



3270 Interactive Communications Software (ICS) User's Guide

Part No. 2302670-9701 *A
15 November 1983

TEXAS INSTRUMENTS

A solid red horizontal bar at the bottom of the page, preceded by three thin white lines.

© Texas Instruments Incorporated 1982, 1983

All Rights Reserved, Printed in U.S.A.

The information and/or drawings set forth in this document and all rights in and to inventions disclosed herein and patents which might be granted thereon disclosing or employing the materials, methods, techniques or apparatus described herein, are the exclusive property of Texas Instruments Incorporated.

MANUAL REVISION HISTORY

DNOS 3270 Interactive Communications Software (ICS) User's Guide
(2302670-9701)

Original Issue	15 December 1982
Revision	15 November 1983

The total number of pages in this publication is 210.

The computers offered in this agreement, as well as the programs that TI has created to use with them, are tools that can help people better manage the information used in their business; but tools—including TI computers—cannot replace sound judgement nor make the manager's business decisions.

Consequently, TI cannot warrant that its systems are suitable for any specific customer application. The manager must rely on judgment of what is best for his or her business.

Preface

The DNOS 3270 Interactive Communications Software (ICS) allows a Texas Instruments 990 computer system, operating with the DNOS operating system, to emulate the operation of a subset of the IBM 3270 Information Display System. This emulation allows an ICS-supported video terminal or any DNOS-supported sequential output device to receive data from an IBM host application program. When the data is received at an ICS display station, the operator can read, delete, or modify it, and then send it back to the host application. The operator can also put the data onto a file or print a copy of it.

An ICS printer station is a receive-only device. Once activated, the printer station can receive data transmitted by the host application program and either print a copy of it or write it onto a file without requiring an operator. ICS can also use a program that you design as an ICS display or printer station.

This manual describes the operation of the ICS display station and the ICS printer station and provides instructions for activating and terminating ICS. Also, this document describes how a user-written program can operate as an ICS display or printer station.

NOTE

In the January 1983 *Texas Instruments Computer Family Catalog* (and all subsequent catalogs), the names for several communications products have been changed. The Communications Interface Module (CIM or COMMIF) appears in the catalog as CI401. The Bit-Oriented/Character-Oriented Asynchronous Interface Module (BCAIM) appears in the catalog as CP501. The X.21 BCAIM appears in the catalog as CP502. The Four-Channel Communications Controller (FCCC) appears in the catalog as CP503. This manual uses the common abbreviations for these communications products. All references to BCAIM encompass either the CP501 or the CP502 communications processor. In correspondence or formal discussions concerning system configurations, refer to the catalog names.

This manual addresses three audiences. Section 1 contains general information about the scope of the ICS package. Section 2 provides background information about the IBM 3270 Information Display System and the DNOS operating system. Section 2 requires no technical knowledge and serves as an introduction for those users unfamiliar with either of these systems. Section 3 contains all the information required for a data entry operator to operate and terminate an ICS station. Section 4 contains information useful to both the systems operator and the systems analyst. Section 5 and the appendixes contain technical information for the systems analyst. Appendix A contains generic keyboard references. Appendix B contains explanations of the errors returned by ICS. Most errors can be corrected by the data entry operator, though some require attention by the systems operator or the systems analyst.

This manual is organized into the following sections and appendixes:

Section

- 1 General Description — Introduces Interactive Communications Software (ICS) and its capabilities, and Programmed Station Control (PSC) and its capabilities.
- 2 Functional Environment — Describes the components and operational requirements of ICS, and the supported configurations. This section also explains how the operating system treats user responses to command prompts.
- 3 ICS Station Operation — Describes the activation, termination, and operation of the ICS printer and video stations.
- 4 Functional Description — Describes the installation process, ICS line activation, configuration management utilities, and the statistics kept by ICS.
- 5 Programmed Station Control (PSC) — Describes PSC, its capabilities, and its operation.

Appendix

- A Generic Keynames — Describes the generic keynames for video display terminals and their cross reference names.
- B ICS Messages and Codes — Describes the error messages that may be logged or displayed during operation of ICS or PSC.
- C ICS Hardware Requirements — Describes the memory and hardware requirements of ICS and PSC.
- D Supported Character Set — Describes the EBCDIC character set supported for an ICS station and provides the ASCII-to-EBCDIC and EBCDIC-to-ASCII code translations.
- E Summary of ICS Commands and Key Functions — Provides a complete list of ICS SCI commands and special key functions supported by ICS.
- F Binary Synchronous Communications(BSC) Procedures — Describes the protocol that ICS uses to communicate with an IBM-compatible host computer.

The following documents contain information about the DNOS operating system, the video terminals supported by ICS, and the printers supported by the operating system.

Title	Part Number
<i>DNOS 3270 Interactive Communications Software (ICS) Object Installation</i>	2302671-9701
<i>Release Information, DNOS 3270 Interactive Communications Software (ICS), Current Release</i>	2276894-9901
<i>DNOS COBOL Programmer's Guide</i>	2270516-9701
<i>DNOS COBOL Reference Manual</i>	2270518-9701
<i>DNOS FORTRAN-78 Programmer's Guide</i>	268680-9701
<i>FORTRAN-78 Reference Manual</i>	2268681-9701
<i>DNOS TI Pascal Programmer's Guide</i>	2270517-9701
<i>TI Pascal Reference Manual</i>	2270519-9701
<i>990/99000 Assembly Language Reference Manual</i>	2270509-9701
<i>990/10 Computer Hardware Reference Manual</i>	945417-9701
<i>DNOS Link Editor Reference Manual</i>	2270522-9701
<i>DNOS Messages and Codes Reference Manual</i>	2270506-9701
<i>DNOS System Command Interpreter (SCI) Reference Manual</i>	2270503-9701
<i>Model 911 Video Display Terminal (VDT) Installation and Operation Manual</i>	945423-9701
<i>Model 931 Video Display Terminal (VDT) General Description</i>	2229228-0001
<i>Model 940 Electronic Video Terminal (EVT) Installation and Operation Manual</i>	2250368-9701
<i>Model 810 Printer Installation and Operation</i>	939460-9701
<i>Model 2230 and 2260 Line Printers Installation and Operation</i>	946256-9701
<i>Model LP300 and LP600 Line Printers Installation and Operation Manual</i>	2250364-9701

Title	Part Number
<i>Model LQ45 Letter Quality Printer System Installation and Operation Manual</i>	2268695-9701
<i>Bit-Oriented/Character-Oriented/Asynchronous Interface Module Installation and Operation Manual</i>	2263886-9701
<i>Four Channel Communications Controller Installation and Operation Manual</i>	2263878-9701
<i>DNOS Remote Terminal Subsystem (RTS) Hardware Installation Manual, Domestic Edition</i>	2272053-9702
<i>Remote Terminal Subsystem (RTS) Site Preparation Manual</i>	2250385-9701
<i>DNOS Remote Terminal Subsystem (RTS) User's Guide</i>	2302676-9701

The following IBM documents provide additional information concerning the interaction between an IBM host system and an IBM 3270 Information Display System.

Title	Part Number
<i>IBM 3270 Information Display System Component Description</i>	GA27-2749
<i>IBM General Information — Binary Synchronous Communications</i>	GA27-3004

Contents

Paragraph	Title	Page
1 — General Description		
1.1	Introduction	1-1
1.2	Terminology	1-1
1.3	Conventions Used in This Manual	1-2
1.4	Emulator Components	1-3
1.4.1	IBM 3270 Information Display System	1-3
1.4.2	ICS Configuration	1-5
1.4.2.1	Video Terminals	1-6
1.4.2.2	Printer Stations	1-7
1.5	Operational Description	1-7
1.6	Supported Features	1-9
1.7	Extended Features	1-10
1.8	Unsupported Features	1-11
2 — Functional Environment		
2.1	Introduction	2-1
2.2	Supported Configurations	2-1
2.2.1	Supported Video Terminals	2-1
2.2.2	Remote Configurations	2-4
2.2.2.1	Remote Terminal Controller (RTC)	2-5
2.2.3	Multiple Channel Installation	2-5
2.3	3270 Identifiers	2-5
2.3.1	IBM Control Unit Address	2-6
2.3.2	IBM Terminal Address	2-7
2.4	Requirements for ICS Operation	2-7
2.4.1	Hardware Requirements	2-7
2.4.1.1	Communications Hardware	2-8
2.4.1.2	ICS Station Hardware	2-8
2.4.2	Software Requirements	2-8
2.5	Operating System Concepts	2-9
2.5.1	System Command Interpreter (SCI)	2-9
2.5.2	SCI Command Prompts	2-10
2.5.2.1	Device Names	2-10
2.5.2.2	File Pathnames	2-10
2.5.2.3	Synonym Responses	2-11
2.5.2.4	ICS Command Prompts	2-11
2.6	SCI Commands Unique to ICS	2-14

Paragraph	Title	Page
3 — ICS Station Operation		
3.1	Introduction	3-1
3.2	Activating An ICS Station	3-1
3.2.1	Execute Interactive Communications Terminal — XICT	3-2
3.2.1.1	COMMUNICATIONS LINE NAME Prompt	3-2
3.2.1.2	TERMINAL ADDRESS (0 – 31) Prompt	3-2
3.2.1.3	PRINT KEY PATHNAME Prompt	3-3
3.2.1.4	UNPROTECTED FIELD INDICATOR Prompt	3-3
3.2.1.5	Examples of ICS Video Terminal Activation	3-4
3.2.1.6	Verifying Video Terminal Activation	3-5
3.2.2	Execute Interactive Communications Printer — XICP	3-6
3.2.2.1	COMMUNICATIONS LINE NAME Prompt	3-6
3.2.2.2	TERMINAL ADDRESS(0 – 31) Prompt	3-6
3.2.2.2	OUTPUT PATHNAME Prompt	3-7
3.2.2.3	Examples of ICS Printer Station Activation	3-7
3.2.2.4	Verifying Printer Station Activation	3-8
3.3	Exiting ICS	3-8
3.3.1	Suspending ICS	3-9
3.3.2	Terminating ICS	3-9
3.3.2.1	Quit Interactive Communications Terminal — QICT	3-10
3.3.2.2	Terminating ICS at an Active Station	3-12
3.4	Video Display Terminal Operation	3-12
3.4.1	Keyboard Layout	3-13
3.4.2	Displays	3-17
3.4.2.1	Formatted Displays	3-17
3.4.2.2	Unformatted Displays	3-19
3.4.2.3	Cursor	3-19
3.4.2.4	Display Messages	3-19
3.4.3	Special Features	3-19
3.4.3.1	Automatic Skip	3-19
3.4.3.2	Audible Alarm	3-20
3.4.4	Key Functions	3-20
3.4.4.1	Repeat Function	3-21
3.4.4.2	Cursor Movement Functions	3-21
3.4.4.3	Field Movement Functions	3-22
3.4.4.4	Tab Functions	3-22
3.4.4.5	Editing Functions	3-24
3.4.4.6	Suspend Function	3-28
3.4.4.7	Terminate Function	3-28
3.4.4.8	Print Functions	3-28
3.4.4.9	Reset Function	3-29
3.4.4.10	Duplicate Functions	3-29
3.4.4.11	Field-Mark Function	3-30
3.4.4.12	Alpha Function	3-30
3.4.4.13	Program-Attention Functions	3-31

Paragraph	Title	Page
4 — Functional Description		
4.1	Introduction	4-1
4.2	Installation	4-1
4.2.1	ICS Installation and Generation	4-2
4.2.2	Initialization of ICS	4-2
4.2.3	Execute Interactive Communications Controller Task — XICC	4-2
4.2.4	Downloading Character Detect Routines	4-3
4.3	Configuration Management	4-3
4.3.1	List Interactive Communications Configuration — LICC	4-4
4.3.2	Modify Interactive Communications Configuration — MICC	4-6
4.3.3	Modify Control Unit Address — MCUA	4-8
4.4	List Interactive Communications Statistics — LICS	4-8
4.4.1	Message	4-9
4.4.2	Block	4-9
4.4.3	Retransmission	4-9
4.4.4	Abort	4-9
4.4.5	Command Reject	4-9
4.4.6	Polls	4-10
4.4.7	Poll Error	4-10
4.4.8	Select Error	4-10
4.4.9	Buffer Error	4-10
4.5	Execute Poll Analyzer — XPA	4-10
4.6	Controller Task	4-11
4.7	ICS Video Stations	4-12
4.8	ICS Printer Stations	4-13
4.8.1	Supported Printers	4-13
4.8.2	Printing Initiated by the Host Application Program	4-13
4.8.3	Print Line Formatting	4-13
4.8.4	Error Conditions	4-15
4.9	Local Format Feature	4-15

5 — Programmed Station Control

5.1	General Description	5-1
5.2	PSC Program Requirements	5-3
5.2.1	PSC Station Buffers	5-3
5.2.2	Attribute Data	5-4
5.2.2.1	Attribute Argument Returned by Fill and Get Routines	5-4
5.2.2.2	Attribute List Returned by Retrieve Attributes	5-5
5.2.3	Host Application Commands and Orders	5-6
5.2.4	ICS Station Selection	5-9
5.2.4.1	First Word	5-10
5.2.4.2	Second Word	5-11
5.2.5	Completion Codes	5-13
5.3	PSC Routines	5-13

Paragraph	Title	Page
5.3.1	Open Routine — Open Station for PSC Use	5-14
5.3.1.1	Open Call Statements	5-14
5.3.1.2	Arguments for the Open Routine	5-16
5.3.1.3	Results of the Open	5-16
5.3.1.4	Examples of the Open Call	5-17
5.3.2	Fill Routine — Fill Station Buffer	5-17
5.3.2.1	Fill Call Statements	5-18
5.3.2.2	Arguments for the Fill Routine	5-18
5.3.2.3	Results of the Fill	5-19
5.3.2.4	Examples of the Fill Call	5-20
5.3.3	Get Routine — Get Data From the Station Buffer	5-21
5.3.3.1	Get Call Statements	5-21
5.3.3.2	Arguments for the Get Routine	5-22
5.3.3.3	Results of the Get	5-23
5.3.3.4	Examples of the Get Call	5-23
5.3.4	Retrieve Attributes Routine — Return Field Attributes	5-24
5.3.4.1	Retrieve Attributes Call Statements	5-25
5.3.4.2	Arguments for the Retrieve Attributes Routine	5-25
5.3.4.3	Results of the Retrieve Attributes	5-26
5.3.4.4	Examples of the Retrieve Attributes Call	5-27
5.3.5	Transmit Routine — Send Station Buffer to Host	5-28
5.3.5.1	Transmit Call Statements	5-28
5.3.5.2	Arguments for the Transmit Routine	5-28
5.3.5.3	Results of the Transmit	5-29
5.3.5.4	Examples of the Transmit Call	5-30
5.3.6	Time-Out Routine — Wait for Host to Respond	5-31
5.3.6.1	Time-Out Call Statements	5-31
5.3.6.2	Arguments for the Time-Out Routine	5-32
5.3.6.3	Results of the Time-Out	5-32
5.3.6.4	Examples of the Time-Out Call	5-32
5.3.7	Close Routine — Close Station	5-33
5.3.7.1	Close Call Statements	5-33
5.3.7.2	Arguments for the Close Routine	5-34
5.3.7.3	Results of the Close	5-34
5.3.7.4	Examples of the Close Call	5-35
5.4	Writing a PSC Program	5-35
5.4.1	COBOL Call Statements	5-37
5.4.2	FORTRAN Call Statements	5-40
5.4.3	Pascal Call Statements	5-42
5.4.4	990 Assembly Language Call Statements	5-45
5.4.5	Calling Sequence Examples	5-48
5.5	Linking PSC Programs	5-56
5.6	Patch ICS PSC User Task — PTPSC	5-58
5.7	Execute PSC Task — XPSC	5-59

Appendixes

Appendix	Title	Page
A	Keycap Cross-Reference	A-1
B	ICS Messages and Codes	B-1
C	ICS Hardware Requirements	C-1
D	ASCII/EBCDIC, EBCDIC/ASCII Translation Tables	D-1
E	Summary of ICS Commands and Key Functions	E-1
F	Binary Synchronous Communications (BSC) Procedures	F-1

Index

Illustrations

Figure	Title	Page
1-1	IBM 3270 Information Display System Configuration	1-4
1-2	3271 Control Unit Cluster	1-5
1-3	ICS Configuration	1-6
1-4	3270 Clusters and ICS Sharing a Line	1-8
2-1	Typical 990 System Configured With ICS	2-2
2-2	ICS Supported Video Terminals	2-3
2-3	Multiple ICS Configuration	2-7
2-4	ICS Menu	2-15
3-1	Unprotected Field Indicator	3-4
3-2	Display Screen Format	3-18
3-3	Remote Duplicate Graphics Character	3-30
3-4	Field-Mark Graphics Character	3-30
5-1	PSC Overview	5-2
5-2	Attribute Argument	5-4
5-3	Attribute List	5-6
5-4	Word One of the First Argument	5-11
5-5	Word Two of the First Argument	5-12
5-6	PSC Activation Flow	5-15
5-7	Example COBOL Calls to PSC Routines (3 Sheets)	5-48
5-8	Example FORTRAN Calls to PSC Routines (3 Sheets)	5-50
5-9	Example Pascal Calls to PSC Routines (2 Sheets)	5-52
5-10	Example 990 Assembly Language Calls to PSC Routines (3 Sheets)	5-54

Tables

Table	Title	Page
2-1	Decimal-to-EBCDIC Address Mapping	2-6
2-2	Device Name Identifiers	2-10
2-3	Pathname Examples	2-11
3-1	Identical ICS/3277 Keys	3-13
3-2	ICS/3277 Single Key Equivalents	3-14
3-3	ICS Two-Key Sequences That Emulate a 3277 Key	3-15
3-4	ICS Keys Without IBM 3277 Equivalents	3-16
3-5	Cursor Movement and Edit Function Key Mapping	3-27
3-6	Special Function Key Mapping	3-32
4-1	Terminal States	4-5
5-1	Attention ID (AID) Codes	5-29

General Description

1.1 INTRODUCTION

The DNOS 3270 Interactive Communications Software (ICS) allows a Texas Instruments 990 computer system to emulate a subset of the IBM 3270 Information Display System. This emulation enables the 990 system to communicate interactively with a host computer in much the same way that an IBM 3270 Information Display System does. Therefore, you can interact with a wide range of inquiry and/or data entry applications designed for the IBM 3270 Information Display System. ICS provides this capability without losing any of the multitasking capabilities of the 990. ICS also provides Programmed Station Control (PSC), which allows a user-written program to substitute as operator of an ICS station.

This section provides a general description of the 3270 ICS emulator used with the 990. It discusses the terminology and conventions used in this manual and identifies the components necessary for 3270 emulation. This section lists the features of standard IBM 3270 that are supported and those that are not. This section also lists features of this package that are not standard to IBM 3270. Consult the *Release Information, DNOS 3270 Interactive Communications Software ICS, Current Release* (part number 2276894-9901) to find out which releases of the DNOS operating system support the current release of DNOS 3270 ICS.

1.2 TERMINOLOGY

ICS is an acronym for Interactive Communications Software. Throughout this manual, the DNOS 3270 ICS emulator is abbreviated as *ICS*.

A *local station* is a terminal connected directly to the computer that controls it, without using modems and telephone (or equivalent) lines. Using digital signals, local stations send data back and forth across relatively short distances.

A *remote station* is a terminal that communicates with the computer that controls it through modems and some form of communications line. Modems transform the digital output of the computer (discrete logic that is either on or off) into analog form suitable for long-distance transmission. The distance between the remote terminal and its computer need not be long-distance, but there is no limit to the distance possible between the two.

The *host computer* is any IBM 3270-compatible computer. This computer is not necessarily an IBM computer. It can be any computer that serves as an IBM 3270 host. The host is the computer with which an ICS terminal communicates interactively.

A *communications session* is defined as the period from ICS station activation to termination. During this time, necessary system resources (such as a unique 3270 terminal address) are reserved for the active local station.

A *poll* is the operation performed by the host when it invites transmission. There are two types of polls. A *general poll* invites transmission from any device or emulator on a particular control unit. A *specific poll* invites transmission only from a specific device or emulator on a particular control unit. Upon receiving the poll, the device or emulator either sends data or responds that it currently has no data to send.

A *select* is the operation performed by the host when it initiates transmission to a particular device on a 3270 control unit or emulator on the line. After the control unit acknowledges the select, the host begins transmitting.

DNOS System Command Interpreter (SCI) commands activate ICS and provide auxiliary services. Those SCI commands that service ICS are not standard DNOS commands but are added during ICS installation. Appendix E contains a complete list of DNOS SCI commands unique to ICS installation.

1.3 CONVENTIONS USED IN THIS MANUAL

The video display terminals (VDTs) supported by ICS can display characters on the screen in high and low intensity. These terminals display SCI prompts in low intensity. When you enter characters from the keyboard, the system displays those characters in high intensity. This manual shows operator entries underlined. Low intensity displays appear in standard type. SCI frequently supplies initial values to the prompts it displays. These initial values are displayed in low intensity on the VDT and in standard uppercase type in this manual. The following is an example of how this manual depicts operator entries:

```
[ ] XICT
```

```
EXECUTE INTERACTIVE COMMUNICATIONS TERMINAL  
COMMUNICATIONS LINE NAME: CM01  
TERMINAL ADDRESS(0-31):  
PRINT KEY PATHNAME: DATA.MARCH.WIGITS  
UNPROTECTED FIELD INDICATOR: YES
```

The brackets ([]) represent the SCI prompt, and XICT represents your response to SCI.

NOTE

DNOS allows the setting of an .OPTION primitive (LOWERCASE = YES) to enable lowercase to uppercase mapping of characters. You can then use uppercase or lowercase characters in response to SCI prompts. If the lowercase option is not set, SCI recognizes only uppercase characters. In that case you must use the SHIFT key in conjunction with the data keys or be sure that the UPPER CASE LOCK key is engaged. For consistency in this manual, all responses to the SCI prompt are shown in uppercase.

After typing the SCI command XICT, press the RETURN key to initiate servicing by SCI. Other keys, such as the ENTER and NEXT FIELD, accomplish the same function. The ENTER key performs the same function as typing a / (slash) after the command and then pressing RETURN. That is, the system accepts all existing default and initial values for the command and goes directly to execution without presenting the prompts for the given command. Throughout this manual, it is assumed that you press the RETURN (or equivalent) key after entering values into a field. After you enter XICT, ICS displays on the screen the SCI command procedure name, the field prompts for the procedure, and any initial values that may be supplied.

The CM01 next to the COMMUNICATIONS LINE NAME prompt is depicted in standard type indicating that it is an initial value supplied by SCI. If you press the RETURN key, the value CM01 is entered as if you had manually typed it. No response is shown next to the TERMINAL ADDRESS (0-31) prompt. None is required, and no initial value is given for this field. Pressing the RETURN key moves the cursor to the next prompt and enters a null value into the terminal address field. Where null responses are accepted, ICS uses a default value. DATA.MARCH.WIGITS is underlined to indicate that the operator typed it.

The XICT command is shown here only to illustrate the conventions used throughout this manual. Paragraph 3.3.1 details the XICT command.

1.4 EMULATOR COMPONENTS

A 990 computer can use ICS to communicate with any host computer that supports an IBM 3271 Model 2 Control Unit. The following paragraphs compare the components of the IBM 3270 Information Display System with the components used by ICS.

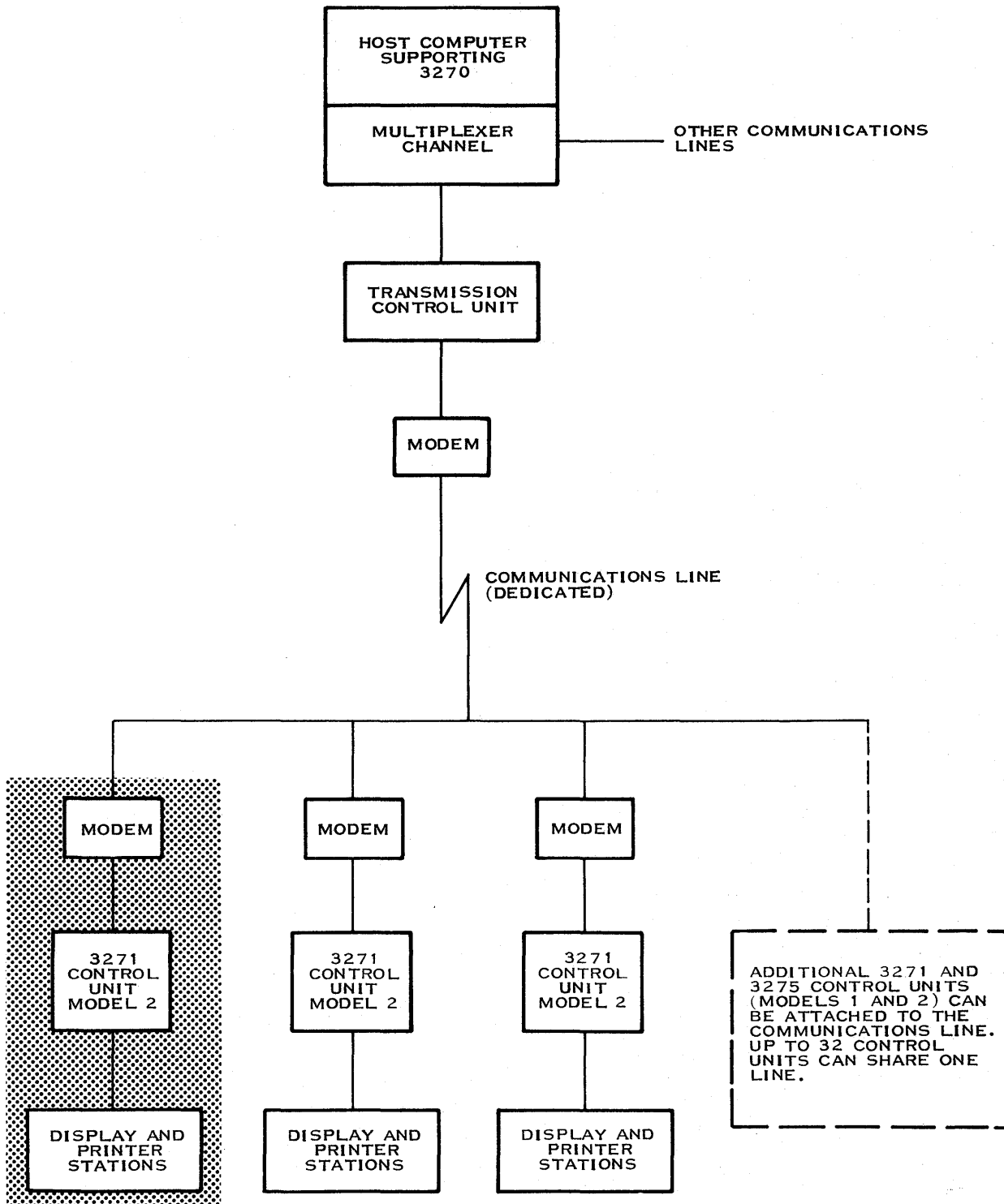
1.4.1 IBM 3270 Information Display System

This paragraph describes the IBM 3270 Information Display System that ICS emulates. Figure 1-1 shows a typical configuration of the IBM 3270 Information Display System. The main components of the system are a communications line, a modem, a transmission control unit, display stations, and printer stations.

The communications line is a dedicated line, usually a leased telephone line. Each control unit on the line uses a modem to translate digital signals into a form suitable for transmission across the communications line.

The host computer identifies each individual 3271 control unit by a unique *IBM control unit address*. Up to 32 control units can share the same communications line. The host can communicate with any individual 3271 control unit on the line. Two 3271 control units, however, cannot communicate with each other directly. Each 3271 control unit on the communications line can have several 3277 display stations and 3284 printer stations attached to it. The shaded area of Figure 1-1 shows the components emulated by ICS installed on a 990 computer.

Figure 1-2 provides a more detailed view of the IBM components that ICS emulates. The figure shows a single IBM 3271 Model 2 Control Unit and its modem, display stations, and printer stations. At least one 3277 display station must be attached. Usually, at least one 3284 printer station is present. Up to 32 stations (display or printer) can be attached to one control unit. Each station is assigned an *IBM terminal address* to identify it to the host.



2281808

Figure 1-1. IBM 3270 Information Display System Configuration

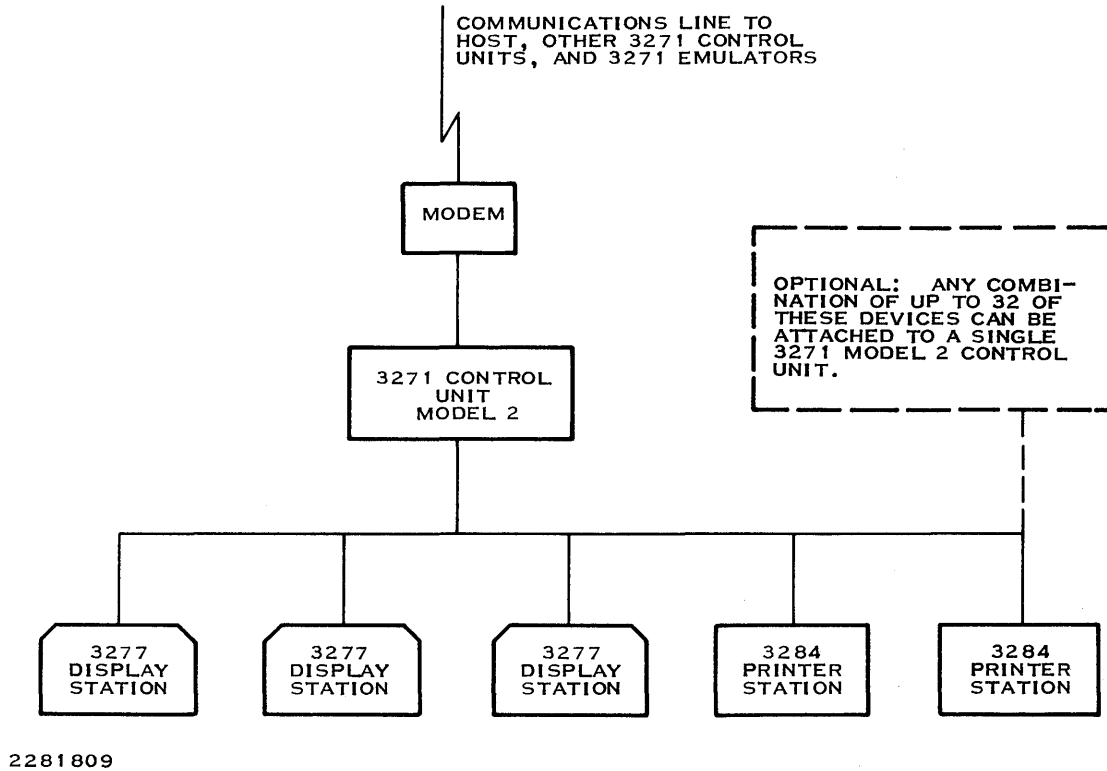


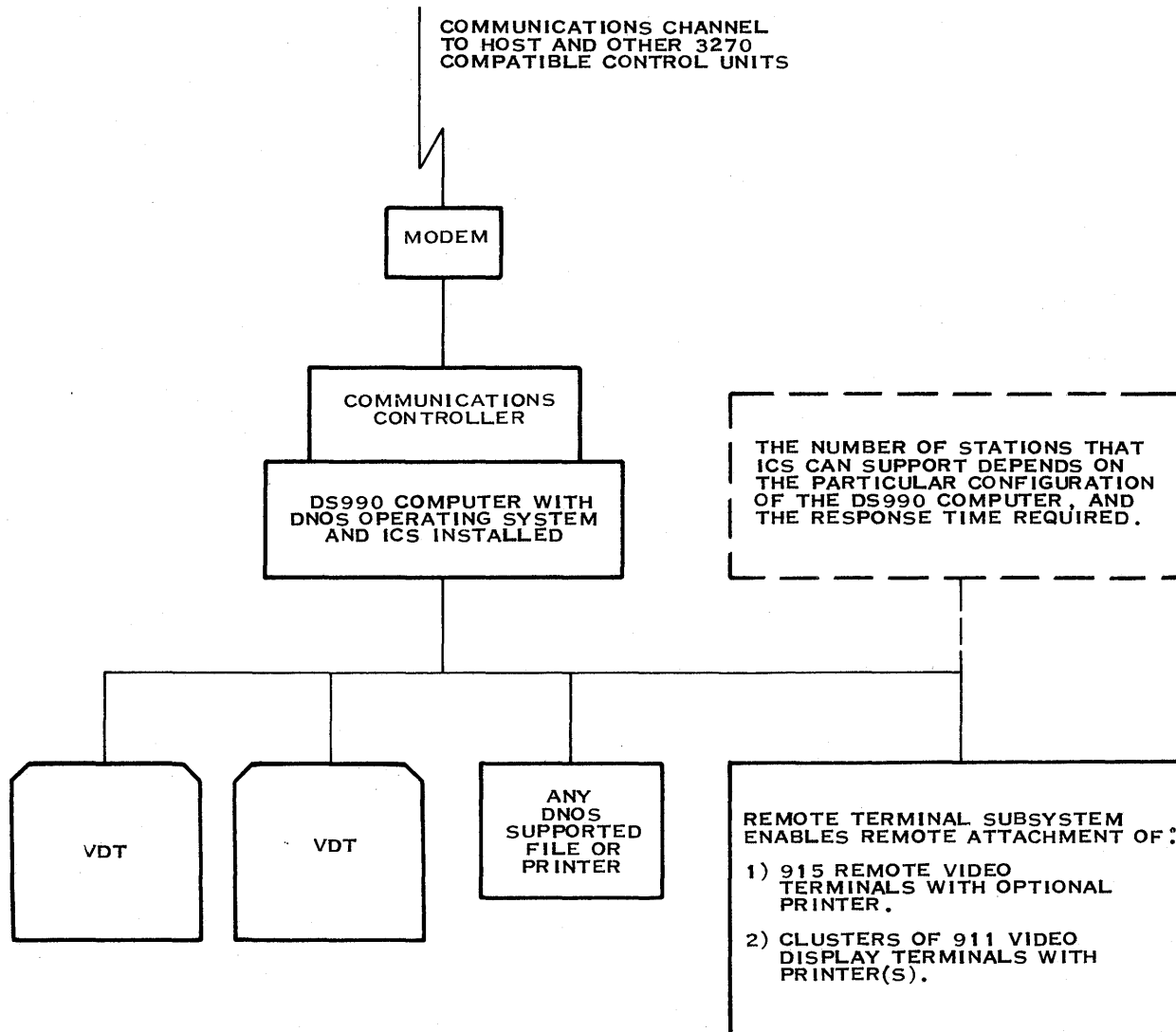
Figure 1-2. 3271 Control Unit Cluster

1.4.2 ICS Configuration

This paragraph briefly describes the components that ICS uses to emulate the 3270 system. Section 2 gives a more detailed description, and Appendix C summarizes hardware requirements.

Figure 1-3 shows a typical configuration of a 990 computer using ICS to communicate with a host computer.

The practical maximum number of active ICS stations that DNOS can support depends on memory size, the hardware supported by the system, and the acceptable response time for a particular host application. ICS supports the maximum number of terminal addresses supported by a standard 3271 control unit, which is 32. The minimum configuration required for operation of ICS is a video terminal, a communications controller, a modem, and a communications line. PSC programs can serve in place of ICS video and printer stations. Printer devices are not required.



2283591

Figure 1-3. ICS Configuration

1.4.2.1 Video Terminals. ICS supports the following video terminals:

- Model 911 Video Display Terminal (VDT)
- Model 915 Remote Video Terminal (RVT)
- Model 931 Video Display Terminal (VDT)
- Model 940 Electronic Video Terminal (EVT)

When installed on a DNOS operating system, these terminals can serve as ICS stations. An ICS station requires no special installation procedure. Any of these terminals can be attached remotely using an additional communications link (Figure 1-4). If a terminal is connected to a communications port, no other terminal can use that port until the first terminal terminates the communications session. When used as ICS stations, remotely attached terminals operate the same as locally attached terminals.

1.4.2.2 Printer Stations. Any printer or file device that runs under DNOS can serve as an ICS printer station. In addition to accommodating a printer, ICS can copy data onto a file on a magnetic medium. You can print this file at any convenient time. DNOS supports several types of hard disk media, magnetic tape, single-sided flexible disks, and double-sided, double-density (DSDD) flexible disks. DNOS also supports several types of printers, including dot matrix, high-speed drum, and letter quality. If logical names have been defined for DNOS spooler operation, those spooler logical names can be used as printer stations. In this case, the spooler chooses the particular printer available for use according to the class name supplied or routes the output to a specific printer if so specified by operator intervention. (Refer to the *DNOS Operations Guide*, part number 2270502-9701, for information concerning spooler facilities and operation.) Any sequential output device installed on the DNOS operating system can serve as an ICS printer station. DNOS requires no special installation procedures.

1.5 OPERATIONAL DESCRIPTION

When you install ICS on a 990 system, you must specify several parameters to identify the components of the system.

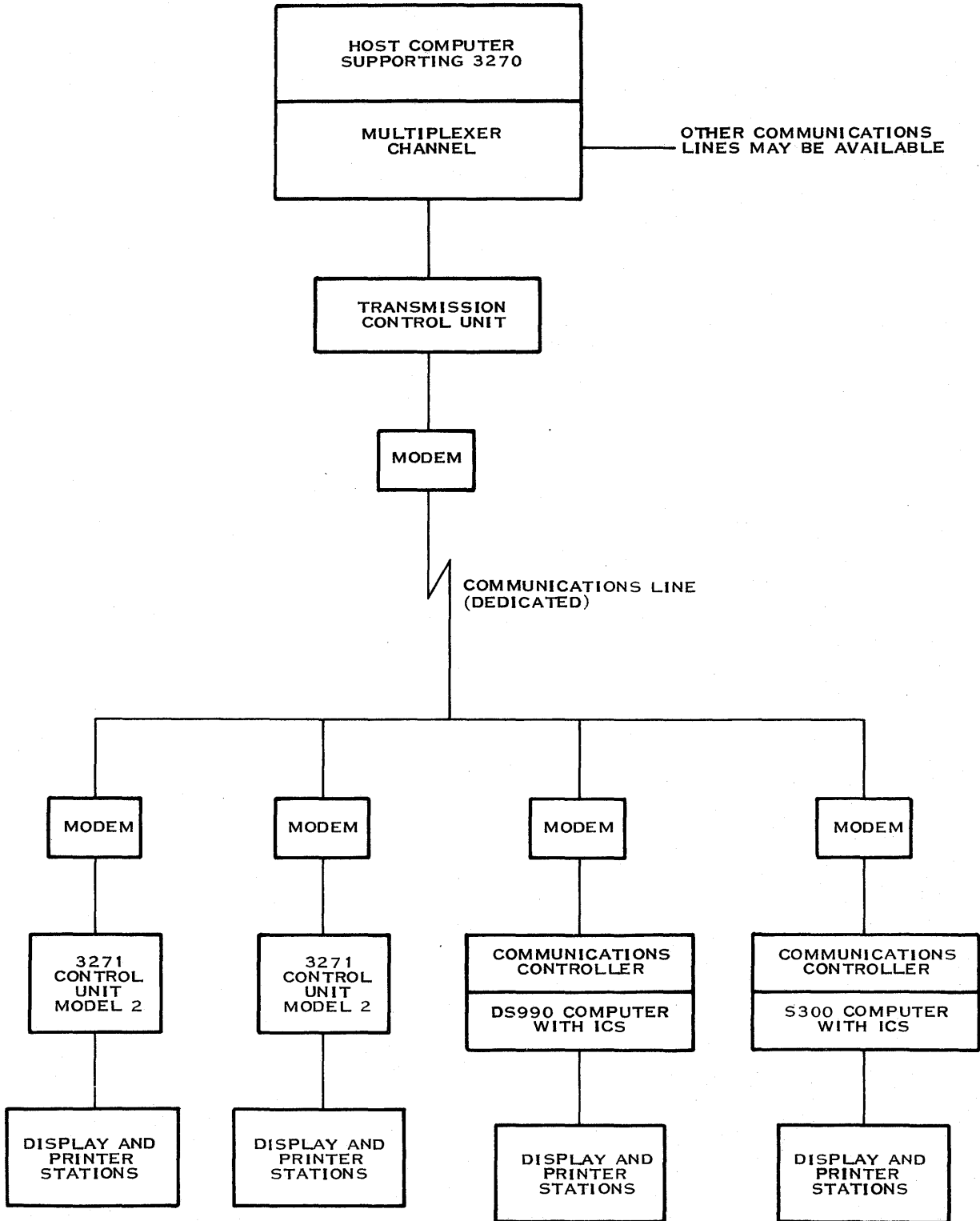
Up to 32 control units or emulators can share the same communications line. ICS and 3271 control unit clusters can share the same line, as shown in Figure 1-4.

The 3271 control unit address identifies a particular control unit or 3270 emulator to the host. When polling or selecting, the host uses this value to specify one control unit or emulator on the communications line. You must install this value on both the host and local 990 system. A 990 system can have more than one ICS communications line installed. When two lines are installed, you must specify the desired communications line.

In addition to the control unit address, all display stations and printer stations are assigned terminal addresses so that the host can direct its communications to a specific ICS station. These station addresses are allocated during installation but are not assigned to a particular station until ICS is activated. When ICS is not active at a particular station, that station is available to service any other application running on the 990 system.

Before a particular ICS station can be activated, the ICS line (controller task) must be activated. When activated, ICS indicates to the host that the status of the ICS display stations and printer stations has changed from not ready to ready. The stations can then be activated to send or receive data. An ICS printer station can only receive data.

Under Programmed Station Control (PSC), ICS responds to the PSC task (user-written program linked with the PSC routines) request to read data, modify fields, and transmit modified data to the host by means of subroutine calls. With PSC, a physical device is not required for use as a station.



2281811

Figure 1-4. 3270 Clusters and ICS Sharing a Line

1.6 SUPPORTED FEATURES

ICS supports the following features of the IBM 3271 Model 2 Control Unit, the IBM 3277 Model 2 Display Station, and the IBM 3284 Model 2 Printer:

- EBCDIC character set — ICS provides ASCII-to-EBCDIC and EBCDIC-to-ASCII translation so that ASCII terminals can communicate with hosts using the EBCDIC character set.
- Data link control — ICS supports all facets of the Binary Synchronous Communications (BSC) protocol used by the IBM 3271 Model 2 Control Unit.
- Key functions — ICS honors all functions of the 66-key and 78-key typewriter and key-punch keyboards available for the 3277 display station, except TEST REQ.
- Line speed — ICS supports line speeds of up to 9600 bits per second.
- Audible alarm — ICS supports an audible alarm that sounds when the operator presses keys in certain situations or when the host sends data and wants to alert the operator.
- Commands — ICS supports the following 3270 Information Display System commands:

Command	Description
Write	Put data into designated fields of the station buffer
Erase/Write	Clear the station buffer before loading with data
Read Modified	Retrieve all modified fields from the station buffer
Erase All Unprotected	Clear all unprotected fields in a station buffer

- Orders — ICS supports all of the IBM 3271 Model 2 Control Unit orders as follows:

Order	Description
New Line	Begin new line in printed output
Form Feed	Execute form feed to the top of the next printed page
End of Message	End of printed output
Start of Field	Mark the beginning of a field
Set Buffer Address	Select buffer address
Insert Cursor	Position cursor

Order	Description
Program Tab	Tab to next unprotected field
Repeat to Address	Insert the character specified into the station buffer
Erase Unprotected to Address	Change characters to null values

- International character sets — ICS supports special character sets for the United Kingdom, France/Belgium, Germany/Austria, Norway/Denmark, Sweden/Finland, and Japan. Refer to the *Release Information, DNOS 3270 Interactive Communications Software (ICS)*, Current Release (part number 2276894-9901) for more information concerning this feature.

1.7 EXTENDED FEATURES

The following features are ICS extensions to the IBM features supported:

- Lowercase display — An ICS display station can display lowercase characters.
- File or sequential input/output (I/O) device — A sequential I/O device or a sequential or relative record file can serve as an ICS printer station instead of a printer. If a relative record file serves as the ICS printer station, the logical record length for that file must be equal to or greater than the number of characters per line for the incoming data.
- Local print capability (PRINT key) — When you press the PRINT key, an ICS display station can store the contents of the screen in a file or print a copy at any DNOS-supported printer. ICS supports the identification function and also allows you to cancel a print in progress.
- Statistics display — ICS provides displays of ICS activity for accounting or analysis purposes.
- Programmed Station Control (PSC) — A user-written program can serve as an ICS station by using the PSC routines.
- Station suspension — ICS permits the user to temporarily suspend an active ICS display station without terminating the session. The operator can temporarily exit ICS (saving any transaction in progress if the local format option has been installed), execute SCI commands, and then reenter ICS without losing any screen data.
- Virtual operation — Once activated, ICS can receive transmissions from the host even though no ICS station is active. ICS acknowledges reception to the host and stores the data in a file. When an ICS station is activated, ICS outputs any queued transmissions.
- Erase to end-of-screen function — ICS supports a key function that fills all modifiable fields with nulls from the position of the cursor to the end of the display screen.

- **Local duplicate** — You can copy the contents of the previous modifiable field into the current field. ICS performs this copy locally before it transmits the data to the host.
- **Tab functions** — ICS provides two modes of tabbing that allow stopping either at fixed tabs or according to the data within a field. You can append the cursor to the end of a word, tab to the beginning of a word, or tab to the first character position of the next unprotected field.
- **Dynamic configuration** — ICS provides utilities that can dynamically modify the installed configuration.
- **Limited poll trace** — ICS provides a limited poll trace of line activity.
- **Host application access** — The installed ICS configuration serves as a pool of station addresses and device types from which any DNOS user can gain access to host applications from any ICS supported station by activating an ICS command procedure.

1.8 UNSUPPORTED FEATURES

The following features, available on IBM 3271 Model 2 Control Units, IBM 3277 Model 2 Display Stations, and IBM 3284 Model 2 Printers, are not supported by ICS:

- **Selector pen (light pen)** — This feature permits you to use a light-pen device to select fields of data for input to the host system.
- **Security keylock** — This feature allows you to lock the keyboard so that data cannot be displayed or modified.
- **Operator identification card reader** — A 3277 display station card reader can read a small card that identifies the operator to the system, allowing access to the data.
- **Read Buffer and Copy commands** — The Read Buffer command allows the application program executing at the host to specify that buffer data be transferred to the host system. The Copy command transfers buffer data from one station to another station attached to the same 3271 control unit.
- **ASCII transmission code** — This feature permits you to transfer data in ASCII code.
- **TEST REQ key** — This key initiates a test performed on test data that you supply.

Functional Environment

2.1 INTRODUCTION

This section describes the configurations that support the DNOS 3270 Interactive Communications Software (ICS). Included are the following topics:

- The identifiers that the standard 3270 information display system uses and how ICS processes them
- The prerequisites for ICS operation
- Operating system concepts relevant to ICS operation
- How ICS uses the DNOS System Command Interpreter (SCI), the user interface to the operating system
- SCI commands that provide control of ICS

2.2 SUPPORTED CONFIGURATIONS

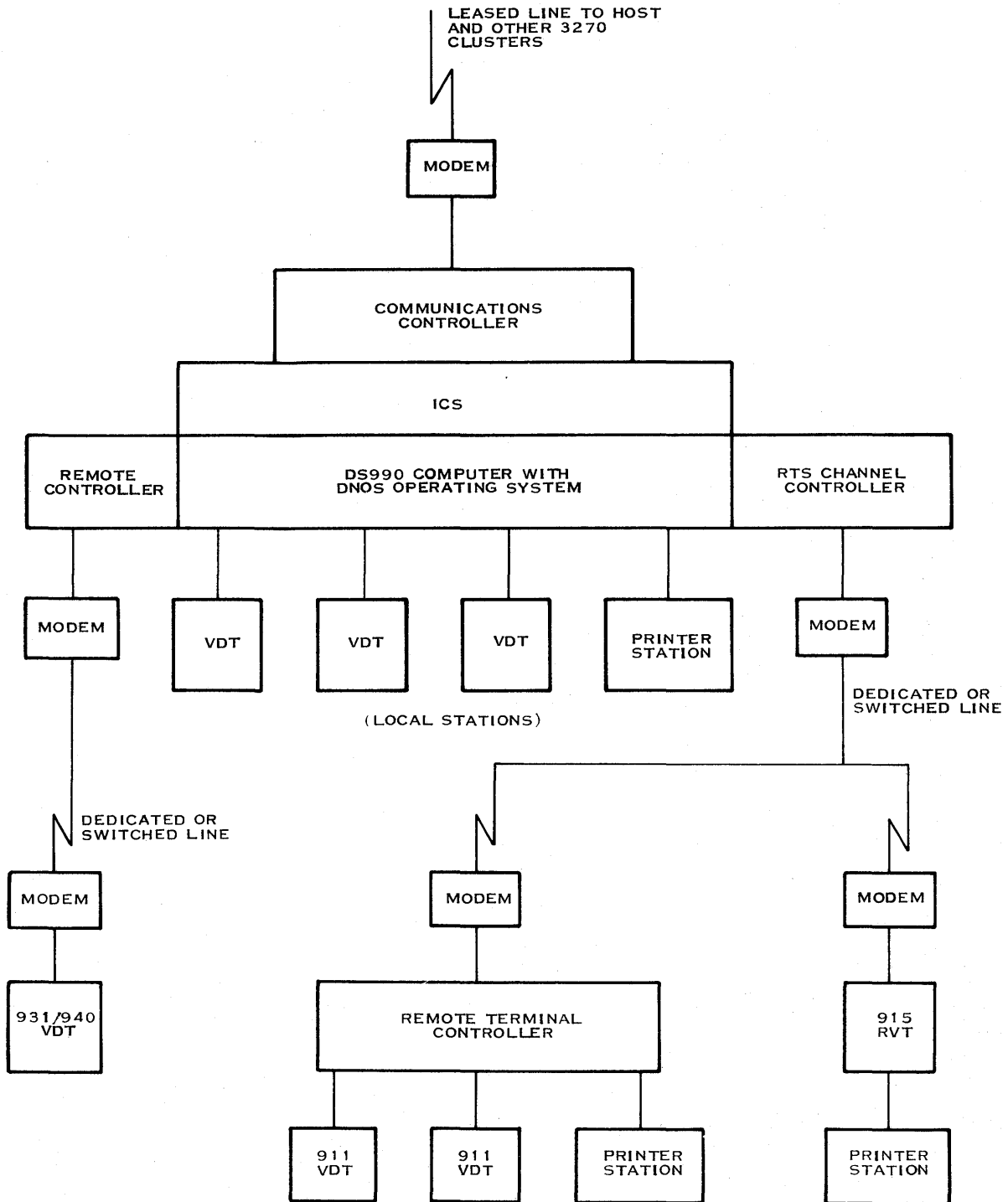
ICS can use several models of video terminals and any printers supported by the DNOS operating system. The video and printer stations can be attached locally or remotely. For example, remote attachment enables the user of a terminal in California to use the resources of a 990 computer in Texas to communicate with a 3270-compatible host computer in Maine. This requires an additional communications link consisting of a communications controller, a modem, and a communications line (usually a telephone line) between the remote terminal and the 990 computer.

Figure 2-1 shows one possible system configuration for the supported video terminals. Video terminals can be attached either locally or remotely, depending upon the system configuration. Terminals that are attached remotely require that the Remote Terminal Subsystem (RTS) be included. RTS is an add-on package that supports video terminals attached to a Remote Terminal Controller (RTC).

2.2.1 Supported Video Terminals

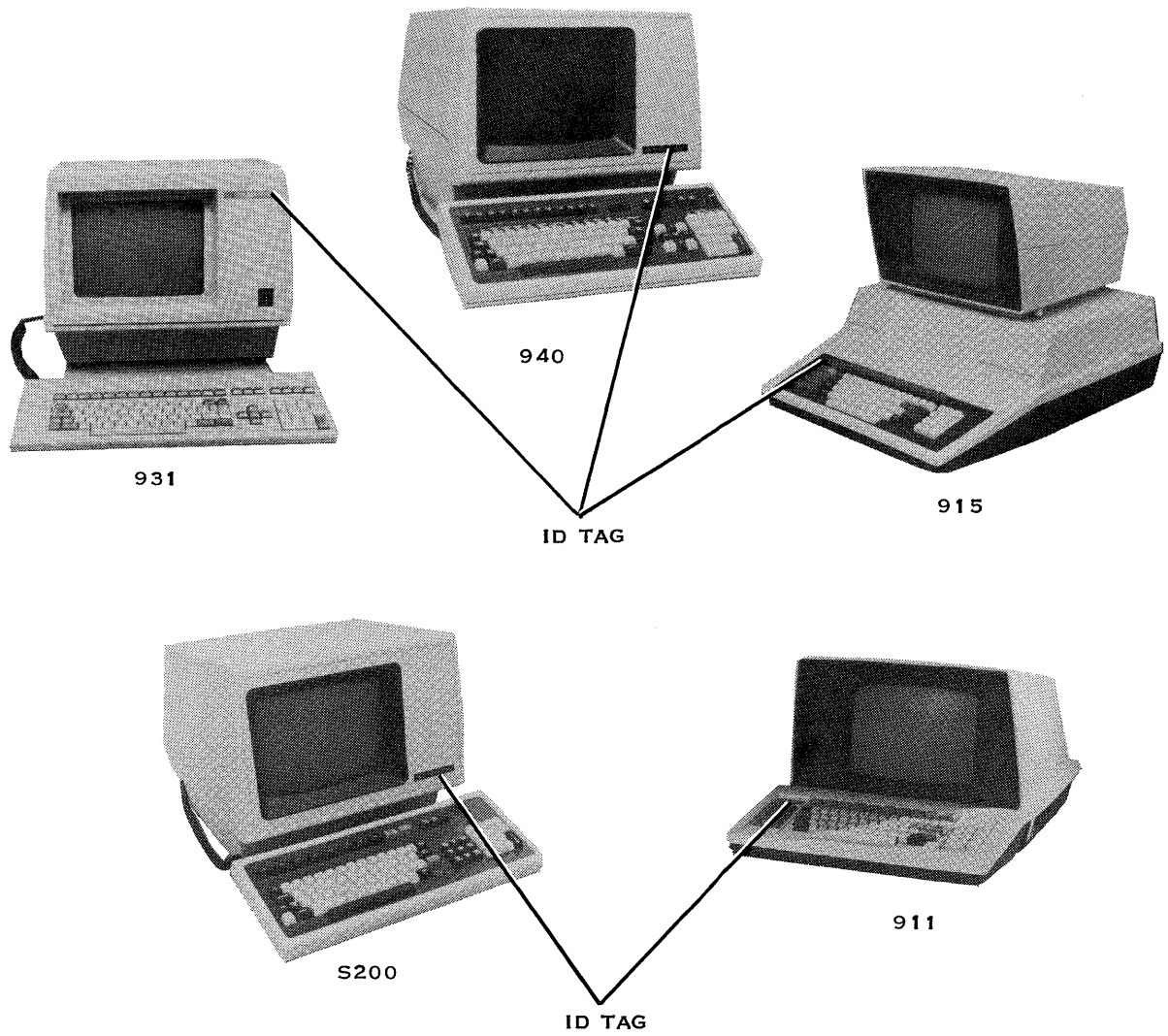
ICS can support several models of video terminals attached locally or remotely. The 911 VDT, 915 RVT, 931 VDT, and the 940 EVT, are all supported by ICS (Figure 2-2). These terminals consist of a video display unit and a cable-attached keyboard.

The 911 VDT and 915 RVT can be identified by the decal in the upper left corner of the keyboard. The 931 VDT and 940 EVT can be identified by the decal on the right side of the display housing.



2283592

Figure 2-1. Typical 990 System Configured With ICS



2282251

Figure 2-2. ICS Supported Video Terminals

ICS supports local or remote attachments of the 911 VDT, 931 VDT, and 940 EVT. The 915 RVT can only be attached remotely and requires that the Remote Terminal Subsystem (RTS) be included. RTS is an add-on package that supports the 915 RVTs. When attached remotely, the 911 and 915 terminals rely on the 990 computer to initially load their memories. Once the download is successfully completed, these terminals operate the same as they do when attached locally. The download is a function of RTS and is required before you can access any application, including ICS. The 931 and 940 do not require downloading.

All terminals operate in the same basic manner, although each terminal may require different key stroke sequences to accomplish these operations. Each terminal can display uppercase and lowercase characters. Some terminals support graphics characters and character sets for different countries.

The 915, 931, and the 940 terminals have their own memory and central processing unit (CPU), but they rely on a 990 computer for their initial program load (IPL), task execution, and file access, plus other operating system resources.

For more information regarding a particular terminal(s), refer to the reference material listed below:

- Model 911 Video Display Terminal Installation and Operations Guide
- Model 915 Remote Video Terminal General Description
- Model 931 Video Display Terminal General Description
- Model 940 Electronic Video Terminal Installation and Operations Manual
- Remote Terminal Subsystem Operators Guide

2.2.2 Remote Configurations

All supported video terminals can be remotely attached. RTS supports the 911 and 915 video terminals. Each installed 940 EVT uses its own communications controller, modem, and communications line. The communications link between the 3270-compatible host and the ICS package installed on a 990 computer must be a dedicated (leased or private) line. The links between ICS and remotely configured terminals, however, may be either dial-up or dedicated. If the link is dial-up, you must call and make the connection to the DNOS operating system before you can download the 915 RVT or the Remote Terminal Controller (RTC). This is the only additional step necessary for remote operation of ICS terminals.

As shown in Figure 2-1, RTS enables the remote attachment of 915 RVTs and RTC clusters to the 990 computer. An RTC supports clusters of 911 VDTs and printer stations. For detailed information about RTS, refer to the *Remote Terminal Subsystem (RTS) Operator's Guide* and the *Remote Terminal Subsystem (RTS) System Programmer's Reference Manual*.

2.2.2.1 Remote Terminal Controller (RTC). Remote VDTs operate under the control of an RTC, which is a 990 Model 5 Computer with built-in communications ports. Each RTC can control up to eight VDTs and up to three printer devices. Supported printer devices include the Model 810 Printer, the Model LQ45 Printer, the Model 820 Keyboard Send/Receive (KSR) Data Terminal, and the Model 743 KSR Data Terminal. Although ICS does not support keyboard input from the 820 and 743 KSRs, ICS can use them as printer stations. Other applications can use these terminals as data entry terminals. RTS supports multiple RTCs and remote video terminals on the same communications channel (port).

Remote video terminals can share an RTS communications channel with RTCs and other RVTs using either a switched or dedicated line. Each RVT can optionally support one Model 810 or Model LQ45 Printer.

VDTs and EVT's are attached remotely through a dedicated controller and cannot share a port with RTS. Each remote terminal installed on a 990 computer must have its own controller. For remote terminals, modems are also required for the 990 computer site and the remote terminal. The communications line can be either a switched (dial-up) or a dedicated (leased) line. With a switched line, more than one terminal can use the communications port (controller), but only one terminal can access it at one time. The line remains busy and inaccessible to other terminals until the first terminal terminates the communications session and releases the line. Only one remote terminal can be attached to a dedicated line. Each remote terminal can optionally support one printer.

2.2.3 Multiple Channel Installation

The 990 computer can support more than one ICS communications line. Figure 2-3 illustrates a 990 computer with two ICS lines installed. Each ICS line requires a separate communications controller port, modem, and leased line. With this configuration, you can access either host computer through any ICS-supported video terminal installed on the 990 computer. Additional ICS channels require additional memory. Each additional ICS channel requires about 4K bytes. The exact memory requirements depend on how many terminal addresses are installed on the line, the hardware used, and other factors. See Appendix C for detailed information.

2.3 3270 IDENTIFIERS

This paragraph describes the function of IBM controller and terminal addresses and how ICS uses them. These values identify the ICS line and individual ICS stations on an ICS line. During installation and operation, ICS uses decimal values for these addresses. The range is 0 through 31. ICS maps these decimal values into EBCDIC values according to Table 2-1. The EBCDIC values are transmitted and received across the line.

Table 2-1. Decimal-to-EBCDIC Address Mapping

Line or Terminal Address (Decimal)	IBM Address (EBCDIC)*	Line or Terminal Address (Decimal)	IBM Address (EBCDIC)*
00	> 40	16	> 50
01	> C1	17	> D1
02	> C2	18	> D2
03	> C3	19	> D3
04	> C4	20	> D4
05	> C5	21	> D5
06	> C6	22	> D6
07	> C7	23	> D7
08	> C8	24	> D8
09	> C9	25	> D9
10	> 4A	26	> 5A
11	> 4B	27	> 5B
12	> 4C	28	> 5C
13	> 4D	29	> 5D
14	> 4E	30	> 5E
15	> 4F	31	> 5F

Note:

The right angle bracket (>) preceding a value indicates that the value is a hexadecimal number.

2.3.1 IBM Control Unit Address

The host computer uses an IBM controller address to identify a particular 3271 control unit or emulator among the 32 possible clusters attached to a communications line. Each 3271 control unit or emulator on the line responds only to its control unit address. Two clusters on the same line should not have the same control unit address. If a duplication occurs, both clusters try to respond to a single host poll. Neither cluster can successfully transmit when a line contention problem exists.

The 990 computer has a control unit address assigned to each installed ICS line. These values need not be unique to the 990 computer. If two ICS lines are installed as shown in Figure 2-3, both can have the same control unit address since each is connected to a different line. The assigned address must be unique to the host.

The host polls each address individually to determine whether any control unit requires servicing. Each control unit or emulator on the line responds when it recognizes its EBCDIC control unit address. ICS maps this EBCDIC value into decimal as shown in Table 2-1. These values range from 0 through 31. The control unit address is assigned to a particular ICS line during ICS installation. Refer to paragraph 4.2 for further information about installation.

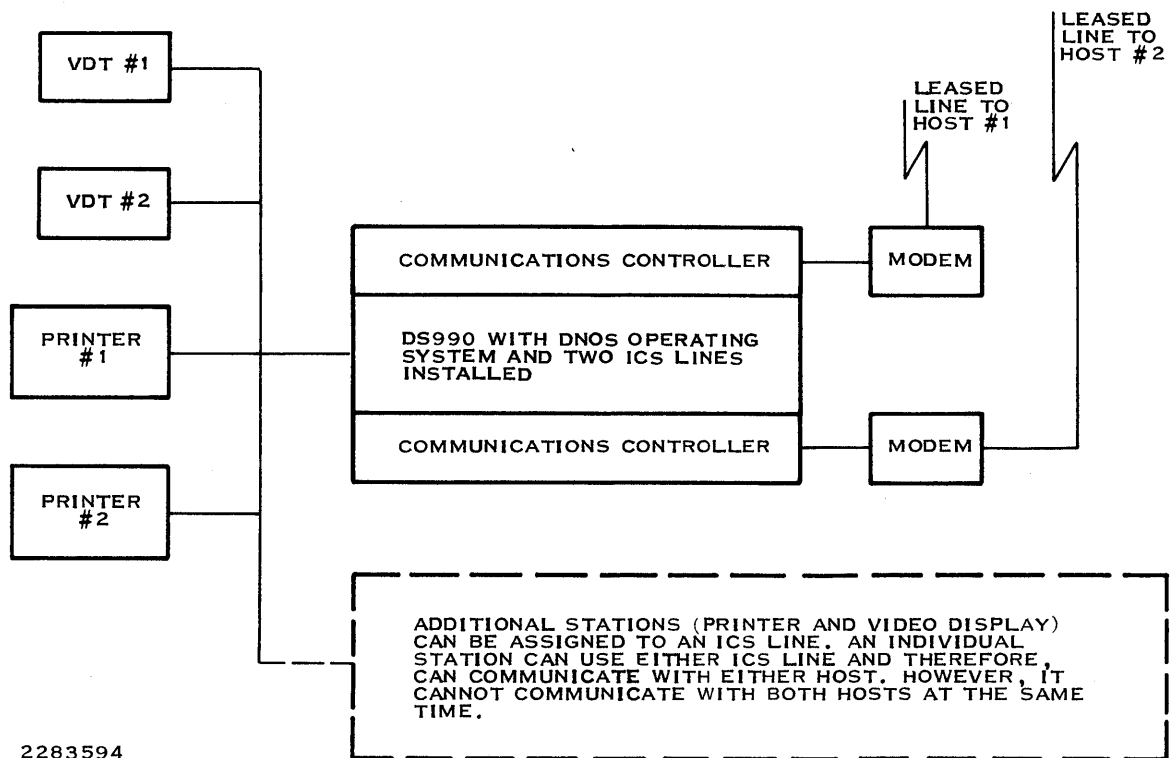


Figure 2-3. Multiple ICS Configuration

2.3.2 IBM Terminal Address

The control unit address identifies a particular cluster on a multidrop line to the host. The terminal address identifies a particular station on that cluster. The number of terminal addresses available to a particular ICS line is determined during ICS installation. If only five terminal addresses are reserved for a particular ICS line during installation, only five ICS stations can be active on that ICS line at a given time. Any installed terminal, however, can be activated as an ICS station on any installed ICS line, provided that not all of the reserved terminal addresses for that line are in use. The number of terminal addresses reserved must be specified during ICS installation.

ICS maps the terminal address from decimal (0–31) to the EBCDIC value used during transmission as shown in Table 2-1.

2.4 REQUIREMENTS FOR ICS OPERATION

The following paragraphs outline ICS hardware requirements and the preliminary software procedures that you must successfully complete before you can use ICS.

2.4.1 Hardware Requirements

ICS requires communications hardware to transmit data between computers and station hardware to present, modify, or create data.

2.4.1.1 Communications Hardware. The communications link from the 3270-compatible host to the 990 computer with ICS requires the following components:

- Communications line — A dedicated (nonswitched) line, either leased or private.
- Modem — A synchronous modem, externally attached to the 990 computer. Where digital lines are available, a data services unit (DSU) can be used instead of a synchronous modem.
- Communications controller — Any one of the following communications controllers (with the appropriate cable):
 - Communications Interface Module CI401 (CIM or COMMIF)
 - Four-Channel Communications Controller CP503 (FCCC)
 - Bit-Oriented, Character-Oriented, Asynchronous Interface Module CP501 or CP502 (BCAIM)

These elements are required to make the connection from the host to ICS. They should be properly installed and tested before you attempt to use ICS. Appendix C provides more detailed information about the components that make the link between the 3270 host and the 990 computer with ICS. Additional hardware components are required to make the link from the 990 computer and remote terminals attached to it by RTS. These components are essentially the same but may also include different types of communications lines for switched facilities. Refer to the *Remote Terminal Subsystem (RTS) Hardware Installation Manual* and the *Remote Terminal Subsystem (RTS) Site Preparation Manual*.

2.4.1.2 ICS Station Hardware. ICS can operate without video terminals or printer stations when a user-written Programmed Station Control (PSC) task is installed on the system. The programmer, however, must use a terminal to generate and debug a PSC task. Most 990 computers are packaged with multiple terminals and at least one printer.

2.4.2 Software Requirements

This paragraph lists the software procedures that you must perform successfully to make ICS operational. Section 3 describes ICS station operation and assumes that all of these preliminary operations are complete. Note that ICS requires access to the System Command Interpreter (SCI). Although this access is automatic in some configurations, generally you must log on to a terminal. This means bidding the SCI task, which handles the interface between you and the operating system. The *DNOS Operations Guide* describes this operation.

The process is more complex when using an RTS terminal. If your RTS uses switched facilities, you must first make the connection to the 990 computer and download the remote terminal software from the 990 computer to the remote device. Exactly how this is done depends on which remote terminal you are using.

Refer to the *Remote Terminal Subsystem (RTS) Operator's Guide*. The following is a list of the preliminary software procedures required for ICS operation:

- RTS installation and generation (if used)
- DNOS operating system installation and generation
- ICS installation and generation

Unless you need to reconfigure your system, you need only perform these steps once. However, you may be required to bid and log on to SCI, depending on your particular configuration, after each initial program load (IPL).

Paragraph 4.2 describes these procedures in more detail, and the *DNOS 3270 ICS Object Installation* document provides complete instructions for ICS installation.

2.5 OPERATING SYSTEM CONCEPTS

The following paragraphs describe features of the DNOS operating system used by ICS. These paragraphs provide information for readers not familiar with the DNOS operating system, including descriptions of the following items:

- Device names
- Pathnames
- Synonyms
- ICS command prompts

The following paragraphs also describe how the ICS procedures handle your responses to SCI command prompts. Paragraph 2.6 provides a complete list of SCI commands unique to ICS installation. Refer to the *DNOS Operations Guide*.

2.5.1 System Command Interpreter (SCI)

SCI is a special task installed on the operating system to handle the interface between you and the resources available to the operating system. Several SCI commands are unique to ICS installation. You use these SCI commands to control ICS.

When SCI is bid successfully, the standard installation yields an SCI prompt followed by the cursor. The SCI prompt is user-defined and typically consists of a pair of brackets, as follows:

[]

NOTE

Avoid using the Hard Break key sequence to terminate tasks in a job. This is especially critical for the job in which the controller task is running.

2.5.2 SCI Command Prompts

When you enter an SCI command at a terminal, a list of command prompts appears on the screen. These command prompts indicate the parameters that you can specify to control the execution of the command. Several prompts require you to respond by entering an access name. An access name can be a device name, a pathname, or a synonym.

2.5.2.1 Device Names. If an access name represents a device, the device name is the name assigned to that device during system generation. The device name can be from one to four characters in length. (Although DNOS accepts eight-character device names, ICS accepts only four-character device names.) The first character must be alphabetic; the others can be alphabetic or numeric. The general form for device names is as follows:

AAxx

where:

AA identifies the type of device.

xx is a decimal number from 01 through 99.

Table 2-2 describes the commonly used identifiers and what they mean.

Table 2-2. Device Name Identifiers

Prefix	Meaning
LP	Line printer
RP	Remote line printer
ST	VDT or ASR/KSR
DS	Disk device
CM	Communications controller port

Two decimal numbers follow the prefix. These numbers identify a specific device among those of the type indicated by the prefix. For instance, CM01 indicates the first communications controller declared during system generation. An ST prefix identifies, for example, the 820 KSR data terminal. Although this terminal has a keyboard, it is not a video terminal, and ICS cannot use it as such. ICS can use it, however, as a printer station.

2.5.2.2 File Pathnames. The pathname of a file is a concatenation of the volume name, the directory levels leading to the file, and the file name itself. Periods separate the components of the pathname. The maximum length of a pathname is 50 characters. The volume, directory, and file identifiers can each be from one to eight characters in length. The first character must be alphabetic, and the others can be alphabetic or numeric. The volume name need not be included if the volume is the system disk. A file name can be concatenated directly to a volume name without requiring a directory name.

Table 2-3 shows examples of valid DNOS pathnames.

Table 2-3. Pathname Examples

Pathname	Description
VOLN.DIR.FILE	The pathname of a file identified by FILE, located in directory DIR, and located on volume VOLN
DS02.DIR.FILE	The pathname of a file identified by FILE, located in directory DIR, and located on device DS02
.DIR.FILE	The pathname of a file identified by FILE, located in directory DIR, and located on the system disk
.FILE	The pathname of a file identified by FILE and located on the system disk
VOLN.FILE	The pathname of a file identified by FILE and located on the volume VOLN

2.5.2.3 Synonym Responses. A synonym is a string of one or more characters that can substitute for a device name or pathname. When SCI prompts you for a device name or a pathname response, you can use a synonym. A synonym can represent an entire pathname or the first components of a pathname. To assign a synonym, use the Assign Synonym (AS) SCI command. See the *DNOS Operations Guide* for more information about synonyms. When SCI prompts you for a device name or filename, you can use a defined synonym in its place.

After you enter a synonym in response to a command prompt, the operating system expands it to its assigned value. Synonyms can save keystrokes, which is important to production operation. Synonyms can also provide unexpected results, however, if you are unaware of their values.

Many SCI procedures use synonyms to pass initial values to command prompts. For instance, the ICS SCI procedures display a prompt for the communications line name. The first time you use one of these procedures at a terminal, no initial value appears for the COMMUNICATIONS LINE NAME prompt. The next time you invoke an ICS procedure at the same terminal, the COMMUNICATIONS LINE NAME prompt displays as its initial value the same value that you entered the last time. To use the displayed initial value, press the RETURN key.

2.5.2.4 ICS Command Prompts. This paragraph describes how ICS interprets your responses to the following command prompts:

- COMMUNICATIONS LINE NAME
- TERMINAL ADDRESS(0 – 31)
- PRINT KEY PATHNAME
- OUTPUT PATHNAME

For some prompts, ICS does not require a response, and SCI accepts a null value. In those instances, ICS uses default values. A null value is a field containing only spaces (blanks). If an initial value carried over from a previous execution appears and you press the RETURN key, the initial value is accepted. This initial value is not a null value; it is treated the same as if you had manually entered the response.

COMMUNICATIONS LINE NAME Prompt. This value identifies the communications line, the communications controller port, and the host computer to ICS. You can enter one of the following values:

- A device name of an installed ICS communications controller port. The general form is CMxx.
- A synonym or logical name for such a device name. ICS uses the value assigned to the synonym.
- A null value (sometimes). When ICS accepts a null value, a default value is used. ICS accepts a null value for the line name in the following command procedures:
 - Execute Interactive Communications Controller (XICC)
 - Execute Interactive Communications Printer (XICP)
 - Execute Interactive Communications Terminal (XICT)
 - Execute ICS Program Station Control Task (XPSC)

If you enter a null value in response to the COMMUNICATIONS LINE NAME prompt when activating an ICS video station, printer station, or PSC task, ICS searches the list of installed ICS lines and selects the first active line that it finds. ICS lines are ordered within this list in the same sequence that you define them when generating the ICS tasks. If two active ICS lines exist and you want to activate a station on the second installed line, you must enter the device name of the line to select it.

If you enter a null value in response to the COMMUNICATIONS LINE NAME prompt when activating an ICS line, ICS searches the list of installed ICS lines and selects the first inactive line that it finds. This is the same ordered list used for auto-selecting the line when activating an ICS video or printer station. If you have only one ICS line installed and you enter a null value in response to the communications line prompt when activating an ICS line or station, ICS always auto-selects the correct line. If the line that ICS auto-selects is already active, ICS displays an error message.

Some ICS utility prompts do not accept null values. In these cases, you must enter a valid identifier.

TERMINAL ADDRESS(0 – 31) Prompt. The terminal address is a decimal number in the range of 0 to 31. ICS displays no initial value for this prompt. When you enter a null value, ICS selects the terminal address in one of the following ways:

- If ICS has been previously activated at that station but is now suspended, ICS selects the terminal address selected during the previous activation. ICS reserves the terminal address of a station when it is suspended.
- If ICS is not in the suspended state at that station, ICS searches a table of terminal addresses and selects the first available address of the appropriate type. If all of the addresses are in use, ICS displays a message to that effect. The type is either PRT (for printer station addresses) or CRT (for video terminal addresses).

PRINT KEY PATHNAME Prompt. The response to the PRINT KEY PATHNAME prompt determines what happens during ICS operation when you press the PRINT key. ICS accepts the following types of responses:

- Printer device names or the system spooler name (for example, LP02)
- File name of a sequential or relative record file
- Name of any sequential device (for example, ST02) supported by the operating system other than a communications controller port
- DUMMY, a specific response that turns off the PRINT key so that nothing happens when you press it
- A synonym or logical name for any of the preceding responses

If you specify a valid printer such as LP03, pressing the PRINT key prints the contents of the screen at LP03. If you enter a valid pathname and then press the PRINT key, ICS writes the contents of the screen to the file specified by the pathname.

If you select a valid pathname, ICS writes the contents of the screen to the file specified by that pathname. If that file does not already exist, ICS creates it automatically, provided that the directory in which it is to reside exists and is not full. If the file already exists, ICS appends the contents of the screen to the existing file each time you press the PRINT key. ICS supports sequential and relative record files.

To change the PRINT key pathname during ICS operation, perform the identification function (paragraph 3.4.4.8).

ICS displays no initial value for the PRINT KEY PATHNAME prompt. The default value for the PRINT KEY PATHNAME prompt is defined by terminal address. The default value for this prompt is the value last specified for the terminal address you use. ICS accepts and evaluates synonyms for this field. Once you assign a value to the PRINT key pathname, that value becomes the default value for this prompt for subsequent activations of the terminal address you use.

OUTPUT PATHNAME Prompt. The response to the OUTPUT PATHNAME prompt identifies the device or file that ICS uses as the ICS printer station. ICS accepts the following types of responses:

- Printer device names or system spooler name (for example, LP02)
- File name of a sequential or relative record file
- Name of any sequential device (for example, ST02) supported by the operating system other than a communications controller port
- A synonym or logical name for any of the preceding responses

The response to the OUTPUT PATHNAME prompt can be any of the responses that ICS accepts as the PRINT KEY PATHNAME prompt. Usually, the output pathname is designated as a file. ICS provides no initial value for this field. If you enter a null value for the OUTPUT PATHNAME prompt, ICS uses the default value. For the first execution, the default value is the value declared during installation. Once a new value is assigned to the OUTPUT PATHNAME prompt, it becomes the default value for subsequent activations of the terminal address you use. However, ICS does not display it as an initial value. The current value for OUTPUT PATHNAME can be obtained by executing the LICC command for the desired communications line, which shows it under PRINTER PATHNAME.

If you enter a file name, ICS writes the output to the file specified. If the file specified does not already exist, ICS creates it, provided that the directory in which it is to reside exists and is not full. If the file already exists, ICS appends the data received from the host to the existing file.

2.6 SCI COMMANDS UNIQUE TO ICS

This paragraph lists all of the ICS-specific System Command Interpreter (SCI) command procedures available on a DNOS operating system that has ICS installed. These procedures are not part of the standard operating system but are incorporated into it when ICS is installed. The commands and their functions are listed, and the specific paragraphs that detail the procedures are referenced, as follows:

- List Interactive Communications Configuration (LICC) — Displays the current installation configuration for a specific communications line name. (See paragraph 4.3.1.)
- List Interactive Communications Statistics (LICS) — Displays the statistical information about an individual ICS line. (See paragraph 4.4.)
- Modify ICS Control Unit Address (MCUA) — Changes the control unit address of an inactive ICS line. (See paragraph 4.3.3.)
- Modify Interactive Communications Configuration (MICC) — Changes the state (enabled/disabled) and the type (CRT or PRT) of a particular terminal address. (See paragraph 4.3.2.)

- Patch ICS PSC User Task (PTPSC) — Applies the latest patches to the PSC Emulator routines that are linked to the user-written PSC task. (See paragraph 5.6.)
- Quit Interactive Communications Terminal (QICT) — Terminates a particular ICS display or printer station. (See paragraph 3.3.2.1.)
- Execute Interactive Communications Controller (XICC) — Activates an ICS communications line. (See paragraph 4.2.2.)
- Execute Interactive Communications Printer (XICP) — Activates an ICS printer station. (See paragraph 3.2.2.)
- Execute Interactive Communications Terminal (XICT) — Activates a particular ICS display station. (See paragraph 3.2.1.)
- Execute Poll Analyzer (XPA) — Provides a listing of buffer activity for the systems analyst. (See paragraph 4.5.)
- Execute Programmed Station Control Task (XPSC) — Activates a user-written program that acts in place of an ICS station. (See paragraph 5.7.)

Under the DNOS operating system, ICS provides a menu that lists all of these commands for quick reference. To display this menu at a video terminal, enter the following characters:

/ICS

Figure 2-4 shows the menu that ICS displays.

```

DNOS ICS 3270 COMMANDS

XICC - EXECUTE INTERACTIVE COMMUNICATIONS CONTROLLER
XICT - EXECUTE INTERACTIVE COMMUNICATIONS TERMINAL
XICP - EXECUTE INTERACTIVE COMMUNICATIONS PRINTER
XPSC - EXECUTE ICS PROGRAMMED STATION CONTROL
LICC - LIST INTERACTIVE COMMUNICATIONS CONFIGURATION
LICS - LIST INTERACTIVE COMMUNICATIONS STATISTICS
MCUA - MODIFY ICS CONTROL UNIT ADDRESS
MICC - MODIFY INTERACTIVE COMMUNICATIONS CONFIGURATION
QICT - QUIT INTERACTIVE COMMUNICATIONS TERMINAL
XPA  - EXECUTE POLL ANALYZER

```

[]

Figure 2-4. ICS Menu

ICS Station Operation

3.1 INTRODUCTION

This section describes how to operate both the display and printer stations of the DNOS 3270 Interactive Communications Software (ICS). It assumes that you have successfully completed all of the preliminary steps listed in Section 2. This section tells you how to activate and terminate an ICS station and how to enter and send data. This section covers the operation of the ICS-supported video terminals, which are referenced individually only in those cases in which they operate differently. Printer operation and error reporting are also detailed. This section describes the System Command Interpreter (SCI) commands that are designed for use by the data entry operator.

ICS allows any ICS-supported video terminal to emulate the operation of IBM Model 2 3277 and 3278 Display Stations. The ICS display station communicates with application programs executing at the host system. An application program can transmit a display to your ICS video station or require you to enter data and transmit it to the host. When the host transmits a display to your station, the display can be formatted or unformatted and can appear with or without data. Then you take one of the following actions:

- Read the data presented (inquiry-only applications)
- Edit the modifiable data display on the ICS display station
- Enter data

Any edited or entered data can then be transmitted to the application program. Data displays are transferred back and forth between the application program and your ICS video station until the data exchange is complete.

ICS allows any DNOS-supported printer or file storage device to emulate the operation of IBM Model 2 3284, 3286, and 3287 Printer Stations. Printer stations only receive data. ICS translates the format-control characters transmitted by the host and either passes them to a printer for a hard copy or writes them in a file.

3.2 ACTIVATING AN ICS STATION

The following paragraphs describe how to activate an ICS station using SCI. The two SCI commands used for this are Execute Interactive Communications Terminal (XICT) and Execute Interactive Communications Printer (XICP). The XICT command activates an ICS video station (type CRT), whereas the XICP activates an ICS printer station (type PRT). Messages associated with the use of these commands are also described.

3.2.1 Execute Interactive Communications Terminal — XICT

To activate ICS at any of the supported video terminals, enter XICT in response to the SCI prompt. Then the following appears on the screen:

```
[ ] XICT
```

```
EXECUTE INTERACTIVE COMMUNICATIONS TERMINAL
COMMUNICATIONS LINE NAME:
  TERMINAL ADDRESS(0-31):
    PRINT KEY PATHNAME:
UNPROTECTED FIELD INDICATOR?: YES
```

3.2.1.1 COMMUNICATIONS LINE NAME Prompt. Your response to the COMMUNICATIONS LINE NAME prompt identifies to ICS which communications line to use for the communications session. Since only one host can be configured per communications line, this response also identifies the host with which ICS is to communicate. This value has a control unit address assigned to it during ICS installation. You must activate the communications line you specify prior to activating a particular station. Paragraph 4.2.2 explains how to activate an ICS line.

Valid line identifiers can be device names of communications controllers or synonyms for such device names.

ICS does not require a response for the COMMUNICATIONS LINE NAME prompt. If you enter a null value, ICS searches a list of the currently active communications lines and selects the active line that is first declared during ICS installation. Paragraph 2.5.2.4 details how this auto-select works.

3.2.1.2 TERMINAL ADDRESS (0 – 31) Prompt. ICS does not require a response to this prompt. If you enter a null value, ICS selects the terminal address. If ICS is in a suspended state at your terminal, ICS uses the address already reserved for that terminal. Otherwise, ICS searches a list of installed 3270 terminal addresses for one that is currently idle and is of the CRT (video) type. ICS assigns the first available address to the ICS station for the duration of the communications session. If all of the installed terminal addresses of the CRT type are in use, ICS displays an error message and does not activate the station. Paragraph 2.5.2.4 details how the auto-select works.

You can specify a particular terminal address. The address you specify must satisfy the following conditions:

- It must be the CRT (video) type.
- It must not be currently assigned to another station.
- It must have been reserved during ICS installation.
- It must not be disabled.

3.2.1.3 PRINT KEY PATHNAME Prompt. Your response to the PRINT KEY PATHNAME prompt determines what happens during ICS operation when you press the PRINT key. If you enter a valid printer device name such as LP02 and then press the PRINT key, ICS prints the contents of the screen at LP02. If you enter a valid pathname and then press the PRINT key, ICS writes the contents of the screen to the file specified by the pathname.

Your response to the PRINT KEY PATHNAME prompt can be any of the following:

- Printer device name or system spooler name (for example, LP02)
- File name of a sequential or relative record file
- Name of any sequential device (for example, ST02) other than a communications controller port
- DUMMY, which is a specific response that turns off the PRINT key so that nothing happens when you press it

If you enter a valid pathname and then press the PRINT key, ICS writes the contents of the screen to the file specified by that pathname. If that file does not already exist, ICS creates it automatically, provided the directory in which it is to reside exists and is not full. If the file already exists, ICS appends the contents of the screen to the data already in the file.

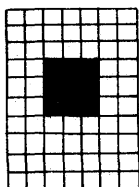
NOTE

If a spooler logical name is assigned as the PRINT key pathname, you must follow the procedure described in Section 8 of the *DNOS ICS Object Installation Guide* for enabling spoolers.

ICS provides no initial value for this prompt. If you enter a null value, ICS uses the default value for the PRINT key pathname. The default value is the value that was specified the last time that any ICS station activated the terminal address that you are now activating. If the terminal address has not been activated since the initial program load (IPL), the default value is the value specified during ICS installation.

You can change the PRINT key pathname during ICS operation by performing the identification function as described in paragraph 3.4.4.8.

3.2.1.4 UNPROTECTED FIELD INDICATOR Prompt. An unprotected field indicator is a special graphics character that ICS uses to mark the beginning of an unprotected field on the screen. You can enter or erase data only when the cursor is in an unprotected field. The character marking the beginning of a field is called the attribute character. Attribute characters cannot be overwritten or erased. If you respond with YES to the UNPROTECTED FIELD INDICATOR prompt, ICS displays each attribute character that marks the beginning of an unprotected field. This shows you where you can modify the screen. The initial value for this prompt is YES. When you accept this value (or enter YES on the keyboard), ICS displays the special graphics character shown in Figure 3-1 immediately to the left of the beginning of every unprotected field on the screen.



2282169

Figure 3-1. Unprotected Field Indicator

NOTE

Some early versions of 911 and 915 terminals do not include graphics support. If your terminal does not support these graphics characters, the unprotected field indicator still appears as a lighted rectangle, the unprotected field indicator still appears as a lighted rectangle except that it is larger than that shown in Figure 3-1. It looks like a cursor displayed in low intensity.

If you position the cursor immediately to the right of any of these characters, the cursor does not blink, indicating that ICS accepts your data entries or erasures. If you position the cursor within a protected field, the cursor blinks. If you press a data or erase key while the cursor is in a protected field, the audible alarm sounds. If you respond NO to the UNPROTECTED FIELD INDICATOR prompt, ICS displays no attribute characters. The only responses that ICS accepts for the UNPROTECTED FIELD INDICATOR prompt are as follows:

- YES
- Y — Short for yes
- NO
- N — Short for no

3.2.1.5 Examples of ICS Video Terminal Activation. This paragraph shows two examples of ICS video terminal activation. In the first example, you enter CM02 in response to the COMMUNICATIONS LINE NAME prompt. You do not enter the terminal address, and ICS selects one automatically(See paragraph 2.5.2.4.) The response to the PRINT KEY PATHNAME prompt is LP03, indicating that when you press the PRINT key, ICS prints the contents of the screen at LP03. ICS accepts the initial value for the UNPROTECTED FIELD INDICATOR prompt, YES.

EXAMPLE 1

[] XICT

```

EXECUTE INTERACTIVE COMMUNICATIONS TERMINAL
COMMUNICATIONS LINE NAME: CM02
TERMINAL ADDRESS(0-31):
PRINT KEY PATHNAME: LP03
UNPROTECTED FIELD INDICATOR?: YES

```

In the second example, you enter only the terminal address. ICS uses the first installed ICS line that is currently active for the communications line name and accepts the initial value provided for the UNPROTECTED FIELD INDICATOR prompt (YES). ICS uses the default value for the PRINT KEY PATHNAME prompt. This default value is whatever value that was used the last time that the terminal address was activated. If the terminal address has not been activated since the last IPL, the default value is the initial default value that you declare during ICS installation. The List Interactive Communications Configuration (LICC) output listing shows the current default value for each terminal address.

EXAMPLE 2

[] XICT

```

EXECUTE INTERACTIVE COMMUNICATIONS TERMINAL
COMMUNICATIONS LINE NAME:
TERMINAL ADDRESS(0-31): 5
PRINT KEY PATHNAME:
UNPROTECTED FIELD INDICATOR?: YES

```

3.2.1.6 Verifying Video Terminal Activation. If the activation is successful, the cursor appears in the upper left corner of the screen. The rest of the screen is blank unless the ICS station was in a suspended state prior to activation. When you activate a suspended station, the screen appears exactly as it did prior to suspension.

If an error occurs, ICS clears the screen and writes an error message on the bottom line. Parts of these messages are variables whose content depends on your responses to ICS command prompts. The following list shows typical error messages generated by an unsuccessful attempt to activate an ICS station:

```

*XICT-CM01 IS NOT AN ICS COMMUNICATIONS LINE:
*XICT-HIGHEST TERMINAL ADDRESS IS 06:
*XICT-TERMINAL ADDRESS 02 BELONGS TO A PRINTER:
*XICT-ALL ICS VIDEO ADDRESSES ARE IN USE:
*XICT-CM01 HAS NOT BEEN ACTIVATED BY "XICC":
*XICT-INVALID PRINTER PATHNAME LP12:

```

See Appendix B for an explanation of these and other messages, the source of the variable parts, and the recommended corrective action.

3.2.2 Execute Interactive Communications Printer — XICP

An ICS printer station can either print data from an ICS line or copy the data onto a file or sequential device. To activate an ICS printer station, enter XICP in response to the SCI prompt. ICS displays the following prompts:

```
[ ] XICP  
  
EXECUTE INTERACTIVE COMMUNICATIONS PRINTER  
COMMUNICATIONS LINE NAME:  
TERMINAL ADDRESS(0-31):  
OUTPUT PATHNAME: LP$1
```

3.2.2.1 COMMUNICATIONS LINE NAME Prompt. Your response to the COMMUNICATIONS LINE NAME prompt identifies to ICS which communications line to use for the communications session. Since only one host can be configured per communications line, this response also identifies the host with which ICS is to communicate. This value has a control unit address assigned to it during ICS installation. The communications line that you specify must already be active. Paragraph 4.2.2 explains how to activate an ICS line.

Valid line identifiers can be device names of communications controller ports or synonyms for such device names.

ICS does not require a response to the COMMUNICATIONS LINE NAME prompt. If you enter a null value, ICS searches a list of the currently active communications lines and selects the active line that was first declared during ICS installation.

3.2.2.2 TERMINAL ADDRESS(0 – 31) Prompt. The TERMINAL ADDRESS(0 – 31) prompt allows you to specify a particular terminal address. If you enter a null value in response to this prompt, ICS assigns the first available terminal address of the PRT (printer) type. The address you specify must satisfy the following conditions:

- It must be the PRT (printer) type.
- It must not be currently assigned to another station.
- It must have been reserved during ICS installation.
- It must not be disabled.

3.2.2.3 OUTPUT PATHNAME Prompt. The response to the OUTPUT PATHNAME prompt can be any of the following identifiers:

- Printer device name or the system spooler name (for example, LP02)
- File name of a sequential or relative record file
- Name of any sequential device name (for example, ST02) other than a communications controller port

NOTE

If a spooler is assigned as the OUTPUT PATHNAME, you must follow the procedures described in Section 8 of the *DNOS ICS Object Installation* for enabling the spooler.

If you enter a file name, ICS writes the data to the specified file. If the file specified does not already exist, ICS creates it, provided the directory in which it is to reside exists and is not full. If the file already exists, ICS appends the new data after the data is in the file.

If you enter a null value, ICS assigns the default value as the output pathname. For the initial execution of this command, the default value is the value declared during installation. For subsequent executions, ICS uses the value assigned during the previous execution as the default value.

3.2.2.4 Examples of ICS Printer Station Activation. This paragraph shows two examples of ICS printer station activation. In the first example, you specify CM02 as the communications line name. You specify no value for the terminal address, so ICS selects one. You enter LP02, so the data is printed at LP02.

EXAMPLE 1

[] XICP

```
EXECUTE INTERACTIVE COMMUNICATIONS PRINTER
COMMUNICATIONS LINE NAME: CM02
TERMINAL ADDRESS(0-31):
OUTPUT PATHNAME: LP02
```

The next example shows no values entered for the COMMUNICATIONS LINE NAME or TERMINAL ADDRESS prompt, so ICS selects these values. Since the response to the OUTPUT PATHNAME prompt is VOL.DIR.FILE, ICS writes the data to the file named FILE residing under the directory DIR which is on the volume VOL.

EXAMPLE 2

[] XICP

```
EXECUTE INTERACTIVE COMMUNICATIONS PRINTER
COMMUNICATIONS LINE NAME:
TERMINAL ADDRESS(0-31):
OUTPUT PATHNAME: VOL.DIR.FILE
```

3.2.2.5 Verifying Printer Station Activation. If ICS detects any errors, it displays an error message and does not perform the activation. Determine the problem, correct it, and reenter the command. If ICS reports no errors, you can assume that output has been properly directed to the designated device. If you enter a null value in response to the TERMINAL ADDRESS(0-31) prompt, ICS selects the terminal address and displays the following type of message at your terminal:

```
*XICP- TERMINAL ADDRESS xx ON COMMUNICATIONS LINE yy ACTIVATED:
```

The following list shows error messages commonly generated during an unsuccessful activation attempt:

```
*XICP-CM01 IS NOT AN ICS COMMUNICATIONS LINE:
*XICP-HIGHEST TERMINAL ADDRESS IS 06:
*XICP-TERMINAL ADDRESS 02 BELONGS TO A VIDEO TERMINAL:
*XICP-ALL ICS PRINTER ADDRESSES ARE IN USE:
*XICP-CM01 HAS NOT BEEN ACTIVATED BY "XICC":
*XICP-INVALID PRINTER PATHNAME LP12:
```

See Appendix B for an explanation of these and other messages, the source of the variable parts, and the recommended corrective action.

3.3 EXITING ICS

When an ICS station is activated, control of the station passes to ICS. You can suspend an ICS video station without terminating it. You can terminate any active or suspended ICS station with the Quit Interactive Communications Terminal (QICT) command. You can also terminate an ICS video station directly. Exactly which keystrokes accomplish these actions depends on the terminal being used.

3.3.1 Suspending ICS

You can suspend ICS at a video station after it has been activated. Once suspended, you must reactivate the station by entering XICT before it can be used again. Suspending ICS at a station does not release the terminal address allocated during activation.

You cannot suspend ICS printer stations. You can only terminate them.

If you enter no value for the terminal address when you are reactivating ICS at a suspended station, ICS uses the terminal address that is still reserved for that station. If you enter a different value for the terminal address when reactivating, ICS allocates the additional address, if possible. This process is useful when an ordinary transaction is interrupted to process a priority order.

When you suspend ICS, SCI is reactivated. The default menu for SCI, the SCI prompt, and the cursor (immediately to the right of the prompt) appear on the screen. Once SCI is reactivated, you can execute any of the command procedures available from the operating system. When you wish to reactivate your video station, enter the following characters:

```
[ ] XICT/
```

When you initially activated the station, you supplied all of the necessary values in response to each prompt. The slash (/) after XICT is an SCI primitive command that allows you to use the initial values without pressing the RETURN key for each prompt. In this instance, after you enter XICT/ and press the RETURN key once, SCI uses the same values that you entered initially and activates ICS without stopping the cursor at each prompt.

NOTE

Entering any command and then pressing the ENTER key sequence performs the same function as the slash (/) primitive. That is, SCI uses the existing initial and default values to execute the command without stopping the cursor at each prompt.

3.3.2 Terminating ICS

You can terminate an active ICS display station in two ways. If SCI is active, execute the QICT command procedure. If ICS is active at the station, use the terminate function keys. (See paragraph 3.3.2.2.) You can also terminate an active ICS printer station using the QICT command procedure. Terminating an ICS station releases the terminal address for use by another station.

3.3.2.1 Quit Interactive Communications Terminal — QICT. The QICT command can terminate any active ICS station. To terminate either an ICS printer or video terminal communications session, enter QICT in response to the SCI prompt. The following prompts appear on the screen:

```
[ ] QICT
```

```
QUIT INTERACTIVE COMMUNICATIONS TERMINAL
COMMUNICATIONS LINE NAME:
  TERMINAL ADDRESS(0-31):
  TERMINAL TYPE(CRT/PRT): CRT
  TERMINAL PATHNAME: STxx
```

The COMMUNICATIONS LINE NAME prompt is the same value specified during activation. Usually, the initial value supplied for this prompt is the one you use. The only time you may need to use a value other than the one supplied is when two ICS lines are installed and you use both of them at your terminal. If no value is entered < Null>, ICS defaults to the first active installed line.

You must respond to the TERMINAL ADDRESS(0 – 31) prompt only when more than one terminal address is assigned to the station designated by the TERMINAL PATHNAME prompt. If you specify the terminal address, you do not have to specify the terminal pathname.

The initial value for the TERMINAL TYPE prompt is CRT, which specifies an ICS video station. When terminating an ICS printer station, change this to PRT.

The initial value of the TERMINAL PATHNAME prompt is the device name of the station at which you entered QICT. Specify either the terminal device name (or a synonym) or the terminal address. Both are not required unless more than one terminal address is assigned to the terminal specified. You may specify both. If ICS determines, however, that the terminal name you entered does not have the specified terminal address assigned to it, ICS displays an error message and takes no further action.

When you terminate ICS using the QICT command, ICS displays a configuration listing (described in paragraph 4.3.1). SCI returns, displaying the SCI prompt and cursor on the bottom line.

In the following example, an operator at ST01 (a display terminal) terminates the ICS station active at ST05.

EXAMPLE 1

```
[ ] QICT
```

```
QUIT INTERACTIVE COMMUNICATIONS TERMINAL
COMMUNICATIONS LINE NAME: CM02
  TERMINAL ADDRESS(0-31):
  TERMINAL TYPE(CRT/PRT): CRT
  TERMINAL PATHNAME: ST05
```

If only one terminal address is assigned to ST05, ICS terminates at ST05, clears the screen there, and displays the following message:

```
*XICT-TERMINATED BY ANOTHER USER OR HARD BREAK KEY
```

If the termination is successful, the system displays a LICC message at the terminal at which you entered QICT. If the termination is unsuccessful, ICS displays a message at the terminal where you entered QICT. For instance, if ST05 is not an active (or suspended) ICS station, the following message appears at ST01:

```
*QICT-ICS NOT RUNNING AT ST05:
```

The following example demonstrates how to terminate an ICS printer station.

EXAMPLE 2

```
[ ] QICT
```

```
QUIT INTERACTIVE COMMUNICATIONS TERMINAL
COMMUNICATIONS LINE NAME: CM02
TERMINAL ADDRESS(0-31):
TERMINAL TYPE(CRT/PRT): PRT
TERMINAL PATHNAME: LP02
```

If the termination is successful, the system displays a LICC listing at the terminal where you entered QICT. If for any reason the termination is unsuccessful, a message appears at the terminal where you entered QICT. For instance, if two addresses of the same terminal type are assigned to LP02, the following message appears at the terminal where you entered QICT:

```
*QICT-TERMINAL ADDRESS NEEDED:
```

When two terminal addresses are assigned to one ICS device, you must specify the terminal address. When you specify the terminal address, you need not specify the terminal pathname.

Under normal operation, when only one terminal address is assigned to your ICS video station, you can use the SCI primitive form with QICT to terminate your station, as follows:

```
[ ] QICT/
```

The slash (/) specifies that SCI uses the initial values assigned to the prompts without stopping the cursor at each prompt. When you press the RETURN key, SCI terminates the ICS video station, except when more than one terminal address is assigned to your station or if you have deleted or modified your synonyms. You should also include the QICT command in the M\$01 command procedure. This ensures release of any suspended ICS station when you log off of SCI. Refer to the *DNOS ICS Object Installation Guide* for details.

3.3.2.2 Terminating ICS at an Active Station. To terminate ICS at an active ICS video station, press the Exit key on your keyboard, release it, and then press Q.

When you press the Exit key, the cursor blinks without moving. This indicates that ICS is waiting for you to press another key. If you press the Exit key again (instead of Q), the cursor stops blinking and your station remains active. When you press the second key (Q), ICS terminates and clears the screen.

NOTE

Throughout this manual, the notation in the format Exit + n indicates that you press and release the Exit key, then press the key indicated by n. You do not actually enter the plus sign (+) that immediately follows Exit.

In all cases where the Exit key is the first key and an alphabetic key is the second key, the alphabetic key performs the desired function whether it is uppercase or lowercase. For example, Exit + Q is functionally equivalent to Exit + q.

Finally, the notation in the format (x)y indicates that you press and hold the key indicated by x, then press the key indicated by y. For instance, (SHIFT)ALT indicates that you press and hold the SHIFT key, then press the ALT key. Where a key is labeled with two characters (such as 1 and !) and performs the same as a standard typewriter key, this manual omits this notation. For instance, ! represents the exclamation point, rather than (SHIFT)1.

When you terminate ICS using the Exit + Q sequence, ICS clears the local format file. When you reactivate a terminal address that was terminated by this key sequence, the screen is blank.

3.4 VIDEO DISPLAY TERMINAL OPERATION

The following paragraphs detail the operation of the video terminals as ICS stations. Note that the description that follows is valid only when ICS controls the station. When other applications control these devices, the keyboard can function differently.

These paragraphs also describe the displays, data entry, special functions, and program attention keys.

3.4.1 Keyboard Layout

ICS-supported video terminals are interactive display terminals with keyboards. These keyboards have the following groups of keys:

- Alphanumeric typewriter keyboard
- Numeric pad
- Cursor control and edit key pad
- Function keys

ICS terminals emulate every key on the IBM 3277 typewriter keyboard, except the TEST REQ key. Some of the 3277 keys are emulated by two-key functions on ICS terminals, and some are labeled differently. Table 3-1 lists all of the keys that are identical on ICS and IBM 3277 keyboards. Some of these keys may be located in a different position, but they perform the same function and are labeled the same. Table 3-2 lists single keys that perform the same functions but are labeled differently.

Table 3-1. Identical ICS/3277 Keys

Key(s)	Description	Key(s)	Description
A – Z	Alphabetic	+	Plus sign
0 – 9	Numeric	!	Exclamation point
SHIFT	SHIFT	–	Hyphen
	Cursor up	=	Equal sign
	Cursor down	_	Underscore
	Cursor right	;	Semicolon
	Cursor left	:	Colon
ERASE INPUT	Erase input	“	Quotation mark
ENTER	Send to host	'	Apostrophe
	At sign	<	Less-than sign
	Pound sign	,	Comma
\$	Dollar sign	>	Greater-than sign
%	Percent sign	.	Period

Table 3-1. Identical ICS/3277 Keys (Continued)

Key(s)	Description	Key(s)	Description
&	Ampersand	?	Question mark
*	Asterisk	/	Slash
(Left parenthesis	Space bar	Blank
)	Right parenthesis		

Table 3-2. ICS/3277 Single Key Equivalents

IBM 3277 Key(s)	Generic Key(s)	Key Function
CLEAR	Initialize Input	Clear screen
ENTER	ENTER	Send to host
LOCK	UPPERCASE LOCK	Uppercase alphabetic
	Next Field	Field right
	Previous Field	Field left
	RETURN	New line
INS MODE	Insert Character	Insert character
DEL	Delete Character	Delete character
PF1-PF8	F1-F8	Program function 1 through 8
(vertical bar)]	Right bracket
] (logical NOT)	^	Caret
¢ (cent sign)	[Left bracket

ICS emulates several IBM 3277 typewriter keys by using a two-key sequence. Table 3-3 lists these IBM keys and the two-key ICS equivalent. The first key in each sequence for the 911, 915, and 940 terminals is the Exit key. When you press the EXIT key, the cursor does not move but begins to blink, indicating that ICS is waiting for you to press another key. If you press the EXIT key again, the cursor stops blinking and you can continue to enter data or edit the contents of the screen (the cursor does not blink on a 940 EVT or 931 VDT).

Table 3-3. ICS Two-Key Sequences That Emulate a 3277 Key

3277 Key(s)	Generic Key Sequence(s)	Function
ALPHA	Exit + A	Allow the entry of nonnumeric input into numeric fields
DUP	Exit + D	Duplicate previous field (remote)
FLD MARK	Exit + F	Field mark
IDENT	Exit + I	Identification
PA1	Exit + + (plus)	Program access key 1 ¹
PA2	Exit + - (hyphen)	Program access key 2 ²
PA3	Exit + _ (underscore)	Program access key 3 ³
PF1 through PF12	Exit + 01 through Exit + 12	Program function keys ⁴
RESET	Exit + R	Reset keyboard

Notes:

¹ This key sequence on a 940 EVT is (ALT)P1

² This key sequence on a 940 EVT is (ALT)P2

³ This key sequence on a 940 EVT is (ALT)P3

⁴ This key sequence on a 940 EVT is F1 through F12

ICS performs several functions not supported by IBM 3277 typewriter keyboards. Table 3-4 shows ICS key functions that have no IBM equivalent.

Table 3-4. ICS Keys Without IBM 3277 Equivalents

Generic Key Sequence(s)	Function
Command	CMD Suspend ICS station
Exit + E	Erase to end-of-screen (EOS)
Exit + L	Locally duplicate previous field
Exit + M	Switch tab mode TABS and WORD mode
Exit + P	Append the cursor to the end of a word
Exit + Q	Terminate ICS video station
Exit + 13 through Exit + 24	Program function keys PF13 through PF24 (these are 3278 keys) ¹
Exit + Exit	Cancel Exit mode
ERASE FIELD	Erase entire field (including data to the left of the cursor)
HOME	Cursor to the first character position
PRINT	Print screen
PRINT + PRINT	Cancel print
REPEAT	Emulate an extension of the IBM typamatic feature ²
SKIP	Erase to the end-of-field and tab (can be optionally switched with TAB)
TAB	Tab according to field, depending on the tab mode (can be optionally switched with SKIP)

Notes:

¹ The key sequence on a 940 EVT is PF13 through PF24

² Data and cursor control keys are typamatic

3.4.2 Displays

After ICS is activated and the ICS display station is interacting with the host application program(s), a display, ready for editing or data entry, appears on the screen. The display may be formatted or unformatted. A formatted display is organized into fields; an unformatted display has no fields.

You should be familiar with the operating instructions of the application before altering the display.

The data for display is stored in a 1920-character buffer and is displayed in 24 lines of 80 characters each. Each buffer location is referenced by its buffer address. The buffer is addressed from 0 (the upper left corner character position) through 1919 (the bottom right corner character position). The character positions on the display have a direct relationship to the buffer positions. The data you enter on the keyboard updates the buffer and appears on the screen as it is entered.

These video terminals display both uppercase and lowercase letters. Under ICS control, these terminals support the 89-character EBCDIC character set shown in Appendix D.

3.4.2.1 Formatted Displays. A formatted display is divided into fields. The host application program defines each field with an attribute character. The attribute character determines the characteristics of the field. Usually, an attribute character appears as a blank preceding the first character position of a field. ICS provides the option of displaying attribute characters that mark the beginning of unprotected fields. (See paragraph 3.2.1.4.) The attribute character can define a field in the following ways:

- Protected or unprotected. You cannot enter or erase data when the cursor is located in a protected field. You can modify an unprotected field. ICS optionally provides the display of a special graphics character to mark the beginning of unprotected fields.
- Alphanumeric or numeric. You can enter alphabetic, numeric, or special characters in an alphanumeric field. Numeric fields are reserved for numeric input and accept only the following characters:
 - 0 through 9
 - The minus sign (–)
 - The decimal point (.) (The comma replaces the decimal point on some international keyboards.)
- Display, nondisplay, or highlighted display. This characteristic determines whether the fields are displayed and, if so, specifies high-intensity or low-intensity display.
- Modified or unmodified. This characteristic determines whether the ICS display station operator or host application program has modified the field. This is the only attribute characteristic that operator input changes. Editing or entering data into an unprotected field automatically changes the attribute character for that field from unmodified to modified.

3.4.2.2 Unformatted Displays. With an unformatted screen, you use the keyboard to edit existing data or to enter new data without the restrictions imposed by defined fields. An unformatted display contains no attribute characters. It is assumed that the characteristics of the display are unprotected, alphanumeric, and display. ICS treats an unformatted display as one large field.

3.4.2.3 Cursor. The cursor appears as a lighted rectangle that occupies a character position on the screen. The cursor can share a character position with a character. The data sharing a character position with the cursor is not erased but is displayed in reverse video (silhouetted) to indicate the cursor location. The position of the cursor indicates where the next entered character is displayed.

Typically, the cursor is initially positioned in the first position of the first unprotected field of a formatted screen or in the upper left corner of an unformatted screen.

Normal movement of the cursor on the display is continuous from left to right and from top to bottom. The cursor wraps from the bottom of the screen to the top of the screen. The alphanumeric keys on the keyboard move the cursor one character position at a time from left to right. Special keys move the cursor without changing any data on the display. The cursor movement for these special keys is described later in this section.

Certain ICS functions require a two-key sequence. The first key of these functions is the Exit key. Once you press this key, the cursor begins to blink. It continues to blink until you press another key. A blinking cursor also occurs in the bottom left of the screen if the keyboard is locked or if a transmission is pending.

3.4.2.4 Display Messages. ICS displays a message on the bottom line of the screen when it detects an error. If you pressed a program attention key immediately prior to the message display, you should assume that the transmission did not take place. When an error occurs, the ICS station either suspends or terminates, depending on the type of error. Further information about these messages can be found in Appendix B.

Some errors cause the keyboard to lock, and a blinking cursor appears on the bottom line of the screen in the first character position. Either the host application or the ICS station operator can reset the keyboard. When the keyboard is reset, you can continue to enter data or edit the contents of the screen. ICS redisplay the original bottom line and repositions the cursor to its previous location.

3.4.3 Special Features

Two special features, automatic skip and audible alarm, aid in data entry and editing a display. Automatic skip speeds up the editing and entry of data by skipping protected fields. The audible alarm sounds when you use keys improperly. Also, some host applications may sound the alarm before updating the screen.

3.4.3.1 Automatic Skip. When you enter a character into the last character position of an unprotected field, the cursor automatically skips any protected fields and stops at the first character position of the next unprotected field. All protected fields are skipped.

If the cursor is in the last character position of the last unprotected field of a display and you press a data key, the cursor wraps around to the first character position of the first unprotected field.

3.4.3.2 Audible Alarm. An audible alarm sounds when you press a data key or an erase key in any of the following situations:

- The cursor is positioned on an attribute character or in a protected field.
- The keyboard is locked.
- The keyboard is in the insert mode, but the field is already full.
- A hardware error is detected on the video terminal.
- The terminal receives some host application screens.

The audible alarm also sounds when you position the cursor over an attribute character or in a protected field and attempt to perform an erase (by pressing a key such as SKIP, ERASE FIELD, or ERASE INPUT).

3.4.4 Key Functions

The alphabetic, numeric, and special character keys operate as typewriter keys. Other keys perform functions unique to ICS. The following paragraphs describe these keys. Table 3-5 shows which keys on each ICS-supported video terminal emulate keys on an IBM 3277 Model 2 Display Station equipped with an EBCDIC typewriter keyboard. The following list shows the types of key functions supported by ICS:

- Repeat
- Cursor movement
- Field movement
- Tab
- Editing
- Suspend
- Terminate
- Print
- Reset
- Duplicate
- Field mark
- Alpha override
- Program attention

3.4.4.1 Repeat Function. The REPEAT key performs the repeat function on the 911 and 915 terminals by emulating an extension of the IBM typamatic feature. An IBM typamatic key repeats when you hold it down. The data keys and the cursor control keys on the 931 and 940 terminals are also typamatic. The keys on the numeric pad on these terminals are not typamatic. Only the space bar and the cursor movement keys are typamatic on IBM keyboards. The REPEAT key on a 911 or 915 ICS display station works in conjunction with all data keys as well as with the cursor movement keys. The REPEAT key causes the next key pressed to repeat as long as the REPEAT key is held down. To use the repeat function, you must press the REPEAT key before you press the key whose action is to be repeated. The only keys that do not repeat are mode keys that produce no key codes themselves. These keys generate code only when used in conjunction with a data key.

The following list shows mode keys that do not repeat:

- SHIFT
- CONTROL
- UPPERCASE LOCK
- INSERT CHARACTER

3.4.4.2 Cursor Movement Functions. The Previous line (↑), Next line (↓), left arrow (←), right arrow (→), Next CHAR (←), and Next CHAR (→) keys move the cursor one character position at a time. These keys can move the cursor into any position on the screen, without concern for attribute characters or protected fields. Using these keys in a field does not mark the field as modified. These keys do not move the cursor off the screen. If the cursor is already at an edge of the screen, the cursor wraps horizontally and vertically as follows:

- When you move the cursor off the screen to the left of the first character position while on lines 2 through 24 (the bottom line), it reappears in the eightieth character position of the line above.
- When you move the cursor off the screen to the left of the first character position while on line 1, it reappears in the eightieth character position of the bottom line.
- When you move the cursor off the screen to the right of the eightieth character position while on lines 1 through 23, it reappears in the first character position of the line below.
- When you move the cursor off the screen to the right of the eightieth character position while on the bottom line, it reappears in the first character position of line 1.
- When you move the cursor vertically below line 24 (the bottom line) of the screen, it reappears on line 1 in the same character position it occupied on line 24.
- When you move the cursor vertically above line 1 of the screen, it reappears on the bottom line in the same character position it occupied on line 1.

The field movement, SKIP, and RETURN keys move the cursor left and right in the same manner except that they position the cursor in the first character position of unprotected fields only. The skip function is a combination of the tab and erase to end-of-field functions. An option is available that switches the function of the tab and skip keys on the 911 and 915 terminals. Consult your systems operator to determine whether this option is enabled.

3.4.4.3 Field Movement Functions. The left-field, right-field, home-cursor, and new-line functions move the cursor by whole fields, according to the fields on a screen.

Field Left Function. The field-left operation moves the cursor to the first position of the first unprotected field to the left of and/or above the position of the cursor. If there are no unprotected fields, the cursor moves to the first character position (upper left corner) of the screen. Table 3-5 shows which keys perform the field-left function for each ICS-supported terminal.

Field-Right Function. The field-right operation moves the cursor to the first position of the first unprotected field to the right of and/or below the position of the cursor. If there are no unprotected fields, the cursor moves to the first character position of the screen. Table 3-5 shows which keys perform the field-right function for each ICS-supported terminal.

Cursor Home Function. The HOME key moves the cursor to the first position of the first unprotected field of a formatted screen or to the first character position of an unformatted screen.

New Line Function. The RETURN key moves the cursor to the first position of the first unprotected field on a subsequent line. If there are no unprotected fields, the cursor moves to the first character position on the screen. If there are no fields, the cursor moves to the first character position of the next line.

3.4.4.4 Tab Functions. ICS provides two modes of tabbing: TABS and WORD. ICS lets you alternate between WORD mode and TABS mode. You can also set fixed tabs for use in the tab mode. Finally, ICS provides a special type of tab that positions the cursor at the end of a word. ICS provides the following tab function keys:

- TAB — Moves the cursor forward to a position determined by which mode is active:
 - TABS mode positions the cursor according to preset tabs
 - WORD mode positions the cursor according to the data within the field
- Alternate from one tab mode to the other — Exit + M
- Append cursor to the position after the last nonblank character in the field — Exit + P

TAB Key (SKIP Key). To perform the tab function on a 911 or 915 terminal, press and hold the SHIFT key, then press the SKIP key. In this manual, this sequence is indicated as TAB. To perform the tab function on a 940 terminal, use the TAB key.

The tab function on the 911 and 915 terminals can be optionally interchanged with the skip function so that when you press the SKIP key, the tab function is performed. (See paragraph 4.7.) Alternately, when this option is used, the skip function is performed when you press TAB. Consult your systems operator to determine whether this switching option is enabled.

Switch Between Tab Modes. The Exit + M key combination allows you to alternate tab modes between TABS and WORD. After activation, ICS begins in TABS mode. Pressing TAB moves the cursor to preset tab stops, retained from session to session according to user ID. Pressing Exit + M changes the tab mode to WORD, where pressing (SHIFT) SKIP moves the cursor to the next word in the field. Like a toggle switch, subsequent use of the Exit + M sequence changes tab modes from one to the other.

The tabs for the TABS mode are set the same as tabs used by the DNOS operating system Text Editor. The SCI command Modify Tabs (MT) enables you to change your tab settings. To use the MT command, you must first suspend ICS at your terminal to regain access to SCI. To suspend ICS, press the orange Command key on a 911, the F9 key on a 915 terminal, or the FORM key on a 940 terminal. Then enter MT to display the existing tab stops. Enter new tabs as desired, separated by commas. Refer to the *DNOS Text Editor Reference Manual* for further information concerning the MT command.

ICS uses these tabs within each modifiable field. If you set tabs at 1, 5, and 15 and place the cursor at the first character position of an unprotected field (immediately to the right of the unprotected field indicator), the following occurs:

- When you tab the first time, the cursor moves to the fifth character position within the field (the cursor is already at position one).
- When you tab the second time, the cursor moves to the fifteenth character position within the field. If the field is less than 15 characters in length, the cursor moves to the first character position of the next unprotected field.
- When you tab the third time, the cursor moves to the first character position of the next unprotected field.

Whenever the tab position exceeds the length of the field, the cursor moves to the first character position of the next unprotected field. The TABS mode treats the beginning of a field as a permanent tab stop. The tab function moves the cursor to the next tab stop or the beginning of the next unprotected field, depending on which comes next on the screen.

In WORD mode, the cursor stops at the beginning of each word, which is a string of letters. It skips over spaces, nulls, numerals, and symbols — including the decimal point. When the field has no further words, or the cursor has reached the end of the field, tabbing advances the cursor to the beginning of the next unprotected field. For example, tabbing at the beginning of the field “(** JULY 4, 1776 **)” moves the cursor to J and then on to the beginning of the next field.

The DNOS operating system keeps a record of tab stops used by ICS for each station under the user ID given during SCI log-on. When you execute ICS, it retrieves the positions of the stops saved when ICS was terminated, and these become the stops for the new session.

Append-Cursor Function. The append-cursor function moves the cursor to the end of the data in a field. If you press Exit + P, the cursor moves to the position after the last nonblank character in the field. If the field contains nothing but blanks and nulls, the cursor moves to the beginning of the field. If the last position in the field holds a nonblank character, the cursor moves to the position to the right of the nonblank character. The following example demonstrates the results of using Exit + P on a 911 terminal and then entering the letter X:

Field		Result	
"ABC	"ABCX	"	"
"ABC D	"ABC DX	"	"
"ABC DEF	"ABC DEF	"	"X
"	"X	"	"

3.4.4.5 Editing Functions. The erase-field, erase-input, skip, insert-character, and delete-character functions edit the display supplied by the host application program or entered by the operator. The ERASE EOS key on the 940 terminal performs the erase to end-of-screen (EOS) function, whereas a two-key sequence on the other terminals performs the erase to EOS function.

Erase-Field Function. The erase-field operation erases the field from the beginning of the field to the end of the field, even if that field continues to another line. The erase-field operation replaces with nulls all the information in an unprotected field and marks the field as modified. The nulls appear as blank characters. After the erase-field operation, the cursor appears at the beginning of the field. If you position the cursor in a protected field and perform the erase-field operation, the audible alarm sounds and no data is cleared, but the cursor is repositioned over the first character of the protected field. The cursor blinks when you position it over a protected field or an attribute character. Table 3-5 shows which keys perform the erase-field function for the ICS-supported terminals.

SKIP Key. The SKIP key replaces with nulls all the data from (and including) the cursor position to the end of the field. After the operation, the cursor appears at the beginning of the next unprotected field.

The SKIP key on the 940 terminal performs the skip function. Usually, you perform the skip function on the 911 and 915 terminals by pressing the SKIP key without holding the SHIFT key down. An option is available to interchange the tab and skip functions so that when you press the SKIP key unshifted, you perform the tab function. When this option is enabled, the (SHIFT) SKIP performs the skip function. This option prevents accidental erasure of data caused by keystroke errors. The SKIP key is immediately below the RETURN key. Usually, if you press the SKIP instead of (or in addition to) the RETURN key, you erase data. Consult your systems operator to determine whether this option is enabled.

The SKIP key performs a similar operation to the erase-field key, except that it does not erase the data within the field to the left of the cursor. Also, the SKIP key positions the cursor differently after the operation. The erase-field operation erases all of the data in the field regardless of the cursor's position within the field. The SKIP key performs both erase and cursor movement operations and marks the field as modified. If the cursor is blinking (because it is in a protected field or over an attribute character) and you press the SKIP key, the audible alarm sounds, the operation is not performed, but the cursor does advance to the first position of the next unprotected field.

Erase-Input Function. The erase-input operation replaces with nulls (displayed as blanks) all unprotected fields and moves the cursor to the first unprotected character position on the screen. The cursor's position does not affect the operation of the erase-input function. If all of the fields on the screen are protected, the cursor moves to the first position on the screen and the data in the protected fields is not modified. If there are no fields on the screen, the entire screen is cleared and the cursor moves to the first position on the screen. This operation does not mark the fields as modified. Table 3-5 shows which keys perform the erase-input operation for each of the ICS-supported terminals.

Erase to End-of-Screen (EOS) Function. The erase to EOS function is not a standard 3270 function. This function replaces with nulls all data in every unprotected field from the current cursor position to the end of the screen. This function does not modify the protected fields. Table 3-5 shows which keys perform the erase to EOS operation for each of the ICS-supported terminals.

Delete Character Function. If you press the delete character key while the cursor is in an unprotected field, you delete the character at the cursor position. As you delete each character, the characters to the right of the cursor in the same field shift one character position to the left. The cursor does not move. The field in which the deletion takes place is marked as modified. As the characters shift left, nulls replace the character positions at the end of the field. If the field continues to the next row, the characters on the next row are moved one at a time from the first character position of the next row to the last character position of the row containing the cursor. When the cursor is on an attribute character or in a protected field, pressing the delete character key does not modify character positions.

Insert-Character Function. The insert-character operation puts the keyboard in the insert mode. In this mode, you can insert characters into unprotected fields. The insert operation can take place only if nulls or blanks are in the field to the right of the cursor. If the field continues to the next row, characters are moved one at a time from the last character position of the row containing the cursor to the first character position of the next row. However, the characters do not advance to the next row if you are working with an unformatted screen. If the field is already filled with characters, no insert takes place. In addition, the audible alarm sounds when you press a data key while in the insert-character mode.

The Insert Character key performs the insert-character operation on all ICS-supported terminals. Once you press the Insert Character key, each character you enter appears at the cursor position. The cursor and the data at and to the right of the cursor shift to the right one character position for each character you insert. When an insert operation takes place, the field is marked as modified. When alphanumeric characters occupy all positions of the field, no more insertions are allowed in that field. The insert operation does not shift attribute characters. When a field occupies more than one row and the characters in the field reach the end of a row, any additional inserts cause the characters to shift to the next line, provided that there are blank or null characters at the end of the field and you are not working with an unformatted screen. You can terminate the insert operation by performing any of the following operations:

- Erase input
- Erase field
- Cursor movement: (next char (→), previous char (←), previous line (↑), next line (↓), Home)
- Field left
- Field right
- Delete Character
- Suspend ICS Function
- Tab
- Skip
- New Line (RETURN)
- Extended Function (EXIT)
- Program attention
- Print screen
- Clear

Table 3-5. Cursor Movement and Edit Function Key Mapping

Function	Generic Key Sequence(s)
Cursor right	Next Character
Cursor left	Previous Character
Cursor up	Previous Line
Cursor down	Next Line
Cursor to the first character position within the first unprotected field	HOME
Cursor to the first character position within the next unprotected field	Next Field
Cursor to the first character position within the previous unprotected field	Previous Field
Switch tab mode between TABS and WORD mode	Exit + M
Tab according to the current tab mode	TAB
Append cursor to the end of a word	Exit + P
Erase entire field	ERASE FIELD
Erase to end-of-field	SKIP
Erase input	ERASE INPUT
Erase to end-of-screen	Exit + E
Delete character	Delete Character
Insert character	Insert Character

3.4.4.6 Suspend Function. This function suspends an ICS video station. To suspend ICS at a 911 terminal, press the Command key. To suspend ICS at a 915 terminal, press the F9 key. To suspend ICS at a 940 terminal, press the FORM key (unshifted). Paragraph 3.3.1 provides a complete explanation of ICS suspension.

3.4.4.7 Terminate Function. This function terminates the ICS video station. To terminate an active ICS display station, press Exit + Q. Paragraph 3.3.2.2 provides more detail about ICS station termination.

3.4.4.8 Print Functions. The following paragraphs describe the PRINT key, how to cancel a print order, and how to change the pathname assigned to the PRINT key.

PRINT Key. The PRINT key allows you to print the contents of the screen or copy it onto a file. When you press the PRINT key, the cursor does not move but begins to blink. When the print operation is completed, the cursor stops blinking and you can continue to enter data or edit the display. See paragraph 3.2.1.3 for PRINT key screen-image destination.

If the PRINT key pathname is assigned to a file that already contains data, that data is not lost. The contents of the screen are appended to the file.

Cancel Print Function. If you have pressed the PRINT key and the cursor is still blinking, you can cancel the print order by pressing the PRINT key again. Output at the PRINT key device halts, the cursor stops blinking, and you can continue to enter data or edit the display. If the cursor has already stopped blinking, the print has already completed, and there is no need to cancel it.

Identification Function (ident). The identification function changes the PRINT key pathname assigned to your ICS display station. Table 3-6 shows which keys perform the identification operation for each ICS-supported terminal. After you perform the identification function, ICS suspends itself, the bottom line is erased, and a prompt is displayed in its place. The prompt has the following general form:

Line = CMxx Term = yy, Printer: zz

Line = CMxx identifies the ICS communications line that you are using. The value xx is the numerical portion of the device access name. Term = yy identifies the terminal address that you are using. The value yy is the decimal terminal address in the range of 0 through 31. The communications line name and terminal address are displayed only for your information; you cannot modify them. The only element that you can modify is zz. The initial value displayed for zz is the value currently installed as the PRINT key pathname. You can set this value when activating the ICS station.

To change the PRINT key pathname, enter the desired pathname. ICS returns, and the screen appears exactly as it did before you performed the identification function. If you enter a null value or DUMMY into this field, your station has no PRINT key pathname and no action is taken when you press the PRINT key. If you disable your PRINT key and perform the identification function again, the initial value is represented as DUMMY. The PRINT key pathname value is the same value specified when activating an ICS video terminal. The same rules apply. (See paragraph 3.2.1.5.)

NOTE

If the PRINT key pathname is assigned to a spooler, you must follow the procedures discussed in Section 8 of the *DNOS ICS Object Installation Guide*.

3.4.4.9 Reset Function. The reset function unlocks the keyboard. If the cursor is blinking at the first character position of the bottom line of the screen, the keyboard is locked. The host application usually locks and unlocks the keyboard to inhibit input while it is processing data. Unless a transmission is currently in progress, you can unlock the keyboard by performing the reset function. Table 3-6 shows which keys perform the reset function for each ICS-supported terminal.

If there is a pending transmission, no action is taken.

3.4.4.10 Duplicate Functions. You can duplicate the contents of the previous field into the current field by using either the local duplicate or the remote duplicate function. The remote duplicate function enters a special code into the current field, and this special code is transmitted to the host with the rest of the data entered. The host application serving the ICS station interprets this code and performs the duplicate function. The local duplicate function duplicates the contents of the previous field before the transmission occurs.

Local Duplicate Function. The local duplicate function copies the contents of the previous modifiable field into the current field. The data appears on the screen the same as if you had entered it manually, except that it appears in low intensity. Table 3-6 shows which keys perform this operation on each of the ICS-supported terminals. If the cursor is in the first modifiable field of the screen, the contents of the last modifiable field of the screen are copied into the current field. Protected (unmodifiable) fields are ignored by this function. If the current field is defined to accept only numeric data and the previous field contains nonnumeric data, the audible alarm sounds and no action is taken. Similarly, if the current field is protected (indicated by a blinking cursor), the audible alarm sounds and no action is taken.

Remote Duplicate Function. When you perform this function and the cursor is in an unprotected field, a special code is entered at the cursor position. This special code indicates to the host application program that a duplicate operation should be performed for the rest of the field. When the host application performs the duplicate operation, it usually copies the contents of the previous modifiable field into the field containing the special code. ICS merely sends the code; the host application can interpret it in a number of different ways. The special code appears as the graphics character shown in Figure 3-3.

When this character appears, the cursor moves to the first character position of the next unprotected field. The field displaying the graphics character is marked as modified. Performing the duplicate function while the cursor is in a protected field or on an attribute character does not write the code or move the cursor.

Table 3-6 shows which keys perform the remote duplicate operation on each ICS-supported terminal.

3.4.4.11 Field-Mark Function. The field-mark function enters an end-of-field mark in an unformatted display image or, in a formatted display image, enters an end-of-subfield mark. ICS displays the end-of-field mark character as the graphics character shown in Figure 3-4.

Table 3-6 shows which keys perform the field-mark operation on each ICS-supported terminal. The field is marked as modified. If the cursor is in a protected field or on an attribute character, the cursor blinks. If you try to perform the field mark operation when the cursor is in a protected field or over an attribute character, no action is taken.

3.4.4.12 Alpha Function. The alpha function allows you to enter a nonnumeric character into a numeric field. Numeric characters include 0 through 9, the minus sign (–), and the decimal point (.). Some international keyboards use the comma instead of the period as a valid numeric character.

Table 3-6 shows which keys perform the alpha function on each ICS-supported terminal. To perform this function, first position the cursor at the beginning of the numeric field, then press the appropriate keys.

2282170

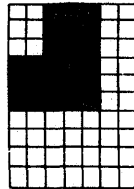


Figure 3-3. Remote Duplicate Graphics Character

2282171

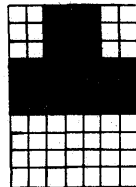


Figure 3-4. Field-Mark Graphics Character

3.4.4.13 Program-Attention Functions. The program-attention operation requests the attention of the host application program. Performing one of the program-attention functions locks the keyboard and sends an attention identification (AID) code to the host application program, identifying which action is requested. The application program takes the appropriate action. After the application program has acted on the attention request, the program(s) may reinitialize the keyboard. ICS supports the following program-attention functions:

- Clear
- Enter
- Program function 1 through 24
- Program access 1 through 3

Clear Function. The clear operation clears the entire screen to nulls, moves the cursor to the first character position on the screen, and transmits an attention identification (AID) code to the host.

Table 3-6 shows which keys on each ICS-supported terminal perform the clear function.

Enter Function. The enter operation transmits the display image to the host application program. Only modified fields (whose attribute characters indicate that they are modified) are transmitted.

Table 3-6 shows which keys on each ICS-supported terminal perform the enter function.

Program Function (PF) Keys. The host application program defines program function keys on the IBM 3277 Model 2 Display Station to request program action, which usually involves reading data from the buffer of the ICS display station. When you press a PF key, the AID code and the modified data in the buffer of that station are transmitted to the host application program.

Table 3-6 shows which keys on each ICS-supported terminals perform the program function operation. There are several ways to perform the program function operation for most of the emulated AID codes.

Program Access Function. The host application program defines program access keys on the IBM 3277 Model 2 Display Station to request program action without requiring that data be read from the buffer of the ICS video station. When you press one of the program access keys, only the program access key code is transferred to the application program. The only difference between program access keys and PF keys is that a program access key sends only the key code, while a PF key transmits the modified fields as well as the key code. ICS emulates three program access keys, PA1 through PA3. The 940 terminal uses the P1, P2, and P3 keys. Table 3-6 shows which keys on each of the ICS-supported terminals emulate program access keys.

Once you press the first key of this sequence on any of these terminals, the keyboard locks, and the cursor blinks in the first column of the bottom row. When you press the second key, the program access key code is transmitted to the host application. What happens next depends on how the host application is programmed to handle the particular program access code. To unlock the keyboard, use the reset function. Paragraph 3.4.4.9 explains the reset function.

Table 3-6. Special Function Key Mapping

Special Function	Generic Key Sequence(s)
Suspend ICS station	Command
Terminate ICS station	Exit + Q
Print screen	PRINT
Cancel print	PRINT + PRINT
Cancel Exit mode	Exit + Exit
Identify printer (also ICS line and terminal address)	Exit + I
Reset (unlock keyboard)	Exit + R
Local duplicate	Exit + L
Remote duplicate	Exit + D
Field mark	Exit + F
Alpha (override numeric protection)	Exit + A
Clear	Initialize Input
Enter (send to host)	ENTER
Program function keys PF1 through PF8	F1 through F8
Program function keys PF9 through PF24 ¹	Exit + 09 through Exit + 24
Program access key PA1 ²	Exit + + (plus)
Program access key PA2 ³	Exit + - (hyphen)
Program access key PA3 ⁴	Exit + _ (underscore)

Notes:

¹ These keys are F9 through F24 on a 940 EVT.

² This sequence is (ALT)P1 on a 940 EVT.

³ This sequence is (ALT)P2 on a 940 EVT.

⁴ This sequence is (ALT)P3 on a 940 EVT.

Functional Description

4.1 INTRODUCTION

This section is for systems personnel responsible for the installation and maintenance of the DNOS 3270 Interactive Communications Software (ICS) package. The procedures it describes are not necessary for typical everyday operation.

This section provides a more detailed explanation of how ICS uses the hardware. It explains the meaning of the parameters used in ICS installation and how they can be modified. It also explains how to enable options such as switching SKIP key functions and what the ICS local format feature provides. In addition, this section also describes the following SCI (System Command Interpreter) command procedures:

- List Interactive Communications Configuration (LICC)
- Modify Interactive Communications Configuration (MICC)
- Modify ICS Control Unit Address (MCUA)
- List Interactive Communications Statistics (LICS)
- Execute Interactive Communications Controller (XICC)
- Execute Poll Analyzer (XPA)

4.2 INSTALLATION

The following paragraphs describe the installation of ICS. Besides installing the hardware in the correct slots of the backpanel with the correct jumpers and necessary cables, the software must also be correctly generated and installed for a specific configuration. The *DNOS 3270 ICS Object Installation* document briefly describes the following software:

- Remote Terminal Subsystem (RTS) installation and generation
- DNOS operating system installation and generation
- ICS installation and generation

Any Texas Instruments video display terminal installed on the DNOS operating system can be used as an ICS station. During this procedure, you must define to the operating system all the hardware necessary for ICS operation. Only the communications controller must be specifically defined as ICS hardware. You do not define the 911 and 915 terminals as ICS stations during system generation. If you use Programmed Station Control (PSC), ICS requires that you specify more intertask communication area to handle the interface between the PSC Emulator and your PSC program. This requirement is described in the *DNOS 3270 ICS Object Installation* document. For complete information about system generation and installation, see the *DNOS System Generation Reference Manual*.

4.2.1 ICS Installation and Generation

The ICS package must be tailored to the specific configuration of your system. This is done by the Execute System Generation Utility (XSGU) procedure and the Build Emulator Tasks (BET) procedure, which builds the ICS tasks. You must specify configuration-dependent parameters in XSGU, such as the number of IBM terminal addresses reserved for a given ICS line and the Poll Timeout Value.

4.2.2 Initialization of ICS

The following modifications to the .S\$ISBTCH procedure file are recommended in order to automate the initialization of ICS. You do not have to make these modifications. However, if you don't you must perform the procedures manually after each initial program load (IPL) of the 990 computer. The following modifications are recommended:

- Downloading of the character detect routines for some communications controllers
- Modify Control Unit Address (MCUA) or Modify Interactive Communications Configuration (MICC) (if your configuration requires these modifications)
- Activation of the ICS line (XICC)

4.2.3 Execute Interactive Communications Controller Task — XICC

Before you can activate any ICS stations, you must activate the communications line. For multiple line configurations, you must activate each line individually, once after each IPL. The line responds to host polls even when no ICS stations are active on that line. When the local format option is installed, the line accepts selects when no ICS stations are active on that line.

To activate an ICS communications line, enter XICC in response to the SCI prompt. The following display appears on the screen:

```
[ ] XICC
```

```
EXECUTE INTERACTIVE COMMUNICATIONS CONTROLLER  
COMMUNICATIONS LINE NAME:
```

The only parameter displayed is the COMMUNICATIONS LINE NAME prompt, which is the device name of the communications port or a synonym for such a value. The general form is CMxx, where xx indicates a specific port. If you enter a null value, ICS searches the list of installed ICS lines in the order that you declare them during installation and selects (as the default) the first inactive line.

Upon successful activation, ICS returns a message. For example, if you activate CM03, ICS displays the following message:

```
*XICC-COMMUNICATIONS LINE CM03 ACTIVATED:
```

Since you must activate an ICS line only once after each IPL, you may find it convenient to include the activation command in the `.$SISBTCH` procedure file. If you have only one ICS line on your system, you can add `XICC/` to your `.$SISBTCH` procedure file. (The slash (/) is an SCI primitive command and is required here.) You can only activate or deactivate a line using the IPL procedure. Using the Kill Job (KJ) command on an ICS job leaves the shared procedure as if ICS were still active and XICT cannot reactivate the line. You do not need to specify the communications line name since ICS searches a list of installed lines and selects the first inactive installed ICS line it finds when no line is specified. If you have more than one line installed, include `XICC/` twice in the `.$SISBTCH` procedure. The only time you need to specify the line name is when you have more than one line installed and you do not want to activate the one that was declared first during installation. For example, if you declared CM01 first during installation and CM02 second, you can activate only CM02 by including the following line in the `.$SISBTCH` procedure:

```
XICC COMMUNICATIONS LINE NAME=CM02
```

Usually, the controller task remains active until you perform another IPL. Paragraph 4.6 contains a discussion of abnormal termination.

If you want the ICS line to use a control unit address other than the one you assigned during installation, you must execute the Modify Control Unit Address (MCUA) command before you activate the ICS line. Paragraph 4.3.3 describes the MCUA command.

4.2.4 Downloading Character Detect Routines

All versions of the Four-Channel Communications Controller (FCCC) and the Bit-Oriented, Character-Oriented Asynchronous Interface Module (BCAIM) require downloading of character detect routines. The modifications to the `.$SISBTCH` file that ICS requires for the downloading of these communications controllers are detailed in the *DNOS 3270 ICS Object Installation* document.

4.3 CONFIGURATION MANAGEMENT

ICS provides three commands that manage the ICS configuration:

- List Interactive Communications Configuration — LICC
- Modify Interactive Communications Configuration — MICC
- Modify ICS Control Unit Address — MCUA

4.3.1 List Interactive Communications Configuration — LICC

The LICC command lists the installed configuration for a particular ICS line and current information about the use of resources assigned to that line. To execute this procedure, enter LICC in response to SCI. The following appears on the screen:

```
[ ] LICC
```

```
LIST INTERACTIVE COMMUNICATIONS CONFIGURATION
COMMUNICATIONS LINE NAME:
OUTPUT PATHNAME:
```

Your response to the COMMUNICATIONS LINE NAME prompt identifies the communications controller, the specific port, and the associated IBM-compatible host. You can use any valid ICS line identifier (such as CM01) or synonym, or logical name. The default value for this field is the first installed ICS line.

Your response to the OUTPUT PATHNAME prompt identifies the location to which ICS writes this information. The default value for this field is the device name of the station at which you entered the command. The output pathname can be any valid pathname of a file, or any sequential output device name, such as LP02 for a printer. If you enter a null value, ICS uses the terminal at which you entered the LICC command as the output pathname.

The following example shows how to generate the configuration listing for the ICS line identified by CM01.

EXAMPLE

```
[ ] LICC
```

```
LIST INTERACTIVE COMMUNICATIONS CONFIGURATION
COMMUNICATIONS LINE NAME: CM01
OUTPUT PATHNAME:
```

ICS accepts the null value in response to the OUTPUT PATHNAME prompt and displays the following information at your terminal:

3270 ICS RELEASE 1.1 - CONFIGURATION LISTING

ID	TYPE	STATE	JOB NAME	RID	INPUT	PRINTER PATHNAME
---	---	---	---	---	---	-----
0	CTL	ACTIVE	ERMIDDLE	13	CM01	
0	CRT	IDLE				LP02
1	CRT	ACTIVE	RB	DA	ST14	LP02
2	CRT	PSC-WAIT	YUAN	23		LP01
3	CRT	PSC-ACTV	YUAN	20		LP01
4	PRT	ACTIVE	YUAN	11		.SYS1.P3270.MARCH.SALES
5	CRT	IDLE				LP01
6	CRT	SUSPENDED			ST08	DS02.JACK.MISC.PRINTER

The first line of the display identifies the release number of the ICS package that your system uses and identifies the display as a configuration listing. The next line is a header that identifies the type of information for each field. The line immediately below the header contains information about the communications line.

The number just below the column header ID identifies the decimal control unit address assigned to the ICS line. CTL (an abbreviation for controller) under the column header TYPE identifies the communications line. The state under the column header STATE can be either active or inactive. The character string below JOB NAME identifies the job name of the controller task. This is the job from which the ICS line was activated. The two-digit hexadecimal number below the column header RID identifies the task run ID of the controller task for this ICS line. ICS displays CM01 beneath the column header INPUT to identify the device name of the communications controller that this ICS line uses. No printer pathname is displayed since printers are not assigned directly to communications lines.

A blank line separates the communications line information from the information about the individual terminal addresses reserved for that line. This display lists each reserved terminal address in numerical order from zero to the highest installed terminal address in the far left column headed ID.

The next column, headed TYPE, indicates the installed station type for each terminal address. Station type can be either CRT for video terminal or PRT for printer station. The third column, headed STATE, indicates the current state of a particular terminal.

Table 4-1 identifies and defines all of the possible states:

Table 4-1. Terminal States

State	Description
ACTIVE	Terminal address is in use by the station identified by the device name listed under INPUT. The listing shows the ICS station task's run ID (beneath RID) and the printer pathname currently assigned to the terminal address.
SUSPENDED	Terminal address has been activated and suspended, but not terminated. The terminal address remains reserved and unavailable for use by any other station until the station is terminated. Since the ICS station is no longer in the system, the display shows no task run ID beneath RID. The station that reserves the terminal address is identified by the device name listed under INPUT. This display also shows the printer pathname currently assigned to this terminal address.
PSC-ACTV	ICS has assigned the terminal address to a currently active user-written PSC program. The task run ID listed identifies the copy of the PSC Emulator (ICSPSC) that serves the terminal address, not the user-written PSC task. No input device is shown for the terminal address. This display also shows the printer pathname currently assigned to this terminal address.

Table 4-1. Terminal States (Continued)

State	Description
PSC-WAIT	ICS has assigned the terminal address to a PSC Emulator. The PSC Emulator (ICSPSC) waits until the host sends more data across the line or until a user-written task issues an Open call. The task run ID identifies the ICSPSC that serves the terminal address, not the user-written PSC task which may or may not still be active. No input device is shown for the terminal address. This display also shows the printer pathname currently assigned to this terminal address.
IDLE	Terminal address is not active and is available for use. No input device or task run ID is associated with an idle terminal address. An idle terminal address does have a printer pathname assigned to it.
DISABLED	Terminal address is not active and not available for use. To enable a station, use the Modify Interactive Communications Configuration (MICC) command. Even though the terminal address is disabled, a printer pathname is assigned to it.

4.3.2 Modify Interactive Communications Configuration — MICC

The MICC command modifies either the state or type of a particular ICS terminal address. MICC can change the station type from CRT to PRT and vice versa, and it can change the state from enabled to disabled and vice versa. The terminal address must be idle or disabled. MICC cannot modify a terminal address that is in use. The MICC command permits dynamic configuration, which means that you do not have to reinstall ICS to modify your configuration.

NOTE

Use of this command should be limited to modifications that match the configuration of the host. If the host expects a particular terminal address to be a printer station and you have modified that address to the video type, errors may occur when the host sends printer commands and orders to your video station. MICC cannot be used after XICC has been performed. An attempt to do so results in the error message:

CANNOT MODIFY ICS CONFIGURATION FOR ACTIVE LINE CMxx.

To execute this command, enter MICC in response to the SCI prompt. The following display appears on the screen:

[] MICC

```

MODIFY INTERACTIVE COMMUNICATIONS CONFIGURATION
COMMUNICATIONS LINE NAME:
  TERMINAL ADDRESS(0-31):
  STATE(ENABLED/DISABLED):
  TERMINAL TYPE(CRT/PRT):

```

The first two prompts identify the specific terminal address to be modified. ICS requires a valid response for both prompts. No default value exists for either the COMMUNICATIONS LINE NAME or the TERMINAL ADDRESS(0-31) prompts in this procedure.

MICC modifies only one terminal address at a time. To change the type or state of a terminal address, enter the desired change in the appropriate field. If you enter a null value, no change is made. You must have at least one video terminal address (type CRT) enabled on each installed ICS communications line. If you try to disable the last video terminal address or modify it to the PRT type, ICS displays the following error message:

*MICC-CANNOT MODIFY LAST CRT ADDRESS:

As an example, assume that CM01 is inactive as shown in the previous output example for LICC. Terminal address 05 is in the enabled state and is the CRT type. Enter MICC and respond as follows:

[] MICC

```

MODIFY INTERACTIVE COMMUNICATIONS CONFIGURATION
COMMUNICATIONS LINE NAME: CM01
  TERMINAL ADDRESS(0-31): 05
  STATE(ENABLED/DISABLED): DISABLED
  TERMINAL TYPE(CRT/PRT):

```

This changes terminal address 05 on the CM01 line from an enabled state to a disabled state (unavailable). Since you enter no response to the TERMINAL TYPE (CRT/PRT) prompt, the type of terminal address 05 remains CRT. The MICC command automatically executes the LICC command for the communications line name that you entered. The following listing appears at your terminal:

3270 ICS RELEASE 1.1 - CONFIGURATION LISTING

ID	TYPE	STATE	JOB NAME	RID	INPUT	PRINTER	PATHNAME
0	CTL	INACTIVE			CM01		
0	CRT	IDLE				LP02	
1	CRT	IDLE				LP02	
2	CRT	IDLE				LP01	
3	CRT	IDLE				LP01	
4	PRT	IDLE				.SYS1.P3270.MARCH.SALES	
5	CRT	DISABLED				LP01	
6	CRT	IDLE				DS02.JACK.MISC.PRINTER	

4.3.3 Modify Control Unit Address — MCUA

The MCUA command changes the control unit address assigned to an inactive ICS communications line. You should use this command only to select the control unit address that the host configuration expects for your ICS line. To execute this command, enter MCUA in response to the SCI prompt. The following prompts appear:

```
[ ] MCUA
```

```
MODIFY CONTROL UNIT ADDRESS
COMMUNICATIONS LINE NAME:
CONTROL UNIT ADDRESS(0-31):
```

Enter the device name of the communications controller port assigned to the ICS line in response to the COMMUNICATIONS LINE NAME prompt. Respond to the CONTROL UNIT ADDRESS(0-31) prompt with the desired decimal control unit address.

If for any reason MCUA cannot perform the modification, ICS returns an error message. If the modification is successful, ICS also returns a message. For example, if you use MCUA to change the control unit address of the ICS line identified by CM01 from 0 to 3, ICS displays the following message:

```
*MCUA-CONTROL UNIT ADDRESS FOR CM01 MODIFIED FROM 0 TO 3:
```

4.4 LIST INTERACTIVE COMMUNICATIONS STATISTICS — LICS

ICS maintains statistics for all activity that occurs after the activation of a communications line. This information aids in recognizing problems in hardware and user-written software. To execute this command, enter LICS in response to the SCI prompt. The following prompts appear on the screen:

```
[ ] LICS
```

```
LIST INTERACTIVE COMMUNICATIONS STATISTICS
COMMUNICATIONS LINE NAME:
OUTPUT PATHNAME:
```

Your response to the COMMUNICATIONS LINE NAME prompt is the device name of the ICS communications controller port. Your response to the OUTPUT PATHNAME prompt can be the name of any file or sequential device. If you enter a null value, ICS writes the statistics listing to the video terminal at which you entered LICS. The following is an example of the displayed listing:

3270 ICS RELEASE 1.1 - STATISTICS LISTING

CM01 - 15:19:32 WEDNESDAY, NOVEMBER 04, 1981

	INBOUND -----	OUTBOUND -----
MESSAGE/EOT	82	21625
BLOCK	132	11
RETRANSMISSION	0	43
ABORT/RVI	0	0
COMMAND REJECT	N/A	0
POLL	21684	N/A
GENERAL POLL	21684	N/A
SPECIFIC POLL	0	N/A
POLL ERROR	0	N/A
SELECT ERROR	0	N/A
BUFFER ERROR	0	N/A

Only the host performs selects and polls, so the select and poll information is listed only in the inbound column. Other statistics are kept for both as follows:

- Inbound — From host to ICS
- Outbound — From ICS to host

4.4.1 Message

ICS increments the inbound message count every time it receives an end-of-transmission (EOT) character. ICS increments the outbound message count every time it transmits an EOT.

4.4.2 Block

ICS increments the inbound block count every time it receives a start-of-text (STX) character. ICS increments the outbound block count every time it transmits an STX.

4.4.3 Retransmission

The retransmission count is the number of blocks that ICS (outbound) and the host (inbound) retransmit in response to a negative acknowledgment (NAK).

4.4.4 Abort

The abort count indicates the number of times that ICS sends a reverse interrupt (RVI) line control character (outbound) and the number of times ICS receives an RVI (inbound).

4.4.5 Command Reject

ICS counts the number of 3270 commands, such as Copy or Read Buffer, that it has rejected. The outbound count is the number of commands from the host that ICS has rejected. Rejection means that ICS recognized a command but did not execute it.

4.4.6 Polls

ICS maintains a count of all polls it receives. The poll count is the total number of polls ICS receives for the control unit address assigned to the ICS line. The total number of polls is the number of specific polls plus the number of general polls. The host uses a specific poll to request the attention of a particular terminal address configured on the ICS cluster. The host uses a general poll to ask if any of the terminal addresses on the ICS cluster require servicing.

Each ICS station or Programmed Station Control task serving as an ICS station has a terminal address assigned to it when activated. If the ICS device service routine (DSR) receives a specific poll, it transmits data only when the station with the terminal address specified in the poll has a transmit pending. If the ICS DSR receives a general poll, it transmits when any of its stations has data to send.

4.4.7 Poll Error

ICS increments the poll error count every time the host polls ICS and specifies a terminal address that ICS cannot access. This error can occur if you do not reserve the address during installation.

4.4.8 Select Error

A select station error occurs when the host tries to initiate a transmission to a specific terminal address on the cluster but ICS cannot access that terminal address. This error can occur if you do not reserve the specified address during installation.

4.4.9 Buffer Error

The buffer error occurs when the host selects ICS to receive a transmission and ICS has no available buffers in which to store the transmission.

4.5 EXECUTE POLL ANALYZER — XPA

The XPA command generates a limited trace of polls passing between ICS and the host. XPA works by writing the poll buffer and status message buffer to the specified listing device in a manner similar to the List System Memory (LSM) command.

To execute the poll analyzer, enter XPA in response to the SCI prompt. The following prompts appear:

```
[ ] XPA
```

```
EXECUTE POLL ANALYZER
COMMUNICATIONS LINE NAME:
NUMBER OF SAMPLES: 7
LISTING ACCESS NAME: ME
```

Your response to the NUMBER OF SAMPLES prompt determines how many lines of trace output ICS returns. This prompt requires a response. Enter a decimal value.

Your response to the LISTING ACCESS NAME prompt determines where ICS writes the output listing. The initial value for this prompt is the station at which you entered the command.

The trace output has the following format:

```

CCF6   4040   7F7F   2D00   0901   3105   0240   40C2   4003   ....-...1..@@.
  A     B     C     D     E     F     G                               ASCII image

```

where:

- A is the beginning address of the buffer that was read.
- B is the controller address that was last polled or selected. This value is displayed twice.
- C is the terminal address or a general poll. A general poll is defined as terminal address > 7F. This value is also displayed twice.
- D is the enquiry (ENQ) poll command.
- E contains random information unrelated to ICS operation.
- F contains the highest terminal address reserved for the ICS line in the right byte.
- G and the following two words of memory contain random information unrelated to ICS operation.

ASCII image is the ASCII representation of the information contained in items B through G. This is not a valid representation for hexadecimal values greater than > 7F. All values greater than > 7F are shown as a period (.).

If the values for B, C, and D are all zeros, ICS has received no polls since the last IPL. If B contains > 3D, the last polled controller did not respond. If B, C, and D always contain the same poll or select address, either the host is polling only that controller or the line has stopped operation since the last poll was read.

4.6 CONTROLLER TASK

When an ICS video or printer station is active, it sends the data it receives across the communications line from the host directly to the device associated with the control unit address and terminal address during activation. The 3270 communications Device Service Routine (DSR) handles the link with the host. When receiving, the DSR reads the data link control characters, responds to selects, and routes the data to the controller task. The controller task then routes the data to the appropriate device. When sending, the DSR routes the data to the specified communications controller. The DSR for the ICS communications controller inserts the data link control characters and responds to the polls of the host. The controller task does not communicate with the host application programs. ICS stations generate or receive data from the host applications through the ICS controller task, the communications DSR, and the IBM-compatible front-end control unit.

If local format is installed and the host selects an ICS terminal address when the terminal address is idle, the communications DSR acknowledges the select as if that station were active and ready to receive. When the host begins the transmission, the controller task forwards the data to a local format file. During installation you can either specify no local format file for a particular ICS line or accept the following pathname:

`.$ICS.CMxxTMyy`

CMxx identifies the communications port name used by the ICS line, where xx is a two-digit decimal number. TMyy identifies the specific terminal address, where yy is a two-digit decimal number. Each terminal address has its own local format file. (Note that these files contain untranslated data from the host application.) The local format file is not a standard ASCII text edit file. It has a physical record size of 1920 characters. If you attempt to look at a local format file by using the SCI procedure Show File (SF), the display is not a true reflection of the contents of the file. The data is in EBCDIC form and contains IBM codes for commands, orders, and attribute characters. This data requires additional processing. The ICS printer, video station, and PSC tasks perform this processing. See paragraph 4.9 for more information on local format feature.

ICS uses the data link control procedures supported by the IBM 3271 Model 2 Control Unit. This protocol is a subset of the Binary Synchronous Communications (BSC) procedures described in Appendix F.

4.7 ICS VIDEO STATIONS

ICS display stations emulate the IBM 3277 Model 2 and IBM 3278 Model 2 Display Stations. Every time an ICS video station is activated, the ICS video station task examines the local format file assigned to the activated terminal address. If there is any data in the local format file, ICS forwards it to the specified video station. The local format file can be empty or contain one or more screens. These screens can be either from previous suspended transactions or from transmissions that the host sent while the station was inactive.

An option is available that switches the functions of the SKIP key. Skip is the unshifted function of the SKIP key. The skip function erases data from the cursor to the end of the field. The SKIP key is located between the RETURN and SHIFT keys, both of which are frequently used. Consequently, you can inadvertently erase data by pressing the SKIP key when intending to press RETURN or SHIFT. If you enable the switch option, such a mistake performs the tab function rather than the skip function, and you do not lose data. To enable this option, execute a Text Edit (XE) of the Execute Interactive Communications Terminal (XICT) command procedure, specifying `.$$CMDS.XICT` as the file access name. The next-to-last line contains the parameters (enclosed in parentheses) used with the bid task primitive. Commas separate the parameters. Counting from the left, (beginning with one) the eighth parameter specifies whether this option is turned on or off. When you receive ICS, the eighth parameter is null. Change this to Y to specify on. After you replace the contents of the file with this modification, the tab function is switched with the skip function. ICS displays no prompt for this option.

4.8 ICS PRINTER STATIONS

ICS emulates the operation of the IBM Model 2 3284, 3286, or 3287 Printer using one of the DNOS-supported sequential input/output (I/O) devices or a sequential or relative record file. The ICS printer station provides a copy of information it receives to the printer station from the application program executing at the host.

When you specify a sequential file, relative record file, or sequential output device as the ICS printer station, the print operation is the same as that of a printer except that the data is written to a file or device other than a printer. If the file you specify does not exist, ICS creates a sequential file that is not blank suppressed, providing that the directory in which the file will exist is accessible and not full. When the file or device is initially opened, it is opened with an open extend. The data from the host is appended to the file or device. When the station is terminated, ICS writes an end-of-file (EOF) to the file or the device. If you specify a file as the output pathname, you can display the contents of that file using the SCI Show File (SF) command. To print a copy of a file used as the ICS printer station, use the SCI Print File (PF) command.

4.8.1 Supported Printers

The printers emulating IBM printers have a 1920-character buffer exactly as the ICS display station does. The character set emulated is the same as the ICS display station character set (refer to Appendix D). Therefore, all of the characters that can be displayed on an ICS display station can be printed on the printer station. Certain printers cannot print lowercase characters.

4.8.2 Printing Initiated by the Host Application Program

The host application program can direct the printer station to print information that the host application transmits. To do this, the host program issues a 3270 Write command addressed to the printer station. This command is transparent to the operator. An intermediate buffer receives the data accompanying the Write command before the printer buffer receives it. The data is scanned and formatted according to a line-length specification in the Write command or according to printer orders that are embedded in the data. As each line is formatted, it is transferred to the printer buffer.

4.8.3 Print Line Formatting

The host application determines the format of the data it sends to the ICS printer station. The host application can embed printer orders in the data to control the format or it can specify a line length to format the printed information. If printer data has no line length specified, it is unformatted. If printer data has a line length specified, it is formatted.

Three printer orders may be used by the host application program to format the data:

- New Line (NL)
- Form Feed (FF)
- End of Message (EM)

The host application embeds the printer orders in the data stream. When the ICS printer station encounters the orders, it performs the formatting function.

New Line (NL) Order. When the ICS printer station encounters an NL order in the data stream, it performs a carriage return and line feed before it continues to print. If the printer station reaches the end of the line before it encounters an NL order, it automatically performs a carriage return and line feed and continues to print. When the ICS printer station encounters an NL order in a nondisplay/nonprint field, it does not execute the order. Instead, it treats the order as an alphanumeric character and prints it as a blank (space). When the ICS station encounters an NL order in a printable field of a printout that has a line length specified, it does not execute the order. Instead, the ICS printer station prints it as a 5.

Form Feed (FF) Order. When an ICS printer station encounters an FF order in the data stream, the printer station advances the paper to the top of the form. The printer station begins the print on this line. The ICS printer station prints the character position containing the FF order as a space. If the paper is located at the top of the form when ICS encounters the FF order, the paper advances to the next top of form, resulting in a blank page.

End of Message (EM) Order. The EM order indicates the end of the printout specified by the most recent Write command. ICS printer stations do not execute the EM order in the following situations:

- When it is encountered in a nonprint/nondisplay field. Instead, the ICS printer station prints it as a space.
- When the most recent Write command specified a line length for the printout. Instead, the ICS printer station prints it as a 9.

When ICS encounters an EM order, it performs a final line feed, terminates the printing operation, and ignores any further information in the buffer. The ICS printer station does not perform the final line feed if the EM order is in the first position of a print line.

Line Length Specification. A line length of 40, 64, or 80 characters can be specified in the Write command. If the Write command does not specify line length and embeds no printer orders, the ICS printer station prints 132 characters on each line. After the ICS printer station prints character 132, it performs a carriage return and line feed. When the Write command specifies a line length, ICS writes the data from the intermediate buffer to the printer buffer with the specified line length. ICS then scans the data characters in the printer buffer one line at a time before it prints them. When a line contains one or more space characters only, a blank line is printed. When ICS encounters null, alphanumeric, or attribute characters in nonprint fields, it prints them as spaces when printable characters appear in the same line. When the entire line is a nonprint field, ICS ignores the line; it does not generate a blank line. When ICS encounters a printer order in a line with a line length specification, ICS replaces it with a graphics character. For the printer order NL, the graphic character is 5. For EM, the graphic character is 9. ICS performs a line feed after every line is printed. When the Write command does not specify a line length, the embedded printer orders control the line length.

4.8.4 Error Conditions

A not-ready condition occurs when the printer station is out of paper or is mechanically disabled. The not-ready condition also occurs when ICS cannot access the file or sequential device acting as the printer station. If an ICS printer station is active when the host application program tries to select one, ICS directs the data from the host to the device identified by your response to the OUTPUT PATHNAME prompt. If this device is not ready or becomes not ready during the printing operation, ICS automatically retries the printout for a specified time interval. If the retry is successful, the printing continues; if the retry is not successful, the ICS printer station writes the current buffer onto the local format file for the terminal address in use, writes a message to the system log, and terminates.

If you specify no local format file for the ICS line during installation and a time-out occurs, ICS discards the data. If you specify no local format file for the ICS line during installation and no ICS printer station is active when the host tries to select a printer on the line, ICS does not accept the select.

If no printer station is active when the host application selects an ICS printer station, ICS accepts the select and writes the data from the host to the local format file. If for any reason ICS cannot access the local format file, it retries the write until the time-out interval has expired.

The time-out interval is a function of a parameter within the Execute Interactive Communications Printer (XICP) command and the time-out value specified for the particular device during DNOS system generation (XSGU). The *DNOS 3270 ICS Release Information* describes how ICS determines the printer time-out interval.

Other ICS error conditions, if detected by the printer station, also terminate the printer station. Appendix B describes the ICS error conditions.

4.9 LOCAL FORMAT FEATURE

ICS operation includes an important optional feature, called local format, which provides you several extended capabilities. When the ICS line has been activated and the host has data to send to a device address which has no emulator active, ICS accepts the data sent and directs it to the local format file allocated for that device address and that line. However, if no local format has been specified, then ICS does not accept data from the host for a device address if the emulator for that device address is not active. When the emulator is activated for a device address, ICS first reads the local format file and, for a CRT type device, updates the screen with any data that was saved. For a PRT type device, ICS first reads the local format file and directs all of the data to the OUTPUT PATHNAME before accepting data directly from the host. Once the data has been processed by ICS, it is no longer available from the local format file.

Another use of the local format file is to preserve the screen image and cursor position when an operator presses the suspend ICS function key to return to SCI. The ICS device address remains reserved for that station so that later, when the operator reactivates ICS at the same terminal, the previous image is redisplayed and the cursor is placed in the position it occupied just before the ICS station was suspended. However, if the operator terminates the session ICS does not save the screen image.

The local format file is also very important for the printer emulator when an unrecoverable error occurs. If the printer emulator output is directed to the spooler or a hard copy device and the device runs out of paper, jams, or in some way becomes inoperable, ICS retries for a user-defined length of time and then writes the current print image to the local format file. If a local format file has not been installed, the current print image is lost. Refer to the *DNOS ICS Release Information* for information concerning modifying printer-timeouts.

The local format feature is also available to PSC applications. In all cases, the local format feature must be specified during ICS installation and is available to all device addresses on the communications line for which it is enabled.

Programmed Station Control

5.1 GENERAL DESCRIPTION

The Programmed Station Control (PSC) facility in ICS enables an application program to serve as an ICS display or printer station. This section provides the systems programmer with a description of PSC routines, their capabilities, and how they work in a DNOS environment. The PSC software consists of a set of procedural language routines and an interface to ICS. The user-written program can perform input/output (I/O) across a 3270 line via ICS by calling the following routines:

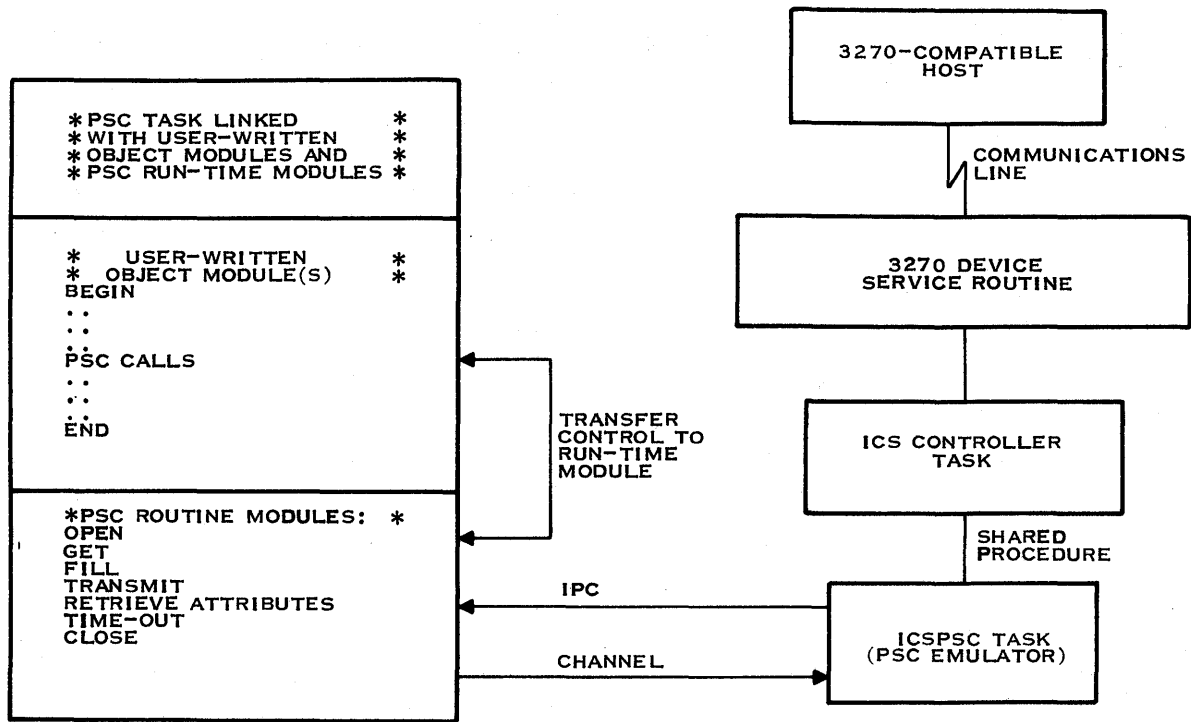
- Open — Allows program access to an ICS station and line
- Fill — Modifies data in the station buffer
- Get — Retrieves data from the station buffer
- Transmit — Sends modified data across the line to the host
- Time-out — Waits for the host to reply
- Retrieve Attributes — Retrieves field attribute information from the buffer
- Close — Ends access to the 3270 line

PSC gives the application program many of the features of an ICS station, plus a number of additional capabilities. A user-written PSC program emulates an ICS display or printer station. The program replaces the ICS station. The PSC program runs as a task under DNOS and exchanges data with the host application through ICS via the PSC Emulator as shown in Figure 5-1.

This section refers to the task that you write to emulate an ICS station as the *PSC program* or *PSC task*. Your PSC program calls the *PSC routines* that you link with your program. As shown in Figure 5-1, the interface between the PSC routines and the ICS controller task is called the *PSC Emulator*.

PSC provides the user-written program with the following capabilities:

- Control over station status and operation
- Data transmission and reception along a 3270 line
- Data buffering in the familiar display screen format
- Manipulation of data in the screen buffer
- Access to attribute information



2283597

Figure 5-1. PSC Overview

Operating by itself, a PSC program can take over any repetitive task usually performed by a station operator. The program receives screens from the host, modifies them as necessary, transmits them back along the ICS line, and waits for the host to reply. At the same time, the PSC program enjoys all the facilities provided by DNOS for using files, devices, and utilities. By coupling the PSC program with user-written interfaces to data entry devices, the programmer can often enhance the performance of an application. The PSC program can obtain data from one or more data entry stations, check for entry errors, and transmit the data to the host. By detecting errors before the data goes to the host, the program reduces traffic along the communications link.

This section provides the application programmer with the information needed to write a program employing the PSC routines. The discussion of program requirements provides essential background information on the data structures used in a PSC program. Next, the discussion deals with the routines that enable the PSC program to emulate an ICS station. This includes the calling syntax in COBOL, FORTRAN, Pascal, and 990 assembly language, as well as examples. The section concludes with directions for linking, patching, and executing PSC programs.

The discussions assume the reader has a programming knowledge of COBOL, FORTRAN, Pascal, or 990 Assembly Language. For descriptions of these languages and details on their implementation under DNOS, consult the applicable programmer's guides and reference manuals. It is also assumed that the programmer understands the structure and requirements of the host application that works with the PSC program.

5.2 PSC PROGRAM REQUIREMENTS

The PSC routines pass data between the PSC program and a host application. The following paragraphs describe the structures and characteristics of the data exchanged, indicating how the program should handle the data.

Using the PSC routines requires setting up the variables that become arguments. This involves declaring them at the appropriate part of the program and supplying them with values before calling them. Paragraphs 5.4.1 through 5.4.4 show the arguments for the routines, along with their data types, lengths, valid values, and uses in each of the programming languages — COBOL, FORTRAN, Pascal, and 990 Assembly Language.

The following paragraphs describe the data structures used by the PSC program and suggest ways to declare them in each of the languages. Beyond declaring the arguments and providing their values, all that remains is to follow the requirements of the programming language for calling an external routine.

When you activate the PSC program, it does not have access to the ICS line until it issues an Open call. If the Open call succeeds, the program can then transmit something to the host, call the Time-Out routine to wait for the host to respond, and issue a Get call to obtain a reply. PSC programs emulating printer stations can only receive data. The message and reply appear in the station buffer, exactly as the data passes between the host and a device serving as an ICS station.

5.2.1 PSC Station Buffers

The PSC program controls a station by manipulating its buffer. The buffer holds up to 1920 characters, representing the screen of an ICS display or printer station. The PSC program fills the buffer, retrieves data from it, and transmits the data to the host application.

Buffer operations address data by field and position. Fields are numbered sequentially from the top line of the screen to the bottom and from left to right on each line. The screen buffer begins with the first position on line 1 and ends with the last position on line 24 — an array of 1920 buffer addresses. Addresses range from 0 through 1919, and each address holds either a data character or an attribute character. (PSC treats nulls as data characters.) The PSC program that calls the Get or Fill commands needs its own buffer large enough to hold the largest field retrieved or modified. Since field sizes range from a few characters to the entire 1920-character buffer, the program buffer can be quite small or large, depending on the host application.

A data structure for the program buffer for each supported language is described below:

- **COBOL** — The buffer can be either a single DISPLAY data item with a picture large enough to hold the largest field, a group item with subitems for each field, or an array with enough word-sized data items to hold each character.
- **FORTRAN** — The buffer can be a one-dimensional integer array large enough to hold the largest field.
- **Pascal** — The buffer can be a string (packed array) with either a dynamic index or a static index large enough for the largest field.
- **990 assembly language** — The buffer should be a block of words large enough to hold the largest field.

5.2.2 Attribute Data

Three of the PSC routines provide access to attribute information from the station buffer. The Fill and Get routines return as the eighth argument (< a8>) the attribute character of the field modified or retrieved. The Attribute routine returns the attributes of up to 40 fields in a special attribute buffer. Your program reads the attribute character to determine the structure of the data. Your program cannot modify the attribute character.

NOTE

In this manual, the notation < a1> signifies the first argument, < a2> signifies the second argument, < a3> the third, and so on.

5.2.2.1 Attribute Argument Returned by Fill and Get Routines. The attribute information returned by Fill and Get routines arrives as two words as shown in Figure 5-2. The high-order byte of the first word contains the attribute character for the field. The low-order byte of the first word and both bytes of the second word are reserved for future enhancements and contain no information of use to the program.

	BYTE 1	BYTE 2
WORD 1	ATTRIBUTE	UNUSED
WORD 2	UNUSED	UNUSED

2281815

Figure 5-2. Attribute Argument

The attribute character has the following bit pattern:

Byte	Description										
0	Not used (depends on values for bits 2 through 7)										
1	Not used (depends on values for bits 2 through 7)										
2	Protection (0 = unprotected, 1 = protected; cannot be modified)										
3	Contents (0 = alphanumeric, 1 = numeric)										
4 – 5	Variable meaning as shown:										
	<table border="1"> <thead> <tr> <th>Value</th> <th>Meaning</th> </tr> </thead> <tbody> <tr> <td>00</td> <td>Display/non-select pen detectable</td> </tr> <tr> <td>01</td> <td>Display/select pen detectable</td> </tr> <tr> <td>10</td> <td>Initial display/select pen detectable</td> </tr> <tr> <td>11</td> <td>Non-display/non-print/non-select pen detectable</td> </tr> </tbody> </table>	Value	Meaning	00	Display/non-select pen detectable	01	Display/select pen detectable	10	Initial display/select pen detectable	11	Non-display/non-print/non-select pen detectable
Value	Meaning										
00	Display/non-select pen detectable										
01	Display/select pen detectable										
10	Initial display/select pen detectable										
11	Non-display/non-print/non-select pen detectable										
6	Always 0 (not used)										
7	Modification (0 = unmodified, 1 = modified)										

To access the attribute information returned, the calling program needs an appropriate data structure to use for the eighth argument of the call. The data structure used must reserve two words of memory (four bytes). If the PSC program reserves only one byte for the attribute argument, the PSC attribute call writes over the next three bytes in memory when it returns the eighth argument. The following list suggests data structures that each language can use for the attribute argument:

- COBOL — Synchronized block of two COMP4 data items, allowing the attribute information to be extracted, using binary arithmetic, from the first data item. COMP4 is a binary structure.
- FORTRAN — One-dimensional integer array with two elements.
- Pascal — Array of 32 Boolean variables, allowing attribute information to be extracted from the first eight bits.
- 990 Assembly Language — Four bytes beginning on an even word boundary, allowing attribute information to be extracted from the first byte.

5.2.2.2 Attribute List Returned by Retrieve Attributes. The attribute information returned by the Retrieve Attributes routine arrives in up to 120 words (or 240 characters), using three words to describe each field. As shown in Figure 5-3, the second and third words for each field contain the same information as that returned in the eighth arguments of the Fill and Get routines. The first word gives the buffer address of the beginning of the field, a decimal number in the range 0 to 1919.

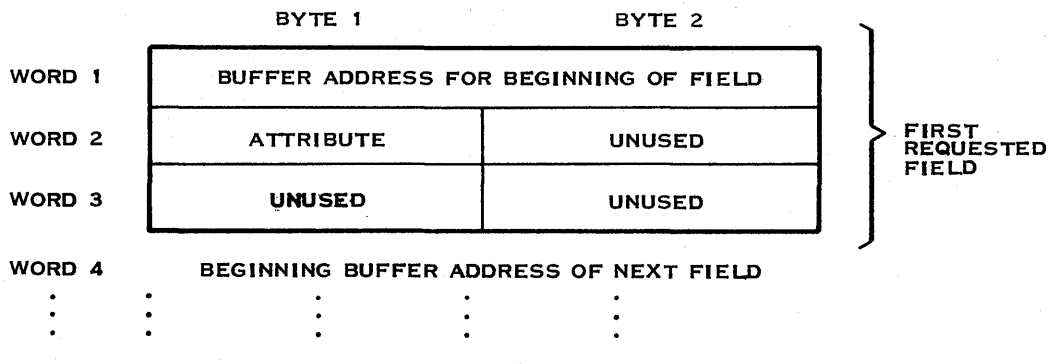


Figure 5-3. Attribute List

The location of the attribute character is one less than the beginning of the field, except for a field that begins in location 0. The unformatted screen is treated as a 1920-character field beginning at location 0 and having alphanumeric, unprotected, and display attributes. On a formatted screen, a field beginning at location 0 has its attribute character at location 1919.

NOTE

In this manual, the right angle bracket (>) preceding a value indicates a hexadecimal number.

5.2.3 Host Application Commands and Orders

Except for three printer orders, the PSC Emulator reads and interprets IBM commands and orders that the host sends. The PSC programmer who designs a program to serve as a PSC video station (type CRT) need not be concerned with these orders since the 3270-host transmits these embedded orders only to printer stations. The PSC program does not need to process the buffer control orders since the PSC Emulator translates them automatically. By the time the program gets control after data comes from the host, the data already occupies the correct field positions. The PSC Emulator helps in this by handling the I/O between the station and the host, executing commands, and manipulating transmission as needed. The PSC program has to take charge of handling the information that arrives at the station.

For a PSC program to emulate an ICS station when it communicates with a host application, the program must respond to the host exactly like an ICS station does — honoring the same commands and transmitting data in the same format. PSC routines help in this by handling the I/O between the station and the host, executing commands, and manipulating transmission as needed. However, the PSC program has to take charge of handling the information that arrives at the station. The first byte of a message contains one of the following command codes for the station:

- > F1 = Write (transmit information to the station buffer)
- > F5 = Erase/Write (erase buffer and fill with data)
- > F6 = Read Modified (transmit all modified fields to host)
- > 6F = Erase All Unprotected (erase all unprotected fields)

The Write command requires special attention, since it conveys a data stream that includes additional instructions for formatting the data — a write control character (WCC) at the beginning of the data stream and 3270 orders embedded in the data. The data stream begins with a WCC byte that specifies additional operations to accompany the write. The PSC Emulator extracts the WCC and performs any operations requested before returning control to the PSC program. Therefore, the data that arrives in the station buffer already reflects the WCC operations requested by the host. The WCC bit pattern and associated operations are as follows:

Bit	Meaning
0	Determined by contents of bits 2 through 7.
1	Always 1 (WCC reset bit).
2-3	Print Line Length. 00 = 132-character line 01 = 40-character line 10 = 64-character line 11 = 80-character line
4	Start Printer. 0 = Do not start printer. 1 = Start printer at end of write.
5	Sound Alarm. 0 = Do not sound alarm. 1 = Sound alarm at end of write.

Bit	Meaning
6	Reset Keyboard. 0 = Do not reset. 1 = Reset keyboard. Restore operation of the keyboard locked by input inhibited condition.
7	Reset Modified. 0 = Do not reset. 1 = Reset MDT bit. Reset all MDT (Modified Data Tag) bits in the selected device's existing data buffer before any data is written or orders are executed.

Unlike host application commands, orders arrive embedded in the transmitted data. Each order occupies at least one byte in the station buffer. The emulated IBM station supports six orders that manipulate the display station buffer and three that control print format. Before it returns control to the PSC program, the PSC Emulator carries out either the buffer orders or the format orders, depending on whether the PSC program is emulating a display station or a printer station.

The list of the six orders that manipulate the display station buffer follows:

EBCDIC Code	Order	Description
>1D	SF	Start Field — Used to indicate that the next byte holds an attribute character.
>11	SBA	Set Buffer Address — Used to set the address by the following two bytes as the buffer location for the next operation.
>13	IC	Insert Cursor — Used to position the cursor to the current buffer address.
>05	PT	Program Tab — Used to advance the buffer address to the next beginning-of-field.
>3C	RA	Repeat to Address — Used to fill all buffer locations (attribute characters included) from the current buffer address up to a specified stop address with a designated fill character. The two bytes following the RA order give the stop address, and the third byte following RA gives the ASCII code of the fill character.
>12	EUA	Erase Unprotected to Address — Used to clear (set to null) all unprotected positions beginning with the current buffer address and continuing up to the stop address specified in two bytes following the EUA order (this does not include attribute byte).

The PSC Emulator removes the command code from the message and performs the operation indicated. The PSC Emulator does not translate the following three printer orders:

EBCDIC Code	Order	Description
>15	NL	New Line — Used for printouts with no line lengths specified. The ASCII equivalent for the NL order is the Line Feed character, >0A.
>0C	FF	Form Feed — Used at the beginning of a printed line to index the form vertically. Some applications may require that the FF order be replaced by the ASCII Vertical Tab order, >0B.
>19	EM	End of Message — Terminates printouts with no line lengths specified. The ASCII equivalent for this order is >13. Some applications may require that the EM order be replaced by the ASCII Form Feed order, >0C.

These orders are not translated so that the PSC program can choose how to handle them. The PSC program can recognize them by their EBCDIC values: >15, >0C, and >19. If the program performs its own printer I/O, it can either replace the orders in the buffer with the appropriate ASCII codes or translate them to print as blanks. The NL order marks the end of a line and needs to be replaced by a Line Feed and carriage return character (>0A0D) before the message goes to the printer unless the PSC program handles line breaks itself. The FF order causes printing to resume at a new form. The EM order marks the end of the message and can be replaced by a Form Feed (>0C) to advance to the next page or a Deselect character (>13) to terminate the printer operation. The PSC Emulator fills the rest of the station buffer with nulls when it detects the EM order.

If you write your PSC program in COBOL, FORTRAN, or Pascal, you cannot directly replace these printer orders with ASCII equivalents. You must consider how the language your program uses handles output to whatever device your program uses.

5.2.4 ICS Station Selection

Every call to a PSC routine specifies the control unit address of the ICS line and the terminal address of the station emulated. In each PSC routine, the first argument <a1> consists of two words containing these addresses or a specification for auto-select. Only the Open routine selects addresses automatically. The other routines use the addresses selected by the Open routine. The first word contains three flags that specify whether to use the local format file and either the IBM control unit address, the device name, or a specification for auto-select. The second word contains the device type and either the terminal address or a specification that ICS select an available address.

5.2.4.1 First Word. The PSC Emulator treats the first word of this argument as two separate bytes as shown in Figure 5-4. Bit 7 designates the least significant byte, and bit 0 designates the most significant byte. The first byte contains three flags that specify the following options:

- Use or do not use the local format file
- Identifies the type of value to be found in byte two
- Automatically selects a control unit address

Bit 5 is the flag specifying whether to use the local format file. If bit 5 is a one, the PSC Emulator stores the contents of the local format file for the terminal address used into the station buffer when the Open is performed. If this bit is zero, the local format file is not used and the station buffer is cleared to nulls after the Open is performed.

Bit 6 indicates whether ICS automatically selects the communications line or uses the value in the second byte. One indicates auto-select. Zero indicates that ICS uses the value found in the second byte.

Usually, if the program uses auto-select for the communications line, it also uses auto-select for the terminal address. If the program specifies auto-select for both, ICS searches for the first available active communications line that has an available terminal address of the appropriate type.

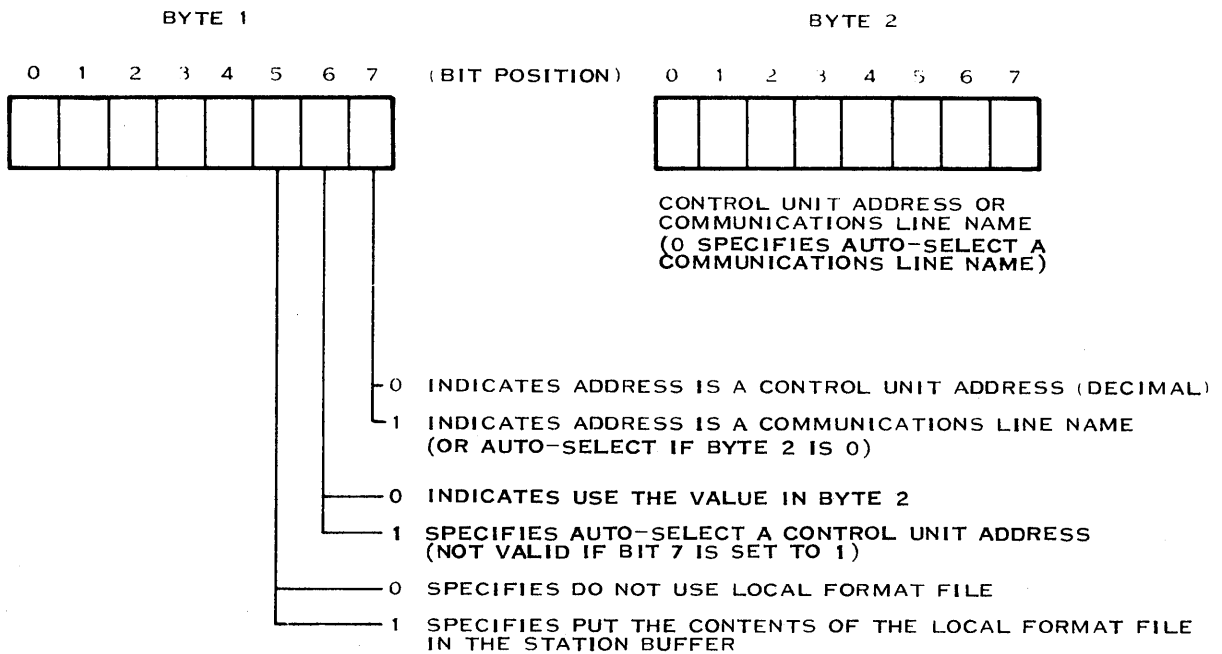
If the program specifies auto-select for the communications line but specifies a particular terminal address, ICS selects the first available active line at which the specified terminal address of the specified type (see second word) is available.

Bit 7 of the first byte contains a flag indicating to ICS the kind of value in the second byte. Zero in bit 7 indicates that the value in the second byte contains a decimal control unit address. One indicates that the second byte contains the numerical portion of the communications line name or, if the second byte contains a zero, that ICS selects the first available communications line.

The second byte is either a decimal control unit address, the numerical portion of the communications line name, or a zero for automatic selection. You can use any one to identify the communications controller port (communications line name), the communications line, and consequently, the host.

Valid control unit addresses range in value from 0 through 31. ICS maps this decimal value to an EBCDIC IBM control unit address according to Table 2-1.

The communications line name address is the numerical portion of the device name for a communications device. For example, CM03 is identified as 03. If bit 7 of byte 1 is a one and byte 2 contains a zero, ICS selects the first available communications line.



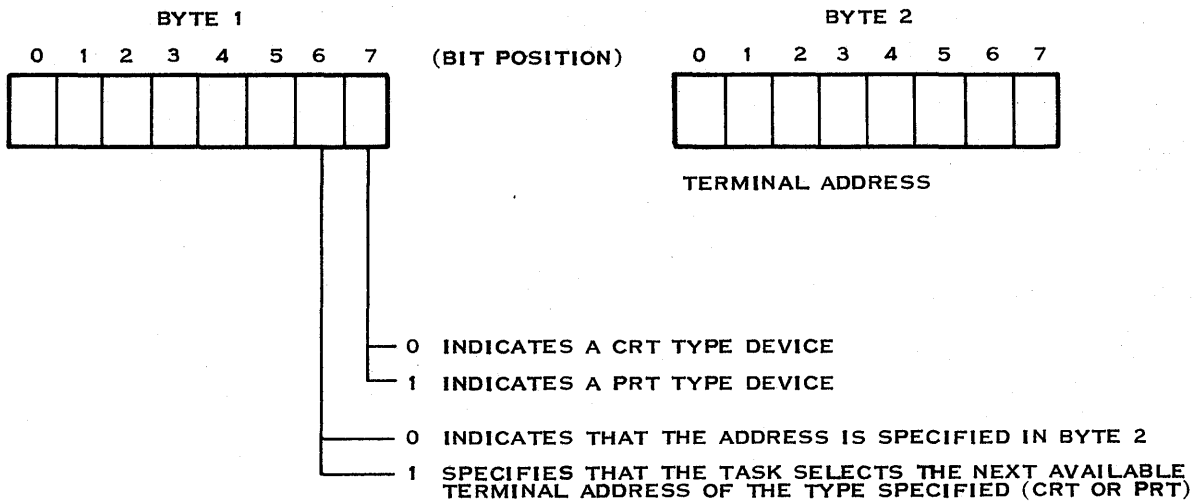
2281817

Figure 5-4. Word One of the First Argument

5.2.4.2 Second Word. As with the first word of this argument, ICS treats the second word as two separate bytes. As shown in Figure 5-5, the first byte contains two flags. Bit 6 contains the flag that designates whether ICS makes the device selection or uses the terminal address in the second byte. Bit 7 contains the second flag that specifies the terminal type, either video station (CRT) or printer station (PRT). The PSC Emulator examines only bit 7 of the second word if bit 6 is set to one.

The second byte contains the terminal address that ICS maps to an IBM EBCDIC address. Like the communications line address, ICS maps the terminal address according to Table 2-1.

When the PSC program calls the Open routine, it opens the ICS terminal address on a communications line. The terminal address stays open until the program calls the Close routine. Between the Open and Close calls, the PSC Emulator uses the ICS station to exchange data with the host via ICS.



2281818

Figure 5-5. Word Two of the First Argument

The terminal address is one of the addresses reserved during ICS generation. ICS can select the terminal address automatically as it can when activating a physical ICS station. A one in bit 6 of byte 1 of the second word in < a1> specifies to ICS that it is to make the terminal address selection of the type specified in bit 7 of the same byte. If your program specifies auto-select, the PSC Emulator returns the control unit and terminal addresses that it has selected in < a1> upon successful activation of the terminal address. A zero in bit 6 specifies that the terminal address to be used is in byte 2 of the second word of < a1>.

The PSC program can emulate either an ICS display or printer station. The type indicator bit is the least significant bit of the high-order byte in the second word of < a1>. Zero indicates a display station. One indicates a printer station.

NOTE

A user-written PSC application can open only one ICS terminal address at a given time. Your application can use more than one ICS terminal address, but it must close the first one before it can open the second.

The following are suggestions for each supported programming language for the data structures your program can use for the first argument:

- COBOL — Data group consisting of two COMP-1 data items
- FORTRAN — A one-dimensional integer array with two elements
- Pascal — Address data type consisting of four byte-sized variables
- 990 assembly language — Four bytes of data

5.2.5 Completion Codes

PSC routines return completion codes that report the success or failure of their operations. A zero completion code always indicates success, while nonzero codes indicate error conditions. Appendix B lists all of the completion codes for PSC, along with an explanation of the condition. The PSC program should test the condition code from a routine immediately after the call. Since most PSC errors are not recorded in the system log or elsewhere, the program should at least record the station address and completion code as part of its error-handling routine. If a nonrecoverable error occurs, the abnormal exit routine of the PSC program should record these values and always issue a Close call to PSC.

Some errors returned to video stations and the system log are identified as PSC errors by their prefix, *XPSC. The DNOS System Command Interpreter (SCI) actually detects and reports these errors. They usually occur when a bad parameter is entered in response to one of the prompts. The first paragraph of Appendix B explains these messages.

5.3 PSC ROUTINES

The following paragraphs discuss the PSC routines individually. This discussion explains how the PSC routines control the flow of information between the station and the host. These routines enable a program to open and close stations, read and write data to the station buffer, and send and receive data from the host application.

The description of a routine consists of the following five parts:

- Name of the routine and a brief functional description
- Syntax for calling the routine from COBOL, FORTRAN, Pascal, and 990 assembly language programs
- Description of each argument
- Explanation of the results of the call
- Examples in COBOL, FORTRAN, Pascal, and 990 assembly language

Language-specific portions of the description are labeled to avoid confusion. The call statements show symbolic arguments enclosed in angle brackets (< >). In actual programs, however, suitable programming language variables replace the symbolic arguments, as shown in the examples presented for each call. Paragraphs 5.4.1 through 5.4.4 provide capsule descriptions of the arguments for each routine called in each programming language.

NOTE

Some of the routines have dummy arguments that preserve compatibility with other PSC implementations. You must reserve memory space for these arguments in the calls, but you need not supply any values.

5.3.1 Open Routine — Open Station for PSC Use

The Open routine requests exclusive access to an ICS station and sets its initial mode of operation. This call is always required before your program can call any other PSC routine. If the user-written task is bid from the Execute Programmed Station Control (XPSC) command procedure, the station specified by the Open routine is already active, and, to avoid an error, < a2> should specify *Do not activate*. If the user-written PSC task is not bid by XPSC, < a2> of the Open routine must specify *Activate*. Figure 5-6 contrasts the program flow of a PSC program activated by XPSC with that of a PSC program activated by one of the language activation procedures.

The Close routine uses the same kind of first argument to specify *Terminate* or *Do not terminate*. (See paragraph 5.3.7.) Most applications can use either method. Paragraph 5.4 contrasts both methods and discusses the merits of each.

5.3.1.1 Open Call Statements. The following examples demonstrate the Open call statement in each of the supported programming languages.

COBOL:

```
CALL 'ICS$OPN' USING <a1> <a2> <a3> <a4>.
```

FORTRAN:

```
CALL IFSOPN (<a1>, <a2>, <a3>, <a4>)
```

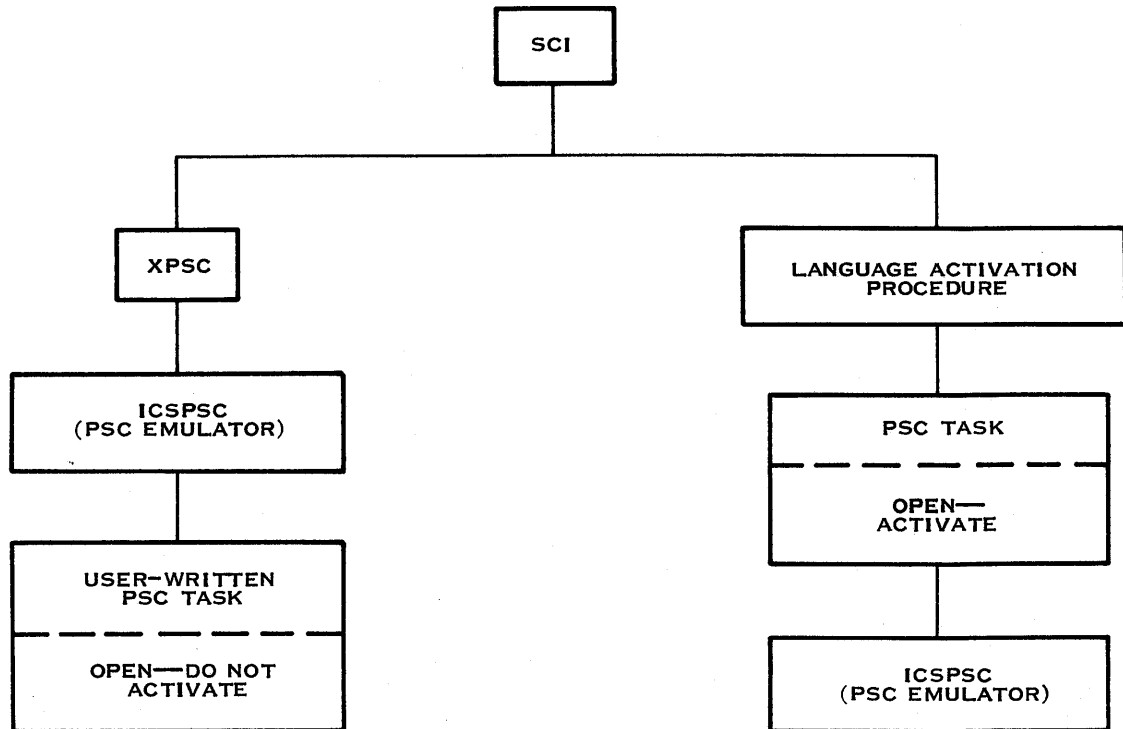
Pascal:

```
IFSOPN (<a1>, <a2>, <a3>, <a4>);
```

990 Assembly Language:

```

        LI    R0,ARGLST  POINTER TO ARGUMENT LIST
        BLWP @IC$$OPN  SUBROUTINE LINKAGE
        .
        .
        .
ARGLST  DATA  4*2      FOUR TWO-BYTE POINTERS TO ARGUMENTS
        DATA <a1>     COMM LINE & TERMINAL ID POINTER
        DATA <a2>     ACTIVATE FLAG POINTER
        DATA <a3>     NOT USED, DUMMY POINTER
        DATA <a4>     COMPLETION CODE POINTER
    
```



2281819

Figure 5-6. PSC Activation Flow

5.3.1.2 Arguments for the Open Routine. The following list describes the arguments for the Open routine:

- < a1> is the control unit address and terminal address of the PSC program. This argument consists of two words. The first word contains the control unit address or a specification to let ICS choose the control unit address. It also contains a flag that indicates whether PSC fills the station buffer with the contents of the local format file. The second word contains the terminal address or a specification to let ICS choose the terminal address. It also specifies the type (CRT or PRT) of the terminal address. Paragraph 5.2.4 describes this argument in detail.
- < a2> indicates whether to activate the station, as follows:
 - 0 = The station is already active, so do not attempt to activate it. Use this when the user's task is activated with the XPSC command or when the station was previously closed with the *Do not terminate* option. (See paragraph 5.3.7.)
 - 1 = Activate the station.
- < a3> is a dummy argument, not used by the routine but required for compatibility with previous releases.
- < a4> is a completion code. The routine returns a zero completion code when the operation succeeds. For nonzero codes, see Appendix B.

5.3.1.3 Results of the Open. A successful call to the Open routine makes the ICS communications line and terminal address available to the PSC program. The value of the second argument determines whether the routine attempts to activate the PSC Emulator. Zero indicates that the PSC Emulator is already active and requires no further action. One indicates that the PSC Emulator is to be activated. After a successful call to the Open routine, the program can send and receive data across the ICS line by issuing calls to other PSC routines.

If < a1> specifies auto-select, the PSC Emulator returns the control unit and terminal addresses selected in < a1>. The only other argument that the PSC Emulator returns is < a4>. The second and third arguments are not changed. Be sure and check the return code and handle any detected errors.

If the XPSC command procedure bids the user-written task, you should specify *Do not activate* when calling the Open routine by assigning < a2> a value of zero. If you activate your PSC program by using XPSC and enter a null value in response to the COMMUNICATIONS LINE NAME and TERMINAL ADDRESS prompts to let ICS choose the communications line and terminal address, you should also specify auto-select in < a1> when your PSC program calls the Open routine.

5.3.1.4 Examples of the Open Call. The following examples demonstrate the Open call statement in each of the supported programming languages.

COBOL:

```
CALL 'IC$OPN' USING STATION-ADDRESS ACTIVATE DUMMY COMPLETION-KODE.
```

FORTRAN:

```
CALL IFSOPN (STADDR, OPNACT, OPNDUM, CCODE)
```

Pascal:

```
IFSOPN (ADDR, ACTIVATE, DUMMY, CCODE);
```

990 Assembly Language:

```

                LI    R0,OPNARG    POINTER TO ARGUMENT LIST
                BLWP  @IC$OPN     SUBROUTINE LINKAGE
                .
                .
                .
OPNARG          DATA  4*2        FOUR TWO-BYTE POINTERS TO ARGUMENTS
                DATA  OPN1       COMM LINE & TERMINAL ID POINTER
                DATA  OPN2       ACTIVATE FLAG POINTER
                DATA  OPN3       DUMMY POINTER; DUMMY IS NOT USED
                DATA  OPNCC      COMPLETION CODE POINTER
                .
                .
                .
OPN1            DATA  >0200      LET ICS SELECT THE COMM LINE
                DATA  >0200      LET ICS SELECT A TERMINAL ADDRESS OF
                                THE CRT TYPE
OPN2            DATA  0          DO NOT ACTIVATE
OPN3            DATA  0          DUMMY; NOT USED
OPNCC           BSS    2          COMPLETION CODE

```

5.3.2 Fill Routine — Fill Station Buffer

This routine places ASCII data into a specified field of a program buffer. The call indicates how many characters go to the buffer, which field receives them, where they go in the field, and what happens to the buffer next. The arguments returned indicate the length of the fill, the success of the operation, and the attributes of the field to be filled.

5.3.2.1 Fill Call Statements. The following examples demonstrate the Fill call statement in each of the supported programming languages.

COBOL:

```
CALL 'IC$$FIL' USING <a1>, <a2>, <a3>, <a4>, <a5>, <a6>, <a7>, <a8>.
```

FORTRAN:

```
CALL IFSFIL (<a1>, <a2>, <a3>, <a4>, <a5>, <a6>, <a7>, <a8>)
```

Pascal:

```
IFSFIL (<a1>, <a2>, <a3>, <a4>, <a5>, <a6>, <a7>, <a8>);
```

990 Assembly Language:

```

        LI      R0, FILARG  POINTER TO ARGUMENTS LIST
        BLWP   @IC$FIL    SUBROUTINE LINKAGE
        .
        .
        .
FILARG  DATA  8*2        EIGHT TWO-BYTE POINTERS TO ARGUMENTS
        DATA  <a1>      COMM LINE & TERMINAL ID POINTER
        DATA  <a2>      PROGRAM STATION BUFFER POINTER
        DATA  <a3>      LENGTH OF DATA TO FILL POINTER
        DATA  <a4>      FIELD NUMBER POINTER
        DATA  <a5>      POSITION WITHIN THE FIELD POINTER
        DATA  <a6>      NEXT ACTION POINTER
        DATA  <a7>      COMPLETION CODE POINTER
        DATA  <a8>      ATTRIBUTE POINTER

```

5.3.2.2 Arguments for the Fill Routine. The following list describes the arguments for the Fill routine:

- < a1 > is the control unit address and terminal address of the PSC program. This argument consists of two words. The first word contains the control unit address or a specification to let ICS choose the control unit address. It also contains a flag that indicates whether PSC fills the station buffer with the contents of the local format file. The second word contains the terminal address or a specification to let ICS choose the terminal address. It also specifies the type (CRT or PRT) of the terminal address. Paragraph 5.2.4 describes this argument in detail.
- < a2 > is the name of the buffer in the PSC program that holds the fill data. This argument must be an alphanumeric data type and must start on an even address boundary.
- < a3 > is the length of the fill data in characters. This number must be no greater than the length of the field filled and no more than 240 characters in any case.
- < a4 > is the number of the field to be filled, counting from the beginning of the station buffer.

- < a5> is the position in the field where the fill starts. The fill begins at this character location in the designated field, counting from the first position after the attribute character. The first position is one.
- < a6> is an indicator for the action to follow the fill operation, as follows:
- 0 = No other action is required.
 - 1 = Transmit any modified data of the program buffer to the host. If the terminal address used by the PSC program is the PRT (printer) type, ICS transmits a message to the host that indicates the host can send more data.
 - 3 = Clear all unprotected fields in the program buffer. If this is specified, < a2> through < a5> are ignored, and the attribute byte of the first unprotected field is returned in < a8> .
- < a7> is a completion code returned after the fill. Zero indicates success. Nonzero completion codes indicate an error condition. If the PSC Emulator detects an error, it does not modify the station buffer. Refer to Appendix B for a list of messages and their meanings.
- < a8> is the attribute information for the field. This argument consists of two words. The high-order byte of the first word returns the attribute byte for the field filled. The low-order byte of the first word and both bytes of the second word are reserved for future enhancements. If clear (option three of < a6>) is specified, the attribute byte of the first unprotected field is returned in < a8> .

5.3.2.3 Results of the Fill. Calling the Fill routine sends the fill data to the station buffer. The data goes into the designated field and starts at the designated position. This modifies the field, and the host application receives the data on the next transmission. If the fill data overflows the field, the remainder is not used. If the fill data runs out before the field is full, the PSC Emulator does not modify the remainder of the field.

The Fill routine returns four values of interest:

- The third argument indicates the length of the data filled. You can use this value to check the success of the operation.
- The fifth argument indicates the position in the field where the fill data stopped. If the end of the field was reached, a 1 is returned. If the field was exceeded, a 2 is returned. Neither condition causes a nonzero completion code to be returned in the seventh argument.
- The seventh argument returns a completion code for the operation, with zero indicating success and nonzero codes (listed in Appendix B) indicating error conditions.
- The eighth argument returns with the attribute byte for the field filled as the high-order byte in the first word. Refer to paragraph 5.2.2.

5.3.2.4 Examples of the Fill Call. The following examples demonstrate the Fill call statement in each of the supported programming languages.

COBOL:

```
CALL 'IC$FIL' USING STATION-ADDRESS FILL-BUFFER FILL-HOW-MANY FILL-
FIELD FILL-BEGIN FILL-ACTION COMPLETION-KODE FILL-
ATTRIBUTE
```

FORTRAN:

```
CALL IFSFIL (STADDR,BUFFER,FILLEN,FILFLD,FIL1,FILACT,CCODE,FILATR)
```

Pascal:

```
IFSFIL (ADDR,B,FILL_COUNT,FILL_FIELD,FILL_START,FILL_ACTION,
CCODE,ATTR);
```

990 Assembly Language:

```
LI    R0,FILARG  POINTER TO ARGUMENTS LIST
BLWP  @IC$FIL   SUBROUTINE LINKAGE
.
.
.
FILARG  DATA  8*2      EIGHT 2-BYTE POINTERS TO ARGUMENTS
        DATA  OPN1     COMM LINE & TERMINAL ID POINTER
        DATA  FIL2     PROGRAM STATION BUFFER POINTER
        DATA  FIL3     DATA LENGTH POINTER
        DATA  FIL4     FIRST FIELD POINTER
        DATA  FIL5     START POSITION POINTER
        DATA  FIL6     LAST FIELD POINTER
        DATA  FILCC    COMPLETION CODE POINTER
        DATA  FIL8     ATTRIBUTE POINTER
.
.
.
OPN1    DATA  >0200    USE THE SAME AS USED FOR THE OPEN
        DATA  >0200    ROUTINE; AUTOMATICALLY SELECT
                        CRT TYPE
FIL2    BSS    2        BEGINNING OF STATION BUFFER
FIL3    BSS    2        LENGTH OF THE DATA TO FILL
FIL4    BSS    2        FIRST FIELD TO FILL
FIL5    BSS    2        POSITION WITHIN THE FIELD TO
                        BEGIN FILL
FIL6    BSS    2        LAST FIELD TO FILL
FILCC   BSS    2        COMPLETION CODE
FIL8    BSS    2        FIRST WORD OF ATTRIBUTE ARGUMENT
        DATA  0        2ND WORD OF ATTRIBUTE; NOT USED
```


5.3.3 Get Routine — Get Data from the Station Buffer

The Get routine enables the PSC program to retrieve information from a station buffer. The program specifies which field, where in the field to begin, and how many characters to retrieve. The routine returns the field specified, the number of characters retrieved, a completion code, and the attributes of the field.

5.3.3.1 Get Call Statements. The following examples demonstrate the Get call statement in each of the supported programming languages.

COBOL:

```
CALL 'IC$GET' USING <a1> <a2> <a3> <a4> <a5> <a6> <a7> <a8>.
```

FORTRAN:

```
CALL IFSGET (<a1>,<a2>,<a3>,<a4>,<a5>,<a6>,<a7>,<a8>)
```

Pascal:

```
IFSGET (<a1>,<a2>,<a3>,<a4>,<a5>,<a6>,<a7>,<a8>);
```

990 Assembly Language:

```

                LI    R0,GETARG  POINTER TO ARGUMENTS LIST
                BLWP @IC$GET  SUBROUTINE LINKAGE
                .
                .
                .
GETARG          DATA  8*2      EIGHT TWO-BYTE POINTERS TO ARGUMENTS
                DATA  <a1>    COMM LINE & TERMINAL ID POINTER
                DATA  <a2>    PROGRAM STATION BUFFER POINTER
                DATA  <a3>    DATA LENGTH POINTER
                DATA  <a4>    FIRST FIELD POINTER
                DATA  <a5>    START POSITION POINTER
                DATA  <a6>    NEXT ACTION POINTER
                DATA  <a7>    COMPLETION CODE POINTER
                DATA  <a8>    ATTRIBUTE POINTER
    
```

5.3.3.2 Arguments for the Get Routine. The following list describes the arguments for the Get routine:

- < a1> is the control unit address and terminal address of the PSC program. This argument consists of two words. The first word contains the control unit address or a specification to let ICS choose the control unit address. It also contains a flag that indicates whether PSC fills the station buffer with the contents of the local format file. The second word contains the terminal address or a specification to let ICS choose the terminal address. It also specifies the type (CRT or PRT) of the terminal address. Paragraph 5.2.4 describes this argument in detail.
- < a2> is the name of the buffer in the PSC program to receive the data. The value for < a2> must be an alphanumeric data type and must begin on an even address boundary.
- < a3> is the length of the data retrieved. This value determines the number of characters that the routine copies from the station buffer into the program buffer indicated by < a2>. The value supplied should be no greater than the size of the buffer receiving the data and, in any case, no more than 240 characters. Fields that contain more than 240 characters require successive Get calls.
- < a4> is the number of the first field containing the data to retrieve, counting from the first field in the station buffer.
- < a5> is the position where the retrieval begins. This number tells the Get routine where to start retrieving data in the field that the fourth argument indicates. Each field begins counting with one (not a buffer position).
- < a6> is an indicator for the action to follow the Get operation, as follows:
 - 0 = No action is required.
 - 1 = Transmit the station buffer to the host. If the terminal address used by the PSC program is the PRT (printer) type, ICS transmits a device end to the host and no data is transmitted.
 - 3 = Clear all unprotected fields in the station buffer.
- < a7> is the completion code returned after the Get. Zero indicates success. Nonzero completion codes indicate an error condition. Refer to Appendix B for a list of messages and their meanings.
- < a8> is the attribute information for the field retrieved. This argument consists of two words. The high-order byte of the first word returns the attribute byte for the field. The low-order byte of the first word and both bytes of the second word are reserved for future enhancements.

5.3.3.3 Results of the Get. The Get routine attempts to retrieve data from a specified station, field, and position. The operation succeeds when the field contains data at the specified position. The program buffer indicated by < a2> receives the data so that the program can work on it.

Four of the arguments return values of interest, as follows:

- The third argument indicates the length of the data retrieved. You can use this value to check the success of the operation.
- The fifth argument indicates the position in the field where the retrieval stopped. If positive, it represents the position after the last character retrieved. This is the position where the next retrieval begins when the program retrieves more of the field. If the fifth argument returns – 1, it means the last character retrieved is the last character in the field. If the fifth argument returns – 2, the routine came to the end of the field before it retrieved all of the characters requested.
- The seventh argument returns a completion code back to the PSC program. Zero indicates success. Nonzero completion codes indicate an error condition. In particular, a >0D code indicates that no new data is available for retrieval. Refer to Appendix B for a list of messages and their meanings.
- The eighth argument returns the attribute byte for the field that was retrieved as the high-order byte in the first word. Refer to paragraph 5.2.2.

5.3.3.4 Examples of the Get Call. The following examples demonstrate the Get call statement in each of the supported programming languages.

COBOL:

```
CALL ''IC$GET'' USING STATION-ADDRESS GET-BUFFER GET-HOW-MANY
                    GET-FIELD GET-BEGIN GET-ACTION COMPLETION-KODE
                    GET-ATTRIBUTE.
```

FORTRAN:

```
CALL IFSGET (STADDR, BUFFER, GETLEN, GETFLD, GET1, GETACT,
            CCODE, ATR)
```

Pascal:

```
IFSGET (ADDR,B,GET_COUNT,GET_FIELD,GET_START,GET_ACT,CCODE,GET_ATR);
```

990 Assembly Language:

```

        LI    R0,GETARG  POINTER TO ARGUMENTS LIST
        BLWP  @IC$GET   SUBROUTINE LINKAGE
        .
        .
GETARG  DATA  8*2      EIGHT TWO-BYTE POINTERS TO ARGUMENTS
        DATA  OPN1     COMM LINE & TERMINAL ID POINTER
        DATA  GET2     PROGRAM STATION BUFFER POINTER
        DATA  GET3     DATA LENGTH POINTER
        DATA  GET4     FIRST FIELD POINTER
        DATA  GET5     START POSITION POINTER
        DATA  GET6     NEXT ACTION POINTER
        DATA  GETCC    COMPLETION CODE POINTER
        DATA  GET8     ATTRIBUTE POINTER
        .
        .
OPN1    DATA  >0200   USE THE SAME AS USED FOR THE OPEN
        DATA  >0200   ROUTINE; AUTOMATICALLY SELECT CRT
                        TYPE
GET2    BSS    2       BEGINNING OF STATION BUFFER
GET3    BSS    2       LENGTH OF DATA TO GET
GET4    BSS    2       FIRST FIELD TO GET
GET5    BSS    2       POSITION WITHIN FIELD TO BEGIN GET
GET6    BSS    2       ACTION TO BE TAKEN AFTER GET
GETCC   DATA  0       COMPLETION CODE
GET8    BSS    4       2ND WORD ATTRIBUTE ARGUMENT; NOT USED
    
```

5.3.4 Retrieve Attributes Routine — Return Field Attributes

The Retrieve Attributes routine returns the attributes of up to 40 fields in the station buffer. In addition to the list of attributes, this routine returns arguments set to the number of the field following the last attribute retrieved, the current position of the cursor, a count of attribute characters returned, a completion code, and the Write Control Character (WCC) of the current screen image.

The Retrieve Attributes routine optionally returns the value of the Write Control Character (WCC) that the host sends with each Write command. The purpose of this routine is to aid you in writing new PSC applications or help you convert existing PSC applications when the host application changes formats.

5.3.4.1 Retrieve Attributes Call Statements. The following examples demonstrate the Retrieve Attributes call statement in each of the supported programming languages.

COBOL:

```
CALL 'ICSATR' USING <a1> <a2> <a3> <a4> <a5> <a6> <a7> <a8>.
```

FORTRAN:

```
CALL IFSATR (<a1>,<a2>,<a3>,<a4>,<a5>,<a6>,<a7>,<a8>)
```

Pascal:

```
IFSATR (<a1>,<a2>,<a3>,<a4>,<a5>,<a6>,<a7>,<a8>);
```

990 Assembly Language:

```

                LI    R0,ATRARG  POINTER TO ARGUMENTS LIST
                BLWP @ICSATR  SUBROUTINE LINKAGE
                .
                .
                .
ATRARG          DATA 8*2      EIGHT TWO-BYTE POINTERS TO ARGUMENTS
                DATA <a1>    COMM LINE & TERMINAL ID POINTER
                DATA <a2>    ATTRIBUTE LIST POINTER
                DATA <a3>    ATTRIBUTE LIST LENGTH POINTER
                DATA <a4>    FIRST FIELD POINTER
                DATA <a5>    CURSOR POSITION POINTER
                DATA <a6>    RETURN LENGTH POINTER
                DATA <a7>    COMPLETION CODE POINTER
                DATA <a8>    WCC POINTER

```

5.3.4.2 Arguments for the Retrieve Attributes Routine. The following list describes the arguments for the Retrieve Attributes routine:

- < a1 > is the control unit address and terminal address of the PSC program. This argument consists of two words. The first word contains the control unit address or a specification to let ICS choose the control unit address. It also contains a flag that indicates whether PSC fills the station buffer with the contents of the local format file. The second word contains the terminal address or a specification to let ICS choose the terminal address. It also specifies the type (CRT or PRT) of the terminal address. Paragraph 5.2.4 describes this argument in detail.
- < a2 > is the name of the list in the PSC program to receive the attribute information. A maximum of 120 words is returned, three words for each attribute.
- < a3 > is the number of attribute characters requested. This value can be any value from 1 to 40.

- < a4> is the number of the first field whose attribute character is retrieved.
- < a5> is the buffer address of the cursor (from 0 to 1919) that the routine returns if the program reserved a video terminal address, or the printer line length if the program reserved a printer terminal address.
- < a6> is the number of attribute characters actually retrieved (from 1 to 40). The routine returns this argument.
- < a7> is the completion code. Zero indicates success. Nonzero completion codes indicate an error condition. (Refer to Appendix B for a list of error messages and their meanings.)
- < a8> is the WCC of the last write command that the host sent. (Refer to the *IBM 3270 Information Display System Component Description* for a complete description of the WCC.) This argument requires a full word (16 bits) of memory, but the WCC occupies only the right (least significant) byte.

5.3.4.3 Results of the Retrieve Attributes. The Retrieve Attributes routine inspects the station buffer and returns a list of attribute characters structured as described in paragraph 5.2.2. Each attribute character requires three words. The first word locates the beginning of the field in the station buffer, the high-order byte of the second word contains the attribute character, and the remainder is not used.

In addition to the list of attribute characters, the routine returns five values of interest, as follows:

- The fourth argument, which the program sets to the field number of the first attribute wanted, returns the number of the field following the last attribute character retrieved. This facilitates a subsequent call to retrieve attributes for the following field. If the last attribute character retrieved is the last one in the buffer, this argument returns - 1. If the attribute characters in the buffer are exhausted before the number requested by the third argument can be retrieved, this argument returns - 2.
- The fifth argument returns one of the following:
 - If your program emulates a video station, < a5> returns the current location of the cursor. Though the PSC program need not position the cursor to modify a field, the 3270 orders that the host sends manipulate the cursor. This argument allows the program to locate the cursor so that it can decide where to perform the next operation, if necessary.
 - If your program emulates a printer station, < a5> returns the printer line length.
- The sixth argument returns the number of attributes actually retrieved. This number matches the number requested by the third argument unless the routine comes to the end of the buffer before retrieving the number requested.

- The seventh argument returns a completion code. Zero means a successful transmission. A nonzero code indicates an error condition. (Refer to Appendix B for a list of error messages and their meanings.)
- The eighth argument returns the WCC of the last Write command sent by the host. This argument is optional.

5.3.4.4 Examples of the Retrieve Attributes Call. The following examples demonstrate the Retrieve Attributes call statement in each of the supported programming languages.

COBOL:

```
CALL 'IC$ATR' USING STATION-ADDRESS ATR-LIST ATR-HOW-MANY ATR-BEGIN
ATR-CURSOR COMPLETION-KODE ATR-RETURNED WCC.
```

FORTRAN:

```
CALL IFSATR (ADDR, ATRB, ATRNUMER, ATRSTRT, ATRCSR, CODE, ATRBKC,
WCC)
```

Pascal:

```
IFSATR (ADDR, ATR_B, ATR_NUMBER, ATR_START, ATR_CURSOR, CCODE,
ATR_BACK, WCC);
```

990 Assembly Language:

```

                LI    R0,ATRARG  POINTER TO ARGUMENTS LIST
                BLWP @IC$ATR  SUBROUTINE LINKAGE
                .
                .
                .
ATRARG          DATA  8*2      EIGHT TWO-BYTE POINTERS TO ARGUMENTS
                DATA  OPN1     COMM LINE & TERMINAL ID POINTER
                DATA  ATR2     ATTRIBUTE LIST POINTER
                DATA  ATR3     ATTRIBUTE LIST LENGTH POINTER
                DATA  ATR4     START FIELD POINTER
                DATA  ATR5     CURSOR POSITION POINTER
                DATA  ATR6     RETURN LENGTH POINTER
                DATA  ATRCC    COMPLETION CODE POINTER
                DATA  ATR8     WCC POINTER
                .
                .
                .
OPN1            DATA  >0200    USE THE SAME AS USED BY THE OPEN
                DATA  >0200    ROUTINE; AUTOMATICALLY SELECT CRT
                                TYPE
ATR2            BSS    240     ATTRIBUTE LIST;
ATR3            BSS    2       ATTRIBUTE LIST LENGTH
ATR4            BSS    2       START FIELD
ATR5            BSS    2       CURSOR POSITION
ATR6            BSS    2       RETURN LENGTH
ATRCC           BSS    2       COMPLETION CODE
ATR8            BSS    2       WCC
```

5.3.5 Transmit Routine — Send Station Buffer to Host

The Transmit routine can transmit all of the modified fields in the station buffer to the host application. This routine generates an attention ID (AID) code for the transmission and returns a completion code indicating its success. The local print option is included for debugging purposes and transmits nothing to the host.

5.3.5.1 Transmit Call Statements. The following examples demonstrate the Transmit routine call statement for each supported programming language.

COBOL:

```
CALL 'IC$XMT' USING <a1> <a2> <a3>.
```

FORTRAN:

```
CALL IFSXMT (<a1>,<a2>,<a3>)
```

Pascal:

```
IFSXMT (<a1>,<a2>,<a3>);
```

990 Assembly Language:

	LI	R0,XMTARG	POINTER TO ARGUMENTS LIST
	BLWP	@IC\$XMT	SUBROUTINE LINKAGE
	.		
	.		
	.		
XMTARG	DATA	3*2	THREE TWO-BYTE POINTERS TO ARGUMENTS
	DATA	<a1>	COMM LINE & TERMINAL ID POINTER
	DATA	<a2>	AID CODE POINTER
	DATA	<a3>	COMPLETION CODE POINTER

5.3.5.2 Arguments for the Transmit Routine. The following list describes the arguments for the Transmit routine:

- < a1> is the control unit address and terminal address of the PSC program. This argument consists of two words. The first word contains the control unit address or a specification to let ICS choose the control unit address. It also contains a flag that indicates whether PSC fills the station buffer with the contents of the local format file. The second word contains the terminal address or a specification to let ICS choose the terminal address. It also specifies the type (CRT or PRT) of the terminal address. Paragraph 5.2.4 describes this argument in detail.
- < a2> is the AID code for the transmission. This argument conveys an AID to the host along with the modified fields in the station buffer. Table 5-1 lists the AID codes that correspond to the program attention keys on the display station keyboard.
- < a3> is a completion code. The routine returns a zero completion code when the operation succeeds. A nonzero code indicates an error condition. (Refer to Appendix B for a list of error messages and their meanings.)

Table 5-1. Attention ID (AID) Codes

Program Attention Key	Attention ID Code	Program Attention Key*	Attention ID Code
Enter	0	Clear	16
PF1	1	Reset	17
PF2	2	PF13	18
PF3	3	PF14	19
PF4	4	PF15	20
PF5	5	PF16	21
PF6	6	PF17	22
PF7	7	PF18	23
PF8	8	PF19	24
PF9	9	PF20	25
PF10	10	PF21	26
PF11	11	PF22	27
PF12	12	PF23	28
PA1	13	PF24	29
PA2	14	SYSREQ	Not supported
PA3	15	Print	31

Note:

* Program Function (PF) keys 13 through 24 are not supported by the IBM 3271 Model 2 Control Unit and the IBM 3277 Model 2 Display Station. Those keys are supported by the IBM 3274 Model 1 Control Unit, the IBM 3276 Model 2 Control Unit, and the IBM 3278 Model 2 Display Station.

5.3.5.3 Results of the Transmit. The Transmit routine typically sends a message across the ICS line to a remote host application. The message begins with the AID code specified in the call. Then, the message lists the contents of every modified field in the station buffer. Fields become modified when filled by the Fill routine.

The following list describes four of the AID codes that warrant special mention:

Code:	Description:
Enter	The Enter code signifies that the station has a message for the host application. Once the program has a buffer ready to transmit, it calls the Transmit routine with this AID code. Then, all of the modified data is transmitted to the host.
Clear	The Clear code replaces all data (including attribute characters) in the station buffer with nulls. Only the Clear code is transmitted to the host.
Reset	If the keyboard is locked, Reset unlocks it. ICS transmits no data.
Print	The Print code is an internal code provided to help you debug a program. No transmission is made to the host. Instead, the Print code writes the contents of the station buffer to the default print pathname (the value last assigned to the terminal address in use). Once your PSC program begins execution, you cannot change the current value of the PRINT key pathname.

The Transmit routine returns a completion code in its third argument. Zero means a successful transmission. A nonzero code indicates an error condition. (Refer to Appendix B for a list of error messages and their meanings.)

5.3.5.4 Examples of the Transmit Call. The following examples demonstrate the Transmit routine call statement in each supported programming language.

COBOL:

```
CALL 'IC$XMT' USING STATION-ADDRESS AID-KODE COMPLETION-KODE.
```

FORTRAN:

```
CALL IFSXMT (STADDR, XMTAID, CCODE)
```

Pascal:

```
IFSXMT (ADDR, AID CODE, CCODE);
```

990 Assembly Language:

```

        LI    R0,XMTARG  POINTER TO ARGUMENTS LIST
        BLWP  @IC$XMT   SUBROUTINE LINKAGE
        .
        .
        .
XMTARG  DATA  3*2      THREE TWO-BYTE POINTERS TO ARGUMENTS
        DATA  OPN1     COMM LINE & TERMINAL ID POINTER
        DATA  XMT2     AID CODE POINTER
        DATA  XMTCC    COMPLETION CODE POINTER
        .
        .
        .
OPN1    DATA  >0200    USE THE SAME AS USED BY THE OPEN
        DATA  >0200    ROUTINE; AUTOMATICALLY SELECT
                        CRT TYPE
XMT2    BSS    2        AID CODE
XMTCC   DATA  0        COMPLETION CODE
    
```

5.3.6 Time-Out Routine — Wait for Host to Respond

The Time-Out routine temporarily suspends execution of the PSC program for a specified period. This routine returns a completion code that indicates the success of the operation.

5.3.6.1 Time-Out Call Statements. The following examples demonstrate the Time-Out routine call statement for each supported programming language.

COBOL:

```
CALL 'IC$TMO' USING <a1> <a2> <a3>.
```

FORTRAN:

```
CALL IFSTMO (<a1>,<a2>,<a3>)
```

Pascal:

```
FSTMO (<a1>,<a2>,<a3>);
```

990 Assembly Language:

```

        LI    R0,TMOARG  POINTER TO ARGUMENTS LIST
        BLWP  @IC$TMO   SUBROUTINE LINKAGE
        .
        .
        .
TMOARG  DATA  3*2      THREE TWO-BYTE POINTERS TO ARGUMENTS
        DATA  <a1>     COMM LINE & TERMINAL ID POINTER
        DATA  <a2>     TIME INTERVAL POINTER
        DATA  <a3>     COMPLETION CODE POINTER
    
```

5.3.6.2 Arguments for the Time-Out Routine. The following list describes the arguments for the Time-Out routine:

- < a1> is the control unit address and terminal address of the PSC program. This argument consists of two words. The first word contains the control unit address or a specification to let ICS choose the control unit address. It also contains a flag that indicates whether PSC fills the station buffer with the contents of the local format file. The second word contains the terminal address or a specification to let ICS choose the terminal address. It also specifies the type (CRT or PRT) of the terminal address. Paragraph 5.2.4 describes this argument in detail.
- < a2> is the number of time-out intervals. This argument specifies the number of 50-millisecond intervals to elapse before the program resumes operation.
- < a3> is a completion code. The routine returns a zero completion code when the operation succeeds. Nonzero codes indicate error conditions. (Refer to Appendix B for a list of error messages and their meanings.)

5.3.6.3 Results of the Time-Out. The Time-Out routine causes the PSC program task to wait for the specified number of 50-millisecond intervals to elapse or until the PSC Emulator receives data from the host. The program resumes execution with the statement following the call. You can call this routine after a Transmit call to give the host sufficient time to respond and write new data into the station buffer. If you are unsure of how long the time-out should be, specify a value larger than you need and rely on the PSC Emulator to terminate the time-out when it receives data from the host.

The routine returns a completion code in its third argument. Zero means success. A nonzero completion code indicates an error condition.

5.3.6.4 Examples of the Time-Out Call. The following examples demonstrate the Time-Out routine call statement for each supported programming language.

COBOL:

```
CALL 'IC$TMO' USING STATION-ADDRESS TMO-HOW-LONG COMPLETION-KODE.
```

FORTRAN:

```
CALL IFSTMO (STADDR, TMOCNT, CCODE)
```

Pascal:

```
IFSTMO (ADDR, TIME_OUT_INTERVALS, CCODE);
```

990 Assembly Language:

```

        LI    R0,TMOARG  POINTER TO ARGUMENTS LIST
        BLWP @IC$TMO    SUBROUTINE LINKAGE
        .
        .
TMOARG  DATA 3*2        THREE 2-BYTE POINTERS TO ARGUMENTS
        DATA OPN1      COMM LINE & TERMINAL ID POINTER
        DATA TMO2      TIME INTERVAL POINTER
        DATA TMOCC     COMPLETION CODE POINTER
        .
        .
OPN1    DATA >200     USE THE SAME AS USED FOR THE OPEN
        DATA >200     ROUTINE; AUTOMATICALLY SELECT
                          CRT TYPE
        DATA 2         TIME INTERVAL; WAIT 100 MILLISECONDS
        BSS  2          COMPLETION CODE
    
```

5.3.7 Close Routine — Close Station

The Close routine releases a station from PSC program control. The call indicates what status to report to ICS. One argument specifies to the PSC Emulator whether the PSC Emulator is to terminate itself or remain active. If a user-written PSC program closes the PSC Emulator using the *Do not terminate* option and then terminates itself, the PSC Emulator automatically rebids your PSC program when it receives data from the host that is directed to the terminal address reserved for your PSC program. The PSC Emulator can successfully rebid your PSC program only if it resides in the .S\$PROGA program file. If you use the *Do not terminate* option but do not terminate your program, the PSC Emulator does not attempt to bid a new copy of your program.

5.3.7.1 Close Call Statements. The following examples demonstrate the Close routine call statement for each supported programming language.

COBOL:

```
CALL 'IC$CLS' USING <a1> <a2> <a3>.
```

FORTRAN:

```
CALL IFSCLS (<a1>,<a2>,<a3>)
```

Pascal:

```
IFSCLS (<a1>,<a2>,<a3>);
```

990 Assembly Language:

```

        LI      R0,CLSARG  POINTER TO ARGUMENTS LIST
        BLWP   @IC$CLS   SUBROUTINE LINKAGE
        .
        .
        .
CLSARG  DATA   3*2      THREE TWO-BYTE POINTERS TO ARGUMENTS
        DATA   <a1>    COMM LINE & TERMINAL ID POINTER
        DATA   <a2>    TERMINATE FLAG POINTER
        DATA   <a3>    COMPLETION CODE POINTER
    
```

5.3.7.2 Arguments for the Close Routine. The following list describes the arguments for the Close routine:

- < a1> is the control unit address and terminal address of the PSC program. This argument consists of two words. The first word contains the control unit address or a specification to let ICS choose the control unit address. It also contains a flag that indicates whether PSC fills the station buffer with the contents of the local format file. The second word contains the terminal address or a specification to let ICS choose the terminal address. It also specifies the type (CRT or PRT) of the terminal address. Paragraph 5.2.4 describes this argument in detail.
- < a2> indicates whether the station remains active after the call, as follows:
 - 0 = Suspend the station until data arrives from the host.
 - 2 = Terminate operations at the station.
- < a3> is a completion code. The routine returns a zero completion code when the operation succeeds. A nonzero code indicates error. (Refer to Appendix B for a list of error messages and their meanings.)

5.3.7.3 Results of the Close. The Close routine releases the station from the control of the PSC program. If the second argument equals zero, the PSC Emulator waits until a message arrives from the host or the PSC application issues an Open *Do not activate* call. If data arrives from the host, the PSC program is rebid. If the second argument equals two, the Close routine terminates the PSC Emulator and releases the ICS terminal address.

The third argument returns a completion code to the program after the routine closes the station. Zero means the operation succeeded. A nonzero code indicates an error condition listed in Appendix B.

NOTE

Anytime the user program terminates abnormally, a Close specifying *Terminate* (a value of two for < a2>) must be issued to the PSC Emulator. The Close call should always be included as part of the abnormal termination procedure of a PSC program.

5.3.7.4 Examples of the Close Call. The following examples demonstrate the Close routine call statement for each of the supported programming languages.

COBOL:

```
CALL 'IC$CLS' USING STATION-ADDRESS DUMMY COMPLETION-KODE.
```

FORTRAN:

```
CALL IFSCLS(STADDR, CLSDUM, CCODE)
```

Pascal:

```
IFSCLS(ADDR, DUMMY, CCODE);
```

990 Assembly Language:

```

          LI    R0,CLSARG  POINTER TO ARGUMENTS LIST
          BLWP  @IC$CLS   SUBROUTINE LINKAGE
          .
          .
          .
CLSARG   DATA  3*2      THREE TWO-BYTE POINTERS TO ARGUMENTS
          DATA  OPN1    COMM LINE & TERMINAL ID POINTER
          DATA  CLS2    TERMINATE FLAG POINTER
          DATA  CLSCC   COMPLETION CODE POINTER
          .
          .
          .
OPN1     DATA  >0200   USE THE SAME AS USED FOR THE OPEN
          DATA  >0200   ROUTINE; AUTOMATICALLY SELECT
                          CRT TYPE
CLS2     DATA  2       TERMINATE
CLSCC    DATA  0       COMPLETION CODE

```

5.4 WRITING A PSC PROGRAM

Writing a PSC program requires the same steps as developing any other application program under DNOS plus one additional step, which is patching. You compile or assemble the program in the usual way, link it with the PSC routines it calls, patch those routines, and execute it with an SCI command. The Execute Programmed Station Control (XPSC) command activates a user-written PSC program.

You must know the generated configuration of the host so that your PSC program uses the correct control unit and terminal addresses. For some host configurations, your PSC program must log on to the desired application. Writing a PSC program also requires that you be familiar with the following aspects of the host application:

- Screen format
- Valid responses for each field
- Use of special function keys (PF1 – PF24, PA1 – PA3)

ICS contains a useful program development aid that allows you to execute PSC call statements individually, providing parameters in a menu-driven fashion. The *Release Information, DNOS 3270 /ICS* document contains a description of how to install and execute this program.

The paragraphs in this section contain no discussion of the process of compiling or assembling the program since the individual language manuals provide all the needed information.

To create a source program, you insert calls to the following PSC routines in place of routines that perform I/O to host:

Routine:	Description:
Fill	Sends output to the station buffer
Get	Retrieves information from the station buffer
Transmit	Sends information to the host

PSC returns a completion code for each call to a PSC routine. Zero indicates a successful operation. Nonzero codes indicate error conditions. (Refer to Appendix B for a list of error messages and their meanings.)

If your PSC program issues a Close call specifying *Do not terminate*, the PSC Emulator rebids your program when data arrives from the host. When the PSC Emulator rebids your program, it passes two parameters that your program can access using the Retrieve Task Parameters call appropriate to the language that you are using. Each parameter is one word (16 bits). The first parameter contains the decimal control unit address that the PSC Emulator has reserved for your program in the first byte (bits 0 through 7). The second byte (bits 8 through 15) of the first parameter contains the decimal terminal address that the PSC Emulator has reserved for your program. The second parameter is always zero. Your program must use these passed addresses for the first argument in the Open *Do not activate* in order for your program to regain access to the PSC Emulator. Alternatively, your program can specify Auto-Select and let ICS find the first available station in the system that is in a PSC wait state.

A PSC program that infrequently processes host data, such as a printer, can use the Close call specifying *Do not terminate* and subsequently, the Open call specifying *Do not activate*. When your PSC program issues the Close call specifying *Do not terminate* and terminates itself, idle memory and system resources belonging to the PSC program are returned to the operating system and made available for other uses.

NOTE

The PSC runtime linked with your application makes use of the DNOS event task synchronization feature. In particular, event number one is used to synchronize events between the PSC runtime and the PSC Emulator. Avoid using event number one if you plan to use events in your PSC application program.

5.4.1 COBOL Call Statements

The following lists suggest data structures that your COBOL program can use for each of the arguments of each PSC routine. It also shows the COBOL call statement for each PSC routine.

Open Call:

CALL "IC\$OPN" USING < a1> < a2> < a3> < a4> .

Argument	Data Type	Picture	Values	Meaning of Values
< a1>	GROUP			First argument 2 words
	COMP-1	9(2)	> 0 - > 600	Control unit address
	COMP-1	9(2)	> 0 - > 300	Terminal address
< a2>	COMP-1	9(2)	0 or 2	Activate interface
< a3>	COMP-1	9(2)	0	Dummy argument
< a4>	COMP-1	9(4)	> 0 - > 00FF	Completion code

Fill Call:

CALL "IC\$FIL" USING < a1> < a2> < a3> < a4> < a5> < a6> < a7> < a8> .

Argument	Data Type	Picture	Values	Meaning of Values
< a1>	GROUP			First argument 2 words
	COMP-1	9(2)	> 0 - > 600	Control unit address
	COMP-1	9(2)	> 0 - > 300	Terminal address
< a2>	DISPLAY	X(< n>)	< n> chars	Data to fill station buffer
< a3>	COMP-1	9(4)	> 1 - > 240	Number of characters in fill
< a4>	COMP-1	9(4)	1 - 1919	Number of field to fill
< a5>	COMP-1	9(4)	1 - 1920	Position in field to start fill
< a6>	COMP-1	9(4)	0,1,or 3	Next action: none, transmit, clear
< a7>	COMP-1	9(4)	> 0 - > 00FF	Completion code
< a8>	GROUP			Attributes of field
	COMP-1	9(1)	192 - 255	Real attribute byte
	COMP-1	9(3)		Slack bytes

Get Call:

CALL "IC\$GET" USING < a1> < a2> < a3> < a4> < a5> < a6> < a7> < a8> .

Argument	Data Type	Picture	Values	Meaning of Values
< a1>	GROUP			First argument 2 words
	COMP-1	9(2)	> 0 - > 600	Control unit address
	COMP-1	9(2)	> 0 - > 300	Terminal address
< a2>	DISPLAY	X(< n>)	< n> chars	Data obtained from station buffer
< a3>	COMP-1	9(4)	1 - 240	Number of characters to get
< a4>	COMP-1	9(4)	1 - 1920	Number of field to get
< a5>	COMP-1	9(4)	1 - 1920	Position in field to start
< a6>	COMP-1	9(4)	0,1,or 3	Next action: none, transmit, clear
< a7>	COMP-1	9(4)	> 0 - > 00FF	Completion open
< a8>	GROUP			Attributes of field
	COMP-1	9(1)	192 - 255	Real Attribute byte
	COMP-1	9(3)	0	Slack bytes

Attribute Call:

CALL "IC\$ATR" USING <a1> <a2> <a3> <a4> <a5> <a6> <a7> <a8>.

Argument	Data Type	Picture	Values	Meaning of Values
<a1>	GROUP			First argument 2 words
	COMP-1	9(2)	>0->600	Control unit address
<a2>	COMP-1	9(2)	>0->300	Terminal address
	Occurs 40 times			Attribute information
	COMP-1	9(4)	0-1919	Position
	DISPLAY	X	64-255	Attribute
<a3>	DISPLAY	X(3)		Not used
	COMP-1	9(2)	1-40	Max number of attribute returned
<a4>	COMP-1	9(4)	1-1920	Number of first attribute to get
<a5>	COMP-1	9(4)	0-1919	Current cursor location
<a6>	COMP-1	9(2)	1-40	Number of attributes returned
<a7>	COMP-1	9(4)	>1->00FF	Completion code
<a8>	COMP-1	9(4)	>0->00FF	WCC of current screen image

Time-Out Call:

CALL "IC\$TMO" USING <a1>, <a2>, <a3>.

Argument	Data Type	Picture	Values	Meaning of Values
<a1>	GROUP			First argument 2 words
	COMP-1	9(2)	>0->600	Control unit address
<a2>	COMP-1	9(2)	>0->300	Terminal address
	COMP-1	9(4)	>0->7FFF	Time-out in 50-msec intervals
<a3>	COMP-1	9(4)	>0->00FF	Completion code

Transmit Call:

CALL "IC\$XMT" USING <a1>, <a2>, <a3>.

Argument	Data Type	Picture	Values	Meaning of Values
<a1>	GROUP			First argument 2 words
	COMP-1	9(2)	>0->600	Control unit address
	COMP-1	9(2)	>0->300	Terminal address
<a3>	COMP-1	9(2)	0-31	AID code (see Table 5-1)
	COMP-1	9(4)	>0->00FF	Completion code

Close Call:

CALL "IC\$CLS" USING < a1> , < a2> , < a3> .

Argument	Data Type	Picture	Values	Meaning of Values
< a1>	GROUP			First argument 2 words
	COMP-1	9(2)	> 0 - > 600	Control unit address
	COMP-1	9(2)	> 0 - > 300	Terminal address
< a2>	COMP-1	9	0 or 2	Reactivate: yes or no
< a3>	COMP-1	9(4)	> 0 - > 00FF	Completion code

5.4.2 FORTRAN Call Statements

The following list suggests data structures that your FORTRAN program can use for each argument of each PSC routine. This list also shows the FORTRAN call statement for each PSC routine.

Open Call:

IFSOPN (< a1> , < a2> , < a3> , < a4>)

Argument	Data Type	Size	Values	Meaning of Values
< a1>	Integer(2)	Two words	> 0 - > 600	Communications line and terminal address
			> 0 - > 300	
< a2>	Integer	Word	0 or 2	Activate interface
< a3>	Integer	Word	0	Dummy argument
< a4>	Integer	Word	> 0 - > 00FF	Completion code

Fill Call:

IFSFIL (< a1> , < a2> , < a3> , < a4> , < a5> , < a6> , < a7> , < a8>)

Argument	Data Type	Size	Values	Meaning of Values
< a1>	Integer(2)	Two words	> 0 - > 600	Communications line and terminal address
			> 0 - > 300	
< a2>	Integer(120)	N/2 words	Any ASCII	Data to fill station buffer
< a3>	Integer	Word	0 - 240	Number of characters in fill
< a4>	Integer	Word	1 - 1920	Number of field to fill
< a5>	Integer	Word	1 - 1920	Position in field to start fill
< a6>	Integer	Word	0,1,or 3	Next action: none, transmit, clear
< a7>	Integer	Word	> 0 - > 00FF	Completion code
< a8>	Integer(2)	Two words	192 - 255	Attribute byte + 3 slack bytes

Get Call:

IFSGET (< a1> ,< a2> ,< a3> ,< a4> ,< a5> ,< a6> ,< a7> ,< a8>)

Argument	Data Type	Size	Values	Meaning of Values
< a1>	Integer(2)	Two words	> 0 – > 600 > 0 – > 300	Communications line and terminal address
< a2>	Integer(120)	120 words	Any ASCII	Data obtained from station buffer
< a3>	Integer	Word	1 – 240	Number of characters to get
< a4>	Integer	Word	1 – 1919	Number of field to get
< a5>	Integer	Word	1 – 1920	Position in field to start
< a6>	Integer	Word	0,1,or 3	Next action: none, transmit, clear
< a7>	Integer	Word	> 0 – > 00FF	Completion code
< a8>	Integer(2)	Two words	192 – 255	Attribute byte + 3 slack bytes

Attribute Call:

IFSATR (< a1> ,< a2> ,< a3> ,< a4> ,< a5> ,< a6> ,< a7> ,< a8>)

Argument	Data Type	Size	Values	Meaning of Values
< a1>	Integer(2)	Two words	> 0 – > 600 > 0 – > 300	Communications line and terminal address
< a2>	Integer(120)	240 bytes	192 – 255	Attributes of up to 40 fields
< a3>	Integer	Word	0 – 40	Number of attributes
< a4>	Integer	Word	1 – 1920	Number of first field
< a5>	Integer	Word	0 – 1919	Current location of cursor
< a6>	Integer	Word	0 – 40	Number of attributes returned
< a7>	Integer	Word	> 0 – > 00FF	Completion code
< a8>	Integer	Word	> 0 – > 00FF	WCC of current screen image

Time-Out Call:

IFSTMO (< a1> , < a2> , < a3>)

Argument	Data Type	Size	Values	Meaning of Values
< a1>	Integer(2)	Two words	> 0 – > 600 > 0 – > 300	Communications line and terminal address
< a2>	Integer	Word	> 1 – > 7FFF	Time-out in 50-msec intervals
< a3>	Integer	Word	> 0 – > 00FF	Completion code

Transmit Call:

IFSXMT (< a1> , < a2> , < a3>)

Argument	Data Type	Size	Values	Meaning of Values
< a1>	Integer(2)	Two words	> 0 – > 600 > 0 – > 300	Communications line and terminal address
< a2>	Integer	Word	0 – 31	AID code (see Table 5-1)
< a3>	Integer	Word	> 0 – > 00FF	Completion code

Close Call:

IFSCLS (< a1> , < a2> , < a3>)

Argument	Data Type	Size	Values	Meaning of Values
< a1>	Integer(2)	Two words	> 0 – > 600 > 0 – > 300	Communications line and terminal address
< a2>	Integer	Word	0 or 2	Reactivate: yes, no
< a3>	Integer	Word	> 0 – > 00FF	Completion code

5.4.3 Pascal Call Statements

Pascal programs normally allocate their stack and heap sizes dynamically. If you design your Pascal PSC program to be rebid by the PSC Emulator, you must not use this feature. The PSC Emulator cannot pass the necessary parameters to dynamically allocate the stack and heap. If you do not design your Pascal PSC program to be rebid by the PSC Emulator, you can use the dynamic allocation of stack and heap sizes. You must, however, activate your program by using the Execute Pascal Task (XPT) command.

The PSC Emulator can successfully rebid your Pascal PSC program only if your program allocates the stack and heap size statically and you activate it by using XPSC. The *DNOS TI Pascal Programmer's Manual* describes how to statically allocate the stack and heap.

When you declare the PSC routines that your program calls, you must define them as external FORTRAN procedures.

The following list suggests data structures that your Pascal program can use for each argument of each PSC routine. It also shows the Pascal call statement for each PSC routine.

Open Call:

IFSOPN (< a1> , < a2> , < a3> , < a4>);

Argument	Data Type	Size	Values	Meaning of Values
< a1>	Array[0..1] of Integer	Two words	> 0 – > 600 > 0 – > 300	Communications line and terminal address
< a2>	Integer	Word	0 or 2	Activate interface
< a3>	Integer	Word	0	Dummy argument
< a4>	Integer	Word	> 0 – > 00FF	Completion code

Fill Call:

IFSFIL (<a1>,<a2>,<a3>,<a4>,<a5>,<a6>,<a7>,<a8>);

Argument	Data Type	Size	Values	Meaning of Values
<a1>	Array[0..1] of integer	Two words	>0->600 >0->300	Communications line and terminal address
<a2>	Array[1..N] of characters	N/2 words	Any ASCII	Data to fill station buffer
<a3>	Integer	Word	1-240	Number of characters in fill
<a4>	Integer	Word	1-1920	Number of field to fill
<a5>	Integer	Word	1-1920	Position in field to start fill
<a6>	Integer	Word	0,1,or 3	Next action: none, transmit, clear
<a7>	Integer	Word	>0->00FF	Completion code
<a8>	Packed Array[1..4] of byte	Two words	192-255	Attribute byte + 3 slack bytes

Get Call:

IFSGET (<a1>,<a2>,<a3>,<a4>,<a5>,<a6>,<a7>,<a8>);

Argument	Data Type	Size	Values	Meaning of Values
<a1>	Array[0..1] of integer	Two words	>0->600 >0->300	Communications line and terminal address
<a2>	Array[0..N] of characters	N/2 words	Any ASCII	Data obtained from station buffer
<a3>	Integer	Word	1-240	Number of characters to get
<a4>	Integer	Word	1-1920	Number of field to get
<a5>	Integer	Word	1-1920	Position in field to start
<a6>	Integer	Word	0,1,or 3	Next action: none, transmit, clear
<a7>	Integer	Word	>0->00FF	Completion code
<a8>	Packed Array [1..4] of byte	Two words	192-255	Attribute byte + 3 slack bytes

Attribute Call:

IFSATR (<a1>,<a2>,<a3>,<a4>,<a5>,<a6>,<a7>,<a8>);

Argument	Data Type	Size	Values	Meaning of Values
<a1>	Array[0..1] of integer	Two words	>0->600 >0->300	Communications line and terminal address
<a2>	Packed Array[1..n] of Attr	3* <n> Bytes	192-255	Attributes of up to 40 fields
<a3>	Integer	Word	0-40	Number of attributes
<a4>	Integer	Word	1-1920	Number of first field
<a5>	Integer	Word	0-1919	Current cursor location
<a6>	Integer	Word	0-40	Number of attributes returned
<a7>	Integer	Word	>0->00FF	Completion code
<a8>	Integer	Word	>0->00FF	WCC of current screen image

Time-Out Call:

IFSTMO (<a1>,<a2>,<a3>);

Argument	Data Type	Size	Values	Meaning of Values
<a1>	Array[0..1] of integer	Two words	>0->600 >0->300	Communications line and terminal address
<a2>	Integer	Word	>0->7FFF	Time-out in 50-msec intervals
<a3>	Integer	Word	>0->00FF	Completion code

Transmit Call:

IFSXMT (<a1>,<a2>,<a3>);

Argument	Data Type	Size	Values	Meaning of Values
<a1>	Array[0..1] of integer	Two words	>0->600 >0->300	Communications line and terminal address
<a2>	Integer		0-31	AID code (see Table 5-1)
<a3>	Integer		>0->00FF	Completion code

Close Call:

IFSCLS (<a1>,<a2>,<a3>);

Argument	Data Type	Size	Values	Meaning of Values
<a1>	Integer(2)	Two words	>0->600 >0->300	Communications line and terminal address
<a2>	Integer	Word	0 or 2	Reactivate: yes, no
<a3>	Integer	Word	>0->00FF	Completion code

5.4.4 990 Assembly Language Call Statements

The following list suggests data structures that your 990 assembly language program can use for each argument of each PSC routine. The list also shows the 990 assembly language call statement for each PSC routine.

Open Call:

```

                LI      R0,OPNARG
                BLWP   @IC$OPN
OPNARG        DATA  4*2
                DATA  <a1>
                DATA  <a2>
                DATA  <a3>
                DATA  <a4>
    
```

Argument	Value Type	Size	Values	Meaning of Values
< a1 >	Hex integer	Two words	> 0 – > 600 > 0 – > 300	Control unit and terminal address
< a2 >	Hex integer	Word	0 or 2	Activate interface: yes or no
< a3 >	Dec integer	Word	> 0	Dummy argument
< a4 >	Hex integer	Word	> 0 – > 00FF	Completion code

Fill Call:

```

                LI      R0,FILARG
                BLWP   @IC$FIL
FILARG        DATA  8*2
                DATA  <a1>
                DATA  <a2>
                DATA  <a3>
                DATA  <a4>
                DATA  <a5>
                DATA  <a6>
                DATA  <a7>
                DATA  <a8>
    
```

Argument	Value Type	Size	Values	Meaning of Values
< a1 >	Hex integer	2 words	> 0 – > 600 > 0 – > 300	Control unit and terminal address
< a2 >	Character	N words	N chars	Data to fill station buffer
< a3 >	Dec integer	Word	1 – 240	Number of characters in fill
< a4 >	Dec integer	Word	1 – 1919	Number of first field to fill
< a5 >	Dec integer	Word	1 – 1920	Position in field to start fill
< a6 >	Dec integer	Word	0,1,or 3	Next action: none, transmit, clear
< a7 >	Hex integer	Word	> 0 – > 00FF	Completion code
< a8 >	Hex integer	Two words	> 192 – > 255	Attribute in high-order byte

Get Call:

```

GETARG      LI      R0,GETARG
            BLWP    @IC$GET
            DATA   8*2
            DATA   < a1>
            DATA   < a2>
            DATA   < a3>
            DATA   < a4>
            DATA   < a5>
            DATA   < a6>
            DATA   < a7>
            DATA   < a8>
    
```

Argument	Value Type	Size	Values	Meaning of Values
< a1>	Hex integer	Two words	> 0 – > 600 > 0 – > 600	Control unit and terminal address
< a2>	Character	N words	N chars	Data obtained from station buffer
< a3>	Dec integer	Word	1-240	Number of characters to get
< a4>	Dec integer	Word	1-1919	Number of field to get
< a5>	Dec integer	Word	1-1920	Position in field to start
< a6>	Dec integer	Word	0,1,or 3	Next action: none, transmit, clear
< a7>	Hex integer	Word	> 0 – > 00FF	Completion code
< a8>	Hex integer	Two words	> 192 – > 255	Attribute in high-order byte

Attribute Call:

```

ATRARG      LI      R0,ATRARG
            BLWP    $IC$ATR
            DATA   8*2
            DATA   < a1>
            DATA   < a2>
            DATA   < a3>
            DATA   < a4>
            DATA   < a5>
            DATA   < a6>
            DATA   < a7>
            DATA   < a8>
    
```

Argument	Value Type	Size	Values	Meaning of Values
< a1>	Hex integer	Two words	> 0 – > 600 > 0 – > 300	Control unit and terminal address
< a2>	Hex integer	Three words	Per attr	Up to 40 attributes
< a3>	Dec integer	Word	0-40	Max attributes to get
< a4>	Dec integer	Word	1-1920	Number of first field
< a5>	Dec integer	Word	0-1919	Current cursor location
< a6>	Dec integer	Word	0-40	Number of attributes
< a7>	Hex integer	Word	> 0 – > 00FF	Completion code
< a8>	Hex integer	Word	> 0 – > 00FF	WCC of current screen location

Time-Out Call:

TMOARG	LI	R0,TMOARG
	BLWP	@IC\$TMO
	DATA	3*2
	DATA	< a1>
	DATA	< a2>
	DATA	< a3>

Argument	Value Type	Size	Values	Meaning of Values
< a1>	Hex integer	2 words	> 0 – > 600	Control unit and terminal address
< a2>	Hex integer	Word	> 0 – > 7FFF	Time-out in 50-msec intervals
< a3>	Hex integer	Word	> 0 – > 00FF	Completion code

Transmit Call:

XMTARG	LI	R0,XMTARG
	BLWP	@IC\$XMT
	DATA	3*2
	DATA	< a1>
	DATA	< a2>
	DATA	< a3>

Argument	Value Type	Size	Values	Meaning of Values
< a1>	Hex integer	Two words	> 0 – > 600	Control unit and terminal address
< a2>	Dec integer	Word	0-31	AID code (see Table 5-1)
< a3>	Hex integer	Word	> 0 – > 00FF	Completion code

Close Call:

```

          LI      R0,CLSARG
          BLWP   @IC$CLS
CLSARG   DATA  3*2
          DATA  <a1>
          DATA  <a2>
          DATA  <a3>
    
```

Argument	Value Type	Size	Values	Meaning of Values
<a1>	Hex integer	Two words	>0 - >600	Control unit terminal address
<a2>	Hex integer	Word	>0 or >2	Reactivate: yes or no
<a3>	Hex integer	Word	>0 - >00FF	Completion code

5.4.5 Calling Sequence Examples

Figures 5-7 through 5-10 illustrate a proper calling sequence for each PSC routine in each supported programming language.

```

IDENTIFICATION DIVISION.
PROGRAM-ID. PSC.
*
* THIS EXAMPLE ILLUSTRATES PSC CALLS AND THE ASSOCIATED
* DATA STRUCTURES FOR COBOL. HOWEVER, THE SEQUENCE OF THE CALLS
* DOES NOT REFLECT HOW A TYPICAL PSC PROGRAM SHOULD BE STRUCTURED.
*

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SOURCE-COMPUTER. TI-990.
OBJECT-COMPUTER. TI-990.

DATA DIVISION.
WORKING-STORAGE SECTION.
01 STATION-ADDRESS.
   03 ICS-LINE      PIC 9(2)  COMP-1.
   03 TERM-ADDRESS PIC 9(2)  COMP-1.

01 OPN-ARGUMENTS.
   03 ACTIVATE     PIC 9      COMP-1.
   03 DUMMY        PIC 9      COMP-1.

01 FIL-ARGUMENTS.
   03 FILL-BUFFER  PIC X(64) DISPLAY.
   03 FILL-HOW-MANY PIC 9(4)  COMP-1.
   03 FILL-FIELD   PIC 9(4)  COMP-1.
   03 FILL-BEGIN   PIC 9(4)  COMP-1.
   03 FILL-ACTION  PIC 9(4)  COMP-1.
03 FILL-ATTRIBUTE.
   05 ATTR-BYTE    PIC X      DISPLAY.
   05 FILLER       PIC X(3)  DISPLAY.
    
```

Figure 5-7. Example COBOL Calls to PSC Routines (Sheet 1 of 3)

```

01 GET-ARGUMENTS.
03 GET-BUFFER      PIC X(64) DISPLAY.
03 GET-HOW-MANY    PIC 9(4)  COMP-1.
03 GET-FIELD       PIC 9(4)  COMP-1.
03 GET-BEGIN       PIC 9(4)  COMP-1.
03 GET-ACTION      PIC 9(4)  COMP-1.
03 GET-ATTRIBUTE.
    05 ATTR-BYTE    PIC X      DISPLAY.
    05 FILLER       PIC X(3)  DISPLAY.

01 ATR-BUFFER      PIC X(240).
01 ATR-BUF REDEFINES ATR-BUFFER.
03 ATR-BU OCCURS 40 TIMES
    INDEXED BY ATR-INDX.
    05 ATR-POS      PIC 9(4)  COMP-1.
    05 ATTR-BYTE    PIC X      DISPLAY.
    05 FILLER       PIC X(3)  DISPLAY.
03 ATR-MAXNO       PIC 9(2)  COMP-1.
03 ATR-BEGIN       PIC 9(4)  COMP-1.
03 ATR-CURSOR      PIC 9(4)  COMP-1.
03 ATR-RETURNED    PIC 9(2)  COMP-1.
03 WCC             PIC 9(4)  COMP-1.

01 TMO-ARGUMENTS.
03 TMO-HOW-LONG    PIC 9(4)  COMP-1.

01 XMT-ARGUMENTS.
03 AID-KODE        PIC 9(2)  COMP-1.

01 CLS-ARGUMENTS.
03 REACTIVATE      PIC 9      COMP-1.

01 COMPLETION-KODE PIC 9(4)  COMP-1.

PROCEDURE DIVISION.
OPN-CALL.
    MOVE 0 TO ICS-LINE.
    MOVE 05 TO TERM-ADDRESS.
    MOVE ZERO TO ACTIVATE.
    MOVE ZERO TO COMPLETION-KODE.
    CALL 'ICSOPN' USING STATION-ADDRESS ACTIVATE DUMMY
        COMPLETION-KODE.
    IF COMPLETION-KODE NOT = ZERO, GO TO ERROR-ROUTINE.
FIL-CALL.
    MOVE "INFORMATION" TO FILL-BUFFER.
    MOVE 11 TO FILL-HOW-MANY.
    MOVE 1 TO FILL-FIELD.
    MOVE 1 TO FILL-BEGIN.
    MOVE 1 TO FILL-ACTION.
    CALL 'IC$FIL' USING STATION-ADDRESS FILL-BUFFER
        FILL-HOW-MANY FILL-FIELD FILL-BEGIN
        FILL-ACTION COMPLETION-KODE FILL-ATTRIBUTE.
    IF COMPLETION-KODE NOT = ZERO, GO TO ERROR-ROUTINE.
GET-CALL.
    MOVE 0 TO GET-BUFFER.
    MOVE 64 TO GET-HOW-MANY.
    MOVE 1 TO GET-FIELD.
    MOVE 1 TO GET-BEGIN.
    MOVE 3 TO GET-ACTION.
    MOVE 0 TO GET-ATTRIBUTE.
    CALL 'IC$GET' USING STATION-ADDRESS GET-BUFFER
        GET-HOW-MANY GET-BEGIN GET-ACTION
        COMPLETION-KODE GET-ATTRIBUTE.
    IF COMPLETION-KODE NOT = ZERO, GO TO ERROR-ROUTINE.

```

Figure 5-7. Example COBOL Calls to PSC Routines (Sheet 2 of 3)

```

ATR-CALL.
  MOVE 24 TO ATR-MAXNO.
  MOVE 5 TO ATR-BEGIN.
  CALL 'IC$ATR' USING STATION-ADDRESS ATR-BUFFER ATR-MAXNO
    ATR-BEGIN ATR-CURSOR ATR-RETURNED COMPLETION-KODE WCC.
  IF COMPLETION-KODE NOT = ZERO, GO TO ERROR-ROUTINE.
TMO-CALL.
  MOVE 2 TO TMO-HOW-LONG.
  CALL "IC$TMO" USING STATION-ADDRESS TMO-HOW-LONG
    COMPLETION-KODE.
  IF COMPLETION-KODE NOT = ZERO, GO TO ERROR-ROUTINE.
XMT-CALL.
  MOVE 0 TO AID-KODE.
  CALL 'IC$XMT' USING STATION-ADDRESS AID-KODE COMPLETION-KODE.
  IF COMPLETION-KODE NOT = ZERO, GO TO ERROR-ROUTINE.
CLS-CALL.
  MOVE 0 TO REACTIVATE.
  CALL 'IC$CLS' USING STATION-ADDRESS REACTIVATE
    COMPLETION-KODE.
  IF COMPLETION-KODE NOT = ZERO, GO TO ERROR-ROUTINE.

ERROR-ROUTINE.
  MOVE 2 TO REACTIVATE.
  CALL 'IC$CLS' USING STATION-ADDRESS REACTIVATE
    COMPLETION-KODE.
END-PROGRAM.

```

Figure 5-7. Example COBOL Calls to PSC Routines (Sheet 3 of 3)

```

C THIS EXAMPLE IS WRITTEN FOR THE FORTRAN 78 COMPILER AND
C ILLUSTRATES PSC CALLS AND THE ASSOCIATED DATA
C STRUCTURES. THE SEQUENCE OF THE CALLS IS NOT
C REPRESENTATIVE OF THE STRUCTURE OF A TYPICAL
C PSC PROGRAM.

```

```

      INTEGER CCODE
C COMPLETION CODE FOR ALL ARGUMENTS
      INTEGER ADDR(2)
C CONTROL UNIT ADDRESS AND TERMINAL ADDRESS, <A1> FOR EACH CALL
      CHARACTER * 240 BUFFER
C ATTRIBUTE BUFFER - <A2> OF FILL AND GET ROUTINES
      INTEGER ATR CHAR(2)
C ATTRIBUTE CHARACTER FOR FILL, GET AND RETRIEVE ATTRIBUTES ROUTINE
      INTEGER DUMMY
C DUMMY -- <A3> OF OPEN, NOT USED
      INTEGER FILL CNT, FILL FIELD, FILL START
      INTEGER FILL NEXT
C FILL COUNT, FIELD, START FIELD, AND NEXT ACTION FOR FILL ROUTINE
      INTEGER GET CNT, GET FIELD, GET START, GET NEXT

```

Figure 5-8. Example FORTRAN Calls to PSC Routines (Sheet 1 of 3)

```

C GET COUNT, FILED, START FIELD, AND NEXT ACTION FOR GET ROUTINE
      INTEGER ATR COUNT,  ATR START,  ATR CURSOR,  ATR TOT,  WCC

C NUMBER OF CHARACTERS REQUESTED, START FIELD, CURSOR LOCATION,
C NUMBER OF CHARACTERS ACTUALLY RETRIEVED FOR RETRIEVE ATTRIBUTES
C ROUTINE, AND WCC.
      INTEGER TIME OUT INTERVALS

C TIME OUT INTERVALS FOR WAIT ROUTINE
      INTEGER TRANSMIT AID CODE

C AID CODE FOR TRANSMIT ROUTINE
      INTEGER CLS OP

C TERMINATE/DO NOT TERMINATE ARGUMENT FOR CLOSE ROUTINE
      EXTERNAL IFSOPN
      EXTERNAL IFSFIL
      EXTERNAL IFSGET
      EXTERNAL IFSATR
      EXTERNAL IFSTMO
      EXTERNAL IFSXMT
      EXTERNAL IFSCLS

C DECLARE PSC ROUTINES AS EXTERNAL SUBPROGRAMS

C OPEN ROUTINE CALL
C -- AUTOSELECT CONTROL UNIT AND CRT TERMINAL ADDRESS,
C DO NOT ACTIVATE.
      ADDR(1)=>200
      ADDR(2)=>200
      ACT=0
      DUMMY=0
      CCODE=0
      CALL IFSOPN(ADDR, ACT, DUMMY, CCODE)
      IF (CCODE.NE.0) GO TO 999

C FILL ROUTINE CALL
      BUFFER='Log-on sequence here.'
      FILL CNT=20
      FILL FIELD=2
      FILL START=1
      FILL NEXT=1
      CALL IFSFIL(ADDR,BUFFER,FILL CNT,FILL FIELD,FILL START,FILL NEXT,
X CCODE, ATR CHAR)
      IF (CCODE.NE.0) GO TO 999

C GET ROUTINE CALL
      GET CNT=80
      GETFIELD=4
      GET START=1
      GET NEXT=0
      CALL IFSGET(ADDR,BUFFER,GET CNT,GET FIELD,GET START,
X GET NEXT,GET ATTR,CCODE)
      IF (CCODE.NE.0) GO TO 999

C ATTRIBUTE ROUTINE
      ATR COUNT=40
      ATR START=10
      CALL IFSATR(ADDR,ATR BUF,ATR COUNT,ATR START,
X ATR CURSOR,ATR TOT,CCODE,WCC)
      IF (CCODE.NE.0) GO TO 999

```

Figure 5-8. Example FORTRAN Calls to PSC Routines (Sheet 2 of 3)

```

C TIME-OUT ROUTINE
    TIME OUT INTERVALS=2
    CALL IFSTMO(ADDR, TIME OUT INTERVALS, CCODE)
    IF (CCODE.NE.0) GO TO 999

C TRANSMIT ROUTINE
    TRANSMIT AID CODE=0
    CALL IFSXMT(ADDR, TRANSMIT AID CODE, CCODE)
    IF (CCODE.NE.0) GO TO 999

C CLOSE ROUTINE WITH TERMINATE OPTION
999  CLOSE OP=2
    CALL IFSCLS(ADDR, CLOSE OP, CCODE)

    END
    
```

Figure 5-8. Example FORTRAN Calls to PSC Routines (Sheet 3 of 3)

```

PROGRAM PSC EXAMPLE;

(* THIS EXAMPLE ILLUSTRATES PSC CALLS AND ASSOCIATED DATA STRUCTURES.
   FOR PASCAL. THE SEQUENCE OF THE CALLS DOES NOT REFLECT THE
   TYPICAL STRUCTURE OF A PSC PROGRAM. *)

TYPE
    ID          = ARRAY[0..1] OF INTEGER;
    BUFFER      = ARRAY[1..80] OF CHAR;
    ATTR_TYPE   = PACKED ARRAY[1..8] OF BOOLEAN;
    ATTR_RECORD = PACKED RECORD ADDRESS: INTEGER;
                   ATR_BYTE: ATTR_TYPE;
                   DUMY1: ATTR_TYPE;
                   DUMY2: INTEGER;
    END;
    ATTR_BUFFER = ARRAY [1..40] OF ATTR_RECORD;

VAR
    CC : PACKED ARRAY[1..20] OF CHAR;          (*BUFFER MESSAGE*)
    CCODE : INTEGER;                          (*COMPLETION CODE*)
    ADDR : ID;                                (*STATION ADDRESS*)
    B : BUFFER;                               (*FILL/GET DATA*)
    ACT, DUMMY : INTEGER;                    (*OPN PARMS*)
    FILL_CNT, FILL_FIELD, _FILL_START,        (*FIL PARMS*)
    FILL_NEXT : INTEGER; FILL_ATTR : ATTR_RECORD;
    GET_CNT, GET_FIELD, _GET_START,          (*GET PARMS*)
    GET_NEXT : INTEGER; GET_ATTR : ATTR_RECORD;
    ATR_BUF : ATTR_BUFFER; ATR_COUNT,        (*ATR PARMS*)
    ATR_START, ATR_CURSOR, ATR_TOT, WCC : INTEGER;
    TIME_OUT_INTERVALS : INTEGER;           (*TMO PARM*)
    TRANSMIT_AID_CODE : INTEGER;           (*XMT PARM*)
    CLS_OP : INTEGER;                       (*CLS PARM*)

PROCEDURE IFSOPN(VAR A1:ID; VAR A2, A3, A4: Integer); EXTERNAL FORTRAN;
PROCEDURE IFSFIL(VAR A1:ID; VAR A2:BUFFER; VAR A3,A4,A5,A6,A7: INTEGER;
    VAR A8:ATTR_RECORD); EXTERNAL FORTRAN;
PROCEDURE IFSGET(VAR A1:ID; VAR A2:BUFFER; VAR A3,A4,A5,A6,A7: INTEGER;
    VAR A8:ATTR_RECORD); EXTERNAL FORTRAN;
PROCEDURE IFSATR(VAR A1:ID; VAR A2:ATTR_BUFFER; VAR A3,A4,A5,A6,A7 ,A8:
    INTEGER); EXTERNAL FORTRAN;
PROCEDURE IFSTMO(VAR A1:ID; VAR A2,A3: INTEGER); EXTERNAL FORTRAN;
PROCEDURE IFSXMT(VAR A1:ID; VAR A2,A3: INTEGER); EXTERNAL FORTRAN;
PROCEDURE IFSCLS(VAR A1:ID; VAR A2,A3: INTEGER); EXTERNAL FORTRAN;
    
```

Figure 5-9. Example Pascal Calls to PSC Routines (Sheet 1 of 2)


```

PROCEDURE RPTERR(VAR A1:ID; VAR A2,A3: INTEGER);
  BEGIN
    (* REPORT ERROR CONDITION TO SOME RETRIEVABLE STRUCTURE*)
    CLS_OP := 2; (*ALWAYS CLOSE PSC EMULATOR WITH TERMINATE OPTION*)
    IFSCLS(ADDR, CLS_OP, CCODE)
  END;

BEGIN

  OPEN: BEGIN
    ADDR[0]:=#0200; ADDR[1]:=#0200; (*STATION ADDRESS: AUTOSELECT
    TYPE CRT*)
    ACT:=0; (*SET ACTIVATION VARIABLE TO ZERO*)
    DUMMY:=0; (*DUMMY ARGUMENT*)
    CCODE:=0; (*SET COMPLETION CODE TO ZERO*)
    IFSOPN(ADDR,ACT,DUMMY,CCODE); (*CALL OPEN ROUTINE*)
    IF CCODE <> 0 THEN RPTERR(ADDR,CLS_OP ,CCODE)
    (*REPORT ERROR PROCEDURE*)
  END;

  FILL: BEGIN
    B::CC:='Please Enter Your ID';(*MESSAGE TO PUT IN STATION BUFFER*)
    FILL_CNT:=20; FILL_FIELD:=2; (*TWENTY CHARACTERS TO FIELD TWO*)
    FILL_START:=1; FILL_NEXT:=1; (*START AT POSITION ONE, TRANSMIT*)
    IFSFIL(ADDR,B, FILL_CNT,FILL_FIELD,FILL_START,FILL_NEXT ,CCODE,FILL_ATTR);
    IF CCODE <> 0 THEN RPTERR(ADDR, CLS_OP,CCODE)
    (*REPORT ERROR PROCEDURE*)
  END;

  GET: BEGIN
    GET_CNT:=80; GET_FIELD:=4; (*EIGHTY CHARACTERS FROM FIELD FOUR*)
    GET_START:=1; GET_NEXT:=5; (*START AT POSITION ONE, NO ACTION*)
    IFSGET(ADDR,B, GET_CNT,GET_FIELD,GET_START,GET_NEXT, CCODE,GET_ATTR);
    IF CCODE <> 0 THEN RPTERR(ADDR, CLS_OP,CCODE)
    (*REPORT ERROR CONDITION*)
  END;

  ATTRIBUTE: BEGIN
    ATR_COUNT:=40; ATR_START:=10; (*FORTY ATTRIBUTES, START FIELD 10*)
    IFSATR(ADDR, ATR_BUF,ATR_COUNT,ATR_START,ATR_CURSOR, ATR_TOT,CCODE,WCC);
    IF CCODE <> 0 THEN RPTERR(ADDR, CLS_OP ,CCODE)
    (*REPORT ERROR PROCEDURE*)
  END;

  TIME_OUT: BEGIN
    TIME_OUT_INTERVALS:=2; (*WAIT TWO 50 MSEC INTERVALS*)
    IFSTMO(ADDR, TIME_OUT_INTERVALS, CCODE);
    IF CCODE <> 0 THEN RPTERR(ADDR, CLS_OP ,CCODE)
    (*REPORT ERROR PROCEDURE*)
  END;

  TRANSMIT: BEGIN
    TRANSMIT_AID_CODE:=0; (*SET AID CODE TO ENTER*)
    IFSXMT(ADDR, TRANSMIT_AID_CODE, CCODE);
    IF CCODE <> 0 THEN RPTERR(ADDR, CLS_OP ,CCODE)
    (*REPORT ERROR PROCEDURE*)
  END;

  CLOSE: BEGIN
    IFSCLS(ADDR, CLOSE_OP, CCODE);
    IF CCODE <> 0 THEN RPTERR(ADDR, CLS_OP, CCODE)
    (*REPORT ERROR PROCEDURE*)
  END;

  (* OTHER ROUTINES AND CODE... *)

END.

```

Figure 5-9. Example Pascal Calls to PSC Routines (Sheet 2 of 2)

* THIS EXAMPLE ILLUSTRATES PSC CALLS AND THE ASSOCIATED
 * DATA STRUCTURES FOR 990 ASSEMBLY LANGUAGE. HOWEVER,
 * THE SEQUENCE OF THE CALLS DOES NOT REFLECT HOW A TYPICAL
 * PCS PROGRAM SHOULD BE STRUCTURED.

IDT 'PSEX'

* THE FOLLOWING REF STATEMENTS ARE USED FOR THE BLWP
 * INSTRUCTIONS TO THE PSC RUN-TIME MODULES.

REF IC\$OPN	OPEN ROUTINE
REF IC\$FIL	FILL ROUTINE
REF IC\$GET	GET ROUTINE
REF IC\$XMT	TRANSMIT ROUTINE
REF IC\$TMO	TIME-OUT ROUTINE
REF IC\$ATR	RETRIEVE ATTRIBUTES ROUTINE
REF IC\$CLS	CLOSE ROUTINE

* SET-UP ARGUMENT LISTS

OPNARG DATA 4*2	FOUR TWO-BYTE POINTERS
DATA OPN1	STATION ID POINTER
DATA OPN2	ACTIVATION FLAG POINTER
DATA OPN3	DUMMY POINTER
DATA OPNCC	COMPLETION CODE POINTER
FILARG DATA 8*2	EIGHT TWO-BYTE POINTERS
DATA OPN1	STATION ID POINTER
DATA FIL2	STATION BUFFER POINTER
DATA FIL3	LENGTH OF FILL POINTER
DATA FIL4	FIELD NUMBER POINTER
DATA FIL5	POSITION POINTER
DATA FIL6	NEXT ACTION POINTER
DATA FILCC	COMPLETION CODE POINTER
DATA FIL8	ATTRIBUTE POINTER
GETARG DATA 8*2	EIGHT TWO-BYTE POINTERS
DATA OPN1	STATION ID POINTER
DATA GET2	STATION BUFFER POINTER
DATA GET3	LENGTH OF GET POINTER
DATA GET4	FIELD NUMBER POINTER
DATA GET5	POSITION POINTER
DATA GET6	NEXT ACTION POINTER
DATA GETCC	COMPLETION CODE POINTER
DATA GET8	ATTRIBUTE POINTER
ATRARG DATA 8*2	EIGHT TWO-BYTE POINTERS
DATA OPN1	STATION ID POINTER
DATA ATR2	ATTRIBUTE BUFFER POINTER
DATA ATR3	NUMBER OF ATTRIBUTES POINTER
DATA ATR4	FIRST FIELD POINTER
DATA ATR5	CURSOR ADDRESS POINTER
DATA ATR6	NUMBER OF ATTRIBUTES RECEIVED POINTER
DATA ATRCC	COMPLETION CODE POINTER
DATA WCC	WCC OF CURRENT SCREEN IMAGE
TMOARG DATA 3*2	THREE TWO-BYTE POINTERS
DATA OPN1	STATION ID POINTER
DATA TMO2	TIME-OUT POINTER
DATA TMOCC	COMPLETION CODE POINTER
XMTARG DATA 3*2	THREE TWO-BYTE POINTERS
DATA OPN1	STATION ID POINTER
DATA XMT2	AID CODE POINTER
DATA XMTCC	COMPLETION CODE POINTER
CLSARG DATA 3*2	THREE TWO-BYTE POINTERS
DATA OPN1	STATION ID POINTER
DATA CLS2	TERMINATE FLAG POINTER
DATA CLSCC	COMPLETION CODE POINTER

* VARIABLES USED BY ALL PSC ROUTINES

Figure 5-10. Example 990 Assembly Language Call Examples (Sheet 1 of 3)

```

OPN1  DATA >200      ICS LINE -- AUTO-SELECT CU ADDR
      DATA >200      TERMINAL ADDRESS -- AUTO-SELECT CRT
CCODE  DATA 0        COMPLETION CODE

*      VARIABLES SHARED BY GET AND FILL ROUTINES

PSCBUF BSS 240        RESERVE 240 BYTES OF MEMORY FOR BUFFER
ATTR   BSS 4          RESERVE TWO WORDS OF MEMORY FOR ATTRIBUTE INFO

*      CALL OPEN ROUTINE

OPN2  DATA 1        ACTIVATE INTERFACE
OPN3  DATA 0        DUMMY ARGUMENT
OPN4  DATA 0        COMPLETION CODE
      LI  RO,OPNARG  PARAMETER LIST POINTER
      BLWP @IC$OPN  SUBROUTINE LINKAGE

*      CALL FILL ROUTINE

FIL2  DATA PSCBUF   ADDRESS OF FILL DATA
FIL3  DATA 10       LENGTH OF FILL DATA
FIL4  DATA 1        FIRST FIELD
FIL5  DATA 1        FIRST POSITION
FIL6  DATA 0        NO ADDITIONAL ACTION
FILCC DATA 0        COMPLETION CODE
FIL8  BYTE 0        ATTRIBUTE BYTE
      BYTE 0        RESERVED FOR
      BYTE 0        FUTURE
      BYTE 0        ENHANCEMENTS
      LI  RO,FILARG  PARAMETER LIST POINTER
      BLWP @IC$FIL  SUBROUTINE LINKAGE

*      CALL GET ROUTINE

GET2  DATA FLDBUF   ADDRESS FOR DATA BUFFER
GET3  DATA 80       LENGTH OF DATA
GET4  DATA 2        START IN SECOND FIELD
GET5  DATA 1        FIRST POSITION
GET6  DATA 3        CLEAR BUFFER
GETCC DATA 0        COMPLETION CODE
GET8  BYTE 0        ATTRIBUTE BYTE
      BYTE 0        RESERVED FOR
      BYTE 0        FUTURE
      BYTE 0        ENHANCEMENTS
      LI  RO,GETARG  PARAMETER LIST POINTER
      BLWP @IC$GET  SUBROUTINE LINKAGE

*      CALL ATTRIBUTE ROUTINE

ATR2  DATA ATRBUF   ADDRESS FOR ATTRIBUTE BUFFER
ATR3  DATA 20       RETRIEVE TWENTY ATTRIBUTES
ATR4  DATA 10       STARTING WITH TENTH FIELD
ATR5  DATA 0        RECEIVE CURSOR POSITION
ATR6  DATA 0        RECEIVE COUNT OF FIELDS
ATRCC DATA 0        COMPLETION CODE
WCC   DATA 0        WCC OF CURRENT SCREEN

*      CALL TIME-OUT ROUTINE

TMO2  DATA 2        TWO 50 MS INTERVALS
TMOCC DATA 0        COMPLETION CODE
      LI  RO,TMOARG  PARAMETER LIST POINTER
      BLWP @IC$TMO  SUBROUTINE LINKAGE

*      CALL TRANSMIT ROUTINE

```

Figure 5-10. Example 990 Assembly Language Call Examples (Sheet 2 of 3)

```

XMT2  DATA 2          AID CODE
XMTCC DATA 0          COMPLETION CODE
      LI  R0,XMTARG    PARAMETER LIST POINTER
      BWLP @IC$XMT     SUBROUTINE LINKAGE

*      CALL CLOSE ROUTINE

CLS2  DATA 0          ALLOW REACTIVATION
CLSCC DATA 0          COMPLETION CODE
      LI  R0,CLSARG    PARAMETER LIST POINTER
      BLWP @IC$CLS     SUBROUTINE LINKAGE

      END

```

Figure 5-10. Example 990 Assembly Language Call Examples (Sheet 3 of 3)

5.5 LINKING PSC PROGRAMS

User-written programs must link in the PSC routines before execution. To include the necessary modules in a link stream, each individual object module is explicitly referenced by using an **INCLUDE** command, and a **LIBRARY** command is used to reference the DNOS 3270 ICS object library directory.

In addition to the modules required specifically for PSC programs, each language requires one or more libraries to be referenced in the link control stream. Exactly which modules are required depends upon the language utilities you use in your program. Many of these modules are optional. Refer to the programmer's manual for the language you are using to find out exactly what modules are required for your PSC program.

The DNOS Link Editor resolves references to the PSC routines when you include a **LIBRARY** statement referencing the DNOS 3270 ICS object library directory in your link control stream along with the required PSC object module and any optional modules. The patch object module is referenced in the following examples by the pathname (PSCPAT). The patch module is required because it provides patch area to the other PSC object modules. After each relink of your PSC program, you must then apply the patches to the PSC object modules before executing your program. The procedure that applies the patches is described in paragraph 5.6. You need to save the link map that the DNOS linker produces from your link control stream in order to apply the patches.

The following three facts must be considered when linking the user task:

- You must assign the PSC task a priority that matches the priority level of the ICSPSC task. The ICSPSC task is installed with a priority of three, and unless that is modified, your user-written PSC task must also be installed with a priority level of three. The ICSPSC task is located under the pathname **.\$\$COMMDN**.
- You should *not* install your PSC task as replicatable if you link the PSC subroutine object modules in the procedure segment of your task. If your link control stream specifies **FORMAT IMAGE REPLACE**, the link editor installs your program as replicatable. The Modify Task Entry (MTE) procedure allows you to change your program to nonreplicatable. Refer to the *DNOS System Command Interpreter Reference Manual* for information concerning the MTE procedure.

- If you intend to activate your PSC program with one of the language activation commands and you want the PSC Emulator to rebid your task, your task must reside in the DNOS system program file, S\$SHARED. If your program closes the PSC Emulator with the *Do not terminate* option, the PSC Emulator rebids your task when it receives data from the ICS line. If your task does not reside in .S\$SHARED, the PSC Emulator returns a program file error.

The actual pathnames of the PSC object modules vary depending on whether you received ICS on disk media as opposed to magnetic tape media or as an add-on package. If you received the object package as an add-on package or on magnetic tape media, the object library directory pathname is as follows:

.DNO3270.OBJ

If you received ICS on disk media, the object library directory pathname is as follows:

DNO3270.OBJ

You can place the object library on any available disk media that you desire. If you plan to do extensive PSC development work on your system, it is recommended that you copy the necessary ICS object modules onto a permanently installed disk. This operation can be specified during ICS object installation. Refer to Section 6 of the *DNOS ICS Object Installation Guide*. If you relocate the object library, remember that you must use the new pathname in the link control stream. The following examples assume that the object library directory is DNO3270.OBJ.

EXAMPLE

```

FORMAT IMAGE,REPLACE,3;      ASSIGN PRIORITY LEVEL 3
LIBRARY .FORT78.OSLOBJ;     STANDARD FORTRAN DIRECTORY
LIBRARY .FORT78.STLOBJ;     STANDARD FORTRAN DIRECTORY
LIBRARY DNO3270.OBJ;
TASK MYPSCTSK;              PSC ROUTINES INCLUDED IN TASK SEGMENT
    
```

—Required Modules: Use these for all applications.—

```

INCLUDE (PSCPAT)             PSC PATCH MODULE
INCLUDE (IFSOPN)             FORTRAN/PASCAL INTERFACE ROUTINE
INCLUDE (ICSOPN)             ICS PSC OPEN ROUTINE
INCLUDE (ICSCLS)             ICS PSC CLOSE ROUTINE
INCLUDE (ICSSUB)             ICS PSC ROUTINES
INCLUDE (THCBAS)             ICS PSC TASK VARIABLES
INCLUDE (THCONS)            ICS PSC TASK CONSTANTS
    
```

— Optional Modules: Use these if needed by application. —

```

INCLUDE (ICSFIL)             ICS PSC FILL ROUTINE
INCLUDE (ICSGET)             ICS PSC GET ROUTINE
INCLUDE (ICSXMT)             ICS PSC TRANSMIT ROUTINE
INCLUDE (ICSATR)             ICS PSC RETRIEVE ATTRIBUTES ROUTINE
INCLUDE (ICSTMO)             ICS PSC TIME-OUT ROUTINE
INCLUDE .OBJECT.MYPSCOBJ:    USER WRITTEN PSC OBJECT MODULE
END
    
```

Note that two of the libraries referenced are specifically required for FORTRAN programs. Refer to the programmer's guide for the language you are using to determine which libraries or modules are required for your program. For further information concerning the Link Editor refer to the *DNOS Link Editor Reference Manual*. Refer to the *DNOS 3270 ICS Release Information*, current release, for other programming languages examples.

5.6 PATCH ICS PSC USER TASK — PTPSC

The following paragraphs describe how to apply the current patches to the PSC routines linked with your PSC program. You must perform this procedure after each relink of your program before you can successfully execute your PSC program. It is assumed that you included the object module DNO3270.OBJ.PSCPAT (patch module) in your link control stream or that you used the library method when linking. If you have relocated the ICS object library, the first element (or elements) of this pathname are different. Use the pathname that defines the current location of this library. It is also assumed that the object modules for the PSC routines were included in the task segment of your program.

ICS provides a procedure that executes a batch stream that applies the current patches to the object modules of the PSC routines. Patches are sent with each new update of ICS. To apply the latest patches to the PSC routines, enter PTPSC in response to the SCI prompt. The following display appears:

```
[ ] PTPSC
      PATCH ICS USER TASK
      SYSTEM DISK/VOLUME NAME:
      EMULATOR OBJECT PATHNAME:
      LISTING ACCESS NAME:
      PSC USER LINKMAP:
      PSC USER PROGRAM FILE:
      PSC USER TASK NAME:
```

The command prompts and appropriate responses are as follows:

SYSTEM DISK/VOLUME NAME

Enter the name of the system disk where the communications utilities reside. This information is needed for the system to keep track of patch information for debugging purposes. Use the same system disk name used throughout the communications procedure.

EMULATOR OBJECT PATHNAME

Enter the pathname of the directory where the ICS PSC patch file resides. Remember where you store this patch file when you receive it from your customer representative. If your customer representative has sent you no new patch files since you received ICS, you must still apply the PSC patches that accompany the ICS release that you received. This patch file is located under the pathname <ICS object>.PAT.PSC3270, where <ICS object> represents the directory containing the ICS object directory. Standard object installation from hard disk media uses the pathname DNO3270.OBJ for the ICS object directory. If you receive ICS as an add on package use the pathname .DNO3270.

LISTING ACCESS NAME

Enter the name of the file where the PTPSC procedure writes the patch listing. Upon completion of this procedure, examine this file to determine if any errors occurred. The volume and directories specified in the pathname must already exist, and the directory under which the file is to be written must have space for one more file. If these conditions are met, the procedure automatically creates the listing file.

PSC USER LINKMAP

Enter the pathname of the file containing the link map generated when you linked your PSC program. That pathname should be the same pathname specified as the LISTING ACCESS NAME when you perform the Execute Link Editor (XLE) command.

PSC USER PROGRAM FILE

Enter the name of the program file in which the user tasks resides.

PSC USER TASK NAME

Enter the name of the PSC task that you wrote. This is the same name used in the PHASE or TASK statement of your link control stream.

Upon completion of the patch application, the following message is written to the station where the procedure was invoked:

```
ICS PSC USER TASK PATCH PROCESS COMPLETE
  n ERRORS IN PATCH STREAM           <listing access name>
```

If the value of *n* is nonzero, examine the listing file you specified to determine what error(s) occurred. Refer to the *DNOS Messages and Codes Reference Manual* to discover the meaning of any errors returned. If any errors occur, correct the problem and reexecute the procedure. If no errors occur, you are ready to execute your PSC program. If your application links the PSC runtime in a nonreplicable procedure, then a dummy link map must be generated in order to apply PSC patches. You can do this by editing the link control stream to include the PSC runtime and other modules of the procedure segment in the task segment. The resulting linkmap should return the same value for the starting address of the PSC runtime. You need to do so because PTPSC assumes the PSC runtime is linked in the task segment.

5.7 EXECUTE PSC TASK — XPSC

The SCI command procedure XPSC bids a PSC task to run in the background mode. Enter XPSC in response to the SCI prompt. The following display appears:

```
[ ] XPSC
EXECUTE ICS PROGRAMMED STATION CONTROL
COMMUNICATIONS LINE NAME:
TERMINAL ADDRESS(0-31):
TERMINAL TYPE(CRT/PRT):
USER PROGRAM FILE ACNM:
USER TASK NAME:
```

If you enter null values in response to the COMMUNICATIONS LINE NAME and the TERMINAL ADDRESS(0-31) prompts, ICS automatically selects them for you, provided that you specify the type of terminal address that your program requires. If you do not specify the type of terminal address required, ICS cannot automatically select the communications line name and an available terminal address.

The user-written PSC program can also select these addresses in the Open call. If your program specifies values for the communications line name or terminal address in < a1> for all of the PSC call statements, you must specify the same values when executing the XPSC command. You can use a control unit address in < a1> and a communications line name during XPSC if they both identify the same ICS communications line. If you specify auto-select by using a null value when executing XPSC, you must also specify auto-select in < a1> for the ICSOPN call in PSC program (see paragraph 5.2.4.1). In both cases, you must specify the terminal type. The program file name and task name or ID must be specified to identify the user-written task to be bid.

To simplify the operation for your data entry personnel, you may wish to create your own command procedure for activating each PSC task installed on your system. Modify the XPSC procedure by placing the required values for your PSC task in the parameters used in the bid primitive. Remember not to duplicate the name of an installed SCI command procedure when naming your customized procedure.

You can also use one of the following language activation procedures provided by the DNOS operating system:

Command Procedure	Description
Execute Task (XT)	Activates any program
Execute Pascal Task (XPT)	Activates a Pascal program
Execute COBOL Task (XCT)	Activates a COBOL program
Execute FORTRAN Task (XFT)	Activates a FORTRAN program

If you use one of these procedures to activate your PSC program, your program must provide the parameters that the XPSC command prompts identify by using the PSC routine call arguments. If your program issues a Close call specifying *Do not terminate*, the PSC Emulator attempts to rebid your program when it receives data from the ICS line. If you activate your program by using one of the language activation procedures and your program subsequently issues a Close call that specifies *Do not terminate*, the PSC Emulator can only rebid your task if you installed it in the .S\$\$SHARED system program file.

Appendix A

Keycap Cross-Reference

Generic keycap names that apply to all terminals are used for keys on keyboards throughout this manual. This appendix contains specific keyboard information to help you identify individual keys on any supported terminal. For instance, every terminal has an Attention key, but not all Attention keys look alike or have the same position on the keyboard. You can use the terminal information in this appendix to find the Attention key on any terminal.

The terminals supported are the 931 VDT, 911 VDT, 915 VDT, 940 EVT, the Business System terminal, and hard-copy terminals (including teleprinter devices). The 820 KSR has been used as a typical hard-copy terminal. The 915 VDT keyboard information is the same as that for the 911 VDT except where noted in the tables.

Appendix A contains three tables and keyboard drawings of the supported terminals.

Table A-1 lists the generic keycap names alphabetically and provides illustrations of the corresponding keycaps on each of the currently supported keyboards. When you need to press two keys to obtain a function, both keys are shown in the table. For example, on the 940 EVT the Attention key function is activated by pressing and holding down the Shift key while pressing the key labeled PREV FORM NEXT. Table A-1 shows the generic keycap name as Attention, and a corresponding illustration shows a key labeled SHIFT above a key named PREV FORM NEXT.

Function keys, such as F1, F2, and so on, are considered to be already generic and do not need further definition. However, a function key becomes generic when it does not appear on a certain keyboard but has an alternate key sequence. For that reason, the function keys are included in the table.












































Multiple key sequences and simultaneous keystrokes can also be described in generic keycap names that are applicable to all terminals. For example, you use a multiple key sequence and simultaneous keystrokes with the log-on function. You log on by *pressing the Attention key, then holding down the Shift key while you press the exclamation (!) key*. The same information in a table appears as *Attention/(Shift)!*.

Table A-2 shows some frequently used multiple key sequences.

Table A-3 lists the generic names for 911 keycap designations used in previous manuals. You can use this table to translate existing documentation into generic keycap documentation.

Figures A-1 through A-5 show diagrams of the 911 VDT, 915 VDT, 940 EVT, 931 VDT, and Business System terminal, respectively. Figure A-6 shows a diagram of the 820 KSR.

Table A-1. Generic Keypcap Names

Generic Name	911 VDT	940 EVT	931 VDT	Business System Terminal	820' KSR
Alternate Mode	None				None
Attention ²		 			 
Back Tab	None	 	 	None	 
Command ²					 
Control					
Delete Character					None
Enter					 
Erase Field					 






















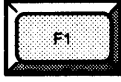





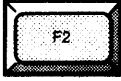











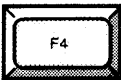





Notes:

¹The 820 KSR terminal has been used as a typical hard-copy terminal with the TPD Device Service Routine (DSR). Keys on other TPD devices may be missing or have different functions.

²On a 915 VDT the Command Key has the label F9 and the Attention Key has the label F10.

2284734 (2/14)



















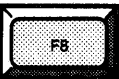


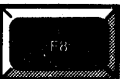














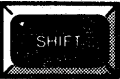



Table A-1. Generic Keycap Names (Continued)

Generic Name	911 VDT	940 EVT	931 VDT	Business System Terminal	820' KSR
Erase Input					 
Exit			 	 	
Forward Tab	 			 	 
F1					 
F2					 
F3					 
F4					 

Notes:

*The 820 KSR terminal has been used as a typical hard-copy terminal with the TPD Device Service Routine (DSR). Keys on other TPD devices may be missing or have different functions.














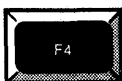




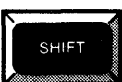

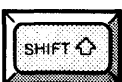

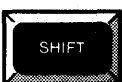
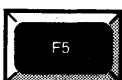






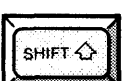


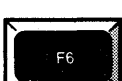









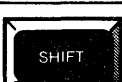





Table A-1. Generic Keycap Names (Continued)

Generic Name	911 VDT	940 EVT	931 VDT	Business System Terminal	820' KSR
F5					 
F6					 
F7					 
F8					 
F9	 			 	 
F10	 			 	 

Notes:

*The 820 KSR terminal has been used as a typical hard-copy terminal with the TPD Device Service Routine (DSR). Keys on other TPD devices may be missing or have different functions.

Table A-1. Generic Keycap Names (Continued)

Generic Name	911 VDT	940 EVT	931 VDT	Business System Terminal	820' KSR
F11	 			 	 
F12	 			 	 
F13	 	 	 	 	 
F14	 	 	 	 	 
Home					 
Initialize Input		 			 

Notes:

The 820 KSR terminal has been used as a typical hard-copy terminal with the TPD Device Service Routine (DSR). Keys on other TPD devices may be missing or have different functions.































Table A-1. Generic Keycap Names (Continued)

Generic Name	911 VDT	940 EVT	931 VDT	Business System Terminal	820' KSR
Insert Character					None
Next Character	 or 				None
Next Field	 		 	 	None
Next Line					 or
Previous Character	 or 				None
Previous Field		 			None

Notes:

*The 820 KSR terminal has been used as a typical hard-copy terminal with the TPD Device Service Routine (DSR). Keys on other TPD devices may be missing or have different functions.

Table A-1. Generic Keycap Names (Continued)

Generic Name	911 VDT	940 EVT	931 VDT	Business System Terminal	820' KSR
Previous Line					 
Print					None
Repeat		See Note 3	See Note 3	See Note 3	None
Return					
Shift					
Skip					None
Uppercase Lock					

Notes:

¹The 820 KSR terminal has been used as a typical hard-copy terminal with the TPD Device Service Routine (DSR). Keys on other TPD devices may be missing or have different functions.

²The keyboard is typamatic, and no repeat key is needed.

2284734 (7/14)

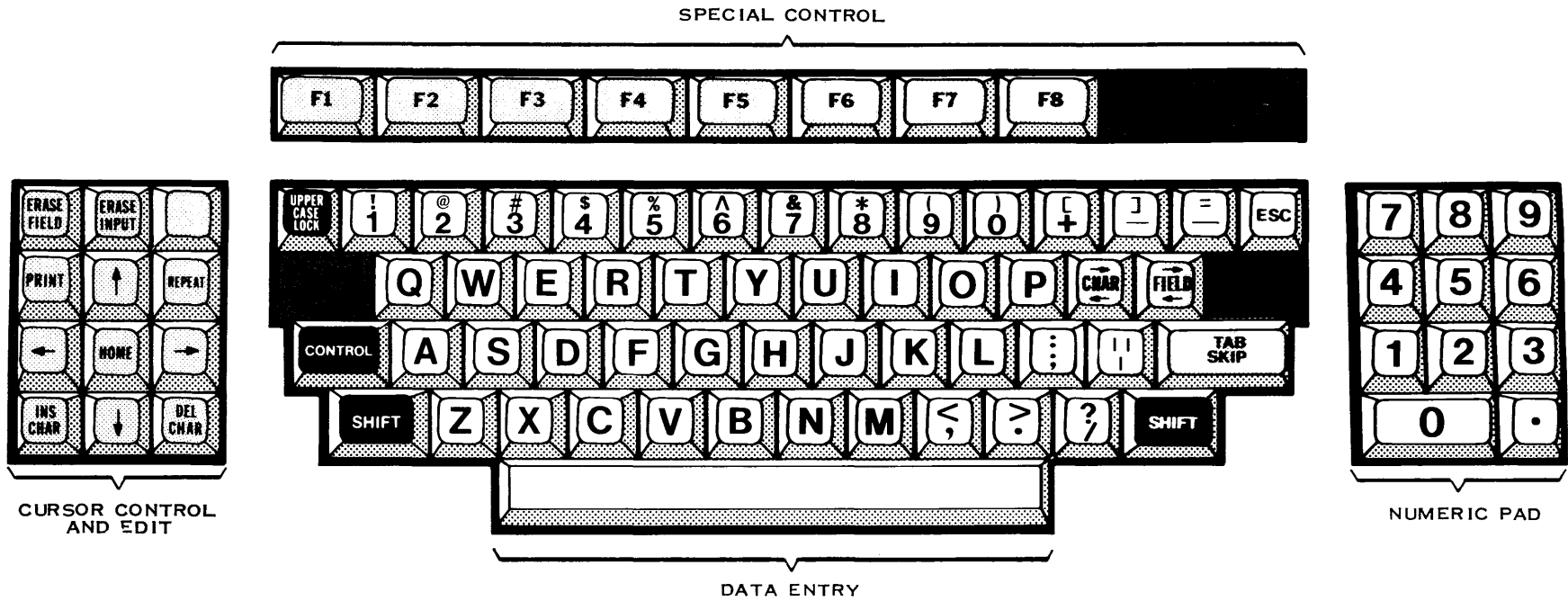
Table A-2. Frequently Used Key Sequences

Function	Key Sequence
Log-on	Attention/(Shift)!
Hard-break	Attention/(Control)x
Hold	Attention
Resume	Any key

Table A-3. 911 Keycap Name Equivalents

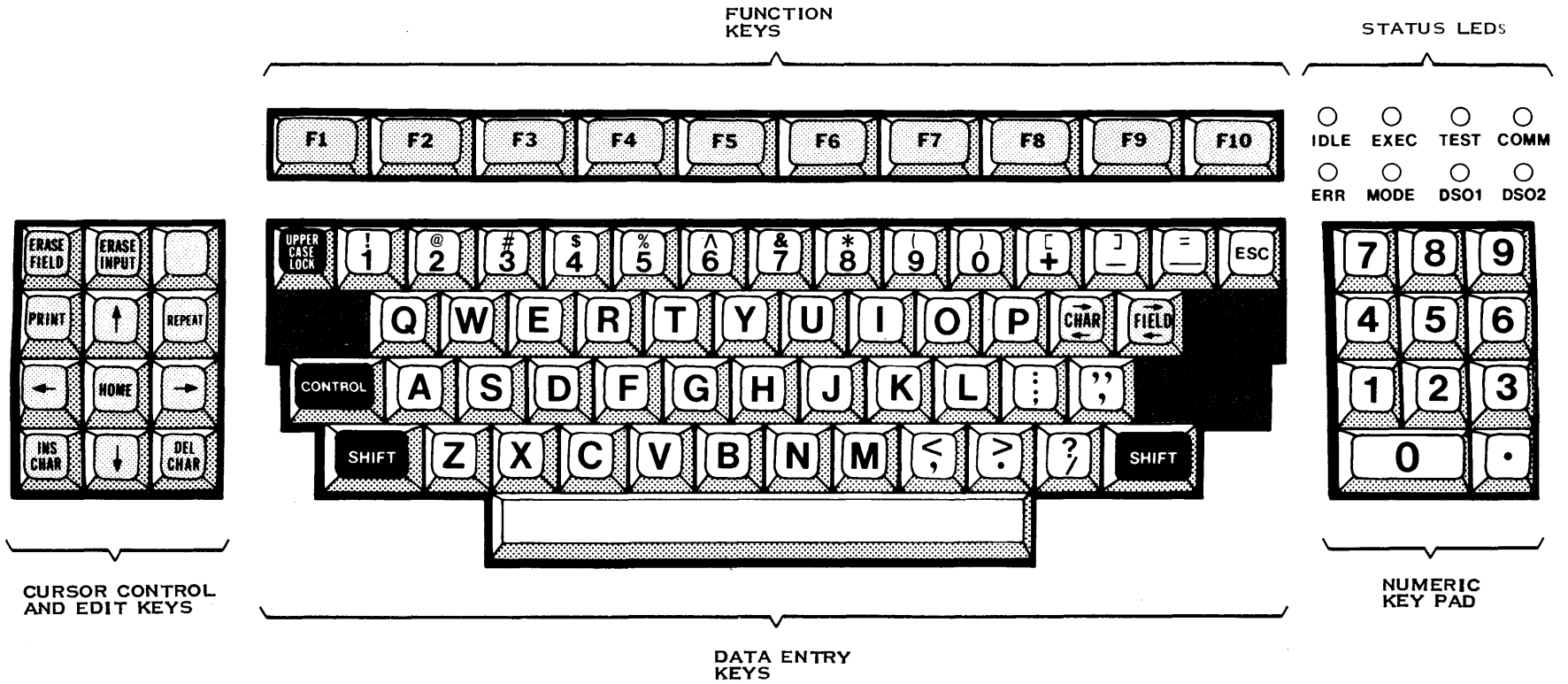
911 Phrase	Generic Name
Blank gray	Initialize Input
Blank orange	Attention
Down arrow	Next Line
Escape	Exit
Left arrow	Previous Character
Right arrow	Next Character
Up arrow	Previous Line

2284734 (8/14)



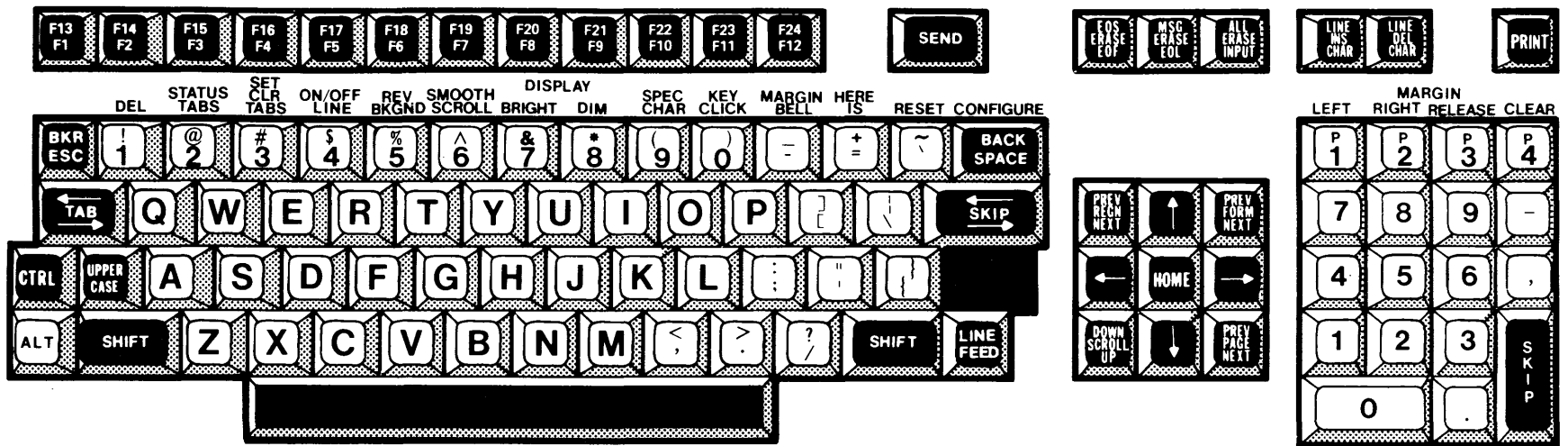
2284734 (9/14)

Figure A-1. 911 VDT Standard Keyboard Layout



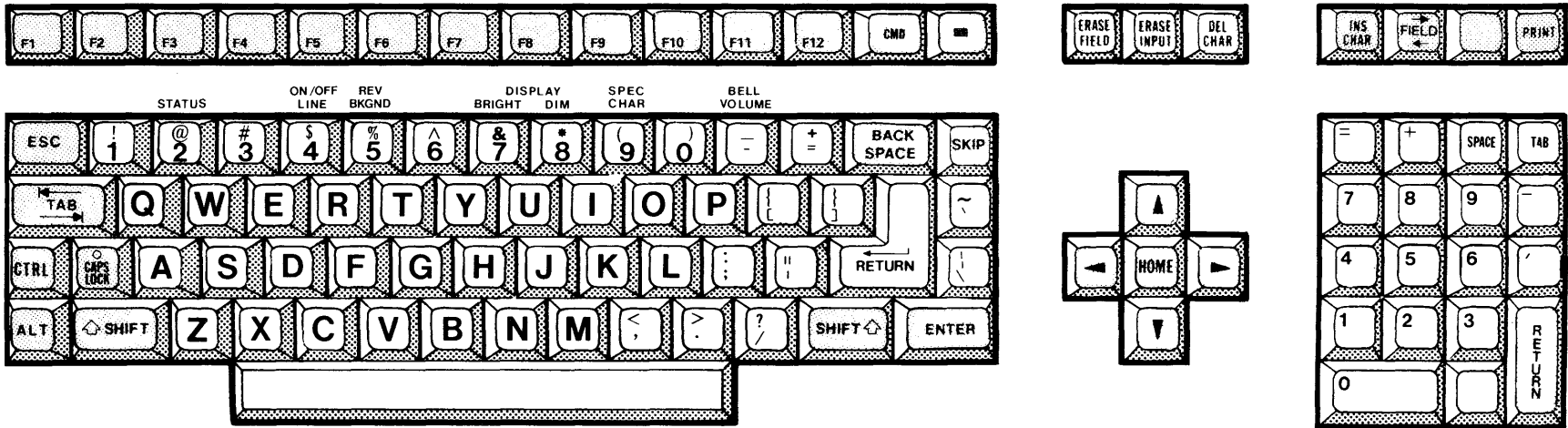
2284734 (10/14)

Figure A-2. 915 VDT Standard Keyboard Layout



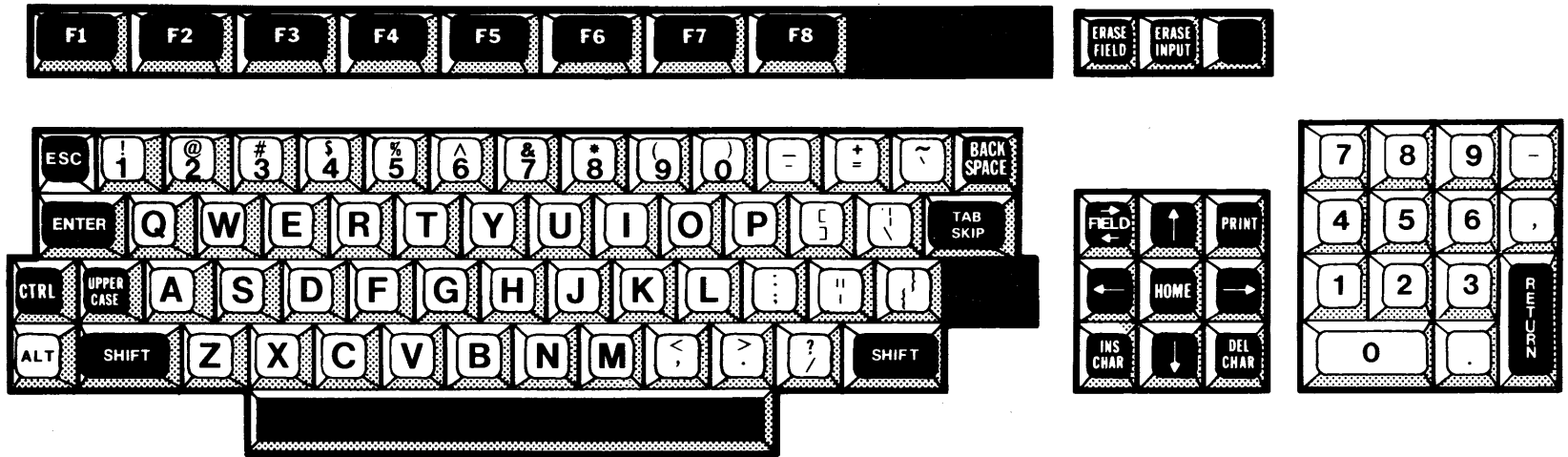
2284734 (11/14)

Figure A-3. 940 EVT Standard Keyboard Layout



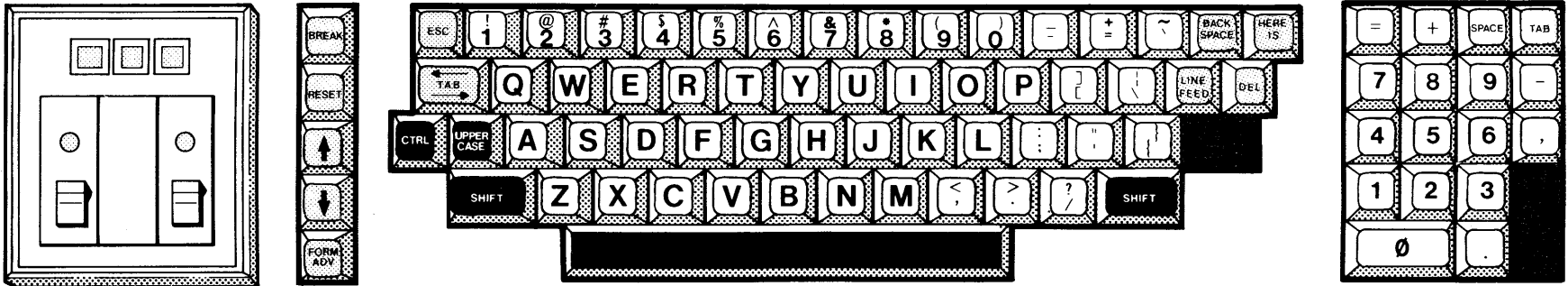
2284734 (12/14)

Figure A-4. 931 VDT Standard Keyboard Layout



2284734 (13/14)

Figure A-5. Business System Terminal Standard Keyboard Layout



2284734 (14/14)

Figure A-6. 820 KSR Standard Keyboard Layout

Appendix B

ICS Messages and Codes

B.1 INTRODUCTION

This appendix lists all of the messages that ICS generates. These messages are divided into the following three groups:

- Messages returned to a station or log file
- Internal device service routine (DSR) Supervisor Call (SVC) error codes
- Programmed Station Control (PSC) codes returned to the user-written PSC task

This appendix provides an explanation of each message and the source of each variable part. Corrective action is recommended where the message indicates an error condition. Some ICS messages return internal DSR errors as SVC return codes. Table B-1 lists the meaning of each internal DSR error code.

B.2 MESSAGES RETURNED TO ICS STATION OR SYSTEM LOGGING DEVICE

ICS displays its messages on the bottom line of video terminals in use as ICS stations. ICS also writes messages to the system logging device when no video terminals are associated with ICS operation. For instance, if an error occurs while an ICS line is active but no video terminal addresses on that line are active, ICS writes the error message to the system log.

Usually, the messages occur when you invoke an SCI command that controls ICS. In these cases, ICS writes the messages at the terminal where you enter the command. The SCI command procedure that you use is identified by a prefix consisting of an asterisk (*), the SCI command, and a dash (-). An example of this message format is as follows:

```
*XICT-TERMINAL ADDRESS 5 BELONGS TO A PRINTER:
```

The *XICT portion of the message indicates that the error occurred while using the Execute Interactive Communications Terminal (XICT) command. Most of the error messages use variants. In the previous example, 5 is a variant. You enter this value in response to a prompt; the program finds it unacceptable and returns it as part of the error message. In addition to user-entered variants, ICS returns variant values such as internal, SVC, and SCI error codes. The explanations of the error messages in this appendix identify the type of each variant. To find the meaning of a particular value of an SVC error code, see paragraph B.3. For an SCI error code, refer to the *DNOS Messages and Codes Reference Manual*.

Table B-1. 3270 DSR Error Codes

Hexadecimal Code	Description
02	Illegal operation. Contact your customer representative.
03	Device in use. The DSR returns this error code when the ICS station task has specified a station that is already active when a second ICS controller task is bid on a communications controller channel whose memory has been purged.
06	I/O aborted. The DSR returns this error when the I/O in progress is aborted.
07	Device error. The DSR has detected a hardware error.
18	Line disconnected.
26	Read Only Memory (ROM) code not correct for specified patch file. You do not have the required firmware installed on the ICS communications controller. The revision level of your FCCC should be revision P or later. The version level of your BCAIM should be A or later.
43	Write parameters not legal with I/O in progress. Write parameters have been specified by the MICA task while an I/O was in progress.
44	Invalid protocol. The wrong character detect module has not been downloaded to BCAIM or FCCC. Refer to the <i>ICS Object Installation Guide</i> for corrective information.
5E	An illegal station. The ICS station task has specified an illegal station during an open, write, or close. The ICS and DSR parameters are not consistent. Ensure that you have correctly installed and generated the ICS tasks and the communications DSR.
72	Extended flags not specified. Either the controller task did not specify the controller ID and number of stations during an open, or a station task did not specify the station during an open, write, or close.

When ICS sends one of these messages to the system log file or device, the Julian date and time precede the message. ICS provides an additional prefix for some messages to indicate the communications line and terminal address associated with the error. This prefix is of the following general form:

CMxx,TMyy

where:

xx represents the numerical portion of the communications device name

yy represents the decimal value for the terminal address

The text of these messages can be modified by the systems programmer. *Release Information, DNOS 3270 ICS* contains instructions for modifying ICS messages.

The following discussion lists and explains all of the ICS messages. Variants are identified by their general type. For example, CM03 is a communications device name. The messages in this appendix are listed in alphabetical order without the prefix. Two of the ICS error messages begin with variants. The explanations for these messages are at the end of this paragraph (B.2), alphabetized by the second element of the message.

ALL ICS COMMUNICATIONS LINES ARE ACTIVE:

You executed the Execute Interactive Communications Controller (XICC) command and entered no response to the COMMUNICATIONS LINE NAME prompt. ICS searches a list of installed ICS lines and tries to select the first inactive installed line. ICS displays this message when all installed ICS communications lines are already active. You cannot activate a line that is already active.

ALL ICS PRINTER ADDRESSES ARE IN USE:

You entered a null value for the TERMINAL ADDRESS(0-31) prompt while using the Execute Interactive Communications Printer (XICP) command. The ICS task searches for an available printer terminal address, finds none, and displays this message at your terminal. Verify this by executing the List Interactive Communications Configuration (LICC) command. The configuration listing shows all of the reserved terminal addresses, their type (CRT or PRT), and their current state (active, suspended, idle, disabled, active PSC, or waiting PSC).

ALL ICS VIDEO ADDRESSES ARE IN USE:

ICS displays this message at your terminal when there are no available ICS video terminal addresses. The List Interactive Communications Configuration (LICC) command returns a list of installed terminal addresses, their types (CRT or PRT), and their current states (active, suspended, idle, disabled, active PSC, or waiting PSC). Wait until one of the currently active or suspended ICS stations terminates.

CANNOT MODIFY CONTROL UNIT ADDRESS FOR ACTIVE ICS LINE xx:

You attempted to modify the control unit address assigned to the ICS line identified by xx. This line is currently active, and ICS does not allow you to modify the control unit address of an active line. You should perform this modification immediately after the initial program load (IPL) and before activating the ICS line. ICS displays this message at the terminal at which you entered the Modify Control Unit Address (MCUA) command.

CANNOT MODIFY ICS CONFIGURATION FOR ACTIVE LINE xx:

You attempted to modify the configuration of the ICS line identified by xx. This line is currently active, and ICS does not allow you to modify the state or type of a terminal address assigned to an active line. You should perform this modification immediately after the initial program load (IPL) and before activating the ICS line. ICS displays this message at the terminal at which you entered the MICC command.

CANNOT MODIFY LAST CRT ADDRESS:

You attempted to disable the only video terminal address on an ICS line or attempted to change its type to PRT (printer). At least one terminal address on each installed ICS line must be of the CRT (video) type and cannot be disabled. ICS displays this message at your terminal.

COMMUNICATIONS LINE NAME REQUIRED:

You invoked an SCI command procedure that requires a response to the COMMUNICATIONS LINE NAME prompt, but you responded with a null value. Reexecute the command specifying the desired communications line name. ICS displays this message at your terminal.

COMMUNICATIONS LINE xx ACTIVATED:

ICS displays this message to verify the successful activation of the ICS line identified by xx. This is not an error message.

COMMUNICATIONS LINE xx ALREADY ACTIVE:

You attempted to activate a communications line that is already active. The line is identified by xx. You need to activate each line only once after each initial program load (IPL). ICS displays this message at your terminal.

COMMUNICATIONS LINE DISCONNECTED FOR xx:

ICS detected a hardware error at the communications line identified by xx. This error indicates the loss of the Data Set Ready signal. Ensure that the cable from the communications line to the modem is properly connected. This error can also occur if the modem is in the loopback mode, so check the AL or DL switches on the modem. ICS writes this message to the system logging device, and if you have a transmission pending, ICS displays this message at your terminal.

CONTROL UNIT ADDRESS FOR xx MODIFIED FROM yy TO zz:

You have successfully modified the control unit address for the ICS line identified by xx from the value identified by yy to the new value identified by zz. ICS displays this message at your terminal.

CONTROL UNIT ADDRESS REQUIRED:

You have changed the Modify Control Unit Address (MCUA) command procedure. You must enter a new value for the control unit address. Each installed ICS communications line must have a valid control unit address assigned to it. ICS displays this message at your terminal.

DATA CONVERSION ERROR:

ICS detected an error while attempting to convert a character code to a binary value or while attempting to convert a binary value to a character code. This is an internal error that you should report to your customer representative.

ERROR xx IN ACCESS TO PARAMETER yy:

ICS detected an SCI error. The parameter yy is a user-entered pathname or device name. This is usually a communications line name. Check to see if the device specified is installed on the operating system. Error xx is an SCI error. The *DNOS Messages and Codes Reference Manual* details SCI error codes. ICS displays this message at your terminal.

HIGHEST TERMINAL ADDRESS IS xx:

You specified a terminal address that exceeds the highest terminal address reserved for the ICS line during installation. The List Interactive Communications Configuration (LICC) command provides a list of installed terminal addresses. Choose an address that is listed as enabled, ensuring that the type is correct. ICS displays this message at your terminal.

HOST NOT POLLING COMMUNICATIONS LINE xx:

ICS detected no polling activity on the communications link for one minute. The absence of this message does not mean that the particular control unit address associated with ICS is being polled. Usually, when this message appears, the host is inoperative. The ICS line is identified by xx. Remember that the host identifies ICS by its EBCDIC control unit address. The List Interactive Communications Configuration (LICC) command shows the decimal value of the control unit address assigned to an ICS line. Table 2-1 lists the EBCDIC value equivalent to the ICS decimal value. The systems programmer can use the Execute Poll Analyzer (XPA) command to verify this problem. If this message occurs frequently, you may need to increase the poll time-out (XSGU parameter) as discussed in the *DNOS 3270 ICS Object Installation Guide*. ICS displays this message at your terminal.

ICS NOT RUNNING AT xx:

You specified a terminal pathname (identified by xx) that is not assigned to an active or suspended ICS station. You cannot terminate an ICS station that is already inactive. Use the List Interactive Communications Configuration (LICC) command to find out which ICS terminal addresses are active or suspended. LICC also lists any video stations that may be associated with those ICS terminal addresses. ICS displays this message at your terminal.

ICSPSC TERMINATED:

The Programmed Station Control (PSC) interface to ICS has been terminated. This is an informative message, not necessarily an error. Your PSC program can terminate ICSPSC by issuing a Close call with the second argument (<a2>) specifying *terminate*. Issuing an Open call with the second argument (<a2>) specifying *activate* reactivates ICSPSC. ICS writes this message to the system logging device.

INTERNAL ERROR, INVALID DEVICE ADDRESS FOR yy:

ICS detected an internal controller error. The device (terminal) address identified by yy is invalid. Consult your systems programmer. ICS writes this message to the system logging device.

INTERNAL ERROR, UNDEFINED TERMINAL STATE FOR xx:

ICS attempted to access a terminal address whose state cannot be determined. The terminal address is identified by xx. This is a fatal error. The ICS task that controls this terminal address should be terminated. Consult your systems programmer. ICS writes this message to the system logging device.

INTERNAL ERROR, xx READ QUEUE:

ICS detected an internal controller task error. The ICS communications line where the error occurred is identified by xx. One possible cause of this error is that ICS can no longer access an internal buffer. The ICS controller task terminates. Consult your systems programmer. ICS writes this message to the system logging device.

INTERNAL ERROR ww, WP:xx, PC:yy, ST:zz:

ICS detected an internal task error identified by ww. The xx after WP indicates the value of the workspace pointer when ICS detected the error. The yy after PC indicates the program counter contents, and the zz after ST indicates the value of the status register when ICS detected the error. Consult your customer representative. ICS writes this message to your terminal.

INVALID CONTROL UNIT ADDRESS xx:

You entered a value (identified by xx) that is not in the range of 0 through 31 in response to the CONTROL UNIT ADDRESS prompt. Reexecute the Modify Control Unit Address (MCUA) command, specifying a control unit address in the range of 0 through 31. You can confirm the results of this operation by using the List Interactive Communications Configuration (LICC) command. ICS displays this message at your terminal.

INVALID DATA RECEIVED:

The host application sent a code that ICS does not support. This is not necessarily a fatal error. Some applications can service different types of 3270 stations and emulators for which these codes can have special meaning. Report these errors to your systems programmer. ICS writes this message to the systems logging device and displays it at your terminal.

INVALID ICS PROTOCOL FOR xx:

You responded to the COMMUNICATIONS LINE NAME prompt with a value that does not identify a valid ICS line. The value that you entered is identified by xx. This release of ICS supports only the Binary Synchronous Communications (BSC) procedures that the IBM 3271 Model 2 Control Unit uses. Consult your customer representative. ICS displays this message at your terminal.

INVALID ICS TASK NAME: xx:

An ICS command procedure has been incorrectly modified. The task name sent as a parameter of a .BID primitive statement is identified by xx. Your systems programmer should correct the error in the command procedure. ICS displays this message at your terminal.

INVALID LOCAL FORMAT: zz:

The ICSPSC task encountered an error while accessing the local format file. If you delete the local format file and start over, this error should not reoccur. The pathname of the local format file (represented by zz) is of the general form CMxxTMy under the directory .S\$ICS where xx represents the decimal control unit address of the ICS line and yy represents the terminal address. If this error persists, contact your customer representative. ICS writes this message to your terminal or the system logging device.

INVALID PRINTER PATHNAME xx:

The value you entered in response to the OUTPUT PATHNAME prompt is invalid. This value is identified by xx and must be a valid pathname or a device name of an installed sequential output device (other than a communications controller). ICS displays this message at your terminal.

INVALID PRINT KEY PATHNAME xx:

ICS displays this message at your terminal when you have specified an invalid pathname in response to the PRINT KEY PATHNAME prompt and you press the PRINT key. ICS writes this message to the system logging device when an invalid pathname is assigned to the PRINT key pathname and your PSC program generates the print code. Reactivate the display station and enter a valid pathname, or use the identification function. If your PSC program reports this error, you must activate the terminal address used by your program, specify a valid PRINT key pathname, and then terminate the station before reexecuting your PSC program.

INVALID PROGRAM FILE PATHNAME: xx:

ICS cannot access the program file pathname identified by xx. This is the value you specified during the Execute ICS Programmed Station Control Task (XPSC) command. Make sure that you enter a valid DNOS pathname for the program file containing your PSC task. You can determine whether the value entered is a valid program file pathname by using the Map Program File (MPF) command. ICS displays this message at your terminal.

INVALID PSC USER TASK NAME: xx:

The task name you entered in response to the TASK NAME prompt is not found in the program file specified. Make sure that the program file name specified is the desired program file, and check to see if the specified task name is in that program file. You can verify this with the Map Program File (MPF) command. ICS displays this message at your terminal.

INVALID TERMINAL ADDRESS: xx:

Your response to the TERMINAL ADDRESS(0-31) prompt is not a decimal value in the range of 0 through 31. The value you enter is identified by xx. Reexecute the command and specify a valid terminal address, or let ICS select one for you (that is, enter a null value). ICS displays this message at your terminal.

INVALID TERMINAL STATE xx:

You entered an invalid modifiable terminal state. When modifying the state of a terminal address using the Modify Interactive Communications Configuration (MICC) command, the only valid responses to the STATE prompt are ENABLED and DISABLED. Reexecute the command, specifying one of these states. ICS displays this message at your terminal.

INVALID TERMINAL TYPE xx:

You entered an invalid terminal type in response to the TERMINAL TYPE(CRT/PRT) prompt. The value identified by xx is the value you entered. Retry, specifying CRT (video) or PRT (printer). ICS displays this message at your terminal.

NO TASK MEMORY FOUND AT >xxxx:

The Execute Poll Analyzer (XPA) command procedure does not match your configuration. Each time you reinstall ICS, you must also generate the XPA command procedure. The *DNOS 3270 ICS Object Installation* document describes how to do this. The memory location, identified by xxxx, is where the current XPA command procedure searches for the ICS task area.

ONLY VALID FOR A VIDEO TERMINAL:

Your response to the TERMINAL PATHNAME prompt is not a supported ICS video terminal. Use a device name of a video terminal that ICS supports, and reexecute the command. ICS displays this message at your terminal.

PSC USER PROGRAM FILE ACNM REQUIRED:

You entered a null value in response to the USER PROGRAM FILE prompt, and the Execute ICS Programmed Station Control Task (XPSC) command procedure has been modified. You must enter the program file pathname where your PSC program is located. If you do not know the correct program file pathname, consult your systems programmer. ICS displays this message at your terminal.

SVC ERR: xx, OP: yy, PN: zz:

ICS encountered an I/O SVC error identified by xx. If the value identified by zz is a communications line name, you should first consult the device service routine (DSR) internal error codes listed in Table B-1. If the value identified by xx is not listed in Table B-1, consult the installation and operation manual appropriate for the communications controller that the ICS line uses. These manuals are listed in paragraph A.3. If the returned code is not listed in one of these manuals, look it up in the *DNOS Messages and Codes Reference Manual*. The SVC operation code is identified by yy. The value zz identifies the pathname or access name of the device that was being accessed by the supervisor call (SVC) block.

TASK AND DSR INCONSISTENT FOR xx: COMPLETE SYSGEN:

ICS detected an inconsistency in the tables generated during installation. This message indicates that inconsistent data was entered for the HIGHEST TERMINAL ADDR prompts during ISC DSR and task installation. Refer to the *DNOS 3270 ICS Object Installation* document and the *DX10 Common Communications DSR Object Installation* document for installation instructions. This is only a warning message.

TASK ERROR ww, WP: xx, PC: yy, ST: zz:

This internal error is identified by ww and generated by the ICS video station task. The value represented by xx is the workspace pointer, yy is the program counter, and zz is the status register. You should report this error to your customer representative.

TERMINAL ADDRESS NEEDED:

A single physical device has more than one terminal address assigned to it, and you did not provide the terminal address when terminating the ICS station. To terminate both ICS stations at a single physical device, you must execute the Quit Interactive Communications Terminal (QICT) command twice. The first termination requires that the terminal address be specified. The terminal pathname is not required. After you terminate the first terminal address, the second execution of QICT does not require that you specify the terminal address. When only one terminal address is assigned to a single physical device serving as an ICS station, you can designate either the terminal pathname or the terminal address when terminating ICS. ICS displays this message at your terminal.

TERMINAL ADDRESS OR TYPE REQUIRED:

When activating a user-written Programmed Station Control (PSC) task, you must specify either the terminal address or the terminal type that your program requires. If you do not specify the terminal address, ICS selects one for you, but ICS must know what type of terminal address to select (PRT or CRT). If you specify a particular terminal address, ICS assumes that the type of the specified address is correct. If your PSC program uses auto-select, do not specify a particular terminal address; specify only the type. If your PSC program uses a particular terminal address, use that terminal address when activating your program. ICS displays this message at the terminal at which you enter the XPSC command.

TERMINAL PATHNAME OR ADDRESS MUST BE ENTERED:

When terminating an ICS station, you must specify either the terminal address or the terminal pathname assigned to that address. Either one of these values identifies a specific ICS station except when one physical device has more than one terminal address assigned to it. ICS displays this message at the terminal at which you enter the Quit Interactive Communications Terminal (QICT) command.

TERMINAL ADDRESS xx BELONGS TO A PRINTER:

You attempted to use a terminal address that is reserved for a printer, but a video terminal address is required. You can use the List Interactive Communications Configuration (LICC) command to get a list of all of the installed addresses and their type (CRT or PRT). Reselect a terminal address of the CRT type that is idle, or enter a null value and let ICS select one. ICS displays this message at your terminal.

TERMINAL ADDRESS xx BELONGS TO A VIDEO TERMINAL:

The terminal address you specify during an ICS printer station activation is the CRT (video) type. Either select another address or let ICS select the address by entering a null value in response to the TERMINAL ADDRESS(0-31) prompt. The List Interactive Communications Configuration (LICC) command returns a list of the terminal addresses and the states and types of each. If all video terminal addresses are in use, you must wait until one is terminated. ICS displays this message at your terminal.

TERMINAL ADDRESS xx IS BUSY:

The terminal address you specify is already in use by another user. The requested address is identified by xx. Choose another terminal address or let ICS choose one for you when you activate an ICS station. ICS automatically selects an idle terminal address of the correct type when you enter a null value in response to the TERMINAL ADDRESS(0-31) prompt. ICS displays this message at your terminal.

TERMINAL ADDRESS xx NOT IN USE:

The terminal address you entered (represented by xx) is not in an active or suspended state. You cannot terminate an ICS station that is inactive. The List Interactive Communications Configuration (LICC) command provides a listing of the states of each terminal address reserved for a particular ICS communications line. LICC shows the state of an inactive terminal address as either IDLE or DISABLED. ICS displays this message at your terminal.

TERMINAL ADDRESS xx ON COMMUNICATIONS LINE yy ACTIVATED:

An ICS station has been successfully activated. The value xx represents the terminal address that is activated, and yy indicates the ICS communications line to which the terminal address belongs. ICS writes this message to the system logging device.

TERMINATED BY ANOTHER USER OR HARD BREAK KEY:

Another user has terminated your ICS station by using the Quit Interactive Communications Terminal (QICT) command, or you terminated your ICS station by typing a Hard Break key sequence. This message indicates a normal termination. ICS displays this message at your terminal.

UNABLE TO TERMINATE ICS TASK, RUN ID xx:

The ICS task could not be terminated, with the value xx representing the task run ID of the task. You should retry the termination since there may be a transmission pending. When the pending transmission is complete, you can successfully perform the termination. This is an internal ICS task error. If it persists, report it to your customer representative.

USER TASK NOT FOUND, RUN ID:xx, JOBNAME:yyyyyyyy

The PSC emulator cannot find user task with a RUN ID of xx and JOBNAME of yyyyyyyy. This may be caused by failure of OPEN in user task. If you are bidding a task with XPSC, verify that the prompt responses are consistent with the user application OPEN request.

WARNING: DATA BUFFER HAS BEEN WRITTEN OVER BY NEW DATA:

The station buffer belonging to a user-written Programmed Station Control (PSC) task has received new data from the host. Any data put into the station buffer that has not been transmitted is lost. Refill the station buffer before transmitting. You can avoid this error by specifying transmit as the action to follow a Get or Fill call. To do this, put a one in the sixth argument (<a6>). Paragraph 5.3 describes the PSC routines. ICS writes this message to the system logging device.

The following messages are alphabetized according to their second elements since their first elements are variants:

xx HAS NOT BEEN ACTIVATED BY 'XICC':

The ICS communications line identified by xx has not been activated. To activate an ICS station, you must first activate the ICS line. Activate the line by using the Execute Interactive Communications Controller (XICC) command before retrying the station activation. ICS displays this message at your terminal.

xx IS NOT AN ICS COMMUNICATIONS LINE:

Your response to the COMMUNICATIONS LINE NAME prompt is not a valid ICS communications line. The value you entered is identified by xx. Reenter the SCI command, specifying a valid ICS communications line. ICS displays this message at your terminal.

B.3 DEVICE SERVICE ROUTINE (DSR) ERROR CODES

Table B-1 lists the internal DSR error codes that ICS returns in some messages as SVC errors. If the error you received is not in this list, consult the appropriate hardware reference manual, as follows:

- *CP501 and CP502 Installation and Operation Manual*
- *CP503 Installation and Operation Manual*

The Communications Interface Module (CIM) and the Two-Channel Communications Interface Module do not return error codes to the communications DSR. If your ICS line uses a CIM as its communications controller, you do not need to reference the CIM hardware manual.

If the SVC error that ICS returned is not listed in either Table B-1 or the BCAIM or FCCC hardware manual, look up the SVC error in the *DNOS Messages and Codes Reference Manual*.

B.4 PSC ERROR CODES AND DESCRIPTIONS

The following list describes the error codes reported to a user-written program making use of the capability to emulate an ICS station. If an error is detected when you activate a PSC program by using the Execute ICS Programmed Station Control Task (XPSC) command, the error is returned to the station at which you entered the command. These messages are listed along with the other ICS messages in paragraph B.2.

Error Code	Description
0001	ICS station already open. A PSC task attempted to open an ICS station already opened by another ICS task.
0002	Illegal number of arguments in a call statement. Some PSC routines generate this error code when too few or too many arguments are specified by the call statement. Count the arguments specified in the call statement and either add the missing argument or delete the extra one.
0003	Invalid field or subfield. Compare the fourth and fifth arguments (< a4> and < a5>) of the most recent Fill or Get call with the format of the screen buffer to ensure that the fill or get operation begins within a field on the screen.
0004	Invalid control unit or station address. The first argument (< a1>) of some call statement is not in the range of valid values. This usually occurs with the Open call statement. Reference the Open call statement description and Table 2-1 for the valid values for the IBM control unit and terminal addresses.
0005	ICS station already active (ready). The terminal address specified in your Open call is already in use (active). Either your program has already activated the PSC Emulator, or someone else is already using the requested address. Ensure that your program specified the correct control unit address.
0006	Control unit or terminal address not found. The control unit terminal address specified in the Open call statement has not been reserved during ICS installation. The List Interactive Communications Configuration (LICC) command displays the available addresses for each ICS line.
0007	ICS station not active. The terminal address specified in any call statement, except the Open, is that of a station that has not been opened with an Open call statement. Ensure that your program specified the correct control unit address.
0008	Invalid terminal type.

Error Code	Description
0009	Invalid function key code in a Transmit call statement. The value specified by the second argument (< a2>) of the Transmit call statement is not a valid program attention key code. See Table 5-1 for the list of supported attention ID (AID) codes.
000A	Invalid character in a fill buffer. See that the buffer specified as the second argument (< a2>) for the Fill routine contains only valid seven-bit ASCII characters.
000B	Attempt to call a PSC routine before the previous transmission has been processed by the host. Call the time-out routine after each call to the Get or Fill routine that transmits or clears the station buffer and after every call to the Transmit routine. If the error persists, call the Transmit routine specifying the reset attention ID (AID) code (17).
000C	Invalid operation for ICS printer station. The value specified for the first argument (< a1>) of the Open routine indicates that the PSC program is serving as a printer station, but the host transmitted a command with orders appropriate for a display station. Either change the value of the high-order byte of the word containing the terminal address or use a PSC printer station program to respond to the host application.
000D	No new data available. There is no data available for the PSC Transmit or Get routine called, so the PSC Emulator returns this informative message to indicate that no data was retrieved or transmitted.
000E	Host computer not polling (only on Transmit routine call statement). Ninety seconds have elapsed without a general or specific poll from the host. Your program should return this message to some logging device so that corrective action can be taken when necessary.
000F	Reserved.
0010	ICS system error. This problem may be caused by a lack of response from the communications hardware. Check the installation and test the hardware. If the error persists, contact your customer representative. See the system log file for further information.
0011	Reserved.

Error Code	Description
0012	Communications hardware error. A hardware error occurred in the communications equipment. Try your PSC request again.
0013	Reserved.
0014	Invalid option specified. The value of the second argument (< a2>) of an Open, Close, or Transmit call statement or the sixth argument (< a6>) of a Fill or Get call statement is not supported. See paragraph 5.3 for the valid values.
0015	Invalid address (odd address). The buffer defined by the second argument (< a2>) in a Fill or Get call statement does not start on an even address boundary. Check your data structures to ensure that the buffer begins on a word boundary (even address). If using 990 assembly language, an EVEN statement prior to the second argument (< a2>) data block ensures a word boundary.
0016	Accessed field protected on Fill call statement. The fourth and fifth arguments (< a4> and < a5>) of the Fill call statement specify a protected field. Check the attribute byte for protection before attempting to modify the field.
0017	Reserved.
0018	Terminal address not active. A PSC task has referenced an ICS terminal address that is no longer active. Check the system log for error messages.
0019	Invalid data length. Correct the length argument that indicates a field length that is negative, zero, or greater than 1920.
001A	Station not open. Make sure the PSC program calls the Open routine before it calls any other routines.
001B	Port/Controller conflict. Bits 6 and 7 of the first byte of the first word of the first argument (< a1>) in the Open call statement are both one. This is a conflict because bit 6 specifies auto-select a control unit address and bit 7 specifies that PSC uses a communications device name. Use one or the other, but not both.
001C	IPC channel error.

Error Code	Description
001D	Device not available. The terminal address specified in the first argument (< a1 >) of the Open call statement is either in use, disabled, or not reserved. To avoid this, use either auto-select or a terminal address that you are sure no one else is using.
001E	Emulator execution error. This is an internal error that should occur only if someone has modified the ICSPSC task. If this error occurs, report it to your customer representative.
001F	Program file error. This is an internal error that should occur only if someone has modified the ICSPSC task. It indicates that the start-up task is unable to successfully assign a logical unit number (LUNO) to the program file containing the ICSPSC task. If this error occurs, report it to your customer representative.
0020	Activate option error. This occurs if your program specifies <i>do not activate</i> when PSC is not active or if your program specifies <i>activate</i> when PSC is already active.
00FF	Station not responding (system error). ICSPSC has terminated abnormally. Check the system logging device to determine the cause of this termination. Consult your systems programmer.

The *DNOS Messages and Codes Reference Manual* describes other error codes that may be encountered.

Appendix C

ICS Hardware Requirements

C.1 INTRODUCTION

ICS runs with the DNOS standard equipment configuration. A Texas Instruments 990 computer system is required with the following minimum equipment:

- 512K bytes of memory
- A system disk (of at least 16M bytes capacity)
- A system console that can be either a video display terminal or an automatic send/receive data terminal
- A means of disk backup
- One of the communications interface sets listed in Table C-3 (for TI modems) or Table C-4 (for non-TI modems)

C.2 MEMORY REQUIREMENTS

ICS uses more memory space when its lines and terminals are activated than when they are inactive. The *static memory* requirements refer to the memory that ICS requires when it is inactive. The *dynamic memory* requirements refer to the memory that ICS uses when it is active.

The static memory requirements for ICS depend on the installed configuration. The following three considerations affect the total memory size of ICS:

- Receive buffer size for each ICS line
- Number of ICS lines installed
- Number of terminal addresses on each line

Since ICS shares a number of its tasks, you cannot determine the memory requirements by simple addition. Installations with two ICS lines do not require twice as much memory as those with only one ICS line. You can determine the approximate memory size of ICS by using the information in the following paragraphs.

The memory size of ICS for an installation with only one ICS line having only one reserved terminal address depends on the size specified for the receive buffer. The *DNOS 3270 ICS Object Installation* explains how the size of the receive buffer is determined. Table C-1 lists the amount of memory allocated for each declared buffer size.

Table C-1. Allocated Memory for Each Buffer Size

Declared Buffer Size (in Bytes)	Allocated Memory (in Bytes)
3,840	4,172
5,120	5,580
5,440	5,900
5,760	6,220
6,080	6,540
6,400	6,860

Where X represents allocated memory for a declared receive buffer size in bytes, the static memory required for ICS with only one line installed and only one terminal address reserved is as follows:

$$660 + X$$

Each additional terminal address (beyond one) reserved during installation requires 68 bytes of additional memory. So the static memory requirements for an ICS installation with only one line having a receive buffer size of X and Y number of terminal addresses can be computed as follows:

$$592 + X + (Y - 1) \times 68$$

For each additional installed ICS line, 68 additional bytes of memory are required plus the memory required for the receive buffer and terminal addresses for the additional line. For example, if your second ICS line has a receive buffer size of 3840 bytes (4172 bytes memory required) and three terminal addresses reserved, the total static memory required for it can be computed as follows:

$$76 + 4172 + (3 \times 68) = 4452$$

To determine the total static memory requirements for your particular installation, calculate the size of each additional line and the first installed line, and add them up.

When you activate an ICS line, ICS uses more memory. You can determine the dynamic memory requirements for ICS by referring to Table C-2. If you activate one ICS line, the amount of memory that the operating system allocates is equal to the nonreplicable requirements plus the replicatable requirements. When you activate a second ICS line, the operating system allocates only the replicatable requirements. The same is true for video stations, printers stations, and Programmed Station Control (PSC) Emulators. For example, when you activate an ICS line, the operating system allocates 6,000 plus 5,000 bytes of memory for a total of 11,000 bytes. When you activate a second ICS line, the operating system allocates only an additional 5,000 bytes for a total of 16,000 bytes. All of the figures in Table C-2 are rounded up to the nearest hundred bytes.

Table C-2. Dynamic Memory Requirements for ICS Tasks

ICS Task	Nonreplicable	Replicable
Controller (communications line)	6,000	5,000
Video station	20,000	7,500
Printer station	13,500	6,500
PSC Emulator	18,500	7,000

C.3 SUPPORTED HARDWARE

The supported non-TI modems are the Western Electric (Bell) modems (listed in Table C-5). The options recommended are listed with each modem. Table C-5 also lists those modem options required when the specified modem is used with ICS. The non-TI communications components (Table C-4) support the Western Electric (Bell) modems. Table C-3 lists the TI modems and communications interface kits that support ICS.

In recent years, the telephone company has begun to offer facilities that are capable of transmitting digital information directly without converting it to analog. For digital service, a modem is not used, but a similar device called a data service unit (DSU) provides the interface between the communications controller and the communications line. Digital data service is generally used only for high-speed information exchange between computers and is not yet available in all areas.

Digital lines offer a major advantage over conventional analog transmissions. Amplifiers placed approximately every mile along the line counteract signal degradation. In analog transmissions, noise and signal distortion are amplified. With digital transmissions, the signal is restored without any of the noise and distortion that may have occurred during transmission.

For ICS transmissions across the Digital Data System (DDS), DSUs using the RS-232C interface at 2400, 4800, or 9600 bits per second are used instead of modems. Consult your Bell representative to determine availability of service, the rates, and the specific DSU required.

Table C-3. TI Communications Interface Kits

Communications Interface	TI Modem (201C Equivalent)
CI401 Kit: 946104-0001	Synchronous Modem Kit: 945094-0003
CP503 Kit: ¹ 2261954-0001	Synchronous Modem Kit: 945094-0003 (one per channel)
CP501 Kit: ² 2303091-0001	Synchronous Modem Kit: 945094-0004
CP502 Kit: 2303091-0001	Synchronous Modem Kit: 945094-0003
Notes:	
¹ The revision level for the FCCC in this kit should be revision P or later.	
² The revision level for the BCAIM controller in this kit should be revision A or later.	

Table C-4. Non-TI Communications Interface Components

Communications Interface	Non-TI Modem Components
Communications Interface Module (CI401) Kit: 946104-0002	The kit comprises: CIM — 946105-0001 Cable — 946117-0001
Four-Channel Communications Controller (CP503) Kit: ¹ 2261954-0001	The FCCC requires: Cable — 946117-0001 or 0002 (one per channel)
Bit-Oriented, Character-Oriented, Asynchronous Interface Module (CP501) Kit: ² 2265184-0001	The kit comprises: BCAIM — 2265165-0001 Cable — 2303070-0002
X.21 Bit-Oriented, Character-Oriented, Asynchronous Interface Module: (CP502) 2303091-0002	The kit comprises: X.21 BCAIM — 2303085-0001 Cable — 2303070-0002
Notes:	
¹ The revision level for the FCCC should be revision P or later.	
² The revision level for the BCAIM should be revision A or later.	

Table C-5. Supported Non-TI Modems

Western Electric (Bell) Modem	Recommended Options	Required Options
Model 201B leased-line operation 2400 bps	None	Without new sync, EIA option Internal timing
Model 201C leased-line 2400 bps	Grounding option (AB not connected to AA)	Without new sync, Internal timing, EIA option
Model 2024A DATAPHONE II leased-line operation, 2400 bps	None	Internal timing
Model 208A leased-line operation, 4800 bps	Grounding option (AB not connected TO AA)	Internal timing
Model 2048C DATAPHONE II* leased-line operation, 4800 bps	None	Internal timing
Model 209A multiplexed, leased-line operation, 2400, 4800, 7200, or 9600 bps	Grounding option (AB not connected to AA) Elastic store out	Internal timing
Model 2096A DATAPHONE II multiplexed leased-line operation, 2400, 4800, or 9600 bps	None	Internal timing

Note:

* The C indicates quick start-up and is recommended for use with ICS lines.

Appendix D

ASCII/EBCDIC, EBCDIC/ASCII Translation Tables

The IBM Binary Synchronous Communications (BSC) procedures employed by ICS require that all data be translated and transmitted as EBCDIC-encoded data. Those EBCDIC characters less than >40 for which ASCII has no equivalent are translated to null characters. Those EBCDIC characters greater than or equal to > 40 for which ASCII has no equivalent are translated to space characters.

The tables included in this appendix contain the following information:

- ASCII-to-EBCDIC translations; entire ASCII character set (Table D-1).
- EBCDIC-to-ASCII translations; entire EBCDIC character set (Table D-2).

Table D-1. ASCII-to-EBCDIC Translation

ASCII VALUE	ASCII CHAR	EBCDIC VALUE	EBCDIC CHAR	ASCII VALUE	ASCII CHAR	EBCDIC VALUE	EBCDIC CHAR
00	NUL	00	NUL	20	SPACE	40	SPACE
01	SOH	01	SOH	21	!	5A	!
02	STX	02	STX	22	”	7F	”
03	ETX	03	ETX	23	#	7B	#
04	EOT	37	EOT	24	\$	5B	\$
05	ENQ	2D	ENQ	25	%	6C	%
06	ACK	2E	ACK	26	&	50	&
07	BEL	2F	BEL	27	'	7D	'
08	BS	16	BS	28	(4D	(
09	HT	05	HT	29)	5D)
0A	LF	25	LF	2A	*	5C	*
0B	VT	0B	VT	2B	+	4E	+
0C	FF	0C	FF	2C	,	6B	,
0D	CR	0D	CR	2D	-	60	-
0E	SO	0E	SO	2E	.	4B	.
0F	SI	0F	SI	2F	/	61	/
10	DLE	10	DLE	30	0	F0	0
11	DC1	11	DC1	31	1	F1	1
12	DC2	12	DC2	32	2	F2	2
13	DC3	13	TM	33	3	F3	3
14	DC4	3C	DC4	34	4	F4	4
15	NAK	3D	NAK	35	5	F5	5
16	SYN	32	SYN	36	6	F6	6
17	ETB	26	ETB	37	7	F7	7
18	CAN	18	CAN	38	8	F8	8
19	EM	19	EM	39	9	F9	9
1A	SUB	3F	SUB	3A	:	7A	:
1B	ESC	27	ESC	3B	;	5E	;
1C	FS	1E	IRS	3C	<	4C	<
1D	GS	1C	IFS	3D	=	7E	=
1E	RS	1E	IRS	3E	>	6E	>
1F	US	1F	IUS	3F	?	6F	?

Table D-1 ASCII-to-EBCDIC Translation (Continued)

ASCII VALUE	ASCII CHAR	EBCDIC VALUE	EBCDIC CHAR	ASCII VALUE	ASCII CHAR	EBCDIC VALUE	EBCDIC CHAR
00	NUL	00	NUL	20	SPACE	40	SPACE
01	SOH	01	SOH	21	!	5A	!
02	STX	02	STX	22	"	7F	"
03	ETX	03	ETX	23	#	7B	#
04	EOT	37	EOT	24	\$	5B	\$
05	ENQ	2D	ENQ	25	%	6C	%
06	ACK	2E	ACK	26	&	50	&
07	BEL	2F	BEL	27	'	7D	'
08	BS	16	BS	28	(4D	(
09	HT	05	HT	29)	5D)
0A	LF	25	LF	2A	*	5C	*
0B	VT	0B	VT	2B	+	4E	+
0C	FF	0C	FF	2C	,	6B	,
0D	CR	0D	CR	2D	-	60	-
0E	SO	0E	SO	2E	.	4B	.
0F	SI	0F	SI	2F	/	61	/
10	DLE	10	DLE	30	0	F0	0
11	DC1	11	DC1	31	1	F1	1
12	DC2	12	DC2	32	2	F2	2
13	DC3	13	TM	33	3	F3	3
14	DC4	3C	DC4	34	4	F4	4
15	NAK	3D	NAK	35	5	F5	5
16	SYN	32	SYN	36	6	F6	6
17	ETB	26	ETB	37	7	F7	7
18	CAN	18	CAN	38	8	F8	8
19	EM	19	EM	39	9	F9	9
1A	SUB	3F	SUB	3A	:	7A	:
1B	ESC	27	ESC	3B	;	5E	;
1C	FS	1C	IFS	3C	<	4C	<
1D	GS	1D	IGS	3D	=	7E	=
1E	RS	1E	IRS	3E	>	6E	>
1F	US	1F	IUS	3F	?	6F	?

Table D-2. EBCDIC-to-ASCII Translation

EBCDIC VALUE	EBCDIC CHAR	ASCII VALUE	ASCII CHAR	EBCDIC VALUE	EBCDIC CHAR	ASCII VALUE	ASCII CHAR	EBCDIC VALUE	EBCDIC CHAR	ASCII VALUE	ASCII CHAR
00	NUL	00	NUL	20	DS	00	NUL	40	SPACE	00	NUL
01	SOH	01	SOH	21	SOS	00	NUL	41		00	NUL
02	STX	02	STX	22	FS	00	NUL	42		00	NUL
03	ETX	03	ETX	23		00	NUL	43		00	NUL
04	PF	00	NUL	24	BYP	00	NUL	44		00	NUL
05	HT	09	HT	25	LF	0A	LF	45		00	NUL
06	LC	00	NUL	26	ETB	17	ETB	46		00	NUL
07	DEL	7F	DEL	27	ESC	00	NUL	47		00	NUL
08		00	NUL	28		00	NUL	48		00	NUL
09	RLF	00	NUL	29		00	NUL	49		00	NUL
0A	SMM	00	NUL	2A	SM	00	NUL	4A	Ç	5B	[
0B	VT	0B	VT	2B	CU2	00	NUL	4B	.	2E	.
0C	FF	0C	FF	2C		00	NUL	4C	<	3C	<
0D	CR	0D	CR	2D	ENQ	05	ENQ	4D	(28	(
0E	SO	0E	SO	2E	ACK	06	ACK	4E	+	2B	+
0F	SI	0F	SI	2F	BEL	07	BEL	4F		7C	
10	DLE	10	DLE	30		00	NUL	50	&	26	&
11	DC1	11	DC1	31		00	NUL	51		00	NUL
12	DC2	12	DC2	32	SYN	16	SYN	52		00	NUL
13	DC3	13	DC3	33		00	NUL	53		00	NUL
14	RES	00	NUL	34	PN	00	NUL	54		00	NUL
15	NL	15	NAK	35	RS	00	NUL	55		00	NUL
16	BS	08	BS	36	UC	00	NUL	56		00	NUL
17	IL	00	NUL	37	EOT	04	EOT	57		00	NUL
18	CAN	18	CAN	38		00	NUL	58		00	NUL
19	EM	19	EM	39		00	NUL	59		00	NUL
1A	CC	00	NUL	3A		00	NUL	5A	!	21	!
1B	CU1	00	NUL	3B	CU3	00	NUL	5B	\$	24	\$
1C	IFS	ID	GS	3C	DC4	14	DC4	5C	*	2A	*
1D	IGS	ID	GS	3D	NAK	15	NAK	5D)	29)
1E	IRS	IC	FS	3E		00	NUL	5E	;	3B	;
1F	IUS	1F	US	3F	SUB	1A	SUB	5F	¬	5E	^

Table D-2 EBCDIC-to-ASCII Translation (Continued)

EBCDIC VALUE	EBCDIC CHAR	ASCII VALUE	ASCII CHAR	EBCDIC VALUE	EBCDIC CHAR	ASCII VALUE	ASCII CHAR	EBCDIC VALUE	EBCDIC CHAR	ASCII VALUE	ASCII CHAR
60	-	2D	-	80		00	NUL	A0		00	NUL
61	/	2F	/	81	a	61	a	A1	~	7E	~
62		00	NUL	82	b	62	b	A2	s	73	s
63		00	NUL	83	c	63	c	A3	t	74	t
64		00	NUL	84	d	64	d	A4	u	75	u
65		00	NUL	85	e	65	e	A5	v	76	v
66		00	NUL	86	f	66	f	A6	w	77	w
67		00	NUL	87	g	67	g	A7	x	78	x
68		00	NUL	88	h	68	h	A8	y	79	y
69		00	NUL	89	i	69	i	A9	z	7A	z
6A		7C		8A		00	NUL	AA		00	NUL
6B	,	2C	,	8B		00	NUL	AB		00	NUL
6C	%	25	%	8C		00	NUL	AC		00	NUL
6D	-	5F	-	8D		00	NUL	AD		00	NUL
6E	>	3E	>	8E		00	NUL	AE		00	NUL
6F	?	3F	?	8F		00	NUL	AF		00	NUL
70		00	NUL	90		00	NUL	B0		00	NUL
71		00	NUL	91	j	6A	j	B1		00	NUL
72		00	NUL	92	k	6B	k	B2		00	NUL
73		00	NUL	93	l	6C	l	B3		00	NUL
74		00	NUL	94	m	6D	m	B4		00	NUL
75		00	NUL	95	n	6E	n	B5		00	NUL
76		00	NUL	96	o	6F	o	B6		00	NUL
77		00	NUL	97	p	70	p	B7		00	NUL
78		00	NUL	98	q	71	q	B8		00	NUL
79	\	60	\	99	r	72	r	B9		00	NUL
7A	:	3A	:	9A		00	NUL	BA		00	NUL
7B	#	23	#	9B		00	NUL	BB		00	NUL
7C	@	40	@	9C		00	NUL	BC		00	NUL
7D	'	27	'	9D		00	NUL	BD		00	NUL
7E	=	3D	=	9E		00	NUL	BE		00	NUL
7F	”	22	”	9F		00	NUL	BF		00	NUL

Table D-2 EBCDIC-to-ASCII Translation (Continued)

EBCDIC VALUE	EBCDIC CHAR	ASCII VALUE	ASCII CHAR	EBCDIC VALUE	EBCDIC CHAR	ASCII VALUE	ASCII CHAR
C0	}	7B	}	E0	\	5C	\
C1	A	41	A	E1		00	NUL
C2	B	42	B	E2	S	53	S
C3	C	43	C	E3	T	54	T
C4	D	44	D	E4	U	55	U
C5	E	45	E	E5	V	56	V
C6	F	46	F	E6	W	57	W
C7	G	47	G	E7	X	58	X
C8	H	48	H	E8	Y	59	Y
C9	I	49	I	E9	Z	5A	Z
CA		00	NUL	EA		00	NUL
CB		00	NUL	EB		00	NUL
CC	⌈	00	NUL	EC	⌈	00	NUL
CD		00	NUL	ED		00	NUL
CE	⌋	00	NUL	EE		00	NUL
CF		00	NUL	EF		00	NUL
D0	}	7D	}	F0	0	30	0
D1	J	4A	J	F1	1	31	1
D2	K	4B	K	F2	2	32	2
D3	L	4C	L	F3	3	33	3
D4	M	4D	M	F4	4	34	4
D5	N	4E	N	F5	5	35	5
D6	O	4F	O	F6	6	36	6
D7	P	50	P	F7	7	37	7
D8	Q	51	Q	F8	8	38	8
D9	R	52	R	F9	9	39	9
DA		00	NUL	FA		00	NUL
DB		00	NUL	FB		00	NUL
DC		00	NUL	FC		00	NUL
DD		00	NUL	FD		00	NUL
DE		00	NUL	FE		00	NUL
DF		00	NUL	FF		00	NUL

Appendix E

Summary of ICS Commands and Key Functions

This appendix provides a quick-reference summary of all of the System Command Interpreter (SCI) command procedures used by ICS and the special key functions of ICS. Table E-1 lists the SCI procedures in alphabetical order along with the paragraphs that detail the operation of the command. The key functions are listed twice. Table E-2 is alphabetized according to the function description. Tables E-3 through E-5 are alphabetized according to the keys you enter. Table E-3 lists the key mapping for the 911 and 915 terminals. Table E-4 lists the key mapping for the 940 EVT, and Table E-5 lists the key mapping for the Business System terminal. Table E-6 lists the key mapping for the 931 VDT.

For quick reference when using the DNOS operating system, ICS provides a menu that lists the commonly used commands. To display this menu at a video terminal, enter the following characters in response to the SCI prompt:

/ICS

Table E-1. ICS SCI Commands

Command	Command Name	Description
LICC	List Interactive Communications Configuration	Displays the current installed configuration for a specific line. (See paragraph 4.3.1.)
LICS	List Interactive Communications Statistics	Displays statistical information about an individual ICS line. (See paragraph 4.4.)
MCUA	Modify Control Unit Address	Changes the control unit address of an inactive ICS line. (See paragraph 4.3.3.)
MICC	Modify Interactive Communications Configurations	Changes the state (enabled or disabled) and the type (CRT or PRT) of a particular terminal address. (See paragraph 4.3.2.)
PTPSC	Patch ICS PSC User Task	Applies the latest patches to the PSC Emulator routines linked to the user-written PSC task. (See paragraph 5.6.)
QICT	Quit Interactive Communications Terminal	Terminates a particular ICS display or printer station. (See paragraph 3.3.2.1.)

Table E-1. ICS SCI Commands (Continued)

Command	Command Name	Description
XICC	Execute Interactive Communications Controller	Activates an ICS communications line. (See paragraph 4.2.2.)
XICP	Execute Interactive Communications Printer	Activates an ICS printer station. (See paragraph 3.2.2.)
XICT	Execute Interactive Communications Terminal	Activates an ICS video station. (See paragraph 3.2.1.)
XPA	Execute Poll Analyzer	Provides a listing of ICS buffer activity for the systems analyst. (See paragraph 4.5.)
XPSC	Execute Programmed Station Control	Activates a user-written program that acts in place of an ICS station. (See paragraph 5.7.)

Table E-2. Key Functions of ICS — Listed by Function

Key Function:	Generic Key Sequence(s):
Alpha (overrides numeric protection)	Exit + A
Append cursor (at end of a word)	Exit + P
Cancel function	Exit + Exit
Clear screen	Initialize Input ¹
Duplicate, remote (duplicate previous field)	Exit + D
Duplicate, local (duplicate previous field)	Exit + L
Erase to end-of-screen (EOS)	Exit + E ²
Field mark (identifies the field as modified)	Exit + F
Home cursor (cursor to the first character position)	HOME

Table E-2. Key Functions of ICS — Listed by Function (Continued)

Key Function:	Generic Key Sequence(s):
Identification function (identity or change the print key pathname)	Exit + I
Print screen	PRINT
Print screen cancel	PRINT + PRINT
Program access key one, PA1	Exit + + (plus) ³
Program access key two, PAS2	Exit + - (hyphen) ⁴
Program access key three, PAS3	Exit + _ (underscore) ⁵
Program function keys, PF1 through PF24	Exit + 01 through Exit + 24 ⁶
Repeat (extension of the typamatic feature)	None—Data and cursor control keys are typamatic ⁷
Reset (unlock keyboard)	Exit + R ⁸
Suspend ICS station	Command
Tab mode toggle (TABS and WORD)	Exit + M
Terminate ICS station	Exit + Q

Notes:¹ 940 EVT key sequence: ERASE ALL² 940 EVT key sequence: ERASE EOS³ 940 EVT key sequence: (ALT)P1⁴ 940 EVT key sequence: (ALT)P2⁵ 940 EVT key sequence: (ALT)P3⁶ 940 EVT key sequence: F1 to F24⁷ 911 and 915 have REPEAT keys⁸ 940 EVT key sequence: (ALT)

Table E-3. Key Functions of 911 and 915 ICS Terminals — Listed by Key

Key Sequence(s) for 911 and 915 Terminals	Key Function
Initialize Input	Clear screen
Command	Suspend 911 terminal
Enter	Transmit modified fields to host
Exit + A	Alpha (override numeric protection)
Exit + D	Duplicate, remote (duplicate previous field)
Exit + E	Erase to end-of-screen (EOS)
Exit + F	Field mark
Exit + I	Identification function (identify or change the print key pathname)
Exit + L	Duplicate, local (duplicate previous field)
Exit + M	Tab mode toggle (between TABS and WORD)
Exit + P	Append cursor to the end of a word
Exit + Q	Terminate station
Exit + R	Reset (unlock keyboard)
Exit + + (plus)	Program access key one, PA1
Exit + - (hyphen)	Program access key two, PA2
Exit + _ (underscore)	Program access key three, PA3
Exit + 01 to EXIT + 24	Program function keys, PF1 through PF24
F9 (915 keyboard)	Suspend 915 terminal
HOME	Home cursor (to the first character position)
PRINT	Print screen
PRINT + PRINT	Cancel screen print
REPEAT	Repeat (extension of the typamatic feature)

Table E-4. Key Functions of 931 ICS Terminals — Listed by Key

Key Sequence(s) For 931 Terminal	Key Function
Initialize Input	Clear screen
Command	Suspend 931 terminal
Exit + A	Alpha (override numeric protection)
Exit + D	Duplicate, remote (duplicate previous field)
Exit + E	Erase to end-of-screen (EOS)
Exit + F	Field mark
Exit + I	Identification function (identify or change the print key pathname)
Exit + L	Duplicate, local (duplicate previous field)
Exit + M	Tab mode toggle (between TABS and WORD)
Exit + P	Append cursor to the end of a word
Exit + Q	Terminate station
Exit + R	Reset (unlock keyboard)
Exit + + (plus)	Program access key one, PA1
Exit + - (hyphen)	Program access key two, PA2
Exit + _ (underscore)	Program access key three, PA3
F1 to F12	Program function keys, PF1 to PF12
HOME	Home cursor (to the first character position)
PRINT	Print screen
PRINT + PRINT	Cancel screen print
REPEAT	No repeat key, keyboard is typamatic

Table E-5. Key Functions of 940 ICS Terminal — Listed by Key

Key Sequence(s) for 940 EVT	Key Function
(ALT)P1	Program access key one, PA1
(ALT)P2	Program access key two, PA2
(ALT)P3	Program access key three, PA3
(ALT)\	Reset (unlock keyboard)
ERASE ALL	Clear screen
ERASE EOF	Erase field (entire field)
ERASE EOL	Skip (erase to end-of-field)
ERASE EOS or PAGE + E	Erase to end-of-screen (EOS)
ERASE INPUT	Erase input
F1 through F24	Program function keys, PF1 through PF24
HOME	Home cursor (to the first character position)
LINEFEED	Field right (cursor to next field)
NEXT FORM	Suspend 940 terminal
Exit + A	Alpha (override numeric protection)
Exit + D	Duplicate, remote (duplicate previous field)
Exit + F	Field mark
Exit + I	Identification function (identify or change the print key pathname)
Exit + L	Duplicate, local (duplicate previous field)
Exit + M	Tab mode toggle (between TABS and WORD)
Exit + P	Append cursor to the end of a word

Table E-5. Key Functions of 940 ICS Terminal — Listed by Key (Continued)

Key Sequence(s) for 940 EVT	Key Function
Exit + Q	Terminate station
PRINT	Print screen
PRINT + PRINT	Cancel print
SEND	Enter (send to host)
SKIP	Cursor to the previous field

Appendix F

Binary Synchronous Communications (BSC) Procedures

ICS uses the IBM 3270 Binary Synchronous Communications (BSC) protocol. This protocol uses the multipoint data link mode of operation over nonswitched facilities. ICS cannot be attached to a line that supports other types of procedures.

The data transmitted from the ICS control unit is sent over communications lines in blocks of data. Each block can have a maximum of 256 characters. The host application program is not limited by the 256-character maximum. Every 256-character block includes a start-of-text (STX) character and an end-of-transmission block (ETB) or an end-of-text (ETX) character. Block-check characters (BCCs) follow each block. These characters provide an error check to ensure that all characters transmitted are received correctly.

Included in each transmission are data link control characters. The characters or character combinations indicate the beginning and end of each message block, format the block, and respond to message blocks. The data link control characters cause the data transmission to proceed in an orderly fashion.

PAD characters precede and follow each transmission. A PAD character ensures that the last character of the transmission block is properly sent prior to turning off the transmitting station. The PAD character is binary 11111111 (> FF).

This appendix briefly describes the BSC control characters that the ICS control unit supports. For a detailed description and explanation of the 3270 BSC procedures, refer to the *IBM 3270 Information Display System Component Description* and the *IBM General Information — Binary Synchronous Communications*.

ICS supports the following data link control characters:

- Synchronous idle (SYN)
- Start of heading (SOH)
- Start of text (STX)
- End of intermediate transmission block (ITB)
- End of transmission block (ETB)
- End of text (ETX)
- End of transmission (EOT)
- Enquiry (ENQ)

- Even positive acknowledgment (ACK0)
- Odd positive acknowledgment (ACK1)
- Negative acknowledgment (NAK)
- Wait before transmit (WACK)
- Data link escape (DLE)
- Reverse interrupt (RVI)
- Escape (ESC)
- Temporary text delay (TTD)

Synchronous idle (SYN). The SYN character establishes and maintains synchronization of transmission. The synchronization pattern consists of two or more SYN characters that precede any control characters or data transmitted.

Start of heading (SOH). ICS transmits the SOH character in a three-character heading that identifies the accompanying data as a status.

Start of text (STX). The STX character precedes a block of text to place the receiving station in the receive mode. STX begins the accumulation of the BCC value that is to be appended to this text block. STX is also the first character of a TTD sequence.

End of intermediate transmission block (ITB). The ICS control unit does not transmit the ITB character, but does support the reception of ITB. When a block of text containing ITB is received, ITB is removed and the BCC accumulation continues with the value of the BCC character following ITB.

End of transmission block (ETB). The ETB character indicates the end of a block of data that was preceded by STX or SOH and indicates that the BCC for the block follows immediately. The ETB character forces a line turnaround, and the receiving station transmits its response.

End of text (ETX). The ETX character indicates the end of the last block of data of a message that was preceded by STX or SOH. A BCC for the block follows immediately. ETX forces a line turnaround, and the receiving station transmits an acknowledgment.

End of transmission (EOT). The ICS control unit transmits an EOT character in the following situations:

- When the ICS control unit cannot perform a request that the host transmits
- In response to a completed read operation by the host
- When the host polls, and the ICS control unit has no data to transmit
- In response to a received RVI

When ICS receives an EOT character, it returns to the state in which it is awaiting a poll or select.

Enquiry (ENQ). The ICS control unit transmits an ENQ character in the following situations:

- To request a reply from the IBM transmission control unit after a three-second time-out following an ICS transmission
- To request retransmission of the previous reply
- As the last character of a text message in which a parity error was detected

When ICS receives an ENQ character, the last transmission is repeated.

To be addressed successfully, the ICS control unit must receive ENQ as the last character of a poll or select sequence.

The ENQ character is transmitted with a leading PAD and two SYN characters and followed by a trailing PAD character.

Alternating acknowledgment (ACK0/1). BSC uses the following two alternating acknowledgment characters:

- The ACK0 character is an affirmative acknowledgment for line bits and even-numbered text blocks. ICS also transmits ACK0 after a successful select-addressing sequence to indicate that it is ready to accept transmission.
- The ACK1 character is an affirmative acknowledgment for odd-numbered text blocks.

The ACK0 and ACK1 characters are transmitted with leading and trailing PAD characters in the same way as an ENQ character.

Negative acknowledgment (NAK). ICS transmits the NAK character when it receives text that satisfies one of the following conditions:

- Terminates with an ENQ character
- Has an embedded ENQ character
- Has a BCC character with an invalid value
- Contains a TTD sequence (STX ENQ)
- Has an ETX character missing

When ICS receives a NAK character in response to a data transmission, ICS resends the last block of data.

Wait before transmitting, positive acknowledgment (WACK). ICS transmits the WACK character when it is unable to receive data on a select. ICS responds to a WACK character with an ENQ character.

Data link escape (DLE). The DLE character provides supplementary line control characters such as WACK, ACK0/1, and RVI. The DLE character enables the transmission of control characters in transparent text. The ICS control unit does not operate in transparent mode, but it can operate on a communications line with stations that use the transparent mode.

Reverse interrupt (RVI). ICS transmits the RVI character when it receives a select while it has a pending status or sense message to send. ICS transmits an EOT character and resets all pending status and sense information whenever it receives an RVI character.

Escape (ESC). The ESC character precedes the command code in each command sequence data stream that ICS receives.

Temporary text delay (TTD). ICS does not transmit a TTD sequence, but it responds to TTD by transmitting NAK and then awaiting EOT.

Line error detection. ICS detects transmission errors by using a cyclic redundancy check (CRC) that it performs on the received data. The transmitting station makes CRCs possible by generating a value for the block check character (BCC) that it appends to the end of each message block transmitted (after an ITB, ETB, or ETX character). The value of the BCC (16 bits) is generated by a cyclic process in which each data bit is manipulated by a preset check polynomial. This polynomial checks that the data transmitted is exactly the same as that received. All characters, except SYN following an STX or SOH character, are included in the CRC. The CRC value also counts the ETB and ETX characters.

If ICS detects an error during transmission of a data block, it responds with a NAK character. If ICS detects an error when it is expecting a control function, it responds with an ENQ character.

Alphabetical Index

Introduction

HOW TO USE INDEX

The index, table of contents, list of illustrations, and list of tables are used in conjunction to obtain the location of the desired subject. Once the subject or topic has been located in the index, use the appropriate paragraph number, figure number, or table number to obtain the corresponding page number from the table of contents, list of illustrations, or list of tables.

INDEX ENTRIES

The following index lists key words and concepts from the subject material of the manual together with the area(s) in the manual that supply major coverage of the listed concept. The numbers along the right side of the listing reference the following manual areas:

- Sections — Reference to Sections of the manual appear as “Sections x” with the symbol x representing any numeric quantity.
- Appendixes — Reference to Appendixes of the manual appear as “Appendix y” with the symbol y representing any capital letter.
- Paragraphs — Reference to paragraphs of the manual appear as a series of alphanumeric or numeric characters punctuated with decimal points. Only the first character of the string may be a letter; all subsequent characters are numbers. The first character refers to the section or appendix of the manual in which the paragraph may be found.
- Tables — References to tables in the manual are represented by the capital letter T followed immediately by another alphanumeric character (representing the section or appendix of the manual containing the table). The second character is followed by a dash (-) and a number.

Tx-yy

- Figures — References to figures in the manual are represented by the capital letter F followed immediately by another alphanumeric character (representing the section or appendix of the manual containing the figure). The second character is followed by a dash (-) and a number.

Fx-yy

- Other entries in the Index — References to other entries in the index preceded by the word “See” followed by the referenced entry.

- Abort Count 4.4.4
- Access Key, Program 3.4.4.13
- Activate:
 - Communications Controller Task ... 4.2.3
 - Option 5.3.1
- Activation:
 - Printer 3.2.2
 - Video Terminal 3.2.1
- Active:
 - State 4.3.1
 - PSC 4.3.1
- Address:
 - Auto-Select:
 - Control Unit 5.2.4.1, 5.7
 - Terminal 2.5.2.4, 5.2.4.2, 5.7
 - Control Unit 2.3.1, 4.3.1
 - Mapping:
 - Decimal to EBCDIC 2.3
 - EBCDIC to Decimal 2.3
 - Poll 4.3.1
 - Terminal 2.3.2, 2.5.2.4, 3.2.1.2, 3.2.2.2, 3.3.2.1, 4.3.1, 5.2.4.2
- Add-On Package 5.5
- AID Code 3.4.4.13, 5.3.5.2
- Alarm, Audible 3.4.3.2, 5.2.3
- Alpha Function 3.4.4.12
- Analyzer, Execute Poll 4.5
- Application:
 - Command, Host 5.2.3
 - Order, Host 5.2.3
- Arguments:
 - Close 5.3.7.2
 - Fill 5.3.2.2
 - Get 5.3.3.2
 - Retrieve Attributes 5.3.4.2
 - Time-Out 5.3.6.2
 - Transmit 5.3.5.2
- ASCII/EBCDIC Translation Appendix D
- Attention ID Code 5.3.5.2
- Attribute Character 3.4.2.1, 5.2.2
- Attributes:
 - Arguments, Retrieve 5.3.4.2
 - Call, Retrieve 5.3.4.1
 - Example, Retrieve 5.3.4.4
 - Routine, Retrieve 5.3.4
- Audible Alarm 3.4.3.2, 5.2.3
- Automatic Skip 3.4.3.1
- Auto-Select:
 - Communications Line
 - Name 2.5.2.4, 5.2.4.1, 5.7
 - Control Unit Address 5.2.4.1, 5.7
 - Terminal Address 2.5.2.4, 5.2.4.2, 5.7
- Binary Synchronous Communications
 - Procedures Appendix F
- Blinking Cursor 3.4.2.1, 3.4.4.8, 3.4.4.8
- Block Count 4.4.2
- Buffer:
 - Error 4.4.9
 - PSC Station 5.2.1
- Business System Terminal 2.2.1
- Call:
 - Close 5.3.7.1
 - Example 5.4.5
 - Fill 5.3.2.1
 - Get 5.3.3.1
 - Open 5.3.1.1
 - Retrieve Attributes 5.3.4.1
 - Time-Out 5.3.6.1
 - Transmit 5.3.5.1
- Cancel Print 3.4.4.8
- CHAR Key, INS 3.4.4.5
- Character:
 - Attribute 3.4.2.1, 5.2.2
 - Write Control 5.2.3
- Clear Function Key 3.4.4.13
- Close:
 - Arguments 5.3.7.2
 - Call 5.3.7.1
 - Example 5.3.7.4
 - Routine 5.3.7
- Code:
 - AID 3.4.4.13, 5.3.5.2
 - Attention ID 5.3.5.2
 - Completion 5.2.5
- Command:
 - Erase All Unprotected 5.2.3
 - Erase/Write 5.2.3
 - Host Application 5.2.3
 - Interpreter, System 2.5.1
 - Read Modified 5.2.3
 - Reject 4.4.5
 - Summary, SCI 2.6, Appendix E
 - Write 5.2.3
- Communications:
 - Controller Appendix C
 - Controller Task, Activate 4.2.3
 - Line:
 - Name 2.5.2.4, 3.2.2.1, 3.3.2.1
 - Name, Auto-Select .. 2.5.2.4, 5.2.4.1, 5.7
 - Procedures, Binary
 - Synchronous Appendix F
 - Completion Code 5.2.5
 - Components, Emulator 1.4
 - Conditions, Printer Error 4.8.4
 - Configuration:
 - List Interactive Communications ... 4.3.1
 - Modify Interactive
 - Communications 4.3.2
 - Control Character, Write 5.2.3
 - Control Unit Address 2.3.1, 4.3.1
 - Auto-Select 5.2.4.1, 5.7
 - Controller:
 - Communications Appendix C
 - Remote Terminal 2.2, 2.2.2.1
- Conventions 1.3
- Count:
 - Abort 4.4.4
 - Block 4.4.2
 - Message 4.4.1
 - Retransmission 4.4.3
- Cursor 3.4.2.3

- Blinking 3.4.2.1, 3.4.4.8, 3.4.4.8
- Movement Key 3.4.4.2
- Wrap 3.4.4.2
- Data Services Unit 2.4.1.1
- Decimal to EBCDIC Address Mapping ... 2.3
- Delete Character Function 3.4.4.5
- Development Aid, Program 5.4
- Device Name 2.5.2.1
- Digital Services Unit Appendix C
- Disabled State 4.3.1
- Display, Format 3.4.2.1
- DSU 2.4.1.1, Appendix C
- DUMY 2.5.2.4
- DUP 3.4.4.10
- Duplicate:
 - Function 3.4.4.10
 - Local 3.4.4.10
 - Remote 3.4.4.10
- EBCDIC to Decimal Address Mapping ... 2.3
- EBCDIC/ASCII Translation Appendix D
- Edit Key 3.4.4.5
- EM Order 4.8.3, 5.2.3
- Emulator Components 1.4
- End of Message Order 4.8.3, 5.2.3
- End-of-Screen, Erase To 3.4.4.5
- Enter:
 - Function 3.4.4.13
 - Key 3.4.4.13
- EOS Function, Erase 3.4.4.5
- Erase:
 - All Unprotected Command 5.2.3
 - EOS Function 3.4.4.5
 - To End-of-Screen 3.4.4.5
- Erase-Field Function 3.4.4.5
- Erase-Input Function 3.4.4.5
- Erase/Write Command 5.2.3
- Error:
 - Buffer 4.4.9
 - Conditions, Printer 4.8.4
 - Messages Appendix B
 - PSC Appendix B
 - Poll 4.4.7
 - Printer Station 3.2.2.4
 - Select 4.4.8
- Example:
 - Call 5.4.5
 - Close 5.3.7.4
 - Fill 5.3.2.4
 - Get 5.3.3.4
 - Open 5.3.1.4
 - Retrieve Attributes 5.3.4.4
 - Time-Out 5.3.6.4
 - Transmit 5.3.5.4
- Execute:
 - Poll Analyzer 4.5
 - PSC Task 5.7
- Extended:
 - Features 1.7
 - Function Mode 3.4.2.3
 - Key Function 3.4.1
- Features:
 - Extended 1.7
 - Supported 1.6
 - Unsupported 1.8
- FF Order 4.8.3, 5.2.3
- Field Movement Key 3.4.4.3
- Field-Mark Function 3.4.4.11
- File:
 - Pathname 2.5.2.2
 - Program 5.5, 5.6
- Fill:
 - Arguments 5.3.2.2
 - Call 5.3.2.1
 - Example 5.3.2.4
 - Routine 5.3.2
- Form Feed Order 4.8.3, 5.2.3
- Format Display 3.4.2.1
- Format File, Local 4.6, 5.2.4.1
- Format, Print Line 4.8.3, 5.2.3
- Function:
 - Alpha 3.4.4.12
 - Delete Character 3.4.4.5
 - Duplicate 3.4.4.10
 - Enter 3.4.4.13
 - Erase EOS 3.4.4.5
 - Erase-Field 3.4.4.5
 - Erase-Input 3.4.4.5
 - Field-Mark 3.4.4.11
 - Identification 3.4.4.8
 - Insert-Character 3.4.4.5
- Key:
 - Clear 3.4.4.13
 - Program 3.4.4.13
- Local Duplicate 3.4.4.10
- Mode, Extended 3.4.2.3
- Print 3.4.4.8
- Program-Attention 3.4.4.13
- Remote Duplicate 3.4.4.10
- Get:
 - Arguments 5.3.3.2
 - Call 5.3.3.1
 - Example 5.3.3.4
 - Routine 5.3.3
- Host:
 - Application:
 - Command 5.2.3
 - Order 5.2.3
- ID:
 - Code, Attention 5.3.5.2
 - Task 5.6, 5.7
 - Run 4.3.1
- Ident 3.4.4.8
- Identification:
 - Function 3.4.4.8
 - PSC Station 5.2.4
- Idle State 4.3.1
- Initial Value 2.5.2.4
- Initialization ICS 4.2.2
- Initialize System 4.2.3

- Patch PSC Programs 5.6
- Pathname:
 - File 2.5.2.2
 - Output 3.2.2.2
 - Print Key 2.5.2.4, 3.2.1.3, 3.4.4.8
 - Prompt, Output 2.5.2.4
 - Terminal 3.3.2.1
- Poll 1.2
 - Address 4.3.1
 - Analyzer, Execute 4.5
 - Error 4.4.7
- Primitive, SCI 3.3.1, 3.3.2.1, 4.2.3
- Print:
 - Cancel 3.4.4.8
 - Function 3.4.4.8
 - Key 3.4.4.8
 - Pathname 2.5.2.4, 3.2.1.3, 3.4.4.8
 - Line:
 - Format 4.8.3, 5.2.3
 - Order 5.2.3
- Printer:
 - Activation 3.2.2
 - Error Conditions 4.8.4
 - Order, Start 5.2.3
 - Station Error 3.2.2.4
- Printer Station Time-Out 4.8.4
- Printers, Supported 1.4.2.2
- Priority Level, PSC 5.5
- Procedures, Binary Synchronous Communications Appendix F
- Program:
 - Access Key 3.4.4.13
 - Development Aid 5.4
 - File 5.5, 5.6
 - Function Key 3.4.4.13
- Program-Attention Function 3.4.4.13
- Prompt, Output Pathname 2.5.2.4
- PSC:
 - Active State 4.3.1
 - Error Messages Appendix B
 - Link 5.5
 - Priority Level 5.5
 - Programs, Patch 5.6
 - Station:
 - Buffer 5.2.1
 - Identification 5.2.4
 - Task, Execute 5.7
 - Waiting State 4.3.1
- PTPSC 5.6

- Read Modified Command 5.2.3
- Reject, Command 4.4.5
- Remote:
 - Duplicate Function 3.4.4.10
 - Terminal:
 - Controller 2.2, 2.2.2.1
 - Subsystem 2.2, 2.2.2
 - Video Terminal 2.2.1
 - Repeat Key 3.4.4.1
- Requirements, Memory Appendix C

- Reset 3.4.4.9
 - Keyboard 5.2.3
 - Modified 5.2.3
- Retransmission Count 4.4.3
- Retrieve Attributes:
 - Arguments 5.3.4.2
 - Call 5.3.4.1
 - Example 5.3.4.4
 - Routine 5.3.4
- Return Key 3.4.4.3
- Routine:
 - Close 5.3.7
 - Fill 5.3.2
 - Get 5.3.3
 - Open 5.3.1
 - Retrieve Attributes 5.3.4
 - Time-Out 5.3.6
 - Transmit 5.3.5
- RTC 2.2, 2.2.2.1
- RTS 2.2, 2.2.2
- Run ID, Task 4.3.1
- RVT, Model 915 2.2.1

- SCI 2.5.1
 - Command Summary 2.6, Appendix E
 - Primitive 3.3.1, 3.3.2.1, 4.2.3
- Select 1.2
 - Error 4.4.8
- Send Key 3.4.4.13
- Services Unit, Digital Appendix C
- Skip:
 - Automatic 3.4.3.1
 - Key 3.4.4.5
- Start Printer Order 5.2.3
- State:
 - Active 4.3.1
 - Disabled 4.3.1
 - Idle 4.3.1
 - PSC:
 - Active 4.3.1
 - Waiting 4.3.1
 - Suspended 4.3.1
 - Terminal 4.3.1
- Station:
 - Buffer, PSC 5.2.1
 - Error, Printer 3.2.2.4
 - Identification, PSC 5.2.4
 - Messages Appendix B
 - Terminate 3.3.2.1, 3.4.4.7
- Statistics 4.4
 - List Interactive Communications 4.4
- Subsystem, Remote Terminal 2.2, 2.2.2
- Summary, SCI Command Appendix E
- Supported:
 - Features 1.6
 - Printers 1.4.2.2
 - Video terminals 1.4.2.1
- Suspend Video Terminal 3.3.1, 3.4.4.6
- Suspended State 4.3.1

Synchronous Communications Procedures, Binary	Appendix F	EBCDIC/ASCII	Appendix D
Synonym	2.5.2.3	Transmit:	
System:		Arguments	5.3.5.2
Command Interpreter	2.5.1	Call	5.3.5.1
Initialize	4.2.3	Example	5.3.5.4
Terminal, Business	2.2.1	Routine	5.3.5
		Type, Terminal	3.3.2.1, 4.3.1, 5.2.4.2
Tab Mode	3.4.4.4	Unit:	
Task:		Address, Auto-Select Control ..	5.2.4.1, 5.7
Execute PSC	5.7	Digital Services	Appendix C
ID	5.6, 5.7	Unprotected Command, Erase All	5.2.3
Name	5.6, 5.7	Unsupported Features	1.8
Run ID	4.3.1	Variants, Message	Appendix B
Task Segment	5.5	VDT Model:	
Terminal:		911	2.2.1
Activation, Video	3.2	931	2.2.1
Address	2.3.2, 2.5.2.4, 3.2.1.2, 3.2.2.2, 3.3.2.1, 4.3.1, 5.2.4.2	Video Terminal:	
Auto-Select	2.5.2.4, 5.2.4.2, 5.7	Activation	3.2
Business System	2.2.1	Remote	2.2.1
Controller, Remote	2.2, 2.2.2.1	Suspend	3.3.1, 3.4.4.6
Pathname	3.3.2.1	Waiting State, PSC	4.3.1
Remote Video	2.2.1	WCC	5.2.3
State	4.3.1	Word Mode	3.4.4.4
Subsystem, Remote	2.2, 2.2.2	Wrap, Cursor	3.4.4.2
Suspend Video	3.3.1, 3.4.4.6	Write:	
Type	3.3.2.1, 4.3.1, 5.2.4.2	Command	5.2.3
Terminals Supported, Maximum	1.4.2	Control Character	5.2.3
Terminate:		XICC	4.2.3
Option	5.3.7	XPA	4.5
Station	3.3.2.1, 3.4.4.7	XPSC	5.7
Time-Out:		911 VDT, Model	2.2.1
Arguments	5.3.6.2	915 RVT, Model	2.2.1
Call	5.3.6.1	931 VDT, Model	2.2.1
Example	5.3.6.4	940 EDT, Model	2.2.1
Time-Out, Printer Station	4.8.4		
Time-Out Routine	5.3.6		
To End-of-Screen, Erase	3.4.4.5		
Translation:			
ASCII/EBCDIC	Appendix D		

FOLD



NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO. 7284 DALLAS, TX

POSTAGE WILL BE PAID BY ADDRESSEE

TEXAS INSTRUMENTS INCORPORATED
DIGITAL SYSTEMS GROUP

ATTN: TECHNICAL PUBLICATIONS
P.O. Box 2909 M/S 2146
Austin, Texas 78769



FOLD



**TEXAS
INSTRUMENTS**

