HONEYWELL

DPS 6 & LEVEL 6 SNA INTERACTIVE TERMINAL FACILITY USER'S GUIDE

SOFTWARE

DPS 6 & LEVEL 6 SNA INTERACTIVE TERMINAL FACILITY USER'S GUIDE

SUBJECT

Operation and Use of the Interactive Terminal Facility in a Systems Network Architecture Environment

SOFTWARE SUPPORTED

SNA Release 1.1.

SPECIAL INSTRUCTIONS

This manual has been extensively revised and rewirtten; therefore, change bars have not been used.

ORDER NUMBER

CR58-01

April 1983

Honeywell

PREFACE

This manual is written for those responsible either for the entry and manipulation of data through Honeywell WST/VIP terminals or for the processing of data through application programs executing on DPS 6 or Level 6 systems. It describes in detail the Systems Network Architecture (SNA) Interactive Terminal Facility (ITF), which, through emulation of an IBM 3274 control unit, is used for the transmission of data to an IBM host system in an SNA network environment.

Unless stated otherwise, the term "host system" refers to an IBM system capable of interfacing with the SNA ITF; and the term "WST/VIP terminal" refers to either the WST/VIP7200, WST/VIP7800, or WST/VIP7300 terminal display stations. In this manual, references to "DPS 6" also apply to the Level 6, unless otherwise specified.

The reader is assumed to have read any appropriate Honeywell MOD 400 and distributed data processing (DDP) documentation. This manual also assumes general knowledge about IBM DDP concepts and does not explain such concepts.

The major topics discussed are:

- Introduction to the capabilities of the SNA ITF
- Operating procedures for WST/VIP terminals when used in conjunction with the SNA ITF

USER COMMENTS FORMS are included at the back of this manual. These forms are to be used to record any corrections, changes, or additions that will make this manual more useful.

Honeywell disclaims the implied warranties of merchantability and fitness for a particular purpose and makes no express warranties except as may be stated in its written agreement with and for its customer.

In no event is Honeywell liable to anyone for any indirect, special or consequential damages. The information and specifications in this document are subject to change without notice. This document contains information about Honeywell products or services that may not be available outside the United States.

File No.: 1R63, 1S63 CR58-01

- Functions of the ITF-specific keys on the WST/VIP terminal keyboards

- Data fields and data field attributes, buffer manipulation, user exits, user application program interfacing, and error handling

- Error messages generated by the ITF

- Examples of typical Assembly language and COBOL application programs that interface with the SNA ITF.

The following conventions are used to indicate the relative levels of topic headings used in this manual:

Level 1 (highest)	ALL CAPITALS, UNDERLINED
Level 2	Initial Capitals, Underlined
Level 3	ALL CAPITALS, NOT UNDERLINED
Level 4	Initial Capitals, Not Underlined

MANUAL DIRECTORY

The following publications constitute the DPS 6/SNA manual set for release 1.1.

Manuals are obtained by submitting a Publications Requisition to:

Honeywell Information Systems Inc.
47 Harvard Street
Westwood, MA 02090

Attn: Publications Services

Order Number	Manual Title
CR56-01	IBM Distributed Data Processing Overview
CR57-01	DPS 6/SNA Administrator's Guide
CR58-01	SNA Interactive Terminal Facility User's Guide
CR59-01	SNA Remote Job Entry Facility User's Guide
CR60-00	SNA File Transfer Facility User's Guide
CZ74-00	GCOS 6 Data Base Augmented Real-Time Tracing System User's Guide
GB88-00	SNA Host System Programmer's Guide

The MOD 400 manual set provides supplementary information to the SNA manual set. See the following manual for directory and Master index information:

CZ01-00	GCOS MOD 400 Guide to Software Documentation.

In addition, appropriate IBM documentation should be consulted for host programming and configuration information.

Each software product has a Software Release Bulletin that users should consult before using the software. The DPS 6/SNA Software Release Bulletins are as follows.

GD72-00	SNA Transport Facility Software Release Bulletin
GD73-00	SNA Interactive Terminal Facility Software Release Bulletin
GD74-00	SNA Remote Job Entry Facility Software Release Bulletin
GD75-00	SNA File Transmission-6 Software Release Bulletin
GD76-00	SNA File Transmission-Host Software Release Bulletin

CONTENTS

Page

SECTION 1 INTRODUCTION.................................... 1-1

SNA Interactive Terminal Facility Capabilities............ 1-2
Typical Configuration..................................... 1-3
How to Use This Manual.................................... 1-4

SECTION 2 STARTING UP AND SHUTTING DOWN THE SNA INTERACTIVE TERMINAL FACILITY.................. 2-1

DPS 6 Start-up and Log-in................................. 2-1
ITF Startup... 2-1
  Starting Up the ITF Using MOD 400 Menus................. 2-2
  Starting Up the ITF Using MOD 400 Commands.............. 2-4
Shutting Down the ITF..................................... 2-4

SECTION 3 USING THE SNA INTERACTIVE TERMINAL FACILITY.... 3-1

Logging In to the ITF..................................... 3-1
  ITF Login Using Menus................................... 3-2
  ITF Login Using Commands................................ 3-4
Logging Off the ITF....................................... 3-5
Using Your Terminal for Data Entry........................ 3-5
  Entering Data... 3-9
    Protected and Unprotected Fields...................... 3-10
    Formatted and Unformatted Displays.................... 3-10
Keyboard Functions.. 3-10
  Data Entry.. 3-12
    Shift... 3-12
    Caps Lock... 3-12
  Cursor Control.. 3-12
    Moving the Cursor From Character to Character......... 3-12
      Cursor Up... 3-12
      Cursor Down... 3-13
      Cursor Right.. 3-13
      Cursor Left... 3-13
    Moving the Cursor From Field to Field................. 3-13
      Backtab... 3-14
      Home.. 3-14
      New Line.. 3-14
      Tab Forward... 3-14

Page

Field Control.................................. 3-15
Alpha Override................................ 3-15
Dup... 3-15
Erase Input................................... 3-15
Erase-to-End-of-Field......................... 3-15
Field Mark.................................... 3-15
Editing.. 3-16
Insert Mode................................... 3-16
Delete.. 3-16
Starting and Stopping Data Entry............... 3-17
Clear... 3-17
Cursor Select................................. 3-17
Device Cancel................................. 3-17
Enter... 3-17
Function Keys (PFl Through PF24).............. 3-17
Ident... 3-18
Print... 3-18
Reset... 3-19
Special Functions.............................. 3-19
Help.. 3-19
Load Matrix................................... 3-19
Logoff.. 3-19
System Request................................ 3-19
ITF-Supported Terminals.......................... 3-20
Key Equivalents.................................. 3-31

SECTION 4 APPLICATION PROGRAMS AND PROCEDURES............ 4-1

Configuring An Application Program.............. 4-1
User Program Exits.............................. 4-1
Application Program/Data Field Di.splay Interface.......... 4-3
Unformatted Display............................ 4-3
Formatted Display.............................. 4-3
Data Fields.................................... 4-3
Data Field Description........................ 4-4
Data Field Attributes......................... 4-4
ITF-User Application Interface.................. 4-5
Screen Image Buffer (SIB)....................... 4-6
Screen Image Data Area (SIDA)................... 4-6
User Program Interface Area (UPIA).............. 4-6
Return and Function Codes..................... 4-7

CONTENTS

Page

Return Code.. 4-7
Function Code Character.............................. 4-7
Terminal Identification.............................. 4-8
Cursor Location...................................... 4-9
Host Response Command................................ 4-9
Request Header, Request Unit, and Logical-Unit Address.. 4-9
Application Program Data Area........................ 4-9
COBOL Application Program............................ 4-9
Assembly Language Application Program............... 4-10

APPENDIX A ITF PROGRAM MESSAGES..................... A-1

APPENDIX B ASCII AND EBCDIC CHARACTER SETS........... B-1

APPENDIX C SAMPLE ITF COBOL APPLICATION PROGRAM....... C-1

APPENDIX D SAMPLE ITF ASSEMBLY LANGUAGE APPLICATION PROGRAM.. D-1

ILLUSTRATIONS

Figure Page

1-1 SNA Interactive Terminal Facility Environment....... 1-3

2-1 Sample ITF Invocation Using MOD 400 Menus........... 2-3

3-1 Sample ITF Login.................................... 3-3
3-2 Status Line Displays................................ 3-8
3-3 Typical Display Screen During Data Entry............ 3-9
3-4 VIP7200 Terminal.................................... 3-21
3-5 WST/VIP7200 Standard Keyboard....................... 3-22
3-6 WST/VIP7200 Data Entry Keyboard..................... 3-23
3-7 VIP7300 Terminal.................................... 3-24

ILLUSTRATIONS

Figure Page

3-8 WST/VIP7300 Standard Keyboard........................ 3-25
3-9 WST/VIP7300 Word Processing Keyboard................ 3-26
3-10 WST/VIP7300 Data Entry Keyboard..................... 3-27
3-11 VIP7800 Terminal.................................... 3-28
3-12 WST/VIP7800 Standard Keyboard....................... 3-29
3-13 WST/VIP7800 Word Processing Keyboard................ 3-30

4-1 Types of User Program Exits......................... 4-2
4-2 Display Showing Attribute Character Locations....... 4-4

C-1 Sample COBOL Application Program.................... C-2

D-1 Sample Assembly Language Application Program........ D-2

TABLES

Table Page

3-1 Status Line Indicators.............................. 3-5
3-2 IBM 3270 Functions.................................. 3-11
3-3 WST/VIP7200 Key Equivalents to 3274 Functions....... 3-32
3-4 WST/VIP7300 Key Equivalents to 3274 Functions....... 3-33
3-5 WST/VIP7800 Key Equivalents to 3274 Functions....... 3-34

B-1 ASCII/Hexadecimal Equivalents....................... B-3
B-2 EBCDIC/Hexadecimal/Binary Equivalents............... B-3

Section 1
INTRODUCTION

The SNA Interactive Terminal Facility (ITF) is a configurable software application package that operates under the GCOS 6 MOD 400 Executive. It enables a DPS 6 system to emulate an IBM 3274 control unit (model 1C) with IBM 3278 terminals and IBM 3287 printers attached. The ITF can operate concurrently with other application program functions executing on the DPS 6. For a description of ITF configuration in a DPS 6/SNA network environment, see the DPS 6/SNA Administrator's Guide.

Under the ITF, DPS 6 devices communicate with the host using a synchronous data link control (SDLC) line protocol. Input is in the form of commands and data streams sent to the host. Output can be directed immediately to one or more DPS 6 terminals or to one or more matrix or line printers.

The ITF is supported by the SNA Transport Facility, which allows end users to utilize standard MOD 400 functions and devices to interface with multiple IBM host systems. For a description of the SNA Transport Facility, see the IBM Distributed Data Processing Overview manual.

You can dynamically reassign DPS 6 terminals from operation with the ITF to other DPS 6 functions, and then back to the ITF. Devices that use the MOD 400 Listener are not dedicated to a single task but may be reassigned to meet the changing demands at the work center.

The ITF also provides a User Exit facility that permits user-written COBOL and Assembly Language programs to be executed before display and after transmission for each ITF terminal. The User Exit facility also supports "virtual terminals," which allow a DPS 6 application program to appear as a terminal to the host.

SNA INTERACTIVE TERMINAL FACILITY CAPABILITIES

The SNA Interactive Terminal Facility (ITF) provides the following capabilities:

- Communication with an IBM host system using either point-to-point or multipoint SDLC protocol
- Emulation of an IBM 3274 control unit cluster controller with attached 3277/78 display stations and 3287 matrix printers (up to 32 devices in combination)
- Multiple, simultaneous logical-unit connections to the SNA ITF
- Support of various session types
- Support of the ASCII character set on the DPS 6 system and the EBCDIC character set on the data communications link
- Acceptance, translation, and display of data from the IBM host system on WST/VIP7200, WST/VIP7800, and WST/VIP7300 terminals
- Acceptance, translation, and printing of data on either a line or serial (matrix) printer
- Support of 3270 field attributes (as described in the IBM Distributed Data Processing Overview)
- Support of buffered printing, including the following controls: new line (NL), forms feed (FF), carriage return (CR), and end-of-message (EM)
- Performance of all the IBM 3270 keyboard functions except Attention and cursor left/right into protected fields (cursor left/right is supported on fully protected screens)
- Support of half-duplex flip-flop transmission
- Ability of WST/VIP terminals to connect and disconnect to and from SNA with the execution of DPS 6 log-in procedures
- SCS printer support
- SNA error and message logging (as described in the DPS 6/SNA Administrator's Guide).

TYPICAL CONFIGURATION

Figure 1-1 illustrates a typical DPS 6 configuration that can support the SNA ITF. Detailed information for configuring such a system can be found in the DPS 6/SNA Administrator's Guide.

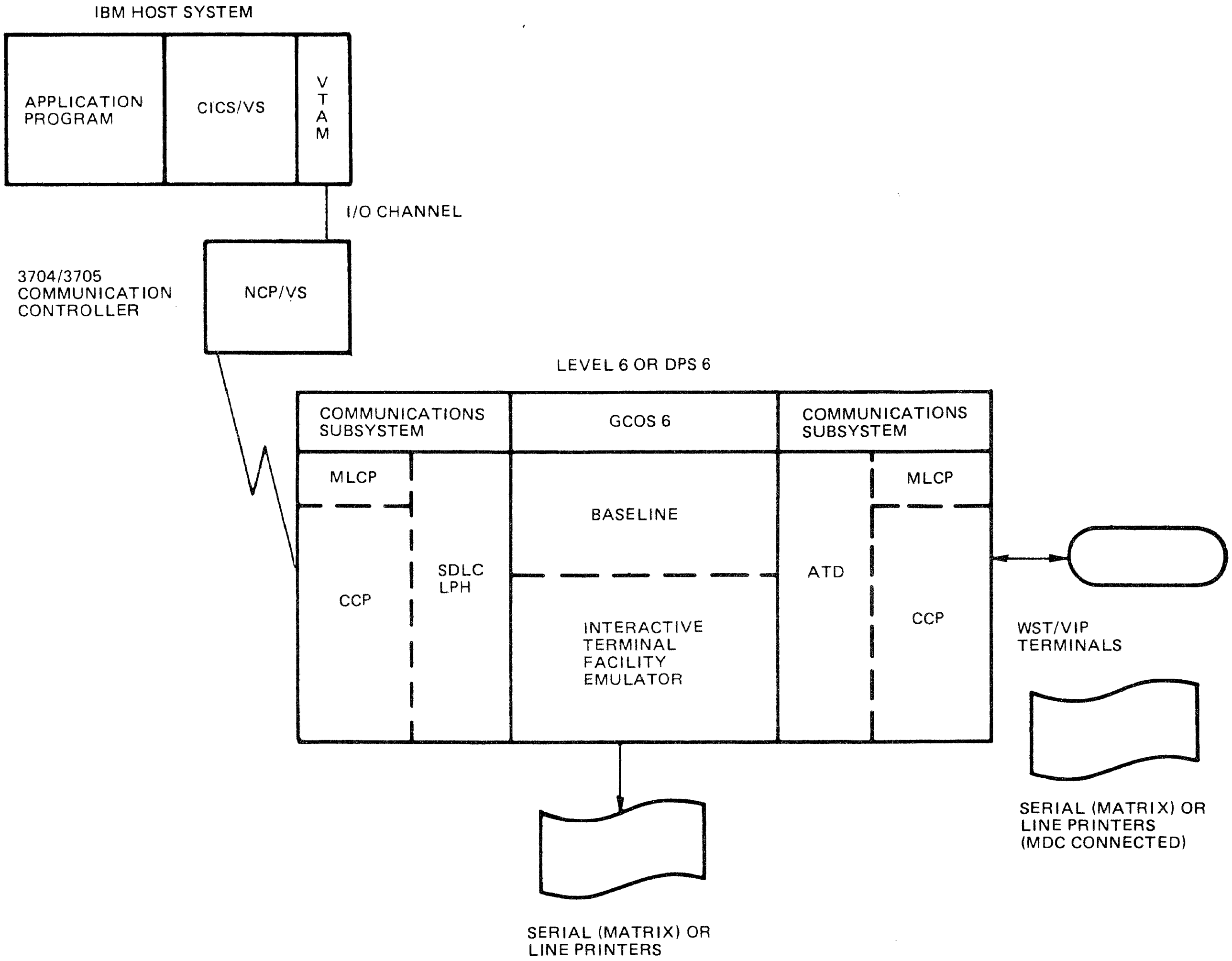


Figure 1-1. SNA Interactive Terminal Facility Environment

HOW TO USE THIS MANUAL

This manual is written for three distinct audiences: ITF operators, data entry specialists, and applications programmers.

If you are an ITF operator, you will want to read Section 2, which describes how to start up and shut down the ITF. You will also want to refer to Appendix A, which lists ITF messages that can appear at the operator terminal.

If you are a data entry specialist, you will want to read Section 3, which describes: 1) how to log in and out of the ITF, and 2) IBM 3270 functions and those DPS 6 terminal keys or key combinations that emulate those functions.

If you are an applications programmer, you will want to read Section 4, which describes application programs and procedures. You will also want to refer to Appendix C, which contains a sample COBOL application program, and Appendix D, which contains a sample Assembly language application program.

Section 2
STARTING UP AND SHUTTING DOWN THE SNA INTERACTIVE TERMINAL FACILITY

This section describes how to start up and shut down the SNA Interactive Terminal Facility (ITF). Before you can connect to the IBM host, you must do certain things locally (in other words, at the DPS 6). Once you have connected to the IBM host, data entry specialists can begin sessions to exchange data and messages with the host.

DPS 6 START-UP AND LOG-IN

If you are the DPS 6 system operator as well as the ITF operator, you must first start up the DPS 6 system. To start up the system, refer to the MOD 400 System User's Guide for directions. In either case, the DPS 6 must be operating before you can log in and start up the ITF.

The next step is to log in to MOD 400. If you are the system operator, you will already be logged in once you start up the DPS 6. If you are not the system operator, you will have to log in. Refer to the System User's Guide for information on how to log in.

ITF STARTUP

Once you have logged in to MOD 400, you must start up the ITF. You can do this using MOD 400 menus or GCOS 6 commands. MOD 400 menus run under the Subsystem Switcher and allow you to make selections from menus instead of entering command lines. To use menus, your system must be configured for that purpose and you must be a registered menu user on the system. For more information on the Subsystem Switcher and menus, refer to the MOD 400 Menu Management/Maintenance Guide. MOD 400 commands run under the Command Processor. For more information on commands and the Command Processor, refer to the Commands manual.

Before you invoke the ITF and connect to the IBM host computer, you must:

- Create a configuration file for ITF, normally done by the SNA administrator (for more information, see the DPS 6/SNA Administrator's Guide)

- Learn the pathname of the configuration file

- Learn the host application's logon command.

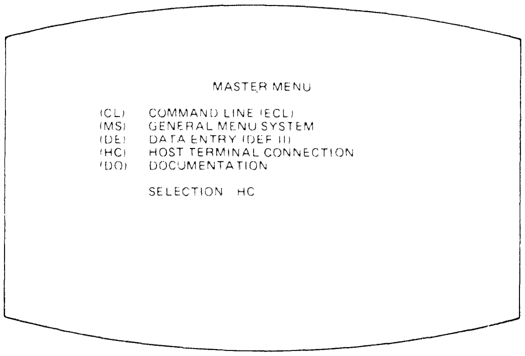
Starting Up the ITF Using MOD 400 Menus

To start up the ITF using MOD 400 menus, your system must be configured for that purpose and you must be registered as a system operator. These directions start at the Master Menu, which will appear when you have successfully logged in to your terminal.

From the Master Menu select "Host terminal connection" (HC). This will bring up the Host Connect Menu. From the Host Connect Menu select "SNA Facility" (SN). This will bring up the SNA Facility Menu. As you are an SNA operator, your SNA Facility menu will include three more selections than an SNA user menu. These three selections are: 1) SNA Administrator, 2) Configurator, and 3) Interactive. (A detailed description of these selections is available in the DPS 6/SNA Administrator's Guide.) Select "Interactive invocation" (II) from the SNA Facility menu. This selection brings up the ITF Invocation Form.

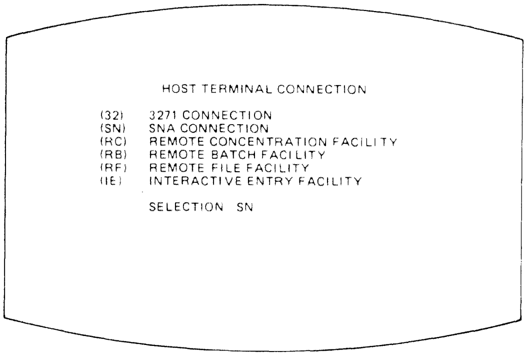
To abbreviate getting from the Master Menu to the ITF Invocation Form by skipping over two menus, enter "HC SN II" at the Master Menu. This will bring you directly to the ITF Invocation Form.

There are two prompts on the ITF Invocation Form: "Number of logical resource numbers" and "Number of logical file numbers." Default values of 50 each are provided. You can leave these defaults or enter new values. When this form has been transmitted, the ITF Login Form is displayed. This form prompts you to enter a logical node name. Once you do this, ITF is started up and becomes ready to accept logins from users. See Figure 2-1 for an example of ITF invocation using menus.



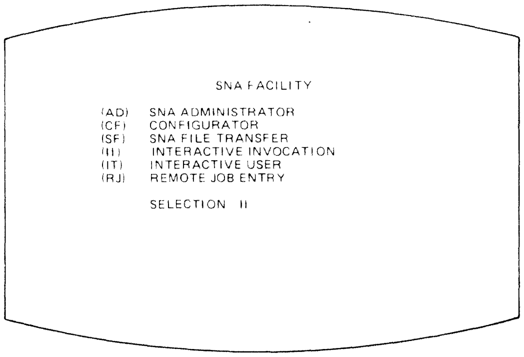
```
                    MASTER MENU

(CL)    COMMAND LINE (ECL)
(MS)    GENERAL MENU SYSTEM
(DE)    DATA ENTRY (DEF II)
(HC)    HOST TERMINAL CONNECTION
(DO)    DOCUMENTATION

        SELECTION  HC
```

This is the MOD 400 Master Menu with the selection HC (Host terminal connection). This selection will bring up the Host terminal connection Menu.



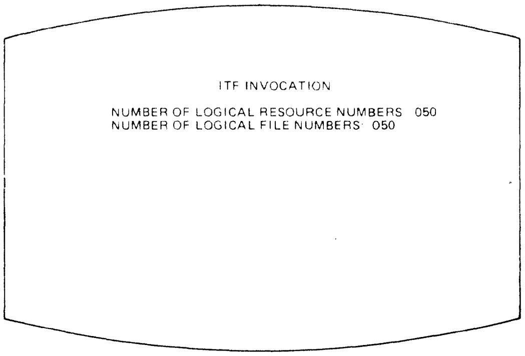
```
            HOST TERMINAL CONNECTION

(32)    3271 CONNECTION
(SN)    SNA CONNECTION
(RC)    REMOTE CONCENTRATION FACILITY
(RB)    REMOTE BATCH FACILITY
(RF)    REMOTE FILE FACILITY
(IE)    INTERACTIVE ENTRY FACILITY

        SELECTION  SN
```

This is the Host terminal connection Menu with the selection SN (SNA Facility). This selection will bring up the SNA Facility Menu.



```
                    SNA FACILITY

(AD)    SNA ADMINISTRATOR
(CF)    CONFIGURATOR
(SF)    SNA FILE TRANSFER
(II)    INTERACTIVE INVOCATION
(IT)    INTERACTIVE USER
(RJ)    REMOTE JOB ENTRY

        SELECTION  II
```

This is the SNA Facility Menu with the selection II (Interactive invocation). This selection will bring up the ITF Invocation Form.



```
                 ITF INVOCATION

NUMBER OF LOGICAL RESOURCE NUMBERS  050
NUMBER OF LOGICAL FILE NUMBERS  050
```

This is the ITF Invocation Form. This is the last form you must fill before ITF users can begin to log in to the host.

Figure 2-1. Sample ITF Invocation Using MOD 400 Menus

Starting Up the ITF Using MOD 400 Commands

To invoke the ITF for the correct host using MOD 400 commands, enter the SNA?ITF command from the operator terminal:

FORMAT:

```
SNA?ITF  node_name
```

ARGUMENT:

node_name

Name of the requested ITF node. (Obtain this information from the SNA administrator.)

SHUTTING DOWN THE ITF

Any registered SNA operator can shut down the ITF by executing the SNA STOP command from an SNA operator terminal. The command-line format of this command is:

FORMAT:

```
$A STOP ctl_args
```

ARGUMENTS:

-GP group_id

Task group identification of the executing ITF node.

-ND node_name

Name as entered in the configuration file for this node.

-LU lu_addr

Within the specified node or task group, shut down the specified logical unit.

-TM nn

Execute the shutdown in nn minutes, where nn is from 0 to 99. The default is ten minutes.

Menu Subsystem users can use the Stop Request on the SNA Operator Menu to accomplish the same purpose.

The SNA ABORT command (or, for menu users, the Abort Request) will also shut down the ITF, but STOP is recommended because: 1) the optional time delay allows for orderly user logoffs; and 2) ITF users are notified via a status-line message that a shutdown is imminent. For more information on SNA Operator commands and menus, see the DPS 6/SNA Administrator's Guide.

Section 3
USING THE SNA INTERACTIVE TERMINAL FACILITY

This section describes how to use the SNA Interactive Terminal Facility (ITF). There are four parts to this section:

1. Instructions on logging in and out

2. A brief description of IBM 3270 functions

3. Pictures of the terminals and terminal keyboards supported by the ITF

4. A table listing what key or key combination on your terminal corresponds to each 3270 function.

LOGGING IN TO THE ITF

To begin using the ITF, you must first log in. The process of logging in to MOD 400 is fully described in the MOD 400 System User's Guide, and will not be repeated here. Once you have logged in to MOD 400, you can log in to the ITF, assuming that it is currently executing. Depending on how you are registered on MOD 400, you will be using either menus or commands. ITF login for both menus and commands is described in the following paragraphs.

ITF Login Using Menus

To use MOD 400 menus, your system must be configured for that purpose and you must be a registered user. Login to ITF can be accomplished only after the system operator has started up the ITF node. Once the node has been invoked, start the login, beginning at the Master Menu (which will appear when you have successfully logged in to your terminal). The steps are:

1. From the Master Menu select the Host terminal connection Menu (HC)

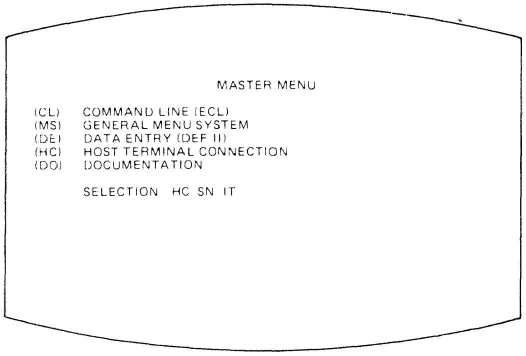
2. From the Host terminal connection Menu, select the SNA Facility Menu (SN)

3. From the SNA Facility Menu, select the ITF Secondary User Login Form (IT).

(To abbreviate getting from the Master Menu to the ITF Secondary User Login Form by skipping over two menus, enter "HC SN IT" at the Master Menu. This will bring you directly to the ITF Secondary User Login Form.)

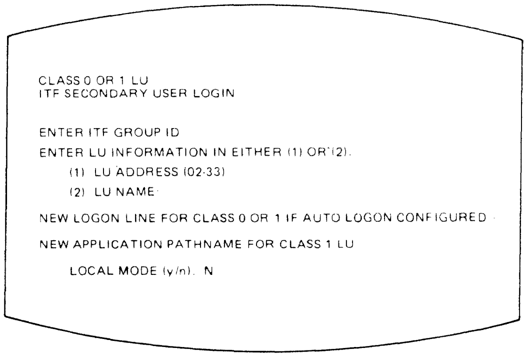
4. The ITF Secondary User Login Form requests your ITF group ID and either a LU address or a LU name. You may also fill in: (1) a new logon line if automatic logon is configured or (2) a new application pathname. The third prompt on this form is "Local Mode." Enter "Y" (yes), "N" (no), or leave the default provided. Transmit the Login form. You are now ready to log on to the host application program.

See Figure 3-1 for an example of ITF login using MOD 400 menus.



```
                    MASTER MENU

(CL)    COMMAND LINE (ECL)
(MS)    GENERAL MENU SYSTEM
(DE)    DATA ENTRY (DEF II)
(HC)    HOST TERMINAL CONNECTION
(DO)    DOCUMENTATION

        SELECTION  HC  SN  IT
```

This is the MOD 400 Master Menu with the abbreviated selections: HC (Host terminal connection), SN (SNA Facility) and IT (Interactive User). This series of selections will bring up the ITF Secondary User Login Form, skipping over the Host terminal connection and SNA Facility menus.



```
CLASS 0 OR 1 LU
ITF SECONDARY USER LOGIN

ENTER ITF GROUP ID
ENTER LU INFORMATION IN EITHER (1) OR (2).
     (1)  LU ADDRESS (02-33)
     (2)  LU NAME
NEW LOGON LINE FOR CLASS 0 OR 1 IF AUTO LOGON CONFIGURED
NEW APPLICATION PATHNAME FOR CLASS 1 LU
     LOCAL MODE (y/n).  N
```

This is the ITF Secondary User Login Form. You must fill in your group ID. The logical-unit address, logical-unit name, logon line and pathname fields are optional; and the local mode field has a default value provided.

Figure 3-1. Sample ITF Login

ITF Login Using Commands

You can log in directly to the ITF through the Listener. The login command is described below. Your SNA or system administrator will know the exact form of login line you require:

FORMAT:

```
LOGIN  user_id  $H -ARG   [ctl_arg]
```

ARGUMENTS:

user_id

You user identification, as assigned by your system administrator.

$H

The task group under which the ITF is executing.

-ARG

A keyword that introduces other control arguments.

-A nn

Identifies the LU address of the terminal (02 to 33). The -A or -N argument is required for class 1 sessions.

-H alternate_login_string

The alternate login string used if the logical unit is configured for automatic login.

-N cccccccc

Identifies the logical unit name as entered during SNA configuration. This value must be eight characters including trailing blanks. For class 1 sessions, the -A or -N argument is required; otherwise, they are optional.

-H alternate_login_string

Alternate login string to be used. This argument is valid only if the ITF logical unit is configured to accept automatic logins.

-P (path)

Pathname of a user application program. It overrides the pathname specified in the configuration for class 1 LUs. This argument is optional.

-L

Specifies local use when no host is required. Used for class 1 sessions only.

Logging Off the ITF

When you log off from the ITF, the current session with the host is closed and the terminal is returned to the DPS 6. You can now use this terminal for other applications.

If the automatic logoff (AUTO LOGOFF) feature is configured in the SNA network, you can log off from the ITF at any time by using the Logoff function.

If automatic logoff is not configured for your terminal, you can log off by using the System Request function and then the Logoff function (to return the terminal to the DPS 6).

USING YOUR TERMINAL FOR DATA ENTRY

When you enter data at your terminal, it is recorded on 24 lines of the display screen. A cursor indicates where on the screen your next keystroke will appear.

The 25th (bottom line) of the screen is a status line, used to indicate the status of your terminal and printer (see Table 3-1 and Figure 3-2). All ITF terminals except VIP7200 terminals have status lines.

Table 3-1. Status Line Indicators

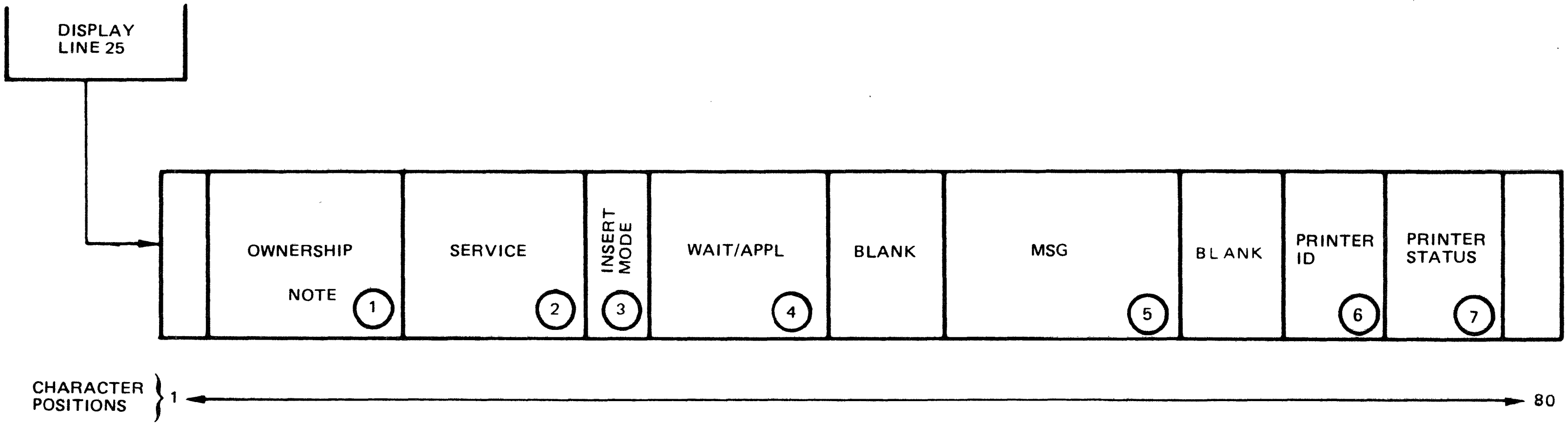
Indicator	Location	Description	Operator Action
Terminal Ownership		Current owner of the terminal display.	See the applicable ownership state.
INACTIVE	Field 1	Terminal not active and keyboard locked.	Wait for terminal to be activated by host or log off and log in later.
???	Field 1	Terminal not owned. Keyboard enabled; print operations allowed; but no transmissions to the host can be initiated.	Use SYS REQUEST to transfer display ownership to the system session.

Table 3-1 (cont). Status Line Indicators

Indicator	Location	Description	Operator Action
SYS	Field 1	Terminal owned by the system session.	Use SYS REQUEST to transfer display ownership to the unowned state if no job is in progress, or to the job if one is in progress.
JOB	Field 1	Terminal owned by the user session; SERVICE indicator illuminates following the JOB indicator.	Use SYS REQUEST to transfer display ownership to the system session.
SERVICE=Y N	Field 2	Terminal out of service at the host when SERVICE=N appears; terminal in service at the host when SERVICE=Y appears.	None if terminal not in service at the host.
IM	Field 3	Terminal in data entry Insert Mode.	None; information only. The IM indicator is turned off when terminal no longer in Insert Mode.
WAIT/APPL	Field 4	If WAIT, no transmission until a response from the host is received and the WAIT indicator goes off. If APPL, the application has control.	Wait for the host to respond and turn off WAIT before initiating another transmission.
MSG	Field 5	ITF error messages in normal blinking video.	For recoverable errors, the message is removed when error recovery is complete.

Table 3-1 (cont). Status Line Indicators

Indicator	Location	Description	Operator Action
PRINT=nn (01-31 or 70-85)	Field 6	Current printer assignment/class. If printer assigned, "nn" displayed. If none assigned and one is available, PRINT=?? displayed. If none assigned and none available, no display. If printer assigned, port address (01 to 31) or class (70 to 85) displayed.	Not applicable. (Information only.) Use IDENT to assign first available printer.
Printer Status display	-------	Printer status when a local copy operation performed.	Use PRINT key combination for status.
ACTV	Field 7	Printer in a local print operation.	Not applicable (information only).
BUSY	Field 7	Printer busy with other copy requests.	Not applicable (information only); use DEV CANCEL to cancel print request or wait for available printer.
VBSY	Field 7	Printer busy with a host print request.	Wait for available printer.
UNAV	Field 7	Printer busy with a DPS 6 print request.	Not applicable (information only); use DEV CANCEL to cancel print request or wait for available printer.

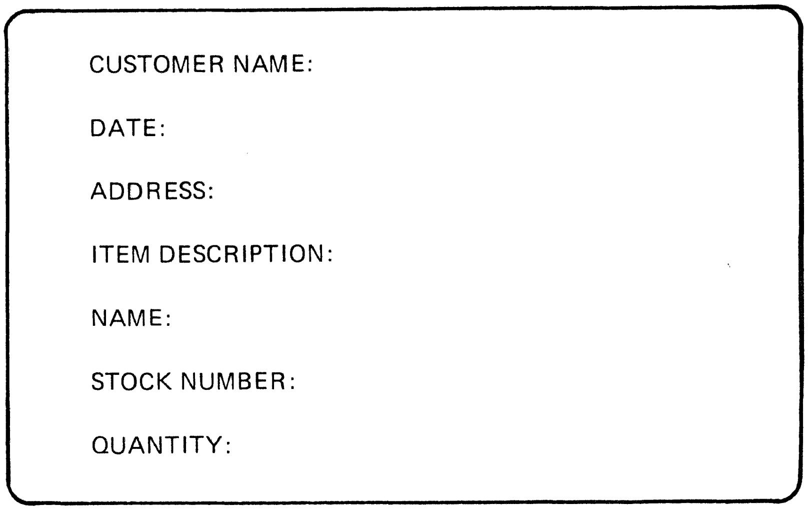


NOTE:
ENCIRCLED NUMBERS ARE THE DISPLAY FIELD LOCATION REFERENCE NUMBERS (SEE TABLE 5-3). THESE NUMBERS ARE FOR CONVENIENCE ONLY, AND DO NOT APPEAR ON THE SCREEN.

Figure 3-2. Status Line Displays

Entering Data

Figure 3-3 shows a typical display screen during data entry.



CUSTOMER NAME:

DATE:

ADDRESS:

ITEM DESCRIPTION:

NAME:

STOCK NUMBER:

QUANTITY:

Figure 3-3. Typical Display Screen During Data Entry

As you enter each character, the ITF validates it against field attributes (see Section 4) to ensure that the character belongs where you are entering it. If the data character is valid, it is displayed on the screen; if the data character is invalid, an alarm sounds and the cursor does not move. An example of an invalid character is a letter entered in a numeric field. To recover, you simply enter a valid character.

The cursor indicates where the next entered character will appear on the screen. The cursor advances automatically (unless you enter an invalid character) as you enter data. If a field is not completed, Return or one of the cursor movement keys moves the cursor to the next data entry field. Also, you can correct mistakes by positioning the cursor at the text to be changed and typing over the error. Cursor movement keys are described later in this section.

When you are finished entering data, you can instruct the ITF to send it to the host program by touching the ENTER key, one of the function keys, or one of the Attention keys. Which key you should use depends on the requirements of the host program. (You should find the information you need in the documentation of the host program.)

While the ITF is transmitting your data to the host program, the keyboard is "locked." The keyboard will not accept keystrokes until: 1) data transmission is complete, 2) the host system sends a message to the ITF to be sent to your screen, 3) you enter the Reset function, 4) you enter the Logoff function, or 5) you enter the System Request function.

PROTECTED AND UNPROTECTED FIELDS

An unprotected field is a region of the screen where you can enter data. A protected field is a region of the screen where you cannot enter data. An example of a protected field is an on-screen prompt such as "Customer name:".

In a partially protected screen, you can only enter protected fields using Cursor Up and Cursor Down. In a fully protected screen, you can use all the cursor-control keys.

See Section 4 for a detailed description of protected and unprotected fields and how to create them when writing an application program.

FORMATTED AND UNFORMATTED DISPLAYS

In a formatted display, data fields are formatted by an attribute character. In an unformatted display, data fields aren't formatted by an attribute character.

See Section 4 for a detailed description of formatted and unformatted displays and how to create them when writing an application program.

KEYBOARD FUNCTIONS

The following paragraphs describes what ITF can do, without considering what keys you use on the particular terminal and keyboard you're using. (That information is provided later in this section.) For discussion purposes, functions are divided into six categories:

1. Data entry functions used to enter data

2. Functions that terminate or enable data entry, or transmit data

3. Cursor control functions that position the cursor on the screen

4. Field control functions that affect either the format or structure of fields

5. Editing mode functions used to insert and delete characters

6. Special functions.

The IBM 3270 functions are listed in Table 3-2.

Table 3-2. IBM 3270 Functions

Function Name	Category
Alpha Override	Field control
Backtab	Cursor control
Clear	Start/stop data entry
Cursor Down	Cursor control
Cursor Left	Cursor control
Cursor Right	Cursor control
Cursor Select	Start/stop data entry
Cursor Up	Cursor control
Delete	Editing mode
Device Cancel	Start/stop data entry
DUP	Field control
Enter	Start/stop data entry
Erase Field	Field control
Erase Input	Field control
Field Mark	Field control
Function Keys (PF1-24)	Start/stop data entry
Home	Cursor control
Ident	Start/stop data entry
Insert Mode	Editing mode
Load Matrix	Special function
Logoff	Special function
New Line	Cursor control
Print	Start/stop data entry
Program Access (PA1-3)	Start/stop data entry
Reset	Start/stop data entry
System Request	Special function
Tab Forward	Cursor control

Data Entry

The data entry keys are the same on all terminals. The data entry functions include:

- All alphabetic characters (upper- and lowercase A through Z)
- Numeric characters (above the alphabetic keys, on a keypad, or both)
- Non-alphabetic characters such as the asterisk (*)
- The space bar. (Do not use the space bar to position the cursor unless text is to be replaced by spaces!)

SHIFT

SHIFT capitalizes all alphabetic characters. It also causes the top legend of any other key to be entered when pressed in combination with that key.

CAPS LOCK

CAPS LOCK capitalizes all alphabetic characters until the key is touched again. The CAPS LOCK key does not affect any other keys.

Cursor Control

The cursor control functions position the cursor without changing the contents of the screen. These functions include:

1. Functions that move the cursor one character position at a time
2. Functions that position the cursor to the first character position in a data entry field.

Cursor control is subject to limitations imposed by data field attributes defined by an application program. For example, you cannot use all the cursor-control functions to move the cursor into a _protected_ field.

MOVING THE CURSOR FROM CHARACTER TO CHARACTER

These cursor control functions position the cursor one character position at a time.

Cursor Up

Cursor Up moves the cursor up one line at a time in _either_ non-data entry or data entry fields.

If the cursor is on line 1 in any column, the cursor moves to line 24 in the same column.

Cursor Down

Cursor Down moves the cursor down one line at a time in _either_ non-data or data entry fields.

If the cursor is on line 24 in any column, the cursor moves to line 1 in the same column.

Cursor Right

Cursor Right moves the cursor one character position to the right in data entry fields.

If the cursor is in the last character position of a field, the cursor moves to the first character position of the next data entry field.

If a data entry field occupies more than one display line and the cursor is in the last character position of a line, the cursor moves from the end of the line to the first character position of the next line.

If a data entry field includes the last character position of line 24, and the cursor is in the last character position, the cursor moves to the first character position of the first data entry field, starting at line 1.

Cursor Left

Cursor Left moves the cursor one character position to the left in data entry fields.

If the cursor is in the first character position of a field, the cursor moves to the last character position of the previous data entry field.

If a data entry field occupies more than one display line, and the cursor is in the first character position of a line, the cursor moves to the last character position of the previous line.

If the cursor is in line 1, column 1, the cursor moves to the last character position of the first data entry field, starting at line 24.

MOVING THE CURSOR FROM FIELD TO FIELD

These cursor control functions position the cursor to the first character position in a data field.

Backtab

Backtab moves the cursor back to the first character position of a data field.

If a display consists of mixed (non-data entry and data entry) fields and the cursor is located either in the attribute character position of one of the mixed fields, or in the first character position of a data entry field, or in any character position of a non-data entry field, the cursor moves back to the first alphanumeric character position of the preceding data entry field.

If the cursor is located in an alphanumeric character position (other than the first position) of a data entry field, the cursor moves back to the first alphanumeric character position in the same data entry field.

If a display is either unformatted or consists only of non-data entry fields, the cursor moves back to the first character position on line 1 of the screen.

Home

If a display is formatted, Home moves the cursor to the first character position of the first data entry field on the screen.

If a display is unformatted, Home moves the cursor to line 1, column 1.

New Line

New Line moves the cursor to the first character position of the next display line into which you can enter data.

If a display consists of either mixed (non-data entry and data entry) fields or only data entry fields, the cursor moves to the first position of the next data entry field.

If a display consists of only non-data entry fields, the cursor moves to the first character position on line 1 of the screen.

If a display is formatted, the cursor wraps to the first character position of the next line.

Tab Forward

Tab moves the cursor forward to the first character position of the next data field.

If a display is either unformatted or consists only of non-data entry fields, the cursor moves forward to the first character position on line 1 of the screen.

Field Control

Field control functions affect either the format or structure of fields.

ALPHA OVERRIDE

Alpha Override changes an unprotected numeric field to an alphanumeric field. To use this function, position the cursor in the unprotected numeric field and touch the Alpha Override key combination.

DUP

DUP duplicates the information from the previous record in a data entry field. When you use this function, the ITF displays an asterisk (*) at the cursor.

The ITF transmits a code, represented by the displayed asterisk, to the host application program. Upon receipt of the code, the host application program initiates a duplication operation where the asterisk (code) is located.

ERASE INPUT

Erase Input erases all data entry fields. The cursor is repositioned to the first location of the first possible data entry field.

If the display consists of only non-data entry fields, the cursor moves to line 1, column 1.

If a display is unformatted, all character locations are cleared to null characters. The cursor moves to line 1, column 1.

ERASE-TO-END-OF-FIELD

Erase-to-End-of-Field erases the current data entry field, from the cursor position to the end of the field. The cursor does not move from its current location.

If either a display consists of non-data entry fields or the cursor is located in an attribute character position, an alarm sounds.

FIELD MARK

Field Mark displays a semicolon, used as a field delimiter.

If a display consists of data entry fields, subfields within a data entry field are delimited with semicolons.

If a display is unformatted, the semicolon indicates the end of a field.

If the cursor is either located in an attribute character position or in a non-data entry field, an alarm sounds and the cursor does not move.

Editing

Editing functions insert and delete characters in data entry fields.

INSERT MODE

Insert Mode allows you to enter consecutive characters into a data entry field without altering characters already there. Any non-alphanumeric key will terminate the insert mode. The position of the cursor after the insert mode terminates is determined by the key that you touch.

When you insert a character, any character already at that position, and all remaining characters within the field (except for null characters), are shifted one character position to the right.

If a data entry field continues from one line to the next and a character insertion on the first line causes a "character overflow" situation, the character occupying the last character position (position 80) in the first line is shifted to the first character position (position 1) of the next line.

If the cursor is located at a null (non-character) position in a data entry field and you insert a character, the character is inserted in the null position and no character shifting occurs.

If you try to insert characters in a filled field, an alarm sounds.

DELETE

Delete deletes the character at the cursor in a data entry field. The cursor does not move from its current location, and any characters to the right shift one position to the left. Vacated character positions at the end of the field are filled with null characters.

If a data entry field continues from one line to the next, characters in the next line shift to the left and, if necessary, wrap to the end of the previous line.

Starting and Stopping Data Entry

These functions enable or terminate data entry from the terminal. Some functions also cause the ITF to transmit data to the host computer.

CLEAR

If you use the Clear function while SYS or ??? is displayed on the status line, then your screen is cleared immediately.

If you use the Clear function while JOB or WAIT is displayed on the status line, your screen does not clear immediately; the keyboard is locked and only the Reset function will unlock it. Anything but the Reset function sounds the alarm. When the host computer sends an appropriate code, the WAIT state is extinguished, and your terminal screen clears.

CURSOR SELECT

Cursor Select lets you select one of a list of items displayed on the screen. To use this function, position the cursor within a selection field (selection fields are designated by a ? character) and touch the Cursor Select keys. The ? character changes to a > character. (If the character is already >, it changes back to ?, allowing you to change your selections before they are sent to the host.) Cursor Select detects characters in a field on the same line; it does not recognize characters in the same field on the following line.

You can also use the Cursor Select function to transmit data to the host computer, the way the Enter function does. To send the data to the host, you must select an attention field. Attention fields are designated by an & character. Position the cursor in the attention field and touch the Cursor Select keys.

DEVICE CANCEL

The Device Cancel function cancels an outstanding print request. When you enter this function, the print request is dequeued, the keyboard is unlocked, and the printer busy status (BUSY, VBSY, or UNAVL) clears.

ENTER

Enter sends the currently displayed data to the host computer and locks the keyboard. Until the keyboard is unlocked, if you touch a data entry key the alarm sounds.

FUNCTION KEYS (PF1 THROUGH PF24)

Function keys terminate data entry and cause the ITF to transmit the displayed data to the host computer. There are 24 IBM 3270 function keys, designated PF1 through PF24.

An application program defines what happens when you touch a function key. Each application program can assign a separate function to each function key. For example, an application program might indicate replacement of one display data entry form with another display data entry form whenever you touch the PF5 function key. Or, if you touch the PF2 function key, the application program could be notified that all necessary data is entered.

If you touch any of the function keys, the keyboard is locked. Until the keyboard is unlocked, if you touch a data entry key the alarm sounds.

IDENT

The Ident function assigns a printer or printer class for future local copies (using the Print function, described below). When you enter this function, the cursor disappears from the screen and the prompt "Enter printer port/class number" is displayed. Enter the number. If the printer you want is available, PRINT=nn is displayed, where nn is the number you entered.

If the number you entered (the port) is invalid, the previous assignment is displayed. Enter a valid port address. If the printer class you requested is invalid, ?? appears on the status line. Use the Ident function to reestablish the previous assignment.

PRINT

The Print function initiates a local hard-copy printing of the contents of your terminal screen.

If the printer assigned to your terminal is available and not busy, the contents of your screen are printed. If your printer is busy with other operations, your print request is queued, the keyboard is locked, and BUSY is displayed on the status line. You can wait for the printer to become available, or you can dequeue the print request by using the Device Cancel function.

If the printer is performing a host-initiated print operation, your print request (considered a local print request) is queued, the keyboard is locked, and VBSY is displayed on the status line. Again, you can wait for the printer to become available, or you can dequeue the print request by using the Device Cancel function.

If the printer is unavailable (because of a malfunction or because the printer is under control of someone else), your print request is queued, the keyboard is locked, and UNAV is displayed on the status line.

RESET

Reset unlocks your keyboard so you can enter more data.

Special Functions

These functions are not used frequently for data entry.

HELP

Help displays a table of key bindings for your terminal. Each of the functions described here is mapped to the key or key combination you use to invoke it. This is an ITF function, not an IBM 3270 function.

LOAD MATRIX

Load Matrix loads the printer matrix for an existing printer. A printer matrix assigns a printer to a class and defines its access. The host is responsible for defining the matrix. You verify matrix entries at your terminal. When you invoke Load Matrix, the matrix is validated and loaded. If you don't want to use the displayed matrix, use any function other than Load Matrix.

For a new printer, you must have the system reconfigured before you can change its matrix.

Printer matrixes are described in the IBM 3270 Component Description manual.

LOGOFF

Logoff logs you off the ITF and returns you to MOD 400.

SYSTEM REQUEST

The System Request function transfers ownership of the screen display. Display ownership is indicated by the status line, and only one session can be the display device owner at a time. During the session or the ownership period, only the device owner can display data. Any attempt by a non-owner to use the display is rejected.

If the display device is not owned, as indicated on the status line, you can use the System Request function to control which session owns the device.

You can also use the System Request function to transfer display ownership between sessions. This transfer interrupts communications taking place during the session without waiting for completion of outbound transmission (transmissions from the host). The inbound transmissions (transmissions to the host) normally are completed before this type of transfer takes place.

ITF-SUPPORTED TERMINALS

The following figures illustrate the terminals supported by the ITF and their various keyboards:

- Figure 3-4 illustrates the VIP7200 terminal
- Figure 3-5 illustrates the VIP7200 standard keyboard
- Figure 3-6 illustrates the VIP7200 data entry keyboard
- Figure 3-7 illustrates the VIP7300 terminal
- Figure 3-8 illustrates the VIP7300 standard keyboard
- Figure 3-9 illustrates the VIP7300 word processing keyboard
- Figure 3-10 illustrates the VIP7300 data entry keyboard
- Figure 3-11 illustrates the VIP7800 terminal
- Figure 3-12 illustrates the VIP7800 standard keyboard
- Figure 3-13 illustrates the VIP7800 word processing keyboard.

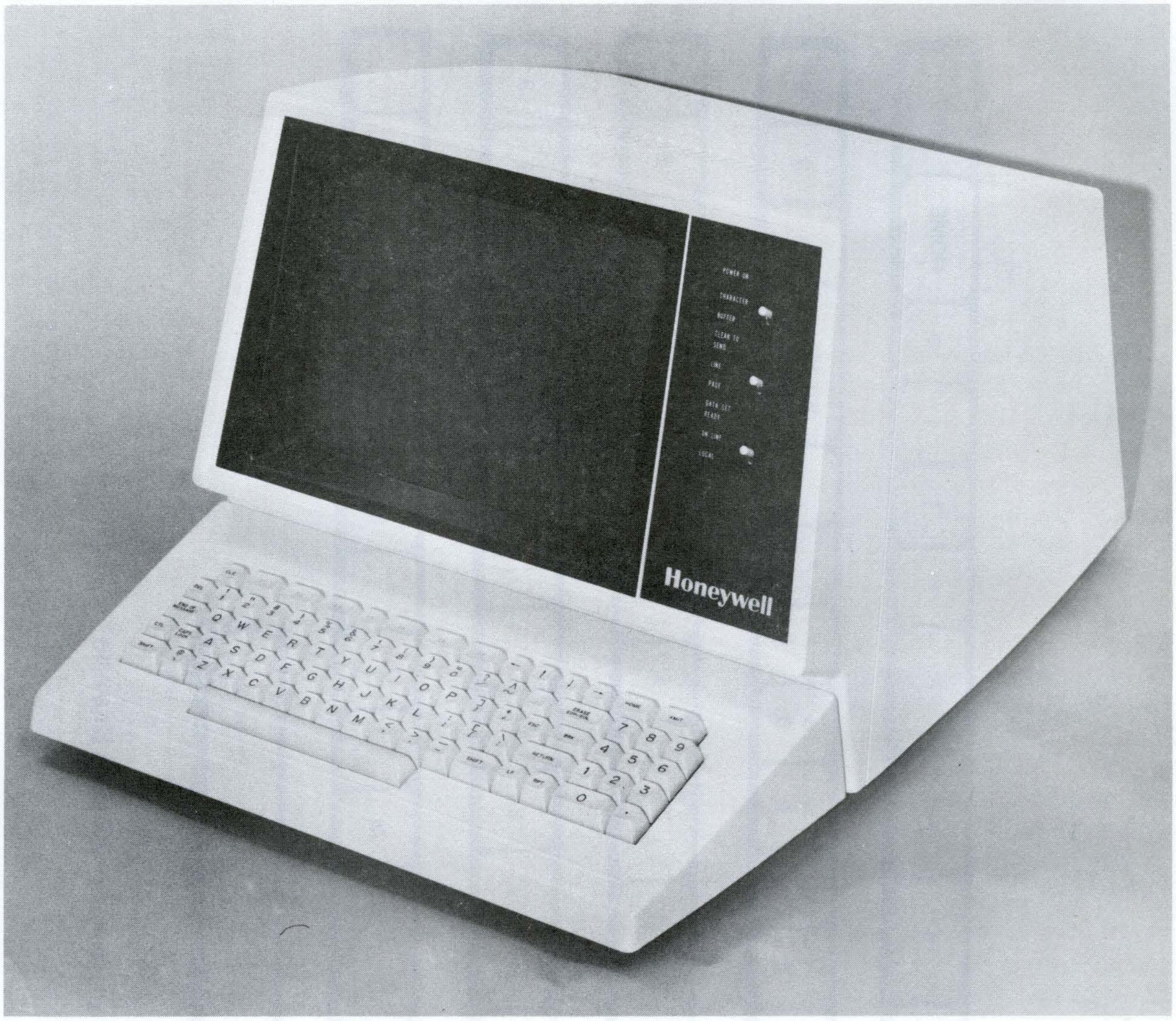


Figure 3-4. VIP7200 Terminal

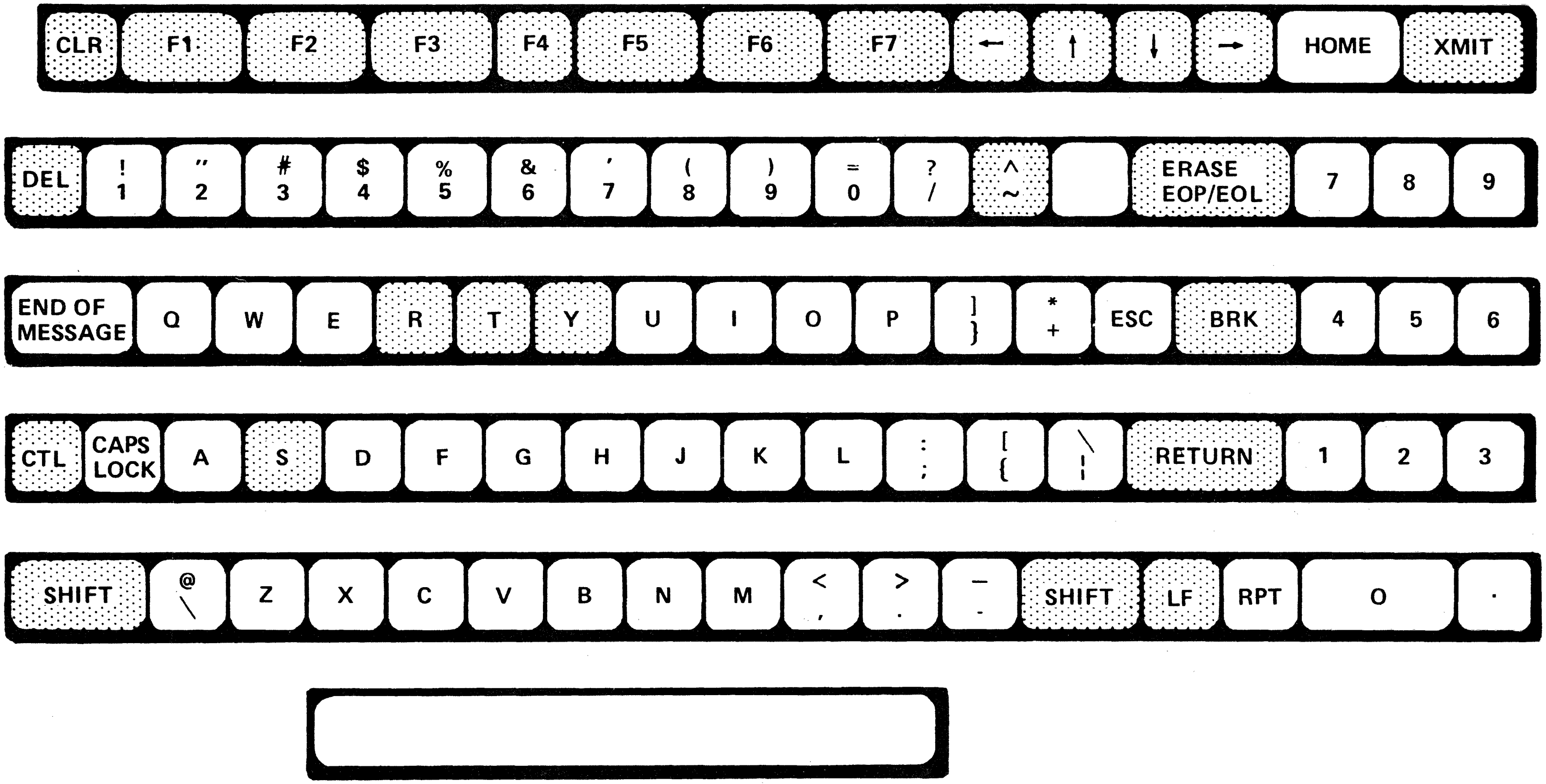


Figure 3-5. WST/VIP7200 Standard Keyboard

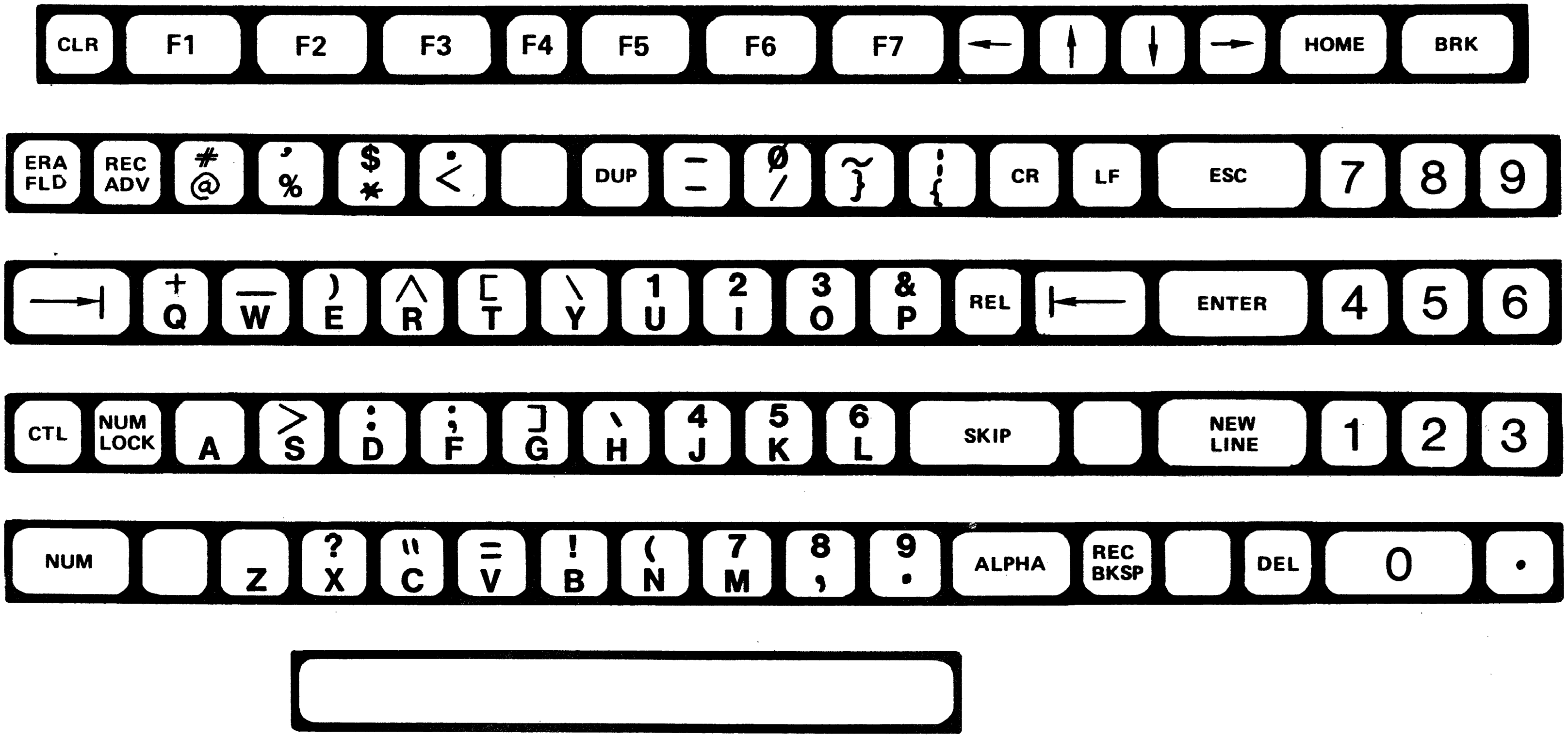


Figure 3-6. WST/VIP7200 Data Entry Keyboard

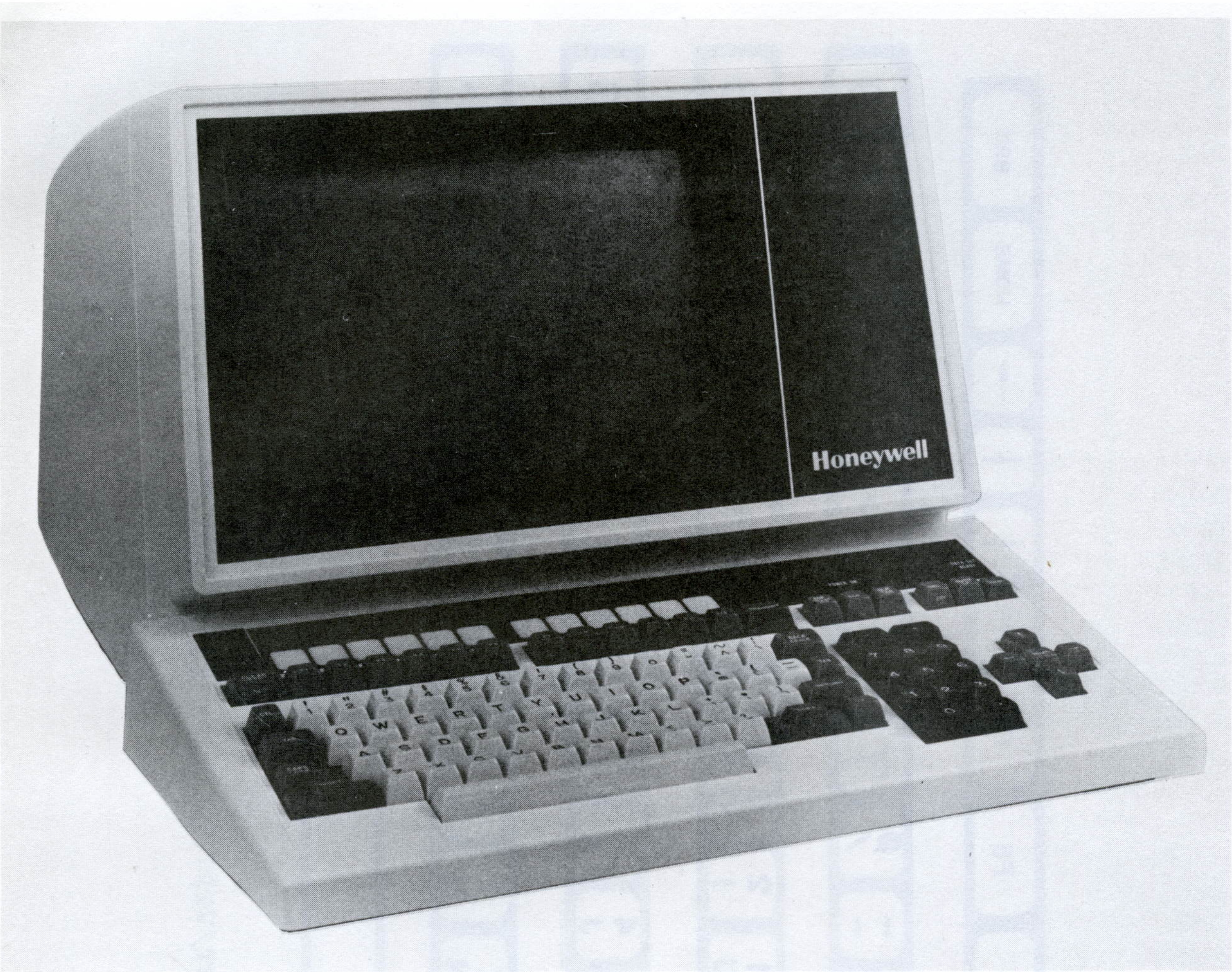


Figure 3-7. VIP7300 Terminal

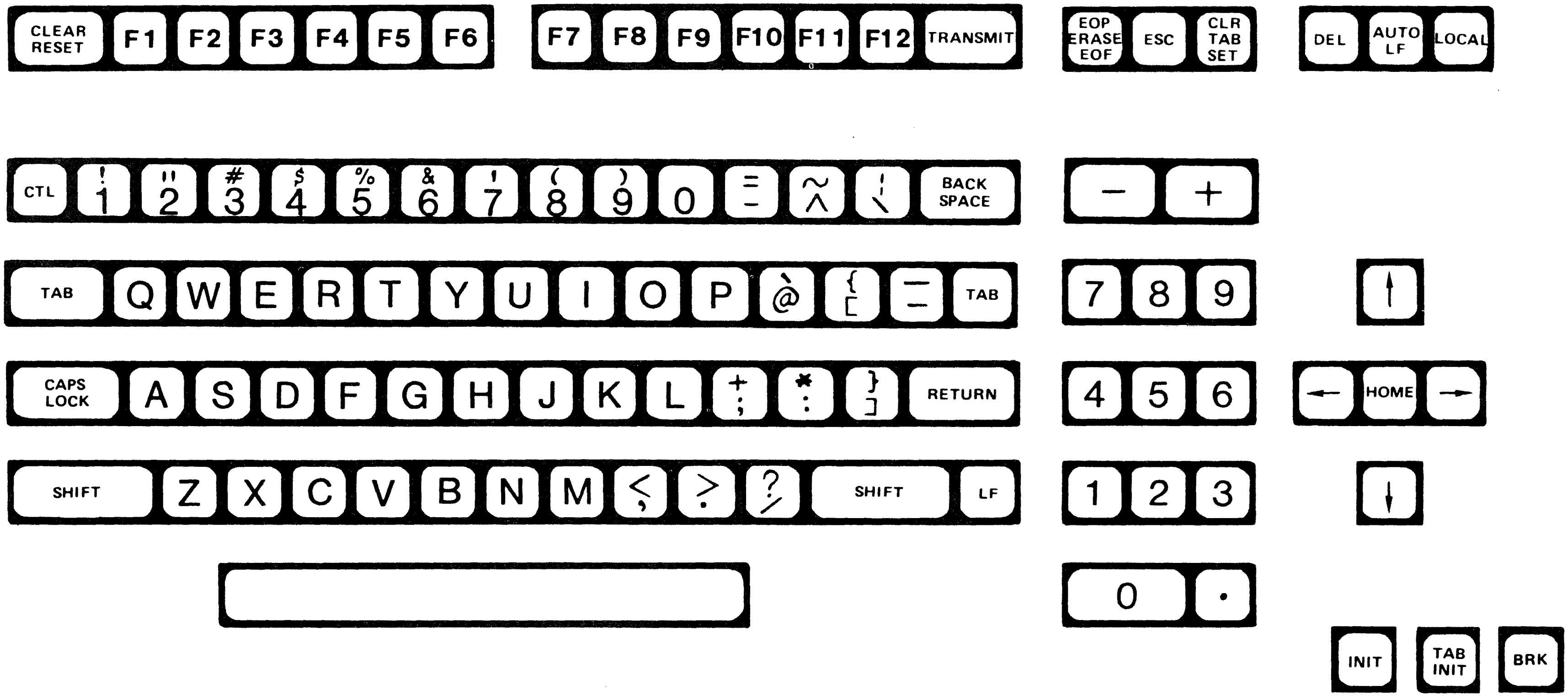


Figure 3-8. WST/VIP7300 Standard Keyboard

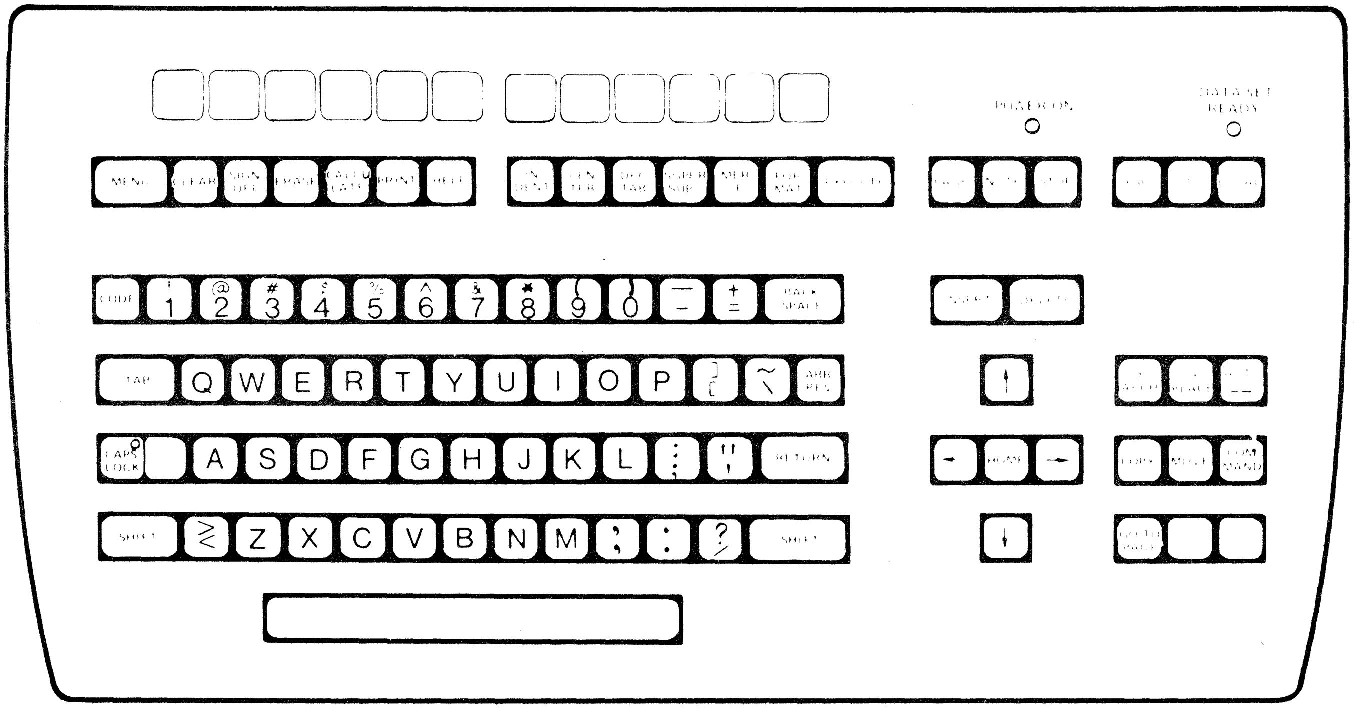


Figure 3-9. WST/VIP7300 Word Processing Keyboard

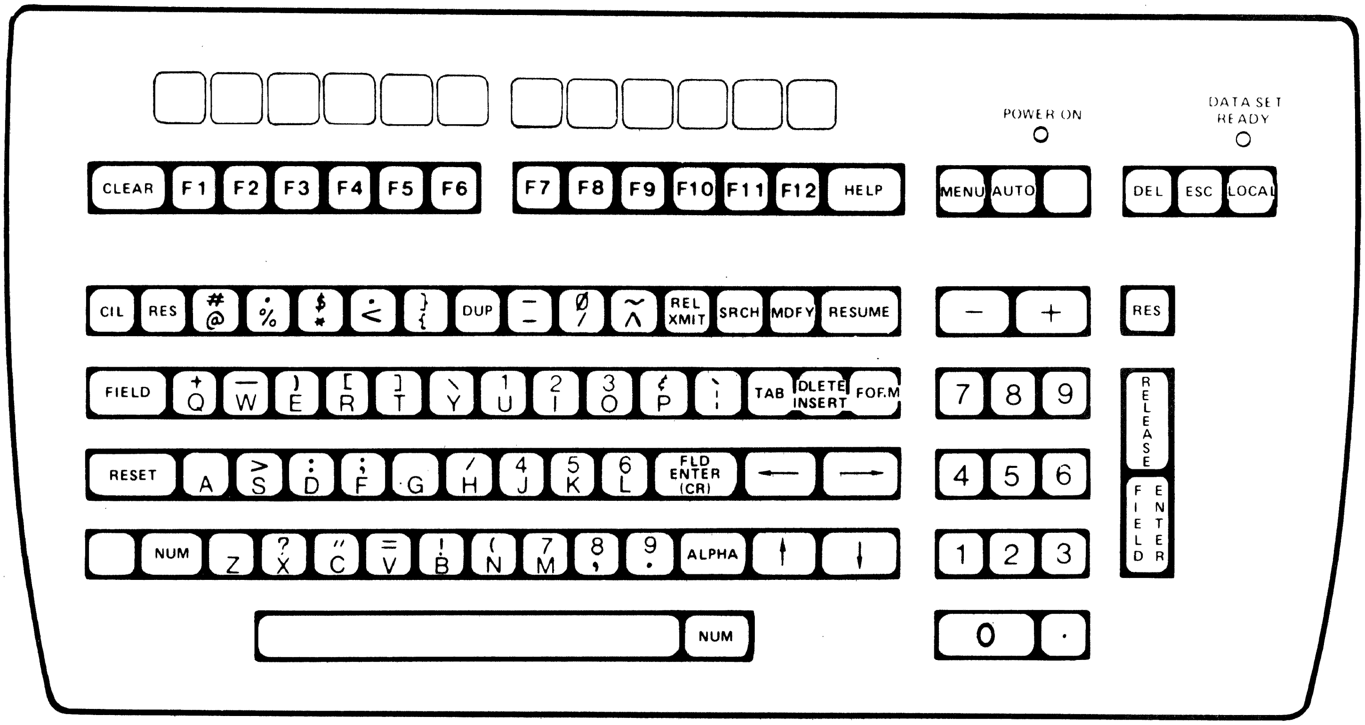


Figure 3-10. WST/VIP7300 Data Entry Keyboard

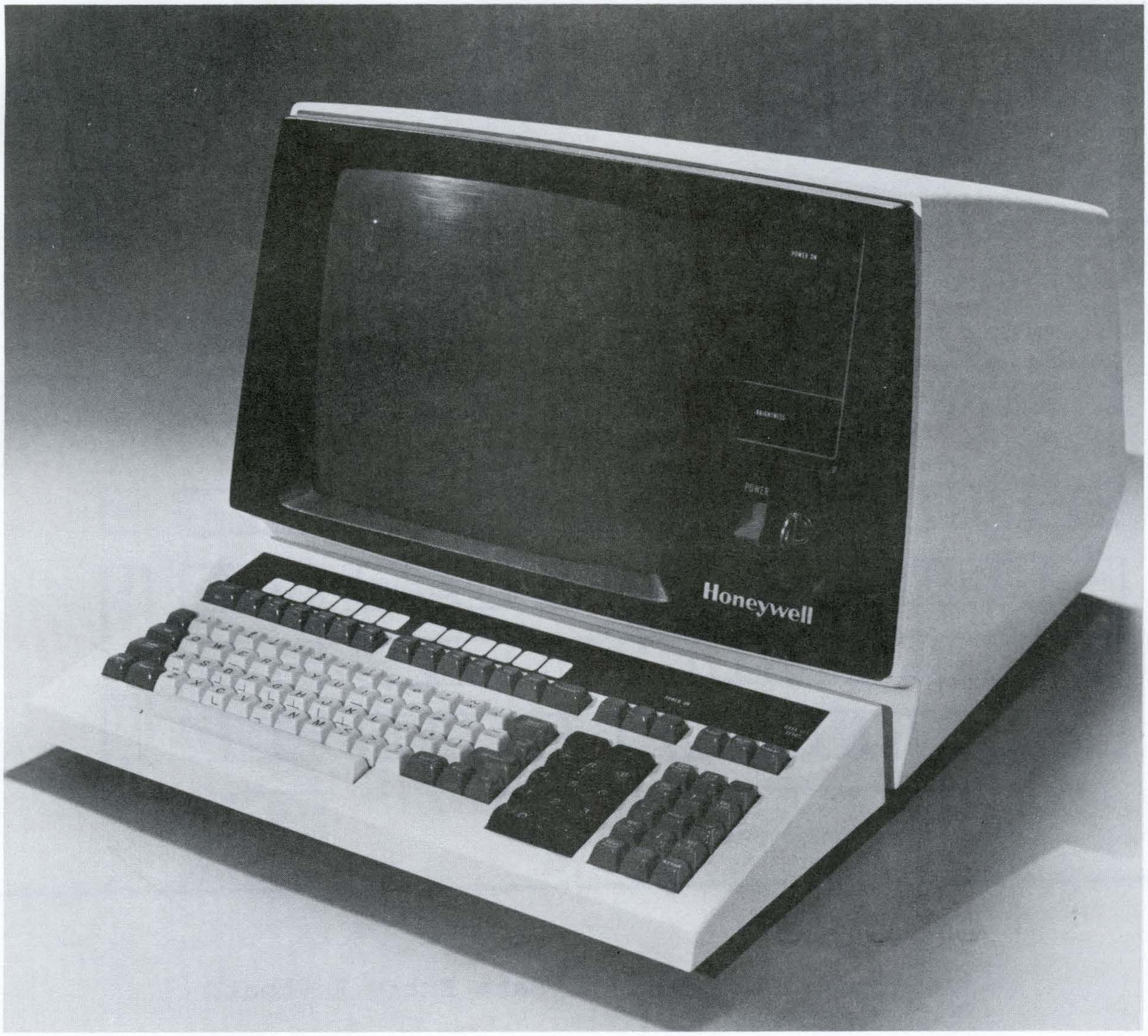


Figure 3-11. VIP7800 Terminal

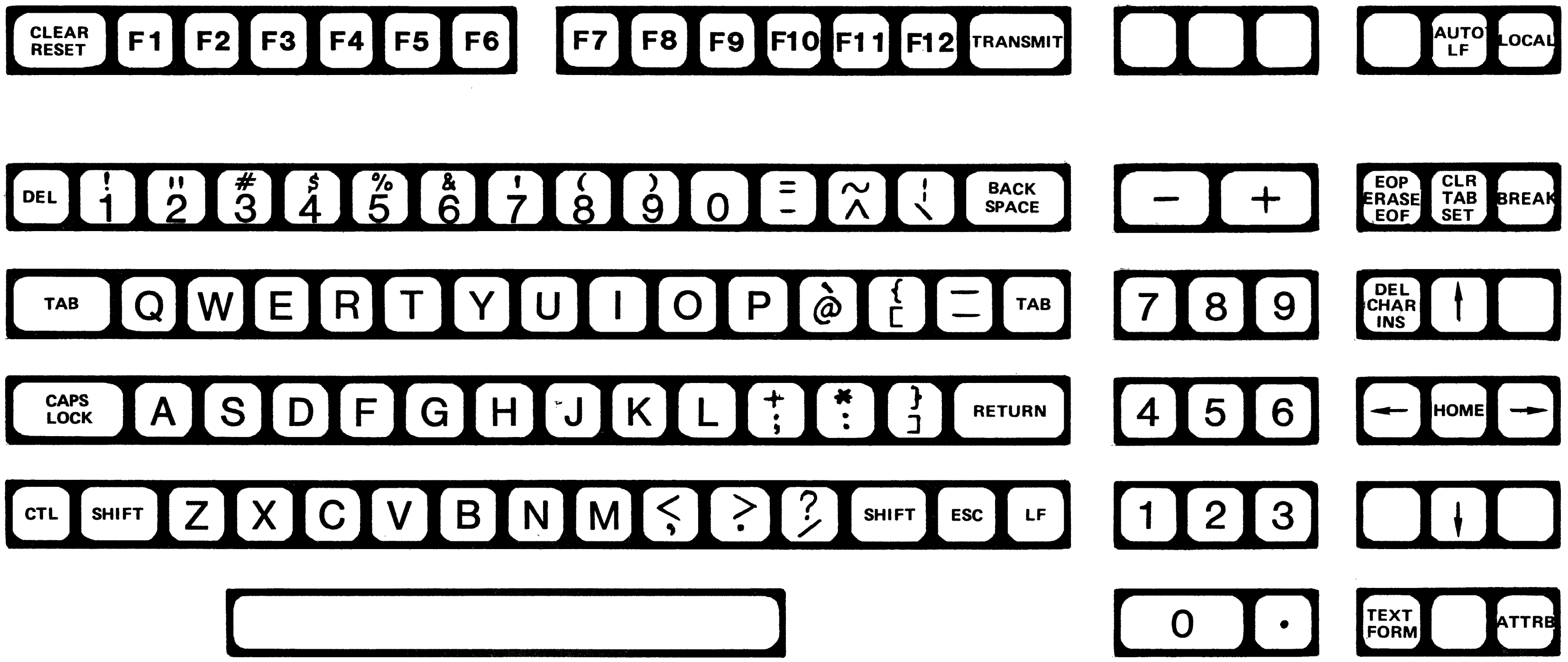


Figure 3-12. WST/VIP7800 Standard Keyboard

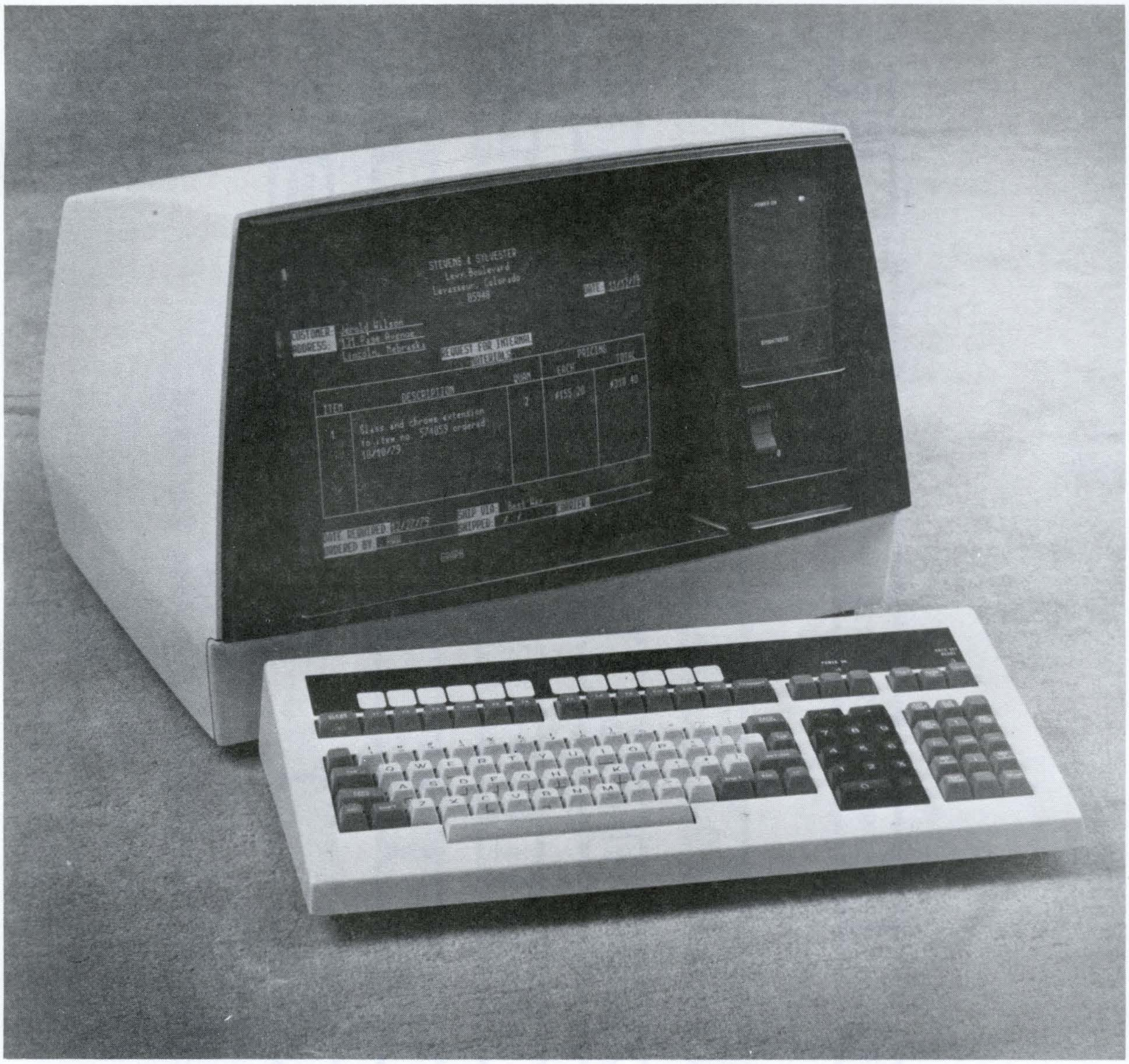


Figure 3-13. WST/VIP7800 Word Processing Keyboard

KEY EQUIVALENTS

Now that you know what the ITF can do, what keys do you use to invoke these functions? Tables 3-3, 3-4, and 3-5 will tell you. Table 3-3 lists IBM 3274 keyboard functions and the VIP7200 keys or key combinations that emulate those functions. Table 3-4 lists IBM 3274 keyboard functions and the WST/VIP7300 (standard, data entry, or word processing) keys or key combinations that emulate those functions. Table 3-5 lists IBM 3274 keyboard functions and the WST/VIP7800 (standard, data entry, or word processing) keys or key combinations that emulate those functions.

You need to know two things to use this table. One is what the prefix keys are. One prefix key is easy -- the SHIFT key. You know that to get a capital letter, you touch the SHIFT key and the letter key simultaneously. The notation for a capital "a" is SHIFT/A. There are two other prefix keys on these keyboards. One is the Control key, labeled CTL. The other is the Numeric key, labeled NUM. Use them just like the SHIFT key where indicated.

The other thing you need to know to use this table is the notation for double-labeled keycaps. For instance, the WST/VIP7801 standard keyboard has a key with a three-line label:

EOF
ERASE
EOP

This key will erase to the end of a field if you touch it unshifted, and erase to the end of a page (screen) if you touch it shifted. The corresponding entries in the table are ERASE (EOF) and SHIFT/ERASE (EOF).

Table 3-3. WST/VIP7200 Key Equivalents to 3274 Functions

IBM 3274 Function	7200 Standard Keyboard	7200 Data Entry Keyboard
ALPHA OVERRIDE	CTL/O	CTL/O
BACKTAB	DEL	DEL
CLEAR	CLR	CLR
CURSOR DOWN	↓	↓
CURSOR LEFT	←	←
CURSOR RIGHT	→	→
CURSOR UP	↑	↑
CURSOR SELECT	CTL/K	CTL/K
DELETE	CTL/R	CTL/R
DEVICE CANCEL	CTL/C	CTL/C
DUP	CTL/Z	CTL/Z
ENTER	XMIT	ENTER
ERASE FIELD	ERASE (EOL)	NUM/ERA FLD
ERASE INPUT	SHIFT/ERASE (EOP)	ERA FLD
FIELD MARK	CTL/Y	CTL/Y
HELP	CTL/H	CTL/H
HOME	HOME	HOME
IDENT	CTL/D	CTL/D
INSERT	CTL/S	CTL/S
LOAD MATRIX	CTL/X	CTL/X
LOGOFF	CTL/L	CTL/L
NEW LINE	LF	LF
PA1	F6	F6
PA2	F7	F7
PA3	CTL/T	CTL/T
PF1	F1	F1
PF2	F2	F2
PF3	F3	F3
PF4	F4	F4
PF5	F5	F5
PF6	SHIFT/F1	NUM/F1
PF7	SHIFT/F2	NUM/F2
PF8	SHIFT/F3	NUM/F3
PF9	SHIFT/F4	NUM/F4
PF10	SHIFT/F5	NUM/F5
PF11	SHIFT/F6	NUM/F6
PF12	SHIFT/F7	NUM/F7
PF13--PF24	Not supported	Not supported
PRINT	CTL/W	CTL/W
RESET	BRK	BRK
SYSTEM REQUEST	CTL/E	CTL/E
TAB	RETURN	SKIP

Table 3-4. WST/VIP7300 Key Equivalents to 3274 Functions

IBM 3274 Function	7300 Standard Key(s)	7300 Data Entry Key(s)	7300 Word Processing Key(s)
ALPHA OVERRIDE	CTL/O	CTL/O	AUTO
BACKTAB	CTL/TAB	CTL/TAB	ABBREV
CLEAR	SHIFT/(CLEAR)	NUM/(CLEAR)	SHIFT/MENU
CURSOR DOWN	↓	↓	↓
CURSOR LEFT	←	←	←
CURSOR RIGHT	→	→	→
CURSOR UP	↑	↑	↑
CURSOR SELECT	CTL/K	CTL/K	SHIFT/PAGE
DELETE	DEL	NUM/DLETE	DELETE
DEVICE CANCEL	CTL/C	CTL/C	SHIFT/SEARCH
DUP	CTL/Z	CTL/Z	SEARCH
ENTER	TRANSMIT	REL [XMT]	EXECUTE
ERASE FIELD	ERASE (EOF)	MDFY	COPY
ERASE INPUT	SHIFT/ERASE (EOP)	CTL/MDFY	SHIFT/COPY
FIELD MARK	CTL/Y	CTL/Y	REPLACE
HELP	HELP	HELP	HELP
HOME	HOME	CLEAR	HOME
IDENT	CTL/D	CTL/D	SHIFT/MOVE
INSERT MODE	TAB SET	INSRT	INSERT
LOAD MATRIX	CTL/X	CTL/X	CTL/X or SHIFT/REPLACE
LOGOFF	CTL/L	CTL/L	SHIFT/COMMAND
NEW LINE	RETURN	FLD ENTER	RETURN
PA1	CTL/V	MENU	PAGE
PA2	CTL/B	AUTO	NOTE
PA3	CTL/N	BLANK	STOP
PF1	F1	F1	CLEAR
PF2	F2	F2	SIGN OFF
PF3	F3	F3	ERASE
PF4	F4	F4	CALCULATE
PF5	F5	F5	PRINT
PF6	F6	F6	HELP
PF7	F7	F7	INDENT
PF8	F8	F8	CENTER
PF9	F9	F9	DEC TAB
PF10	F10	F10	SUPER SUB
PF11	F11	F11	MERGE
PF12	F12	F12	FORMAT
PF13	SHIFT/F1	NUM/F1	SHIFT/CLEAR
PF14	SHIFT/F2	NUM/F2	SHIFT/SIGN OFF
PF15	SHIFT/F3	NUM/F3	SHIFT/ERASE
PF16	SHIFT/F4	NUM/F4	SHIFT/CALCULATE
PF17	SHIFT/F5	NUM/F5	SHIFT/PRINT
PF18	SHIFT/F6	NUM/F6	SHIFT/COMMAND
PF19	SHIFT/F7	NUM/F7	SHIFT/INDENT
PF20	SHIFT/F8	NUM/F8	SHIFT/CENTER

Table 3-4 (cont). WST/VIP7300 Key Equivalents to 3274 Functions

IBM 3274 Function	7300 Standard Key(s)	7300 Data Entry Key(s)	7300 Word Processing Key(s)
PF21	SHIFT/F9	NUM/F9	SHIFT/DEC TAB
PF22	SHIFT/F10	NUM/F10	SHIFT/SUPER SUB
PF23	SHIFT/F11	NUM/F11	SHIFT/MERGE
PF24	SHIFT/F12	NUM/F12	SHIFT/FORMAT
PRINT	CTL/W	CTL/W	MOVE
RESET	RESET	RESET	MENU
SYSTEM REQUEST	CTL/E	CTL/E	SHIFT/AUTO
TAB FORWARD	TAB	TAB	TAB

Table 3-5. WST/VIP7800 Key Equivalents to 3274 Functions

IBM 3274 Function	7800 Standard Key(s)	7800 Data Entry Key(s)	7800 Word Processing Key(s)
ALPHA OVERRIDE	CTL/O	CTL/O	CTL/O or AUTO
BACKTAB	CTL/TAB	CTL/TAB	ABBREV
CLEAR	SHIFT/(CLEAR)	SHIFT/(CLEAR)	SHIFT/MENU
CURSOR DOWN	↓	↓	↓
CURSOR LEFT	←	←	←
CURSOR RIGHT	→	→	→
CURSOR UP	↑	↑	↑
CURSOR SELECT	CTL/K	CTL/K	CTL/K or PAGE
DELETE	SHIFT/CHAR (DEL)	NUM/CHAR (DEL)	DELETE
DEVICE CANCEL	CTL/C	CTL/C	CTL/C or SHIFT/SEARCH
DUP	CTL/Z	CTL/Z	CTL/Z
ENTER	TRANSMIT	TRANSMIT	EXECUTE
ERASE FIELD	ERASE (EOF)	ERASE (EOF)	COPY
ERASE INPUT	SHIFT/ERASE (EOP)	SHIFT/ERASE (EOP)	SHIFT/COPY
FIELD MARK	CTL/Y	CTL/Y	CTL/Y
HELP	CTL/H	CTL/H	CTL/H or COMMAND
HOME	HOME	HOME	HOME
IDENT	CTL/D	CTL/D	CTL/D or SHIFT/MOVE
INSERT MODE	CHAR (INS)	CHAR (INS)	INSERT
LOAD MATRIX	CTL/X	CTL/X	CTL/X or SHIFT/REPLACE
LOGOFF	CTL/L	CTL/L	CTL/L or SHIFT/COMMAND
NEW LINE	RETURN	RETURN	RETURN
PA1	CTL/V	CTL/V	PAGE
PA2	CTL/B	CTL/B	NOTE
PA3	CTL/N	CTL/N	STOP

Table 3-5 (cont). WST/VIP7800 Key Equivalents to 3274 Functions

IBM 3274 Function	7800 Standard Key(s)	7800 Data Entry Key(s)	7800 Word Processing Key(s)
PF1	F1	F1	CLEAR
PF2	F2	F2	SIGN OFF
PF3	F3	F3	ERASE
PF4	F4	F4	CALCULATE
PF5	F5	F5	PRINT
PF6	F6	F6	HELP
PF7	F7	F7	INDENT
PF8	F8	F8	CENTER
PF9	F9	F9	DEC TAB
PF10	F10	F10	SUPER SUB
PF11	F11	F11	MERGE
PF12	F12	F12	FORMAT
PF13	SHIFT/F1	NUM/F1	SHIFT/CLEAR
PF14	SHIFT/F2	NUM/F2	SHIFT/SIGN OFF
PF15	SHIFT/F3	NUM/F3	SHIFT/ERASE
PF16	SHIFT/F4	NUM/F4	SHIFT/CALCULATE
PF17	SHIFT/F5	NUM/F5	SHIFT/PRINT
PF18	SHIFT/F6	NUM/F6	SHIFT/HELP
PF19	SHIFT/F7	NUM/F7	SHIFT/INDENT
PF20	SHIFT/F8	NUM/F8	SHIFT/CENTER
PF21	SHIFT/F9	NUM/F9	SHIFT/DEC TAB
PF22	SHIFT/F10	NUM/F10	SHIFT/SUPER SUB
PF23	SHIFT/F11	NUM/F11	SHIFT/MERGE
PF24	SHIFT/F12	NUM/F12	SHIFT/FORMAT
PRINT	CTL/W	CTL/W	CTL/W
RESET	RESET	RESET	MENU
SYSTEM REQUEST	CTL/E	CTL/E	CTL/E or AUTO
TAB	TAB	TAB	TAB

Section 4
APPLICATION PROGRAMS AND PROCEDURES

GCOS 6 Assembly language or COBOL can be used to write an application program to interface with the SNA Interactive Terminal Facility (ITF). Details of application programs, data fields, program interfaces, and data flow are described in this section.

CONFIGURING AN APPLICATION PROGRAM

An application program appears to the host system as a logical unit (LU). An application program is identified to the ITF by specifying LU characteristics when configuring the ITF. For a description of how to specify LU characteristics, see the DPS 6/SNA Administrator's Guide.

Specification of LU characteristics permits association of the application program with a particular terminal (i.e., LU class 1) or, if interaction with a terminal is not desired, to execute the application program as a "virtual" terminal (i.e., LU class 2) not associated with any physical terminal. An application program can be shared by more than one terminal or represent more than one "virtual" terminal.

USER PROGRAM EXITS

The ITF provides exits to user application programs before data transmission and data display. User program exits can be used:

- With a terminal and a host link (Class 1 logical unit executing with remote host)
- With a terminal and no host link (Class 1 logical unit local)
- With a host link and no terminal (Class 2). In this case, the application program acts as a "virtual" terminal.

Figure 4-1 shows the types of user program exits.

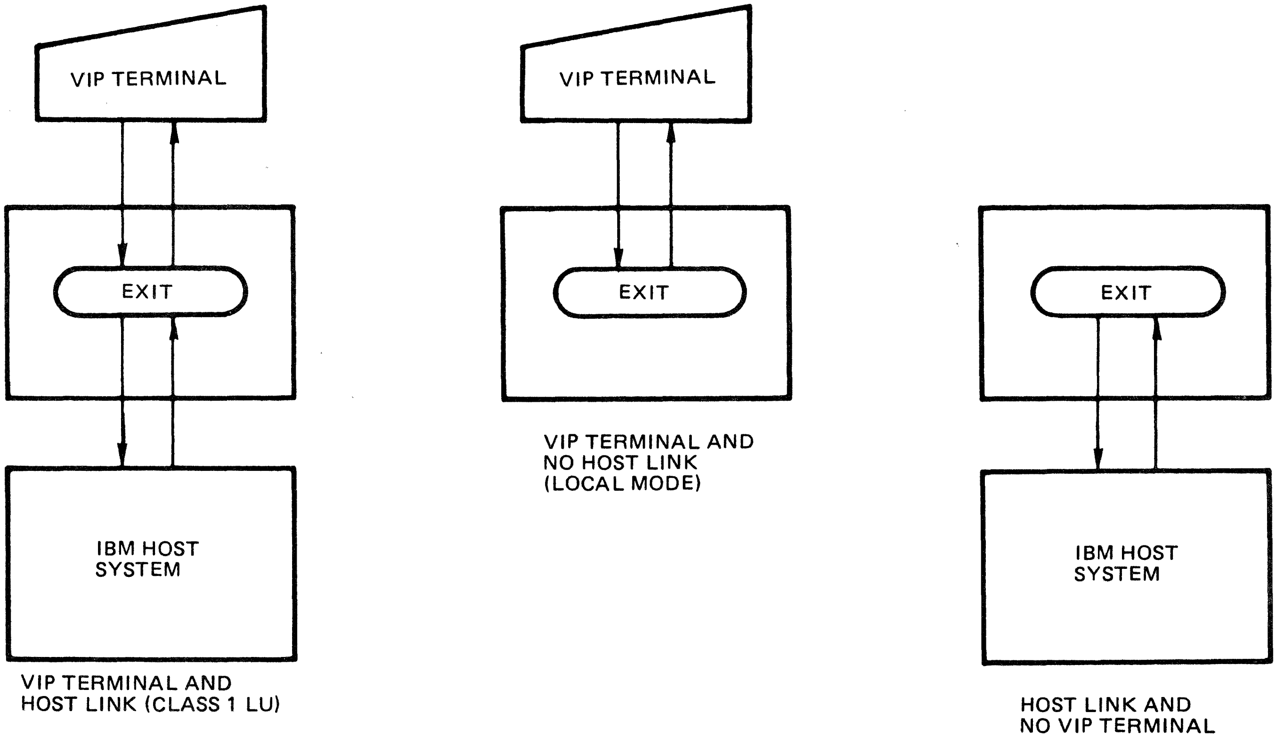

VIP TERMINAL
EXIT
IBM HOST SYSTEM
VIP TERMINAL AND HOST LINK (CLASS 1 LU)
VIP TERMINAL
EXIT
VIP TERMINAL AND NO HOST LINK (LOCAL MODE)
EXIT
IBM HOST SYSTEM
HOST LINK AND NO VIP TERMINAL

Figure 4-1. Types of User Program Exits

When the user program exit has occurred, an area called the screen image buffer (SIB), containing the screen image and certain control information, is passed to the application program. The program can manipulate data in the SIB and use GCOS 6 files, devices, and processing support. The application then passes a return code to the ITF, indicating how to process the data.

APPLICATION PROGRAM/DATA FIELD DISPLAY INTERFACE

A display consisting of data fields can either be formatted or unformatted. A data field is defined as an attribute character (described later in this section) and all the data following it, up to the next attribute character.

Unformatted Display

An unformatted display has no defined data fields, i.e., an attribute character is not used.

Formatted Display

A formatted display consists of data fields originating from the SIB. An attribute character is located at the start of each data field and defines the attributes (characteristics) of the field. Attribute characters are placed in the SIB by either a user application program or the ITF (as a result of commands and orders sent by the host system).

Data Fields

Data fields are areas on a screen used to contain data to be sent or received. These fields can be:

- Alphanumeric or numeric: An alphanumeric field consists of any alphabetic, numeric, or special character available on the terminal keyboard. A numeric field consists only of numeric characters available on the terminal.

- Displayable or non-displayable: Data fields can either be displayed (shown on the display screen) or non-displayed (not shown on the display screen).

- Protected or unprotected: When a data field is protected, no data can be entered in the field through a terminal keyboard in the field. When a data field is unprotected, data can be entered or deleted and any character position in the field can be modified.

- Modified or unmodified: When a data field is modified, the contents of the field are changed and transmitted to the host system. When a data field is unmodified, the contents of the field are unchanged and not transmitted to the host system.

These characteristics are established through use of the attribute character.

DATA FIELD DESCRIPTION

In a formatted display, each data field begins with an attribute character. The attribute character occupies the leftmost position of the data field. Although associated with the data field, the attribute character is not included in the data portion of the field. It is not visible when the data field is displayed on the display screen.

Field length is defined as the number of characters between, but not including, attribute characters. Successive data fields are thus separated from each other by attribute characters, as shown by the //// symbol in Figure 4-2.

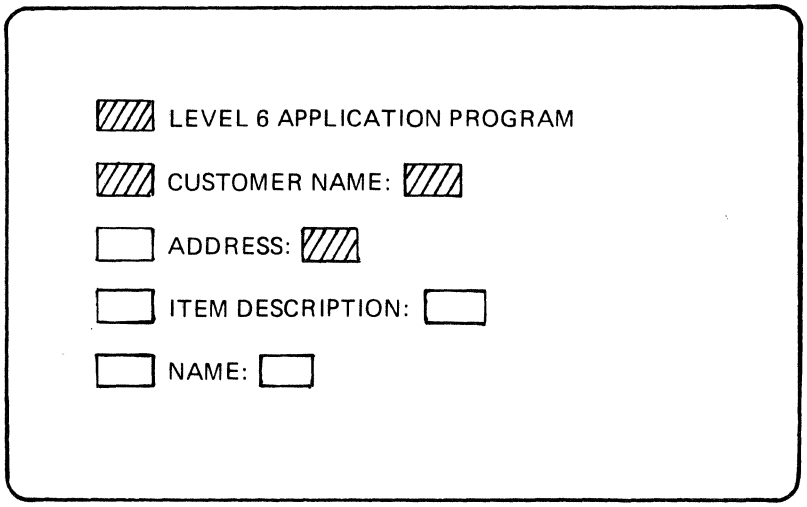


Figure 4-2. Display Showing Attribute Character Locations

DATA FIELD ATTRIBUTES

In addition to defining the beginning and end of a data field, an attribute character defines properties of the data field according to the bit configuration of the attribute character, which is set by the application program executing either on the host system or on the DPS 6. Attribute character bit assignments that can be set by a DPS 6 <u>application program</u> are:

Bit 0 - ACF (Attribute Character Flag): Must always be set to 1

Bit 1 - P/U (Protected/Unprotected): Protected field if set to 1; unprotected field if set to 0

Bits 2, 3 - No Extended Attributes = 00
Blink = 01
Underscore = 10
Inverse Video = 11

Bit 4 - MDT (Modified Data Tag): Field has been modified by a DPS 6 application program if set to 1; not modified if set to 0

Bit 5, 6 - Displayed/Not Detectable = 00
Displayed/Detectable = 01
High Intensity/Detectable = 10
Non-Displayed = 11

Bit 7 - N/A (Numeric/Alphanumeric): Numeric field only if set to 1; alphanumeric field if set to 0

For example, if only bit 0 is set to 1, a designated field is assigned the following default attributes:

- Unprotected
- No extended attributes
- Not modified
- Displayed/Not detectable
- Alphanumeric.

Data is transmitted to the host system without the attribute character.

ITF-USER APPLICATION INTERFACE

The ITF consists of code that provides an interface between the host system link and the user application program. The user application program (written in Assembly language or COBOL) executes on the DPS 6.

Once you have written the application program, it is linked as an independent bound unit. You must then provide the bound unit pathname to the ITF as part of logical unit configuration. For a description of how to specify the necessary LU characteristics, see the DPS 6/SNA Administrator's Guide.

Each time the application program is requested by the ITF, the address of the SIB is passed:

- For Assembly language programs, to the Task Request Block in location t_prm + 1 (see the description of the $TRBD and $RQTSK macro calls in the System Programmer's Guide)

- For COBOL programs, to the Linkage Section, to the appropriate data-name.

For Assembly language programs, the ITF waits for the application program to terminate with a Terminate Request ($TRMRQ) call. For COBOL programs the ITF waits for the application program to terminate with an EXIT statement. The relationship between the ITF and the application program is asynchronous. Notice of significant events that occur while the application is active (e.g., the logical unit is deactivated) is given to the ITF immediately after the application program terminates. These significant events can be obtained by requesting the program again and passing the proper function code.

SCREEN IMAGE BUFFER (SIB)

Screen images are maintained on the DPS 6 in a 2048-character buffer called the screen image buffer (SIB). Each emulated device, including terminals, has an accompanying SIB.

SIB management is handled through use of a pool of disk- or memory-resident buffers. The number of resident buffers is configured into the ITF by specifying certain physical unit (PU) characteristics when configuring the ITF. For a description of how to specify these characteristics, see the DPS 6/SNA Administrator's Guide.

A SIB is always memory resident while it is being manipulated. The number of resident SIBs is related to the number of active logical units (LUs). Thus, SIB is always memory resident during the time a user application program (LU) is active.

The SIB is divided into three areas:

1. Screen Image Data Area (SIDA)
2. User Program Interface Area (UPA)
3. Application Program Data Area (APDA).

The three areas of the SIB are described in more detail in the following paragraphs.

Screen Image Data Area (SIDA)

The screen image data area (SIDA) consists of an array of 1920 ASCII characters (0-1919) that represents the current image on the screen. When an operator initiates a change to the screen image, the change is reflected in the SIDA. It is the contents of the SIDA that is transmitted to the host system.

User Program Interface Area (UPIA)

The user program interface area (UPIA) contains data that is used as an interface between the ITF emulator and the user application program. The UPIA occupies characters 1920 through 1939 of the SIB. The UPIA contains:

1. Return and function codes
2. Terminal identification
3. Cursor location
4. Host response command
5. Request header, request unit, and logical-unit address.

RETURN AND FUNCTION CODES

Characters 1920 and 1921 are used to communicate commands between the ITF and the application program. The ITF initiates commands to the application program and sets hexadecimal values for the function code; the application program sets hexadecimal values for the return code.

Return Code

The application program sets the return code. The following are the valid return codes:

- X'01' - Initiate Session: This code causes ITF to transmit the logon string to the host. If the LU is configured for automatic logon, the string is taken from the configuration file. If no logon string is configured, the application must put the logon string in the SIB before passing this return code to the ITF

- X'02' Terminate Session: This causes the ITF to transmit the logoff string to the host . If the LU is configured for automatic logon/logoff, the logoff string is taken from the configuration file, otherwise the application must put the logoff string in the SIB before passing this return code to the ITF

- X'03' - This code causes the ITF to send the SIB to the host application program

- X'04' - This code causes the ITF to send the SIB to the Host System Services Control Program (SSCP)

- X'05' - This code causes the ITF to send the SIB to the terminal display

- X'06' - This code causes the ITF to wait for host action

- X'07' - Final Return: The application is terminated for the last time.

Function Code Character

When the ITF passes commands to the application program, the function code has one of the following hexadecimal settings:

- X'01' - First call to application program: This code is passed when the LU is activated by the host, or if the user is logged in in local mode at login time

- X'02' - Traffic Active: This code is passed to the application when the Start Data Traffic command is received by the logical unit from the host

- X'03' - This code is passed to the application when data is received from the Host System Services Control Program (SSCP)

- X'04' - This code is passed to the application when data is received from the host application program

- X'05' - This code is passed to the application in response to an Initiate Session return code if the LU is already in session, i.e., logged in to a host application

- X'06' - This code is passed to the application if traffic is inactive when the application requests transmission to the host application

- X'07' - This code indicates to the application that the session with the host application is terminated

- X'08' - This code indicates that transmission to the host cannot be initiated because the session with the host application is terminated

- X'09' - This code indicates that the SIB contains data from the operator's display

- X'0A' - This code indicates that the logical unit has been deactivated by the host or by an SNA operator. When this code is received, the application program must perform its termination logic

- X'0B' - This code has the same as the preceding code (X'0A') but the deactivation command was received while the application was busy

TERMINAL IDENTIFICATION

These characters are meaningful only if the function code (character 1921) equals X'09' upon entry to the application.

Character 1922 indicates that the AID key was pressed by the operator. A virtual application program must set character 1922 upon return to the ITF.

Character 1923 indicates the logical resource number of the terminal.

CURSOR LOCATION

These bytes contain the current cursor location expressed as an offset into the SIB (0 to 1919). Before returning to the ITF, the application program must set the current cursor location in these bytes.

HOST RESPONSE COMMAND

Character 1926 is reserved.

Character 1927: If the function code is X'04' (data from the host application), this character contains the command sent by the host. Valid host commands are:

```
Erase/Write - X'F5'
Write       - X'F1'
```

REQUEST HEADER, REQUEST UNIT, AND LOGICAL-UNIT ADDRESS

Characters 1928-1931 contain the three-character request header and the first character of the request unit; if the request unit is null, character 1931 is undetermined.

Character 1932 contains the logical-unit address (2-33) expressed in hexadecimal notation.

Application Program Data Area

Characters 1940 to 2047 contain the Application Program Data Area (APDA). Your application program can use this as a work area. Any data entered in this work area is maintained; however, you should store no addresses (i.e., pointers) in this area (especially addresses of locations in the SIB).

COBOL APPLICATION PROGRAM

A COBOL application program must be a called subroutine of the supplied object unit ZCOBIF.O. The entry point must be named "APPL." in the IDENTIFICATION DIVISION. The program can then be compiled and linked with ZCOBIF.O as the root. Use the -OC argument when invoking the Advanced COBOL compiler. The object unit ZCOBIF.O provides a mechanism whereby the screen image buffer can be passed to the COBOL application program. Once linked, the application program object unit and ZCOBIF.O form a single bound unit, having any user-supplied name, representing the executable task (i.e., the application program) that interfaces with the ITF.

Typical Advanced COBOL arguments to create a COBOL object unit are:

```
COBOLA APPL -OC -LSSU 3 -XREF -COUT !LPT00
```

Typical Linker directives to create a COBOL application are:

```
LINKER xxxxxx
LIB  >LDD>ZCART
LINKN >LDD>ZXOSRT>ZCOBIF; APPL; MAP; QUIT
```

where ZCART is the argument chosen when using MOD 400 Advanced COBOL; xxxxxx is the bound unit name; and APPL is the name of the COBOL object unit.

For an example of a typical COBOL application program written to generate a formatted data entry display form, see Appendix C.

ASSEMBLY LANGUAGE APPLICATION PROGRAM

A program written in Assembly language is assembled and linked as a single bound unit. You can give the bound unit any name you want. The bound unit represents the executable task, (i.e., the application program) that interfaces with the ITF.

For an example of a typical Assembly language application program, that was written to generate a formatted data entry display form, see Appendix D.

Appendix A
ITF PROGRAM MESSAGES

xx5B40 PRINTER CONFIGURATION ERROR

Cause: ITF cannot locate a specified LU configuration table.

Effect: The ITF task group cannot be started; processing terminates.

Action: Contact your local Customer Service representative if the problem persists after checking the specified printer configuration.

xx5B41 INVALID REQUEST FROM ACS

Cause: An internal processing error has occurred between ITF tasks.

Effect: The ITF task group terminates processing.

Action: Contact your local Customer Service representative if the problem persists after retry.

xx5B42 CANNOT FIND LUCT FOR LU NAME

Cause: An internal processing error has occurred between ITF tasks.

Effect: The ITF task group terminates processing.

Action: Contact your local Customer Service representative if the problem persists after retry.

xx5B43 CANNOT FIND LUCT FOR APPLICATION PATH

Cause: An internal processing error has occurred between ITF tasks in locating a Logical Unit Control Table.

Effect: The ITF task group terminates processing.

Action: Contact your local Customer Service representative if the problem persists after retry.

xx5B44 FAILURE TO ACTIVATE THE NODE

Cause: Failure to establish a connection to the host over the communications link.

Effect: No communication to the host is possible.

Action: None; ITF will automatically retry on an indefinite basis. If the failure persists, check the status of the physical link and host system.

xx5B45 PU TO LU BROADCAST FAILURE

Cause: An internal processing error has occurred between ITF tasks.

Effect: The ITF task group terminates processing.

Action: Contact your local Customer Service representative if the problem persists after retry.

xx5B46 CANNOT FIND LUCT FOR APPLICATION ASSIGNMENT

Cause: An internal processing error has occurred between ITF tasks.

Effect: The ITF task group terminates processing.

Action: Contact your local Customer Service representative if the problem persists after retry.

xx5B47 LU TO PU SEMAPHORE ERROR

Cause: An internal processing error has occurred between ITF tasks.

Effect: The ITF task group terminates processing.

Action: Contact your local Customer Service representative if the problem persists after retry.

xx5B48 LU TASK TERMINATION ERROR

Cause: An internal processing error has occurred in the termination of a logical unit task.

Effect: The ITF task group terminates processing.

Action: Contact your local Customer Service representative if the problem persists after retry.

xx5B49 INVALID REASON CODE FOR PU START-UP

Cause: An internal processing error has occurred in the activation of a physical unit.

Effect: The ITF task group terminates processing.

Action: Contact your local Customer Service representative if the problem persists after retry.

xx5B4B INVALID CURSOR LOCATION FOR SBA ORDER

Cause: An invalid screen cursor address has been sent by a host application program.

Effect: The data is rejected via a negative response to the host.

Action: Contact the host application programmer for a correction.

xx5B4C INVALID CURSOR LOCATION FOR IC ORDER

Cause: An invalid cursor location has been sent by a host application program.

Effect: The data is rejected via a negative response to the host.

Action: Contact the host application programmer for a correction.

xx5B4D INVALID DFC COMMAND RECEIVED

Cause: The host has sent an invalid Data Flow Control command according to SNA protocol.

Effect: The command is rejected via a negative response to the host.

Action: Contact the host administrator for a correction.

xx5B4E DATA FLOW CONTROL ERROR

Cause: An SNA protocol error has been generated by a host application.

Effect: A negative response is sent to the host.

Action: Contact the host application programmer for a correction.

xx5B4F DATA FLOW CONTROL ERROR

Cause: ITF has experienced an internal SNA protocol error between itself and the SNA Transport Facility.

Effect: ITF processing is suspended for the affected logical unit.

Action: Contact your local Customer Service representative if the problem persists after retry.

xx5B50 ACTIVE NODE RECEIVED

Cause: Informational message stating a successful link establishment for a host.

Effect: None; information only.

Action: None.

xx5B52 INVALID RETURN CODE FROM APPLICATION

Cause: An application program has returned an invalid return code.

Effect: The application program terminates.

Action: Correct the program logic and recompile.

Appendix B
ASCII AND EBCDIC CHARACTER SETS

Tables B-1 and B-2 illustrate the ASCII and EBCDIC character sets, respectively. In addition to the ASCII characters, Table B-1 shows the hexadecimal equivalents, and Table B-2 shows the binary and hexadecimal equivalents of the EBCDIC character set.

ASCII data in an SNA Interactive Terminal Facility (ITF) input file is converted to equivalent EBCDIC data for transmission to a host system. A reverse translation occurs for data received by the ITF from a host system. Three ASCII characters do not have equivalent EBCDIC representation and vice versa; they are converted as follows:

ASCII		EBCDIC	
Character	Hexadecimal Representation	Character	Hexadecimal Representation
^	5E	¬	5F
]	5D	¢	4A
[	5B	|	FA

Following are lists of the control characters and special graphic characters that appear in the two tables.

CONTROL CHARACTERS

BS	Backspace
BYP	Bypass
CC	Cursor control
CR	Carriage return
DC1	Device control 1
DC2	Device control 2
DC3	Device control 3
DC4	Device control 4
DS	Digit select
ENQ	Enquiry
EO	Eight ones
ESC	Escape
FF	Form feed
FS	Field separator
RLF	Reverse line feed
RS	Reader stop
SM	Set mode
SMM	Start of manual message
SOS	Start of Significance
GE	Graphic escape
GS	Group separator
HT	Horizontal tab
IFS	Interchange file separator
IGS	Interchange group separator
IL	Idle
IRS	Interchange record separator
IUS	Interchange unit separator
LC	Lowercase
LF	Line feed
NL	New line
PF	Punch off
PN	Punch on
RES	Restore
SP	Space
TM	Tape mark
UC	Uppercase
VT	Vertical tab

SPECIAL GRAPHIC CHARACTERS

¢	Cent sign
.	Period, decimal point
<	Less-than sign
(	Left parenthesis
+	Plus sign
&	Ampersand
!	Exclamation point
$	Dollar sign
*	Asterisk
)	Right parenthesis
;	Semicolon
¬	Logical NOT
-	Minus sign
/	Slash
|	Vertical line
,	Comma
%	Percent
_	Underscore
>	Greater-than sign
?	Question mark
'	Grave accent
:	Colon
#	Number sign
@	At sign
'	Prime, apostrophe
=	Equal sign
"	Quotation mark
~	Tilde
{	Opening brace
\	Reverse slash
⑁	Chair
¦	Logical OR
[	Opening bracket
]	Closing bracket
^	Circumflex

Table B-1. ASCII/Hexadecimal Equivalents

	H1							
H2	0	1	2	3	4	5	6	7
0			SP	0	@	P	`	p
1		DC1	!	1	A	Q	a	q
2		DC2	“	2	B	R	b	r
3		DC3	#	3	C	S	c	s
4		DC4	$	4	D	T	d	t
5	ENQ		%	5	E	U	e	u
6			&	6	F	V	f	v
7			'	7	G	W	g	w
8	BS		(	8	H	X	h	x
9	HT	EM	)	9	I	Y	i	y
A	LF		*	:	J	Z	j	z
B	VT	ESC	+	;	K	[	K	{
C	FF	FS	,	<	L	\	l	¦
D	CR	GS	—	=	M	]	m	}
E	SO	RS	.	>	N	∧	n	~
F	SI	US	/	?	O	—	o	DEL

Table B-2. EBCDIC/Hexadecimal/Binary Equivalents

Bit Positions 4, 5, 6, 7	Second Hexadecimal Digit	00				01				10				11				Bit Positions 0,1
		00	01	10	11	00	01	10	11	00	01	10	11	00	01	10	11	Bit Positions 2,3
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	First Hexadecimal Digit
0000	0			DS		SP	&	.						{[a]	}[a]	\[a]	0	
0001	1		DC1	SOS						a	j	~[a]		A	J		1	
0010	2		DC2	FS						b	k	s		B	K	S	2	
0011	3		TM							c	l	t		C	L	T	3	
0100	4	PF	RES	BYP	PN					d	m	u		D	M	U	4	
0101	5	HT	NL	LF	RS					e	n	v		E	N	V	5	
0110	6	LC	BS		UC					f	o	w		F	O	W	6	
0111	7		IL	ESC						g	p	x		G	P	X	7	
1000	8									h	q	y		H	Q	Y	8	
1001	9								\[a]	i	r	z		I	R	Z	9	
1010	A	SMM	CC	SM		¢	!	¦[a]									|[a]	
1011	B	VT				.	$	,	#									
1100	C	FF	IFS		DC4	<	*	%	@					ʃ[a]		⑂[a]		
1101	D	CR	IGS	ENQ		(	)	—	'									
1110	E		IRS			+	.	>	=					Y[a]				
1111	F		IUS			|	¬	?	"								EO[a]	

Appendix C
SAMPLE ITF COBOL APPLICATION PROGRAM

Figure C-1 shows a sample of a logical unit class 1 local mode COBOL program (whose bound unit and program name must be "APPL"), executing on the DPS 6, that creates a formatted data entry order form to be displayed on a WST/VIP terminal screen. A WST/VIP terminal operator can then enter (or modify) data on the order form. This sample program illustrates the use of various ITF function and return codes.

Comparable display forms can be created by a host application program and sent to the ITF.

NOTE

For a detailed description of the SIB and other material pertinent to writing a COBOL application program, see Section 4 of this manual.

```
GCOS6 MOD400-R3.0-09/17/1525     COBOLA   4.0                       APPL    83/05/16    1001    PAGE 0001

COMPILATION LISTING OF ^ITF>COBOL>APPL.C
COMPILED BY:  COBOLA VERSION  4.0  08/25/0843
COMPILED ON: 83/05/16  1001
    OPTIONS: -XREF  -LIST_SVTY 3 -COUT  MANUAL_C -OLD_CALL

                  IDENTIFICATION DIVISION.
                  PROGRAM-ID. APPL.
                ** *
                ** THIS APPLICATION IS DESIGNED TO EXERCISE THE INTERFACE AREA
                ** OF THE SNA/ITF PRODUCT.  IT SHOULD BE ASSOCIATED WITH A
                ** LU TYPE 2 CLASS 1 DISPLAY.  IT CAN BE RUN EITHER LOCALLY OR CAN
                ** COMMUNICATE WITH SRC.  IT IS SET UP TO LOGON TO MVS, BUT CAN
                ** EASILY BEALTERED SO THAT IT CAN LOG ONTO DOS.
                **
                **
                **             MENU SCREEN
                **
                ** ITF/SNA COBOL APPLICATION PROGRAM THAT IS DESIGNED TO LOGON TO SRC (MVS)
                ** WHEN CALLED WITH A FUNCTION CODE OF X'01'.A RECORD OF ALL FUNCTION CODES
                ** RECEIVED EXCEPT X'04' AND X'09' ARE WRITTEN ONTO THE DEVICE THAT ITF WAS
                ** STARTED FROM.
                **
                **  PA1 - DISPLAY THIS SCREEN  (MENU SCREEN)
                **  PA2 - SUSPEND FOR 2 MINUTES
                **  PA3 - CLEAR SCREEN OF DATA IN ORDER TO ENTER DATA TO SEND TO HOST
                **  PF24 - EXTENDED HILITE ATTRIBUTE FORM
                **  XMIT - SEND SCREEN IMAGE TO HOST
                **  CLEAR - CLEAR SCREEN OF ALL DATA, SENDS CLEAR CODE TO HOST
                **
                ** ALL OTHER FUNCTION AND PROGRAM ACCESS KEYS DISPLAY A DATA ENTRY FORM
                ** (WITH VARIOUS ATTRIBUTE CHARACTERS) WHILE WRITING TO A OUTPUT DEVICE,
                ** THAT MUST BE ASSOCIATED TO LFN 03 BEFORE ACTIVATION OF THIS PROGRAM, THE
                ** AID KEY THAT WAS PRESSED
                **
                ** *
                ** *
                **       ITF USER PROGRAM INTERFACE AREA (UPIA) CODES
                **       --------------------------------------------
                ** -
                ** THIS AREA (UPIA) CONTAINS DATA THAT IS USED AS AN
                ** INTERFACE BETWEEN ITF AND A USER APPLICATION PROGRAM.
                ** -
                **      SIDA - BYTES - 1-1919
                **      UPIA - BYTES - 1920-1959
                **      APDA - BYTES - 1940-2047
                ** -
                ** BYTES 1920 (RETURN CODE) AND 1921 (FUNCTION CODE) ARE USED TO COMM.
                ** COMMANDS BETWEEN ITF AND A USER APPL..  ITF INITIATES COMMANDS TO THE
                ** USER APPL AND SETS HEXADECIMAL VALUES FOR THE FUNCTION CODE.  THE USER
                ** APPL SETS HEXADECIMAL VALUES FOR THE RETURN CODE.
                **   ---------------
                **  \_______________\---------RETURN CODE - PASSED FROM APPL TO ITF
                **      BYTE 1920
                ** X'01' = INITIATE SESSION
                ** X'02' = TERMINATE SESSION
                ** X'03' = SEND SIB TO HOST APPL
                ** X'04' = SEND SIB TO SSCP
                ** X'05' = SEND SIB TO TERMINAL DISPLAY
                ** X'06' = WAIT FOR HOST OR DISPLAY ACTION
                ** X'07' = FINAL RETURN FROM APPL TO ITF
```

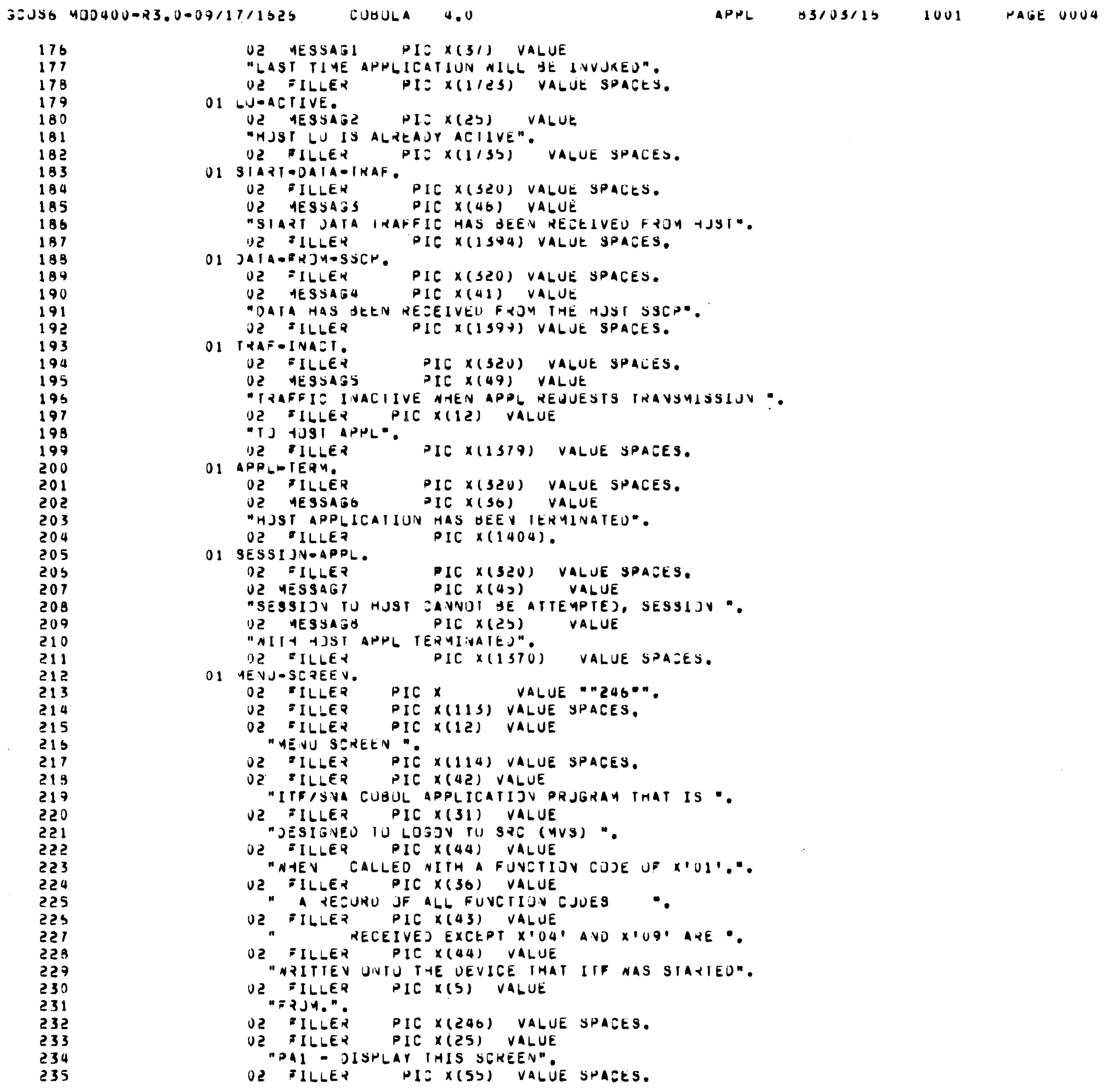
Figure C-1. Sample COBOL Application Program

```
GCOS6 MOD400-R3.0-09/17/1525     COBOLA   4.0                        APPL    83/03/15    1001    PAGE 0002

**   ----------------
**   \---------------\---------FUNCTION CODE - PASSED FROM ITF TO APPL
**      BYTE 1921
** X'01' = FIRST CALL TO APPLICATION
** X'02' = TRAFFIC ACTIVE - START DATA TRAFFIC RECEIVED FROM HOST
** X'03' = DATA IN SIB IS FROM SSCP
** X'04' = DATA IN SIB IS FROM HOST APPL
** X'05' = LU ALREADY IN SESSION - INITIATE SESSION REQUEST IGNORED
** X'06' = TRAFFIC INACTIVE - SIB NOT SEND TO HOST
** X'07' = SESSION WITH HOST APPL IS TERMINATED
** X'08' = TRANSMISSION TO HOST NOT HONERED - SESSION TERMINATED
** X'09' = DATA IN SIB IS FROM OPERATOR'S DISPLAY UNIT
** X'0A' = LU HAS BEEN DEACTIVATED BY HOST OR SNA OPERATOR
** X'0B' = SAME AS '0A' BUT EVENT OCCURED WHILE APPL WAS BUSY
**
**   ---------------
**   \--------------\--------AID KEY
**      BYTE 1922
**
**    ----------------
**    \---------------\--------LRN OF TERMINAL
**      BYTE 1923
**
**    --------------------------
**    \------------\------------\-------CURSOR POSITION
**      BYTES 1924 + 1925
**
**    -----------------
**    \----------------\-------RESERVED
**      BYTE 1926
**
**     -----------------
**     \----------------\-------HOST FUNCTION CODE
**      BYTE 1927      ERASE/WRITE - X'F5'
**                          OR
**                     WRITE - X'F1'
**
**      ----------------------------------
**     !        !        !        !      !----RH/RU DATA
**      -----------------------------------
**           BYTES 1928 - 1951
**
**
**
**      -------------------
**      !                 !--------LU ADDRESS (02 - 33)
**      -------------------
**         BYTE 1932
**
**
**    BYTES 1933 TO 1939 = RESERVED FOR FUTURE USE
**    BYTES 1940 TO 2047 = APDA - APPLICATION PROGRAM DATA AREA
**    CAN BE USED BY THE APPL AS A WORK AREA.
**    NO ADDRESSES HOWEVER SHOULD BE STORED, ONLY DATA
** \\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
*     FLOWCHART:
* LOG ONTO AN ITF TERMINAL WITH ITFMVS B.O. ATTACHED - WITHOUT AUTOLOGON
* HAVE MVS UP - ONCE LOGGED ON ITF SHOULD CALL US WITH A FC OF X'02'
* WE WILL STORE THE MVS LOGON STRING INTO THE SIB AND RETURN WITH A RC OF X'01'
* WE NOW CHECK THE FUNCTION CODE TO DETERMINE WHAT COURSE TO PERSUE.
*
** \\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\

*
  ENVIRONMENT DIVISION.
  CONFIGURATION SECTION.
  SOURCE-COMPUTER.
```

Figure C-1 (cont). Sample COBOL Application Program

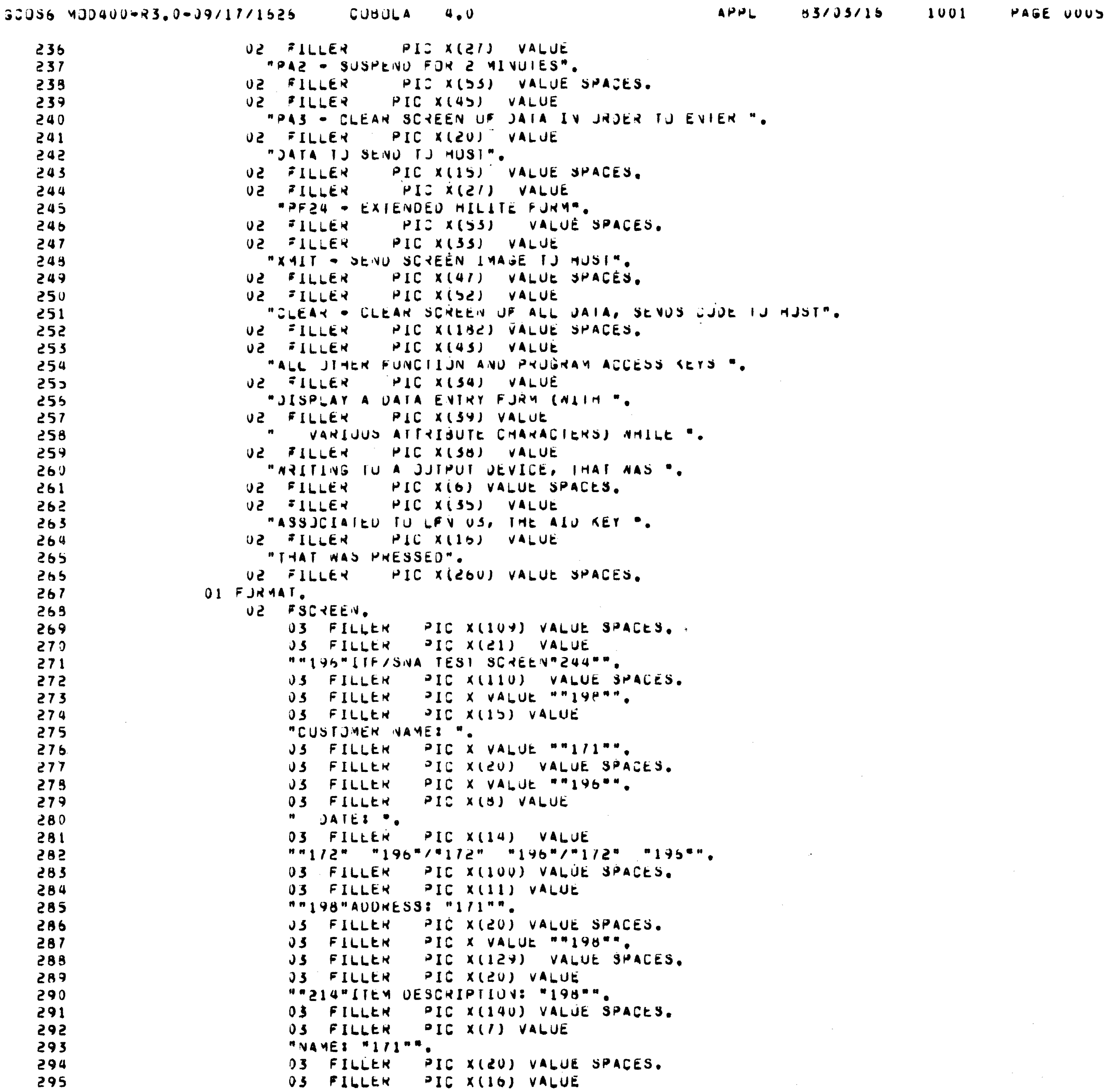
```
  116              HIS-SERIES-60 LEVEL-6.
  117          OBJECT-COMPUTER.
  118              HIS-SERIES-60 LEVEL-6.
  119        *
  120          INPUT-OUTPUT SECTION.
  121          FILE-CONTROL.
  122              SELECT PRINTFILE
  123                  ASSIGN TO OC-PRINTER
  124                  ORGANIZATION IS SEQUENTIAL
  125                  ACCESS MODE IS SEQUENTIAL
  126                  FILE STATUS IS PRT-STAT.
  127          DATA DIVISION.
  128          FILE SECTION.
  129          FD  PRINTFILE
  130                  BLOCK CONTAINS 1 RECORDS
  131                  LABEL RECORDS ARE OMITTED.
  132          01  PRT-REC           PIC  X(132).
  133          WORKING-STORAGE SECTION.
  134          77  PF1      PIC X  VALUE  ""242"".
  135          77  PF2      PIC X  VALUE  ""243"".
  136          77  PF3      PIC X  VALUE  ""244"".
  137          77  PF4      PIC X  VALUE  ""245"".
  138          77  PF5      PIC X  VALUE  ""246"".
  139          77  PF6      PIC X  VALUE  ""247"".
  140          77  PF7      PIC X  VALUE  ""248"".
  141          77  PF8      PIC X  VALUE  ""249"".
  142          77  PF9      PIC X  VALUE  ""250"".
  143          77  PF10     PIC X  VALUE  ""123"".
  144          77  PF11     PIC X  VALUE  ""124"".
  145          77  PF12     PIC X  VALUE  ""125"".
  146          77  PF13     PIC X  VALUE ""194"".
  147          77  PF14     PIC X  VALUE ""195"".
  148          77  PF15     PIC X  VALUE ""196"".
  149          77  PF16     PIC X  VALUE ""197"".
  150          77  PF17     PIC X  VALUE ""198"".
  151          77  PF18     PIC X  VALUE ""199"".
  152          77  PF19     PIC X  VALUE ""200"".
  153          77  PF20     PIC X  VALUE ""201"".
  154          77  PF21     PIC X  VALUE ""202"".
  155          77  PF22     PIC X  VALUE ""75"".
  156          77  PF23     PIC X  VALUE ""76"".
  157          77  PF24     PIC X  VALUE ""77"".
  158          77  PA1      PIC X  VALUE  ""109"".
  159          77  PA2      PIC X  VALUE ""111"".
  160          77  PA3      PIC X  VALUE  ""108"".
  161          77  XMIT     PIC X  VALUE  ""126"".
  162          77  CLEAR    PIC X  VALUE  ""110"".
  163          77  ZERO0           PIC X  VALUE ""01"".
  164          77  PRT-STAT        PIC XX  VALUE SPACES.
  165          77  TWO0            PIC X   VALUE ""5"".
  166          77  ONEE             PIC X    VALUE ""2"".
  167          77  BINVAL           COMP-1.
  168          77  HEXVAL           PIC X(4).
  169        *
  170        *
  171          01 MVS-LOGON.
  172              02  LOGON       PIC X(18)  VALUE
  173              "LOGON APPLID(CICS)".
  174              02  FILLER      PIC X(1742)  VALUE SPACES.
  175          01 LAST-TIME.
```

Figure C-1 (cont). Sample COBOL Application Program



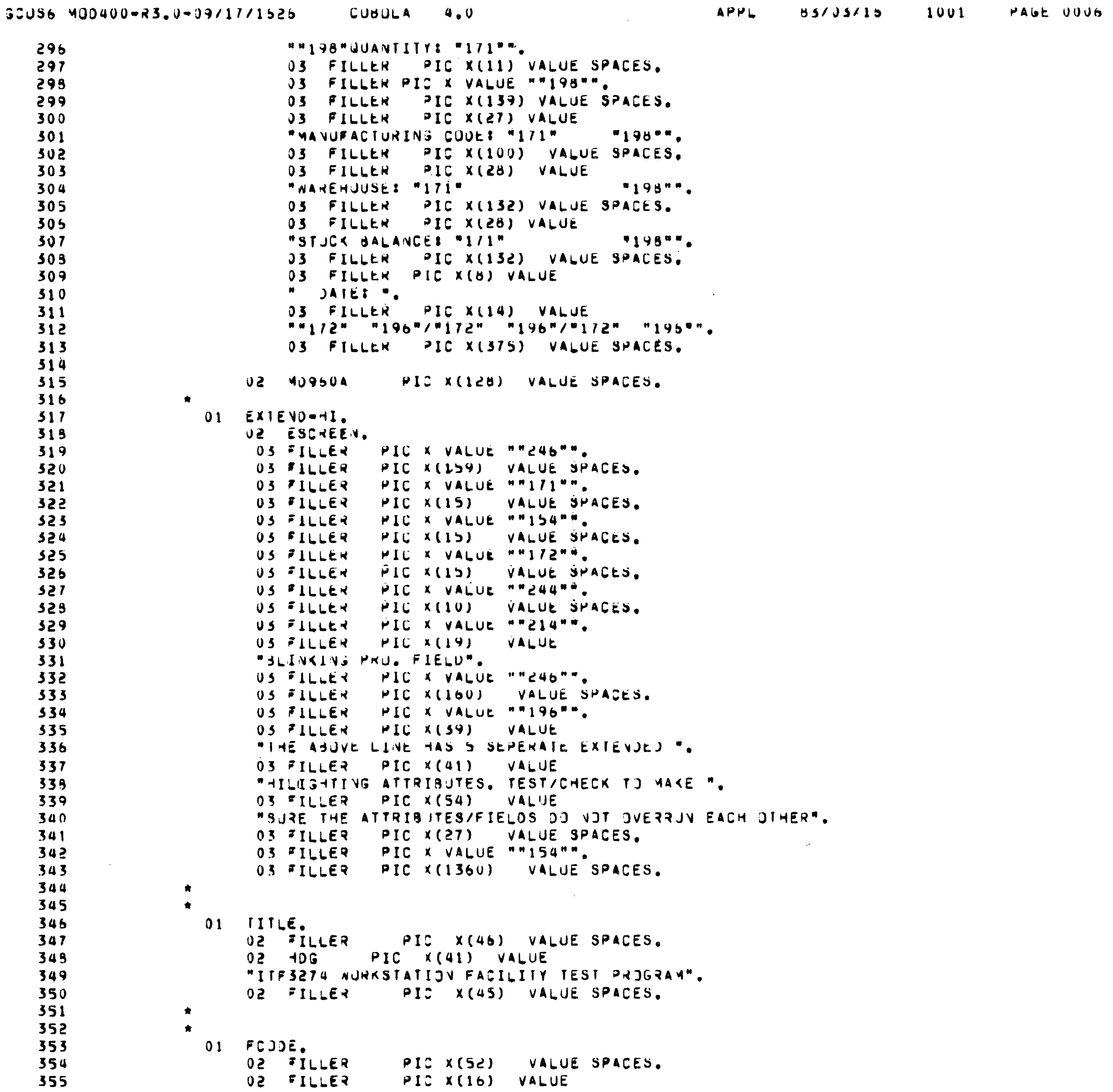
```
GCOS6 MOD400-R3.0-09/17/1525      COBOLA    4.0                       APPL    83/03/15    1001    PAGE 0004
                    02  MESSAG1     PIC X(37)  VALUE
                    "LAST TIME APPLICATION WILL BE INVOKED".
                    02  FILLER      PIC X(1723)  VALUE SPACES.
                01 LU-ACTIVE.
                    02  MESSAG2     PIC X(25)    VALUE
                    "HOST LU IS ALREADY ACTIVE".
                    02  FILLER      PIC X(1755)   VALUE SPACES.
                01 START-DATA-TRAF.
                    02  FILLER       PIC X(520) VALUE SPACES.
                    02  MESSAG3      PIC X(46)  VALUE
                    "START DATA TRAFFIC HAS BEEN RECEIVED FROM HOST".
                    02  FILLER       PIC X(1594) VALUE SPACES.
                01 DATA-FROM-SSCP.
                    02  FILLER       PIC X(520) VALUE SPACES.
                    02  MESSAG4      PIC X(41)  VALUE
                    "DATA HAS BEEN RECEIVED FROM THE HOST SSCP".
                    02  FILLER       PIC X(1599) VALUE SPACES.
                01 TRAF-INACT.
                    02  FILLER        PIC X(520)  VALUE SPACES.
                    02  MESSAG5       PIC X(49)  VALUE
                    "TRAFFIC INACTIVE WHEN APPL REQUESTS TRANSMISSION ".
                    02  FILLER     PIC X(12)  VALUE
                    "TO HOST APPL".
                    02  FILLER        PIC X(1579)  VALUE SPACES.
                01 APPL-TERM.
                    02  FILLER        PIC X(520)  VALUE SPACES.
                    02  MESSAG6       PIC X(36)   VALUE
                    "HOST APPLICATION HAS BEEN TERMINATED".
                    02  FILLER         PIC X(1404).
                01 SESSION-APPL.
                    02  FILLER         PIC X(520)  VALUE SPACES.
                    02 MESSAG7         PIC X(45)    VALUE
                    "SESSION TO HOST CANNOT BE ATTEMPTED, SESSION ".
                    02  MESSAG8        PIC X(25)    VALUE
                    "WITH HOST APPL TERMINATED".
                    02  FILLER         PIC X(1570)   VALUE SPACES.
                01 MENU-SCREEN.
                    02  FILLER    PIC X        VALUE ""246"".
                    02  FILLER    PIC X(113) VALUE SPACES.
                    02  FILLER    PIC X(12)  VALUE
                      "MENU SCREEN ".
                    02  FILLER    PIC X(114) VALUE SPACES.
                    02  FILLER    PIC X(42) VALUE
                      "ITF/SNA COBOL APPLICATION PROGRAM THAT IS ".
                    02  FILLER    PIC X(31)  VALUE
                      "DESIGNED TO LOGON TO SRC (MVS) ".
                    02  FILLER    PIC X(44)  VALUE
                      "WHEN   CALLED WITH A FUNCTION CODE OF X'01'.".
                    02  FILLER    PIC X(36)  VALUE
                      "  A RECORD OF ALL FUNCTION CODES    ".
                    02  FILLER    PIC X(43)  VALUE
                      "       RECEIVED EXCEPT X'04' AND X'09' ARE ".
                    02  FILLER    PIC X(44)  VALUE
                      "WRITTEN ONTO THE DEVICE THAT ITF WAS STARTED".
                    02  FILLER    PIC X(5)  VALUE
                      "FROM.".
                    02  FILLER    PIC X(246)   VALUE SPACES.
                    02  FILLER    PIC X(25)  VALUE
                      "PA1 - DISPLAY THIS SCREEN".
                    02  FILLER     PIC X(55)  VALUE SPACES.
```

Figure C-1 (cont). Sample COBOL Application Program



```
GCOS6 MOD400-R3.0-09/17/1525     COBOLA   4.0                    APPL     83/03/15     1001     PAGE 0005
                 02  FILLER      PIC X(27)  VALUE
                   "PA2 - SUSPEND FOR 2 MINUTES".
                 02  FILLER      PIC X(53)  VALUE SPACES.
                 02  FILLER    PIC X(45)  VALUE
                   "PA3 - CLEAR SCREEN OF DATA IN ORDER TO ENTER ".
                 02  FILLER    PIC X(20)  VALUE
                   "DATA TO SEND TO HOST".
                 02  FILLER    PIC X(15)  VALUE SPACES.
                 02  FILLER      PIC X(27)  VALUE
                    "PF24 - EXTENDED HILITE FORM".
                 02  FILLER      PIC X(53)   VALUE SPACES.
                 02  FILLER    PIC X(33)  VALUE
                   "XMIT - SEND SCREEN IMAGE TO HOST".
                 02  FILLER    PIC X(47)  VALUE SPACES.
                 02  FILLER    PIC X(52)  VALUE
                   "CLEAR - CLEAR SCREEN OF ALL DATA, SENDS CODE TO HOST".
                 02  FILLER    PIC X(182) VALUE SPACES.
                 02  FILLER    PIC X(43)  VALUE
                   "ALL OTHER FUNCTION AND PROGRAM ACCESS KEYS ".
                 02  FILLER    PIC X(34)  VALUE
                   "DISPLAY A DATA ENTRY FORM (WITH ".
                 02  FILLER    PIC X(39) VALUE
                   "   VARIOUS ATTRIBUTE CHARACTERS) WHILE ".
                 02  FILLER    PIC X(38)  VALUE
                   "WRITING TO A OUTPUT DEVICE, THAT WAS ".
                 02  FILLER    PIC X(6) VALUE SPACES.
                 02  FILLER    PIC X(35)  VALUE
                   "ASSOCIATED TO LFN 05, THE AID KEY ".
                 02  FILLER    PIC X(16)  VALUE
                   "THAT WAS PRESSED".
                 02  FILLER    PIC X(260) VALUE SPACES.
             01 FORMAT.
                 02  FSCREEN.
                     03  FILLER   PIC X(109) VALUE SPACES.
                     03  FILLER   PIC X(21)  VALUE
                     ""196"ITF/SNA TEST SCREEN"244"".
                     03  FILLER   PIC X(110)  VALUE SPACES.
                     03  FILLER   PIC X VALUE ""198"".
                     03  FILLER   PIC X(15) VALUE
                     "CUSTOMER NAME: ".
                     03  FILLER   PIC X VALUE ""171"".
                     03  FILLER   PIC X(20)  VALUE SPACES.
                     03  FILLER   PIC X VALUE ""196"".
                     03  FILLER   PIC X(8) VALUE
                     "  DATE: ".
                     03  FILLER   PIC X(14)  VALUE
                     ""172"  "196"/"172"  "196"/"172"  "195"".
                     03  FILLER   PIC X(100) VALUE SPACES.
                     03  FILLER   PIC X(11) VALUE
                     ""198"ADDRESS: "171"".
                     03  FILLER   PIC X(20) VALUE SPACES.
                     03  FILLER   PIC X VALUE ""198"".
                     03  FILLER   PIC X(129)  VALUE SPACES.
                     03  FILLER   PIC X(20) VALUE
                     ""214"ITEM DESCRIPTION: "198"".
                     03  FILLER   PIC X(140) VALUE SPACES.
                     03  FILLER   PIC X(7) VALUE
                     "NAME: "171"".
                     03  FILLER   PIC X(20) VALUE SPACES.
                     03  FILLER   PIC X(16) VALUE
```

Figure C-1 (cont). Sample COBOL Application Program



```
GCOS6 MOD400-R3.0-09/17/1526     COBOLA   4.0                     APPL    83/05/15    1001    PAGE 0006
                  ""198"QUANTITY: "171"".
                  03  FILLER   PIC X(11) VALUE SPACES.
                  03  FILLER PIC X VALUE ""198"".
                  03  FILLER   PIC X(139) VALUE SPACES.
                  03  FILLER   PIC X(27) VALUE
                  "MANUFACTURING CODE: "171"     "198"".
                  03  FILLER   PIC X(100)  VALUE SPACES.
                  03  FILLER   PIC X(28)  VALUE
                  "WAREHOUSE: "171"                "198"".
                  03  FILLER   PIC X(132) VALUE SPACES.
                  03  FILLER   PIC X(28) VALUE
                  "STOCK BALANCE: "171"            "198"".
                  03  FILLER   PIC X(132)  VALUE SPACES.
                  03  FILLER  PIC X(8) VALUE
                  "  DATE: ".
                  03  FILLER   PIC X(14)  VALUE
                  ""172"  "196"/"172"  "196"/"172"  "196"".
                  03  FILLER   PIC X(375)  VALUE SPACES.

              02  MD960A     PIC X(128)  VALUE SPACES.
      *
        01  EXTEND-HI.
            02  ESCREEN.
             03 FILLER   PIC X VALUE ""246"".
             03 FILLER   PIC X(159)  VALUE SPACES.
             03 FILLER   PIC X VALUE ""171"".
             03 FILLER   PIC X(15)   VALUE SPACES.
             03 FILLER   PIC X VALUE ""154"".
             03 FILLER   PIC X(15)   VALUE SPACES.
             03 FILLER   PIC X VALUE ""172"".
             03 FILLER   PIC X(15)   VALUE SPACES.
             03 FILLER   PIC X VALUE ""244"".
             03 FILLER   PIC X(10)   VALUE SPACES.
             03 FILLER   PIC X VALUE ""214"".
             03 FILLER   PIC X(19)   VALUE
             "BLINKING PRO. FIELD".
             03 FILLER   PIC X VALUE ""246"".
             03 FILLER   PIC X(160)   VALUE SPACES.
             03 FILLER   PIC X VALUE ""196"".
             03 FILLER   PIC X(39)   VALUE
             "THE ABOVE LINE HAS 5 SEPERATE EXTENDED ".
             03 FILLER   PIC X(41)   VALUE
             "HILIGHTING ATTRIBUTES. TEST/CHECK TO MAKE ".
             03 FILLER   PIC X(54)   VALUE
             "SURE THE ATTRIBUTES/FIELDS DO NOT OVERRUN EACH OTHER".
             03 FILLER   PIC X(27)   VALUE SPACES.
             03 FILLER   PIC X VALUE ""154"".
             03 FILLER   PIC X(1360)   VALUE SPACES.
      *
      *
        01  TITLE.
            02  FILLER      PIC  X(46)  VALUE SPACES.
            02  HDG     PIC  X(41)  VALUE
            "ITF3274 WORKSTATION FACILITY TEST PROGRAM".
            02  FILLER      PIC  X(45)  VALUE SPACES.
      *
      *
        01  FCODE.
            02  FILLER      PIC X(52)   VALUE SPACES.
            02  FILLER      PIC X(16)  VALUE
```

Figure C-1 (cont). Sample COBOL Application Program

```
3CJS6 MOD400-R3.0-09/17/1525     CUBULA   4.0                    APPL    83/J3/16    1001    PAGE 0007

            "FJNCTIJN CUDE = ".
            02  CODE1       PIC  X(3)  VALUE SPACES.
            02  FILLER       PIC  X(8)  VALUE
            " =  GJJD".
            02  FILLER       PIC  X(52)  VALUE SPACES.
      *
      *
      *
       LINKAGE SECTIJN.
       01  SI3.
            02  SI3A1       PIC  X(1760).
            02  SU3SI3      PIC  X(160).
            02  31920       PIC X.
            02  31921       PIC X.
            02  31922        PIC X.
            02  31923        PIC X.
            02  31924        PIC XX.
            02  31926        PIC X.
            02  31927         PIC X.
            02  SRSVD     PIC X(120).
      *
      *
      *
       PROCEDJRE DIVISION USING SI3.
       VENTS2.
                IF 31921 = ""2"" PERFORM SEND-HJST-MVS GO TJ LEAVE.
                IF 31921 = ""11"" PERFORM FC80A GJ TJ LEAVE.
                IF 31921 = ""12"" PERFORM FC808 GJ TJ LEAVE.
                IF 31921 = ""6"" PERFORM FC805 GO TJ LEAVE.
                IF 31921 = ""5"" PERFORM DISPLAY-SI3 GO TJ LEAVE.
                IF 31921 = ""3"" PERFORM FC802 GJ TJ LEAVE-1.
                IF 31921 = ""4"" PERFORM FC803 GJ TJ LEAVE-2.
                IF 31921 = ""7"" PERFORM FC806 GJ TJ LEAVE-1.
                IF 31921 = ""8"" PERFORM FC807 GJ TJ LEAVE-2.
                IF 31921 = ""9"" PERFORM FC808 GJ TJ LEAVE-1.
                 IF 31922 = PF1 MJVE "PF1" TO CJDE1
                 PERFORM PRINT GO TJ NEXTT.
                 IF 31922 = PF2 MJVE "PF2" TO CJDE1
                 PERFJRM PRINT GO TJ NEXTT.
                 IF 31922 = PF3 MJVE "PF3" TO CJDE1
                 PERFJRM PRINT GO TJ NEXTT.
                 IF 31922 = PF4 MJVE "PF4" TO CJDE1
                 PERFJRM PRINT GO TJ NEXTT.
                 IF 31922 = PF5 MJVE "PF5" TO CJDE1
                 PERFJRM PRINT GO TJ NEXTT.
                 IF 31922 = PF6 MJVE "PF6" TO CJDE1
                 PERFJRM PRINT GO TJ NEXTT.
                 IF 31922 = PF7 MJVE "PF7" TO CJDE1
                 PERFJRM PRINT GO TJ NEXTT.
                 IF 31922 = PF8 MJVE "PF8" TO CJDE1
                 PERFJRM PRINT GO TJ NEXTT.
                 IF 31922 = PF9 MJVE "PF9" TO CJDE1
                 PERFJRM PRINT GO TJ NEXTT.
                 IF 31922 = PF10 MOVE "P10" TJ CODE1
                 PERFJRM PRINT GO TJ NEXTT.
                 IF 31922 = PF11 MJVE "P11" TJ CODE1
                 PERFJRM PRINT GO TJ NEXTT.
                 IF 31922 = PF12 MJVE "P12" TJ CODE1
                 PERFJRM PRINT GO TJ NEXTT.
                 IF 31922 = PF13 MOVE "P13" TO CODE1
```

Figure C-1 (cont). Sample COBOL Application Program

```
GCOS6 MOD400-R3.0-09/17/1525      COBOLA   4.0                 APPL    83/05/15    1001    PAGE 0008

                       PERFORM PRINT GO TO NEXTT.
                       IF B1922 = PF14 MOVE "P14" TO CODE1
                       PERFORM PRINT GO TO NEXTT.
                       IF B1922 = PF15 MOVE "P15" TO CODE1
                       PERFORM PRINT GO TO NEXTT.
                       IF B1922 = PF16 MOVE "P16" TO CODE1
                       PERFORM PRINT GO TO NEXTT.
                       IF B1922 = PF17 MOVE "P17" TO CODE1
                       PERFORM PRINT GO TO NEXTT.
                       IF B1922 = PF18 MOVE "P18" TO CODE1
                       PERFORM PRINT GO TO NEXTT.
                       IF B1922 = PF19 MOVE "P19" TO CODE1
                       PERFORM PRINT GO TO NEXTT.
                       IF B1922 = PF20 MOVE "P20" TO CODE1
                       PERFORM PRINT GO TO NEXTT.
                       IF B1922 = PF21 MOVE "P21" TO CODE1
                       PERFORM PRINT GO TO NEXTT.
                       IF B1922 = PF22 MOVE "P22" TO CODE1
                       PERFORM PRINT GO TO NEXTT.
                       IF B1922 = PF23 MOVE "P23" TO CODE1
                       PERFORM PRINT GO TO NEXTT.
                       IF B1922 = PF24 MOVE "P24" TO CODE1
                       PERFORM PRINT  MOVE SPACES TO SIBA1
                        MOVE ESCREEN TO SIBA1
                        MOVE ""6"" TO B1920 GO TO LEAVE.
                       IF B1922 = PA1 MOVE "PA1" TO CODE1
                       PERFORM PRINT MOVE SPACES TO SIBA1
                       MOVE MENU-SCREEN TO SIBA1
                       MOVE ""6"" TO B1920 GO TO LEAVE.
                       IF B1922 = PA2 MOVE "PA2" TO CODE1
                       PERFORM PRINT
                       CALL "SUSP02"
                       GO TO NEXTT.
                       IF B1922 = PA3 MOVE "PA3" TO CODE1
                       PERFORM PRINT
                       MOVE SPACES TO SIBA1 SUBSIB
                       MOVE ""6"" TO B1920
                       GO TO LEAVE.
                       IF B1922 = XMIT MOVE "XMT" TO CODE1
                       PERFORM PRINT GO TO SEND-HOST.
                       IF B1922 = CLEAR MOVE "CLR" TO CODE1
                       PERFORM PRINT GO TO SEND-HOST.
                 *
                  NEXTT.
                      MOVE SPACES TO SIBA1.
                      MOVE FSCREEN TO SIBA1.
                      CALL "SOPIA" USING SIB.
                      MOVE ""5"" TO B1920.
                  LEAVE.
                      EXIT PROGRAM.
                  PRINT.
                      OPEN OUTPUT PRINTFILE.
                       MOVE SPACES TO PRT-REC MOVE FCODE TO PRT-REC.
                      WRITE PRT-REC AFTER ADVANCING 1.
                      CLOSE PRINTFILE.
                  SEND-HOST-MVS.
                      DISPLAY "FIRST CALL TO APPLICATION - LOGON - FC = 01".
                      OPEN OUTPUT PRINTFILE.
                      MOVE SPACES TO PRT-REC MOVE TITLE TO PRT-REC.
                      WRITE PRT-REC AFTER ADVANCING PAGE.
```

Figure C-1 (cont). Sample COBOL Application Program

```
SCOS6 MOD400-R5.0-09/17/1525      COBOLA   4.0                           APPL    83/03/16    1001    PAGE 0009

                   CLOSE PRINTFILE.
                   MOVE SPACES TO SIBA1.
                   MOVE MVS-LOGON TO SIBA1.
                   MOVE ""2"" TO B1920.
                   CALL "SOPIA" USING SIB.
               FC30A.
                   DISPLAY "LU DEACTIVATED - FC = 0A".
                   MOVE ""8"" TO B1920.
               FC30B.
                   DISPLAY "LU DEACTIVATED - FC = 0B".
                   MOVE ""8"" TO B1920.
               FC305.
                   DISPLAY "LU ALREADY IN SESSION WITH HOST - FC = 05".
                   MOVE SPACES TO SIBA1.
                   MOVE LU-ACTIVE TO SIBA1.
                   MOVE ""5"" TO B1920.
                   CALL "SOPIA" USING SIB.
               SEND-HOST.
                   MOVE ""4"" TO B1920.
                   GO TO LEAVE.
               DISPLAY-SIB.
                   MOVE ""6"" TO B1920.
               FC302.
                   DISPLAY "START DATA TRAFFIC RECEIVED FROM HOST - FC = 02".
                   MOVE SPACES TO SIBA1.
                   MOVE MENU-SCREEN TO SIBA1.
               FC303.
                   DISPLAY "DATA RECEIVED FROM HOST SSCP - FC = 03".
                   MOVE ""6"" TO B1920.
               FC306.
                   DISPLAY "TRAFFIC INACT FOR APPL TO REQUEST TRAN - FC = 06".
                   MOVE SPACES TO SIBA1.
                   MOVE TRAF-INACT TO SIBA1.
               FC307.
                   DISPLAY "HOST APPL HAS BEEN TERMINATED - FC = 07".
                   MOVE ""3"" TO B1920.
               FC308.
                   DISPLAY "SESSION TO HOST CANNOT BE ATTEMPTED - FC = 08".
                   MOVE SPACES TO SIBA1.
                   MOVE SESSION-APPL TO SIBA1.
               DISPLAY-MENU.
                   MOVE SPACES TO SIBA1.
                   MOVE MENU-SCREEN TO SIBA1.
                   MOVE ""5"" TO B1920.
                   CALL "SOPIA" USING SIB.
                   GO TO LEAVE-2.
               LEAVE-1.
                   MOVE SPACES TO SUBSIB.
                   MOVE ""6"" TO B1920.
                   CALL "SOPIA" USING SIB.
               LEAVE-2.
                   EXIT PROGRAM.
               END COBOL.

-> NO FATAL ERRORS; NO WARNINGS
```

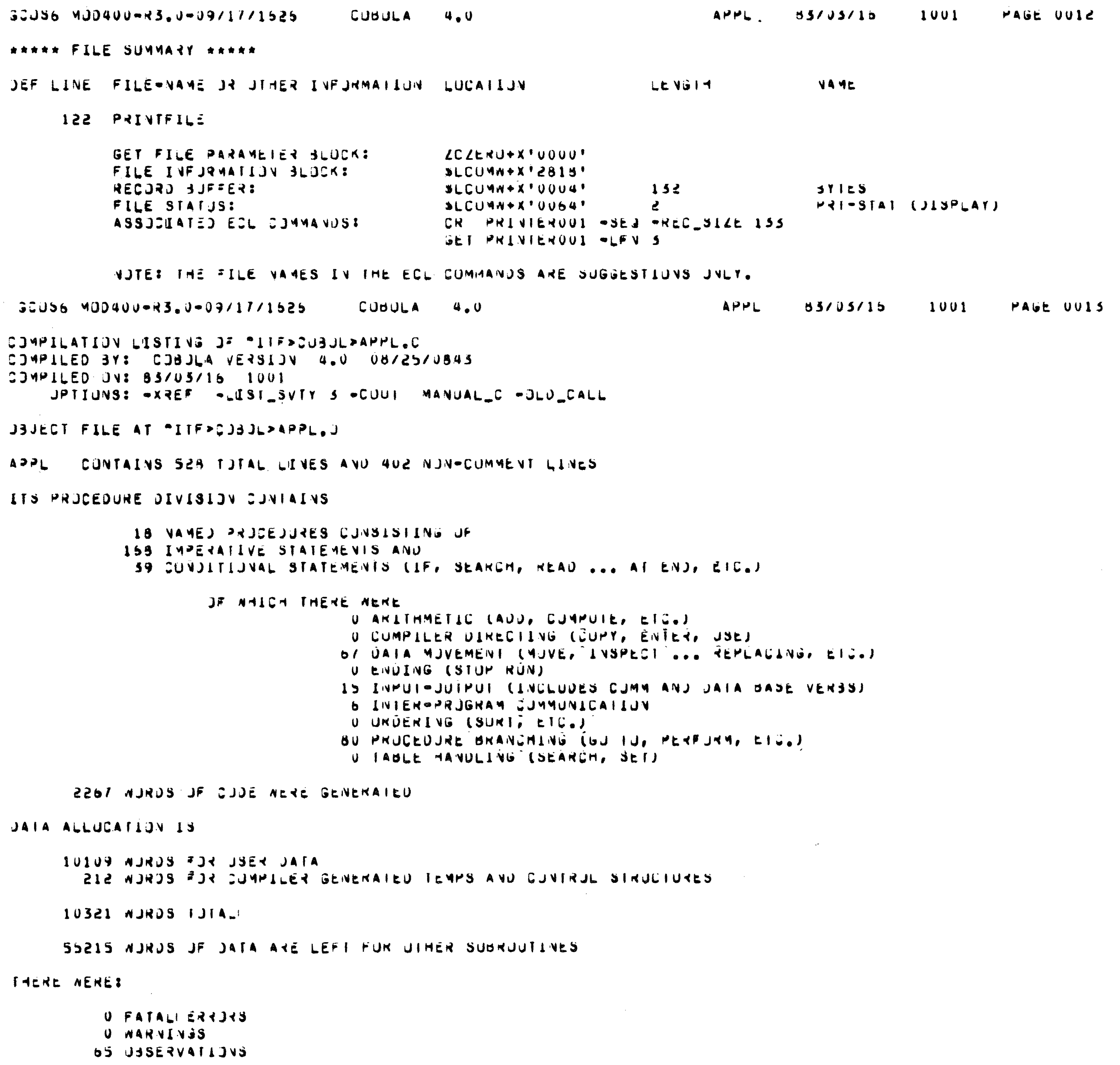
Figure C-1 (cont). Sample COBOL Application Program

```
GCOS6 MOD400-R3.0-09/17/1525     COBOLA   4.0                       APPL    83/03/15    1001    PAGE 0010

CROSS-REFERENCE LISTING (ALPHABETIC ORDER)

LN  IDENTIFIER           TYPE  OFFSET  USAGE/CLASS    RJ PICTURE          DEFINED     REF LINES

01  APPL-TERM            WORK  150A    GROUP ALPHANUM    X(1760)           DEF 200     NO REF
02  B1920                LINK  03C0    DSPLY ALPHANUM    X(1)              DEF 368     REF 440 444 452 465 479 483 485 491 494
                                                                                       497 504 511 519 524
02  B1921                LINK  03C0(8) DSPLY ALPHANUM    X(1)              DEF 369     REF 381 382 383 384 385 386 387 388 389
                                                                                       390
02  B1922                LINK  03C1    DSPLY ALPHANUM    X(1)              DEF 370     REF 391 393 395 397 399 401 403 405 407
                                                                                       409 411 413 415 417 419 421 423 425 427
                                                                                       429 431 433 435 437 441 443 449 454 456
02  B1923                LINK  03C1(8) DSPLY ALPHANUM    X(1)              DEF 371     NO REF
02  B1924                LINK  03C2    DSPLY ALPHANUM    X(2)              DEF 372     NO REF
02  B1926                LINK  03C3    DSPLY ALPHANUM    X(1)              DEF 373     NO REF
02  B1927                LINK  03C3(8) DSPLY ALPHANUM    X(1)              DEF 374     NO REF
77  BINVAL               WORK  0057    COMP-1 NUMERIC    S9(5)  BIN(15)    DEF 167     NO REF
77  CLEAR                WORK  0062    DSPLY ALPHANUM    X(1)              DEF 162     REF 456
02  CODE1                WORK  2756    DSPLY ALPHANUM    X(5)              DEF 357     REF 391 393 395 397 399 401 403 405 407
                                                                                       409 411 413 415 417 419 421 423 425 427
                                                                                       429 431 433 435 437 441 443 449 454 456
01  DATA-FROM-SSCP       WORK  0E2A    GROUP ALPHANUM    X(1760)           DEF 188     NO REF
    DISPLAY-MENU         PROC          PARAGRAPH NAME                      DEF 516     NO REF
    DISPLAY-SI3          PROC          PARAGRAPH NAME                      DEF 495     REF 385
02  ESCREEN              WORK  2330    GROUP ALPHANUM    X(1923)           DEF 318     REF 459
01  EXTEND-HI            WORK  2330    GROUP ALPHANUM    X(1923)           DEF 317     NO REF
    FC802                PROC          PARAGRAPH NAME                      DEF 498     REF 386
    FC803                PROC          PARAGRAPH NAME                      DEF 502     REF 387
    FC805                PROC          PARAGRAPH NAME                      DEF 487     REF 384
    FC806                PROC          PARAGRAPH NAME                      DEF 505     REF 388
    FC807                PROC          PARAGRAPH NAME                      DEF 509     REF 389
    FC808                PROC          PARAGRAPH NAME                      DEF 512     REF 390
    FC80A                PROC          PARAGRAPH NAME                      DEF 481     REF 382
    FC80B                PROC          PARAGRAPH NAME                      DEF 484     REF 383
01  FCODE                WORK  2754    GROUP ALPHANUM    X(131)            DEF 355     REF 468
01  FORMAT               WORK  1F80    GROUP ALPHANUM    X(1887)           DEF 267     NO REF
02  FSCREEN              WORK  1F80    GROUP ALPHANUM    X(1759)           DEF 268     REF 461
02  HDG                  WORK  2709    DSPLY ALPHANUM    X(41)             DEF 348     NO REF
77  HEXVAL               WORK  0068    DSPLY ALPHANUM    X(4)              DEF 168     NO REF
01  LAST-TIME            WORK  030A    GROUP ALPHANUM    X(1760)           DEF 173     NO REF
    LEAVE                PROC          PARAGRAPH NAME                      DEF 464     REF 381 382 383 384 385 440 444 453 495
    LEAVE-1              PROC          PARAGRAPH NAME                      DEF 522     REF 386 388 390
    LEAVE-2              PROC          PARAGRAPH NAME                      DEF 525     REF 387 389 521
02  LOGON                WORK  006A    DSPLY ALPHANUM    X(18)             DEF 172     NO REF
01  LO-ACTIVE            WORK  074A    GROUP ALPHANUM    X(1760)           DEF 179     REF 490
02  MD960A               WORK  22EF(8) DSPLY ALPHANUM    X(128)            DEF 315     NO REF
01  MENU-SCREEN          WORK  18EA    GROUP ALPHANUM    X(1836)           DEF 212     REF 443 501 518
02  MESSAG1              WORK  030A    DSPLY ALPHANUM    X(37)             DEF 176     NO REF
02  MESSAG2              WORK  074A    DSPLY ALPHANUM    X(25)             DEF 180     NO REF
02  MESSAG3              WORK  085A    DSPLY ALPHANUM    X(46)             DEF 185     NO REF
02  MESSAG4              WORK  0ECA    DSPLY ALPHANUM    X(41)             DEF 190     NO REF
02  MESSAG5              WORK  123A    DSPLY ALPHANUM    X(49)             DEF 195     NO REF
02  MESSAG6              WORK  15AA    DSPLY ALPHANUM    X(56)             DEF 202     NO REF
02  MESSAG7              WORK  191A    DSPLY ALPHANUM    X(45)             DEF 207     NO REF
02  MESSAG8              WORK  1950(8) DSPLY ALPHANUM    X(25)             DEF 209     NO REF
01  MVS-LOGON            WORK  006A    GROUP ALPHANUM    X(1760)           DEF 171     REF 478
    NEWTS2               PROC          PARAGRAPH NAME                      DEF 380     NO REF
    NEXIT                PROC          PARAGRAPH NAME                      DEF 459     REF 392 394 396 398 400 402 404 406 408
                                                                                       410 412 414 416 418 420 422 424 426 428
                                                                                       430 432 434 436 446
```

Figure C-1 (cont). Sample COBOL Application Program

```
03056 MOD400-R3.0-09/17/1525     COBOLA   4.0                    APPL    83/03/15    1001    PAGE 0011

CROSS-REFERENCE LISTING (ALPHABETIC ORDER)

LN  IDENTIFIER              TYPE  OFFSET  USAGE/CLASS     RJ PICTURE        DEFINED      REF LINES

77  ONEE                    WORK  0066    DSPLY ALPHANUM     X(1)           DEF 166      NO REF
77  PA1                     WORK  005E    DSPLY ALPHANUM     X(1)           DEF 158      REF 441
77  PA2                     WORK  005F    DSPLY ALPHANUM     X(1)           DEF 159      REF 445
77  PA3                     WORK  0060    DSPLY ALPHANUM     X(1)           DEF 160      REF 449
77  PF1                     WORK  0046    DSPLY ALPHANUM     X(1)           DEF 134      REF 391
77  PF10                    WORK  004F    DSPLY ALPHANUM     X(1)           DEF 143      REF 409
77  PF11                    WORK  0050    DSPLY ALPHANUM     X(1)           DEF 144      REF 411
77  PF12                    WORK  0051    DSPLY ALPHANUM     X(1)           DEF 145      REF 413
77  PF13                    WORK  0052    DSPLY ALPHANUM     X(1)           DEF 146      REF 415
77  PF14                    WORK  0053    DSPLY ALPHANUM     X(1)           DEF 147      REF 417
77  PF15                    WORK  0054    DSPLY ALPHANUM     X(1)           DEF 148      REF 419
77  PF16                    WORK  0055    DSPLY ALPHANUM     X(1)           DEF 149      REF 421
77  PF17                    WORK  0056    DSPLY ALPHANUM     X(1)           DEF 150      REF 423
77  PF18                    WORK  0057    DSPLY ALPHANUM     X(1)           DEF 151      REF 425
77  PF19                    WORK  0058    DSPLY ALPHANUM     X(1)           DEF 152      REF 427
77  PF2                     WORK  0047    DSPLY ALPHANUM     X(1)           DEF 135      REF 393
77  PF20                    WORK  0059    DSPLY ALPHANUM     X(1)           DEF 153      REF 429
77  PF21                    WORK  005A    DSPLY ALPHANUM     X(1)           DEF 154      REF 431
77  PF22                    WORK  005B    DSPLY ALPHANUM     X(1)           DEF 155      REF 433
77  PF23                    WORK  005C    DSPLY ALPHANUM     X(1)           DEF 156      REF 435
77  PF24                    WORK  005D    DSPLY ALPHANUM     X(1)           DEF 157      REF 437
77  PF3                     WORK  0048    DSPLY ALPHANUM     X(1)           DEF 136      REF 395
77  PF4                     WORK  0049    DSPLY ALPHANUM     X(1)           DEF 137      REF 397
77  PF5                     WORK  004A    DSPLY ALPHANUM     X(1)           DEF 138      REF 399
77  PF6                     WORK  004B    DSPLY ALPHANUM     X(1)           DEF 139      REF 401
77  PF7                     WORK  004C    DSPLY ALPHANUM     X(1)           DEF 140      REF 403
77  PF8                     WORK  004D    DSPLY ALPHANUM     X(1)           DEF 141      REF 405
77  PF9                     WORK  004E    DSPLY ALPHANUM     X(1)           DEF 142      REF 407
    PRINT                   PROC          PARAGRAPH NAME                    DEF 466      REF 392 394 396 398 400 402 404 406 408
                                                                                         410 412 414 416 418 420 422 424 426 428
                                                                                         430 432 434 436 438 442 446 450 455 457
FD  PRINTFILE               FILE  0004    SEQ/SEQ PRINTER    X(132)         DEF 129      REF 467 470 473 476
01  PRT-REC                 FILE  0004    DSPLY ALPHANUM     X(132)         DEF 132      REF 465 468 469 474 474 475
77  PRT-STAT                WORK  0064    DSPLY ALPHANUM     X(2)           DEF 164      NO REF
    SEND-HOST               PROC          PARAGRAPH NAME                    DEF 493      REF 455 457
    SEND-HOST-MVS           PROC          PARAGRAPH NAME                    DEF 471      REF 381
01  SESSION-APPL            WORK  187A    GROUP ALPHANUM     X(1760)        DEF 203      REF 515
01  SIB                     LINK  0000    GROUP ALPHANUM     X(2048)        DEF 365      REF 379 462 480 492 520 525
02  SIBAI                   LINK  0000    DSPLY ALPHANUM     X(1760)        DEF 366      REF 438 439 442 443 451 460 461 477 478
                                                                                         489 490 500 501 507 508 514 515 517 518

02  SRSVD                   LINK  05C4    DSPLY ALPHANUM     X(120)         DEF 375      NO REF
01  START-DATA-TRAF         WORK  0A3A    GROUP ALPHANUM     X(1760)        DEF 183      NO REF
02  SUBSIB                  LINK  0370    DSPLY ALPHANUM     X(160)         DEF 367      REF 451 523
01  TITLE                   WORK  26F2    GROUP ALPHANUM     X(132)         DEF 346      REF 474
01  TRAF-INACT              WORK  119A    GROUP ALPHANUM     X(1760)        DEF 193      REF 508
77  TWOO                    WORK  0065    DSPLY ALPHANUM     X(1)           DEF 165      NO REF
77  XMIT                    WORK  0061    DSPLY ALPHANUM     X(1)           DEF 161      REF 454
77  ZEROO                   WORK  0063    DSPLY ALPHANUM     X(1)           DEF 163      NO REF
```

Figure C-1 (cont). Sample COBOL Application Program



```
SCOS6 MOD400-R3.0-09/17/1525     COBOLA   4.0                        APPL    83/05/16    1001    PAGE 0012

***** FILE SUMMARY *****

DEF LINE  FILE-NAME OR OTHER INFORMATION  LOCATION            LENGTH          NAME

     122  PRINTFILE

          GET FILE PARAMETER BLOCK:       ZCZERO+X'0000'
          FILE INFORMATION BLOCK:         $LCOMM+X'2818'
          RECORD BUFFER:                  $LCOMM+X'0004'      132             BYTES
          FILE STATUS:                    $LCOMM+X'0064'      2               PRT-STAT (DISPLAY)
          ASSOCIATED ECL COMMANDS:        CR  PRINTER001 -SEQ -REC_SIZE 133
                                          GET PRINTER001 -LFN 3

          NOTE: THE FILE NAMES IN THE ECL COMMANDS ARE SUGGESTIONS ONLY.
 SCOS6 MOD400-R3.0-09/17/1525     COBOLA   4.0                        APPL    83/05/16    1001    PAGE 0013

COMPILATION LISTING OF ^ITF>COBOL>APPL.C
COMPILED BY:  COBOLA VERSION  4.0  08/25/0843
COMPILED ON: 83/05/16  1001
    OPTIONS: -XREF  -LIST_SVTY 3 -COUT  MANUAL_C -OLD_CALL

OBJECT FILE AT ^ITF>COBOL>APPL.O

APPL   CONTAINS 528 TOTAL LINES AND 402 NON-COMMENT LINES

ITS PROCEDURE DIVISION CONTAINS

            18 NAMED PROCEDURES CONSISTING OF
           158 IMPERATIVE STATEMENTS AND
            59 CONDITIONAL STATEMENTS (IF, SEARCH, READ ... AT END, ETC.)

                 OF WHICH THERE WERE
                               0 ARITHMETIC (ADD, COMPUTE, ETC.)
                               0 COMPILER DIRECTING (COPY, ENTER, USE)
                              67 DATA MOVEMENT (MOVE, INSPECT ... REPLACING, ETC.)
                               0 ENDING (STOP RUN)
                              15 INPUT-OUTPUT (INCLUDES COMM AND DATA BASE VERBS)
                               6 INTER-PROGRAM COMMUNICATION
                               0 ORDERING (SORT, ETC.)
                              80 PROCEDURE BRANCHING (GO TO, PERFORM, ETC.)
                               0 TABLE HANDLING (SEARCH, SET)

      2267 WORDS OF CODE WERE GENERATED

DATA ALLOCATION IS

     10109 WORDS FOR USER DATA
       212 WORDS FOR COMPILER GENERATED TEMPS AND CONTROL STRUCTURES

     10321 WORDS TOTAL

     55215 WORDS OF DATA ARE LEFT FOR OTHER SUBROUTINES

THERE WERE:

         0 FATAL ERRORS
         0 WARNINGS
        65 OBSERVATIONS
```

Figure C-1 (cont). Sample COBOL Application Program

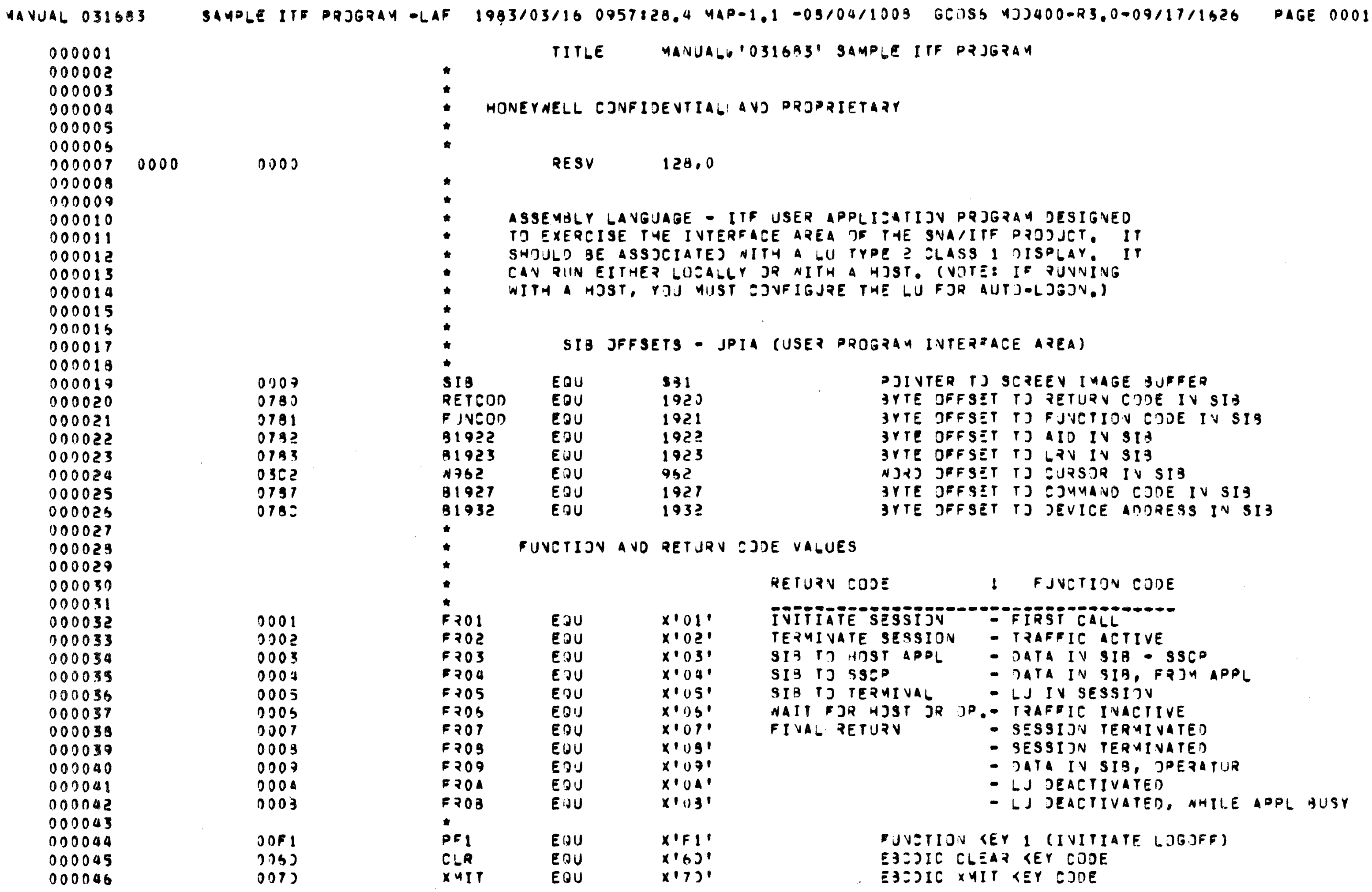
Appendix D
SAMPLE ITF ASSEMBLY LANGUAGE APPLICATION PROGRAM

Figure D-1 shows a sample of a logical unit class 1 local mode assembly language program (whose bound unit and program name is "LOCAL"), executing on the DPS 6, that creates a formatted data entry form to be displayed on a WST/VIP terminal screen. A WST/VIP terminal operator can then enter (or modify) data on the form. This sample program (LOCAL) illustrates the use of various ITF function and return codes. This program will display on the user terminal the EBCDIC code for the function key struck by the operator. Function Key 1 (PF-1) is interpreted by this program to initiate a logoff.

Comparable display forms can be created by a host application program and sent to the Interactive Terminal Facility (ITF).

NOTE

For a detailed description of the SIB, and other material pertinent to writing an assembly language program, see Section 4 of this manual.



```
MANUAL 031683     SAMPLE ITF PROGRAM -LAF  1983/03/16 0957:28.4 MAP-1.1 -08/04/1008  GCOS6 MOD400-R3.0-09/17/1626   PAGE 0001

                                   TITLE     MANUAL,'031683' SAMPLE ITF PROGRAM
                           *
                           *
                           *   HONEYWELL CONFIDENTIAL AND PROPRIETARY
                           *
                           *
       0000     0000               RESV      128,0
                           *
                           *
                           *     ASSEMBLY LANGUAGE - ITF USER APPLICATION PROGRAM DESIGNED
                           *     TO EXERCISE THE INTERFACE AREA OF THE SNA/ITF PRODUCT.  IT
                           *     SHOULD BE ASSOCIATED WITH A LU TYPE 2 CLASS 1 DISPLAY.  IT
                           *     CAN RUN EITHER LOCALLY OR WITH A HOST. (NOTE: IF RUNNING
                           *     WITH A HOST, YOU MUST CONFIGURE THE LU FOR AUTO-LOGON.)
                           *
                           *
                           *          SIB OFFSETS - UPIA (USER PROGRAM INTERFACE AREA)
                           *
                0009       SIB     EQU       $B1                  POINTER TO SCREEN IMAGE BUFFER
                0780       RETCOD  EQU       1920                 BYTE OFFSET TO RETURN CODE IN SIB
                0781       FUNCOD  EQU       1921                 BYTE OFFSET TO FUNCTION CODE IN SIB
                0782       B1922   EQU       1922                 BYTE OFFSET TO AID IN SIB
                0783       B1923   EQU       1923                 BYTE OFFSET TO LRN IN SIB
                03C2       W962    EQU       962                  WORD OFFSET TO CURSOR IN SIB
                0787       B1927   EQU       1927                 BYTE OFFSET TO COMMAND CODE IN SIB
                078C       B1932   EQU       1932                 BYTE OFFSET TO DEVICE ADDRESS IN SIB
                           *
                           *      FUNCTION AND RETURN CODE VALUES
                           *
                           *                            RETURN CODE          !   FUNCTION CODE
                           *
                0001       FR01    EQU       X'01'      -------------------------------------
                                                        INITIATE SESSION    - FIRST CALL
                0002       FR02    EQU       X'02'      TERMINATE SESSION   - TRAFFIC ACTIVE
                0003       FR03    EQU       X'03'      SIB TO HOST APPL    - DATA IN SIB - SSCP
                0004       FR04    EQU       X'04'      SIB TO SSCP         - DATA IN SIB, FROM APPL
                0005       FR05    EQU       X'05'      SIB TO TERMINAL     - LU IN SESSION
                0006       FR06    EQU       X'06'      WAIT FOR HOST OR OP.- TRAFFIC INACTIVE
                0007       FR07    EQU       X'07'      FINAL RETURN        - SESSION TERMINATED
                0008       FR08    EQU       X'08'                          - SESSION TERMINATED
                0009       FR09    EQU       X'09'                          - DATA IN SIB, OPERATOR
                000A       FR0A    EQU       X'0A'                          - LU DEACTIVATED
                000B       FR0B    EQU       X'0B'                          - LU DEACTIVATED, WHILE APPL BUSY
                           *
                00F1       PF1     EQU       X'F1'                FUNCTION KEY 1 (INITIATE LOGOFF)
                006D       CLR     EQU       X'6D'                EBCDIC CLEAR KEY CODE
                007D       XMIT    EQU       X'7D'                EBCDIC XMIT KEY CODE
```

Figure D-1. Sample Assembly Language Program

```
MANUAL 031683     SAMPLE ITF PROGRAM -LAF  1983/03/16 0957:28.4 MAP-1.1 -08/04/1008  GCOS6 MOD400-R3.0-09/17/1626   PAGE 0002

  000047                          /
  000048                          *
  000049                          *
  000050                          *         TASK REQUEST BLOCK DISPLACEMENTS MACRO
  000051                          *
  000052                          ***** GCOS6-1 TASK REQUEST BLOCK DISPLACEMENTS *** 1979/08/14 *********
  000053                          *                           '\0\1\2\3\4\5\6\7\8\9\A\B\C\D\E\F\'      *
  000054            FFFE          T_RRB      EQU      -$AF          '\      RETURN RB (R=1)          \'   *
  000055            FFFF          T_SEM      EQU      -1            '\      SEMAPHORE (S=1)          \'   *
  000056            0000          T_RS       EQU      0             '\      RETURN STATUS            \'   *
  000057            0002          T_CT1      EQU      $AF           '\      RS       \T\W\U\S\ \R\D\I\'   *
  000058            0003          T_CT2      EQU      T_CT1+1       '\      LRN      \ \0\   \0\0\0\0\'   *
  000059            0004          T_ADR      EQU      T_CT2+1       '\      START ADDRESS (I=0)      \'   *
  000060            0005          T_PRM      EQU      T_ADR+$AF     '\      START OF PARAMETER BLOCK \'   *
  000061                          * CONTROL WORD 1 (CT1) BIT MASKS                                        *
  000062            0080          MT_T       EQU      X'80'                  NOT TERMINATED BIT           *
  000063            0040          MT_W       EQU      X'40'                  DON'T WAIT BIT               *
  000064            0020          MT_U       EQU      X'20'                  USER BIT                     *
  000065            0010          MT_S       EQU      X'10'                  POST SEMAPHORE BIT           *
  000066            0004          MT_R       EQU      4                      RETURN RB INDICATOR          *
  000067            0002          MT_D       EQU      2                      DELETE TRB (RTN TO MEM POOL) *
  000068            0001          MT_I       EQU      1                      I/O BIT (ABSENCE OF START AD) *
  000069                          ***********************************************************************

MANUAL 031683     SAMPLE ITF PROGRAM -LAF  1983/03/16 0957:28.4 MAP-1.1 -08/04/1008  GCOS6 MOD400-R3.0-09/17/1626   PAGE 0003

  000070                          /
  000071                          *
  000072                          *
  000073            0080          APPL       EQU      $
  000074                          *
  000075                          *   GET ADDRESS OF SIB FROM TASK REQUEST BLOCK
  000076                          *
  000077  0080  9CC4 0007                    LDB      SIB,$B4.T_ADR+$AF+1          ADD. OF SIB
  000078  0082  A870 0781                    LDR      $R2,=FUNCOD                  FUN. CODE BYTE OFFSET
  000079  0084  92A1                         LLH      $R1,SIB.$R2                  GET FUNCTION CODE
  000080  0085  A3C0 0003                    LAB      $B2,FCNTBL                   ADD. OF BRANCH TABLE
  000081  0087  0C92                         LDB      $B5,$B2.$R1                  ADDRESS OF ROUNTINE
  000082  0088  8385                         JMP      $B5                          GO TO IT.....
  000083                          *
  000084                          *
  000085                          *
  000086            0089          FCNTBL     EQU      $
  000087  0089  0000 00A1                    DC       <FC00                INVALID FUNCTION CODE
  000088  008B  0000 00CB                    DC       <FC01                FIRST CALL INT --> APPLICATION
  000089  008D  0000 00CE                    DC       <FC02                TRAFFIC ACTIVE (NOT USED IN LOCAL MODE)
  000090  008F  0000 00F1                    DC       <FC03                DATA FROM SSCP (NOT USED IN LOCAL MODE)
  000091  0091  0000 00F3                    DC       <FC04                DATA FROM PLU (NOT USED IN LOCAL MODE)
  000092  0093  0000 00F5                    DC       <FC05                ALREADY IN SESSION (NOT USED IN LOCAL MODE)
  000093  0095  0000 010F                    DC       <FC06                TRAFFIC INACTIVE (NOT USED IN LOCAL MODE)
  000094  0097  0000 0117                    DC       <FC07                SESSION TERMINATED (NOT USED IN LOCAL MODE)
  000095  0099  0000 0117                    DC       <FC08                SESSION TERMINATED (NOT USED IN LOCAL MODE)
  000096  009B  0000 0853                    DC       <FC09                DATA FROM DISPLAY
  000097  009D  0000 0873                    DC       <FC0A                LU DEACTIVATED (OPERATOR LOGOFF/STOP IN LOCAL MODE)
  000098  009F  0000 0895                    DC       <FC0B                LU DEACTIVATED (NOT USED IN LOCAL MODE)
```

Figure D-1 (cont). Sample Assembly Language Program

```
MANUAL 031683     SAMPLE ITF PROGRAM -LAF  1983/03/16 0957:28.4 MAP-1.1 -08/04/1008  GCOS6 MOD400-R3.0-09/17/1626   PAGE 0004
000099                          /
000100                          *
000101                          ********************
000102                          *
000103                          *    FC00:   INVALID FUNCTION CODE
000104                          *
000105                          ********************
000106                          *
000107                 00A1     FC00      EQU       $
000108  00A1  C3C0 000C                   LAB       $B4,ERRMS1                 ERROR MESSAGE TEXT
000109  00A3  E870 0034                   LDR       $R6,=ERRML1                ERROR MESSAGE LENGTH
000110  00A5  7C00                        LDV       $R7,0                      LEFT BYTE
000111                          *                   WRITE TO USER OUT
000112  00A6  0001                        MCL
000113  00A7  0801                        DC        X'0801'
000114                          *
000115  00A8  A870 0780                   LDR       $R2,=RETCOD                BYTE OFFSET OF RETURN CODE
000116  00AA  1C07                        LDV       $R1,=FR07                  FINAL RETURN RC
000117  00AB  97A1                        STH       $R1,$B3.$R2                STORE INTO $B3
000118  00AC  0F81 0847                   B         EXIT1                      GO TERMINATE
000119                          *
000120                          *
000121  00AE  4120              ERRMS1    TEXT      'A INVALID FUNCTION CODE(00) - APPLICATION TERMINATED'
        00AF  494E
        00B0  5641
        00B1  4C49
        00B2  4420
        00B3  4655
        00B4  4E43
        00B5  5449
        00B6  4F4E
        00B7  2043
        00B8  4F44
        00B9  4528
        00BA  3030
        00BB  2920
        00BC  2D20
        00BD  4150
        00BE  504C
        00BF  4943
        00C0  4154
        00C1  494F
        00C2  4E20
        00C3  5445
        00C4  524D
        00C5  494E
        00C6  4154
        00C7  4544
000122                 0034     ERRML1    EQU       ($-ERRMS1)*2        MESSAGE LENGTH
000123
000124                          *
000125                          ****************
000126                          *
000127                          *   FC01:  INITIAL CALL FROM ITF
```

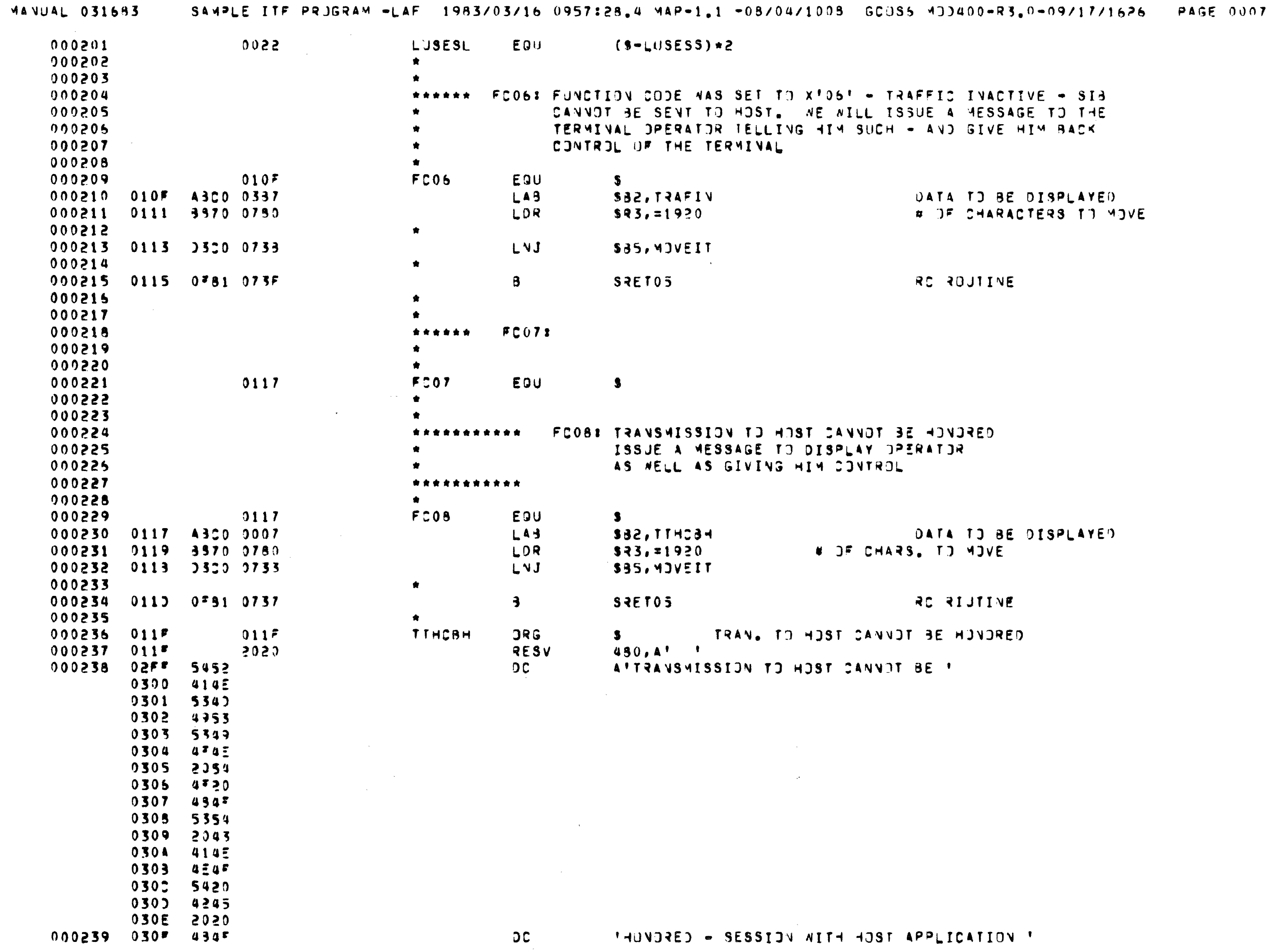
Figure D-1 (cont). Sample Assembly Language Program

```
MANUAL 031683     SAMPLE ITF PROGRAM -LAF  1983/03/16 0957:28.4 MAP-1.1 -08/04/1008  GCOS6 MOD400-R3.0-09/17/1626   PAGE 0005
000128                             *          PUT A RETURN CODE OF X'01' IN THE SIB
000129                             *          ITF SHOULD SEND THE AUTO LOGON STRING FROM
000130                             *          THE CONFIGURATION FILE
000131                             *
000132                             ****************
000133                             *
000134                             *
000135                 00C8        FC01      EQU      $
000136  00C8  A870  0780                     LDR      $R2,=RETCOD                 BYTE OFFSET OF RETURN CODE
000137  00CA  1C01                           LDV      $R1,=FR01                   INITIATE SESSION RC
000138  00CB  97A1                           STH      $R1,SIB.$R2                 STORE IT INTO THE SIB
000139  00CC  0F81  0827                     B        EXIT1                       RETURN TO ITF
000140                             *
000141                             *
000142                             ****************
000143                             *
000144                             *  FC02: START DATA TRAFFIC RECEIVED
000145                             *        WILL DISPLAY A FORMATTED SCREEN
000146                             *
000147                             ****************
000148                             *
000149                 00CE        FC02      EQU      $
000150  00CE  C3C0  000E                     LAB      $B4,MESS1T                  MESSAGE TEXT
000151  00D0  E870  0028                     LDR      $R6,=MESS1L                 MESSAGE LENGTH
000152  00D2  7C00                           LDV      $R7,0                       LEFT BYTE
000153                             *                  WRITE TO USER OUT
000154  00D3  0001                           MCL
000155  00D4  0801                           DC       X'0801'
000156                             *
000157                             *
000158  00D5  D3C0  0821                     LNJ      $B5,FORMAT                  TRANSFER FORMATTED IMAGE TO SCREEN
000159                             *
000160  00D7  A870  0780                     LDR      $R2,=RETCOD                 BYTE OFFSET OF RC
000161  00D9  1C05                           LDV      $R1,=FR05                   DISPLAY SIB RC
000162  00DA  97A1                           STH      $R1,SIB.$R2                 STORE RETURN CODE
000163                             *
000164  00DB  0F81  0818                     B        EXIT1                       TERMINATE BACK TO ITF
000165                             *
000166  00DD  4120                 MESS1T    TEXT     'A START DATA TRAFFIC RECEIVED FROM HOST'
        00DE  5354
        00DF  4152
        00E0  5420
        00E1  4441
        00E2  5441
        00E3  2054
        00E4  5241
        00E5  4646
        00E6  4943
        00E7  2052
        00E8  4543
        00E9  4549
        00EA  5645
        00EB  4420
        00EC  4652
```

Figure D-1 (cont). Sample Assembly Language Program

```
        00ED  4F4D
        00EE  2048
        00EF  4F53
        00F0  5400
000167              0028          MESSIL    EQU       ($-MESSIT)*2
000168                            *
000169                            *****   FC03 - DATA RECEIVED FROM HOST SSCP
000170                            *****          DISPLAY ANY DATA ONTO TERMINAL
000171                            *
000172              00F1          FC03      EQU       $
000173  00F1  0F81 0763                     B         SRET05                 DISPLAY SIB RETURN CODE ROUTINE
000174                            *
000175                            *****   FC04 - DATA RECEIVED FROM HOST APPLICATION
000176                            *              AND IS SITTING IN THE SIB, DISPLAY
000177                            *              THIS DATA ONTO THE TERMINAL
000178                            *
000179              00F3          FC04      EQU       $
000180  00F3  0F81 0761                     B         SRET05                 DISPLAY SIB RETURN CODE ROUTINE
000181                            *
000182                            *
000183                            ******  FC05 - LU IS ALREADY IN SESSION
000184                            *              DISPLAY MESSAGE ONTO OPERATOR CONSOLE
000185                            *              RETURN CONTROL TO TERMINAL USER
000186                            *******
000187                            *
000188              00F5          FC05      EQU       $
000189  00F5  C3C0 0009                     LAB       $B4,LUSESS                       MESSAGE TEXT
000190  00F7  E570 0022                     LDR       $R6,=LUSESL                      MESSAGE LENGTH
000191  00F9  7C00                          LDV       $R7,0                            LEFT BYTE
000192                            *                   WRITE TO USER OUT
000193  00FA  0001                          MCL
000194  00FB  0801                          DC        X'0801'
000195                            *
000196                            *
000197  00FC  0F81 0759                     B         SRET05                           DISPLAY SIB RETURN CODE ROUTINE
000198                            *
000199                            *
000200  00FE  4120                LUSESS    TEXT      'A LU ALREADY IN SESSION WITH HOST'
        00FF  4C55
        0100  2041
        0101  4C52
        0102  4541
        0103  4459
        0104  2049
        0105  4E20
        0106  5345
        0107  5353
        0108  494F
        0109  4E20
        010A  5749
        010B  5448
        010C  2048
        010D  4F53
        010E  5400
```

Figure D-1 (cont). Sample Assembly Language Program



```
MANUAL 031683     SAMPLE ITF PROGRAM -LAF  1983/03/16 0957:28.4 MAP-1.1 -08/04/1008  GCOS6 MOD400-R3.0-09/17/1626   PAGE 0007

000201              0022           LUSESL    EQU      ($-LUSESS)*2
000202                             *
000203                             *
000204                             ******  FC06: FUNCTION CODE WAS SET TO X'06' - TRAFFIC INACTIVE - SIB
000205                             *             CANNOT BE SENT TO HOST.  WE WILL ISSUE A MESSAGE TO THE
000206                             *             TERMINAL OPERATOR TELLING HIM SUCH - AND GIVE HIM BACK
000207                             *             CONTROL OF THE TERMINAL
000208                             *
000209              010F           FC06      EQU      $
000210   010F  A3C0 0337                     LAB      $B2,TRAFIN                      DATA TO BE DISPLAYED
000211   0111  B970 0780                     LDR      $R3,=1920                       # OF CHARACTERS TO MOVE
000212                             *
000213   0113  D3C0 0733                     LNJ      $B5,MOVEIT
000214                             *
000215   0115  0F81 073F                     B        SRET05                          RC ROUTINE
000216                             *
000217                             *
000218                             ******     FC07:
000219                             *
000220                             *
000221              0117           FC07      EQU      $
000222                             *
000223                             *
000224                             ************   FC08: TRANSMISSION TO HOST CANNOT BE HONORED
000225                             *                    ISSUE A MESSAGE TO DISPLAY OPERATOR
000226                             *                    AS WELL AS GIVING HIM CONTROL
000227                             ************
000228                             *
000229              0117           FC08      EQU      $
000230   0117  A3C0 0007                     LAB      $B2,TTHCBH                      DATA TO BE DISPLAYED
000231   0119  B970 0780                     LDR      $R3,=1920            # OF CHARS. TO MOVE
000232   011B  D3C0 0733                     LNJ      $B5,MOVEIT
000233                             *
000234   011D  0F81 0737                     B        SRET05                          RC ROUTINE
000235                             *
000236   011F       011F           TTHCBH    ORG      $         TRAN. TO HOST CANNOT BE HONORED
000237   011F       2020                     RESV     480,A'  '
000238   02FF  5452                          DC       A'TRANSMISSION TO HOST CANNOT BE '
         0300  414E
         0301  534D
         0302  4953
         0303  5349
         0304  4F4E
         0305  2054
         0306  4F20
         0307  484F
         0308  5354
         0309  2043
         030A  414E
         030B  4E4F
         030C  5420
         030D  4245
         030E  2020
000239   030F  484F                          DC       'HONORED - SESSION WITH HOST APPLICATION '
```

Figure D-1 (cont). Sample Assembly Language Program

```
MANUAL 031683     SAMPLE ITF PROGRAM -LAF  1983/03/16 0957:28.4 MAP-1.1 -08/04/1008  GCOS6 MOD400-R3.0-09/17/1626   PAGE 0008
               0310   4E4F
               0311   5245
               0312   4420
               0313   2020
               0314   5345
               0315   5353
               0316   494F
               0317   4E20
               0318   5749
               0319   5448
               031A   2048
               031B   4F53
               031C   5420
               031D   4150
               031E   504C
               031F   4943
               0320   4154
               0321   494F
               0322   4E20
     000240    0323   4953                                   DC          'IS NOW TERMINATED - CONTROL IS YOURS!'
               0324   204E
               0325   4F57
               0326   2054
               0327   4552
               0328   4D49
               0329   4E41
               032A   5445
               032B   4420
               032C   2020
               032D   434F
               032E   4E54
               032F   524F
               0330   4C20
               0331   4953
               0332   2059
               0333   4F55
               0334   5253
               0335   2120
     000241    0336         2020                             RESV        400,A'  '
     000242    04C6   0000                                   DC          0
     000243                                     *
     000244                                     *
     000245    04C7         04C7                TRAFIN       ORG         $           TRAFFIC INACTIVE SCREEN DISPLAY
     000246    04C7         2020                             RESV        480,A'  '
     000247    06A7   5452                                   DC          A'TRAFFIC INACTIVE - SIB CANNOT BE SENT TO HOST'
               06A8   4146
               06A9   4649
               06AA   4320
               06AB   494E
               06AC   4143
               06AD   5449
               06AE   5645
               06AF   2020
               06B0   2053
```

Figure D-1 (cont). Sample Assembly Language Program

```
0631  4942
0632  2043
0633  414E
0634  4E4F
0635  5420
0636  4245
0637  2053
0638  454E
0639  5420
063A  544F
063B  2048
063C  4F53
063D  5420
063E        2020                      RESV      400,A'  '
084E  0000                            DC        0
                                *
                                *
                                ***********************************
                                ***  MOVEIT: ROUNTINE TO MOVE DATA FROM USER BUFFER
                                *            TO SIB BUFFER, SO THAT ITF WILL DISPLAY
                                *            IT
                                *
                                *    UPON ENTRY:
                                *                  $B2 = POINTS TO DATA TO BE MOVED INTO SIB
                                *                  SIB ($B1) = LOCATION OF SIB (DESTINATION)
                                *                  $R3 = # OF CHARACTERS TO BE MOVED
                                *                  $R4 = TEMP. STORAGE
                                *                  $R2 = COUNTER
                                *                  $R1 = COUNTER
                                ************************************
                                *;-
                  084F          MOVEIT    EQU      $
084F  1C00                                LDV      $R1,0
0850  2C00                                LDV      $R2,0                     SET TO ZERO
                                *
0851  C0EE                      $A        LDH      $R4,$B2.+$R2              GET CHAR. FROM BUFFER
0852  C700                                STH      $R4,SIB.+$R1              STORE INTO SIB BUFFER
0853  377E            T                   BDEC     $R3,>-$A                  LOOP UNTIL = 0
0854  8385                                JMP      $85
                                *
                                *
                                ******* SRETOS: SET RETURN CODE = X'05'
                                *               DISPLAY SIB ONTO TERMINAL DISPLAY
                                *
                  0855          SRETOS    EQU      $
0855  A870 0780                           LDR      $R2,=RETCOD               BYTE OFFSET OF RETURN CODE
0857  1C05                                LDV      $R1,=FR05                 RC = 05
0858  97A1                                STH      $R1,SIB.$R2               STORE THE RC INTO THE SIB
0859  0F81 009A                           B        EXIT1                     TERMINATE - BACK TO ITF
                                *
                                *****************
                                *
                                *   FC09: SIB ORIGINATED AT THE DISPLAY
                                *
```

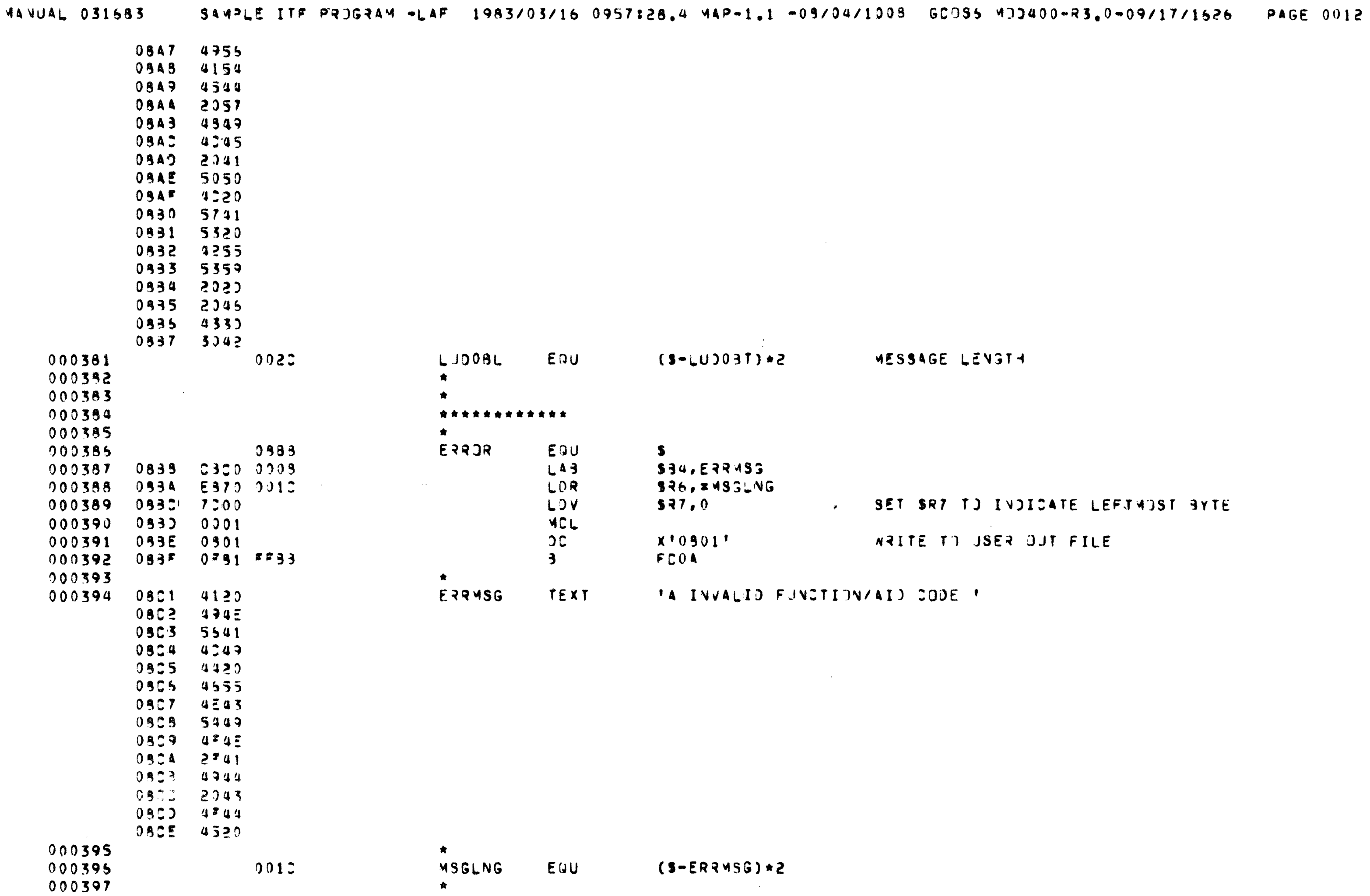
Figure D-1 (cont). Sample Assembly Language Program

```
                        ******************
                        *
                        *
           0853         FC09      EQU       $
0853  A870 0782                   LDR       $R2,=31922
085D  D2A1                        LLH       $R5,SIB.$R2               GET AID KEY HIT
085E  D970 00F1                   CMR       $R5,=PF1                  CHECK FOR FUNCTION KEY 1
0860  0985                        BNE       >CHKCLR                   NOT HIT: CHECK FOR CLEAR KEY
                        *
                        *   FUNCTION KEY 1 SIGNIFIES HERE THAT THE OPERATOR
                        *   WANTS THE APPLICATION TO TERMINATE THE SESSION.
                        *
0861  C870 0002                   LDR       $R4,=FR02                 SET FOR RETCOD 2: TERMINATE SESSION
0863  0F81 008D                   B         EXIT
                        *
           0865         CHKCLR    EQU       $
0865  D970 006D                   CMR       $R5,=CLR .                CLEAR KEY?
0867  0987                        BNE       >XMITT          NOT HIT: CHECK FOR XMIT KEY
                        *
                        *         LOAD RETURN CODE OF X'03' - SEND CLEAR CODE TO HOST
                        *
0868  A870 0780                   LDR       $R2,=RETCOD               BYTE OFFSET TO RC
086A  1C03                        LDV       $R1,=FR03                 WAIT RC
086B  9741                        STH       $R1,SIB.$R2               STORE RC
086C  0F81 0087                   B         EXIT1                     RETURN TO ITF
                        *
                        *         XMITT: SEND MODIFIED DATA TO HOST APPL
                        *         RETURN CODE = X'03'
                        *
           086E         XMITT     EQU       $
086E  D970 007D                   CMR       $R5,=XMIT                 IS IT XMIT?
0870  0987                        BNE       >PFKEY                    NO HIT: GO DISPLAY THE KEY HIT
                        *
0871  A870 0780                   LDR       $R2,=RETCOD               BYTE OFFSET TO RETURN CODE
0873  1C03                        LDV       $R1,=FR03                 RC = 03, SEND SIB TO HOST
0874  9741                        STH       $R1,SIB.$R2               STORE IT INTO SIB
0875  0F81 007E                   B         EXIT1                     GO TERMINATE
                        *
                        *
                        *  IF NOT PF1/CLR/XMIT, REFRESH THE FORMAT AND ECHO THE AID KEY
                        *
           0877         PFKEY     EQU       $
0877  D5C0 007F                   LNJ       $B5,FORMAT
0879  0F81 0055                   B         RETURN
                        *
                        *
                        *
                        ******************
                        *
                        *   FC0A: FINAL CALL FROM ITF: OPERATOR LOGOFF/STOP
                        *
                        ******************
                        *
           087B         FC0A      EQU       $
```

Figure D-1 (cont). Sample Assembly Language Program

```
000343                          *
000344  087B  C3C0 000C                   LAB      $B4,TERMMT                    TERMINATE RECEIVED MESSAGE
000345  087D  E870 001A                   LDR      $R6,=TERMML                   MESSAGE LENGTH
000346                          *
000347  087F  7C00                        LDV      $R7,0               SET $R7 TO INDICATE LEFTMOST BYTE
000348  0880  0001                        MCL
000349  0881  0801                        DC       X'0801'             WRITE TO USER OUT FILE
000350                          *
000351  0882  A870 0780                   LDR      $R2,=RETCOD                   BYTE OFFSET OF RC
000352  0884  1C07                        LDV      $R1,=FR07                     FINAL RETURN RC
000353  0885  9741                        STH      $R1,SIB,$R2                   STORE IT
000354  0886  0F81 006D                   B        EXIT1
000355                          *
000356                          *
000357  0888  4120              TERMMT    TEXT     'A LU DEACTIVATED - FC = 0A'
        0889  4C55
        088A  2044
        088B  4541
        088C  4354
        088D  4956
        088E  4154
        088F  4544
        0890  202D
        0891  2046
        0892  4320
        0893  3D20
        0894  3041
000358             001A         TERMML    EQU      ($-TERMMT)*2
000359                          *
000360                          *
000361                          ********************
000362                          *  FC08: FINAL CALL FROM ITF: APPL. WAS BUSY WHEN RECEIVED
000363                          **
000364                          *
000365                          *
000366             0895         FC08      EQU      $
000367                          *
000368  0895  C3C0 000C                   LAB      $B4,LUD08T                    TERMINATE RECEIVED MESSAGE
000369  0897  E870 002C                   LDR      $R6,=LUD08L                   MESSAGE LENGTH
000370                          *
000371  0899  7C00                        LDV      $R7,0               SET $R7 TO INDICATE LEFTMOST BYTE
000372  089A  0001                        MCL
000373  089B  0801                        DC       X'0801'             WRITE TO USER OUT FILE
000374                          *
000375  089C  A870 0780                   LDR      $R2,=RETCOD                   BYTE OFFSET OF RC
000376  089E  1C07                        LDV      $R1,=FR07                     FINAL RETURN RC
000377  089F  9741                        STH      $R1,SIB,$R2                   STORE IT INTO SIB
000378  08A0  0F81 0053                   B        EXIT1                         GO TERMINATE
000379                          *
000380  08A2  4120              LUD08T    TEXT     'A LU DEACTIVATED WHILE APPL WAS BUSY - FC=08'
        08A3  4C55
        08A4  2044
        08A5  4541
        08A6  4354
```

Figure D-1 (cont). Sample Assembly Language Program



```
MANUAL 031683     SAMPLE ITF PROGRAM -LAF  1983/03/16 0957:28.4 MAP-1.1 -09/04/1008  GCOS6 MOD400-R3.0-09/17/1626   PAGE 0012
               08A7  4955
               08A8  4154
               08A9  4544
               08AA  2057
               08AB  4849
               08AC  4C45
               08AD  2041
               08AE  5050
               08AF  4C20
               08B0  5741
               08B1  5320
               08B2  4255
               08B3  5359
               08B4  2020
               08B5  2046
               08B6  4330
               08B7  5042
000381                    002C           LUDOBL    EQU       ($-LUDOBT)*2         MESSAGE LENGTH
000382                                   *
000383                                   *
000384                                   *************
000385                                   *
000386                    08B8           ERROR     EQU       $
000387         08B8  C3C0 0008                     LAB       $B4,ERRMSG
000388         08BA  E370 001C                     LDR       $R6,=MSGLNG
000389         08BC  7C00                          LDV       $R7,0          .     SET $R7 TO INDICATE LEFTMOST BYTE
000390         08BD  0001                          MCL
000391         08BE  0801                          DC        X'0801'              WRITE TO USER OUT FILE
000392         08BF  0F81 FF83                     B         FC0A
000393                                   *
000394         08C1  4120                ERRMSG    TEXT      'A INVALID FUNCTION/AID CODE '
               08C2  494E
               08C3  5641
               08C4  4C49
               08C5  4420
               08C6  4655
               08C7  4E43
               08C8  5449
               08C9  4F4E
               08CA  2F41
               08CB  4944
               08CC  2043
               08CD  4F44
               08CE  4520
000395                                   *
000396                    001C           MSGLNG    EQU       ($-ERRMSG)*2
000397                                   *
```

Figure D-1 (cont). Sa ple Assembly Language Program

```
MANUAL 031683     SAMPLE ITF PROGRAM -LAF  1983/03/16 0957:28.4 MAP-1.1 -08/04/1008  GCOS6 MOD400-R3.0-09/17/1626   PAGE 0013

000398                               /
000399                               *
000400                               *
000401                               *******************
000402                               *
000403                               *    RETURN: MAP THE AID KEY TO ITS ASCII VALUE AND SET LINE 24
000404                               *            OF THE SIB TO ECHO THE AID KEY THAT WAS HIT
000405                               *
000406                               *******************
000407                               *
000408                               *
000409                               *
000410                08CF           RETURN    EQU      $
000411  08CF  B370 074E                        LDR      $R3,=1870                       SIB INDEX
000412  08D1  A3C0 0420                        LAB      $B2,AIDTXT                      POINT TO TEXT
000413  08D3  F370 000D                        LDR      $R7,=AIDMSG-1                   LENGTH FOR LOOP
000414  08D5  2C00                             LDV      $R2,0                           INDEX TO TEXT
000415  08D6  C2EE                   RTNTXT    LLH      $R4,$B2.+$R2                    MOVE TEXT
000416  08D7  C7FD                             STH      $R4,SIB.+$R3                    TO THE SIB
000417  08D8  777E                             BDEC     $R7,>RTNTXT
000418                               *
000419                               *
000420                               *    TRANSLATE AID CODE TO ASCII VIA TABLE LOOKUP
000421                               *
000422  08D9  A370 0782                        LDR      $R2,=31922
000423  08DB  D2A1                             LLH      $R5,SIB.$R2                     GET AID CODE
000424  08DC  B3C0 03E8                        LAB      $B3,AIDTBL                      CODE TABLE
000425  08DE  3C00                             LDV      $R3,0                           INDEX
000426  08DF  C2FF                   RTNTX1    LLH      $R4,$B3.+$R3                    GET TABLE ENTRY
000427  08E0  4901 FFD7                        BEZ      $R4,ERROR                       ZERO TERMINATOR MEANS INVALID CODE
000428  08E2  D954                             CMR      $R5,=$R4                        IS IT A HIT?
000429  08E3  09FC                             BNE      >RTNTX1                         CONTINUE THE SEARCH
000430                               *
000431                               *
000432                               *    MAP AID CODE TO ASCII VALUE
000433                               *
000434  08E4  B3C0 03F0                        LAB      $B3,AIDASC                      ASCII TABLE
000435  08E6  3EFF                             ADV      $R3,-1                          ADJUST INDEX
000436  08E7  B333                             LAB      $B3,$B3.$R3                     POINT TO TEXT
000437  08E8  A370 075E                        LDR      $R2,=1886                       SIB LOCATION
000438  08EA  3C00                             LDV      $R3,0
000439  08EB  C2FF                             LLH      $R4,$B3.+$R3                    MOVE ASCII TEXT TO SIB
000440  08EC  C7ED                             STH      $R4,SIB.+$R2
000441  08ED  C233                             LLH      $R4,$B3.$R3
000442  08EE  C7A1                             STH      $R4,SIB.$R2
000443                               *
000444                               *
000445                               *    SET RETURN CODE TO DISPLAY THE SIB
000446                               *
000447  08EF  C370 0005                        LDR      $R4,=FR05
000448                08F1           EXIT      EQU      $
000449  08F1  A370 0780                        LDR      $R2,=RETCOD
000450  08F3  C7A1                             STH      $R4,SIB.$R2
000451                08F4           EXIT1     EQU      $
```

Figure D-1 (cont). Sample Assembly Language Program

```
MANUAL 031683     SAMPLE ITF PROGRAM -LAF  1983/03/16 0957:28.4 MAP-1.1 -08/04/1008  GCOS6 MOD400-R3.0-09/17/1626   PAGE 0014

    000452  08F4  2000                           LDV        $R2,0                    SET GOOD RETURN STATUS
    000453  08F5  0001                           MCL
    000454  08F6  0103                           DC         X'0103'          TERMINATE WITHOUT MODIFYING START ADDRESS
    000455                        *
```

Figure D-1 (cont). Sample Assembly Language Program

```
000456                          /
000457                          *
000458                          *
000459                          ********************
000460                          *
000461                          *   FORMAT: SUBROUTINE TO TRANSFER FORMAT IMAGE TO THE SIB
000462                          *           AND TO SET THE CURSOR ADDRESS AT THE 'NAME' FIELD
000463                          *
000464                          ********************
000465                          *
000466                          *
000467                          *
000468              08F7        FORMAT    EQU     $
000469  08F7  A300 0000                   LAB     $32,IMAGE
000470  08F9  F370 077F                   LDR     $R7,=IMGLNG-1            FOR BDEC LOOP
000471  08FB  2200                        LDV     $R2,0                    INDEX TO IMAGE
000472  08FC  3300                        LDV     $R3,0                    INDEX TO SIB
000473  08FD  C2EE              FORMT1    LLH     $R4,$32.+$R2
000474  08FE  C7FD                        STH     $R4,SIB.+$R3
000475  08FF  777E                        BDEC    $R7,>FORMT1
000476                          *
000477                          *   SET CURSOR AT START OF NAME FIELD
000478                          *
000479  0900  C370 0059                   LDR     $R4,=89                  SIB INDEX
000480  0902  CF41 03C2                   STR     $R4,SIB.N962
000481  0904  8385                        JMP     $35                      RETURN TO CALLER
000482                          *
000483                          *
000484                          *   PRE-FORMATTED SCREEN IMAGE
000485                          *
000486              0905        IMAGE     EQU     $
000487  0905  C000                        DC      Z'C000'                  PROTECTED FIELD: LINE 1
000488  0906        0000                  RESV    39,0
000489  092D  C220                        DC      Z'C220',A' NAME:'        PROTECTED,HIGH INTENSITY: LINE 2
        092E  204E
        092F  414D
        0930  453A
000490  0931  AA20                        DC      Z'AA20'                  UNPRO/UNDERLINE FOR DATA ENTRY
000491  0932        0000                  RESV    35,0
000492  0955  C000                        DC      Z'C000'                  PROTECTED: LINE 3
000493  0956        0000                  RESV    39,0
000494  097D  C220                        DC      Z'C220',A'ADDRESS:'      PROTECTED,HIGH INTENSITY: LINE 4
        097E  4144
        097F  4452
        0980  4553
        0981  533A
000495  0982  AA20                        DC      Z'AA20'                  UNPRO/UNDERLINE FOR DATA ENTRY
000496  0983        0000                  RESV    34,0
000497  09A5  C000                        DC      Z'C000'                  PROTECTED: LINE 5
000498  09A6        0000                  RESV    39,0
000499  09CD  C200                        DC      Z'C200',A'ZIP:'          PROTECTED, HIGH INTENSITY: LINE 6
        09CE  5449
        09CF  503A
000500  09D0  A320                        DC      Z'A320'                  UNPROTECTED,NUMERIC,UNDERLINE
```

Figure D-1 (cont). Sample Assembly Language Program

```
MANUAL 031683      SAMPLE ITF PROGRAM -LAF  1983/03/16 0957:28.4 MAP-1.1 -08/04/1008  GCOS6 MOD400-R3.0-09/17/1626   PAGE 0016

   000501  09D1        0000                  RESV      36,0
   000502  09F5  F500                        DC        Z'F500'                        PROTECT/INVERSE VIDEO: REMAINDER OF SCREEN
   000503  09F6        0000                  RESV      39,0
   000504  0A1D        0000                  RESV      5,0
   000505  0A22  4C4C                        TEXT 'LL            OOO          CCCCC        AA       LL                    '
           0A23  2020
           0A24  2020
           0A25  2020
           0A26  2020
           0A27  2020
           0A28  2020
           0A29  204F
           0A2A  4F4F
           0A2B  2020
           0A2C  2020
           0A2D  2020
           0A2E  2020
           0A2F  2043
           0A30  4343
           0A31  4343
           0A32  2020
           0A33  2020
           0A34  2020
           0A35  2020
           0A36  4141
           0A37  2020
           0A38  2020
           0A39  2020
           0A3A  4C4C
           0A3B  2020
           0A3C  2020
           0A3D  2020
           0A3E  2020
           0A3F  2020
           0A40  2020
           0A41  2020
           0A42  2020
           0A43  2020
           0A44  2000
   000506  0A45        0000                  RESV      5,0
   000507  0A4A  4C4C                        TEXT 'LL           OOOOO        CCCCCCC      AAAA      LL
           0A4B  2020
           0A4C  2020
           0A4D  2020
           0A4E  2020
           0A4F  2020
           0A50  2020
           0A51  4F4F
           0A52  4F4F
           0A53  4F20
           0A54  2020
           0A55  2020
           0A56  2020
           0A57  4343
```

Figure D-1 (cont). Sample Assembly Language Program

```
              0A58  4343
              0A59  4343
              0A5A  4320
              0A5B  2020
              0A5C  2020
              0A5D  2041
              0A5E  4141
              0A5F  4120
              0A60  2020
              0A61  2020
              0A62  4C4C
              0A63  2020
              0A64  2020
              0A65  2020
              0A66  2020
              0A67  2020
              0A68  2020
              0A69  2020
              0A6A  2020
              0A6B  2020
              0A6C  2000
       000508 0A6D        0000                RESV      5,0
       000509 0A72  4C4C                      TEXT 'LL          DD   DD      CC      CC    AA  AA     LL
              0A73  2020
              0A74  2020
              0A75  2020
              0A76  2020
              0A77  2020
              0A78  2044
              0A79  4420
              0A7A  2020
              0A7B  4444
              0A7C  2020
              0A7D  2020
              0A7E  2043
              0A7F  4320
              0A80  2020
              0A81  2020
              0A82  4343
              0A83  2020
              0A84  2020
              0A85  4141
              0A86  2020
              0A87  4141
              0A88  2020
              0A89  2020
              0A8A  4C4C
              0A8B  2020
              0A8C  2020
              0A8D  2020
              0A8E  2020
              0A8F  2020
              0A90  2020
              0A91  2020
```

Figure D-1 (cont). Sample Assembly Language Program

```
        0A92  2020
        0A93  2020
        0A94  2000
000510  0A95        0000                  RESV      5,0
000511  0A9A  4C4C                        TEXT 'LL          OO      OO   CC            AA     AA   LL
        0A9B  2020
        0A9C  2020
        0A9D  2020
        0A9E  2020
        0A9F  2020
        0AA0  4F4F
        0AA1  2020
        0AA2  2020
        0AA3  204F
        0AA4  4F20
        0AA5  2020
        0AA6  4343
        0AA7  2020
        0AA8  2020
        0AA9  2020
        0AAA  2020
        0AAB  2020
        0AAC  2041
        0AAD  4120
        0AAE  2020
        0AAF  2041
        0AB0  4120
        0AB1  2020
        0AB2  4C4C
        0AB3  2020
        0AB4  2020
        0AB5  2020
        0AB6  2020
        0AB7  2020
        0AB8  2020
        0AB9  2020
        0ABA  2020
        0ABB  2020
        0ABC  2000
000512  0ABD        0000                  RESV      5,0
000513  0AC2  4C4C                        TEXT 'LL          O       O    CC           AA      AA  LL
        0AC3  2020
        0AC4  2020
        0AC5  2020
        0AC6  2020
        0AC7  2020
        0AC8  4F20
        0AC9  2020
        0ACA  2020
        0ACB  2020
        0ACC  4F20
        0ACD  2020
        0ACE  4343
        0ACF  2020
```

Figure D-1 (cont). Sample Assembly Language Program

```
        0AD0  2020
        0AD1  2020
        0AD2  2020
        0AD3  2020
        0AD4  4141
        0AD5  2020
        0AD6  2020
        0AD7  2020
        0AD8  4141
        0AD9  2020
        0ADA  4C4C
        0ADB  2020
        0ADC  2020
        0ADD  2020
        0ADE  2020
        0ADF  2020
        0AE0  2020
        0AE1  2020
        0AE2  2020
        0AE3  2020
        0AE4  2000
000514  0AE5       0000                           RESV      5,0
000515  0AEA  4C4C                                TEXT 'LL          O       O   CC          AA      AA  LL
        0AEB  2020
        0AEC  2020
        0AED  2020
        0AEE  2020
        0AEF  2020
        0AF0  4F20
        0AF1  2020
        0AF2  2020
        0AF3  2020
        0AF4  4F20
        0AF5  2020
        0AF6  4343
        0AF7  2020
        0AF8  2020
        0AF9  2020
        0AFA  2020
        0AFB  2020
        0AFC  4141
        0AFD  2020
        0AFE  2020
        0AFF  2020
        0B00  4141
        0B01  2020
        0B02  4C4C
        0B03  2020
        0B04  2020
        0B05  2020
        0B06  2020
        0B07  2020
        0B08  2020
        0B09  2020
```

Figure D-1 (cont). Sample Assembly Language Program

```
MANUAL 031683     SAMPLE ITF PROGRAM -LAF  1983/03/16 0957:28.4 MAP-1.1 -08/04/1008  GC386 MD3400-R3.0-09/17/1626  PAGE 0020
        030A  2020
        030B  2020
        030C  2000
000516  030D       0000                        RESV      5,0
000517  0312  4C4C                             TEXT 'LL          O       O   CC          AAAAAAAAAA  LL
        0313  2020
        0314  2020
        0315  2020
        0316  2020
        0317  2020
        0318  4F20
        0319  2020
        031A  2020
        031B  2020
        031C  4F20
        031D  2020
        031E  4343
        031F  2020
        0320  2020
        0321  2020
        0322  2020
        0323  2020
        0324  4141
        0325  4141
        0326  4141
        0327  4141
        0328  4141
        0329  2020
        032A  4C4C
        032B  2020
        032C  2020
        032D  2020
        032E  2020
        032F  2020
        0330  2020
        0331  2020
        0332  2020
        0333  2020
        0334  2000
000518  0335       0000                        RESV      5,0
000519  033A  4C4C                             TEXT 'LL          O       O   CC          AAAAAAAAAA  LL
        033B  2020
        033C  2020
        033D  2020
        033E  2020
        033F  2020
        0340  4F20
        0341  2020
        0342  2020
        0343  2020
        0344  4F20
        0345  2020
        0346  4343
        0347  2020
```

Figure D-1 (cont). Sample Assembly Language Program

```
MANUAL 031683      SAMPLE ITF PROGRAM -LAF  1983/03/16 0957:28.4 MAP-1.1 -08/04/1008  GCOS6 MOD400-R3.0-09/17/1626   PAGE 0021
               0848  2020
               0849  2020
               084A  2020
               084B  2020
               084C  4141
               084D  4141
               084E  4141
               084F  4141
               0850  4141
               0851  2020
               0852  4C4C
               0853  2020
               0854  2020
               0855  2020
               0856  2020
               0857  2020
               0858  2020
               0859  2020
               085A  2020
               085B  2020
               085C  2000
      000520   085D        0000                      RESV      5,0
      000521   0862  4C4C                            TEXT 'LL          OO      OO  CC          AA      AA  LL
               0863  2020
               0864  2020
               0865  2020
               0866  2020
               0867  2020
               0868  4F4F
               0869  2020
               086A  2020
               086B  204F
               086C  4F20
               086D  2020
               086E  4343
               086F  2020
               0870  2020
               0871  2020
               0872  2020
               0873  2020
               0874  4141
               0875  2020
               0876  2020
               0877  2020
               0878  4141
               0879  2020
               087A  4C4C
               087B  2020
               087C  2020
               087D  2020
               087E  2020
               087F  2020
               0880  2020
               0881  2020
```

Figure D-1 (cont). Sample Assembly Language Program

```
         0382  2020
         0383  2020
         0384  2000
000522   0385        0000                   RESV      5,0
000523   038A  4C4C                         TEXT 'LL         DD   DD     CC     CC  AA      AA  LL
         038B  2020
         038C  2020
         038D  2020
         038E  2020
         038F  2020
         0390  204F
         0391  4F20
         0392  2020
         0393  4F4F
         0394  2020
         0395  2020
         0396  2043
         0397  4320
         0398  2020
         0399  2020
         039A  4343
         039B  2020
         039C  4141
         039D  2020
         039E  2020
         039F  2020
         03A0  4141
         03A1  2020
         03A2  4C4C
         03A3  2020
         03A4  2020
         03A5  2020
         03A6  2020
         03A7  2020
         03A8  2020
         03A9  2020
         03AA  2020
         03AB  2020
         03AC  2000
000524   03AD        0000                   RESV      5,0
000525   03B2  4C4C                         TEXT 'LLLLLLLLL     DDDDD      CCCCCCC    AA      AA  LLLLLLLLL
         03B3  4C4C
         03B4  4C4C
         03B5  4C4C
         03B6  4C20
         03B7  2020
         03B8  2020
         03B9  4F4F
         03BA  4F4F
         03BB  4F20
         03BC  2020
         03BD  2020
         03BE  2020
         03BF  4343
```

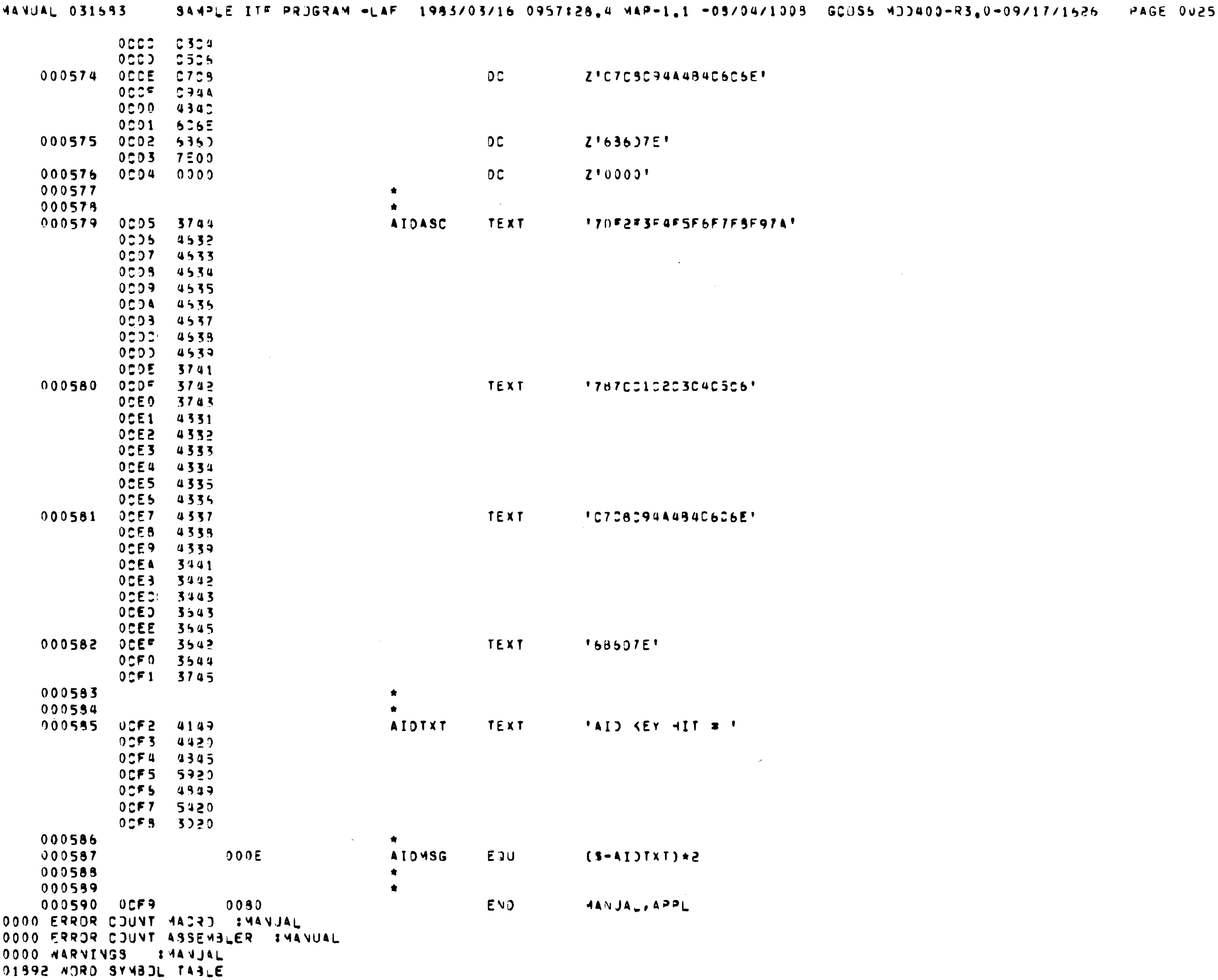
Figure D-1 (cont). Sample Assembly Language Program

```
MANUAL 031683     SAMPLE ITF PROGRAM -LAF  1983/03/16 0957:28.4 MAP-1.1 -08/04/1008  GCOS6 MOD400-R3.0-09/17/1626   PAGE 0023
            0BC0   4343
            0BC1   4343
            0BC2   4320
            0BC3   2020
            0BC4   4141
            0BC5   2020
            0BC6   2020
            0BC7   2020
            0BC8   4141
            0BC9   2020
            0BCA   4C4C
            0BCB   4C4C
            0BCC   4C4C
            0BCD   4C4C
            0BCE   4C20
            0BCF   2020
            0BD0   2020
            0BD1   2020
            0BD2   2020
            0BD3   2020
            0BD4   2000
    000526  0BD5        0000                       RESV       S,0
    000527  0BDA   4C4C                            TEXT 'LLLLLLLLL         OOO          CCCCC     AA        AA  LLLLLLLLL
            0BDB   4C4C
            0BDC   4C4C
            0BDD   4C4C
            0BDE   4C20
            0BDF   2020
            0BE0   2020
            0BE1   204F
            0BE2   4F4F
            0BE3   2020
            0BE4   2020
            0BE5   2020
            0BE6   2020
            0BE7   2043
            0BE8   4343
            0BE9   4343
            0BEA   2020
            0BEB   2020
            0BEC   4141
            0BED   2020
            0BEE   2020
            0BEF   2020
            0BF0   4141
            0BF1   2020
            0BF2   4C4C
            0BF3   4C4C
            0BF4   4C4C
            0BF5   4C4C
            0BF6   4C20
            0BF7   2020
            0BF8   2020
            0BF9   2020
```

Figure D-1 (cont). Sample Assembly Language Program

```
        0BFA  2020
        0BFB  2020
        0BFC  2000
000528  0BFD       0000                     RESV      200,0
000529                            *
000530                            *
000531             0780           IMGLNG    EQU       ($-IMAGE)*2
000532                            *
000533                            *
000534                            *   MAPPING TABLES FOR AID KEY TRANSLATION
000535                            *
000536                            *
000537                            *         KEY                   EBCDIC
000538                            *         ---                   ------
000539                            *
000540                            *         ENTER                 7D
000541                            *         PF1                   F1
000542                            *         PF2                   F2
000543                            *         PF3                   F3
000544                            *         PF4                   F4
000545                            *         PF5                   F5
000546                            *         PF6                   F6
000547                            *         PF7                   F7
000548                            *         PF8                   F8
000549                            *         PF9                   F9
000550                            *         PF10                  7A
000551                            *         PF11                  7B
000552                            *         PF12                  7C
000553                            *         PF13                  C1
000554                            *         PF14                  C2
000555                            *         PF15                  C3
000556                            *         PF16                  C4
000557                            *         PF17                  C5
000558                            *         PF18                  C6
000559                            *         PF19                  C7
000560                            *         PF20                  C8
000561                            *         PF21                  C9
000562                            *         PF22                  4A
000563                            *         PF23                  4B
000564                            *         PF24                  4C
000565                            *         LITE-PEN              7E
000566                            *         PA1                   6C
000567                            *         PA2                   6E
000568                            *         PA3                   6B
000569                            *         CLEAR                 6D
000570                            *
000571                            *
000572  0CC5  7DF2                AIDTBL    DC        Z'7DF2F3F4F5F6F7F8F97A'
        0CC6  F3F4
        0CC7  F5F6
        0CC8  F7F8
        0CC9  F97A
000573  0CCA  7B7C                          DC        Z'7B7CC1C2C3C4C5C6'
        0CCB  C1C2
```

Figure D-1 (cont). Sample Assembly Language Program



```
MANUAL 031683     SAMPLE ITF PROGRAM -LAF  1983/03/16 0957:28.4 MAP-1.1 -09/04/1008  GCOS6 MOD400-R3.0-09/17/1626   PAGE 0025
          0CCC  C3C4
          0CCD  C5C6
  000574  0CCE  C7C8                      DC       Z'C7C8C94A4B4C6C6E'
          0CCF  C94A
          0CD0  4B4C
          0CD1  6C6E
  000575  0CD2  636D                      DC       Z'636D7E'
          0CD3  7E00
  000576  0CD4  0000                      DC       Z'0000'
  000577                         *
  000578                         *
  000579  0CD5  3744             AIDASC   TEXT     '7DF2F3F4F5F6F7F8F97A'
          0CD6  4532
          0CD7  4533
          0CD8  4534
          0CD9  4535
          0CDA  4536
          0CDB  4537
          0CDC  4538
          0CDD  4539
          0CDE  3741
  000580  0CDF  3742                      TEXT     '7B7CC1C2C3C4C5C6'
          0CE0  3743
          0CE1  4331
          0CE2  4332
          0CE3  4333
          0CE4  4334
          0CE5  4335
          0CE6  4336
  000581  0CE7  4337                      TEXT     'C7C8C94A4B4C6C6E'
          0CE8  4338
          0CE9  4339
          0CEA  3441
          0CEB  3442
          0CEC  3443
          0CED  3643
          0CEE  3645
  000582  0CEF  3642                      TEXT     '6B6D7E'
          0CF0  3644
          0CF1  3745
  000583                         *
  000584                         *
  000585  0CF2  4149             AIDTXT   TEXT     'AID KEY HIT = '
          0CF3  4420
          0CF4  4B45
          0CF5  5920
          0CF6  4849
          0CF7  5420
          0CF8  3D20
  000586                         *
  000587             000E        AIDMSG   EQU      ($-AIDTXT)*2
  000588                         *
  000589                         *
  000590  0CF9       0080                 END      MANUAL,APPL
0000 ERROR COUNT MACRO  :MANUAL
0000 ERROR COUNT ASSEMBLER  :MANUAL
0000 WARNINGS  :MANUAL
01892 WORD SYMBOL TABLE
```

Figure D-1 (cont). Sample Assembly Language Program

```
MANUAL 031683     SAMPLE ITF PROGRAM -LAF  1983/03/16 0957:28.4 MAP-1.1 -08/04/1008  GCOS6 MOD400-R3.0-09/17/1626   PAGE 0026

  $        ****   73   85  107  122  135  149  167  172  179  188  201  209  221  229  236  245  256  279  292
                 304  318  330  342  358  366  381  386  396  410  448  451  468  486  531  587
  $A        270  272
  $AF      ****   54   57   60   77
  $B1      ****   19
  $B2      ****   80   81  210  230  270  412  415  469  473
  $B3      ****  424  425  434  436  439  441
  $B4      ****   77  108  150  189  344  368  387
  $B5      ****   81   82  158  213  232  273  331  481
  $R1      ****   79   81  116  117  137  138  161  162  267  271  281  282  311  312  323  324  352  353  376
                 377
  $R2      ****   78   79  115  117  136  138  160  162  268  270  280  282  293  294  310  312  322  324  351
                 353  375  377  414  415  422  423  437  440  442  448  450  452  471  473
  $R3      ****  211  231  272  411  416  425  426  435  436  438  439  441  472  474
  $R4      ****  270  271  301  415  416  426  427  428  439  440  441  442  447  450  473  474  479  480
  $R5      ****  294  295  305  319  423  428
  $R6      ****  109  151  190  345  369  388
  $R7      ****  110  152  191  347  371  389  413  417  470  475
  AIDASC    579  434
  AIDMSG    587  413
  AIDTBL    572  424
  AIDTXT    585  412  587
  APPL       73  590
  B1922      22  293  422
V B1923      23
V B1927      25
V B1932      26
  CHKCLR    304  296
  CLR        45  305
  ERRML1    122  109
  ERRMS1    121  108  122
  ERRMSG    394  397  396
  ERROR     386  427
  EXIT      448  302
  EXIT1     451  118  139  164  283  313  325  354  376
  FC00      107   87
  FC01      135   88
  FC02      149   89
  FC03      172   90
  FC04      179   91
  FC05      188   92
  FC06      209   93
  FC07      221   94
  FC08      229   95
  FC09      292   96
  FC0A      342   97  392
  FC0B      366   98
  FCVTBL     86   80
  FORMAT    468  158  331
  FORMT1    473  475
  FR01       32  137
  FR02       33  301
  FR03       34  311  323
V FR04       35
```

Figure D-1 (cont). Sample Assembly Language Program

```
MANUAL 031683     SAMPLE ITF PROGRAM -LAF  1983/03/16 0957:28.4 MAP-1.1 -08/04/1008  GCOS6 MOD400-R3.0-09/17/1626   PAGE 0027
  FR05        36   151   281   447
V FR06        37
  FR07        38   116   352   376
V FR08        39
V FR09        40
V FR0A        41
V FR0B        42
  FUNCOD      21    78
  IMAGE      486   459   531
  IMGLNG     531   470
  LUDOBL     381   359
  LUDOBT     380   359   381
  LUSESL     201   190
  LUSESS     200   199   201
  MESS1L     167   151
  MESS1T     166   150   167
  MOVEIT     266   213   232
  MSGLNG     396   399
V MT_D        67
V MT_I        68
V MT_R        66
V MT_S        65
V MT_T        62
V MT_U        64
V MT_W        63
  PF1         44   295
  PFKEY      330   320
  RETCOD      20   115   136   160   280   310   322   351   375   449
  RETURN     410   332
  RTNTX1     426   429
  RTNTXT     415   417
  SIB         19    77    79   117   138   162   271   282   294   312   324   353   377   416   423   440   442   450   474   480
  SRET05     279   173   180   197   215   234
  T_ADR       59    60    77
  T_CT1       57    58
  T_CT2       58    59
V T_PRM       60
V T_RRB       54
V T_RS        56
V T_SEM       55
  TERMML     358   345
  TERMMT     357   344   358
  TRAFIN     245   210
  TTHCBH     236   230
  W962        24   480
  XMIT        46   319
  XMITT      318   306
   84 LABELS
  297 REFERENCES
  590 RECORDS
    0 U FLAGS
    0 M FLAGS
   20 V FLAGS
 1146 WORD CROSS REFERENCE TABLE
```

Figure D-1 (cont). Sample Assembly Language Program

INDEX

3274 FUNCTIONS
- WST/VIP7200 KEY EQUIVALENTS TO 3247 FUNCTIONS (TBL), 3-32
- WST/VIP7300 KEY EQUIVALENTS TO 3274 FUNCTIONS (TBL), 3-33
- WST/VIP7800 KEY EQUIVALENTS TO 3274 FUNCTIONS (TBL), 3-34

ADDRESS
- REQUEST HEADER, REQUEST UNIT, AND LOGICAL-UNIT ADDRESS, 4-9

ALPHA OVERRIDE
- ALPHA OVERRIDE FUNCTION, 3-15

APPLICATION PROGRAM
- APPLICATION PROGRAM DATA AREA, 4-9
- APPLICAITON PROGRAM/DATA FIELD DISPLAY INTERFACE, 4-3
- APPLICATION PROGRAMS AND PROCEDURES, 4-1
- ASSEMBLY LANGUAGE APPLICATION PROGRAM, 4-10
- COBOL APPLICATION PROGRAM, 4-9
- CONFIGURING AN APPLICATION PROGRAM, 4-1
- SAMPLE ITF ASSEMBLY LANGUAGE APPLICATION PROGRAM, D-1
- SAMPLE ITF COBOL APPLICATION PROGRAM, C-1

ASSEMBLY LANGUAGE
- ASSEMBLY LANGUAGE APPLICATION PROGRAM, 4-10
- SAMPLE ITF ASSEMBLY LANGUAGE APPLICATION PROGRAM, D-1

ATTRIBUTE CHARACTER
- DISPLAY SHOWING ATTRIBUTE CHARACTER LOCATIONS (FIG), 4-4

ATTRIBUTES
- DATA FIELD ATTRIBUTES, 4-4

BACKTAB
- BACKTAB FUNCTION, 3-14

CAPS
- CAPS LOCK, 3-12

CLEAR
- CLEAR FUNCTION, 3-17

COBOL
- COBOL APPLICATION PROGRAM, 4-9
- SAMPLE ITF COBOL APPLICATION PROGRAM, C-1

CURSOR
- CURSOR CONTROL, 3-12
- CURSOR DOWN, 3-13
- CURSOR LEFT, 3-13
- CURSOR LOCATION, 4-9
- CURSOR RIGHT, 3-13
- CURSOR SELECT, 3-17
- CURSOR UP, 3-12
- MOVING THE CURSOR FROM CHARACTER TO CHARACTER, 3-12
- MOVING THE CURSOR FROM FIELD TO FIELD, 3-13

DATA ENTRY
- DATA ENTRY, 3-12
- STARTING AND STOPPING DATA ENTRY, 3-17
- TYPICAL DISPLAY SCREEN DURING DATA ENTRY (FIG), 3-17
- USING YOUR TERMINAL FOR DATA ENTRY, 3-5

DATA ENTRY KEYBOARD
- WST/VIP7200 DATA ENTRY KEYBOARD (FIG), 3-23
- WST/VIP7300 DATA ENTRY KEYBOARD (FIG) 3-27

DATA FIELD
- DATA FIELD, 4-3
- DATA FIELD ATTRIBUTES, 4-4
- DATA FIELD DESCRIPTION, 4-4

DELETE
- DELETE FUNCTION, 3-16

DEVICE CANCEL
- DEVICE CANCEL FUNCTION, 3-17

DTSPLAY
- APPLICATION PROGRAM/DATA FIELD DISPLAY INTERFACE, 4-3
- FORMATTED DISPLAY. 4-3
- FORMATTED AND UNFORMATTED DISPLAYS, 3-10
- STATUS LINE DISPLAYS (FIG). 3-8
- UNFORMATTED DISPLAYS (FIG), 3-8

DUP
- DUP FUNCTION, 3-15

EDITTNG
- EDITTNG, 3-16

ENTER
- ENTER FUNCTION. 3-17

ERASE TNPUT
- ERASE TNPUT FUNCTION. 3-15

ERASE-TO-END-OF-FIELD
- ERASE-TO-END-OF-FIELD FUNCTION, 3-15

EXIT
- TYPES OF USER PROGRAM EXITS (FIG). 4-2
- USER PROGRAM EXITS. 4-1

FIELD
- APPLICATION PROGRAM/DATA FIELD DISPLAY INTERFACE, 4-3
- MOVTNG THE CURSOR FROM FIELD TO FIELD. 3-13
- PROTECTED AND UNPROTECTED FIELDS. 3-10

FIFLD CONTROL
- FIELD CONTROL, 3-15

FIELD MARK
- FIELD MARK FUNCTION, 3-15

FORMATTED
- FORMATTED DISPLAY. 4-3
- FORMATTED AND UNFORMATTED DISPLAYS, 3-10

FUNCTION CODE
- FUNCTION CODE CHARACTER, 4-7
- RETURN AND FUNCTION CODES, 4-7

FUNCTION KEYS
- FUNCTION KEYS (PF1 THROUGH PF24). 3-17

HELP
- HELP FUNCTION, 3-19

HOME
- HOME FUNCTION, 3-14

HOST RESPONSE
- HOST RESPONSE COMMAND. 4-9

IBM 3270
- IBM 3270 FUNCTIONS (TBL). 3-11

INSERT MODE
- INSERT MODE FUNCTION, 3-16

INTERACTIVE TERMINAL FACILITY
- SNA INTERACTIVE TERMINAL FACILITY CAPABILITIES, 1-2
- SNA INTERACTIVE TERMINAL FACILITY ENVIRONMENT (FIG), 1-3

INTERFACE
- APPLICATION PROGRAM/DATA FIELD DISPLAY INTERFACE, 4-3

INVOCATION
- SAMPLE ITF INVOCATION USING MOD 400 MENUS (FIG). 2-3

ITF
- LOGGIN IN TO THE ITF. 3-1

INDEX

ITF FUNCTIONS, LIST OF
- ALPHA OVERRIDE, 3-15
- BACKTAB, 3-14
- CLEAR, 3-17
- DELETE, 3-16
- DEVICE CANCEL, 3-17
- DUP, 3-15
- ENTER, 3-17
- ERASE INPUT, 3-15
- ERASE-TO-END-OF-FIELD, 3-15
- FIELD MARK, 3-15
- HELP, 3-19
- HOME, 3-14
- INSERT MODE, 3-16
- LOAD MATRIX, 3-19
- LOGOFF, 3-19
- NEW LINE, 3-14
- PRINT, 3-17
- RESET, 3-19
- SYSTEM REQUEST, 3-19
- TAB, 3-14

KEY EQUIVALENTS
- KEY EQUIVALENTS, 3-31
- WST/VIP7200 KEY EQUIVALENTS TO 3247 FUNCTIONS (TBL), 3-32
- WST/VIP7300 KEY EQUIVALENTS TO 3274 FUNCTIONS (TBL), 3-33
- WST/VIP7800 KEY EQUIVALENTS TO 3274 FUNCTIONS (TBL), 3-34

KEYBOARD FUNCTION
- KEYBOARD FUNCTIONS, 3-10

LANGUAGE
- ASSEMBLY LANGUAGE APPLICATION PROGRAM, 4-10

LOAD MATRIX
- LOAD MATRIX, 3-19

LOGICAL UNIT
- REQUEST HEADER, REQUEST UNIT, AND LOGICAL-UNIT ADDRESS, 4-9

LOGIN
- DPS 6 START-UP AND LOG-IN, 2-1

LOGIN (CONT)
- ITF LOGIN USING COMMANDS, 3-4
- ITF LOGIN USING MENUS, 3-2
- LOGGING IN TO THE ITF, 3-1
- SAMPLE ITF LOGIN (FIG), 3-3

LOGOFF
- LOGGING OFF THE ITF, 3-5
- LOGOFF, 3-19

MENU
- ITF LOGIN USING MENUS, 3-2

MESSAGES
- ITF PROGRAM MESSAGES, A-1

MOD 400 COMMAND
- STARTING UP THE ITF USING MOD 400 COMMANDS, 2-4

MOD 400 MENU
- SAMPLE ITF INVOCATION USING MOD 400 MENUS (FIG), 2-3
- STARTING UP THE ITF USING MOD 400 COMMANDS, 2-4

NEW LINE
- NEW LINE FUNCTION, 3-14

PRINT
- PRINT FUNCTION, 3-17

PROTECTED FIELD
- PROTECTED AND UNPROTECTED FIELDS, 3-10

REQUEST HEADER
- REQUEST HEADER, REQUEST UNIT, AND LOGICAL-UNIT ADDRESS, 4-9

RESET
- RESET FUNCTION, 3-19

RETURN CODE
- RETURN CODE, 4-7

SCREEN IMAGE BUFFER
- SCREEN IMAGE BUFFER (SIB), 4-6

INDEX

- SCREEN IMAGE DATA AREA
  - SCREEN IMAGE DATA AREA (SIDA), 4-6
- SHUTTING DOWN
  - SHUTTING DOWN THE ITF, 2-4
- SIB
  - (SEE SCREEN IMAGE BUFFER)
- SIDA
  - (SEE SCREEN IMAGE DATA AREA)
- STANDARD KEYBOARD
  - WST/VIP7200 STANDARD KEYBOARD (FIG), 3-22
  - WST/VIP7300 STANDARD KEYBOARD (FIG), 3-25
  - WST/VIP7800 STANDARD KEYBOARD (FIG), 3-29
- START-UP
  - DPS 6 START-UP AND LOG-IN, 2-1
  - STARTING UP THE ITF USING MOD 400 COMMANDS, 2-4
  - STARTING UP THE ITF USING MOD 400 MENUS, 2-2
  - STARTING UP AND SHUTTING DOWN THE SNA INTERACTIVE TERMINAL FACILITY, 2-1
- STATUS LINE
  - STATUS LINE DISPLAYS (FIG), 3-8
  - STATUS LINE INDICATORS (TBL), 3-5
- SYSTEM REQUEST
  - SYSTEM REQUEST FUNCTION, 3-19
- TAB FORWARD
  - TAB FORWARD FUNCTION, 3-14
- TERMINAL
  - ITF-SUPPORTED TERMINALS, 3-20
  - TERMINAL IDENTIFICATION, 3-8
- TERMINAL (CONT)
  - USING YOUR TERMINAL FOR DATA ENTRY, 3-5
- UNFORMATTED DISPLAY
  - FORMATTED AND UNFORMATTED DISPLAYS, 3-10
  - UNFORMATTED DISPLAY, 4-3
- UNPROTECTED FIELD
  - PROTECTED AND UNPROTECTED FIELDS, 3-10
- USER PROGRAM
  - USER PROGRAM EXITS, 4-1
  - USER PROGRAM INTERFACE AREA (UPIA), 4-6
- WORD PROCESSING KEYBOARD
  - WST/VIP7300 WORD PROCESSING KEYBOARD (FIG), 3-16
  - WST/VIP7800 WORD PROCESSING KEYBOARD (FIG), 3-30
- WST/VIP7200
  - WST/VIP7200 DATA ENTRY KEYBOARD (FIG), 3-23
  - WST/VIP7200 KEY EQUIVALENTS TO 3247 FUNCTIONS (TBL), 3-32
  - WST/VIP7200 STANDARD KEYBOARD (FIG), 3-22
- WST/VIP7300
  - WST/VIP7300 DATA ENTRY KEYBOARD (FIG) 3-27
  - WST/VIP7300 KEY EQUIVALENTS TO 3274 FUNCTIONS (TBL), 3-33
  - WST/VIP7300 STANDARD KEYBOARD (FIG), 3-25
  - WST/VIP7300 WORD PROCESSING KEYBOARD (FIG), 3-16
- WST/VIP7800
  - WST/VIP7800 KEY EQUIVALENTS TO 3274 FUNCTIONS (TBL), 3-34
  - WST/VIP7800 STANDARD KEYBOARD (FIG), 3-29
  - WST/VIP7800 WORD PROCESSING KEYBOARD (FIG), 3-30

HONEYWELL INFORMATION SYSTEMS

Technical Publications Remarks Form

TITLE: DPS 6 & LEVEL 6
SNA INTERACTIVE TERMINAL
FACILITY USER'S GUIDE

ORDER NO. CR58-01

DATED APRIL 1983

ERRORS IN PUBLICATION

SUGGESTIONS FOR IMPROVEMENT TO PUBLICATION

Your comments will be investigated by appropriate technical personnel and action will be taken as required. Receipt of all forms will be acknowledged; however, if you require a detailed reply, check here. □

FROM: NAME ______________________ DATE ____________

TITLE ______________________

COMPANY ______________________

ADDRESS ______________________

____

PLEASE FOLD AND TAPE–
NOTE: U. S. Postal Service will not deliver stapled forms

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 39531 WALTHAM, MA02154

POSTAGE WILL BE PAID BY ADDRESSEE

HONEYWELL INFORMATION SYSTEMS
200 SMITH STREET
WALTHAM, MA 02154

ATTN: PUBLICATIONS, MS486

Honeywell

HONEYWELL INFORMATION SYSTEMS
Technical Publications Remarks Form

TITLE: DPS 6 & LEVEL 6
SNA INTERACTIVE TERMINAL
FACILITY USER'S GUIDE

ORDER NO. CR58-01

DATED APRIL 1983

ERRORS IN PUBLICATION

SUGGESTIONS FOR IMPROVEMENT TO PUBLICATION

Your comments will be investigated by appropriate technical personnel and action will be taken as required. Receipt of all forms will be acknowledged; however, if you require a detailed reply, check here. □

FROM: NAME ______________________________ DATE ______________

TITLE ______________________________

COMPANY ______________________________

ADDRESS ______________________________

____

PLEASE FOLD AND TAPE—
NOTE: U. S. Postal Service will not deliver stapled forms

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 39531 WALTHAM, MA02154

POSTAGE WILL BE PAID BY ADDRESSEE

HONEYWELL INFORMATION SYSTEMS
200 SMITH STREET
WALTHAM, MA 02154

ATTN: PUBLICATIONS, MS486

Honeywell

Together, we can find the answers.

Honeywell

Honeywell Information Systems
U.S.A.: 200 Smith St., MS 486, Waltham, MA 02154
Canada: 155 Gordon Baker Rd., Willowdale, ON M2H 3N7
U.K.: Great West Rd., Brentford, Middlesex TW8 9DH **Italy**: 32 Via Pirelli, 20124 Milano
Mexico: Avenida Nuevo Leon 250, Mexico 11, D.F. **Japan**: 2-2 Jinbou-Cho Kanda, Chiyoda-Ku Tokyo
Australia: 124 Walker St., North Sydney, N.S.W. 2060 **S.E. Asia**: Mandarin Plaza, Tsimshatsui East, H.K.

37075, 1.5583, Printed in U.S.A.