X.25 Network Control Program
Packet Switching Interface

# Host Programming

*Version 3*

SC30-3502-08

X.25 Network Control Program
Packet Switching Interface

# Host Programming

*Version 3*

SC30-3502-08

**Note**

Before using this document, read the general information under “Notices” on page vii.

## Ninth Edition, October 1998

This is a revision of, and obsoletes, SC30-3502-07.

This edition applies to:

| Version 3 of the IBM licensed program X.25 Network Control Program Packet Switching Interface (program number 5688-035)

and to all subsequent releases and modifications until otherwise indicated in new editions or technical newsletters. See the “What Is New in This Book” section for the changes made to this manual. Technical changes or additions to the text and illustrations are indicated by a vertical line to the left of the change. Make sure you are using the correct edition for the level of the product.

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

IBM welcomes your comments. A form for readers' comments is provided at the back of this publication. If the form has been removed, you may address your comments to:

Design and Information Development
Dept. CGF/ Bldg. 656
International Business Machines Corporation
P.O. Box 12195
Research Triangle Park, North Carolina 27709-9990

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

# Contents

# 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 IBM's intellectual property rights may be used instead of the IBM product, program, or service. Evaluation and verification of operation in conjunction with other products, except those expressly designated by IBM, are the user's responsibility.

IBM may have patents or pending patent applications covering subject matter 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 Road
Research Triangle Park, NC 27709-2195
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.

## Programming Interface Information

This *NPSI Host Programming* manual is intended to help application programmers in writing programs to interface with IBM’s X.25 Network Control Program Packet Switching Interface (NPSI) licensed program. This publication documents general-use programming interface and associated guidance information provided by NPSI.

General-use programming interfaces allow customers to write programs that obtain the services of NPSI.

## Notice to Users of Online Versions of This Book

For online versions of this book, you are authorized to:

- Copy, modify, and print the documentation contained on the media, for use within your enterprise, provided you reproduce the copyright notice, all warning statements, and other required statements on each copy or partial copy.
- Transfer the original unaltered copy of the documentation when you transfer the related IBM product (which may be either machines you own, or programs, if the program's license terms permit a transfer). You must, at the same time, destroy all other copies of the documentation.

You are responsible for payment of any taxes, including personal property taxes, resulting from this authorization.

THERE ARE NO WARRANTIES, EXPRESS OR IMPLIED, INCLUDING THE WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

Some jurisdictions do not allow the exclusion of implied warranties, so the above exclusion may not apply to you.

Your failure to comply with the terms above terminates this authorization. Upon termination, you must destroy your machine-readable documentation.

## Trademarks

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

ACF/VTAM
Advanced Peer-to-Peer Networking
AIX
AnyNet
APPN
AS/400
BookManager
CICS
Common User Access
CUA
IBM
IMS
Library Reader
MVS/ESA
MVS/XA
NetFinity
NTune
NTuneMON
NTuneNCP
NetView
OS/2
OS/390
OS/400
SystemView
System/370
Virtual Machine/Enterprise System Architecture
VM/ESA
VSE/ESA
VTAM

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

*X.25 NPSI Host Programming* assists application programmers in writing programs to interface with the IBM X.25 Network Control Program Packet Switching Interface (X.25 NPSI) licensed program. NPSI offers Systems Network Architecture (SNA) users the ability to use communication facilities that support the X.25 interface as defined by the International Telegraph and Telephone Consultative Committee (CCITT), also known as the International Telecommunication Union-Telecommunication Standardization Sector (ITU-TS), Geneva in 1976, Malaga-Torremolinos in 1984, and Melbourne in 1988.

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

## Who Should Use This Book

*X.25 NPSI Host Programming* is intended for the following audiences:

- Application programmers who are responsible for designing, coding, compiling, executing, testing, and debugging application programs that work with NPSI connections.
- System programmers who are customizing subsystems that support application programs. These application programs might include Time Sharing Option (TSO), Customer Information Control System (CICS), and Information Management System (IMS).
- Programmers who write communication and transmission control programs (CTCPs) to interface with the:
  - General access to X.25 transport extension (GATE) function
  - Dedicated access to X.25 transport extension (DATE) function

## How to Use This Book

Use this book to help when designing application programs, customizing subsystems, and writing CTCPs to interface with NPSI.

### How CSS, 37CS, and 3746 Model 900 Are Used

The terms *connectivity subsystem (CSS)* and *37CS* 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.

### Symbols Used in This Book

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

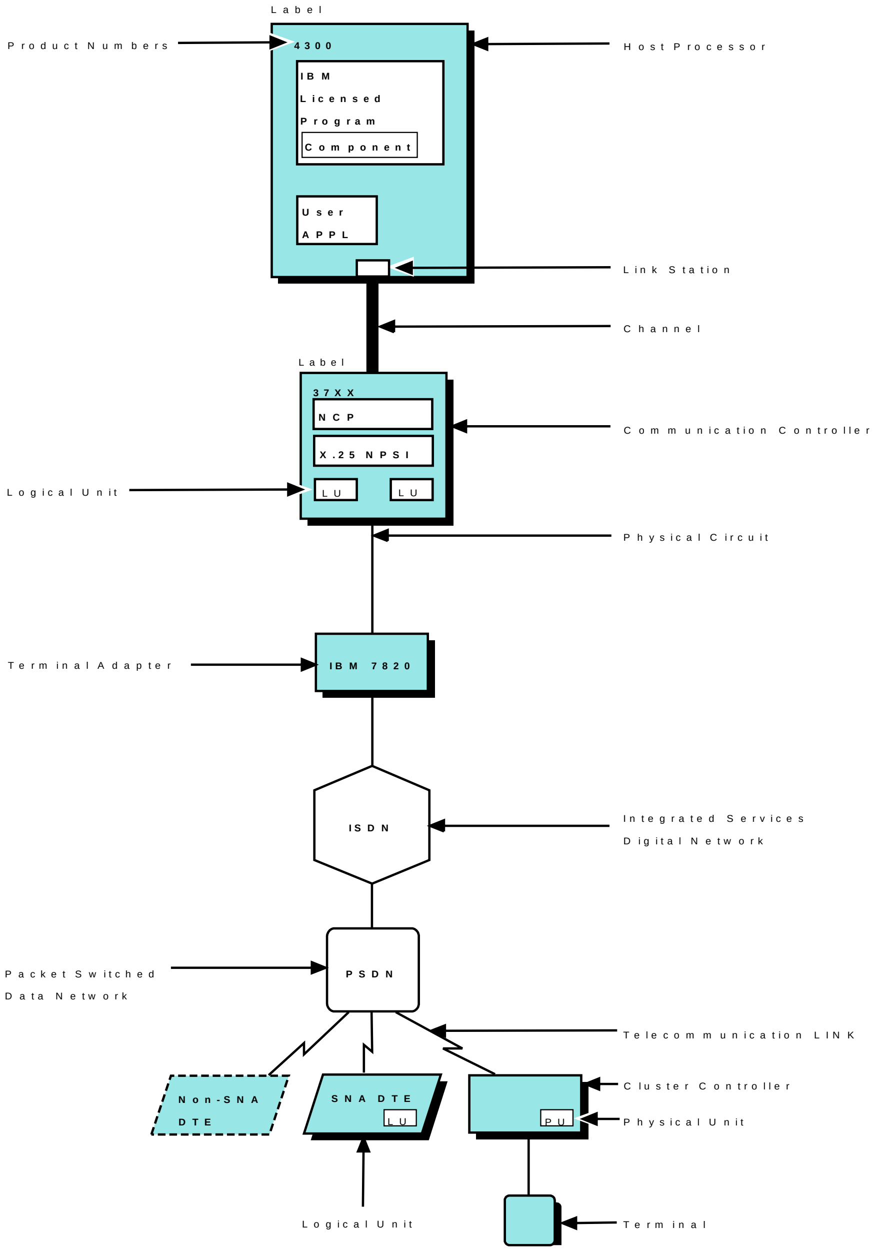


*Figure 0-1. Networking Symbols Used in This Book*

## What Is New in This Book

This edition contains editorial and minor technical changes. New or changed information is identified by a vertical bar (|) in the left margin.

## Where to Find More Information

For more information on NPSI, see the hardcopy and softcopy library information described in the following sections.

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

### NPSI Hardcopy Library

The NPSI hardcopy library includes:

- *X.25 NPSI Data Areas*
- *X.25 NPSI Diagnosis, Customization, and Tuning*
- *X.25 NPSI General Information*
- *X.25 NPSI Planning and Installation*

### NPSI Softcopy Library

Those publications available as softcopy books have cross-document search and hypertext links for speedy, online information retrieval. These softcopy books are grouped together on an electronic bookshelf 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 information on CD-ROMs and softcopy books, see *IBM Online Libraries: Softcopy Collection 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. Methods of Communication Using NPSI

# Chapter 1. Methods of Communication Using NPSI

NPSI is an IBM licensed program that provides Virtual Telecommunications Access Method (VTAM) users with an X.25 interface to connect remote data terminal equipment (DTEs) with SNA and non-SNA networks. This chapter provides you with information to design or modify a host application program to allow communication with remote DTEs through NPSI.

NPSI makes a virtual circuit appear to the access method and Network Control Program (NCP) as a Synchronous Data Link Control (SDLC) link with an associated physical unit (PU) and logical unit (LU). Three types of SNA sessions result when the virtual circuit is active: a system services control point (SSCP)-PU session, an SSCP-LU session, and an LU-LU session.

- An SSCP-PU session is established with each multichannel link physical unit (MCH PU) and each virtual circuit PU. The SSCP-PU session allows the SSCP to monitor and change the status of the circuits by sending SNA requests to the PU. These commands are processed by the NCP and NPSI.
- An SSCP-LU session is associated with each multichannel link logical unit (MCH LU) and virtual circuit LU. The SSCP-LU session allows the SSCP to transfer commands to and from the LUs.
- An LU-LU session occurs between an LU associated with the host application program and the virtual circuit LU. This LU-LU session allows the application program to transfer messages to and from the virtual circuit LUs. The MCH-LU-to-application-LU session exists only if the application LU is in a CTCP.
- The general access to the X.25 transport extension (GATE) function of NPSI allows a CTCP to monitor virtual circuits to non-SNA X.25 DTEs.
- The dedicated access to the X.25 transport extension (DATE) function of NPSI allows a CTCP to manage virtual circuits to SNA and non-SNA X.25 DTEs.

## Communicating with an SNA DTE

NPSI supports connection to SNA devices, including the following session types:

- Type 0
- Type 1
- Type 2
- Type 6.1
- Type 6.2

There are no special considerations for the host application program when it communicates with an SNA device. However, the host application program must ensure that the remote device has the correct device specifications. If the application uses CICS or IMS, entries for the correct specifications must be added to the table of devices.

NPSI must have the virtual circuit defined as defining logical link control (LLC) type 2 or type 3. The type you use depends on the network path configuration. If you are using a Network Interface Adapter (NIA) or a physical services header (PSH) interface at the remote DTE, use LLC type 2; otherwise, use LLC type 3.

## Switched Virtual Circuit Subarea Communication (SVCSC)

SVCSC allows two subarea nodes to be connected by a switched virtual circuit (SVC). The connection is used primarily for the following reasons:

- Connecting two subarea nodes in different packet switched data networks (PSDNs)
- Adding transfer capability through new virtual routes that map to virtual circuits of different MCHs

### NPSI Considerations

NPSI supports SVCSC. With SVCSC, communication is possible with another communication controller running NPSI, or with a host computer that has the X.25 telecommunication subsystem controller (TSC).

SVCSC supports SNA traffic between subareas. The NCPs supporting these subareas can provide SNA network interconnection (SNI) functions. SVCSC provides connectivity between any combination of 3720, 3745, and 3746 Model 900 Communication Controllers and the X.25 TSC. The subarea node qualified logical link control (QLLC) is used by SVCSC.

Both the access method and the NCP are aware that an SVC virtual circuit is used. All connection parameters, such as dial number, VCCPT index, and OUFT index, are defined to the access method. A VTAM operator or an automated operator facility, such as a NetView program command list, must initiate the dial connection.

SVCs used for SVCSC communication do not have to be dedicated to this function.

### Short Hold Mode (SHM) Feature

SHM, managed by NPSI, can provide a significant savings in network fees. When traffic is absent from an X.25 SVC for an amount of time specified by the user, and if SHM is supported by both the calling and called sides, NPSI breaks the virtual connection without informing the NCP or VTAM. From an SNA point of view, the connection is unchanged, and a logical connection continues to be in place. When traffic resumes from either side, NPSI automatically reestablishes the connection.

## SNA Type 2.1 Node Support

NPSI, in conjunction with VTAM V3R2 or later releases, allows LUs that are connected to an SNA type 2.1 peripheral node to communicate through a PSDN.

Two peer systems can communicate using the SNA backbone network without requiring the participation of a host application program for relay purposes. NPSI also allows SNA LU type 6.2 multiple session communication for the LUs that reside in the SNA type 2.1 node.

## Enhanced SNA Type 2.1 Boundary Function Support (Casual Connection)

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

The enhanced SNA type 2.1 boundary function support is also called a casual connection. Casual connection allows a VTAM, NCP, and NPSI configuration, working as a type 2.1 node, to communicate with another type 2.1 node. This adjacent type 2.1 node can be another VTAM, NCP, and NPSI configuration, or a native type 2.1 node. With casual connection, you do not have to predefine the identity of the adjacent link station.

## Communicating with a Non-SNA DTE

Communication between a non-SNA DTE and a host application program is *always* performed through the NPSI LU simulator (LUSIM). LUSIM is used for LLC types 0, 4, and 5, and for CTCP communication with the MCH LU. LUSIM allows communication with applications that support SNA LU type 1 devices, including CICS, IMS, and TSO. These applications are described in Chapter 5, “Programming Host Applications.”

## LUSIM

The LUSIM converts the SNA transmission header (TH) and request/response header (RH) into appropriate X.25 headers when SNA data is sent to NPSI for transmission to a non-SNA device. NPSI then performs any other processing necessary for transmission of the message to the PSDN.

Communication from a non-SNA DTE to an SNA host application program flows in the reverse direction. After NPSI satisfies all X.25 processing considerations, the LUSIM then converts the X.25 headers into appropriate SNA headers and forwards the message to the destination application.

The NPSI LUSIM operates as an SNA secondary logical unit (SLU) type 1 in a half-duplex mode using either contention or flip-flop protocol. Only one LU-LU session is allowed on a virtual circuit to a non-SNA DTE at any given time.

For a description of the SNA functions and concepts involved in a type 1 LU-LU session, see the following publications:

- *SNA Concepts and Products*
- *SNA Technical Overview*

### Bracket Protocol

Bracket protocol is initiated by an issuer placing the begin bracket indicator in the SNA header. An end bracket is used to end the exchange with the LU.

Most exchanges between LUs are organized into units of data for transmission or reception. The LUs ensure that the data units are not broken up during the exchange.

When neither LU is engaged in an exchange of data, a state known as *between bracket* exists. An SLU, as defined in the BIND, can bid for initiation of a bracket. The partner LU sends a positive or negative response. If a positive response is received, the bidding LU can begin the bracket. The partner can choose to reject the bid if, for instance, the partner has already initiated a bracket. LUSIM sends an `X'0813'` negative response to reject the bid request if the bid request cannot be accepted.

Bracket protocol works the same with the LUSIM as it does with all LU type 1 devices.

## Response Request Types

SNA defines two different types of response requests: *definite* and *exception*.

***Definite*** response requests are used to ensure that a message has arrived at its destination. When this type of message is received by the LUSIM, a response is immediately returned. Unless the delivery confirmation bit (D bit) is used, the LUSIM does not wait for the data to be sent out or for the data to arrive at the destination before responding to the host.

***Exception*** response requests are used when the arrival of the message is not critical or when there is a high degree of traffic on the communication path. In such cases, the sender is notified, by means of a negative response, only when there is an error.

## Half-Duplex Flip-Flop Session

In a half-duplex flip-flop session, the LU that is the first speaker to send is the owner of the change of direction indicator. The change of direction owner continues sending requests until it reaches the end of the data to be sent, or until the receiver of the data issues a SIGNAL command. In either case, after the sender (change of direction owner) sends the last request of the last chain, the sender turns on the change of direction indicator to give control of the session to the other session partner.

An LU that is awaiting a change of direction is prohibited from sending data. It is free to send response and expedited-flow requests, such as SIGNAL.

If a receiving LU is to initiate data transmission, it must obtain the CD indicator. It can notify the sender by issuing the SIGNAL command. The sender (CD owner) can then either relinquish the CD indicator or refuse the request.

The LUSIM always sends a CD indicator in host-bound path information units (PIUs), which allows the host to respond immediately.

## Half-Duplex Contention Session

In a half-duplex contention session, either LU can initiate data transmission. If both LUs send at the same time, the *contention winner* is the winner defined in the BIND image at the session's initiation. The *contention loser*, as defined in the BIND, receives a negative response with a sense code of `X'081B'`.

With NPSI, this negative response occurs only with definite response request type. In the case of exception response, data from either side is processed immediately or queued in NPSI.

## Session-Level Pacing

For session-level pacing, LUSIM supports only send pacing. The simulator sends a pacing response each time it receives a request with the pacing bit set to 1 in the request/response header (RH). For an LU associated with a virtual circuit or a physical circuit, the pacing parameter must be 1.

Although the pacing information placed by the LUSIM in a response does not cause the release of the buffers containing the corresponding request/response unit (RU), session-level pacing allows the data flow between the host and the NCP to be regulated. An request/response unit (RU) from the host entering the NCP is processed by the LUSIM only when the packet window is open: that is, when the previous RUs in the window have been sent to the network.

## Non-Support of SNA Segmenting

NPSI does not support SNA segmenting with the LUSIM. If the value specified for MAXDATA is not large enough, NPSI returns a negative response with a sense code of X'08F5' to the host.

## Support of RU Chaining

NPSI supports RU chaining for long non-SNA messages on both inbound and outbound flows. This optional support is selected at SYSGEN by coding MBITCHN=YES on the X25.MCH statement. If the RU chaining option is not selected, NPSI:

- Considers outbound chained PIUs as only-in-chain PIUs and maps each of them into an X.25 packet sequence
- Assembles X.25 packet sequences into only-in-chain PIUs

The RU chaining support is fully available for virtual circuits working with the protocol converter for non-SNA equipment (PCNE) or integrated packet assembler/disassembler (PAD), and it is partially available for virtual circuits working with GATE or transparent PAD. See “RU Chaining for Virtual Circuits Working in GATE and Transparent PAD” on page 1-6 for more information about GATE and transparent PAD processing.

## RU Chaining for Virtual Circuits Working in PCNE or Integrated PAD

On inbound flows, NPSI converts each long non-SNA message into an SNA RU chain. The conversion produces a first-in-chain (FIC) PIU, middle-in-chain (MIC) PIU, or last-in-chain (LIC) PIU. On outbound flows, each SNA chain is converted into a packet sequence. NPSI can create two types of packet sequences:

- Complete packet sequence (CPS)

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

- M-bit sequence

  An M-bit sequence contains a CPS series. Each last packet of a CPS within the series has both the M bit and D bit set to 1, except for the last packet of the last CPS, which has the M bit set to 0.

On inbound flows, NPSI provides optional support for RU chaining through the MBITCHN keyword on the X25.MCH statement. When a CPS is received, if MBITCHN=YES and DBIT=NO is specified, NPSI accumulates the data packets and builds an RU chain (FIC, MIC, or LIC). The size of the PIU in the chain is determined by either the value of the MAXRU size on the BIND command, or the value of the X25.MAXPIU keyword on the BUILD statement. When MBITCHN=YES and DBIT=YES are specified, NPSI accumulates CPS data packets and builds an RU. The length of the RU is determined by the length of the CPS series. When the M-bit sequence is received (CPS series), RUs are chained as FIC, MIC, or LIC, depending on the length of the CPS series.

**Note:** If the last packet of an M-bit sequence has the D bit set to 1, NPSI sends an Receive Ready (RR) packet for end-to-end acknowledgment.

If the application sends a negative response to NPSI, NPSI builds a CANCEL CHAIN command and sends the DTE a Reset Request packet containing a diagnostic code of X'B3'.

On outbound flows, if MBITCHN=YES and DBIT=YES is specified, each RU chain is converted into a CPS series that is linked together using the D bit. The last packet in the last CPS has the M bit set to 0. This series is sent to the non-SNA DTE as an M-bit sequence. If MBITCHN=YES is specified and DBIT=NO, each RU chain is converted into a single CPS. The CPS is then sent to the non-SNA DTE.

On outbound flows, NPSI processes the CANCEL CHAIN command by sending to the DTE a Reset Request packet containing a diagnostic code of X'B3'.

For both inbound and outbound flows, when MBITCHN=YES is specified, and a Reset Indication packet is received during the process of an M-bit sequence, NPSI builds a CANCEL CHAIN command RU and sends it to the host.

**Note:** If the LU-simulator LU-application supports a definite response, NPSI sets the D bit to 1 in the last packet of an M-bit sequence.

## RU Chaining for Virtual Circuits Working in GATE and Transparent PAD

When the RU chaining function is selected (MBITCHN=YES), only inbound flow is different from the existing process. In this case, NPSI accumulates the data packets of a CPS, up to the value of either the MAXRU size in the BIND, or the X25.MAXPIU on the BUILD NDF statement. When this value is reached, NPSI builds and sends an only-in-chain (OIC) PIU. Consequently, a single CPS can result in a series of OIC PIUs.

Figure 1-1 on page 1-7 shows the data flow for a network configuration that uses GATE.

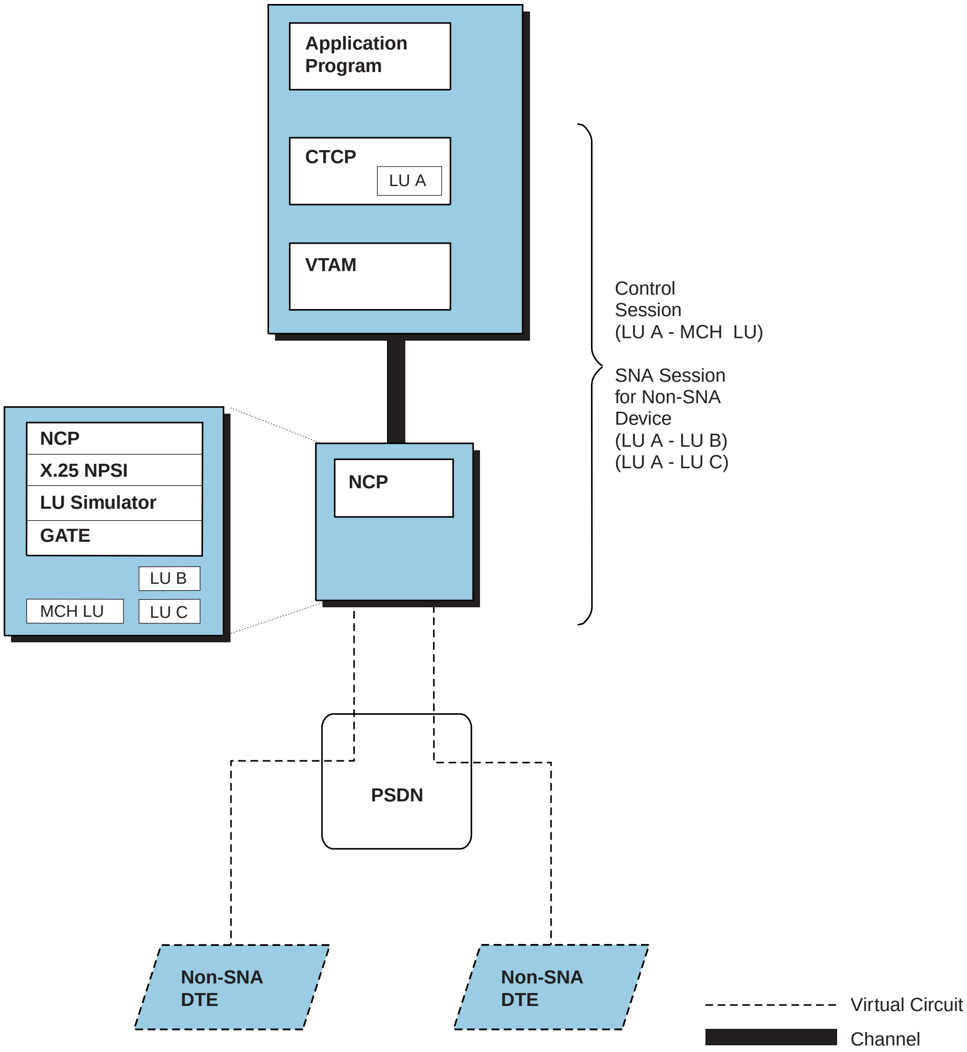


*Figure 1-1. SNA Host Node to Non-SNA DTE (GATE)*

## Actions on Control and Data RUs

NPSI does not perform any processing on the SNA data RU *except* when you use the optional character code translation between ASCII and EBCDIC, and when you use integrated PAD support with password protection. The two types of bidirectional control RUs that flow between the LU and the LUSIM are session control RUs and data flow RUs.

- Session control RUs manage:
  - LU activation and deactivation
  - Session initiation and disconnection
  - Data traffic initiation
  - LU status
- Data flow RUs manage:
  - Directions of the data on the LU-LU session
  - Traffic assessment between LUs

Some of these RUs are partial implementations of SNA. The extent of implementation for each supported RU type is specified in “Session Control RUs” and “Data Flow Control RUs.”

***Session Control RUs:*** The LUSIM recognizes and processes the following session-control RUs:

**ACTLU** An ACTIVATE LOGICAL UNIT command is received from the SSCP. The LUSIM returns a positive response to the SSCP checking for the activation type. If the session is not bound, the ACTLU is for an initial activation and is followed by a BIND. If the session is already bound, the ACTLU is for a session continuation case and is not followed by a BIND. The ACTLU establishes a connection between VTAM (SSCP) and the target LU. Logon requests can then be transmitted to the host.

**BIND** A BIND SESSION command is received from the host LU. The LUSIM validates the bind image and completes the session connection. BIND establishes a connection between the host application program and the virtual circuit LU. This is a non-negotiable BIND. The validity of the BIND parameters are checked against the values listed in Table 1-1 on page 1-11. If BIND is accepted, the LUSIM sends a positive response to the primary LU (PLU).

**SDT** A START DATA TRAFFIC command is received from the host LU. The LUSIM becomes eligible to send and receive data and returns a positive response to the application LU.

**CLEAR** A CLEAR command is received from the host LU. The LUSIM clears all the local pending conditions and returns a positive response to the LU. The status of the LU, as managed by the LUSIM, is reinitialized so that further processing can continue.

**UNBIND** An UNBIND SESSION command is received from the host LU. The LUSIM returns a positive response to the host and releases the remote DTE from the session. LOGON requests for a new session can then be transmitted to the host.

**DACTLU** A DEACTIVATE LOGICAL UNIT command is received from the SSCP. The LUSIM returns a positive response to the SSCP and breaks the connection between the virtual circuit LU and VTAM.

***Data Flow Control RUs:*** The LUSIM accepts and processes the following data-flow control RUs:

**CANCEL** A CANCEL command is received from the host LU. The LUSIM returns a positive response and takes no other action. Because the chaining concept is local to the LUSIM, the host application program needs a higher-level end-to-end protocol to cancel a chain of RUs being sent to a remote non-SNA DTE.

The LUSIM never sends a CANCEL command to the host.

When MBITCHN=YES, if a CANCEL command is received from the host LU, the LUSIM returns a positive response and resets the virtual circuit with a diagnostic code of X'B3'. On receipt of a RESET INDICATION, the LUSIM sends a CANCEL command to the host.

**SIGNAL** A SIGNAL command is received from the host LU. The LUSIM returns a positive response and takes no other action unless NPSI integrated PAD support is used. The SLU never generates a SIGNAL command unless the integrated PAD function is used to support the remote DTE. When the integrated PAD function receives a SIGNAL from the host, the integrated PAD sends an INDICATION OF BREAK message to the remote PAD. On receipt of an Interrupt, Reset, or Qualified packet from the remote PAD, the integrated PAD interface sends a SIGNAL request to the host (depending on the setting of PAD parameter 7).

**BID** A BID command is received from the host LU. The LUSIM enters begin bracket pending status if its status is such that it can accept the request and return a positive response. An exception response is returned if the LUSIM cannot accept the bid. All data packets received from the network are buffered until an RU is received specifying begin bracket.

**CHASE** A CHASE command is received from the host LU. The LUSIM returns a positive response and takes no other action.

**SHUTD** A SHUTDOWN command is received from the host LU. The LUSIM returns SHUTDOWN COMPLETE (SHUTC) as a response as soon as the LU enters the between bracket state. Any additional incoming data packets are discarded. If the LUSIM can close the bracket, the SHUTC command is returned after receiving the first packet with the more data bit (M bit) set to 0 (end of packet sequence). With integrated PAD, a SHUTDOWN can be optionally converted to an INVITATION TO CLEAR message for an SVC, or a Reset packet for a PVC, and sent to the remote PAD.

## Session Initiation

A host application program establishes communication with an NPSI remote DTE in the same way it does with any LU type 1 device. The session initiation requests used to establish this communication can originate from any one of the following sources:

- Remote users

  Remote users can initiate a session from their devices by entering a LOGON command.

  The LOGON command is translated into a formatted initiate (INIT) request by the unformatted system services (USS) table in use at the host. The application program need not have previous knowledge of the LUs existence. However, if your network uses non-EBCDIC devices, the USS table must be coded to recognize the characters used by these devices.

- Network operators

  Network operators can establish a session by issuing:

  – The V NET,LOGON=*application* command

  – The V NET,ACT command when LOGAPPL=*application* is coded on the LU statement

  When either of these commands is issued, a session is initiated between the controlling application (specified by *application*) and the LUSIM by the application issuing the OPNDST macro in its logon exit.

- Application programs

Application programs can initiate a session by issuing:

- The OPNDST OPTCD=ACQUIRE macroinstruction
- The SIMLOGON and OPNDST OPTCD=ACCEPT macroinstructions

In the second case, a logon exit must be coded as well. The application program must have previous knowledge of the LUs existence to use these instructions.

- Network definition

  The remote user can be connected directly to a specific application if you code both of the following:

  - The LOGAPPL=*application* keyword on the LU macro
  - The OPNDST OPTCD=ACCEPT macro in the logon exit of the CTCP or application program

  In the LOGAPPL keyword, *application* is the name of the application program to which the remote user is connected.

- NetView command list

  If the processing initiated by the entry of a V NET, LOGON command, or the activation of a resource with the LOGAPPL keyword specified does not result in a logon, the request is not queued and the request is not reattempted. In this instance, you could write a command list to periodically check if the LU is in session, and, if not, the command list could issue a V NET,LOGON=appl name.

The request parameter list (RPL) and the node initialization block (NIB) are additional control blocks used for session establishment. The RPL contains information that describes the session and how to establish it. The NIB contains additional information about the session. The NIB must specify the symbolic name of the LU (NIBSYM).

You also have the option of specifying the BIND image in the NIBNDAR field of the NIB. The BIND image contains the session parameters that establish the communication rules to be followed for session establishment. The session parameters enable each end of the session to know what the other end of the session will or will not do in different communication situations. If you do not specify a BIND image, the VTAM default is used.

If a mode name is required, a corresponding mode table must be defined to VTAM. Figure 1-2 shows a typical mode table entry that can be used with the NPSI LUSIM.

```
name   MODEENT FMPROF=X'03',                       X
               TSPROF=X'03',                       X
               PRIPROT=X'B1',                      X
               SECPROT=X'B0',                      X
               COMPROT=X'3040',                    X
               PSNDPAC=1
```

*Figure 1-2. Typical Mode Table Entry for Use with NPSI LUSIM*

***where:***

`name` is the mode name used in the NIB definition.

Table 1-1 represents a BIND image required to establish a session.

See *VTAM Customization* for information about coding and installing a mode table and *VTAM Programming* for more information about controlling session initiation through a CTCP.

*Table 1-1. Bind Parameter List*

| Byte | Bits | Value | Meaning |
|---|---|---|---|
| 0 | 0–7 | X'31' | BIND code |
| 1 | 0–3 | B'0000' | BIND format = 0 |
|  | 4–7 | B'0001' | BIND type = 1 |
| 2 | 0–7 | X'03' | FM profile = 3 |
| 3 | 0–7 | X'03' | TS profile = 3 |
|  |  |  | **Definition of Primary Protocol**: |
| 4 | 0 | B'1' | Multiple request chains |
|  | 1 | B'0' | Required value |
|  | 2–3 | B'01' | Exception response |
|  |  | B'10' | Definite response |
|  |  | B'11' | Exception or definite response |
|  | 4–5 |  | Not used |
|  | 6 | B'0' | Compression not used |
|  | 7 | B'1' | PLU can send EB |
|  |  | B'0' | PLU does not send EB |
|  |  |  | **Definition of Secondary Protocol**: |
| 5 | 0 | B'1' | Multiple request chains (the SLU always sends OIC to the host) |
|  | 1 | B'0' | Required value |
|  | 2–3 | B'01' | Exception response |
|  |  | B'10' | Definite response |
|  |  | B'11' | Exception or definite response |
|  | 4–5 |  | not used |
|  | 6 | B'0' | Compression not used |
|  | 7 | B'1' | Secondary can send EB (Not allowed with flip-flop) |
|  |  | B'0' | Secondary does not send EB |
|  |  |  | **Common Protocol**: |
| 6 | 0 |  | Not used |
|  | 1 | B'0' | FM header not used |
|  | 2 | B'1' | Bracket used |
|  |  | B'0' | Bracket not used |
|  | 3 | B'1' | Bracket termination rule 1 |
|  | 4 | B'0' | EBCDIC |
|  | 5–7 |  | Not used |
|  |  |  | **Common Protocol**: |
| 7 | 0–1 | B'01' | Half-duplex contention |
|  |  | B'10' | Half-duplex flip-flop |
|  | 2 | B'0' | Primary LU recovery responsibility |
|  | 3 | B'0' | First speaker is secondary |
|  | 4–7 |  | Not used |
| 8 | 0–7 |  | Not used |
| 9 | 0–1 |  | Not used |
|  | 2–7 | B'000001' | Pacing |

## Session Termination

Termination of a session between a host application program and an NPSI resource occurs in much the same way as it does for any other LU type 1 device. Session termination requests can come from any of the following sources:

- Remote users

  Remote users can terminate sessions from their devices by:

  – Entering the LOGOFF command
  – Pressing a key that causes the X.25 connection to be terminated

- Network operators

  Network operators can terminate sessions by issuing one of the following commands:

  – V NET,INACT,ID=*luname*

    When this command is issued, the LU is deactivated.

  – V NET,TERM,ID=*luname*

    When this command is issued, a termination request is sent to the SSCP.

- Application programs

  Application programs can terminate sessions normally by issuing the CLSDST macroinstruction.

- Network failures

  Network failures can originate from many sources, such as the failure of a hardware or software component. In most cases, the session is abnormally terminated.

See Chapter 2, "Programming Using PCNE and PAD Support," Chapter 3, "Programming a GATE CTCP," and Chapter 4, "Programming a DATE CTCP," for more information about how the CTCP controls session termination.

For more specific information about controlling session termination through a CTCP, see *VTAM Programming*.

## Session Continuation

NPSI, in conjunction with VTAM V3R2 and NCP V5R2 and later releases, provides optional session continuation capabilities, including takeover and return of ownership.

Ownership of resources by an SSCP is determined by either initial activation or acquisition.

Session continuation provides communication with no disruptions for active LU-LU sessions, if the owning SSCP fails or connectivity between the LUSIM and the owning SSCP is lost. The LINE, PU, and LU ownership can be taken over by another SSCP, and given back to the original SSCP when it becomes active again. With NPSI, the NCP allows a switched line, PU, and LU to remain active. Therefore, when the session partners are still accessible, the active LU-LU sessions belonging to that PU are not disturbed. Session continuation operates in both SNA and non-SNA environments, and applies to PVC and SVC.

## Using GATE and DATE for Communicating

When using GATE, with or without fast connect, FTPI, or DATE, use a CTCP as an interface between host application programs and NPSI. The CTCP must communicate with the application program it controls. In particular, the CTCP and the application program must be able to communicate to accomplish the following:

- Perform a Call Request
- Handle SIGNAL outbound
- Handle SIGNAL inbound (caused by BREAK at a remote DTE)

The CTCP must be able to process all valid control and qualified packets for the virtual circuits. A list of the control packets for GATE can be found in Table 3-1 on page 3-7. The control packets for FTPI can be found in Table 3-5 on page 3-52. The control packets for DATE are listed in Table 4-1 on page 4-11 and Table 4-2 on page 4-12. The CTCP is able to specify any valid options within these packets.

An example of these options is the CALL REQUEST command from the CTCP to NPSI. In a message to DATE or GATE, the CTCP can specify *any* combination of facilities that are acceptable to the PSDN within the FACILITIES field of the Call Request packet or the Call-Accepted packet.

It is *not* the responsibility of the CTCP to manage the packet level processor (PLP) counters, the packet-level modulo, or the M bit. They are managed by NPSI. When communicating through a CTCP, the enhancements for ISO are not implemented by NPSI. It is the responsibility of the host programmer of the CTCP to ensure that the ISO functions are implemented.

## Selecting GATE or DATE

While GATE and DATE both operate in conjunction with the LUSIM, these NPSI functions are designed to satisfy different operating requirements. Table 1-2 on page 1-14 shows the major differences between GATE and DATE processing.

*Table 1-2. GATE and DATE Processing Differences*

| GATE | DATE |
|---|---|
| The logon is always built by NPSI. | Building of the logon by NPSI is optional when using DATE. The logon can be built by NPSI if this service is requested by the CTCP in the CALL REQUEST or CALL CONFIRMATION command. |
| GATE always uses LLC type 4. | For DATE, the LLC type is determined by the CTCP, including SNA sessions to SNA destinations. See the list of virtual circuits types and corresponding hexadecimal codes in “Determination of Virtual Circuit Type” on page 4-3. |
| RU byte 0 contains control information for qualified, control, and data packets. | RU byte 0 contains no control information when DATE is communicating with the VC LU, except when transparent PAD is used. RU byte 3 contains the packet type for PIUs passed to or from the MCH LU. |
| The CTCP is in session with the MCH LU and the VC LU. | The CTCP is in session only with the MCH LU. The application program is in session with the VC LU. |
| After CLEAR exchange, the CTCP must issue the CLSDST macro. | After CLEAR exchange, NPSI requests that the NCP send an INOP message (INOP RU) to VTAM. If the application issues the CLSDST macro, NPSI sends an information message to the CTCP, and the CTCP must initiate a CLEAR exchange. |
| NPSI manages the packet level processor timers. | The CTCP manages the packet level processor timers. |
| The CTCP acts as a relay program for session data. This GATE function provides better transmission synchronization than is possible with DATE. | The CTCP does not act as a relay program for session data, because the data is transmitted directly to the application program. Because the CTCP does not process session data, DATE can be used to achieve better host performance than is possible with GATE. |
| The CTCP cannot send the CLEAR CONFIRMATION command or the RESET CONFIRMATION command. | The CTCP must send the CLEAR CONFIRMATION and RESET CONFIRMATION commands. |

## Application Programming Requirements

When using either GATE or DATE, you must satisfy the following requirements for an interface between the CTCP and a host application program:

- virtual circuit establishment
- virtual circuit termination

Additional requirements specific to GATE and DATE are described in Chapter 3, “Programming a GATE CTCP,” and in Chapter 4, “Programming a DATE CTCP.”

## Virtual Circuit Establishment

Virtual circuit management is performed on the CTCP LU to MCH LU session when GATE and DATE are used. When using fast connect, the virtual circuit is established on the CTCP LU to VC LU session. For these reasons, establishment of the control session must be the first step in virtual circuit establishment. The control session can be established using the procedures described in “Session Initiation” on page 1-9.

***Call-In:*** When a call comes in to the application program from a remote DTE, the Incoming Call packet is translated by NPSI into a CALL REQUEST command. This command is then sent to the CTCP.

When GATE=GENERAL is coded on the X25.MCH statement, NPSI determines the LLC type by using CUD0 or the subaddressing digit. In the case of DATE, the CTCP determines the LLC type.

The packet length and window size are determined by the CTCP.

The packet length, window size, and LLC type (for DATE) are communicated back to NPSI through a CALL ACCEPTED command. NPSI then converts this command to a Call Accepted packet and forwards it to the PSDN.

The CTCP is responsible for all required facility negotiations. In particular, the CTCP can negotiate the packet and window sizes with the PSDN. NPSI does not participate in these negotiations.

If the CTCP cannot accommodate the packet and window sizes specified by the PSDN, the CTCP must clear the call. When the negotiations are successful, the CTCP passes the negotiated packet and window sizes to NPSI in the CALL ACCEPTED command.

***Call-Out:*** When the user application or the CTCP operator requests that the CTCP connect with a call-out destination, the operation performed is similar to the one executed for call-in. The CTCP builds a corresponding CALL REQUEST command, including the Call Request packet that is sent to the DCE. The CTCP also builds parameters that include window and packet size.

The CALL REQUEST command is received by the GATE or DATE function in NPSI. NPSI then selects an available virtual circuit, updates the associated control blocks with the provided parameters, and sends the Call Request packet to the PSDN.

If an answer is not received for the Call Request packet within the time limit set for the T21 timer, the CTCP can retry the call by resending the same CALL REQUEST command to NPSI. It is up to the CTCP to decide how many retries must be performed and what must be done if the retries are unsuccessful.

When the remote DTE accepts the Call Request, a Call Connected packet is sent to NPSI. This Call Connected packet is forwarded to the CTCP as a CALL CONNECTED command.

***Logon Processing:*** If NPSI builds the logon, it does so when the call setup is completed. This logon request is issued to the access method after the ACTLU command has been processed for the LU associated with the virtual circuit for the duration of the call. The input queue for that virtual circuit is locked. All incoming data is queued until SDT (start data traffic) is received from the user application. At this point, all data traffic is passed to the application.

**Note:** The INTFAC keyword for the X25.MCH statement gives NPSI the option to participate in packet and window size negotiation. If you specify INTFAC=NO on the X25.MCH statement, NPSI does not participate in packet or window size negotiation. If you specify INTFAC=YES on the X25.MCH statement, NPSI interprets and uses the packet and window sizes that are contained in the Call Connected packet under GATE and DATE.

### Virtual Circuit Termination

A virtual circuit can be deactivated in any of the following ways:

- The remote DTE requests the session termination.

  In this instance, a Clear Request packet is received from the PSDN, and NPSI transfers this information to the CTCP as a CLEAR REQUEST command.

- The user application program can initiate session termination by issuing the CLSDST macroinstruction.

  In this case, NPSI receives an ABCONN request from VTAM.

- The user application program can notify the CTCP to disconnect the virtual circuit.

  When the CTCP receives this request, the CTCP builds a CLEAR command, including the cause and diagnostic bytes as determined by the CTCP. This command is sent to NPSI. NPSI then sends a Clear Request packet to the PSDN. When the Clear Confirmation packet is received, NPSI transfers it to the CTCP in the CLEAR CONFIRMATION command.

- The network operator can terminate the session by issuing a V NET,INACT command for the VC PU or the VC LU.

  When this command is processed, the status of the virtual circuit is set to INACT (inactive). This status change is communicated to the host and the CTCP. The SNA and X.25 resources are disassociated at this point.

For additional information about virtual circuit termination, see one of the following:

- “Virtual Circuit Termination” on page 3-4 (GATE)
- “Virtual Circuit Termination” on page 4-9 (DATE).

# Chapter 2. Programming Using PCNE and PAD Support

# Chapter 2. Programming Using PCNE and PAD Support

This chapter explains how to program virtual circuits (VCs) when you use either of the following:

- The protocol converter for non-SNA equipment (PCNE)
- Integrated and transparent packet assembler/disassembler (PAD) support functions of NPSI.

## PCNE Function of NPSI

The host application programs exchange data with the LUSIM using only LU type 1 protocol in half-duplex contention mode or half-duplex flip-flop mode. The LUSIM provides buffering of inbound data for asynchronous data flow. In the case of exception response mode and contention mode without brackets, buffering of outbound data is also provided by NPSI.

When operating in half-duplex contention mode, the LUSIM immediately sends incoming data to the host, unless it is awaiting a response from the host. If the LUSIM is waiting for a definite response, and the host sends data to the LUSIM instead, NPSI sends an SNA sense code of X'081B' to the host application program.

When operating in half-duplex flip-flop mode, the LUSIM must wait to be in the correct state before transmitting or receiving data with the host. This process is accomplished by the CD bit in the SNA RH. If the host sends data to NPSI and it is not allowed, because NPSI has the CD bit, the data is rejected with a sense code of X'081B'.

**Note:** NPSI supports RU chaining for the PCNE function.

Figure 2-1 on page 2-2 shows the data flow for a network configuration that uses PCNE.

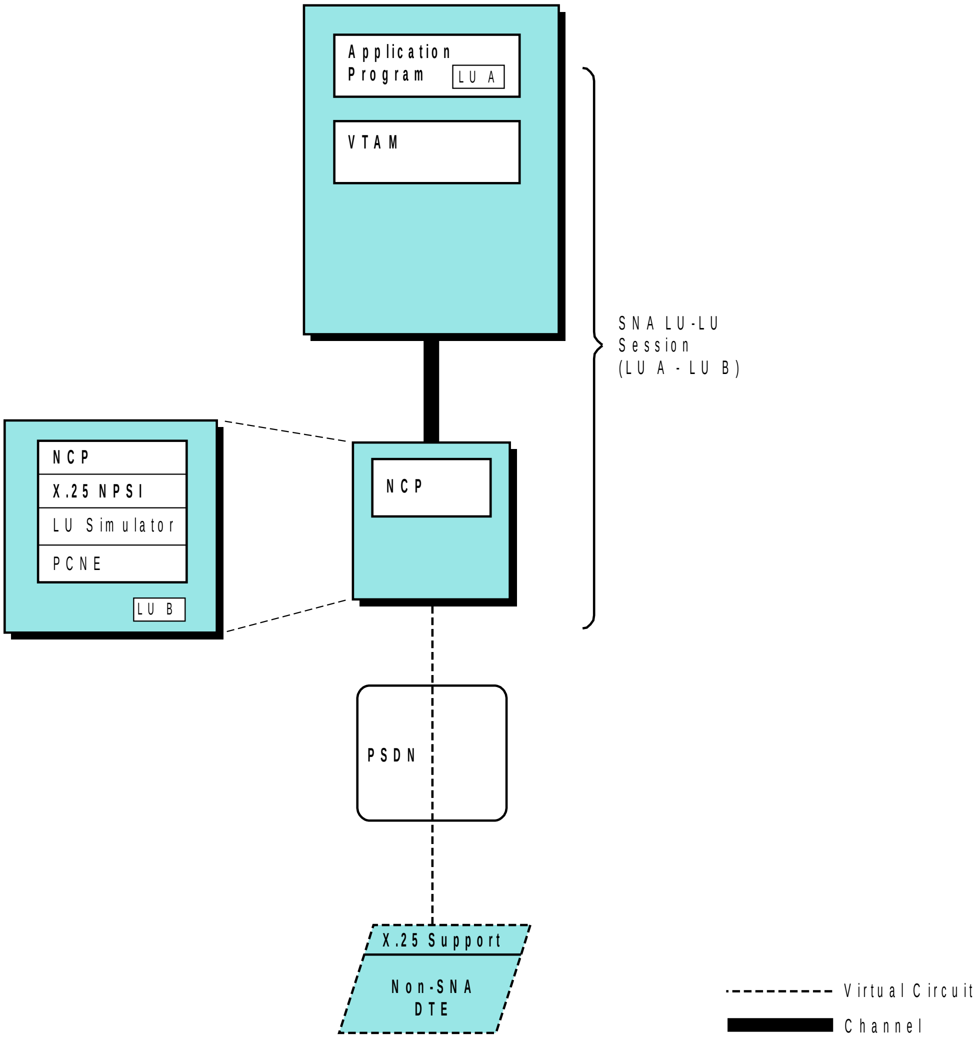


*Figure 2-1. SNA Host Node to Non-SNA DTE (PCNE)*

## Delivery Confirmation Bit

When a type 0 virtual circuit is used, NPSI can support an end-to-end delivery confirmation service to verify that an outbound RU, which has been converted to X.25 data, has reached its destination. The LUSIM does this through a bit in the packet header known as the D bit. NPSI supports RU chaining for non-SNA connections. You can specify whether you wish to use RU chaining by coding the MBITCHN keyword on the X25.MCH statement at system generation. If MBITCHN=YES is specified, RU chaining is supported. The following describes the levels of D-bit support provided.

1. The secondary protocol for the LUSIM supports *only* definite response when the following occur:

   An RU is presented with a definite response requested, the LUSIM converts this request to a complete packet sequence with the D bit of the last packet set to 1. Upon receipt of a Receive Ready (RR) packet, which acknowledges the last packet of the M-bit sequence, the LUSIM constructs a definite response and presents it to the request sender (point of origin).

Conversely, if NPSI receives a packet with the D bit set to 1 and the M bit set to 0, the LUSIM sets the definite response indicator on in the request/response header (RH) of the message sent to the host application program. When the response is returned, NPSI responds to the remote DTE's request for delivery confirmation by sending an RR packet to the network.

2. The secondary protocol for the LUSIM allows *either* definite response or exception response for all releases.

**Notes:**

1. The X.25 D bit indicates delivery to the destination, and whether the delivery was successful. If the application responds negatively to an RU that was received with the D bit set to 1, NPSI resets the virtual circuit with a diagnostic code of X'B3'. The sending of the RESET is dependent upon the setting of the RESETINO keyword on the X25.NET statement.
2. When NPSI issues a Call Request packet with the D bit set to 1, but receives a Call Connected packet with the D bit set to 0, the data packets should have the D bit set to 0.

   However, NPSI still accepts data packets with the D bit set to 1 without resetting the logical channel. This avoids migration problems with DTEs that are not aware of how the D-bit procedure is used during call establishment.

## NPSI Support for RU Chaining

This function allows non-SNA equipment to use the SNA RU chaining function when communicating with an SNA application through NPSI. This function can be used for outbound and inbound flows using a generation parameter.

The chain of PIUs is composed of:

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

**Note:** If MBITCHN=YES, NPSI sends the entire chain of PIUs as either one CPS if DBIT=NO, or a CPS series (an M bit sequence, or MBS) if DBIT=YES. See Chapter 1, "Methods of Communication Using NPSI" for additional information about RU chaining support.

On inbound flow, NPSI maps each CPS on one or more SNA elements of the chain.

Figure 2-2 on page 2-4 and Figure 2-3 on page 2-5 show examples of outbound flow.

Figure 2-4 on page 2-6 and Figure 2-5 on page 2-6 show examples of inbound flow.

## Outbound Flow

Figure 2-2 illustrates the conversion of each chain of RUs into one CPS when the CPS packets are linked together with the M bit.

For this example, assume packet size=64.

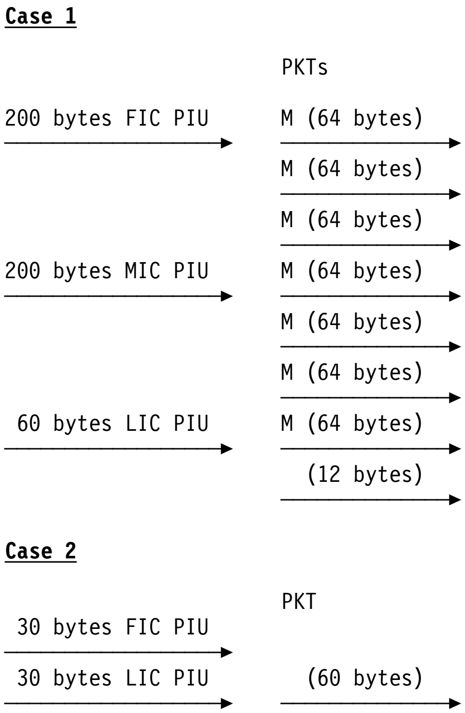


*Figure 2-2. NPSI RU Chaining Outbound Flow (MBITCHN=YES, DBIT=NO)*

**Note:** When MBITCHN=YES and DBIT=NO, each packet sent into the network with the M bit on must be a full data packet. If the last packet from an FIC or MIC RU is not full, NPSI must wait for the next RU to fill the non-full packet with the data from this RU. Figure 2-3 on page 2-5 illustrates the conversion of each RU chain into a CPS chain linked together with the M and D bits, if D bit support is allowed.

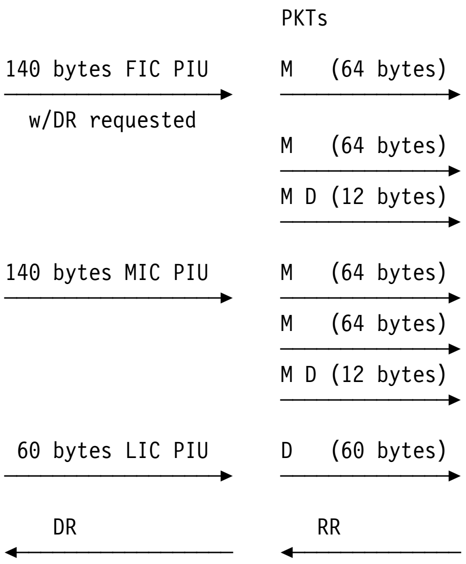


*Figure 2-3. NPSI RU Chaining Outbound Flow (MBITCHN=YES, DBIT=YES)*

**Note:** When MBITCHN=YES and DBIT=YES, each chain element is put into a packet independently. As a result, the last packet from an FIC or MIC RU may not be full, and has both the M bit and the D bit on.

## Inbound Flow

Figure 2-4 on page 2-6 illustrates the conversion of a CPS into an SNA RU chain.

Packets containing the M bit are accumulated until MAXRU of the BIND command is reached. The PIU is then forwarded to the host. Otherwise, the size of the accumulated data needed to build a PIU is determined by the length of the CPS received.

For this example, assume MAXRU=150 and packet size=64.

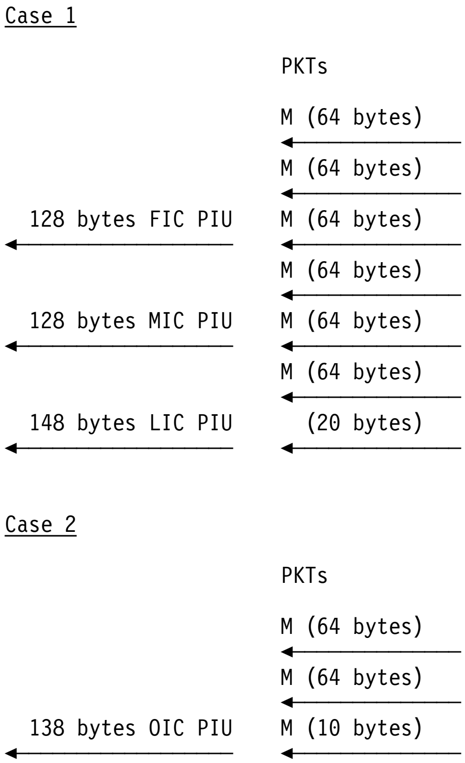


*Figure 2-4. NPSI RU Chaining Inbound Flow (MBITCHN=YES, DBIT=NO)*

Figure 2-5 illustrates the conversion of each CPS chain linked together with the D bit into an RU chain, if D-bit support is allowed. The size of the PIU forwarded to the host is determined by either the length of the CPS received, or the size of the MAXRU of the BIND, whichever is smaller.

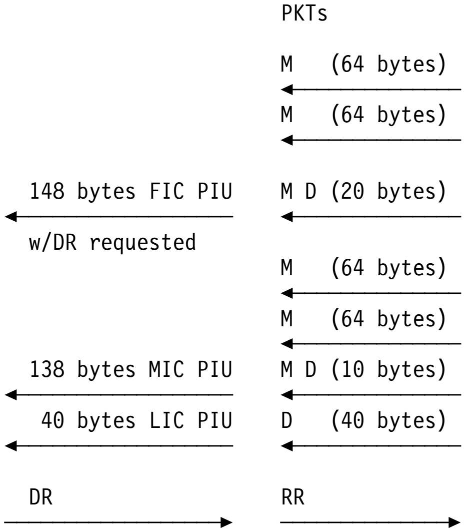


*Figure 2-5. NPSI RU Chaining Inbound Flow (MBITCHN=YES, DBIT=YES)*

**Note:** NPSI can send other RR packets, but this figure shows only those RR packets that are mapped to a definite response PIU.

## Establishing a Session over a PCNE-to-PCNE Connection

In addition to an SNA-to-SNA session link, two application programs can establish a PCNE-to-PCNE session link across a type 0 virtual circuit. This connection can be between any two programs.

The process of session initiation in a PCNE-to-PCNE environment can be the following:

1. APPL 1 initiates a session with the LU of the virtual circuit that communicates with the other host.
2. APPL 1 constructs a LOGON message acceptable to APPL 2, and sends it to APPL 2.
3. APPL 1 and APPL 2 exchange data using either the half-duplex flip-flop protocol or the half-duplex contention protocol.
4. APPL 1 issues a CLSDST to terminate the session.

In a PCNE-to-PCNE session, the initiator of the session can construct a LOGON message to cause the LOGON exit routine to activate in the partner LU. The LOGON message must be formatted as follows:

```
LOGON APPLID(aaaaaaaa) DATA(dddddddd)
```

***where:***

`aaaaaaaa` Is the LU partner name (APPLID).
`dddddddd` Is any appropriate user data that is to be sent to the partner LU.

Figure 2-6 on page 2-8 is an example of a session that can be established. It shows the data flow for a session that uses PCNE.

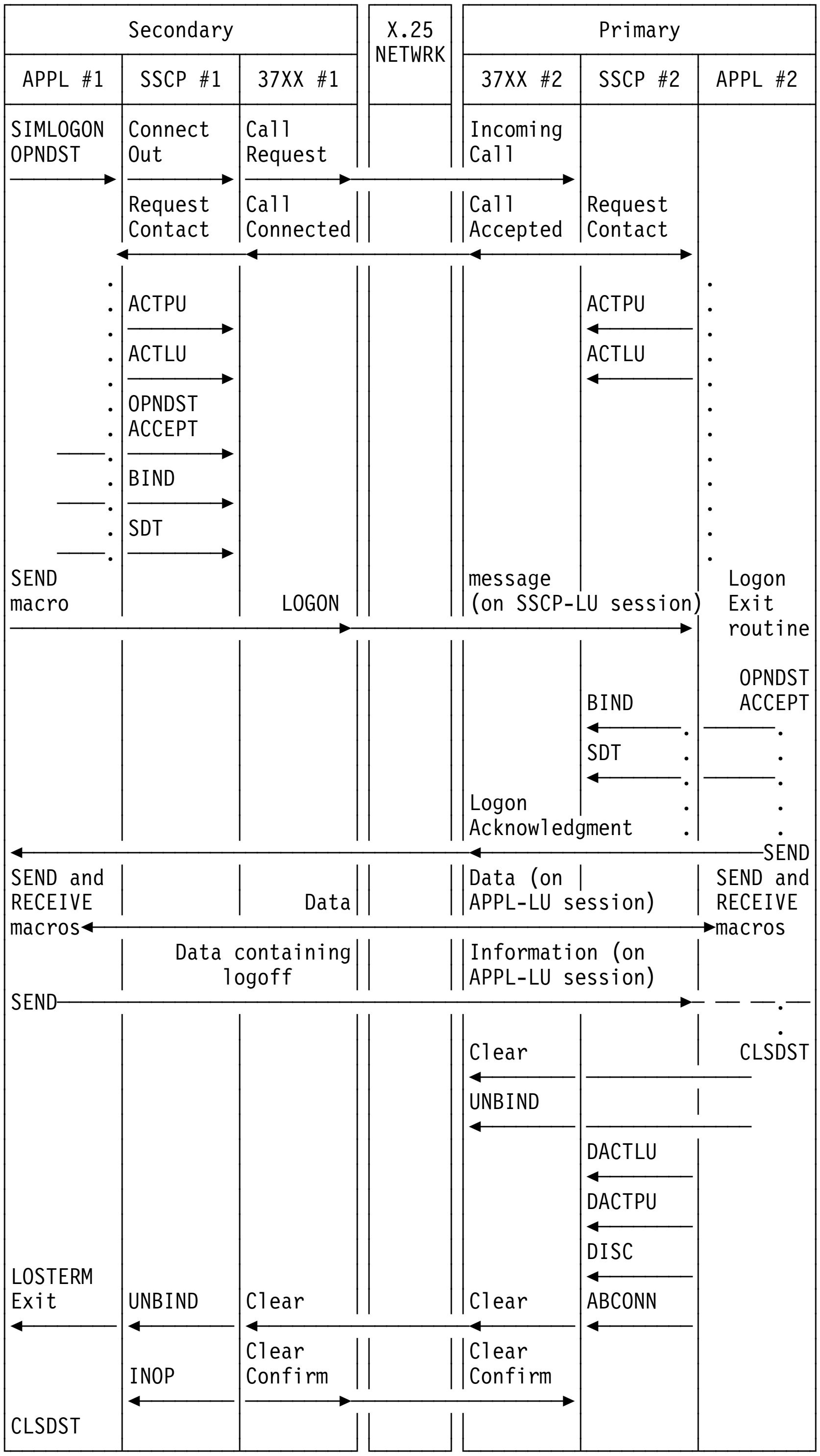


*Figure 2-6. APPL-to-APPL Session through PCNE-to-PCNE*

## PCNE-to-PCNE Considerations

PCNE-to-PCNE sessions require special planning. You must solve some PCNE-specific problems when choosing the method of synchronization between endpoints and the level of delivery confirmation to be used for a PCNE-to-PCNE session.

### Application-Level Synchronization

The session between the two PCNE endpoints must be controlled carefully. The connection is not controlled by any X.25 or SNA flow control. In addition, the line between the endpoints is a duplex line. Because the connection can support communication in both directions from the host *simultaneously*, the session can lose synchronization easily.

If synchronization is required, the PCNE applications must create their own synchronization method. Methods for creating synchronization include (but are not limited to):

- The inclusion of special data to signify the passage of the flow direction
- The use of half-duplex flip-flop protocol

## PAD Support Functions of NPSI

NPSI provides two levels of support for a PAD:

- Integrated PAD support

  If the PAD complies with Recommendations X.3, X.28, and X.29, you can use the integrated PAD support function of NPSI to interface with the PAD. Integrated PAD support features are based on those provided by PCNE.

- Transparent PAD support

  If you need to use the facilities of the PAD, as defined by Recommendations X.3, X.28, and X.29, other than those provided by integrated PAD support, or the PAD service does not follow Recommendation X.29, you can use NPSI’s transparent PAD function. Transparent PAD support provides an interface between host application programs and any type of PAD.

  When the application wants to use Interrupt packets to control the flow direction or the Reset packet to monitor the operation of the network, transparent PAD can be used to control an X.25 remote DTE that does not use a PAD.

**Note:** The NPSI PAD support should not be used for connections through SDLC PADs. These connections are supported by the QLLC (or LLC 3) or through the SPNQLLC specification on the X25.MCH statement.

## Integrated PAD Support

A host application program communicating through the integrated PAD function of NPSI is largely isolated from the intricacies of communication with the remote PAD. The application program operates in its normal fashion without regard for the interface at the remote end. Integrated PAD provides support for the extra processing that is required for operation of the PAD. The application program is generally not involved in sending or receiving PAD commands or responses.

The considerations for running with the LUSIM apply in the case of integrated PAD support. Integrated PAD code within NPSI interacts with the remote PAD according to CCITT Recommendation X.29. NPSI supports RU chaining for the integrated PAD function.

## PAD Parameter Settings

PAD parameters 1, 7, and 8 are of particular interest in an NPSI environment. When under the control of the integrated PAD support function, NPSI sends a PAD message to the PAD at ACTLU for SVC and at BIND for PVC. NPSI uses this PAD message to set the following PAD parameter values:

| PAD<br>Parameter | Default<br>Setting (Decimal) |
|---|---|
| 1 | 0 |
| 7 | 21 |
| 8 | 0 |

For more information about NPSI PAD parameters, see *NPSI Planning and Installation*. For more information about NPSI PAD parameter settings, see the customization section of *NPSI Diagnostics, Customization, and Tuning*.

## PAD Parameter Strings on the PADPARM keyword

NPSI allows you to specify, at system generation, one or more strings of PAD parameters using the PADPARM keyword on the X25.PAD statement. The PADINDX keyword on the X25.MCH statement provides an index, which points to a selected parameter string.

The PAD parameter string is passed to the PAD at ACTLU for SVC and at BIND for PVC.

For more information about PAD parameter generation, see *NPSI Planning and Installation*.

## SIGNAL and BREAK Processing

When NPSI receives a SIGNAL from the host application program, the type of action taken is dependent on the setting of the PAD parameter 7.

When the PAD parameter setting is 0705 or 0721, NPSI converts a SIGNAL from the host application program to an indication of BREAK message that is sent to the PAD.

Additionally, if the PAD parameter setting is 0702, NPSI sends a Reset Indication message to the virtual circuit. If the PAD parameter setting is 0701, NPSI sends an Interrupt Indication message to the virtual circuit.

When you press the **Break** key, NPSI's actions are dependent on the setting of PAD parameter 7.

If the PAD parameter setting is 0701, NPSI receives an Interrupt packet from the PAD. NPSI then sends a SIGNAL to the host application and an Interrupt Confirmation packet to the PAD. This sequence causes the application to discontinue output to the terminal until a CD flag is received.

If the PAD parameter setting is 0702, NPSI receives a Reset packet from the PAD, and then sends a Reset Confirmation packet to the PAD. If the cause and diagnostic code in the Reset packet matches one of the values in PADBRKCD, NPSI sends a SIGNAL to the host application. The PADBRKCD keyword is coded on the

X25.MCH statement. Additionally, NPSI purges its input and output queues. This sequence causes the application to discontinue output to the terminal.

If the PAD parameter setting is 0705, NPSI receives an Interrupt packet and indication of BREAK message from the PAD. NPSI then sends a SIGNAL, which is built on the Break packet, to the host and an Interrupt Confirmation packet to the PAD. When a DATE CTCP is used, a SIGNAL is sent to the application on the data session. If the BREAK was entered during output data, this sequence causes the application to discontinue output to the terminal until a CD flag is received. If the BREAK was entered during input data, this sequence causes the application to delete the characters that were entered just before the SIGNAL arrived.

If the PAD parameter setting is 0708 (which leads to a DATA ESCAPE PAD state when you press the Break key, NPSI sends a SIGNAL, which is built on a Reset or an Interrupt packet, to the host application. If the cause and diagnostic code in the RESET packet matches one of the values in PADBRKCD, NPSI sends a SIGNAL to the host application. If the BREAK was entered during data input, the host application purges the last block of data that was entered. If the BREAK was entered during output data, the host application discontinues output until a CD flag is received.

The sequence of events that occurs when the PAD parameter setting is 0721 remains as in previous NPSI releases, except that the SIGNAL that is sent to the host application contains PAD setting information.

The SIGNAL RU is formatted as shown in Figure 2-7.

| Byte | 00 | 01 | 03 |
|---|---|---|---|
| | X'C9' | 0 0 0 1 | Extension |

*Figure 2-7. Format of SIGNAL RU*

The SIGNAL Extension field contains the PAD parameter setting information.

**Note:** Integrated PAD, PCNE, and GATE devices can be on the same MCH if you code different IDBLK numbers for the IDBLKP, IDBLKC, and IDBLKG keywords. On a DATE MCH, the SIGNAL/BREAK processing is the same for an Integrated PAD.

## SHUTD Considerations

When a terminal or printer is connected to a host through a PAD, the device may need to be attached to more than one VTAM application program. The device might need to move between application programs or, as is true of a printer, the device itself might be passed between application programs. NPSI provides two methods to allow devices to be shared by more than one application program. To specify the option you want to use, you must code the SHUTD keyword of the X25.MCH statement in one of the following ways:

SHUTD=INVCLR

Code the INVCLR parameter for the SHUTD keyword when you want the device to be disconnected from one application program before being connected to another.

When an SNA SHUTD command is sent by the application and the SHUTD keyword is coded as INVCLR, NPSI interprets the SHUTD command and converts it into one of the following:

- An X.25 Invitation to Clear packet, if the virtual circuit is switched
- A Reset packet, if the circuit is permanent

NPSI immediately returns to the host a positive response to the SHUTD command.

The PSDN responds to the packet by returning either a Clear Indication packet (for SVCs) or a Reset Confirmation packet (for PVCs). These packets return the virtual circuit to its initial status, breaking the connection to the host.

Reception of these packets by NPSI results in an SNA SHUTC command being sent to the host. In response to this command, the application program sends an UNBIND command to the LUSIM function of NPSI, which unbinds the SNA session.

Once the session is broken, the device must either redial the host, if the connection was through an SVC, or logon to the new application program, if the connection was by way of a PVC.

Figure 2-8 on page 2-13 shows the data flow of the SNA commands and the X.25 packets used to switch the attachment of a device from one application program to another when the INVCLR parameter is specified.

SHUTD=NOINVCLR

Code the NOINVCLR parameter for the SHUTD keyword when you want to keep the X.25 connection active on SHUTD. In that case, an Invitation to Clear packet or Reset packet is not issued to the PSDN. With this option, the virtual circuit is not cleared, and the device can be passed between application programs.

Figure 2-9 on page 2-14 shows the data flow when the NOINVCLR parameter is specified. In the case of a PVC, if the SNA UNBIND command includes an indication that an SNA BIND command is forthcoming, a Set PAD message is not sent with the following SNA BIND command.

**Note:** SHUTD=NOINVCLR is mandatory when you code PAD=INTEG and GATE=DEDICAT on the same MCH.

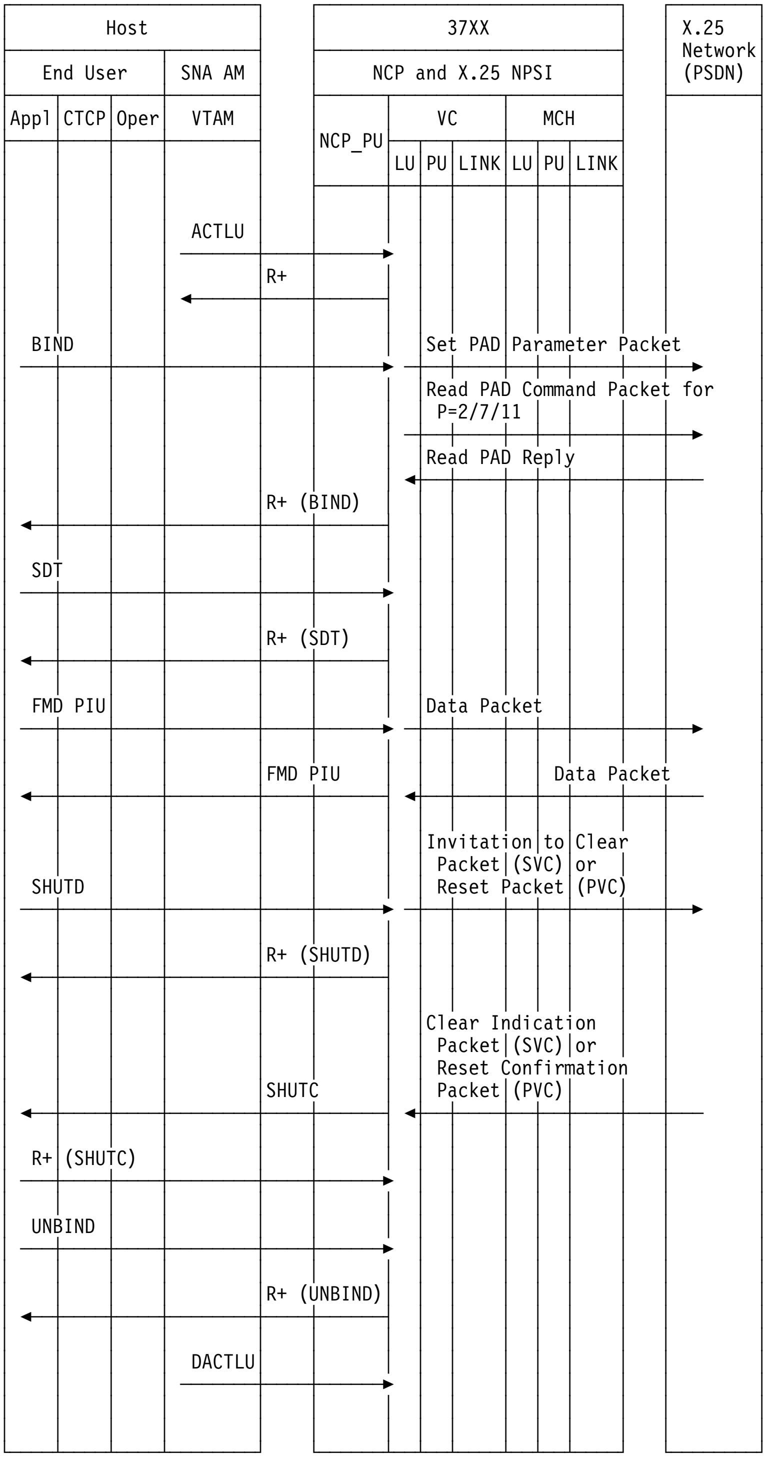


*Figure 2-8. Data Flow with SHUTD=INVCLR Option*

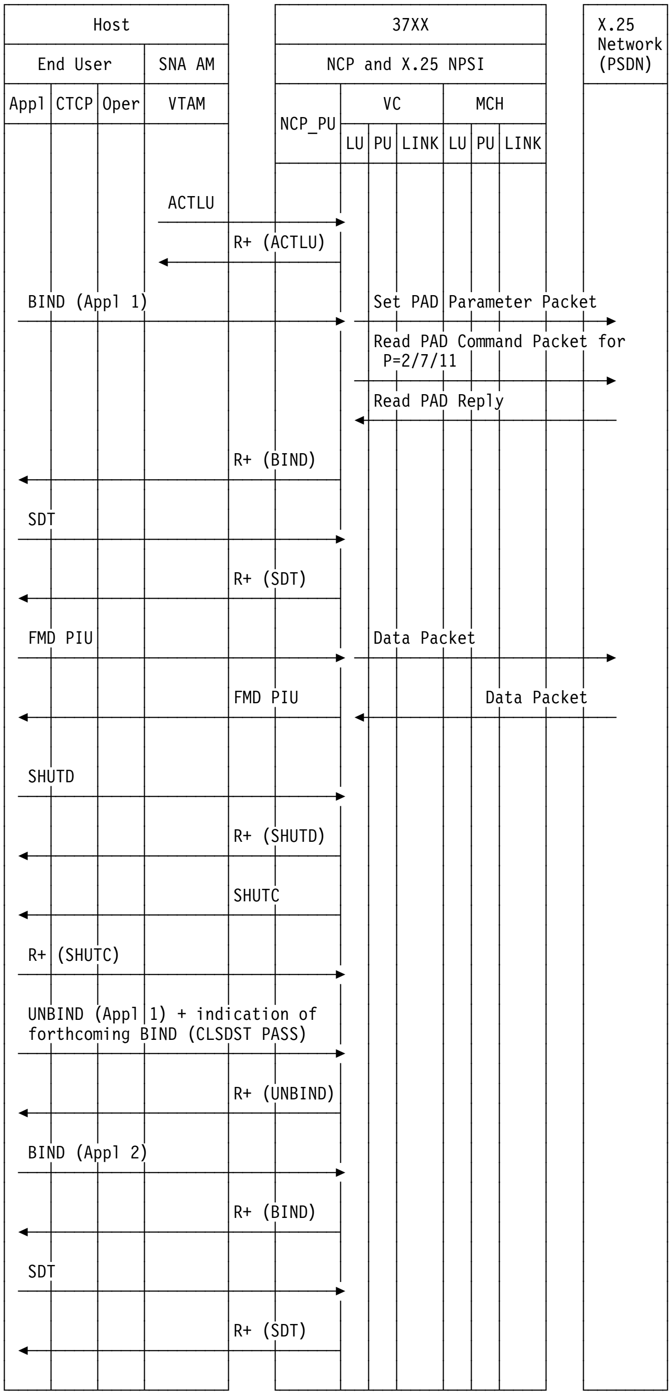


*Figure 2-9. Data Flow with SHUTD=NOINVCLR Option*

## Application Program Support for Password Protection

If the application program wants to use the password protection feature of NPSI, it can use the enable presentation (ENP) and inhibit presentation (INP) characters to define when a field is not to be echoed to the terminal. The hexadecimal values for these control characters are:

| | |
|---|---|
| **X'24'** | INP |
| **X'14'** | ENP |

**Note:** If NPSI does not translate EBCDIC to ASCII, the control character used for inhibit presentation is X'12' rather than X'24', and the overstrike message is translated to ASCII EVEN code. If this default value must be changed, see the customization section of *NPSI Diagnosis, Customization, and Tuning*.

To use the password protection function, the application program places an INP character at the end of the data stream prompting the protected information. The protected information is interpreted and converted into the appropriate PAD commands to disable the display at the device. Disabling the display at the device can be accomplished through one of the following methods:

- The inhibition of echoing data back to the device, if the device is operating with echoplexing
- The transmission of an 8-character blackout string, if the device is not operating with echoplexing

**Note:** Both of the options for this function work with typewriter-like devices. However, the inhibition of echoing data back to a device works only with video display terminals.

NPSI determines whether to inhibit echo or to transmit a blackout message to support password protection. This determination is made by interrogating the setting of PAD parameter 2 (echo). If the echo mode is ON, the echo facility is used to inhibit the display. Otherwise, the blackout message is used to hide the printing.

Upon receiving and processing the response, the host application program starts the next output buffer with the ENP character to initiate the redisplaying of characters on the device.

NPSI is responsible for all processing required for the password protection function. To implement this function, you must specify PWPROT=YES on the X25.MCH statement. If NPSI processing of the password protection function does not meet your requirements, you can use the GATE, DATE, or transparent PAD function rather than the integrated PAD support function.

## RU Chaining Support for Integrated PAD

NPSI supports RU chaining for long non-SNA messages on both inbound and outbound flows. This optional support is selected at system generation by coding MBITCHN=YES on the X25.MCH statement. This support is fully available for virtual circuits working in PCNE or integrated PAD, and partially available for virtual circuits working in GATE or transparent PAD. Without this support (MBITCHN=NO), when receiving an RU, NPSI (under LLC 0, 4, and 5) acts as if it were an OIC RU, that is, after removing the TH and RH fields, it sends a CPS. These packets are linked together with an M bit by setting the M bit to 1 in all but the last packet. On inbound flow, when receiving a CPS, NPSI builds an OIC RU.

This function allows non-SNA equipment to use the SNA RU chaining function when communicating with an SNA application through NPSI. This function can be used for outbound and inbound flows using the generation parameter MBITCHN.

The chain of PIUs may be composed of:

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

**Note:** If you specify MBITCHN=YES, NPSI sends the entire chain of PIUs as either one CPS if DBIT=NO, or a CPS series (an M bit sequence, or MBS) if DBIT=YES.

On inbound flow, NPSI maps each CPS on one or more SNA elements of the chain.

# Transparent PAD Support

Transparent PAD support is provided to allow an application program to control the functions of a PAD. When a program uses transparent PAD, the first byte of the data passed between the application program and NPSI is used as a packet type indicator. This byte notifies NPSI and the host application program of the type of packet and allows each end to process the packet accordingly.

## Uses of Transparent PAD Support

Transparent PAD is used for two reasons. The first reason is to permit you to use the facilities of the PAD, as defined by Recommendations X.3, X.28, and X.29, other than those provided by integrated PAD support. Using transparent PAD support, a host application program can control its interaction with a remote DTE connected through a PAD. This higher level of control is required by some application programs and users of NPSI.

The second reason to use transparent PAD is to provide control of PAD services, which *do not* follow Recommendation X.29. Transparent PAD support allows any type of PAD to be supported and controlled by an application program through NPSI. An example of an X.3 PAD service is a bisynchronous PAD. This type does not support Recommendations X.3, X.28, and X.29 and often uses a vendor-specific interface that must be provided for by the host application program.

Transparent PAD support can be used on a physical circuit defined for any mode of operation; that is, the X25.MCH statement can specify GATE=NO, GATE=GENERAL, or GATE=DEDICAT. Even when MBITCHN=YES is coded on the X25.MCH statement, the SNA RU chaining function is not supported. On inbound flow, packets containing the M bit are accumulated until MAXRU of the BIND command is reached. The PIU is then forwarded to the host. Otherwise, the size of the accumulated data needed to build a PIU is determined by the length of the CPS received. On outbound flow, NPSI converts each FIC, MIC, or LIC PIU into an OIC, and then builds and sends a corresponding CPS.

Figure 2-10 on page 2-17 shows the data flow in a network configuration using transparent PAD support.

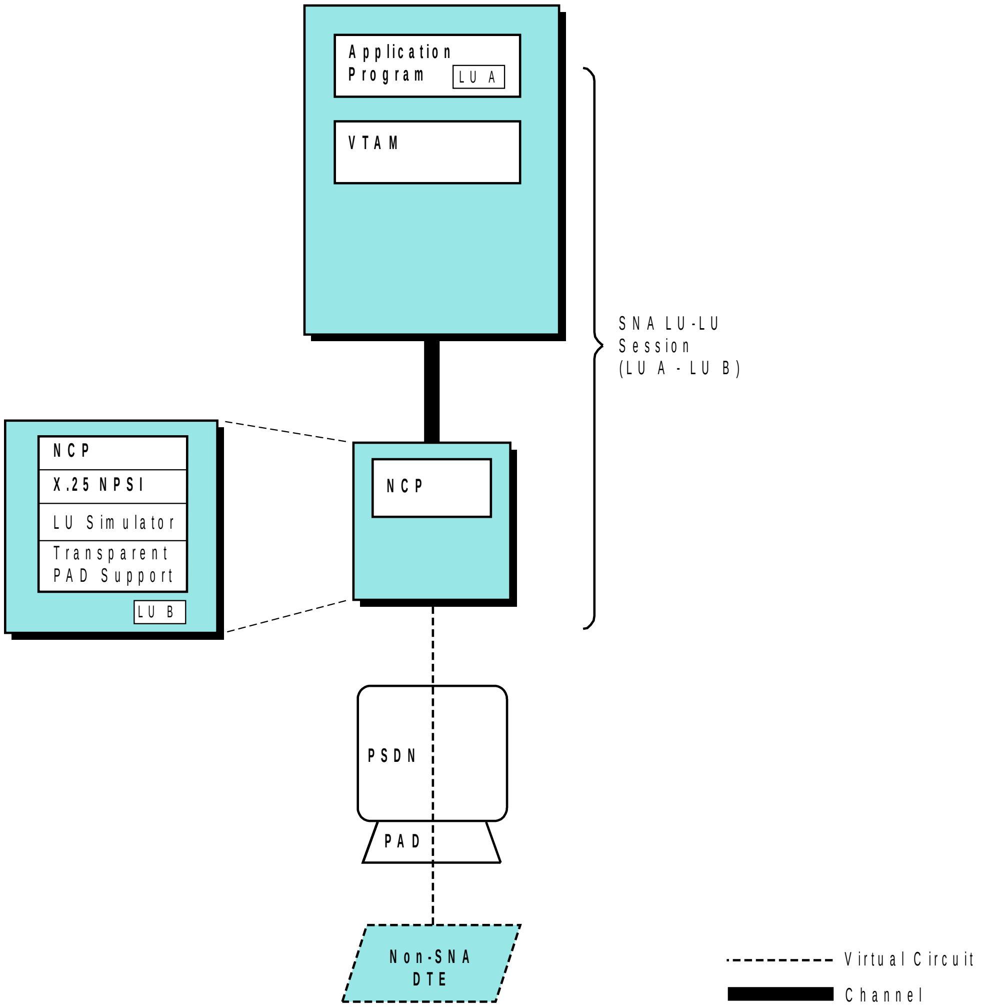


*Figure 2-10. SNA Host Node to Non-SNA DTE (Transparent PAD)*

## Programming Requirements for Transparent PAD

The transparent PAD support function is designed for application programs that need to control the remote PAD by means of commands exchanged between the application program and NPSI.

For application programs using transparent PAD support, the virtual circuit setup and takedown is identical to that for the type 0 virtual circuit. However, the contents of the following types of packets are sent from or routed to the application program over the LU-LU session between the application program LU and the virtual circuit LU.

- Data packets
- Qualified Data packets
- Interrupt packets
- Reset packets

Commands and information for PAD control are contained in Qualified Data packets.

An application program must use the first byte (byte 0) of the RU to specify the packet type indicator, which denotes the type of data contained in the packet. This byte is used in communication between the application program and the transparent PAD function within NPSI. The first byte of each RU must contain one of the values shown in Table 2-1.

*Table 2-1. Command Formats for Transparent PAD RUs*

| RU Byte 0 | Packet Type |
|---|---|
| X'00' | Data without Q bit |
| X'02' | Data with Q bit |
| X'1B' | Reset |
| X'1F' | Reset Confirmation |
| X'23' | Interrupt |
| X'27' | Interrupt Confirmation |

Qualified Data packets (that is, data packets with the Q bit set to 1) are used to exchange information between the application program and the remote PAD. Data packets with the Q bit set to 0 are used to exchange information between the application program and the remote DTE.

For Reset packets, the second and third bytes of the RU contain cause and diagnostic values. When Reset packets are exchanged, the only action of NPSI is to set the packet received P(R) and packet sent P(S) counters to 0. The application is responsible for issuing a CLSDST if required, or to perform a checkpoint/restart with the remote DTE, if this was planned in the convention between the application and the remote DTE.

Upon receipt of a Reset Indication packet from the network, the application program must send the reset confirmation to NPSI. NPSI then forwards the Reset Confirmation to the network.

For Interrupt packets, the second byte of the RU contains the interrupt cause (usually set to 0). For Interrupt packets from the PAD, the application program must initiate the transmission of the Interrupt Confirmation packet.

NPSI performs normal packet functions and recombining through the M bit in packet headers.

Depending on the value of the TRAN keyword coded on the X25.MCH statement, NPSI can translate between EBCDIC and ASCII. When translation is requested, data beyond byte 0 (the first byte) in RUs flowing on transparent PAD sessions is translated by NPSI in the case of Unqualified Data packets.

The following describes the format for each transparent PAD RU.

***Data without a Q Bit:*** The data packet without a Q bit is used to signify that the associated data is only user data. There is no special significance to the attached data. The format of this packet is:

| | |
|---|---|
| Byte 0 | X'00' |
| Bytes 1 through n | Variable-size field containing the data packet |

***Data with a Q Bit:*** The data packet with a Q bit is used for data that is sent to or from the remote PAD. All PAD commands are contained in data that start with this indicator. To instruct the PAD to perform a function, place the value X'02' in byte 0 of the data and place the PAD command directly behind it. NPSI recognizes your desire to send a command to the PAD. It responds by placing the PAD command into the packet and by setting the Q bit to 1. The packet is then sent to the remote PAD. The PAD performs the desired function, and the host application program continues with its required processing.

The format of the data packet with a Q bit is:

| | |
|---|---|
| Byte 0 | X'02' |
| Bytes 1 through n | Variable-size field containing the data packet |

Qualified data can be used to perform any type of PAD command. The valid X.29 PAD commands are shown in Table 2-2.

*Table 2-2. Valid X.29 PAD Commands*

| Code | Command |
|---|---|
| X'00' | Parameter Indication |
| X'01' | Invitation to Clear |
| X'02' | Set PAD |
| X'03' | Indication of Break |
| X'04' | Read PAD |
| X'05' | Error |
| X'06' | Set and Read PAD |
| X'07' | Reselection |

For detailed information about the coding requirements for these PAD commands, see *NPSI Diagnosis, Customization, and Tuning*. This book contains the formats of all commands and the status and error codes.

***Reset:*** A Reset packet is used to reset the PAD. When the PAD is reset, the communication between NPSI and the remote DTE is reset to its initial state. This reset can be totally disruptive to the session state; all outstanding packets are discarded.

When an application program receives a Reset Indication packet, the application program must send a Reset Confirmation command back to NPSI. NPSI then forwards a Reset Confirmation packet to the PAD. If an application sends a Reset command, a Reset Confirmation packet is returned to the application. The format of this packet is:

| | |
|---|---|
| Byte 0 | X'1B' |
| Bytes 1 and 2 | Cause and diagnostic fields |
| Bytes 3 through n | Variable-size field containing optional user data |

***Reset Confirmation:*** Reset requires the use of a Confirmation packet called the Reset Confirmation packet. The format of this packet is:

| | |
|---|---|
| Byte 0 | X'1F' |
| Bytes 1 through n | Variable-size field containing optional user data |

***Interrupt:*** An application program can start the Interrupt procedure by sending an Interrupt Indication command to the PAD through NPSI. When an Interrupt packet is received by NPSI, it is passed to the PAD through the network. The PAD returns an interrupt confirmation to NPSI, which relays it to the application. When an application program receives an Interrupt Indication packet, the application program must send an Interrupt Confirmation command back to NPSI. NPSI then forwards an Interrupt Confirmation packet to the PAD. The format of this packet is:

| | |
|---|---|
| Byte 0 | X'23' |
| Byte 1 | Interrupt cause |
| Bytes 2 through n | Variable-size field containing optional data |

***Interrupt Confirmation:*** An application program must be able to handle either sending or receiving an Interrupt Confirmation command. The format of the Interrupt Confirmation packet is:

| | |
|---|---|
| Byte 0 | X'27' |
| Byte 1 | Optional user data |

## Recommendation for Transparent PAD

Half-duplex contention is recommended, because unexpected control packets can come from the network at any time.

For information about RU Chaining Support, see “RU Chaining for Virtual Circuits Working in GATE and Transparent PAD” on page 1-6.

# Chapter 3. Programming a GATE CTCP

# Chapter 3. Programming a GATE CTCP

This chapter explains how to program virtual circuits when you use one of the following:

- GATE
- The fast connect (FC) GATE function of NPSI
- The X.21 switched connection using GATE
- FTPI

## GATE Function of NPSI

When you use the GATE function of NPSI, communication is through the CTCP. All traffic passes through the CTCP. GATE uses the NPSI LUSIM to operate and communicate with non-SNA LUs. For GATE, you do not need to specify the LLC type in the CALL REQUEST command, because GATE always uses a type 4 LLC.

The CTCP is responsible for handling all commands and data that pass between the application program and NPSI. GATE resides in NPSI and communicates with the CTCP, which contains a VTAM application interface. Two types of LU-LU sessions perform the command and data communication.

- The CTCP conducts a *control session* to control the initiation and termination of sessions with remote DTEs. The control session is between the CTCP and the MCH LU.
- A *data session* is created for each virtual circuit for data transmission. The data session is between the CTCP and the LU associated with the virtual circuit for the duration of the call. This LU is called virtual circuit LU in this chapter for wording simplicity.

**Note:** The PLP timers are started and controlled by NPSI. The CTCP does not need to start these timers. When the network does not respond to a message, NPSI sends an INFORMATION REPORT message to the CTCP on the MCH LU session.

## Control Session Establishment

The control session for GATE can be established using any of the following session initiation requests:

- The network operator can issue one of the following commands:
  - V NET,LOGON=*application*
  - V NET,ACT with LOGAPPL=*application* coded on the LU statement

  When either of these commands is issued, a session is automatically initiated between the controlling application (specified by *application*) and the LUSIM.
- The application program can issue one of the following:
  - The OPNDST OPTCD=ACQUIRE statement instruction
  - The SIMLOGON and OPNDST OPTCD=ACCEPT statement instructions

  In either case, the application program must have previous knowledge of the LU’s existence to use these instructions. In the second case, a LOGON exit must be coded as well.

The following processes are unique to GATE:

- All data for GATE processing passes between the CTCP and the virtual circuit LU. The CTCP must either process the request or pass it to the appropriate application.
- The use of GATE is determined on a virtual circuit-by-virtual circuit basis. Therefore, a single physical circuit can support the GATE LLC, as well as other types of LLCs.
- Call-out is simulated as a call-in to NCP and the access method. For this reason, you do not need to code a PATH statement in the switched major node definition.

## Virtual Circuit Establishment

You can use different methods to establish virtual circuits, depending on whether the circuits are PVCs or SVCs. The methods available for the establishment of PVC and SVC circuits are described in the following sections.

### Session Establishment through a PVC

The control session (CTCP LU to MCH LU) performs virtual circuit management. For this reason, establishment of the control session must be the first step in virtual circuit establishment. The PVC can be established by issuing any of the following activation requests and statement instructions:

- V NET,ACT with LOGAPPL=*appl1* (coded on the virtual circuit LU definition statement)
- V NET,LOGON=*appl1*
- OPNDST OPTCD=ACQUIRE
- SIMLOGON and OPNDST OPTCD=ACCEPT
- LOGON from a remote DTE

### Session Establishment through an SVC

To set up SVCs, the CTCP must send commands to NPSI over the APPL LU-to-MCH LU session.

Your CTCP must open the session with the physical circuit LU before GATE virtual circuit sessions can be established. The GATE code in NPSI finds the name of the CTCP application program in the PLU name field of the BIND SESSION command.

One CTCP can manage several physical circuits, so the user data field of the BIND SESSION command must contain the symbolic name of the physical circuit LU, (MCH LU name). This symbolic name, which is associated with the SNA resource identifier, enables the CTCP to identify the SNA resource and MCH when the logon message is received.

***V3R4 and Later:*** For GATE SVCs, the quality of service (QOS) is defined at Call setup.

***Call Out:*** The Call Request packet (with or without the QOS facility) is passed unchanged from the CTCP to the network. The selected value, if any, is saved by NPSI from the Call Connected packet that is received from the network.

***Call In:*** The Incoming Call packet (with or without the QOS facility) is passed unchanged from the network to the CTCP. The selected value, if any, is saved by NPSI from the Call Accepted packet that is passed by the CTCP.

You can select the following priorities:

| | |
|---|---|
| LOW | 0 to 3 (X'00' to X'03') |
| NORMAL | 4 to 7 (X'04' to X'07') |
| HIGH1 | 8 to 11 (X'08' to X'0B') |
| HIGH2 | 12 to 14 (X'0C' to X'0E') |
| Reserved | 15 to 254 (X'0F' to X'FE') |
| Unspecified | 255 (X'FF') |

Like the packet-level window, the QOS facility can be negotiated. See *NPSI Planning and Installation* for more information. If the negotiated value is unspecified or reserved, the default priority on the X25.MCH statement is used.

**V3R6 and Later:** When the BIND Session command user data field does not carry the MCH LU name, the CTCP can issue a new command over the LU-LU control session. This new command, MLUNAME, carries the MCH LU name. MLUNAME, when used, should be the first command issued by the GATE CTCP.

Once the session between the LU associated with the CTCP and the LU associated with the physical circuit is established, a connection from an SVC between the CTCP and a remote DTE can be established in one of two ways:

- A CALL REQUEST command from the CTCP to the physical circuit LU
- An Incoming Call packet containing a type 4 virtual circuit identifier

**V3R7 and later:** In prior releases of NPSI the CTCP was required to pass the X25 flow control values to the GATE into the call setup commands. The CTCP can now let the GATE determine these values. The CTCP does this by setting the PW (packet window) and PSIZ (packet window size) fields to zero in the CALL REQUEST or CALL ACCEPTED commands that it sends to GATE.

## BIND Format for an MCH LU (Control Session)

The format of the BIND between the CTCP LU and the physical circuit LU (MCH LU) is the same as that described in Table 1-1 on page 1-11 with the following additions:

| **Position in**<br>**BIND Image** | **Description** |
|---|---|
| Byte 7 | The physical circuit LU must work in contention mode |
| Byte 26 | Cryptographic options field length |
| Bytes 27 through k | Crypto options |
| Byte k+1 | Length of PLU name (L1) |
| Bytes k+2 through m | Primary LU name (CTCP name) |
| Byte m+1 | Length of SLU name (L2) |
| Bytes m+2 through n | Secondary LU name (physical circuit LU name) |

The following is the layout of this section of the BIND command without crypto fields:

| **Position in**<br>**BIND Image** | **Description** |
|---|---|
| 26 | Length of crypto field (Value = 0) |
| 27 | Length of PLU name (Value = L1) |
| 28 | PLU name |
| 28+L1+1 | Length of SLU name |
| 28+L1+2 | SLU name |

## BIND Format for a Virtual Circuit LU

The BIND format for a virtual circuit LU is the same as shown in Table 1-1 on page 1-11.

## Logon Message Format for GATE

Once an SVC is set up between the CTCP and a remote DTE, the NPSI GATE function generates a logon message. This message sets up the LU-LU session between the LU associated with the CTCP and the LU associated with the virtual circuit for the duration of the call. The logon message has the following format:

```
LOGON APPLID(xxxxxxxx) DATA(yyyyzzzzzzzz)
```

***where:***

`xxxxxxx` Is the PLU (CTCP) name of up to 8 characters.

`yyyy` Is a 2-byte connection identifier (CNID) in the case of a call request made by the CTCP, or a 2-byte SNA resource identifier (RESID) in the case of an incoming call from the network. RESID has an `X'F'` as the first digit. See the CNID and RESID definitions on page 3-10.

`zzzzzzzz` Is the SLU (physical circuit) name of 8 characters. It is padded to the right with blanks. zzzzzzzz is obtained from the control session BIND or from the MLUNAME command. See "Session Establishment through an SVC" on page 3-2.

The following example assumes that the CTCP application name is CTCP1 and that the physical circuit LU name is XU038:

```
LOGON APPLID(CTCP1) DATA(0001XU038)
```

GATE constructs and sends to the SSCP a logon message that allows the LU associated with the virtual circuit to enter into a session with the CTCP.

## Processing of Data Received before Session Establishment

Data received before the session is established between the virtual circuit LU and the CTCP is treated in the following manner:

**PVCs** After the ACTLU command is processed, the first message is treated as a logon and is passed to the SSCP on the SSCP to LU session. The other data is passed to the CTCP on the CTCP to virtual circuit LU session after SDT processing.

**SVCs** The data is queued in NPSI. The data is passed to the CTCP on the CTCP to virtual circuit LU session after SDT processing. The logon message is built by the NPSI GATE function and passed to the SSCP on the SSCP to virtual circuit LU session after the ACTLU command is processed.

## Virtual Circuit Termination

A virtual circuit can be deactivated using any of the following requests and commands:

- Clear Request from remote DTE
- CLSDST from the application program
- Request from the application program for the CTCP to CLEAR
- V NET,INACT command

- V NET,TERM command

In the simplest case, the remote DTE requests the session termination. In this instance, a Clear Request packet is received from the PSDN, and NPSI transfers this packet to the CTCP as a CLEAR REQUEST command.

The user application program can initiate session termination. In this case, NPSI receives an ABCONN request from VTAM and clears the SVC, or sends a reset out for a PVC.

The user application program can notify the CTCP to disconnect the virtual circuit. When the CTCP receives this request:

- The CTCP builds a CLEAR command, including the cause and diagnostic bytes as determined by the CTCP.
- The CLEAR command is sent to NPSI.
- NPSI then sends a Clear Request packet to the PSDN.
- A Clear Confirmation packet is sent to NPSI. NPSI transfers this packet to the CTCP in the CLEAR CONFIRMATION command.
- A CLSDST macro is issued to end the session with the virtual circuit LU. NPSI does not generate an INOP message.

The network operator can terminate the session by issuing a V NET,INACT or V NET,TERM command for the virtual circuit PU or the virtual circuit LU. The SNA and X.25 resources are disassociated when this command is issued. NPSI then clears the SVC or sends a reset out for a PVC.

## Control Session Termination

The control session for GATE can be terminated in any of the following ways:

- The application program can request that the CTCP issue a CLSDST macro for the MCH LU.
- The network operator can issue one of the following commands:

  – V NET,INACT,ID=*mch lu*
  – V NET,TERM,LU=*mch lu*
  – CANCEL *the CTCP name*

  In the first and second commands, *mch lu* is the identifier of the physical circuit to be terminated. In the third command, the CTCP is canceled.

- If the CTCP abends, the physical circuit can no longer process GATE calls.

**Note:** If the CTCP is canceled or abends, the resources used with this CTCP must be deactivated and reactivated. This reactivation is necessary, because any LUs undergoing logon processing at the time the CTCP fails will be waiting for a response from the CTCP that will not be forthcoming. Therefore, the switched major nodes where the LUs are defined must be deactivated and reactivated.

## Command Interface between GATE and the CTCP

In the following command descriptions, the RESID field in the RU format corresponding to the resource identifier designates either the set of SNA resources (LINE, PU, and LU) that are associated with a virtual circuit, or the X.25 virtual circuit number (systems generation option VCID). SNA resource identifiers are associated with virtual circuit for each MCH. They correspond to the number of the SNA resource within the set of SNA resources associated with the MCH. The *range* of resource identifier numbers is the same as the *range* of virtual circuit numbers. An SNA resource number consists of three digits in hexadecimal format. The first digit is the logical group number and the next two digits are the logical channel numbers.

The following example illustrates how resource identifier numbers are determined.

```
X25.LCG  LCGN=0
X25.VC   LCN=(1,2),TYPE=P ....  2 PVCs
X25.VC   LCN=(3,99),TYPE=S ... 97 SVCs
X25.LCG  LCGN=2
X25.VC   LCN=(1,3),TYPE=P ....  3 PVCs
X25.VC   LCN=(4,98),TYPE=S ... 95 SVCs
```

In this example, SNA identifiers X'003' through X'063' and SNA identifiers X'204' through X'262' are used to map all SVCs of this MCH. This setup allows CTCPs to continue to run without change.

The commands recognized or generated by GATE are listed in Table 3-1 on page 3-7.

*Table 3-1. CTCP-GATE Control Commands*

| Command Code | Command | Session with CTCP |
|---|---|---|
| 00 | DATA EXCHANGE (Q bit set to 0) | Data Session |
| 02 | DATA EXCHANGE (Q bit set to 1) | Data Session |
| 0B | INCOMING CALL/CALL REQUEST | Control Session |
| 0F | CALL CONNECTED/CALL ACCEPTED | Control Session |
| 13 | CLEAR indication/CLEAR request | Control Session or Data Session* |
| 17 | CLEAR CONFIRMATION | Control Session or Data Session* |
| 1B | RESET | Control Session or Data Session* |
| 1F | RESET CONFIRMATION | Data Session |
| 23 | INTERRUPT | Data Session |
| 27 | INTERRUPT CONFIRMATION | Data Session |
| F1 | DIAGNOSTIC | Control Session |
| F4 | MLUNAME | Control Session |
| FF | INFORMATION REPORT Message | Control Session or Data Session* |

**Note:** An asterisk (*) indicates that the command flows on the data session if the VC LU is in session; otherwise, the command flows on the control session.

The following describes the data exchange commands and the control commands used by GATE and the CTCP. The data exchange commands are followed by user data. The control commands are followed by additional control information.

## DATA EXCHANGE without Q Bit (CTCP to GATE, GATE to CTCP through the Data Session)

The DATA EXCHANGE command with the Q bit set off is issued from the CTCP to GATE and from GATE to the CTCP through the data session. The command is sent in the following format:

```
     ┌────┬──//──┐
RU:  │ cc │ user │
     └────┴──//──┘
```

| Bytes | Field | Description |
|---|---|---|
| 0 | cc | Command code. For this command, the value is X'00'. |
| 1–n | user | Variable-size field containing user data following the packet header. |

**Note:** The command byte is set only in FIC and OIC RUs.

## DATA EXCHANGE with Q Bit (CTCP to GATE, GATE to CTCP through the Data Session)

The DATA EXCHANGE command with the Q bit set on is issued from the CTCP to GATE and from GATE to the CTCP through the data session. The command is sent in the following format:

```
     ┌─────┬──//──┐
RU:  │ cc  │ user │
     └─────┴──//──┘
```

| Bytes | Field | Description |
|---|---|---|
| 0 | cc | Command code. For this command, the value is X'02'. |
| 1–n | user | Variable-size field containing user data following the packet header. |

**Note:** The command byte is set only in FIC and OIC RUs.

## INCOMING CALL (GATE to CTCP through the Control Session)

The INCOMING CALL command is issued from GATE to the CTCP through the control session. The command is sent in the following format:

```
     ┌─────┬────────┬──//──┐
RU:  │ cc  │ resid  │ icpk │
     └─────┴────────┴──//──┘
```

| Bytes | Field | Description |
|---|---|---|
| 0 | cc | Command code. For this command, the value is X'0B'. |
| 1–2 | resid | Resource identifier. The leftmost halfbyte is set to X'F'. |
| 3–n | icpk | Variable-size field containing the Incoming Call packet received from the network (without the 3-byte packet header). |

**Note:** If the CTCP does not receive the X.25 disconnection command (CLEAR), it can receive a CALL command for a resource that it assumes is already connected. The CTCP disregards the first call as cleared and processes the second call. If this is a common occurrence, verify the PU switched major node definitions.

## CALL REQUEST (CTCP to GATE through the Control Session)

The CALL REQUEST command is issued from the CTCP to GATE through the control session. The command is sent in the following format:

```
     ┌─────┬───────┬─────┬───────┬──//──┐
RU:  │ cc  │ cnid  │ pw  │ psiz  │ crpk │
     └─────┴───────┴─────┴───────┴──//──┘
```

| Bytes | Field | Description |
|---|---|---|
| 0 | cc | Command code. For this command, the value is X'0B'. |
| 1–2 | cnid | Unique connection identifier determined by the CTCP. This 3-digit number is right-aligned. The leftmost halfbyte **must** be X'0'. To avoid conflicts, use numbers greater than the highest SNA resource identifier numbers (virtual circuit numbers). |
| 3 | pw | Packet window size. |
| 4–5 | psiz | Packet size, in hexadecimal, used for this virtual circuit. |
| 6–n | crpk | Variable-size field containing the Call Request packet without the packet header. |

**Note:** Packet and window sizes specified in the CALL REQUEST command are recognized by NPSI. The flow control parameters included in the facilities of the Call Connected packet are ignored by NPSI. The CTCP should clear the call and perform a recall if these flow control parameters do not match the flow control parameters that were passed to NPSI in the CALL REQUEST command.

**Note:** Depending on the generation option specified, the flow control parameters included in the facilities of the Call Connected packet are recognized by NPSI if INTFAC=YES is specified.

**V3R7 and later:** The CTCP might set packet window and packet window size fields to zero to allow NPSI to determine the X.25 flow control values for a virtual call. See “Session Establishment through an SVC” on page 3-2 for information.

## CALL CONNECTED (GATE to CTCP through the Control Session)

This CALL CONNECTED command is issued from GATE to the CTCP through the control session. The command is sent in the following format:

```
                                   //
RU:  │ cc │ cnid │ resid │ ccpk │
                                   //
```

| Bytes | Field | Description |
|---|---|---|
| 0 | cc | Command code. For this command, the value is X'0F'. |
| 1–2 | cnid | Connection identifier, as specified in the CALL REQUEST command. |
| 3–4 | resid | Resource identifier. The leftmost halfbyte is set to X'F'. |
| 5–n | ccpk | Variable-size field containing optional bytes that followed the packet header of the Call Connected packet. |

**V3R4 and Later:** Depending on the generation option specified on the DEFPRI keyword, the QOS facility, (X'D2') contained in the Call Connected packet, determines the OSI priority of data that is used during the session.

## CALL ACCEPTED (CTCP to GATE through the Control Session)

The CALL ACCEPTED command is issued from the CTCP to GATE through the control session. The command is sent in the following format:

```
                                        //
RU:  │ cc │ resid │ pw │ psiz │ capk │
                                        //
```

| Bytes | Field | Description |
|---|---|---|
| 0 | cc | Command code. For this command, the value is X'0F'. |
| 1–2 | resid | Resource identifier, as found in the INCOMING CALL command. |
| 3 | pw | Packet window size. |
| 4–5 | psiz | Packet size in hexadecimal used for this virtual circuit. |
| 6–n | capk | Variable-size field containing the Call Accepted packet to be sent to the network without the 3-byte packet header. |

**V3R4 and Later:** Depending on the generation option specified on the DEFPRI keyword, the QOS facility, (X'D2') contained in the Call Accepted packet, determines the OSI priority of data that is used during the session.

**V3R7 and later:** The CTCP might set packet window and packet window size fields to zero to allow NPSI to determine the X.25 flow control values for a virtual call. See “Session Establishment through an SVC” on page 3-2 for information.

## CLEAR on Incoming Call (CTCP to GATE through the Control Session)

The CLEAR command can be issued from the CTCP to GATE through the control session to refuse an incoming call. The command is sent in the following format:

```
    ┌────┬───────┬──────┬──//──┐
RU: │ cc │ resid │ crdf │ user │
    └────┴───────┴──────┴──//──┘
```

| Bytes | Field | Description |
|---|---|---|
| 0 | cc | Command code. For this command, the value is X'13'. |
| 1–2 | resid | Resource identifier, as found in the INCOMING CALL command. |
| 3–4 | crdf | CAUSE and DIAGNOSTIC fields that are to be inserted in the Clear Request packet. |
| 5–n | user | Variable-size field containing optional user data to be put in the Clear Request packet after the CAUSE and DIAGNOSTIC fields. |

## CLEAR after Incoming Call (GATE to CTCP through the Control Session)

The CLEAR command can be issued after the incoming call is sent from GATE to the CTCP through the control session. A Clear Indication clears an unacknowledged incoming call from the network. The command is sent in the following format:

```
    ┌────┬───────┬──────┬──//──┐
RU: │ cc │ resid │ crdf │ user │
    └────┴───────┴──────┴──//──┘
```

| Bytes | Field | Description |
|---|---|---|
| 0 | cc | Command code. For this command, the value is X'13'. |
| 1–2 | resid | Resource identifier. The leftmost halfbyte is set to X'F'. |
| 3–4 | crdf | CAUSE and DIAGNOSTIC fields from the Clear Request packet. |
| 5–n | user | Variable-size field containing optional user data found in the Clear Indication packet after the CAUSE and DIAGNOSTIC fields. |

## CLEAR on Outgoing Call (GATE to CTCP through the Control Session)

The CLEAR command on the outgoing call is issued from GATE to the CTCP through the control session, if an outgoing call cannot be completed. The command is sent in the following format:

```
    ┌─────┬──────┬──────┬─//──┐
RU: │ cc  │ cnid │ crdf │ user│
    └─────┴──────┴──────┴─//──┘
```

| Bytes | Field | Description |
|---|---|---|
| 0 | cc | Command code. For this command, the value is X'13'. |
| 1–2 | cnid | Connection identifier, as specified in the CALL REQUEST command. The leftmost halfbyte is 0. |
| 3–4 | crdf | CAUSE and DIAGNOSTIC fields from the Clear Request packet. |
| 5–n | user | Variable-size field containing optional user data found in the Clear Indication packet after the CAUSE and DIAGNOSTIC fields. |

**Note:** If the NPSI timer elapses on a call request, NPSI sends a Clear packet to the PSDN and sends this CLEAR command to the CTCP.

## CLEAR (GATE to CTCP, CTCP to GATE through the Data Session)

This CLEAR command is issued from GATE to the CTCP and from the CTCP to GATE through the data session. The command is sent in the following format:

```
    ┌─────┬──────┬─//──┐
RU: │ cc  │ ccdf │ user│
    └─────┴──────┴─//──┘
```

| Bytes | Field | Description |
|---|---|---|
| 0 | cc | Command code. For this command, the value is X'13'. |
| 1–2 | ccdf | Clear CAUSE and DIAGNOSTIC fields. |
| 3–n | user | Variable-size field containing optional user data following the CAUSE and DIAGNOSTIC fields. |

## CLEAR CONFIRMATION after CLEAR Command for Incoming Call Refused (GATE to CTCP through the Control Session)

The CLEAR CONFIRMATION command is issued after the CLEAR command for the incoming call that is refused. The command is issued from GATE to the CTCP through the control session. The command is sent in the following format:

```
     ┌──────┬────────┬──//──┐
RU:  │  cc  │ resid  │ user │
     └──────┴────────┴──//──┘
```

| Bytes | Field | Description |
|---|---|---|
| 0 | cc | Command code. For this command, the value is X'17'. |
| 1–2 | resid | Resource identifier, as found in the CLEAR command on an incoming call. The leftmost halfbyte is set to X'F'. |
| 3–n | user | Variable-size field containing any optional user data found in the DCE Clear Confirmation packet. |

## CLEAR CONFIRMATION (GATE to CTCP through the Data Session)

The CLEAR CONFIRMATION command is issued from GATE to the CTCP through the data session. The command is sent in the following format:

```
     ┌──────┬──//──┐
RU:  │  cc  │ user │
     └──────┴──//──┘
```

| Bytes | Field | Description |
|---|---|---|
| 0 | cc | Command code. For this command, the value is X'17'. |
| 1–n | user | Variable-size field containing any optional user data found in the DCE Clear Confirmation packet. |

## RESET (GATE to CTCP, CTCP to GATE through the Data Session)

The RESET command is issued from GATE to the CTCP and from the CTCP to GATE through the data session. The command is sent in the following format:

```
         ┌────┬──────┬──//──┐
RU:      │ cc │ rcdf │ user │
         └────┴──────┴──//──┘
```

| Bytes | Field | Description |
|---|---|---|
| 0 | cc | Command code. For this command, the value is X'1B'. |
| 1–2 | rcdf | Reset CAUSE and DIAGNOSTIC fields. |
| 3–n | user | Variable-size field containing optional user data following the CAUSE and DIAGNOSTIC fields. |

If NPSI detects an unrecoverable situation, GATE sends a RESET command to the CTCP and resets the virtual circuit according to the type of error, as shown in the following table.

| Type of Error | Reset CMD to CTCP | Reset Req to Network |
|---|---|---|
| Inconsistent Q-bit setting | Cause X'00' Diag X'53' | Diag for index X'53' |
| T25 timeout | Cause X'00' Diag X'92' | Diag for index X'92' |
| Timeout after received RNR | Cause X'00' Diag X'94' | Diag for index X'94' |
| Received packet longer than maximum packet size | Cause X'00' Diag X'A7' | Diag for index X'A7' |
| Packet received that is not valid P(S) | Cause X'00' Diag X'AB' | Diag for index X'AB' |
| Packet received that is not valid P(R) | Cause X'00' Diag X'AC' | Diag for index X'AC' |
| Packet received that is not valid D bit | Cause X'00' Diag X'AD' | Diag for index X'AD' |
| Chain canceled by APPL | Cause X'00' Diag X'B3' | Diag for index X'B3' |

**Note:** See *NPSI Diagnosis, Customization, and Tuning* for more information about NPSI Cause and Diagnostic codes.

## RESET because PVC Not in SNA Session (GATE to CTCP through the Control Session)

A RESET command is issued from GATE to the CTCP when a Reset packet is received and the PVC is not in session with the host. This RESET command is issued through the control session and is sent in the following format:

```
    ┌────┬───────┬──────┬──//──┐
RU: │ cc │ resid │ rcdf │ user │
    └────┴───────┴──────┴──//──┘
```

| Bytes | Field | Description |
|---|---|---|
| 0 | cc | Command code. For this command, the value is X'1B'. |
| 1–2 | resid | Resource identifier. The leftmost halfbyte is set to X'0'. |
| 3–4 | rcdf | Reset CAUSE and DIAGNOSTIC fields. |
| 5–n | user | Variable-size field containing optional user data. |

The network has sent a Reset packet on the indicated PVC that is not yet in session. The CTCP might want to establish a session using this PVC. NPSI sends the RESET confirmation automatically; however, the CTCP must not send the RESET confirmation to NPSI.

## RESET CONFIRMATION (GATE to CTCP through the Data Session)

The RESET CONFIRMATION command is issued from GATE to the CTCP through the data session. The command is sent in the following format:

```
    ┌────┬──//──┐
RU: │ cc │ user │
    └────┴──//──┘
```

| Bytes | Field | Description |
|---|---|---|
| 0 | cc | Command code. For this command, the value is X'1F'. |
| 1–n | user | Variable-size field containing any optional user data found in the DCE Reset Confirmation packet. |

## INTERRUPT (CTCP to GATE, GATE to CTCP through the Data Session)

The INTERRUPT command is issued from the CTCP to GATE and from GATE to the CTCP through the data session. The command is sent in the following format:

```
    ┌────┬────┬──//──┐
RU: │ cc │ iu │ user │
    └────┴────┴──//──┘
```

| Bytes | Field | Description |
|---|---|---|
| 0 | cc | Command code. For this command, the value is X'23'. |
| 1 | iu | Interrupt user data byte. |
| 2–n | user | Variable-size field containing optional user data following the interrupt user data byte. |

## INTERRUPT CONFIRMATION (CTCP to GATE, GATE to CTCP through the Data Session)

The INTERRUPT CONFIRMATION command is issued from the CTCP to GATE and from GATE to the CTCP through the data session. The command is sent in the following format:

```
     ┌─────┬───//──┐
RU:  │ cc  │ user  │
     └─────┴───//──┘
```

| Bytes | Field | Description |
|---|---|---|
| 0 | cc | Command code. For this command, the value is X'27'. |
| 1–n | user | Variable-size field containing optional user data following the packet header. |

## DIAGNOSTIC (GATE to CTCP through the Control Session)

The DIAGNOSTIC command is issued from GATE to the CTCP through the control session. The command is sent in the following format:

```
     ┌─────┬─────┬─//─┐
RU:  │ cc  │ dc  │ od │
     └─────┴─────┴─//─┘
```

| Bytes | Field | Description |
|---|---|---|
| 0 | cc | Command code. For this command, the value is X'F1'. |
| 1 | dc | Diagnostic code. |
| 2–n | od | Variable-size field containing diagnostic explanation. |

## MLUNAME (CTCP to GATE through the Control Session)

The MLUNAME (MCH LU name) command is issued from the CTCP to the GATE through the control session. This command passes the MCH LU name to GATE when the user data portion of the BIND command does not carry it. The command is sent in the following format:

```
     ┌─────┬─────────┐
RU:  │ cc  │ mchslu  │
     └─────┴─────────┘
```

| Bytes | Field | Description |
|---|---|---|
| 0 | cc | Command code. For this command, the value is X'F4'. |
| 1-n | mchslu | Variable-size field (up to 8 bytes) containing the MCH LU name. |

## ERROR/INFORMATION REPORT (GATE to CTCP through the Control Session or Data Session)

The ERROR/INFORMATION REPORT command is issued from GATE to the CTCP through the control session for a CLEAR command sent on the incoming call. Otherwise, the command is sent through the data session. The command is sent in the following format:

| | | | | |
|---|---|---|---|---|
| RU: | cc | resid | ce | ec |

| Bytes | Field | Description |
|---|---|---|
| 0 | cc | Command code. For this command, the value is X'FF'. |
| 1–2 | resid | Resource identifier. The leftmost halfbyte is set to X'0'. |
| 3 | ce | Command code for the command in error. |
| 4 | ec | Error cause with a value of X'30'. Timer elapsed before a response from the network was received for the last control packet. All other values indicate the low order sense byte returned for the command issued. |

**Note:** No ERROR/INFORMATION REPORT is sent to the CTCP if the NPSI timer elapses on a call request. Instead, a CLEAR command is sent to the CTCP, and a Clear packet is sent to the PSDN.

**Notes:**

1. When control commands from the CTCP to GATE contain optional user data, such data is put into the resulting control packet without being checked by GATE. The M bit is not used with control packets. For this reason, all user data must fit into a single packet.

2. When NPSI receives a RESET from the network, NPSI responds immediately with either a RESET CONFIRMATION or a RESET REQUEST. This is the reason why the CTCP does not have to send a RESET CONFIRMATION to GATE.

   When NPSI receives a CLEAR from the network, NPSI responds with a CLEAR CONFIRMATION at ABCONN time from VTAM. This is to avoid having the next call arrive while the previous session is closing down. The CTCP does not send a CLEAR CONFIRMATION to GATE.

   Although the CTCP cannot send the CLEAR CONFIRMATION command or the RESET CONFIRMATION command, the CTCP must be able to receive these commands.

3. To prevent returning a negative SNA response on the CTCP to MCH-LU session, the keyword 'GATEINFO' must be coded on the X25.MCH statement.

## Interfacing with Multiple CTCPs

If more than one CTCP is specified in an X25.MCH statement, the MCH has more than one LU. The CTCPs can reside in the same host or in different hosts. Each MCH LU is in session with a single CTCP. Once a CTCP to MCH LU session is established, NPSI knows the CTCP application name and places it in the logon command that is generated by NPSI when a call-in or call-out occurs on a virtual circuit.

The selection of the CTCP is based on subaddressing or on the value of the first byte of the CALL USER DATA (CUD) field. If subaddressing is used and the subaddressing digit is found, only CTCP 0 can be chosen. If the subaddressing digit is *not* found, CUD0 is used. See *NPSI Planning and Installation* for more information about CTCP selection.

**V3R4 and Later:** A third CTCP selection exists based on the value of the last byte of the Called Address Extension Facility (AEF).

Once the appropriate MCH LU and CTCP are selected, the interface between these two LUs is the same as the interface for GATE with a single CTCP. If several CTCPs can be accessed from a given MCH, all virtual circuit LUs that can be associated with this MCH must be defined in all the CTCPs, if the CTCPs require the definition of the LUs that go in session with them. In the case of GTMOSI, for example, all virtual circuit LUs associated with an MCH must be defined in each generation of GTMOSI related to this MCH.

## Fast Connect Using GATE

Fast connect (FC) is an option of the NPSI GATE function. FC is used only for SVC connected to non-SNA DTEs.

FC operates similarly to GATE. With FC, X.25 switched resources are dynamically connected to preestablished SNA sessions with the host application.

The reduction in session setup time results in a faster connection to the host. This faster connection is useful when connections are characterized by:

- Short connection duration
- Small data size
- Quick response time for connection establishment
- Heavy calling and clearing rate

VTAM is not aware that the SNA sessions are carrying data from different virtual circuits, because only NPSI is involved in the mapping of the virtual circuit to the SNA session.

Specifying the FC option allows for the creation of a number of SNA resources and a *different* number of X.25 resources. You can create a pool of available SNA resources that can be dynamically connected to different X.25 resources on demand. Virtual calls can be directed to different CTCPs, depending on the amount of traffic at the time.

FC supports only the logmode table entry shown in Figure 3-1.

```
FAST1   MODEENT   LOGMODE=FAST,
                  TSPROF=X'03',
                  FMPROF=X'03',
                  PRIPROT=X'B0',  (- multi RU chain from PLU   )
                                  (- definite or exception resp)
                                  (- primary will not send EB  )
                  SECPROT=X'B0',  (- multi RU chain from PLU   )
                                  (- definite or exception resp)
                                  (- secondary will not send EB)
                  COMPROT=X'0040' (- bracket not used          )
                                  (- half-duplex contention    )
```

*Figure 3-1. Logmode Table Entry Supported by FC*

Figure 3-2 on page 3-20 shows the data flow for a network configuration that uses FC GATE.

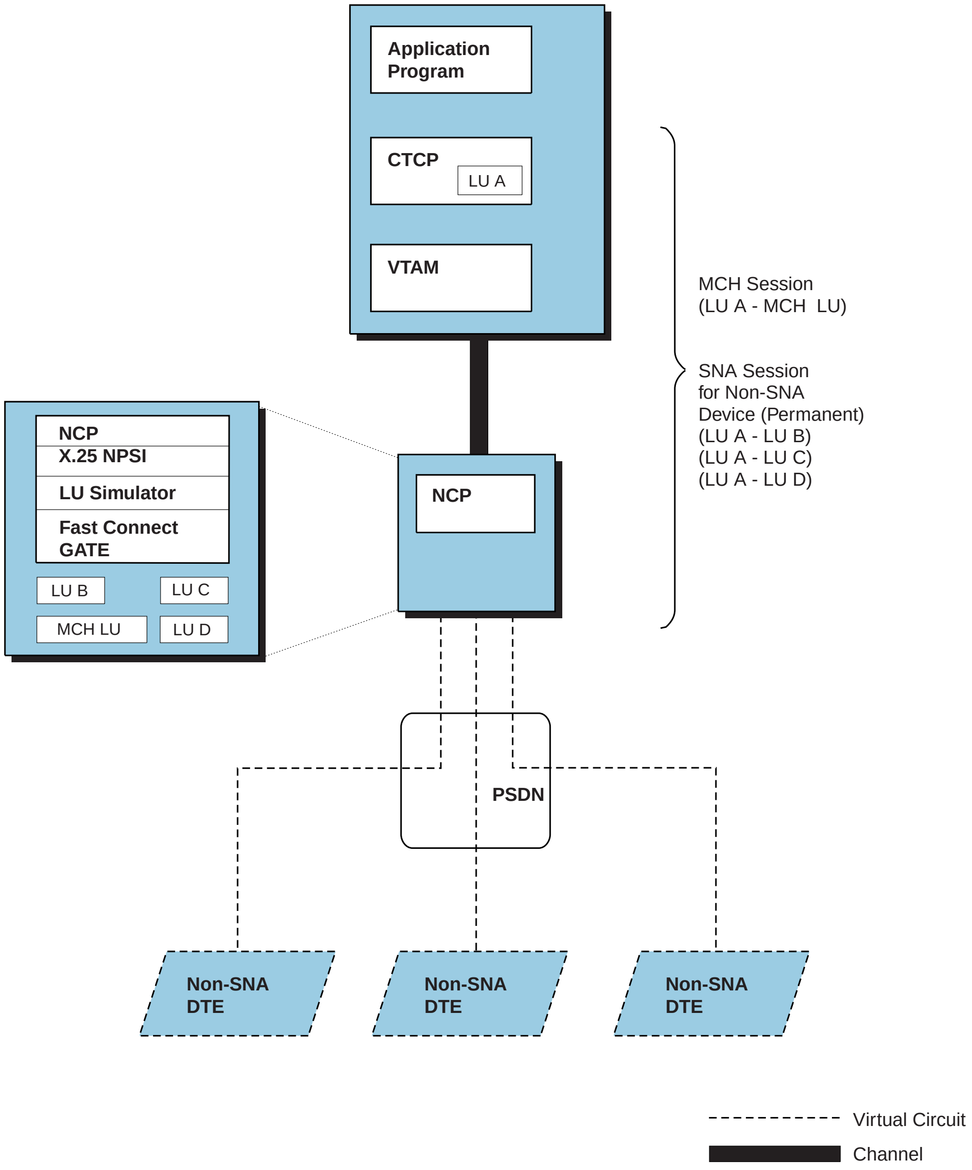


*Figure 3-2. SNA Host Node to Non-SNA DTE (FC GATE)*

## Call-In (FC GATE to CTCP)

When an incoming call is received from the PSDN, FC GATE sends an INCOMING CALL command to the host processor through a CTCP LU to the virtual circuit LU session. This processing differs from non-FC GATE processing, in which CALL commands are sent on the CTCP LU to the MCH LU session. This difference is caused by a high calling rate that can occur with the FC option. If this traffic was placed on the CTCP LU to MCH LU session, a session bottleneck for setup could occur. Placing the control requests on the CTCP LU to virtual circuit LU session spreads the control requests across more sessions, thereby reducing the delay.

NPSI uses either byte 0 of the incoming CUD field or the subaddressing digit(s) to determine the desired CTCP and to select the pool of SNA resources connected to this CTCP.

NPSI chooses an available resource in the selected pool to place the incoming call and allocates that SNA resource to the calling virtual circuit. NPSI then associates the SNA resource and the X.25 resource.

Once the virtual circuit is established, fast connect operates similarly to the GATE interface. The call operation commands do not change when you use the FC option. (See “Command Interface between FC and the CTCP” on page 3-26 for the exact format of the commands passed between the CTCP and FC GATE operating in NPSI.)

If the CTCP does not accept the call, the CTCP returns a CLEAR command that carries the resource identifier of the call to FC GATE. FC transforms the CLEAR command into a Clear packet and forwards the packet to the PSDN along with the CAUSE and DIAGNOSTIC fields specified by the CTCP. See *NPSI Diagnosis, Customization, and Tuning* for a list of supported cause and diagnostic codes.

## Call-Out (CTCP to FC GATE)

When an FC GATE CTCP performs a call-out procedure, the CTCP sends a CALL REQUEST command on a CTCP to the virtual circuit LU session.

If a CTCP tries to send a CALL REQUEST command to the LU on the physical circuit, fast connect GATE returns an X'0801' sense code (resource not available). This negative response tells the CTCP that it is communicating with NPSI, rather than with the FC request for price quote (RPQ).

When receiving a CALL REQUEST command from an FC CTCP, FC GATE selects a free X.25 resource, dynamically links it to an SNA resource, and uses the information provided in the command to create a Call Request packet that is sent to the PSDN.

The PSDN acknowledges the Call Request packet with a Call Connected packet. FC GATE returns a CALL CONNECTED command to the CTCP. The resource identifier in the CALL CONNECTED command is used by the CTCP and FC GATE for subsequent commands and data exchange.

If the PSDN responds to a Call Request with a Clear packet, a CLEAR command is sent to the CTCP. The CLEAR command contains a clearing cause and diagnostic byte.

If a response to the Call Request is not received within the duration of the X.25 T21 timer, NPSI sends a CLEAR command (built by FC GATE) to the network and then to the CTCP through the data session.

## Call Collision

Figure 3-3 on page 3-22 shows the order of virtual circuit and SNA resource assignment for call-in and call-out.

If contention occurs, and depending on the call that NPSI receives last, NPSI performs one of the following actions:

- Discards Incoming Call packets from the PSDN using the same virtual circuit as the call-out session.
- Responds negatively to a CALL REQUEST command from the CTCP using the same SNA resource ID as the call-in session.

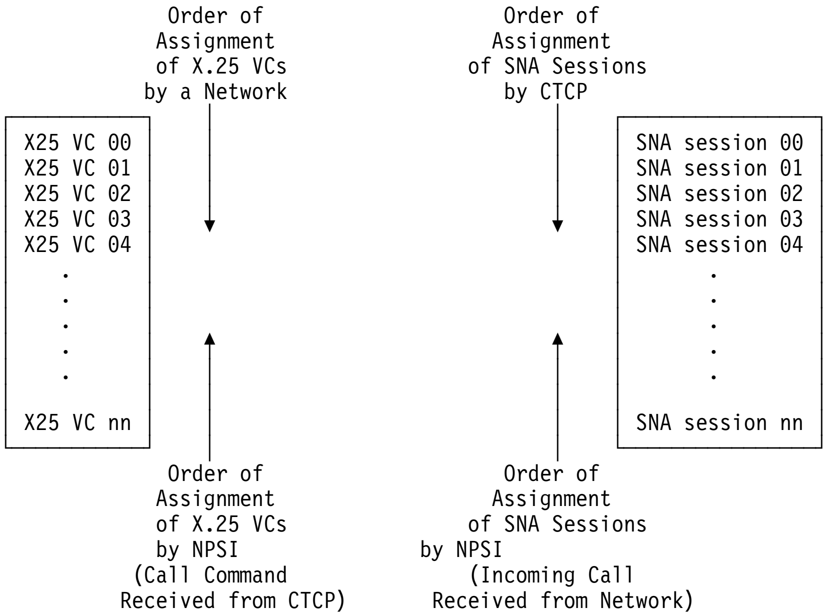

Order of Assignment of X.25 VCs by a Network
Order of Assignment of SNA Sessions by CTCP
X25 VC 00
X25 VC 01
X25 VC 02
X25 VC 03
X25 VC 04
X25 VC nn
SNA session 00
SNA session 01
SNA session 02
SNA session 03
SNA session 04
SNA session nn
Order of Assignment of X.25 VCs by NPSI (Call Command Received from CTCP)
Order of Assignment of SNA Sessions by NPSI (Incoming Call Received from Network)

*Figure 3-3. Virtual Circuit and SNA Resource Assignment Order*

## FC MCH Reinitialization Consideration

When an MCH reinitialization occurs, the FC SNA resources are not changed to INOP status. For SNA resources engaged in an active call, a CLEAR command with a diagnostic code of X'30' is sent to the CTCP. For SNA resources not engaged in an active call, no action is taken.

## FC and CTCP Interface During Virtual Circuit Connection

Once an X.25 connection is established, data and commands (for example, RESET and INTERRUPT) can be exchanged between FC GATE and a CTCP.

The data exchanged can be either qualified or unqualified. Qualified data is used to communicate with a PAD; unqualified data is used to communicate with an end-user. An example of qualified (control) data is a packet containing a PAD command.

If the data is specified as qualified, the Q bit is on in the X.25 packet. The data type is communicated to FC GATE through the use of the first byte of the RU. The distinction is:

| **RU Byte 0** | **Usage** |
|---|---|
| X'00' | Unqualified data |
| X'02' | Qualified data |

## NPSI FC Connection Termination

The CLEAR command terminates NPSI communication. The CLEAR command is processed differently by NPSI depending on whether a normal session termination is possible or a collision must be resolved.

## Normal Clearing

An X.25 communication can end either on a CTCP request (a CLEAR command) or on request from either the PSDN or the terminal (a Clear Indication packet).

CLEAR commands (X'13') flow between the CTCP and FC GATE on the CTCP LU to virtual circuit LU session.

After a CLEAR command exchange, the X.25 virtual circuit is freed by the PSDN. However, because the associated SNA resource remains bound and available for a subsequent call, NPSI discards any unexpected packet (except an Incoming Call packet) received from the PSDN. Requests from the CTCP, with the exception of CALL REQUEST commands, are rejected with an SNA sense code of X'081C'.

When a Clear packet is received from the PSDN, FC GATE sends a Clear Confirmation packet to the network and a Clear command to the CTCP. The SNA resource set and the X.25 VC are immediately available for the next call.

If you request the billing record option by specifying the TAXUNIT keyword of the X25.MCH statement, the CLEAR or CLEAR CONFIRMATION command returned to the CTCP contains billing information. The TAXUNIT keyword specifies the size of the X.25 accounting unit.

The billing function of fast connect GATE provides the CTCP with the option of obtaining the following information:

- Number of X.25 accounting units
- User session start time and date
- User session end time and date

The billing information can be used by the CTCP or an auxiliary program to assist in utilization tracking, accounting, and other user requirements. When you request the billing option, billing information is appended to the last command (CLEAR, CLEAR CONFIRMATION, or ERROR/INFORMATION REPORT) sent by NPSI to a CTCP. The RU of the last command is sent in the format shown in Figure 3-4.

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| CLEAR,<br>CLEAR<br>CONFIRM,<br>or ERROR/<br>INFO. RPT.<br>as<br>Received<br>from<br>Network | ( | Session<br>Start<br>Time<br>and<br>Date | Session<br>End<br>Time<br>and<br>Date | Re-<br>served | Number<br>of<br>X.25<br>Accounting<br>Units<br>Sent | Number<br>of<br>X.25<br>Acct.<br>Units<br>Rcvd. | ( |

*Figure 3-4. Format of the CLEAR, CLEAR CONFIRMATION, and ERROR/INFORMATION REPORT Commands with the Billing Option*

**Notes:**

1. The two left parentheses are delimiters and can be used by the CTCP to locate the billing information.
2. An X.25 unit is the network billing unit as defined in the TAXUNIT keyword of the X25.MCH statement at NPSI generation.

The CLEAR command is sent in the following format:

| **Bytes** | **Value** |
|---|---|
| 0 | X'13' |
| 1 through 2 | Resource ID |
| 3 through 4 | Cause and diagnostic fields |
| 5 through n | User data |
| n+1 | ( |
| n+2 through n+21 | Session start time and date |
| n+22 through n+41 | Session end time and date |
| n+42 through n+44 | Reserved |
| n+45 through n+48 | Number of X.25 accounting units sent |
| n+49 through n+52 | Number of X.25 accounting units received |
| n+53 | ( |

The CLEAR CONFIRMATION command is sent in the following format:

| **Bytes** | **Value** |
|---|---|
| 0 | X'17' |
| 1 | ( |
| 2 through 21 | Session start time and date |
| 22 through 41 | Session end time and date |
| 42 through 44 | Reserved |
| 45 through 48 | Number of X.25 accounting units sent |
| 49 through 52 | Number of X.25 accounting units received |
| 53 | ( |

The ERROR/INFORMATION REPORT command is sent in the following format:

| **Bytes** | **Value** |
|---|---|
| 0 | X'FF' |
| 1 through 2 | Resource identifier |
| 3 | Command that failed |
| 4 | Error cause |
| 5 | ( |
| 6 through 25 | Session start time and date |
| 26 through 45 | Session end time and date |
| 46 through 48 | Reserved |
| 49 through 52 | Number of X.25 accounting units received |
| 53 through 56 | Number of X.25 accounting units sent |
| 57 | ( |

The billing information is sent to the CTCP in the ERROR/INFORMATION REPORT command when NPSI does not receive a response to a CLEAR command within the value set for the T23 timer (CLEAR timeout).

No billing information is provided, even if it is requested, when the following occurs:

- Incoming or outgoing call is cleared before confirmation is received.
- LU-LU session is disrupted.
- MCH experiences a failure.

### CLEAR Collision

CLEAR collision is detected in NPSI when either:

- A CLEAR command is received while a Clear packet is processed.
- A Clear packet is received while a CLEAR command is processed.

If a CLEAR command collides with a previously received Clear Indication packet from the PSDN, FC GATE rejects the CLEAR command with an SNA sense code of X'081C' (function not executable).

When NPSI receives a Clear Request packet from the PSDN, before receiving the Clear Confirmation packet for a previously sent CLEAR command, NPSI sends a CLEAR command to the CTCP to confirm the Clear Request packet.

## FC NOGO and GO Command Processing (V3R4 and Later)

NOGO and GO command processing enhances the inbound flow control for FC. The FC CTCP issues NOGO commands to NPSI on appropriate virtual circuit LU sessions when a problem is detected at the host, such as a buffer shortage. NPSI then stops the flow of incoming data for these sessions. The FC CTCP must send GO commands to NPSI on the virtual circuit LU sessions to signal that it is ready to process additional data again.

A virtual circuit enters the NOGO state when its LU has received a NOGO command from the FC CTCP. NPSI queues the data packets received from the network destined for that virtual circuit LU session. In addition, if there are more than four PIUs queued on the inbound queue, NPSI stops packet acknowledgment in an effort to stop inbound packets.

**Note:** Once the CTCP has issued a NOGO command, NPSI stops the inbound flow of PIUs. NPSI still accepts outbound PIUs, transforming them and sending the packets to the network.

If you are a remote DTE operator sending data when NPSI is in the NOGO state, your request can remain unanswered, or you may not be able to enter additional data. You can correct the problem by pressing the Break key to generate an Interrupt or a Reset packet. NPSI forwards these packets in the NOGO state to the CTCP, which takes action, such as canceling the current transaction.

When an X.25 connection linked to an SNA session in the NOGO state is cleared, NPSI reestablishes the GO state of this session without requiring the CTCP to issue the GO command.

**V3R3 Only:** NPSI responds to the NOGO and GO commands with the sense code X'080C' (function not supported).

Table 3-2 shows the effects of receiving X.25 packets when NPSI is in the NOGO state.

*Table 3-2. NPSI Behavior in NOGO State*

| Command code | Session with CTCP |
|---|---|
| CLEAR Command | The CLEAR command is passed from the CTCP to GATE or from GATE to the CTCP through the data session, purging all data packets on the inbound queue. The NOGO state is reset when the SNA resource is freed. |
| CLEAR CONFIRMATION Command | The CLEAR CONFIRMATION command is passed from GATE to the CTCP through the data session, purging all data packets on the inbound queue, and resetting the NOGO state. |
| DATA Command | When NPSI receives data from the CTCP, if inbound data packets are queued because of the NOGO state, NPSI does not update the P(R) in the outbound data packets to keep the window from being reopened. |
| Data Packets | Data packets from NPSI remain queued on the NPSI inbound queue. Currently, if more than four PIUs are on the inbound queue, data packet acknowledgment is stopped to avoid receiving more data packets from the network. |
| INTERRUPT CONFIRMATION Command | The INTERRUPT CONFIRMATION command is passed from the CTCP to GATE or from GATE to the CTCP through the data session. The NPSI inbound queue is not purged. |
| INTERRUPT Command | The INTERRUPT command is passed from the CTCP to GATE or from GATE to the CTCP through the data session. The NPSI inbound queue is not purged. |
| MCH Failure | When an MCH failure occurs, a pseudo CLEAR command is passed to the CTCP, and all data packets are purged from the inbound queue. |

**Note:** RESET processing is the same in both the NOGO and GO state.

## Command Interface between FC and the CTCP

In the command descriptions in Table 3-3 on page 3-27, the RESID field in the RU format that corresponds to the resource identifier designates either the set of SNA resources (LINE, PU, and LU) that are associated with a virtual circuit, or the X.25 VC number (systems generation option VCID). SNA resource identifiers are associated with virtual circuits for each MCH. The range for the resource identifier numbers is the same as the range for the virtual circuit numbers, if there is only one CTCP for each MCH. Otherwise, the range is defined by the X25.FCG statements.

An SNA resource number consists of three digits in hexadecimal format. The left-most halfbyte maps the logical channel group number and the next two digits map the logical channel number. The following example shows how you can code the X25.MCH statement to define SVCs. In this example, SNA IDs X'003' through X'063' and SNA IDs X'204' through X'262' are used to map all SVCs of this MCH. This setup allows CTCPs to continue to run without change.

```
X25.LCG  LCGN=0
X25.VC   LCN=(1,2),TYPE=P ....  2 PVCs
X25.VC   LCN=(3,99),TYPE=S ... 97 SVCs
X25.LCG  LCGN=2
X25.VC   LCN=(1,3),TYPE=P ....  3 PVCs
X25.VC   LCN=(4,98),TYPE=S ... 95 SVCs
```

Table 3-3 lists the commands recognized or generated by FC GATE.

*Table 3-3. CTCP-FC GATE Control Commands*

| **Command code** | **Command** | **Session with CTCP** |
|---|---|---|
| 00 | DATA EXCHANGE (Q bit set to 0) | Data Session |
| 02 | DATA EXCHANGE (Q bit set to 1) | Data Session |
| 0B | INCOMING CALL/CALL REQUEST | Data Session |
| 0F | CALL CONNECTED/CALL ACCEPTED | Data Session |
| 13 | CLEAR indication/CLEAR request | Data Session |
| 17 | CLEAR CONFIRMATION | Data Session |
| 1B | RESET | Data Session |
| 1F | RESET CONFIRMATION | Data Session |
| 23 | INTERRUPT | Data Session |
| 27 | INTERRUPT CONFIRMATION | Data Session |
| F1 | DIAGNOSTIC | Control Session |
| F2 | NOGO | Data Session |
| F3 | GO | Data Session |
| FF | INFORMATION REPORT Message | Data Session |

The following sections describe the DATA EXCHANGE commands and control commands. The DATA EXCHANGE commands are followed by user data. The control commands are followed by additional control information.

## DATA EXCHANGE without Q Bit (CTCP to FC GATE, FC GATE to CTCP through the Data Session)

A DATA EXCHANGE command, with the Q bit set off, is issued from the CTCP to GATE and from GATE to the CTCP through the data session. The command is sent in the following format:

```
    ┌────┬───//──┐
RU: │ cc │  user  │
    └────┴───//──┘
```

| Bytes | Field | Description |
|---|---|---|
| 0 | cc | Command code. For this command, the value is X'00'. |
| 1–n | user | Variable-size field containing user data following the packet header. |

**Note:** The command byte is set only in FIC and OIC RUs.

## DATA EXCHANGE with Q Bit (CTCP to FC GATE, FC GATE to CTCP through the Data Session)

A DATA EXCHANGE command, with the Q bit set on, is issued from the CTCP to GATE and from GATE to the CTCP through the data session. The command is sent in the following format:

```
    ┌────┬───//──┐
RU: │ cc │  user  │
    └────┴───//──┘
```

| Bytes | Field | Description |
|---|---|---|
| 0 | cc | Command code. For this command, the value is X'02'. |
| 1–n | user | Variable-size field containing user data following the packet header. |

**Note:** The command byte is set only in FIC and OIC RUs.

## INCOMING CALL (FC GATE to CTCP through the Data Session)

The INCOMING CALL command is issued from GATE to the CTCP through the data session. The command is sent in the following format:

```
    ┌────┬───────┬───//──┐
RU: │ cc │ resid │  icpk  │
    └────┴───────┴───//──┘
```

| Bytes | Field | Description |
|---|---|---|
| 0 | cc | Command code. For this command, the value is X'0B'. |
| 1–2 | resid | Resource identifier. The leftmost halfbyte is set to X'F'. |
| 3–n | icpk | Variable-size field containing the Incoming Call packet received from the network (without the 3-byte packet header). |

## CALL REQUEST (CTCP to FC GATE through the Data Session)

The CALL REQUEST command is issued from the CTCP to GATE through the data session. The command is sent in the following format:

```
    ┌────┬──────┬────┬──────┬──//──┐
RU: │ cc │ cnid │ pw │ psiz │ crpk │
    └────┴──────┴────┴──────┴──//──┘
```

| Bytes | Field | Description |
|---|---|---|
| 0 | cc | Command code. For this command, the value is X'0B'. |
| 1–2 | cnid | Unique connection identifier determined by the CTCP. This 3-digit number is right-aligned. The leftmost halfbyte *must* be X'0'. |
| 3 | pw | Packet window size. |
| 4–5 | psiz | Packet size in hexadecimal used for this virtual circuit. |
| 6–n | crpk | Variable-size field containing the Call Request packet without the packet header. |

**Note:** Packet and window sizes specified in the CALL REQUEST command are recognized by NPSI. The flow control parameters included in the facilities of the Call Connected packet are ignored by NPSI. The CTCP should clear the call and try again if these flow control parameters do not match the flow control parameters that were passed to NPSI in the CALL REQUEST command.

**Note:** The flow control parameters included in the facilities of the Call Connected packet are recognized by NPSI if INTFAC is specified.

**V3R7 and later:** The CTCP might set packet window and packet window size fields to zero to allow NPSI to determine the X.25 flow control values for a virtual call. See “Session Establishment through an SVC” on page 3-2 for information.

## CALL CONNECTED (FC GATE to CTCP through the Data Session)

The CALL CONNECTED command is issued from GATE to the CTCP through the data session. The command is sent in the following format:

```
    ┌────┬──────┬───────┬──//──┐
RU: │ cc │ cnid │ resid │ ccpk │
    └────┴──────┴───────┴──//──┘
```

| Bytes | Field | Description |
|---|---|---|
| 0 | cc | Command code. For this command, the value is X'0F'. |
| 1–2 | cnid | Connection identifier, as specified in the CALL REQUEST command. |
| 3–4 | resid | Resource identifier. The leftmost halfbyte is set to X'F'. |
| 5–n | ccpk | Variable-size field containing optional bytes that followed the packet header of the Call Connected packet. |

**V3R4 and Later:** Depending on the generation option specified on the DEFPRI keyword, the QOS facility, (X'D2') contained in the Call Connected packet, determines the QOS value that is used during the session.

## CALL ACCEPTED (CTCP to FC GATE through the Data Session)

The CALL ACCEPTED command is issued from the CTCP to GATE through the data session. The command is sent in the following format:

```
    ┌────┬───────┬────┬──────┬─//──┐
RU: │ cc │ resid │ pw │ psiz │ capk │
    └────┴───────┴────┴──────┴─//──┘
```

| Bytes | Field | Description |
|---|---|---|
| 0 | cc | Command code. For this command, the value is X'0F'. |
| 1–2 | resid | Resource identifier, as found in the INCOMING CALL command. |
| 3 | pw | Packet window size. |
| 4–5 | psiz | Packet size in hexadecimal used for this virtual circuit. |
| 6–n | capk | Variable-size field containing the Call Accepted packet to be sent to the network without the 3-byte packet header. |

**V3R4 and Later:** Depending on the generation option specified on the DEFPRI keyword, the QOS facility, (X'D2') contained in the Call Accepted packet, determines the quality of service value that is used during the session.

**V3R7 and later:** The CTCP might set packet window and packet window size fields to zero to allow NPSI to determine the X.25 flow control values for a virtual call. See “Session Establishment through an SVC” on page 3-2 for information.

## CLEAR on Incoming Call (CTCP to FC GATE through the Data Session)

The CLEAR command to clear an incoming call is issued from the CTCP to GATE through the data session. The command is sent in the following format:

```
    ┌────┬───────┬──────┬─//──┐
RU: │ cc │ resid │ crdf │ user │
    └────┴───────┴──────┴─//──┘
```

| Bytes | Field | Description |
|---|---|---|
| 0 | cc | Command code. For this command, the value is X'13'. |
| 1–2 | resid | Resource identifier, as found in the Incoming call command. |
| 3–4 | crdf | CAUSE and DIAGNOSTIC fields that are to be inserted in the Clear Request packet. |
| 5–n | user | Variable-size field containing optional user data to be put in the Clear Request packet after the CAUSE and DIAGNOSTIC fields. |

## CLEAR (FC GATE to CTCP through the Data Session)

The CLEAR command is issued from GATE to the CTCP through the data session. The command is sent in the following format:

```
    ┌────┬───────┬────┬─//──┬─//─────┐
RU: │ cc │ resid │ccdf│ user │ billing │
    └────┴───────┴────┴─//──┴─//─────┘
```

| Bytes | Field | Description |
|---|---|---|
| 0 | cc | Command code. For this command, the value is X'13'. |
| 1–2 | resid | Resource identifier, as specified in the INCOMING CALL or CALL CONNECTED command. The leftmost halfbyte is X'F' for the direction GATE to CTCP, and X'0' for the direction CTCP to GATE. |
| 3–4 | ccdf | Clear CAUSE and DIAGNOSTIC fields. |
| 5–n | user | Variable-size field containing optional user data following the CAUSE and DIAGNOSTIC fields. |
| n–m | billing | Variable-size field containing billing information, if the billing option is requested at NPSI generation. See “Normal Clearing” on page 3-23 for billing format details. |

**Note:** When the CLEAR command is sent by GATE to the CTCP, a definite response is requested.

## CLEAR (CTCP to FC GATE through the Data Session)

The CLEAR command is issued from the CTCP to GATE through the data session. The command is sent in the following format:

```
     ┌────┬──────┬──//──┐
RU:  │ cc │ ccdf │ user │
     └────┴──────┴──//──┘
```

| Bytes | Field | Description |
|---|---|---|
| 0 | cc | Command code. For this command, the value is X'13'. |
| 1–2 | ccdf | CAUSE and DIAGNOSTIC fields that are to be inserted in the Clear Request packet. |
| 3–n | user | Variable-size field containing optional user data to be put in the Clear Request packet after the CAUSE and DIAGNOSTIC fields. |

## CLEAR on Call Request (FC GATE to CTCP through the Data Session)

The CLEAR command on the Call Request is issued from GATE to the CTCP through the data session. The command is sent in the following format:

```
    ┌────┬──────┬──────┬─//─┐
RU: │ cc │ cnid │ crdf │user│
    └────┴──────┴──────┴─//─┘
```

| Bytes | Field | Description |
|---|---|---|
| 0 | cc | Command code. For this command, the value is X'13'. |
| 1–2 | cnid | Connection identifier, as specified in the CALL REQUEST command. The leftmost halfbyte is X'0'. |
| 3–4 | crdf | CAUSE and DIAGNOSTIC fields from the Clear packet. |
| 5–n | user | Variable-size field containing optional user data found in the Clear packet after the CAUSE and DIAGNOSTIC fields. |

**Note:** This CLEAR command can be received by the CTCP either when the network clears the outgoing call or if a call confirmation is not received from the network within the value set for the T23 timer (X.25 clear timeout).

## CLEAR CONFIRMATION after CLEAR Command for Incoming Call Refused (FC GATE to CTCP through the Data Session)

The CLEAR CONFIRMATION command is issued after the CLEAR command for the incoming call that is refused. It is issued from GATE to the CTCP through the data session. The command is sent in the following format:

```
    ┌────┬───────┬─//─┐
RU: │ cc │ resid │user│
    └────┴───────┴─//─┘
```

| Bytes | Field | Description |
|---|---|---|
| 0 | cc | Command code. For this command, the value is X'17'. |
| 1–2 | resid | Resource identifier, as found in the CLEAR command on the incoming call. The leftmost halfbyte is set to X'F'. |
| 3–n | user | Variable-size field containing optional user data if present in the DCE Clear Confirmation packet. |

## CLEAR CONFIRMATION (FC GATE to CTCP through the Data Session)

The CLEAR CONFIRMATION command is issued from GATE to the CTCP through the data session. The command is sent in the following format:

```
     ┌────┬───//────┐
RU:  │ cc │ billing │
     └────┴───//────┘
```

| Bytes | Field | Description |
|---|---|---|
| 0 | cc | Command code. For this command, the value is X'17'. |
| 1–n | billing | Variable-size field containing billing information, if the billing option is requested at NPSI generation. See “Normal Clearing” on page 3-23 for billing format details. |

## RESET (FC GATE to CTCP, CTCP to FC GATE through the Data Session)

The RESET command is issued from GATE to the CTCP and from the CTCP to GATE through the data session. The command is sent in the following format:

```
     ┌────┬──────┬──//──┐
RU:  │ cc │ rcdf │ user │
     └────┴──────┴──//──┘
```

| Bytes | Field | Description |
|---|---|---|
| 0 | cc | Command code. For this command, the value is X'1B'. |
| 1–2 | rcdf | Reset CAUSE and DIAGNOSTIC fields. |
| 3–n | user | Variable-size field containing optional user data following the CAUSE and DIAGNOSTIC fields. |

If NPSI detects an unrecoverable situation, GATE sends a RESET command to the CTCP and resets the virtual circuit according to the type of error, as shown in the following table.

| Type of Error | Reset CMD to CTCP | Reset Req to Network |
|---|---|---|
| Inconsistent Q-bit setting | Cause X'00' Diag X'53' | Diag for index X'53' |
| T25 timeout | Cause X'00' Diag X'92' | Diag for index X'92' |
| Timeout after received RNR | Cause X'00' Diag X'94' | Diag for index X'94' |
| Received packet longer than maximum packet size | Cause X'00' Diag X'A7' | Diag for index X'A7' |
| Packet received with not valid P(S) | Cause X'00' Diag X'AB' | Diag for index X'AB' |
| Packet received with not valid P(R) | Cause X'00' Diag X'AC' | Diag for index X'AC' |
| Packet received with not valid D bit | Cause X'00' Diag X'AD' | Diag for index X'AD' |
| Chain canceled by APPL | Cause X'00' Diag X'B3' | Diag for index X'B3' |

**Note:** See *NPSI Diagnosis, Customization, and Tuning* for more information about NPSI Cause and Diagnostic codes.

## RESET CONFIRMATION (FC GATE to CTCP through the Data Session)

The RESET CONFIRMATION command is issued from GATE to the CTCP through the data session. The command is sent in the following format:

```
    ┌────┬──//──┐
RU: │ cc │ user │
    └────┴──//──┘
```

| Bytes | Field | Description |
|---|---|---|
| 0 | cc | Command code. For this command, the value is X'1F'. |
| 1–n | user | Variable-size field containing any optional user data found in the DCE Reset Confirmation packet. |

Although the CTCP cannot send CLEAR CONFIRMATION or RESET CONFIRMATION commands, the CTCP must be able to receive these commands. NPSI automatically sends out Clear Confirmation and Reset Confirmation packets.

## INTERRUPT (CTCP to FC GATE, FC GATE to CTCP through the Data Session)

The INTERRUPT command is issued from the CTCP to GATE and from GATE to the CTCP through the data session. The command is sent in the following format:

```
    ┌────┬────┬──//──┐
RU: │ cc │ iu │ user │
    └────┴────┴──//──┘
```

| Bytes | Field | Description |
|---|---|---|
| 0 | cc | Command code. For this command, the value is X'23'. |
| 1 | iu | Interrupt cause byte. |
| 2–n | user | Variable-size field containing optional user data following the interrupt user data byte. |

## INTERRUPT CONFIRMATION (CTCP to FC GATE, FC GATE to CTCP through the Data Session)

The INTERRUPT CONFIRMATION command is issued from the CTCP to GATE and from GATE to the CTCP through the data session. The command is sent in the following format:

```
    ┌────┬──//──┐
RU: │ cc │ user │
    └────┴──//──┘
```

| Bytes | Field | Description |
|---|---|---|
| 0 | cc | Command code. For this command, the value is X'27'. |
| 1–n | user | Variable-size field containing optional user data following the packet header. |

## DIAGNOSTIC (FC GATE to CTCP through the Control Session)

The DIAGNOSTIC command is issued from GATE to the CTCP through the control session.  The command is sent in the following format:

```
        ┌──────┬──────┬──//─┐
RU:     │  cc  │  dc  │  od │
        └──────┴──────┴──//─┘
```

| Bytes | Field | Description |
|---|---|---|
| 0 | cc | Command code.  For this command, the value is X'F1'. |
| 1 | dc | Diagnostic code. |
| 2–n | od | Variable-size field containing diagnostic explanation. |

## NOGO (CTCP to FC GATE through the Data Session)

The NOGO command is issued from the CTCP to FC GATE in the following format:

```
        ┌──────┐
RU:     │  cc  │
        └──────┘
```

| Bytes | Field | Description |
|---|---|---|
| 0 | cc | 1-byte command code.  For this command, the value is X'F2'. |

## GO (CTCP to FC GATE through the Data Session)

The GO command is issued from the CTCP to GATE in the following format:

```
        ┌──────┐
RU:     │  cc  │
        └──────┘
```

| Bytes | Field | Description |
|---|---|---|
| 0 | cc | 1-byte command code.  For this command, the value is X'F3'. |

## ERROR/INFORMATION REPORT (FC GATE to CTCP through the Data Session)

The ERROR/INFORMATION REPORT command is issued from FC GATE to the CTCP through the data session. The command is sent in the following format:

```
     ┌────┬───────┬────┬────┬───//───┐
RU:  │ cc │ resid │ ce │ ec │ billing│
     └────┴───────┴────┴────┴───//───┘
```

| Bytes | Field | Description |
|---|---|---|
| 0 | cc | Command code. For this command, the value is X'FF'. |
| 1–2 | resid | Resource identifier. |
| 3 | ce | Command code for the command in error. |
| 4 | ec | Error cause. |
| 5–n | billing | Variable-size field containing billing information, if the billing option is requested at NPSI generation. See “Normal Clearing” on page 3-23 for billing format details. |

The only error cause that currently exists is X'30'. The timer elapsed before a response from the network was received for the control packet.

## X.21 Switched Connections Using GATE

NPSI provides a connection to an integrated services digital network (ISDN) over a switched connection. To provide this connection capability, NPSI uses two logical interfaces.

- NPSI uses the standard GATE to CTCP upstream interface.
- NPSI uses a downstream interface between NPSI link access control and the X.21 switched function of either the transmission subsystem (TSS) or, since V3R8, the CSS, to control the X.21 physical line.

**Note:** Only SVCs are supported on the MCH that is dedicated to X.21 connections.

Figure 3-5 on page 3-38 illustrates an X.21 switched connection using an IBM 7820 terminal adapter.

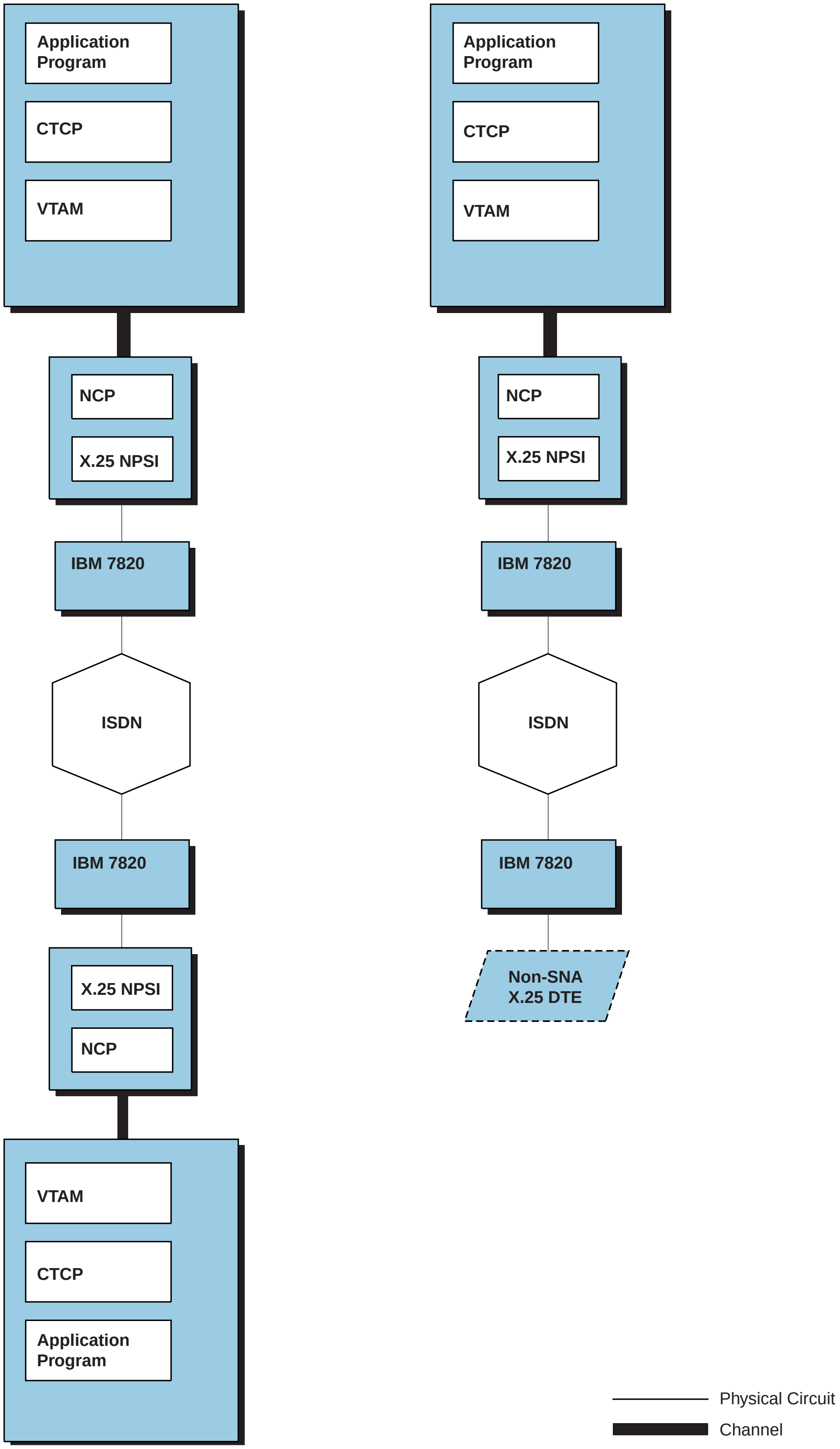


*Figure 3-5. SNA Host to ISDN (X.21 Switched Connection)*

## Session Establishment on X.21 Switched Connections

From the CTCP point of view, session establishment for an X.21 switched connection using GATE is similar to session establishment for a non-X.21 switched connection using GATE. See “Session Establishment through an SVC” on page 3-2 for more information.

First, the CTCP activates the control session with the physical circuit (CTCP to MCH_LU). When the CTCP to MCH_LU session is established, a session between the CTCP and a remote DTE can be established over an SVC in one of the following ways:

1. A CALL REQUEST command from the CTCP

   When it is the first virtual circuit session establishment, you must use a new CALL REQUEST command (with command code `X'8B'`). When it is the next virtual circuit session establishment, you must use the `X'0B'` CALL REQUEST command. See “CALL REQUEST `X'8B'` (CTCP to GATE through the Control Session)” on page 3-43 for more information.

2. An X.25 Incoming Call packet

   The X.25 Incoming Call packet is received from the remote DTE, after the establishment of an X.21 Incoming connection on a physical interface.

## Virtual Circuit Establishment

The following sections describe establishment procedures for X.21 incoming and outgoing calls.

### X.21 Incoming Call

When an X.21 Incoming Call signal is detected, the X.21 physical connection is performed by the TSS or CSS. Only after a successful X.21 connection can NPSI perform the X.25 link level and packet level setup. NPSI then waits for an X.25 Incoming Call packet to be received. When a valid X.25 Incoming Call packet is received, the virtual circuit session is established. If a valid X.25 Incoming Call packet is not received within the time specified on the X21INACT keyword of the X25.MCH statement, the X.21 connection is cleared.

**V3R7 and later:** It is possible to have several CTCPs. NPSI selects the appropriate CTCP using the last byte of the called address extension facility (AEFBYTE sysdef option).

### X.21 Outgoing Call

When the CTCP wants to establish the first session through an SVC, it sends a CALL REQUEST command (`X'8B'`) to GATE, which initiates the X.21 Outgoing Call. The called DTE address, which is contained within the X.25 Call Request packet, contains the X.21 dial digits that are used to perform X.21 outgoing calls. These dial digits represent the destination of the call.

When the X.21 outgoing connection is successful, NPSI performs the X.25 link level and packet level setup.

Only when the link level and packet level are set up does GATE resume the processing of the CALL REQUEST command. The CALL REQUEST command is then processed by GATE as it would be processed on a non-X.21 connection to establish the first virtual circuit session.

Subsequent virtual circuit sessions must be established by CTCP using a X'0B' CALL REQUEST command that has the same called DTE address value as the one set in the X'8B' CALL REQUEST command used to establish the first virtual circuit session.

**V3R7 and later:** To separate the X.21/ISDN address digits from the X.21/ISDN subaddress digits, you can specify a delimiter digit (X'A') in the Called DTE Address field of the X'8B' CALL REQUEST command. When you specify a delimiter digit:

- All the called DTE address digits (including the delimiter digit) are used to perform the X.21 outgoing call. NPSI translates the delimiter into an asterisk (*) character before dialing the number on the X.21 switched interface.
- Only the X.21/ISDN address digits are set by NPSI in the Called DTE Address field of the X.25 Call Request packet sent on the line to establish the first virtual circuit session.
- The Called DTE Address field of all the X'0B' CALL REQUEST commands the CTCP uses to establish subsequent virtual circuit sessions must contain the same X.21 and ISDN address without any delimiter or X.21 and ISDN subaddress.

## CALL REQUEST Commands (CTCP to GATE)

***Commands Received during or after an X.21 Outgoing Connection:*** While NPSI is establishing an X.21 outgoing connection, any subsequent CALL REQUEST commands are refused until packet level setup is complete (X.25 Restart packets exchange completed).

Once the packet level is set up, NPSI compares the X.21 dial digits that are received in each subsequent CALL REQUEST command with the digits used for the X.21 outgoing connection. If the X.21 dial digits are different, NPSI refuses the CALL REQUEST command by sending a negative response to the CTCP. If the X.21 dial digits match, the CALL REQUEST command is processed by GATE the same way a CALL REQUEST command would be processed on a non-X.21 connection.

- A free virtual circuit number is assigned to the call.
- The X.25 Call Request packet is sent on the SVC to the remote DTE.

***Commands Received during or after an X.21 Incoming Connection:*** Once NPSI detects the X.21 Incoming Call signal, it refuses any CALL REQUEST commands that the CTCP sends until the X.25 link level and packet level are set up. To refuse these commands, NPSI returns negative responses to the CTCP.

Because calling party identification is not supported on X.21 switched connection, the origin of the call is not known by NPSI. In this case, checking of X.21 dial digits is not performed on CALL REQUEST commands following an X.21 incoming connection.

Once the packet level is set up, NPSI processes all CALL REQUEST commands received from the CTCP, as would occur in a typical CTCP to GATE operation.

## Virtual Circuit Session Termination

A virtual circuit can be terminated by using the set of requests described for GATE non-X.21 switched connections. See “Virtual Circuit Termination” on page 3-4 for more information. At the end of deactivation of a virtual circuit session on an X.21 switched connection, NPSI checks if it was the last virtual circuit active on the line.

- If the answer is yes, NPSI:
  1. Disconnects the X.25 link level
  2. Releases the X.21 connection
  3. Sets the physical interface in MONITOR INCOMING CALL state
- If the answer is no, NPSI resumes the process of virtual circuit deactivation as it would occur for a non-X.21 connection.

## MCH Failure Handling for X.21 Connections

When an MCH INOP (non recoverable line failure) occurs, the MCH LU becomes inoperable, and the session with the CTCP fails.

When either the X.21 switched path is cleared or the X.25 link is reinitialized, the MCH PU and MCH LU remain active and the control session with the CTCP does not fail. The X.21 connection is cleared, and the physical interface is set again to MONITOR INCOMING CALL state. However, the active virtual circuit PUs are all made inoperable.

See the “RECFMS Byte Expansions” section of the *NCP Version 7 and X.25 NPSI Version 3 Diagnosis, Customization, and Tuning* (LY30-5610) manual for RECFMS error bytes (MKBMDREC) values associated with different MCH failures.

## Command Interface between GATE and the CTCP for X.21 Switched Connections

In the following command descriptions, the RESID field in the RU format corresponding to the resource identifier designates either the set of SNA resources (LINE, PU, and LU) that are associated with a virtual circuit, or the X.25 virtual circuit number (system generation option VCID). SNA resource identifiers are associated with virtual circuits for each MCH. They correspond to the number of the SNA resource within the set of the SNA resources associated with the MCH. The *range* of the resource identifier numbers is the same as the *range* of virtual circuit numbers. An SNA resource number consists of three digits in hexadecimal format. The first digit is the logical group number and the next two digits are the logical channel numbers.

The following example illustrates how resource identifier numbers are determined.

```
X25.LCG  LCGN=0
X25.VC   LCN=(1,2),TYPE=P ....  2 PVCs
X25.VC   LCN=(3,99),TYPE=S ... 97 SVCs
X25.LCG  LCGN=2
X25.VC   LCN=(1,3),TYPE=P ....  3 PVCs
X25.VC   LCN=(4,98),TYPE=S ....95 SVCs
```

In this example, SNA identifiers X'003' through X'063' and SNA identifiers X'204' through X'262' are used to map all SVCs of this MCH. This setup allows CTCPs to continue to run without change.

The commands recognized or generated by GATE for an X.21 switched connection are listed in Table 3-4.

*Table 3-4. CTCP-GATE Control Commands*

| **Command code** | **Command** | **Session with CTCP** |
|---|---|---|
| 00 | DATA EXCHANGE (Q bit set to 0) | Data Session |
| 02 | DATA EXCHANGE (Q bit set to 1) | Data Session |
| 0B | INCOMING CALL | Control Session |
| 0B or 8B | CALL REQUEST | Control Session |
| 0F | CALL CONNECTED/CALL ACCEPTED | Control Session |
| 13 | CLEAR REQUEST/CLEAR INDICATION | Control Session or Data Session* |
| 17 | CLEAR CONFIRMATION | Control Session or Data Session* |
| 1B | RESET | Control Session or Data Session* |
| 1F | RESET CONFIRMATION | Data Session |
| 23 | INTERRUPT | Data Session |
| 27 | INTERRUPT CONFIRMATION | Data Session |
| F1 | DIAGNOSTIC | Control Session |
| FF | INFORMATION REPORT | Control Session or Data Session* |

**Note:** An asterisk (*) indicates that the command flows on the data session if the virtual circuit LU is in session; otherwise, the command flows on the control session.

For X.21 switched connections, there are two CALL REQUEST commands (X'0B' and X'8B'). A detailed description of these CALL REQUEST commands begins with “CALL REQUEST X'8B' (CTCP to GATE through the Control Session).”

For a detailed description of all other commands listed in Table 3-4, see “Command Interface between GATE and the CTCP” on page 3-6.

## CALL REQUEST X'8B' (CTCP to GATE through the Control Session)

The CALL REQUEST command is issued from CTCP to GATE through the control session to establish the first virtual circuit session on an X.21 switched connection. This command starts the X.21 switched connection. The command is sent in the following format:

```
          ┌───┬─────┬────┬─────┬──//──┐
RU:       │ cc│ cnid│ pw │ psiz│ crpk │
          └───┴─────┴────┴─────┴──//──┘
```

| Bytes | Field | Description |
|---|---|---|
| 0 | cc | Command code. For this command, the value is X'8B'. |
| 1-2 | cnid | Unique connection identifier determined by the CTCP. This 3-digit number is right-aligned. The leftmost halfbyte must be X'0'. To avoid conflicts, you should use numbers greater than the largest SNA resource identifier numbers (virtual circuit numbers). |
| 3 | pw | Packet window size. |
| 4-5 | psiz | Packet size, in hexadecimal, used for this virtual circuit. |
| 6-n | crpk | Variable size field containing the Call Request packet without the packet header. The X.21/ISDN number (dial digits used to perform the X.21 switched connection to ISDN) is contained in the Called DTE Address field of the Call Request packet.<br>**V3R7 and later:** You can specify a delimiter digit (X'A') in the Called DTE Address field to allow specific processing of X.21/ISDN subaddress digits. See "X.21 Outgoing Call" on page 3-39 for a description of this processing. |

When the X.21 connection is already active, any CALL REQUEST command with command code X'8B' received from the CTCP is rejected by a negative response (X'081C'—function not executable).

**V3R7 and later:** The CTCP might set packet window and packet window size fields to zero to allow NPSI to determine the X.25 flow control values for a virtual call. See "Session Establishment through an SVC" on page 3-2 for information.

### CALL REQUEST X'0B' (CTCP to GATE through the Control Session)

The CALL REQUEST command is issued from CTCP to GATE through the control session to establish the next virtual circuit sessions once the X.21 switched connection is active. The command is sent in the following format:

```
     ┌────┬──────┬────┬──────┬──//──┐
RU:  │ cc │ cnid │ pw │ psiz │ crpk │
     └────┴──────┴────┴──────┴──//──┘
```

| Bytes | Field | Description |
|---|---|---|
| 0 | cc | Command code. For this command, the value is X'0B'. |
| 1-2 | cnid | Unique connection identifier determined by the CTCP. This 3-digit number is right-aligned. The leftmost halfbyte must be X'0'. To avoid conflicts, you should use numbers greater than the largest SNA resource identifier numbers (virtual circuit numbers). |
| 3 | pw | Packet window size. |
| 4-5 | psiz | Packet size, in hexadecimal, used for this VC. |
| 6-n | crpk | Variable size field containing the Call Request packet without the packet header. The X.21/ISDN number (dial digits used to perform the X.21 switched connection to ISDN) is contained in the DTE called address field of the Call Request packet.<br>Delimiter digit X'A' is not allowed in the Called DTE Address field. |

When the X.21 connection is not active, any CALL REQUEST command with command code X'0B' received from the CTCP is rejected by a negative response (X'081C'—function not executable).

**V3R7 and later:** The CTCP might set packet window and packet window size fields to zero to allow NPSI to determine the X.25 flow control values for a virtual call. See “Session Establishment through an SVC” on page 3-2 for information.

**For the Migration Facility:** When only one SVC is defined at SYSGEN on the MCH dedicated to X.21 switched access to ISDN, the X.21 switched connection can be initiated by a CTCP with either a CALL REQUEST X'8B', or a CALL REQUEST X'0B'.

To manage data exchange and data session control, a CTCP uses the set of commands defined for GATE non-X.21 switched connections.

## Fast Transaction Processing Interface (FTPI) Using GATE (V3R4 and Later)

FTPI is a protocol of CTCP and NPSI. You can have 1 to 28 CTCPs in session with NPSI. Each of these sessions, called CTCP pseudo-link (CPL) sessions, multiplexes the traffic to and from the virtual circuit of FTPI MCHs between the CTCP and NPSI.

FTPI enhances the performance of the system with the following improvements:

- Reduction of session setup time

- Increased FTPI GATE traffic performance
- Quick recovery after failure of a network component
- Quick response time for connection establishment
- Handling of unusual MCH and virtual circuit numbers
- **For V3R8 and later**: Dedication of an entire FTPI MCH to PVCs

In a network environment where large numbers of non-SNA terminals are connected to a host server, communication must be optimized. To accomplish this, FTPI does the following:

- Reduces the number of PIUs
- Reduces extra SNA traffic

You can code a suggested logmode table as shown in Figure 3-6.

```
FTPI    MODEENT   LOGMODE=FAST,
                  TSPROF=X'03',
                  FMPROF=X'03',
                  PRIPROT=X'10',  (- multi RU chain from PLU   )
                                  (- definite or exception resp)
                                  (- primary does not send EB  )
                  SECPROT=X'10',  (- multi RU chain from PLU   )
                                  (- definite or exception resp)
                                  (- secondary does not send EB)
                  COMPROT=X'0020' (- bracket not used          )
                                  (- half-duplex contention    )
```

*Figure 3-6. Suggested Coding for Logmode Table*

## Message Traffic Multiplexing

If there is only one CTCP defined, all of the traffic coming over all of the FTPI MCHs is multiplexed by FTPI on the unique LU-LU session with the CTCP. Conversely, the CTCP multiplexes all of the commands it sends to the virtual circuits of the FTPI MCHs on the unique LU-LU session. In addition to this multiplexing function, the blocking function allows one PIU to carry several commands related to different virtual circuits of different FTPI MCHs. A multiplexing timer prevents poor response time in the case of low data traffic. FTPI inbound blocking waits for the reception of several packets until the PIU is filled. When traffic is low, the packets that are blocked by FTPI stay in the queue for an extended amount of time, resulting in poor response time. To avoid this, a timer covers the blocking period. When the timer elapses, the PIU is forwarded, regardless of whether it is full.

The FTPI inbound traffic regulation blocks incoming traffic according to a packet counter. The average number of packets received from the network between the time the timer starts and stops is computed. This computed value is called a packet counter, and is an NPSI internal value (for more information about FTPI traffic regulation blocks, see “Blocking Regulation” on page 3-50).

## FTPI Protocols

FTPI is an extension of the GATE interface of NPSI. In this role, an FTPI CTCP controls the session establishment and takedown of all virtual circuits on the FTPI MCHs. The FTPI CTCP also processes all of the data flowing over these FTPI virtual circuits.

## Command Format

The FTPI command differs from the FC command in that it has an additional 8-byte header.

Packets coming from the X.25 network are converted to FTPI commands by NPSI and sent to the FTPI CTCP. In the other direction, CTCP FTPI commands are transformed to packets by NPSI. The packets are then sent to the network.

## Identification of the Virtual Circuit

To demultiplex the data traffic on the FTPI session, a virtual circuit identifier (VCID) is created. This identifier consists of two components, the multichannel link identifier (MCHID) and the virtual circuit identifier (VCI). The MCHID is the line address of the MCH, and VCID is the VC number. Together, these uniquely identify the virtual circuit, which can be either the origin or destination of the data.

MCHID and VCID are 12 bits in length. Each accommodates up to 4096 elements (`'FFF'`). The MCHID and VCID are affected by FTPI NPSI during the Call setup.

Some FTPI commands do not carry MCHID and VCID. In this case, the entire session is affected by the GO and NOGO commands. See “NOGO and GO Commands” for more information on these commands.

## Correlation Number

When either an Incoming Call packet or a Call Connected packet is received from the X.25 network, NPSI adds a *correlation number* to the INCOMING CALL command or the CALL CONNECTED command as a new identification element. This number is used during the entire connection.

The FTPI CTCP must add the correlation number, without change, to all of the commands sent to NPSI. NPSI maintains a unique correlation number for each virtual circuit. Conversely, NPSI adds this correlation number to all of the commands it sends to the CTCP.

NPSI verifies the correlation number in all of the commands it receives from the FTPI CTCP. If there is a discrepancy, the command is discarded, and an ERROR INFO REPORT is sent to the FTPI CTCP.

## NOGO and GO Commands

The FTPI CTCP can implement some inbound flow control by using the NOGO and GO commands. The NOGO command alerts NPSI of problems, such as a buffer shortage at the host. Upon receiving the NOGO command, NPSI either reroutes the Incoming Call packets to any available alternative CTCP, if the ALTCTCP keyword is coded on the X25.CPL statement, or clears them with a diagnostic code of *busy condition in host* (index `X'EB'`). See “Load Balancing” on page 3-48 for more information.

The GO command signals NPSI that the CTCP is once again available to accept Incoming Call packets. Once the NOGO command has been sent, NPSI no longer sends Incoming Calls to that CTCP. To reinitiate processing of Incoming Call packets for that CTCP, you must send the GO command.

**Note:** The NOGO and GO commands affect the entire CTCP session. During the period between issuing NOGO and GO, Incoming Call packets cannot be passed to the CTCP that issued the NOGO command. However, these packets can be accepted by other CTCPs if the ALTCTCP keyword is coded.

## FTPI Outbound Flow Control

All multiplexed traffic must be handled equally. If the session traffic is homogenous, this is enough flow control. Problems can arise if the traffic has different throughput requirements.

An example of this problem is when small interactive traffic and large file transfers are intermixed on the multiplexed session. In this case, the interactive traffic has a difficult time getting sufficient throughput. In these cases, you should separate the interactive and file transfer traffic to different sessions. This can be done in two different ways:

- Dial into a specific DTE address for each type of traffic.

  For example, you can dedicate DTE address 1 to interactive traffic, and DTE address 2 to a file transfer.

- Using a menu system to process applications with different data requirements.

  For example, a menu system can be defined as the LOGAPPL for the LU. The session is then passed (CLSDST PASS) to the data application. When the session is ended, the LU is put back in session with the menu system.

Figure 3-7 shows how a menu system could be used to manage FTPI outbound flow control.

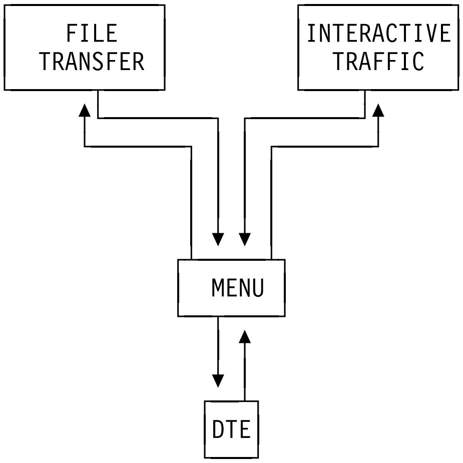


*Figure 3-7. Potential Menu Management for FTPI Outbound Control*

**V3R6 and Later:** An outbound flow control mechanism has been adopted to prevent a shortage of buffer space when outbound traffic is heavy.

When the number of packets waiting in the virtual circuit outbound queues reaches a critical threshold (1500 buffers in use), NPSI suspends processing of new PIUs until the number of packets in the outbound queues falls below the critical threshold. At that point, processing of new PIUs resumes.

## Load Balancing

When one CTCP is overloaded or in the process of being stopped, the CTCP notifies NPSI that new incoming calls should not be forwarded to that CTCP by sending a NOGO command. Rather than having the calls cleared, load balancing allows FTPI to reroute the incoming calls to an alternative CTCP.

A table is created in NPSI at system generation using the ALTCTCP keyword. This table defines one or several backup CTCPs for a given CTCP.

When FTPI receives an Incoming Call packet, it checks the session with the desired CTCP. If the CTCP cannot be reached (for example, the session is not active or is in the NOGO state), the INCOMING CALL command is forwarded to the first available alternative CTCP.

**Note:** When a NOGO command is received, data transfer continues for connections established before the NOGO state.

## Qualified and Unqualified Data

The data exchanged can be qualified or unqualified. Qualified data is used to communicate control data. Unqualified data is used to communicate user data. An example of qualified (control) data is a packet containing a PAD command.

If the data is specified as qualified, the Q bit is turned on in the X.25 packet header. The data type is communicated between FTPI GATE and FTPI CTCP through the first byte of the RU as:

| **RU byte 0** | **Usage** |
|---|---|
| X'00' | Unqualified data |
| X'02' | Qualified data |

## Outgoing Call

FTPI converts the CALL REQUEST command to a Call Request packet, and sends it to the network.

***MCHID:*** The MCHID on the CALL REQUEST command must be set by the CTCP, because the CTCP builds the Call Request packet. The MCHID is the line address of the physical line to be used for callout. The high-order 4 bits of the MCHID are set to X'0'. The called address must be set in the CALL REQUEST command.

***VCID:*** When the CTCP sends the CALL REQUEST command, the CTCP does not know the logical channel to use to send the call. Therefore, the CTCP sets the VCID to X'x000'. The high-order 4 bits (x) must be set to X'0'.

On the CALL CONNECTED response, NPSI determines the logical channel of the connection and sets the VCID. The CTCP must save the VCID for use in the connection process. If a logical channel is not available, NPSI sends a CLEAR command back to the CTCP with the same correlation number, and the VCID set to X'0000'.

***Correlation Number:*** When the CTCP sends the CALL REQUEST command, the CTCP does not know the correlation number that NPSI assigns to this connection. Therefore, the CTCP sets the correlation number to X'0000'. On the CALL CONNECTED response, the CTCP must save the correlation number for use in all of the commands sent for this connection.

The Correlation number is not useful for PVCs, so commands related to PVCs have their correlation number field set to 0000.

***Flow Sequence:*** Figure 3-8, and Figure 3-9 on page 3-50 show the flow sequences for a Call Request.

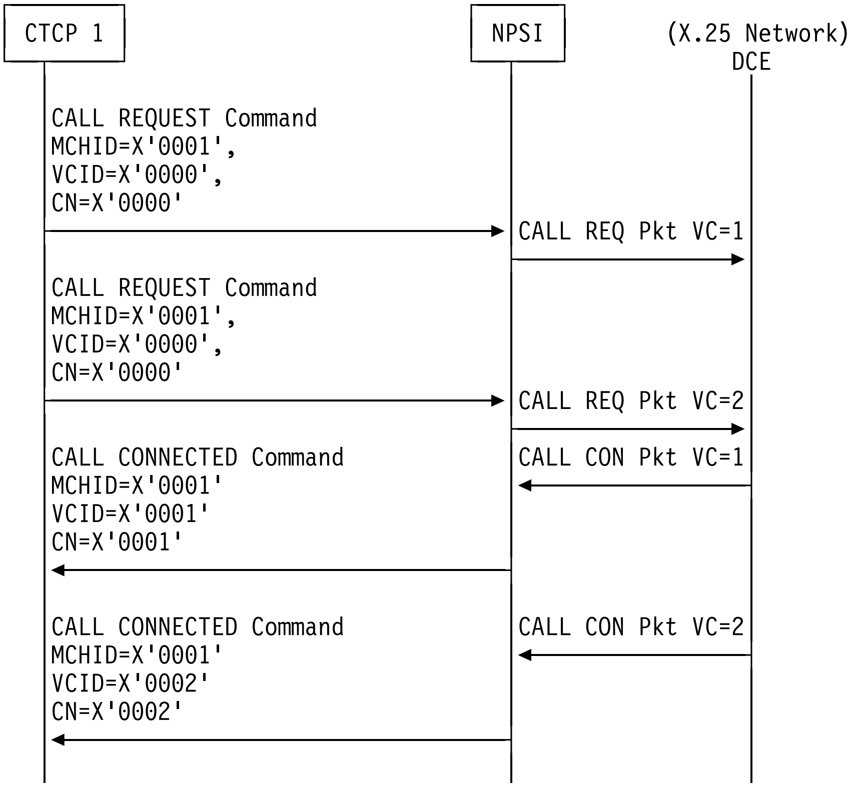


*Figure 3-8. Call Request Sequence with One CTCP*

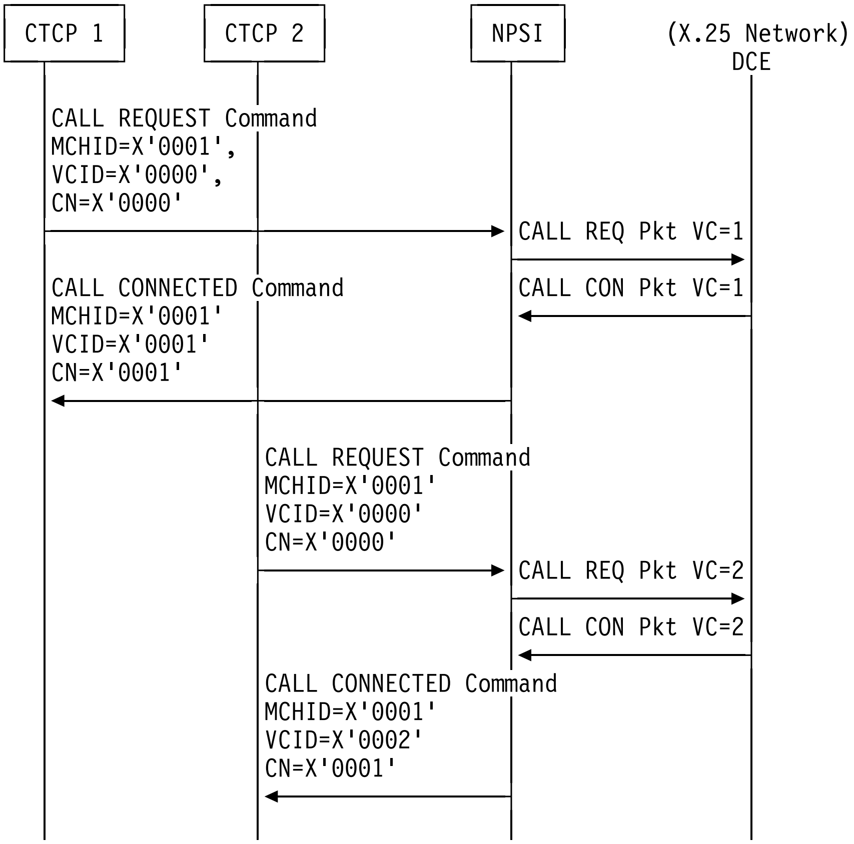


*Figure 3-9. Call Request Sequence with Two CTCPs*

## PVC Support

FTPI allows the CTCP host application to exchange commands and data on PVCs. FTPI provides the following enhancements for PVC support:

- Commands for a PVC can be received from any of the active CTCPs.
- Packets are forwarded to the PVC preferred CTCP.
- The preferred CTCP is defined in NPSI generation on the X25.LINE and X25.VC statements.

## Blocking Regulation

Blocking regulation allows NPSI to control traffic flow for improved response time. There are two regulation modes for FTPI: blocked and unblocked.

***Blocked Mode:*** FTPI inbound blocked mode allows NPSI to wait for the reception of several packets before multiplexing all of them into one PIU. In blocked mode, there are two conditions that determine when to forward a PIU:

- The size of the PIU has reached the maximum PIU size defined during CPL session establishment.
- The X25.CPL MAXTIME timer has elapsed.

***Unblocked Mode:*** FTPI inbound unblocked mode allows NPSI to build and send a PIU as soon as one packet is received from the network.

FTPI inbound traffic regulation is a continual process in NPSI that determines whether blocked or unblocked mode is used, depending on the amount of traffic from the network.

A packet counter is incremented each time a packet is received between the starting and elapsing of the timer. When the timer elapses, an average is computed using the following formula:

```
average = ( packet counter + 15 previous average ) ÷ 16
```

If this average is greater than the blocking factor value, blocked mode is set. If it is less than the blocking factor value, unblocked mode is set.

**Note:** The blocking factor value is defined at system generation using the MINDATA keyword on the X25.CPL statement.

**CPL Session Failure:** When a CPL session fails, NPSI clears the active SVCs and resets the active PVCs linked through this session with a clear or reset diagnostic code X'C5'.

## FTPI Commands

The FTPI header is used for the command descriptions in Table 3-5 on page 3-52. The following fields are in the FTPI header:

LENGTH OF COMMAND DATA This is the length of the data area starting at, and including, the command code.

MCHID The line address of the MCH on which the virtual circuit resides.

VCID Virtual circuit identification (LCGN/LCN).

**Note:** The combination of MCHID and VCID uniquely identifies the virtual circuit that is the origin or destination of the data.

RESID The RESID field in the RU format designates either the SNA resources (LINE, PU, and LU) that are associated with a virtual circuits, or the X.25 virtual circuit number (systems generation option VCID).

Correlation Number A unique identifier set by NPSI in the INCOMING CALL command and the CALL CONNECTED command. The correlation number is attached to all processing commands for this resource by both the CTCP and NPSI. The correlation number is set to 0000 in PVC related commands.

Figure 3-10 shows the command format sent between CTCP and NPSI:

| Length of Command Data | MCHID | VCID | Correlation Number | Cmd Code | DATA |
|---|---|---|---|---|---|
| 0 1 | 2 3 | 4 5 | 6 7 | 8 | 9 n |

*Figure 3-10. Command Format Sent between CTCP and NPSI*

**Note:** In most commands, the user field, which is part of the data field (bytes 9—n), not only contains user data, but can also contain control data. Control data can be DTE addresses or facilities, or any information set by either remote DTE, or by the network.

Table 3-5 shows the GATE control commands for FTPI.

*Table 3-5. CTCP-FTPI GATE Control Commands*

| **Command Code** | **Command** | **Session with CTCP** |
|---|---|---|
| 00 | DATA EXCHANGE without Q bit | CPL Session |
| 02 | DATA EXCHANGE with Q bit | CPL Session |
| 0B | CALL REQUEST/INCOMING CALL | CPL Session |
| 0F | CALL CONNECTED/CALL ACCEPTED | CPL Session |
| 13 | CLEAR on Incoming Call | CPL Session |
| 13 | CLEAR on Call Request | CPL Session |
| 13 | CLEAR | CPL Session |
| 13 | CLEAR Request | CPL Session |
| 17 | CLEAR CONFIRMATION | CPL Session |
| 1B | RESET | CPL Session |
| 1F | RESET CONFIRMATION | CPL Session |
| 23 | INTERRUPT | CPL Session |
| 27 | INTERRUPT CONFIRMATION | CPL Session |
| F1 | DIAGNOSTIC | CPL Session |
| F2 | NOGO | CPL Session |
| F3 | GO | CPL Session |
| FF | ERROR INFORMATION REPORT | CPL Session |

The data packets transferred are complete messages that are prefixed by a command header. The header provides the data length, X.25 MCHID, and the VCID. The header also contains a correlation number to use when handling collisions.

## DATA EXCHANGE without Q Bit (CTCP to FTPI, FTPI to CTCP through the CPL Session)

A DATA EXCHANGE command with the Q bit set off is issued from the CTCP to FTPI and from FTPI to the CTCP. The command is sent in the following format:

```
    ┌─────┬───────┬──────┬────┬────┬──//──┐
RU: │ lcd │ mchid │ vcid │ cn │ cc │ user │
    └─────┴───────┴──────┴────┴────┴──//──┘
```

| Bytes | Field | Description |
|---|---|---|
| 0–1 | lcd | Length of command data. |
| 2–3 | mchid | Multichannel link identifier. |
| 4–5 | vcid | VCI. |
| 6–7 | cn | Correlation number. |
| 8 | cc | Command code. For this command, the value is X'00'. |
| 9–n | user | Variable-size field containing user data following the packet header. |

## DATA EXCHANGE with Q Bit (CTCP to FTPI, FTPI to CTCP through the CPL Session)

A DATA EXCHANGE command with the Q bit set on is issued from the CTCP to FTPI and from FTPI to the CTCP. The command is sent in the following format:

```
    ┌─────┬───────┬──────┬────┬────┬──//──┐
RU: │ lcd │ mchid │ vcid │ cn │ cc │ user │
    └─────┴───────┴──────┴────┴────┴──//──┘
```

| Bytes | Field | Description |
|---|---|---|
| 0–1 | lcd | Length of command data. |
| 2–3 | mchid | Multichannel link identifier. |
| 4–5 | vcid | VCI. |
| 6–7 | cn | Correlation number. |
| 8 | cc | Command code. For this command, the value is X'02'. |
| 9–n | user | Variable-size field containing user data following the packet header. |

## CALL REQUEST (CTCP to FTPI through the CPL Session)

The CALL REQUEST command is issued from the CTCP to FTPI. The command is sent in the following format:

```
                                                    //
RU:  | lcd | mchid | vcid | cn | cc | cnid | pw | psiz | crpk |
                                                    //
```

| Bytes | Field | Description |
|---|---|---|
| 0–1 | lcd | Length of command data. |
| 2–3 | mchid | Multichannel link identifier. |
| 4–5 | vcid | VCI. Set to X'000' by the CTCP. This field is right aligned. The leftmost halfbyte must be X'0'. |
| 6–7 | cn | Correlation number (reserved by NPSI). |
| 8 | cc | Command code. For this command, the value is X'0B'. |
| 9–10 | cnid | Connection identifier determined by the CTCP. The leftmost halfbyte must be X'0'. This field is right aligned. |
| 11 | pw | Packet window size. |
| 12-13 | psiz | Packet size, in hexadecimal. |
| 14-n | crpk | Variable-size field containing the Call Request packet without the packet header. |

**V3R7 and later:** The CTCP might set packet window and packet window size fields to zero to allow NPSI to determine the X.25 flow control values for a virtual call. See "Session Establishment through an SVC" on page 3-2 for information.

## INCOMING CALL (FTPI to CTCP through the CPL Session)

The INCOMING CALL command is issued from FTPI to the CTCP. The command is sent in the following format:

```
                                             //
RU:  | lcd | mchid | vcid | cn | cc | resid | icpk |
                                             //
```

| Bytes | Field | Description |
|---|---|---|
| 0–1 | lcd | Length of command data. |
| 2–3 | mchid | Multichannel link identifier. |
| 4–5 | vcid | VCI. |
| 6–7 | cn | Correlation number. |
| 8 | cc | Command code. For this command, the value is X'0B'. |
| 9–10 | resid | Resource identifier. The leftmost halfbyte is set to X'F'. |
| 11–n | icpk | Variable-size field containing the Incoming Call packet received from the network (without the 3-byte packet header). |

## CALL CONNECTED (FTPI to CTCP through the CPL Session)

The CALL CONNECTED command is issued from FTPI to the CTCP. The command is sent in the following format:

```
     ┌─────┬───────┬──────┬────┬────┬──────┬───────┬──//──┐
RU:  │ lcd │ mchid │ vcid │ cn │ cc │ cnid │ resid │ ccpk │
     └─────┴───────┴──────┴────┴────┴──────┴───────┴──//──┘
```

| Bytes | Field | Description |
|---|---|---|
| 0–1 | lcd | Length of command data. |
| 2–3 | mchid | Multichannel link identifier. |
| 4–5 | vcid | VCI. Set to X'000' by the CTCP. This field is right aligned. The leftmost halfbyte must be X'0'. |
| 6–7 | cn | Correlation number (reserved by NPSI). |
| 8 | cc | Command code. For this command, the value is X'0F'. |
| 9-10 | cnid | Connection identifier as specified in the CALL REQUEST command. |
| 11-12 | resid | Resource identifier. The leftmost half-byte is set to F. |
| 13-n | ccpk | Variable-size field containing optional bytes that follow the packet header of the Call Connected packet. |

**Note:** Depending on the generation option specified on the DEFPRI keyword, the QOS facility (X'D2', contained in the Call Connected packet) determines the OSI priority of data that is used during the session.

**V3R7 and later:** The CTCP might set packet window and packet window size fields to zero to allow NPSI to determine the X.25 flow control values for a virtual call. See “Session Establishment through an SVC” on page 3-2 for information.

## CALL ACCEPTED (CTCP to FTPI through the CPL Session)

The CALL ACCEPTED command is issued from the CTCP to FTPI. The command is sent in the following format:

```
                                                             ┌─//──┐
RU:  │ lcd │ mchid │ vcid │ cn │ cc │ resid │ pw │ psiz │ capk │
                                                             └─//──┘
```

| Bytes | Field | Description |
|---|---|---|
| 0–1 | lcd | Length of command data. |
| 2–3 | mchid | Multichannel link identifier. |
| 4–5 | vcid | VCI. |
| 6–7 | cn | Correlation number. |
| 8 | cc | Command code. For this command, the value is X'0F'. |
| 9–10 | resid | Resource identifier as found in the INCOMING CALL command. |
| 11 | pw | Packet window size. |
| 12–13 | psiz | Packet size in hexadecimal used for this virtual circuit. |
| 14–n | capk | Variable-size field containing the Call Accepted packet to be sent to the network without the 3-byte packet header. |

**Note:** Depending on the generation option specified on the DEFPRI keyword, the QOS facility (X'D2', contained in the Call Accepted packet) determines the OSI priority of data that is used during the session.

**V3R7 and later:** The CTCP might set packet window and packet window size fields to zero to allow NPSI to determine the X.25 flow control values for a virtual call. See “Session Establishment through an SVC” on page 3-2 for information.

## CLEAR on Incoming Call (CTCP to FTPI through the CPL Session)

The CLEAR command to clear an incoming call is issued from the CTCP to FTPI. The command is sent in the following format:

```
    ┌─────┬───────┬──────┬────┬────┬───────┬──────┬──//──┐
RU: │ lcd │ mchid │ vcid │ cn │ cc │ resid │ crdf │ user │
    └─────┴───────┴──────┴────┴────┴───────┴──────┴──//──┘
```

| Bytes | Field | Description |
|---|---|---|
| 0–1 | lcd | Length of command data. |
| 2–3 | mchid | Multichannel link identifier. |
| 4–5 | vcid | VCI. |
| 6–7 | cn | Correlation number. |
| 8 | cc | Command code. For this command, the value is X'13'. |
| 9–10 | resid | Resource identifier as found in the INCOMING CALL command. |
| 11–12 | crdf | CAUSE and DIAGNOSTIC fields that are to be inserted in the Clear Request packet. |
| 13–n | user | Variable-size field containing optional user data to be put in the Clear Request packet after the CAUSE and DIAGNOSTIC fields. |

## CLEAR on Call Request (FTPI to CTCP through the CPL Session)

The CLEAR command to clear a call request is issued from FTPI to CTCP. The command is sent in the following format:

```
    ┌─────┬───────┬──────┬────┬────┬──────┬──────┬──//──┐
RU: │ lcd │ mchid │ vcid │ cn │ cc │ cnid │ crdf │ user │
    └─────┴───────┴──────┴────┴────┴──────┴──────┴──//──┘
```

| Bytes | Field | Description |
|---|---|---|
| 0–1 | lcd | Length of command data. |
| 2–3 | mchid | Multichannel link identifier. |
| 4–5 | vcid | VCI. |
| 6–7 | cn | Correlation number. |
| 8 | cc | Command code. For this command, the value is X'13'. |
| 9-10 | cnid | Connection identifier as specified in the CALL REQUEST command. |
| 11–12 | crdf | CAUSE and DIAGNOSTIC fields that are to be inserted in the Clear Request packet. |
| 13–n | user | Variable-size field containing optional user data to be put in the Clear Request packet after the CAUSE and DIAGNOSTIC fields. |

## CLEAR (FTPI to CTCP through the CPL Session)

The CLEAR command is issued from FTPI to CTCP. The command is sent in the following format:

```
     ┌────┬───────┬──────┬────┬────┬───────┬──────┬──//──┬─────────┐
RU:  │ lcd│ mchid │ vcid │ cn │ cc │ resid │ ccdf │ user │ billing │
     └────┴───────┴──────┴────┴────┴───────┴──────┴──//──┴─────────┘
```

| Bytes | Field | Description |
|---|---|---|
| 0–1 | lcd | Length of command data. |
| 2–3 | mchid | Multichannel link identifier. |
| 4–5 | vcid | VCI. |
| 6–7 | cn | Correlation number. |
| 8 | cc | Command code. For this command, the value is X'13'. |
| 9–10 | resid | Resource identifier. The leftmost halfbyte is set to X'F'. |
| 11–12 | ccdf | Clear CAUSE and DIAGNOSTIC fields. |
| 13–n | user | Variable-size field containing optional user data following the CAUSE and DIAGNOSTIC fields. |
| n–m | billing | Variable-size field containing billing information, if the billing option is requested at NPSI generation. See “Normal Clearing” on page 3-23 for billing format details. |

**Note:** In case of MCH failure, for each active SVC of the MCH, NPSI builds a pseudo CLEAR command with ccdf=0030, and sends it to the CTCP.

## CLEAR REQUEST (CTCP to FTPI through the CPL Session)

The CLEAR REQUEST command is issued from CTCP to FTPI. The command is sent in the following format:

```
     ┌────┬───────┬──────┬────┬────┬──────┬──//──┐
RU:  │ lcd│ mchid │ vcid │ cn │ cc │ ccdf │ user │
     └────┴───────┴──────┴────┴────┴──────┴──//──┘
```

| Bytes | Field | Description |
|---|---|---|
| 0–1 | lcd | Length of command data. |
| 2–3 | mchid | Multichannel link identifier. |
| 4–5 | vcid | VCI. |
| 6–7 | cn | Correlation number. |
| 8 | cc | Command code. For this command, the value is X'13'. |
| 9–10 | ccdf | Clear CAUSE and DIAGNOSTIC fields. |
| 11–n | user | Variable-size field containing optional user data following the CAUSE and DIAGNOSTIC fields. |

## CLEAR CONFIRMATION after CLEAR Command for Incoming Call Refused (FTPI to CTCP through the CPL Session)

The CLEAR CONFIRMATION command, which acknowledges the clearing of an incoming call by the CTCP, is issued from FTPI to CTCP. The command is sent in the following format:

```
                                                  ┌──//──┐
RU:   │ lcd │ mchid │ vcid │ cn │ cc │ resid │ user │
                                                  └──//──┘
```

| Bytes | Field | Description |
|---|---|---|
| 0-1 | lcd | Length of command data. |
| 2–3 | mchid | Multichannel link identifier. |
| 4–5 | vcid | VCI. |
| 6–7 | cn | Correlation number. |
| 8 | cc | Command code. For this command, the value is X'17' |
| 9–10 | resid | Resource identifier as found in the INCOMING CALL command. |
| 11–n | user | Variable-size field containing optional user data if present in the DCE Clear Confirmation packet. |

## CLEAR CONFIRMATION (FTPI to CTCP through the CPL Session)

The CLEAR CONFIRMATION command is issued after the CLEAR command if the incoming call is refused. It is issued from FTPI to the CTCP. The command is sent in the following format:

```
                                          ┌──//──┐
RU:   │ lcd │ mchid │ vcid │ cn │ cc │ user │ billing │
                                          └──//──┘
```

| Bytes | Field | Description |
|---|---|---|
| 0–1 | lcd | Length of command data. |
| 2–3 | mchid | Multichannel link identifier. |
| 4–5 | vcid | VCI. |
| 6–7 | cn | Correlation number. |
| 8 | cc | Command code. For this command, the value is X'17'. |
| 9–n | user | Variable-size field containing optional user data if present in the DCE Clear Confirmation packet. |
| n–m | billing | Variable-size field containing billing information, if the billing option is requested at NPSI generation. See “Normal Clearing” on page 3-23 for billing format details. |

## RESET (FTPI to CTCP, CTCP to FTPI through the CPL Session)

The RESET command is issued from FTPI to the CTCP and from the CTCP to FTPI. The command is sent in the following format:

```
                                                   //
RU:   | lcd | mchid | vcid | cn | cc | rcdf |  user
                                                   //
```

| Bytes | Field | Description |
|---|---|---|
| 0–1 | lcd | Length of command data. |
| 2–3 | mchid | Multichannel link identifier. |
| 4–5 | vcid | VCI. |
| 6–7 | cn | Correlation number. |
| 8 | cc | Command code. For this command, the value is X'1B'. |
| 9–10 | rcdf | Reset CAUSE and DIAGNOSTIC fields. |
| 11–n | user | Variable-size field containing optional user data following the CAUSE and DIAGNOSTIC fields. |

**Note:** In case of MCH failure, for each active PVC of the MCH, NPSI builds a RESET command with rcdf=0030 and sends it to the CTCP.

If NPSI detects an unrecoverable situation, FTPI builds and sends a pseudo-RESET command to the CTCP and resets the virtual circuit according to the type of error, as shown in the following table:

| Type of Error | Reset CMD to CTCP | Reset Req to Network |
|---|---|---|
| Inconsistent Q-bit setting | Cause X'00' Diag X'53' | Diag for index X'53' |
| T25 time out | Cause X'00' Diag X'92' | Diag for index X'92' |
| Time out after received RNR | Cause X'00' Diag X'94' | Diag for index X'94' |
| Received packet longer than maximum packet size | Cause X'00' Diag X'A7' | Diag for index X'A7' |
| Packet received that is not valid P(S) | Cause X'00' Diag X'AB' | Diag for index X'AB' |
| Packet received that is not valid P(R) | Cause X'00' Diag X'AC' | Diag for index X'AC' |
| Packet received that is note valid D bit | Cause X'00' Diag X'AD' | Diag for index X'AD' |

**Note:** See *NPSI Diagnosis, Customization, and Tuning* for more information about NPSI Cause and Diagnostic codes.

## RESET CONFIRMATION (FTPI to CTCP through the CPL Session)

The RESET CONFIRMATION command is issued from FTPI to the CTCP. The command is sent in the following format:

```
     ┌─────┬───────┬──────┬────┬────┬──//──┐
RU:  │ lcd │ mchid │ vcid │ cn │ cc │ user │
     └─────┴───────┴──────┴────┴────┴──//──┘
```

| Bytes | Field | Description |
|---|---|---|
| 0–1 | lcd | Length of command data. |
| 2–3 | mchid | Multichannel link identifier. |
| 4–5 | vcid | VCI. |
| 6–7 | cn | Correlation number. |
| 8 | cc | Command code. For this command, the value is X'1F'. |
| 9–n | user | Variable-size field containing any optional user data found in the DCE Reset Confirmation packet. |

**CLEAR and RESET CONFIRMATION Commands:** Although the CTCP cannot send CLEAR CONFIRMATION or RESET CONFIRMATION commands, the CTCP must be able to receive these commands. The CTCP cannot send CLEAR CONFIRMATION and RESET CONFIRMATION commands, because NPSI automatically sends out Clear Confirmation and Reset Confirmation packets.

## INTERRUPT (CTCP to FTPI, FTPI to CTCP through the Data Session through the CPL Session)

The INTERRUPT command is issued from the CTCP to FTPI and from FTPI to the CTCP. The command is sent in the following format:

```
     ┌─────┬───────┬──────┬────┬────┬────┬──//──┐
RU:  │ lcd │ mchid │ vcid │ cn │ cc │ iu │ user │
     └─────┴───────┴──────┴────┴────┴────┴──//──┘
```

| Bytes | Field | Description |
|---|---|---|
| 0–1 | lcd | Length of command data. |
| 2–3 | mchid | Multichannel link identifier. |
| 4–5 | vcid | VCI. |
| 6–7 | cn | Correlation number. |
| 8 | cc | Command code. For this command, the value is X'23'. |
| 9 | iu | Interrupt cause byte. |
| 10–n | user | Variable-size field containing optional user data following the interrupt user data byte. |

## INTERRUPT CONFIRMATION (FTPI to CTCP, CTCP to FTPI through the CPL Session)

The INTERRUPT CONFIRMATION command is issued from FTPI to the CTCP and from the CTCP to FTPI. The command is sent in the following format:

```
         ┌─────┬───────┬──────┬────┬────┬──//──┐
RU:      │ lcd │ mchid │ vcid │ cn │ cc │ user │
         └─────┴───────┴──────┴────┴────┴──//──┘
```

| Bytes | Field | Description |
|---|---|---|
| 0–1 | lcd | Length of command data. |
| 2–3 | mchid | Multichannel link identifier. |
| 4–5 | vcid | VCI. |
| 6–7 | cn | Correlation number. |
| 8 | cc | Command code. For this command, the value is X'27'. |
| 9-n | user | Variable-size field containing optional user data following the packet header. |

## DIAGNOSTIC (FTPI to CTCP through the CPL Session)

The DIAGNOSTIC command is issued from FTPI to the CTCP. The command is sent in the following format:

```
         ┌─────┬───────┬──────┬────┬────┬────┬─//─┐
RU:      │ lcd │ mchid │ vcid │ cn │ cc │ dc │ od │
         └─────┴───────┴──────┴────┴────┴────┴─//─┘
```

| Bytes | Field | Description |
|---|---|---|
| 0–1 | lcd | Length of command data. |
| 2–3 | mchid | Multichannel link identifier. |
| 4–5 | vcid | VCI. |
| 6–7 | cn | Correlation number. |
| 8 | cc | Command code. For this command, the value is X'F1'. |
| 9 | dc | Diagnostic code. |
| 10–n | od | Variable-size field containing diagnostic explanation. |

## NOGO (CTCP to FTPI through the CPL Session)

The NOGO command is issued from CTCP to FTPI.  The command is sent in the following format:

|  |  |  |  |  |  |
|---|---|---|---|---|---|
| RU: | lcd | mchid | vcid | cn | cc |

| Bytes | Field | Description |
|---|---|---|
| 0–1 | lcd | Length of command data. |
| 2–3 | mchid | Multichannel link identifier, set to X'0000'. |
| 4–5 | vcid | VCI, set to X'0000'. |
| 6–7 | cn | Correlation number, set to X'0000'. |
| 8 | cc | Command code.  For this command, the value is X'F2'. |

## GO (CTCP to FTPI through the CPL Session)

The GO command is issued from CTCP to FTPI.  The command is sent in the following format:

|  |  |  |  |  |  |
|---|---|---|---|---|---|
| RU: | lcd | mchid | vcid | cn | cc |

| Bytes | Field | Description |
|---|---|---|
| 0–1 | lcd | Length of command data. |
| 2–3 | mchid | Multichannel link identifier.  set to X'0000'. |
| 4–5 | vcid | VCI, set to X'0000'. |
| 6–7 | cn | Correlation number, set to X'0000'. |
| 8 | cc | Command code.  For this command, the value is X'F3'. |

## ERROR/INFORMATION REPORT (FTPI to CTCP through the CPL Session)

The ERROR/INFORMATION REPORT command is issued from FTPI to the CTCP. The command is sent in the following format:

```
        ┌────────┬───────────┬──────────┬──────┬──────┬─────────┬──────┬──────┬─//─┐
RU:     │  lcd   │  mchid    │  vcid    │  cn  │  cc  │  resid  │  ce  │  ec  │ bi │
        └────────┴───────────┴──────────┴──────┴──────┴─────────┴──────┴──────┴─//─┘
```

| Bytes | Field | Description |
|---|---|---|
| 0–1 | lcd | Length of command data. |
| 2–3 | mchid | Multichannel link identifier. |
| 4–5 | vcid | VCI. |
| 6–7 | cn | Correlation number. |
| 8 | cc | Command code. For this command, the value is X'FF'. |
| 9–10 | resid<br>or cnid | Resource identifier. The leftmost halfbyte must be X'0'.<br>Connection identifier if this report applies to a previous call command. |
| 11 | ce | Command code for the command in error. |
| 12 | ec | Error cause (see Table 3-6 on page 3-65). |
| 13–n | billing | Variable-size field containing billing information, if the billing option is requested at NPSI generation. See page 3-24 for billing format details. |

Table 3-6 on page 3-65 shows the error cause codes contained in the FTPI error and information reports.

*Table 3-6. FTPI Error and Information Report Cause Codes*

| Error Cause Code | Error Description |
|---|---|
| X'04' | The MCHID set by the CTCP is unknown. |
| X'05' | The command from the CTCP contains an invalid parameter. |
| X'06' | The VC id set by the CTCP is unknown. |
| X'08' | The X.25 resource selected by the CTCP in the command has no corresponding SNA resource (VC cleared and CTCP not yet informed or CTCP error). |
| X'09' | The correlation number in the CTCP command does not match the NPSI correlation number. |
| X'0C' | The command code set by the CTCP selects a non-supported function. |
| X'10' | Command received for a PVC for which the PU is inactive. |
| X'17' | The command set by the CTCP cannot be executed due to MCH failure. |
| X'1C' | The command set by the CTCP selects a function currently not executable due to a lack of SNA or X.25 resources. |
| X'1F' | VC cleared (SVC) or reset (PVC); overflow threshold reached. |
| X'30' | Timer elapsed before a response to a control packet was received from the network. |
| X'40' | Command received from wrong CTCP. |

# Chapter 4. Programming a DATE CTCP

# Chapter 4. Programming a DATE CTCP

This chapter explains how to program virtual circuits using the DATE function of NPSI.

For an example of programming a DATE CTCP, see Appendix, “Example of Programming a DATE CTCP.”

## DATE Function of NPSI

With the DATE function of NPSI, you can use all the LLC types, except LLC type 4 (GATE) and the integrated PAD support option of LLC type 5. You can use these LLC types to gain greater control of NPSI control information than is possible with just NPSI LLC functions. The DATE CTCP processes all control packets. This feature provides maximum flexibility in the processing of these packet types. The control packets include virtual circuit setup and termination, as well as Interrupt, Reset, and Qualified Data packets.

Figure 4-1 on page 4-2 shows the data flow for a network configuration that uses DATE.

**Note:** You can use the integrated PAD support option of LLC type 5.

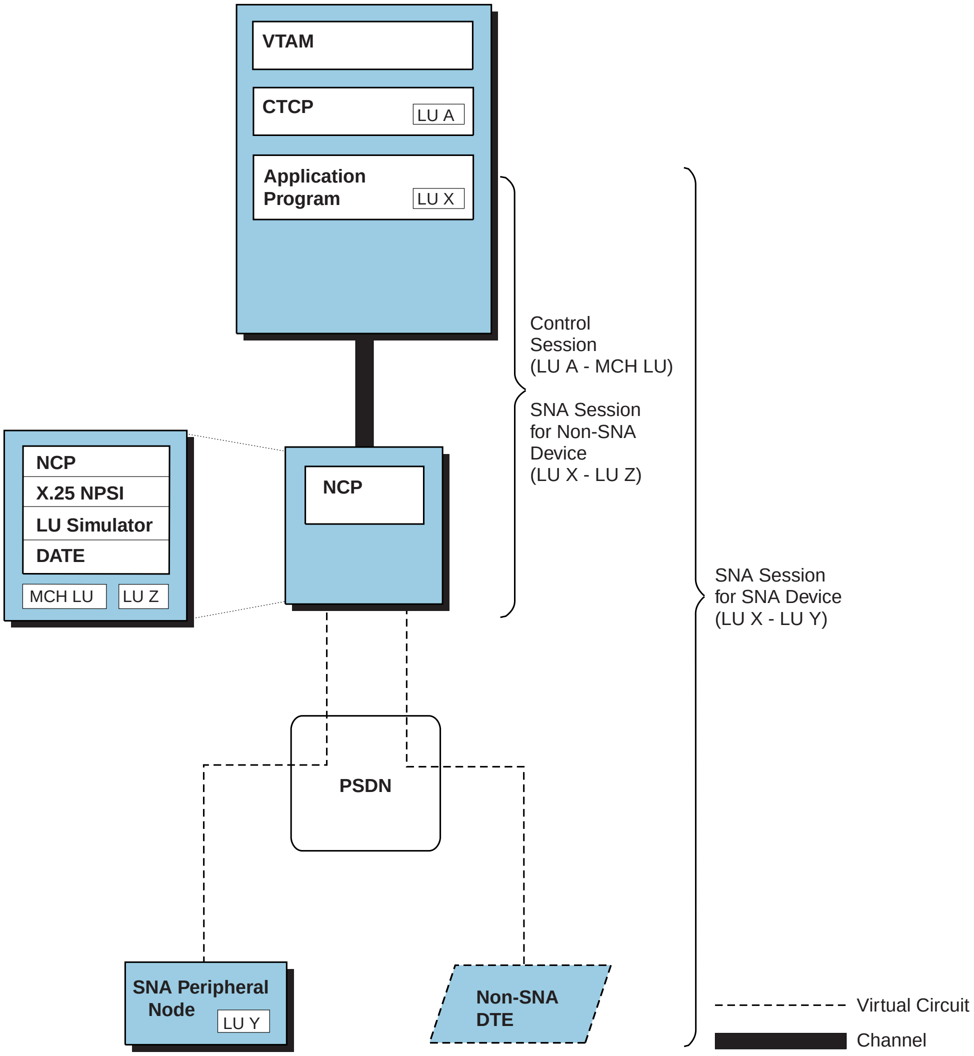


*Figure 4-1. SNA Host Node to SNA Peripheral Node or Non-SNA DTE (DATE)*

## CTCP Interface with DATE

The type of data handled by the application program is determined by the virtual circuit type:

- Type 0

  The application program sends and receives data packets. The CTCP sends or receives Restart, Call, Clear, Reset, Interrupt, and Qualified Data packets.

- Type 2

  The application program sends and receives the data packets. The CTCP sends or receives Restart, Call, Clear, and Reset packets. Interrupt and Qualified Data packets are discarded.

- Type 3

  The application program sends and receives data packets. The CTCP sends or receives Restart, Call, Clear, and Reset packets. Interrupt packets are discarded, and Qualified Data packets are handled by NPSI.

- Type 5

  The application program sends or receives all packets except Restart, Call, and Clear packets. These packets are handled by the CTCP.

## Determination of Virtual Circuit Type

The CTCP communicates the virtual circuit type (LLC type) to NPSI through the use of byte 6 of the CALL REQUEST or CALL ACCEPTED command. The value used in this byte is equal to the value used in the first byte of the CUD field of the Call Request packet to select LLCs. The hexadecimal code for each LLC type is:

| Hex Code | Virtual Circuit Type |
|---|---|
| X'C0' | LLC0 |
| X'C2' | LLC2 |
| X'C3' | LLC3 |
| X'01' | LLC5 |
| X'41' | LLC5 |
| X'51' | LLC5 |
| X'81' | LLC5 |

**Notes:**

1. A type 5 virtual circuit using the DATE function can use either the transparent or integrated PAD support function.
2. SVCSC is not supported on an MCH under DATE control.

## PLP Timers

The CTCP is responsible for managing the PLP states and the PLP timers. When you are not using DATE, the default values used by NPSI for these timers are the values defined in the Recommendation X.25. You can change these values when you program a CTCP; or by using the T20, T21, T22, and T23 keywords on the X.25 NET statement. The defaults are:

- Restart Request time out (T20 timer)

  This timer is started by the CTCP when a RESTART REQUEST command is sent to NPSI. The response must arrive within 180 seconds.

- Call Request time out (T21 timer)

  This timer is started by the CTCP when a CALL REQUEST command is sent to NPSI. The response must arrive within 200 seconds.

- Reset Request timeout (T22 timer)

  This timer is started by the CTCP when a RESET REQUEST command is sent to NPSI. The response to this request must arrive within 180 seconds.

- Clear Request timeout (T23 timer)

  This timer is started by the CTCP when a CLEAR REQUEST command is sent to NPSI. The Clear Confirmation response must be received within 180 seconds.

**Note:** Only the applicable timers are used for a specific virtual circuit, because not all of the control packets are passed to the CTCP for all virtual circuit types.

## Control Session Establishment

To communicate with a PSDN through DATE, the CTCP must first establish a session with the MCH LU. This session is a type 1 LU-LU session. Only after the START DATA TRAFFIC command is exchanged is the CTCP able to monitor the DATE function. The format of the BIND SESSION command is the same as the GATE format. See “Session Establishment through an SVC” on page 3-2.

**Note:** Half-duplex contention mode must be used on the CTCP to MCH LU session.

When the CTCP initiates session setup, the CTCP starts by sending a RESTART command to NPSI. The T20 timer associated with this command must be started by the CTCP. Upon receipt of the RESTART command, NPSI clears all the virtual circuits associated with the physical circuit being restarted and initiates the link setup procedure.

The link-level setup procedure is executed in two steps:

1. Link-level initiation is accomplished by the following:
   - NPSI sends a disconnect (DISC) frame to the DCE and waits for a disconnect mode (DM) frame or an unnumbered acknowledgment (UA) frame.
   - NPSI transmits an set asynchronous balanced mode (SABM) frame to the DCE and waits for a UA frame to be returned.
   - The DCE positively responds with a UA frame. At this point, the link-level setup is complete.
2. Packet level initiation is accomplished by the following:
   - DATE sends the Restart packet to the network.
   - The DCE can respond with a Restart Confirmation packet, or it might have already sent a Restart packet.
   - The packet received is passed on to the CTCP.
   - If the T20 timer has not expired, it is now stopped.

The CTCP must initiate setup of the physical circuit dedicated to the DATE function by sending a RESTART command. When link-level setup is performed correctly, DATE sends the Restart Request packet corresponding to the command passed by the CTCP.

The response from the PSDN is passed to the CTCP as a RESTART CONFIRMATION command or a RESTART command (in restart collision). In the case of restart collision, the CTCP does not need to respond with a RESTART CONFIRMATION command when it receives a RESTART command. When the CTCP receives a RESTART command or a RESTART CONFIRMATION command, it must then stop the T20 timer.

Once the control session with the physical circuit is established, the virtual circuits can be established. At this point, the virtual circuits are ready to operate as specified by the user application.

**V3R6 and Later:** The DATE CTCP can issue the MLUNAME command once RESTARTs have been exchanged. The procedure for issuing the MLUNAME command is the same for GATE.

## Virtual Circuit Establishment

For SVCs, the CTCP must determine whether a user application program is ready to be called, or is waiting to call out. The CTCP must know the LLC type used with a given application program. The CTCP passes this information to DATE in the CALL REQUEST or CALL ACCEPTED command. DATE then updates the corresponding control blocks in NPSI to allow the usual procedure at the data level.

## Call-In

When DATE receives an Incoming Call packet, it transfers the packet to the CTCP as an INCOMING CALL command, along with an indication of the appropriate SNA resource ID. Using the INCOMING CALL command, the CTCP determines the type of virtual circuit to be set up. The CTCP also selects the packet length and packet window size to be used, based on the PSDN subscription parameters. Then, the CTCP passes this information to DATE along with the application PLU name (if appropriate) through a CALL ACCEPTED command. DATE translates the CALL ACCEPTED command into a Call Accepted packet and forwards it to the PSDN.

## Call-Out

When an application program or operator requests that the CTCP connect an application with a destination, the CTCP sends a CALL REQUEST command to DATE. In this command, the CTCP passes on all the necessary information: virtual circuit type, packet length, packet window size, and an application PLU name for logon (if appropriate). The CTCP also sends a connection identifier that is used to associate the resource identifier that is passed to the CTCP in the CALL CONNECTED command, with the CALL REQUEST command.

Upon receipt of the CALL REQUEST command, the DATE code in NPSI selects a free virtual circuit to send a Call Request packet to the PSDN. NPSI then associates that virtual circuit with an SNA resource set. The CALL REQUEST command passed by the CTCP is forwarded by NPSI to the PSDN.

If the connection is successful, the PSDN sends a Call Connected packet back to NPSI. DATE then transfers the packet to the CTCP in a CALL CONNECTED command. The CALL CONNECTED command associates the CTCP connection identifier with the resource ID associated with the virtual circuit for the duration of the call. Thus, the CTCP is aware of the resource ID that is used for this connection.

At this time, DATE causes the NCP to send a REQUEST CONTACT request unit to VTAM for that virtual circuit. Thus, a connection is established directly between the user application program and the corresponding remote DTE over the chosen virtual circuit.

Once a virtual circuit is active, the application program communicates data RUs *directly* with the remote DTE, without going through the CTCP or DATE.

**Notes:**

1. Call request is simulated by an incoming call to NCP and the access method. Consequently, no PATH statement is required in the switched major node (SMN) for a DATE call request.
2. A connection to a virtual circuit attached to a DATE MCH must be performed under the control of the CTCP. If an application tries to connect out directly through the access method on such a virtual circuit, DATE generates an INOP message for the involved virtual circuit.
3. The LLC type is determined by the CTCP. With DATE, byte 0 of the CUD field is not used by NPSI to select the LLC type. The LLC type is passed to NPSI through byte 6 of the CALL REQUEST or CALL ACCEPTED command. Similarly, the packet size and window size are determined by the CTCP. Their values are passed to NPSI in bytes 3, 4, and 5 of the CALL REQUEST or CALL ACCEPTED command.

   In the case of a CALL REQUEST sent by the CTCP, packet and window sizes specified in the CALL REQUEST command are taken into account by NPSI. The flow control parameters contained in the facilities of the Call Connected packet are ignored by NPSI. The CTCP should clear the call and try again if these parameters do not match the flow control parameters that were passed to NPSI in the CALL REQUEST command.

   Depending on the generation option, the flow control parameters included in the facilities of the Call Connected packet are taken into account by NPSI.

If the application is to decide when to initiate a connection, it must send a message to the CTCP, either directly or through the operator, to ask the CTCP to issue a CALL REQUEST. The application cannot directly trigger a CALL REQUEST through an OPNDST OPTCD=ACQUIRE, or a SIMLOGON followed by an OPNDST OPTCD=ACCEPT, as would a normal application.

An SNA session using the virtual circuit can be established in one of the following ways:

- A LOGON command is generated by DATE for LLC0 and LLC5 virtual circuits.
- A remote user enters LOGON or an equivalent command.
- A logon is performed automatically because of LOGAPPL coded on the LU statement at call setup.
- An operator issues a V NET,LOGON=*application* command.

- An operator issues a V NET,ACT command, and LOGAPPL=*application* is coded on the LU statement.
- CLIST does a V NET,LOGON=*application* when the LU is active.

## Logging on to the Application

For DATE, if either LLC0 or LLC5 is selected for the call, the CTCP can request that NPSI build a logon with a specified application. In this case, NPSI builds the logon at call confirmation time.

When the CALL ACCEPTED command or the CALL REQUEST command carries an application name (when byte 7 is not equal to 0), DATE builds a logon message with the following format:

```
LOGON APPLID(xxxxxxxx) DATA(yyyyzzzzzzzz)
```

***where:***

`xxxxxxxx` Is the application program name given in the CALL REQUEST or CALL ACCEPTED command (up to 8 characters).

`yyyy` Is the 2-byte SNA resource ID.

`zzzzzzzz` Is the physical circuit LU name (up to 8 characters; padded to the right with blanks if less than 8 characters). zzzzzzzz is obtained from the control session BIND or from the MLUNAME command. See “Session Establishment through an SVC” on page 3-2.

This logon request is issued to the access method after the ACTLU command is processed for the involved LU associated with the virtual circuit for the duration of the call. When any of the following conditions are true, all data is queued until SDT is received:

- MAXOUT=6 is coded on the X25.PU statement or the X25.VC statement for PVCs.
- MAXOUT=6 is coded on the PU statement of the switched major node for SVCs.
- The logon is generated by NPSI.

Once SDT is received, all data traffic is passed to the application.

## Protocol for a Type 0 Switched Virtual Circuit

When an SVC using LLC type 0 is requested (X'C0' in byte 6 of the CALL REQUEST command or CALL ACCEPTED command), the logon message can be either the first data received by DATE from the remote DTE or a logon message built by DATE. In the second case, the CTCP gives the name of the application program to DATE in the CALL REQUEST or CALL ACCEPTED command.

The logon message is sent as data from the selected LU attached to the virtual circuit for the duration of the call, through the access method, to the LU of the application program. The logon exit routine is scheduled. Execution of the VTAM OPNDST macro that is coded with OPTCD=ACCEPT in the application program causes the SNA BIND SESSION and START DATA TRAFFIC commands to flow. These commands establish the LU-LU session associated with the selected SVC.

The application program then works under standard PCNE protocol, except when control packets, Interrupt packets, or qualified packets are received. The DATE code transmits the control, Interrupt, and qualified packets to the CTCP for processing on the MCH LU to CTCP LU session. DATE acts only on the Restart and Clear packets.

## Protocol for a Type 2 Switched Virtual Circuit

An SVC LLC type 2 is requested by X'C2' in byte 6 of the CALL REQUEST or CALL ACCEPTED command. After the call exchange is performed, the SVC is available to the application program, as in the case of a permanent type 2 virtual circuit.

The CTCP must establish an interface with the user application program to synchronize the different activities. A programmed operator interface should be developed with the CTCP.

## Protocol for a Type 3 Switched Virtual Circuit to SNA Peripheral Node

An SVC LLC type 3 is requested by X'C3' in byte 6 of the CALL REQUEST or CALL ACCEPTED command. After the call exchange is performed, the SVC is available to the application program, as in the case of a permanent type 3 virtual circuit.

The CTCP must establish an interface with the user application program to synchronize the different activities. A programmed operator interface should be developed with the CTCP.

For all type 3 virtual circuits, Interrupt packets and qualified packets are not authorized on the CTCP interface. The Interrupt packets are discarded; the qualified packets are processed by NPSI, as if the MCH was not under DATE.

## Protocol for a Type 3 Switched Virtual Circuit to SNA Subarea Node

For all SVCs using LLC type 3, Interrupt packets and qualified packets are not authorized on the CTCP interface. The Interrupt packets are discarded; the qualified packets are processed by NPSI, as if the MCH was not under DATE.

SVCSC is not supported on a DATE MCH, and only PVCs are supported.

## Protocol for a Type 5 Virtual Circuit

When an SVC using LLC type 5 is selected (X'01', X'41', X'51', or X'81' in byte 6 of the CALL REQUEST or CALL ACCEPTED command), it is treated as if it were a type 0 virtual circuit for the logon message processing. Following the logon, the standard processing for a transparent PAD virtual circuit occurs. During the session, Reset, Interrupt, and qualified packets are handled directly by the application.

**Notes:**

1. NPSI supports both transparent and integrated PAD support function. When an SVC using the integrated PAD support function of LLC type 5 is selected, the SVC is treated as if it were a type 0 virtual circuit for the logon message processing. Following the logon request and upon receipt of Interrupt, Reset, or qualified packets with BREAK specified, NPSI sends an SNA SIGNAL to the application and confirmation to the network. See “SIGNAL and BREAK

Processing” on page 2-10 for more information. During the session, Interrupt, Reset, and qualified packets (because of a BREAK entered at the terminal) are not authorized on the CTCP interface.

2. The association between the selected LU and the virtual circuit lasts for the duration of the call.

## Negotiation

The DCE can negotiate with the CTCP for the packet length and window size that apply to this virtual circuit. DATE keeps the values initially defined by the CTCP. The CTCP must analyze the CALL CONNECTED command to check whether a negotiation was performed, and to clear the connection, if necessary, before sending a new CALL REQUEST command containing negotiated values to update DATE.

**Note:** Depending on the generation option specified, the flow control parameters included in the facilities of the Call Connected packet are taken into account by NPSI.

# Virtual Circuit Termination

A virtual circuit can be deactivated using any of the following requests and commands:

- Clear Request from remote DTE
- CLSDST from the application program
- Request from the application program for the CTCP to CLEAR
- V NET,INACT command
- V NET,TERM command

In the simplest case, the remote DTE requests the session termination. In this instance:

1. A Clear Request packet is received from the PSDN, and NPSI transfers this information to the CTCP as a CLEAR REQUEST command.
2. The CTCP returns a CLEAR CONFIRMATION command to NPSI.
3. NPSI then generates an INOP for the virtual circuit LU.

The user application program can initiate session termination by issuing the CLSDST macroinstruction. In this case:

1. NPSI receives an ABCONN request from the NCP and forwards an INFORMATION REPORT to the CTCP with a cause field of X'07'.
2. The CTCP then sends a CLEAR command to NPSI that is forwarded to the PSDN.
3. When the Clear Confirmation packet returns from the PSDN, it is passed to the CTCP as a CLEAR CONFIRMATION command by NPSI. At this point, deactivation is complete, and the SNA and X.25 resources are disassociated.

The user application program can notify the CTCP to disconnect the virtual circuit. When the CTCP receives this request:

1. The CTCP builds a CLEAR command, including the cause and diagnostic bytes as determined by the CTCP.

2. The CLEAR command is sent to NPSI.
3. NPSI then sends a Clear Request packet to the PSDN.
4. When the Clear Confirmation packet is received, NPSI transfers it to the CTCP in the CLEAR CONFIRMATION command.
5. NPSI then generates an INOP to be sent to VTAM.
6. The controlling application program's LOSTERM exit is scheduled, and the application program issues a CLSDST macroinstruction. At that point, the virtual circuit is freed and available for reuse.

The network operator can terminate the session by issuing a V NET,INACT command for the virtual circuit PU or the virtual circuit LU or a V NET,TERM command for the virtual circuit LU. When this command is issued, an Information Report message is sent to the CTCP. The CTCP must issue a CLEAR command to resynchronize communication between the CTCP and NPSI.

## Control Session Termination

The DATE control session can be terminated in any of the following ways:

- The application program asks the CTCP to disconnect from the MCH LU.
- The operator issues the CANCEL command to cancel the CTCP.
- The operator issues the V NET,INACT,ID=*mch lu* or the V NET,TERM,LU=*mch lu* command to terminate the session with the physical circuit LU. *mch lu* is the name of the physical circuit LU.
- The CTCP abends.

**Notes:**

1. If the CTCP is canceled, abends, or is inactivated (through the V NET,INACT or V NET,TERM command), the SMNs used for communication with this CTCP must be deactivated and reactivated. This deactivation and reactivation clears any condition that is pending on the involved virtual circuits.
2. If the CTCP is terminated, the MCH LINK status is changed to INOP. The MCH LINK must be reactivated before the MCH can be reused.

## Command Interface between DATE and the CTCP

Control and qualified data commands are exchanged between the CTCP LU and the MCH LU. With the exception of commands flowing on type 5 virtual circuits, commands exchanged between the application program LU and the LU associated with the virtual circuit do not contain command codes.

Command codes use conventions defined in the X.25 protocol. The command code is always in byte 2 of an RU flowing on a DATE session. For data exchange, byte 2 can have the value X'00', indicating data exchange without the Q bit, or the value X'02', indicating data exchange with the Q bit. When control commands from the CTCP to DATE contain optional user data, this data is put into the resulting control packet without verification by DATE. Segmenting using the M bit in the packet header is *not* performed for control packets. For this reason, all user data within control packets must fit into a single packet. For qualified or unqualified data packets, user data can be spread over several packets.

For control commands, the other bytes of the RU also have significance to the CTCP and DATE. When sending commands to DATE, the CTCP must adhere to the command formats described in this section. For commands going from DATE to the CTCP, the CTCP uses these formats to interpret the command. The CTCP establishes one session with the DATE function within NPSI to handle the exchange of the following commands. These commands are contained in RUs that flow through the SNA session between the CTCP and the MCH LU.

Table 4-1 on page 4-11 shows the commands that are processed by the CTCP over the physical circuit LU (MCH LU) session for each of these virtual circuit types.

*Table 4-1. Commands on Physical Circuit LU Session*

| **Commands** | **Command Code** | **Type 0** | **Type 2** | **Type 3 (Peripheral Node)** | **Type 3 (Subarea Node)** | **Type 5** |
|---|---|---|---|---|---|---|
| ERROR/INFORMATION REPORT | X'00' | Yes | Yes | Yes | Yes | Yes |
| CALL REQUEST and INCOMING CALL | X'0B' | Yes * | Yes * | Yes * | No | Yes * |
| CALL ACCEPTED and CALL CONNECTED | X'0F' | Yes * | Yes * | Yes * | No | Yes * |
| CLEAR REQUEST and CLEAR INDICATION | X'13' | Yes * | Yes * | Yes * | No | Yes * |
| CLEAR CONFIRMATION | X'17' | Yes * | Yes * | Yes * | No | Yes * |
| DIAGNOSTIC | X'F1' | Yes | Yes | Yes | Yes | Yes |
| MLUNAME | X'F4' | Yes | Yes | Yes | Yes | Yes |
| RESTART REQUEST and RESTART INDICATION | X'FB' | Yes | Yes | Yes | Yes | Yes |
| RESTART CONFIRMATION | X'FF' | Yes | Yes | Yes | Yes | Yes |

**Note:** The commands followed by an asterisk (*) are processed by the CTCP on the MCH LU session *only* for SVCs.

Table 4-2 shows, by virtual circuit type, the commands processed on the application program to virtual circuit LU (VC LU) session and those processed on the CTCP to physical circuit LU (MCH LU) session.

*Table 4-2. Commands on Virtual Circuit or Physical Circuit LU Session*

| **Commands** | **Command Code** | **Type 0** | **Type 2** | **Type 3 (Peripheral Node)** | **Type 3 (Subarea Node)** | **Type 5** |
|---|---|---|---|---|---|---|
| UNQUALIFIED DATA | X'00' | MCH LU or VC LU | MCH LU or VC LU | MCH LU or VC LU | MCH LU or VC LU | MCH LU or VC LU |
| QUALIFIED DATA | X'02' | MCH LU | No | No | No | VC LU * |
| RESET REQUEST and RESET INDICATION | X'1B' | MCH LU | MCH LU | MCH LU | MCH LU | VC LU * |
| RESET CONFIRMATION | X'1F' | MCH LU | MCH LU | MCH LU | MCH LU | VC LU * |
| INTERRUPT REQUEST | X'23' | MCH LU | No | No | No | VC LU * |
| INTERRUPT CONFIRMATION | X'27' | MCH LU | No | No | No | VC LU * |

**Note:** The commands followed by an asterisk (*) are carried on the data session between the host application and the remote DTE. They are processed by NPSI transparent PAD support. The format of these commands is the same as the format of the transparent PAD RUs on page 2-17.

**Note:** When using integrated PAD, RESET INDICATION, and RESET CONFIRMATION commands can flow either on the VC LU or on the MCH LU session.

## Identifiers

The connection identifier (CNID) referred to in the following command descriptions is a 2-byte field. It is chosen by the CTCP, when the CTCP makes a CALL REQUEST. It is then used by the CTCP to correlate the CALL REQUEST with the subsequent commands (CALL CONNECTED or CLEAR) related to the CNID. This field is used temporarily during the CALL REQUEST setup. Once the call is complete, it is no longer used.

The resource identifier described in the following paragraph is used when the call is complete. The leftmost halfbyte of the 2-byte field is X'0', for outbound and inbound commands, except for the CLEAR command from the CTCP after a CALL REQUEST.

The resource identifier (RESID) field designates the set of SNA resources (LINE, PU, and LU) that are associated with either a virtual circuit, or the X.25 virtual circuit number (systems generation option VCID). SNA resource IDs are associated with SVCs for each MCH. The SNA resource IDs correspond to the number of SNA resources within the set of SNA resources associated with the MCH. The range for the resource ID numbers is the same as the range for the virtual circuit numbers. An SNA resource number consists of 3 digits in hexadecimal format. The first digit maps the logical group number, and the next two digits map the logical channel numbers.

The CNID and RESID must have different ranges to avoid problems at Call Collision.

Figure 4-2 shows how you can code the X25.MCH statement to define SVCs. In this example, SNA IDs X'003' to X'063' and SNA IDs X'204' to X'262' are used to map all SVCs of this MCH. This setup allows CTCPs to continue to run without change.

```
X25.LCG  LCGN=0
X25.VC   LCN=(1,2),TYPE=P ....  2 PVCs
X25.VC   LCN=(3,99),TYPE=S ... 97 SVCs
X25.LCG  LCGN=2
X25.VC   LCN=(1,3),TYPE=P ....  3 PVCs
X25.VC   LCN=(4,98),TYPE=S ... 95 SVCs
```

*Figure 4-2. Coding X25.MCH Statement to Define SVCs*

The following descriptions show data exchange commands and control commands. The data exchange commands are followed by user data. The control commands are followed by additional control information.

## DATA EXCHANGE without Q Bit (CTCP to DATE through the Control Session)

Data without a Q bit can be sent out from the CTCP to DATE through the control session. The command is sent in the following format:

```
    ┌───────┬────┬──//──┐
RU: │ resid │ cc │ user │
    └───────┴────┴──//──┘
```

| Bytes | Field | Description |
|---|---|---|
| 0–1 | resid | SNA resource identifier. |
| 2 | cc | Command code. For this command, the value is X'00'. |
| 3–n | user | Variable-size field containing user data to be sent to the remote terminal. For example, this field can be used at connection time to ask the remote user not to clear the connection. |

**Note:** For more information about this data exchange, see DATE Message on page 4-25.

## ERROR/INFORMATION REPORT (DATE to CTCP)

The ERROR/INFORMATION REPORT command is issued from DATE to the CTCP. The command is sent in the following format:

```
    ┌───────┬────┬────┬────┐
RU: │ resid │ cc │ ce │ ec │
    └───────┴────┴────┴────┘
```

| Bytes | Field | Description |
|---|---|---|
| 0–1 | resid | Resource identifier contained in bytes 0 and 1. The leftmost halfbyte is X'F'. |
| 2 | cc | Command code contained in byte 2. For this command, the value is X'00'. |
| 3 | ce | Command code for the command in error (contained in byte 3). |
| 4 | ec | Error cause. (See the list of error causes in Table 4-3 on page 4-25.) |

## DATA EXCHANGE with Q Bit (CTCP to DATE, DATE to CTCP through the Control Session)

A DATA EXCHANGE command with a Q bit is issued from the CTCP to DATE and from DATE to the CTCP through the control session for LLC type 0. (Exchange occurs on the data session for LLC type 5.) The command is sent in the following format:

```
    ┌───────┬────┬──//──┐
RU: │ resid │ cc │ user │
    └───────┴────┴──//──┘
```

| Bytes | Field | Description |
|---|---|---|
| 0–1 | resid | SNA resource identifier. |
| 2 | cc | Command code. For this command, the value is X'02'. |
| 3–n | user | Variable-size field containing user data related to this packet. For example, this field can be used to control a remote PAD. |

**Note:** If you are using transparent PAD, see the command formats for the transparent PAD RUs on page 2-17.

## CALL REQUEST (CTCP to DATE)

A CALL REQUEST command is issued from the CTCP to DATE. The command is sent in the following format:

```
    ┌──────┬────┬────┬──────┬────┬────┬──//──┐
RU: │ cnid │ cc │ pw │ psiz │ vc │ al │ crpk │
    └──────┴────┴────┴──────┴────┴────┴──//──┘
```

<table>
<tr><th>Bytes</th><th>Field</th><th>Description</th></tr>
<tr><td>0–1</td><td>cnid</td><td>Connection identifier contained in bytes 0 and 1.</td></tr>
<tr><td>2</td><td>cc</td><td>Command code contained in byte 2. For this command, the value is X'0B'.</td></tr>
<tr><td>3</td><td>pw</td><td>Packet window size in hexadecimal contained in byte 3.</td></tr>
<tr><td>4–5</td><td>psiz</td><td>Packet size in hexadecimal used for this virtual circuit contained in bytes 4 and 5.</td></tr>
<tr><td>6</td><td>vc</td><td>The type of virtual circuit to be set up. Code in byte 6 the corresponding hexadecimal code shown next to the virtual circuit type in the following list:
<table>
<tr><th>Hexadecimal Code</th><th>Virtual Circuit Type</th></tr>
<tr><td>X'C0'</td><td>0</td></tr>
<tr><td>X'C2'</td><td>2</td></tr>
<tr><td>X'C3'</td><td>3 (to SNA peripheral node)</td></tr>
<tr><td>X'01', X'41', X'51', or X'81'</td><td>5</td></tr>
</table>
<b>Notes:</b><br>
1. You cannot specify a type 4 virtual circuit.<br>
2. You can use transparent PAD or integrated PAD support with a type 5 virtual circuit.</td></tr>
<tr><td>7</td><td>al</td><td>The application name length (X'00' through X'08') contained in byte 7. A length of X'0' means that DATE must not generate the LOGON command.</td></tr>
<tr><td>8–n</td><td>crpk</td><td>Variable-size field starting in byte 8 that contains the application name followed by the Call Request packet (without the packet header).</td></tr>
</table>

## INCOMING CALL (DATE to CTCP)

The INCOMING CALL command is issued from DATE to the CTCP. The command is sent in the following format:

| | | | |
|---|---|---|---|
| RU: | resid | cc | icpk |

| Bytes | Field | Description |
|---|---|---|
| 0–1 | resid | Resource identifier contained in bytes 0 and 1. The leftmost halfbyte is X'F'. See note below. |
| 2 | cc | Command code contained in byte 2. For this command, the value is X'0B'. |
| 3–n | icpk | Variable-size field starting in byte 3 that contains the Call packet received from the network (without the 3-byte packet header). |

In the case of a Call Collision detected by NPSI, NPSI sends the Call Request to the network, and when NPSI receives the Incoming Call, it passes the Incoming Call to the CTCP. This Incoming Call contains a 2-byte connection identifier in bytes 0 and 1. The first digit is X'0', so that the CTCP can distinguish it from a normal Incoming Call. The CTCP can then discard this INCOMING CALL command (under normal circumstances, the network has already discarded the Incoming Call) and wait for the CALL CONNECTED command, or it might send a CLEAR REQUEST command that clears the Call Request.

## CALL ACCEPTED (CTCP to DATE)

The CALL ACCEPTED command is issued from the CTCP to DATE. The command is sent in the following format:

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| RU: | resid | cc | pw | psiz | vc | al | capk |

<table>
<tr><th>Bytes</th><th>Field</th><th>Description</th></tr>
<tr><td>0–1</td><td>resid</td><td>Resource identifier contained in bytes 0 and 1. This resource identifier matches the one given in bytes 0 and 1 of the corresponding INCOMING CALL command from DATE.</td></tr>
<tr><td>2</td><td>cc</td><td>Command code contained in byte 2. For this command, the value is X'0F'.</td></tr>
<tr><td>3</td><td>pw</td><td>Packet window size contained in byte 3.</td></tr>
<tr><td>4–5</td><td>psiz</td><td>Packet size, in hexadecimal, contained in bytes 4 and 5.</td></tr>
<tr><td>6</td><td>vc</td><td>The type of virtual circuit to be set up. Code in byte 6 the corresponding hexadecimal code shown next to the virtual circuit type in the following list:
<table>
<tr><th>Hexadecimal Code</th><th>Virtual Circuit Type</th></tr>
<tr><td>X'C0'</td><td>0</td></tr>
<tr><td>X'C2'</td><td>2</td></tr>
<tr><td>X'C3'</td><td>3 (to SNA peripheral node)</td></tr>
<tr><td>X'01', X'41', X'51', or X'81'</td><td>5.</td></tr>
</table>
<b>Notes:</b>
<ol>
<li>You cannot specify a type 4 virtual circuit.</li>
<li>You can use transparent PAD or integrated PAD support with a type 5 virtual circuit.</li>
</ol></td></tr>
<tr><td>7</td><td>al</td><td>The application name length (X'00' through X'08') contained in byte 7. A length of X'0' means that DATE must not generate a LOGON command and there will not be an application name starting at byte 8.</td></tr>
<tr><td>8–n</td><td>capk</td><td>Variable-size field starting in byte 8 that contains the application name followed by the Call Accepted packet (without the 3-byte packet header).</td></tr>
</table>

## CALL CONNECTED (DATE to CTCP)

The CALL CONNECTED command is issued from DATE to the CTCP. The command is sent in the following format:

```
     ┌───────┬────┬──────┬──//─┐
RU:  │ resid │ cc │ cnid │ ccpk │
     └───────┴────┴──────┴──//─┘
```

| Bytes | Field | Description |
|---|---|---|
| 0–1 | resid | Resource identifier contained in bytes 0 and 1. The leftmost halfbyte is X'F'. |
| 2 | cc | Command code contained in byte 2. For this command, the value is X'0F'. |
| 3–4 | cnid | Connection identifier contained in bytes 3 and 4. This connection identifier matches the one given in bytes 0 and 1 of the corresponding CALL REQUEST command. |
| 5–n | ccpk | Variable-size field starting in byte 5 that contains the Call Connected packet (without the packet header). |

**Note:** Depending on the generation option specified, the flow control parameters included in the facilities of the Call Connected packet are taken into account by NPSI.

## CLEAR on Incoming Call (CTCP to DATE)

The CLEAR command on the incoming call is issued from the CTCP to DATE. The command is sent in the following format:

```
     ┌───────┬────┬──────┬──//─┐
RU:  │ resid │ cc │ crdf │ user │
     └───────┴────┴──────┴──//─┘
```

| Bytes | Field | Description |
|---|---|---|
| 0–1 | resid | Resource identifier contained in bytes 0 and 1. This resource identifier matches the one given in bytes 0 and 1 of the corresponding INCOMING CALL command from DATE. However, the first four bits should be X'0' (CTCP turns off the X'F' in the first 4 bits of the Incoming Call). |
| 2 | cc | Command code contained in byte 2. For this command, the value is X'13'. |
| 3–4 | crdf | The cause and diagnostic fields (contained in bytes 3 and 4) that are to be inserted in the Clear Request packet. |
| 5–n | user | Variable-size field starting in byte 5 that contains optional user data. |

## CLEAR on Outgoing Call (DATE to CTCP)

The CLEAR command on the outgoing call is issued from DATE to the CTCP. The command is sent in the following format:

```
    ┌───────┬────┬──────┬──────┬──//─┐
RU: │ resid │ cc │ cnid │ crdf │ user│
    └───────┴────┴──────┴──────┴──//─┘
```

| Bytes | Field | Description |
|---|---|---|
| 0–1 | resid | Resource identifier contained in bytes 0 and 1. The first digit is X'F'. |
| 2 | cc | Command code contained in byte 2. For this command, the value is X'13'. |
| 3–4 | cnid | Connection identifier contained in bytes 3 and 4. This connection identifier matches the one given in bytes 0 and 1 of the corresponding CALL REQUEST command. |
| 5–6 | crdf | Cause and diagnostic fields contained in bytes 5 and 6. These fields are taken from the Clear Request packet. |
| 7–n | user | Variable-size field starting in byte 7 that contains optional user data. |

## CLEAR after Outgoing Call (CTCP to DATE)

The CLEAR command is issued after the outgoing call from the CTCP to DATE. This CLEAR command can be used when the CTCP detects a time-out after a CALL REQUEST command has been sent. The command is sent in the following format:

```
    ┌──────┬────┬──────┬──//─┐
RU: │ cnid │ cc │ ccdf │ user│
    └──────┴────┴──────┴──//─┘
```

| Bytes | Field | Description |
|---|---|---|
| 0–1 | cnid | Connection identifier in bytes 0 and 1. The first digit must be set to X'F' to signify to DATE that this is a CNID rather than a RESID. |
| 2 | cc | Command code in byte 2. For this command, the value is X'13'. |
| 3–4 | ccdf | Cause and diagnostic bytes contained in bytes 3 and 4. |
| 5–6 | user | Variable-size field starting in byte 5 that contains optional user data. |

## CLEAR (DATE to CTCP, CTCP to DATE)

The CLEAR command is issued from DATE to the CTCP and from the CTCP to DATE. The command is sent in the following format:

```
    ┌───────┬────┬──────┬─//──┐
RU: │ resid │ cc │ ccdf │ user│
    └───────┴────┴──────┴─//──┘
```

| Bytes | Field | Description |
|---|---|---|
| 0–1 | resid | Resource identifier, contained in bytes 0 and 1, that is provided to the CTCP in the CALL CONNECTED or INCOMING CALL command. The leftmost halfbyte is X'0' when the direction is CTCP to DATE, and X'F' when the direction is DATE to CTCP. |
| 2 | cc | Command code contained in byte 2. For this command, the value is X'13'. |
| 3–4 | ccdf | Clear cause and diagnostic fields contained in bytes 3 and 4. |
| 5–n | user | Variable-size field starting in byte 5 that contains optional user data. |

## CLEAR CONFIRMATION after CLEAR Command for Incoming Call Refused (DATE to CTCP)

The CLEAR CONFIRMATION command is issued from DATE to the CTCP after the CLEAR command for an incoming call is refused. The command is sent in the following format:

```
    ┌───────┬────┐
RU: │ resid │ cc │
    └───────┴────┘
```

| Bytes | Field | Description |
|---|---|---|
| 0–1 | resid | Resource identifier contained in bytes 0 and 1. The first digit is X'F'. |
| 2 | cc | Command code contained in byte 2. For this command, the value is X'17'. |

## CLEAR CONFIRMATION (CTCP to DATE, DATE to CTCP)

The CLEAR CONFIRMATION command is issued from the CTCP to DATE and from DATE to the CTCP. The command is sent in the following format:

```
    ┌───────┬────┐
RU: │ resid │ cc │
    └───────┴────┘
```

| Bytes | Field | Description |
|---|---|---|
| 0–1 | resid | Resource identifier contained in bytes 0 and 1. The leftmost halfbyte is X'0' when the direction is CTCP to DATE, and X'F' when the direction is DATE to CTCP. |
| 2 | cc | Command code contained in byte 2. For this command, the value is X'17'. |

## RESET (DATE to CTCP, CTCP to DATE)

The RESET command is issued from DATE to the CTCP and from the CTCP to DATE.

**Note:** When using integrated PAD, the CTCP is not authorized to send Reset packets. Reset packets flow on the physical circuit LU session only if they are *not generated* by a BREAK entered at the terminal.

The command is sent in the following format:

```
    ┌───────┬────┬──────┬──//──┐
RU: │ resid │ cc │ rcdf │ user │
    └───────┴────┴──────┴──//──┘
```

| Bytes | Field | Description |
|---|---|---|
| 0–1 | resid | Resource identifier, contained in bytes 0 and 1, that is provided to the CTCP in the CALL CONNECTED or INCOMING CALL command. The leftmost halfbyte is X'0' when the direction is CTCP to DATE, and X'F' when the direction is DATE to CTCP. |
| 2 | cc | Command code contained in byte 2. For this command, the value is X'1B'. |
| 3–4 | rcdf | Reset cause and diagnostic fields contained in bytes 3 and 4. |
| 5–n | user | Variable-size field starting in byte 5 that contains optional user data. |

When NPSI detects an unrecoverable situation, NPSI generates a RESET command that is passed to the CTCP (or to the application in case of transparent PAD). This Reset packet is not sent to the PSDN. The CTCP (or the PAD application) must take the appropriate action, such as sending a RESET command, or clearing the connection, or ignoring the error.

| Type of Error | Reset CMD to CTCP |
|---|---|
| Packet received with invalid P(S) | Diag X'AB' |
| Packet received with invalid P(R) | Diag X'AC' |
| T25 timeout | Diag X'92' |

**Notes:**

1. See *NPSI Diagnosis, Customization and Tuning* for more information about NPSI Cause and Diagnostic codes.
2. If you are using transparent PAD, see the command formats in Table 2-1 on page 2-18.

# RESET CONFIRMATION (CTCP to DATE, DATE to CTCP)

The RESET CONFIRMATION command is issued from the CTCP to DATE and from DATE to the CTCP. The command is sent in the following format:

```
    ┌───────┬────┐
RU: │ resid │ cc │
    └───────┴────┘
```

| Bytes | Field | Description |
|---|---|---|
| 0–1 | resid | Resource identifier contained in bytes 0 and 1. The leftmost halfbyte is X'0' when the direction is CTCP to DATE, and X'F' when the direction is DATE to CTCP. |
| 2 | cc | Command code contained in byte 2. For this command, the value is X'1F'. |

**Notes:**

1. At RESET CONFIRMATION, the PLP counters are reset to 0, but the virtual circuit remains active. The CTCP may clear the virtual circuit, if necessary.
2. If you are using transparent PAD, see the command formats for the transparent PAD RUs on page 2-17.

# INTERRUPT (CTCP to DATE, DATE to CTCP)

The INTERRUPT command is issued from the CTCP to DATE, from DATE to the CTCP, and is used for LLC0.

**Note:** When using integrated PAD, Interrupt packets flow on the virtual circuit LU session. They are not authorized on the physical circuit LU session.

The command is sent in the following format:

```
    ┌───────┬────┬──────┬─//──┐
RU: │ resid │ cc │  iu  │ user │
    └───────┴────┴──────┴─//──┘
```

| Bytes | Field | Description |
|---|---|---|
| 0–1 | resid | Resource identifier contained in bytes 0 and 1. The leftmost halfbyte is X'0' when the direction is CTCP to DATE, and X'F' when the direction is DATE to CTCP. |
| 2 | cc | Command code contained in byte 2. For this command, the value is X'23'. |
| 3 | iu | Interrupt user data contained in byte 3. |
| 4–n | user | Variable-size field starting in byte 4 that contains optional user data. |

**Note:** If you are using transparent PAD, see the command formats for the transparent PAD RUs on page 2-17.

## INTERRUPT CONFIRMATION (CTCP to DATE, DATE to CTCP)

The INTERRUPT CONFIRMATION command is issued from the CTCP to DATE and from DATE to the CTCP, and is used for LLC0.

**Note:** When using integrated PAD, Interrupt Confirmation packets flow on the virtual circuit LU session. They are not authorized on the physical circuit LU session.

The command is sent in the following format:

```
    ┌───────┬────┐
RU: │ resid │ cc │
    └───────┴────┘
```

| Bytes | Field | Description |
|---|---|---|
| 0–1 | resid | Resource identifier contained in bytes 0 and 1. The leftmost halfbyte is X'0' when the direction is CTCP to DATE, and X'F' when the direction is DATE to CTCP. |
| 2 | cc | Command code contained in byte 2. For this command, the value is X'27'. |

**Note:** If you are using transparent PAD, see the command formats for the transparent PAD RUs on page 2-17.

## DIAGNOSTIC (CTCP to DATE, DATE to CTCP)

The DIAGNOSTIC command is issued from the CTCP to DATE and from DATE to the CTCP. The command is sent in the following format:

```
    ┌──────┬────┬──────┬─//─┐
RU: │ F000 │ cc │ dc   │ od │
    └──────┴────┴──────┴─//─┘
```

| Bytes | Field | Description |
|---|---|---|
| 0–1 | F000 | Null identifier. |
| 2 | cc | Command code contained in byte 2. For this command, the value is X'F1'. |
| 3 | dc | Diagnostic code contained in byte 3. |
| 4–n | od | Variable-size field starting in byte 4 that contains diagnostic explanation. |

## MLUNAME (CTCP to DATE through the Control Session)

The MLUNAME (MCH LU name) command is issued from the CTCP to DATE through the control session. It is used to pass the MCH LU name to DATE when the user data portion of the BIND command does not carry it. The command is sent in the following format:

```
         ┌───────┬────┬──//──┐
RU:      │ F000  │ cc │mchslu│
         └───────┴────┴──//──┘
```

| Bytes | Field | Description |
|---|---|---|
| 0 | F000 | Null identifier containing X'F000' |
| 2 | cc | Command code. For this command, the value is X'F4'. |
| 3–n | mchslu | Variable-size field (up to 8 bytes) containing the MCH LU name. |

## RESTART (CTCP to DATE, DATE to CTCP)

The RESTART command is issued from the CTCP to DATE and from DATE to the CTCP. The command is sent in the following format:

```
         ┌──────┬────┬────┬────┐
RU:      │ x000 │ cc │ ca │ dg │
         └──────┴────┴────┴────┘
```

| Bytes | Field | Description |
|---|---|---|
| 0–1 | x000 | Null identifier containing X'x000', where x is X'0' when the direction is CTCP to DATE and X'F' when the direction is DATE to CTCP. |
| 2 | cc | Command code contained in byte 2. For this command, the value is X'FB'. |
| 3 | ca | Cause code contained in byte 3. |
| 4 | dg | Diagnostic code contained in byte 4. |

## RESTART CONFIRMATION (CTCP to DATE, DATE to CTCP)

The RESTART CONFIRMATION command is issued from the CTCP to DATE and from DATE to the CTCP. The command is sent in the following format:

```
         ┌──────┬────┐
RU:      │ x000 │ cc │
         └──────┴────┘
```

| Bytes | Field | Description |
|---|---|---|
| 0–1 | x000 | Null identifier containing X'x000', where x is X'0' when the direction is CTCP to DATE and X'F' when the direction is DATE to CTCP. |
| 2 | cc | Command code contained in byte 2. For this command, the value is X'FF'. |

# INFORMATION REPORT Messages Sent from DATE to the CTCP

The INFORMATION REPORT command allows for the exchange of information between DATE and the CTCP without involving the X.25 network. This INFORMATION REPORT message is sent from DATE to the CTCP. When the message is sent by DATE, the CTCP must respond to the code in the message, as shown in Table 4-3.

*Table 4-3. CTCP Responses to DATE INFORMATION REPORT Messages*

| **Error Cause** | **Reason Sent by DATE** | **Required Action** |
|---|---|---|
| X'00' | Current PLP state does not permit the sending of Reset, Interrupt, qualified data, or unqualified data packets | Correct the CTCP |
| X'01' | Virtual circuit type not consistent with DATE | Correct the CTCP |
| X'02' | Unknown format | Correct the CTCP |
| X'03' | Resource identifier not generated in NPSI | Correct the CTCP or the NPSI generation |
| X'04' | No virtual circuit available | The CTCP must retry a CALL REQUEST command |
| X'05' | Unsuccessful XIO to X.25 network | Send a message to the operator to check the MCH |
| X'06' | Physical circuit is not running | Issue a RESTART REQUEST command |
| X'07' | Application abandons connection | Issue a CLEAR REQUEST command |
| X'08' | Invalid packet size of the CALL REQUEST or CALL ACCEPTED command, or specified in the CALL REQUEST or CALL ACCEPTED command. | Correct the CTCP |

When NPSI sends an INFORMATION REPORT message, DATE does not take action to resolve the problem reported by the message to the CTCP. The receiving buffer that caused the message is released, and DATE waits for new instructions from the CTCP.

**Note:** When DATE cannot determine the virtual circuit where the problem occurred, the information report message is sent with a RESID of 0.

# DATE Message

A message can be sent from the CTCP LU to the MCH LU. This message is used to notify you of an impending session setup. The message can be used when the session setup will take a considerable amount of time to tell the operator that the session establishment is taking place, and that the operator should not clear the connection, even if this establishment takes a long time.

The CTCP uses the MCH LU session for this message, because of its constant availability and because data can be sent to the remote terminal before the virtual circuit session is active. Unlike the session to the VC LU, the session to the MCH LU is always available for use. This allows the message to be sent to the terminal operator, as soon as the call setup is complete but long before the SNA session setup for the virtual circuit is complete.

## Program Operator Interface (POI)

Before communication between the DATE CTCP and the application program can take place, a program operator interface (POI) must be created. This interface is used to pass requests and the current status information between the two programs.

An example of how the communication interface (created by POI) operates is illustrated by the application program sending the CTCP a request to acquire a particular device. Without the use of the POI, the request cannot be sent because the application program requires a session with the MCH LU. This MCH LU is held by the CTCP, and VC LU can be acquired only through this session. The POI allows the necessary communication to notify the CTCP of the desired application.

In addition to handling requests, the POI must also be able to asynchronously notify the session partner of a change in virtual circuit status. A status change can be caused by a number of events. The following are examples of events that can cause status changes:

- Failure of a virtual circuit
- Activation of a virtual circuit LU
- Operator deactivation of a virtual circuit LU

## Methods of Creating a POI

The CTCP and the application program must be able to communicate before a POI can be created. This communication can use any method, including the following:

- SNA session

  This session can use any session protocol that is acceptable by both sides of the communication channel.

- Cross-memory services

  When cross-memory services are supported by the operating system.

- Connection through NPSI using PCNE.

The method of communication must support asynchronous requests between the CTCP and the application program. Although any SNA protocol allows for this method of communication, the use of SNA indicators signaling a change of direction or bracketing causes additional communication difficulties. These difficulties are caused by the problems associated with asynchronous requests that run counter to the direction of the communication path established at the time of the request.

For example, communication problems occur if the session is in the in-bracket (INB) state, and the contention-losing partner must notify the other partner of an event. To eliminate the difficulty, a private protocol must be established that notifies the other partner that a message needs to be sent. When the private protocol is established, the contention-losing partner transmits a SIGNAL request that notifies the contention-winning partner of the need to reverse the flow of data. The contention-winning partner then responds by relinquishing control of the session, which permits the contention-losing partner to transmit the message.

A requirement for a more complex data flow can demand that a private protocol be established, even though it follows normal SNA request flows. A private protocol allows the data flow between partners to be streamlined by permitting either of the following transmissions:

- Multiple requests transmitted at one time
- Transmission of a different method of asynchronous requests

The communication between the partners must have the flexibility to allow for any request. The POI should not:

- Restrict the type or size of the message
- Restrict the capacity of the interface
- Have a high utilization (this can change during a major network event, such as the failure of a communication controller)

# Chapter 5. Programming Host Applications

# Chapter 5. Programming Host Applications

This chapter presents the definitions and requirements needed to support application programs that communicate with resources through NPSI. Unusual application program occurrences, connection methods, and dial-in and dial-out methods are described.

## Programming CICS

The CICS subsystem can communicate with many types of SNA and non-SNA resources. The communication definitions for CICS are located in the terminal control table (TCT). This table defines device characteristics that include, but are not limited to:

- Network name
- Device type (such as 3270 or 3767)
- Access method
- Screen size and availability of an alternative size
- CICS terminal identifier

The CICS Terminal Control Program (TCP) obtains the profile of the device from the terminal control table to determine the correct terminal driver.

The TCP also performs protocol and device-specific processing. This processing isolates the application program from the terminal-specific requirements. For example, the terminal driver can chain the output data into elements of an acceptable size for the destination device and can request attachment to the destination resource. The application program, unaware of this processing, can therefore support different types of devices simultaneously.

The NPSI LUSIM creates the LU-LU communication between the host application program and a non-SNA DTE for non-SNA devices.

Figure 5-1 on page 5-2 shows the data flow in a network configuration using CICS and NPSI.

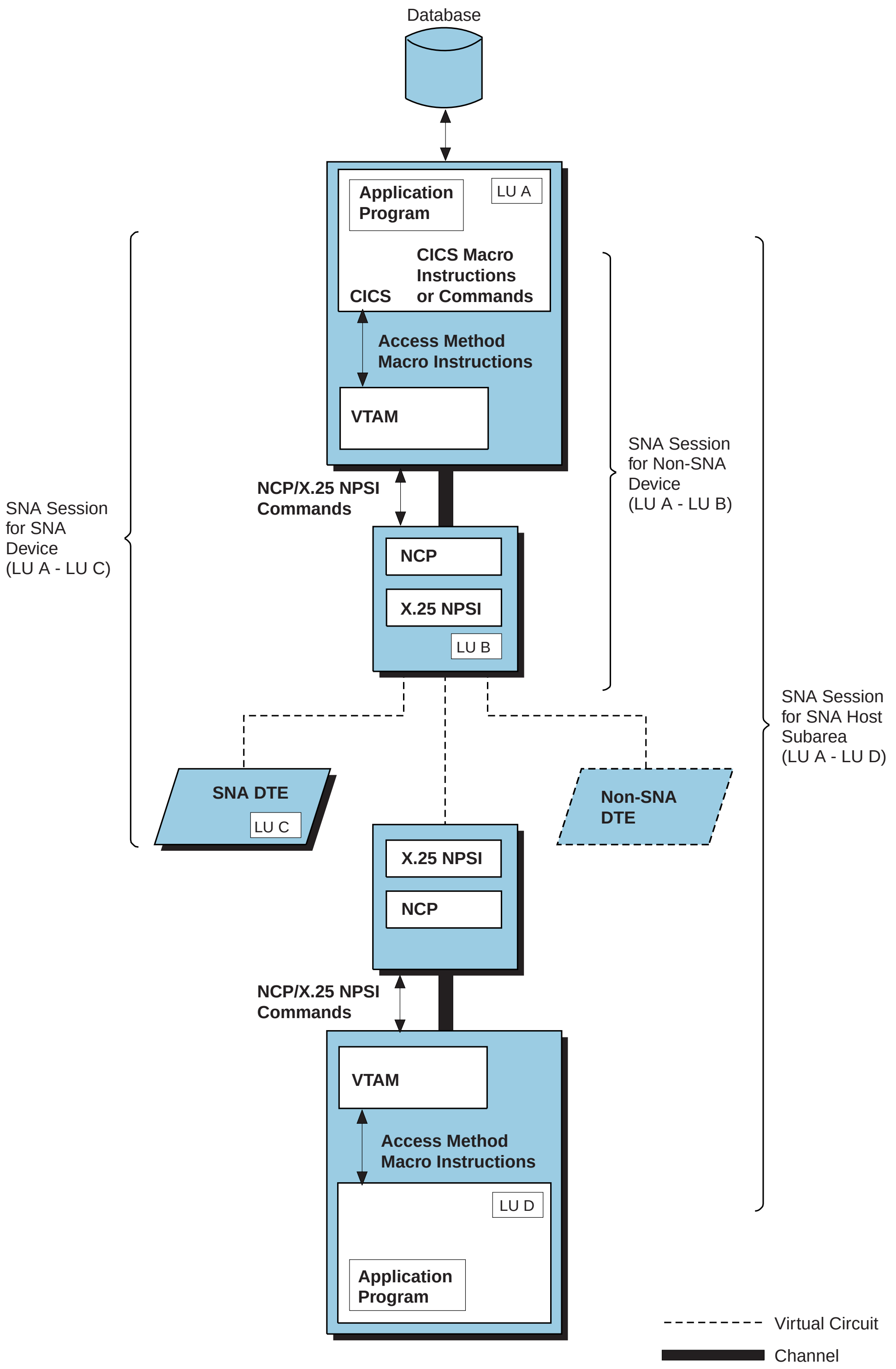

Database
Application Program
LU A
CICS Macro Instructions or Commands
CICS
Access Method Macro Instructions
VTAM
NCP/X.25 NPSI Commands
NCP
X.25 NPSI
LU B
SNA Session for SNA Device (LU A - LU C)
SNA Session for Non-SNA Device (LU A - LU B)
SNA Session for SNA Host Subarea (LU A - LU D)
SNA DTE
LU C
Non-SNA DTE
X.25 NPSI
NCP
NCP/X.25 NPSI Commands
VTAM
Access Method Macro Instructions
LU D
Application Program
Virtual Circuit
Channel

*Figure 5-1. SNA Host Node to SNA Peripheral Node or Non-SNA DTE Using CICS*

## TCT Definition for SNA Resources

NPSI gives the impression of a direct connection (no PSDN in the middle) to a leased SNA device. The host application program is unaware of the session setup, access method, and network communication requirements. Instead, all this knowledge is offloaded onto the applicable software, hardware, and microcode to permit the isolation of these concerns.

The CICS/NPSI pair allows application support of all LUs supported by PU types 1, 2.0, and 2.1 nodes. Thus, this pair can support the following LU session types:

- LU1
- LU2
- LU4
- LU6.1
- LU6.2

This support allows the LU to be connected to CICS as if the PSDN is not there. Because the connection is isolated from the PSDN, it is not necessary to change the SNA resources that communicate through NPSI. Thus, the definition of an SNA resource with NPSI is exactly the same as that without NPSI.

## TCT Definition for Non-SNA Resources

The TCT connection to non-SNA resources appears as an LU type 1 session.

The application programmer must set up a configuration using one of the two interface types supported by the CICS/NPSI interface. The types supported are half-duplex flip-flop mode and half-duplex contention mode.

Half-duplex flip-flop protocol is centered around the change of direction indicator (CD indicator). This request/response header (RH) indicator identifies the partner that can transmit data. The partner holding the CD indicator is the one able to send data. The partner receiving data can send a SIGNAL command if it needs to send data before the CD indicator would normally be relinquished.

Contention protocol is a half-duplex protocol that permits either session partner to initiate data transmissions. Both partners can receive data at any time.

Contention occurs if both session partners want to send data at the same time. In this case, a winner is determined and only the winner’s transmissions continue. The contention winner is predetermined in the session BIND parameters. Thus, the winner of a contention is always known beforehand.

The contention of transmission is done at the *chaining* level. Contention occurs only at the start of a chain of data. The next chain initiates a new contention situation that must be administered individually.

## Generation Requirements

The system programmer defines the LU1 device by coding the following values for the TRMTYPE keyword in the DFHTCT TYPE=TERMINAL macro or the CEDA DEFINE/ALTER TERMINAL transaction in the TYPETERM field.

Figure 5-2 on page 5-4 describes a typical TCT entry for an LU1 device. Options and defaults are noted to the right of some keywords.

```
DFHTCT   TYPE=TERMINAL,
         ACCMETH=VTAM,
         TRMTYPE=                (see list of options)
         NETNAME=xxxxxxxx,
         TRMSTAT=TRANSCEIVE,
         TIOAL=256
         BUFFER=256,
         RUSIZE=                 (default is 256)
         BRACKET=YES,
         PGESTAT=PAGE,
         PAGESIZE=               (default is (12,80))
```

*Figure 5-2. Typical TCT Entry for LU1 Device*

Possible values for the TRMTYPE keyword are shown in Table 5-1.

*Table 5-1. Possible Values for the TRMTYPE keyword*

| **Protocol** | **Acceptable Value** |
|---|---|
| Flip-flop | 3767<br>3767I<br>INTLU |
| Contention | 3767C |

**Note:** Code `TRMTYPE=TWX` if the PAD device requires it at the remote end.

# Application Requirements for Support

Application requirements for support of remote DTEs communicating through NPSI vary depending on whether the remote DTEs are SNA devices or non-SNA devices.

## SNA Resources

The application need not be modified to support these devices. All changes applied to the data are isolated from the application.

## Non-SNA Resources

Non-SNA resources must also be supported by a host application program. Because the host application knows only of the SNA resources, the support is made easier by the CICS/NPSI interface. However, communication needs can vary, so the host application program should be modified to support your network configuration. For example, if you use transparent PAD support, session data is prefixed with an extra byte. You must modify your CICS application programs to accommodate this prefix.

***Integrated PAD Support:*** When a CICS application program communicates through integrated PAD support, the application program operates as if connected to an LU type 1 device.

***Password Protection:*** If the application wants to use the password protection feature of NPSI, it can use the ENP and INP characters to define when a field should not be echoed to the terminal. The hexadecimal values for these control characters are:

| | |
|---|---|
| **X'24'** | Inhibit presentation (INP) |
| **X'14'** | Enable presentation (ENP) |

**Note:** If TRAN=NO is coded on the X25.MCH statement:

- NPSI does not translate EBCDIC to ASCII.
- The control character used for INP is X'12' rather than X'24'.
- The overstrike message is translated to ASCII EVEN code.

If this default value must be changed, see *NPSI Diagnosis, Customization, and Tuning*.

To use the password protection function, the application program places an INP character anywhere in the data stream to prompt the protected information. The protected information is interpreted and converted into the appropriate PAD commands to disable display at the device. Disabling the display can be accomplished through either the inhibiting of the data from being echoed back to the device, if the device is a video display terminal, or the transmission of a blackout character string, if the device is a typewriter-like device.

A user LOGON transaction must be written to use the services of the NPSI password protection.

Upon receiving and processing the response, the next output RU should contain, at any position, the ENP character used to initiate the redisplaying of characters on the device.

To implement the password protection function, you must request that NPSI perform this processing by specifying PWPROT=YES on the X25.MCH statement. NPSI is responsible for all the required processing to implement the password protection function.

***Passing Control to another VTAM Subsystem:*** If a CICS transaction wants to pass the integrated PAD device to another application by using the EXEC CICS ISSUE PASS command, SHUTD=NOINVCLR should be coded in the corresponding X25.MCH statement. This prevents the SHUTD executed by CICS from clearing or resetting the virtual circuit.

NPSI is also responsible for *all* PAD processing for any activity. As soon as the virtual circuit connection is established with the PAD, NPSI sends a PAD message to the PAD to set the values of PAD parameters 1, 7, and 8. See *NPSI Diagnosis, Customization, and Tuning* for more information about the objective of integrated PAD support.

***Transparent PAD Support:*** Communication through transparent PAD support requires that the CICS program be written so that the first byte of the data buffer is reserved for the command code that defines the type of data contained in the buffer. The valid command codes are:

| First Byte | Type of Data |
|---|---|
| X'00' | User data with Q bit set to 0 (unqualified data) |
| X'02' | User data with Q bit set to 1 (qualified data) |
| X'1B' | RESET command |
| X'1F' | RESET CONFIRMATION command |
| X'23' | INTERRUPT command |
| X'27' | INTERRUPT CONFIRMATION command |

Data that is passed between the remote DTE and the application program is sent as data with the Q bit set to 0. This is done by placing X'00' into the first byte of the data buffer sent to the remote DTE. Application data is returned to the application program with this same code in the first byte of the data buffer. The application program must be aware of the extra byte preceding the data processed by the application program.

Qualified data is used to exchange information between the application program and the remote PAD. Qualified data can be used to perform any type of PAD command.

Reset packets contain the cause and diagnostic codes in the second and third bytes, respectively. NPSI resets the send and receive counters to 0.

The second byte of the Interrupt packet contains the interrupt cause (usually set to X'00'). For Interrupt packets, the application program initiates the transmission of an Interrupt Confirmation packet when an Interrupt packet is received.

You can use the TRAN keyword on the X25.MCH statement to request that NPSI translate data when you use PAD support. When you request the TRAN option, NPSI performs translation of all data *after* the first byte of the RU for unqualified data.

Because the first 4 bytes of the input are not consistent, these characters cannot be used for transaction selection. As a result, the TCT entry for the session should specify the TRANSID keyword, or the TRANSID keyword should be used on the CICS RETURN command. This allows input to be processed correctly even though the keywords are preceded by a command code.

***Bracket Protocol:*** A bracket is any complete unit of work that an application program and an LU have been programmed to accomplish. Bracket protocols are used to ensure that one or both ends of a session do not begin processing a new unit of work until the current one is complete.

A CICS transaction is a typical example of a bracket. In a CICS bracket, an LU requests information and CICS responds and ends the bracket. The LU can make further requests depending on the response, and the bracket will continue.

When a session is established, the bits in the session parameters determine who begins the bracket, who ends it, and who wins if contention occurs. One end of the session is assigned as the first speaker, and the other end is the bidder. The bidder normally asks permission to begin a bracket.

***Flip-Flop Protocol:*** Half-duplex flip-flop data flow control protocol operates with the CD indicator. Only the partner that possesses the CD indicator is able to transmit data.

CICS automatically includes the CD indicator with the first data RU that is sent in a session. To change the CD indicator setting, choose one of the following actions:

- Explicitly use the INVITE parameter on the SEND command.

  Use of the INVITE parameter notifies the CICS TCP that the program is requesting that a CD indicator be added to the data when it is sent to the access method.

- Implicitly use a SEND command with a RECEIVE command.

CICS delays the transmission of output data as long as possible so it can piggyback as many indicators onto the data as is feasible. In this case, CICS *does not* actually send data out upon encountering a SEND command. Instead, CICS buffers the data and waits for a command that requires it to transmit the data.

When CICS encounters the RECEIVE command, it transmits the data and adds a CD indicator to the data so that a response is returned. CICS performs the same processing for the CONVERSE command.

Using these methods, the CD indicator is passed between the session partners while data flow is maintained and controlled. Each session partner can control the data flow and manage its own internal resources.

To better manage resources, a session partner can send data to the partner that possesses the CD indicator. This is done by using the ISSUE SIGNAL command. This command indicates to the session partner holding the CD indicator that the partner sending the ISSUE SIGNAL command wants to send data. The partner with the CD indicator can satisfy this request by performing a command that sends the CD indicator to the other partner.

The session partner receiving the SIGNAL command is notified of this request by the execute interface block (EIB) field EIBSIG being set to X'FF' or through a HANDLE CONDITION SIGNAL command. For you to use the EIBSIG method, the application program must be coded to check the EIBSIG field after all terminal control commands (SEND, RECEIVE, CONVERSE, and so forth). Use of the HANDLE CONDITION SIGNAL command is preferable, because the specified routine branches whenever the SIGNAL command is received.

NPSI sends OIC LIC RUs to the host with the CD indicator on. Integrated PAD can exchange SIGNAL commands with the host.

***Contention Mode Protocol:*** In contention mode, contention for the right to send occurs only between *chains* within a bracket. If the session is not in bracket state, the normal rules for bracket initiation apply. NPSI does not have a contention state. It has send, receive, and standby states.

When CICS is in contention mode, it has three states:

- Send
- Receive
- Contention

If a message arrives while a receiver is in contention state, the receiver’s state switches to *receive*. Transmission of the end-of-chain session partners back to contention state.

Contention can occur when both session partners are in send state. The session is always established with CICS as the contention *loser*. SNA specifies the following rules for the resolution of contention:

- If the contention loser is sending, incoming messages must be queued.
- If the contention winner is sending, incoming messages can be either queued or rejected with an appropriate sense code.

For the CICS subsystem, (and its associated application programs), the SNA rules are implemented in the following manner:

- The contention resource sends data to CICS, but the CICS application program has not issued a RECEIVE command.

  In this case, CICS provides the queuing mechanism specified by SNA, if *Read-ahead queuing* is specified for the transaction or the queuing is handled by VTAM. You can specify Read-ahead queuing by coding RAQ=YES in the DFHPCT macro or in the PROFILE command of the CEDA (CICS resource definition on-line transaction) transaction.

  **Note:** If the CEDA transaction is used to create a new transaction profile, then this profile must be specified for all transactions that want to use read-ahead queuing.

  If read-ahead queuing is specified, incoming data is queued by CICS in its *temporary storage queue*. If the transaction subsequently issues a RECEIVE command, the input is recovered from the temporary storage queue rather than from the terminal. If the transaction terminates without issuing a RECEIVE command, CICS automatically purges the queue and the input is lost.

- The CICS application program issues a SEND command, but the contention resource is in *send* state.

  In this case, the contention resource returns a negative response with sense code X'081B'. This sense code specifies that the receiver is in *send* state.

  When the negative response is received, the Node Abnormal Condition Program (DFHZNAC) is scheduled for the task. The following default actions are taken by DFHZNAC:

  1. If the rejected chain is sent with definite response, the VTAM SEND command is purged, and the CICS transaction is abnormally terminated (DFHZNAC action flags X'60E000').

  2. If CICS is not in send state when the response is received, the VTAM SEND command is purged, the transaction is abnormally terminated, and a VTAM CLSDST macro is issued to break the connection (DFHZNAC action flags X'60E001').

  3. If exception response was requested, the VTAM SEND is purged, and the CICS transaction is abnormally terminated (DFHZNAC action flags X'60E000').

  For the first case, the CICS user can write a node error program (NEP) to retry the failing SEND command. The NEP must do the following:

  – Reset the abend flag in TWAOPTL.

  – Reset the VTAM SEND purge flag in TWAOPTL.

  – Set flag TWANPFW in the NEP return code byte TWANEPR. This flag specifies *retry with FORCE* and results in CICS sending the SIGNAL command to NPSI and then retrying the SEND command.

  For the second case, an NEP is not allowed to reset the *break connection* flag, so recovery is not possible.

  For the third case, care should be taken because the terminal input/output area (TIOA) might not be available.

## Dial-In to CICS

An application program does not require any modification to support a dial-in resource. To the host application, the dial-in resource appears as a resource that has attached itself to the host.

Both physical and user security are cause for concern. Physical security *can* be implemented by the access method; however, additional security assurances, such as passwords or security packages, might be required.

## Dial-Out from CICS

CICS dial-out can be performed in three ways:

- The first method uses the START command for a transaction on the terminal to which you want to dial out. This command performs an interval control process that initiates a new task to a specific terminal. When CICS recognizes that the terminal is not attached to the region, it attempts to acquire the terminal. The acquisition is passed to VTAM, and the SNA CONNOUT request is passed to the appropriate communication controller.

  In this case, the transaction program must establish a method of synchronization with the remote device. One method is to define a device that will be the first one to send data. A session partner would not send any data until requested to initiate data transfer. This simple protocol eliminates problems incurred in communication; this is especially true of non-SNA connections, because the flow control on these sessions is difficult to regulate. Once the data flow is synchronized between the session partners, communication can proceed.

  NPSI is invoked, because the resource is identified as belonging to NPSI. An X.25 Call Request packet is transmitted to the remote DTE. Additional commands are passed between the network resources until the session is established (by a BIND being passed between the session partners) and SDT is complete. At this point, the device is attached to CICS, and the transaction that is already started is attached to the device. Communication can now be performed with the dial-out device.

- The second method of initiating a dial-out operation is for an application program (or an operator command) to request the acquisition of a device. For an application program request, use the DFHTC TYPE=CONTROL macro or the EXEC CICS SET TERMINAL command. (The latter is available only in CICS Version 1.7 and later.) An operator can request acquisition of the device through the CEMT transaction.

- The third method of initiating a dial-out operation is for the network operator to issue one of the following commands:

  – V NET,ID=*termid*,LOGON=*appl*
  – V NET,ACT,ID=*termid* with LOGAPPL=*appl* coded on the LU statement

  In both commands, *appl* specifies the CICS subsystem application ID name, and *termid* specifies the terminal to which you want to dial out. When either of these commands is issued, a session is established between the device and CICS.

## Disconnection of SVCs

An application program can disconnect an SVC when communication is complete. This disconnection reduces communication costs by reducing the connection time.

Termination of the virtual circuit is caused by an application program issuing an ISSUE DISCONNECT command to the SLU, which results in an UNBIND command flowing to the remote LU. When this command is processed, the SVC is cleared if DISCNT=YES is coded in the PU of the SMN, and this is the last or only LU to terminate its sessions.

When using CICS with the ISSUE PASS command, CICS causes a CLSDST OPTCD=PASS macroinstruction to be issued. In this case, to prevent a device from being disconnected when using integrated PAD support, specify SHUTD=NOINVCLR on the X25.MCH statement. The NOINVCLR parameter allows the device to be passed between applications. For example, a printer can be acquired by another application after it is passed from CICS without clearing the virtual circuit.

To use the ISSUE PASS command, you must code AUTH=PASS on the VTAM APPL statement for CICS.

## Considerations when Creating an FC CTCP under CICS

You should understand the FC CTCP interface and have a thorough understanding of CICS before attempting to use this application program interface (API). The following considerations should also be understood before using the API:

- The TRANSID must be used on the TCT definition.

  The TCT entries that are in session with the MCH LU and VC LUs should have a TRANSID specified. This results in CICS initiating the specified transaction *whenever* data arriving at CICS contains a Diagnostic packet in the case of the MCH LU, or contains Call, Clear, Reset, or data packets in the case of the VC LUs.

- The FC CTCP created within a CICS application must be able to process *any* of the commands sent by NPSI.

  This includes commands that application programs are not usually able to recognize and process, such as the CLEAR, RESET, and CALL commands. All data from the NPSI LU is routed to the task defined in the TRANSID command located in the appropriate TCT entry. It is the duty of this task to analyze the data packets received from NPSI, and to act accordingly. This task can be programmed to handle X.25 commands, such as:

  – CLEAR
  – RESET
  – CALL

  For example, this task can be programmed to analyze the cause and diagnostic fields received in a Clear packet, and to reestablish the X.25 connection for certain causes.

- The VTAM node error program (DFHZNEP) must be customized.

  DFHZNEP is invoked by CICS whenever the operational status of a device controlled by VTAM changes. The CICS abnormal condition program (DFHZNAC) determines the necessary actions in response to the status change (such as session established, terminated, or device failure). DFHZNEP is able to modify some of the action settings determined by DFHZNAC. Upon returning from DFHZNEP, DFHZNAC performs the set actions. These actions can either be the default settings or those modified by DFHZNEP.

In addition, DFHZNEP can link up to other user-written programs. These user-written programs can determine the required actions for LUs attached to other subsystems, or even the same CICS system.

This process is invoked when the session to the MCH LU is disconnected. Although this is not a normal occurrence, the CICS CTCP must be able to handle this situation if it occurs.

When the session to the MCH LU is disconnected, DFHZNEP is invoked by CICS as a result of the session loss with the MCH LU. The CTCP should determine the cause of the failure by analyzing the cause code, set by CICS within the communications area, that is passed from DFHZNAC to DFHZNEP.

**Note:** These cause codes are not the same as the X.25 CAUSE codes that are contained in many of the packet formats.

## Programming IMS

Through NPSI, both SNA and non-SNA terminals can be connected to the Information Management System (IMS) subsystem. The system programmer defines each device communicating through NPSI with the COMM, TYPE, and TERMINAL macros.

Figure 5-3 on page 5-12 shows the data flow in a network configuration using IMS and NPSI.

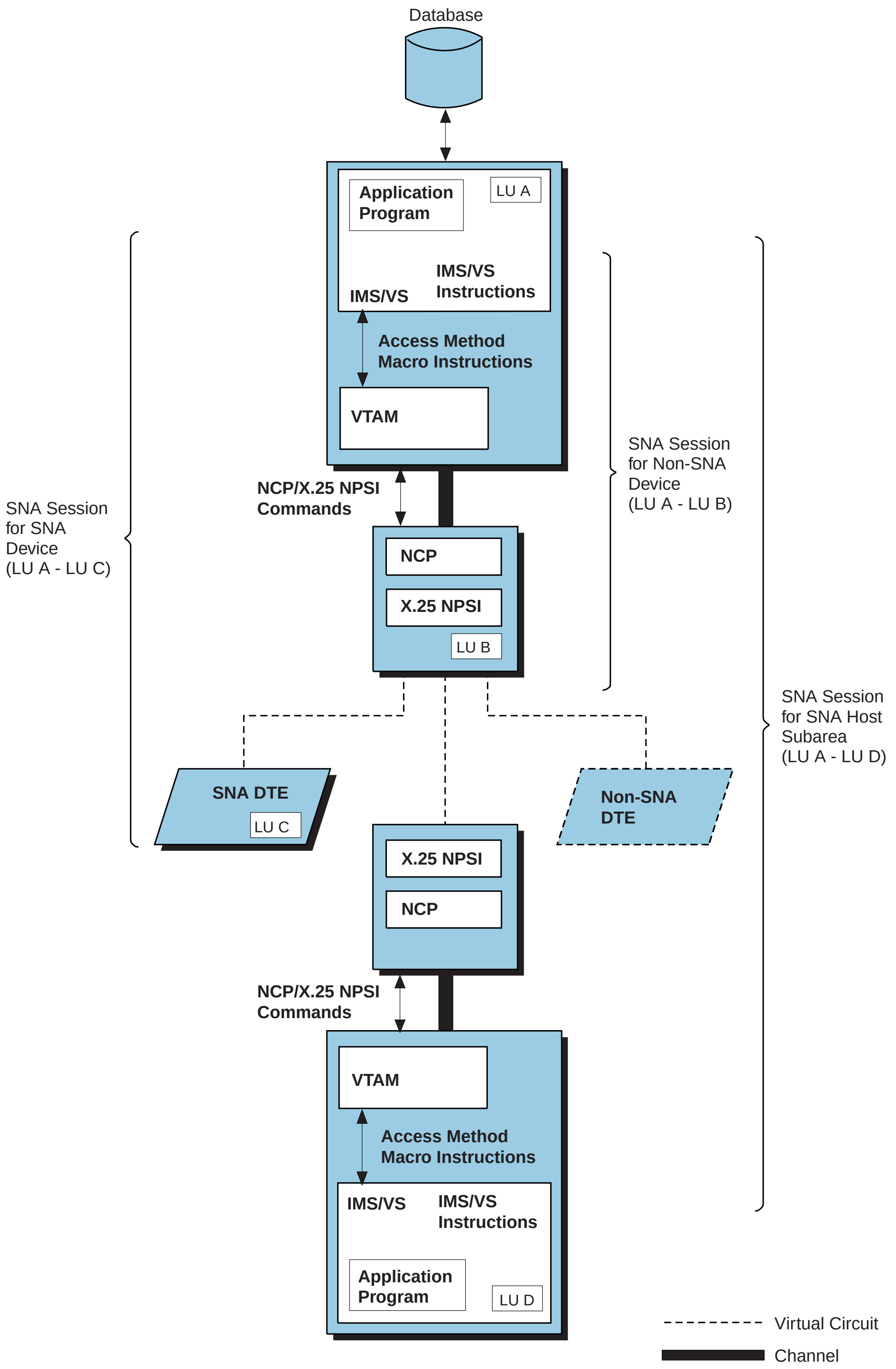


*Figure 5-3. SNA Host Node to SNA Peripheral Node or Non-SNA DTE Using IMS*

## SNA Connection

SNA devices are defined to IMS and communicate with IMS without regard for the connection method. IMS is unaware that NPSI is in the communication path. No modification of IMS or IMS application programs is required to allow communication through NPSI to a remote DTE.

## Non-SNA Connection

Non-SNA devices interface with IMS through the NPSI LUSIM. This function performs a conversion on data coming from the non-SNA device to give the appearance, to the IMS application program, that the data comes from an LU type 1 device.

The LUSIM does *not* map outbound chaining of PIUs to any X.25 protocol. Because there is no mapping, the size of the maximum RU must be set as large as the largest RU that the IMS system needs to transmit.

**Note:** The LUSIM optionally maps chains of PIUs to the X.25 complete packet sequence M-bit sequence (CPS/MBS) when MBITCHN=YES is specified. If this option is not chosen, the size of the maximum RU (coded in the BIND image) must be set as large as the largest RU that the IMS system needs to transmit. When using GATE or Transparent PAD with MBITCHN=YES, only OIC PIUs will be generated and sent to the host.

This RU size must be set in two places. The first place is in the COMM macro. The maximum RU size on the RECANY keyword of the COMM macro must be set large enough so that no RUs are truncated. Secondly, the OUTBUF keyword of the TERMINAL macro must be set. This keyword is used by IMS to set the outbound RU size in the BIND.

These RU values should be optimized to reduce performance impacts. To conserve storage, set the maximum RU size as small as possible.

All devices are defined with the TYPE and TERMINAL macros as SNA devices. The TYPE macro defines some global values used in all subsequent TERMINAL macros.

LU type 1 devices are supported in communication through NPSI. In addition, IMS fast-path devices are also supported. You can find a more extensive list of supported devices in the *IMS General Information* manual and the *IMS Installation Guide* associated with the platform on which you are working.

Figure 5-4 shows an example of a terminal definition for a 3767.

```
COMM     RECANY=(number of buffers,maximum RU size)
TYPE     UNITYPE=3767,
         OPTIONS=NOASR
TERMINAL NAME=nnnnnnnn,
         COMPT=CONSOLE,
         FEAT=IGNORE,
         OPTIONS=(BSELM,...),
         OUTBUF=maximum RU size
```

*Figure 5-4. Terminal Definition for a 3767*

## IMS Application Support for DFC

IMS is responsible for all data flow control (DFC) requirements from the application program. The application program under IMS is not aware of the data flow control requirements. The application program is concerned only with the data; all higher-level protocol requirements are isolated from the application program by IMS.

## Integrated PAD Support

When an IMS application program communicates through integrated PAD support, the application program operates as if connected to an LU type 1 device.

If the application program wants to use the password protection feature of NPSI, the application program can use the ENP and the INP characters to define when a field should not be echoed to the terminal. The hexadecimal values for these control characters are:

| | |
|---|---|
| **X'24'** | INP |
| **X'14'** | ENP |

**Note:** If NPSI does not translate EBCDIC to ASCII, the control character used for inhibit presentation is X'12' rather than X'24', and the overstrike message is translated to ASCII EVEN code. If this default value must be changed, see the Customization chapter of *NPSI Diagnosis, Customization, and Tuning*.

To use the password protection function, the application places an INP character at the end of the data stream to prompt the protected information. NPSI interprets this character and converts it into the appropriate PAD commands to disable the display at the device. Disabling the display can be accomplished through either the inhibition of echoing data back to the device, if the device is a video display terminal, or the transmission of a blackout character string, if the device is a typewriter-like device.

Upon receiving and processing the response, IMS starts the next output buffer with the ENP character to initiate the redisplaying of characters on the device.

NPSI is responsible for all processing required to implement the password protection function. You must request that NPSI perform this processing by specifying PWPROT=YES on the X25.MCH statement.

NPSI is also responsible for *all* PAD processing for any activity. As soon as the virtual circuit connection is established with the PAD, NPSI sends a PAD message to the PAD to set the values of PAD parameters 1, 7, and 8. See *NPSI Diagnosis, Customization, and Tuning* for more information on the objective of integrated PAD support.

**V3R5 and Later:** For the Integrated PAD function, NPSI V3R5 and later releases makes sending the READ PAD PARAMETER command optional at call setup time. In some instances, when the READ PAD PARAMETER command is sent, some data may be lost from the DTE. You can avoid this by coding RPADPARM=NO on the X25.MCH statement.

For more information about coding the RPADPARM keyword, see *NPSI Planning and Installation*.

## Transparent PAD Support

Communication through transparent PAD support requires that the IMS program reserve the first byte of the data buffer for the command code that defines the type of data to be contained in the buffer.

The valid command codes are:

| First Byte | Type of Data |
|---|---|
| X'00' | User data with Q bit set to 0 (unqualified data) |
| X'02' | User data with Q bit set to 1 (qualified data) |
| X'1B' | RESET command |
| X'1F' | RESET CONFIRMATION command |
| X'23' | INTERRUPT command |
| X'27' | INTERRUPT CONFIRMATION command |

Qualified data is used to exchange information between the application program and the remote PAD. Qualified data can also be used to perform any type of PAD command.

Data that is passed between the remote DTE and the application program is sent as data with the Q bit set to 0. This is done by placing X'00' into the first byte of the data buffer being sent to the remote DTE. Application data is returned to the application program with the same code in the first byte of the data buffer. Thus, the application must be aware of the extra byte preceding the data processed by the application program.

Reset packets contain the cause and diagnostic codes in the second and third bytes, respectively. NPSI resets the send and receive counters to 0.

The second byte of the Interrupt packet contains the interrupt cause (usually set to X'00').

You can use the TRAN keyword on the X25.MCH statement to request that NPSI translate data when you are using PAD support. When you request the TRAN option, NPSI performs translation of all data *after* the first byte of the RU for unqualified data.

**Note:** If you are using transparent PAD, see the command formats for the transparent PAD RUs on page 2-17.

## Programming TSO

This section contains considerations specific to the Time Sharing Option (TSO) subsystem when running NPSI. Figure 5-5 shows the data flow in a network configuration using TSO, and NPSI.

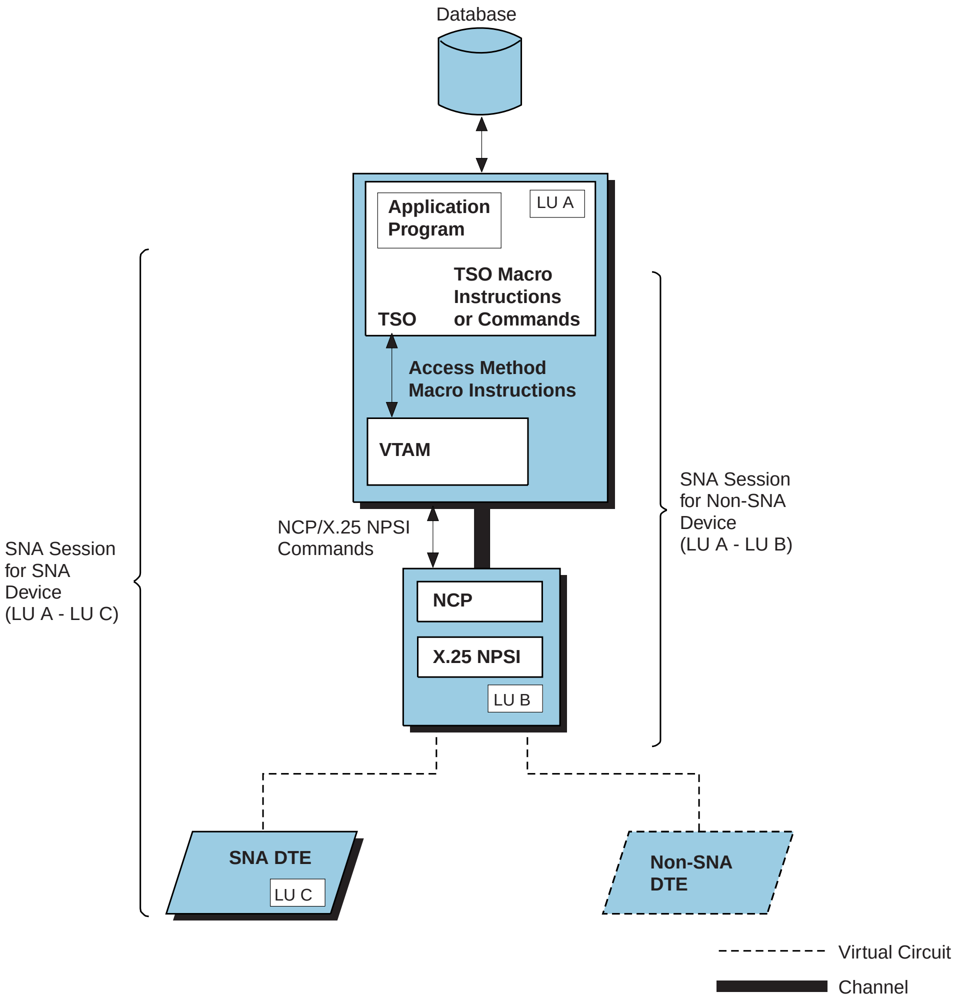

Database
Application Program
LU A
TSO Macro Instructions or Commands
TSO
Access Method Macro Instructions
VTAM
NCP/X.25 NPSI Commands
NCP
X.25 NPSI
LU B
SNA Session for SNA Device (LU A - LU C)
SNA Session for Non-SNA Device (LU A - LU B)
SNA DTE
LU C
Non-SNA DTE
Virtual Circuit
Channel

*Figure 5-5. SNA Host Node to SNA Peripheral Node or Non-SNA DTE Using TSO*

## LU Definition

The LUSIM code in NPSI simulates an LU by a method similar to a Network Terminal Option (NTO) device for each start-stop mode DTE accessing the SNA host through NPSI. These devices appear to the host access method as LU type 1 SNA devices and support the same BIND parameters as the SDLC 3767 Communications Terminal. Therefore, you must define the appropriate profile for an LU type 1 device.

For start-stop mode DTEs running over an SVC using TSO, you should define the LU with the TERM=TWX keyword in the VTAM SMN definition as shown in Figure 5-6 on page 5-17.

```
CA001    PU  ADDR=01,
             PACING=1,
             VPACING=2,
             PUTYPE=1,
             IDBLK=003,
             IDNUM=0F010,
             MAXDATA=261,
             DISCNT=YES
CX001    LU  LOCADDR=0,
             SSCPFM=USSNTO,
             MODTAB=MODRAL,
             DLOGMOD=SSCICSF,
             TERM=TWX,
             USSTAB=USSTAB
```

*Figure 5-6. Non-SNA Entry for NPSI for SMN*

The same parameters can be coded in the X25.PU and X25.LU statements, or in the X25.VC statement at NPSI generation for PVCs. See *NPSI Planning and Installation* for more information about coding these statements.

## Line Control

The use of TSO with an LU type 1 device allows only line mode transactions. Problems controlling carriage return and line feed when simulating LU type 1 devices are quite common. For several start-stop mode DTEs, an End of Text (EOT) character is normally sent as the delimiting character rather than of a carriage return and line feed. To alleviate this problem, use TERM=TWX in the SMN for the LU. The TERM=TWX specification tells TSO that the remote DTE is an NTO device.

Although NTO is not used within NPSI, the start-stop devices are actually simulated as NTO devices within the communication controller. This means VTAM expects to find a carriage return and line feed at the end of every input line. If either or both the carriage return and line feed are missing, VTAM inserts the missing character. The character VTAM inserts can cause a formatting problem. To avoid this problem, you should change the EOT character to a line feed using translation tables.

For more information about TSO LU definition and line control, see *VTAM Installation and Resource Definition*.

## Password Protection

When using the TSO subsystem with LU type 1 devices, you must code the DCODE parameter on the MODEENT keyword of the mode table used for that session. It must be coded as:

`MODEENT DCODE=`*devtype*

See *VTAM Customization* for valid values for *devtype*. The value to code for password protection is `X'80'`. You must code this value whether the terminal at the remote end is typewriter or display type. Coding the DCODE keyword permits TSO to request the USERID first, and then the PASSWORD. The user’s response to the PASSWORD prompt is suppressed. During normal processing, the data entered by the user is returned to the terminal to be displayed. This process, often referred to as *echoing*, is a technique of visually assuring that the data was received correctly. In the special case of a PASSWORD, displaying the data that you entered is a security exposure.

NPSI uses the ENP and the INP characters to define when a field is not to be echoed to the terminal.

TSO places an INP character at the end of the data stream when prompting for the password. NPSI scans the data for the INP character. If the character is present, NPSI ensures that, as the password is keyed, it cannot be read.

For terminals that are using echoing of input data, NPSI performs password protection by setting PAD parameter 2 to a value 0 before entry of the password and then returns PAD parameter 2 to a value of 1 after the password is entered. While PAD parameter 2 has the value 0, no input data is echoed.

For terminals that are not using echoing of input data, NPSI performs password protection by creating an overstrike sequence of 8 positions. Note that the echoing technique is also quite suitable for typewriter type terminals.

Upon receiving and processing the response, TSO starts the next output buffer with the ENP character. This causes the PAD parameter 2 with a value of 1 to be sent for terminals that are using echoing of input data.

## Customization

When accessing TSO, you often need to customize the X.3 PAD parameters to suit your installation needs. For example, you may want to ensure that you can perform error correction in the PAD buffer by setting PAD parameter 16 to 8. This is an alternative to having erroneous data passed over the X.3 PSDN only to be corrected within TSO. To ensure that the PAD parameters are set to your desired values:

1. Select a profile from the supplier of the PAD service with the desired values.
2. Modify the profile immediately after the connection to the PAD service to set the desired values. (This can be automated if you are using a physical circuit.)
3. Customize the NPSI integrated PAD support code so that the desired values are set when the SNA session is started using X.29 PAD messages.

For additional information about PAD parameter customization, including PAD parameter settings, see *NPSI Diagnosis, Customization, and Tuning*.

## Programming NetView

NetView is a licensed program that permits automated operation management of a communication network. This program provides a command facility that contains message automation, hardware monitor, session monitor, and status monitor. The following sections describe how to use the NetView program with NPSI attached terminals, and explain the functions of the NetView program that work on NPSI attached resources.

## Using NetView with NPSI Attached Terminals

The NetView program can be used from terminals connected through LLC type 2 or type 3. The NetView program cannot communicate through LLC types 0, 4, or 5 directly. It can communicate through these LLC types through a relay program, such as the IBM-provided General Teleprocessing Monitor for Open Systems Interconnection (GTMOSI), the Communication Subsystem for Interconnection (CSFI), and the OSI Communication Subsystem. The GTMOSI program maps non-SNA devices (such as a 3101 or a Minitel) into a 3278 to allow connection to the NetView program.

The NetView program also participates fully in the SVCSC environment, and is able to connect with another NetView system through an X.25 interface. This gives you great flexibility in creating a network topology.

Figure 5-7 on page 5-20 shows the data flow for a network configuration that uses the NetView program and VTAM.

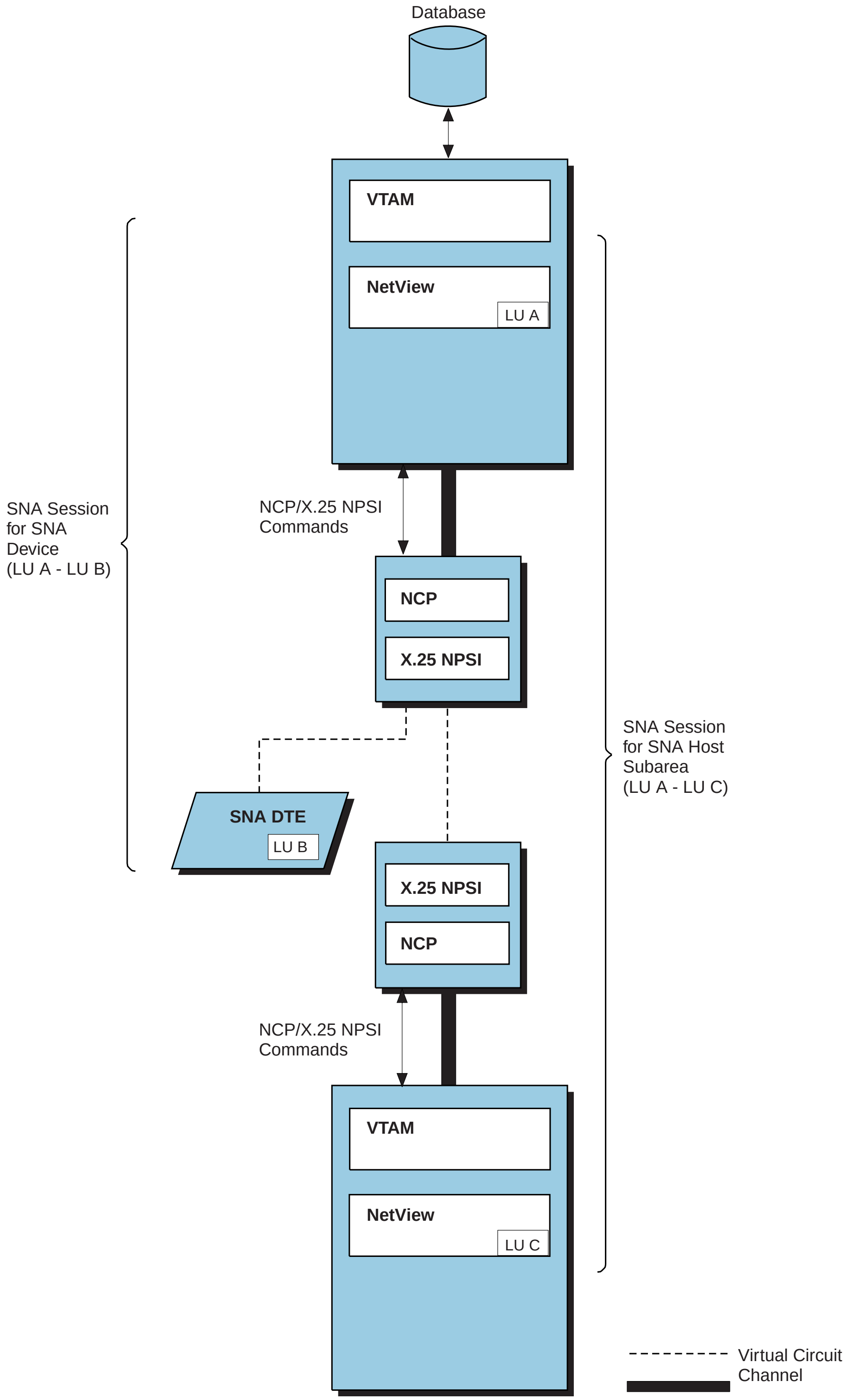


*Figure 5-7. SNA Host Node to SNA DTE or Non-SNA DTE Using NetView and VTAM*

## NetView Functions Usable with NPSI Resources

The following sections describe the high-level functions and list the additional network-management capabilities of the NetView program. The relationship of the NetView program to SNA and non-SNA connections is also described. For more information, see *Planning for NetView, NCP, and VTAM*.

### Command Facility

The command facility provides the environment to enter commands to all components of the network. These components include the host access method, application subsystems, the operating system, and the large and departmental processors comprising the network hosts.

Messages from all these network components can enter the command facility and commands can be automatically sent to the originating component, or any required destination.

To illustrate how these commands are automatically sent with NPSI, use command lists or REXX EXECs to retry the activation of an MCH PU when the PSDN is temporarily out of order. The VTAM message showing that the PU is inactive can be used to initiate an activate command after a period of time, without the need for any operator intervention. If the PSDN is still out of order after the wait, the MCH PU can be reactivated by the command facility.

### Hardware Monitor

The hardware monitor enables you to access problem information generated at resources that are either link-attached or channel-attached to the host system. The information passed to the host consists of the following:

| | |
|---|---|
| **Statistics** | Records of traffic and recoverable errors |
| **Events** | Unusual occurrences detected at a device or program |

The hardware monitor supports NPSI by managing and presenting the RECFMS request units, generated by NPSI, which contain event and statistical data.

**Note:** The use of billing units as statistical data affects the NPDA-51F and NPDA-53F panels in NetView. In these panels, both the TRANSMISSION TOTAL and RECEIVE TOTAL can reflect billing units and not packet totals.

### Session Monitor

The session monitor is used to collect and correlate data about sessions and routes, and to obtain online access to the collected data. It allows you to examine information related to the SNA network and to identify network problems.

Session monitor can be used to measure response time, when using LLC 2 or 3, if the remote DTE has the response time monitor (RTM) facility.

### Status Monitor

The status monitor displays network status and accepts network operator commands. Status information that can be displayed includes the following:

- Status summaries
- Details for a domain
- Status of a single node
- Node descriptions

## SNA Connection

For SNA connections, those connections that flow through virtual circuits using LLC types 2 and 3, NPSI provides path control functions only. Neither the SNA device nor the host destination is aware that the link uses X.25 protocols. Network management, functioning principally through the PU, is also unaffected by the presence of the PSDN. Management is not affected when using either connections to peripheral nodes or subarea nodes, whether through PVCs or SVCs.

### Support for Network Management

Network management using the NetView program is supported across the X.25 connection. The existence of an X.25 interface is transparent to the SNA requests and responses that are used for network management.

## Non-SNA Connection

For non-SNA connections, that is LLC types 0, 4, and 5, NPSI simulates PUs and LUs within the communication controller. Thus, the SNA-oriented network management functions, which depend on the SNA resources, do not extend past the communication controller. The event and statistical data generated by NPSI are the only network management events provided by the NetView program for these types of virtual circuits. However, because the event data provided by NPSI includes diagnostic information created by the PSDN and the remote non-SNA DTE in control packets, as much network management data is presented through the NetView program as possible.

## NPM

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

NPM supports accounting and performance data collection for all SNA LUs (MCH LU and VC LU). As a result, NPSI does not have to be extended to support LU counter management. However, NPSI V3R4 and later releases have been extended to support data collection on the MCH LINE and PU, and on the VC PU.

You must modify your system generation to allow NPSI and NPM to perform the following functions:

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

### Accounting

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

### Performance

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

See *NPSI Diagnosis, Customization, and Tuning Guide* for more information about the performance data that is supported and how to use the data to tune your network.

See *NPSI Planning and Installation* for information about the installation and generation procedure used to invoke this function.

# Chapter 6. NPSI Dial-In Enhancements

# Chapter 6. NPSI Dial-In Enhancements

This chapter describes the definitions and requirements needed to support NPSI dial-in enhancements.

## Summary of NPSI Dial-In Enhancements

**V3R5 and Later:** The dial-in process has been enhanced by two different functions offered by VTAM:

1. The Identification Verification function that uses the VERIFY and VERID keywords of the PATH statement in the VTAM SMN.
   - In addition to the IDBLK and IDNUM verification, this function also checks the calling DTE address.
   - This function only applies to the X.25 DTEs that provide an SNA identification and to those uniquely identified by an existing entry in the VTAM SMN:
     - SNA DTEs (QLLC or PSH)
     - X.25 non-SNA DTEs that provide this identification through the CUD field (PCNE when it is not selected by the subaddressing mechanism)
   - When it finds a match between the Node_ID of the XID and the IDBLK and IDNUM keywords of the PU entries on the SMN, VTAM performs another scan to check whether VERIFY=IN or INOUT and VERID are coded on the PU PATH statement in the SMN. This scan compares the data from an additional REQCONT control vector (CV `X'25'`) with the value of the VERID keyword. The connection request is accepted only if there is a match. For NPSI, the CV `X'25'` carries the calling DTE address from the Incoming Call packet.
2. The configuration services installation exit and the MODEL major node that supports the dynamic definition of previously undefined dial-in devices.
   - When VTAM cannot find a match between the node ID of the XID and the IDBLK and IDNUM keywords of the PU entries on the SMN, it invokes the configuration services installation user exit Routine. This user exit receives both the XID and the additional CV `X'25'` of the received REQCONT PIU. It checks the given X.25 calling DTE address and eventually other given data (LLC type, CUD).
   - Based on these verifications, the User Exit requests VTAM to either reject the connection, or to accept it on a PU entry dynamically created in the SMN. This SMN PU entry is built from information provided by the exit on registered skeletons, and exists only for the duration of the call (each incoming call for a specific DTE).
   - Additional details about how this function processes in VTAM and in the user exit are provided later in this chapter.

In the X.25 environment, both functions require NPSI to provide both NCP physical services and VTAM with a security control vector (X'25') appended to the XID.

**Note:** The installation user exit does not allow a single SMN PU entry to be used for both outgoing and incoming calls between a specific non-identified X.25 DTE. For example, with an incoming call, VTAM does not allow the User Exit to return the name of an existing SMN PU entry, such as the one used for outgoing calls.

## Peripheral Node Dial-In Support Process

For NPSI, the peripheral node dial-in support process is:

1. On incoming calls, NPSI provides NCP physical services with an XID that is:
   - The actual XID received from SNA DTEs
   - An XID built with the DTE identification, if any, received in the CUD field of the Incoming Call packet (PCNE)
   - An XID built with an identification (IDBLK and IDNUM) attached to the virtual circuit line selected by NPSI to connect this incoming call

   **Notes:**

   a. This identifier, previously called *default ID*, is mainly used with GATE and PAD connections.

   b. In the first case, the XID built can be any of the architectural XID formats (0, 1 or 3). In the last two cases, it is always an XID format 0 (6 bytes).
2. NCP physical services builds a request contact (REQCONT) PIU that includes the XID and sends it to the VTAM/SSCP owning the virtual circuit line, which NPSI connects to the incoming call.
3. When VTAM/SSCP receives the REQCONT PIU, it scans the VTAM SMN to find a match on the IDBLK and IDNUM fields between the received XID in the REQCONT, and the PU entries in the SMN.

   **Note:** If VTAM has more than one active SMN, they can all be scanned for the match.
4. Depending on the outcome of the scan, the connection is either accepted or denied. If it is accepted, the SETCV_PU and CONTACT PIUs are sent to NCP. If it is denied, the ABCONN PIU is sent to NCP.

This process has the following limitations:

- No user control access based on the X.25 calling DTE address is authorized.

  **Note:** The calling DTE address is more valuable for security checking purposes than an identification coming from the end-user DTE.
- For the DTEs that provide their identification, these identifications must be coded in the VTAM SMN before the call is submitted.
- For the DTEs that do not provide an SNA type identification (IDBLK and IDNUM), unique name identification is not possible.

For this reason, the statistics accumulated by NetView (alerts and PDStats) cannot be used since they identify an NPSI virtual circuit line through which successive connections with several DTEs have been established.

## Configuration Services Installation Exit and the MODEL Major Node

| The Configuration Services Installation User Exit was a new function of VTAM
| V3R4. In addition to the existing SMN, it created:

1. A model major node (MMN) that defines models of PU and LU statements used to build the switched PUs and their dependent LUs dynamically.
2. A dynamic switched major node (DSMN) where the switched PUs and their dependent LUs are dynamically created when requested by the configuration services installation user exit.

If VTAM does not find a match between the received Node ID in the XID and the IDBLK and IDNUM values of any existing entries in the SMN, the Configuration Services Installation User Exit is provided with the XID and CV X'25' data from the received REQCONT PIU. The exit can:

- Perform various user processing, such as:
  - Control access by verifying the X.25 calling DTE address and the CUD provided by NPSI in the REQCONT CV X'25'
  - Connection time accounting
- Refuse the connection by requesting that VTAM abandon the connection.
- Accept the connection, then provide VTAM with the names of the correct PU and LU models and those of the dynamically created devices.

  **Note:** By using the LOGAPPL keyword in the LU models, the exit can cause a session to be established with a predefined application. This application can then request additional security information from the remote DTE user, such as a password.

## Programming the Configuration Services Installation User Exit

The Configuration Services Installation User Exit uses key data that consists of:

- The received REQCONT PIU provided by VTAM as an input parameter
- A user defined table to drive and perform the X.25 DTE Control Access

### Installation Exit General Information

**Note:** The following information is given as an example only. See VTAM documentation for actual user exit interface information.

- The name of the routine must be ISTEXCCS.
- You are responsible for link editing the exit routine in the appropriate library. The exit routine that performs I/O must be link edited with RMODE=24.
- Unconditional GETMAINs should not be coded.
- The exit routine can be written as serially reusable or reentrant.
- The exit runs under a VTAM subtask, allowing it to perform I/O and other processing without affecting the VTAM main task.
- The exit routine operates enabled in pageable storage and gets control in supervisor state with a VTAM storage key.

- Errors in the exit routine can damage VTAM or system control blocks and modules.
- VTAM macros must not be used in this routine.

## Parameter List

The address of the following parameter list is passed to the user-written exit routine (ISTEXCCS) in register 1:

| Hex Offset | EXIT Parameter List |
|---|---|
| **0** | Pointer to length field:<br>This field points to a value that is the length of the entire vector, including the header portion of 3 bytes. The field length is 2 bytes. |
| **4** | Pointer to Type field:<br>This field points to a 1 byte value that denotes the type of vector being passed. |
| **8** | Pointer to Vector field:<br>The field points to one of the vectors. |
| **C** | Pointer to Exit work area field:<br>This field points to a 4-byte field used by the Installation Exit to store the address of the storage allocated to it. |

In addition:

- The address of the vector must be returned in register 1 when the exit returns control to VTAM.
- A nonzero return code in register 15 indicates a fatal error and causes the VTAM subtask to terminate.

  **Note:** The following figures are given as examples to show the type of information carried by the various vectors. To get the accurate vector formats, refer to the appropriate VTAM documentation.

Table 6-1 shows the different vector headers and their offset values used for connections through NPSI.

*Table 6-1. Vector Header and Types*

| Offset | Vector Header |
|---|---|
| 0 | Length of vector including this field |
| 2 | Vector Type:<br>X'00' Begin<br>X'01' XID<br>X'02' Build<br>X'03' Connection status<br>X'04' Failure<br>X'FF' End |
| 3-*n* | Vector: See following XID figures. |

***Begin Vector (*** X'00'***):*** VTAM calls the exit routine with the begin vector during initialization. The exit then performs its initial setup and returns the begin vector to VTAM. VTAM uses the begin vector to determine which specific tasks the exit is to perform. The exit must return the address of the Configuration Services Exit Routine parameter list in Register 1. Table 6-2 shows the begin vector format.

*Table 6-2. Begin Vector Format*

| Dec (Hex) Offset | Size (Bytes) | Bit setting | Description |
|---|---|---|---|
| 0 (0) | 1 | B'1... ....' | This dynamic build bit indicates whether the exit is to participate in dynamic builds of switched resources. If this bit is on when the vector is returned from the exit, VTAM processes a REQCONT for a switched device that is not defined. To process the REQCONT, VTAM calls the exit with the XID vector. If this bit is not on, VTAM denies the REQCONT. |
| | | B'..1. ....' | This call-if-PU-defined bit tells VTAM to call the exit routine when VTAM receives an XID for a PU that is already defined to VTAM. |
| | | B'...x xxxx' | The remaining bits are reserved. |

**Note:** The vectors shown in Table 6-2 and Table 6-3 are shown only to give an example of the type of information they carry. For accurate vector formats and contents, please refer to *VTAM Programming*.

***XID Vector (X'01'):*** If you code the exit routine to support dynamic builds, VTAM calls it with an XID vector whenever an unknown PU attempts to dial in. Table 6-6 on page 6-7 shows resource entry blocks returned by the exit. Table 6-5 on page 6-6 shows a build vector the exit constructs and returns from the information provided in the XID vector. The exit also returns a return code in Register 15. Table 6-3 shows the format for the XID vector.

*Table 6-3. XID Vector Format*

| Dec (Hex) Offset | Size (Bytes) | Bit setting | Description |
|---|---|---|---|
| 0 (0) | 1 | B'1... ....'<br>B'.xxx xxxx' | This bit tells VTAM whether the REQCONT VTAM received is for a PU that VTAM will take over.<br>The remaining bits are reserved. |
| 1 (1) | 1 | | This field is reserved. |
| 2 (2) | 8 | | This field contains the name of the switched line over which VTAM received the REQCONT. |
| 10 (A) | 1 | | This is byte 0 of the XID, including control vectors. See "XID Vector Data" for additional information. |

***XID Vector Data:*** The XID data format shown in Table 6-4 is only for a Format 0 XID because it is the format built by NPSI for a non-SNA DTE. The XID is followed by the control vector X'25', which is not included in the XID length since it is not part of the XID.

*Table 6-4. XID Data and CV25*

| Dec (Hex) Offset | Size (Bytes) | Bit setting | Description |
|---|---|---|---|
| 0 (0) | 1 | Bits 0-3<br>X'0'<br>Bits 4-7<br>X'1'<br>X'2' | Format of XID:<br>Fixed format, only bytes 0-5 are included.<br>Type of the XID-sending node<br>T1: PU type 1<br>T2: PU type 2 |
| 1 (1) | 1 | X'00' | Length in binary of XID; reserved for fixed-format XID. |
| 2 (2) | 4 | Bits 0-11<br>Bits 12-31 | Node Identification<br>Block number (IDBLK)<br>ID number (IDNUM) |
| 6 (6) | 1 | | Length in binary of the CV X'25', including this field. |
| 7 (7) | 1 | X'25' | Control vector type. |
| 8 (8) | 1 | X'00' | Reserved. |
| 9 (9) | 1 | | Length in binary of following field. |
| 10 (A) | n | | Security Identification. See "Build Vector" for details on this field. |
| 10 (A) | m | | calling DTE address (variable length) This field can be reduced to one byte with an X'FF' value in the event that NPSI did not get an X.25 calling DTE address from the Incoming Call packet. |
| 10+m | 1 | * | Separator character (optional). |
| 11+m | 1 | X'03' | Length of CV X'01', including this field. |
| 12+m | 1 | X'01' | Control Vector Type. |
| 13+m | 1 | <br>X'01'<br>X'02'<br>X'03'<br>X'04' | LLC type for this connection:<br>LLC5: Integrated PAD<br>LLC5: Transparent PAD<br>LLC4: GATE<br>LLC0: PCNE |
| 14+m | 1 | X'03' | Length of CV X'02', including this field. |
| 15+m | 1 | X'02' | Control Vector Type. |
| 16+m | p | - | Call User Data field (from 1st to 16th byte). |

***Build Vector (X'02'):*** The exit returns the build vector to VTAM when it is called by the XID vector (X'01'). The exit must return the address of the configuration services exit routine parameter output list in register 1 and a return code in register 15. The build vector returns MODEL PU and LU names, and real PU and LU names for VTAM to use when defining the switched resource. Table 6-5 shows the build vector format.

*Table 6-5. Build Vector Format*

| Dec (Hex) Offset | Size (Bytes) | Bit setting | Description |
|---|---|---|---|
| 0 (0) | 1 | B'1... ....'<br><br><br>B'.xxx xxx' | This bit tells VTAM how to process the REQCONT. If this bit is on, VTAM processes the REQCONT without building any dynamic resources. The REQCONT eventually fails when VTAM does not find the resources.<br>The remaining bits are reserved. |
| 1 (1) | 1 | | This is byte 0 for the first resource entry block. The resource entry block formats are described in “Resource Entry Blocks” on page 6-7. |

***Resource Entry Blocks:*** Your exit must build a set of resource entry blocks for each dynamic resource associated with the XID vector. The first resource entry block must be for a PU definition. A block is needed for every dependent LU attached to the PU. The exit cannot build independent LUs from a model. If the exit attempts to build an independent LU, the resource entry block for that LU fails. You can have one PU entry and, depending on the PU type, a variable number of LU entries. The format for each entry is the same.

VTAM converts the PU, LU, or model resource names returned in the resource entry blocks to uppercase characters. A resource entry block fails if it returns an invalid PU, LU, or model resource name. Table 6-6 shows the build vector format.

*Table 6-6. Resource Entry Block Format*

| Dec (Hex) Offset | Size (Bytes) | Bit setting | Description |
|---|---|---|---|
| 0 (0) | 1 | <br>X '71'<br>X '81' | Resource Entry Block Type<br>This block is for a PU.<br>This block is for an LU. |
| 1 (1) | 1 | B'1... ....'<br>B'.xxx xxxx' | If this bit is turned on, the value coded in the address override field (byte hex 12 below) overrides the address in the model definition.<br>The remaining bits are reserved. |
| 2 (2) | 8 | | This is the minor node name of the dynamically created PU or LU represented by this resource entry block. VTAM recognizes the dynamic device by this name. |
| 10 (A) | 8 | | This is the name of a model resource defined in a model major node. |
| 18 (12) | 1 | | This address override field contains one of the following addresses:<br>• If this resource entry block represents a PU, this field contains a station address.<br>• If this resource entry block represents an LU, this field contains a local address.<br>The value in this field is ignored unless the bit in byte hex 1 of this resource entry block is set to indicate that the value is valid. For more information on the local and station address fields, see the *VTAM Resource Definition Reference* that applies to your platform. |
| 19 (13) | 1 | | This byte is the start of the next resource entry block. If there are no more resource entry blocks, this byte does not appear. |

***Connection Status Vector (*** X'03'***):*** VTAM calls the exit with the connection status vector whenever the connection status of a switched PU type 1, 2, or 2.1 changes. You can use this vector to record the connection and disconnection times of switched devices. VTAM does not expect the exit to return any information besides a return code. Table 6-7 shows the connection status format.

*Table 6-7. Connection Status Format*

| Dec (Hex) Offset | Size (Bytes) | Bit setting | Description |
|---|---|---|---|
| 0 (0) | 1 | B'1... ....'<br>B'.1.. ....'<br>B'..xx xxxx' | If this bit is turned on, the device named in the vector is connecting to the network.<br>If this bit is turned on, the device named in the vector is disconnecting from the network.<br>The remaining bits are reserved. |
| 1 (1) | 8 | | This field contains the name of the switched line to which the PU is, or was, connected. |
| 9 (9) | 8 | | This field contains the name of the PU whose status changed. |
| 17 (11) | 8 | | This field contains the time stamp that tells when the PU's status changed. The time stamp is retrieved from the system time-of-day clock. |
| 25 (19) | 8 | | This field contains the station ID of the PU, if available. |
| 33 (21) | 8 | | This field contains the CPNAME of the PU, if available. |

***Failure Vector (*** X'04'***):*** VTAM calls the exit with the failure vector when a dynamic build request fails. VTAM does not expect the exit to return any information. Table 6-8 shows the failure vector format.

*Table 6-8. Failure Vector Format*

| Dec (Hex) Offset | Size (Bytes) | Bit setting | Description |
|---|---|---|---|
| 0 (0) | 1 | B'1... ....'<br>B'.1.. ....'<br>B'..1. ....'<br>B'...x xxxx' | If this bit is turned on, the dynamic build of a PU failed.<br>If this bit is turned on, the dynamic build of a LU failed.<br>If this bit is turned on, the ISTDSWMN major node is not active.<br>The remaining bits are reserved. |
| 1 (1) | 8 | | This field contains the name the exit provided for VTAM to use as the name of the dynamically defined device. |
| 9 (9) | 8 | | This field contains the name of the model resource VTAM used to dynamically build the device. |

***End Vector (*** X'FF'***):*** VTAM calls the exit with the end vector during normal HALT processing to tell the exit that VTAM is terminating. VTAM does not expect the exit to return any information. Table 6-9 shows the end vector format.

*Table 6-9. End Vector Format*

| Dec (Hex) Offset | Size (Bytes) | Bit setting | Description |
|---|---|---|---|
| 0 (0) | 2 | | Reserved |

# Installation Exit Process

One of the key data for the exit is a user-defined table indexed by the X.25 calling DTE addresses. After the key data has been defined, the process takes place according to VTAM schedules.

## X.25 DTE Address Table or File

This table is either:

- An actual table assembled with the exit
- An external file

The second solution is recommended for the following reasons:

- The file can be updated more easily by dynamically adding or deleting the X.25 DTE addresses while running.
- The file can be shared by other applications, such as:
  - An application allowing the file to be updated when VTAM and the exit are running
  - A CTCP application, if GATE is used, allowing the CTCP to know which PU and LU names a specific Incoming Call connection used in VTAM and NetView
- This solution avoids the storage problems that can arise with an actual table for a very large number of X.25 DTE addresses.
- The file can be either sequential, if the number of different X.25 DTE addresses allows it, or indexed to reduce the address search time when this number is very large.

With both solutions, each table entry or file record can have the following format and contents:

**Note:** These suggestions do not preclude any other information or special process to be added.

- The KEY X.25 DTE address can have a variable length in bytes depending on the network accessed.
- The length of the MODEL PU name in 1 byte.
  - The MODEL PU name for this DTE, up to 8 bytes.
- The length of the MODEL LU name in 1 byte.
  - The MODEL LU name for this DTE, up to 8 bytes.
- The length of the ACTUAL PU name in 1 byte.
  - The ACTUAL PU name to create for this DTE, up to 8 bytes.
- The length of the ACTUAL LU name in 1 byte.
  - The ACTUAL LU name to create for this DTE, up to 8 bytes.
- A byte for special process flags, such as:
  - FLAG X'80'. This entry represents more than one DTE with several terminals connected through a PAD or a PAD for Videotext terminals (PAV). The X.25 DTE address identifies the PAD or PAV rather than the terminal attached downstream, which is not a pure X.25 DTE, but usually an ASCII terminal.

The ACTUAL PU and LU names shown previously are dynamically updated. They can consist of a fixed header part, for example 4 bytes long, and a variable trailer part. This trailer part can be an EBCDIC numeric value starting at ‘0000’ and incremented by 1 every time the name has to be created. After this entry has been used 9999 times, the value returns to ‘0000’.

Some downstream terminals can provide their unique identification in the CUD field, generally starting from the 5th byte with a maximum of 12 bytes. In this case, a second user table is required to identify the calling party, no longer from the calling DTE address, but from the CUD. When the DTE attached downstream can be uniquely identified, it can get unique names.

- FLAG X'40' This entry is used for the DTE addresses that do not have a corresponding entry. The call can still be accepted, if the customer decides to do so. With this entry, you can force a session (LOGAPPL= keyword) to be established with a specific control access application that can request more valid identification information from the user at the DTE.

  Because this entry can be used more than once, the ACTUAL PU and LU names are dynamically updated.

- FLAG X'FF' This entry is used when NPSI cannot find an calling DTE address in the Incoming Call packet. Since the data in the CV X'25' cannot be NULL, NPSI has built a CV X'25' with a 1 byte data consisting of X'FF'.

  Because this entry can be used more than once, the ACTUAL PU and LU names are dynamically updated.

When the data the exit uses has been described, the different phases of the process can be defined according to the various schedules from VTAM.

## Initialization

When VTAM starts, the exit is entered once with a begin vector (X'00'). to enable it to perform any initialization process required, such as opening the appropriate files, especially those providing the DTE identifications previously described.

Upon returning to VTAM, the begin vector must be set according to the following process specified in the exit:

- Bit 0 must be set to ON to allow the exit to participate in the VTAM REQCONT process and to dynamically build DSMN entries.
- Bit 1 must be set to ON if the exit wants to know the connection status information on all switched devices through the connection status vector (X'03').

## Incoming Call Process

Anytime VTAM receives a REQCONT for which it does not find an IDBLK and IDNUM match in the SMN, the exit is entered with an XID Vector (X'01') containing the XID data and the CV X'25' built by NPSI.

The exit then compares the X.25 DTE address given in the CV25 to the entries in the indexed file. If a match is found, the exit builds a build vector (X'02') and passes it to VTAM. This vector must include resource entry blocks for both the PU and the LU that are the SNA representation of the calling X.25 DTE. Through the build vector bit setting, the exit can refuse to create a dynamic entry for this call in the DSMN and can let VTAM deny the connection.

If the build vector passed to VTAM is incorrect, the exit can be entered again with a failure vector (X'04') that fully explains the error found by VTAM.

## Connection Status

If this function has been requested through the begin vector returned to VTAM by the exit, the exit is entered with a connection status vector (X'03') when the connection status of a switched device changes. This enables the exit to record the connection and disconnection times for the switched devices.

When the X.25 DTE address does not fully identify a terminal, such as one connected through a PAD/PAV, the exit can use this connection status vector to find out when a dynamically named resource is no longer in use. For more information, see “X.25 DTE Address Table or File” on page 6-9.

## Termination

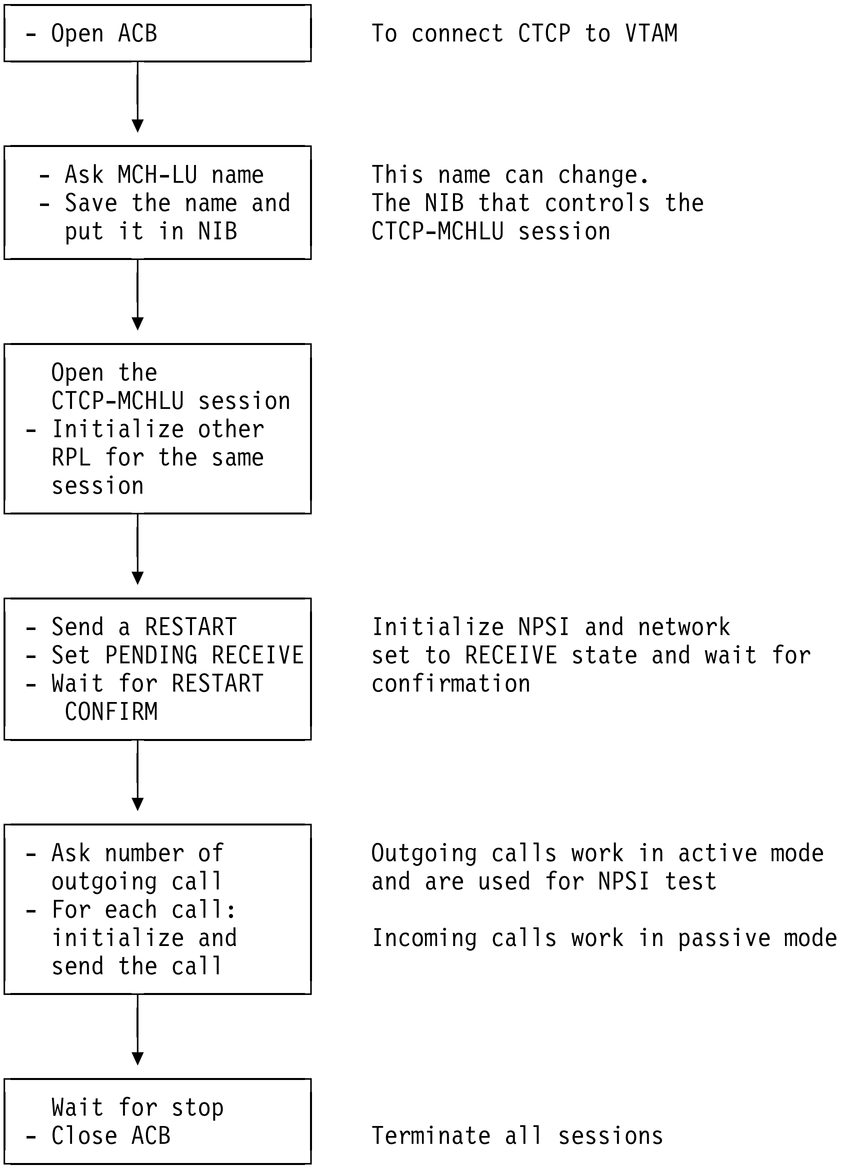
VTAM calls the exit with an end vector (X'FF') during its normal HALT processing. The exit can then close the files opened at initialization.

# Appendix. Example of Programming a DATE CTCP

# Appendix. Example of Programming a DATE CTCP

This appendix provides an example of the programming required to create a CTCP. The programming example includes use of the DATE function of NPSI.

## Main Flow of the CTCP



```
- Open ACB                      To connect CTCP to VTAM

 - Ask MCH-LU name              This name can change.
 - Save the name and            The NIB that controls the
   put it in NIB                CTCP-MCHLU session

  Open the
  CTCP-MCHLU session
- Initialize other
  RPL for the same
  session

- Send a RESTART                Initialize NPSI and network
- Set PENDING RECEIVE           set to RECEIVE state and wait for
- Wait for RESTART              confirmation
   CONFIRM

- Ask number of                 Outgoing calls work in active mode
  outgoing call                 and are used for NPSI test
- For each call:
  initialize and                Incoming calls work in passive mode
  send the call

  Wait for stop
- Close ACB                     Terminate all sessions
```

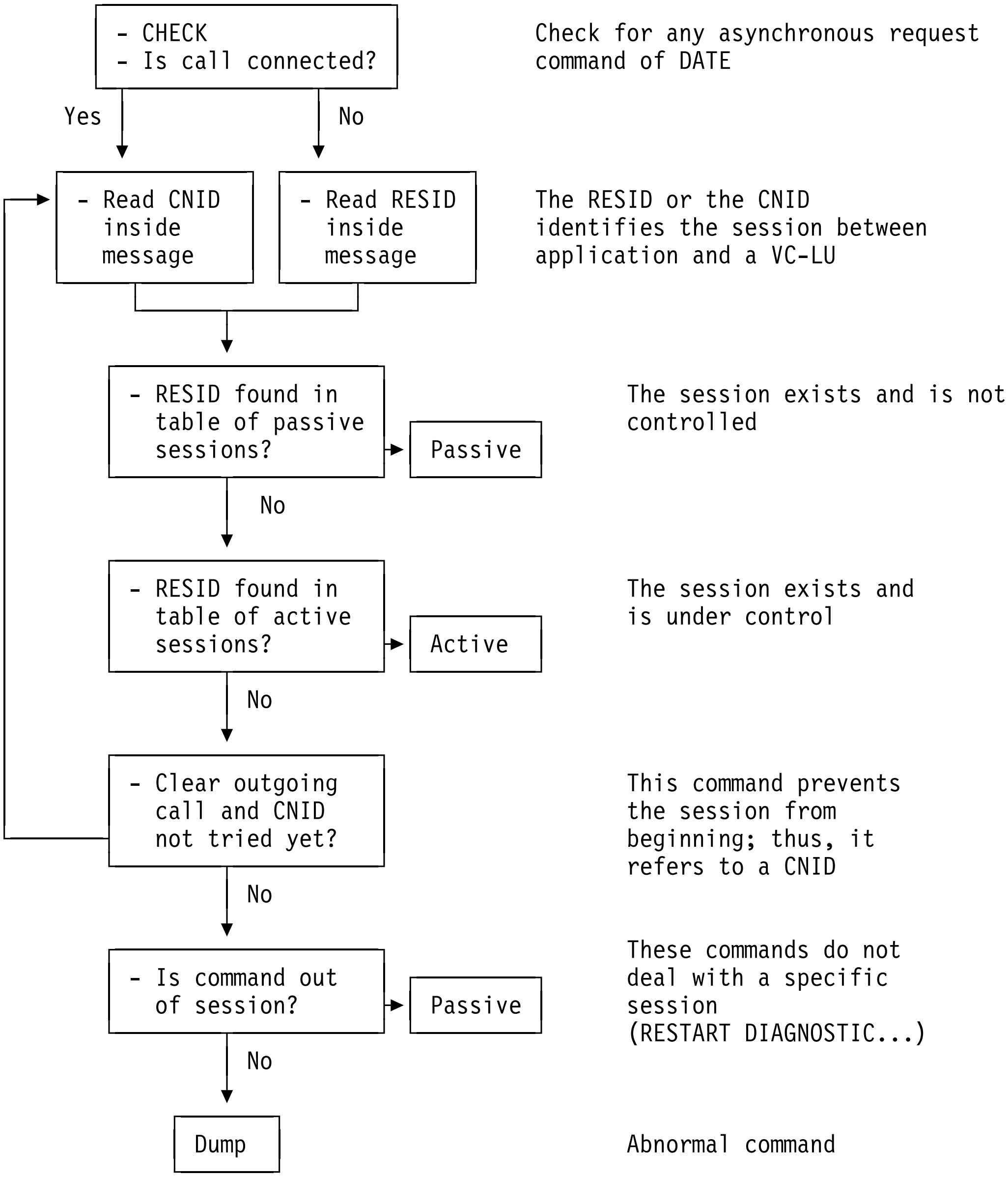
## Description of Sample CTCP

This CTCP has two parts:

- Active part: If callouts are requested, the callout part of this CTCP serves to test the NPSI DATE function. A certain number of commands are sent, and the CTCP verifies that the received responses are the expected ones.

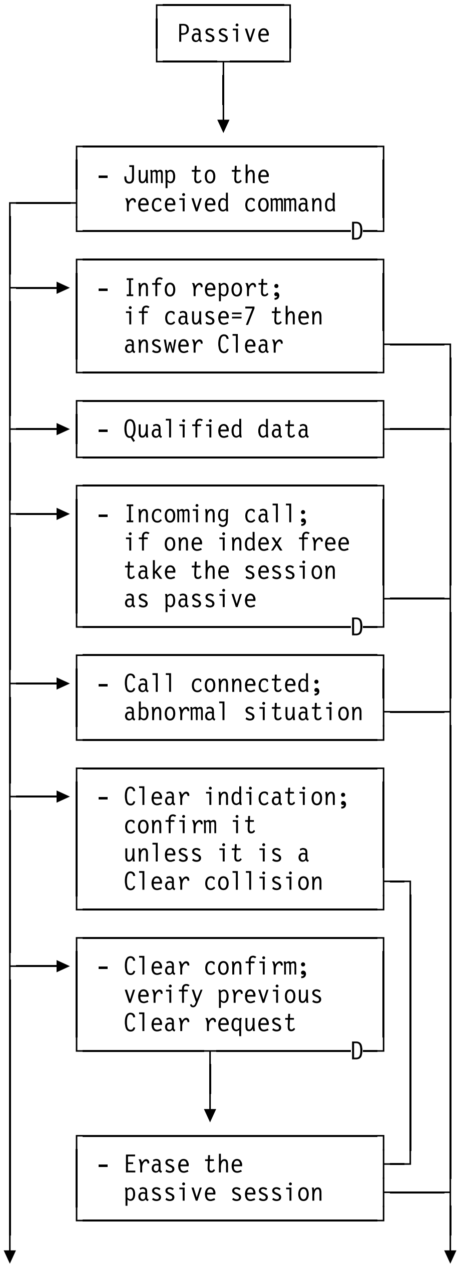
  Because this CTCP is prepared to handle only one MCH, the call requests cause incoming calls on other virtual circuits of the same MCH. Similar logic applies for the other Control or Data packets.
- Passive part: This part receives incoming calls, and does not serve to test the NPSI DATE function. This part has the normal DATE CTCP reactions.

## Receive Exit Routine


- CHECK
- Is call connected?
Yes
No
- Read CNID inside message
- Read RESID inside message
- RESID found in table of passive sessions?
Passive
No
- RESID found in table of active sessions?
Active
No
- Clear outgoing call and CNID not tried yet?
No
- Is command out of session?
Passive
No
Dump
Check for any asynchronous request command of DATE
The RESID or the CNID identifies the session between application and a VC-LU
The session exists and is not controlled
The session exists and is under control
This command prevents the session from beginning; thus, it refers to a CNID
These commands do not deal with a specific session (RESTART DIAGNOSTIC...)
Abnormal command

## Passive Part of the Receive Routine

**Note:** If the stated conditions are not true, a D (dump) appears in the bottom right corner of the frame, indicating that a dump should be performed.


Passive
- Jump to the received command
D
- Info report; if cause=7 then answer Clear
- Qualified data
- Incoming call; if one index free take the session as passive
D
- Call connected; abnormal situation
- Clear indication; confirm it unless it is a Clear collision
- Clear confirm; verify previous Clear request
D
- Erase the passive session

Error if the command is unknown

Application ends session

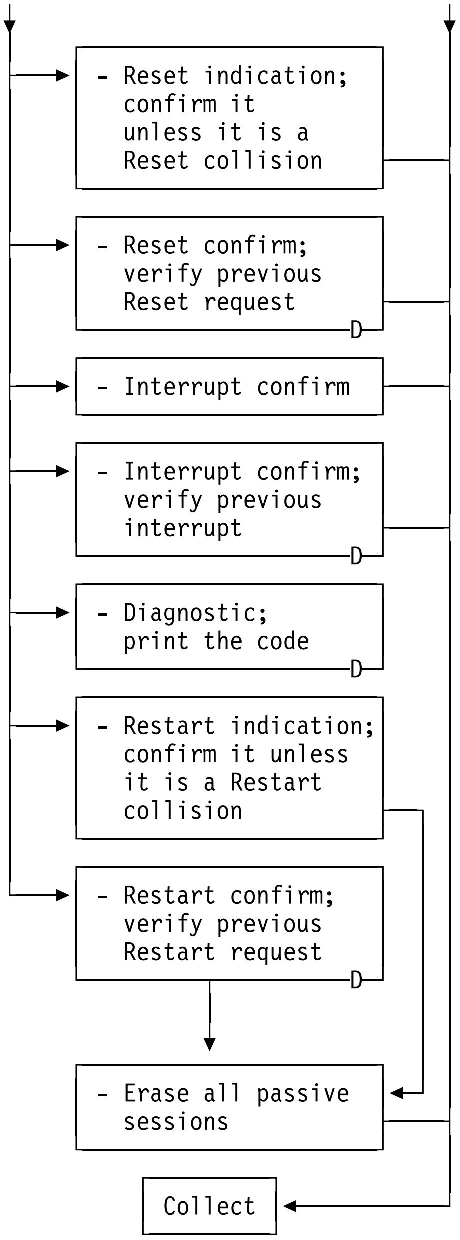
Incoming call already contains the RESID of the session. If the passive table is full, the active part or TPNS has asked more sessions than the CTCP can support

Because the passive part does not send a Call Request

Otherwise it is an error

Clear means end of session

**Note:** This figure is continued on the next page.


- Reset indication; confirm it unless it is a Reset collision
- Reset confirm; verify previous Reset request
D
- Interrupt confirm
- Interrupt confirm; verify previous interrupt
D
- Diagnostic; print the code
D
- Restart indication; confirm it unless it is a Restart collision
- Restart confirm; verify previous Restart request
D
- Erase all passive sessions
Collect

Otherwise it is an error

Interrupt collision does not exist

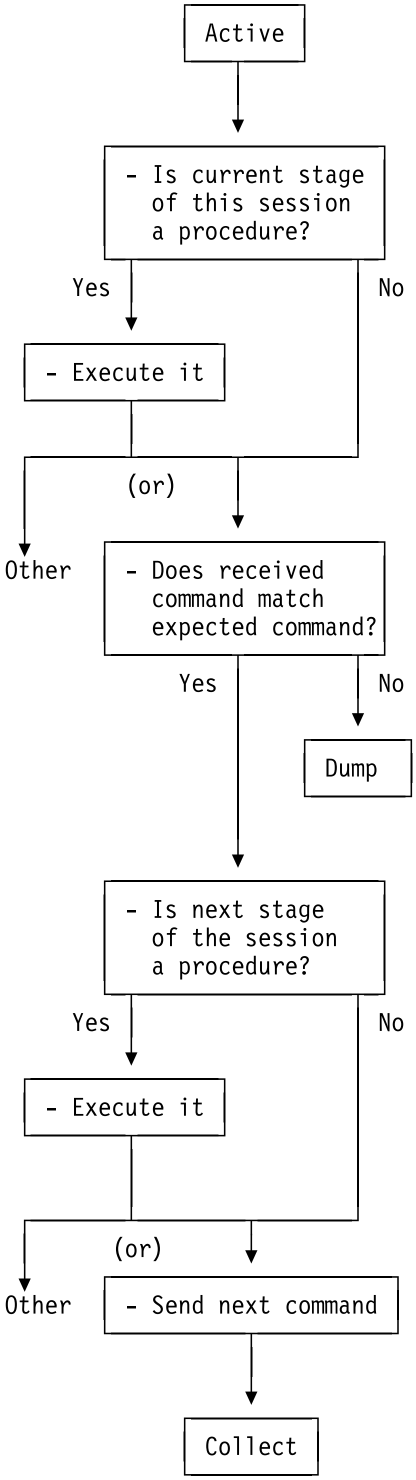
Otherwise it is an error

Situation severe enough to stop

Otherwise it is an error

Restart means end of all sessions

## Active Part of the Receive Routine


Active
- Is current stage of this session a procedure?
Yes
No
- Execute it
(or)
Other
- Does received command match expected command?
Yes
No
Dump
- Is next stage of the session a procedure?
Yes
No
- Execute it
(or)
Other
- Send next command
Collect

The current stage is a pointer excerpt of a table (the script)

The procedure gives control to any point or gives the next command

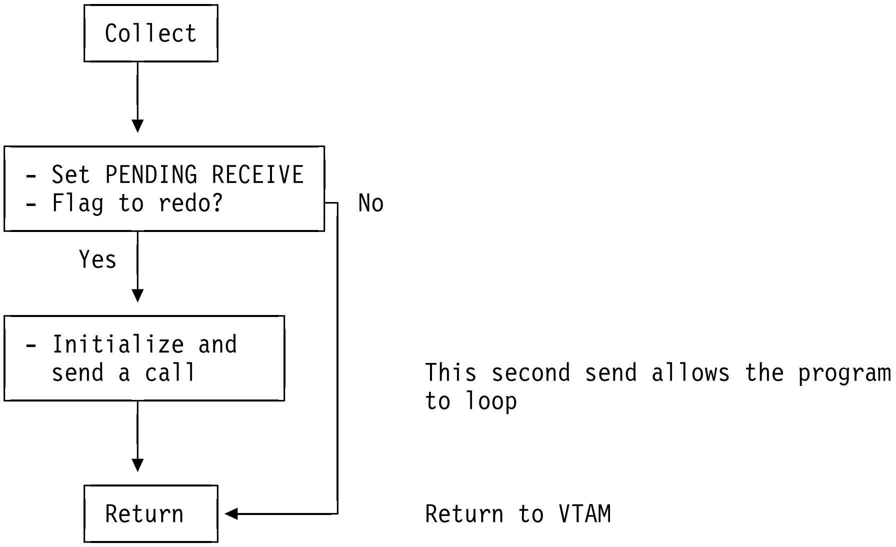
Allow full control of the session

Session is progressing abnormally

The next stage is a pointer excerpt of a table (the script)

The procedure gives control to any point or gives the next command

### End of the Receive Routine



This second send allows the program to loop

Return to VTAM

## Sample Output Assembly of a CTCP Program

The following is a sample of output assembly generated by a CTCP program using the DATE function of NPSI.

```
                                              EXTERNAL SYMBOL DICTIONARY
SYMBOL   TYPE  ID  ADDR  LENGTH  LD ID FLAGS                                                         ASM H V 02 16.17 09/27/88
CTCPSTR   SD 0001 000000 0013D4         00

  LOC  OBJECT CODE    ADDR1 ADDR2   STMT  SOURCE STATEMENT                                           ASM H V 02 16.17 09/27/88
000000                              517+CTCPSTR  CSECT                              * CODE SECTION NAME         01-00055
000000                              518+         DS    0H                                                       02-SAVE
000000 90EC D00C            0000C   519+         STM   14,12,12(13)                      SAVE REGISTERS         02-SAVE
000004 0530                         520+         BALR  R3,R0                        * SET FIRST BASE            01-00057
                            00006   521+         USING *,R3,R4,R5                   * ASSEMBLER DIRECTIVE       01-00058
000006 5840 300E            00014   522+EXTBASES L     R4,B10000                    * SET SECOND BASE           01-00059
00000A 5850 3012            00018   523+         L     R5,B20000                    * SET THIRD BASE            01-00060
00000E 47F0 3016            0001C   524+         B     LB0000                       * SKIP BASES VALUES         01-00061
000012 0000
000014 00001006                     525+B10000   DC    A(EXTBASES+4096)             * VALUE OF SECOND BASE      01-00062
000018 00002006                     526+B20000   DC    A(EXTBASES+8192)             * VALUE OF THIRD BASE       01-00063
00001C 18CD                         527+LB0000   LR    R12,R13                      * OLD SAVE AREA IN R12      01-00064
00001E 41D0 33D6            003DC   528+         LA    R13,MAINSAVE                 * NEW SAVE AREA IN R13      01-00065
000022 50DC 0008            00008   529+         ST    R13,8(R12)                   * NEW POINTER IN OLD AREA   01-00066
000026 50C0 33DA            003E0   530+         ST    R12,MAINSAVE+4               * OLD POINTER IN NEW AREA   01-00067
                                    531+         PRINT NOGEN                                                    02-00042
                                    532 * OPEN ACB MCH ******************************************************** CTC51700
                                    533          OPEN  ACBMCH                                                   CTC51800
                                    539          MVERIFY                            * VERIFY OPEN COMPLETION    CTC51900
                                    540+         PRINT GEN                                                      02-00033
000036 12FF                         541+         LTR   R15,R15                      * TEST RETURN CODE          01-00282
000038 4780 303A            00040   542+         BZ    MV0001                       * NO PROBLEM                01-00283
00003C 45E0 3446            0044C   543+         BAL   R14,FAILEXIT                 * DUMP                      01-00284
                            00040   544+MV0001   EQU   *                            * RESUME STATEMENTS         01-00285
                                    545+         PRINT NOGEN                                                    02-00042
                                    546 * GET MCH LU NAME ***************************************************** CTC52000
000040 D703 3432 3432 00438 00438   547          XC    WTORECB(4),WTORECB           * ERASE WTORECB FOR USE     CTC52100
000046 D207 341E 3426 00424 0042C   548          MVC   REPLY(8),BLANK8              * INITIALISE REPLY          CTC52200
                                    549          WTOR  ' CTCP ||| ENTER MCH LU NAME (DU3) :',REPLY,8,WTORECB    CTC52300
                                    560          WAIT  1,ECB=WTORECB                * WAIT FOR OPERATOR REPLY   CTC52400
00008C D507 341E 3426 00424 0042C   564          CLC   REPLY(8),BLANK8              * COMPARE TO NO ANSWER      CTC52500
000092 4770 3096            0009C   565          BNE   UPCASE                       * THERE IS AN ANSWER        CTC52600
000096 D202 341E 3F88 00424 00F8E   566          MVC   REPLY(3),=C'DU3'             * TAKE DEFAULT ANSWER       CTC52700
00009C D607 341E 3426 00424 0042C   567 UPCASE   OC    REPLY(8),BLANK8              * CONVERT INTO UPPER CASE   CTC52800
0000A2 D207 30B9 341E 000BF 00424   568          MVC   WTONAME+23(8),REPLY          * TRANSFERT TO STRING       CTC52900
                                    569 WTONAME  WTO   ' MCH LU NAME :         '    * CONFIRM NAME              CTC53000
                                    577          MODCB NIB=NIBMCH,NAME=(*,REPLY)    * PUT MCH LU NAME IN NIB    CTC53100
                                    597          MVERIFY                            * VERIFY MODCB COMPLETION   CTC53200
                                    598+         PRINT GEN                                                      02-00033
000104 12FF                         599+         LTR   R15,R15                      * TEST RETURN CODE          01-00282
000106 4780 3108            0010E   600+         BZ    MV0002                       * NO PROBLEM                01-00283
00010A 45E0 3446            0044C   601+         BAL   R14,FAILEXIT                 * DUMP                      01-00284
                            0010E   602+MV0002   EQU   *                            * RESUME STATEMENTS         01-00285
                                    603+         PRINT NOGEN                                                    02-00042
                                    604 * ESTABLISH SESSION BETWEEN CTCP AND MCH, INITIALISE SEND RPL ********* CTC53300
                                    605          OPNDST RPL=RPLMCH,OPTCD=(SYN,ACQUIRE,START)                    CTC53400
                                    664          MVERIFY                            * VERIFY OPNDST COMPLETION  CTC53500
                                    665+         PRINT GEN                                                      02-00033
00015E 12FF                         666+         LTR   R15,R15                      * TEST RETURN CODE          01-00282
000160 4780 3162            00168   667+         BZ    MV0003                       * NO PROBLEM                01-00283
000164 45E0 3446            0044C   668+         BAL   R14,FAILEXIT                 * DUMP                      01-00284
                            00168   669+MV0003   EQU   *                            * RESUME STATEMENTS         01-00285
                                    670+         PRINT NOGEN                                                    02-00042
                                    671          SHOWCB AM=VTAM,RPL=RPLMCH,FIELDS=ARG,AREA=NIBARG,LENGTH=4      CTC53600
                                    689          MVERIFY                            * VERIFY SHOWCB COMPLETION  CTC53700
                                    690+         PRINT GEN                                                      02-00033
000198 12FF                         691+         LTR   R15,R15                      * TEST RETURN CODE          01-00282
00019A 4780 319C            001A2   692+         BZ    MV0004                       * NO PROBLEM                01-00283
00019E 45E0 3446            0044C   693+         BAL   R14,FAILEXIT                 * DUMP                      01-00284
                            001A2   694+MV0004   EQU   *                            * RESUME STATEMENTS         01-00285
                                    695+         PRINT NOGEN                                                    02-00042
                                    696          MODCB AM=VTAM,RPL=RPLMCHS,ARG=(*,NIBARG)                       CTC53800
                                    715          MVERIFY                            * VERIFY MODCB  COMPLETION  CTC53900
                                    716+         PRINT GEN                                                      02-00033
0001D8 12FF                         717+         LTR   R15,R15                      * TEST RETURN CODE          01-00282
0001DA 4780 31DC            001E2   718+         BZ    MV0005                       * NO PROBLEM                01-00283
0001DE 45E0 3446            0044C   719+         BAL   R14,FAILEXIT                 * DUMP                      01-00284
                            001E2   720+MV0005   EQU   *                            * RESUME STATEMENTS         01-00285
                                    721+         PRINT NOGEN                                                    02-00042
                                    722 * SEND RESTART REQUEST ************************************************ CTC54000
0001E2 D701 3F9E 3F9E 00FA4 00FA4   723          XC    RESIVAL(2),RESIVAL           * NULLIFY RESID FOR RESTART CTC54100
0001E8 92FF 3F8C      00F92         724          MVI   RSARBOOL,TRUE                * SET RESTART BOOLEAN       CTC54200
                                    725          SETSEND RSARREQ                    * SEND RESTART REQUEST      CTC54300
                                    726+         PRINT GEN                                                      02-00033
0001EC 4170 4073            01079   727+         LA    R7,RSARREQ                   * GET ADRESS OF DATA        01-00262
0001F0 4187 0001            00001   728+         LA    R8,1(R7)                     * GET ADRESS OF STRING      01-00266
0001F4 1799                         729+         XR    R9,R9                        * NULLIFY R9                01-00267
0001F6 4397 0000            00000   730+         IC    R9,0(R7)                     * GET LENGTH OF STRING      01-00268
0001FA 45E0 3B06            00B0C   731+         BAL   R14,SBRSEND                  * CALL SEND PROCEDURE       01-00269
                                    732+         PRINT NOGEN                                                    02-00042
                                    733 * SET PENDING RECEIVE MCH EXIT ROUTINE ******************************** CTC54400
0001FE D200 42CA 3F8B 012D0 00F91   734          MVC   AREAMCHR(1),=C' '            * ERASE AREAMCHR            CTC54500
000204 D2FE 42CB 42CA 012D1 012D0   735          MVC   AREAMCHR+1(255),AREAMCHR     * ERASE AREAMCHR            CTC54600
00020A D203 43CA 43C9 013D0 013CF   736          MVC   AREAMCHR+256(4),AREAMCHR+255 * ERASE AREAMCHR            CTC54700
                                    737          RECEIVE RPL=RPLMCH,AREA=AREAMCHR,RECLEN=260,                  CCTC54800
                                                       OPTCD=(ASY,SPEC),RTYPE=(DFSYN,DFASY,NRESP),EXIT=RCVMCH   CTC54900
                                    785          MVERIFY                            * VERIFY RECEIVE COMPLETION CTC55000
                                    786+         PRINT GEN                                                      02-00033
000274 12FF                         787+         LTR   R15,R15                      * TEST RETURN CODE          01-00282
000276 4780 3278            0027E   788+         BZ    MV0006                       * NO PROBLEM                01-00283
00027A 45E0 3446            0044C   789+         BAL   R14,FAILEXIT                 * DUMP                      01-00284
                            0027E   790+MV0006   EQU   *                            * RESUME STATEMENTS         01-00285
                                    791+         PRINT NOGEN                                                    02-00042
                                    792 * WAIT FOR RESTART CONFIRM CORRESPONDING TO RESTART SENT ABOVE ******** CTC55100
                                    793          WAIT  1,ECB=RSARECB                * POST ECB IN RECEIVE EXIT  CTC55200
000288 D703 3436 3436 0043C 0043C   797          XC    RSARECB(4),RSARECB           * ERASE RSARECB FOR USE     CTC55300
                                    798 *********************************************************************** CTC55400
                                    799 *        ACTIV PART OF MAINLINE                                       * CTC55500
                                    800 *********************************************************************** CTC55600
                                    801 *                                                                       CTC55700
                                    802 * GET DIALOGUE QUANTITY ********************************************** CTC55800
00028E D703 3432 3432 00438 00438   803 NUMAGAIN XC    WTORECB(4),WTORECB           * ERASE WTORECB FOR USE     CTC55900
000294 D201 341E 3F7E 00424 00F84   804          MVC   REPLY(2),=C'01'              * INITIALISE REPLY          CTC56000
                                    805          WTOR  ' CTCP ||| ENTER NUMBER OF OUTGOING CALL (01-04) :',    CCTC56100
                                                       REPLY,2,WTORECB              * REPLY <= QUANTITY         CTC56200
                                    816          WAIT  1,ECB=WTORECB                * WAIT FOR OPERATOR REPLY   CTC56300
0002EA F221 343F 341E 00445 00424   820          PACK  NUMBER+5(3),REPLY(2)         * CONVERT TO PACKED FORMAT  CTC56400
0002F0 4FC0 343A            00440   821          CVB   R12,NUMBER                   * CONVERT TO BINARY         CTC56500
0002F4 49C0 3F0A            00F10   822          CH    R12,QUANTITY                 * COMPARE TO MAXIMUM        CTC56600
0002F8 4720 3288            0028E   823          BH    NUMAGAIN                     * REPLY TOO HIGH            CTC56700
0002FC 12CC                         824          LTR   R12,R12                      * TEST                      CTC56800
0002FE 4780 3288            0028E   825          BZ    NUMAGAIN                     * REFUSE ZERO               CTC56900
000302 40C0 3F0A            00F10   826          STH   R12,QUANTITY                 * STORE ACCEPTED VALUE      CTC57000
```

```
                                    827 * SEND ALL REQUESTED CALL OUT ***************************************** CTC57100
000306 4890 3F0A            00F10   828          LH    R9,QUANTITY                  * GET TOTAL TO START        CTC57200
00030A 4090 3F96            00F9C   829 NEXTSES  STH   R9,INDEX                     * FREE R9 REGISTER          CTC57300
00030E 45E0 3D30            00D36   830          BAL   R14,INISESS                  * INITIALISE ONE SESSION    CTC57400
000312 4890 3F96            00F9C   831          LH    R9,INDEX                     * RETRIEVE CURRENT INDEX    CTC57500
000316 4690 3304            0030A   832          BCT   R9,NEXTSES                   * REPEAT COMPARISON         CTC57600
                                    833 *********************************************************************** CTC57700
                                    834 *        END OF ACTIV                                                 * CTC57800
                                    835 *********************************************************************** CTC57900
                                    836 *                                                                       CTC58000
                                    837 * GET STOP ************************************************************ CTC58100
00031A D703 3432 3432 00438 00438   838 STOPLOOP XC    WTORECB(4),WTORECB           * ERASE WTORECB FOR USE     CTC58200
000320 D202 341E 342E 00424 00434   839          MVC   REPLY(3),DANSWER             * INITIALISE REPLY          CTC58300
                                    840          WTOR  ' CTCP ||| ENTER END : ',REPLY,3,WTORECB                 CTC58400
                                    851          WAIT  1,ECB=WTORECB                * WAIT FOR OPERATOR REPLY   CTC58500
00035A D602 341E 3426 00424 0042C   855          OC    REPLY(3),BLANK8              * CONVERT INTO UPPER CASE   CTC58600
000360 D502 341E 342E 00424 00434   856          CLC   REPLY(3),DANSWER             * LOOK IF END               CTC58700
000366 4770 3314            0031A   857          BNE   STOPLOOP                     * UNKNOWN ANSWER; RETRY     CTC58800
                                    858 * TERMINATE APPLICATION BY CLOSING MCH ACB **************************** CTC58900
                                    859          CLOSE ACBMCH                                                   CTC59000
                                    865          MVERIFY EXITCTCP                   * VERIFY CLOSE COMPLETION   CTC59100
                                    866+         PRINT GEN                                                      02-00033
000376 12FF                         867+         LTR   R15,R15                      * TEST RETURN CODE          01-00282
000378 4780 337A            00380   868+         BZ    MV0007                       * NO PROBLEM                01-00283
00037C 45E0 3446            0044C   869+         BAL   R14,FAILEXIT                 * DUMP                      01-00284
                            00380   870+MV0007   EQU   *                            * RESUME STATEMENTS         01-00285
                                    871+         PRINT NOGEN                                                    02-00042
                                    872 EXITCTCP MEPILOG                            * RETURN TO MVS             CTC59200
                                    873+         PRINT GEN                                                      02-00033
000380 58D0 33DA            003E0   874+EXITCTCP L     R13,MAINSAVE+4               * RESTORE POINTER TO OLD SA 01-00076
000384 98EC D00C            0000C   875+         LM    14,12,12(13)                     RESTORE THE REGISTERS   02-RETUR
000388 41F0 0000            00000   876+         LA    15,0(0,0)                        LOAD RETURN CODE        02-RETUR
00038C 07FE                         877+         BR    14                               RETURN                  02-RETUR
000390                              878+R14SAVE  DS    F                            * EXIT ROUTINE RETURN POINT 01-00078
000394 0000000000000000             879+SBRSAVE  DC    18F'0'                       * EXIT ROUTINE SAVE AREA    01-00079
0003DC 0000000000000000             880+MAINSAVE DC    18F'0'                       * MAIN PROGRAM SAVE AREA    01-00080
                                    881+         PRINT NOGEN                                                    02-00042
                                    882 * MAINLINE DATA ******************************************************* CTC59300
000424                              883 REPLY    DS    8C                           * FOR WTOR REPLY            CTC59400
00042C 4040404040404040             884 BLANK8   DC    8C' '                        * FOR COMPARE LOGICAL       CTC59500
000434 C5D5C4                       885 DANSWER  DC    C'END'                       * DEFAULT ANSWER            CTC59600
000438                              886 WTORECB  DS    F                            * FOR WTOR COMPLETION       CTC59700
00043C 00000000                     887 RSARECB  DC    F'0'                         * FOR RESTART CONFIRM       CTC59800
000440 0000000000000000             888 NUMBER   DC    D'0'                         * FOR PACKED CONVERSION     CTC59900
000448                              889 NIBARG   DS    F                            * FOR ARG TRANSFERT         CTC60000
                                    891 * PROCEDURE FAILEXIT ************************************************** CTC60200
                                    892 * INPUT : R14.                                                          CTC60300
                                    893 * OUTPUT: NONE.                                                         CTC60400
00044C 900F 34C2            004C8   894 FAILEXIT STM   R0,R15,FAILAREA              * STORE CONTEXT             CTC60500
000450 5BE0 3F6A            00F70   895          S     R14,=A(CTCPSTR)              * FAILURE OFFSET            CTC60600
000454 50E0 3502            00508   896          ST    R14,WORKN                    * PREPARE UNPACK            CTC60700
000458 F363 3506 3503 0050C 00509   897          UNPK  WORKC(7),WORKN+1(4)          * UNPACK HALFBYTE TO EBCDIC CTC60800
00045E DC05 3506 341F 0050C 00425   898          TR    WORKC(6),HEX                 * TRANSLATE EBCDIC TO HEXA  CTC60900
000464 D205 3484 3506 0048A 0050C   899          MVC   WTOMAC+30(6),WORKC           * PLACE RESULT IN WTO MSG   CTC61000
                                    900 WTOMAC   WTO   ' CTCP ||| FAILED AT : ______ '                          CTC61100
                                    908          JCLABEND                           * ABANDON AND DUMP CTCP     CTC61200
                                    909+         PRINT GEN                                                      02-00033
                                    910+*        ABEND 0,DUMP
                                    911+         PRINT NOGEN                        * COMMENT ABOVE IS ENOUGH   01-00249
                                    918+         PRINT NOGEN                                                    02-00042
                                    919 * FAILEXIT DATA ******************************************************* CTC61300
                                    920         MMARK ' FAILAREA '                 * FOR READABLE DUMP         CTC61400
                                    921+        PRINT GEN                                                      02-00033
0004A4 40C6C1C9D3C1D9C5             922+        DC    C' FAILAREA '                * GENERATE CONSTANT STRING  01-00492
0004AE 40C6C1C9D3C1D9C5             923+        DC    C' FAILAREA '                * GENERATE CONSTANT STRING  01-00493
0004B8 40C6C1C9D3C1D9C5             924+        DC    C' FAILAREA '                * GENERATE CONSTANT STRING  01-00494
0004C2 40C6C1C9D3C1D9C5             925+        DC    C' FAILAREA '                * GENERATE CONSTANT STRING  01-00495
0004CC                      004C8   926+        ORG   *-4                          * OVERLAP STRING            01-00496
0004C8                              927+        DS    0F                           * ALIGNEMENT STORAGE        01-00497
                                    928+        PRINT NOGEN                                                    02-00042
0004C8                              929 FAILAREA DS    16F                          * ERROR CONTEXT STORAGE     CTC61500
000508                              930 WORKN    DS    F                            * NUMERIC WORK STORAGE      CTC61600
00050C                              931 WORKC    DS    9C                           * CARACTER WORK STORAGE     CTC61700
                                    932 *                                           * NON REDUCTIBLE SIZE |     CTC61800
000515 F0F1F2F3F4F5F6F7             933 HEXTR    DC    C'0123456789ABCDEF'          * TRANSLATION TABLE         CTC61900
                            00425   934 HEX      EQU   HEXTR-240                    * TRANSLATION POINTER       CTC62000
                                    936 *********************************************************************** CTC62200
                                    937 *                                                                     * CTC62300
                                    938 *     R E C E I V E   O N   M C H   E X I T - R O U T I N E         * CTC62400
                                    939 *                                                                     * CTC62500
                                    940 *********************************************************************** CTC62600
                                    941 RCVMCH   MSBRIN                             * ESTABLISH ADRESSABILITY   CTC62700
                                    942+        PRINT GEN                                                      02-00033
                                    943+        PUSH  USING                        * TEMPORARY CHANGE BASE     01-00116
                                    944+        DROP  R3,R4,R5                     * TEMPORARY CHANGE BASE     01-00117
000525 00
000526 05F0                         945+RCVMCH   BALR  R15,0                        * TEMPORARY CHANGE BASE     01-00118
                            00528   946+        USING *,R15                        * TEMPORARY CHANGE BASE     01-00119
000528 5830 F010            00538   947+        L     R3,B00008                    * SET FIRST BASE            01-00120
00052C 5840 F014            0053C   948+        L     R4,B10008                    * SET SECOND BASE           01-00121
000530 5850 F018            00540   949+        L     R5,B20008                    * SET THIRD BASE            01-00122
000534 47F0 F01C            00544   950+        B     LB0008                       * SKIP BASES VALUES         01-00123
000538 00000006                     951+B00008   DC    A(EXTBASES)                  * VALUE OF FIRST BASE       01-00124
00053C 00001006                     952+B10008   DC    A(EXTBASES+4096)             * VALUE OF SECOND BASE      01-00125
000540 00002006                     953+B20008   DC    A(EXTBASES+8192)             * VALUE OF THIRD BASE       01-00126
                                    954+        DROP  R15                          * RESTORE BASES             01-00127
                                    955+        POP   USING                        * RESTORE BASES             01-00128
000544 50E0 338A            00390   956+LB0008   ST    R14,R14SAVE                  * KEEP RETURN ADRESS        01-00129
000548 41D0 338E            00394   957+        LA    R13,SBRSAVE                  * SET UP NEW SAVE AREA      01-00130
                                    958+        PRINT NOGEN                                                    02-00042
                                    959         CHECK RPL=RPLMCH                   * TO UNLOCK RPLMCH          CTC62800
                                    966         MVERIFY                            * VERIFY CHECK COMPLETION   CTC62900
                                    967+        PRINT GEN                                                      02-00033
00055E 12FF                         968+        LTR   R15,R15                      * TEST RETURN CODE          01-00282
000560 4780 3562            00568   969+        BZ    MV0009                       * NO PROBLEM                01-00283
000564 45E0 3446            0044C   970+        BAL   R14,FAILEXIT                 * DUMP                      01-00284
                            00568   971+MV0009   EQU   *                            * RESUME STATEMENTS         01-00285
                                    972+        PRINT NOGEN                                                    02-00042
                                    973 * CHOOSE BETWEEN CNID OR RESID ACCORDING TO COMMAND CODE ************** CTC63000
000568 950F 42CC      012D2         974         CLI   AREAMCHR+2,X'0F'             * LOOK AT COMMAND CODE      CTC63100
00056C 4780 3578            0057E   975         BE    WANTCNID                     * CALL CONNECTED            CTC63200
000570 D201 3F9A 42CA 00FA0 012D0   976         MVC   WANTED(2),AREAMCHR           * CHOOSE RESID              CTC63300
000576 940F 3F9A      00FA0         977         NI    WANTED,X'0F'                 * ERASE DIRECTION INDICATOR CTC63400
00057A 47F0 3586            0058C   978         B     SEEKID                       * GO TO TABLE SCANNING      CTC63500
00057E D201 3F9A 42CD 00FA0 012D3   979 WANTCNID MVC   WANTED(2),AREAMCHR+3         * CHOOSE CNID               CTC63600
000584 940F 3F9A      00FA0         980         NI    WANTED,X'0F'                 * ERASE DIRECTION INDICATOR CTC63700
000588 9680 3F9A      00FA0         981         OI    WANTED,CNIDSENT              * SET INTERNAL CNID FLAG    CTC63800
                                    982 * LOOK FOR SESSION INDEX ACCORDING TO RESID *************************** CTC63900
00058C 4890 3F0A            00F10   983 SEEKID   LH    R9,QUANTITY                  * POTENTIAL SESSIONS QUANT  CTC64000
000590 4090 3F96            00F9C   984 NEXTIDP  STH   R9,INDEX                     * FREE R9 REGISTER          CTC64100
```

```
                                    985          MTBLGET PASSTBL,INDEX,RESIVAL,HALF * GET RESID FROM PASSIV TBL CTC64200
                                    986+         PRINT GEN                                                      02-00033
000594 48B0 3F96            00F9C   987+         LH    R11,INDEX                    * PREPARE INDEX REGISTER    01-00179
000598 06B0                         988+         BCTR  R11,R0                       * 1 LESS TO CHANGE ORIGINE  01-00180
00059A 89B0 0001            00001   989+         SLL   R11,1                        * MULTIPLY BY 2             01-00190
00059E 48BB 3F14            00F1A   990+         LH    R11,(PASSTBL)(R11)           * EXTRACT TABLE ELEMENT     01-00191
0005A2 40B0 3F9E            00FA4   991+         STH   R11,RESIVAL                  * STORE IN WORK STORAGE     01-00192
                                    992+         PRINT NOGEN                                                    02-00042
0005A6 D501 3F9A 3F9E 00FA0 00FA4   993          CLC   WANTED(2),RESIVAL            * COMPARE RESIDS            CTC64300
0005AC 4780 3672            00678   994          BE    PASSMODE                     * RESID FOUND; KEEP INDEX   CTC64400
0005B0 4890 3F96            00F9C   995          LH    R9,INDEX                     * RETRIEVE CURRENT INDEX    CTC64500
0005B4 4690 358A            00590   996          BCT   R9,NEXTIDP                   * REPEAT COMPARISON         CTC64600
                                    997 *********************************************************************** CTC64700
                                    998 *        ACTIV PART                                                   * CTC64800
                                    999 *********************************************************************** CTC64900
                                   1000 *                                                                       CTC65000
0005B8 4890 3F0A            00F10  1001          LH    R9,QUANTITY                  * POTENTIAL SESSIONS QUANT  CTC65100
0005BC 4090 3F96            00F9C  1002 NEXTIDA  STH   R9,INDEX                     * FREE R9 REGISTER          CTC65200
                                   1003          MTBLGET ACTITBL,INDEX,RESIVAL,HALF * GET RESID FROM ACTIV TBL  CTC65300
                                   1004+         PRINT GEN                                                      02-00033
0005C0 48B0 3F96            00F9C  1005+         LH    R11,INDEX                    * PREPARE INDEX REGISTER    01-00179
0005C4 06B0                        1006+         BCTR  R11,R0                       * 1 LESS TO CHANGE ORIGINE  01-00180
0005C6 89B0 0001            00001  1007+         SLL   R11,1                        * MULTIPLY BY 2             01-00190
0005CA 48BB 3F22            00F28  1008+         LH    R11,(ACTITBL)(R11)           * EXTRACT TABLE ELEMENT     01-00191
0005CE 40B0 3F9E            00FA4  1009+         STH   R11,RESIVAL                  * STORE IN WORK STORAGE     01-00192
                                   1010+         PRINT NOGEN                                                    02-00042
0005D2 D501 3F9A 3F9E 00FA0 00FA4  1011          CLC   WANTED(2),RESIVAL            * COMPARE RESIDS            CTC65400
0005D8 4780 3BDE            00BE4  1012          BE    ACTIMODE                     * RESID FOUND; KEEP INDEX   CTC65500
0005DC 4890 3F96            00F9C  1013          LH    R9,INDEX                     * RETRIEVE CURRENT INDEX    CTC65600
0005E0 4690 35B6            005BC  1014          BCT   R9,NEXTIDA                   * REPEAT COMPARISON         CTC65700
                                   1015 *********************************************************************** CTC65800
                                   1016 *        END OF ACTIV PART                                            * CTC65900
                                   1017 *********************************************************************** CTC66000
                                   1018 *                                                                       CTC66100
                                   1019 * NO MORE TO COMPARE ************************************************** CTC66200
0005E4 9500 42CC      012D2        1020          CLI   AREAMCHR+2,X'00'             * LOOK AT COMMAND CODE      CTC66300
0005E8 4770 3600            00606  1021          BNE   CLEAROUT                     * INFO REPORT               CTC66400
0005EC 9180 3F9A      00FA0        1022          TM    WANTED,CNIDSENT              * AVOID LOOP WITH CNID      CTC66500
0005F0 4710 3610            00616  1023          BO    OTHERCMD                     * NOT RETRY WITH CNID       CTC66600
0005F4 D201 3F9A 42CA 00FA0 012D0  1024          MVC   WANTED(2),AREAMCHR           * CHOOSE CNID               CTC66700
0005FA 940F 3F9A      00FA0        1025          NI    WANTED,X'0F'                 * ERASE DIRECTION INDICATOR CTC66800
0005FE 9680 3F9A      00FA0        1026          OI    WANTED,CNIDSENT              * SET INTERNAL CNID FLAG    CTC66900
000602 47F0 3586            0058C  1027          B     SEEKID                       * RETRY WITH CNID           CTC67000
000606 9513 42CC      012D2        1028 CLEAROUT CLI   AREAMCHR+2,X'13'             * LOOK AT COMMAND CODE      CTC67100
00060A 4770 3610            00616  1029          BNE   OTHERCMD                     * NOT CLEAR ON OUTGOING     CTC67200
00060E 9180 3F9A      00FA0        1030          TM    WANTED,CNIDSENT              * AVOID LOOP WITH CNID      CTC67300
000612 4780 3578            0057E  1031          BZ    WANTCNID                     * RETRY WITH CNID           CTC67400
                                   1032 * LOOK FOR OUT OF SESSION COMMAND ************************************* CTC67500
                                   1033 OTHERCMD TESTCMD 0B                         * INCOMING CALL             CTC67600
                                   1034+         PRINT GEN                                                      02-00033
000616 950B 42CC      012D2        1035+OTHERCMD CLI   AREAMCHR+2,X'0B'             * COMPARE DATE COMMAND      01-00296
00061A 4780 3756            0075C  1036+         BE    CMD0B                        * GO TO COMMAND ACTION      01-00297
                                   1037+         PRINT NOGEN                                                    02-00042
                                   1038          TESTCMD F1                         * DIAGNOSTIC                CTC67700
                                   1039+         PRINT GEN                                                      02-00033
00061E 95F1 42CC      012D2        1040+         CLI   AREAMCHR+2,X'F1'             * COMPARE DATE COMMAND      01-00296
000622 4780 393E            00944  1041+         BE    CMDF1                        * GO TO COMMAND ACTION      01-00297
                                   1042+         PRINT NOGEN                                                    02-00042
                                   1043          TESTCMD FB                         * RESTART INDICATION        CTC67800
                                   1044+         PRINT GEN                                                      02-00033
```

```
000626 95FB 42CC      012D2        1045+         CLI   AREAMCHR+2,X'FB'             * COMPARE DATE COMMAND      01-00296
00062A 4780 399E            009A4  1046+         BE    CMDFB                        * GO TO COMMAND ACTION      01-00297
                                   1047+         PRINT NOGEN                                                    02-00042
                                   1048          TESTCMD FF                         * RESTART CONFIRM           CTC67900
                                   1049+         PRINT GEN                                                      02-00033
00062E 95FF 42CC      012D2        1050+         CLI   AREAMCHR+2,X'FF'             * COMPARE DATE COMMAND      01-00296
000632 4780 39F4            009FA  1051+         BE    CMDFF                        * GO TO COMMAND ACTION      01-00297
                                   1052+         PRINT NOGEN                                                    02-00042
                                   1053          WTO   ' CTCP ||| RESID NOT FOUND AND INVALID COMMAND'          CTC68000
000670 45E0 3446            0044C  1061          BAL   R14,FAILEXIT                 * INVALID COMMAND           CTC68100
000674 47F0 3A7E            00A84  1062          B     COLLECTP                     * IF FAILEXIT DOESNOT ABEND CTC68200
                                   1064 * PASSIV MODE : ******************************************************* CTC68400
                                   1065 * 1 IDENTIFY THE COMMAND ********************************************** CTC68500
                                   1066 * 2 PERFORM CORRESPONDING REACTION ************************************ CTC68600
                                   1067 PASSMODE TESTCMD 00                         * INFO REPORT               CTC68700
                                   1068+         PRINT GEN                                                      02-00033
000678 9500 42CC      012D2        1069+PASSMODE CLI   AREAMCHR+2,X'00'             * COMPARE DATE COMMAND      01-00296
00067C 4780 36EE            006F4  1070+         BE    CMD00                        * GO TO COMMAND ACTION      01-00297
                                   1071+         PRINT NOGEN                                                    02-00042
                                   1072          TESTCMD 02                         * QUALIFIED DATA            CTC68800
                                   1073+         PRINT GEN                                                      02-00033
000680 9502 42CC      012D2        1074+         CLI   AREAMCHR+2,X'02'             * COMPARE DATE COMMAND      01-00296
000684 4780 3718            0071E  1075+         BE    CMD02                        * GO TO COMMAND ACTION      01-00297
                                   1076+         PRINT NOGEN                                                    02-00042
                                   1077          TESTCMD 0B                         * INCOMING CALL             CTC68900
                                   1078+         PRINT GEN                                                      02-00033
000688 950B 42CC      012D2        1079+         CLI   AREAMCHR+2,X'0B'             * COMPARE DATE COMMAND      01-00296
00068C 4780 3756            0075C  1080+         BE    CMD0B                        * GO TO COMMAND ACTION      01-00297
                                   1081+         PRINT NOGEN                                                    02-00042
                                   1082          TESTCMD 0F                         * CALL CONNECTED            CTC69000
                                   1083+         PRINT GEN                                                      02-00033
000690 950F 42CC      012D2        1084+         CLI   AREAMCHR+2,X'0F'             * COMPARE DATE COMMAND      01-00296
000694 4780 3836            0083C  1085+         BE    CMD0F                        * GO TO COMMAND ACTION      01-00297
                                   1086+         PRINT NOGEN                                                    02-00042
                                   1087          TESTCMD 13                         * CLEAR INDICATION          CTC69100
                                   1088+         PRINT GEN                                                      02-00033
000698 9513 42CC      012D2        1089+         CLI   AREAMCHR+2,X'13'             * COMPARE DATE COMMAND      01-00296
00069C 4780 3862            00868  1090+         BE    CMD13                        * GO TO COMMAND ACTION      01-00297
                                   1091+         PRINT NOGEN                                                    02-00042
                                   1092          TESTCMD 17                         * CLEAR CONFIRM             CTC69200
                                   1093+         PRINT GEN                                                      02-00033
0006A0 9517 42CC      012D2        1094+         CLI   AREAMCHR+2,X'17'             * COMPARE DATE COMMAND      01-00296
0006A4 4780 3888            0088E  1095+         BE    CMD17                        * GO TO COMMAND ACTION      01-00297
                                   1096+         PRINT NOGEN                                                    02-00042
                                   1097          TESTCMD 1B                         * RESET INDICATION          CTC69300
                                   1098+         PRINT GEN                                                      02-00033
0006A8 951B 42CC      012D2        1099+         CLI   AREAMCHR+2,X'1B'             * COMPARE DATE COMMAND      01-00296
0006AC 4780 38C2            008C8  1100+         BE    CMD1B                        * GO TO COMMAND ACTION      01-00297
                                   1101+         PRINT NOGEN                                                    02-00042
                                   1102          TESTCMD 1F                         * RESET CONFIRM             CTC69400
                                   1103+         PRINT GEN                                                      02-00033
0006B0 951F 42CC      012D2        1104+         CLI   AREAMCHR+2,X'1F'             * COMPARE DATE COMMAND      01-00296
0006B4 4780 38E8            008EE  1105+         BE    CMD1F                        * GO TO COMMAND ACTION      01-00297
                                   1106+         PRINT NOGEN                                                    02-00042
                                   1107          TESTCMD 23                         * INTERRUPT INDICATION      CTC69500
                                   1108+         PRINT GEN                                                      02-00033
0006B8 9523 42CC      012D2        1109+         CLI   AREAMCHR+2,X'23'             * COMPARE DATE COMMAND      01-00296
0006BC 4780 3908            0090E  1110+         BE    CMD23                        * GO TO COMMAND ACTION      01-00297
                                   1111+         PRINT NOGEN                                                    02-00042
                                   1112          TESTCMD 27                         * INTERRUPT CONFIRM         CTC69600
                                   1113+         PRINT GEN                                               02-00033
0006C0 9527 42CC      012D2        1114+         CLI   AREAMCHR+2,X'27'             * COMPARE DATE COMMAND      01-00296
0006C4 4780 391E            00924  1115+         BE    CMD27                        * GO TO COMMAND ACTION      01-00297
                                   1116+         PRINT NOGEN                                             02-00042
                                   1117 * THE LAST 3 HAVE NO RESID AND ARE HANDLED BEFORE ******************** CTC69700
                                   1118 *        TESTCMD F1                         * DIAGNOSTIC                CTC69800
                                   1119 *        TESTCMD FB                         * RESTART INDICATION        CTC69900
                                   1120 *        TESTCMD FF                         * RESTART CONFIRM           CTC70000
                                   1121          WTO   ' CTCP ||| UNKNOWN COMMAND'  * NO MORE COMMAND           CTC70100
0006EC 45E0 3446            0044C  1129          BAL   R14,FAILEXIT                 * UNKNOWN COMMAND           CTC70200
0006F0 47F0 3A7E            00A84  1130          B     COLLECTP                     * IF FAILEXIT DOESNOT ABEND CTC70300
                                   1131 * ACHIEVE APPROPRIATE COMMAND REACTION ******************************** CTC70400
                            006F4  1132 CMD00    EQU   *                            * INFO REPORT               CTC70500
0006F4 9507 42CE      012D4        1133          CLI   AREAMCHR+4,X'07'             * ERROR CAUSE 7             CTC70600
0006F8 4770 3A7E            00A84  1134          BNE   COLLECTP                     * NO ACTION TO PERFORM      CTC70700
                                   1135          SETSEND CLEAREQ                    * ANSWER TO CAUSE 7         CTC70800
                                   1136+         PRINT GEN                                               02-00033
0006FC 4170 4050            01056  1137+         LA    R7,CLEAREQ                   * GET ADRESS OF DATA        01-00262
000700 4187 0001            00001  1138+         LA    R8,1(R7)                     * GET ADRESS OF STRING      01-00266
000704 1799                        1139+         XR    R9,R9                        * NULLIFY R9                01-00267
000706 4397 0000            00000  1140+         IC    R9,0(R7)                     * GET LENGTH OF STRING      01-00268
00070A 45E0 3B06            00B0C  1141+         BAL   R14,SBRSEND                  * CALL SEND PROCEDURE       01-00269
                                   1142+         PRINT NOGEN                                             02-00042
00070E 4870 3F96            00F9C  1143          LH    R7,INDEX                     * BYTE NUMBER               CTC70900
000712 4177 3F1C            00F22  1144          LA    R7,FLAGTBL(R7)               * BYTE ADRESS               CTC71000
000716 9620 7000      00000        1145          OI    0(R7),CLEASENT               * SET THE CLEASENT FLAG     CTC71100
00071A 47F0 3A7E            00A84  1146          B     COLLECTP                                                 CTC71200
                            0071E  1147 CMD02    EQU   *                            * QUALIFIED DATA            CTC71300
                                   1148          WTO   ' CTCP ||| PASSIV MODE: QUALIFIED DATA COMMAND'          CTC71400
000758 47F0 3A7E            00A84  1156          B     COLLECTP                                                 CTC71500
                            0075C  1157 CMD0B    EQU   *                            * INCOMING CALL             CTC71600
                                   1158 * PREPARE SESSION CONTINUATION                                          CTC71700
                                   1159 * LOOK FOR AVAILABLE ROW IN PASSIV TABLE                                CTC71800
00075C 4890 3F0A            00F10  1160          LH    R9,QUANTITY                  * POTENTIAL SESSIONS QUANT  CTC71900
000760 4090 3F96            00F9C  1161 NEXTPASS STH   R9,INDEX                     * FREE R9 REGISTER          CTC72000
000764 4870 3F96            00F9C  1162          LH    R7,INDEX                     * BYTE NUMBER               CTC72100
000768 4177 3F1C            00F22  1163          LA    R7,FLAGTBL(R7)               * BYTE ADRESS               CTC72200
00076C 9110 7000      00000        1164          TM    0(R7),AVAILABL               * TEST THE AVAILABL FLAG    CTC72300
000770 4770 37AA            007B0  1165          BNZ   FOUNDPAS                     * FOUND PASSIV AVAILABLE    CTC72400
000774 4890 3F96            00F9C  1166          LH    R9,INDEX                     * RETRIEVE CURRENT INDEX    CTC72500
000778 4690 375A            00760  1167          BCT   R9,NEXTPASS                  * REPEAT COMPARISON         CTC72600
                                   1168          WTO   ' CTCP ||| NO PASSIV ROW AVAILABLE '                     CTC72700
0007A8 45E0 3446            0044C  1176          BAL   R14,FAILEXIT                                             CTC72800
0007AC 47F0 3A7E            00A84  1177          B     COLLECTP                     * IF FAILEXIT DOESNOT ABEND CTC72900
                                   1178 * GET SESSION RESID                                                     CTC73000
0007B0 9710 7000      00000        1179 FOUNDPAS XI    0(R7),AVAILABL               * TOGGLE AVAILABLE BIT      CTC73100
0007B4 4870 42CA            012D0  1180          LH    R7,AREAMCHR                  * GET RESIVAL FROM COMMAND  CTC73200
0007B8 4070 3F9E            00FA4  1181          STH   R7,RESIVAL                   * SET RESIVAL FOR SBRSEND   CTC73300
0007BC 940F 3F9E      00FA4        1182          NI    RESIVAL,X'0F'                * ERASE DIRECTION INDICATOR CTC73400
                                   1183          MTBLPUT PASSTBL,INDEX,RESIVAL,HALF * SAVE NEW SESSION'S RESID  CTC73500
                                   1184+         PRINT GEN                                               02-00033
0007C0 48B0 3F96            00F9C  1185+         LH    R11,INDEX                    * PREPARE INDEX REGISTER    01-00207
0007C4 06B0                        1186+         BCTR  R11,R0                       * 1 LESS TO CHANGE ORIGINE  01-00208
0007C6 89B0 0001            00001  1187+         SLL   R11,1                        * MULTIPLY BY 2             01-00218
0007CA 48A0 3F9E            00FA4  1188+         LH    R10,RESIVAL                  * PREPARE NEW VALUE         01-00219
0007CE 40AB 3F14            00F1A  1189+         STH   R10,(PASSTBL)(R11)           * STORE IN TABLE            01-00220
                                   1190+         PRINT NOGEN                                             02-00042
                                   1191 * DETERMINE CUD BYTE 0 ADRESS ***************************************** CTC73600
0007D2 4370 42CD            012D3  1192          IC    R7,AREAMCHR+3                * GET PACKET BYTE 4         CTC73700
0007D6 1887                        1193          LR    R8,R7                        * COPY                      CTC73800
```

```
0007D8 5470 3F92            00F98  1194          N     R7,FOURBITS                 * ISOLATE BITS 0-3         CTC73900
0007DC 8880 0004            00004  1195          SRL   R8,4                        * ISOLATE BITS 4-7         CTC74000
0007E0 5480 3F92            00F98  1196          N     R8,FOURBITS                 * ISOLATE BITS 4-7         CTC74100
0007E4 4177 8001            00001  1197          LA    R7,1(R7,R8)                 * ROUND SUM TO GREATER EVEN CTC74200
0007E8 8870 0001            00001  1198          SRL   R7,1                        * DIVIDE BY TWO            CTC74300
0007EC 4177 42CE            012D4  1199          LA    R7,AREAMCHR+4(R7)           * ADRESS OF FACILITY LENGTH CTC74400
0007F0 1788                        1200          XR    R8,R8                       * ERASE R8                 CTC74500
0007F2 4387 0000            00000  1201          IC    R8,0(R7)                    * READ FACILITY LENGTH     CTC74600
0007F6 4177 8001            00001  1202          LA    R7,1(R7,R8)                 * GET CUD ADRESS           CTC74700
                                   1203 * RECOGNIZE LLC ******************************************************* CTC74800
0007FA 95C0 7000      00000        1204          CLI   0(R7),LLC0                  * COMP CUD BYTE 0          CTC74900
0007FE 4780 381A            00820  1205          BE    ACCEPT                      * GO TO ANSWER             CTC75000
000802 9501 7000      00000        1206          CLI   0(R7),LLC5                  * COMP CUD BYTE 0          CTC75100
000806 4780 381A            00820  1207          BE    ACCEPT                      * GO TO ANSWER             CTC75200
                                   1208          SETSEND CLEAREQ                   * REFUSE INCOMING CALL     CTC75300
                                   1209+         PRINT GEN                                                    02-00033
00080A 4170 4050            01056  1210+         LA    R7,CLEAREQ                  * GET ADRESS OF DATA       01-00262
00080E 4187 0001            00001  1211+         LA    R8,1(R7)                    * GET ADRESS OF STRING     01-00266
000812 1799                        1212+         XR    R9,R9                       * NULLIFY R9               01-00267
000814 4397 0000            00000  1213+         IC    R9,0(R7)                    * GET LENGTH OF STRING     01-00268
000818 45E0 3B06            00B0C  1214+         BAL   R14,SBRSEND                 * CALL SEND PROCEDURE      01-00269
                                   1215+         PRINT NOGEN                                                  02-00042
00081C 47F0 3A7E            00A84  1216          B     COLLECTP                                               CTC75400
                                   1217 * SEND CORRESPONDING CALL ACCEPTED ************************************ CTC75500
000820 D200 4042 7000 01048 00000  1218 ACCEPT   MVC   CALLACC+7(1),0(R7)          * COPY LLC TYPE            CTC75600
                                   1219          SETSEND CALLACC                   * ACCEPT INCOMING CALL     CTC75700
                                   1220+         PRINT GEN                                                    02-00033
000826 4170 403B            01041  1221+         LA    R7,CALLACC                  * GET ADRESS OF DATA       01-00262
00082A 4187 0001            00001  1222+         LA    R8,1(R7)                    * GET ADRESS OF STRING     01-00266
00082E 1799                        1223+         XR    R9,R9                       * NULLIFY R9               01-00267
000830 4397 0000            00000  1224+         IC    R9,0(R7)                    * GET LENGTH OF STRING     01-00268
000834 45E0 3B06            00B0C  1225+         BAL   R14,SBRSEND                 * CALL SEND PROCEDURE      01-00269
                                   1226+         PRINT NOGEN                                                  02-00042
000838 47F0 3A7E            00A84  1227          B     COLLECTP                                               CTC75800
                            0083C  1228 CMD0F    EQU   *                           * CALL CONNECTED           CTC75900
                                   1229          WTO   ' INTERNAL LOGIC ERROR IN CTCP '                       CTC76000
                                   1237 * CNID AND RESID CANNOT BE CONFUSED; EVEN THOUGH,                     CTC76100
                                   1238 * CNID HAS BEEN CONFUSED WITH RESID OF A PASSIV SPEAKER IN SESSION    CTC76200
000864 47F0 3A7E            00A84  1239          B     COLLECTP                                               CTC76300
                            00868  1240 CMD13    EQU   *                           * CLEAR INDICATION         CTC76400
000868 4870 3F96            00F9C  1241          LH    R7,INDEX                    * BYTE NUMBER              CTC76500
00086C 4177 3F1C            00F22  1242          LA    R7,FLAGTBL(R7)              * BYTE ADRESS              CTC76600
000870 9120 7000      00000        1243          TM    0(R7),CLEASENT              * TEST THE CLEAR SENT BIT  CTC76700
000874 4770 38A0            008A6  1244          BNZ   RELEASE                     * DO NOT CONFIRM           CTC76800
                                   1245          SETSEND CLEACNF                   * ANSWER CLEAR CONFIRM     CTC76900
                                   1246+         PRINT GEN                                                    02-00033
000878 4170 4056            0105C  1247+         LA    R7,CLEACNF                  * GET ADRESS OF DATA       01-00262
00087C 4187 0001            00001  1248+         LA    R8,1(R7)                    * GET ADRESS OF STRING     01-00266
000880 1799                        1249+         XR    R9,R9                       * NULLIFY R9               01-00267
000882 4397 0000            00000  1250+         IC    R9,0(R7)                    * GET LENGTH OF STRING     01-00268
000886 45E0 3B06            00B0C  1251+         BAL   R14,SBRSEND                 * CALL SEND PROCEDURE      01-00269
                                   1252+         PRINT NOGEN                                                  02-00042
00088A 47F0 38A0            008A6  1253          B     RELEASE                     * MAKE ROW AVAILABLE       CTC77000
                            0088E  1254 CMD17    EQU   *                           * CLEAR CONFIRMATION       CTC77100
00088E 4870 3F96            00F9C  1255          LH    R7,INDEX                    * BYTE NUMBER              CTC77200
000892 4177 3F1C            00F22  1256          LA    R7,FLAGTBL(R7)              * BYTE ADRESS              CTC77300
000896 9120 7000      00000        1257          TM    0(R7),CLEASENT              * TEST THE CLEAR SENT BIT  CTC77400
00089A 4770 38A0            008A6  1258          BNZ   RELEASE                     * DO NOT CONFIRM           CTC77500
00089E 45E0 3446            0044C  1259          BAL   R14,FAILEXIT                * DUMP                     CTC77600
0008A2 47F0 3A7E            00A84  1260          B     COLLECTP                    * IF FAILEXIT DOESNOT ABEND CTC77700
```

```
0008A6 4870 3F96            00F9C  1261 RELEASE  LH    R7,INDEX                     * BYTE NUMBER               CTC77800
0008AA 4177 3F1C            00F22  1262          LA    R7,FLAGTBL(R7)               * BYTE ADRESS               CTC77900
0008AE 9210 7000      00000        1263          MVI   0(R7),AVAILABL               * INITIALISE ALL BITS       CTC78000
                                   1264          MTBLPUT PASSTBL,INDEX,=X'F000',HALF * ERASE PASSIV RESID       CTC78100
                                   1265+         PRINT GEN                                                      02-00033
0008B2 48B0 3F96            00F9C  1266+         LH    R11,INDEX                    * PREPARE INDEX REGISTER    01-00207
0008B6 06B0                        1267+         BCTR  R11,R0                       * 1 LESS TO CHANGE ORIGINE  01-00208
0008B8 89B0 0001            00001  1268+         SLL   R11,1                        * MULTIPLY BY 2             01-00218
0008BC 48A0 3F80            00F86  1269+         LH    R10,=X'F000'                 * PREPARE NEW VALUE         01-00219
0008C0 40AB 3F14            00F1A  1270+         STH   R10,(PASSTBL)(R11)           * STORE IN TABLE            01-00220
                                   1271+         PRINT NOGEN                                                    02-00042
0008C4 47F0 3A7E            00A84  1272          B     COLLECTP                                                 CTC78200
                            008C8  1273 CMD1B    EQU   *                            * RESET INDICATION          CTC78300
0008C8 4870 3F96            00F9C  1274          LH    R7,INDEX                     * BYTE NUMBER               CTC78400
0008CC 4177 3F1C            00F22  1275          LA    R7,FLAGTBL(R7)               * BYTE ADRESS               CTC78500
0008D0 9140 7000      00000        1276          TM    0(R7),RSETSENT               * TEST THE RESET SENT FLAG  CTC78600
0008D4 4770 3900            00906  1277          BNZ   ERASERSE                     * DO NOT CONFIRM            CTC78700
                                   1278          SETSEND RSETCNF                    * ANSWER RESET CONFIRM      CTC78800
                                   1279+         PRINT GEN                                                      02-00033
0008D8 4170 4060            01066  1280+         LA    R7,RSETCNF                   * GET ADRESS OF DATA        01-00262
0008DC 4187 0001            00001  1281+         LA    R8,1(R7)                     * GET ADRESS OF STRING      01-00266
0008E0 1799                        1282+         XR    R9,R9                        * NULLIFY R9                01-00267
0008E2 4397 0000            00000  1283+         IC    R9,0(R7)                     * GET LENGTH OF STRING      01-00268
0008E6 45E0 3B06            00B0C  1284+         BAL   R14,SBRSEND                  * CALL SEND PROCEDURE       01-00269
                                   1285+         PRINT NOGEN                                                    02-00042
0008EA 47F0 3A7E            00A84  1286          B     COLLECTP                                                 CTC78900
                            008EE  1287 CMD1F    EQU   *                            * RESET CONFIRMATION        CTC79000
0008EE 4870 3F96            00F9C  1288          LH    R7,INDEX                     * BYTE NUMBER               CTC79100
0008F2 4177 3F1C            00F22  1289          LA    R7,FLAGTBL(R7)               * BYTE ADRESS               CTC79200
0008F6 9140 7000      00000        1290          TM    0(R7),RSETSENT               * TEST THE RESET SENT FLAG  CTC79300
0008FA 4770 3900            00906  1291          BNZ   ERASERSE                     * DO NOT CONFIRM            CTC79400
0008FE 45E0 3446            0044C  1292          BAL   R14,FAILEXIT                 * DUMP                      CTC79500
000902 47F0 3A7E            00A84  1293          B     COLLECTP                     * IF FAILEXIT DOESNOT ABEND CTC79600
000906 9740 7000      00000        1294 ERASERSE XI    0(R7),RSETSENT               * TOGGLE RESET SENT BIT     CTC79700
00090A 47F0 3A7E            00A84  1295          B     COLLECTP                                                 CTC79800
                            0090E  1296 CMD23    EQU   *                            * INTERRUPT                 CTC79900
                                   1297          SETSEND INTECNF                    * ANSWER INTERRUPT CONFIRM  CTC80000
                                   1298+         PRINT GEN                                                      02-00033
00090E 4170 4069            0106F  1299+         LA    R7,INTECNF                   * GET ADRESS OF DATA        01-00262
000912 4187 0001            00001  1300+         LA    R8,1(R7)                     * GET ADRESS OF STRING      01-00266
000916 1799                        1301+         XR    R9,R9                        * NULLIFY R9                01-00267
000918 4397 0000            00000  1302+         IC    R9,0(R7)                     * GET LENGTH OF STRING      01-00268
00091C 45E0 3B06            00B0C  1303+         BAL   R14,SBRSEND                  * CALL SEND PROCEDURE       01-00269
                                   1304+         PRINT NOGEN                                                    02-00042
000920 47F0 3A7E            00A84  1305          B     COLLECTP                                                 CTC80100
                            00924  1306 CMD27    EQU   *                            * INTERRUPT CONFIRMATION    CTC80200
000924 4870 3F96            00F9C  1307          LH    R7,INDEX                     * BYTE NUMBER               CTC80300
000928 4177 3F1C            00F22  1308          LA    R7,FLAGTBL(R7)               * BYTE ADRESS               CTC80400
00092C 9108 7000      00000        1309          TM    0(R7),INTRSENT               * TEST THE INTER SENT FLAG  CTC80500
000930 4770 3936            0093C  1310          BNZ   ERASEINT                     * DO NOT CONFIRM            CTC80600
000934 45E0 3446            0044C  1311          BAL   R14,FAILEXIT                 * DUMP                      CTC80700
000938 47F0 3A7E            00A84  1312          B     COLLECTP                     * IF FAILEXIT DOESNOT ABEND CTC80800
00093C 9708 7000      00000        1313 ERASEINT XI    0(R7),INTRSENT               * TOGGLE INTER SENT BIT     CTC80900
000940 47F0 3A7E            00A84  1314          B     COLLECTP                                                 CTC81000
                            00944  1315 CMDF1    EQU   *                            * DIAGNOSTIC                CTC81100
000944 D203 3502 42CD 00508 012D3  1316          MVC   WORKN(4),AREAMCHR+3          * PREPARE UNPACK            CTC81200
00094A F384 3506 3502 0050C 00508  1317          UNPK  WORKC(9),WORKN(5)            * UNPACK HALFBYTE TO EBCDIC CTC81300
000950 DC07 3506 341F 0050C 00425  1318          TR    WORKC(8),HEX                 * TRANSLATE EBCDIC TO HEXA  CTC81400
000956 D201 398A 3506 00990 0050C  1319          MVC   WTODIAG+44(2),WORKC          * PLACE RESULT IN WTO MSG   CTC81500
00095C D205 398D 3508 00993 0050E  1320          MVC   WTODIAG+47(6),WORKC+2        * PLACE RESULT IN WTO MSG   CTC81600
```

```
                                   1321 WTODIAG  WTO   ' CTCP ||| PASSIV MODE: DIAGNOSTIC : __.______ '         CTC81700
00099C 45E0 3BCE            00BD4  1329          BAL   R14,SBRNULL                  * VERIFY RESID=0            CTC81800
0009A0 47F0 3A7E            00A84  1330          B     COLLECTP                                                 CTC81900
                            009A4  1331 CMDFB    EQU   *                            * RESTART INDICATION        CTC82000
                                   1332          WTO   ' CTCP ||| PASSIV MODE: RESTART INDICATION'              CTC82100
0009D8 45E0 3BCE            00BD4  1340          BAL   R14,SBRNULL                  * VERIFY RESID=0            CTC82200
0009DC 95FF 3F8C      00F92        1341          CLI   RSARBOOL,TRUE                * TEST RESTART SENT BOOL    CTC82300
0009E0 4780 3A40            00A46  1342          BE    ERASERSA                     * NO RESTART CONFIRM        CTC82400
                                   1343          SETSEND RSARCNF                    * ANSWER RESTART CONFIRM    CTC82500
                                   1344+         PRINT GEN                                                      02-00033
0009E4 4170 4079            0107F  1345+         LA    R7,RSARCNF                   * GET ADRESS OF DATA        01-00262
0009E8 4187 0001            00001  1346+         LA    R8,1(R7)                     * GET ADRESS OF STRING      01-00266
0009EC 1799                        1347+         XR    R9,R9                        * NULLIFY R9                01-00267
0009EE 4397 0000            00000  1348+         IC    R9,0(R7)                     * GET LENGTH OF STRING      01-00268
0009F2 45E0 3B06            00B0C  1349+         BAL   R14,SBRSEND                  * CALL SEND PROCEDURE       01-00269
                                   1350+         PRINT NOGEN                                                    02-00042
0009F6 47F0 3A4C            00A52  1351          B     ERASEALL                     * PERFORM RESTART           CTC82600
                            009FA  1352 CMDFF    EQU   *                            * RESTART CONFIRMATION      CTC82700
                                   1353          WTO   ' CTCP ||| PASSIV MODE: RESTART CONFIRMATION'            CTC82800
000A32 45E0 3BCE            00BD4  1361          BAL   R14,SBRNULL                  * VERIFY RESID=0            CTC82900
000A36 95FF 3F8C      00F92        1362          CLI   RSARBOOL,TRUE                * TEST RESTART SENT BOOL    CTC83000
000A3A 4780 3A40            00A46  1363          BE    ERASERSA                     * NO RESTART CONFIRM        CTC83100
000A3E 45E0 3446            0044C  1364          BAL   R14,FAILEXIT                 * DUMP                      CTC83200
000A42 47F0 3A7E            00A84  1365          B     COLLECTP                     * IF FAILEXIT DOESNOT ABEND CTC83300
000A46 9200 3F8C      00F92        1366 ERASERSA MVI   RSARBOOL,FALSE               * RESET  RESTART BOOLEAN    CTC83400
                                   1367          POST  RSARECB                      * UNLOCK RESTART POINT      CTC83500
000A52 4890 3F0A            00F10  1371 ERASEALL LH    R9,QUANTITY                  * INITIALISE ERASE LOOP     CTC83600
000A56 4090 3F96            00F9C  1372 ERALOOP  STH   R9,INDEX                     * FREE R9 REGISTER          CTC83700
000A5A 4870 3F96            00F9C  1373          LH    R7,INDEX                     * BYTE NUMBER               CTC83800
000A5E 4177 3F1C            00F22  1374          LA    R7,FLAGTBL(R7)               * BYTE ADRESS               CTC83900
000A62 9210 7000      00000        1375          MVI   0(R7),AVAILABL               * INITIALISE ALL BITS       CTC84000
                                   1376          MTBLPUT PASSTBL,INDEX,=X'F000',HALF * ERASE PASSIV RESID       CTC84100
                                   1377+         PRINT GEN                                                      02-00033
000A66 48B0 3F96            00F9C  1378+         LH    R11,INDEX                    * PREPARE INDEX REGISTER    01-00207
000A6A 06B0                        1379+         BCTR  R11,R0                       * 1 LESS TO CHANGE ORIGINE  01-00208
000A6C 89B0 0001            00001  1380+         SLL   R11,1                        * MULTIPLY BY 2             01-00218
000A70 48A0 3F80            00F86  1381+         LH    R10,=X'F000'                 * PREPARE NEW VALUE         01-00219
000A74 40AB 3F14            00F1A  1382+         STH   R10,(PASSTBL)(R11)           * STORE IN TABLE            01-00220
                                   1383+         PRINT NOGEN                                                    02-00042
000A78 4890 3F96            00F9C  1384          LH    R9,INDEX                     * RETRIEVE CURRENT INDEX    CTC84200
000A7C 4690 3A50            00A56  1385          BCT   R9,ERALOOP                   * REPEAT COMPARISON         CTC84300
000A80 47F0 3A7E            00A84  1386          B     COLLECTP                                                 CTC84400
                                   1387 * SET PENDING RECEIVE MCH EXIT ROUTINE ******************************** CTC84500
000A84 D200 42CA 3F8B 012D0 00F91  1388 COLLECTP MVC   AREAMCHR(1),=C' '            * ERASE AREAMCHR            CTC84600
000A8A D2FE 42CB 42CA 012D1 012D0  1389          MVC   AREAMCHR+1(255),AREAMCHR     * ERASE AREAMCHR            CTC84700
000A90 D203 43CA 43C9 013D0 013CF  1390          MVC   AREAMCHR+256(4),AREAMCHR+255 * ERASE AREAMCHR            CTC84800
                                   1391          RECEIVE RPL=RPLMCH,AREA=AREAMCHR,RECLEN=260,                  CCTC84900
                                                       OPTCD=(ASY,SPEC),RTYPE=(DFSYN,DFASY,NRESP),EXIT=RCVMCH   CTC85000
                                   1439          MVERIFY                            * VERIFY RECEIVE COMPLETION CTC85100
                                   1440+         PRINT GEN                                                      02-00033
000AFC 12FF                        1441+         LTR   R15,R15                      * TEST RETURN CODE          01-00282
000AFE 4780 3B00            00B06  1442+         BZ    MV0010                       * NO PROBLEM                01-00283
000B02 45E0 3446            0044C  1443+         BAL   R14,FAILEXIT                 * DUMP                      01-00284
                            00B06  1444+MV0010   EQU   *                            * RESUME STATEMENTS         01-00285
                                   1445+         PRINT NOGEN                                                    02-00042
                                   1446 EXRCVMCH MSBROUT                            * RETURN TO MVS             CTC85200
                                   1447+         PRINT GEN                                                      02-00033
000B06 58E0 338A            00390  1448+EXRCVMCH L     R14,R14SAVE                  * RESTORE RETURN ADRESS     01-00139
000B0A 07FE                        1449+         BR    R14                          * RETURN TO VTAM            01-00140
                                   1450+         PRINT NOGEN                                                    02-00042
                                   1451 * SEND PROCEDURE ****************************************************** CTC85300
                                   1452 * INPUT : R8, R9, R14.                                                  CTC85400
                                   1453 * OUTPUT: NONE.                                                         CTC85500
000B0C 50E0 3BCA            00BD0  1454 SBRSEND  ST    R14,R14SAVE3                 * KEEP RETURN ADRESS        CTC85600
000B10 D701 8000 8000 00000 00000  1455          XC    0(2,R8),0(R8)                * NULLIFY RESID             CTC85700
000B16 95F1 8002      00002        1456          CLI   2(R8),X'F1'                  * DIAGNOSTIC                CTC85800
000B1A 4780 3B2E            00B34  1457          BE    SENDOK                                                   CTC85900
000B1E 95FB 8002      00002        1458          CLI   2(R8),X'FB'                  * RESTART                   CTC86000
000B22 4780 3B2E            00B34  1459          BE    SENDOK                                                   CTC86100
000B26 95FF 8002      00002        1460          CLI   2(R8),X'FF'                  * RESTART CONFIRM           CTC86200
000B2A 4780 3B2E            00B34  1461          BE    SENDOK                                                   CTC86300
000B2E D201 8000 3F9E 00000 00FA4  1462          MVC   0(2,R8),RESIVAL              * PUT RESID IN SENT AREA    CTC86400
                                   1463 SENDOK   SEND  RPL=RPLMCHS,AREA=(R8),RECLEN=(R9),OPTCD=SYN                CTC86500
                                   1482 * IF RETRIABLE SEND AGAIN ********************************************* CTC86600
000B5A 4870 3F8E            00F94  1483          LH    R7,MAXRETRY                  * TOTAL RETRY               CTC86700
000B5E 4070 3F98            00F9E  1484 RETRLOOP STH   R7,COUNT                     * FREE R7 REGISTER          CTC86800
000B62 49F0 3F82            00F88  1485          CH    R15,=H'4'                                                CTC86900
000B66 4770 3BBE            00BC4  1486          BNE   NORETRY                      * SEND IS ACHIEVED          CTC87000
000B6A 4900 3F84            00F8A  1487          CH    R0,=H'8'                                                 CTC87100
000B6E 4770 3BBE            00BC4  1488          BNE   NORETRY                      * SEND IS ACHIEVED          CTC87200
                                   1489          EXECRPL RPL=RPLMCH                 * RETRY SEND                CTC87300
000B86 4870 3F98            00F9E  1502          LH    R7,COUNT                     * RETRIEVE CURRENT COUNT    CTC87400
000B8A 4670 3B58            00B5E  1503          BCT   R7,RETRLOOP                  * REPEAT COMPARISON         CTC87500
                                   1504          WTO   ' CTCP ||| SEND RETRIED UNSUCCESSFULL '                  CTC87600
000BC0 45E0 3446            0044C  1512          BAL   R14,FAILEXIT                                             CTC87700
000BC4 58E0 3BCA            00BD0  1513 NORETRY  L     R14,R14SAVE3                 * RESTORE RETURN ADRESS     CTC87800
000BC8 12FF                        1514          LTR   R15,R15                      * VERIFY SEND COMPLETION    CTC87900
000BCA 4770 3446            0044C  1515          BNZ   FAILEXIT                     * GO DIRECTLY TO FAILEXIT   CTC88000
                                   1516 *                                           * DO NOT USE MVERIFY |      CTC88100
000BCE 07FE                        1517          BR    R14                          * RETURN TO CALLER          CTC88200
000BD0                             1518 R14SAVE3 DS    F                            * RETURN ADRESS STORAGE     CTC88300
                                   1519 * NULL RESID PROCEDURE ************************************************ CTC88400
000BD4 4870 42CA            012D0  1520 SBRNULL  LH    R7,AREAMCHR                  * LOAD RESID FIELD          CTC88500
000BD8 5470 3F6E            00F74  1521          N     R7,=X'00000FFF'              * ERASE DIRECTION INDICATOR CTC88600
000BDC 1277                        1522          LTR   R7,R7                        * TEST RESID                CTC88700
000BDE 4770 3446            0044C  1523          BNZ   FAILEXIT                     * NON ZERO RESID            CTC88800
000BE2 07FE                        1524          BR    R14                          * RESID=0                   CTC88900
                                   1526 *********************************************************************** CTC89100
                                   1527 *        ACTIV PART OF RECEIVE EXIT                                   * CTC89200
                                   1528 *********************************************************************** CTC89300
                                   1529 *                                                                       CTC89400
                                   1530 * DETERMINE CURRENT STEP ********************************************** CTC89500
                                   1531 ACTIMODE MTBLGET PLAYTBL,INDEX,PLAYVAL,FULL * GET SCRIPT BASE ADRESS    CTC89600
                                   1532+         PRINT GEN                                                      02-00033
000BE4 48B0 3F96            00F9C  1533+ACTIMODE LH    R11,INDEX                    * PREPARE INDEX REGISTER    01-00179
000BE8 06B0                        1534+         BCTR  R11,R0                       * 1 LESS TO CHANGE ORIGINE  01-00180
000BEA 89B0 0002            00002  1535+         SLL   R11,2                        * MULTIPLY BY 4             01-00185
000BEE 58BB 3ECE            00ED4  1536+         L     R11,(PLAYTBL)(R11)           * EXTRACT TABLE ELEMENT     01-00186
000BF2 50B0 3EA2            00EA8  1537+         ST    R11,PLAYVAL                  * STORE IN WORK STORAGE     01-00187
                                   1538+         PRINT NOGEN                                                    02-00042
                                   1539          MTBLGET STEPTBL,INDEX,STEPVAL,HALF * GET CURRENT SCRIPT INDEX  CTC89700
                                   1540+         PRINT GEN                                                      02-00033
000BF6 48B0 3F96            00F9C  1541+         LH    R11,INDEX                    * PREPARE INDEX REGISTER    01-00179
000BFA 06B0                        1542+         BCTR  R11,R0                       * 1 LESS TO CHANGE ORIGINE  01-00180
000BFC 89B0 0001            00001  1543+         SLL   R11,1                        * MULTIPLY BY 2             01-00190
000C00 48BB 3F0C            00F12  1544+         LH    R11,(STEPTBL)(R11)           * EXTRACT TABLE ELEMENT     01-00191
000C04 40B0 3EC6            00ECC  1545+         STH   R11,STEPVAL                  * STORE IN WORK STORAGE     01-00192
                                   1546+         PRINT NOGEN                                                    02-00042
000C08 41E0 3C02            00C08  1547          LA    R14,*                        * USEFUL IF PRELAST CALLED  CTC89800
                                   1548          DOACTION                           * BRANCH IF NO DATA POINTER CTC89900
                                   1549+         PRINT GEN                                                      02-00033
000C0C 58A0 3EA2            00EA8  1550+         L     R10,PLAYVAL                  * SCRIPT BASE POINTER       01-00231
000C10 48B0 3EC6            00ECC  1551+         LH    R11,STEPVAL                  * STEP OF SCRIPT            01-00232
000C14 06B0                        1552+         BCTR  R11,R0                       * ORIGIN=0                  01-00233
000C16 89B0 0002            00002  1553+         SLL   R11,2                        * FULLWORD DEPLACMENT       01-00234
000C1A 587A B000            00000  1554+         L     R7,0(R10,R11)                * GET SCRIPT ACTION         01-00235
000C1E 5070 3ECA            00ED0  1555+         ST    R7,ACTIVAL                   * FOR FURTHER USE           01-00236
000C22 5570 3F72            00F78  1556+         CL    R7,=A(BEGINCMD)              * CMP WITH COMMAND BORDER   01-00237
000C26 0747                        1557+         BLR   R7                           * SPECIFIC ACTION           01-00238
000C28 5570 3F76            00F7C  1558+         CL    R7,=A(ENDCMD)                * CMP WITH COMMAND BORDER   01-00239
000C2C 07B7                        1559+         BNLR  R7                           * SPECIFIC ACTION           01-00240
                                   1560+         PRINT NOGEN                                                    02-00042
                                   1561 * COMPARE RECEIVED DATA IN AREAMCHR WITH DATA EXPECTED IN (R7) ******** CTC90000
000C2E 5870 3ECA            00ED0  1562 LOCKCMD  L     R7,ACTIVAL                   * GET CURRENT ACTION        CTC90100
000C32 D500 42CC 7003 012D2 00003  1563          CLC   AREAMCHR+2(1),3(R7)          * CMP COMMAND PART          CTC90200
000C38 4780 3C3A            00C40  1564          BE    NEXTCMD                      * ADEQUATE COMMAND RECEIVED CTC90300
000C3C 45E0 3446            0044C  1565          BAL   R14,FAILEXIT                 * UNEXPECTED COMMAND        CTC90400
                                   1566 * GET NEXT STEP ACCORDING TO THE PLAY ********************************* CTC90500
                                   1567 NEXTCMD  MINC  STEPVAL,1,HALF               * INCREMENT SCRIPT INDEX    CTC90600
                                   1568+         PRINT GEN                                                      02-00033
000C40 4870 3EC6            00ECC  1569+NEXTCMD  LH    R7,STEPVAL                   * GET TARGET TO INCREMENT   01-00162
000C44 4177 0001            00001  1570+         LA    R7,1(R7)                     * INCREMENT                 01-00163
000C48 4070 3EC6            00ECC  1571+         STH   R7,STEPVAL                   * STORE NEW TARGET VALUE    01-00164
                                   1572+         PRINT NOGEN                                                    02-00042
000C4C 41E0 3C46            00C4C  1573          LA    R14,*                        * USEFUL IF PRELAST CALLED  CTC90700
                                   1574          DOACTION                           * BRANCH IF NO DATA POINTER CTC90800
                                   1575+         PRINT GEN                                                      02-00033
000C50 58A0 3EA2            00EA8  1576+         L     R10,PLAYVAL                  * SCRIPT BASE POINTER       01-00231
000C54 48B0 3EC6            00ECC  1577+         LH    R11,STEPVAL                  * STEP OF SCRIPT            01-00232
000C58 06B0                        1578+         BCTR  R11,R0                       * ORIGIN=0                  01-00233
000C5A 89B0 0002            00002  1579+         SLL   R11,2                        * FULLWORD DEPLACMENT       01-00234
000C5E 587A B000            00000  1580+         L     R7,0(R10,R11)                * GET SCRIPT ACTION         01-00235
000C62 5070 3ECA            00ED0  1581+         ST    R7,ACTIVAL                   * FOR FURTHER USE           01-00236
000C66 5570 3F72            00F78  1582+         CL    R7,=A(BEGINCMD)              * CMP WITH COMMAND BORDER   01-00237
000C6A 0747                        1583+         BLR   R7                           * SPECIFIC ACTION           01-00238
000C6C 5570 3F76            00F7C  1584+         CL    R7,=A(ENDCMD)                * CMP WITH COMMAND BORDER   01-00239
000C70 07B7                        1585+         BNLR  R7                           * SPECIFIC ACTION           01-00240
                                   1586+         PRINT NOGEN                                                    02-00042
                                   1587 * SEND ACTION DATA **************************************************** CTC90900
                                   1588          SETSEND                                                        CTC91000
                                   1589+         PRINT GEN                                                      02-00033
000C72 5870 3ECA            00ED0  1590+         L     R7,ACTIVAL                   * GET CURRENT ACTION        01-00265
000C76 4187 0001            00001  1591+         LA    R8,1(R7)                     * GET ADRESS OF STRING      01-00266
000C7A 1799                        1592+         XR    R9,R9                        * NULLIFY R9                01-00267
000C7C 4397 0000            00000  1593+         IC    R9,0(R7)                     * GET LENGTH OF STRING      01-00268
000C80 45E0 3B06            00B0C  1594+         BAL   R14,SBRSEND                  * CALL SEND PROCEDURE       01-00269
                                   1595+         PRINT NOGEN                                                    02-00042
                                   1596 * PREPARE FOR RECEIVE ************************************************* CTC91100
                                   1597          MINC  STEPVAL,1,HALF               * INCREMENT SCRIPT INDEX    CTC91200
                                   1598+         PRINT GEN                                                      02-00033
000C84 4870 3EC6            00ECC  1599+         LH    R7,STEPVAL                   * GET TARGET TO INCREMENT   01-00162
000C88 4177 0001            00001  1600+         LA    R7,1(R7)                     * INCREMENT                 01-00163
000C8C 4070 3EC6            00ECC  1601+         STH   R7,STEPVAL                   * STORE NEW TARGET VALUE    01-00164
                                   1602+         PRINT NOGEN                                                    02-00042
                                   1603          MTBLPUT STEPTBL,INDEX,STEPVAL,HALF * SAVE CURRENT SCRIPT INDEX CTC91300
                                   1604+         PRINT GEN                                                      02-00033
000C90 48B0 3F96            00F9C  1605+         LH    R11,INDEX                    * PREPARE INDEX REGISTER    01-00207
000C94 06B0                        1606+         BCTR  R11,R0                       * 1 LESS TO CHANGE ORIGINE  01-00208
000C96 89B0 0001            00001  1607+         SLL   R11,1                        * MULTIPLY BY 2             01-00218
000C9A 48A0 3EC6            00ECC  1608+         LH    R10,STEPVAL                  * PREPARE NEW VALUE         01-00219
```

```
000C9E 40AB 3F0C            00F12  1609+         STH   R10,(STEPTBL)(R11)           * STORE IN TABLE            01-00220
                                   1610+         PRINT NOGEN                                                    02-00042
                                   1611 * SET PENDING RECEIVE MCH EXIT ROUTINE ******************************** CTC91400
000CA2 D200 42CA 3F8B 012D0 00F91  1612 COLLECTA MVC   AREAMCHR(1),=C' '            * ERASE AREAMCHR            CTC91500
000CA8 D2FE 42CB 42CA 012D1 012D0  1613          MVC   AREAMCHR+1(255),AREAMCHR     * ERASE AREAMCHR            CTC91600
000CAE D203 43CA 43C9 013D0 013CF  1614          MVC   AREAMCHR+256(4),AREAMCHR+255 * ERASE AREAMCHR            CTC91700
                                   1615          RECEIVE RPL=RPLMCH,AREA=AREAMCHR,RECLEN=260,                  CCTC91800
                                                       OPTCD=(ASY,SPEC),RTYPE=(DFSYN,DFASY,NRESP),EXIT=RCVMCH   CTC91900
                                   1663          MVERIFY                            * VERIFY RECEIVE COMPLETION CTC92000
                                   1664+         PRINT GEN                                                      02-00033
000D18 12FF                        1665+         LTR   R15,R15                      * TEST RETURN CODE          01-00282
000D1A 4780 3D1C            00D22  1666+         BZ    MV0011                       * NO PROBLEM                01-00283
000D1E 45E0 3446            0044C  1667+         BAL   R14,FAILEXIT                 * DUMP                      01-00284
                            00D22  1668+MV0011   EQU   *                            * RESUME STATEMENTS         01-00285
                                   1669+         PRINT NOGEN                                                    02-00042
                                   1670 * REPLAY SCRIPT ******************************************************* CTC92100
000D22 9500 3F8D      00F93        1671          CLI   ENDSBOOL,FALSE               * CMP LOOP INDICATOR        CTC92200
000D26 4780 3B00            00B06  1672          BE    EXRCVMCH                     * LEAVE                     CTC92300
000D2A 9200 3F8D      00F93        1673          MVI   ENDSBOOL,FALSE               * RESET LOOP INDICATOR      CTC92400
000D2E 45E0 3D30            00D36  1674          BAL   R14,INISESS                  * INITIALISE ONE SESSION    CTC92500
000D32 47F0 3B00            00B06  1675          B     EXRCVMCH                     * LEAVE                     CTC92600
                                   1676 * INITIALISE SESSION PROCEDURE **************************************** CTC92700
000D36 50E0 3DDE            00DE4  1677 INISESS  ST    R14,R14SAVE2                 * KEEP RETURN ADRESS        CTC92800
000D3A D201 3F9E 3F96 00FA4 00F9C  1678          MVC   RESIVAL(2),INDEX             * INIT RESIVAL FOR SBRSEND  CTC92900
000D40 D201 3F9C 3F9E 00FA2 00FA4  1679          MVC   CNIDVAL(2),RESIVAL           * INIT CNIDVAL              CTC93000
000D46 9680 3F9C      00FA2        1680          OI    CNIDVAL,CNIDSENT             * SET INTERNAL CNID FLAG    CTC93100
                                   1681          MTBLPUT ACTITBL,INDEX,CNIDVAL,HALF * PUT CNID VALUE IN TBL     CTC93200
                                   1682+         PRINT GEN                                                      02-00033
000D4A 48B0 3F96            00F9C  1683+         LH    R11,INDEX                    * PREPARE INDEX REGISTER    01-00207
000D4E 06B0                        1684+         BCTR  R11,R0                       * 1 LESS TO CHANGE ORIGINE  01-00208
000D50 89B0 0001            00001  1685+         SLL   R11,1                        * MULTIPLY BY 2             01-00218
000D54 48A0 3F9C            00FA2  1686+         LH    R10,CNIDVAL                  * PREPARE NEW VALUE         01-00219
000D58 40AB 3F22            00F28  1687+         STH   R10,(ACTITBL)(R11)           * STORE IN TABLE            01-00220
                                   1688+         PRINT NOGEN                                                    02-00042
                                   1689          MTBLGET PLAYTBL,INDEX,PLAYVAL,FULL * GET SCRIPT BASE ADRESS    CTC93300
                                   1690+         PRINT GEN                                                      02-00033
000D5C 48B0 3F96            00F9C  1691+         LH    R11,INDEX                    * PREPARE INDEX REGISTER    01-00179
000D60 06B0                        1692+         BCTR  R11,R0                       * 1 LESS TO CHANGE ORIGINE  01-00180
000D62 89B0 0002            00002  1693+         SLL   R11,2                        * MULTIPLY BY 4             01-00185
000D66 58BB 3ECE            00ED4  1694+         L     R11,(PLAYTBL)(R11)           * EXTRACT TABLE ELEMENT     01-00186
000D6A 50B0 3EA2            00EA8  1695+         ST    R11,PLAYVAL                  * STORE IN WORK STORAGE     01-00187
                                   1696+         PRINT NOGEN                                                    02-00042
000D6E D201 3EC6 3F86 00ECC 00F8C  1697          MVC   STEPVAL(2),=H'1'             * INIT STEPVAL              CTC93400
                                   1698          MTBLPUT STEPTBL,INDEX,STEPVAL,HALF * PUT CURRENT STEP IN TBL   CTC93500
                                   1699+         PRINT GEN                                                      02-00033
000D74 48B0 3F96            00F9C  1700+         LH    R11,INDEX                    * PREPARE INDEX REGISTER    01-00207
000D78 06B0                        1701+         BCTR  R11,R0                       * 1 LESS TO CHANGE ORIGINE  01-00208
000D7A 89B0 0001            00001  1702+         SLL   R11,1                        * MULTIPLY BY 2             01-00218
000D7E 48A0 3EC6            00ECC  1703+         LH    R10,STEPVAL                  * PREPARE NEW VALUE         01-00219
000D82 40AB 3F0C            00F12  1704+         STH   R10,(STEPTBL)(R11)           * STORE IN TABLE            01-00220
                                   1705+         PRINT NOGEN                                                    02-00042
                                   1706 * DO FIRST STEP OF THE SESSION **************************************** CTC93600
000D86 41E0 3D80            00D86  1707          LA    R14,*                        * USEFUL IF PRELAST CALLED  CTC93700
                                   1708          DOACTION                           * BRANCH IF NO DATA POINTER CTC93800
                                   1709+         PRINT GEN                                                      02-00033
000D8A 58A0 3EA2            00EA8  1710+         L     R10,PLAYVAL                  * SCRIPT BASE POINTER       01-00231
000D8E 48B0 3EC6            00ECC  1711+         LH    R11,STEPVAL                  * STEP OF SCRIPT            01-00232
000D92 06B0                        1712+         BCTR  R11,R0                       * ORIGIN=0                  01-00233
000D94 89B0 0002            00002  1713+         SLL   R11,2                        * FULLWORD DEPLACMENT       01-00234
000D98 587A B000            00000  1714+         L     R7,0(R10,R11)                * GET SCRIPT ACTION         01-00235
```

```
000D9C 5070 3ECA            00ED0  1715+         ST    R7,ACTIVAL                   * FOR FURTHER USE           01-00236
000DA0 5570 3F72            00F78  1716+         CL    R7,=A(BEGINCMD)              * CMP WITH COMMAND BORDER   01-00237
000DA4 0747                        1717+         BLR   R7                           * SPECIFIC ACTION           01-00238
000DA6 5570 3F76            00F7C  1718+         CL    R7,=A(ENDCMD)                * CMP WITH COMMAND BORDER   01-00239
000DAA 07B7                        1719+         BNLR  R7                           * SPECIFIC ACTION           01-00240
                                   1720+         PRINT NOGEN                                                    02-00042
                                   1721 * SEND ACTION DATA **************************************************** CTC93900
                                   1722          SETSEND                                                        CTC94000
                                   1723+         PRINT GEN                                                      02-00033
000DAC 5870 3ECA            00ED0  1724+         L     R7,ACTIVAL                   * GET CURRENT ACTION        01-00265
000DB0 4187 0001            00001  1725+         LA    R8,1(R7)                     * GET ADRESS OF STRING      01-00266
000DB4 1799                        1726+         XR    R9,R9                        * NULLIFY R9                01-00267
000DB6 4397 0000            00000  1727+         IC    R9,0(R7)                     * GET LENGTH OF STRING      01-00268
000DBA 45E0 3B06            00B0C  1728+         BAL   R14,SBRSEND                  * CALL SEND PROCEDURE       01-00269
                                   1729+         PRINT NOGEN                                                    02-00042
                                   1730 * PREPARE FOR RECEIVE ************************************************* CTC94100
                                   1731          MINC  STEPVAL,1,HALF               * INCREMENT SCRIPT INDEX    CTC94200
                                   1732+         PRINT GEN                                                      02-00033
000DBE 4870 3EC6            00ECC  1733+         LH    R7,STEPVAL                   * GET TARGET TO INCREMENT   01-00162
000DC2 4177 0001            00001  1734+         LA    R7,1(R7)                     * INCREMENT                 01-00163
000DC6 4070 3EC6            00ECC  1735+         STH   R7,STEPVAL                   * STORE NEW TARGET VALUE    01-00164
                                   1736+         PRINT NOGEN                                                    02-00042
                                   1737          MTBLPUT STEPTBL,INDEX,STEPVAL,HALF * PUT CURRENT STEP IN TBL   CTC94300
                                   1738+         PRINT GEN                                                      02-00033
000DCA 48B0 3F96            00F9C  1739+         LH    R11,INDEX                    * PREPARE INDEX REGISTER    01-00207
000DCE 06B0                        1740+         BCTR  R11,R0                       * 1 LESS TO CHANGE ORIGINE  01-00208
000DD0 89B0 0001            00001  1741+         SLL   R11,1                        * MULTIPLY BY 2             01-00218
000DD4 48A0 3EC6            00ECC  1742+         LH    R10,STEPVAL                  * PREPARE NEW VALUE         01-00219
000DD8 40AB 3F0C            00F12  1743+         STH   R10,(STEPTBL)(R11)           * STORE IN TABLE            01-00220
                                   1744+         PRINT NOGEN                                                    02-00042
000DDC 58E0 3DDE            00DE4  1745          L     R14,R14SAVE2                 * RESTORE RETURN ADRESS     CTC94400
000DE0 07FE                        1746          BR    R14                          * RETURN TO CALLER          CTC94500
000DE4                             1747 R14SAVE2 DS    F                            * RETURN ADRESS STORAGE     CTC94600
                                   1748 * CALL CONNECT PROCEDURE ********************************************** CTC94700
000DE8 D201 3F9E 42CA 00FA4 012D0  1749 CONNECT  MVC   RESIVAL(2),AREAMCHR          * GET OFFERED RESID         CTC94800
000DEE 940F 3F9E      00FA4        1750          NI    RESIVAL,X'0F'                * ERASE DIRECTION INDICATOR CTC94900
                                   1751          MTBLPUT ACTITBL,INDEX,RESIVAL,HALF * UPDATE RESIDS TABLE       CTC95000
                                   1752+         PRINT GEN                                                      02-00033
000DF2 48B0 3F96            00F9C  1753+         LH    R11,INDEX                    * PREPARE INDEX REGISTER    01-00207
000DF6 06B0                        1754+         BCTR  R11,R0                       * 1 LESS TO CHANGE ORIGINE  01-00208
000DF8 89B0 0001            00001  1755+         SLL   R11,1                        * MULTIPLY BY 2             01-00218
000DFC 48A0 3F9E            00FA4  1756+         LH    R10,RESIVAL                  * PREPARE NEW VALUE         01-00219
000E00 40AB 3F22            00F28  1757+         STH   R10,(ACTITBL)(R11)           * STORE IN TABLE            01-00220
                                   1758+         PRINT NOGEN                                                    02-00042
000E04 D502 42CC 3E9C 012D2 00EA2  1759          CLC   AREAMCHR+2(3),RETRCALL       * INFO REPORT RETRY CALL    CTC95100
000E0A 4770 3E10            00E16  1760          BNE   CLEACALL                                                 CTC95200
000E0E 92FF 3F8D      00F93        1761          MVI   ENDSBOOL,TRUE                * SET LOOP INDICATOR        CTC95300
000E12 47F0 3C9C            00CA2  1762          B     COLLECTA                                                 CTC95400
000E16 9513 42CC      012D2        1763 CLEACALL CLI   AREAMCHR+2,X'13'             * CLEAR ON OUTGOING         CTC95500
000E1A 4770 3E32            00E38  1764          BNE   NORMAL                                                   CTC95600
                                   1765          SETSEND CLEACNF                    * CLEAR CONFIRM             CTC95700
                                   1766+         PRINT GEN                                                      02-00033
000E1E 4170 4056            0105C  1767+         LA    R7,CLEACNF                   * GET ADRESS OF DATA        01-00262
000E22 4187 0001            00001  1768+         LA    R8,1(R7)                     * GET ADRESS OF STRING      01-00266
000E26 1799                        1769+         XR    R9,R9                        * NULLIFY R9                01-00267
000E28 4397 0000            00000  1770+         IC    R9,0(R7)                     * GET LENGTH OF STRING      01-00268
000E2C 45E0 3B06            00B0C  1771+         BAL   R14,SBRSEND                  * CALL SEND PROCEDURE       01-00269
                                   1772+         PRINT NOGEN                                                    02-00042
000E30 45E0 3E3C            00E42  1773          BAL   R14,PRELAST                  * TRIGGER LOOP INDICATOR    CTC95800
000E34 47F0 3C9C            00CA2  1774          B     COLLECTA                                                 CTC95900
```

```
000E38 D203 3ECA 3F7A 00ED0 00F80  1775 NORMAL   MVC   ACTIVAL(4),=A(CALLCNT)       * PREPARE COMMAND CONTROL   CTC96000
000E3E 47F0 3C28           00C2E  1776          B     LOCKCMD                      * RESUME TO COMMAND CONTROL CTC96100
                                   1777 * PRELAST PROCEDURE **************************************************** CTC96200
                                   1778 * INPUT : STEPVAL, R14.                                                 CTC96300
                                   1779 * OUTPUT: ENDSBOOL, STEPVAL.                                            CTC96400
000E42 D201 3502 3F9E 00508 00FA4  1780 PRELAST  MVC   WORKN(2),RESIVAL             * PREPARE UNPACK            CTC96500
000E48 F342 3506 3502 0050C 00508  1781          UNPK  WORKC(5),WORKN(3)            * UNPACK HALFBYTE TO EBCDIC CTC96600
000E4E DC03 3506 341F 0050C 00425  1782          TR    WORKC(4),HEX                 * TRANSLATE EBCDIC TO HEXA  CTC96700
000E54 D203 3E83 3506 00E89 0050C  1783          MVC   WTOENDS+45(4),WORKC          * PLACE RESULT IN WTO MSG   CTC96800
                                   1784 WTOENDS  WTO   ' CTCP ||| ACTIV : END SESSION, RESID=____'              CTC96900
000E90 92FF 3F8D      00F93        1792          MVI   ENDSBOOL,TRUE                * SET LOOP INDICATOR        CTC97000
                                   1793          MINC  STEPVAL,1,HALF               * POINT TO NEXT STEP        CTC97100
                                   1794+         PRINT GEN                                                      02-00033
000E94 4870 3EC6           00ECC  1795+         LH    R7,STEPVAL                   * GET TARGET TO INCREMENT   01-00162
000E98 4177 0001           00001  1796+         LA    R7,1(R7)                     * INCREMENT                 01-00163
000E9C 4070 3EC6           00ECC  1797+         STH   R7,STEPVAL                   * STORE NEW TARGET VALUE    01-00164
                                   1798+         PRINT NOGEN                                                    02-00042
000EA0 07FE                        1799          BR    R14                          * RESUME PLAY AS NO PRELAST CTC97200
                                   1800 * DATA FOR ACTIV PART ************************************************* CTC97300
000EA2 000B04                      1801 RETRCALL DC    X'000B04'                    * INFO REPORT RETRY CALL    CTC97400
000EA8                             1802 PLAYVAL  DS    F                            * EXCERPT OF PLAYTBL        CTC97500
                                   1803          MMARK ' STEPVAL '                  * FOR READABLE DUMP         CTC97600
                                   1804+         PRINT GEN                                                      02-00033
000EAC 40E2E3C5D7E5C1D3            1805+         DC    C' STEPVAL '                 * GENERATE CONSTANT STRING  01-00492
000EB5 40E2E3C5D7E5C1D3            1806+         DC    C' STEPVAL '                 * GENERATE CONSTANT STRING  01-00493
000EBE 40E2E3C5D7E5C1D3            1807+         DC    C' STEPVAL '                 * GENERATE CONSTANT STRING  01-00494
000EC7 40E2E3C5D7E5C1D3            1808+         DC    C' STEPVAL '                 * GENERATE CONSTANT STRING  01-00495
000ED0                      00ECC  1809+         ORG   *-4                          * OVERLAP STRING            01-00496
000ECC                             1810+         DS    0F                           * ALIGNEMENT STORAGE        01-00497
                                   1811+         PRINT NOGEN                                                    02-00042
000ECC                             1812 STEPVAL  DS    H                            * EXCERPT OF STEPTBL        CTC97700
000ED0                             1813 ACTIVAL  DS    F                            * EXCERPT OF A SCRIPT       CTC97800
000ED4                             1814 PLAYTBL  DS    0F                           * PLAYS ADRESSES FOR ACTIV  CTC97900
                                   1815          MAKEPLAY PLAY1,4                                               CTC98000
                                   1816+         PRINT GEN                                                      02-00033
000ED4 00000F30                    1817+         DC    A(PLAY1)                     * TABLE ELEMENT             01-00465
000ED8 00000F30                    1818+         DC    A(PLAY1)                     * TABLE ELEMENT             01-00465
000EDC 00000F30                    1819+         DC    A(PLAY1)                     * TABLE ELEMENT             01-00465
000EE0 00000F30                    1820+         DC    A(PLAY1)                     * TABLE ELEMENT             01-00465
                                   1821+         PRINT NOGEN                                                    02-00042
                                   1822          MAKEPLAY PLAY2,0                                               CTC98100
                                   1823          MMARK ' DECLARATION'               * FOR READABLE DUMP         CTC98200
                                   1824+         PRINT GEN                                                      02-00033
000EE4 40C4C5C3D3C1D9C1            1825+         DC    C' DECLARATION'              * GENERATE CONSTANT STRING  01-00492
000EF0 40C4C5C3D3C1D9C1            1826+         DC    C' DECLARATION'              * GENERATE CONSTANT STRING  01-00493
000EFC 40C4C5C3D3C1D9C1            1827+         DC    C' DECLARATION'              * GENERATE CONSTANT STRING  01-00494
000F08 40C4C5C3D3C1D9C1            1828+         DC    C' DECLARATION'              * GENERATE CONSTANT STRING  01-00495
000F14                      00F10  1829+         ORG   *-4                          * OVERLAP STRING            01-00496
000F10                             1830+         DS    0F                           * ALIGNEMENT STORAGE        01-00497
                                   1831+         PRINT NOGEN                                                    02-00042
                                   1832          MAKEDECL                                                       CTC98300
                                   1833+         PRINT GEN                                                      02-00033
000F10 0004                        1834+QUANTITY DC    H'4'                         * TOTAL SESSIONS QUANTITY   01-00477
000F12                             1835+STEPTBL  DS    4H                           * STEPS IN PLAYS FOR ACTIV  01-00478
000F1A F000F000F000F000            1836+PASSTBL  DC    4X'F000'                     * RESIDS OR CNIDS FOR PASSI 01-00479
000F22 00                          1837+FLAGTBL  DC    X'00'                        * FLAG TABLE ORIGIN IS 1    01-00480
000F23 10101010                    1838+         DC    4AL1(AVAILABL)               * FLAGS FOR PASSI           01-00481
000F28                             1839+         DS    0H                           * ALIGNEMENT STORAGE        01-00482
000F28 F000F000F000F000            1840+ACTITBL  DC    4X'F000'                     * RESIDS OR CNIDS FOR ACTIV 01-00483
                                   1841+         PRINT NOGEN                                                    02-00042
```

```
000F30 00001010                    1842 PLAY1    DC    A(CALLREQ0)                                              CTC98400
000F34 00000DE8                    1843          DC    A(CONNECT)                                               CTC98500
000F38 0000106A                    1844          DC    A(INTEREQ)                                               CTC98600
000F3C 0000106F                    1845          DC    A(INTECNF)                                               CTC98700
000F40 00000FFA                    1846          DC    A(UDATA01)                                               CTC98800
000F44 00001056                    1847          DC    A(CLEAREQ)                                               CTC98900
000F48 00000E42                    1848          DC    A(PRELAST)                                               CTC99000
000F4C 0000105C                    1849          DC    A(CLEACNF)                                               CTC99100
000F50 00001026                    1850 PLAY2    DC    A(CALLREQ5)                                              CTC99200
000F54 00000DE8                    1851          DC    A(CONNECT)                                               CTC99300
000F58 0000106A                    1852          DC    A(INTEREQ)                                               CTC99400
000F5C 0000106F                    1853          DC    A(INTECNF)                                               CTC99500
000F60 00000FFA                    1854          DC    A(UDATA01)                                               CTC99600
000F64 00001056                    1855          DC    A(CLEAREQ)                                               CTC99700
000F68 00000E42                    1856          DC    A(PRELAST)                                               CTC99800
000F6C 0000105C                    1857          DC    A(CLEACNF)                                               CTC99900
                                   1858 *********************************************************************** CTC00000
                                   1859 *        END OF ACTIV PART                                            * CTC00100
                                   1860 *********************************************************************** CTC00200
                                   1861 *                                                                       CTC00300
                                   1863 * DATA FOR RECEIVE ON MCH ********************************************* CTC00500
000F70                             1864          LTORG                          * PLACE OF LITERALS         CTC00600
000F70 00000000                    1865                =A(CTCPSTR)
000F74 00000FFF                    1866                =X'00000FFF'
000F78 00000FFA                    1867                =A(BEGINCMD)
000F7C 00001083                    1868                =A(ENDCMD)
000F80 0000104F                    1869                =A(CALLCNT)
000F84 F0F1                        1870                =C'01'
000F86 F000                        1871                =X'F000'
000F88 0004                        1872                =H'4'
000F8A 0008                        1873                =H'8'
000F8C 0001                        1874                =H'1'
000F8E C4E4F3                      1875                =C'DU3'
000F91 40                          1876                =C' '
                                   1877          MEQUATE                        * EQUATES                   CTC00700
                                   1878+         PRINT GEN                                                      02-00033
                            00000  1879+R0       EQU   0                                                        01-00088
                            00001  1880+R1       EQU   1                                                        01-00089
                            00002  1881+R2       EQU   2                                                        01-00090
                            00003  1882+R3       EQU   3                                                        01-00091
                            00004  1883+R4       EQU   4                                                        01-00092
                            00005  1884+R5       EQU   5                                                        01-00093
                            00006  1885+R6       EQU   6                                                        01-00094
                            00007  1886+R7       EQU   7                                                        01-00095
                            00008  1887+R8       EQU   8                                                        01-00096
                            00009  1888+R9       EQU   9                                                        01-00097
                            0000A  1889+R10      EQU   10                                                       01-00098
                            0000B  1890+R11      EQU   11                                                       01-00099
                            0000C  1891+R12      EQU   12                                                       01-00100
                            0000D  1892+R13      EQU   13                                                       01-00101
                            0000E  1893+R14      EQU   14                                                       01-00102
                            0000F  1894+R15      EQU   15                                                       01-00103
                                   1895+         PRINT NOGEN                                                    02-00042
                            000FF  1896 TRUE     EQU   X'FF'                    * TRUE FLAG VALUE           CTC00800
                            00000  1897 FALSE    EQU   X'00'                    * FALSE FLAG VALUE          CTC00900
                            00008  1898 INTRSENT EQU   X'08'                    * ALLOW INTERRUPT CONFIRM   CTC01000
                            00010  1899 AVAILABL EQU   X'10'                    * INDICATE AVAILABLE PASSIV CTC01100
                                   1900 *                                       * INDIRECT USE IN MAKEDECL  CTC01200
                            00020  1901 CLEASENT EQU   X'20'                    * TO DETECT CLEA COLLISION  CTC01300
                            00040  1902 RSETSENT EQU   X'40'                    * TO DETECT RSET COLLISION  CTC01400
                            00080  1903 CNIDSENT EQU   X'80'                        * CNID AND RESID DIFFERENCE CTC01500
000F92 00                          1904 RSARBOOL DC    AL1(FALSE)                   * TO DETECT RSAR COLLISION  CTC01600
000F93 00                          1905 ENDSBOOL DC    AL1(FALSE)                   * TO RESUME SCRIPT          CTC01700
000F94                             1906 MAXRETRY DS    H'10'                        * MAXIMUM SEND RETRY        CTC01800
000F96 0000
000F98 0000000F                    1907 FOURBITS DC    F'15'                        * MASK FOR BITS 0-3         CTC01900
                            000C0  1908 LLC0     EQU   X'C0'                        * VC TYPE CODE              CTC02000
                            00001  1909 LLC5     EQU   X'01'                        * VC TYPE CODE              CTC02100
000F9C                             1910 INDEX    DS    H                            * SELECT SESSION DATA       CTC02200
000F9E                             1911 COUNT    DS    H                            * RETRY SEND COUNTER        CTC02300
000FA0                             1912 WANTED   DS    H                            * TARGET RESID OR CNID      CTC02400
000FA2                             1913 CNIDVAL  DS    H                            * EXCERPT OF RESITBL        CTC02500
000FA4                             1914 RESIVAL  DS    H                            * EXCERPT OF RESITBL        CTC02600
                                   1915 * THE REST IS TO USE IF ACTIV CODE IS DELETED ************************* CTC02700
                                   1916 * REPLACE ?? BY QUANTITY OF SESSION TO SUPPORT ************************ CTC02800
                                   1917 *QUANTITY DC    H'??'                        * TOTAL SESSIONS QUANTITY  CTC02900
                                   1918 *PASSTBL  DC    ??X'F000'                    * RESIDS OR CNIDS FOR PASS CTC03000
                                   1919 *FLAGTBL  DC    X'00'                        * FLAG TABLE ORIGIN IS 1   CTC03100
                                   1920 *         DC    ??AL1(AVAILABL)              * FLAGS FOR PASSI          CTC03200
                                   1922 *********************************************************************** CTC03400
                                   1923 *                                                                     * CTC03500
                                   1924 *     L O S T E R M   O N   M C H   E X I T - R O U T I N E           * CTC03600
                                   1925 *                                                                     * CTC03700
                                   1926 *********************************************************************** CTC03800
                                   1927 CTCPLOST MSBRIN                             * ESTABLISH ADRESSABILITY   CTC03900
                                   1928+         PRINT GEN                                                      02-00033
                                   1929+         PUSH  USING                        * TEMPORARY CHANGE BASE     01-00116
                                   1930+         DROP  R3,R4,R5                     * TEMPORARY CHANGE BASE     01-00117
000FA6 05F0                        1931+CTCPLOST BALR  R15,0                        * TEMPORARY CHANGE BASE     01-00118
                            00FA8  1932+         USING *,R15                        * TEMPORARY CHANGE BASE     01-00119
000FA8 5830 F010            00FB8  1933+         L     R3,B00012                    * SET FIRST BASE            01-00120
000FAC 5840 F014            00FBC  1934+         L     R4,B10012                    * SET SECOND BASE           01-00121
000FB0 5850 F018            00FC0  1935+         L     R5,B20012                    * SET THIRD BASE            01-00122
000FB4 47F0 F01C            00FC4  1936+         B     LB0012                       * SKIP BASES VALUES         01-00123
000FB8 00000006                    1937+B00012   DC    A(EXTBASES)                  * VALUE OF FIRST BASE       01-00124
000FBC 00001006                    1938+B10012   DC    A(EXTBASES+4096)             * VALUE OF SECOND BASE      01-00125
000FC0 00002006                    1939+B20012   DC    A(EXTBASES+8192)             * VALUE OF THIRD BASE       01-00126
                                   1940+         DROP  R15                          * RESTORE BASES             01-00127
                                   1941+         POP   USING                        * RESTORE BASES             01-00128
000FC4 50E0 338A            00390  1942+LB0012   ST    R14,R14SAVE                  * KEEP RETURN ADRESS        01-00129
000FC8 41D0 338E            00394  1943+         LA    R13,SBRSAVE                  * SET UP NEW SAVE AREA      01-00130
                                   1944+         PRINT NOGEN                                                    02-00042
                                   1945          CLSDST RPL=RPLMCH,OPTCD=SYN                                    CTC04000
                                   1962          MVERIFY                            * VERIFY CLSDST COMPLETION  CTC04100
                                   1963+         PRINT GEN                                                      02-00033
000FEA 12FF                        1964+         LTR   R15,R15                      * TEST RETURN CODE          01-00282
000FEC 4780 3FEE            00FF4  1965+         BZ    MV0013                       * NO PROBLEM                01-00283
000FF0 45E0 3446            0044C  1966+         BAL   R14,FAILEXIT                 * DUMP                      01-00284
                            00FF4  1967+MV0013   EQU   *                            * RESUME STATEMENTS         01-00285
                                   1968+         PRINT NOGEN                                                    02-00042
                                   1969 EXITLOST MSBROUT                            * RETURN TO MVS             CTC04200
                                   1970+         PRINT GEN                                                      02-00033
000FF4 58E0 338A            00390  1971+EXITLOST L     R14,R14SAVE                  * RESTORE RETURN ADRESS     01-00139
000FF8 07FE                        1972+         BR    R14                          * RETURN TO VTAM            01-00140
                                   1973+         PRINT NOGEN                                                    02-00042
                                   1975 *********************************************************************** CTC04400
                                   1976 *                                                                     * CTC04500
                                   1977 *              D A T A    A R E A                                     * CTC04600
                                   1978 *                                                                     * CTC04700
                                   1979 *********************************************************************** CTC04800
                                   1980 * COMMUNICATION CONSTANTS ********************************************* CTC04900
                            00FFA  1981 BEGINCMD EQU   * ..||..||..||..||..         * LOW COMMAND BORDER        CTC05000
000FFA 0F000000                    1982 UDATA01  DC    X'0F000000'                  * UNQUALIFIED DATA  1       CTC05100
000FFE E2E3D6D740E2C5E2            1983          DC    C'STOP SESSION'              * MESSAGE TO APPLICATION    CTC05200
00100A 050000000000                1984 INFOREP  DC    X'050000000000'              * ERROR / INFO REPORT       CTC05300
001010 1500000B030080C0            1985 CALLREQ0 DC    X'1500000B030080C005'        * CALL REQUEST              CTC05400
001019 C5C3C8D6E2                  1986          DC    C'ECHOS'                     *      APPLICATION NAME     CTC05500
00101E 09106040164000C0            1987          DC    X'09106040164000C0'          *      LINE NUMBER          CTC05600
001026 1500000B03008001            1988 CALLREQ5 DC    X'1500000B0300800105'        * CALL REQUEST              CTC05700
00102F C5C3C8D6E2                  1989          DC    C'ECHOS'                     *      APPLICATION NAME     CTC05800
001034 0910604016400001            1990          DC    X'0910604016400001'          *      LINE NUMBER          CTC05900
00103C 0400000B00                  1991 CALLINC  DC    X'0400000B00'                * CALL INCOMING             CTC06000
001041 0D00000F030080AA            1992 CALLACC  DC    X'0D00000F030080AA05'        * CALL ACCEPTED             CTC06100
00104A C5C3C8D6E2                  1993          DC    C'ECHOS'                     *      APPLICATION NAME     CTC06200
00104F 0600000F000000              1994 CALLCNT  DC    X'0600000F000000'            * CALL CONNECTED            CTC06300
                                   1995 *                ..||..||..||..||..                                     CTC06400
001056 050000130000                1996 CLEAREQ  DC    X'050000130000'              * CLEAR REQUEST             CTC06500
00105C 03000017                    1997 CLEACNF  DC    X'03000017'                  * CLEAR CONFIRM             CTC06600
001060 0500001B0000                1998 RSETREQ  DC    X'0500001B0000'              * RESET REQUEST             CTC06700
001066 0300001F                    1999 RSETCNF  DC    X'0300001F'                  * RESET CONFIRM             CTC06800
00106A 0400002300                  2000 INTEREQ  DC    X'0400002300'                * INTERRUPT REQUEST         CTC06900
00106F 03000027                    2001 INTECNF  DC    X'03000027'                  * INTERRUPT CONFIRM         CTC07000
001073 050000F10000                2002 DIAGNOS  DC    X'050000F10000'              * DIAGNOSTIC                CTC07100
001079 050000FB0000                2003 RSARREQ  DC    X'050000FB0000'              * RESTART REQUEST           CTC07200
00107F 030000FF                    2004 RSARCNF  DC    X'030000FF'                  * RESTART CONFIRM           CTC07300
                            01083  2005 ENDCMD   EQU   * ..||..||..||..||..         * HIGH COMMAND BORDER       CTC07400
                                   2006 * COMMUNICATION CONTROL BLOCKS AND VARIABLES ************************** CTC07500
                                   2007 ACBMCH   ACB   AM=VTAM,EXLST=CTCPLST,APPLID=CTCPNAME                    CTC07600
                                   2054 CTCPLST  EXLST AM=VTAM,LOSTERM=CTCPLOST                                 CTC07700
00111D 05                          2076 CTCPNAME DC    X'05'                                                    CTC07800
00111E C5C3C8D6F3                  2077          DC    C'ECHO3'                                                 CTC07900
                                   2078 NIBMCH   NIB   NAME=XU0B2,MODE=RECORD,BNDAREA=BINDMCH                   CTC08000
001164 010303                      2096 BINDMCH  DC    X'010303'                                                CTC08100
001167 B0B000400000                2097          DC    X'B0B000400000'                                          CTC08200
00116D 0000000000000000            2098          DC    17X'00'                                                  CTC08300
00117E 08                          2099          DC    X'08'                                                    CTC08400
00117F C5C3C8D6F3404040            2100          DC    C'ECHO3   '                                              CTC08500
001187 08                          2101          DC    X'08'                                                    CTC08600
001188 C4E4F34040404040            2102          DC    C'DU3     '                                              CTC08700
                                   2103          MMARK ' RPLMCHS '                  * FOR READABLE DUMP         CTC08800
                                   2104+         PRINT GEN                                                      02-00033
001190 40D9D7D3D4C3C8E2            2105+         DC    C' RPLMCHS '                 * GENERATE CONSTANT STRING  01-00492
001199 40D9D7D3D4C3C8E2            2106+         DC    C' RPLMCHS '                 * GENERATE CONSTANT STRING  01-00493
0011A2 40D9D7D3D4C3C8E2            2107+         DC    C' RPLMCHS '                 * GENERATE CONSTANT STRING  01-00494
0011AB 40D9D7D3D4C3C8E2            2108+         DC    C' RPLMCHS '                 * GENERATE CONSTANT STRING  01-00495
0011B4                      011B0  2109+         ORG   *-4                          * OVERLAP STRING            01-00496
0011B0                             2110+         DS    0F                           * ALIGNEMENT STORAGE        01-00497
                                   2111+         PRINT NOGEN                                                    02-00042
                                   2112 RPLMCHS  RPL   AM=VTAM,ACB=ACBMCH,NIB=NIBMCH,AREA=AREAMCHR,AREALEN=255,CCTC08900
                                                       POST=SCHED                                               CTC09000
                                   2174          MMARK ' RPLMCH '                   * FOR READABLE DUMP         CTC09100
                                   2175+         PRINT GEN                                                      02-00033
001220 40D9D7D3D4C3C840            2176+         DC    C' RPLMCH '                  * GENERATE CONSTANT STRING  01-00492
001228 40D9D7D3D4C3C840            2177+         DC    C' RPLMCH '                  * GENERATE CONSTANT STRING  01-00493
001230 40D9D7D3D4C3C840            2178+         DC    C' RPLMCH '                  * GENERATE CONSTANT STRING  01-00494
001238 40D9D7D3D4C3C840            2179+         DC    C' RPLMCH '                  * GENERATE CONSTANT STRING  01-00495
001240                      0123C  2180+         ORG   *-4                          * OVERLAP STRING            01-00496
00123C                             2181+         DS    0F                           * ALIGNEMENT STORAGE        01-00497
                                   2182+         PRINT NOGEN                                                    02-00042
                                   2183 RPLMCH   RPL   AM=VTAM,ACB=ACBMCH,NIB=NIBMCH,AREA=AREAMCHR,AREALEN=255,CCTC09200
```

```
                                                       POST=SCHED                                               CTC09300
                                   2245          MMARK ' AREAMCHR '                 * FOR READABLE DUMP         CTC09400
                                   2246+         PRINT GEN                                                      02-00033
0012AC 40C1D9C5C1D4C3C8            2247+         DC    C' AREAMCHR '                * GENERATE CONSTANT STRING  01-00492
0012B6 40C1D9C5C1D4C3C8            2248+         DC    C' AREAMCHR '                * GENERATE CONSTANT STRING  01-00493
0012C0 40C1D9C5C1D4C3C8            2249+         DC    C' AREAMCHR '                * GENERATE CONSTANT STRING  01-00494
0012CA 40C1D9C5C1D4C3C8            2250+         DC    C' AREAMCHR '                * GENERATE CONSTANT STRING  01-00495
0012D4                      012D0  2251+         ORG   *-4                          * OVERLAP STRING            01-00496
0012D0                             2252+         DS    0F                           * ALIGNEMENT STORAGE        01-00497
                                   2253+         PRINT NOGEN                                                    02-00042
0012D0 4040404040404040            2254 AREAMCHR DC    260C' '                      * RECEIVE AREA              CTC09500
000000                             2255          END   CTCPSTR                                                  CTC09600
                                             CROSS REFERENCE
SYMBOL    LEN    VALUE  DEFN  REFERENCES                                                       ASM H V 02 16.17 09/27/88
ACBMCH   00004   001084 2008  0537 0863 2131 2202
ACCEPT   00006   000820 1218  1205 1207
ACTIMODE 00004   000BE4 1533  1012
ACTITBL  00002   000F28 1840  1008 1687 1757
ACTIVAL  00004   000ED0 1813  1555 1562 1581 1590 1715 1724 1775
AREAMCHR 00001   0012D0 2254  0734 0735 0735 0736 0736 0748 0974 0976 0979 1020 1024 1028 1035 1040 1045 1050 1069 1074
                              1079 1084 1089 1094 1099 1104 1109 1114 1133 1180 1192 1199 1316 1388 1389 1389 1390 1390
                              1402 1520 1563 1612 1613 1613 1614 1614 1626 1749 1759 1763 2134 2205
AVAILABL 00001 00000010 1899  1164 1179 1263 1375 1838
BEGINCMD 00001 00000FFA 1981  1867
BINDMCH  00003   001164 2096  2094
BLANK8   00001   00042C 0884  0548 0564 0567 0855
B00008   00004   000538 0951  0947
B00012   00004   000FB8 1937  1933
B10000   00004   000014 0525  0522
B10008   00004   00053C 0952  0948
B10012   00004   000FBC 1938  1934
B20000   00004   000018 0526  0523
B20008   00004   000540 0953  0949
B20012   00004   000FC0 1939  1935
CALLACC  00009   001041 1992  1218 1221
CALLCNT  00007   00104F 1994  1869
CALLREQ0 00009   001010 1985  1842
CALLREQ5 00009   001026 1988  1850
CLEACALL 00004   000E16 1763  1760
CLEACNF  00004   00105C 1997  1247 1767 1849 1857
CLEAREQ  00006   001056 1996  1137 1210 1847 1855
CLEAROUT 00004   000606 1028  1021
CLEASENT 00001 00000020 1901  1145 1243 1257
CMDFB    00001 000009A4 1331  1046
CMDFF    00001 000009FA 1352  1051
CMDF1    00001 00000944 1315  1041
CMD0B    00001 0000075C 1157  1036 1080
CMD0F    00001 0000083C 1228  1085
CMD00    00001 000006F4 1132  1070
CMD02    00001 0000071E 1147  1075
CMD1B    00001 000008C8 1273  1100
CMD1F    00001 000008EE 1287  1105
CMD13    00001 00000868 1240  1090
CMD17    00001 0000088E 1254  1095
CMD23    00001 0000090E 1296  1110
CMD27    00001 00000924 1306  1115
CNIDSENT 00001 00000080 1903  0981 1022 1026 1030 1680
CNIDVAL  00002   000FA2 1913  1679 1680 1686
COLLECTA 00006   000CA2 1612  1762 1774
COLLECTP 00006   000A84 1388  1062 1130 1134 1146 1156 1177 1216 1227 1239 1260 1272 1286 1293 1295 1305 1312 1314 1330
                              1365 1386
```

```
CONNECT  00006   000DE8 1749  1843 1851
COUNT    00002   000F9E 1911  1484 1502
CTCPLOST 00002   000FA6 1931  2075
CTCPLST  00004   0010F0 2055  2030
CTCPNAME 00001   00111D 2076  2044
CTCPSTR  00001 00000000 0517  1865 2255
DANSWER  00003   000434 0885  0839 0856
ENDCMD   00001 00001083 2005  1868
                                             CROSS REFERENCE
SYMBOL    LEN    VALUE  DEFN  REFERENCES                                                         ASM H V 02 16.17 09/27/88
ENDSBOOL 00001   000F93 1905  1671 1673 1761 1792
ERALOOP  00004   000A56 1372  1385
ERASEALL 00004   000A52 1371  1351
ERASEINT 00004   00093C 1313  1310
ERASERSA 00004   000A46 1366  1342 1363
ERASERSE 00004   000906 1294  1277 1291
EXRCVMCH 00004   000B06 1448  1672 1675
EXTBASES 00004   000006 0522  0525 0526 0951 0952 0953 1937 1938 1939
FAILAREA 00004   0004C8 0929  0894
FAILEXIT 00004   00044C 0894  0543 0601 0668 0693 0719 0789 0869 0970 1061 1129 1176 1259 1292 1311 1364 1443 1512 1515
                              1523 1565 1667 1966
FALSE    00001 00000000 1897  1366 1671 1673 1904 1905
FLAGTBL  00001   000F22 1837  1144 1163 1242 1256 1262 1275 1289 1308 1374
FOUNDPAS 00004   0007B0 1179  1165
FOURBITS 00004   000F98 1907  1194 1196
HEX      00016 00000425 0934  0898 1318 1782
HEXTR    00016   000515 0933  0934
IDA0017A 00004   0000D8 0583  0580 0582 0584
IDA0036A 00004   00016C 0674  0673 0675
IDA0042A 00004   0001B0 0702  0699 0701 0703
IHB0010A 00002   000080 0558  0551
IHB0014A 00002   0000C8 0575  0571
IHB0028  00001 00000150 0659  0615
IHB0077  00001 00000266 0780  0747
IHB0084A 00002   0002DE 0814  0807
IHB0088A 00002   00034E 0849  0842
IHB0100A 00002   000492 0906  0902
IHB0134A 00002   00066E 1059  1055
IHB0165A 00002   0006EA 1127  1123
IHB0169A 00002   000756 1154  1150
IHB0170A 00002   0007A6 1174  1170
IHB0180A 00002   000862 1235  1231
IHB0193A 00002   00099A 1327  1323
IHB0194A 00002   0009D6 1338  1334
IHB0198A 00002   000A30 1359  1355
IHB0204  00001 00000AEE 1434  1401
IHB0213  00001 00000B4C 1477  1473
IHB0216A 00002   000BBE 1510  1506
IHB0242  00001 00000D0A 1658  1625
IHB0274A 00002   000E8E 1790  1786
IHB0298  00001 00000FDC 1957  1955
INDEX    00002   000F9C 1910  0829 0831 0984 0987 0995 1002 1005 1013 1143 1161 1162 1166 1185 1241 1255 1261 1266 1274
                              1288 1307 1372 1373 1378 1384 1533 1541 1605 1678 1683 1691 1700 1739 1753
INISESS  00004   000D36 1677  0830 1674
INTECNF  00004   00106F 2001  1299 1845 1853
INTEREQ  00005   00106A 2000  1844 1852
INTRSENT 00001 00000008 1898  1309 1313
LB0000   00002   00001C 0527  0524
LB0008   00004   000544 0956  0950
LB0012   00004   000FC4 1942  1936
```

```
LLC0     00001 000000C0 1908  1204
LLC5     00001 00000001 1909  1206
LOCKCMD  00004   000C2E 1562  1776
MAINSAVE 00004   0003DC 0880  0528 0530 0874
MAXRETRY 00002   000F94 1906  1483
                                             CROSS REFERENCE
SYMBOL    LEN    VALUE  DEFN  REFERENCES                                                       ASM H V 02 16.17 09/27/88
MV0001   00001 00000040 0544  0542
MV0002   00001 0000010E 0602  0600
MV0003   00001 00000168 0669  0667
MV0004   00001 000001A2 0694  0692
MV0005   00001 000001E2 0720  0718
MV0006   00001 0000027E 0790  0788
MV0007   00001 00000380 0870  0868
MV0009   00001 00000568 0971  0969
MV0010   00001 00000B06 1444  1442
MV0011   00001 00000D22 1668  1666
MV0013   00001 00000FF4 1967  1965
NEXTCMD  00004   000C40 1569  1564
NEXTIDA  00004   0005BC 1002  1014
NEXTIDP  00004   000590 0984  0996
NEXTPASS 00004   000760 1161  1167
NEXTSES  00004   00030A 0829  0832
NIBARG   00004   000448 0889  0680 0698
NIBMCH   00004   001124 2079  0588 2135 2206
NORETRY  00004   000BC4 1513  1486 1488
NORMAL   00006   000E38 1775  1764
NUMAGAIN 00006   00028E 0803  0823 0825
NUMBER   00008   000440 0888  0820 0821
OTHERCMD 00004   000616 1035  1023 1029
PASSMODE 00004   000678 1069  0994
PASSTBL  00002   000F1A 1836  0990 1189 1270 1382
PLAYTBL  00004   000ED4 1814  1536 1694
PLAYVAL  00004   000EA8 1802  1537 1550 1576 1695 1710
PLAY1    00004   000F30 1842  1817 1818 1819 1820
PRELAST  00006   000E42 1780  1773 1848 1856
QUANTITY 00002   000F10 1834  0822 0826 0828 0983 1001 1160 1371
RCVMCH   00002   000526 0945  0754 1408 1632
RELEASE  00004   0008A6 1261  1244 1253 1258
REPLY    00001   000424 0883  0548 0553 0564 0566 0567 0568 0579 0804 0809 0820 0839 0844 0855 0856
RESIVAL  00002   000FA4 1914  0723 0723 0991 0993 1009 1011 1181 1182 1188 1462 1678 1679 1749 1750 1756 1780
RETRCALL 00003   000EA2 1801  1759
RETRLOOP 00004   000B5E 1484  1503
RPLMCH   00004   00123C 2184  0613 0679 0745 0961 1399 1497 1623 1953
RPLMCHS  00004   0011B0 2113  0707 1471
RSARBOOL 00001   000F92 1904  0724 1341 1362 1366
RSARCNF  00004   00107F 2004  1345
RSARECB  00004   00043C 0887  0794 0797 0797 1369
RSARREQ  00006   001079 2003  0727
RSETCNF  00004   001066 1999  1280
RSETSENT 00001 00000040 1902  1276 1290 1294
R0       00001 00000000 1879  0520 0894 0988 1006 1186 1267 1379 1487 1534 1542 1552 1578 1606 1684 1692 1701 1712 1740
                              1754
R10      00001 0000000A 1889  1188 1189 1269 1270 1381 1382 1550 1554 1576 1580 1608 1609 1686 1687 1703 1704 1710 1714
                              1742 1743 1756 1757
R11      00001 0000000B 1890  0987 0988 0989 0990 0990 0991 1005 1006 1007 1008 1008 1009 1185 1186 1187 1189 1266 1267
                              1268 1270 1378 1379 1380 1382 1533 1534 1535 1536 1536 1537 1541 1542 1543 1544 1544 1545
                              1551 1552 1553 1554 1577 1578 1579 1580 1605 1606 1607 1609 1683 1684 1685 1687 1691 1692
                              1693 1694 1694 1695 1700 1701 1702 1704 1711 1712 1713 1714 1739 1740 1741 1743 1753 1754
                              1755 1757
```

```
R12      00001 0000000C 1891  0527 0529 0530 0821 0822 0824 0824 0826
R13      00001 0000000D 1892  0527 0528 0529 0874 0957 1943
                                             CROSS REFERENCE
SYMBOL    LEN    VALUE  DEFN  REFERENCES                                                                      ASM H V 02 16.17 09/27/88
R14      00001 0000000E 1893  0543 0601 0668 0693 0719 0731 0789 0830 0869 0895 0896 0956 0970 1061 1129 1141 1176 1214
                              1225 1251 1259 1284 1292 1303 1311 1329 1340 1349 1361 1364 1443 1448 1449 1454 1512 1513
                              1517 1524 1547 1565 1573 1594 1667 1674 1677 1707 1728 1745 1746 1771 1773 1799 1942 1966
                              1971 1972
R14SAVE  00004   000390 0878  0956 1448 1942 1971
R14SAVE2 00004   000DE4 1747  1677 1745
R14SAVE3 00004   000BD0 1518  1454 1513
R15      00001 0000000F 1894  0541 0541 0599 0599 0666 0666 0691 0691 0717 0717 0787 0787 0867 0867 0894 0945 0946 0954
                              0968 0968 1441 1441 1485 1514 1514 1665 1665 1931 1932 1940 1964 1964
R3       00001 00000003 1882  0520 0521 0944 0947 1930 1933
R4       00001 00000004 1883  0521 0522 0944 0948 1930 1934
R5       00001 00000005 1884  0521 0523 0944 0949 1930 1935
R7       00001 00000007 1886  0727 0728 0730 1137 1138 1140 1143 1144 1144 1145 1162 1163 1163 1164 1179 1180 1181 1192
                              1193 1194 1197 1197 1198 1199 1199 1201 1202 1202 1204 1206 1210 1211 1213 1218 1221 1222
                              1224 1241 1242 1242 1243 1247 1248 1250 1255 1256 1256 1257 1261 1262 1262 1263 1274 1275
                              1275 1276 1280 1281 1283 1288 1289 1289 1290 1294 1299 1300 1302 1307 1308 1308 1309 1313
                              1345 1346 1348 1373 1374 1374 1375 1483 1484 1502 1503 1520 1521 1522 1522 1554 1555 1556
                              1557 1558 1559 1562 1563 1569 1570 1570 1571 1580 1581 1582 1583 1584 1585 1590 1591 1593
                              1599 1600 1600 1601 1714 1715 1716 1717 1718 1719 1724 1725 1727 1733 1734 1734 1735 1767
                              1768 1770 1795 1796 1796 1797
R8       00001 00000008 1887  0728 1138 1193 1195 1196 1197 1200 1200 1201 1202 1211 1222 1248 1281 1300 1346 1455 1455
                              1456 1458 1460 1462 1474 1591 1725 1768
R9       00001 00000009 1888  0729 0729 0730 0828 0829 0831 0832 0983 0984 0995 0996 1001 1002 1013 1014 1139 1139 1140
                              1160 1161 1166 1167 1212 1212 1213 1223 1223 1224 1249 1249 1250 1282 1282 1283 1301 1301
                              1302 1347 1347 1348 1371 1372 1384 1385 1475 1592 1592 1593 1726 1726 1727 1769 1769 1770
SBRNULL  00004   000BD4 1520  1329 1340 1361
SBRSAVE  00004   000394 0879  0957 1943
SBRSEND  00004   000B0C 1454  0731 1141 1214 1225 1251 1284 1303 1349 1594 1728 1771
SEEKID   00004   00058C 0983  0978 1027
SENDOK   00004   000B34 1471  1457 1459 1461
STEPTBL  00002   000F12 1835  1544 1609 1704 1743
STEPVAL  00002   000ECC 1812  1545 1551 1569 1571 1577 1599 1601 1608 1697 1703 1711 1733 1735 1742 1795 1797
STOPLOOP 00006   00031A 0838  0857
TRUE     00001 000000FF 1896  0724 1341 1362 1761 1792
UDATA01  00004   000FFA 1982  1846 1854
UPCASE   00006   00009C 0567  0565
WANTCNID 00006   00057E 0979  0975 1031
WANTED   00002   000FA0 1912  0976 0977 0979 0980 0981 0993 1011 1022 1024 1025 1026 1030
WORKC    00001   00050C 0931  0897 0898 0899 1317 1318 1319 1320 1781 1782 1783
WORKN    00004   000508 0930  0896 0897 1316 1317 1780 1781
WTODIAG  00004   000964 1323  1319 1320
WTOENDS  00004   000E5C 1786  1783
WTOMAC   00004   00046C 0902  0899
WTONAME  00004   0000A8 0571  0568
WTORECB  00004   000438 0886  0547 0547 0554 0561 0803 0803 0810 0817 0838 0838 0845 0852
=A(BEGINCMD)
         00004   000F78 1867  1556 1582 1716
=A(CALLCNT)
         00004   000F80 1869  1775
=A(CTCPSTR)
         00004   000F70 1865  0895
=A(ENDCMD)
         00004   000F7C 1868  1558 1584 1718
=C' '    00001   000F91 1876  0734 1388 1612
=C'DU3'  00003   000F8E 1875  0566
                                             CROSS REFERENCE
```

```
SYMBOL    LEN    VALUE  DEFN  REFERENCES                                                                  ASM H V 02 16.17 09/27/88
=C'01'   00002   000F84 1870  0804
=H'1'    00002   000F8C 1874  1697
=H'4'    00002   000F88 1872  1485
=H'8'    00002   000F8A 1873  1487
=X'F000' 00002   000F86 1871  1269 1381
=X'00000FFF'
         00004   000F74 1866  1521
                                  DIAGNOSTIC CROSS REFERENCE AND ASSEMBLER SUMMARY
                                                                                                      ASM H V 02 16.17 09/27/88
     NO STATEMENTS FLAGGED IN THIS ASSEMBLY
   OVERRIDING PARAMETERS-  OBJECT,NODECK,XREF(SHORT)
   OPTIONS FOR THIS ASSEMBLY
    NODECK, OBJECT, LIST, XREF(SHORT), NORENT, NOTEST, NOBATCH, ALIGN, ESD, RLD, NOTERM, LINECOUNT(55),
    FLAG(0), SYSPARM()
   NO OVERRIDING DD NAMES
    1096 CARDS FROM SYSIN   12525 CARDS FROM SYSLIB
    1992 LINES OUTPUT         113 CARDS OUTPUT
```

# Abbreviations, Glossary, Bibliography, and Index

# List of Abbreviations

| | |
|---|---|
| **AEF** | address extension facility |
| **API** | application programming interface |
| **APPL** | application program |
| **ASCII** | American National Standard Code for Information Interchange |
| **CC** | command code |
| **CCITT** | Consultative Committee on International Telegraph and Telephone |
| **CD** | change direction |
| **CD-ROM** | compact disk read-only memory |
| **CICS/VS** | Customer Information Control System/Virtual Storage |
| **CMD** | command |
| **CNID** | connection identifier |
| **CPL** | CTCP pseudo-link |
| **CPS** | complete packet sequence |
| **CPS/MBS** | complete packet sequence M-bit sequence |
| **CSFI** | Communication Subsystem for Interconnection |
| **CSS** | connectivity subsystem |
| **CTCP** | communication and transmission control program |
| **CUD** | call user data |
| **CV** | control vector |
| **DATE** | dedicated access to X.25 transport extension |
| **DCE** | data circuit-terminating equipment |
| **DCI** | diagnostic code indicator |
| **DFC** | data flow control |
| **DISC** | disconnect |
| **DM** | disconnect mode |
| **DSMN** | dynamic switched major node |
| **DTE** | data terminal equipment |
| **EIB** | execute interface block |
| **ENP** | enable presentation (characters) |
| **FC** | fast connect |
| **FIC** | first-in-chain |
| **FTPI** | Fast Transaction Processing Interface |
| **GATE** | general access to X.25 transport extension |
| **IDBLK** | identification block |
| **IDNUM** | ID number |
| **IMS/VS** | Information Management System/Virtual Storage |
| **INB** | in-bracket |
| **INOP** | inoperative |
| **INP** | inhibit presentation (characters) |
| **ISDN** | integrated services digital network |
| **ISO** | International Organization for Standardization |
| **LIC** | last-in-chain |
| **LLC** | logical link control |
| **LU** | logical unit |
| **LUSIM** | LU simulator |
| **MCH** | multichannel link |
| **MIC** | middle-in-chain |
| **MMN** | model major node |
| **NCP** | Network Control Program |
| **NEP** | node error program |
| **NIA** | Network Interface Adapter |
| **NIB** | node initialization block |
| **NPM** | NetView Performance Monitor |
| **NPSI** | X.25 Network Control Program Packet Switching Interface |
| **NTO** | Network Terminal Option |
| **OIC** | only-in-chain |
| **OSI** | Open Systems Interconnection |
| **OSICS** | Open Systems Interconnection Communication System |
| **PAD** | packet assembler/disassembler |
| **PAV** | PAD for Videotext terminals |
| **PCNE** | protocol converter for non-SNA equipment |
| **PIU** | path information unit |
| **PLP** | packet level processor |
| **PLU** | primary logical unit |
| **POI** | program operator interface |
| **PRPQ** | programming request for price quotation |
| **PSDN** | packet switching data network |
| **PSH** | physical services header |
| **PU** | physical unit |
| **PVC** | permanent virtual circuit |
| **QLLC** | qualified logical link control |

| | |
|---|---|
| **QOS** | quality of service |
| **RECFMS** | record formatted maintenance statistics |
| **req.** | request |
| **REQCONT** | request contact |
| **RESID** | resource identifier |
| **RH** | request/response header |
| **RNR** | receive not ready (packet) |
| **RPL** | request parameter list |
| **RPQ** | request for price quote |
| **RR** | receive ready (packet) |
| **RU** | request/response unit |
| **SABM** | set asynchronous balanced mode |
| **SDLC** | Synchronous Data Link Control |
| **SDT** | start data traffic |
| **SHM** | short hold mode |
| **SLU** | secondary logical unit |
| **SMN** | switched major node |
| **SNA** | Systems Network Architecture |
| **SNI** | SNA network interconnection |
| **SSCP** | system services control point |
| **SVC** | switched virtual circuit |
| **SVCSC** | switched virtual circuit subarea communication |
| **TCP** | Terminal Control Program |
| **TCT** | terminal control table |
| **TH** | transmission header |
| **TIOA** | terminal input/output area |
| **TSC** | telecommunications subsystem controller |
| **TSO** | Time Sharing Option |
| **TSS** | transmission subsystem |
| **UA** | unnumbered acknowledgment |
| **USS** | unformatted system services |
| **VCI** | virtual circuit identifier |
| **VCID** | virtual circuit identifier |
| **VTAM** | Virtual Telecommunications Access Method |
| **XID** | exchange identification |

# Glossary

This glossary includes terms and definitions from:

- The *American National Standard Dictionary for Information Systems*, ANSI X3.172-1990, copyright 1990 by the American National Standards Institute (ANSI). Copies may be purchased from the American National Standards Institute, 11 West 42nd Street, New York, New York 10036. Definitions are identified by the symbol (A) after the definition.
- The ANSI/EIA Standard—440-A, *Fiber Optic Terminology*. Copies may be purchased from the Electronic Industries Association, 2001 Pennsylvania Avenue, N.W., Washington, DC 20006. Definitions are identified by the symbol (E) after the definition.
- The *Information Technology Vocabulary*, developed by Subcommittee 1, Joint Technical Committee 1, of the International Organization for Standardization and the International Electrotechnical Commission (ISO/IEC JTC1/SC1). Definitions of published parts of this vocabulary are identified by the symbol (I) after the definition; definitions taken from draft international standards, committee drafts, and working papers being developed by ISO/IEC JTC1/SC1 are identified by the symbol (T) after the definition, indicating that final agreement has not yet been reached among the participating National Bodies of SC1.
- The *IBM Dictionary of Computing*, New York: McGraw-Hill, 1994.
- Internet Request for Comments: 1208, *Glossary of Networking Terms*
- Internet Request for Comments: 1392, *Internet Users' Glossary*
- The *Object-Oriented Interface Design: IBM Common User Access Guidelines*, Carmel, Indiana: Que, 1992.

The following cross-references are used in this glossary:

**Contrast with:** This refers to a term that has an opposed or substantively different meaning.

**Synonym for:** This indicates that the term has the same meaning as a preferred term, which is defined in its proper place in the glossary.

**Synonymous with:** This is a backward reference from a defined term to all other terms that have the same meaning.

**See:** This refers the reader to multiple-word terms that have the same last word.

**See also:** This refers the reader to terms that have a related, but not synonymous, meaning.

**Deprecated term for:** This indicates that the term should not be used. It refers to a preferred term, which is defined in its proper place in the glossary.

## A

**abend**. (1) Abnormal end of task; the termination of a task before its completion because of an error condition that cannot be resolved by recovery facilities while the task is executing. (2) Synonym for *abnormal termination*.

**abnormal termination**. (1) The cessation of processing prior to planned termination. (T) (2) A system failure or operator action that causes a job to end unsuccessfully. (3) Synonymous with *abend*.

**ACB**. (1) In VTAM, access method control block. (2) In NCP, adapter control block. (3) Application control block.

**ACB name**. (1) The label of an ACB macroinstruction used to name the ACB. (2) A name specified either on the VTAM APPL definition statement or on the VTAM application program's ACB macroinstruction. Contrast with *network name*.

**accept**. (1) In a VTAM application program, to establish a session with a logical unit (LU) in response to a CINIT request from a system services control point (SSCP). The session-initiation request may begin when a terminal user logs on, a VTAM application program issues a macroinstruction, or a VTAM operator issues a command. See also *acquire*. (2) An SMP process that moves distributed code and MVS-type programs to the distribution libraries.

**ACCESS**. In the Simple Network Management Protocol (SNMP), the clause in a Management Information Base (MIB) module that defines the minimum level of support that a managed node provides for an object.

**access control**. In computer security, ensuring that the resources of a computer system can be accessed only by authorized users in authorized ways.

**access method**. (1) A technique, implemented in software, that controls the flow of information through a network. (2) A technique for moving data between main storage and input/output devices.

**access method control block (ACB)**. A control block that links an application program to VSAM or VTAM.

**ACF/NCP**. Advanced Communications Function for the Network Control Program. Synonym for *NCP*.

**ACF/VTAM**. Advanced Communications Function for the Virtual Telecommunications Access Method. Synonym for *VTAM*.

**acknowledgment**. (1) The transmission, by a receiver, of acknowledge characters as an affirmative response to a sender. (T) (2) An indication that an item sent was received.

**acquire**. (1) In VTAM, to take over resources that were formerly controlled by an access method in another domain or to resume control of resources that were controlled by that domain but released. Contrast with *release*. See also *resource takeover*. (2) In a VTAM application program, to initiate and establish a session with another logical unit (LU). The acquire process begins when the application program issues a macroinstruction. See also *accept*.

**action**. (1) An operation on a managed object, the semantics of which are defined as part of the managed object class definition. (2) In the AIX operating system, a defined task that an application performs. An action modifies the properties of an object or manipulates the object in some way.

**activate**. To make a resource ready to perform its function. Contrast with *deactivate*.

**active**. (1) Operational. (2) Pertaining to a node or device that is connected or is available for connection to another node or device. (3) The state of a resource when it has been activated and is operational. (4) In the AIX operating system, pertaining to the window pane in which the text cursor is currently positioned. (5) In VTAM, pertaining to a major or minor node that has been activated by VTAM. Most resources are activated as part of VTAM start processing or as the result of a VARY ACT command. (6) Contrast with *inactive* and *inoperative*. (7) See also *pending active session*.

**ACTLU**. Activate logical unit. In SNA, a command used to start a session on a logical unit.

**adapter**. A part that electrically or physically connects a device to a computer or to another device.

**adapter control block (ACB)**. In NCP, a control block that contains line control information and the states of I/O operations for BSC lines, SS lines, or SDLC links.

**adaptive pacing**. Synonym for *adaptive session-level pacing*.

**adaptive session-level pacing**. A form of session-level pacing in which session components exchange pacing windows that may vary in size during the course of a session. This allows transmission within a network to adapt dynamically to variations in availability and demand of buffers on a session-by-session basis. Session-level pacing occurs within independent stages along the session path according to local congestion at the intermediate and endpoint nodes. Synonymous with *adaptive pacing* and *adaptive session pacing*. Contrast with *fixed session-level pacing*.

**adaptive session pacing**. Synonym for *adaptive session-level pacing*.

**address**. In data communication, the unique code assigned to each device or workstation connected to a network.

**address space manager (ASM)**. A component in an APPN or LEN node that assigns and frees session addresses.

**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.

**Administrative Domain**. A collection of hosts and routers, and the interconnecting networks, managed by a single administrative authority.

**Advanced Peer-to-Peer Networking (APPN)**. An extension to SNA featuring (a) greater distributed network control that avoids critical hierarchical dependencies, thereby isolating the effects of single points of failure; (b) dynamic exchange of network topology information to foster ease of connection, reconfiguration, and adaptive route selection; (c) dynamic definition of network resources; and (d) automated resource registration and directory lookup. APPN extends the LU 6.2 peer orientation for end-user services to network control and supports multiple LU types, including LU 2, LU 3, and LU 6.2.

**Advanced Peer-to-Peer Networking (APPN) end node**. A node that provides a broad range of end-user services and supports sessions between its local control point (CP) and the CP in an adjacent network node. It uses these sessions to dynamically register its resources with the adjacent CP (its network node server), to send and receive directory search requests, and to obtain management services. An APPN end node can also attach to a subarea network as a peripheral node or to other end nodes.

**Advanced Peer-to-Peer Networking (APPN) network**. A collection of interconnected network nodes and their client end nodes.

**Advanced Peer-to-Peer Networking (APPN) network node**. A node that offers a broad range of end-user services and that can provide the following:

- Distributed directory services, including registration of its domain resources to a central directory server
- Topology database exchanges with other APPN network nodes, enabling network nodes throughout the network to select optimal routes for LU-LU sessions based on requested classes of service
- Session services for its local LUs and client end nodes
- Intermediate routing services within an APPN network

**Advanced Peer-to-Peer Networking (APPN) node**. An APPN network node or an APPN end node.

**AID**. Attention identifier.

**all points addressable (APA)**. In computer graphics, pertaining to the ability to address and display or not display each picture element (pel) on a display surface.

**ALS**. Adjacent link station.

**analog**. (1) Pertaining to data consisting of continuously variable physical quantities. (A) (2) Contrast with *digital*.

**AND operation**. Synonym for *conjunction*.

**any-mode**. In VTAM, the following:

- The form of a RECEIVE request that obtains input from any one (unspecified) session
- The form of an ACCEPT request that completes the establishment of a session by accepting any one (unspecified) queued CINIT request.

Contrast with *specific-mode*. See *continue-any mode*.

**APA**. All points addressable.

**API**. Application programming interface.

**APPL**. Application program.

**application**. A collection of software components used to perform specific types of user-oriented work on a computer.

**application control block (ACB)**. The control blocks created from the output of DBDGEN and PSBGEN and placed in the ACB library for use during online and DBB region type execution of IMS/VS.

**application program**. (1) A program written for or by a user that applies to the user's work, such as a program that does inventory control or payroll. (2) A program used to connect and communicate with stations in a network, enabling users to perform application-oriented activities.

**application program interface**. See *application programming interface (API)*.

**application programming interface (API)**. (1) The set of programming language constructs or statements that can be coded in an application program to obtain the specific functions and services provided by an underlying operating system or service program. (2) In VTAM, the language structure used in control blocks so that application programs can reference them and be identified to VTAM.

**apply**. An SMP process that moves distributed code and MVS-type programs to the system libraries.

**Apply**. A push button that carries out the selected choices in a window without closing the window.

**APPN**. Advanced Peer-to-Peer Networking.

**area**. In Internet and DECnet routing protocols, a subset of a network or gateway grouped together by definition of the network administrator. Each area is self-contained; knowledge of an area's topology remains hidden from other areas.

**ASCII (American National Standard Code for Information Interchange)**. The standard code, using a coded character set consisting of 7-bit coded characters (8 bits including parity check), that is used for information interchange among data processing systems, data communication systems, and associated equipment. The ASCII set consists of control characters and graphic characters. (A)

**ASM**. Address space manager.

**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.

**asynchronous balanced mode (ABM)**. An operational mode of a balanced data link in which either combined station can send commands at any time and can initiate transmission of response frames without explicit permission from the other combined station.

**asynchronous request**. In VTAM, a request for an asynchronous operation. Contrast with *synchronous request*.

**attention identifier (AID)**. A character in a data stream indicating that the user has pressed a key, such as the Enter key, that requests an action by the system.

**attribute**. (1) A characteristic that identifies and describes a managed object. The characteristic can be determined, and possibly changed, through operations on the managed object. (2) Information within a managed object that is visible at the object boundary. An attribute has a type, which indicates the range of information given by the attribute, and a value, which is within that range. (3) Variable data that is logically a part of an object and that represents a property of the object. For example, a serial number is an attribute of an equipment object.

**autologon**. Synonym for *automatic logon*.

**automated operator**. In IMS/VS, an application program that can issue a subset of IMS/VS operator commands and receive status information on the execution of the commands.

**automatic logon**. (1) A process by which VTAM automatically creates a session-initiation request to establish a session between two logical units (LUs). The session is between a designated primary logical unit (PLU) and a secondary logical unit (SLU) that is neither queued for nor in session with another PLU. Synonymous with *autologon*. See also *controlling application program* and *controlling logical unit*. (2) In VM, a process by which a virtual machine is initiated by other than the user of that virtual machine; for example, the primary VM operator's virtual machine is activated automatically during VM initialization.

**autotask**. An unattended NetView operator station task that does not require a terminal or a logged-on user. Autotasks can run independently of VTAM and are typically used for automated console operations. Contrast with *logged-on operator*.

**available**. In VTAM, pertaining to a logical unit that is active, connected, enabled, and not at its session limit.

# B

**backbone**. (1) A set of nodes and their interconnecting links providing the primary data path across a network. (2) In a local area network multiple-bridge ring configuration, a high-speed link to which the rings are connected by means of bridges or routers. A backbone may be configured as a bus or as a ring. (3) In a wide area network, a high-speed link to which nodes or data switching exchanges (DSEs) are connected.

**backbone network**. A central network to which smaller networks, normally of lower speed, connect. The backbone network usually has a much higher capacity than the networks it helps interconnect or is a wide-area network (WAN) such as a public packet-switched datagram network.

**batch**. (1) An accumulation of data to be processed. (2) A group of records or data processing jobs brought together for processing or transmission. (3) Pertaining to activity involving little or no user action. Contrast with *interactive*.

**begin bracket**. In SNA, the value (binary 1) of the begin-bracket indicator in the request header (RH) of the first request in the first chain of a bracket; the value denotes the start of a bracket. Contrast with *end bracket*. See also *bracket*.

**behavior**. (1) Ideally, a collection of assertions that describe the allowed states that a managed object can assume. An assertion can be a precondition, a postcondition, or an invariant. In practice, the behavior is often an informal description of the semantics of attributes, operations, and notifications. (2) The way in which managed objects, name bindings, attributes, notifications, and operations interact with the actual resources that they model and with each other.

**bid**. In the contention form of invitation or selection, an attempt by the computer or by a station to gain control of a line in order to transmit data.

**bidder**. See *bidder session*.

**bidder session**. The half-session defined at session activation as having to request and receive permission from the other half-session to begin a bracket. Contrast with *first-speaker session*. Synonym for *contention-loser session*.

**binary synchronous communication (BSC)**. (1) A form of telecommunication line control that uses a standard set of transmission control characters and control character sequences, for binary synchronous transmission of binary-coded data between stations. (2) Contrast with *Synchronous Data Link Control (SDLC)*.

**BIND**. In SNA, a request to activate a session between two logical units (LUs). See also *session activation request*. Contrast with *UNBIND*.

**block**. A string of data elements recorded or transmitted as a unit. The elements may be characters, words, or physical records. (T)

**Boolean**. (1) Pertaining to the processes used in the algebra formulated by George Boole. (A) (2) A value of 0 or 1 represented internally in binary notation.

**bottleneck**. A hardware or software component that can limit the performance of a device, a subsystem (such as an adapter), or a network. For example, if an adapter has hardware that can forward 14 000 packets per second, and software that can process 4000 packets per second, the packet throughput is limited to 4000 packets per second; and the software is the bottleneck.

**boundary function (BF)**. (1) In SNA, a capability of a subarea node to provide protocol support for attached peripheral nodes, such as: (a) interconnecting subarea path control and peripheral path control elements, (b) performing session sequence numbering for low-function peripheral nodes, and (c) providing session-level pacing support. (2) In SNA, the component that provides these capabilities.

**boundary 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.

**bracket**. In SNA, one or more chains of request units and their responses that are exchanged between two session partners and that represent a transaction between them. A bracket must be completed before another bracket can be started. Examples of brackets are database inquiries/replies, update transactions, and remote job entry output sequences to workstations.

**bracket protocol**. In SNA, a data flow control protocol in which exchanges between two session partners are achieved through the use of brackets, with one partner designated at session activation as the first speaker and the other as the bidder. The bracket protocol involves bracket initiation and termination rules.

**bridge**. (1) A functional unit that interconnects two local area networks that use the same logical link control protocol but may use different medium access control protocols. (T) (2) A functional unit that interconnects multiple LANs (locally or remotely) that use the same logical link control protocol but that can use different medium access control protocols. A bridge forwards a frame to another bridge based on the medium access control (MAC) address. (3) In the connection of local loops, channels, or rings, the equipment and techniques used to match circuits and to facilitate accurate data transmission. (4) Contrast with *gateway* and *router*.

**browse**. (1) To look at records in a file. (2) In the NetView Graphic Monitor Facility, to open a view that cannot receive status changes from the NetView program. Contrast with *monitor*.

**BSC**. Binary synchronous communication.

**buffer**. (1) A routine or storage used to compensate for a difference in rate of flow of data, or time of occurrence of events, when transferring data from one device to another. (A) (2) To allocate and schedule the use of buffers. (A) (3) A portion of storage used to hold input or output data temporarily.

**byte**. (1) A string that consists of a number of bits, treated as a unit, and representing a character. (T) (2) A binary character operated upon as a unit and usually shorter than a computer word. (A) (3) A group of 8 adjacent binary digits that represent one EBCDIC character.

# C

**CA**. (1) Channel adapter. (2) Channel attachment.

**call**. (1) The action of bringing a computer program, a routine, or a subroutine into effect, usually by specifying the entry conditions and jumping to an entry point. (I) (A) (2) In data communication, the actions necessary to make a connection between two stations on a switched line. (3) In communications, a conversation between two users. (4) To transfer control to a procedure, program, routine, or subroutine. (5) To attempt to contact a user, regardless of whether the attempt is successful.

**call-accepted packet**. A call supervision packet that a called data terminal equipment (DTE) transmits to indicate to the data circuit-terminating equipment (DCE) that it accepts the incoming call.

**call connected packet**. A call supervision packet that a data circuit-terminating equipment (DCE) transmits to indicate to a calling data terminal equipment (DTE) that the connection for the call has been completely established.

**call establishment**. The complete sequence of events necessary to establish a data connection.

**call request packet**. (1) A call supervision packet that a data terminal equipment (DTE) transmits to ask that a connection for a call be established throughout the network. (2) In X.25 communications, a call supervision packet transmitted by a DTE to ask for a call establishment through the network.

**call user data (CUD)**. In X.25 communications, data optionally included in the call-request packet by the user application.

**CALLIN**. The logical channel type on which the data terminal equipment (DTE) can receive a call, but cannot send one.

**calling**. (1) The process of transmitting selection signals in order to establish a connection between data stations. (I) (A) (2) In X.25 communications, pertaining to the location or user that makes a call.

**calling party**. On a switched line, the location that originates a connection.

**CALLOUT**. The logical channel type on which the data terminal equipment (DTE) can send a call, but cannot receive one.

**Cancel**. A push button that removes a window without applying any changes made in that window.

**casual connection**. (1) In a subarea network, a connection in which type 5 nodes are attached through the boundary function using low-entry networking (LEN). Therefore, the nodes appear as LEN nodes rather than subarea nodes. (2) In an APPN network, a connection between an end node and a network node with different network identifiers.

**CCITT**. International Telegraph and Telephone Consultative Committee. This was an organization of the International Telecommunication Union (ITU). On 1 March 1993 the ITU was reorganized, and responsibilities for standardization were placed in a subordinate organization named the Telecommunication Standardization Sector of the International Telecommunication Union (ITU-TS). “CCITT” continues to be used for recommendations that were approved before the reorganization.

**CD**. Compact disc.

**CD-ROM**. High-capacity read-only memory in the form of an optically read compact disc.

**chain**. (1) A group of logically linked user data records processed by LU 6.2. (2) A group of request units delimited by begin-chain and end-chain. Responses are always single-unit chains. See *RU chain*.

**channel**. (1) A path along which signals can be sent, for example, data channel, output channel. (A) (2) In data communication, a means of one-way transmission. (3) A functional unit, controlled by the processor, that handles the transfer of data between processor storage and local peripheral equipment.

**channel-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*.

**character-coded**. Synonym for *unformatted*.

**checkpoint**. (1) A sequence of instructions in a computer program for recording the status of execution for restarting. (T) (2) A point at which information about the status of a job and the system can be recorded so that the job step can be later restarted. (3) To record such information.

**CICS**. Customer Information Control System.

**CICS/VS**. Customer Information Control System for Virtual Storage.

**circuit**. (1) One or more conductors through which an electric current can flow. See *physical circuit* and *virtual circuit*. (2) A logic device.

**circuit switching**. (1) A process that, on demand, connects two or more data terminal equipment (DTEs) and permits the exclusive use of a data circuit between them until the connection is released. (I) (A) (2) Synonymous with *line switching*. (3) See also *message switching* and *packet switching*.

**class**. (1) In object-oriented design or programming, a group of objects that share a common definition and that therefore share common properties, operations, and behavior. Members of the group are called instances of the class. (2) In the AIX operating system, pertaining to the I/O characteristics of a device. System devices are classified as block or character devices.

**class of service (COS)**. A set of characteristics (such as route security, transmission priority, and bandwidth) used to construct a route between session partners. The class of service is derived from a mode name specified by the initiator of a session.

**clear confirmation packet**. See *DCE clear confirmation packet*.

**clear indication packet**. A call supervision packet that a data circuit-terminating equipment (DCE) transmits to inform a data terminal equipment (DTE) that a call has been cleared.

**clear request packet**. A call supervision packet transmitted by a data terminal equipment (DTE) to ask that a call be cleared.

**CLI**. Command line interface.

**Close**. A choice that removes a window and all of the windows associated with it from the workplace. For example, if a user is performing a task in a window and a message appears, or the user asks for help, both the message and the help windows disappear when the user closes the original window.

**CLSDST**. Close destination.

**collision**. An unwanted condition that results from concurrent transmissions on a channel. (T)

**command**. (1) A request from a terminal for the performance of an operation or the execution of a particular program. (2) In SNA, any field set in the transmission header (TH), request header (RH), and sometimes portions of a request unit (RU), that initiates an action or that begins a protocol; for example: (a) Bind Session (session-control request unit), a command that activates an LU-LU session, (b) the change-direction indicator in the RH of the last RU of a chain, (c) the virtual route reset window indicator in an FID4 transmission header. (3) See also *VTAM operator command*.

**command facility**. The component of the NetView program that is a base for command processors that can monitor, control, automate, and improve the operation of a network.

**command line interface (CLI)**. In Communications Manager/2, the capability to enter Communications Manager/2 commands from the operating system's command line.

**command list**. In the NetView program, a list of commands and statements designed to perform a specific function for the user. Command lists can be written in REXX or in the NetView command list language.

**common operations services (COS)**. The portion of SNA management services that pertains to the major vectors for limited remote operations control.

**communication controller**. A type of communication control unit whose operations are controlled by one or more programs stored and executed in the unit. It manages the details of line control and the routing of data through a network.

**compact disc (CD)**. (1) A disc, usually 4.75 inches in diameter, from which data is read optically by means of a laser. (2) A disc with information stored in the form of pits along a spiral track. The information is decoded by a compact-disc player and interpreted as digital audio data, which most computers can process.

**component**. (1) Hardware or software that is part of a functional unit. (2) A part of a structured type or value, such as an array element or a record field.

**compression**. (1) The process of eliminating gaps, empty fields, redundancies, and unnecessary data to shorten the length of records or blocks. (2) Any encoding to reduce the number of bits used to represent a given message or record.

**computer word**. A word suitable for processing by a given computer, usually treated as a unit. (T) Synonymous with *fullword*.

**configuration**. (1) The manner in which the hardware and software of an information processing system are organized and interconnected. (T) (2) The devices and programs that make up a system, subsystem, or network. (3) In CCP, the arrangement of controllers, lines, and terminals attached to an IBM 3710 Network Controller. Also, the collective set of item definitions that describe such a configuration.

**configuration services**. One of the types of network services in a control point (SSCP, NNCP, or ENCP). Configuration services activates, deactivates, and records the status of physical units, links, and link stations.

**conjunction**. The Boolean operation whose result has the Boolean value 1 if and only if each operand has the Boolean value 1. (I) (A) Synonymous with *AND operation*.

**connected**. In VTAM, the state of a physical unit (PU) or a logical unit (LU) that has an active physical path to the host processor containing the system services control point (SSCP) that controls the respective PU or LU.

**connection**. (1) In data communication, an association established between functional units for conveying information. (I) (A) (2) In Open Systems Interconnection architecture, an association established by a given layer between two or more entities of the next higher layer for the purpose of data transfer. (T) (3) In SNA, the network path that links together two logical units (LUs) in different nodes to enable them to establish communications. (4) In X.25 communication, a virtual circuit between two data terminal equipments (DTEs). A switched virtual circuit (SVC) connection lasts for the duration of a call; a permanent virtual circuit (PVC) is a permanent connection between the DTEs. (5) In TCP/IP, the path between two protocol applications that provides reliable data stream delivery service. In the Internet, a connection extends from a TCP application on one system to a TCP application on another system. (6) In system communications, a line over which data can be passed between two systems or between a system and a device. (7) Synonym for *physical connection*.

**connectivity**. (1) The capability of a system or device to be attached to other systems or devices without modification. (T) (2) The capability to attach a variety of functional units without modifying them.

**connectivity subsystem (CSS)**. An expansion frame, such as the 3746 Model 900, that extends connectivity and enhances the performance of the IBM 3745 Communication Controller.

**constraints**. In NETDA/2, the set of essential requirements specified with the node, connection, or application definitions. A change in a constraint value changes the input to the network design. Contrast with *parameters*.

**contention**. In a session, a situation in which both NAUs attempt to initiate the same action at the same time, such as when both attempt to send data in a half-duplex protocol (half-duplex contention), or both attempt to start a bracket (bracket contention). At session initiation, one 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.

**contention-loser session**. To an NAU, a session for which it was defined during session initiation to be the contention loser. Synonymous with *bidder session*.

**contention-winner session**. To an NAU, a session for which it was defined during session initiation to be the contention winner. Synonymous with *first-speaker session*.

**continue-any mode**. In VTAM, the state of a session or conversation that allows its input to satisfy a RECEIVE request issued in any-mode. While this state exists, input on the session or conversation can also satisfy RECEIVE requests issued in specific-mode. For conversations, continue-any mode is further qualified as either buffer continue-any or logical record continue-any. This specifies whether VTAM is to receive the data in terms of logical records or buffers. Contrast with *continue-specific mode*.

**continue-specific mode**. In VTAM, the state of a session or conversation that allows its input to satisfy only RECEIVE requests issued in specific-mode. Contrast with *continue-any mode*.

**control block**. (1) A storage area used by a computer program to hold control information. (I) (2) In the IBM Token-Ring Network, a specifically formatted block of information provided from the application program to the Adapter Support Interface to request an operation.

**control character**. A character whose occurrence in a particular context specifies a control function. (T)

**control point (CP)**. (1) A component of an APPN or LEN node that manages the resources of that node. In an APPN node, the CP is capable of engaging in CP-CP sessions with other APPN nodes. In an APPN network node, the CP also provides services to adjacent end nodes in the APPN network. (2) A component of a node that manages resources of that node and optionally provides services to other nodes in the network. Examples are a system services control point (SSCP) in a type 5 subarea node, a network node control point (NNCP) in an APPN network node, and an end node control point (ENCP) in an APPN or LEN end node. An SSCP and an NNCP can provide services to other nodes.

**control program**. (1) A computer program designed to schedule and to supervise the execution of programs of a computer system. (I) (A) (2) The part of the AIX Base Operating System that determines the order in which basic functions should be performed. (3) In VM/ESA, see *Control Program (CP)*.

**Control Program (CP)**. In VM/ESA, a component that manages the resources of a single computer so multiple computing systems appear to exist. Each of these apparent systems, or virtual machines, is the functional equivalent of an IBM System/370, 370-XA, or ESA computer.

**control vector**. One of a general class of RU substructures that has variable length, is carried within some enclosing structure, and has a one-byte key used as an identifier.

**controller**. A device that coordinates and controls the operation of one or more input/output devices, such as workstations, and synchronizes the operation of such devices with the operation of the system as a whole.

**controlling application program**. In VTAM, an application program with which a secondary logical unit (other than an application program) is automatically put in session whenever the secondary logical unit is available. See also *automatic logon* and *controlling logical unit*.

**controlling logical unit**. In VTAM, a logical unit with which a secondary logical unit (other than an application program) is automatically put in session whenever the secondary logical unit is available. A controlling logical unit can be either an application program or a device-type logical unit. See also *automatic logon* and *controlling application program*.

**conversation**. A logical connection between two transaction programs using an LU 6.2 session. Conversations are delimited by brackets to gain exclusive use of a session.

**converted command**. An intermediate form of a character-coded command produced by VTAM through use of an unformatted system services definition table. The format of a converted command is fixed; the unformatted system services definition table must be constructed in such a manner that the character-coded command (as entered by a logical unit) is converted into the predefined, converted command format. See also *unformatted*.

**Copy**. A choice that places a copy of a selected object onto the clipboard.

**COS**. (1) Class of service. (2) Common operations services.

**CP**. (1) Control point. (2) In VM, Control Program.

**cryptographic**. Pertaining to the transformation of data to conceal its meaning. See also *decipher* and *encipher*.

**CSS**. Connectivity subsystem.

**CUD**. Call user data.

**Customer Information Control System (CICS)**. An IBM licensed program that enables transactions entered at remote terminals to be processed concurrently by user-written application programs. It includes facilities for building, using, and maintaining databases.

**Customer Information Control System for Virtual Storage (CICS/VS)**. An IBM licensed program that can be used in a communications network.

# D

**D-bit**. Delivery-confirmation bit. In X.25 communications, the bit in a data packet or call-request packet that is set to 1 if end-to-end acknowledgment (delivery confirmation) is required from the recipient.

**DACTLU**. Deactivate logical unit.

**data**. (1) A re-interpretable representation of information in a formalized manner suitable for communication, interpretation, or processing. Operations can be performed upon data by humans or by automatic means. (T) (2) Any representations such as characters or analog quantities to which meaning is or might be assigned. (A) (3) A representation of facts or instructions in a form suitable for communication, interpretation, or processing by human or automatic means. Data include constants, variables, arrays, and character strings.

**Note:** Programmers make a distinction between instructions and the data they operate on; however, in the usual sense of the word, data includes programs and program instructions.

**data circuit**. (1) A pair of associated transmit and receive channels that provide a means of two-way data communication. (I) (2) In SNA, synonym for *link connection*. (3) See also *physical circuit* and *virtual circuit*.

**Notes:**

1. Between data switching exchanges, the data circuit may include data circuit-terminating equipment (DCE), depending on the type of interface used at the data switching exchange.
2. Between a data station and a data switching exchange or data concentrator, the data circuit includes the data circuit-terminating equipment at the data station end, and may include equipment similar to a DCE at the data switching exchange or data concentrator location.

**data circuit-terminating equipment (DCE)**. In a data station, the equipment that provides the signal conversion and coding between the data terminal equipment (DTE) and the line. (I)

**Notes:**

1. The DCE may be separate equipment or an integral part of the DTE or of the intermediate equipment.
2. A DCE may perform other functions that are usually performed at the network end of the line.

**data communication**. (1) Transfer of data among functional units by means of data transmission according to a protocol. (T) (2) The transmission, reception, and validation of data. (A)

**data flow control (DFC)**. In SNA, a request/response unit (RU) category used for requests and responses exchanged between the data flow control layer in one half-session and the data flow control layer in the session partner.

**data flow control (DFC) layer**. In SNA, the layer within a half-session that controls whether the half-session can send, receive, or concurrently send and receive, request units (RUs); groups related RUs into RU chains; delimits transactions via the bracket protocol; controls the interlocking of requests and responses in accordance with control modes specified at session activation; generates sequence numbers; and correlates requests and responses.

**data flow control (DFC) protocol**. In SNA, the sequencing rules for requests and responses by which network addressable units (NAUs) in a session coordinate and control data transfer and other operations; for example, bracket protocol.

**data flow synchronous (DFSYN) response**. In VTAM, a normal-flow response that is treated as a normal-flow request so that it may be received in sequence with normal-flow requests.

**data link**. In SNA, synonym for *link*.

**data link control (DLC)**. A set of rules used by nodes on a data link (such as an SDLC link or a token ring) to accomplish an orderly exchange of information.

**data link level**. (1) In the hierarchical structure of a data station, the conceptual level of control or processing logic between high level logic and the data link that maintains control of the data link. The data link level performs such functions as inserting transmit bits and deleting receive bits; interpreting address and control fields; generating, transmitting, and interpreting commands and responses; and computing and interpreting frame check sequences. See also *packet level* and *physical level*. (2) In X.25 communications, synonym for *frame level*.

**data network**. An arrangement of data circuits and switching facilities for establishing connections between data terminal equipment. (I)

**data packet**. In X.25 communications, a packet used for the transmission of user data on a virtual circuit at the DTE/DCE interface.

**data set**. (1) Synonym for *file*. (2) Deprecated term for *modem*.

**data stream**. (1) All information (data and control commands) sent over a data link usually in a single read or write operation. (2) A continuous stream of data elements being transmitted, or intended for transmission, in character or binary-digit form, using a defined format.

**data terminal equipment (DTE)**. That part of a data station that serves as a data source, data sink, or both. (I) (A)

**DC**. Data chaining.

**DCE**. (1) Data circuit-terminating equipment. (2) Distributed Computing Environment.

**DCE clear confirmation packet**. A call supervision packet that a data circuit-terminating equipment (DCE) transmits to confirm that a call has been cleared.

**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*.

**decipher**. (1) To convert enciphered data in order to restore the original data. (T) (2) In computer security, to convert ciphertext into plaintext by means of a cipher system. (3) To convert enciphered data into clear data. Contrast with *encipher*. Synonymous with *decrypt*.

**decrypt**. (1) In computer security, to decipher or decode. (2) Synonym for *decipher*. (T)

**default**. Pertaining to an attribute, condition, value, or option that is assumed when none is explicitly specified. (I)

**definite response (DR)**. In SNA, a protocol requested in the form-of-response-requested field of the request header that directs the receiver of the request to return a response unconditionally, whether positive or negative, to that request chain. Contrast with *exception response* and *no response*.

**definition statement**. (1) In VTAM, the statement that describes an element of the network. (2) In NCP, a type of instruction that defines a resource to the NCP. See Figure X-1, Figure X-2, and Figure X-3. See also *macroinstruction*.

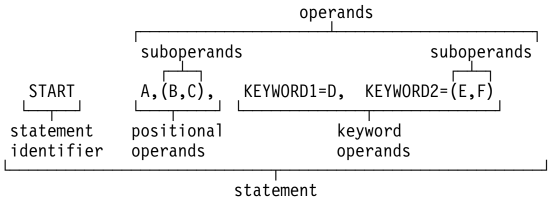


*Figure X-1. Example of a Language Statement*

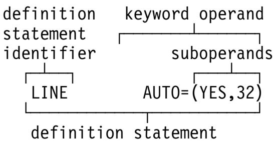


*Figure X-2. Example of an NCP Definition Statement*

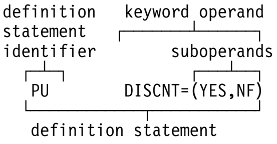


*Figure X-3. Example of a VTAM Definition Statement*

**degree**. In NETDA/2, the maximum number of attachments (such as links, Ethernet connections, or token-ring connections) that a node can have. For example, if a node can have six attachments, that node has a degree of 6.

**Delete**. A choice that removes a selected object. The space it occupied is usually filled by the remaining object or objects in the window.

**delimiter**. (1) A character used to indicate the beginning and end of a character string. (T) (2) A flag that separates and organizes items of data. (A) (3) A character that groups or separates words or values in a line of input.

**delivery-confirmation bit**. See *D-bit*.

**dependent LU**. See *SSCP-dependent LU*.

**destination**. (1) Any point or location, such as a node, station, or a particular terminal, to which information is to be sent. (2) In ACF/TCAM, the place to which a message being handled by a message handler is to be sent. (3) An external logical unit (LU) or application program to which messages or other data are directed.

**device**. (1) A mechanical, electrical, or electronic contrivance with a specific purpose. (2) In the AIX operating system, a valuator, button, or the keyboard.

Buttons have values of 0 or 1 (up or down); valuators return values in a range, and the keyboard returns ASCII values.

**device driver**. (1) A file that contains the code needed to use an attached device. (2) A program that enables a computer to communicate with a specific peripheral device; for example, a printer, a videodisc player, or a CD drive. (3) A collection of subroutines that control the interface between I/O device adapters and the processor.

**DFC**. Data flow control.

**dial-in**. Pertaining to the direction in which a switched connection is requested by any node or terminal other than the receiving host or an NCP.

**dial-out**. Pertaining to the direction in which a switched connection is requested by a host or an NCP.

**digital**. (1) Pertaining to data that consist of digits. (T) (2) Pertaining to data in the form of digits. (A) (3) Contrast with *analog*.

**directory**. (1) A table of identifiers and references to the corresponding items of data. (I) (A) (2) A database in an APPN node that lists names of resources (in particular, logical units) and records the CP name of the node where each resource is located. See *distributed directory database* and *local directory database*. (3) A type of file containing the names and controlling information for other files or other directories. (4) A listing of the files stored on a disk or diskette. (5) In VM, a Control Program (CP) disk file that defines each virtual machine's typical configuration: the user ID, password, regular and maximum allowable virtual storage, CP command privilege class or classes allowed, dispatching priority, logical editing symbols to be used, account number, and CP options desired.

**directory service (DS)**. An application service element that translates the symbolic names used by application processes into the complete network addresses used in an OSI environment. (T)

**directory services (DS)**. A control point component of an APPN node that maintains knowledge of the location of network resources.

**disable**. To make nonfunctional.

**disabled**. (1) Pertaining to a state of a processing unit that prevents the occurrence of certain types of interruptions. (2) Pertaining to the state in which a transmission control unit or audio response unit cannot accept incoming calls on a line. (3) In VTAM, pertaining to a logical unit (LU) that has indicated to its system services control point (SSCP) that it is temporarily not ready to establish LU-LU sessions. An initiate request for a session with a disabled logical unit (LU) can specify that the session be queued by the SSCP until the LU becomes enabled. The LU can separately indicate whether this applies to its ability to act as a primary logical unit (PLU) or a secondary logical unit (SLU). See also *enabled* and *inhibited*.

**DISC**. Disconnect.

**disconnection**. In VTAM, the termination of a physical connection.

**diskette**. (1) A small magnetic disk enclosed in a jacket. (T) (2) A thin, flexible magnetic disk and a semi-rigid protective jacket, in which the disk is permanently enclosed.

**display**. (1) A visual presentation of data. (I) (A) (2) To present data visually. (I) (A) (3) Deprecated term for *panel*.

**display panel**. In computer graphics, a predefined display image that defines the locations and characteristics of display fields on a display surface.

**Distributed Computing Environment (DCE)**. The Open Software Foundation (OSF) specification (or a product derived from this specification) that assists in networking. DCE provides such functions as authentication, directory service (DS), and remote procedure call (RPC).

**distributed directory database**. The complete listing of all the resources in the network as maintained in the individual directories scattered throughout an APPN network. Each node has a piece of the complete directory, but it is not necessary for any one node to have the entire list. Entries are created, modified, and deleted through system definition, operator action, automatic registration, and ongoing network search procedures. Synonymous with *distributed network directory* and *network directory database*.

**distributed network directory**. Synonym for *distributed directory database*.

**DM**. Disconnect mode.

**domain**. (1) That part of a computer network in which the data processing resources are under common control. (T) (2) In SNA, see *end node domain*, *network node domain*, and *system services control point (SSCP) domain*. (3) In Open Systems Interconnection (OSI), a part of a distributed system or a set of managed objects to which a common policy applies. (4) In a database, all the possible values of an attribute or a data element. (5) See *Administrative Domain* and *domain name*.

**domain name**. In the Internet suite of protocols, a name of a host system. A domain name consists of a sequence of subnames separated by a delimiter character. For example, if the fully qualified domain name

(FQDN) of a host system is `ralvm7.vnet.ibm.com`, each of the following is a domain name:

- `ralvm7.vnet.ibm.com`
- `vnet.ibm.com`
- `ibm.com`

**domain operator**. In a multiple-domain network, the person or program that controls operation of resources controlled by one system services control point (SSCP). See also *network operator*.

**downstream**. In the direction of data flow from the host to the end user. Contrast with *upstream*.

**DR**. (1) In VTAM, NCP, and CCP, dynamic reconfiguration. (2) In SNA, definite response.

**drive**. A peripheral device, especially one that has addressed storage media.

**driver**. See *device driver*.

**drop**. In the IBM Cabling System, a cable that runs from a faceplate to the distribution panel in a wiring closet. When the IBM Cabling System is used with the IBM Token-Ring Network, a drop may form part of a lobe. Cables between wiring closets are not classified as drops.

**DS**. (1) Desired state. (2) Directory service. (3) Directory services.

**DTE**. Data terminal equipment. (A)

**dump**. (1) To record, at a particular instant, the contents of all or part of one storage device in another storage device. Dumping is usually for the purpose of debugging. (T) (2) Data that has been dumped. (T) (3) To copy data in a readable format from main or auxiliary storage onto an external medium such as tape, diskette, or printer. (4) To copy the contents of all or part of virtual storage for the purpose of collecting error information.

**duplex**. Pertaining to communication in which data can be sent and received at the same time. Synonymous with *full-duplex*. Contrast with *half-duplex*.

**dynamic**. (1) In programming languages, pertaining to properties that can only be established during the execution of a program; for example, the length of a variable-length data object is dynamic. (I) (2) Pertaining to an operation that occurs at the time it is needed rather than at a predetermined or fixed time. (3) Contrast with *static*.

**dynamic reconfiguration (DR)**. The process of changing the network configuration (peripheral PUs and LUs) without regenerating complete configuration tables or deactivating the affected major node.

# E

**EBCDIC**. Extended binary-coded decimal interchange code. A coded character set of 256 8-bit characters.

**EC**. Engineering change.

**ECB**. Event control block.

**echo**. (1) In computer graphics, the immediate notification of the current values provided by an input device to the operator at the display console. (I) (A) (2) In word processing, to print or display each character or line as it is keyed in. (3) In data communication, a reflected signal on a communications channel. On a communications terminal, each signal is displayed twice, once when entered at the local terminal and again when returned over the communications link. This allows the signals to be checked for accuracy.

**EIB**. External interrupt 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*.

**Emulation Program (EP)**. (1) An IBM control program that allows a channel-attached IBM communication controller to emulate the functions of an IBM 2701 Data Adapter Unit, an IBM 2702 Transmission Control, or an IBM 2703 Transmission Control. (2) See also *network control program*.

**enable**. To make functional.

**enabled**. (1) Pertaining to a state of the processing unit that allows the occurrence of certain types of interruptions. (2) Pertaining to the state in which a transmission control unit or an audio response unit can accept incoming calls on a line. (3) In VTAM, pertaining to a logical unit (LU) that has indicated to its system services control point (SSCP) that it is ready to establish LU-LU sessions. The LU can separately indicate whether this prevents it from acting as a primary logical unit (PLU) or a secondary logical unit (SLU). See also *disabled* and *inhibited*.

**encipher**. (1) To scramble data or to convert data to a secret code that masks the meaning of the data to any unauthorized recipient. (2) In VTAM, to convert clear data into enciphered data. Contrast with *decipher*. Synonymous with *encrypt*.

**encrypt**. Synonym for *encipher*. (T)

**end bracket**. In SNA, the value (binary 1) of the end bracket indicator in the request header (RH) of the first request of the last chain of a bracket; the value denotes the end of the bracket. Contrast with *begin bracket*. See also *bracket*.

**end node domain**. An end node control point, its attached links, and its local LUs.

**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.

**end-of-transmission (EOT) code**. The character or sequence of characters that indicates termination of sending.

**end-of-transmission (EOT) handshaking**. When a 3710 sends EOT characters over an idle line and waits for return characters. If no EOT response is returned, the 3710 breaks the session.

**end user**. (1) A person, device, program, or computer system that utilizes a computer network for the purpose of data processing and information exchange. (T) (2) The ultimate source or destination of application data flowing through an SNA network. An end user can be an application program or a workstation operator.

**entry point (EP)**. (1) The address or label of the first instruction executed on entering a computer program, routine, or subroutine. A computer program, routine, or subroutine may have a number of different entry points, each perhaps corresponding to a different function or purpose. (I) (A) (2) In SNA, a type 2.0, type 2.1, type 4, or type 5 node that provides distributed network management support. It sends network management data about itself and the resources it controls to a focal point for centralized processing, and it receives and executes focal-point initiated commands to manage and control its resources.

**EOT**. The end-of-transmission character. (A)

**EP**. (1) Emulation Program. (2) Entry point.

**equipment**. See *data terminal equipment (DTE).*

**ER**. (1) Explicit route. (2) Exception response.

**error**. A discrepancy between a computed, observed, or measured value or condition and the true, specified, or theoretically correct value or condition. (I) (A)

**event**. (1) An occurrence of significance to a task; for example, the completion of an asynchronous operation, such as an input/output operation. (2) In the NetView program, a record indicating irregularities of operation in physical elements of a network.

**event control block (ECB)**. A control block used to represent the status of an event.

**exception**. An abnormal condition such as an I/O error encountered in processing a data set or a file.

**exception response (ER)**. In SNA, a protocol requested in the form-of-response-requested field of a request header that directs the receiver to return a response only if the request is unacceptable as received or cannot be processed; that is, a negative response, but not a positive response, can be returned. Contrast with *definite response* and *no response*.

**exchange identification (XID)**. A specific type of basic link unit that is used to convey node and link characteristics between adjacent nodes. XIDs are exchanged between link stations before and during link activation to establish and negotiate link and node characteristics, and after link activation to communicate changes in these characteristics.

**exec**. (1) In the AIX operating system, to overlay the current process with another executable program. (2) See also *fork*.

**EXEC**. In a VM operating system, a user-written command file that contains CMS commands, other user-written commands, and execution control statements, such as branches.

**execute**. To perform the actions specified by a program or a portion of a program. (T)

**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*.

**EXLST**. In VSAM and VTAM, exit list.

**expedited flow**. In SNA, a data flow designated in the transmission header (TH) that is used to carry network control, session control, and various data flow control request/response units (RUs); the expedited flow is separate from the normal flow (which carries primarily end-user data) and can be used for commands that affect the normal flow. Contrast with *normal flow*.

**Note:** The normal and expedited flows move in both the primary-to-secondary and secondary-to-primary directions. Requests and responses on a given flow, whether normal or expedited, usually are processed sequentially within the path, but the expedited flow traffic may be moved ahead of the normal-flow traffic within the path at queuing points in the half-sessions and for half-session support in boundary functions.

**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 architecture (XA)**. An extension to System/370 architecture that takes advantage of continuing high performance enhancements to computer system hardware.

# F

**facilities field**. In an X.25 packet switching data network, an optional field used by the data terminal equipment (DTE) to convey information about the call to the network.

**feature**. A part of an IBM product that may be ordered separately by the customer.

**FIC**. First-in-chain.

**field**. An identifiable area in a window. Examples of fields are: an entry field, into which a user can type or place text, and a field of radio button choices, from which a user can select one choice.

**field-formatted**. Pertaining to a request or response that is encoded into fields, each having a specified format such as binary codes, bit-significant flags, and symbolic names. Contrast with *unformatted*.

**file**. A named set of records stored or processed as a unit. (T) Synonymous with *data set*.

**first-in-chain (FIC)**. A request unit (RU) whose request header (RH) begin chain indicator is on and whose RH end chain indicator is off. See also *RU chain*.

**first speaker**. See *first-speaker session*.

**first-speaker session**. The half-session defined at session activation as: (a) able to begin a bracket without requesting permission from the other half-session to do so, and (b) winning contention if both half-sessions attempt to begin a bracket simultaneously. Synonym for *contention-winner session*. Contrast with *bidder session*.

**fixed pacing**. Synonym for *fixed session-level pacing*.

**fixed session-level pacing**. A form of session-level pacing in which the data transfer rate is controlled using fixed pacing-window sizes, which are initialized at session-activation time. Synonymous with *fixed pacing*. Contrast with *adaptive session-level pacing*.

**flag (F) sequence**. A unique sequence of eight bits (01111110) used to delimit the opening and closing of an SDLC frame.

**flow**. In NETDA/2, the amount of traffic that can pass through a node, connection, or route in both directions during a given period of time.

**flow control**. (1) In data communication, control of the data transfer rate. (I) (2) In SNA, the process of managing the rate at which data traffic passes between components of the network. The purpose of flow control is to optimize the rate of flow of message units with minimum congestion in the network; that is, to neither overflow the buffers at the receiver or at intermediate routing nodes, nor leave the receiver waiting for more message units. See also *pacing*.

**FMD**. Function management data.

**fork**. In the AIX operating system, to create and start a child process.

**formatted system services**. A portion of VTAM that provides certain system services as a result of receiving a field-formatted command, such as an Initiate or Terminate command. Contrast with *unformatted system services (USS)*. See also *field-formatted*.

**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 level**. Synonymous with *data link level*. See *link level*.

**full-duplex (FDX)**. Synonym for *duplex*.

**full-screen mode**. A form of screen presentation in which the contents of an entire terminal screen can be displayed at once. Full-screen mode is often used for fill-in-the-blanks prompting.

**fullword**. Synonym for *computer word*.

**function management data (FMD)**. An RU category used for end-user data exchanged between logical units

(LUs) and for requests and responses exchanged between network services components of LUs, PUs, and control points.

## 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) 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. (3) 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. (4) In the AIX operating system, an entity that operates above the link layer and translates, when required, the interface and protocol used by one network into those used by another distinct network. (5) In TCP/IP, synonym for *router*.

**generation**. The process of assembling and link editing definition statements so that resources can be identified to all the necessary programs in a network.

**generic unbind**. Synonym for *session deactivation request*.

**graphic monitor**. The graphical user interface of the NetView Graphic Monitor Facility.

**group**. In the NetView/PC program, to identify a set of application programs that are to run concurrently.

## H

**half-duplex (HD, HDX)**. In data communication, pertaining to transmission in only one direction at a time. Contrast with *duplex*.

**handle**. (1) In the Advanced DOS and OS/2 operating systems, a binary value created by the system that identifies a drive, directory, and file so that the file can be found and opened. (2) In the AIX operating system, a data structure that is a temporary local identifier for an object. Allocating a handle creates it. Binding a handle makes it identify an object at a specific location. (3) In OS/400 application programming interfaces, a variable that represents an object.

**hardcopy**. (1) A permanent copy of a display image generated on an output device such as a printer or plotter, and which can be carried away. (T) (2) A printed copy of machine output in a visually readable form; for example, printed reports, listings, documents, and summaries. (3) Contrast with *softcopy*.

**hardware monitor**. The component of the NetView program that helps identify network problems, such as hardware, software, and microcode, from a central control point using interactive display techniques.

**header**. (1) System-defined control information that precedes user data. (2) The portion of a message that contains control information for the message such as one or more destination fields, name of the originating station, input sequence number, character string indicating the type of message, and priority level for the message.

**Help**. A choice that gives a user access to helpful information about objects, choices, tasks, and products. A Help choice can appear on a menu bar or as a push button.

**help panel**. Information displayed by a system in response to a help request from a user.

**hexadecimal**. (1) Pertaining to a selection, choice, or condition that has 16 possible different values or states. (I) (2) Pertaining to a fixed-radix numeration system, with radix of 16. (I) (3) Pertaining to a system of numbers to the base 16; hexadecimal digits range from 0 through 9 and A through F, where A represents 10 and F represents 15.

**high-level data link control (HDLC)**. In data communication, the use of a specified series of bits to control data links in accordance with the International Standards for HDLC: ISO 3309 Frame Structure and ISO 4335 Elements of Procedures.

**host**. (1) In the Internet suite of protocols, an end system. The end system can be any workstation; it does not have to be a mainframe. (2) See *host processor*.

**host ID**. In the Internet suite of protocols, that part of the IP address that defines the host system on the network. The length of the host ID depends on the type of network or network class (A, B, or C).

**host LU**. An SNA logical unit located in a host processor; for example, a VTAM application program. Contrast with *peripheral LU*.

**host node**. (1) A node at which a host computer is located. (T) (2) A node that provides an application programming interface (API) and a common application interface.

**host processor**. (1) A processor that controls all or part of a user application network. (T) (2) In a network, the processing unit in which the data communication access method resides.

# I

**I/O**. Input/output.

**ICP**. Internet Control Protocol.

**ID**. (1) Identifier. (2) Identification.

**IMS**. Information Management System. Synonym for *IMS/VS*.

**IMS/VS**. Information Management System/Virtual Storage. Synonymous with *IMS*.

**inactive**. (1) Not operational. (2) Pertaining to a node or device not connected or not available for connection to another node or device. (3) In the AIX operating system, pertaining to a window that does not have an input focus. (4) In VTAM, the state of a resource or a major or minor node that has not been activated or for which the VARY INACT command has been issued. (5) Contrast with *active*. (6) See also *inoperative*.

**inbound**. In communications, data that is received from the network.

**incoming call packet**. A call supervision packet transmitted by a data circuit-terminating equipment (DCE) to inform a called data terminal equipment (DTE) that another DTE has requested a call.

**information (I) format**. A format used for information transfer.

**information (I) frame**. A frame in I format used for numbered information transfer.

**Information Management System/Virtual Storage (IMS/VS)**. A database/data communication (DB/DC) system that can manage complex databases and networks.

**Information/Management**. A feature of the Information/System licensed program that provides interactive systems management applications for problem, change, and configuration management.

**inhibited**. In VTAM, pertaining to a logical unit (LU) that has indicated to its system services control point (SSCP) that it is temporarily not ready to establish LU-LU sessions. An initiate request for a session with an inhibited LU will be rejected by the SSCP. The LU can separately indicate whether this applies to its ability to act as a primary logical unit (PLU) or a secondary logical unit (SLU). See also *disabled* and *enabled*.

**initial program load (IPL)**. (1) The initialization procedure that causes an operating system to commence operation. (2) The process by which a configuration image is loaded into storage at the beginning of a work day or after a system malfunction. (3) The process of loading system programs and preparing a system to run jobs. (4) Synonymous with *system restart* and *system startup*.

**INITIATE**. A network services request sent from a logical unit (LU) to a system services control point (SSCP) requesting that an LU-LU session be established.

**inoperative**. The condition of a resource that has been active but is not currently active. A resource may be inoperative for reasons such as the following: a) it may have failed, b) it may have received an INOP request, or c) it may be suspended while a reactivate command is being processed. See also *inactive*.

**input/output (I/O)**. (1) Pertaining to input, output, or both. (A) (2) Pertaining to a device, process, or channel involved in data input, data output, or both.

**insert**. In LANs, to make an attaching device an active part of the LAN.

**installation**. (1) In system development, preparing and placing a functional unit in position for use. (T) (2) A particular computing system, including the work it does and the people who manage it, operate it, apply it to problems, service it, and use the results it produces.

**installation exit**. The means specifically described in an IBM software product's documentation by which an IBM software product may be modified by a customer's system programmers to change or extend the functions of the IBM software product. Such modifications consist of exit routines written to replace one or more existing modules of an IBM software product, or to add one or more modules or subroutines to an IBM software product, for the purpose of modifying or extending the functions of the IBM software product. Synonymous with *installation-wide exit*. See *user exit*.

**installation exit routine**. A routine written by a user to take control at an installation exit of an IBM software product.

**installation-wide exit**. Synonym for *installation exit*.

**instance**. In the AIX operating system, a concrete realization of an abstract object class. An instance of a widget or a gadget is a specific data structure that contains detailed appearance and behavioral information that is used to generate a specific graphical object on-screen at run time.

**integrated services digital network (ISDN)**. A digital end-to-end telecommunication network that supports multiple services including, but not limited to, voice and data.

**Note:** ISDNs are used in public and private network architectures.

**interactive**. (1) Pertaining to a program or system that alternately accepts input and then responds. An interactive system is conversational, that is, a continuous dialog exists between user and system. Contrast with *batch*. (2) Pertaining to the exchange of information between a user and a computer.

**interconnection**. See *SNA network interconnection (SNI)*.

**interface**. (1) A shared boundary between two functional units, defined by functional characteristics, signal characteristics, or other characteristics, as appropriate. The concept includes the specification of the connection of two devices having different functions. (T) (2) Hardware, software, or both, that links systems, programs, or devices.

**International Organization for Standardization (ISO)**. An organization of national standards bodies from various countries established to promote development of standards to facilitate international exchange of goods and services, and develop cooperation in intellectual, scientific, technological, and economic activity.

**International Telecommunication Union (ITU)**. The specialized telecommunication agency of the United Nations, established to provide standardized communication procedures and practices, including frequency allocation and radio regulations worldwide.

**Internet Control Protocol (ICP)**. The VIrtual NEtworking System (VINES) protocol that provides exception notifications, metric notifications, and PING support. See also *RouTing update Protocol (RTP)*.

**interrupt**. (1) A suspension of a process, such as execution of a computer program caused by an external event, and performed in such a way that the process can be resumed. (A) (2) To stop a process in such a way that it can be resumed.

**IPL**. (1) Initial program loader. (A) (2) Initial program load.

**ISDN**. Integrated services digital network.

**ISO**. International Organization for Standardization.

**ITU**. International Telecommunication Union.

**ITU-T**. See *ITU-TS*.

**ITU-TS**. International Telecommunication Union - Telecommunication Standardization Sector. The part of the International Telecommunication Union (ITU) that is responsible for developing recommendations for telecommunications.

# K

**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=`). (6) See also *definition statement* and *keyword operand*. (7) 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

**LAPB**. Link access protocol balanced.

**last-in-chain (LIC)**. A request unit (RU) whose request header (RH) end chain indicator is on and whose RH begin chain indicator is off. See also *RU chain*.

**layer**. (1) In network architecture, a group of services that is complete from a conceptual point of view, that is one out of a set of hierarchically arranged groups, and that extends across all systems that conform to the network architecture. (T) (2) In the Open Systems Interconnection reference model, one of seven conceptually complete, hierarchically arranged groups of services, functions, and protocols, that extend across all open systems. (T) (3) In SNA, a grouping of related functions that are logically separate from the functions in other groups. Implementation of the functions in one layer can be changed without affecting functions in other layers.

**LCD**. Line control definer.

**leased line**. Synonym for *nonswitched line*.

**LEN**. Low-entry networking.

**LIC**. (1) Last-in-chain. (2) In NCP, line interface coupler.

**line**. (1) The portion of a data circuit external to data circuit-terminating equipment (DCE), that connects the DCE to a data switching exchange (DSE), that connects a DCE to one or more other DCEs, or that connects a DSE to another DSE. (I) (2) Synonymous with *channel* and *circuit*.

**line control discipline**. Synonym for *link protocol* and *protocol*.

**line discipline**. Synonym for *link protocol* and *protocol*.

**line mode**. A form of screen presentation in which the information is presented a line at a time in the message area of the terminal screen. Contrast with *full-screen mode*.

**line 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) (3) In SNA, synonymous with *data link*.

**link access protocol balanced (LAPB)**. A protocol used for accessing an X.25 network at the link level. LAPB is a duplex, asynchronous, symmetric protocol, used in point-to-point communication.

**link-attached**. Pertaining to devices that are connected to a controlling unit by a data link. Contrast with *channel-attached*. Synonymous with *remote*.

**link connection**. (1) The physical equipment providing two-way communication between one link station and one or more other link stations; for example, a telecommunication line and data circuit-terminating equipment (DCE). (2) In SNA, synonymous with *data circuit*.

**link 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 protocol**. (1) The rules for sending and receiving data at the link level. (2) Synonymous with *line control discipline* and *line discipline*.

**link station**. (1) The hardware and software components within a node representing a connection to an adjacent node over a specific link. For example, if node A is the primary end of a multipoint line that connects to three adjacent nodes, node A will have three link stations representing the connections to the adjacent nodes. See also *adjacent link station (ALS)*. (2) In VTAM, a named resource within an APPN or a subarea node that represents the connection to another APPN or subarea node that is attached by an APPN or a subarea link. In the resource hierarchy in a subarea network, the link station is subordinate to the subarea link.

**link status (LS)**. Information maintained by local and remote modems.

**LLC**. Logical link control.

**load**. (1) To bring all or part of a computer program into memory from auxiliary storage so that the computer can run the program. (2) To place a diskette into a diskette drive.

**local**. (1) Pertaining to a device accessed directly without use of a telecommunication line. (2) Synonym for *channel-attached.* (3) Contrast with *remote*.

**local address**. In SNA, an address used in a peripheral node in place of a network address and transformed to or from a network address by the boundary function in a subarea node.

**local directory database**. That set of resources (LUs) in the network known at a particular node. The resources included are all those in the node's domain as well as any cache entries.

**Locate**. Synonym for *Locate/CD-Initiate*.

**Locate search message**. Synonym for *Locate/CD-Initiate.*

**Locate/CD-Initiate**. (1) An abbreviated term for a message exchanged between APPN nodes that contains one of the following sets of general data stream (GDS) variables:

- A Locate, a Find Resource, and a Cross-Domain Initiate GDS variable used for a network search request
- A Locate, a Found Resource, and a Cross-Domain Initiate GDS variable used for a search reply when a network resource has been located

These message structures correspond to the CP components that perform the search of the distributed network directory and establish the session. The Locate GDS variable contains information used to control the delivery of the search messages in the network. The Find and Found GDS variables contain information used in the directories: origin cache data (control point information) and search arguments (destination LU name), and located resource information, respectively. The Cross-Domain Initiate GDS variable contains endpoint TG vector information to be used in selecting the route for the session. The length of the Locate/CD-Initiate message is limited to 1024 bytes.

(2) Synonymous with *Locate* and *Locate search message*.

**logged-on operator**. A NetView operator station task that requires a terminal and a logged-on user. Contrast with *autotask*.

**logic error**. In VTAM, an error condition that results from an invalid request (a program logic error).

**logical channel**. In packet mode operation, a sending channel and a receiving channel that together are used to send and receive data over a data link at the same time. Several logical channels can be established on the same data link by interleaving the transmission of packets.

**logical line**. In NCP, the representation of the connection between NCP and a node communicating with NCP over a physical line such as token-ring or frame-relay. A single physical line can support multiple logical lines. Contrast with *physical line*.

**logical link**. (1) A pair of link stations, one in each of two adjacent nodes, and their underlying link connection, providing a single link-layer connection between the two nodes. Multiple logical links can be distinguished while they share the use of the same physical media connecting two nodes. Examples are 802.2 logical links used on local area network (LAN) facilities and LAP E logical links on the same point-to-point physical link between two nodes. The term logical link also includes the multiple X.25 logical channels that share the use of the access link from a DTE to an X.25 network. (2) In APPNTAM, the unidirectional representation in a node of a link.

**logical link control (LLC)**. The data link control (DLC) LAN sublayer that provides two types of DLC operation for the orderly exchange of information. The first type is connectionless service, which allows information to be sent and received without establishing a link. The LLC sublayer does not perform error recovery or flow control for connectionless service. The second type is connection-oriented service, which requires establishing a link prior to the exchange of information. Connection-oriented service provides sequenced information transfer, flow control, and error recovery.

**logical link control (LLC) protocol**. In a local area network, the protocol that governs the exchange of transmission frames between data stations independently of how the transmission medium is shared. (T) The LLC protocol was developed by the IEEE 802 committee and is common to all LAN standards.

**logical link control (LLC) protocol data unit**. A unit of information exchanged between link stations in different nodes. The LLC protocol data unit contains a destination service access point (DSAP), a source service access point (SSAP), a control field, and user data.

**logical unit (LU)**. A type of network accessible unit that enables end users to gain access to network resources and communicate with each other.

**logmode table**. Synonym for *logon mode table*.

**logoff**. In VTAM, an unformatted session-termination request.

**logon**. (1) The procedure by which a user begins a terminal session. (2) In VTAM, an unformatted session-initiation request for a session between two logical units.

**logon data**. In VTAM, either of the following:

- The user data portion of a field-formatted or unformatted session-initiation request
- The entire logon sequence or message from a logical unit (LU).

Synonymous with *logon message*.

**logon message**. Synonym for *logon data*.

**logon mode**. In VTAM, a subset of session parameters specified in a logon mode table for communication with a logical unit. See also *session parameters*.

**logon mode table**. In VTAM, a set of entries for one or more logon modes. Each logon mode is identified by a logon mode name. Synonymous with *logmode table*.

**low-entry networking (LEN)**. A capability of nodes to attach directly to one another using basic peer-to-peer protocols to support multiple and parallel sessions between logical units.

**low-entry networking (LEN) end node**. A LEN node receiving network services from an adjacent APPN network node.

**low-entry networking (LEN) node**. A node that provides a range of end-user services, attaches directly to other nodes using peer protocols, and derives network services implicitly from an adjacent APPN network node, that is, without the direct use of CP-CP sessions.

**LU**. Logical unit.

**LU-LU session**. A logical connection between two logical units (LUs) in an SNA network that typically provides communication between two end users.

**LU type**. The classification of an LU in terms of the specific subset of SNA protocols and options it supports for a given session, namely:

- The mandatory and optional values allowed in the session activation request
- The usage of data stream controls, function management headers (FMHs), request unit parameters, and sense data values
- Presentation services protocols such as those associated with FMH usage

LU types 0, 1, 2, 3, 4, 6.1, 6.2, and 7 are defined.

**LU 6.2**. A type of logical unit that supports general communication between programs in a distributed processing environment. LU 6.2 is characterized by (a) a peer relationship between session partners, (b) efficient utilization of a session for multiple transactions, (c) comprehensive end-to-end error processing, and (d) a generic application programming interface (API) consisting of structured verbs that are mapped into a product implementation.

**LUS**. Logical unit services.

# M

**M-bit**. More-data bit. In X.25 communications, the bit in a data packet that indicates that there is more data to follow in another data packet, when a message is too large for one packet.

**machine check handler (MCH)**. A feature that analyzes errors and attempts recovery by retrying the failing instruction. If retry is unsuccessful, it attempts to correct the malfunction or to isolate the affected task.

**macroinstruction**. (1) An instruction in a source language that is to be replaced by a defined sequence of instructions in the same source language and that may also specify values for parameters in the replaced instructions. (T) (2) In assembler programming, an assembler language statement that causes the assembler to process a predefined set of statements called a macro definition. The statements normally produced from the macro definition replace the macroinstruction in the program. See also *definition statement*.

**maintenance analysis procedure (MAP)**. A maintenance document that gives an IBM service representative a step-by-step procedure for tracing a symptom to the cause of a failure.

**major node**. In VTAM, a set of resources that can be activated and deactivated as a group. See *minor node*.

**management services (MS)**. (1) One of the types of network services in control points (CPs) and physical units (PUs). Management services are the services provided to assist in the management of SNA networks, such as problem management, performance and accounting management, configuration management, and change management. (2) Services that assist in the management of systems and networks in areas such as problem management, performance management, business management, operations management, configuration management, and change management.

**manager**. (1) In OSI management, a user that, for a particular interaction, has assumed a manager role. (2) A system that assumes a manager role.

**MAP**. Maintenance analysis procedure.

**mapping**. The process of converting data that is transmitted in one format by the sender into the data format that can be accepted by the receiver.

**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)

**MCH**. Machine check handler.

**menu**. (1) A list of options displayed to the user by a data processing system, from which the user can select an action to be initiated. (T) (2) In text processing, a list of choices displayed to the user by a text processor from which the user can select an action to be initiated. (T) (3) A list of choices that can be applied to an object. A menu can contain choices that are not available for selection in certain contexts. Those choices are indicated by reduced contrast.

**message**. (1) An assembly of characters and sometimes control codes that is transferred as an entity from an originator to one or more recipients. A message consists of two parts: envelope and content. (T) (2) In VTAM, the amount of function management data (FMD) transferred to VTAM by the application program with one SEND request.

**message switching**. The process of receiving a message, storing it, and forwarding it to its destination unaltered. (T)

**method**. In the NetView program, the code that runs within the Resource Object Data Manager (RODM) address space. Methods are used to implement behavior specified by an operation.

**MIC**. Middle-in-chain.

**microcode**. (1) One or more microinstructions. (2) A code, representing the instructions of an instruction set, that is implemented in a part of storage that is not program-addressable. (3) To design, write, and test one or more microinstructions.

**middle-in-chain (MIC)**. A request unit (RU) whose request header (RH) begin chain indicator and RH end chain indicator are both off. See also *RU chain*.

**migration**. The installation of a new version or release of a program to replace an earlier version or release.

**minor node**. In VTAM, a uniquely defined resource within a major node. See *major node* and *node*.

**mode**. See *mode name*.

**mode name**. The name used by the initiator of a session to designate the characteristics desired for the session, such as traffic pacing values, message-length limits, sync point and cryptography options, and the class of service within the transport network.

**modem (modulator/demodulator)**. (1) A functional unit that modulates and demodulates signals. One of the functions of a modem is to enable digital data to be transmitted over analog transmission facilities. (T) (A) (2) A device that converts digital data from a computer to an analog signal that can be transmitted on a telecommunication line, and converts the analog signal received to data for the computer.

**modulo**. (1) Pertaining to a modulus; for example, 9 is equivalent to 4 modulo 5. (2) See also *modulus*.

**modulus**. A number, such as a positive integer, in a relationship that divides the difference between two related numbers without leaving a remainder; for example, 9 and 4 have a modulus of 5 (9 - 4 = 5; 4 - 9 = -5; and 5 divides both 5 and -5 without leaving a remainder).

**monitor**. (1) A device that observes and records selected activities within a data processing system for analysis. Possible uses are to indicate significant departure from the norm, or to determine levels of utilization of particular functional units. (T) (2) Software or hardware that observes, supervises, controls, or verifies operations of a system. (A) (3) The function required to initiate the transmission of a token on the ring and to provide soft-error recovery in case of lost tokens, circulating frames, or other difficulties. The capability is present in all ring stations. (4) In the NetView Graphic Monitor Facility, to open a view that can receive status changes from the NetView program. Problem determination and correction can be performed directly from the view. Contrast with *browse*.

**more-data bit**. See *M-bit*.

**MSG**. Console messages.

**Multiple Virtual Storage (MVS)**. See *MVS*.

**MVS**. Multiple Virtual Storage. Implies MVS/370, the MVS/XA product, and the MVS/ESA product.

# N

**native**. In MPTN architecture, pertaining to the relationship between a transport user and a transport provider that are both based on the same transport protocol.

**NCP**. (1) Network Control Program. (2) Synonymous with *ACF/NCP*.

**NCP/EP definition facility (NDF)**. A program that is part of System Support Programs (SSP) and that is used to generate a load module for a partitioned emulation program (PEP), a Network Control Program (NCP), or an Emulation Program (EP).

**NDF**. NCP/EP definition facility.

**negative response (NR)**. In SNA, a response indicating that a request did not arrive successfully or was not processed successfully by the receiver. Contrast with *positive response*.

**negotiable BIND**. In SNA, a capability that allows two half-sessions to negotiate the parameters of a session when the session is being activated.

**negotiation**. The process of deciding what packet size to transmit between a network and a 3710 Network Controller.

**net ID**. See *network identifier*.

**NETID**. See *network identifier*.

**netname**. In Communications Manager/2, the name used in conjunction with the server name to identify a resource on the network when it is shared. See also *universal naming convention (UNC)*.

**NetView Bridge**. A set of application programming interfaces that allow the NetView program to interact with various types of databases in the MVS environment.

**NetView Graphic Monitor Facility (NGMF)**. A function of the NetView program that provides the network operator with a graphic topological presentation of a network controlled by the NetView program and that allows the operator to manage the network interactively.

**NetView Installation and Administration Facility/2 (NIAF/2)**. An OS/2-based tool that allows new NetView users or users migrating from a prior release of NetView to install, administer, and maintain the NetView program. NIAF/2 replaces the Interactive System Productivity Facility-based (ISPF-based) NetView Installation Facility.

**NetView-NetView task (NNT)**. The task under which a cross-domain NetView operator session runs. See *operator station task*.

**NetView Performance Monitor (NPM)**. An IBM licensed program that collects, monitors, analyzes, and displays data relevant to the performance of a VTAM telecommunication network. It runs as an online VTAM application program.

**NetView program**. An IBM licensed program used to monitor and manage a network and to diagnose network problems.

**network**. (1) An arrangement of nodes and connecting branches. (T) (2) A configuration of data processing devices and software connected for information interchange. (3) A group of nodes and the links interconnecting them.

**network address**. (1) According to ISO 7498-3, a name, unambiguous within the OSI environment, that identifies a set of network service access points. (2) An identifier for a node, station, or unit of equipment in a network. (3) 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*.

**network architecture**. The logical structure and operating principles of a computer network. (T)

**Note:** The operating principles of a network include those of services, functions, and protocols.

**network control (NC)**. In SNA, a request/response unit (RU) category used for requests and responses exchanged between physical units (PUs) for such purposes as activating and deactivating explicit and virtual routes and sending load modules to adjust peripheral nodes. See also *data flow control*, *function management data*, and *session control*.

**network control program**. A program, generated by the user from a library of IBM-supplied modules, that controls the operation of a communication controller.

**Network Control Program (NCP)**. An IBM licensed program that provides communication controller support for single-domain, multiple-domain, and interconnected network capability.

**network directory database**. Synonym for *distributed directory database*.

**network identifier**. (1) In TCP/IP, that part of the IP address that defines a network. The length of the network ID depends on the type of network class (A, B, or C). (2) A 1- to 8-byte customer-selected name or an 8-byte IBM-registered name that uniquely identifies a specific subnetwork. (3) In MPTN architecture, the address qualifier of a transport provider address that identifies a group of nodes according to the network in which they reside.

**network management**. The process of planning, organizing, and controlling a communication-oriented data processing or information system.

**network name**. (1) The symbolic identifier by which end users refer to a network accessible unit, a link, or a link station within a given subnetwork. In APPN networks, network names are also used for routing purposes. Contrast with *network address*. (2) In a multiple-domain network, the name of the APPL statement defining a VTAM application program. The network name must be unique across domains. Contrast with *ACB name*. See *uninterpreted name*.

**network node (NN)**. See *Advanced Peer-to-Peer Networking (APPN) network node*.

**network node domain**. An APPN network-node control point, its attached links, the network resources for which it answers directory search requests (namely, its local LUs and adjacent LEN end nodes), the adjacent APPN end nodes with which it exchanges directory search requests and replies, and other resources (such as a local storage device) associated with its own node or an adjacent end node for which it provides management services.

**network operator**. (1) A person who controls the operation of all or part of a network. (2) In a multiple-domain network, a person or program responsible for controlling all domains. See also *domain operator*.

**network services (NS) header**. In SNA, a 3-byte field in a function management data (FMD) request/response unit (RU) flowing in an SSCP-LU, SSCP-PU, or SSCP-SSCP session. The network services header is used primarily to identify the network services category of the request unit (RU) (for example, configuration services and session services) and the particular request code within a category.

**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.

**NIAF/2**. NetView Installation and Administration Facility/2.

**NIB**. Node initialization block.

**NN**. Network node.

**no response**. In SNA, a protocol requested in the form-of-response-requested field of the request header that directs the receiver of the request not to return any response, regardless of whether or not the request is received and processed successfully. Contrast with *definite response* and *exception response*.

**node**. (1) In a network, a point at which one or more functional units connect channels or data circuits. (I) (2) In network topology, the point at an end of a branch. (T) (3) The representation of a state or an event by means of a point on a diagram. (A) (4) In a tree structure, a point at which subordinate items of data originate. (A) (5) Any device, attached to a network, that transmits and receives data. (6) 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. (7) In VTAM, a point in a network defined by a symbolic name. See *major node* and *minor node*.

**node initialization block (NIB)**. In VTAM, a control block associated with a particular node or session that contains information used by the application program to identify the node or session and to indicate how communication requests on a session are to be handled by VTAM.

**node name**. In VTAM, the symbolic name assigned to a specific major or minor node during network definition.

**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. Synonymous with *leased line*. Contrast with *switched line*.

**normal flow**. In SNA, a data flow designated in the transmission header (TH) that is used primarily to carry end-user data. The rate at which requests flow on the normal flow can be regulated by session-level pacing. Normal and expedited flows move in both the primary-to-secondary and secondary-to-primary directions. Contrast with *expedited flow*.

**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.

**NPDA**. (1) Network Problem Determination Application. (2) A command that starts the NetView hardware monitor. NPDA also identifies various panels and functions as part of the hardware monitor.

**NPM**. NetView Performance Monitor.

**NPSI**. X.25 NCP Packet Switching Interface.

**NS**. Network services.

**NTO**. Network Terminal Option.

**NTune**. A set of programs (NTuneMON and NTuneNCP) that allow monitoring and tuning of active NCPs.

**NTuneNCP**. A program that runs in a communication controller and, with NTuneMON and VTAM, enables a network administrator to tune NCP interactively.

# O

**object**. (1) In object-oriented design or programming, an abstraction consisting of data and the operations associated with that data. See also *class*. (2) An item that a user can manipulate as a single unit to perform a task. An object can appear as text, an icon, or both.

**Off**. A choice that appears in the cascaded menu from the Refresh choice. It sets the refresh function to off.

**offline**. (1) Pertaining to the operation of a functional unit that takes place either independently of, or in parallel with, the main operation of a computer. (T) (2) Neither controlled by, nor communicating with, a computer. Contrast with *online*.

**offset**. The number of measuring units from an arbitrary starting point in a record, area, or control block, to some other point.

**OIC**. Only-in-chain.

**On**. A choice that appears in a cascaded menu from the Refresh choice. It immediately refreshes the view in a window.

**online**. (1) Pertaining to the operation of a functional unit when under the direct control of the computer. (T) (2) Pertaining to a user's ability to interact with a computer. (A) (3) Pertaining to a user's access to a computer via a terminal. (A) (4) Controlled by, or communicating with, a computer. (5) Contrast with *offline*.

**online information**. Information stored in a computer system that can be displayed, used, and modified in an interactive manner without any need to obtain hardcopy.

**only-in-chain (OIC)**. A request unit for which the request header (RH) begin chain indicator and RH end chain indicator are both on. See also *RU chain*.

**open**. (1) A break in an electrical circuit. (2) To make an adapter ready for use.

**Open**. A choice that leads to a window in which users can select the object they want to open.

**open system**. A system whose characteristics comply with standards made available throughout the industry and that therefore can be connected to other systems complying with the same standards. (T)

**Open Systems Interconnection (OSI)**. (1) The interconnection of open systems in accordance with standards of the International Organization for Standardization (ISO) for the exchange of information. (T) (A) (2) The use of standardized procedures to enable the interconnection of data processing systems.

**Note:** OSI architecture establishes a framework for coordinating the development of current and future standards for the interconnection of computer systems. Network functions are divided into seven layers. Each layer represents a group of related data processing and communication functions that can be carried out in a standard way to support different applications.

**Open Systems Interconnection (OSI) architecture**. Network architecture that adheres to that particular set of ISO standards that relates to Open Systems Interconnection. (T)

**Open Systems Interconnection (OSI) reference model**. A model that describes the general principles of the Open Systems Interconnection, as well as the purpose and the hierarchical arrangement of its seven layers. (T)

**operand**. (1) An entity on which an operation is performed. (I) (2) That which is operated upon. An operand is usually identified by an address part of an instruction. (A) (3) Information entered with a command name to define the data on which a command processor operates and to control the execution of the command processor. (4) An expression to whose value an operator is applied. See also *definition statement*, *keyword*, *keyword parameter*, and *parameter*.

**operating system (OS)**. Software that controls the execution of programs and that may provide services such as resource allocation, scheduling, input/output control, and data management. Although operating systems are predominantly software, partial hardware implementations are possible. (T)

**operation**. In object-oriented design or programming, a service that can be requested at the boundary of an object. Operations include modifying an object or disclosing information about an object.

**operator**. (1) In a language statement, the lexical entity that indicates the action to be performed on operands. See also *definition statement*. (2) A person or program responsible for managing activities controlled by a given piece of software such as MVS, the NetView program, or IMS. (3) A person who operates a device. (4) A person who keeps a system running. (5) See also *autotask*, *logged-on operator*, *network operator*, and *operator station task*.

**operator station task (OST)**. The NetView task that establishes and maintains the online session with the network operator. There is one operator station task for each network operator who logs on to the NetView program. See *NetView-NetView task*.

**OPNDST**. Open destination.

**origin**. An external logical unit (LU) or application program from which a message or other data originates. See also *destination*.

**OSI**. Open Systems Interconnection.

**outbound**. In communications, data that is transmitted to the network.

**output**. Pertaining to a device, process, or channel involved in an output process, or to the associated data or states. The word “output” may be used in place of “output data,” “output signal,” “output process,” when such a usage is clear in a given context. (T)

# P

**pacing**. A technique by which a receiving component controls the rate of transmission of a sending component to prevent overrun or congestion. See also *flow control*, *receive pacing*, *send pacing*, *session-level pacing*, and *virtual route (VR) pacing*.

**pacing group**. Synonym for *pacing window*.

**pacing response**. In SNA, an indicator that signifies the readiness of a receiving component to accept another pacing group. The indicator is carried in a response header (RH) for session-level pacing and in a transmission header (TH) for virtual route pacing.

**pacing window**. (1) The path information units (PIUs) that can be transmitted on a virtual route before a virtual-route pacing response is received, indicating that the virtual route receiver is ready to receive more PIUs on the route. (2) The requests that can be transmitted on the normal flow in one direction on a session before a session-level pacing response is received, indicating that the receiver is ready to accept the next group of requests. (3) Synonymous with *pacing group*.

**package**. A collection of attributes, notifications, operations, or behaviors that are treated as a single module in the specification of a managed object class. Packages can be mandatory or conditional when referenced in a definition of a managed object class.

**packet**. In data communication, a sequence of binary digits, including data and control signals, that is transmitted and switched as a composite whole. The data, control signals, and, possibly, error control information are arranged in a specific format. (I)

**packet assembler/disassembler (PAD)**. A functional unit that enables data terminal equipment (DTEs) not equipped for packet switching to access a packet switched network. (T) (A)

**packet level**. (1) The packet format and control procedures for exchange of packets containing control information and user data between data terminal equipment (DTE) and data circuit-terminating equipment (DCE). See also *data link level* and *physical level*. (2) A part of Recommendation X.25 that defines the protocol for establishing logical connections between two DTEs and for transferring data on these connections.

**packet mode operation**. Synonym for *packet switching*.

**packet size**. (1) In X.25 communications, the length of the user data in a data packet. (2) The maximum amount of user data in a packet.

**packet switching**. (1) The process of routing and transferring data by means of addressed packets so that a channel is occupied only during transmission of a packet. On completion of the transmission, the channel is made available for transfer of other packets. (I) (2) Synonymous with *packet mode operation*. See also *circuit switching*.

**packet switching data network (PSDN)**. A network that uses packet switching as a means of transmitting data.

**packet window**. The maximum number of consecutive data packets that are allowed to flow between a data terminal equipment (DTE) and a data circuit-terminating equipment (DCE) before an acknowledgment is received for a given logical channel.

**PAD**. Packet assembler/disassembler.

**page**. (1) In a virtual storage system, a fixed-length block that has a virtual address and is transferred as a unit between real storage and auxiliary storage. (I) (A) (2) A printed form. (3) The information displayed at the same time on the screen of a display device. (4) In VSE, a fixed-length block of instructions, data, or both that can be located in processor storage or in the page data set on disk. (5) To replace the information displayed on the screen with prior or subsequent information from the same file.

**panel**. (1) See *window*. (2) A formatted display of information that appears on a display screen. See *help panel* and *task panel*. (3) In computer graphics, a display image that defines the locations and characteristics of display fields on a display surface.

**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. (5) See also *keyword*, *keyword parameter*, and *operand*.

**parameters**. In NETDA/2, the set of restrictions that affect only the output of a network design. A change in a parameter value does not change the input to the network design. Contrast with *constraints*.

**password**. (1) A value used in authentication or a value used to establish membership in a set of people having specific privileges. (2) A unique string of characters known to a computer system and to a user, who must specify the character string to gain access to a system and to the information stored within it. (3) In computer security, a string of characters known to the computer system and a user, who must specify it to gain full or limited access to a system and to the data stored within it.

**path**. (1) In a network, any route between any two nodes. A path may include more than one branch. (T) (2) The series of transport network components (path control and data link control) that are traversed by the information exchanged between two network accessible units. See also *explicit route (ER)*, *route extension*, and *virtual route (VR)*. (3) In VTAM when defining a switched major node, a potential dial-out port that can be used to reach that node. (4) In the NetView/PC program, a complete line in a configuration that contains all of the resources in the service point command service (SPCS) query link configuration request list.

**path control (PC)**. The function that routes message units between network accessible units in the network and provides the paths between them. It converts the basic information units (BIUs) from transmission control (possibly segmenting them) into path information units (PIUs) and exchanges basic transmission units containing one or more PIUs with data link control. Path control differs by node type: some nodes (APPN nodes, for example) use locally generated session identifiers for routing, and others (subarea nodes) use network addresses for routing.

**path information unit (PIU)**. A message unit consisting of a transmission header (TH) alone, or a TH followed by a basic information unit (BIU) or a BIU segment.

**path statement**. In NETDA/2, a type of route statement used for subarea routing. Existing VTAM and NCP path statements can be included in a network design, or new path statements can be generated.

**PC**. (1) Personal computer. (2) Path control.

**PDB**. Pool descriptor block.

**peer**. In network architecture, any functional unit that is in the same layer as another entity. (T)

**pending active session**. In VTAM, the state of an LU-LU session recorded by the system services control point (SSCP) when it finds both logical units (LUs) available and has sent a CINIT request to the primary logical unit (PLU) of the requested session.

**peripheral LU**. In SNA, a logical unit in a peripheral node. Contrast with *subarea LU*.

**peripheral node**. A node that uses local addresses for routing and therefore is not affected by changes in network addresses. A peripheral node requires boundary-function assistance from an adjacent subarea node. A peripheral node can be a type 1, 2.0, or 2.1 node connected to a subarea boundary node.

**peripheral PU**. In SNA, a physical unit in a peripheral node. Contrast with *subarea PU*.

**permanent virtual circuit (PVC)**. (1) In X.25 and frame-relay communications, a virtual circuit that has a logical channel permanently assigned to it at each data terminal equipment (DTE). Call-establishment protocols are not required. Contrast with *switched virtual circuit (SVC)*. (2) The logical connection between two frame-relay terminating equipment stations, either directly or through one or more frame-relay frame handlers. A PVC consists of one or more PVC segments.

**personal computer (PC)**. (1) A microcomputer primarily intended for stand-alone use by an individual. (T) (2) A desktop, floor-standing, or portable microcomputer that usually consists of a system unit, a display monitor, a keyboard, one or more diskette drives, internal fixed-disk storage, and an optional printer. PCs are designed primarily for stand-alone operation but may be connected to mainframes or networks.

**physical circuit**. A circuit established without multiplexing. See also *data circuit*. Contrast with *virtual circuit*.

**physical connection**. (1) A connection that establishes an electrical circuit. (2) A point-to-point or multipoint connection. (3) Synonymous with *connection*.

**physical level**. In X.25, the mechanical, electrical, functional, and procedural media used to activate, maintain, and deactivate the physical link between the data terminal equipment (DTE) and the data circuit-terminating equipment (DCE). See *data link level* and *packet level*.

**physical line**. In NCP, the physical connection between NCP and an adjacent device or local area network (LAN). A single physical line, such as token-ring or frame-relay, can support multiple logical lines. Contrast with *logical line*.

**physical services header (PSH)**. The protocol above the X.25 packet level used to provide adjacent node services for SNA-to-SNA communication to IBM 5973 Network Interface Adapters.

**physical unit (PU)**. The component that manages and monitors the resources (such as attached links and adjacent link stations) associated with a node, as requested by an SSCP via an SSCP-PU session. An SSCP activates a session with the physical unit in order to indirectly manage, through the PU, resources of the node such as attached links. This term applies to type 2.0, type 4, and type 5 nodes only. See also *peripheral PU* and *subarea PU*.

**physical unit (PU) services**. In SNA, the components within a physical unit (PU) that provide configuration services and maintenance services for SSCP-PU sessions.

**PIU**. Path information unit.

**PLU**. Primary logical unit.

**POI**. Program operator interface.

**pointer**. (1) A data element that indicates the location of another data element. (T) (2) An identifier that indicates the location of an item of data. (A)

**pop**. To remove an item from the top of a pushdown list. Contrast with *push*.

**POP**. Post Office Protocol.

**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*.

**post**. (1) In the AIX operating system, the action required to make a pop-up or pull-down menu appear. This action is normally a click or a button press on one of the mouse buttons. (2) To enter a unit of information on a record. (3) To note the occurrence of an event. (4) To add information to a record in order to keep the record current.

**POST**. Power-on self test.

**Post Office Protocol (POP)**. A protocol used for exchanging network mail and accessing mailboxes.

**Prepare**. A presentation services header that flows as part of commit processing, indicating the partner has begun the first phase of the two-phase commit process.

**primary logical unit (PLU)**. In SNA, the logical unit (LU) that sends the BIND to activate a session with its partner LU. Contrast with *secondary logical unit (SLU)*.

**problem determination**. The process of determining the source of a problem; for example, a program component, machine failure, telecommunication facilities, user or contractor-installed programs or equipment, environmental failure such as a power loss, or user error.

**process**. (1) A course of the events defined by its purpose or by its effect, achieved under given conditions. (2) In data processing, the course of events that occurs during the execution of all or part of a program. (T) (3) Any operation or combination of operations on data. (4) A function being performed or waiting to be performed. (5) To perform operations on data in a process. (I) (A)

**process identification number (process ID)**. A unique number assigned to a process by the operating system. The number is used internally by processes to communicate.

**processor**. In a computer, a functional unit that interprets and executes instructions. A processor consists of at least an instruction control unit and an arithmetic and logic unit. (T)

**profile**. Data that describes the significant characteristics of a user, a group of users, or one or more computer resources.

**program operator**. A VTAM application program that is authorized to issue VTAM operator commands and receive VTAM operator awareness messages. See also *solicited message* and *unsolicited message*.

**program operator interface (POI)**. A VTAM function that allows programs to perform VTAM operator functions.

**programmed operator**. The Emulator High-Level Language Application Programming Interface (EHLLAPI) application program that performs and monitors activities in a workstation presentation space. Contrast with *terminal operator*.

**protocol**. (1) A set of semantic and syntactic rules that determine the behavior of functional units in achieving communication. (I) (2) In Open Systems Interconnection architecture, a set of semantic and syntactic rules that determine the behavior of entities in the same layer in performing communication functions. (T) (3) In SNA, the meanings of, and the sequencing rules for, requests and responses used for managing the network, transferring data, and synchronizing the states of network components. Synonymous with *line control discipline* and *line discipline*. See *bracket protocol* and *link protocol*.

**PSDN**. Packet switching data network.

**PSH**. Physical services header.

**PU**. Physical unit.

**PU type**. (1) Deprecated term for *node type*. (2) The type of physical unit in a node.

**push**. To add an item to the top of a pushdown list. Contrast with *pop*.

**PUT**. Program update tape.

**PVC**. Permanent virtual circuit.

# Q

**QLLC**. Qualified logical link control.

**QOS**. Quality of service.

**qualified logical link control (QLLC)**. An X.25 protocol that allows the transfer of data link control information between two adjoining SNA nodes that are connected through an X.25 packet-switching data network. The QLLC provides the qualifier “Q” bit in

X.25 data packets to identify packets that carry logical link protocol information.

**quality of service (QOS)**. A set of communication characteristics required by an application. Each QOS defines a specific transmission priority, level of route reliability, and security level. Each QOS also defines whether the sessions are interactive. Contrast with *class of service (COS)*.

**queue**. (1) A list constructed and maintained so that the next data element to be retrieved is the one stored first. (T) (2) A line or list of items waiting to be processed; for example, work to be performed or messages to be displayed. (3) To arrange in or form a queue.

# R

**read-only**. A type of access to data that allows data to be read but not copied, printed, or modified.

**read-only memory (ROM)**. Memory in which stored data cannot be modified by the user except under special conditions.

**receive not ready (RNR)**. In communications, a data link command or response that indicates a temporary condition of being unable to accept incoming frames.

**receive not ready (RNR) packet**. See *RNR packet*.

**receive pacing**. In SNA, the pacing of message units that a component is receiving. Contrast with *send pacing*.

**receive ready packet**. See *RR packet*.

**RECFMS**. Record formatted maintenance statistics.

**Recommendation X.25**. See *X.25*.

**Recommendation X.28**. See *X.28*.

**Recommendation X.29**. See *X.29*.

**Recommendation X.3**. See *X.3*.

**record**. (1) In programming languages, an aggregate that consists of data objects, possibly with different attributes, that usually have identifiers attached to them. In some programming languages, records are called structures. (I) (2) A set of data treated as a unit. (T) (3) A set of one or more related data items grouped for processing. (4) In VTAM, the unit of data transmission for record mode. A record represents whatever amount of data the transmitting node chooses to send.

**record formatted maintenance statistics (RECFMS)**. A statistical record built by an SNA controller and usually solicited by the host.

**reentrant**. The attribute of a program or routine that allows the same copy of the program or routine to be used concurrently by two or more tasks.

**release**. (1) A distribution of a new product or new function and APAR fixes for an existing product. Normally, programming support for the prior release is discontinued after some specified period of time following availability of a new release. The first version of a product is announced as Release 1, Modification Level 0. (2) In VTAM, to relinquish control of resources (communication controllers or physical units). See also *resource takeover*. Contrast with *acquire*.

**remote**. (1) Pertaining to a system, program, or device that is accessed through a telecommunication line. (2) Synonym for *link-attached*. (3) Contrast with *local*.

**request header (RH)**. The control information that precedes a request unit (RU). See also *request/response header (RH)*.

**request parameter list (RPL)**. In VTAM, a control block that contains the parameters necessary for processing a request for data transfer, for establishing or terminating a session, or for some other operation.

**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)*.

**reset**. On a virtual circuit, reinitialization of data flow control. At reset, all data in transit are eliminated.

**reset confirmation packet**. In X.25 communications, a packet transmitted by the data terminal equipment (DTE) to inform the data circuit-terminating equipment (DCE) that a reset operation has been processed.

**reset indication packet**. In X.25 communications, a packet transmitted by the data circuit-terminating equipment (DCE) to inform the data terminal equipment (DTE) that a virtual call or a permanent virtual circuit is being reset and the reason for the resetting.

**reset request packet**. In X.25 communications, a packet transmitted by the data terminal equipment (DTE) to the data circuit-terminating equipment (DCE) to request that a virtual call or a permanent virtual circuit be reset. The reason for the request can also be specified in the packet.

**resource**. (1) Any facility of a computing system or operating system required by a job or task, and including main storage, input/output devices, the processing unit, data sets, and control or processing programs. (2) In the NetView program, any hardware or software that provides function to the network.

**Resource Object Data Manager (RODM)**. A component of the NetView program that operates as a cache manager and that supports automation applications. RODM provides an in-memory cache for maintaining real-time data in an address space that is accessible by multiple applications.

**resource takeover**. In VTAM, an action initiated by a network operator to transfer control of resources from one domain to another without breaking the connections or disrupting existing LU-LU sessions on the connection. See also *acquire* and *release*.

**response**. (1) In data communication, a reply represented in the control field of a response frame. It advises the primary or combined station of the action taken by the secondary or other combined station to one or more commands. (2) See also *command*.

**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 indicator**. In the AS/400 system, a 1-character field passed with an input record from the system to a program to provide information about the data record or actions taken by the workstation user.

**response time**. (1) The elapsed time between the end of an inquiry or demand on a computer system and the beginning of the response; for example, the length of time between an indication of the end of an inquiry and the display of the first character of the response at a user terminal. (I) (A) (2) For response time monitoring, the time from the activation of a transaction until a response is received, according to the response time definition coded in the performance class.

**response time monitor (RTM)**. A feature available with certain hardware devices to allow measurement of response times, which may be collected and displayed by the NetView program.

**response unit (RU)**. A message unit that acknowledges a request unit. It may contain prefix information received in a request unit. If positive, the response unit may contain additional information (such as session parameters in response to BIND SESSION). If negative, the response unit contains sense data defining the exception condition.

**Restructured Extended Executor (REXX)**. A general-purpose, procedural language for end-user personal programming, designed for ease by both casual general users and computer professionals. It is also useful for application macros. REXX includes the capability of issuing commands to the underlying operating system from these macros and procedures. Features include powerful character-string manipulation, automatic data typing, manipulation of objects familiar to people, such as words, numbers, and names, and built-in interactive debugging.

**return code**. (1) A code used to influence the execution of succeeding instructions. (A) (2) A value returned to a program to indicate the results of an operation requested by that program.

**REX**. Route extension.

**REXX**. Restructured Extended Executor.

**RH**. Request/response header.

**RNR**. Receive not ready.

**RNR packet**. A packet used by a data terminal equipment (DTE) or by a data circuit-terminating equipment (DCE) to indicate a temporary inability to accept additional packets for a virtual call or permanent virtual circuit.

**ROM**. Read-only memory. (A)

**route**. (1) An ordered sequence of nodes and transmission groups (TGs) that represent a path from an origin node to a destination node traversed by the traffic exchanged between them. (2) The path that network traffic uses to get from source to destination.

**route daemon**. A program that runs under 4BSD UNIX to propagate route information among machines on a local area network. Also referred to as *routed* (pronounced “route-d”).

**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)*.

**routed**. Pronounced “route-d.” See *route daemon*.

**router**. (1) A computer that determines the path of network traffic flow. The path selection is made from several paths based on information obtained from specific protocols, algorithms that attempt to identify the shortest or best path, and other criteria such as metrics or protocol-specific destination addresses. (2) An attaching device that connects two LAN segments, which use similar or different architectures, at the reference model network layer. (3) In OSI terminology, a function that determines a path by which an entity can be reached. (4) In TCP/IP, synonymous with *gateway*. (5) Contrast with *bridge*.

**RouTing update Protocol (RTP)**. The VIrtual NEtworking System (VINES) protocol that maintains the routing database and allows the exchange of routing information between VINES nodes. See also *Internet Control Protocol (ICP)*.

**RPL**. Request parameter list.

**RR**. Receive ready.

**RR packet**. A packet used by a data terminal equipment (DTE) or by a data circuit-terminating equipment (DCE) to indicate that it is ready to receive data packets within the window.

**RTM**. Response time monitor.

**RTP**. RouTing update Protocol.

**RU**. Request/response unit.

**RU chain**. In SNA, a set of related request/response units (RUs) that are consecutively transmitted on a particular normal or expedited data flow. The request RU chain is the unit of recovery: if one of the RUs in the chain cannot be processed, the entire chain is discarded. Each RU belongs to only one chain, which has a beginning and an end indicated by means of control bits in request/response headers within the RU chain. Each RU can be designated as first-in-chain (FIC), last-in-chain (LIC), middle-in-chain (MIC), or only-in-chain (OIC). Response units and expedited-flow request units are always sent as only-in-chain.

# S

**SABM**. Set asynchronous balanced mode.

**screen**. (1) The physical surface of a display device upon which information is shown to users. (2) In the AIX extended curses library, a window that is as large as the display screen of the workstation. (3) Deprecated term for *display panel*.

**SDLC**. Synchronous Data Link Control.

**SDT**. Start data traffic.

**secondary logical unit (SLU)**. In SNA, the logical unit (LU) that contains the secondary half-session for a particular LU-LU session. An LU may contain secondary and primary half-sessions for different active LU-LU sessions. Contrast with *primary logical unit (PLU)*.

**secondary logical unit (SLU) key**. A key-encrypting key used to protect a session cryptography key during its transmission to the secondary half-session.

**segmenting**. In OSI, a function performed by a layer to map one protocol data unit (PDU) from the layer it supports into multiple PDUs.

**select**. To explicitly identify one or more objects to which a subsequent choice will apply.

**selection**. The process of explicitly identifying one or more objects to which a subsequent choice will apply.

**send pacing**. In SNA, the pacing of message units that a component is sending. Contrast with *receive pacing*.

**server**. (1) A functional unit that provides shared services to workstations over a network; for example, a file server, a print server, a mail server. (T) (2) In a network, a data station that provides facilities to other stations; for example, a file server, a print server, a mail server. (A) (3) In the AIX operating system, an application program that usually runs in the background and is controlled by the system program controller. (4) In AIX Enhanced X-Windows, a program that provides the basic windowing mechanism. It handles interprocess communication (IPC) connections from clients, demultiplexes graphics requests onto screens, and multiplexes input back to clients.

**service point (SP)**. An entry point that supports applications that provide network management for resources not under the direct control of itself as an entry point. Each resource is either under the direct control of another entry point or not under the direct control of any entry point. A service point accessing these resources is not required to use SNA sessions (unlike a focal point). A service point is needed when entry point support is not yet available for some network management function.

**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 data**. Data about a session, collected by the NetView program, that consists of session awareness data, session trace data, and session response time data.

**session deactivation request**. In SNA, a request that deactivates a session between two network accessible units (NAUs); for example, UNBIND and DACTPU. Synonymous with *generic unbind*. Contrast with *session activation request*.

**session-level pacing**. A flow control technique that permits a receiving half-session or session connector to control the data transfer rate (the rate at which it receives request units) on the normal flow. It is used to prevent overloading a receiver with unprocessed requests when the sender can generate requests faster than the receiver can process them. See also *adaptive session-level pacing*, *fixed session-level pacing*, and *virtual route (VR) pacing*.

**session monitor**. The component of the NetView program that collects and correlates session-related data and provides online access to this information.

**session parameters**. In SNA, the parameters that specify or constrain the protocols (such as bracket protocol and pacing) for a session between two network accessible units. See also *logon mode*.

**session partner**. In SNA, one of the two network accessible units (NAUs) having an active session.

**set asynchronous balanced mode (SABM)**. In communications, a data link control command used to establish a data link connection with the destination in asynchronous balanced mode.

**shared**. Pertaining to the availability of a resource for more than one use at the same time.

**shutdown**. The process of ending operation of a system or a subsystem, following a defined procedure.

**SLU**. Secondary logical unit.

**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.

**SNA terminal**. A terminal that supports SNA protocols.

**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. (3) Contrast with *hardcopy*.

**solicited message**. A response from VTAM to a command entered by a program operator. Contrast with *unsolicited message*.

**SP**. Service point.

**specific-mode**. In VTAM, the following:

- The form of a RECEIVE request that obtains input from one specific session
- The form of an ACCEPT request that completes the establishment of a session by accepting a specific queued CINIT request.

Contrast with *any-mode*. See *continue-specific mode*.

**SSCP**. System services control point.

**SSCP-dependent LU**. An LU that requires assistance from a system services control point (SSCP) in order to initiate an LU-LU session. It requires an SSCP-LU session.

**SSCP-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-PU session**. In SNA, a session between a system services control point (SSCP) and a physical unit (PU); SSCP-PU sessions allow SSCPs to send requests to and receive status information from individual nodes in order to control the network configuration.

**SSP**. System Support Programs.

**ST**. Session configuration screen abbreviation.

**stage**. A program that processes messages in a NetView pipeline. Stages send messages to each other serially.

**statement**. A language syntactic unit consisting of an operator, or other statement identifier, followed by one or more operands. See *definition statement*.

**static**. (1) In programming languages, pertaining to properties that can be established before execution of a program; for example, the length of a fixed length variable is static. (I) (2) Pertaining to an operation that occurs at a predetermined or fixed time. (3) Contrast with *dynamic*.

**station**. An input or output point of a system that uses telecommunication facilities; for example, one or more systems, computers, terminals, devices, and associated programs at a particular location that can send or receive data over a telecommunication line.

**status**. The condition or state of hardware or software, usually represented by a status code.

**status monitor**. A component of the NetView program that collects and summarizes information on the status of resources defined in a VTAM domain.

**stream**. (1) To send data from one device to another. (2) See *data stream*.

**subarea**. A portion of the SNA network consisting of a subarea node, attached peripheral nodes, and associated resources. Within a subarea node, all network accessible units (NAUs), links, and adjacent link stations (in attached peripheral or subarea nodes) that are addressable within the subarea share a common subarea address and have distinct element addresses.

**subarea LU**. In SNA, a logical unit in a subarea node. Contrast with *peripheral LU*.

**subarea node (SN)**. A node that uses network addresses for routing and maintains routing tables that reflect the configuration of the network. Subarea nodes can provide gateway function to connect multiple subarea networks, intermediate routing function, and boundary function support for peripheral nodes. Type 4 and type 5 nodes can be subarea nodes.

**subarea PU**. In SNA, a physical unit in a subarea node. Contrast with *peripheral PU*.

**subsystem**. A secondary or subordinate system, usually capable of operating independently of, or asynchronously with, a controlling system. (T)

**supervisor**. The part of a control program that coordinates the use of resources and maintains the flow of processing unit operations.

**supervisor call (SVC)**. A request that serves as the interface into operating system functions, such as allocating storage. The SVC protects the operating system from inappropriate user entry. All operating system requests must be handled by SVCs.

**SVC**. (1) Supervisor call. (2) Switched virtual circuit.

**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 virtual circuit (SVC)**. An X.25 circuit that is dynamically established when needed. The X.25 equivalent of a switched line. Contrast with *permanent virtual circuit (PVC)*.

**synchronous**. (1) Pertaining to two or more processes that depend upon the occurrence of specific events such as common timing signals. (T) (2) Occurring with a regular or predictable time relationship.

**Synchronous Data Link Control (SDLC)**. (1) A discipline conforming to subsets of the Advanced Data Communication Control Procedures (ADCCP) of the American National Standards Institute (ANSI) and High-level Data Link Control (HDLC) of the International Organization for Standardization, for managing synchronous, code-transparent, serial-by-bit information transfer over a link connection. Transmission exchanges may be duplex or half-duplex over switched or nonswitched links. The configuration of the link connection may be point-to-point, multipoint, or loop. (I) (2) Contrast with *binary synchronous communication (BSC)*.

**synchronous request**. In VTAM, a request for a synchronous operation. Contrast with *asynchronous request*.

**SYSDEF**. System definition.

**system**. In data processing, a collection of people, machines, and methods organized to accomplish a set of specific functions. (I) (A)

**system definition**. The process, completed before a system is put into use, by which desired functions and operations of the system are selected from various available options. Synonymous with *system generation*.

**system generation**. Synonym for *system definition*.

**system restart**. Synonym for *initial program load (IPL)*.

**system services control point (SSCP)**. A component within a subarea network for managing the configuration, coordinating network operator and problem determination requests, and providing directory services and other session services for end users of the network. Multiple SSCPs, cooperating as peers with one another, can divide the network into domains of control, with each SSCP having a hierarchical control relationship to the physical units and logical units within its own domain.

**system services control point (SSCP) domain**. The system services control point, the physical units (PUs), the logical units (LUs), the links, the link stations, and all the resources that the SSCP has the ability to control by means of activation and deactivation requests.

**system startup**. Synonym for *initial program load (IPL)*.

**System Support Programs (SSP)**. An IBM licensed program, made up of a collection of utilities and small programs, that supports the operation of the NCP.

**Systems Network Architecture (SNA)**. The description of the logical structure, formats, protocols, and operational sequences for transmitting information units through, and controlling the configuration and operation of, networks. The layered structure of SNA allows the ultimate origins and destinations of information, that is, the end users, to be independent of and unaffected by the specific SNA network services and facilities used for information exchange.

# T

**table**. A repository for data that NETDA/2 uses to design a network. Each table contains information related to the network.

**takeover**. The process by which the failing active subsystem is released from its extended recovery facility (XRF) sessions with terminal users and replaced by an alternate subsystem. See *resource takeover*.

**task**. In a multiprogramming or multiprocessing environment, one or more sequences of instructions treated by a control program as an element of work to be accomplished by a computer. (I) (A)

**task panel**. Online display from which you communicate with the program in order to accomplish the program's function, either by selecting an option provided on the panel or by entering an explicit command. See also *help panel*.

**Task-Related User Exit (TRUE)**. A CICS module used for invoking resource managers that are outside of CICS. A TRUE is one of the components of CICS sockets.

**TCP**. Transmission Control Protocol.

**Telecommunications Access Method (TCAM)**. An access method used to transfer data between main storage and remote or local terminals.

**teleprocessing network simulator (TPNS)**. A testing package that enables a user to test and evaluate teleprocessing systems before actual terminal installation.

**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 operator**. The user of an Emulator High-Level Language Application Programming Interface (EHLLAPI) application program. Contrast with *programmed operator*.

**TERMINATE**. In SNA, a request unit that is sent by a logical unit (LU) to its system services control point (SSCP) to cause the SSCP to start a procedure to end one or more designated LU-LU sessions.

**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.

**Time Sharing Option (TSO)**. An operating system option; for the System/370 system, the option provides interactive time sharing from remote terminals.

**time stamp**. (1) To apply the current system time. (2) The value on an object that is an indication of the system time at some critical point in the history of the object. (3) In query, the identification of the day and time when a query report was created that query automatically provides on each report.

**timeout**. (1) An event that occurs at the end of a predetermined period of time that began at the occurrence of another specified event. (I) (2) A time interval allotted for certain operations to occur; for example, response to polling or addressing before system operation is interrupted and must be restarted.

**toggle**. (1) Pertaining to any device having two stable states. (A) (2) Pertaining to a switching device, such as a toggle key on a keyboard, that allows a user to switch between two types of operations. (3) To switch between two modes; for example, on a personal computer connected to a network, to switch between the data entry and command entry modes or between stand-alone operation and terminal emulation.

**topology**. In communications, the physical or logical arrangement of nodes in a network, especially the relationships among nodes and the links between them.

**TPNS**. Teleprocessing network simulator.

**TR**. Trace.

**transaction**. In IMS/VS, a specific set of input data that triggers execution of a specific process or job. A transaction is a message destined for an application program.

**transaction program (TP)**. (1) A program that processes transactions in an SNA network. There are two kinds of transaction programs: application transaction programs and service transaction programs. See also *conversation*. (2) In VTAM, a program that performs services related to the processing of a transaction. One or more transaction programs may operate within a VTAM application program that is using the VTAM application program interface (API). In that situation, the transaction program would request services from the application program, using protocols defined by that application program. The application program, in turn, could request services from VTAM by issuing the APPCCMD macroinstruction.

**translation table**. A table used to replace one or more characters with alternative characters; for example, to translate characters representing a virtual address to those representing a real address, characters representing an event to those representing a procedure call, characters of a national character set to those of another national language, or characters representing a relocated address to those representing an absolute address.

**Transmission Control Protocol (TCP)**. A communications protocol used in the 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 uses the Internet Protocol (IP) as the underlying protocol.

**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 services (TS) profile**. In SNA, a specification in a session activation request (and optionally, in the responses) of transmission control (TC) protocols (such as session-level pacing and the usage of session-level requests) to be supported by a particular session. Each defined transmission services profile is identified by a number.

**transmission subsystem (TSS)**. A line adapter that attaches to the IBM 3745 Communication Controller.

**transmission subsystem component (TSC)**. The component of VTAM that comprises the transmission control, path control, and data link control layers of SNA.

**TRUE**. Task-Related User Exit.

**TSC**. (1) Telecommunications subsystem controller. (2) Transmission subsystem component.

**TSO**. Time Sharing Option.

**TSS**. Transmission subsystem.

**tuning**. The process of adjusting an application or a system to operate in a more efficient manner in the work environment of a particular installation.

**TWX**. Teletypewriter exchange service.

**type 2.1 node**. A node that can be an APPN network node, an APPN end node, or a LEN node. It can also attach as a peripheral node to a subarea boundary node in the same way as a type 2.0 node.

**T1**. In the United States, a 1.544-Mbps public access line. It is available in twenty-four 64-Kbps channels. The European version (E1) transmits 2.048 Mbps. The Japanese version (J1) transmits 1.544 Mbps.

# U

**UA**. Unnumbered acknowledgment.

**UNBIND**. In SNA, a request to deactivate a session between two logical units (LUs). See also *session deactivation request*. Contrast with *BIND*.

**UNC**. Universal naming convention.

**unformatted**. In VTAM, pertaining to commands (such as LOGON or LOGOFF) entered by an end user and sent by a logical unit in character form. The character-coded command must be in the syntax defined in the user's unformatted system services definition table. Synonymous with *character-coded*. Contrast with *field-formatted*.

**unformatted system services (USS)**. In SNA products, a system services control point (SSCP) facility that translates a character-coded request, such as a logon or logoff request, into a field-formatted request for processing by formatted system services and that translates field-formatted replies and responses into character-coded requests for processing by a logical unit. Contrast with *formatted system services*. See also *converted command*.

**uninterpreted name**. In SNA, a character string that a system services control point (SSCP) can convert into the network name of a logical unit (LU). Typically, an uninterpreted name is used in a logon or Initiate request from a secondary logical unit (SLU) to identify the primary logical unit (PLU) with which the session is requested.

**universal naming convention (UNC)**. In Communications Manager/2, a name used to identify the server name and the network name (netname) of a resource. This name is in one of the following formats:

```
\\servername\netname\path\filename
\\servername\netname\devicename
```

**unsolicited message**. A message, from VTAM to a program operator, that is unrelated to any command entered by the program operator. Contrast with *solicited message*.

**UP**. Unnumbered poll.

**upstream**. In the direction of data flow from the end user to the host. Contrast with *downstream*.

**user**. (1) Any person or any thing that may issue or receive commands and messages to or from the information processing system. (T) (2) Anyone who requires the services of a computing system.

**user exit**. (1) A point in an IBM-supplied program at which a user exit routine may be given control. (2) A programming service provided by an IBM software product that may be requested during the execution of an application program for the service of transferring control back to the application program upon the later occurrence of a user-specified event.

**user exit routine**. A user-written routine that receives control at predefined user exit points. User exit routines can be written in assembler or a high-level language.

**user table**. In TPNS, one or more text data entries contained in a table format, which may be referred to for logic testing and message generation.

**user variable**. See *USERVAR*.

**USERVAR**. A variable whose value is the name of the logical unit to which VTAM routes session-establishment requests.

**USS**. Unformatted system services.

# V

**value**. (1) A specific occurrence of an attribute; for example, “blue” for the attribute “color.” (T) (2) A quantity assigned to a constant, a variable, a parameter, or a symbol.

**variable**. (1) In programming languages, a language object that may take different values, one at a time. The values of a variable are usually restricted to a certain data type. (I) (2) A quantity that can assume any of a given set of values. (A) (3) A name used to represent a data item whose value can be changed while the program is running. (4) In the Simple Network Management Protocol (SNMP), a match of an object instance name with an associated value. (5) In the NetView command list language, a character string beginning with “&” that is coded in a command list and is assigned a value during execution of the command list.

**vector**. The MAC frame information field.

**version**. A separately licensed program that usually has significant new code or new function.

**view**. In IBM network management products, a graphical representation of a network or a part of a network. A view consists of resource symbols and resource labels; it may also include a background picture or text that a network operator has placed within it.

**virtual circuit**. (1) In packet switching, the facilities provided by a network that give the appearance to the user of an actual connection. (T) See also *data circuit*. Contrast with *physical circuit*. (2) A logical connection established between two DTEs.

**virtual machine (VM)**. (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). Contrast with *explicit route (ER)*. See also *path* and *route extension (REX)*.

**virtual route (VR) pacing**. In SNA, a flow control technique used by the virtual route control component of path control at each end of a virtual route to control the rate at which path information units (PIUs) flow over the virtual route. VR pacing can be adjusted according to traffic congestion in any of the nodes along the route. See also *session-level pacing*.

**Virtual 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.

**VS**. Virtual storage.

**VSE (Virtual Storage Extended)**. A system that consists of a basic operating system (VSE/Advanced Functions) and any IBM supplied and user-written programs required to meet the data processing needs of a user. VSE and the hardware it controls form a complete computing system. Its current version is called VSE/ESA.

**VSE/ESA**. Virtual Storage Extended/Enterprise Systems Architecture.

**VTAM**. (1) Virtual Telecommunications Access Method. (2) Synonymous with *ACF/VTAM*.

**VTAM application program**. A program that has opened an access method control block (ACB) to identify itself to VTAM and that can therefore issue VTAM macroinstructions.

**VTAM operator**. A person or program authorized to issue VTAM operator commands. See *domain operator*, *program operator*, and *network operator*.

**VTAM operator command**. A command used to monitor or control a VTAM domain. See also *definition statement*.

# W

**window**. (1) A portion of a display surface in which display images pertaining to a particular application can be presented. Different applications can be displayed simultaneously in different windows. (A) (2) An area with visible boundaries that presents a view of an object or with which a user conducts a dialog with a computer system. (3) In data communication, the number of data packets a data terminal equipment (DTE) or data circuit-terminating equipment (DCE) can send across a logical channel before waiting for authorization to send another data packet. The window is the main mechanism of pacing, or flow control, of packets. (4) See *pacing window*.

**window size**. The specified number of frames of information that can be sent before receiving an acknowledgment response.

**write-to-operator (WTO)**. An optional user-coded service that allows a message to be written to the system console operator informing the operator of errors and unusual system conditions that may need to be corrected.

**WTO**. Write-to-operator.

# 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**. (1) An International Telegraph and Telephone Consultative Committee (CCITT) recommendation for the interface between data terminal equipment and packet-switched data networks. (2) See also *packet switching*.

**X.25 interface**. An interface consisting of a data terminal equipment (DTE) and a data circuit-terminating equipment (DCE) in communication over a link using the procedures described in the CCITT Recommendation X.25.

**X.25 NCP Packet Switching Interface (NPSI)**. An IBM licensed program that allows SNA users to communicate over packet switching data networks that have interfaces complying with CCITT Recommendation X.25. It allows SNA programs to communicate with SNA or non-SNA equipment over such networks.

**X.25_1984**. CCITT Recommendation X.25 (Malaga-Torremolinos, 1984).

**X.28**. An International Telegraph and Telephone Consultative Committee (CCITT) recommendation for the DTE/DCE interface for a start-stop mode data terminal equipment (DTE) accessing the packet assembly/disassembly (PAD) facility in a public data network situated in the same country.

**X.29**. An International Telegraph and Telephone Consultative Committee (CCITT) recommendation for procedures for the exchange of control information and user data between a packet assembly/disassembly (PAD) facility and a packet mode data terminal equipment (DTE) or another PAD facility.

**X.3**. An International Telegraph and Telephone Consultative Committee (CCITT) recommendation for packet assembly/disassembly (PAD) in a public data network.

**X.32**. An International Telegraph and Telephone Consultative Committee (CCITT) recommendation that defines the interface between data terminal equipment (DTE) and packet-switching networks through a public switched network, such as a public telephone network.

**XA**. Extended architecture.

**XI**. X.25 SNA Interconnection.

**XID**. Exchange identification.

# 3

**37CS**. Deprecated term for *3746 Model 900 connectivity subsystem (CSS)*.

# Bibliography

## X.25 NCP Packet Switching Interface Publications

The following paragraphs briefly describe the library of books associated with X.25 NCP Packet Switching Interface Version 3.

*X.25 NPSI Version 3 Release 9 Data Areas* (LY30-5627)

This book is written for system programmers to assist them in troubleshooting and diagnosing problems with the X.25 NCP Packet Switching Interface. It provides control block information for NPSI.

*NCP Version 7 and X.25 NPSI Version 3 Diagnosis, Customization, and Tuning* (LY30-5610)

This book is written for system programmers to assist them in troubleshooting and diagnosing problems with the X.25 NCP Packet Switching Interface. It helps programmers to diagnose problems, resolve common errors, and describe problems to and interface with the IBM Support Center.

*X.25 NPSI Version 3 General Information* (GC30-3469)

This book introduces managers, system designers, programmers, and other data processing personnel to the basic concepts of packet-switching, X.25, and IBM's X.25 NCP Packet Switching Interface licensed program.

*X.25 NPSI Version 3 Host Programming* (SC30-3502)

This book is written for application and system programmers to assist them in writing application programs that use the X.25 NCP Packet Switching Interface. Application programmers should have some knowledge of DATE and GATE functions and the operating systems that support them. System programmers should be knowledgeable in SNA architecture.

*NCP Version 7 and X.25 NPSI Version 3 Planning and Installation* (SC30-3470)

This book provides system programmers and analysts with the information required to plan and implement the installation of NPSI. The topics include hardware and software requirements, preinstallation system performance considerations, instructions for defining and generating NPSI, and installation examples.

## Other Networking System Product Libraries

The following books provide cross-product information for NCP, NetView, and VTAM. For detailed information about these products refer to the library for each.

*ACF/NCP, ACF/SSP, EP, NPSI, and NTuneMON Softcopy Collection Kit* (LK2T-0414, CD-ROM)

*IBM Online Libraries: Softcopy Collection Kit User's Guide* (GC28-1700)

*Planning for Integrated Networks* (SC31-8062)

*Planning for NetView, NCP, and VTAM* (SC31-8063)

## NCP, SSP, and EP Library

The following publications apply to the NCP, SSP, and EP library.

*NCP and EP Reference* (LY43-0029)

*NCP and EP Reference Summary and Data Areas* (LY43-0030)

*NCP and SSP Customization Guide* (LY43-0031)

*NCP and SSP Customization Reference* (LY43-0032)

*NCP, SSP, and EP Diagnosis Guide* (LY43-0033)

*NCP, SSP, and EP Generation and Loading Guide* (SC31-6221)

*NCP, SSP, and EP Messages and Codes* (SC31-6222)

*NCP, SSP, and EP Resource Definition Guide* (SC31-6223)

*NCP, SSP, and EP Resource Definition Reference* (SC31-6224)

*NCP, SSP, and EP Trace Analysis Handbook* (LY43-0037)

*NCP V7R7 Migration Guide* (SC30-3889)

*NCP V7R7, SSP V4R7, and EP R14 Library Directory* (SC30-3971)

*NTuneNCP Feature Reference* (LY43-0039)

*NTuneMON V2R5 User's Guide* (SC31-6266)

## VTAM Library

The following list shows the books in the VTAM library.

*VTAM Customization* (LY43-0068)

*VTAM Data Areas for MVS/ESA* (LY43-0071)

*VTAM Diagnosis* (LY43-0069)

*VTAM Messages and Codes* (SC31-6546)

*VTAM Migration Guide* (GC31-6547)

*VTAM Network Implementation Guide* (SC31-6548)

*VTAM Operation* (SC31-6549)

*VTAM Programming* (SC31-6550)

*VTAM Programming for LU 6.2* (SC31-6551)

*VTAM Release Guide* (GC31-6555)

*VTAM Resource Definition Reference* (SC31-6552)

*VTAM Resource Definition Samples* (SC31-6554)

## NetView Library

The following list shows the books in the NetView library.

*TME 10 NetView for OS/390 NGMF User's Guide* (SC31-8234)

*TME 10 NetView for OS/390 User's Guide* (SC31-8241)

*TME 10 NetView for OS/390 Administration and Security Reference* (SC31-8222)

*TME 10 NetView for OS/390 Application Programmer's Guide* (SC31-8223)

*TME 10 NetView for OS/390 Automation Guide* (SC31-8225)

*TME 10 NetView for OS/390 Bridge Implementation* (SC31-8238)

*TME 10 NetView for OS/390 Command Reference* (SC31-8227)

*TME 10 NetView for OS/390 Customization Guide* (SC31-8228)

*TME 10 NetView for OS/390 Customization: Using Assembler* (SC31-8229)

*TME 10 NetView for OS/390 Customization: Using PL/I and C* (SC31-8230)

*TME 10 NetView for OS/390 Customization: Using REXX and the NetView Command List Language* (SC31-8231)

*TME 10 NetView for OS/390 Installation and Administration Guide* (SC31-8236)

*TME 10 NetView for OS/390 Messages* (SC31-8237)

*TME 10 NetView for OS/390 Diagnosis Guide* (LY43-0108)

*TME 10 NetView for OS/390 Resource Object Data Manager and GMFHS Programmer's Guide* (SC31-8233)

*TME 10 NetView for OS/390 Tuning Guide* (SC31-8240)

*TME 10 NetView for OS/390 NGMF User's Guide* (SC31-8234)

## NPM Library

The following list shows the books in the NPM library.

*NetView Performance Monitor at a Glance* (GH19-6960)

*NetView Performance Monitor Concepts and Planning* (GH19-6961)

*NetView Performance Monitor Desk/2 User's Guide* (SH19-6963)

*NetView Performance Monitor Diagnosis* (LY19-6381)

*NetView Performance Monitor Graphic Subsystem* (SH19-6967)

*NetView Performance Monitor Installation and Customization* (SH19-6964)

*NetView Performance Monitor Messages and Codes* (SH19-6966)

*NetView Performance Monitor Reports and Record Formats* (SH19-6965)

*NetView Performance Monitor User's Guide* (SH19-6962)

## SNA Publications

The following publications contain information on SNA:

*Systems Network Architecture Formats* (GA27-3136)

*Systems Network Architecture: Management Services Reference* (SC30-3346)

*Systems Network Architecture Technical Overview* (GC30-3073)

*X.25 DTE/DCE and DTE/DTE Interface Architecture Reference* (SC30-3409)

*X.25 1980 Interface for Attaching IBM SNA Nodes to Packet-Switched Data Networks: General Information Manual* (GA27-3345)

*X.25 1984/1988 DTE/DCE and DTE/DTE Interface General Information Manual* (GA27-3761)

## 372x Publications

The following list shows selected publications for the IBM 372x.

*3720/3721 Communication Controllers Configuration Guide* (GA33-0063)

*3720/3721 Communication Controllers Introduction* (GA33-0060)

*3725 Communication Controllers Principles of Operation* (GA33-0013)

## 3745 Publications

The following list shows the publications for the IBM 3745.

*IBM 3745 Basic Operations Guide for MOSS Operations* (SA33-0098)

*IBM 3745 Communication Controller Introduction* (GA33-0092 for 3745 Models 210, 310, 410 and 610)

*IBM 3745 Communication Controller Introduction* (GA33-0138 for 3745 Models 130, 150, and 170)

*IBM 3745 Communication Controller (all models): Principles of Operation* (SA33-0102)

*IBM 3745 Models 21A through 61A Migration and Planning Guide* (GA33-0183)

## CCITT Recommendations

The following book provides information about CCITT recommendations X.1–X.32.

*Blue Book, Volume VIII—Fascicle VIII.2, Data Communication Networks:  Services and Facilities, Interfaces, Recommendations X.1—X.32.*

## GOSIP

The following book provides information about the United States Government Open System Interconnection Profile (GOSIP).

*Federal Information Processing Standard (FIPS) 146-2*

## ISO Publications

The following books provide information about ISO standards 7776 and 8208.

*ISO 7776 Information Processing Systems—Data Communication— High-Level Data Link Control Procedures—Description of the X.25 LAPB-Compatible DTE Data Link Procedures*

*ISO 8208 Information Processing Systems—Data Communication—X.25 Packet Level Protocol for Data Terminal Equipment.*

# Index

# Tell Us What You Think!

**X.25 Network Control Program**
**Packet Switching Interface**
**Host Programming**
**Version 3**

**Publication No. SC30-3502-08**

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 representative 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!**
SC30-3502-08

Cut or Fold
Along Line

Fold and Tape **Please do not staple** Fold and Tape

NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

# BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

Design & Information Development
Dept. CGF/Bldg. 656
International Business Machines Corporation
PO BOX 12195
RESEARCH TRIANGLE PARK NC 27709-9990

Fold and Tape **Please do not staple** Fold and Tape

Cut or Fold
Along Line

File Number: S370/4300-30
Program Number: 5688-035

Printed in the United States of America
on recycled paper containing 10%
recovered post-consumer fiber.

SC30-3502-08