SC33-0074-2

Customer Information Control System/Virtual Storage (CICS/VS) Version 1 Release 5

Program Product

IBM 3767, 3770, and 6670 Guide

Program Numbers 5740-XX1 (CICS/OS/VS)
5746-XX3 (CICS/DOS/VS)

IBM

| Third Edition (May 1980)

| This edition applies to Version 1 Release 5 (Version 1.5) of the IBM
| program product Customer Information Control System/Virtual Storage
| (CICS/VS), program numbers 5746-XX3 (for DOS/VS) and 5740-XX1 (for
| OS/VS). Until the OS/VS version is released, the information applicable
| to that version is for planning purposes only.

This edition is based on the CICS/VS Version 1.4.1 edition, and changes from that edition are indicated by vertical lines to the left of the changes. Note, however, that the 1.4.1 edition remains current and applicable for users of Version 1.4.1 of CICS/VS.

Information in this publication is subject to change. Changes will be published in new editions or technical newsletters. Before using this
| publication, consult the latest IBM System/370 and 4300 Processors Bibliography, GC20-0001, to learn which editions and technical newsletters are current and applicable.

It is possible that this material may contain references to, or information about, IBM products (machines and programs), programming, or services that are not announced in your country. Such references or information must not be construed to mean that IBM intends to announce such IBM products, programming, or services in your country.

Publications are not stocked at the addresses given below; requests for copies of IBM publications should be made to your IBM representative or to the IBM branch office serving your locality.

A form for reader's comments is provided at the back of this publication; if the form has been removed, comments may be addressed either to:

| International Business Machines Corporation,
| Department 812HP,
| 1133 Westchester Avenue
| White Plains, New York 10604.

| or to:

| IBM United Kingdom Laboratories Limited,
| Programming Publications, Mail Point 095,
| Hursley Park,
| Winchester, Hampshire SO21 2JN, England.

Preface

This book provides information for CICS/VS users who intend to install a CICS/OS/VS or CICS/DOS/VS system that communicates with an IBM 3767 Communication Terminal, an IBM 3770 Communication System, or an IBM 6670 Information Distributor. Brief information is also given on 3767 support for certain devices connected via the Network Terminal Option of NCP/VS. The book is directed to system designers, system programmers, and application programmers.

Scope of this Book

This book is intended to supplement CICS/VS publications, 3767 Communication Terminal publications, 3770 Communication System publications, and IBM 6670 Information Distributor publications. It is assumed that the reader is familiar with the standard CICS/VS facilities that are provided for communication with remote subsystems, and also with the principles of operation of the subsystems and their host communication facilities. Some familiarity with IBM Systems Network Architecture is also assumed.

Except where stated, the information in this book applies only to a CICS/VS system that is part of a Systems Network Architecture (SNA) network and employs VTAM (Virtual Telecommunications Access Method). For details of the support that CICS/VS provides when TCAM (Telecommunications Access Method) is used, refer to the CICS/VS System Programmer's Reference Manual.

Note: References to CICS/VS application programming in this book are, in general, made in terms of the command-level interface. Lists of macro-level/command-level equivalents are given in the CICS/VS command-level and RPGII Application Programmer's Reference Manuals.

Organization of this Book

The manner in which application programs are designed and coded for a CICS/VS system that communicates with an IBM 3767, 3770, or 6670 subsystem is determined to some extent by the type of support CICS/VS provides. CICS/VS support defines several types of logical unit that have characteristics different from each other. Accordingly, this publication consists of an introductory chapter followed by a series of chapters describing the procedures that should be followed in designing and coding application programs for each of the logical unit types.

The book contains the following sections:

"Chapter 1: Introduction", which introduces the basic concepts and requirements of a CICS/VS-3767 , CICS/VS-3770, or CICS/VS-6670 system.

"Chapter 2: The Full Function Logical Unit", which describes CICS/VS support for online programmable models of the 3770.

"Chapter 3: The Interactive Logical Unit", which describes CICS/VS support for interactive communication between CICS/VS and a 3767/3770

terminal. Mention is also made of certain other devices that appear to CICS/VS as a 3767 interactive logical unit.

"Chapter 4: The Batch Logical Unit", which describes CICS/VS support for batch transfers of data between CICS/VS and a 3770 terminal.

"Chapter 5: The Batch Data Interchange Logical Unit", which describes CICS/VS support for batch transfers of data between CICS/VS and programmable models of the 3770 when the facilities of the CICS/VS batch data interchange program are used.

"Chapter 6: The Type 4 Logical Unit", which describes CICS/VS support for SNA Type 4 Logical Unit, and, in particular, for the 6670 Information Distributor.

"Appendix A: Sample Program for the Full Function LU", which describes the sample 3770 online communication program distributed with CICS/VS.

"Appendix B: Sample CICS/VS Program for the Batch LU", which describes the sample CICS/VS application program (distributed with CICS/VS) for the 3770.

"Appendix C: System Sense Codes Sent By CICS/VS", which lists the sense codes that the 3770 programmer may receive from CICS/VS.

"Appendix D: BIND Formats", which discusses the binds used by CICS/VS for the various 3767, 3770, and 6670 logical unit types.

Related Publications

The following publications contain additional information needed when installing and using a CICS/VS-3767/3770 system:

- Customer Information Control System/Virtual Storage (CICS/VS):
  - System/Application Design Guide, SC33-0068.
  - Application Programmer's Reference Manual (Command Level), SC33-0077.
  - Application Programmer's Reference Manual (RPGII), SC33-0085.
  - Application Programmer's Reference Manual (Macro Level), SC33-0079.
  - System Programmer's Guide (DOS/VS), SC33-0070.
  - System Programmer's Guide (OS/VS), SC33-0071.*
  - System Programmer's Reference Manual, SC33-0069.
  - Operator's Guide, SC33-0080.
  - Messages and Codes, SC33-0081.
  - Master Index, SC33-0095.*
- IBM 3767 Models 1 and 2 Communication Terminal Component Description, GA27-3096
- IBM 3770 Data Communication System: System Components, GA27-3097

- IBM 3773, 3774, and 3775 Programmable Communication Terminals Programmer's Guide, GC30-3028
- ACF/VTAM Concepts and Planning, GC38-0282
- Introduction to the IBM 3704 and 3705 Communications Controllers, GA27-3051
- Systems Network Architecture: Reference Summary, GA27-3136
- Systems Network Architecture: Functional Description of Logical Unit Types, GC20-1868
- Systems Network Architecture: Types of Logical Unit to Logical Unit Sessions, GC20-1869

* Available at the same time as CICS/OS/VS Version 1 Release 5

Contents

Figures

Summary of Amendments for Version 1 Release 5

The following technical changes have been made to cover new support provided by CICS/VS Version 1 Release 5:

- The guide describes CICS/VS Release 1 Version 5 support for certain non-SNA (System Network Architecture) devices which are connected through ACF/VTAM and the Network Terminal Option (NTO) of NCP/VS.
- References to the Application Programming Interface (API) have been changed to describe the Command Level Interface instead of the Macro Level Interface.

Chapter 1. Introduction

This chapter provides an overview of a teleprocessing system that contains the IBM Customer Information Control System/Virtual Storage (CICS/VS) and the IBM 3767 Communication Terminal, IBM 3770 Data Communication System, or the 6670 Information Distributor. The chapter introduces the major components of such a system. The information about CICS/VS support for the 3767 also applies to Common Carrier Teletypewriter Exchange (TWX) and World Trade Teletypewriter (TLX) terminals running under the Network Expansion Terminal Option program (NETO).

The IBM Customer Information Control System/Virtual Storage (CICS/VS) can be used in conjunction with remote terminals or subsystems to provide a general-purpose teleprocessing system. The whole system runs under Virtual Storage Extended (VSE), the IBM Operating System/Virtual Storage 1 (OS/VS1), or the IBM Operating System/Virtual Storage 2 (OS/VS2).

The IBM 3767 Communication Terminal is a general-purpose keyboard-printer terminal. The IBM 3770 Data Communication System is a family of keyboard-printer terminals and attachable I/O devices. The IBM 6670 is a printer/copier device with facilities for magnetic card input and output.

CICS/VS is a general-purpose data base/data communication control system that provides the necessary support for an advanced teleprocessing network.

Note: In this chapter, the term "CICS/VS-SNA system" refers to a CICS/VS-3767 system, a CICS/VS-3770 system, or a CICS/VS-6670 system.

System Components

When operated online, a CICS/VS-SNA system requires the following components:

- CICS/VS and its application programs in the host processor
- The Virtual Telecommunications Access Method (VTAM)
- An operating system (VSE, OS/VS1, or OS/VS2)
- IBM 3704 and/or 3705 Communications Controllers and Network Control Programs/Virtual Storage (NCP/VS)
- An SNA terminal or subsystem

The relationship of the components to each other is shown in Figure 1-1.

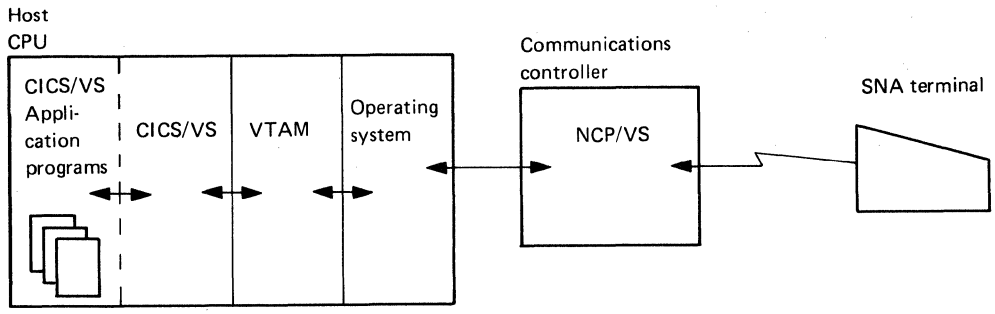


Figure 1-1. Major Components of a CICS/VS-SNA System

Planning the system requires an understanding of each component and of its relationship to the others. This information is provided in the following sections.

CICS/VS AND ITS APPLICATION PROGRAMS

CICS/VS is an IBM System/370 program product that controls online data base/data communication (DB/DC) applications. CICS/VS provides (1) most of the standard functions required by application programs for communication with remote and local terminals and subsystems; (2) control for concurrently running user application programs serving many online users; and (3) a full data base capability. CICS/VS, and the application programs under its control, run under any System/370 virtual storage operating system - VSE, OS/VS1, or OS/VS2.

CICS/VS application programs provide whatever processing logic the user requires. These user-written application programs use CICS/VS command-level statements or macro instructions to request the services of CICS/VS.

For a full description of CICS/VS, see the CICS/VS General Information manual.

VTAM

VTAM controls the allocation of the network resources and enables these resources to be shared among VTAM's own users.

To VTAM, CICS/VS is a single VTAM user; VTAM is unaware of the CICS/VS application programs. Thus, the CICS/VS application programs use CICS/VS commands or macro instructions to request CICS/VS services; in turn, CICS/VS uses VTAM macro instructions to invoke VTAM facilities.

For a description of VTAM and how it is used, see VTAM Concepts and Planning.

OPERATING SYSTEM

CICS/VS, its application programs, and VTAM use the services of the operating system to perform activities such as input/output, storage management, and task management.

When the operating system is generated, VTAM must also be generated.

COMMUNICATIONS CONTROLLER AND THE NCP/VS

The IBM 3704 and 3705 Communications Controllers provide communications network control by means of the NCP/VS. VTAM uses the facilities of the NCP/VS to control lines and devices attached to the controllers, to transmit data between the 3767/3770/6670 and the host processor, to perform error recovery, and to collect statistics about the network.

For additional information on the communications controllers and NCP/VS, see Introduction to the IBM 3704 and 3705 Communications Controllers.

SNA TERMINALS

The IBM 3767 Communication Terminal is a movable desk-top terminal available in a number of different models. It is designed primarily for interactive communication with a host processor.

The IBM 3770 Data Communication System is a family of multi-purpose terminals and I/O devices. Programmable models of the 3770, including an online programmable communications model, are available.

The IBM 6670 Information Distributor is a member of the IBM Office System/6 family of products and is designed for word-processing and data-processing applications. It comprises a laser printer, an electrophotographic copier, a magnetic card reader/recorder, and a processor. It serves as a text processor, as a convenience copier, as a high-quality printer, and as a communication terminal.

A full list of supported terminals is given in the CICS/VS General Information manual.

The CICS/VS-SNA System

To CICS/VS, the host communication interfaces of 3767, 3770, and 6670 devices appear as secondary logical units in the SNA network. Each logical unit is defined to CICS/VS by the system programmer through the DFHTCT TYPE=TERMINAL macro instruction.

Various types of logical unit may be defined; each implying a particular SNA protocol that will be used for sessions between CICS/VS and the terminal. The types of logical unit described in this book are:

- The full function LU

  This logical unit, described in chapter 2, provides communication between CICS/VS and online programmable models of the 3770. It enables the maximum function of 3770 programmable communications to be realized in a CICS/VS-3770 system.

- The interactive LU

  This logical unit, described in chapter 3, provides communication between CICS/VS and the 3767. It may also be used with some models of the 3770, and with certain other devices that appear to CICS/VS as 3767 interactive logical units. Either flip-flop or contention mode may be selected for the interactive logical unit. Other devices that use the same support include the Common Carrier Teletypewriter Exchange (TWX) and World Trade Teletypewriter (TLX) terminals running under the Network Expansion Terminal Option program (NETO).

- The batch LU

  This logical unit, described in chapter 4, provides batch-mode communication between CICS/VS and 3770 terminals.

- The batch data interchange LU

  This logical unit, described in chapter 5, provides communication between CICS/VS and programmable models of the 3770. CICS/VS application programs that use this LU have all the facilities provided for batch LU support, and may also employ batch data interchange commands or macro instructions.

- The type 4 LU

  This logical unit, described in Chapter 6, provides batch-mode communication between CICS/VS and terminals using the SNA logical unit type 4 protocols. CICS/VS application programs that use this LU have all the facilities provided for batch LU support for data processing media. In addition, word processing media may be selected, and Batch Data Interchange commands, as well as BMS commands, may be used. Batch LU support is compatible with LU4 support for data processing media.

A 3767 terminal is always associated with an interactive logical unit.

Any model of the 3770 family can be associated with the batch logical unit or the batch data interchange logical unit. Models of the 3770 with keyboard/printer-to-line capability can also be associated with the interactive logical unit.

For non-programmable and offline-programmable models of the 3770, the protocol (batch, batch data interchange, or interactive) must be

selected by the system programmer when the DFHTCT TYPE=TERMINAL macro is coded; it cannot be modified dynamically.

The online-programmable model of the 3770 contains two logical units. One of these logical units supports full function LU protocols; the other supports batch, batch data interchange, or interactive protocols. However, only one of the two logical units can be in session with CICS/VS at any one time.

The functions available to 3770 online programs include:

- Editing and verifying data received from the 3770 keyboard.
- Reading and writing to the host processor
- Editing and verifying data received from the host processor
- Formatting, displaying, and printing data sets
- Operate offline when the host processor, VTAM, CICS/VS, or NCP/VS is unavailable

A 6670 Information Distributor is always associated with a type 4 logical unit.

The remaining chapters in this manual describe the logical units in more detail.

Operation of a CICS/VS-SNA System

The basic operation of a CICS/VS-SNA system involves:

- Establishing sessions between CICS/VS and logical units
- Sending data between CICS/VS application programs and these logical units
- Terminating sessions between CICS/VS and logical units

A session may be initiated by the logical unit, by the VTAM network operator, by the CICS/VS master-terminal operator, automatically by VTAM, or by CICS/VS itself. CICS/VS uses VTAM's simulated logon (SIMLOGON) macro to initiate a logon on behalf of logical units such as those that:

- Contain output-only terminals
- Are defined by the user as secure terminals, to which access is controlled by CICS/VS

Logical units for which CICS/VS is to initiate logons are identified to CICS/VS through the terminal control table (TCT). CICS/VS also uses the SIMLOGON macro instruction to obtain logical units that are requested by the master-terminal operator, or that are involved in an automatic transaction initiation (ATI), but are not currently connected.

Once a logical unit is connected to CICS/VS, it remains connected until:

- Another VTAM application program requests connection to it (if specified by the CICS/VS system programmer).

- The logical unit itself requests disconnection.
- The CICS/VS master terminal operator requests disconnection.
- CICS/VS, VTAM, the NCP/VS, the logical unit, or the entire system is deactivated.
- The CICS/VS application program requests disconnection.

CICS/VS uses VTAM's close destination (CLSDST) macro instruction with the RELEASE option to disconnect logical units.

While a transaction is in progress, the CICS/VS application program uses CICS/VS commands or macro instructions to request CICS/VS services (including data transfer services for the logical unit). CICS/VS, in turn, uses VTAM record-mode macro instructions to request the data transfer operations from VTAM. VTAM adds routing information to the data stream and uses the facilities of the operating system and NCP/VS to transmit the requests to the appropriate logical unit.

The interfaces between components in a CICS/VS-SNA system are shown in Figure 1-2.

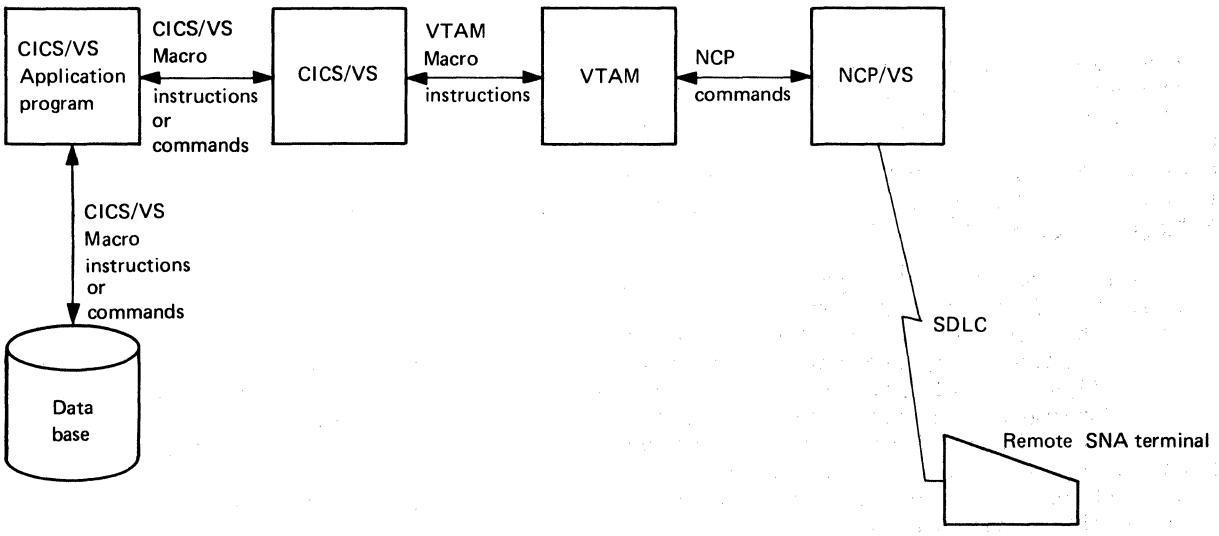


Figure 1-2. Data Flow and Interfaces in a CICS/VS-SNA System

Chapter 2. The Full Function Logical Unit

Introduction

The 3770 full function logical unit is designed to enable the 3770 user to add, change, and delete CICS/VS data base records with data base integrity. The full function LU thus extends to the 3770 user the full capability of data base inquiry and updating from remote operator stations. The major facilities that the full function LU gives the CICS/VS-3770 user are:

- Data base update
- Host-initiated sessions
- Host-initiated transactions
- Message resynchronization and re-presentation
- Chain assembly
- Logical record presentation (for SCS data streams)
- Message integrity
- Basic mapping support (BMS) for SCS data streams

CICS/VS Application Programming

Although CICS/VS management modules use SNA (VTAM) commands to control the session with the 3770 full function LU, the CICS/VS application programmer does not need to concern himself with these commands. He requests terminal services through the RECEIVE and SEND commands. These commands, and other commands applicable to terminal control, are described in detail in the CICS/VS Application Programmer's Reference Manual (Command Level).

Both the CICS/VS application programmer and the 3770 programmer enjoy complete freedom of data format. The conventions are determined by these two programmers; CICS/VS will accept any combination of bits. As a consequence of the fact that the data stream format will not be known to CICS/VS, full BMS support is not available through the full function LU. However, in order to provide the user with message switching services, limited BMS support is provided. If the CICS/VS application program uses BMS commands, the data stream generated by CICS/VS and passed to the 3770 will be in SNA character string (SCS) format. When assembling his maps, the user must specify TERM=SCS in the DFHMSD macro.

The use of a function management header (FMH) in communications between CICS/VS and the full function LU is optional in both directions. If an FMH is required in an outbound message, it must be built by the CICS/VS application program; no FMHs will be generated by CICS/VS. On input, CICS/VS will accept an inbound FMH and either pass it on to the CICS/VS application program or discard it (according to the specification made by the system programmer in the PCT).

CICS/VS System Programming

The CICS/VS system programmer defines the full function LU by coding SESTYPE=USERPROG in addition to TRMTYPE=3770P in the DFHTCT TYPE=TERMINAL macro.

3770 Programming

The 3770 Programmable Communications feature provides the 3770 programmer with a host communication facility – a programming interface that gives the 3770 programmer the ability to issue all the SNA commands for communication with a logical unit in the host, in this case CICS/VS. Since the 3770 programmer has direct control over the issue and receipt of SNA commands, he will need to be aware of the way that CICS/VS expects these commands to be used. The section, "application programming for the full function logical unit", later in this chapter discusses and illustrates how programs should be constructed to observe the protocols associated with these commands.

The sample 3770 program supplied with CICS/VS provides an example of how the 3770 host communication statements are used for communication with CICS/VS. Further details are given in appendix A.

There are two features associated with the full function LU that require awareness and action from the 3770 programmer: host-initiated sessions, and host-initiated transactions.

Host-initiated sessions mean that CICS/VS can initiate a session with a 3770 full function LU without receiving a logon request. The 3770 then activates the program associated with that LU. The 3770 program must thus contain the logic necessary to recognize and deal with the unsolicited request sent from the host to initiate the session.

When CICS/VS initiates a transaction, it sends an SNA BID command to the 3770 to request permission to begin message transmission. The 3770 program must contain the logic necessary to recognize and respond to the BID command.

CICS/VS service routines, such as the sign-on/sign-off and master terminal functions, use the SCS data stream when sending messages to the full function LU. Since both BMS and other CICS/VS services use SCS, and in particular the NL (new line) control character, the 3770 program must contain the logic necessary to recognize and process this character. The sample 3770 program contains an example of such logic.

Another facility for which the 3770 program must contain the appropriate logic is message resynchronization and re-presentation. The 3770 program must be capable of handling sequence numbers and of receiving and dealing with the SNA set-and-test-sequence-numbers (STSN) command.

As described earlier for the CICS/VS application programmer, the 3770 programmer enjoys complete freedom of data formats, subject only to the conventions agreed between him and the CICS/VS application programmer.

If an inbound FMH is required, it must be built by the 3770 program.

Application Programming for the 3770 Full Function LU

The purpose of this section is to show how a complementary pair of programs must be written to ensure orderly communication between CICS/VS and a full function LU in a 3770 terminal. The pair of programs are known as the CICS/VS application program and the 3770 program. Orderly communication requires the use of control information. Control information exists at two levels. There are SNA commands, which can be sent by either session partner to request control actions, and SNA control indicators, which are sent as part of a message to inform a session partner on the state of the dialog.

Although a 3770 program communicates with a CICS/VS application program, CICS/VS itself retains much of the responsibility for sending and receiving control information. The CICS/VS application programmer is thus shielded from having to code such information in his program. However, this is not true for the 3770 programmer who must be aware of how CICS/VS expects a session with a 3770 program to be controlled. This section therefore, by means of representative examples, describes how application programs should be constructed to ensure that synchronization of data flow can be maintained.

SNA Control Indicators

Messages are transmitted between application programs in units of data called Request/Response Units (RUs). The SNA control indicators are transmitted in a header that precedes each RU and is called the Request/Response Header (RH). The control indicators in an RH that can be set and tested during a session between CICS/VS and a 3770 program are as follows:

BB	Begin-Bracket
EB	End-Bracket
RQD1	Request-Definite-1
RQE1	Request-Exception-1
DR1	Definite-Response-1
DR2	Definite-Response-2
BC	Begin-Chain
EC	End-Chain
CD	Change-Direction
FMH	Function Management Header at beginning of RU

The meaning and use of these indicators is explained later in this section. In practice, neither the CICS/VS application programmer nor the 3770 programmer builds an RH directly. Instead, the 3770 programmer is provided with programming statements that request setting or testing of these indicators on a particular data transmission. Similarly, the CICS/VS application programmer controls these indicators by specifying options on CICS/VS commands. However, the CICS/VS application programmer is normally much less involved with setting and analyzing RH indicators than the 3770 programmer.

SNA Commands

SNA commands are special transmissions that control the session and the flow of information within a session. Unlike SNA indicators, they are not transmitted with user data. In fact, an SNA command includes an RH, together with a code identifying the command type, and, depending on the type of command, command data. The SNA commands that may be used in a session are determined by the SNA BIND command, which is used to establish a session. In communication between CICS/VS and a 3770 program, some SNA commands are restricted for use by CICS/VS, some by the 3770 program, or some may be used by both. Figure 2-1 defines the SNA commands and transmission direction permitted between CICS/VS and a 3770 program. The meaning and use of these SNA commands is described later in this section. As with RH indicators, application programmers use programming statements and commands to cause SNA commands to be transmitted. However, since SNA commands are largely concerned with maintaining orderly communication between session partners, CICS/VS normally sends and responds to them without intervention by a CICS/VS application program.

SNA session control commands	CICS/VS to 3770	3770 to CICS/VS
BIND	Y	
CLEAR	Y	
INITIATE-SELF		Y
REQUEST-RECOVERY (RQR)		Y
SET-AND-TEST-SEQUENCE-NUMBERS (STSN)	Y	
START-DATA-TRAFFIC (SDT)	Y	
TERMINATE-SELF		Y
UNBIND	Y	
SNA data flow control commands		
BID	Y	
CANCEL	Y	Y
CHASE	Y	Y
LOGICAL-UNIT-STATUS (LUSTAT)	Y	Y
QUIESCE-AT-END-OF-CHAIN (QEC)	Y	
QUIESCE-COMPLETE (QC)		Y
READY-TO-RECEIVE (RTR)		Y
RELEASE-QUIESCE (RELQ)	Y	
REQUEST-SHUTDOWN (RSHUTD)		Y
SHUTDOWN (SHUTD)	Y	
SHUTDOWN-COMPLETE (SHUTC)		Y
SIGNAL	Y	Y

Figure 2-1. SNA Commands

A SIMPLE CICS/VS TRANSACTION USING THE FULL FUNCTION LOGICAL UNIT

Figure 2–2 illustrates how a 3770 program and a CICS/VS application program communicate using the SNA control indicators. The figure also illustrates the various protocols associated with the control indicators.

Figure 2–2 and subsequent figures in this section illustrate the protocols using flow sequence diagrams. The following conventions are used in these figures:

- Vertical lines represent interfaces. The principal interfaces are CICS/VS and the 3770 program; these are represented by continuous lines. Only those interfaces that are pertinent to an example are included.
- Horizontal lines represent flow direction of RUs and messages. Request units are represented by continuous lines and are labeled either with the pertinent SNA indicators being sent or with an SNA command type. Response units are represented by dotted lines and are labeled either positive (+ resp) or negative (– resp). A negative response also includes the system sense codes that are sent.

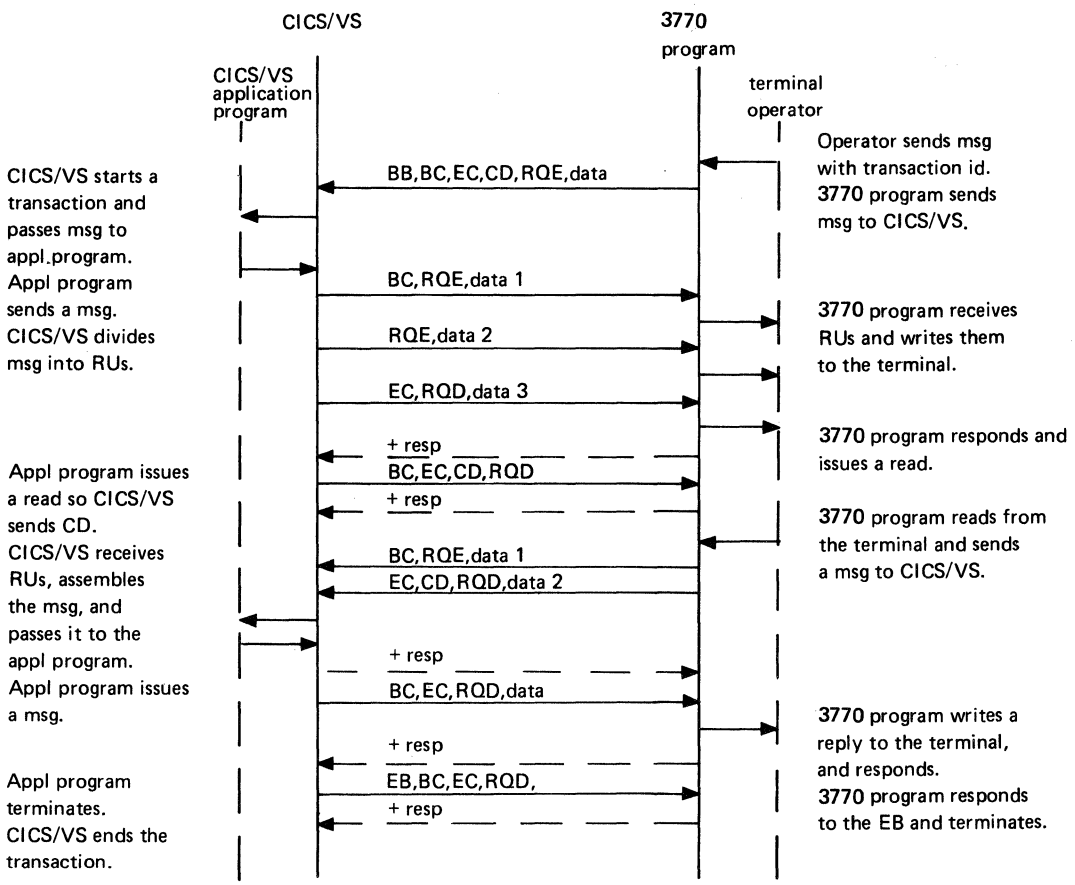


Figure 2–2. A Simple Transaction Using the Full Function LU

This example assumes that the CICS/VS application program is to be invoked by a message received initially by the 3770 program. The CICS/VS application program will expect further input from the 3770 program. The CICS/VS application program will write a short reply and terminate. For this example it is assumed that the CICS/VS system programmer has requested that all output from CICS/VS be sent requesting a definite response from the 3770 program upon successful processing of the message by coding DFHPCT TYPE=OPTGRP,MSGPREQ=MSGINTEG.

Processing commences with the 3770 terminal operator invoking the 3770 program.

The 3770 program issues a read request for initial input. Upon receipt of the input it finds that it must pass the message to CICS/VS (the assumption being that the data also contains the appropriate CICS/VS transaction identification). The 3770 program requests that the 3770 initiate a session with CICS/VS. The 3770 program must ensure that the 3770 uses a logical unit that CICS/VS recognizes as a full function logical unit. The logical unit number must also be recorded by the 3770 program so that, in the event of a session failure, the 3770 program can reestablish the session on the same logical unit and participate in message resynchronization. (Message resynchronization is discussed later in this chapter.)

Once the session has been established, the 3770 program sends the initial message to CICS/VS. The example assumes that the initial message is not greater than 256 bytes and therefore is sent as a single RU with both the begin-chain (BC) and end-chain (EC) indicators. Because the message is initiating a CICS/VS application program, the begin-bracket (BB) indicator is also required so that CICS/VS will recognize that a transaction identification is included. Since the 3770 program is written to expect a reply from CICS/VS, it decides not to ask for definite response to the message. Thus the request-exception (RQE) indicator is included. In addition, since the 3770 program is expecting to receive a reply, the change-direction (CD) indicator is set to give control of the session to CICS/VS.

When CICS/VS receives the message, it inspects the transaction identification and attaches the appropriate CICS/VS application program. The message is passed to the CICS/VS application program in a TIOA.

The CICS/VS application program processes the message and issues a write. The example assumes that the message exceeds 512 bytes. CICS/VS thus breaks the message up into a chain of RUs. The initial RU is sent with the begin-chain (BC) indicator. Because a chained message is not considered to be delivered until all elements of the chain have been sent, CICS/VS requires a response to an element of a chain only if an error is detected by the 3770 program. Thus all RUs, except the last, carry the request-exception (RQE) indicator. The final RU carries the end-chain (EC) indicator. (Note that an element is assumed to be within a chain when neither the BC nor EC indicators are coded.) Because the system programmer specified that definite responses are required, the final RU carries the request-definite (RQD) indicator.

The 3770 program reads each RU and writes it to the terminal, performing whatever terminal dependent processing is necessary to produce a readable format at the terminal. Following successful output of the message, it writes the definite response to CICS/VS.

The 3770 program issues another read request for a message from CICS/VS because the last message did not contain the CD indicator.

Following receipt by CICS/VS of the definite response, the CICS/VS application program is passed control at the next instruction following the write request. The CICS/VS application program issues a read to

receive the next input from the terminal. This causes CICS/VS to send an RU indicating that the CICS/VS application program is now prepared to receive data. The RU does not contain data but has the BC, EC, CD, and RQD indicators set. Upon receiving a chain containing a CD indicator, the 3770 program must not expect to receive more data from the CICS/VS application program until it has sent data. However, before the 3770 program sends any data to the CICS/VS application program it must send a response to CICS/VS, since a definite response was requested.

The 3770 program issues another read to the terminal. The example assumes that, as a result, a message longer than 256 bytes must be sent to CICS/VS. The message is sent as a two element-chain. Note that, apart from the fact that it is a two-element chain rather than a three-element chain, the process is equivalent to that described above for the data transmitted to the 3770 program. However, the CD indicator is included with the final RU because, in the example, the 3770 program expects a reply.

CICS/VS assembles the chain into a single message and passes it to the CICS/VS application program to satisfy the read request. The chain assembly process is specified by the system programmer using the DFHTCT TYPE=TERMINAL, CHNASSY=YES. This option is recommended for use with the full function LU.

Following receipt of the message, the CICS/VS application program performs further processing and replies to the 3770 program. CICS/VS sends the message as a single RU chain and requests definite response. Control returns immediately to the CICS/VS application program, which may continue processing until a wait is issued. The 3770 program receives the message and writes it to the terminal. It then writes a positive response to CICS/VS and issues a read request to await further messages (if any).

The positive response from the 3770 program causes CICS/VS to return control to the CICS/VS application program following the wait request. The CICS/VS application program performs some cleanup processing and then terminates. This causes CICS/VS to send a single RU to the 3770 program. The RU carries the BC, EC, RQD, and EB indicators. The 3770 program must send a definite response to CICS/VS to acknowledge receipt of the RU.

The End-Bracket (EB) indicator instructs the 3770 program that it is now free to perform any of the following:

- Monitor the terminal for further input which may result in the 3770 program commencing another transaction.
- Terminate, so closing the session with CICS/VS.
- Monitor for unsolicited messages from CICS/VS resulting from transactions commencing as a result of CICS/VS Automatic Transaction Initiation.

RESPONSE PROTOCOL

Response protocol enables the sender of a RU to verify that the RU actually reached its destination. Two types of response protocol are used by CICS/VS, definite and exception. Note that the SNA no-response protocol is not supported by CICS/VS.

Each RU transmitted between CICS/VS and a logical unit requests either definite or exception response protocol. With definite response

protocol, the receiver of a RU must return a positive response if the RU is received correctly. If the RU is not received correctly, a negative response must be returned. With exception response protocol, the receiver of the RU must return a response only if the RU is not received correctly. In this case, the response would be a negative response. A positive response is not allowed with exception response protocol; if a RU arrives successfully, no response is returned. Each RU transmitted between CICS/VS and a logical unit must request a response protocol.

Along with the two types of response and the two response protocols, CICS/VS supports the two levels of response DR1 and DR2. Use of the two levels is decided by the session partners who may attach significance to the two levels. For example, a program may choose to send a DR2 response when it has received, validated, and enqueued a message for subsequent processing, and a DR1 response when it has received, validated, and processed a message. The level of response chosen can be significant where such a distinction can be used to take appropriate error recovery actions.

Transmissions from CICS/VS normally request a DR1 response unless overridden by the RESP operand of the DFHTCT TYPE=INITIAL macro instruction. CICS/VS responds as requested to by a 3770 program.

Note that all the examples given in this section assume that only the DR1 response is used and therefore it is not shown in the flow sequences. Unless special user conditions apply, the DR1 level of response should always be chosen.

Responses to SNA Commands

CICS/VS always requests definite response protocol for any SNA command. The 3770 program must be prepared to respond to all commands except STSN, SDT, and UNBIND, which are handled automatically by the 3770 on behalf of the logical unit. (In some circumstances, the 3770 program is also required to indicate how the 3770 should respond to BIND command.)

There are circumstances under which the 3770 will respond to the BID and CLEAR commands. Circumstances in which the 3770 program must respond are discussed later in this chapter.

Definite response protocol should be requested whenever the 3770 program sends a message containing an SNA command to CICS/VS.

Responses to Data Received from CICS/VS

CICS/VS may send data with either definite or exception response protocol. Data is sent with exception response protocol requested whenever message protection or message integrity is not specified in the program control table (PCT) entry for the transaction. If a negative response is returned to data transmitted by CICS/VS with exception response protocol requested, retransmission of the message is not possible as CICS/VS cannot guarantee the availability of the TIOA from which the message was transmitted.

Data is sent with definite response protocol requested whenever message protection or message integrity is specified in the PCT entry for the transaction. If an output message is chained for a transaction specifying message integrity or protection, each element except the last has exception response protocol requested. The last element has

definite response protocol requested. As a good practice, the 3770 program should attempt the operation requested and then respond according to the completion status of the operation. If a negative response is returned for a message or element of a message, retransmission of the complete message (all elements in a chain) may be attempted by the user node error program (ZNEP) in CICS/VS.

Any RU sent from CICS/VS and containing the change-direction (CD) indicator (discussed below) is sent with definite response requested.

A CICS/VS application program may request definite response protocol by use of the DEFRESP option of the SEND or CONVERSE command. In this way, the PCT option can be overridden.

Responses to Data Sent to CICS/VS

The 3770 program is responsible for requesting a response protocol and level of response for messages sent to CICS/VS. Definite or exception response protocol may be requested with DR1 or DR2 response level. The response protocol and level requested does not alter the way in which CICS/VS processes the message.

With definite response protocol requested, CICS/VS may return a positive or negative response to a message. A positive response returned during a conversational transaction indicates that CICS/VS received the message, but the message may still be backed out if a system failure occurs. A positive response returned for a one-message input-only transaction indicates that CICS/VS has successfully completed the transaction.

Any negative response sent by CICS/VS is accompanied by four bytes of sense information. This sense information defines the reason for the negative response in the first two bytes and provides the CICS/VS error message number (if relevant) in the last two bytes. The 3770 program should interpret the sense information and take appropriate action to notify the terminal operator. See "Handling Errors" later in this chapter for the sense information format and techniques for using the information.

DATA TRANSMISSION PROTOCOL

Communication between CICS/VS and the full function logical unit is designed to use the half-duplex flip-flop (HDX FF) protocol. In this protocol, one session partner is first the sender and, after sending the change-direction (CD) indicator, becomes the receiver. The other session partner is now the sender. A receiver may not send data, although it may send responses and some types of command. If a receiver attempts to send data before receiving the CD indicator, the request may be rejected with a negative response.

If a receiver must send data urgently, the SNA SIGNAL command may be sent. The SIGNAL command is discussed in the section, "Interrupting Transmission between CICS/VS and the Logical Unit," later in this chapter.

DATA CHAINING

During system generation, CICS/VS is informed, through the BUFFER operand of the DFHTCT TYPE=TERMINAL macro, of the maximum data length permitted for a single outbound transmission to the logical unit. The maximum value that can be specified in the BUFFER operand for a full function LU is 256. If a CICS/VS application program sends a message longer than the maximum length specified, CICS/VS automatically divides the message and sends it as a chain of RUs. The 3770 program should make chained output appear to the terminal operator as a series of writes from CICS/VS, with no provision to interrupt receipt of the output.

To provide full support for data chaining, the 3770 program should be prepared to recognize and process an SNA CANCEL command. CICS/VS sends a CANCEL command whenever the 3770 program returns a negative response to any element of a data chain except the final element. Even though a negative response is sent for an element of a chain, CICS/VS may have already sent the rest of the chain; therefore, the 3770 program must be prepared to delete the rest of the chain. The 3770 program may also have to inform the terminal operator that the message is to be ignored.

BRACKET PROTOCOL

Within CICS/VS, a transaction represents a unit of work that must be completely processed before another unit can be processed. This transaction concept must be extended to the 3770 program so that each works on the same transaction simultaneously.

A transaction is delimited by identifying uniquely the first and last messages of the transaction. The SNA bracket protocol provides the means of achieving this. The begin-bracket (BB) indicator is sent with the first message and the end-bracket (EB) indicator is sent with the last message of a transaction. Within brackets, that is, once the BB indicator has been sent and acknowledged, any number of messages may be sent but they will not carry either the BB or EB indicators. Bracket protocol also defines which session partner may terminate a bracket and which session partner must ask permission to begin a bracket. A bracket (and therefore a transaction) may be begun only when the session partners are between brackets, that is, when either no message has yet been sent by a session partner or the last message sent by a session partner carried the EB indicator.

The use of bracket protocol resolves the potential contention problem that arises when both session partners attempt to begin a transaction simultaneously.

When communicating with a full function logical unit, CICS/VS requires bracket protocols to be observed. The system programmer must define the logical unit with bracket protocol enforcement requested (BRACKET=YES in DFHTCT TYPE=TERMINAL macro). CICS/VS always ends a bracket, the 3770 program may never do so. CICS/VS always requests a definite response when it sends the EB indicator, regardless of the PCT options selected. If CICS/VS sends the EB indicator with a multi-element chain, the EB indicator will appear with the first RU of the chain and the RQD indicator will be sent with the last element of the chain. A 3770 program may begin a bracket at any time unless CICS/VS has already requested and received permission to begin a bracket. This latter sequence is described later under "Automatic Transaction Initiation".

Bracket Communication Techniques

The responsibility for implementing bracket communication for the logical unit falls on the 3770 program. The 3770 maintains a record of the bracket state and sets indicators that the 3770 program can inspect. Alternatively, the 3770 program can keep its own record of the current bracket state of the session. This record enables the 3770 program to provide necessary bracket protocol and determine when a new transaction may begin.

The 3770 program may maintain a table of valid transaction codes for CICS/VS. This table should also contain pertinent information such as response protocol required for input with each transaction. Whenever a session is between brackets, the 3770 program should scan this table for each input from a terminal. Once a transaction is initiated, the table need not be scanned until an EB indicator is received from CICS/VS.

As an added function, local transactions that are handled entirely within the 3770 may be supported by the 3770 program. An example of a local transaction is an inquiry from a terminal operator to determine the processing status of the 3770 program. Such an inquiry should be supported at all times, even if the logical unit is within brackets. To implement such a function, the 3770 program should keep a table of valid local transaction codes. This table should be scanned for every keyboard input to the logical unit regardless of the bracket state.

CONTROL OF SNA INDICATORS BY A CICS/VS APPLICATION PROGRAM

Normally CICS/VS takes full responsibility for setting the required SNA indicators on RUs sent to a 3770. However, there are two methods that a CICS/VS application program may use to reduce the number of RUs sent.

Use of the CONVERSE Command

Figure 2-2 illustrates a CICS/VS application program sequence of a SEND command followed by RECEIVE commands. The receive request causes CICS/VS to send a null RU with the CD indicator. If a CONVERSE command is used to replace both the send and the receive requests, CICS/VS sets the CD indicator on the last RU of the chain sent to the 3770 program; thereby avoiding the sending of a null RU.

Use of the LAST Option

Figure 2-2 also illustrates the sequence of a final write request being followed by a null RU to send the EB indicator. By including the LAST option in the write request, the application programmer would cause CICS/VS to send the EB indicator with the first RU of the message and the RQD indicator with the last RU; thereby avoiding the sending of a null RU.

MESSAGE FORMATS

The format of messages is a matter of convention established between the 3770 program and the CICS/VS application program. CICS/VS imposes no restrictions. The data stream can be constructed in any way that ensures the stream is convenient for the programs to process.

A function management header (FMH) is not required for input messages to CICS/VS. However, if an FMH is supplied, it can either be passed unchanged to the CICS/VS application program or ignored by CICS/VS and not passed on; the INBFMH operand of the DFHPCT TYPE=ENTRY macro is used to determine how an inbound FMH will be handled. If FMHs are to be passed to a CICS/VS application, the application must contain a HANDLE CONDITION INBFMH command. If required by the application, the CICS/VS application program may include an FMH in a message to be sent to the 3770 program. However, CICS/VS will not inspect the FMH and will simply transmit it as part of the data.

If BMS is used to format messages to the 3770 program, it will use the SCS data stream, in which each line is delimited by a new line (NL) control character. This BMS support is thus particularly suitable for output to printers that accept SCS data streams. This BMS support will not generate an FMH on output, nor accept one on input.

AUTOMATIC TRANSACTION INITIATION

Automatic transaction initiation (ATI) may occur, for example, as a result of a message being routed to the logical unit. Since, in most cases, the message will be unexpected by the 3770 program, the 3770 program can only pass the message on to the terminal under the direction of the SNA indicators set by CICS/VS.

To participate in ATI, the 3770 program must be prepared to recognize receipt of a SNA BID command and to grant permission for CICS/VS to proceed with ATI. The BID command is a CICS/VS request to allow transaction initiation. If permission to proceed is not granted by the 3770 program, ATI is suspended by CICS/VS.

The 3770 program can grant permission to proceed with ATI by:

- Responding to the bid with a positive response
- Responding to the bid with an negative response and then transmitting an SNA ready-to-receive (RTR) command at a later time

When permission is granted using the RTR command, CICS/VS normally returns a positive response and proceeds with ATI. The first output from the transaction is sent with BB indicator. However, CICS/VS returns an negative response if the condition causing the ATI no longer exists when the RTR command is received. For example, the request may have been canceled by the master terminal operator.

When ATI is requested for a logical unit, CICS/VS waits (if necessary) until there is no active transaction associated with the logical unit, that is, the logical unit is in the between-brackets state. CICS/VS then requests permission to proceed by sending the BID command. The responsibility of determining if the ATI should be allowed rests with the 3770 program. To make the 3770 program ATI support independent of the particular transaction or series of transactions that may be currently in progress, the support should be implemented in a generalized manner. One method is to allow the terminal operator to

indicate when to accept an ATI. This method could be implemented in the following manner:

- Always respond to the bid with the negative response (code '0814') that indicates that RTR will be sent later.
- Inform the terminal operator of the ATI request by displaying an appropriate message.
- When it is convenient to accept the ATI, the terminal operator informs the 3770 program by entering a code defined as 'accept ATI'.
- Issue the RTR command to CICS/VS.

For output-only stations, a bid may be accepted as soon as it is received, provided the required resources are available.

Figure 2-3 illustrates how a transaction is initiated as a result of ATI. The example assumes that the 3770 program has previously established a session with CICS/VS but is not currently engaged in a transaction. CICS/VS commences by sending a BID command to the 3770 program to ensure that it is prepared to accept unsolicited data from CICS/VS. When the 3770 program detects receipt of the bid, it will be in one of the following states:

1. Able to accept the message.
2. Unable to accept the message because the 3770 program has just sent, or is about to send, a message to CICS/VS to initiate a transaction.
3. Unable to accept the message because either the 3770 program is engaged in a dialog with the terminal operator and wishes to complete processing, or the 3770 program needs to acquire an output printer before receiving the first message to be output.

The flow for state 1 is shown in Figure 2-3. Upon receiving a definite response to the BID command, CICS/VS will pass control to the transaction associated with the unsolicited data to permit it to commence output. The 3770 program must commence reading from CICS/VS and continue to do so until an RU having a CD indicator or a chain with an EB indicator is received. The first chain received from CICS/VS will have the BB indicator.

Figure 2-4 illustrates the flow for state 2. The 3770 program must write a negative response to CICS/VS rejecting the BID command, and then continue as in Figure 2-2. The 3770 program has a choice of two negative responses. With a response including system sense code X'0813', CICS/VS will resend the BID command once the current transaction has completed. With a response including system sense code X'0814', CICS/VS will await a ready-to-receive (RTR) command from the 3770 program. The 3770 program must ensure that the session is in between-bracket state before sending the RTR command.

For state 3, the 3770 program will immediately send a response including sense code X'0814' to reject the BID command. It may then either inform the operator that an unsolicited request is pending or attempt to obtain an output printer as appropriate. When the 3770 program is informed that the operator is willing to accept the output or a printer is available, it must send an RTR command to CICS/VS. CICS/VS can then pass control to the associated transaction to commence output. The first chain from CICS/VS will have the BB indicator set. The BID command will not be retransmitted. If, in the intervening period, the ATI has been canceled, CICS/VS will send a negative response indicating

that the 3770 program should inform the operator that the request has been canceled, or that the printer may be released.

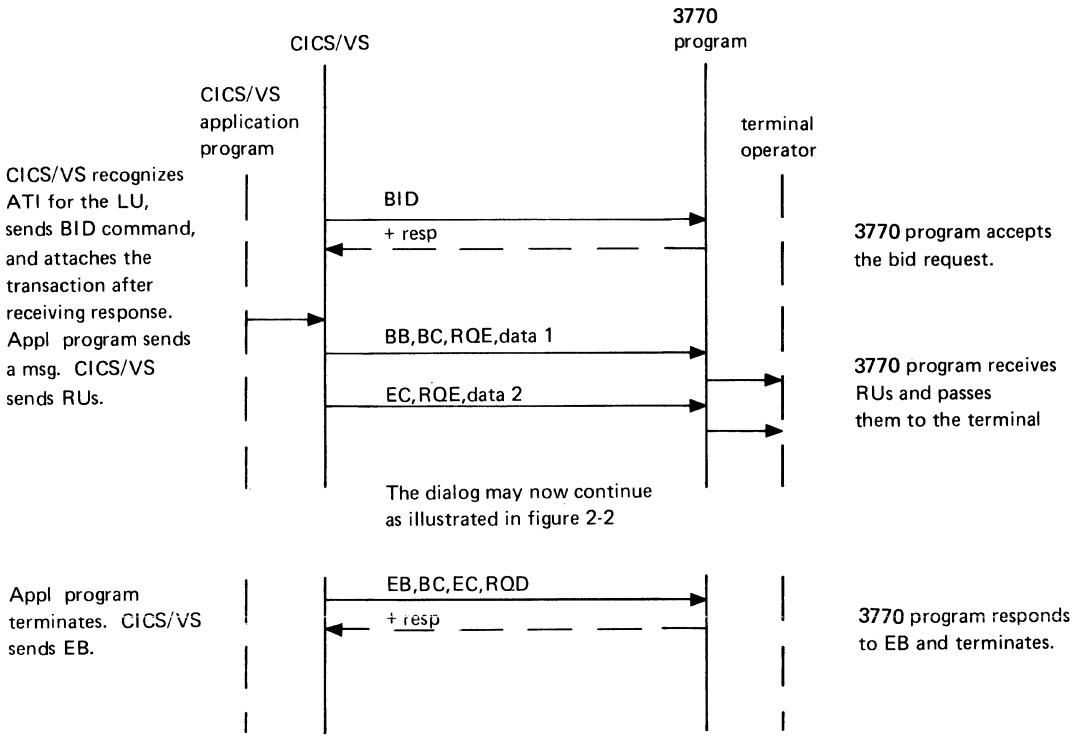


Figure 2-3. Acceptance of an Automatic Transaction Initiation

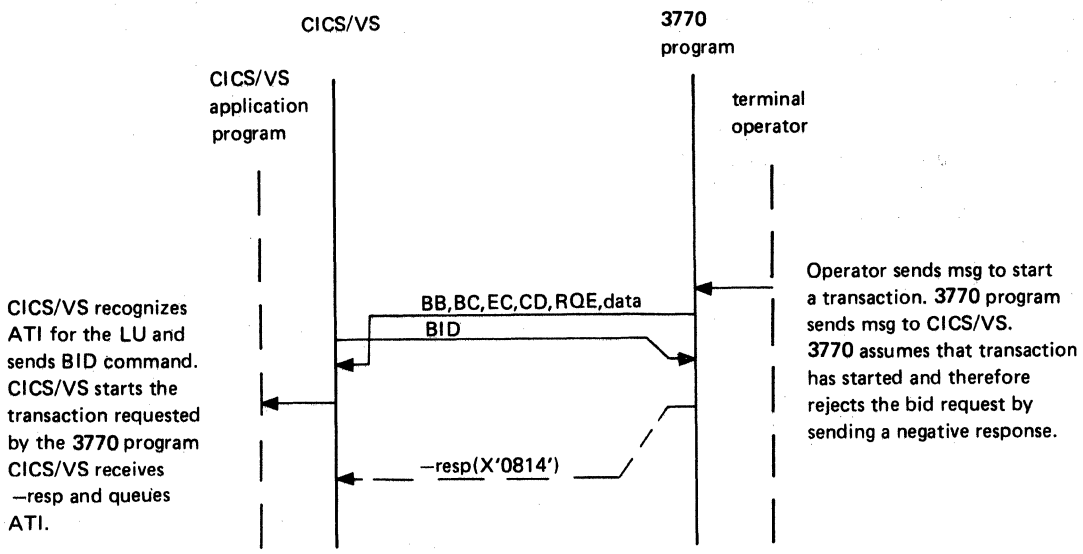


Figure 2–4. Rejection of an Automatic Transaction Initiation

ESTABLISHING A SESSION

Before any data can be transmitted between a logical unit and CICS/VS, a session must be established. The access method allows only session control information to pass between a logical unit and CICS/VS before a session is established. Data, data flow control information, or session control information is allowed after a session is established.

A session may be initiated by a 3770 program issuing the 3770 OPNSESS statement which causes the 3770 to send the SNA INITIATE–SELF command.

To the 3770 program, a session appears either as a 3770–initiated session or as a CICS/VS–initiated session. A session initiated by the access method network operator, the master–terminal operator, or by CICS/VS during CICS/VS initialization appears as a CICS/VS–initiated session to a 3770 program.

CICS/VS-Initiated Sessions for Automatic Transaction Initiation

The unsolicited message sequence illustrated in Figure 2-3 assumes that the 3770 program is active and in session with CICS/VS at the time the message arrives. However, since a session may not be in progress, CICS/VS must determine if a session exists and initiate one if necessary.

Figure 2-5 illustrates the process. On recognizing an ATI request for a logical unit for which no session exists, CICS/VS requests the access method to initiate a session using the OPNDST macro. This causes the SNA BIND command to be sent to the 3770. The BIND command includes command data that dictates the SNA protocols CICS/VS expects a 3770 program to observe when communicating with CICS/VS. When the 3770 receives the BIND command, it determines which 3770 program should be activated. The 3770 passes control to the selected 3770 program and makes the bind parameters available to it in a 3770 buffer. The 3770 program may inspect the bind parameters and continue or reject the session depending on whether or not it is prepared to communicate with CICS/VS.

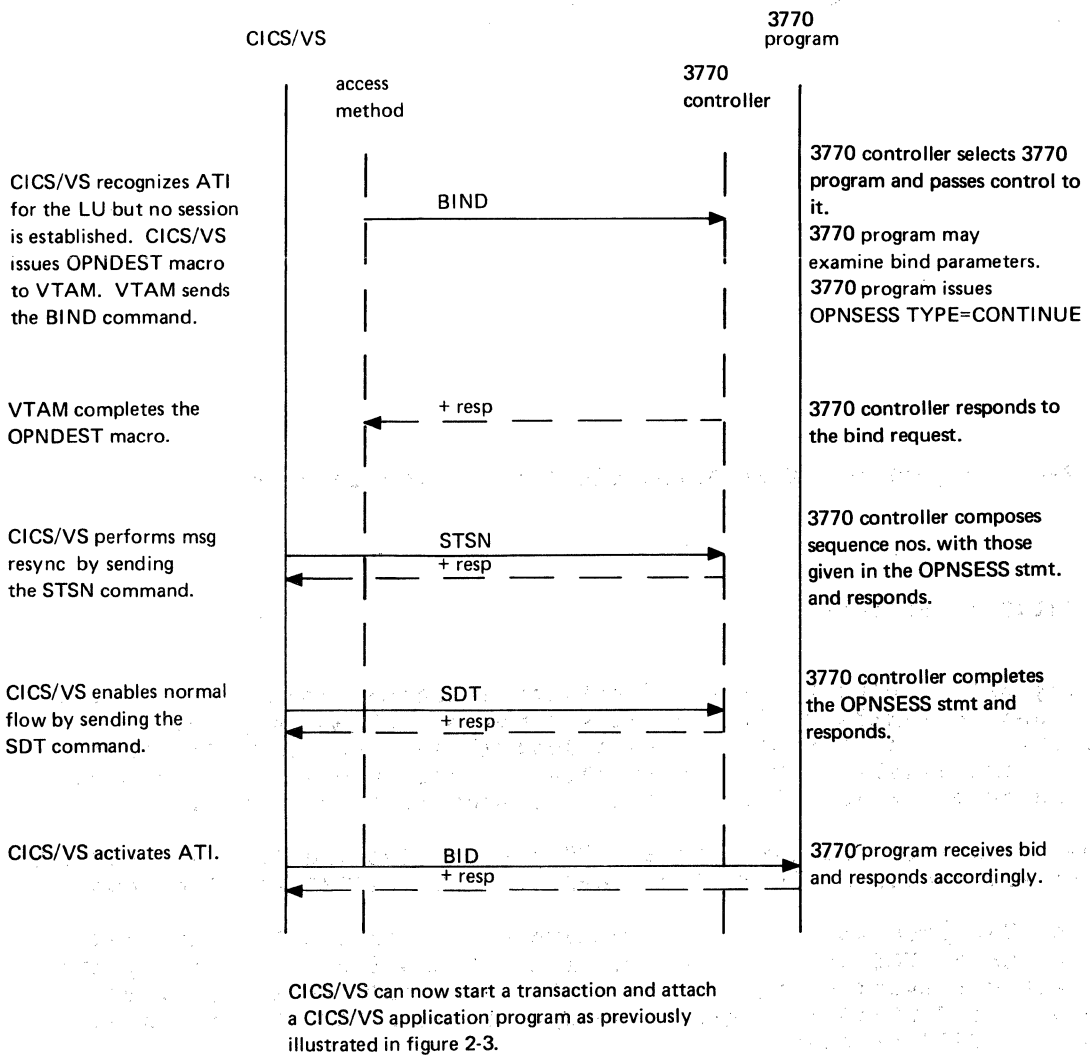


Figure 2-5. CICS/VS-Initiated Session for Automatic Transaction Initiation

If the 3770 program rejects the session (using OPNSESS TYPE=REJECT) the 3770 sends a negative response to CICS/VS. CICS/VS marks the logical unit as being out-of-service and queues the ATI for a subsequent attempt following CICS/VS master terminal intervention.

Normally the 3770 program will continue the session (using OPNSESS TYPE=CONTINUE) in which case the 3770 will respond positively to CICS/VS. CICS/VS then enters its message resynchronization sequence (if message resynchronization is required) by sending the STSN command to the 3770 so that the last known message sequence numbers can be compared. If the 3770 reports no mismatch, CICS/VS enables the session by sending the SNA Start-Data-Traffic (SDT) command. This causes the 3770 to complete the OPNSESS statement with suitable return codes to the 3770 program indicating the outcome of the resynchronization activity. The 3770 also responds to the SDT command.

Figure 2-5 assumes that no message retransmission is required and shows CICS/VS proceeding with ATI as previously shown in Figure 2-3.

Alternatively, the 3770 programmer can arrange for a 3770 program, already activated by the 3770, to be prepared to accept a CICS/VS-initiated session. Such a 3770 program is activated for the particular logical unit and commences by issuing a 3770 OPNSESS TYPE=ACCEPT statement. When an unsolicited bind is accepted by the 3770 as described above, the bind parameters are passed to the waiting 3770 program which can then elect to continue or reject the session.

Other CICS/VS-Initiated Sessions

CICS/VS may establish a session as a result of CICS/VS system generation options during CICS/VS initialization, as a result of CICS/VS emergency restart, or as a result of a request by either the CICS/VS master terminal operator or the access method network operator. The session initiation sequence shown in Figure 2-5, beginning with the BIND command and ending with the SDT command, is the same regardless of the reason for starting the session. If the STSN command indicates that message retransmission for a committed message is required because the previous session terminated abnormally, the message will be sent immediately after the SDT command. It will contain both BB and EB indicators. If there is no ATI request pending, BID will not be sent.

3770 Program-Initiated Sessions

When a 3770 program is activated and requires communication with a CICS/VS application program, it issues a 3770 OPNSESS TYPE=ACQUIRE statement. The statement causes the 3770 to request CICS/VS to initiate a session. CICS/VS responds by sending the BIND command as previously described. Again, the 3770 program may inspect the bind parameters and elect to continue or reject the session. Note that the example given in Figure 2-2 assumed that the session was already established.

If the 3770 program continues the session, the resynchronization procedure is followed and, provided message retransmission by CICS/VS is not required, the application proceeds as in Figure 2-2.

MESSAGE RESYNCHRONIZATION

During a session, messages transmitted between CICS/VS and a logical unit are numbered. Outbound messages sent by CICS/VS are consecutively numbered, beginning with one. Inbound messages sent by the logical unit are also consecutively numbered, beginning with one. These sequence numbers provide:

- A means for the message receiver to ensure that messages are not lost during transmission.
- A means for the message sender to correlate a response with the original message.
- A means of identifying a message restart point (last recoverable message) for cases in which a message is lost.

The process of reestablishing sequence number synchronization between CICS/VS and a logical unit is called message resynchronization.

Message resynchronization may be initiated because the session failed, the 3770 program issued an SNA request-recovery (RQR) command, or a host failure occurred causing a CICS/VS emergency restart. Once initiated, message resynchronization must complete successfully before CICS/VS permits normal data transmission to resume.

The CICS/VS message protection feature provides for the resynchronization of the message flow between CICS/VS and a full function logical unit. The feature allows a session to continue from the last completed logical-unit-of-work (LUW) prior to resynchronization. A LUW is a transaction or part of a transaction that is delimited by synchronization points. Automatic re-presentation of a committed message is performed when it is necessary to complete message flow up to the last synchronization point. A discussion of the concepts of the LUW and committed output is given in the CICS/VS System/Application Design Guide.

CICS/VS performs resynchronization during emergency restart, or when a previous session was abnormally terminated, or when requested to do so by a 3770 program. A 3770 program requests resynchronization using the WRTCTL RESYNC=REQ statement. In any of these cases CICS/VS sends the SNA Set-and-Test-Sequence-Numbers (STSN) command while the normal message flow for the session is suspended.

The STSN command carries the sequence numbers of the last input and output message for the completed LUW for which CICS/VS is directing resynchronization. The STSN command is processed by the 3770 without the direct involvement of the 3770 program. However, the 3770 program must supply the sequence numbers that the 3770 uses to compare with the CICS/VS sequence numbers in order to correctly respond to the STSN command. The recording of message sequence numbers for this purpose by the 3770 program is discussed below. The numbers are supplied to the 3770 when the 3770 program specifies the OPNSESS statement with the RESYNC=YES operand for warm start session initiation or the WRTCTL statement with the RESYNC=YES operand when the 3770 program requests resynchronization. In either case, if the 3770 detects a mismatch between CICS/VS and 3770 program sequence numbers, the statement completes with error number 104.

The previous figure, Figure 2-5, showed how message resynchronization takes place at the start of a session. Figure 2-6 shows an example of the flow when resynchronization is performed at the request of the 3770 program. In this example, the resynchronization is requested just as a committed message is being sent. The message is deleted.

Resynchronization detects the loss of the message and automatically retransmits it.

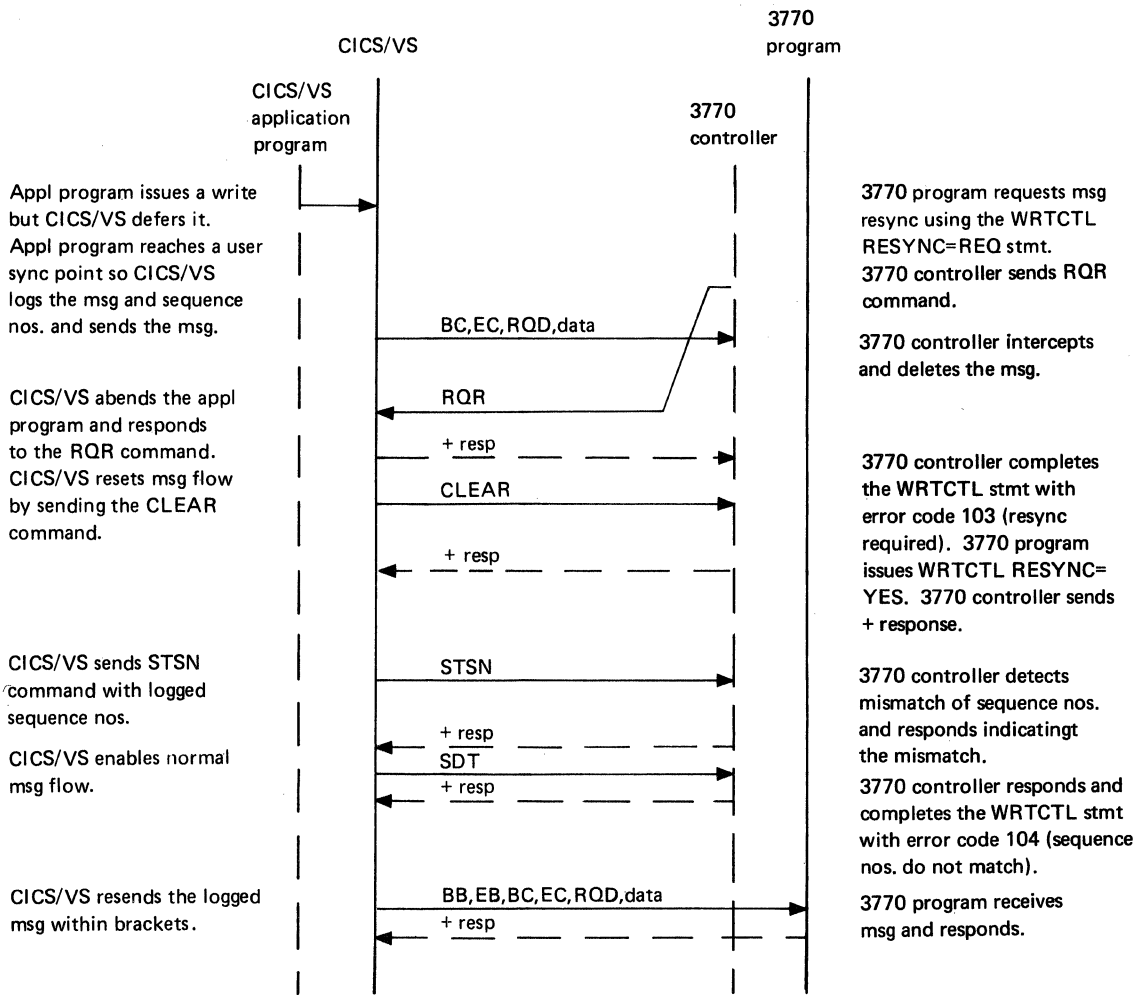
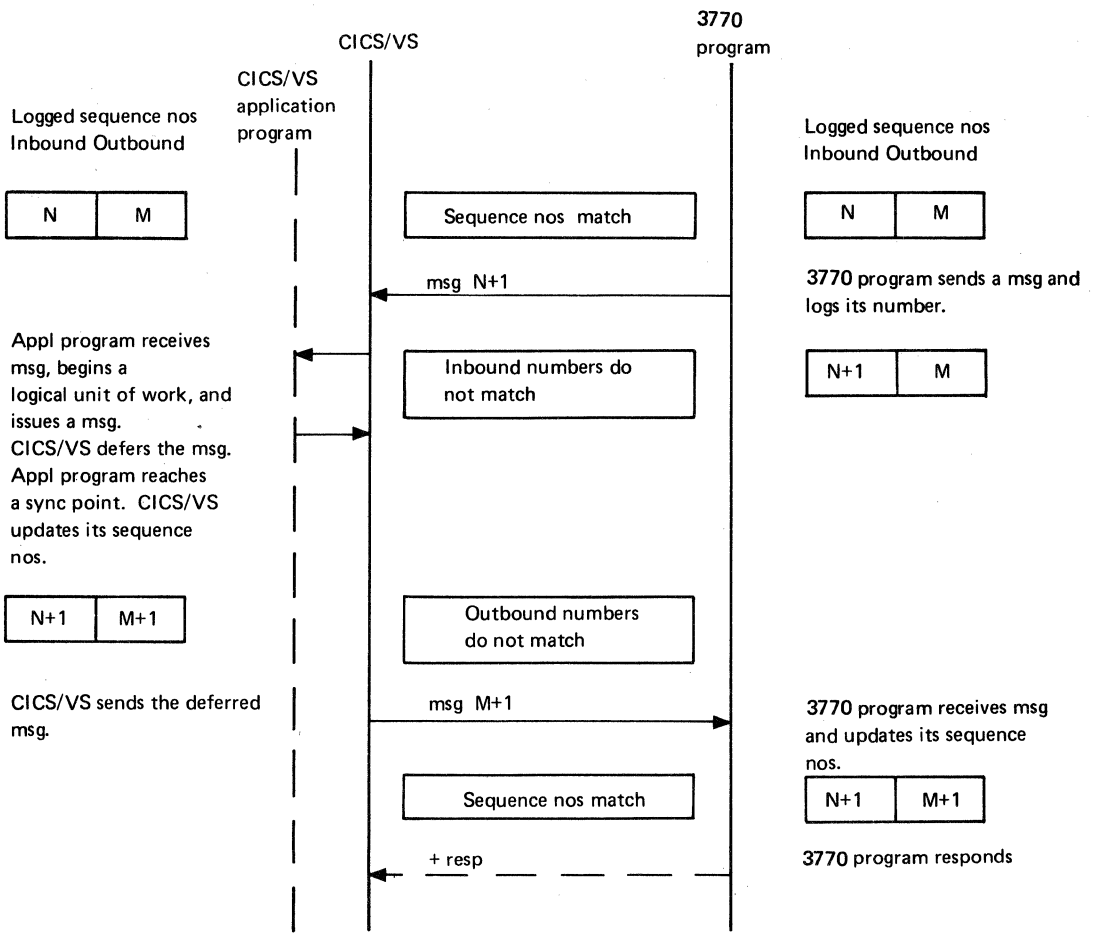


Figure 2-6. Message Resynchronization Initiated by a 3770 Program

Sequence Number Management

In order to supply sequence numbers to the 3770, the 3770 program must retain numbers that are used during normal message flow. The numbers are obtained by issuing the RDSTUS statement after each RDDATA and WRTDATA statement. The 3770 program would normally log the numbers to a file so as to preserve them during a lengthy session failure. CICS/VS logs sequence numbers for any transaction that specifies message protection. Normally the CICS/VS application program should be designed so that each LUW is composed of an input message, processing of the message, for example, a file update, followed by a committed output message. In such a case, sequence numbers logged by CICS/VS and the 3770 program will match except during the actual transmission of a message (there is a delay between the sender of a message logging the sequence number sent while the message flows through the network, before the receiver is able to log his copy of the sequence number). Because CICS/VS uses half-duplex flip-flop protocol with the full function logical unit, only one message will be in transmission at any instant. Therefore, provided the CICS/VS application program is composed of LUWs in the manner described, the sequence numbers can mismatch only by a single message on either inbound or outbound flow. Figure 2-6 illustrates this potential problem.



Notes: 1. The 3770 program must log the sequence number of a received message before it responds to CICS/VS. When CICS/VS receives the response, the deferred message is no longer eligible for retransmission.

2. This figure applies only to single-element chains. If a message is sent as a multi-element chain, CICS/VS and the 3770 program should log only the sequence numbers of the last RUs of messages. This is acheived automatically for CICS/VS by specifying the chain assembly option.

Figure 2–7. Sequence Number Management

If a session fails and resynchronization is performed, the 3770 has to handle the following cases:

1. Sequence numbers match.
2. Inbound sequence numbers do not match, but outbound sequence numbers do match.
3. Inbound sequence numbers match, but outbound sequence numbers do not match.

For case 1, no action is taken by the 3770 program. Cases 2 and 3 are indicated to the 3770 program by error number 104. The 3770 program must issue RDSTUS statement to obtain the sequence numbers supplied by CICS/VS and compare them with the numbers it logged in order to distinguish between the two cases. Case 2 occurs when the session fails during an incomplete LUW and the effect of the LUW, including the inbound message number was "backed out" by CICS/VS. The 3770 program may choose to retransmit the message (if possible), or perform other processing according to the application design. Case 3 occurs when the session is interrupted after completion of the LUW, but before the committed output message could be acknowledged by a definite response from the 3770 program. CICS/VS automatically re-sends the message and therefore the 3770 program must be prepared to receive and process the message.

If a CICS/VS application program does not restrict message transmission to one input message followed by one output message for each LUW, or if message protection is not specified for the CICS/VS application program, CICS/VS will not be logging sequence numbers of all messages. However, if the 3770 program is logging all sequence numbers, there will be no correspondence between the CICS/VS and 3770 program numbers. The 3770 program would therefore need to be designed in such a way that the effect of those messages for which CICS/VS is not logging sequence numbers can be either ignored or "backed out" in the event of a session failure.

Refer to the CICS/VS System/Application Design Guide for suggested transaction restart techniques.

HANDLING ERRORS

When communicating with CICS/VS, the 3770 program must be able to handle error conditions as they occur and respond appropriately to CICS/VS and the terminal operator. Methods used to convey error information between CICS/VS and the 3770 program and techniques for handling such error information are discussed in this section.

Receiving Sense Information From CICS/VS

When CICS/VS detects an error during receipt of a message from the logical unit, it sends a negative response to the chain containing the error followed by a null RU with the EB indicator, thereby ending the transaction. In severe cases, the CLEAR and UNBIND commands are also sent to terminate the session.

Some errors leave the logical unit in the between-bracket state, for example, response X'0819' indicating that an RTR command sent by the 3770 program is being rejected. Both CICS/VS and the 3770 program are free to continue the session. Some errors occur asynchronously, for example, a transaction abend, when there is no outstanding response to a chain sent by the 3770 program. In this case, CICS/VS uses the SNA logical-unit-status (LUSTAT) command to transmit the error code. A 3770 program should therefore also be prepared to receive the LUSTAT command.

If the 3770 program receives a negative response or the LUSTAT command, unless already in between-bracket state it should continue to receive from CICS/VS until the EB indicator is received indicating that CICS/VS has ended the transaction. Depending on the error sense code, the 3770 program may begin a new transaction or reattempt the same

transaction, possibly after allowing a decision to made by the 3770 terminal operator. However, the 3770 program must be prepared to receive the CLEAR and UNBIND commands at any time. This sequence may be imposed by the access method rather than by CICS/VS.

Figures 2-8 and 2-9 show CICS/VS negative responses to a message attempting to initiate a transaction. Similar sequences will occur even if the inbound message is not the first of a transaction. Figure 2-10 shows an asynchronous error condition occurring when there is no outstanding response that CICS/VS can use to convey the error condition to the 3770 program. If, in that example, the write had resulted in a multiple-element chain under the control of the 3770 program, it would had been terminated by an SNA CANCEL command if the application had abended before sending the last element.

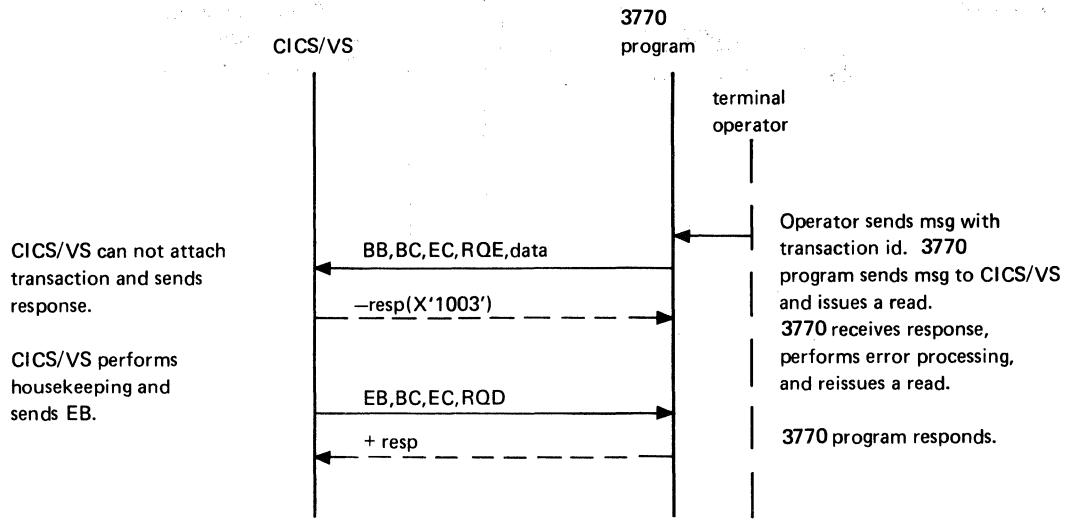


Figure 2-8. Error Response to a Single-Element Chain by CICS/VS

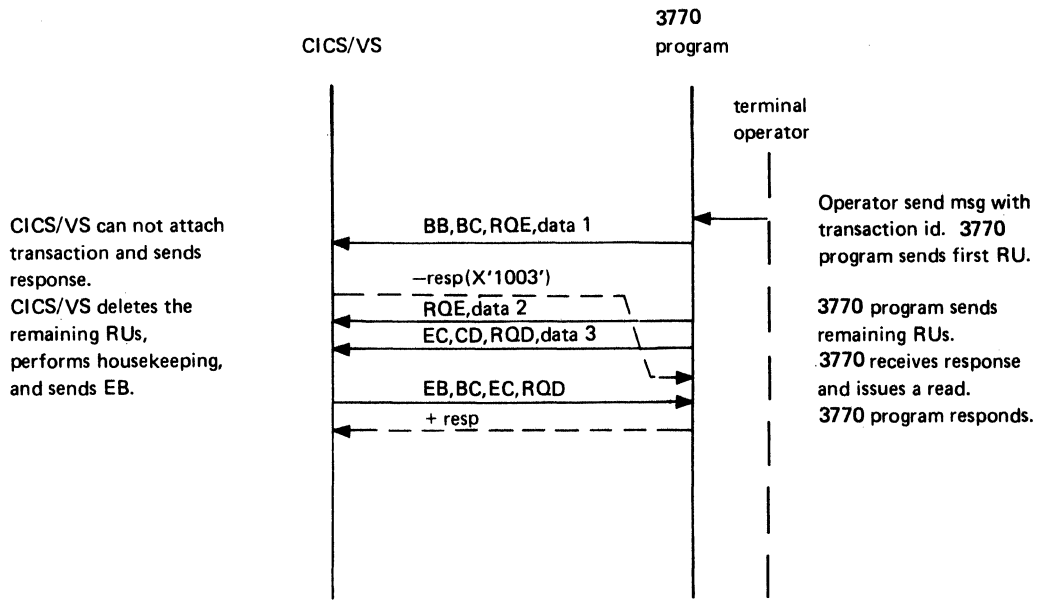


Figure 2–9. Error Response to a Multi–Element Chain by CICS/VS

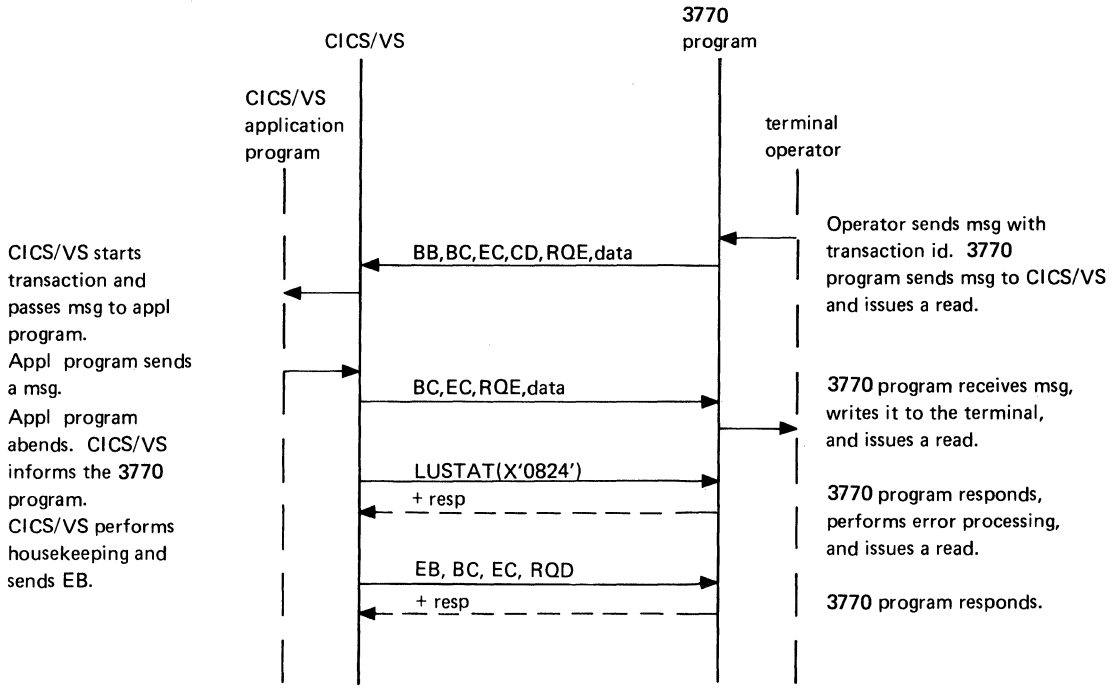


Figure 2-10. Error Reporting by CICS/VS using the LUSTAT Command

Format of an Error Response

An error response is made up of two bytes of system sense information and two bytes of user sense information. Under some circumstances, CICS/VS sends a CICS/VS error message number (converted to binary) in the user sense bytes.

In particular, when a transaction abend occurs, the user sense bytes will indicate that, for a transaction eligible for dynamic transaction backout, all data base changes made for the LUW have been backed out. The 3770 programmer must be aware of the relationship of input messages to the CICS/VS synchronization point technique being employed, so that the condition may be correctly relayed to the operator responsible for understanding which input messages have been backed out.

Upon receiving a negative response from CICS/VS the 3770 program may relay the condition to the operator using a message similar to the following:

INPUT REJECTED — SYSTEM SENSE ssss REASON DFHuuuu

where ssss is the system sense and uuuu is the user sense. The operator may then look up the DFHuuuu error message (provided uuuu is not zero) in the CICS/VS Messages and Codes. A list of system sense codes that CICS/VS may send to a 3770 program is given in Appendix C.

Sending Sense Information to CICS/VS

The 3770 program may send negative responses to CICS/VS with two bytes of system sense information and, optionally, two bytes of user sense information. CICS/VS error recovery routines examine only the system sense bytes. The permissible system sense codes and the error recovery actions that are taken are described in the CICS/VS System Programmer's Reference Manual. Any of these system sense codes can be reported as appropriate to CICS/VS using the LUSTAT command if detected while the 3770 program is in send mode. User conditions to be reported to a user-written node error program (NEP) should be sent to CICS/VS using the LUSTAT command with a system sense code X'0000' and user sense codes defined by the user. CICS/VS treats this form of the LUSTAT command in a manner similar to a negative response with zero system sense.

With the exception of the system sense codes X'0813' and X'0814', which are discussed later in this section, CICS/VS and an NEP have responsibility for error recovery. Therefore, if a 3770 program sends a negative response to a CICS/VS chain, it must delete the remainder of the chain (if any) and issue a read request to CICS/VS. Upon receiving a negative response from a 3770 program, CICS/VS analyzes the system sense bytes and sets default error action flags (also described in the CICS/VS System Programmer's Reference Manual). If an NEP exists, it may reset the error action flags (except for "disconnect session"). Note that the default action taken by CICS/VS will prove adequate. Only when the processing required is dependent on the CICS/VS application program and the content of the rejected message should a NEP reset error action flags.

In most cases, the default action is to cancel the outstanding request and abend the transaction. CICS/VS will inform the 3770 program by sending null RU with an EB indicator. In certain cases, session termination is also carried out and therefore the CLEAR and UNBIND commands will also be sent. If an NEP is coded to override a transaction abend, the 3770 program must be aware of the system sense codes for which this is being done and be prepared to receive further data and return to normal data flow.

Where the default action does not require a transaction abend, the 3770 program should accept a return to normal data flow. If such an error occurs on a read request, or on a write request for a chain that requested definite response protocol, CICS/VS automatically re-issues the request. However, if a write request has requested exception response protocol, CICS/VS can not re-issue the message because the data will no longer be available. The CICS/VS application program must therefore re-create the message and reissue the write, or abend the transaction. To do this the NEP and the CICS/VS application program must have set up a communication technique using the transaction work area (TWA) in which the CICS/VS application program can check codes that are set by the NEP.

Bid Reject, System Sense Codes X'0813' and X'0814'

In these cases CICS/VS simply re-queues the pending ATI request to be reattempted at a later time. See discussion on ATI earlier in this chapter.

Intervention Required, System Sense Code X'0802'

The default action taken by CICS/VS for this response is to recommend a retry, but to delay the retry in the expectation that the 3770 program will send a LUSTAT command with system sense X'0001' and user sense X'0000' to indicate that the condition has been cleared. When CICS/VS receives the command with these codes, CICS/VS reattempts the request. If the condition giving rise to the original negative response cannot be cleared, the 3770 program should send a LUSTAT command with system sense X'081C' and user sense X'0000' indicating a permanent error. The CD indicator should be sent with this command to return control of the session to CICS/VS.

When a 3770 program elects to send a negative response indicating intervention required, it should initiate whatever action is required to clear or further analyze the condition and then send an LUSTAT command to report on the outcome. After sending the LUSTAT command the 3770 program should either issue a read to either resume processing or receive the EB indicator if the error was reported to be permanent (as shown in Figure 2-11).

If the 3770 program is sending data and an intervention required condition occurs, the 3770 program may use the LUSTAT command with system sense code X'0802' and user sense code X'0000'. The CICS/VS recovery action is as described above, except that if the condition is reported as cleared, the CICS/VS read will be attempted so the 3770 program must continue to send rather than go to receive mode.

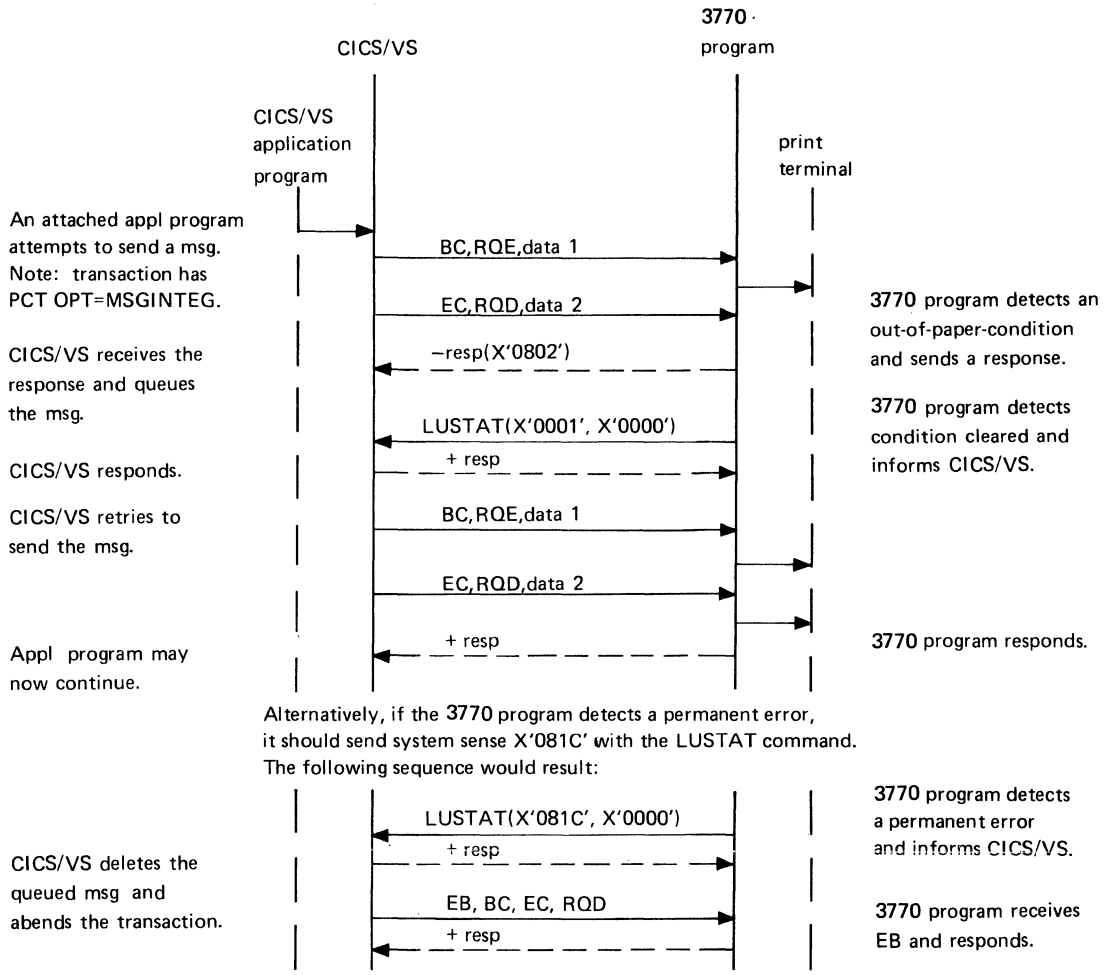


Figure 2–11. Error Reporting by a 3770 Program using the LUSTAT Command

SUSPENDING TRANSMISSION BETWEEN CICS/VS AND THE LOGICAL UNIT

Whenever CICS/VS receives a request from the master–terminal operator to place a logical unit out of service or in receive–only mode, CICS/VS sends the SNA quiesce–at–end–of–chain (QEC) command to the logical unit. This command signifies that CICS/VS, temporarily, will not accept any further data from the logical unit. The 3770 program should send a response to the command, complete any current output, send the SNA quiesce–complete (QC) command to CICS/VS, and prepare to receive from CICS/VS. When CICS/VS is enabled to resume normal communication with the logical unit, it will send the SNA release–quiesce (RELQ) command. Note that a 3770 program may continue to receive data from CICS/VS during the quiesced period.

INTERRUPTING TRANSMISSION BETWEEN CICS/VS AND THE LOGICAL UNIT

The normal data flow between CICS/VS and the logical unit can be interrupted by either session partner by issuing the SNA SIGNAL command. The SIGNAL command provides a means for one session partner to indicate to the other, who is currently in send mode, that the receiving partner wishes to send data. For example, the 3770 program may receive a request from the terminal operator that output that is being sent from CICS/VS is not required and should be discontinued.

The SIGNAL command may be sent at any time by the 3770 program, but is sent only under certain circumstances by CICS/VS.

The CICS/VS application program can detect the inbound SIGNAL by use of a HANDLE CONDITION command for the SIGNAL condition. The HANDLE, which must have been executed before SIGNAL arrives, specifies the label of a routine that is to be given control when the SIGNAL condition is raised. Note that the branch is not necessarily taken immediately on receipt of SIGNAL; unless a WAIT SIGNAL command has been issued, the raising of the condition is deferred until the next SEND, CONVERSE, or RECEIVE command.

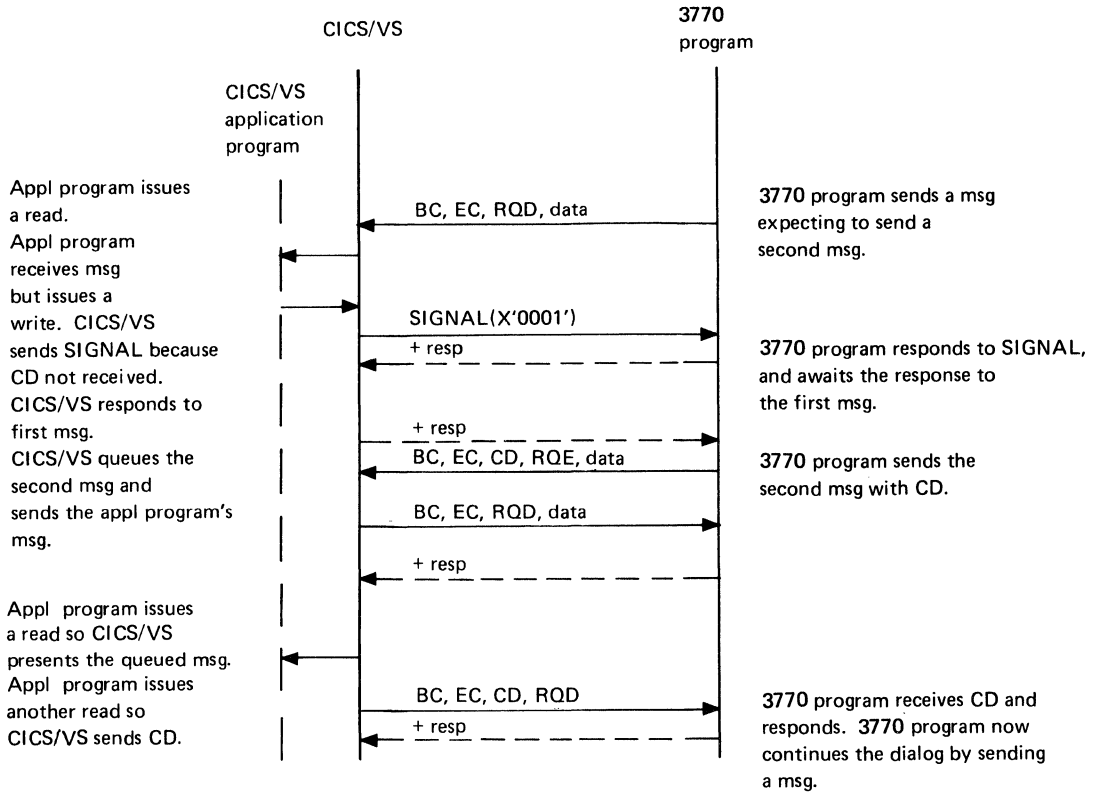
In the absence of a HANDLE for the SIGNAL condition, the condition is ignored.

Use of SIGNAL should be careful planned. For example, the 3770 program may never send the SIGNAL command and therefore,if the CICS/VS application program is waiting for the command, the transaction will be locked. If the SIGNAL command is used it should carry the system sense code X'0001' and user sense code of X'0000', which indicates a request for change direction. The access method responds positively on behalf of CICS/VS. The CICS/VS application program should stop writing and commence receiving as soon as convenient after detecting a signal. The 3770 program should only send a signal if it is reading and requires to write urgently to the CICS/VS application program. The 3770 program must not attempt to write until it receives a chain including the CD indicator.

CICS/VS uses the SIGNAL command when the CICS/VS application program issues a write request while the 3770 program is still in send mode, that is, has not sent a chain including the CD indicator. In all sequences so far illustrated, every chain sent by the 3770 program has included the CD indicator. Thus, whenever the CICS/VS chooses to write, it may do so because the 3770 program will have reverted to receive mode. However, the 3770 program may be constructed such that is sends a series of messages to the CICS/VS application program before issuing a read and therefore the CICS/VS application program must be aware that it should not send a message until the series is complete. A convention could be established between the CICS/VS application program and the 3770 program based upon the content of the messages being sent as to when a read may be issued by the CICS/VS application program.

If the CICS/VS application program does issue a write before it is expected by the 3770 program, CICS/VS responds by sending a SIGNAL command to the 3770 program. The 3770 program should send a chain including the CD indicator as soon as possible and issue a read request. CICS/VS cannot honour the CICS/VS write request until it receives a CD indicator. Thus CICS/VS continues to read from the 3770 program until the CD indicator is received. Data received in this way is queued and subsequent read requests by the CICS/VS application program are satisfied from the queue, provided that the read-ahead queuing option (RAQ operand of the DFHSPCT TYPE=ENTRY macro) is used for the transaction; this practice is recommended. Careful planning is required

if this mode of operation is to be employed. Figure 2–12 shows the sequence that will cause CICS/VS to send the SIGNAL command.



Note that if the CICS/VS application program had issued a write after receiving the queued message, CICS/VS would have sent the message directly since control of the session was with CICS/VS.

Figure 2–12. Use of the SIGNAL Command by CICS/VS

Figure 2–13 shows the sequence that will occur if the 3770 program loses synchronism with the CICS/VS application program and issues a read while the CICS/VS application program is still reading. In this case, the 3770 program can either request further operator input or decide that there is sufficient lack of synchronism to warrant termination of the transaction. The transaction may be terminated by sending the LUSTAT command with the system sense code, X'0002' and user sense code X'0000' indicating to CICS/VS that it should discontinue reading and abend the transaction. The CD indicator should be included in the LUSTAT command to return control to CICS/VS.

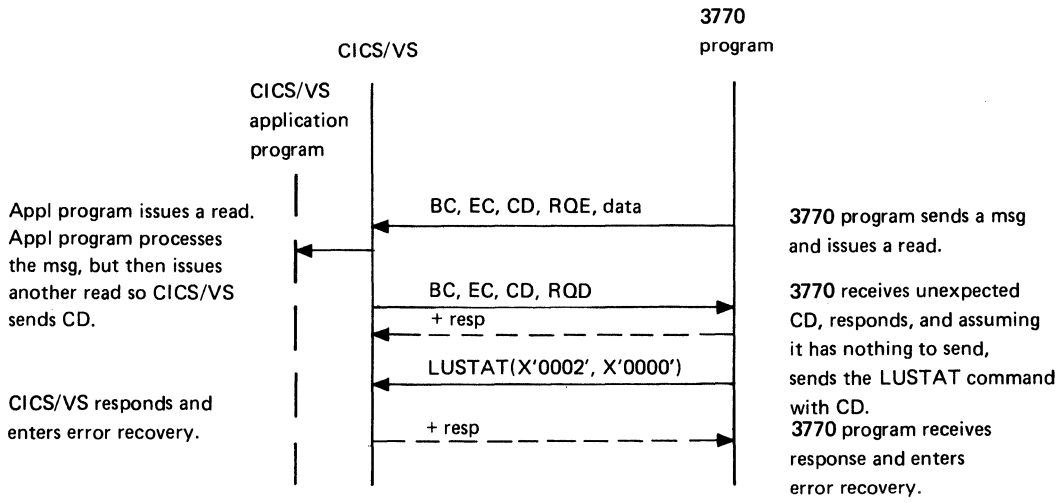


Figure 2-13. Error Response to Loss of Synchronism by a 3770 Program

TERMINATING A SESSION

A session is terminated by removing the logical connection between a logical unit in the 3770 and CICS/VS. Two types of session termination exist — orderly and immediate.

Orderly Termination

An orderly termination occurs when the logical unit is allowed to complete any transactions currently in progress before the session is terminated.

To support orderly termination, the 3770 program must be prepared to receive the SNA shutdown (SHUTD) command and to send the SNA shutdown-complete (SHUTC) and the request-shutdown (RSHUTD) commands.

If the 3770 program is to request an orderly termination, it should first ensure that any outstanding transactions and housekeeping are completed. Then the RSHUTD command can be issued to CICS/VS. CICS/VS interprets the RSHUTD command as a request for orderly termination and assumes that all processing by the logical unit has been completed.

CICS/VS issues a CLSDST macro instruction for the logical unit causing the CLEAR command followed by the UNBIND command to be sent. This sequence terminates the session.

CICS/VS initiates orderly termination by issuing the SHUTD command. The 3770 program must recognize the SHUTD command and send a definite response immediately. It must then complete any outstanding housekeeping, and issue the SHUTC command to CICS/VS. If a transaction is in progress, the 3770 program should continue until CICS/VS sends a message with the EB indicator before sending the SHUTC command. CICS/VS then terminates the session.

If the 3770 program does not provide support for orderly termination, it must be able to tolerate receipt of the CLEAR and UNBIND commands at almost any time.

Immediate Termination

Immediate termination is unconditional session termination without consideration for outstanding transactions or the processing state of the logical unit. CICS/VS flags immediate session terminations as abnormal. During any future session initiation with the same logical unit, CICS/VS automatically initiates message resynchronization.

While processing transactions, the 3770 program may encounter a condition such as a program check which precludes further transaction processing. In such cases, the 3770 program should issue the 3770 CLOSESS statement which causes the 3770 to send the TERMINATE-SELF command. This command signifies to CICS/VS that some abnormal condition occurred in the logical unit. CICS/VS terminates the session immediately. Prior warning is not given to the 3770 program nor can the program stop the termination of the session. The 3770 does not transmit data to CICS/VS on behalf of the logical unit following receipt of the CLEAR-UNBIND command sequence.

Chapter 3. The Interactive Logical Unit

Introduction

CICS/VS support for interactive logical units is designed for use with the 3767 and with the interactive logical unit of the 3770.

The support may also be used for certain devices connected via ACF/VTAM and the Network Terminal Option (NTO) of NCP/VS. These devices are:

1. TWX (model 33/35) Common Carrier Teletypewriter Exchange.

2. Teletypewriter (World Trade only)

When connected via NTO, the devices use the 3767 interactive logical unit protocols. However, the data transmitted to and from the device is not translated into a 3767 data stream, but remains a TWX/TTY data stream.

The interactive logical unit can be used in either flip-flop or contention mode. In flip-flop mode, each party sends messages in turn, the current sender being allowed to send only when the receiver invites him to do so, at which point the roles are reversed. In contention mode, conditions can arise in which each party has an equal right to send; when both attempt to do so at the same time, contention occurs, and one party, the contention loser, must give way. In sessions with the interactive logical unit, CICS/VS is always the contention loser.

Flip-flop mode is suitable for applications in which the message flow approaches a conversational exchange between the terminal operator and the CICS/VS transaction. In this mode, the terminal keyboard will lock when the operator presses the EOM key and thus despatches a message. CICS/VS will not unlock the keyboard until the transaction either issues a further receive request or terminates. For teletypewriter devices that do not have keyboard lock, keyboard rattle is employed to dissuade the operator from entering data when CICS/VS is sending.

Contention mode is suitable for use when the flow of data is predominantly one way. It is also useful for the support of teletypewriter devices, to avoid the annoyance of keyboard rattle to the operator. This mode usually requires operator discipline in the following of agreed procedures if efficient use is to be made of the session.

System Programming for the Interactive Logical Unit

GENERATING THE TERMINAL CONTROL TABLE TERMINAL ENTRY

The system programmer defines the interactive logical unit by coding the following values of TRMTYPE and SESTYPE in the DFHTCT TYPE=TERMINAL macro.

Terminal	Mode	TRMTYPE=	SESTYPE=
3767	flip-flop	INTLU or 3767I or 3767	—
	contention	3767C	—
3770	flip-flop	INTLU or 3770I	—
	contention	3770C	—
TWX	flip-flop	TWX	INTLU
	contention	TWX	CONTLU
TLX	flip-flop	TLX	INTLU
	contention	TLX	CONTLU

Where alternative TRMTYPE options are given, they are synonyms; the same support is generated whichever keyword is used.

The following example shows a typical DFHTCT TYPE=TERMINAL macro for the interactive logical unit.

```
DFHTCT TYPE=TERMINAL,
       ACCMETH=VTAM,
       TRMTYPE=              see table
       SESTYPE=              see table
       TRMIDNT=yyyy,
       NETNAME=xxxxxxxx,
       TRMSTAT=TRANSCEIVE,   or NOINTLOG - see Automatic Transaction
       TIOAL=256,                          Initiation
       BUFFER=256,
       RUSIZE=               default is 256
       BRACKET=YES,
       CONNECT=AUTO,         do not specify for devices supported via NTO
       PGESTAT=PAGE,
       PAGESIZE=             default is (12,80)
       VF=YES,HF=YES         do not specify for devices supported via NTO
```

READ-AHEAD QUEUING

Read-ahead queuing can be selected for any transaction designed to run with the interactive logical unit by specifying RAQ=YES in the DFHPCT TYPE=ENTRY macro for the transaction.

This specification causes CICS/VS to maintain an outstanding VTAM receive for the logical unit during the lifetime of the transaction. Input received from the logical unit is queued on CICS/VS temporary storage, and input requests from the transaction are satisfied from this queue. The queue is purged when the transaction terminates.

For information on the use of this facility, see "Contention Mode" and "Unsolicited Input" later in this chapter.

SYSTEM GENERATION

Suitable versions of the Terminal Control Program and the Basic Mapping Support Program must be generated by specifying the interactive logical unit in the VTAMDEV operand of the DFHSG PROGRAM=TCP macro and the BMSDEV operand of the DFHSG PROGRAM=BMS macro; see CICS/VS System Programmer's Reference Manual.

Alternatively, pregenerated versions of these programs may be used; see the appropriate CICS/VS System Programmer's Guide.

SNA Protocols and CICS/VS Programming

This section discusses the SNA protocols that are used for the interactive logical unit and relates them to CICS/VS application and system programming requirements.

BIND COMMAND

CICS/VS uses the VTAM open-destination macro to transmit a BIND command to the interactive logical unit. The bind formats that CICS/VS uses for the interactive (flip-flop) and interactive (contention) logical units are described in appendix D. The binds are identical except for the specification of flip-flop or contention mode. Note that an additional bind section (bytes X'23' to X'29') is used for devices supported via the NTO option of NCP/VS.

BRACKET PROTOCOL

Bracket protocol is used for both the interactive (flip-flop) and the interactive (contention) logical units. Initiation and termination of brackets is illustrated in Figure 3-1.

Bracket Initiation

The interactive logical unit may initiate a bracket at any time by sending a request unit (RU) with the begin-bracket (BB) indicator set. CICS/VS must accept this request; unless, of course, CICS/VS is already in bracket state (a bracket state error).

CICS/VS requests permission to begin a bracket by sending an SNA BID command to the logical unit. This may occur because a transaction has been started by Automatic Transaction Initiation and wishes to communicate with the logical unit, or the logical unit has been named in a CICS/VS ROUTE list for a routed message. The interactive logical unit can respond to the BID in a number of ways.

1. Positive response - CICS/VS may begin a bracket by sending begin-bracket. The interactive logical unit will respond positively to a BID if it is not in send state and is not already in bracket state.

2. Negative response with sense code X'0813' – the BID has been rejected. For the interactive logical unit, this situation arises because of a race condition; the logical unit has sent begin-bracket, and switched to bracket state at the same time that CICS/VS, which is not yet in bracket state, has sent the BID. The sense code indicates that the logical unit will not subsequently send the SNA ready-to-receive (RTR) command; CICS/VS must BID again.

3. Negative response with sense code X'081B' – the BID has been rejected because the logical unit is in send state but not in bracket state.

The negative responses are handled by CICS/VS, which queues the BID for retry when the session is available. Note that the node error program (DFHZNAC) is not driven by these responses.

Bracket Termination

Once a bracket has been initiated, it can be terminated only by CICS/VS; the bind specifies that the logical unit may not send end-bracket (EB).

If the CICS/VS application program terminates while the session is still in bracket state, CICS/VS will send a null RU carrying the EB indicator. The CICS/VS application programmer can avoid this extra flow by specifying the LAST operand on the last SEND command that the application issues. This operand tells CICS/VS to insert the EB indicator on this final transmission.

The application will be abnormally terminated if it attempts to issue a RECEIVE command after issuing a SEND LAST command.

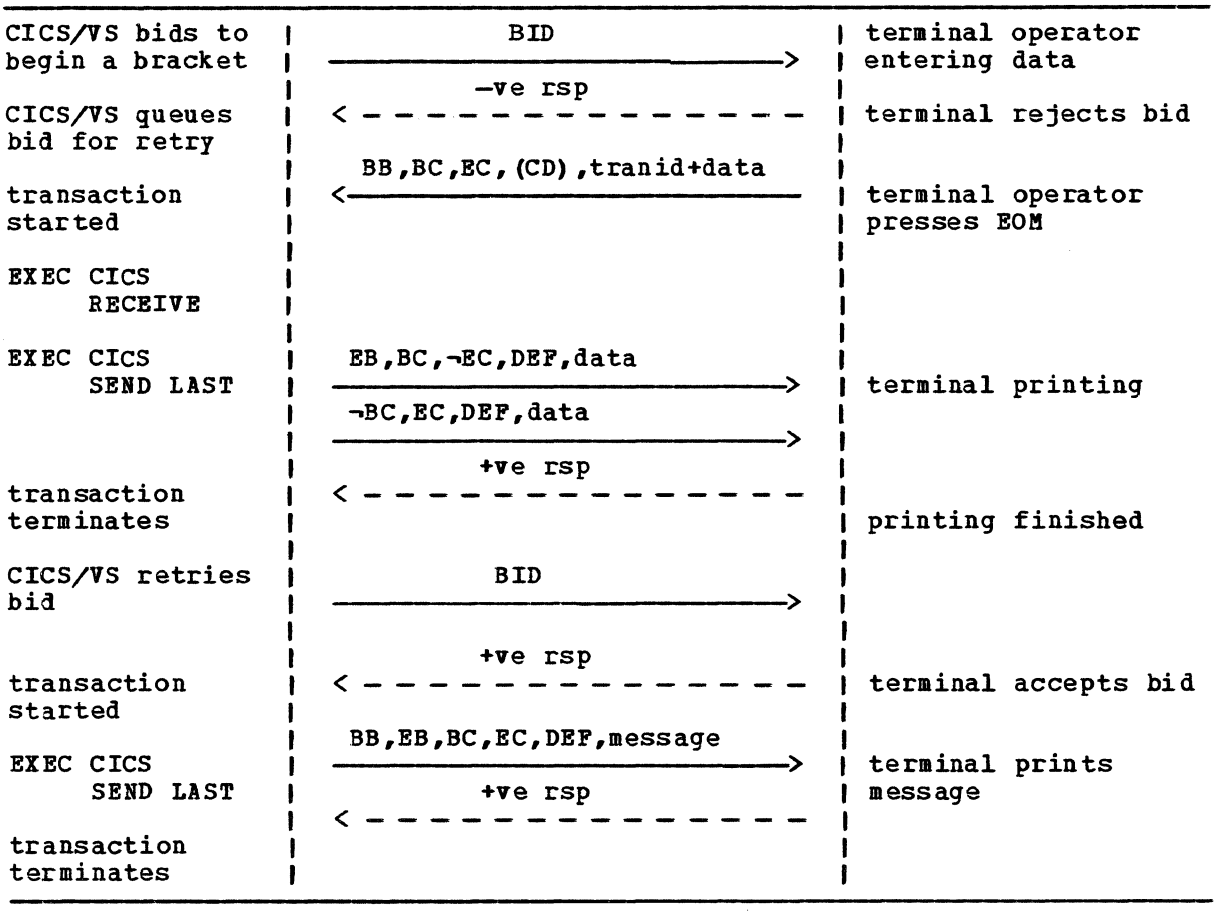


Figure 3-1. Bracket Initiation and Termination

CHAINING

Request unit chaining is used for all transmissions between CICS/VS and the interactive logical unit. A chain consists of a sequence of one or more RUs with the following properties.

- The first RU is marked "begin-chain" (BC).
- The last RU is marked "end-chain" (EC).
- RUs that are neither first nor last are marked "not-begin-chain, not-end-chain" (¬BC,¬EC).

The following alternative terminology is sometimes used to refer to the chaining indicator settings:

```
 BC,¬EC  -  first in chain   (FIC)
¬BC,¬EC  -  middle in chain  (MIC)
¬BC, EC  -  last in chain    (LIC)
 BC, EC  -  only in chain    (OIC)
```

The bind image transmitted to the interactive logical unit contains the maximum RU sizes for RUs transmitted between CICS/VS and the logical unit. The CICS/VS system programmer can specify the RU sizes in the terminal control table entry (DFHTCT TYPE=TERMINAL) by means of the BUFFER operand for RUs outbound from CICS/VS to the logical unit and the RUSIZE operand for RUs inbound from the logical unit to CICS/VS. In general, a value of 256 bytes is suitable for the interactive logical unit.

CICS/VS should normally be allowed to control data chaining for transactions that run with the interactive logical unit. CICS/VS will assemble inbound RUs into complete chains provided that CHNASSY=YES is specified for the terminal control program (DFHSG PROGRAM=TCP) and also in the terminal control table terminal entry (DFHTCT TYPE=TERMINAL). On output, CICS/VS will, when necessary, split the output message into separate RUs, unless the MSGPREQ=CCONTRL operand of the DFHPCT TYPE=OPTGRP macro is used to specify that the transaction will control its own chaining.

RESPONSE MODE

The CICS/VS bind for the interactive logical unit specifies that both exception and definite response modes are permitted.

Note that the unit of transmission to which a response applies is a chain. In exception response mode, all RUs in the chain are marked exception-response. In definite response mode, the last RU in the chain is marked definite-response, all others are marked exception response.

CICS/VS always requests definite response to SNA commands.

For all other flows, CICS/VS requests exception response unless:

1. The chain carries the EB indicator.
2. The message protection option is specified for the transaction (MSGINTEG option of DFHPCT TYPE=ENTRY).
3. The CICS/VS application programmer specifies the DEFRESP option on a SEND or CONVERSE command.

The specification of both definite and exception response protocols in the bind causes the interactive logical unit to select a default protocol which is dependent on the terminal type:

1. The IBM 3767 defaults to exception response request mode.
2. The IBM 3770 interactive logical unit defaults to definite response request mode.

FLIP-FLOP MODE

In flip-flop mode, the exchange of messages between CICS/VS and the interactive logical unit is structured by means of the change-direction (CD) indicator. Initially, the sender of begin-bracket has the flow, and has the right to continue sending until it invites the receiver to send by including the CD indicator on the last RU in the chain, at which point the roles are reversed.

Figure 3-2 shows a simple exchange using flip-flop protocols.

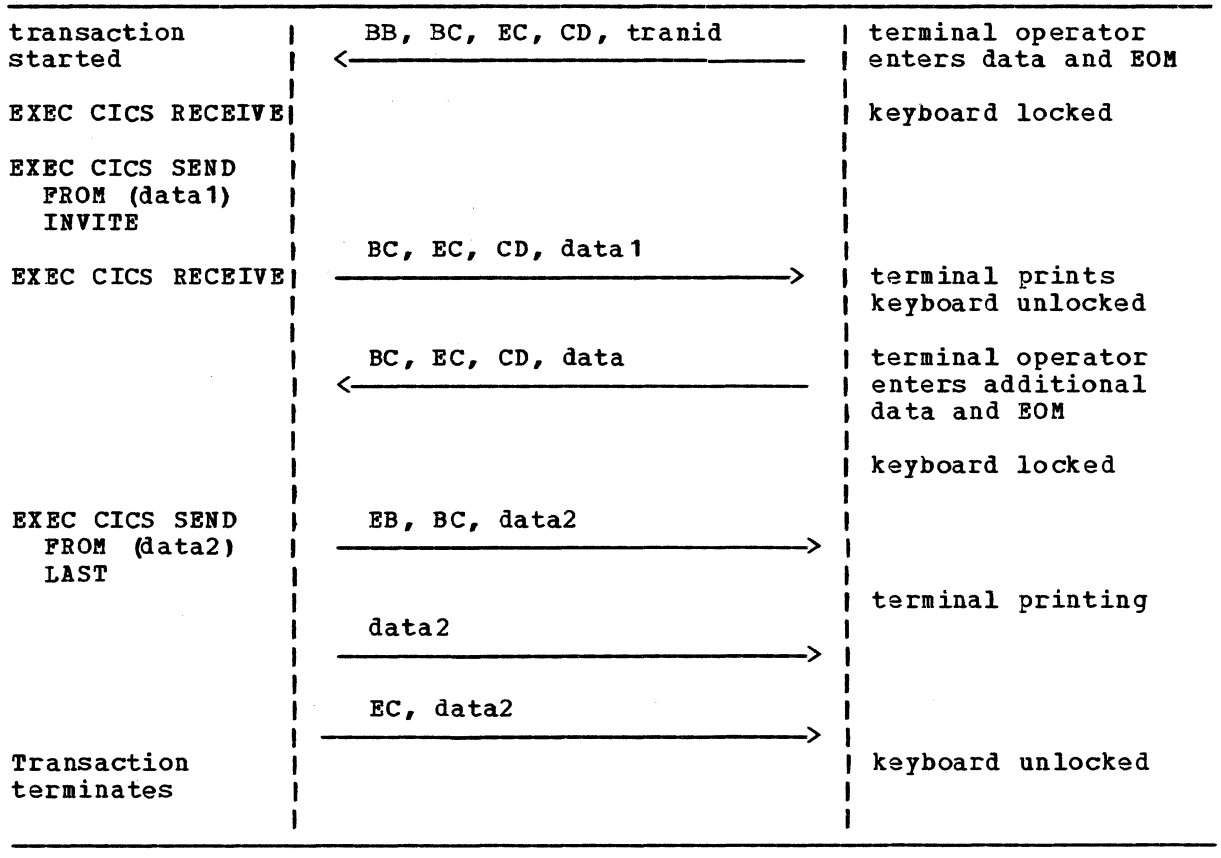


Figure 3-2. Flip-Flop Mode Protocol

The 3767 or 3770 transmits CD with the last RU of the chain whenever the operator presses the EOM key, so that CICS/VS is given the flow after each chain. The CICS/VS application programmer has a number of ways of influencing the sending of CD to the logical unit:

1. The issuing of a RECEIVE command when CICS/VS is in send state - CICS/VS will send an RU carrying the CD indicator before issuing the VTAM receive macro.

2. The use of the CONVERSE command - CD is sent with the message in preparation for the implied receive operation.

3. The use of the INVITE option on a SEND command - CD is sent with the message.

Note that CICS/VS often defers the sending of messages (unless the WAIT option is specified). For instance, the message generated by a SEND command will not be transmitted until a later event in the program, such as another SEND command being executed or the program terminating. This technique frequently reduces the number of messages, compared with immediate transmission, because indicators such as CD or EB can often be sent with the data.

Flip-flop protocols apply only to normal-flow RUs. The SIGNAL command, which is an expedited-flow RU, can be sent regardless of the current normal-flow direction, and provides a way for the current

receiver to request permission to send. See "The SIGNAL Command" later in this chapter.

CONTENTION MODE

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.

Figure 3-3 shows an exchange in contention mode. Note that CICS/VS defers output, as described under "Flip-Flop Mode".

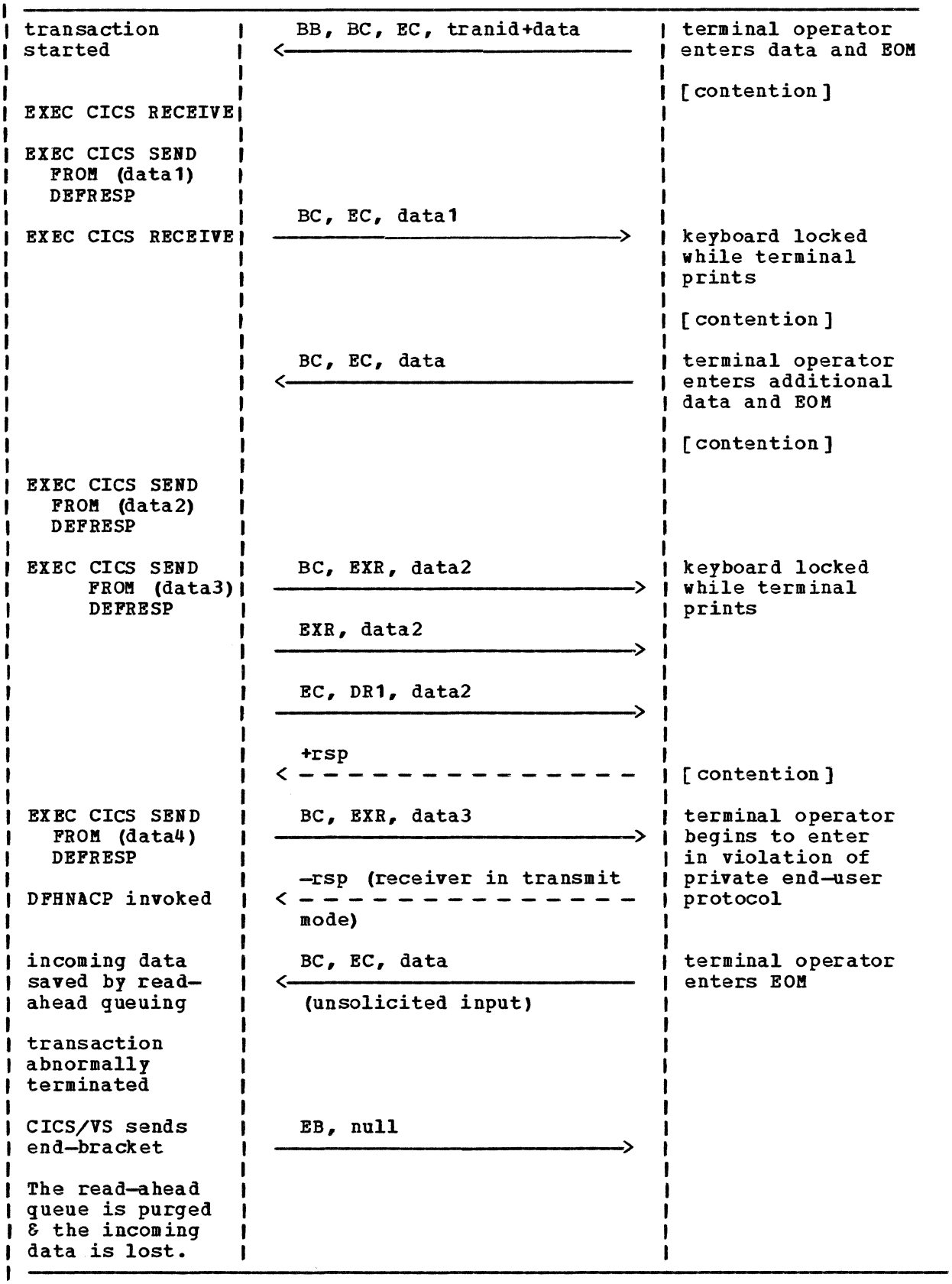


Figure 3-3. Contention Mode Protocol

In contention mode, both CICS/VS and the interactive logical unit have three conceptual states: send state, receive state, and contention state. The arrival of a message when the receiver of the message is in contention state will cause it to switch to receive state; when end-of-chain is transmitted, both transmitter and receiver revert to contention state. The flip-flop session may therefore be used for sequences of message exchanges provided that a suitable end-user protocol is established for the application and the terminal operator. Contention occurs when both parties attempt to send at the same time, so that each is in send state when the message from the other arrives.

In a session between CICS/VS and the interactive LU, CICS/VS is designated the "contention loser" and the interactive logical unit is designated the "contention winner". SNA specifies the following rules for the resolution of contention.

1. Messages arriving at the contention loser, if it is sending, must be queued.

2. Messages arriving at the contention winner, if it is sending, can either be queued or be rejected with an appropriate negative response. The interactive LU always returns a negative response.

To the CICS/VS user, these two aspects of contention manifest themselves in the following ways:

1. The interactive logical unit sends data to CICS/VS, but the CICS/VS application progam has not issued a receive command.

   Under these circumstances, the queuing mechanism specified by SNA is provided by CICS/VS if read-ahead queuing has been specified for the transaction, or by ACF/VTAM otherwise.

2. The CICS/VS application issues a SEND command, but the interactive logical unit is in send state.

   In this case, the interactive logical unit returns a negative response with sense code X'081B', signifying "receiver in transmit mode". The default action taken by CICS/VS on receipt of the negative response depends upon a number of factors; also, the user may modify the default action by means of a suitable node error program (NEP). For further information refer to "Receiver in Transmit Mode" later in this chapter.

The use of the CD indicator is not precluded in contention mode operation. This indicator tells the receiver that it should switch directly from receive state to send state, without entering contention state; it is an assurance that the sender will not attempt to send again before it has issued a receive.

The interactive logical unit does not send CD to CICS/VS in contention mode. The CICS/VS application programmer, however, may use SEND INVITE to switch the logical unit into send state.

SIGNAL COMMAND

The SIGNAL command is an expedited-flow SNA command that may be sent by either CICS/VS or the interactive logical unit regardless of the current direction of the normal-flow. It carries a four-byte signal code that specifies the request that is being signaled. The only signal code that is sent between CICS/VS and the interactive logical unit is "request change direction" (X'0001' plus a two byte reserved field).

The SIGNAL command may be used in either flip-flop or contention mode. In flip-flop mode, it requests the current sender to send CD. In contention mode, it requests the current sender to desist from sending.

SIGNAL Outbound from CICS/VS

The CICS/VS application programmer can cause SIGNAL to be sent by means of the ISSUE SIGNAL command (or the DFHTC TYPE=FORCE macro). When the signal command is received by the interactive logical unit, the keyboard is locked, and any data remaining in the device buffers is transmitted to CICS/VS, with the CD indicator if flip-flop mode is being used. The application may then issued a SEND command.

The sending of SIGNAL may also be specified in a node error program as a way of forcing through a message that has been rejected because of contention; see "Receiver in Transmit Mode" later in this chapter.

The use of the SIGNAL command in CICS/VS application programs should normally be restricted to cases in which the application has detected an error and has an urgent need to get a message to the the terminal operator.

SIGNAL Inbound to CICS/VS

The interactive logical unit sends SIGNAL to CICS/VS as the result of some operator action, such as depression of the attention key on the 3767.

The CICS/VS application program can detect the inbound SIGNAL by use of a HANDLE CONDITION command for the SIGNAL condition. The HANDLE, which must have been executed before SIGNAL arrives, specifies the label of a routine that is to be given control when the SIGNAL condition is raised. Note that the branch is not necessarily taken immediately on receipt of SIGNAL; unless a WAIT SIGNAL command has been issued, the raising of the condition is deferred until the next SEND, CONVERSE, or RECEIVE command.

In the absence of a HANDLE for the SIGNAL condition, the condition is ignored.

Inbound SIGNAL provides a useful way for a terminal operator to signal a "ready" state to the application. For example, an application that is printing multiple pages at the terminal could issue a WAIT command at the completion of each page. The operator then uses the attention key to indicate readiness for the next page.

The application designer, however, must be aware that the operator may violate established end-user protocols at any time by hitting the attention key.

ERROR CONDITIONS

Unsolicited Input

Unsolicited input is most likely to be received from the terminal during contention mode sessions, and is usually the result of violation of the private protocols established for the CICS/VS application and the terminal operator. The CICS/VS user should be aware of two possible types of unsolicited input:

1. Input for which the CICS/VS application has not issued a receive command.

2. Input which is passed to the CICS/VS application program in response to a receive command, but which is not the input that was expected.

For case 1, the way in which the unsolicited input is handled depends on whether or not read-ahead queuing has been specified for the transaction.

If read-ahead queuing is specified, the incoming data is queued on temporary storage by CICS/VS. If the transaction subsequently issues a receive command, the input is recovered from the temporary storage queue. If the transaction terminates without issuing a receive command, the queue is purged and the input is lost.

If read-ahead queuing is not specified, the incoming data is retained in VTAM's buffers. CICS/VS uses either a VTAM receive-any macro or a VTAM receive-specific macro to recover this data.

When no CICS/VS transaction is attached for a particular terminal, the terminal is in VTAM receive-any mode, and input from the terminal is received into the receive-any pool that CICS/VS maintains for VTAM terminals. When input from the terminal causes a transaction to be attached, the terminal is placed in receive-specific mode, and CICS/VS now issues VTAM receive-any macros in response to RECEIVE commands from the CICS/VS application. When the transaction terminates, the terminal reverts to receive-any mode.

Therefore, if the transaction subsequently issues a receive command, the input is recovered from VTAM and passed to the application; if the transaction terminates without issuing a receive command, the data is acquired by CICS/VS. Provided that the incoming data is a valid transaction request, the new transaction is started; otherwise, the input is discarded with an appropriate error message.

Unsolicited input may be received whenever the terminal operator enters data at an unexpected time. For example, a transaction may be designed to prompt the terminal operator by means of a SEND command, and then to issue a RECEIVE command to receive the inbound data. If the operator does not wait for the prompt, but starts to enter another message, contention occurs. The prompt message will be rejected (see "Contention Mode" and "Receiver in Transmit Mode"), and the inbound data will be queued, either by CICS/VS or by VTAM.

Note that if the application is allowed to proceed and issue a receive command, the queued input data will be acquired; but it may be totally unrelated to the transaction. This is the form of unsolicited input referred to in (2) at the beginning of this section.

| Receiver in Transmit Mode
|

| The interactive logical unit sends a negative response signifying
| "receiver in transmit mode" whenever it receives a normal-flow request
| while it is in send state or while it is awaiting a definite response to
| a previous transmission. The sense code is X'081B' and the associated
| CICS/VS error message is DFH2843.

| Note: The interactive logical unit may also send sense code X'081B' in
| response to BID. CICS/VS treats this response as a normal bid reject,
| sense code X'0813'; this section does not apply to this case. See
| "Bracket Protocol" earlier in this chapter.

| When the negative response is received, the node abnormal condition
| program (DFHZNAC) is scheduled. The default action taken by DFHZNAC is
| as follows:

| 1. If CICS/VS is not in send state when the response is received, the
| VTAM send is purged, the transaction is abnormally terminated, and
| a VTAM close destination macro is issued to break the connection
| (DFHZNAC action flags X'60E001').

| 2. If the rejected chain was sent with definite response requested:

| a. For the interactive flip-flop logical unit, the send is retried
| (DFHZNAC action flags X'600000').

| b. For the interactive contention logical unit, the VTAM send is
| purged, and the CICS/VS transaction is abnormally terminated
| (DFHZNAC action flags X'60E000').

| 3. If exception response was requested, the VTAM send is purged, and
| the CICS/VS transaction is abnormally terminated (DFHZNAC action
| flags X'60E000').
|
| For case 2.b (definite-response send in contention mode) the CICS/VS
| user can write a node error program (NEP) to retry the failing send.
| 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", that is, send SIGNAL to the
| logical unit and then retry the send.

| Note that for case 1, a NEP is not allowed to reset the "break
| connection" flag, so that recovery is not possible. Also, for case 3, a
| retry should not be enforced as there is no guarantee that the original
| TIOA is still available.

| Automatic Transaction Initiation
|

| Automatic Transaction Initiation (ATI) can be used to initiate a
| transaction that communicates with the interactive logical unit. The
| steps that CICS/VS takes during ATI are:

1. Establish a session with the logical unit if one does not already exist.

   CICS/VS issues a VTAM SIMLOGON macro to acquire the logical unit. VTAM in turn drives the CICS/VS logon exit, and CICS/VS issues a VTAM OPENDST macro to initiate the session.

   This logon mechanism should not be used for devices connected via the NTO option of NCP/VS. The terminal control table terminal entry for such devices should specify TRMSTAT=NOINTLOG; this causes the ATI request to be queued until a session is acquired by some other means; for example, by operator logon.

2. When a session is available, and the session is not in bracket state, issue a BID command.

3. If the BID is accepted, initiate the transaction.

   If the BID is rejected, it is queued for retry when the session is once more out of bracket state.

Basic Mapping Support (BMS)

The full range of BMS function is available when BMS is used to construct data streams for an interactive logical unit. In addition, some extra operands are supplied for use with the various BMS macro instructions. Only the operands and related facilities that are used for communication with interactive logical units, but that are not otherwise applicable, are discussed here; for all the general BMS macro instruction and function descriptions, see the CICS/VS Application Programmer's Reference Manual (Command Level).

OFFLINE MAP BUILDING

The DFHMSD macro instruction includes the TERM operand, which is used by BMS to select the correct map(s) for a particular device. The parameter to be coded in the TERM operand for an interactive logical unit (flip-flop or contention mode) is INTLU (or 3767 or 3770I) or SCS for SNA character strings. The alternatives, in the case of INTLU, result in exactly the same support being generated; the alternatives are given in case your documentation would be made clearer by referring to the device types.

As a further alternative, interactive logical units can use the ALL parameter, with the usual proviso that the page size used must be limited to that of the smallest device to be used with the map set.

The HTAB and VTAB Operands

The HTAB and VTAB operands of the DFHMSD macro are available to specify the logical tab settings to be used when communicating with 3767 and 3770 logical units with the horizontal and vertical forms control features. If these operands are specified in a DFHMSD macro, they will apply to all maps defined in the map set.

BMS will use horizontal and vertical tab control characters only if HF=YES and VF=YES respectively are specified in the DFHTCT TYPE=TERMINAL macro; otherwise, line spaces and line feeds will be used to obtain the equivalent result. Horizontal and vertical tabs should not be specified for devices supported via the NTO option of NCP/VS.

The physical tabs on the 3767 or 3770 can be set to correspond to the logical tab settings either manually at the terminal, or by a user-written program that sends a stream of special characters to the printer. An example of a user-written program is shown in Figure 3-4. To ensure that the output data begins at the correct position, a new line (NL) or carriage return (CR) character should be sent to the printer before the first line of data.

Logical tab settings do not alter the printed output resulting from a BMS output request; however, careful setting of tabs can result in printing time being saved. When used for input and output, tabs can result in fewer data characters being sent to and from the logical unit, because of suppressed blank characters in a line, or of totally suppressed blank lines when VTAB is used.

```
TCTTEAR  EQU   11
TIOABAR  EQU   10
         SPACE 1
         PRINT OFF
         COPY  DFHCSADS
         COPY  DFHTCADS
         COPY  DFHTCTTE
         COPY  DFHTIOA
         PRINT ON
         SPACE 1
PGMX     CSECT
         BALR  3,0
         USING *,3
         SPACE 1
         B     START
         SPACE 1
*
* TAB SETTING DATA STREAM
*
TABS     DS    0H
         SPACE 1
HTAB     DC    X'2BC1' CONTROL CHAR INDICATING HTAB'S
         DC    AL1(HTABLEN-2)  COUNTER
         DC    AL3(0)    RESERVED FOR MPP(MAX.PRINT.POS.),
*                         LM(LEFT MARG.),RM(RIGHT MARG.)
         DC    AL1(1,11,21,31,41,51)  HTAB'S : -1-11-21-31-41-51-
HTABLEN  EQU   *-HTAB
         SPACE 1
VTAB     DC    X'2BC2'  CONTROL CHAR INDICATING VTAB'S
         DC    AL1(VTABLEN-2)  COUNTER
         DC    AL3(0)    RESERVED FOR MPL(MAX.PRINT.LINES),
*                         TM(TOP MARG.),BM(BOTTOM MARG.)
         DC    AL1(10,20,30)          VTAB's : -10-20-30-
VTABLEN  EQU   *-VTAB
         SPACE 1
TABSLEN  EQU   *-TABS  DATA STREAM LENGTH
         SPACE 1
START    DS    0H
         L     TCTTEAR,TCAFCAAA
*
         MVC   TCASCNB,=AL2(TABSLEN)
         DFHSC TYPE=GETMAIN,CLASS=TERMINAL
         L     TIOABAR,TCASCSA
         MVC   TIOADBA(TABSLEN),TABS
         MVC   TIOATDL,=AL2(TABSLEN)
         ST    TIOABAR,TCTTEDA
         DFHTC TYPE=(WRITE,WAIT)
*
         DFHPC TYPE=RETURN
         LTORG
         END   PGMX
```

Figure 3-4. (Part 1 of 2) Physical Tab Setting Sample Program

```
PGMX     CSECT
         EXEC CICS SEND FROM(TABS) LENGTH(TABSLEN) WAIT LAST
         EXEC CICS RETURN
         SPACE 1
*
* TAB SETTING DATA STREAM
*
TABS     DS    0H
         SPACE 1
HTAB     DC    X'2BC1'          CONTROL CHAR INDICATING HTAB'S
         DC    AL1(HTABLEN-2)   COUNTER
         DC    AL3(0)           RESERVED FOR MPP(MAX.PRINT.POS.),
*                               LM(LEFT MARG.),RM(RIGHT MARG.)
         DC    AL1(1,11,21,31,41,51)  HTAB'S : -1-11-21-31-41-51-
HTABLEN  EQU   *-HTAB
         SPACE 1
VTAB     DC    X'2BC2'          CONTROL CHAR INDICATING VTAB'S
         DC    AL1(VTABLEN-2)   COUNTER
         DC    AL3(0)           RESERVED FOR MPL(MAX.PRINT.LINES),
*                               TM(TOP MARG.),BM(BOTTOM MARG.)
         DC    AL1(10,20,30)          VTAB's : -10-20-30-
VTABLEN  EQU   *-VTAB
         SPACE 1
TABSLEN  EQU   *-TABS  DATA STREAM LENGTH
         SPACE 1
         END   PGMX
```

Figure 3-4. (Part 2 of 2) Physical Tab Setting Sample Program

Chapter 4. The Batch Logical Unit

Introduction

This chapter deals with communication between a CICS/VS application program and a batch logical unit (BCHLU). It describes only those aspects that are peculiar to such a session; general facilities and services available to the CICS/VS application programmer are described in the CICS/VS Application Programmer's Reference Manuals. (The batch logical unit must not be confused with the batch data interchange logical unit, which is described in Chapter 5 of this manual.)

The batch logical unit is designed for use when there are only infrequent changes in the direction of data flow.

System Programming for the Batch Logical Unit

It is necessary to specify to CICS/VS the type of logical unit that will be used with the 3770. The way this is done is through the TRMTYPE operand of the DFHTCT TYPE=TERMINAL system generation macro. To specify a batch logical unit, use any of the following parameters:

BCHLU or 3770B or 3770

The alternatives are synonymous: the same support will be generated whichever keyword is chosen.

BCHLU or 3770B (but not simply 3770) must also be specified in both the VTAMDEV operand of the DFHSG PROGRAM=TCP system generation macro and in the BMSDEV operand of the DFHSG PROGRAM=BMS macro.

If the application program is to use BMS commands or macros to communicate with the batch logical unit, the DFHSG PROGRAM=DIP macro must be coded to generate the batch data interchange program (described in Chapter 5). This is because BMS uses the services of the batch data interchange program to construct function management headers (described below). Also, INBFMH=DIP must be specified in the DFHPCT TYPE=ENTRY macro.

Full details of all operands for these macros are given in the CICS/VS System Programmer's Reference Manual.

The keyboard-printer of the 3770 can be used with interactive application programs during the same session as batch exchanges. The mode of operation will be interactive flip-flop, as described in Chapter 3.

In order to use the CICS/VS logical device code (LDC) facility for device selection, the system programmer must define a system LDC table containing the LDCs corresponding to the devices to be used, and identify the valid LDC names on each TCTTE defining a batch logical unit.

VTAM Requirements

For a card reader in a 3770 system, the BUFLIM operand of the LU definition statement must be specified. The default value of BUFLIM is 2, and this value would probably result in the session being terminated under peak load conditions. The value specified should be sufficiently large to cater for the maximum number of punched cards that will constitute the input for a single transaction.

Application Programming for the Batch Logical Unit

CICS/VS provides the application program with various methods of communicating with a batch logical unit. One method is through the commands provided by terminal services; another is through the commands provided by basic mapping support (BMS). The use of these commands is described in the CICS/VS Application Programmer's Reference Manual (Command Level). Alternatively, the application program may be written using CICS/VS macros, which are described in the CICS/VS Application Programmer's Reference Manual (Macro Level). Some considerations that apply particularly to the batch logical unit are dealt with below.

FMH HANDLING

The component devices of a batch logical unit may be explicitly selected for output. Such selection remains in effect until explicitly terminated. Selection is performed by means of a control header called the function management header (FMH). All data flowing to a particular component is termed a data set. The component is thus selected by the begin-dataset FMH and selection is terminated by the end-dataset FMH. Explicit selection by FMH is required for all components except the console printer.

On input to CICS/VS, the source is indicated by means of a begin-dataset FMH, for all components except the keyboard. The selected input component remains active until an end-dataset FMH is sent.

When the CICS/VS application program uses BMS to communicate with the batch logical unit, the programmer does not need to consider FMHs. Destination selection and deselection are carried out automatically by CICS/VS. If, for any reason, the application program must inspect the inbound FMH or must itself construct the outbound FMH, it must use terminal control commands to do so. When using terminal control commands to communicate with the batch logical unit, the application programmer is fully responsible for destination selection and deselection; the procedures for doing so are described in detail in the sections below.

An output destination selection remains in effect until explicitly terminated. It is thus important that the application program ensures a begin-dataset FMH request has been positively accepted before sending further data for that component. Also, since a data set remains selected until an end-dataset FMH is sent, it is important to ensure that a data set is correctly deselected before attempting to send console messages (which may normally be sent without an FMH). If the CICS/VS application abends, the user program error program (PEP) may issue a WRITE request to deselect the active destination, provided that the abend was not caused by a condition notified via the node error program (NEP).

To ensure correct output component selection, the DEFRESP and FMH options must be specified the WRITE or CONVERSE commands when the TIOA contains a begin- or resume-dataset FMH.

Inbound Function Management Header (FMH) Notification

The CICS/VS application program can request notification when a function management header (FMH) is included in the data received during a read from a batch logical unit. The FMH for the batch logical unit is a six-byte field that can be sent to, and received from, the batch logical unit via the TIOA; when present, the FMH occupies the first six bytes of the TIOA. The format of the FMH used for batch logical units is given in Figure 4-1.

Byte	Bits	Meaning
0	0-7	FMH length; coded as X'06'.
1	0-7	FMH type (1); coded as X'01'.
2	0-7	Device code; coded as follows: X'00' - console printer X'10' - disk 1 (input and output) X'11' - disk 2 (input and output) X'20' - card punch X'20' - card reader X'30' - line printer
3	0-7	Reserved; coded as X'00'.
4	0-2	Data set control; coded as follows: '000'B - Resume data set '001'B - End data set '010'B - Begin data set '011'B - Chain contains begin and end data set '100'B - Suspend data set
4	3	Feature not supported; coded as '0'B.
4	4	Reserved; coded as '0'B.
4	5-7	Reserved; coded as '0'B.
5	0-7	Reserved; coded as X'00'.

Figure 4-1. Function Management Header (FMH) Format

Whether or not inbound FMHs will be passed to the application program is specified by the system programmer in the PCT. He can specify that no inbound FMHs will be passed, or that only the FMH at the end of the data set will be passed, or that all inbound FMHs will be passed. If he specifies that all inbound FMHs will be passed to the application program, you should code a HANDLE CONDITION INBFMH command. This command will instruct CICS/VS to give control to a user-written routine whenever an inbound FMH is received.

Your inbound-FMH routine could investigate the contents of the FMH and take some action depending on, for example, which device the data has come from. The routine would then scan the TIOA for input data, starting at the appropriate byte, located by means of the FMH length byte at the start of the TIOA. If the data is initial data from a logical unit, the transaction identification will start at the seventh byte. However, input data from the console keyboard will not be

preceded by an FMH; in this case, of course, the transaction ID would appear in the first bytes of the TIOA in the normal way.

When input data is received as a chain of RUs, only the first (or only) RU of the chain is ever preceded by an FMH.

End of Data Set (EODS) Notification

The CICS/VS application program can request notification when end of data set (EODS) is received during a read from a batch logical unit. To request EODS notification, the application programmer must code a HANDLE CONDITION EODS command. This command specifies the address of a routine that is to receive control when EODS is encountered.

The end-of-data-set routine will receive control when EODS is encountered, regardless of whether or not the FMH (which contains the EODS indication) is to be passed to the application program. If the INBFMH operand is also coded, the EODS operand overrides it. To give control to an inbound-FMH routine (probably the one that would have been used if EODS had not been encountered) you can issue a DFHTC TYPE=WAIT macro, with the INBFMH operand specified, within your end-of-data-set routine.

Outbound Function Management Header (FMH)

When sending output data to any batch logical unit device other than the console printer, it is also necessary to send a function management header (FMH) to select the required output component. The FMH for the batch logical unit is a six-byte field and, when sent, must occupy the first six bytes of the TIOA to be used for the write operation. Any data to be sent to a device will be placed after the FMH. Whenever an FMH is supplied in a TIOA, the FMH option of the SEND or CONVERSE command must be specified.

The format of the FMH is given in Figure 4-1. The functions of the FMH when sent to a batch logical unit are:

- Data set control: this consists of setting on or off the three bits named in the fifth byte (byte 4) of the FMH; these are: beginning of data set (BODS), end of data set (EODS), suspend data set, and resume data set.

- Device control: this is achieved by setting the third byte (byte 2) of the FMH to a value, selected from those given in Figure 4-1, which will determine the device to which the output data will be directed.

When sending output to the console printer, the use of FMHs is optional. For communication with all other devices in a batch logical unit, an FMH must be supplied at the beginning of a data set and another at the end of the data set. Each of these FMHs must specify the device code that corresponds to the device that is being written to. The first FMH must have the BODS bit on (set to '1'B); the last FMH must have the EODS bit on. The first FMH can be followed in the TIOA by the data; FMH and data together must not exceed 256 bytes. The last FMH must be a stand-alone FMH sent after all of the data has been sent.

A data set sent to the batch logical unit can consist of multiple chains, but an FMH should be provided only at the beginning or end of

the data set or when suspending or resuming the data set. If the CCOMPL=NO operand is specified on a write request, the following write request must not supply an FMH.

The suspend-data-set bit is used when, at some point during a series of write requests to output a data set to, for example, the card punch, a need arises to give some information to the device operator. Rather than wait until the end of the data set, it is possible to send an FMH with the suspend-data-set bit on (and the device-code byte set to indicate that the card punch is the device that has the suspended data set) followed, in a separate TIOA, by a message directed to the console printer. The message can be preceded by an optional FMH, in the normal manner when communicating with the console printer. The suspended data set is then resumed by sending the next portion of the data set, with a preceding FMH. In this FMH, the device-code byte must specify the original device that has the suspended device set, and none of the three data set control bits (suspend, BODS, and EODS) should be on.

Only one data set can be suspended at any one time for one logical unit, and suspension can only be used to send data to the console printer. (If the user wishes to interrupt the process of reading a data set to send a message to the console printer, the suspension is handled by CICS/VS. The user need not take the actions described above for writing.)

Whenever an FMH is sent, unless the output is to be sent to the console printer, the device-code byte must be supplied. Any data set sent to a batch logical unit either without a preceding FMH, or with an FMH that has a device-code byte coded X'00', or with an FMH specifying an output device that does not exist, will be directed to the console printer or to a default component defined for that logical unit.

As an alternative to filling the device-code byte directly, you can use the symbolic LDC representing the required device, and the DFHTC CTYPE=LOCATE, LDC=YES macro instruction, which is normally used by the system programmer, to provide you with the actual device code. The LDC is a two-character identification given to each device that will communicate with VTAM. The system programmer prepares an LDC table in the TCT, containing each LDC, its corresponding "LDC numeric value" (the equivalent of the device-code byte contents in the FMH) and other information used for basic mapping support (BMS) operations. Full details on the use of the DFHTC CTYPE=LOCATE macro are given in the CICS/VS System Programmer's Reference Manual. In simplified terms, the LDC (mnemonic) is placed in TCATPLDM, and the LDC numeric value is returned in TCATPLDC, from which it must be loaded into the FMH.

Messages may be routed to the batch logical unit by means of the CMSG transaction. If no LDC is specified in the ROUTE operand of this transaction, the message is routed to the console printer.

SIGNAL Data-Flow-Control Commands

One of the SNA data-flow-control commands is SIGNAL. The SIGNAL command can be both sent and received by CICS/VS. When sent, it causes the logical unit to stop sending data and prepare to receive data written to it by CICS/VS. When received, an indicator is set in the TCTTE which can be tested by the application program by means of the DFHTC TYPE=SIGNAL macro.

The 3767 and 3770 batch logical units send SIGNAL to CICS/VS as the result of the operator pressing the attention key.

The CICS/VS application program can detect the inbound SIGNAL by use of a HANDLE CONDITION command for the SIGNAL condition. The HANDLE, which must have been executed before SIGNAL arrives, specifies the label of a routine that is to be given control when the SIGNAL condition is raised. Note that the branch is not necessarily taken immediately on receipt of SIGNAL; unless a WAIT SIGNAL command has been issued, the raising of the condition is deferred until the next SEND, CONVERSE, or RECEIVE command.

In the absence of a HANDLE for the SIGNAL condition, the condition is ignored.

Inbound SIGNAL provides a useful way for a terminal operator to signal a "ready" state to the application. For example, an application that is printing multiple pages at the terminal could issue a WAIT command at the completion of each page. The operator then uses the attention key to indicate readiness for the next page.

Card Punch Output

When writing data to a card punch logical unit, card boundaries must be indicated by inter-record-separator (IRS) characters. To save transmission time, the IRS character can be inserted after the last non-blank character in each card. The size of an output message to a card punch must not exceed 256 bytes (the RU size), including the FMH and the IRS characters. This restriction is necessary to avoid CICS/VS splitting a card image, as it would do if more than one RU were to be sent.

Bracket Protocol

Bracket protocol is used when CICS/VS communicates with a batch logical unit. For the most part the use of brackets is transparent to the CICS/VS application program.

Only on the last write operation of a task to a logical unit does the bracket protocol become apparent to the CICS/VS application program. On the last output request to a logical unit, the CICS/VS application program may specify LAST on the SEND command. The last output request is defined as either the last DFHTC TYPE=WRITE macro specified for a transaction not using chain control, or as the write operation that transmits the first or only request of the last output chain for a transaction employing output chain control. For further information, refer to the description in the CICS/VS System Programmer's Reference Manual of the CCONTRL parameter of the DFHPCT TYPE=OPTGRP macro. The LAST option causes CICS/VS to transmit an end-bracket indicator with the final output message to the logical unit. This indicator notifies the batch logical unit that the current transaction is ending. If the LAST option is not specified, CICS/VS waits until the task detaches before sending the end-bracket indicator. Since an end-bracket indicator is transmitted only with the first RU of the chain, the LAST operand is ignored for a transaction using chain control unless the request is the first or only one in the chain.

Basic Mapping Support (BMS)

The full range of BMS function is available when BMS is used to construct data streams for a batch logical unit. In addition, some extra operands are supplied for use with the various BMS macro instructions. Only the operands and related facilities that are used for communication with batch logical units, but that are not otherwise applicable, are discussed here; for all the general BMS macro instruction and function descriptions, see the CICS/VS Application Programmer's Reference Manual (Macro Level).

When constructing data streams for batch logical units, BMS builds any necessary function management headers (FMHs), and the application program need only supply logical device codes (LDCs), using the appropriate command options, to tell BMS which device the output is to be directed to.

OFFLINE MAP BUILDING

The DFHMSD macro instruction includes the TERM operand, which is used by BMS to select the correct map(s) for a particular device. The parameter to be coded in the TERM operand for a batch logical unit is BCHLU or 3770B. Each of the alternatives results in exactly the same support being generated; the alternatives are given in case your documentation would be made clearer by referring to the device type.

As a further alternative, batch logical units can use the ALL parameter, with the usual proviso that the page size used must be limited to that of the smallest device to be used with the map set.

When defining input maps, the user should bear in mind that, unlike terminal control commands, BMS input commands provide entire card images.

The HTAB and VTAB Operands

The HTAB and VTAB operands of the DFHMSD macro are available to specify the logical tab settings to be used when communicating with 3767 and 3770 logical units with the horizontal and vertical forms control features. If these operands are specified in a DFHMSD macro, they will apply to all maps defined in the map set.

The physical tabs on the 3767 or 3770 can be set to correspond to the logical tab settings either manually at the terminal, or by a user-written program that sends a stream of special characters to the printer. An example of a user-written program is shown in Figure 3-1. To ensure that the output data begins at the correct position, a new line (NL) or carriage return (CR) character should be sent to the printer before the first line of data.

Logical tab settings do not alter the printed output resulting from a BMS output request; however, careful setting of tabs can result in printing time being saved. When used for input and output, tabs can result in fewer data characters being sent to and from the logical unit, because of suppressed blank characters in a line, or of totally suppressed blank lines when VTAB is used.

The LDC Option

The LDC option is used to tell BMS the device to which it should send the data resulting from a BMS output request. It is used for communication with logical units that have more than one output component. The LDC option specifies the logical device code (LDC) for the required device. BMS uses the LDC numeric value to build the FMHs that must be sent to the logical unit. The mnemonic LDCs that you supply will be provided by the system programmer. The LDC option is specified in the DFHMSD macro instruction; the option is overridden by an LDC option specified in the first BMS output request for a particular logical message; if no LDC option is specified anywhere, the console is used as the default component.

Output to Card Punch

For output mapping of card images to be sent to a batch logical unit card punch, the logical message must not be too large to be transmitted in a single RU. This means that the number of characters that can be transmitted must not exceed 247-n, where n is the number of card images in the logical message (sent by a BMS TYPE=OUT or PAGEOUT request). In applying this formula, all characters up to and including the last non-blank character in the card image must be counted.

ONLINE BMS REQUESTS

The LDC Option

The LDC option can be specified in BMS SEND MAP, SEND TEXT, and CONVERSE commands. Its function is the same as that of the LDC option of the DFHMSD macro, described in the section "OFFLINE MAP BUILDING", earlier in this chapter. If an LDC is not specified in the first BMS output request for a particular logical message, the LDC option of the DFHMSD macro applies. If an LDC is specified in the first BMS output request, the LDC in this and subsequent BMS requests for the logical message applies.

When specified in a BMS ROUTE command, the LDC option overrides any LDC specified in a route list.

Data Set Considerations with BMS

When sending or receiving data sets to or from the batch logical unit with BMS input or output commands, the user does not need to indicate the beginning or end of the data set; suspending or resuming data set transmission, when data sets change, is also handled by BMS.

Sample CICS/VS 3770 Program

Appendix B describes the sample CICS/VS application program that is provided with the CICS/VS distribution tape. The program is an example of a CICS/VS application program designed for communication with a 3770 batch logical unit.

Chapter 5. The Batch Data Interchange Logical Unit

The batch data interchange program (DFHDIP), a component of CICS/VS, is designed to work in conjunction with the programmable models of the IBM 3770 Data Communication System. Communication between DFHDIP and the 3770 takes place through a logical unit assigned specially for the purpose and referred to as a batch data interchange logical unit. Batch data interchange sessions allow transaction and user records prepared at the 3770 to be transmitted to the host, and records resulting from host processing to be transmitted to the 3770.

System Programming for the Batch Data Interchange Logical Unit

The CICS/VS system programmer defines the batch data interchange LU by coding SESTYPE=BATCHDI in addition to TRMTYPE=3770 in the DFHTCT TYPE=TERMINAL macro.

BCHLU or 3770B must be coded in the VTAMDEV operand of the DFHSG PROGRAM=TCP macro and in the BMSDEV operand of the DFHSG PROGRAM=BMS macro.

The DFHSG PROGRAM=DIP macro must be coded to generate the batch data interchange program. Also, INBFMH=DIP must be specified in the DFHPCT TYPE=ENTRY macro.

Full details of all the system generation macros are given in the CICS/VS System Programmer's Reference Manual.

VTAM Requirements

In a 3770 system, the BUFLIM operand of the LU definition statement must be specified. The default value of BUFLIM is 2, and this value would probably result in the session being terminated under peak load conditions. The value specified should be sufficiently large to cater for the largest data set that will constitute the input for a single transaction.

Master Terminal Operations

Since the programmable 3770 models do not have a direct keyboard-to-line function, they cannot be used interactively with CICS/VS. These models are thus unsuitable for CICS/VS master terminal operations or other interactive CICS/VS services.

Application Programming for the Batch Data Interchange Logical Unit

The CICS/VS application programmer invokes the facilities provided by the batch data interchange program through batch data interchange commands. In addition, all BMS facilities available to the batch logical unit are available to the batch data interchange logical unit. BMS uses the services of DFHDIP to provide these facilities.

Note: The batch data interchange macro instruction DFHDI may be used only in assembler programs.

The CICS/VS batch data interchange program and the 3770 exchange data in the form of data streams. Each data stream contains a function management header (FMH) that defines the destination of the data stream; the FMH can also contain requests for data management operations to be performed on the data set. The CICS/VS application programmer requests these data management operations through the batch data interchange commands. He does not himself have to construct the outgoing FMH. The following table shows the data management operations available through the batch data interchange commands and indicates which operations can be performed on the various 3770 data sets. (The 3770 data sets are described in detail later in this guide.)

	Transaction data set (SYS.TDS)	Interrupt data set (SYS.INTR)*	User data sets
ADD record	No	No	Yes
REPLACE record	No	No	Yes
QUERY data set	Yes	No	Yes
NOTE	No	No	Yes

* SYS.INTR is not referenced directly. Printer or card punch output, generated in the same way as for non-programmable 3770 models, is directed to SYS.INTR.

The ADD and REPLACE record operations are self-explanatory. QUERY requests that an entire data set is transmitted by the 3770 to CICS/VS.

NOTE requests the 3770 to return the number of the next available record.

The CICS/VS application programmer can also use BMS to format the output to the 3770. The CICS/VS logical device code (LDC) facility can be used in the same manner as for non-programmable 3770 models. In addition, destination names can be specified via the LDC. BMS directs the output to SYS.INTR if no destination is specified. BMS can also be used to process input from the transaction data set.

The Batch Data Interchange Session

Communication between a CICS/VS application program and the 3770 takes place within an SNA session established between CICS/VS in the host and the batch data interchange logical unit in the 3770. The session is initiated by CICS/VS, but the stimulus for initiation of the session may originate either in the host system or in the 3770. Possible origins are:

- CICS/VS initialization, if the batch data interchange LU is defined in the TCT with CONNECT=AUTO specified.
- Creation of a CICS/VS task by the automatic transaction initiation (ATI) feature.
- Entry of a CSMT ACQ transaction by the CICS/VS master terminal operator to acquire the batch data interchange LU.
- Issue of a VARY LOGON command for the LU by the VTAM network operator.
- Issue of an appropriate command by the 3770 operator.

Once the session has been initiated, the 3770 can initiate data transfer provided that the operator has indicated, through the appropriate 3770 command, the data sets for transmission to the host. Either the data must contain a transaction identifier as the first four bytes, or a transaction identifier must be specified for the LU in the TCTTE.

Alternatively, the CICS/VS task can initiate data transfer.

Although CICS/VS allows the user to intersperse input requests with output requests, this leads to inefficient use of the batch protocols, which are designed for mass data transfer in one direction at a time. In practical terms, this means that batch transactions should be designed to read entire data sets before processing the data, and subsequently to send the output in a single operation.

Function Management Header (FMH) Handling

Provided that the CICS/VS application program communicating with the batch data interchange logical unit issues batch data interchange or BMS commands rather than terminal control commands, the batch data interchange program will construct the necessary outbound FMHs, strip off all inbound FMHs, and maintain the control blocks necessary for correct termination by CICS/VS if the transaction ends abnormally.

When a CICS/VS application program issues a batch data interchange command that specifies a new outbound destination, the batch data interchange program (DFHDIP) constructs and sends the appropriate Begin Destination FMH to select the destination. This FMH is preceded by an End Destination FMH for the currently selected destination, if there is one. CICS/VS application programs should normally be designed to terminate the currently-selected destination, by means of an ISSUE END command, before they themselves end. However, if an application program is terminated abnormally with an active selected destination outstanding, CICS/VS issues an Abort Destination FMH to terminate the selection.

If the CICS/VS application program uses BMS commands to send data to the batch data interchange LU, BMS uses the services of DFHDIP to construct the necessary selection FMHs, and, in general, destination selection and termination is handled as described in the preceding paragraph. The only exception occurs when the new destination (specified by its logical device code) is the console printer destination. In this case, the following action is taken:

1. A Suspend Destination FMH is issued for the currently-selected destination.

2. The output for the console printer destination is sent, without a selection FMH.

3. If, following the console output, the previous destination is reselected, a Resume Destination FMH is issued for that destination.

4. If, following the console output, a different destination is selected, a combination of Resume and End Destination FMHs is issued for the current destination, followed by a Begin Destination FMH for the new destination.

The inbound FMHs removed by the batch data interchange program are used to maintain a record of the destination currently selected. The CICS/VS application is advised of changes in the selected destination by means of return codes resulting from invocations of the batch data interchange program (see "Request Completion" later in this chapter).

In addition to the facilities provided by DFHDIP, there are some other 3770 data set operations that can be performed from the host processor using terminal control commands. If the CICS/VS application program issues terminal control commands, the program must contain its own logic for handling FMHs and abend processing.

Whereas the FMHs used with the batch logical unit (BCHLU) are of fixed length (six bytes), those used with the batch data interchange logical unit may be of either fixed length (six bytes) or variable length. Variable length FMHs contain a destination name in addition to the information contained in six-byte FMHs. The length of every FMH is always specified in its first byte.

Furthermore, the batch data interchange logical unit can accept two different types of FMH. The type 1 FMH used for selecting and deselecting named destinations can be followed by one or more type 2 FMHs that specify the operation to be performed on the data set.

For an introduction to FMH concepts, refer to the section "FMH Handling" in Chapter 4. Further details are also given later in this chapter.

The 3770 Transaction Data Set

The transaction data set contains a batch of transaction records collected at the 3770. Each record comprises those fields, processed during the execution of a 3770 program, that are designated for host processing. The 3770 program controls the format of the record by:

- Designating fields, in a fixed sequence, for host processing.
- Optionally specifying packing of the fields (by suppression of nulls).
- Optionally specifying use of a field delimiter character.
- Optionally using the END-CYCLE function to terminate a repeated sequence of fields (causing an IRS character to be inserted in the record).

There are two ways of initiating transmission of the transaction data set from the 3770 to the host. The 3770 operator can designate data sets for transmission to the host by the 3770 CODE 9 operator command. In this case, the data stream must include a transaction identifier.

Alternatively, a CICS/VS transaction can be attached for the batch data interchange LU by means of ATI and then issue the appropriate FMH through the ISSUE QUERY command. The name of the 3770 transaction data set is SYS.TDS.

The operator should not designate more than one data set at a time for transmission to CICS/VS, unless the transaction identified by the first record of the first data set is prepared to process all the data sets.

The form of transmission of the data set is not dependent on the method used to initiate transmission. Each record is transmitted as one SNA chain. Read requests must be issued to make the data available to a CICS/VS application program. The application program can have read requests issued on its behalf by means of either the RECEIVE MAP or the ISSUE RECEIVE command. Only one type of command (either batch data interchange or BMS) should be used during the processing of a data set. They should not be mixed. The amount of data returned for each request is determined by existing terminal control options. The correspondence of one chain to one transmit record makes the CICS/VS chain assembly option appropriate for the logical unit; if it is specified, logical record presentation can also be used for the transaction, which has the effect of splitting the record into sections at the points where END-CYCLE was used during its creation.

The application program can use a CICS/VS mapping service function for the data read. This is appropriate when a packing option is used that suppresses null characters, since the fields do not then appear at fixed offsets in the input record. When RECEIVE MAP is used to obtain data, mapping is performed during the input process. Each portion of the record delimited by END-CYCLE is mapped as one input line (since IRS is mapped as NL). It is appropriate for the 3770 program to terminate each field with HT so that the CICS/VS program may specify a BMS map with a tab map so that each field is positioned at a fixed offset during input mapping. The offsets in the map must be the same as the offsets in the tab map if HT is used in this way. If necessary, logical record presentation can be used so that each input "line" is processed with a different map. (Alternatively, at the macro level, DFHBIF TYPE=INFORMAT can be used to map the data after input by DFHDI TYPE=RECEIVE.)

The 3770 Interrupt Data Set

The BMS support for output to the non-programmable models of the 3770 can be used with the programmable 3770. The BMS output is directed to a particular component by means of the LDC operand. If the associated LDC value indicates line printer or card punch without a data set name, BMS will construct a six-byte function management header (FMH) to precede the output. This header will cause the 3770 to direct the output to the interrupt data set (SYS.INTR). If the associated LDC value indicates console ouput, no FMH will be appended to the data. In this case the 3770 will direct the output either directly to the console or to SYS.INTR, depending upon the setting of the 3770 DISK switch. If output is routed to SYS.INTR, the 3770 system data set index contains an indication of the destination selected (line printer, console, or punch).

The interrupt data set is also selected if a message is routed to the batch data interchange logical unit by means of the CMSG transaction and no LDC is specified in the ROUTE operand.

BMS output commands (SEND MAP ACCUM, SEND TEXT ACCUM, SEND TEXT, and ROUTE) can be used in conjunction with the LDC operand to build the

appropriate console, printer or punch output as required, in the same way as for the non-programmable models of 3770.

3770 User Data Sets

The batch data interchange program is a service program that assists a CICS/VS application program to communicate with the programmable models of the 3770. This assistance includes the maintenance of user data sets at the 3770, by use of the batch data interchange commands, which enable the CICS/VS application program to add and replace records in 3770 relative data sets. Typically, the CICS/VS application program will be initiated by ATI to perform these functions.

Additionally, entire user data sets can be queried (transmitted to CICS/VS).

User data sets can be placed on a diskette volume and referenced by means of a VOLID number.

User Data Set Record Manipulation

3770 data sets can be created empty or full. If created empty, records must be added before they can be replaced. If created full, records can only be replaced (even when priming the dataset initially).

Different transactions may add records to the same data set. If records that are added to a data set are subsequently to be replaced, the ADD request should be preceded by a NOTE request. This will cause the 3770 to return the number of the next available record, which will be assigned to the record that is being added. This number must be saved to permit subsequent reference to the record.

User Data Sets as Print Data Sets

User data sets can be allocated as named print data sets to which BMS printer output can be directed by means of an LDC defined with a line printer LDC value and the appropriate data set name. Such data sets can subsequently be printed at the 3770 under control of the operator and a user-written 3770 program. This facility permits the correct forms to be inserted before requesting a particular print data set by name. To take advantage of this facility, the logical unit should be in AUTOPAGE paging status.

Refer to the description of the DFHTCT TYPE=LDC macro in the CICS/VS System Programmer's Reference Manual for further details on specifying LDC values when allocating user data sets as print data sets.

User Data Set Control Functions

Data sets can be created, erased, or deleted by means of a CICS/VS application program only by using terminal control commands or macros. When using terminal control, the CICS/VS application program is fully responsible for selecting and deselecting the required data sets.

The function management headers used for the batch data interchange logical unit are described in the IBM 3773, 3774, and 3775 Programmable Communication Terminals Programmer's Guide. For control operations, these FMHs can be concatenated to specify fully the functions required.

When specifying a data set, the name of the data set can be qualified by specifying the name of the diskette volume on which the data set is located. In this case, the 3770 will select the specified volume first when searching for the data set.

The first operation is to select the required data set using a terminal control WRITE operation with the TIOA (or the FROM data-area) configured as in 1 or 2 in Figure 5-1. This selection remains in force until the destination is deselected by issuing a WRITE operation with the output data area configured as in 3 of Figure 5-1. Between selection and deselection, any number of control operations can be performed on the named data set. The 3770 will assume that all requests sent apply to the selected destination and will reject invalid requests. The console cannot be accessed while another outbound destination is still active. A new outbound destination (such as SYS.INTR) cannot be selected before the current outbound destination has been deselected. However, an active inbound destination (for example, SYS.TDS) can be interrupted for output to the console.

Datasets can be created, erased, or deleted by a CICS/VS application program using a DFHTC TYPE=WRITE with TIOA containing a single FMH as in 4 of Figure 5-1, or the equivalent command. This WRITE must immediately follow the dataset-select WRITE for the create operation.

	Required control function	Contents of TIOA for DFHTC TYPE=WRITE macro or of output area for SEND command
1	Select destination	FMH1:BDS,dataset-name.
2	Select destination on named volume	FMH1:FMHC,BDS,dataset-name;FMH2: VOLID.
3	End destination selection	FMH1:EDS,dataset-name.
4	Create, erase, or delete a data set	FMH2:CREATE/ERASE/DELETE
5	Store a program in SYS.PGM	FMH2:FMHC,STORE-PGM;FMH2:RECID; new program.
6	Execute a program in SYS.SUPR	FMH2:FMHC,EXECUTE-PGM;FMH2: RECID.

Key

FMH1 — A function management header type 1, which selects or deselects a data set. It contains the name of the required data set.

FMH2 — A function management header type 2, which specifies an operation on a data set or supplies information to supplement a preceding FMH. When it follows another FMH (with the FMHC bit set on), it must follow it immediately in the TIOA.

FMHC — Bit 0 of byte 1 of the FMH is set on to indicate that another FMH follows this one immediately in the TIOA.

BDS — Byte 4 of the FMH is configured to indicate the beginning of the data set.

EDS — Byte 4 of the FMH is configured to indicate the end of the data set.

VOLID — Volume name

RECID — Relative record number

CREATE
ERASE
DELETE — Data set operations
STORE-PGM
EXECUTE-PGM

Figure 5-1. TIOA and FMH Configurations for 3770 Data Set Operations

The required operands of the DFHTC macro for these operations are:

```
DFHTC TYPE=(WRITE[,WAIT])
      ,FMH=YES
      ,DEFRESP=YES
```

The equivalent command is:

```
SEND FROM(data-area) LENGTH(data-value)
     [WAIT]
     DEFRESP
     FMH
```

SYS.PGM and SYS.SUPR Operations

CICS/VS transactions can be written (using DFHTC TYPE=WRITE only) to store programs in SYS.PGM and subsequently request that they be executed. Similarly, transactions can request that procedures in SYS.SUPR be executed. Refer to the 3773, 3774, and 3775 Programmer's Guide for formats of the required function management headers and for details of program preparation required prior to transmission of the program to the 3770. The VSAM file BQILIBI built in the Program Validation Services preparation phase must be defined in the CICS/VS file control table.

As for other 3770 data sets, SYS.PGM and SYS.SUPR must be selected prior to sending control function FMHs, and deselected when all operations are complete. TIOA formats are shown in 1, 3, 5 and 6 of Figure 5-1.

Error Conditions in User Data Set Operations

Various error conditions may occur during the manipulation of records in the 3770 data sets. Errors are reported to CICS/VS by means of a negative response with system sense value X'1008' to indicate an invalid FMH. The associated user sense will indicate more precisely the cause of the error. The user sense codes received by DFHZNAC are to be found in fields TWAUR1 and TWAUR2. The system sense code mentioned above may be given in response to a chain that does not contain an FMH. In this case, it refers to the last FMH request that is still active, and means that the current chain has been associated with that request and has caused an error.

Error conditions arising from the use of the batch data interchange program are reported to the application program as described under "Request Completion" below.

Request Completion

The result of each invocation of DFHDIP is indicated by a one-byte category code and a one-byte return code. When an exception response is received, the sense information is saved and normal error recovery actions (by DFHZNAC) are modified to allow the user task to resume. When this resumption occurs, DFHDIP uses the sense information to generate the return code before returning control to the application program itself.

The category and return codes are available in EIBR, and have the meanings shown in Figure 5-2. The HANDLE CONDITION command may be used to test the category code.

Note: If the batch data interchange macro instruction DFHDI is being used, the category and return codes are available in the TCA; see the application programmers reference manual(macro level).

An exceptional case is when the DFHDIP invocation was through BMS: in this case, the DFHZNAC error recovery actions are not modified. This is to maintain compatibility with earlier programs.

Category Byte	Return Byte	Condition	HANDLE Condition
X'00'	X'00'	Successful completion	(Normal Response)
	X'01'	Begin destination FMH received	
	X'02'	Resume destination FMH received	
X'04'	X'11'	End destination FMH received	EODS
X'04'	X'12'	Suspend destination FMH received	DSSTAT
	X'13'	Abort destination FMH received	
X'08'	X'21'	Request invalid for data set organization	FUNCERR
	X'22'	Record too long	
	X'23'	Data set full	
	X'24'	Invalid keyword or record identifier	
	X'25'	Resource not available	
	X'26'	Invalid NUMREC option	
	X'28'	Insufficient resource	
	X'60'	Transient Data error during logging	
X'0C'	X'29'	Data set not found	SELERR
	X'41'	Destination does not exist	
	X'43'	Media not supported	
	X'44'	Invalid destination name	
	X'60'	Transient Data error during logging	
X'10'	X'F1'	Unexpected sense	UNEXPIN
	X'F2'	Unexpected FMH	
	X'F3'	Unexpected input	
X'E1'	X'00'	Retrieved data length to great	LENGERR

Figure 5-2. EIBR Return Codes for Batch Data Interchange

The user can control the response mode used for terminal control WRITE requests made by DFHDIP. Definite responses are requested by the use of the DEFRESP option of the batch data interchange commands or by specifying message integrity for the transaction; otherwise, only exception responses are requested. Exception response mode may allow an application to achieve a higher data rate than is possible with definite response mode, but it precludes synchronous error detection.

On completion of a DFHDIP request, the sense value from the latest exception response received is reflected in the return code. When the request was made in exception response mode and an error is detected by the 3770, this will be indicated by the return code to a subsequent request. If the same situation occurs in definite response mode, the

exception response is received and the return code set prior to completion of the request.

DFHDIP always requests a definite response to those FMHs that it builds for data management requests or destination selection. If previous requests were made in exception response mode, an SNA CHASE command is issued to force the processing of any outstanding exception responses before sending the FMH.

Data Integrity

As already described, a CICS/VS application program sending data to the 3770 can have assurance of the reception and processing of the data by the use of definite response mode.

The 3770 sends all chains in definite response mode.

Exception Responses

Exception responses are routed to the node abnormal condition program (DFHZNAC), which selects recovery actions and passes control to a user-written node error program (DFHZNEP). The user has the option, in his node error program, of overriding the recovery action selected by DFHZNAC in favour of some other action. In practice, such overriding action will seldom be necessary for errors arising in batch data interchange functions, because the default actions are designed to allow feedback of return codes to the application program.

Problem Determination

The batch data interchange program offers the normal CICS/VS problem determination facilities:

- Trace table entries
- Formatted dump

Chapter 6. The Type 4 Logical Unit

This chapter deals with communication between a CICS/VS application program and a type 4 logical unit (LUTYPE4). It describes CICS/VS support for logical unit type 4 protocols, with particular reference to the IBM 6670 Information Distributor.

System Programming for the Type 4 Logical Unit

Support for a type 4 logical unit in the terminal control program is generated by specifying ACCMETH=VTAM,VTAMDEV=LUTYPE4 in the DFHSG PROGRAM=TCP macro. Other operands of this macro which may be required are CHNASSY=YES and LOGREC=YES; see "Chain Assembly and Logical Record Presentation" later in this chapter.

If batch data interchange is to be used, the batch data interchange program must be generated (DFHSG PROGRAM=DIP). If basic mapping support is to be used, both the batch data interchange program and the basic mapping support program (DFHSG PROGRAM=BMS,BMSDEV=BCHLU) must be generated.

The terminal control table terminal entry (TCTTE) for an LUTYPE4 is generated by specifying TRMTYPE=LUTYPE4,ACCMETH=VTAM in the DFHTCT TYPE=TERMINAL macro.

In order to use the CICS/VS logical device code (LDC) facility for device selection, the system programmer must define a system LDC table containing the LDCs corresponding to the devices to be used, and identify the valid LDC names on each TCTTE defining an LUTYPE4. Alternatively, a local extended LDC list may be specified for a TCTTE.

Full details of all operands of the system programming macros are given in the CICS/VS System Programmer's Reference Manual.

Application Programming for the Type 4 Logical Unit

This section describes those aspects of CICS/VS application programming that are peculiar to a session between CICS/VS and an LUTYPE4. General facilities and services available to the CICS/VS application programmer are described in the CICS/VS Application Programmer's Reference Manuals.

CICS/VS support for LUTYPE4 is designed primarily to enable batch data to be transmitted between the logical unit and the CICS/VS application program. The support is an extended version of that provided for the 3770 Batch Logical Unit (a logical unit type 1). A description of LUTYPE1/LUTYPE4 compatibility is included later in this chapter.

A Type 4 logical unit provides access to a number of discrete destinations, or media, within the physical device that the logical unit represents. The following media are defined:

- Data Processing Media
  - console

- print
- card

- Word Processing Media
  - WPMEDIA1
  - WPMEDIA2
  - WPMEDIA3
  - WPMEDIA4 (defaulted to WPMEDIA1 by IBM 6670)

The word processing destinations represent destinations that are defined by the LUTYPE 4 implementation.

Each of data processing and word processing media may be further qualified by a one-byte subaddress (0 through 15). The subaddresses that may be used are determined by the specific LUTYPE4 implementation. Subaddress 15 is defined by SNA to mean any medium of the specified type.

Media selection for an LUTYPE4 is effected by means of Type 1 Function Management Headers (FMH). The format of an FMH Type 1 is shown in Figure 6-1. The CICS/VS application programmer's involvement with FMHs depends upon the application programming interface that is being used. Further information is given under the headings "Application Program Interfaces" and "End-of-Data-Set Notification" later in this chapter.

For messages outbound from CICS/VS, the first request unit (RU) of the message carries a "begin destination" function management header (FMH) which selects the medium for which the message is intended. This selection remains in effect until it is deselected by means of an "end destination" or an "abort destination" FMH.

As an exception to this selection mechanism, messages that do not have a begin destination FMH are routed to the default destination "console" (a particular printer output hopper for the IBM 6670). Note, however, that the IBM 6670 does not support "suspend destination" or "resume destination" FMHs, so that messages may not be sent to the console while another destination is currently selected. If "console" is selected during transmission of data to another medium, CICS/VS transmits an "end-destination" FMH to deselect the currently-selected medium.

For messages inbound to CICS/VS, the first RU carries a "begin destination" FMH which indicates the source of the inbound message. Similarly, the first RU of the final chain of the message carries an "end destination" FMH.

Messages inbound from an IBM 6670 are transmitted as a single chain; the first RU of this chain therefore carries a combined begin destination and end destination FMH. LUTYPE4 protocols require all FMHs other than combined begin-destination/end-destination FMHs to be transmitted stand-alone, that is, without accompanying user data in the same chain.

Byte	Bits	Meaning
0	0–7	FMH length; coded as X'06' or X'09' (note 3)
1	0–7	FMH type (1); coded as X'01'.
2	0–7	Media code; coded as follows: (note 1) X'0x' – console printer X'2x' – card X'3x' – print X'8x' – WP media 1 X'9x' – WP media 2 X'Ax' – WP media 3 X'Cx' – WP media 4
3	0–3	Reserved; coded as X'0'.
	4–7	Data stream profile; coded as X'0'. (note 2)
4	0–7	Data set control; coded as follows: X'00' – Resume destination (note 2) X'20' – End destination X'40' – Begin destination X'60' – Chain contains begin and end destination X'80' – Suspend destination (note 2) X'A0' – Abort destination
5	0–7	Exchange record length; coded as X'00'.
6	0–7	Reserved, coded as X'00' (note 3)
7	0–7	Reserved, coded as X'00' (note 3)
8	0–7	Length of destination name, coded as X'00' (note 3)

Note 1: x is a 4–bit subaddress coded as X'0' through X'F'.
Subaddress X'F' means any available subaddress.

Note 2: Not supported by IBM 6670.

Note 3: CICS/VS transmits a 9–byte FMH to an LUTYPE4.
The IBM 6670 transmits a 6–byte FMH.

Figure 6–1. Function Management Header (FMH) Format

APPLICATION PROGAM INTERFACES

This section of this chapter discusses the application programming interfaces available for LUTYPE4. It is followed by descriptions of those aspects of application programming which must be considered whichever interface is used; namely:

- Bracket protocol
- Chain assembly and logical record presentation
- End–of–data–set notification
- Inbound SIGNAL commands

- Inbound LUSTAT commands
- CICS/VS Transaction Design

Application programs designed to communicate with an LUTYPE4 may be written using any of the following interfaces.

- Batch Data Interchange commands or macro instructions.
- Basic Mapping Support commands or macro instructions.
- Terminal Control commands or macro instructions.

In all cases, the use of macro instructions is allowed only in assembler programs.

Full details of these interfaces are given in the CICS/VS Application Programmer's Reference Manual (Command Level), the CICS/VS Application Programmer's Reference Manual (Macro Level), and the CICS/VS Application Programmer's Reference Manual (RPGII).

Batch Data Interchange

The batch data interchange interface may be used for all CICS/VS application programs that communicate with an LUTYPE4, and is particularly appropriate for programs that use the word processing media. Using this interface, the application programmer does not have to concern himself with FMHs, nor does the system programmer have to build LDC lists.

Application programs may use the following batch data interchange commands, or the equivalent macro instructions, to communicate with an LUTYPE4:

```
ISSUE SEND
ISSUE RECEIVE
ISSUE END
ISSUE ABORT
ISSUE WAIT
```

Media selection is performed by means of options on the ISSUE SEND command. The same medium and subaddress must be specified on every ISSUE SEND command, not just the first, and on all subsequent batch data interchange commands for the selected medium until a new selection is made. If the medium and subaddress are not specified, CICS/VS will assume that the message is for the default console, and will deselect any currently selected destination.

Basic Mapping Support

The BMS interface for LUTYPE4 provides output mapping support for messages to the console, print, and card media of the IBM 6670. Output mapping support is also provided for the word processing media. BMS, however, does not use any of the additional control characters available for word processing text; if these are required, the CICS/VS application must format the output data stream and use the terminal control or batch data interchange interface.

BMS input mapping is provided for data streams from LUTYPE4 data processing media. If BMS input mapping is required, INBFMH=DIP should be coded on the DFHPCT TYPE=ENTRY macro for the transaction. This operand allows the batch data interchange program to handle the inbound FMHs and to specify to BMS the LUTYPE4 medium from which the input originated. If BMS input requests are used to read data from word processing media, no input mapping is performed, and the input is passed unchanged to the application program.

If INBFMH=DIP is not specified, attempts to map data from word processing media will have unpredictable results. Input data from data processing media containing NL or IRS control characters will, however, be mapped correctly.

When constructing outbound data streams for an LUTYPE4, BMS uses the services of the batch data interchange program to build the necessary FMHs. The application programmer has only to supply a logical device code to inform BMS which medium the output is intended for.

BMS formats the output for an LUTYPE4 by means of new line (NL) or inter-record separator (IRS) control characters. In addition, form feed (FF), vertical tab (VT), and horizontal tab (HT) characters may be used, depending upon the output medium and the options specified in the TCTTE for the logical unit. The control characters that BMS may use for the various output media are:

Console — NL, FF, HT, and VT

Card — IRS only

Printer — NL, FF, HT, and VT

All WP media — NL, FF, and HT

The use of FF, VT, and HT control characters must be enabled by specifying FF=YES, VF=YES, and HF=YES respectively on the DFHTCT TYPE=TERMINAL macro for the logical unit.

Horizontal and vertical tab settings are defined in the HTAB and VTAB operands of the map set definition macro DFHMSD. If they are used, the user must ensure that the physical tab settings at the IBM 6670 match those defined for the map set.

Terminal Control

If the terminal control interface is used for communication with an LUTYPE4, the application progammer is responsible for building the FMHs required to select and deselect the required destinations. The format of the FMH is shown in figure 6-1. FMHs, other than combined begin-destination/end-destination FMHs, must be transmitted without accompanying data, and the presence of an FMH must be indicated to CICS/VS by means of the FMH option of the SEND command. The application programmer is also responsible for formatting the output, using the correct control characters for the medium.

The disposition of inbound FMHs is determined by the INBFMH operand of the DFHPCT TYPE=ENTRY macro for the transaction. If all FMHs are passed to the application, the INBFMH condition is raised whenever an inbound FMH is detected. Note, however, that the EODS (end-of-data-set) condition overrides the INBFMH condition (see "End of Data Set Notification" later in this chapter).

BRACKET PROTOCOL

Bracket protocol is used when CICS/VS communicates with a type 4 logical unit. Note that the bind for an LUTYPE4 (Appendix D) specifies that the session is to be brought up in "in-bracket" state. After receiving a positive response to the bind, CICS/VS transmits a null RU carrying and end-bracket indicator to set the session to "between-brackets" state.

For the most part the use of brackets is transparent to the CICS/VS application program. However, if the application is using terminal control or BMS commands to send data to the logical unit, the LAST option may be specified on the last output request. The last output request is defined as either the last output command for a transaction not using chain control, or as the output operation that transmits the first or only RU of the last output chain for a transaction employing output chain control. For further information, refer to the description in the CICS/VS System Programmer's Reference Manual of the CCONTRL parameter of the DFHPCT TYPE=OPTGRP macro.

The LAST specification causes CICS/VS to transmit an end-bracket indicator with the final output message to the logical unit. This indicator notifies the batch logical unit that the current transaction is ending. If the LAST operand is not specified, CICS/VS waits until the task detaches before sending the end-bracket indicator, which is transmitted with a null RU. Since an end-bracket indicator is transmitted only with the first RU of the chain, the LAST operand is ignored for a transaction using chain control unless the request is the first or only one in the chain.

CHAIN ASSEMBLY AND LOGICAL RECORD PRESENTATION

CICS/VS allows inbound data to be presented to the application program in the form of single request units (RUs), complete chains, or logical records. Similarly, the chaining of outbound data may either be handled automatically by CICS/VS or be controlled by the application program.

For inbound data, chain assembly is specified by means of the CHNASSY operand of the DFHTCT TYPE=TERMINAL macro for the logical unit, and therefore applies to all applications which run with the logical unit. CHNASSY=YES specifies that CICS/VS is to assemble a complete chain before any further processing is carried out; CHNASSY=NO specifies that each RU is to be processed individually.

Logical record presentation enables inbound data to be presented to the application program in the form of logical records, rather than single RUs or complete chains, irrespective of whether or not chain assembly is specified. Logical record presentation is specified for the transaction by means of the LOGREC operand of the DFHPCT TYPE=ENTRY macro for the transaction.

Logical records are delimited by new line (NL), inter-record separator (IRS), or transparent (TRN) characters. If inbound chain assembly is not being used, the end of an RU also delimits a logical record. Because the IBM 6670 may transmit logical records that span RUs, chain assembly should be specified whenever logical record presentation is specified.

Note that documents from an IBM 6670 are transmitted as single chains, and are potentially very large. If the application is designed to process complete chains, the use of the SET option on the input

commands may be appropriate to reduce the storage requirements of the application itself.

Also, it is possible for the length of the inbound chain to exceed the maximum TIOA size specified in the TCTTE. The default action taken by CICS/VS for this condition (error code X'45') is to abort the VTAM receive command and to abend the CICS/VS transaction. To avoid this action, the user should write a node error program (NEP) for the condition which sets action flags TWAOPT1 to X'00', to avoid printing the TCTTE; and sets TWAOPT2 to X'40' so that the VTAM receive command is still aborted but the transaction is allowed to resume. When the transaction resumes, the first part of the chain is available in the TIOA. The next VTAM receive command that CICS/VS issues, as a result of an application program issuing a RECEIVE command, will recover the rest of the chain, or the next section of it if the remaining data is still too long.

Outbound chaining is normally handled by CICS/VS. Under these circumstances each output command results in the transmission of a complete chain consisting of one or more RUs, depending upon the length of the output data and the maximum RU size.

Alternatively, applications that use the terminal control interfaces can be designed to control their own outbound chaining. This is done in the following way:

1. The transaction must be identified as one allowed to control outbound chaining, by specifying MSGOPT=CCONTRL and MSGPREQ=CCONTRL in the DFHPCT TYPE=OPTGRP macro.

2. Each SEND command, except the command that completes the chain, must have the CNOTCOMPL option specified. The omission of this option indicates that the chain is complete.

END-OF-DATA-SET NOTIFICATION

The end of a data set transmission from an LUTYPE4 is indicated by an "end-destination" FMH carried in the first RU of the final chain.

Because the IBM 6670 transmits data sets as single chains containing a combined begin- and end-destination FMH, the end-destination indication is received on the first read operation. CICS/VS therefore delays the raising of the EODS condition until the whole chain has been received.

The EODS condition is raised on the first read operation that is attempted after the whole of the data set has been passed to the application, irrespective of whether chain assembly or logical record presentation is being used.

When all data sets have been transmitted, a further read will raise the DSSTAT condition "currently no data to send", signifying the end of a batch (see "INBOUND LUSTAT COMMANDS" later in this chapter).

INBOUND SIGNAL COMMANDS

The SIGNAL command provides a means for an LUTYPE4 that is currently in receive mode to indicate to CICS/VS that it wishes to send data.

The inbound SIGNAL command is always detected by the node abnormal condition program. The action flags for this condition (error code X'66') specify no action; the system programmer may, however, write a node error program (NEP) to handle the inbound SIGNAL command if required.

After NEP processing, CICS/VS raises the SIGNAL condition for the application program when it resumes. The application program can specify the address of a routine to handle the inbound SIGNAL command by means of the HANDLE CONDITION SIGNAL command. In this routine, the contents of the 4-byte signal code can be examined by means of the ASSIGN SIGDATA command. The SIGNAL codes have the following meanings:

X'00010000' - hard request change direction

X'00010001' - soft request change direction

The "hard" change direction request is a mandatory request. CICS/VS enforces this protocol for LUTYPE4 (but not for other logical unit types) by raising the IGREQCD condition if the application attempts an output operation after the request has been received. Note that all change direction requests from an IBM 6670 are "hard" requests.

The IBM 6670 does not send a "change direction" request until it receives an end-of-data set indicator. This provides a measure of synchronization, ensuring that the direction change occurs between jobs.

The IGREQCD condition can be handled by means of a HANDLE CONDITION IGREQCD command. The attempted output operation, which caused IGREQCD to be raised, will not have been performed. The application program should either issue a receive command or terminate.

INBOUND LUSTAT COMMANDS

The LOGICAL UNIT STATUS (LUSTAT) command is used by a logical unit to send status information to its session partner. Details of LUSTAT codes are given in Systems Network Architecture Reference Summary, GA27-3136.

The following inbound LUSTAT codes from an LUTYPE4 are particularly relevant to the CICS/VS application programmer:

X'0003' - entering attended mode (not sent by IBM 6670)

X'0004' - entering unattended mode (not sent by IBM 6670)

X'0007' - currently no data to send

CICS/VS keeps a record in the terminal control table of whether the logical unit is in attended or unattended mode. At session initialization, unattended mode is assumed. Thereafter, the mode is controlled by receipt of LUSTAT (X'0003') and LUSTAT (X'0004') commands from the logical unit. The application programmer can test the attended/unattended mode of the logical unit by means of the ASSIGN UNATTEND command.

The IBM 6670 does not send LUSTATUS (X'0003') or (X'0004'); it is therefore always in "unattended" mode.

The IBM 6670 transmits LUSTAT(X'0007') to indicate that it currently has no data to send. The inbound LUSTAT command is detected by the node abnormal condition program (DFHZNAC), which drives the node error program exit, and then takes the action specified by the action flags. The default action depends on whether or not there is an active task associated with the terminal.

1. If a CICS/VS task is attached at the time that the LUSTAT command is received, the action flags are all zero, signifying a no-operation. Control is returned to the application, and, if the application had a batch data interchange input command oustanding, the DSSTAT condition is raised.

2. If a CICS/VS task is not attached at the time that the LUSTAT command is received, the following action is taken:

   a. If a write operation is outstanding (because, for example, the transaction has issued a write command and then terminated), it is completed.

   b. If a transaction associated with the terminal is awaiting activation (by automatic transaction initiation, for example), it is activated.

   c. Otherwise, if the terminal is "unattended", the session is terminated by an UNBIND command. Note that session termination may be avoided by altering the appropriate action flag in a node error program.

CICS/VS TRANSACTION DESIGN FOR THE IBM 6670

To make efficient use of the CICS/VS-6670 session, the design of CICS/VS transactions must take into account the expected method of operation of the 6670. The session is controlled by two basic implementation-defined aspects of the 6670:

1. The 6670 transmits LUSTAT(X'0007') when it is in send state and currently has no data to send. The sending of LUSTAT is delayed to give the operator time to insert further cards into the magnetic card reader. Sending LUSTAT (X'0007') does not prevent transmission of further data if the operator inserts more cards.

2. The 6670 transmits SIGNAL (hard request change direction) when it is in receive state and the operator inserts a "send" job into the card reader. However, if an outbound destination selection is current, SIGNAL will be deferred until EODS is received.

Two basic categories of CICS/VS transaction, each corresponding to a particular method of 6670 operation, can be defined:

1. Long-running CICS/VS transactions designed to read multiple batches of data (documents) from the 6670. This form of transaction requires the 6670 operator to enter a send job that transmits a single CICS/VS transaction identifier followed by one or more complete documents.

   The transaction can be designed to read a document, carry out any required EODS processing when the EODS condition is raised, read the next document, and so on. The 6670 indicates that the last document has been transmitted by sending LUSTAT(X'0007') (see "Inbound LUSTAT Commands" earlier in this chapter).

   Such a transaction could also be designed to start a separate output transaction to transmit documents to the 6670 after LUSTAT(X'0007') has been received (DSSTAT condition "currently no data to send").

2. Transactions designed to receive a single document from, or transmit requested data to, a 6670. This form of transaction requires the 6670 operator to insert a discrete "send" job, carrying a CICS/VS transaction identifier, into the card reader, and to wait for a console message from the transaction before inserting the next job.

   The transaction, after performing its required function and before terminating, should transmit a message to the 6670 console to indicate that next job may be inserted. It should also be designed to handle SIGNAL (hard request change direction) commands from the 6670, in case the 6670 operator violates the operational protocol by inserting another job when the 6670 is in receive state.

By following the operational procedures described above, the frequency of automatic transmission of SIGNAL by the IBM 6670, and the associated overhead, can be kept to a minimum.

LUTYPE1/LUTYPE4 COMPATIBILITY

Because the support provided by CICS/VS for a type 4 logical unit is similar to that provided for the 3770 batch logical unit (an LU type 1), some existing applications designed to operate with the console, card, or printer components of a 3770 batch LU may also be suitable for operation with an LUTYPE4. However, the following potential incompatibilities exist:

1. Records spanning RUs

   For an LUTYPE4, logical records may span RUs. LU type 1 applications that read card input, and are designed to run without chain assembly, may therefore not receive input data in the form expected in the following cases:

   a. When the application uses logical record presentation

   b. When BMS input mapping is used.

   In both cases, the specification of chain assembly in the TCTTE will enable the application to run as intended. However, the specification of chain assembly for the logical unit may require applications to be modified.

2. Inbound SIGNAL command

   Existing applications will not contain a HANDLE CONDITION IGREQCD command. If such an application attempts an output operation after a "hard" change direction request has been received, it will be abnormally terminated.

3. Console messages

   If a current application selects CONSOLE during transmission of a data set to an LUTYPE1 destination, the destination will be suspended, but will be resumed for a further send. If CONSOLE is selected during transmission of a data set to an LUTYPE4 destination, however, the destination will be deselected. A further send to the same destination will cause it to be reselected, but the terminal will see the two parts of the transmitted document as separate documents.

Appendix A. Sample Program for the Full Function LU

The sample program supplied with CICS/VS for the full function LU is contained in file DFHSPPCO in the CICS/VS source library. It is a 3770 program designed to serve the following purposes:

1. It provides an example of how 3770 host communication statements are used for communication with CICS/VS.

2. It provides the basic logic needed to exchange data between a 3770 terminal and a CICS/VS application program.

3. It provides the 3770 terminal operator access to certain fundamental CICS/VS services, namely

   - Sign-on and sign-off
   - Supervisory terminal services
   - Master terminal services
   - The CSFE transaction.

The sample program has several modes of operation. Three of these modes are used by the 3770 terminal operator to direct the execution of the program. These modes are:

1. Local mode, in which the operator may request host communication or local functions. Local mode corresponds to the SNA out-of-session state.

2. Communication mode, in which the operator may begin a transaction with CICS/VS or request user-defined functions. Communication mode corresponds to SNA in-session and between-brackets states.

3. Transaction mode, in which the operator may send and receive messages to and from a CICS/VS transaction. Transaction mode corresponds to SNA in-session and within-brackets states.

4.

   In addition, the sample program has three internal modes of operation that are required to maintain orderly communication with CICS/VS. These modes are:

5. Read mode, which is entered to receive output from CICS/VS. In this mode, SNA commands and indicators as well as user data are handled.

6. State test mode, which is entered from read mode to test and take action on any SNA commands or indicators received from CICS/VS.

7. Error recovery mode, which is entered whenever an error code is issued on the completion of an HCF statement.

The operation of the sample program in these six modes is described below.

If the 3770 sample program is used with a CICS/VS application program containing DFHBMS macros, the PGESIZE operand of the DFHTCT TYPE=LDC macro must specify a column value of 79.

Local Mode

The operator is prompted by message CSP001 for a line of input. The program scans the input for the following commands:

1. QUIT or Q

   The program issues message CSP020 and terminates.

2. HOST or H

   The program issues message CSP010 to determine the host system name, and CSP012 to determine the required LU number.

   For error recovery purposes, the program must also determine how message sequence numbers are to be handled. The program issues message CSP014 to allow the operator to choose one of the following commands:

   a. COLD or C

      Sequence numbers will be numbered from zero.

   b. WARM or W

      Sequence numbers will continue numbering from values recorded during a previous session.

   c. NONE or N

      Sequence numbers will not be recorded.

   Note: When sequence numbers are to be recorded, a predefined 3770 data set (with record length 4) is required for each logical unit.

   The program then issues message CSP016 to determine whether the operator requires a CICS/VS-initiated session or a 3770-initiated session. The operator may reply with either of the following commands:

   a. WAIT or W

      The program issues an OPNSESS TYPE=(ACCEPT,WAIT) statement and waits for CICS/VS to initiate a session.

   b. SELF or S

      The program prompts the operator, using message CSP010, for the name by which CICS/VS is known to the access method. The operator is also prompted, using message CSP012, for the LU number of his terminal. The program issues an OPNSESS TYPE=ACQUIRE statement to obtain the session.

   The program enters communication mode.

3. For any other input, the program issues message CSP030 to indicate invalid input and allows a retry.

Communication Mode

On entering communication mode, the program tests if any output has been received from CICS/VS. If so, the program enters read mode. Otherwise, the program issues message CSP101 to prompt the operator to enter input. The program scans the input for the following commands:

1. LOCAL or L

   The program terminates the session and reverts to local mode.

2. Any user-defined commands

   The sample program may be modified to include user-defined commands and functions.

3. Other

   The program assumes that any undefined input is a message for CICS/VS requesting a transaction initiation. The program first tests whether a current session has been quiesced. If so, message CSP502 is issued to the operator. Otherwise, the message is sent to CICS/VS with the BB indicator set. If the data entered did not fill a line, the BC, EC, and CD indicators are also set. If the data did fill a line, the operator may continue to enter data until a short line is entered. Thus a short line denotes either a complete message or the final part of a message. After sending the message to CICS/VS, the program enters transaction mode.

Transaction Mode

In transaction mode, the program manages message flow between the CICS/VS transaction and the terminal. The program maintains the half-duplex flip-flop protocol between CICS/VS and the full function LU.

When in send state, the program reads data from the terminal and sends it to CICS/VS. Each full line of input is treated as an RU and is sent with the appropriate SNA indicators. A short line is either the last RU in a chain or a single-element chain and is sent with the CD indicator.

When in receive state, the program enters read mode to receive output from CICS/VS. Data is presented to the terminal until either a CD or EB indicator is received.

Read Mode

When in read mode, the program receives RUs from CICS/VS. The program determines the nature of the RU by inspecting the SNA indicators. The RU will be one of the following types:

1. Data Request

   The RU contains data for the terminal. If the SNA indicators specify that an FMH is included, the program sets a flag (SSENS) to indicate that a negative response is required. (The sample program does not handle function management headers.) If neither of these conditions occur, the data is presented to the terminal.

   If an RU is received containing a CANCEL command, canceling elements of a chain that have already been presented to the terminal, the sample program issues message CSP450, informing the 3770 that the message has been canceled by the host. No further elements of the canceled chain are transmitted.

2. Positive Response

   The RU is a reference to data or an SNA command sent by the program.

3. Negative Response

   The RU is a response denoting that either CICS/VS could not initiate a transaction or the CICS/VS transaction has abended. The reason for the failure (a CICS/VS error message number) is issued to the terminal, using messages CSP400 and CSP401.

4. Command Request

   The RU contains an SNA command. The commands supported by the program are CANCEL, QEC, RELQ, SHUTD, BID, LUSTAT, or SIGNAL. For any other command, the program sets flag (SSENS) to indicate an error. The actions for the supported commands are as follows:

CANCEL	issue message CSP450 and send a positive response.
QEC	issue message CSP501 and send a positive response.
RELQ	issue message CSP452 and send a positive response.
SHUTD	send a positive response.
BID	set the flag SSENS to indicate bid reject (the RTR command will be sent when appropriate) and send a positive response.
LUSTAT	issue message CSP400.
SIGNAL	if the system sense bytes are X'0100' (request change direction), send a positive response; otherwise, treat as an unexpected command and send a negative response.

In all the above cases, the next action of the program is to enter state test mode.

State Test Mode

In state test mode, the program tests the state machine indicators maintained by the 3770 to maintain the SNA protocols required for the full function LU. The program tests for each of the conditions described below, in the order given.

1. Within-brackets state

   The program enters transaction mode.

2. Quiesce-pending state

   The program sends the QC command to CICS/VS and continues with the next test.

3. Ready-to-receive pending state

   The program issues message CSP503, which invites the operator to decline or accept an outstanding bid request from CICS/VS. If the bid is accepted, the program sends the RTR command to CICS/VS and enters read mode. If the bid is rejected, the program continues with the next state test.

4. Shutdown-complete pending state

   The program issues message CSP504, executes the CLOSESS statement after sending the SHUTC command to CICS/VS, and enters local mode.

If none of the state tests apply, the program enters communication mode.

Error Recovery Mode

Error recovery mode is entered whenever the 3770 completes an HCF statement with an error code. The program issues message CSP600, which specifies the error code number. The following list describes the action taken by the program for each error code:

No session resources available, code 002
OPNSESS failure, code 003
Session terminated, code 102

For the above three errors, the program issues CLOSESS statement and enters local mode.

Protocol violation, code 101

Sends a negative response and enters state test mode.

Session resynchronization required, code 103

Obtains sequence numbers from the 3770 data set and issues WRTCTL RESYNC=YES statement if the operator has requested sequence number logging. Otherwise, issues the WRTCTL RESYNC=NO statement. Enters state test mode.

Sequence numbers do not match, code 104

Issues the RDSTUS TYPE=RCVRY to obtain CICS/VS sequence numbers. Compares the CICS/VS and 3770 sequence numbers and issues one of the following messages:

CSP605 Both inbound and outbound numbers do not match. Enters state test mode.

CSP606 Inbound (3770 to CICS/VS) numbers do not match. Enters communication mode.

CSP607 Outbound (CICS/VS to 3770) numbers do not match. Enters read mode.

No sequence numbers from host, code 105

Enters state test mode.

Recovery required, code 106

Issues message CSP602 and then issues the WRTCTL RESYNC=REQ statement.

Recovery failed, code 107

Issues message CSP603 and then issues the CLOSESS statement. Enters local mode.

Messages Issued by the Sample Program

CSP001 - LOCAL MODE - ENTER FUNCTION REQUEST :

CSP005 - INVALID OPNSESS TYPE, REENTER :

CSP010 - ENTER HOST SYSTEM NAME :

CSP012 - ENTER LU NUMBER :

CSP014 - ENTER ''COLD'' (C) ''WARM'' (W) or ''NONE'' (N):

CSP016 - ENTER ''WAIT'' (W) or ''SELF'' (S):

CSP018 - HOST INITIATED SESSION NOW STARTED

CSP020 - PROGRAM TERMINATED

CSP030 - UNKNOWN FUNCTION, REENTER :

CSP101 - HOST COMMUNICATE MODE - ENTER TRANSACTION

CSP400 - CICS SENSE CONDITION

CSP401 - USER SENSE MESSAGE :

CSP450 - MESSAGE CANCELED BY HOST

CSP452 - QUIESCE RELEASED, LU IN SERVICE

CSP501 - LU QUIESCED, OUT OF SERVICE

CSP502 - LU STILL QUIESCED, OUT OF SERVICE

CSP503 - HOST OUTPUT PENDING, ENTER A TO ACCEPT :

CSP504 - LU SHUTDOWN

CSP600 - HOST COMMUNICATION ERROR NO.
(NO SESSION RESOURCES AVAILABLE)
(OPNSESS FAILURE)
(PROTOCOL VIOLATION - EXCEPTION RESPONSE SENT)
(HOST SESSION TERMINATED)
(RESYNCHRONIZATION REQUIRED)

CSP601 - CICS/VS DID NOT RESYNCHRONIZE

CSP602 - RECOVERY REQUIRED

CSP603 - RECOVERY FAILED

CSP604 - SEQUENCE NUMBER MISMATCH, ERROR IGNORED

CSP605 - RETRY LAST TRANSACTION OR CONTINUE

CSP606 - FURTHER CICS/VS OUTPUT IS EXPECTED

Appendix B. Sample CICS/VS Program for the Batch LU

A sample CICS/VS application program for communication with a 3770 batch logical unit is provided with the CICS/VS distribution tape. The file name of the sample program is DFHSAMP. The program consists of three steps — two processing steps followed by a backout step. At the beginning and end of each step, a message is sent to the 3770 keyboard—printer. Detailed information on the processing of the sample is contained in its descriptive header.

Processing Step 1

A card data set is read from the batch logical unit and transmitted to the application program as logical records. Each card is used to create a new record on an existing VSAM file. All cards are saved on temporary storage for subsequent printing. When the end of the data set is reached, the records are transmitted from temporary storage to the printer. The printer represents another data set associated with the same batch logical unit.

Processing Step 2

Another card data set is read from the batch logical unit and transmitted to the application program, again as logical records. The cards are used to update the records on the VSAM file created in processing step 1. Some cards will have errors and in these cases a "record not found" condition will occur, resulting in an interrupting message to the console. Processed cards will be saved on temporary storage and printed out when the end of the data set is reached.

Backout Step

The VSAM file is restored to its original condition (that is, its condition before records were created in processing step 1).

Appendix C. System Sense Codes Sent by CICS/VS

This appendix lists the system sense codes and their meanings that CICS/VS may send to a full function LU when CICS/VS detects an abnormal condition. For each system sense code, the SNA definition of the code is followed by the reason why CICS/VS sends the code. For further information on CICS/VS error recovery actions, refer to the CICS/VS System Programmer's Guide.

System sense code	Reason
X'0809'	Request reject – Mode inconsistency. Data has been received from an LU defined as permanently out-of-service or in receive-only state.
X'080B'	Request reject – Bracket race error. CICS/VS received a begin-bracket prior to receiving the response to end-bracket.
x'080F'	Request reject – End user not authorized. The security code of the requested transaction did not match that of the operator signed on to the LU.
X'0812'	Request reject – Insufficient resource. The transaction has been purged because of a resource constraint or interlock.
X'0819'	Request reject – RTR not required. The condition that caused a previous BID command to be sent is no longer valid.
X'081C'	Request reject – Request not executable. 1. The PCT entry for the requested transaction was was disabled. 2. The PPT entry associated with the PCT entry was disabled. 3. An RU requesting transaction initiation was too long and was truncated by the access method.
X'081F'	Request reject – Reserved (defined by CICS/VS). CICS/VS received an SNA command that it does not support.
X'0824'	Request reject – Component aborted. A transaction abnormal termination has been caused by a program check, by a stall purge condition, or by a user request.
X'1002'	Request error – RU length error. An RU has been received that exceeds the maximum size established by the RAMAX operand of the DFHTCT TYPE=INITIAL macro instruction.
X'1003'	Request error – Function not supported. CICS/VS could not identify the requested transaction.
X'1008'	Request error – Invalid FMH. The length byte in an FMH was incorrect.

System sense code	Reason
X'2003'	State error - Bracket (catastrophic). An RU contained in an end-bracket but not a begin-bracket. CICS/VS does not permit 3770 LUs to terminate a bracket.
X'400A'	RH usage error - No-response not allowed. An RU requested SNA no-response protocol, which is not supported.

Appendix D. Bind Formats

The bind image that accompanies the BIND command transmitted by VTAM to a 3767, a 3770, or a 6670 is supplied by CICS/VS in a bind area addressed from the node intialization block (NIB).

This appendix lists the bind image formats for the Full Function logical unit (Figure D-1), the Interactive (Flip/Flop) logical unit (Figure D-2), the Interactive (Contention) logical unit (Figure D-3), the Batch logical unit (Figure D-4), the Batch Data Interchange logical unit (Figure D-5), and the Type 4 logical unit (Figure D-6).

Byte	Value	Meaning
0	X'31'	BIND Request Code
1		
	0000....	Bind Format 0
	0001	Bind Type 1 (cold)
2	X'04'	FM Profile 4 (LU-LU)
3	X'04'	TS Profile 4 (LU-LU)
4		Primary LU Protocol
	1.......	Multiple RU Chains
	.0......	Immediate Request Mode
	..11....	Definite/Exception Response
	....00..	
	......0.	No Compression
	1	Primary may send EB
5		Secondary LU Protocol
	1.......	Multiple RU Chains
	.0......	Immediate Request Mode
	..11....	Definite/Exception Response
	....00..	
	......0.	No Compression
	0	Secondary may not send EB
6		Common Protocol
	0.......	
	.1......	FMH Allowed
	..1.....	Bracket Protocol Used
	...1....	Bracket Termination Rule 1
	....0...	Alternate Code not Allowed
	000	

Figure D-1 (Part 1 of 2). Bind Format for Full Function LU

Byte	Value	Meaning
7		Common Protocol
	10.....	Half Duplex Flip/Flop
	..0.....	PLU has Recovery Responsibility
	...0....	SLU is Contention Winner
	....000.	
	0	
8	00.....	
	..xxxxxx	SLU Send Pacing Count (Note 1)
9	00.....	
	..xxxxxx	SLU Receive Pacing Count (Note 1)
10		SLU to PLU RU Size (Note 2)
	xxxx....	Mantissa
	xxxx	Exponent
11		PLU to SLU RU Size (Note 2)
	xxxx....	Mantissa
	xxxx	Exponent
12	00.....	
	..xxxxxx	PLU CPMGR Send Pacing Count (Note 1)
13	00.....	
	..xxxxxx	PLU CPMER Send Pacing Count (Note 1)
14	0.....	
	.0000000	LU Type 0
15-	XL22'00'	(Remainder of Bind Area

Notes:
1. Supplied by VTAM
2. As specified in DFHTCT Macro

Figure D—1 (Part 2 of 2). Bind Format for Full Function LU

Byte	Value	Meaning
0	X'31'	BIND Request Code
1		
	0000....	Bind Format 0
	0001	Bind Type 1 (cold)
2	X'03'	FM Profile 3 (LU–LU)
3	X'03'	TS Profile 3 (LU–LU)
4		Primary LU Protocol
	1.......	Multiple RU Chains
	.0......	Immediate Request Mode
	..11....	Definite/Exception Response
	....00..	
	......0.	No Compression
	1	Primary may send EB
5		Secondary LU Protocol
	1.......	Multiple RU Chains
	.0......	Immediate Request Mode
	..01....	Exception Response
	....00..	
	......0.	No Compression
	0	Secondary may not send EB
6		Common Protocol
	0.......	
	.0......	FMH not Allowed
	..1.....	Bracket Protocol Used
	...1....	Bracket Termination Rule 1
	....0...	Alternate Code not Allowed
	000	
7		
	10......	Half Duplex Flip/Flop
	..0.....	PLU has Recovery Responsibility
	...0....	SLU is First Speaker
	....000.	
	0	SLU is Contention Winner
8		
	00......	
	..xxxxxx	SLU Send Pacing Count (Note 1)
9		
	00......	
	..xxxxxx	SLU Receive Pacing Count (Note 1)
10		SLU to PLU RU Size (Note 2)
	xxxx....	Mantissa
	xxxx	Exponent

Figure D–2 (Part 1 of 2). Bind Format for Interactive (Flip–Flop) LU

Byte	Value	Meaning
11		PLU to SLU RU Size (Note 2)
	xxxx....	Mantissa
	xxxx	Exponent
12		
	00.....	
	..xxxxxx	PLU CPMGR Send Pacing Count (Note 1)
13		
	00.....	
	..xxxxxx	PLU CPMGR Receive Pacing Count (Note 1)
14		
	0.....	
	.0000001	LU Type 1
15—	XL22'00'	(Remainder of Bind Area)

Notes:
1. Supplied by VTAM
2. As specified on DFHTCT Macro

Figure D—2 (Part 2 of 2). Bind Format for Interactive (Flip—Flop) LU

Byte	Value	Meaning
0	X'31'	BIND Request Code
1		
	0000....	Bind Format 0
	0001	Bind Type 1 (cold)
2	X'03'	FM Profile 3 (LU-LU)
3	X'03'	TS Profile 3 (LU-LU)
4		Primary LU Protocol
	1.......	Multiple RU Chains
	.0......	Immediate Request Mode
	..11....	Definite/Exception Response
	....00..	
	......0.	No Compression
	1	Primary may send EB
5		Secondary LU Protocol
	1.......	Multiple RU Chains
	.0......	Immediate Request Mode
	..01....	Exception Response
	....00..	
	......0.	No Compression
	0	Secondary may not send EB
6		Common Protocol
	0.......	
	.0......	FMH not Allowed
	..1.....	Bracket Protocol Used
	...1....	Bracket Termination Rule 1
	....0...	Alternate Code not Allowed
	000	
7		Common Protocol
	01......	Half Duplex Contention
	..0.....	PLU has Recovery Responsibility
	...0....	SLU is First Speaker
	....000.	
	0	SLU is Contention Winner
8		
	00......	
	..xxxxxx	SLU Send Pacing Count (Note 1)
9		
	00......	
	..xxxxxx	SLU Receive Pacing Count (Note 1)
10		SLU to PLU RU Size (Note 2)
	xxxx....	Mantissa
	xxxx	Exponent

Figure D-3 (Part 1 of 2). Bind Format for Interactive (Contention) LU

Byte	Value	Meaning
11		PLU to SLU RU Size (Note 2)
	xxxx....	Mantissa
	xxxx	Exponent
12		
	00......	
	..xxxxxx	PLU CPMGR Send Pacing Count (Note 1)
13		
	00......	
	..xxxxxx	PLU CPMGR Receive Pacing Count (Note 1)
14		
	0.......	
	.0000001	LU Type 1
15--	XL22'00'	(Remainder of Bind Area)

Notes:
1. Supplied by VTAM
2. As specified on DFHTCT Macro

Figure D-3 (Part 2 of 2). Bind Format for Interactive (Contention) LU

Byte	Value	Meaning
0	X'31'	BIND Request Code
1		
	0000....	Bind Format 0
	0001	Bind Type 1 (cold)
2	X'03'	FM Profile 3 (LU-LU)
3	X'03'	TS Profile 3 (LU-LU)
4		Primary LU Protocol
	1.......	Multiple RU Chains
	.0......	Immediate Request Mode
	..11....	Definite/Exception Response
	....00..	
	......0.	No Compression
	1	Primary may send EB
5		Secondary LU Protocol
	1.......	Multiple RU Chains
	.0......	Immediate Request Mode
	..01....	Exception Response
	....00..	
	......0.	No Compression
	0	Secondary may not send EB
6		Common Protocol
	0.......	
	.1......	FMH Allowed
	..1.....	Bracket Protocol Used
	...1....	Bracket Termination Rule 1
	....0...	Alternate Code not Allowed
	000	
7		Common Protocol
	10......	Half Duplex Flip/Flop
	..0.....	PLU has Recovery Responsibility
	...0....	SLU is First Speaker
	....000.	
	0	SLU is Contention Winner
8		
	00......	
	..xxxxxx	SLU Send Pacing Count (Note 1)
9		
	00......	
	..xxxxxx	SLU Receive Pacing Count (Note 1)
10		SLU to PLU RU Size (Note 2)
	xxxx....	Mantissa
	xxxx	Exponent

Figure D-4 (Part 1 of 3). Bind Format for Batch LU

Byte	Value	Meaning
11		PLU to SLU RU Size (Note 2)
	xxxx....	Mantissa
	xxxx	Exponent
12		
	00......	
	..xxxxxx	PLU CPMGR Send Pacing Count (Note 1)
13		
	00......	
	..xxxxxx	PLU CPMGR Receive Pacing Count (Note 1)
14		
	0.......	
	.0000001	LU Type 1
15		
	0010....	FMH Subset 2
	0000	Basic SCS Controls
16		(PLU Usage)
	0.......	2 Destinations Outstanding
	.0......	Compact Data not Sent
	..0.....	PDIR Data Set not Allowed
	...00000	
17	X'00'	
18		(PLU Usage - Document Control Characters)
	1.......	BS, CR, INP, EMP, LF, HT, VT sent
	.1......	SHF sent
	..1.....	SVF sent
	...0....	SVF SEL not sent
	....0...	SLD not sent
	.....00.	
	1	TRN and IRS sent
19		(PLU Usage)
	0.......	SLU will Initiate Attended
	.0......	and will not Alternate (Note 3)
	..000000	
20		(PLU Usage - Media Flags)
	1.......	Document Output Allowed
	.1......	Card Format Allowed
	..1.....	Exchange Media Format Allowed
	...1....	Disc Data Management Format Allowed
	....0...	Extended Card Format not Allowed
	.....0..	Extended Document Format not Allowed
	......1.	PLU requires CD on EODS
	0	

Figure D-4 (Part 2 of 3). Bind Format for Batch LU

21	X'00'	(SLU Usage — See Byte 16)
22	X'00'	
23	X'E1'	(SLU Usage — See Byte 18)
24	X'00'	(SLU Usage — See Byte 19)
25	X'F0'	(SLU Usage — See Byte 20)
26	XL11'00'	(Remainder of Bind Area)

Notes:
1. Supplied by VTAM
2. As specified on DFHTCT Macro
3. Attended/Unattended operation is transparent to CICS/VS

Figure D—4 (Part 3 of 3). Bind Format for Batch LU

Byte	Value	Meaning
0	X'31'	BIND Request Code
1		
	0000....	Bind Format 0
	0001	Bind Type 1 (cold)
2	X'03'	FM Profile 3 (LU–LU)
3	X'03'	TS Profile 3 (LU–LU)
4		Primary LU Protocol
	1.......	Multiple RU Chains
	.0......	Immediate Request Mode
	..11....	Definite/Exception Response
	....00..	
	......0.	No Compression
	1	Primary may send EB
5		Secondary LU Protocol
	1.......	Multiple RU Chains
	.0......	Immediate Request Mode
	..11....	Definite/Exception Response
	....00..	
	......0.	No Compression
	0	Secondary may not send EB
6		Common Protocol
	0.......	
	.1......	FMH not Allowed
	..1.....	Bracket Protocol Used
	...1....	Bracket Termination Rule 1
	....0...	Alternate Code not Allowed
	000	
7		Common Protocol
	10......	Half Duplex Flip/Flop
	..0.....	PLU has Recovery Responsibility
	...0....	SLU is First Speaker
	....000.	
	0	SLU is Contention Winner
8		
	00......	
	..xxxxxx	SLU Send Pacing Count (Note 1)
9		
	00......	
	..xxxxxx	SLU Receive Pacing Count (Note 1)
10		SLU to PLU RU Size (Note 2)
	xxxx....	Mantissa
	xxxx	Exponent

Figure D–5 (Part 1 of 3). Bind Format for Batch Data Interchange LU

Byte	Value	Meaning
11		PLU to SLU RU Size (Note 2)
	xxxx....	Mantissa
	xxxx	Exponent
12		
	00......	
	..xxxxxx	PLU CPMGR Send Pacing Count (Note 1)
13		
	00......	
	..xxxxxx	PLU CPMGR Receive Pacing Count (Note 1)
14		
	0.......	
	.0000001	LU Type 1
15		
	0011....	FMH Subset 3
	0001	Basic SCS Controls (and Cards may Span RUs)
16		(PLU Usage)
	0.......	2 Destinations Outstanding
	.0......	Compact Data not Sent
	..0.....	PDIR Data Set not Allowed
	...0....	Keyed Direct Data Set not Allowed
	....1...	Sequential Data Set Allowed
	.....1..	Sequential Access to Direct Data Set
	......0.	Series ID not Allowed
	0	Add or Replace Replicate not Allowed
17		(PLU Usage)
	0.......	
	.1......	Data Set Query Allowed
	..1.....	Data Set Create or Scratch Allowed
	...1....	Execute FP Allowed
	0000	
18		(PLU Usage - Document Control Characters)
	1.......	BS, CR, INP, EMP, LF, HT, VT sent
	.1......	SHF sent
	..1.....	SVF sent
	...0....	SVF SEL not sent
	....0...	SLD not sent
	.....00.	
	1	TRN and IRS sent
19		(PLU Usage)
	0.......	SLU will Initiate Attended
	.0......	and will not Alternate (Note 3)
	..000000	

Figure D-5 (Part 2 of 3). Bind Format for Batch Data Interchange LU

Byte	Value	Meaning
20		(PLU Usage — Media Flags)
	1.......	Document Output Allowed
	.1.......	Card Format Allowed
	..0.....	Exchange Media Format not Allowed
	...1....	Disc Data Management Format Allowed
	....0...	Extended Card Format not Allowed
	.....0..	Extended Document Format not Allowed
	......1.	PLU requires CD on EODS
	0	
21	X'00'	(SLU Usage — See Byte 16)
22	X'00'	(SLU Usage — See Byte 17)
23	X'E1'	(SLU Usage — See Byte 18)
24	X'00'	(SLU Usage — See Byte 19)
25	X'D0'	(SLU Usage — See Byte 20)
26	XL11'00'	(Remainder of Bind Area)

Notes:
1. Supplied by VTAM
2. As specified on DFHTCT Macro
3. Attended/Unattended operation is transparent to CICS/VS

Figure D—5 (Part 3 of 3). Bind Format for Batch Data Interchange LU

Byte	Value	Meaning
0	X'31'	BIND Request Code
1		
	0000....	Bind Format 0
	0001	Bind Type 1 (cold)
2	X'07'	FM Profile 7
3	X'07'	TS Profile 7
4		Bind Sender Protocol
	1.......	Multiple RU Chains
	.0......	Immediate Request Mode
	..11....	Definite/Exception Response
	....00..	
	......0.	No Compression
	1	Primary may send EB
5		Bind Receiver Protocol
	1.......	Multiple RU Chains
	.0......	Immediate Request Mode
	..11....	Definite/Exception Response
	....00..	
	......0.	No Compression
	0	Secondary may not send EB
6		Common Protocol
	0.......	
	.1......	FMH Allowed
	..0.....	Bracket reset state is INB
	...1....	Bracket Termination Rule 1
	....0...	Alternate Code not Allowed
	000	
7		Common Protocol
	10......	Half Duplex Flip/Flop
	..0.....	Bind Sender has Recovery Responsibility
	...0....	Bind Receiver is Contention Winner
	....000.	
	1	Reset state — send for bind sender
8		
	00......	
	..xxxxxx	Bind Receiver Send Pacing Count (Note 1)
9		
	00......	
	..xxxxxx	Bind Receiver Receive Pacing Count (Note 1)
10		Bind Receiver to Bind Sender RU Size (Note 2)
	xxxx....	Mantissa
	xxxx	Exponent

Figure D-6 (Part 1 of 3). Bind Format for Type 4 Logical Unit (LUTYPE4)

Byte	Value	Meaning
11		Bind Sender to Bind Receiver RU Size (Note 2)
	xxxx....	Mantissa
	xxxx	Exponent
12		
	00.....	
	..xxxxxx	Bind Sender CPMGR Send Pacing Count (Note 1)
13		
	00.....	
	..xxxxxx	Bind Sender CPMGR Receive Pacing Count (Note 1)
14		
	0.......	
	.0000100	LU Type 4
15		(Bind Sender Send Protocol - Printer Data Stream Profile)
	1.......	Base DSP supported
	.0.....	General DSP not supported
	..1.....	Job DSP supported
	...0....	
	....1...	WP Raw Form supported
	.....0..	
	......0.	Dual Pitch not supported
	0	Proportional Escapement not supported
16		(Bind Sender Send Protocol - Additional Data Stream Profile)
	0.......	OII level 1 not supported
	.1.....	Card Format supported
	..0.....	OII Level 2 not supported
	...0....	Basic Exchange not supported
	....0...	
	.....0..	WP Exchange Diskette not supported
	......0.	OII Level 3 not supported
	0	
17		(Bind Sender Send Protocol - Console Data Stream Profile)
	1.......	Base DSP supported
	.0.....	General DSP not supported
	..1.....	Job DSP supported
	...00000	
18		(Bind Sender Send Protocol - FM Usage)
	0.......	1-level Stack
	.0.....	No WP Format Headers
	..0.....	No Supervisory Services FM data
	...0....	No Compaction
	....00..	PDIR not supported
	......0.	No Query for Destination
	0	Bind Sender may send CD on EDS

Figure D-6 (Part 2 of 3). Bind Format for Type 4 Logical Unit (LUTYPE4)

Byte	Value	Meaning
19	X'A8'	(Bind Receiver Send Protocol– see byte 15)
20	X'40'	(Bind Receiver Send Protocol– see byte 16)
21	X'A0'	(Bind Receiver Send Protocol– see byte 17)
22	X'00'	(Bind Receiver Send Protocol– see byte 18)
23	X'00'	EBCDIC (no alternate code)
24	0000.... ....1... .....1.. 00	(General Characteristics) Bind Receiver will initialize unattended and can alternate
25	X'00'	NCI not supported
26	X'00'	CRYPTO not supported
27		LUNAME length
28		Length of User Data field
29–		User Data Field

Notes:
1. Supplied by VTAM
2. As specified on DFHTCT Macro

Figure D–6 (Part 3 of 3). Bind Format for Type 4 Logical Unit (LUTYPE4)

Index

Each page number in this index refers to the start of the paragraph containing the indexed item.

International Business Machines Corporation
Data Processing Division
1133 Westchester Avenue, White Plains, N.Y. 10604

IBM World Trade Americas/Far East Corporation
Town of Mount Pleasant, Route 9, North Tarrytown, N.Y., U.S.A. 10591

IBM World Trade Europe/Middle East/Africa Corporation
360 Hamilton Avenue, White Plains, N.Y., U.S.A. 10601

Customer Information Control System/Virtual Storage (CICS/VS)
IBM 3767/3770/6670 Guide

SC33-0074-2

READER'S COMMENT FORM

This manual is part of a library that serves as a reference source for systems analysts, programmers, and operators of IBM systems. This form may be used to communicate your views about this publication. They will be sent to the author's department for whatever review and action, if any, is deemed appropriate. Comments may be written in your own language; use of English is not required.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation whatever. You may, of course, continue to use the information you supply.

Note: *Copies of IBM publications are not stocked at the location to which this form is addressed. Please direct any requests for copies of publications, or for assistance in using your IBM system, to your IBM representative or to the IBM branch office serving your locality.*

Number of your latest Technical Newsletter for this publication. . . . . .

pressure-sensitive or other gummed tape to seal this form.

— Cut Along Dotted Line —

If you want an acknowledgement, give your name and address below.

Name . . . . . .

Job Title . . . . . Company . . . . . .

Address . . . . . .

. . . . . Zip . . . . . .

Thank you for your cooperation. No postage stamp is necessary if mailed in the U.S.A. (Elsewhere, your IBM representative or IBM branch office will be happy to forward your comments.)

SC33-0074-2

Reader's Comment Form

Fold and tape Please do not staple Fold and tape

No postage necessary if mailed in the United States

BUSINESS REPLY MAIL

FIRST CLASS PERMIT 40 ARMONK, NEW YORK

Postage will be paid by addressee:

International Business Machines Corporation
Department 812HP
1133 Westchester Avenue
White Plains, New York 10604

Fold and tape Please do not staple Fold and tape

CICS/VS IBM 3767/3770/6670 Guide Printed in U.S.A. SC33-0074-2

International Business Machines Corporation
Data Processing Division
1133 Westchester Avenue, White Plains, N.Y. 10604

IBM World Trade Americas/Far East Corporation
Town of Mount Pleasant, Route 9, North Tarrytown, N.Y., U.S.A. 10591

IBM World Trade Europe/Middle East/Africa Corporation
360 Hamilton Avenue, White Plains, N.Y., U.S.A. 10601

Customer Information Control System/Virtual Storage (CICS/VS)
IBM 3767/3770/6670 Guide
SC33-0074-2

READER'S COMMENT FORM

This manual is part of a library that serves as a reference source for systems analysts, programmers, and operators of IBM systems. This form may be used to communicate your views about this publication. They will be sent to the author's department for whatever review and action, if any, is deemed appropriate. Comments may be written in your own language; use of English is not required.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation whatever. You may, of course, continue to use the information you supply.

Note: *Copies of IBM publications are not stocked at the location to which this form is addressed. Please direct any requests for copies of publications, or for assistance in using your IBM system, to your IBM representative or to the IBM branch office serving your locality.*

Number of your latest Technical Newsletter for this publication. . . . . .

Note: *Staples can cause problems with automated mail sorting equipment. Please use pressure-sensitive or other gummed tape to seal this form.*

Cut Along Dotted Line

If you want an acknowledgement, give your name and address below.

Name . . . . . .

Job Title . . . . . Company . . . . . .

Address . . . . . .

. . . . . Zip . . . . . .

Thank you for your cooperation. No postage stamp is necessary if mailed in the U.S.A. (Elsewhere, your IBM representative or IBM branch office will be happy to forward your comments.)

SC33-0074-2

Reader's Comment Form

Fold and tape Please do not staple Fold and tape

No postage necessary if mailed in the United States

BUSINESS REPLY MAIL

FIRST CLASS PERMIT 40 ARMONK, NEW YORK

Postage will be paid by addressee:

International Business Machines Corporation
Department 812HP
1133 Westchester Avenue
White Plains, New York 10604

Fold and tape Please do not staple Fold and tape

CICS/VS IBM 3767/3770/6670 Guide Printed in U.S.A. SC33-0074-2

IBM®

International Business Machines Corporation
Data Processing Division
1133 Westchester Avenue, White Plains, N.Y. 10604

IBM World Trade Americas/Far East Corporation
Town of Mount Pleasant, Route 9, North Tarrytown, N.Y., U.S.A. 10591

IBM World Trade Europe/Middle East/Africa Corporation
360 Hamilton Avenue, White Plains, N.Y., U.S.A. 10601