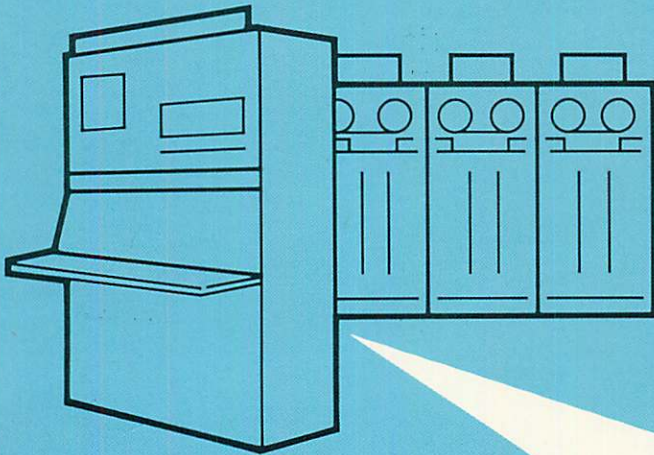


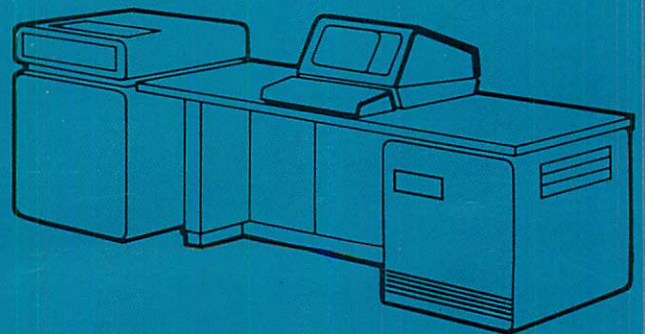
Using and Managing IML/3000



Course Slides and Text



IML/3000



HP 3000 Data Communications

**Using and Managing
IML/3000**

Course Slides and Text



**HEWLETT
PACKARD**

19447 Pruneridge Avenue, Cupertino, California 95014

NOTICE

The information contained in this document is subject to change without notice.

HEWLETT-PACKARD MAKES NO WARRANTY OF ANY KIND WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Hewlett-Packard shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance or use of this material.

Hewlett-Packard assumes no responsibility for the use or reliability of its software on equipment that is not furnished by Hewlett-Packard.

This document contains proprietary information which is protected by copyright. All rights are reserved. No part of this document may be photocopied, reproduced or translated to another program language without the prior written consent of Hewlett-Packard Company.

LIST OF EFFECTIVE PAGES

The List of Effective Pages gives the date of the current edition and of any pages changed in updates to that edition. Within the manual, any page changed since the last edition is indicated by printing the date the changes were made on the bottom of the page. Changes are marked with a vertical bar in the margin. If an update is incorporated when an edition is reprinted, these bars are removed but the dates remain.

First Edition. Nov 1980

PRINTING HISTORY

New editions are complete revisions of the manual. Update packages, which are issued between editions, contain additional and replacement pages to be merged into the manual by the customer. The date on the title page and back cover of the manual changes only when a new edition is published. When an edition is reprinted, all the prior updates to the edition are incorporated. No information is incorporated into a reprinting unless it appears as a prior update. The edition does not change.

The software product part number printed alongside the date indicates the version and update level of the software product at the time the manual edition or update was issued. Many product updates and fixes do not require manual changes, and conversely, manual corrections may be done without accompanying product changes. Therefore, do not expect a one to one correspondence between product updates and manual updates.

First Edition. Nov 1980

USING AND MANAGING IML/3000 is a course in the use and management of Hewlett-Packard's IML/3000. The intended audience is:

- o Personnel who will be responsible for setting up and maintaining the IML/3000 operating conditions on the HP 3000 ("IML/3000 Managers"),
- o Personnel who will be authorized to control communications with the host at the HP 3000 end ("IML/3000 Console Operators"), and
- o Personnel who will write application programs using IML/3000 Intrinsic and/or use the Inquiry and Development Facility for interactive access to the host ("IML/3000 Users"),

The student should expect, at the end of the course to:

- o Have a general understanding of the environment in which IML/3000 operates, the basic software components of the product, and how IML/3000 processes data
- o Be able to prepare an IML/3000 configuration file
- o Be able to control communications activity on an IML line (at the HP 3000 end)
- o Be able to trace communications line activity (and, optionally, to interpret general information from a trace listing)
- o Be able to operate IDF on an HP terminal and printer
- o Be able to write an application program using IML/3000 Intrinsic that will interact with an application on the host. (This assumes the student already knows how to program in COBOL, FORTRAN, BASIC, or SPL.)

Preface

Course materials are:

- o USING AND MANAGING IML/3000, a textbook containing reproductions of the overhead transparencies used in teaching the class, and explanatory text. USING AND MANAGING IML/3000 intended for use in class concurrently with lectures presented by a Hewlett-Packard instructor. The illustrations in the book are copies of the overhead slides the instructor will use in class. Accompanying each illustration is explanatory text; in most cases this text will be sufficiently comprehensive to make extensive note-taking unnecessary.
- o Additional handouts to be supplied by the instructor.
- o IML/3000 INTERACTIVE MAINFRAME LINK REFERENCE MANUAL, the primary product reference document. The student should have access to this manual during the course.

Contents

Introduction.....	7
Managing IML/3000.....	37
Inquiry and Development Facility.....	79
IDF: Introduction.....	81
IDF: Display Terminals.....	93
IDF: Printing Devices.....	117
Programming IML/3000.....	127
Programming Overview.....	129
Sample IML/3000 Application.....	167
Additional Considerations.....	213
CS Trace: Supplement.....	215
PDBIML Source Code Listing	
Blank Screen ID Form	
Blank Host Application Questionare	

USING AND MANAGING IML/3000

Using and Managing IML/3000

iml/3000

Introduction

- **Features of IML/3000**
- **Hardware**
- **Software**
- **Intelligent Network Processor (INP)
and IML/3000 control unit software**
- **Data flow**
- **Internal screen image buffer**

Introduction

Features of IML/3000

- **Programmatic emulation of 3270**
- **Inquiry and Development Facility (IDF)**
- **Configuration files**
- **Management capability**
- **Security provisions**

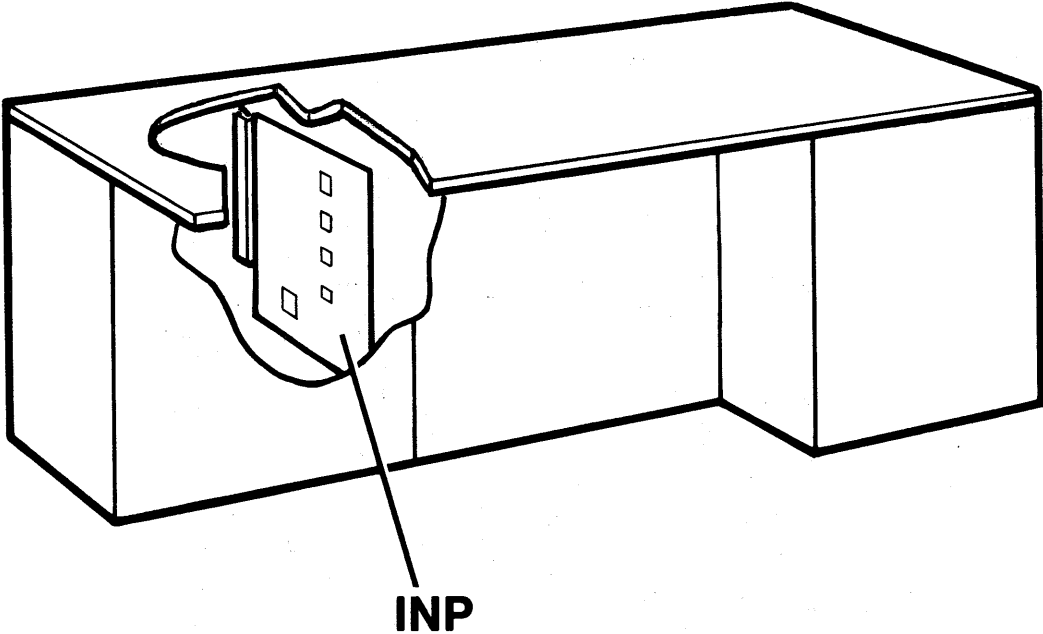
Features of IML/3000

- o Programmatic emulation of 3270 devices via IML/3000 Intrinsic
 - * IML/3000 provides PTOP (Program TO Program) capability between a program executing on an HP 3000 and a program executing on a host system.
 - * An HP 3000 program uses the IML Intrinsic to invoke PTOP communication. The HP 3000 program may be written in:
 - BASIC
 - COBOL
 - COBOL II
 - FORTRAN
 - SPL
- o Inquiry and Development Facility
 - * Permits "pass through" emulation of 3270 terminals and printers on HP devices
 - * Is an HP-supplied application
 - part of IML/3000 software
 - no user programming necessary to run IDF
- o IML/3000 configuration files
 - * Define 3270 devices that will be emulated when IML/3000 is started
 - * Define IML/3000 security
- o Management of IML/3000 provided via
 - * IMLMGR subsystem and its subcommands
 - * :IMLCONTROL console command
- o IML/3000 security provisions
 - * Augment standard MPE security
 - * prevent unauthorized access to the host system
 - * defined in the configuration file

introduction

HP 3000 CPU and INP

HP 3000 CPU

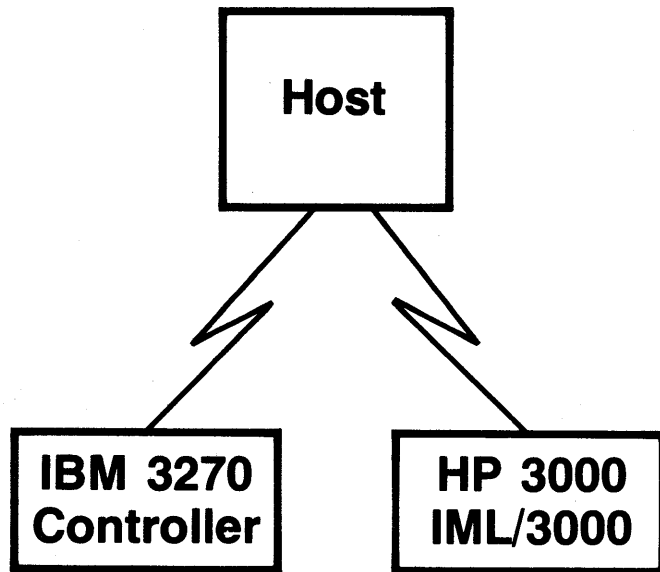


HP 3000 CPU and INP

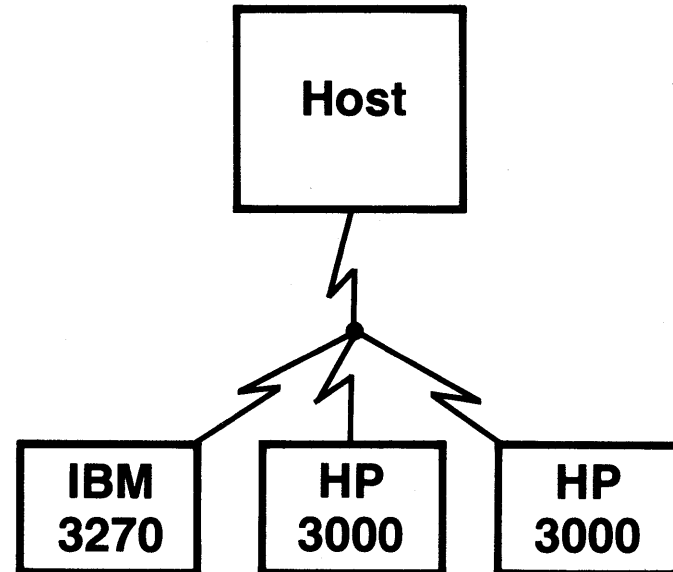
- o HP 3000 CPU
 - * may be Series II, III, 30, 33, or 44
 - * connected to host via modem link
- o INP (Intelligent Network Processor)
 - * Is HP 3000 communications controller used by IML/3000
 - * Performs all of communications line activity
 - * Offloads line handling overhead from the HP 3000 CPU

Data Link Configurations

Point-To-Point



Multi-Drop

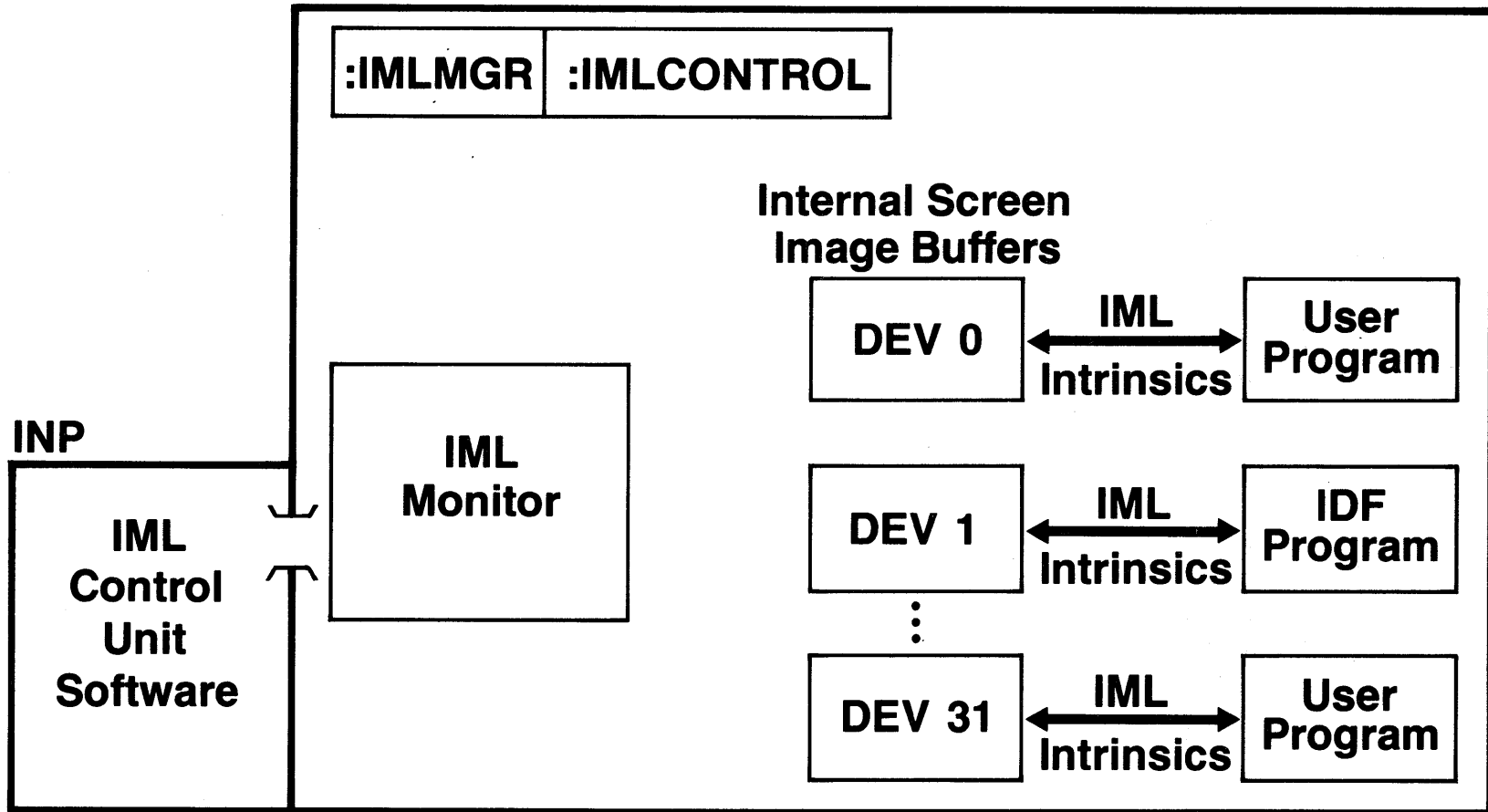


Data Link Configurations

- o Communications line must be private (leased)
- o Configurations may be point-to-point or multi-drop
- o In multi-drop configuration, IML/3000 can operate on one or more drops while IBM 3270's operate on one or more other drops

Software

HP 3000

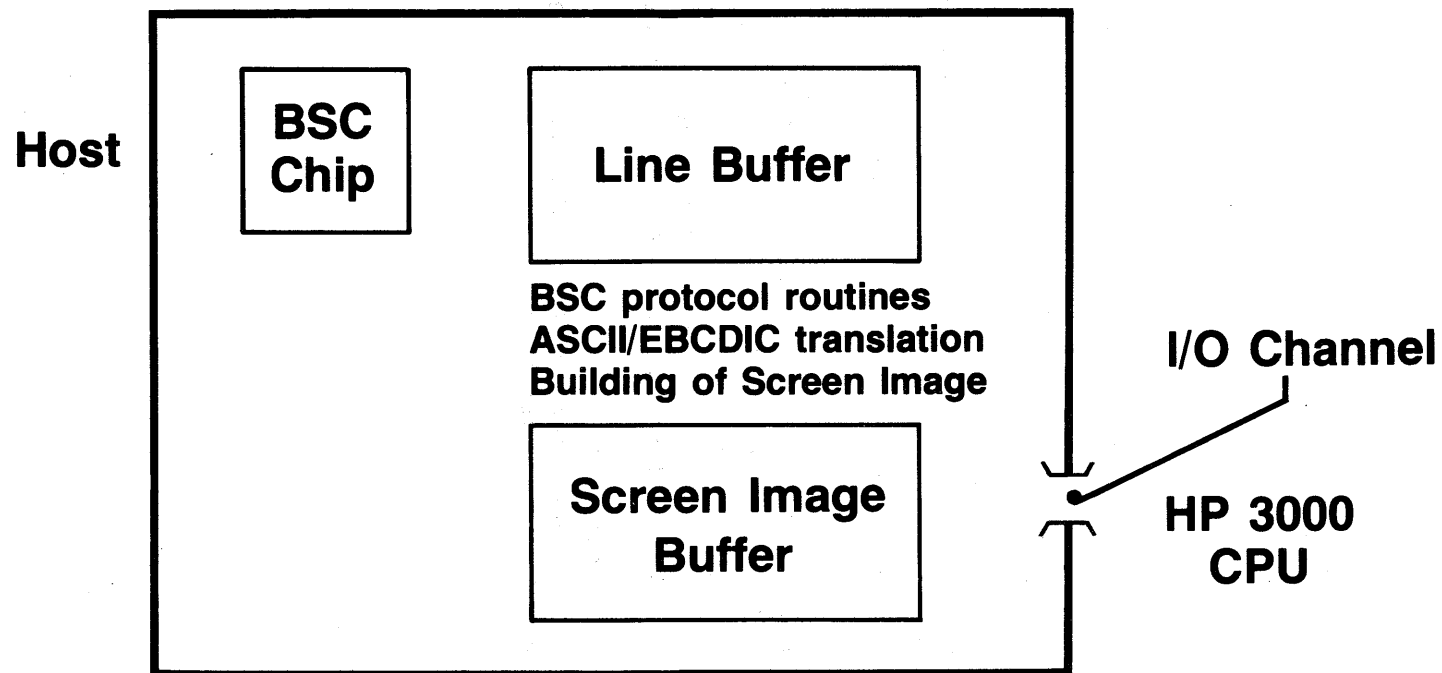


Software

- o IML/3000 Control Unit Software
 - * Performs most of 3270 Control Unit functions
 - * Downloaded to INP when IML/3000 started
- o IMLMGR subsystem and IMLCONTROL commands
 - * provide authorized users control of IML/3000 processes
- o IML Monitor
 - * directs flow of data between INP and one or more internal screen image buffers
 - * processes START, STOP, KILL and TRACE commands
- o Internal Screen Image Buffers
 - * Monitor allocates an MPE extra data segment for each IML/3000 user program
 - extra data segment serves as buffer for user's screen data
 - contains character-for-character copy of the emulated 3270 device screen, called the internal screen image
 - * Internal screen image buffers are accessible to Monitor and to user's program (the latter via the Intrinsic)
 - * There can be as many as 32 internal screen image buffers per IML monitor, one per emulated 3270 device
- o The IML/3000 Intrinsic
 - * used to pass data between the internal screen image buffer and the user program

introduction

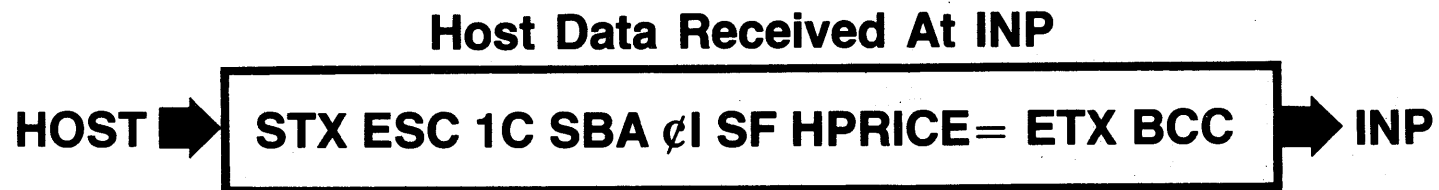
INP and IML/3000 Control Unit Software



INP and IML/3000 Control Unit Software

- o BSC Chip
 - * Does sync-searching for the INP
 - * Handles serial-to-parallel and parallel-to-serial bit conversions
- o Line Buffer holds data received from host and data ready for transmission to host
- o IML Control Unit Software and other software on the INP
 - * Execute BSC protocol routines
 - respond to host system polls and selects
 - do block check calculations
 - strip protocol characters out of data streams (STX,ETX, etc.)
 - * Translate incoming EBCDIC data to ASCII and outgoing ASCII to EBCDIC (for EBCDIC lines)
 - * Interpret 3270 commands and orders to build screen images in screen image buffer on the INP
- o INP Screen Image Buffer
 - * Contains formatted image of user's data screen
 - * When Monitor copies screen image to user's extra data segment, it copies the image from the INP Screen Image Buffer

Data Flow with Sample Data



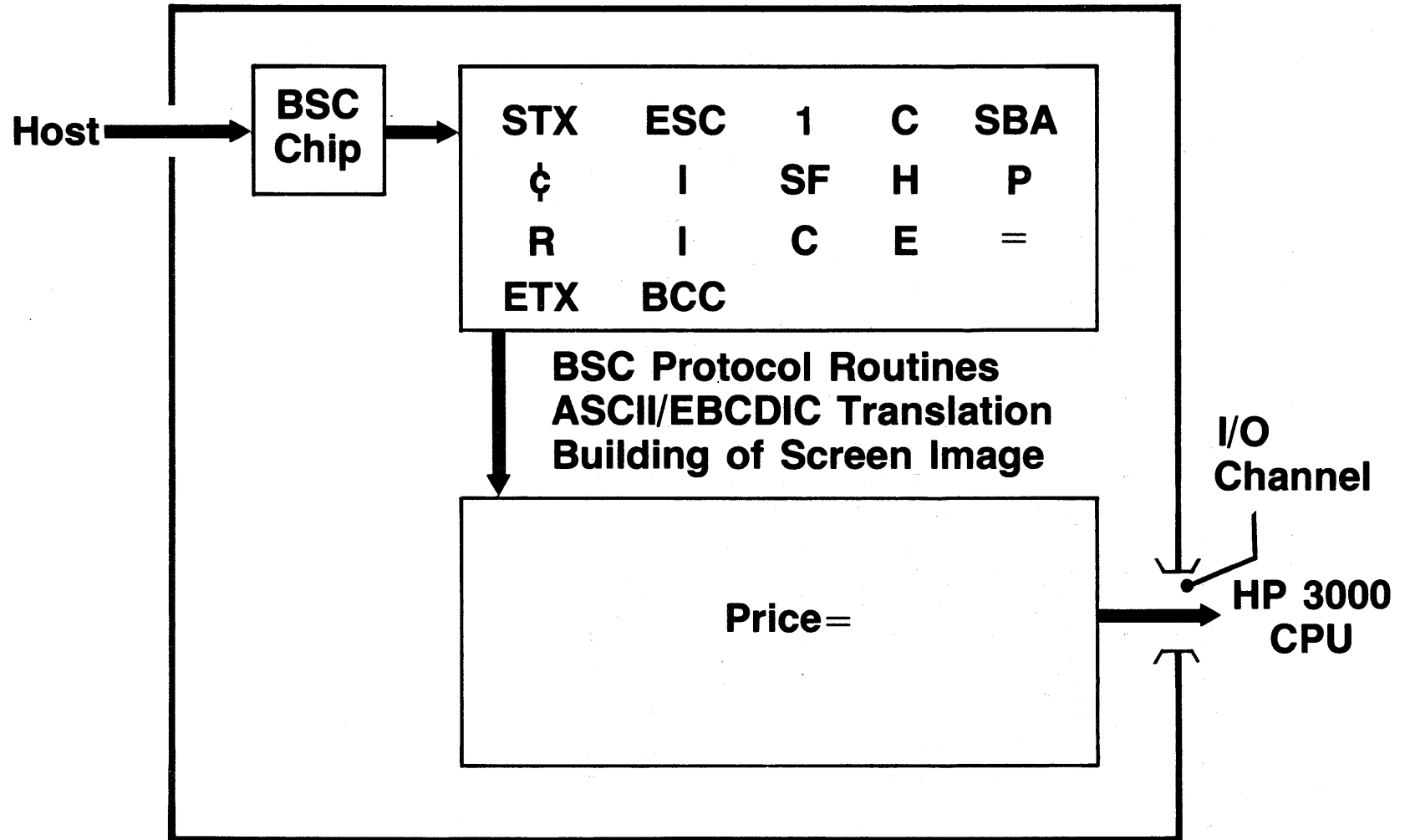
Data Flow with Sample Data

In the following slides we will examine flow of data from the host to the user's program and back to the host.

- o Data stream received from host contains
 - * Data
 - * BSC protocol/3270 framing characters
eg: STX, ETX, BCC
 - * 3270 commands and orders
eg: SBA, SF, WCC

- o Interpreting this example:
 - * STX = BSC Start-of-text character
 - * ESC = 3270 framing character
 - * 1 = 3270 Write command
 - * C = Write Control Character = binary 11000011 (EBCDIC)
= enable keyboard, reset MDT bits
 - * SBA = Set Buffer Address order
 - * (cent) I = buffer address = buffer position 649
 - * SF = Start Field order
 - * H = Attribute Byte = binary 11001000 = unprotected field,
alphanumeric data, intensified display
 - * PRICE= = Data sent to user's screen
 - * ETX = BSC End-of-text character
 - * BCC = Block check character (actually two characters)

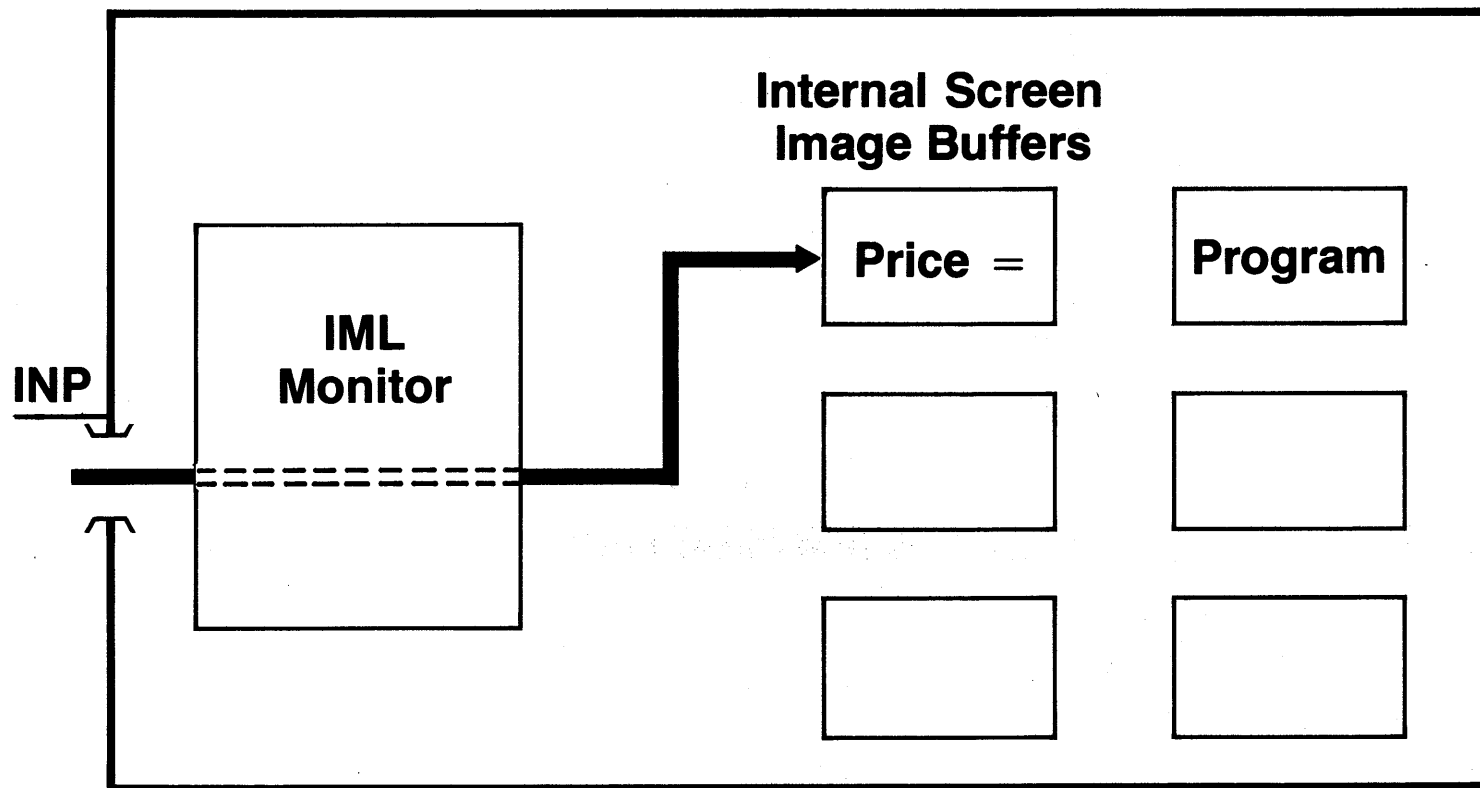
Data Flow Through INP



Data Flow Through INP

- o The bit serial data is received at the BSC chip and converted to characters.
- o The characters are stored in the line buffer.
- o The BSC protocol and 3270 framing characters and 3270 commands and orders are stripped from the data stream.
- o The data is translated into ASCII and written into the INP screen image buffer.
 - * The screen image build routines use the commands and orders in the 3270 data stream to write data into appropriate positions in the screen image buffer.

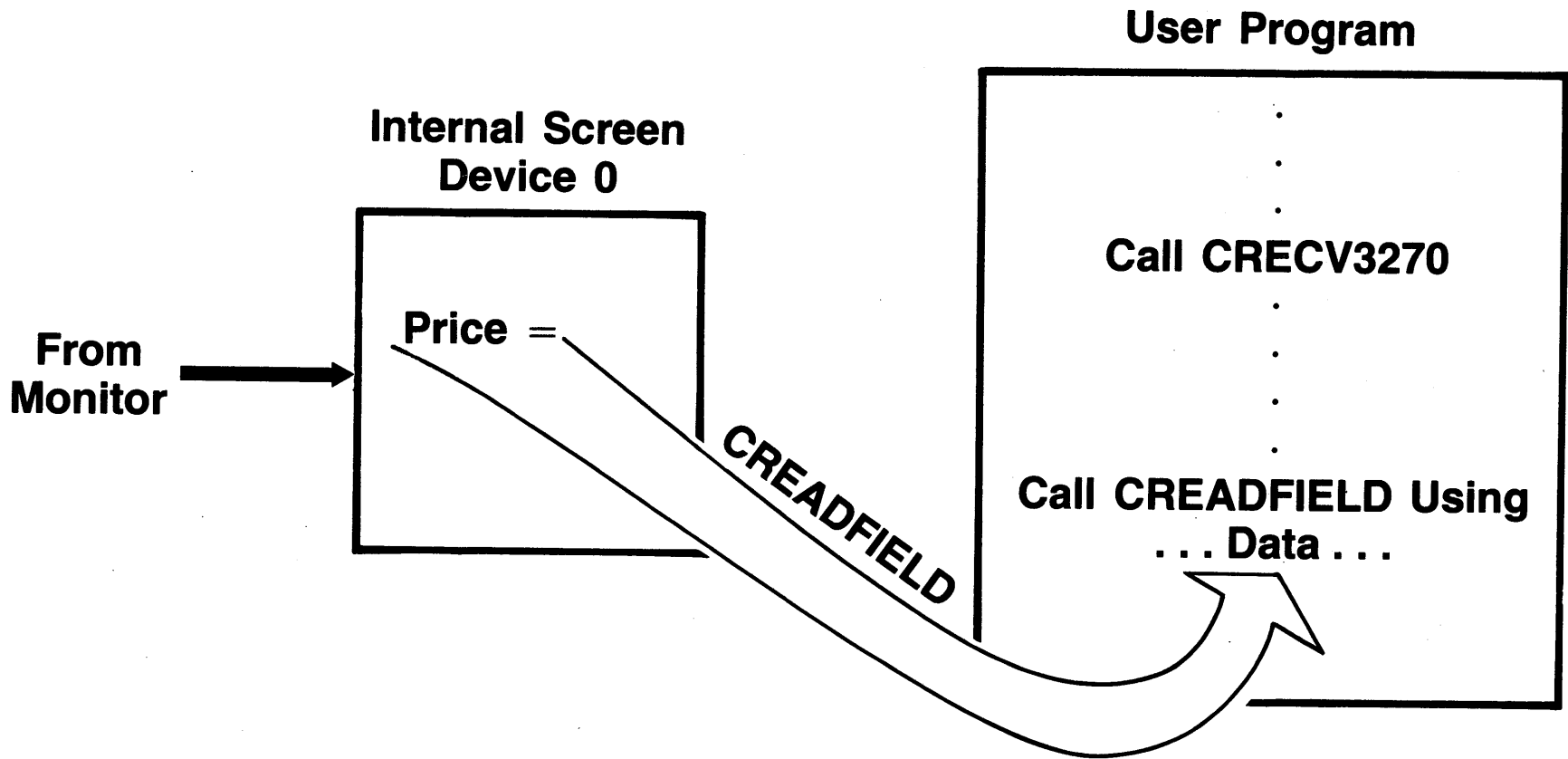
Data Flow from INP to Internal Screen Image Buffer



Data Flow from INP to Internal Screen Image Buffer

- o Screen image copied from INP screen image buffer to the HP 3000 main memory internal screen image buffer
 - * Monitor directs copy action
 - * User's program can then access the data with the IML intrinsics.

Data Flow from Internal Screen Image Buffer to User Program



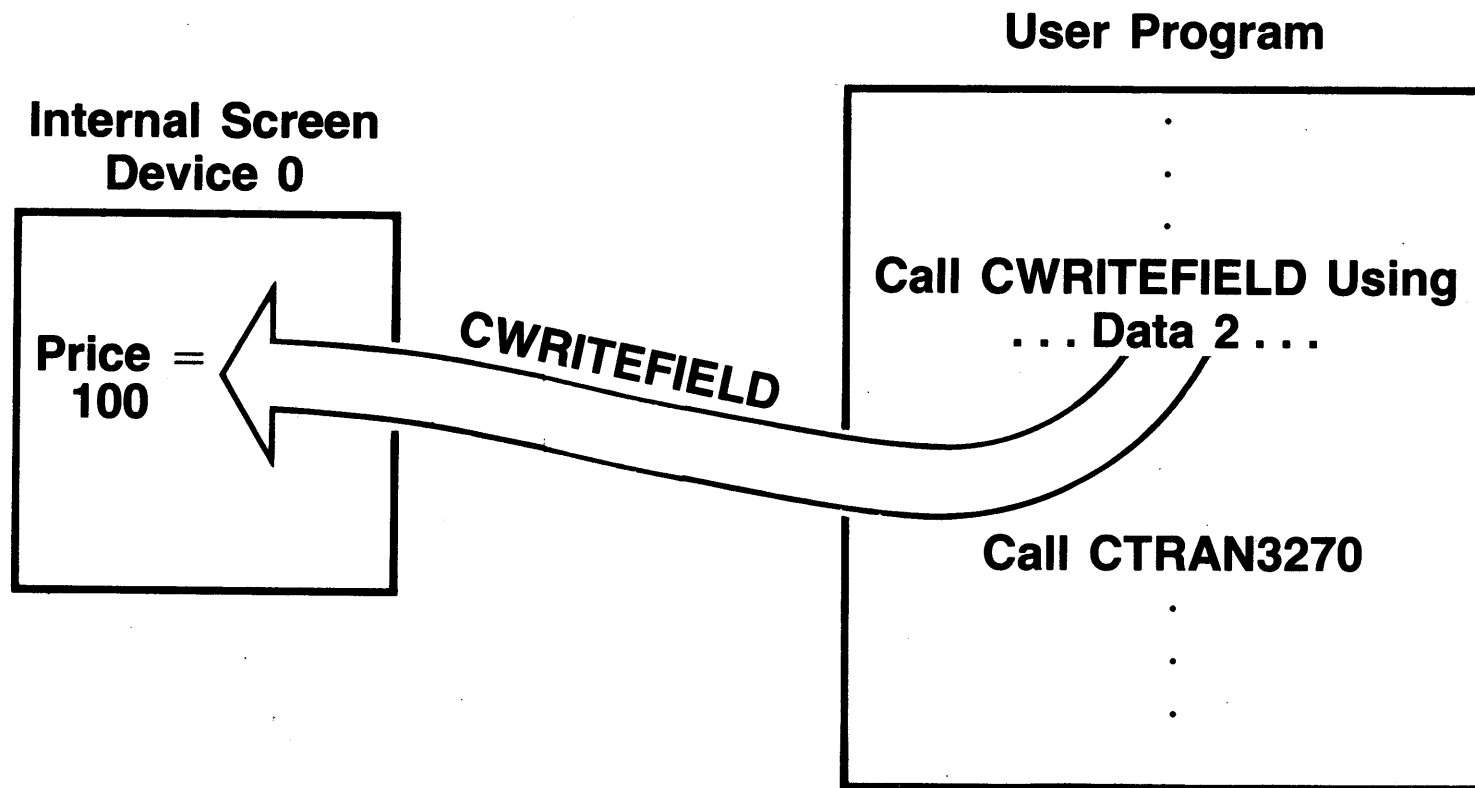
Intrinsics provide access to internal screen buffers

Data Flow from Internal
Screen Image Buffer to User's Program

- o User's program calls CRECV3270 (COBOL version of RECV3270)
- o Program then calls CREADFIELD. The COBOL variable DATA now has the value "PRICE=".

introduction

Data Flow from User Program to Internal Screen Image Buffer

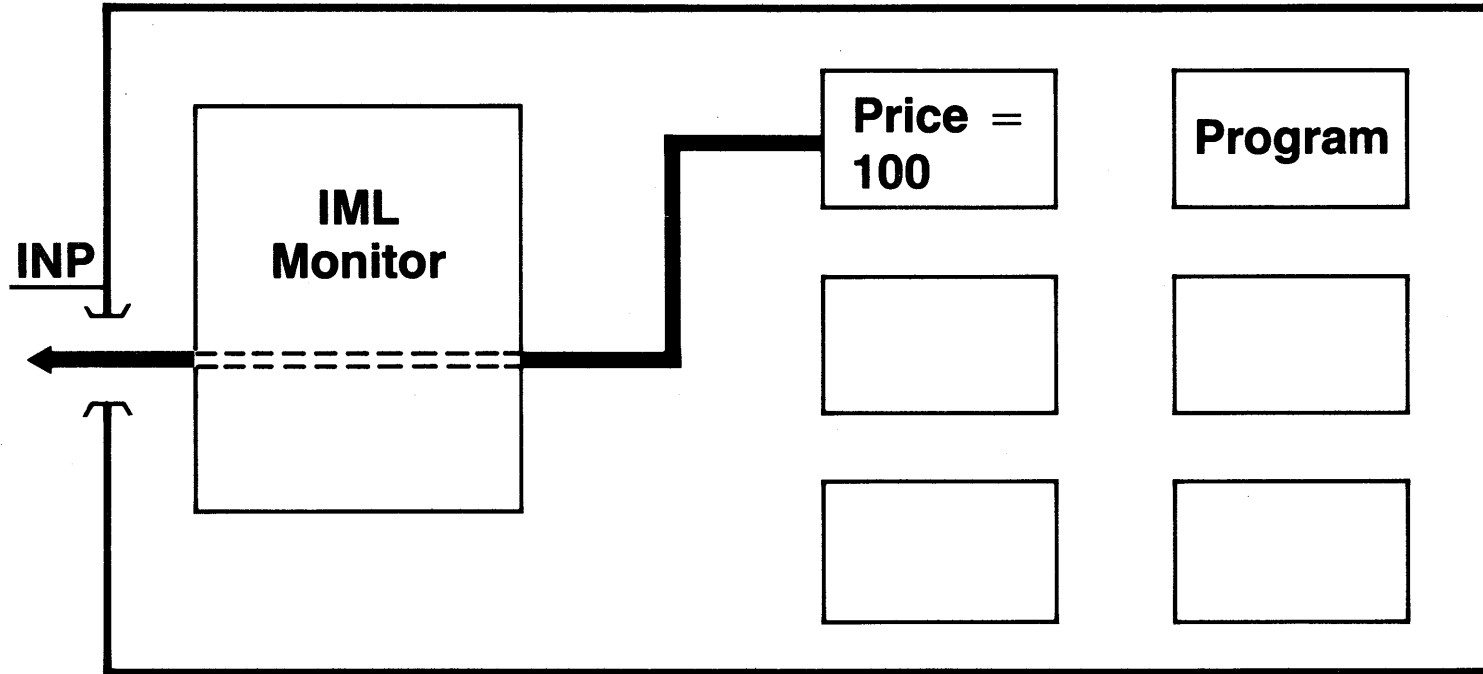


Data Flow from User Program
to Internal Screen Image Buffer

- o User's program writes new data to buffer
 - * Assume value of COBOL variable DATA2 is "100".
 - * CWRITEFIELD writes this into the internal screen image.
- o User's program then calls CTRAN3270 (COBOL version of TRAN3270) to transmit this to the host system.

introduction

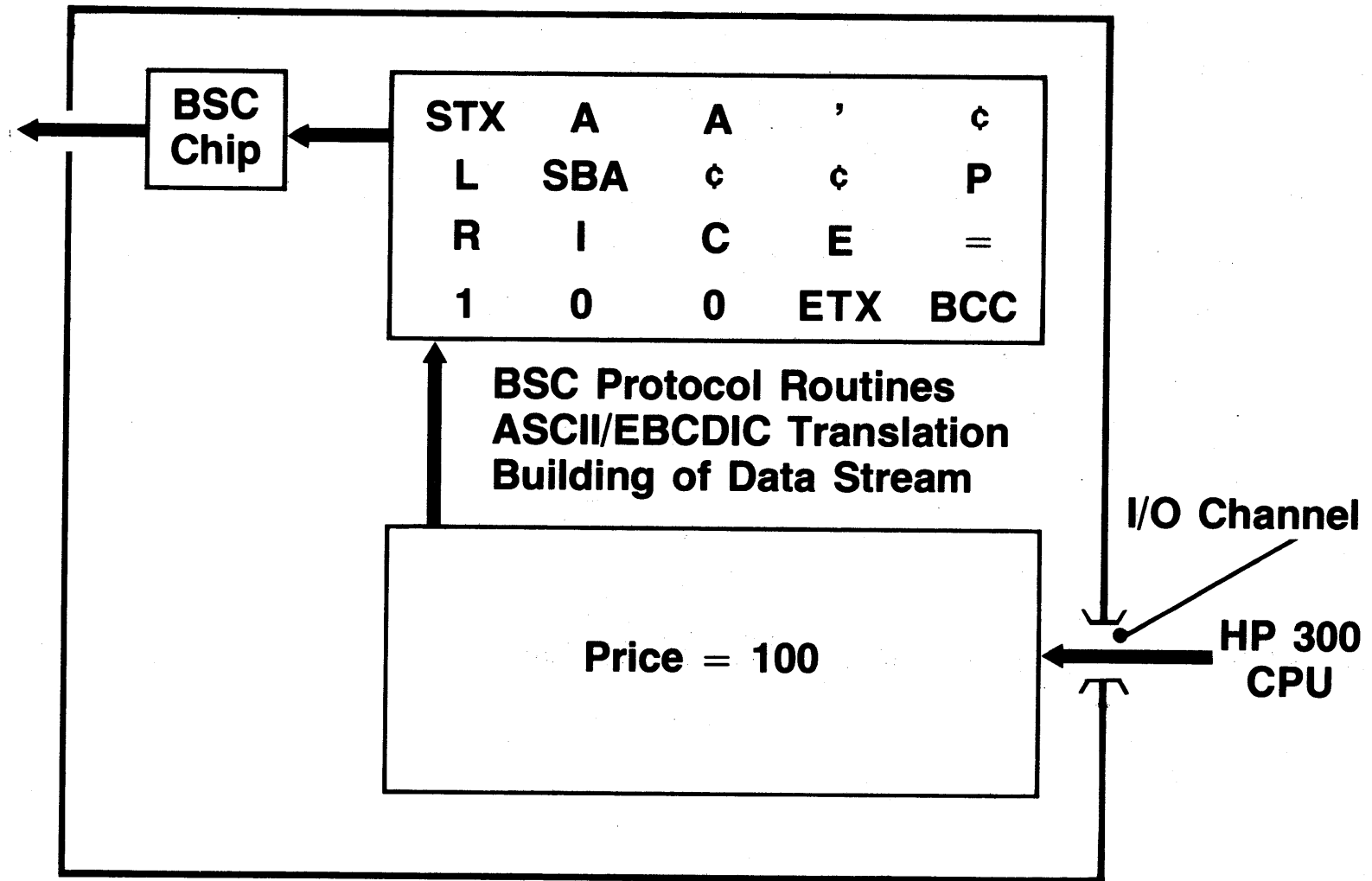
Data Flow from Internal Screen Image Buffer to INP



Data Flow from Internal Screen Image Buffer to INP

- o When CTRAN3270 is called
 - * Monitor is informed that screen image is ready for transmission
 - Monitor locks extra data segment, queues the transmission request (if others are pending), informs Control Unit software on INP that screen image is waiting for transmission
 - when host polls device, Monitor copies image to INP screen buffer

Data Flow Through INP



Data Flow Through INP

- o IML software and INP code build data stream in Line Buffer
 - * A read by poll occurs, resulting in a Read Modified operation being performed
 - * A data stream is built in the line buffer with the appropriate BSC protocol characters and 3270 AID and order codes.
 - * The BSC chip converts this to bit serial format and transmits the text to the host system.

introduction

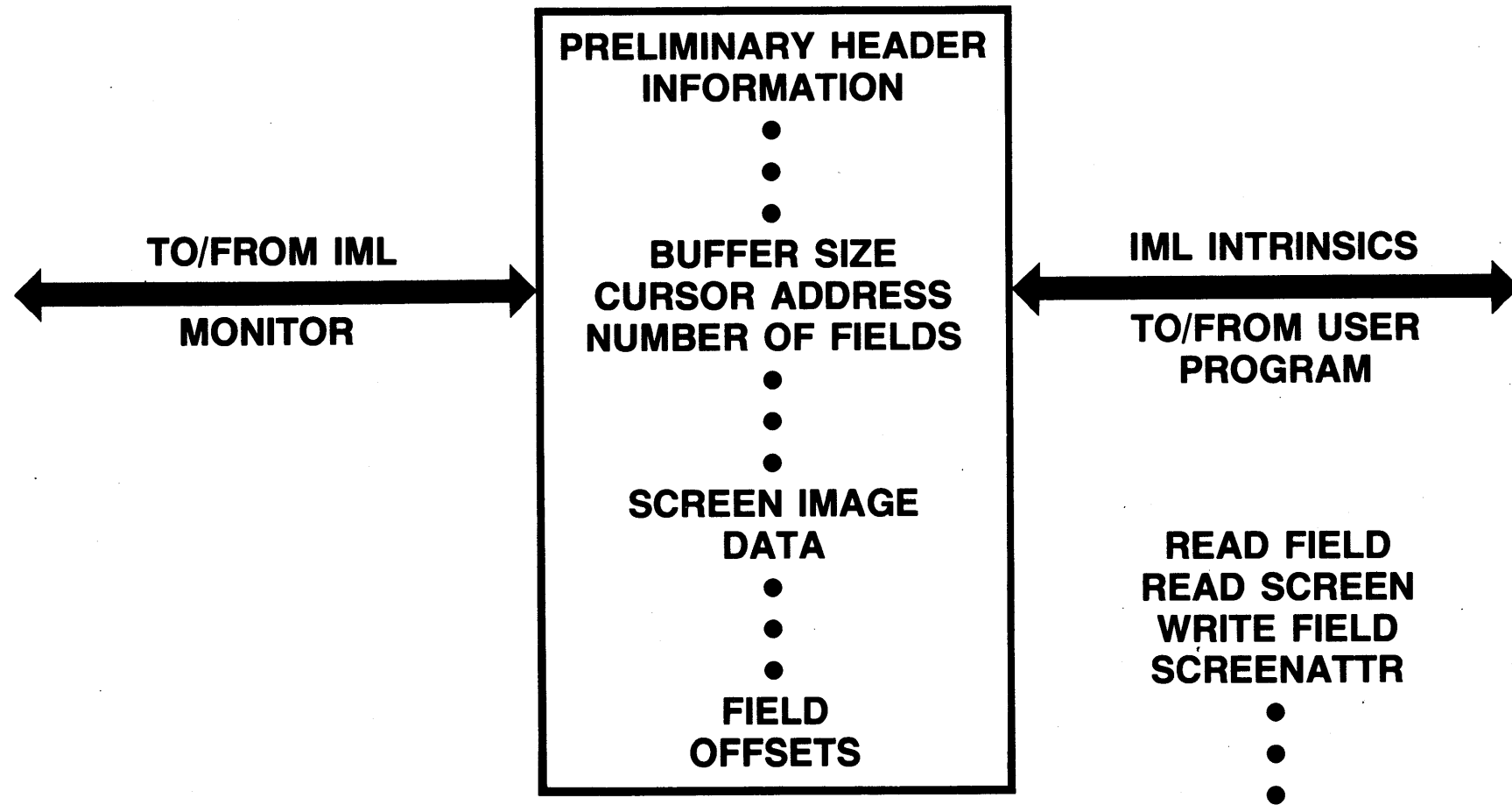
Data Stream Sent to Host



Data Stream Sent to Host

- o STX = start of text
- o AA = address of this terminal (control unit 1, device 1)
- o ' = AID code = ENTER
- o (cent) L = cursor address = buffer position 659
- o SBA = set buffer address
- o (cent) (cent) = address of modified field
= buffer position 650
- o PRICE=100 = contents of modified field, with nulls compressed
out
- o ETX = end of text
- o BCC = block check character (actually two characters if line
code is EBCDIC).

The Internal Screen Image Buffer



The Internal Screen Image Buffer

- o The internal screen image is a representation of 3270 device buffer data
- o The internal screen image buffer holds the image for access by user program
- o The internal screen image buffer contains:
 - * preliminary header information
 - * buffer size, current cursor address, number of fields in data screen
 - * a 480 or 1920 byte character by character representation of a 3270 screen image
 - * field offsets within screen data
- o An IML application program uses the IML intrinsics to access the internal screen image in much the same way as a 3270 user uses the screen and keyboard of a 3270 device to access information in the 3270 terminal buffer.

Managing IML/3000

- **Configuring IML/3000 into MPE I/O system**
- **Configuration files**
- **IMLMGR subsystem**
- **IMLCONTROL console commands**
- **IML/3000 traces, dumps**

Managing IML/3000

- o Management function distinct from user function
- o Management responsibilities include:
 - * Building IML/3000 configuration files
 - * Opening and closing communications link
 - * Handling line traces and INP/IML Monitor dumps
 - * May also include configuring IML/3000 into MPE I/O system
- o Management responsibilities can be shared between IML Manager and Console Operators

Configuring IML/3000 into MPE I/O System

Refer to:

■ **IML/3000 Reference Manual**

— or —

■ **HP 3000 System Manager/System Supervisor
Reference Manual**

managing iml/3000

Configuring IML/3000 into MPE I/O System

IML/3000 Configuration Files

A configuration file defines:

- **Which INP/communications line**
- **What bisync polling addresses**
- **How many/what kind 3270 terminals are emulated**
- **Any automatic IDF devices**
- **IML/3000 security**

IML/3000 Configuration Files

- o Each file is associated with a specific pseudo device and INP
 - * Only one configuration file can be in use with given INP at a time
- o Each file defines a "Control Unit" and "attached devices"
 - * Lists polling addresses for Control Unit and devices
 - addresses are same as configured on host
 - * Defines characteristics of devices--IBM model number, buffer size, etc.
- o Identifies devices on which IDF will be started automatically when IML/3000 started
- o Defines which userid's, programs, MPE ldev's will have access to which 3270 device numbers

IML/3000 Configuration Files

- **Are 80-byte sequential ASCII files**
- **Are built outside IML/3000 (via a text editor)**
- **Have three main sections:**
 - control unit definition**
 - device definitions**
 - security specifications (optional)**

IML/3000 Configuration Files

- o Three main sections:
 - * Control Unit definition
 - * Device definitions
 - * Security specifications
- o Each line in configuration file may consist of:
 - * Statement keywords
 - * Statement parameters
 - * Comments
- o Statement keywords
 - * Are: LDEV, CONTROL UNIT, CODE, ASCII, EBCDIC, MESSAGE, DEVICE TYPE, BUFFER, SIZE, PAGE, ON, ALLOW, and AUTO ACQUIRE
 - * May be abbreviated to only those characters needed to distinguish them from other keywords
 - * Each statement must begin on new line
 - * Only ON/ALLOW and DEVICE statements may extend over multiple lines
 - * ON/ALLOW and DEVICE statements must end with semi-colon
- o Statement parameters
 - * Must be followed by at least one space
 - * An asterisk and comment may follow the space
- o Comments
 - * May occupy part or all of any line
 - * Asterisk anywhere in configuration file denotes start of comment
 - * ILMGR ignores any character string (within a line) preceded by an asterisk
- o Configuration files have usual MPE file security except can't use LOCKWORDS.
- o HP recommends keeping configuration files in PUB.SYS
 - * All IML/3000 users need read access
 - * Console Operator needs read access in order to use IMLCONTROL commands

Configuration File: Defining Control Unit

File name: IMLCONF.PUB.SYS

*SAMPLE IML/3000 CONFIGURATION FILE

*

LDEV 100 19

*

CONTROL UNIT 13⁰ = F

*

CODE EBCDIC

*

MESSAGE IML LINE TO CHICAGO DATA CENTER NOW OPEN

•
•
•

Configuration File: Defining Control Unit

EXAMPLE

- Filename is IMLCONF.PUB.SYS
 - * in PUB.SYS so all IML users have read access to it
- LDEV statement
 - * Identifies MPE logical device number of IML pseudo device
 - * IML pseudo device has a referback in the MPE I/O configuration to the INP which it will use.
- CONTROL UNIT statement defines the BSC address of the emulated 3270 control unit. Control unit 13 corresponds to BSC address character F.
- CODE EBCDIC statement specifies the line code
- MESSAGE statement contains a message that will be displayed on system console when IML/3000 is started

Configuration File: Defining Devices

```
      .  
      .  
      .  
DEVICE          0;  
TYPE           3278  
BUFFER SIZE    1920  
AUTO ACQUIRE  50  
*  
DEVICE          25;  
TYPE           3287  
PAGE SIZE    66  
AUTO ACQUIRE  6  
      .  
      .  
      .
```

device 50 must be 500

Configuration Files: Defining Devices

EXAMPLE (cont.)

- o DEVICE statement
 - * Defines device number(s) for an emulated 3270 display or printer
 - * May be any integer from 0 to 31
 - * May specify more than one at a time. Statement must end in a semi-colon.
- o TYPE statement
 - * Defines the model type of the device(s) being defined
 - model numbers 3277 or 3278 for displays
 - model numbers 3284, 3286, 3287, 3288, or 3289 for printers
 - * Statement is optional
 - default is 3277
- o BUFFER SIZE statement
 - * Defines screen size for device
 - * May be 480 or 1920
 - * May not be switched dynamically
 - IML/3000 treats ERASE/WRITE ALTERNATE like ERASE/WRITE
 - * Statement is optional
 - default is 1920
- o AUTO ACQUIRE statement
 - * Number in statement is MPE logical device number of HP 3000 terminal or printer
 - * When IML/3000 is started using file, IDF will be started automatically on ldev
 - if device not currently in use
 - if in use, IML/3000 will not try again to start IDF on it
 - * MPE logical device 50 will emulate 3270 device 0 (in this example)
- o Device 25 in this example
 - * Is a 3287 printer
 - * With page size of 66 lines
 - * Ldev 6 will be automatically acquired for emulation of device 25 when IML/3000 is started (if available)
- o Note: The use of the word "device" may be confusing at times. Bear in mind the distinction between emulated 3270 devices (i.e. IML emulated 3270 devices) and MPE logical devices (i.e. I/O devices configured under MPE on the HP 3000).

Configuration File: Security

```
      .  
      .  
      .  
ON 0 ALLOW 45, 46, 47, TEST1.NANCY.USERS;  
*  
ON 1 ALLOW @.@.USERS;  
*  
ON # ALLOW MANAGER.PUB.ACCTING
```

Configuration File: Security

EXAMPLE (cont.)

- o ON/ALLOW statement is used to define IML/3000 security
 - * ON phrase specifies 3270 device number(s)
 - * ALLOW phrase specifies MPE ldev's, userid's and program files that are allowed to use these devices

- o First statement
 - * For IML emulated 3270 device 0 ("ON 0")
 - * Allows sessions with \$STDIN on ldev 45, 46, or 47
 - * Allows userid TEST1 logged on to the NANCY group of the USERS account
 - * Allows program file TEST1.NANCY.USERS

- o Second statement
 - * For IML emulated 3270 device 1
 - * Allows any user logged on to any group in the USERS account
 - * Allows any program file in any group in the USERS account

- o Third statement
 - * For all of the devices in this configuration file
 - * Allows userid MANAGER logged on to group PUB of the ACCTING account
 - * Allows program file MANAGER.PUB.ACCTING

Configuration: Minimizing “Device in Use” Problem

- **Definition of problem**
- **Solution: allot device numbers when prepare configuration file**
 - **IDF terminals allotted by MPE Idev or userid**
 - **programs allotted by program name or userid**
programs can keep retrying OPEN3270

Configuration:
Minimizing "Device in Use" Problem

PROBLEM:

- o If IML device already being used
 - * Terminal user will get message "Device Requested is Already in Use" if try to start IDF using that device number
 - * Program will get RESULT code 8 (Device Already in Use) if calls OPEN3270 using that device number
- o In order to start using IML/3000
 - * Terminal user will have to re-issue :IML command with another device number
 - may have to do this many times until free device found
 - * Program will have to call OPEN3270 using different device number
 - may have to do this many times until free device found

SOLUTION:

- o Assign IML device numbers ahead of time, usually when the configuration file is made.
- o You can assign device numbers by userid or by user groups
 - * Example: "Bob, you should always use device number 4"
 - * Example: "The Parts Database group should use devices 5, 6 or 7; the accounting group should use devices 2 and 3."
- o You can assign device numbers by MPE logical device number
 - * Example: "Whoever is using this terminal over here, MPE ldev 25, should use IML device 1".
- o You can assign device numbers to program files in the same way, or code OPEN3270 retry logic into the program
- o Device assignments can be informal, or they can be enforced with configuration file ON/ALLOW list.

Minimizing “Device in Use” Problem: Example

```
      .  
      .  
      .  
DEVICE 0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15;  
TYPE    3278  
BUFFER SIZE 1920  
*  
ON 0 ALLOW 21, @.@.SYS, 23;  
ON 1 ALLOW 22, @.@.SYS, 23;  
ON 2 ALLOW 23, @.@.SYS;  
ON 3 ALLOW CONNIE.@.USER, @.@.SYS, 23;  
ON 4 ALLOW BOB.@.USER, @.@.SYS, 23;  
ON 5,6,7,8,9,10,11,12,13 ALLOW @.@.@;  
ON 14 ALLOW IMSPGM1.@.PROD, @.@.SYS, 23;  
ON 15 ALLOW PARTINVN.@.PROD, @.@.SYS, 23;
```

Minimizing "Device in Use" Problem:

EXAMPLE

- o Configuration file defines 16 IBM 3278 terminals
- o Device allotments are as follows:
 - * All users and programs in SYS account are authorized to emulate all 16 terminals
 - * User of HP ldev 23 also can emulate all 16 terminals
 - * Terminal devices 5 through 13 are available to all users and programs on the system
 - * Terminal device 0 also available to HP device 21
 - * Terminal device 1 also available to HP device 22
 - * Terminal device 3 also available to CONNINE.@.USER
 - * Terminal device 4 also available to BOB.@.USER
 - * Terminal device 14 also available to program IMSPGM1.@.PROD
 - * Terminal device 15 also available to program PARTINVN.@.PROD

IML/3000 Configuration: Exercise

- **Prepare a configuration file for your IML/3000 system. Allot device numbers to minimize “device in use” problem and provide security.**

managing iml/3000

**IML/3000 Configuration:
Exercise**

managing iml/3000

IMLMGR Subsystem

- **Tool for management of IML/3000**
- **Invocation requires “OP” capability**
- **Entered via :IMLMGR session command**
- **Provides set of commands**

managing iml/3000

IMLMGR Subsystem

- o One of two ways provided for management of IML/3000
- o Normally used by IML Manager

IMLMGR Subsystem: Commands

- **Verify**
- **Configuration File**
- **Start**
- **Acquire**
- **Release**
- **Display**
- **Stop**
- **Kill**
- **Trace**
- **Exit**

IMLMGR Subsystem: Commands

- VERIFY configfilename
 - * Checks an IML/3000 configuration file for syntax errors
 - * Also makes the verified file the current configuration file
- CONFIGURATION FILE configfilename
 - * Makes the named configuration file the current configuration file
 - * VERIFY or CONFIGURATION FILE command must be entered before any other IMLMGR command (except DISPLAY and EXIT)
- START
 - * Starts the line (starts the IML Monitor and communication with the INP) specified in the configuration file
 - * If there is more than one IML line in system, they can be distinguished by their configuration files
- ACQUIRE ldev USING device number [ENHANCE=value][;BLANK]
 - * Starts IDF on the specified HP device
 - * Uses IBM device number specified
 - * Command overrides security specifications of configuration file
- RELEASE ldev
 - * Stops IDF on specified HP device
- DISPLAY ldev
DISPLAY ALL
DISPLAY CONFIGURATION FILE
 - * Displays the names of configuration files, emulated 3270 devices, MPE ldevs, and accounting information associated with active IML/3000 pseudo devices
- STOP
 - * Initiates orderly line disconnect and IML/3000 shutdown for current IML/3000 configuration file
- TRACE ON, parms
 - * Turns on CS Trace facility for current IML/3000 line
- KILL
 - * Disconnects line and halts IML/3000 for current configuration file
- EXIT
 - * Terminates IMLMGR subsystem
 - * Does not halt IML/3000

IMLMGR Subsystem: Example of Use

```
: IMLMGR
HP 32229A.000.00 3270 Manager (C) HEWLETT-PACKARD
>DISPLAY ALL
100 Communications line to host not connected
>CONFIGURATION FILE IMLCONF.PUB.SYS
>START
>D A
PSEUDO DEVICE NUMBER: 100
CONFIGURATION FILE: IMLCONF.PUB.SYS
AVAILABLE IBM DEVICES: 0,1
  IBM   LDEV PROGRAM NAME           INTRODUCED
  25    6    IDF--ACQUIRIED        TUES 11:41A
```

IMLMGR Subsystem: Example of Use

In this example, the DISPLAY, CONFIGURATION FILE and START commands are used. When DISPLAY is first used, the only IML/3000 pseudo device configured into MPE (device 100) is inactive. After START is issued, the device becomes active and IDF is started automatically on an HP ldev. This can be seen from the results of the second invocation of DISPLAY ALL (abbreviated to "D A").

managing iml/3000

:IMLCONTROL Command

- **Used by IML/3000 Console Operator**
- **Used to control communications line**
- **Accessed at system console through OPERATOR.SYS**
- **Accessed at user's terminals if user "allowed" to use command**

managing iml/3000

:IMLCONTROL Command

- o The IMLCONTROL command is a console command
- o It can also be used by a user that has been ALLOW'd

:IMLCONTROL Commands

- **:IMLCONTROL START, configfilename**
- **:IMLCONTROL STOP, configfilename**
- **:IMLCONTROL KILL, configfilename**
- **:IMLCONTROL TRACE, configfilename**

:IMLCONTROL Commands

- START
 - * Same function as IMLMGR START command
 - * Starts IML monitor process running, using line and control unit information in specified configuration file
 - * Optionally, can also turn on trace
- STOP
 - * Same function as IMLMGR STOP command
 - * Requests orderly shutdown of IML/3000 subsystem
- KILL
 - * Same function as IMLMGR KILL command
 - * Requests immediate shutdown of IML/3000 subsystem
- TRACE
 - * Same function as IMLMGR TRACE command
 - * Initiates CS Trace facility on communication line defined in specified configuration file

Tracing the Communications Line

- Use CS Trace facility — standard HP 3000 communications line debugging tool

- Turn on Trace facility via
 - : IMLMGR/IMLCONTROL TRACE command
 - or —
 - : IMLCONTROL START command

Tracing the Communications Line

- o Trace facility can be started with the IMLMGR TRACE command, the IMLCONTROL TRACE command, or the TRACE parameter of IMLCONTROL START

managing iml/3000

TRACE Command Parameters

TRACE ON, ALL, mask,, WRAP, tracefile

TRACE Command Parameters

- ALL
 - * Strongly recommend use of this parameter.
If not specified with IML/3000, will result in trace of INP errors only. No line errors will be recorded.
- mask
 - * Specifies type of line events to trace (see IML/3000 Reference Manual)
 - * The default mask is acceptable for most purposes
- WRAP
 - * Causes trace entries that overflow each trace record to overlay prior entries
- tracefile
 - * Name of file where trace records is to be written
 - * Default file name is CSTRACE.PUB.SYS
- The other standard CS Trace parameter, "numentries", has no effect when used with IML/3000.
 - * Numentries is always 14 or 15

Tracing the Communications Line: Example

■ IMLMGR

```
>C IMLCONF.PUB.SYS
>TRACE ON, ALL
.
.
.
>TRACE OFF
```

■ IMLCONTROL console command

```
:IMLCONTROL START, configfile; TRACE ON, ALL
-or-
:IMLCONTROL TRACE configfile, ON
:IMLCONTROL TRACE configfile, OFF
```

managing iml/3000

Tracing the Communications Line: Example

In both these examples, default TRACE parameters are used.

Printing the Trace File

```
:FILE CSTRACE=tracefile  
:RUN CSDUMP.PUB.SYS[,hex]
```

Printing the Trace File

- o The file equation must be used if the name of the trace file is not CSTRACE.PUB.SYS
- o If the "hex" parameter is specified when running CSDUMP, entries in trace file will be printed in hexadecimal

INP RAM Dump

- Triggered by INP failure
- INP memory dumped to: INPLOGnn.PUG.SYS
- IML/3000 permits up to 20 dumps per day
- Console message: INP RAM dump in INPLOGnn
- List INPLOGnn using INPDPAN program
- Report INP failure to HP

INP RAM dump

- o INP "sysfail" will cause dump
- o INPLOGnn
 - * nn = counter from 1 to 99, and is incremented by 1 each time a INP dump occurs

IML Monitor Internal (Stack) Dump

- **Triggered by Monitor failure**
- **Stack dumped to: MONDUxxx.PUB.SYS**
- **Console message: either a CS Irrecoverable Error or subsystem error message**
- **Report stack dump to HP**

IML Monitor Internal (Stack) Dump

o MONDUxxx

* Where xxx is MPE logical device number of IML/3000 pseudo device associated with Monitor

Inquiry and Development Facility (IDF): Introduction

**IDF is an IML/3000 application program. It uses IML/3000
Intrinsics to make**

- **HP terminals appear like 3277/3278 terminals to host**
- **HP printers appear like 328x printers to host**

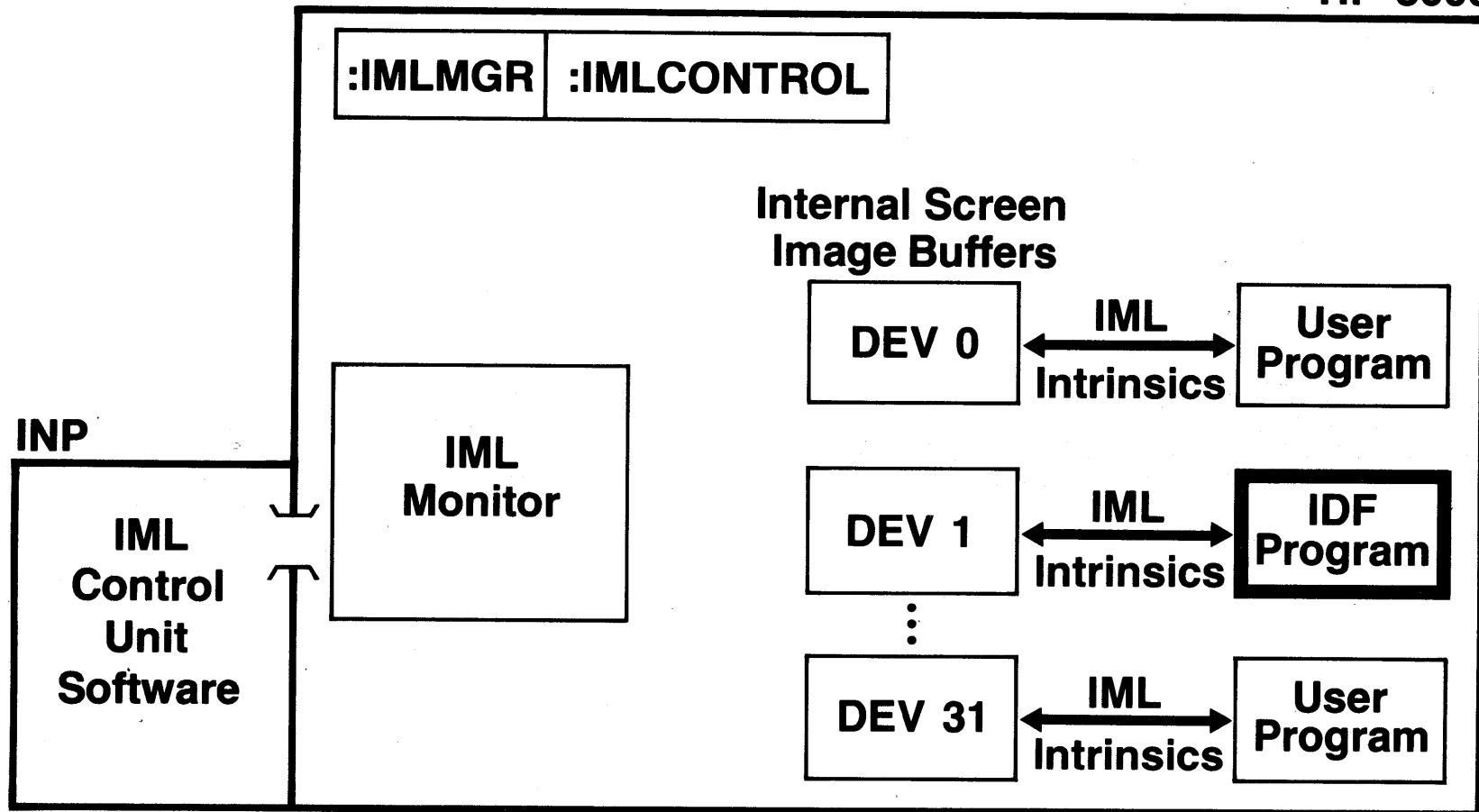
There are differences between IDF and IBM 3270

iml/3000

Inquiry and Development Facility (IDF):
Introduction

A Component of IML/3000 Software

HP 3000



idf: introduction

A Component of IML/3000 Software

- o IDF has same structural relationship to IML/3000 software as any other IML application program
- o IDF calls IML/3000 Intrinsics
- o Each IDF process has its own internal screen image buffer

idf: introduction

HP IDF Terminals and Emulated IBM Devices

IDF allows these HP terminals

HP 2640B/N/S

HP 2641A

HP 2647A

HP 2645A/S/N

HP 2648A

HP 2626A

HP Z624

to appear to host like these terminals:

3277 models 1, 2

3278 models 1, 2

idf: introduction

HP IDF Terminals and Emulated IBM Devices

idf: introduction

HP IDF Printing Devices and Emulated IBM Devices

**IDF allows any HP printer supported by MPE as
“spoolable” (whether spooled or not) to emulate these
printers:**

- **3284**
- **3286**
- **3287 with 3271/72 or
3274/76 adapter**
- **3288**
- **3289**

idf: introduction

HP IDF Printing Devices and Emulated IBM Devices

Starting IDF

Display terminals only:

```
: IML  
CONFIGURATION FILE NAME? configfilename  
DEVICE NUMBER? devicenum
```

Display terminals and printing devices:

IMLMGR >ACQUIRE command

-or-

AUTO ACQUIRE statement in configuration file

Starting IDF

- o Keyboard-display terminals only:
 - * Log on to terminal
 - * Enter :IML command
 - syntax: IML [ENHANCE=1 or 2 or 3][;BLANKS]
 - ENHANCE affects video display (discussed in a subsequent slide)
 - BLANKS affects treatment of blanks/nulls at terminal (discussed in a subsequent slide)
 - * When prompted, enter configuration file name
 - * When prompted, enter host device number of device wish to emulate
- o Keyboard-display terminals and printing devices
 - * IML Manager can issue IMLMGR >ACQUIRE command
 - * IML Manager can include the device in a configuration file AUTO ACQUIRE statement

Stopping IDF

Display terminals only:

- User presses f8 (EXIT) key
- BREAK and subsystem BREAK (control-Y) disabled

Display terminals and printing devices:

- IMLMGR >RELEASE command
- IMLMGR >STOP, >KILL commands
- IMLCONTROL STOP, KILL commands

— IDF exit

} IML exit

Stopping IDF

- o Display terminals
 - * Press f8 special function key on 264x terminal keyboard
 - this is comparable to turning off a 327x device
 - perform soft reset first, if keyboard has been disabled
- o Display terminals and printing devices
 - * IMLMGR >RELEASE command will terminate IDF immediately on printer devices
 - * IMLMGR >RELEASE command will send a message requesting the user exit (press f8) on display terminals
 - * IMLMGR, IMLCONTROL STOP will terminate IDF immediately on printer devices
 - * IMLMGR, IMLCONTROL STOP will send a message requesting the user exit (press f8) on display terminals
 - * IMLMGR, IMLCONTROL KILL will terminate IDF immediately on all displays and printers for that configuration file (It will also terminate IML access for all currently running IML application programs as well.)

Character Display Differences

IBM

HP

EBCDIC			ASCII		
character	octal	hex	character	octal	hex
¢	112	4A	[133	5B
!	132	5A]	135	5D
	117	4F	!	041	21
└	137	5F	^	136	5E

idf: introduction

Character Display Differences

idf: display terminals

Using IDF on HP 264x Display Terminals

- **Comparison of IDF and 3277/3278 terminal features**
- **Comparison of keyboard functions**
- **Additional considerations**
- **LOGIML file and hardcopy of screen image**

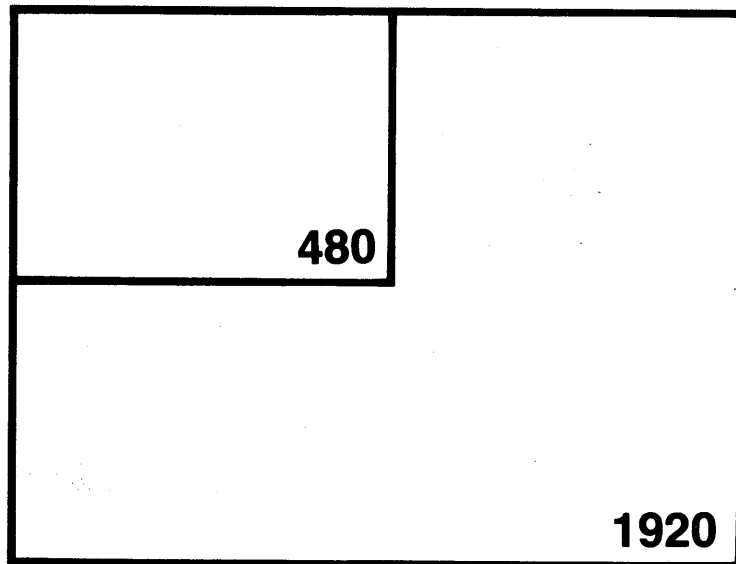
idf

Using IDF on HP 264x Display Terminals

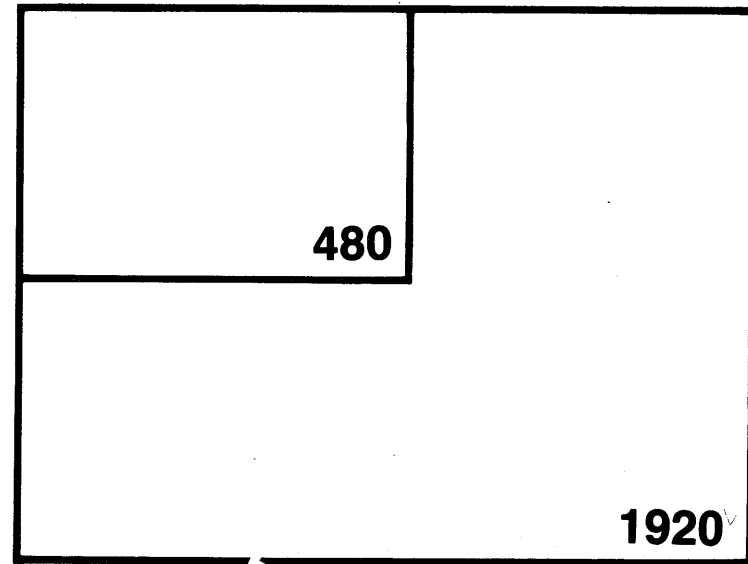
idf: display terminals

Comparison of 3277/3278 and IDF Terminal Features

**3277/3278 Terminals:
Screen sizes**



**HP-IDF Terminals:
Screen sizes**



Also: 960, 2560, 3440, 3654

idf: display terminals

Comparison of 3277/3278 and IDF Terminal Features

3270 Terminals

IDF Terminals

Screen sizes: 480 (12 x 40)
 1920 (24 x 80)
960, 2560, 3440, 3654

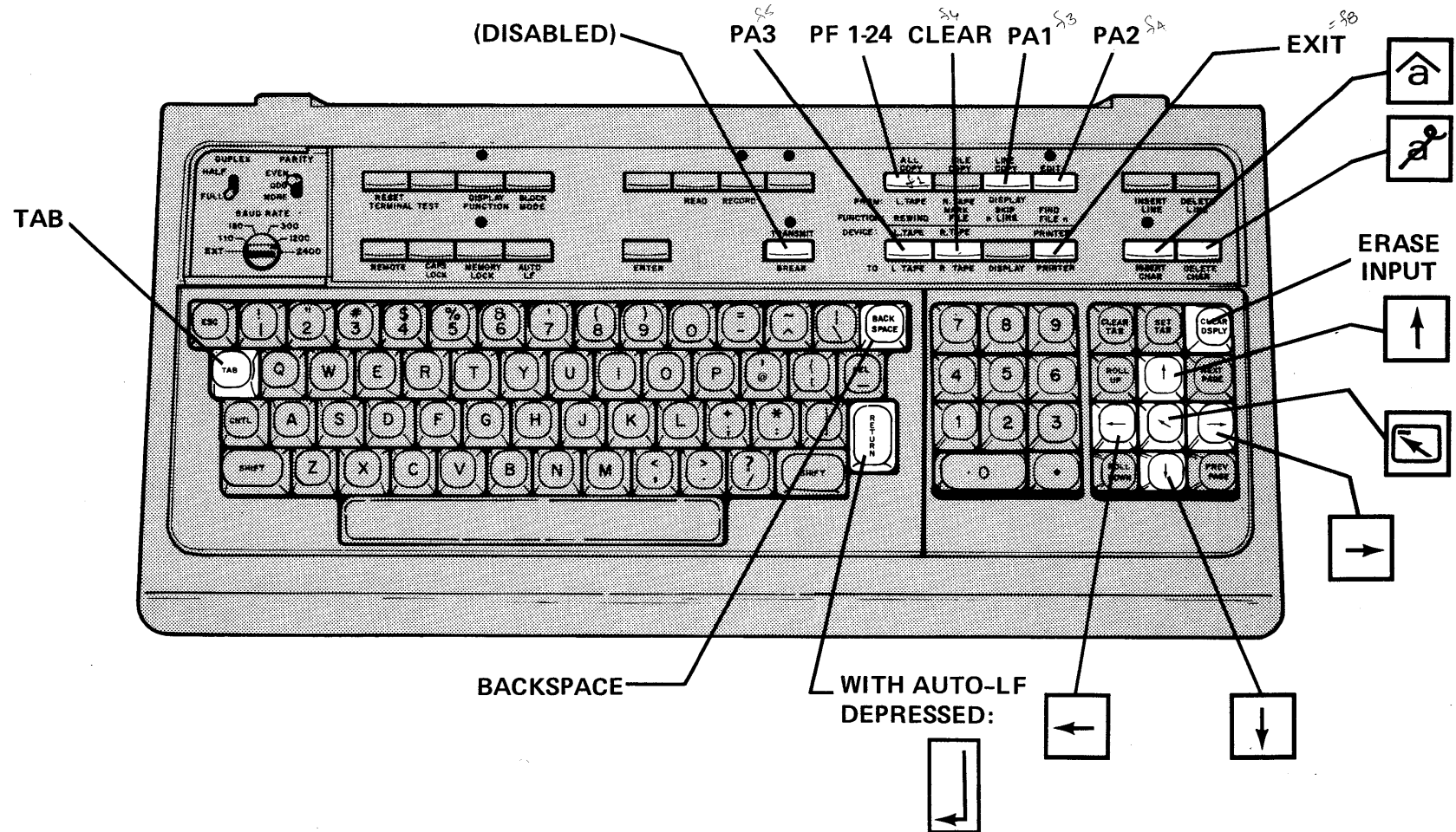
Screen sizes: 480 (12 x 40)
 1920 (24 x 80)
960, 2560, 3440, 3654--
 not available

Formatted screens:
protected/unprotected fields
intensified fields
numeric field edit
automatic skip

Formatted screens:
protected/unprotected fields
enhanced fields
not available
automatic skip

idf: display terminals

Comparison of Keyboard Functions: HP-IDF



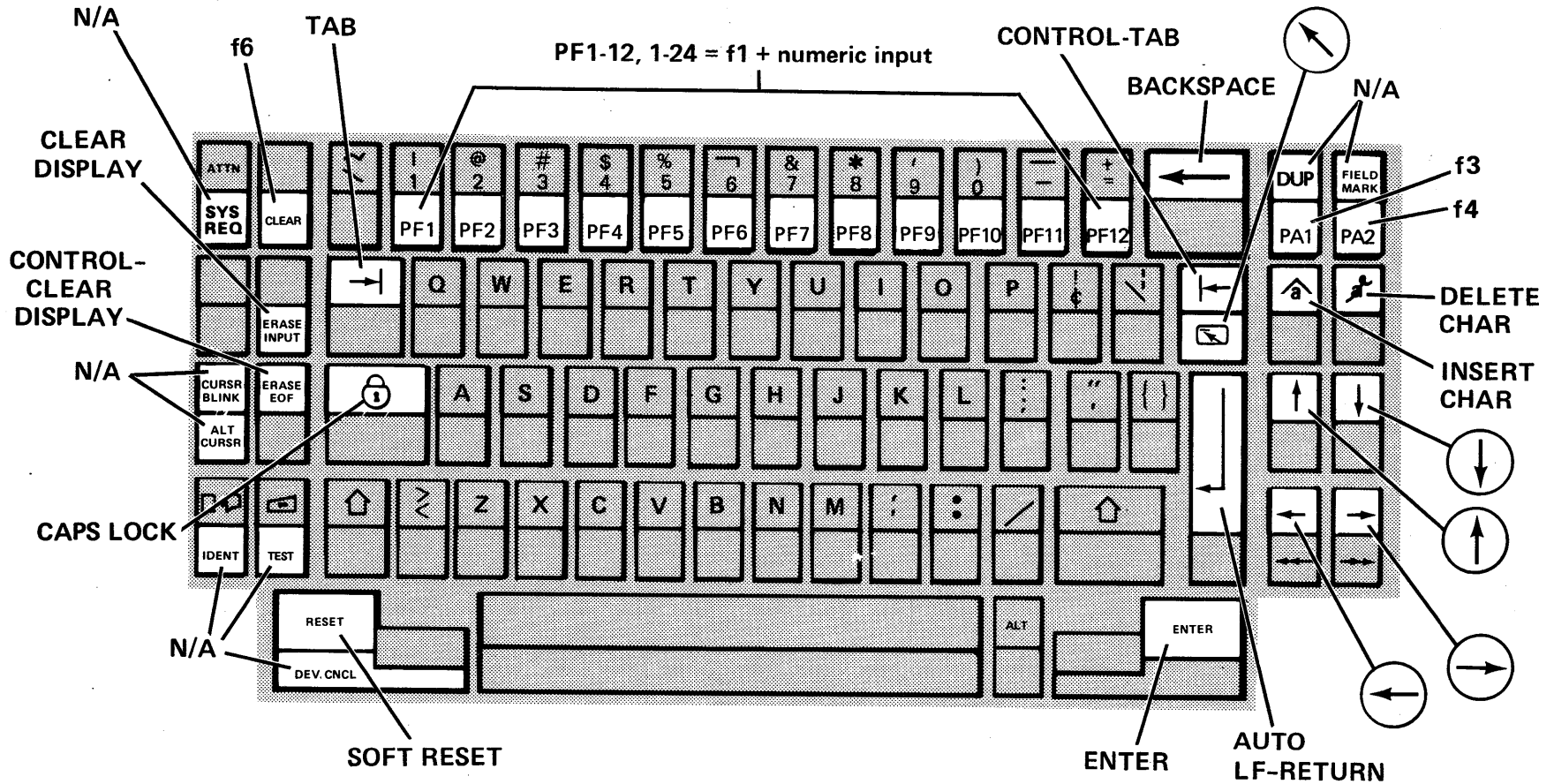
idf: display terminals

Comparison of Keyboard Functions: HP-IDF

- o Break key, control-Y: disabled
- o Control key: use limited
- o Return key: local only
- o Enter key: causes transmission to host
- o Reset key: special meaning with IDF (use with caution)
 - * Discussed further in subsequent slide
- o Special function keys: special definitions with IDF
 - * f1: Program Function (PF) keys
 - * f2: (not used)
 - * f3: Program Attention (PA1) key
 - * f4: PA2
 - * f5: PA3
 - * f6: CLEAR
 - * f7: (not used)
 - * f8: EXIT (equivalent to powering down 327x terminal)
- o These keys work as described in HP terminals users' manuals:
 - * CAPS LOCK, INSERT CHAR, HOME UP, cursor positioning keys

idf: display terminals

Comparison of Keyboard Functions: IBM 3278



idf: display terminals

Comparison of Keyboard Functions: IBM 3278

- o The following key function have been reproduced on the HP-IDF terminal:
 - * Erase input
 - * Erase EOF
 - * Tab
 - * Backtab
 - * Backspace
 - * Newline
 - * Insert character
 - * Enter
 - * PF1-24
 - * Clear
 - * Exit
- o The following functions are not realized on the HP-IDF:
 - * Clicking key
 - * Field mark
 - * Ident
 - * Dev cncl
 - * Test/system request
 - * Dup
- o PF keys realized via 264x "soft" keys

idf: display terminals

Transmitting Data

- **Block mode — no data sent until an I/O key pressed**
- **ENTER — press ENTER key**
- **PA1, PA2, PA3, CLEAR — press appropriate soft key**
- **PF1-PF24 — press f1, indicate which PF key, press ENTER**

idf: display terminals

Transmitting Data

- o All transmissions are in block mode
 - * Press ENTER key to transmit
- o PA keys and CLEAR key:
 - * PA1: f3
 - * PA2: f4
 - * PA3: f5
 - * CLEAR: f6
- o PF keys:
 - * Press f1
 - * Enter program function key number when prompted
 - * Press ENTER key

idf: display terminals

Keyboard Lock and Reset

- **IDF locks keyboard to emulate 3270 “input inhibit”**
- **Exceptions: HP 2640B/N/S not locked because no soft reset**
- **To reset terminals other than 2640, press RESET key once**

idf: display terminals

Keyboard Lock and Reset

idf: display terminals

Recovery from Accidental Hard Reset

- **Hard reset occurs when press RESET key twice (once for 2640's)**
- **No guarantee of recovery from hard reset**
- **For ATC/ADCC terminals: press control-R, RETURN control-R**
- **For multipoint terminals: press ENTER after get "TERMINAL READY" message**

idf: display terminals

Recovery from Accidental Hard Reset

idf: display terminals

Display Enhancement Options for IDF Terminals

:IML; ENHANCE=value

value	converts 3270 normal bright to:	converts 3270 high intensity to:
0	264x half bright	264x normal
1	264x normal	264x underline
2	264x normal	264x inverse video
3	264x inverse video	264x normal

idf: display terminals

Display Enhancement Options for IDF Terminals

- o :IML command "ENHANCE" parameter
 - * Used to alter video display
 - * Enter 0, 1, 2, or 3
 - * Default=0

idf: display terminals

Additional Functional Differences

- **Nulls vs. blanks**
- **Cursor location and MTS/3000 terminals**

idf: display terminals

Additional Functional Differences

- o Nulls vs. blanks in unprotected fields
 - * 264x terminal does not accept null characters on a formatted screen
 - IDF transmits blanks instead of nulls to 264x terminal
 - * When IDF gets data back from 264x terminal, it does not know which blank characters in the unprotected fields are supposed to be null characters. It makes the following assumptions:
 - Leading blanks converted to nulls, unless BLANKS parameter specified, in which case not converted
 - Embedded blanks always transmitted as blanks
 - Trailing blanks up to end of field always converted to nulls
- o Cursor location
 - * HP 264x terminals connected via ATC/ADCC always transmit correct cursor location when transmit key pressed
 - * HP 264x terminals connected via MTS always return cursor location of row 0 column 0 (buffer position 0)
 - * Therefore you should not use an MTS 264x terminal under IDF if the host application program checks the cursor location when ENTER is pressed
 - * Most host application programs ignore the cursor location entirely, so in most cases this is not a problem

idf: display terminals

ATC/ADCC Terminals and TBUFFs

idf: display terminals

ATC/ADCC Terminals and TBUFFs

- o Configure enough HP 3000 system Terminal Buffers for ATC/ADCC IDF terminals. The maximum that may be needed are:
 - * Connected via ATC: approximately 64 TBUFFs for each terminal running IDF concurrently with other IDF terminals
 - * Connected via ADCC: approximately 32 TBUFFs for each terminal running IDF concurrently
 - * These are maximums; in most cases fewer will be needed for succesful IDF operation
- o MTS terminals do not use Terminal Buffers

idf: display terminals

LOGIML File Screen Trace Facility

```
:FILE LOGIML;DEV=LP  
:IML
```

idf: display terminals

LOGIML Screen Trace Facility

- o Screen activity for each IDF device can be traced
 - * By the IDF user
 - * User enters file equation before entering :IML command
- o File equation directs copy of screens to printer

idf: display terminals

LOGIML Sample Output

```

A
*01234567890123456789012345678901234567890123456789012345678901234567890123456789* Called from location # 1
0#-----SPF/TSO PRIMARY OPTION MENU-----0 50
1#SELECT OPTION ==>#X#####HEWLETT-PACKARD BAEDP#
2#
3# 0 #SPF PARMS - SPECIFY TERMINAL AND SPF PARAMETERS #USERID -#s47SAB 2 50 11 50 40
4# 1 #BROWSE - DISPLAY SOURCE DATA OR OUTPUT LISTINGS #TIME -#16:29 3 50 40 40 40
5# 2 #EDIT - CREATE OR CHANGE SOURCE DATA #TERMINAL -#3278 4 50 40 40 40
6# 3 #UTILITY - PERFORM SPF UTILITY FUNCTIONS #PF KEYS -#24 5 50 40 40 40
7# 4 #FOREGROUND - COMPILE, ASSEMBLE, LINK EDIT, OR DEBUG #RELEASE - 2.2.1A 6 50 40 40
8# 5 #BACKGROUND - COMPILE, ASSEMBLE, OR LINK EDIT #REL DATE - 5/21/80 7 50 40 40
9# 6 #TSO - ENTER TSO COMMAND OR CLIST 8 50 40
0# 7 #TUTORIAL - DISPLAY INFORMATION ABOUT SPF 9 50 40
1# X #EXIT - TERMINATE SPF USING LIST/LOG DEFAULTS 0 50 40
2# 1 50 40
3#PRESS#END KEY#TO TERMINATE SPF# 2 50
4# 3 40 50 40 40
5# 4 50
6# 5 40
7# 6
8# 7
9# 8
0# 9
1# 0
2# 1
3# 2
4# 3
*0123456789012345678901234567890123456789012345678901234567890123456789*
B
C
D
E
Cursor: R1 ,C20 Keyboard Enabled

```

idf: display terminals

LOGIML Sample Output

- o This illustration shows a TSO/SPF menu with user's response
- o A: Row and column numbering begins with zero
- o B: Attributes bytes are represented as H's overprinted with T's
- o C: User input is "X". Cursor appears as line under character
- o D: These four columns show octal values of rightmost six bits of each attribute byte
 - * Note that only one, 11, shows the MDT has been set
- o E: These figures show cursor location and keyboard state at time screen transmitted to host
 - * Cursor is at location row 1, column 20
 - * Keyboard enabled

idf: printing devices

What Kind of Device Can Be an IDF Printer?

- **Any HP printer supported by MPE as “spoolable” (whether spooled or not).**

What Kind of Device can be an IDF Printer?

- o Examples of printers supported by IDF on series II/III:
 - * 2608A (parallel diferential interface)
 - * 2613A
 - * 2617A
 - * 2619A
 - * 2631B (ATC connection, when configured as a printer)
- o Examples of printers supported by IDF on series 30/33/44:
 - * 2631 (HP-IB connection)
 - * 2608A
 - * 2631B (ADCC connection, when configured as a printer)
- o Exception:
 - * 2607 lineprinter not supported; it does not have some of the features necessary for proper emulation of 328x devices

idf: printing devices

Device Capabilities

- **40, 64, 80, 132 column platen widths**
- **3278 local copy key function not supported
(Use slaved printer attached directly to 2645 terminal)**
- **3287 PA1/PA2 key option not supported**
- **No emulation of IBM extended character set**

idf: printing devices

Device Capabilities

idf: printing devices

Spooled or Unspooled?

- **If system printer acquired for IDF is spooled, no output until IML communications line closed or printer released (>RELEASE)**
- **If printer acquired unspooled, output listed “hot” and no other HP 3000 program can FOPEN printer**

idf: printing devices

Spooled or Unspooled?

idf: error messages

Error Messages

- **How displayed**
- **What to do next**
- **Some examples follow**

idf: printing devices

Error Messages

- o How Displayed
 - * Displayed at user's terminal or system console
 - * Warnings displayed at HP device
 - * Fatal errors
 - Reported at system console if IDF started by IML Manager or via AUTO ACQUIRE statement
 - Otherwise reported at user's device
- o What to do: Consult IML/3000 Reference Manual for recovery procedure

Error Messages: Examples

■ Received before emulation begins:

- INVALID CONFIGURATION FILE
- EMULATION NOT AUTHORIZED
- DEVICE REQUESTED IS ALREADY IN USE

■ Received after emulation starts:

- INVALID PROGRAM FUNCTION KEY. PLEASE RE-ENTER

Error Messages: Examples

- IDFERR 3: INVALID CONFIGURATION FILE
 - * Can be received by IML Manager or user
 - * IML configuration file contains errors in syntax
 - * File has to be corrected before it can be used
- IDFERR 6: EMULATION NOT AUTHORIZED
 - * Can be received by Manager or user
 - * You are not authorized to run IDF on this device
 - * Check with IML Manager or check contents of configuration file
- IDFERR 8: DEVICE REQUESTED IS ALREADY IN USE
 - * Can be received by IML Manager or user
 - * The requested IML device is already in use
 - * Request another, or wait until this one free
- IDFWARN 15: INVALID PROGRAM FUNCTION KEY NUMBER
 - * Can be received by user
 - * Number entered was <1, or >12 (for 3277) or >24 (for 3278)
 - * Enter a valid number

iml/3000

Programming IML/3000

- **Programming overview**
- **Sample program**
- **Additional considerations**

Programming IML/3000

Programming Overview

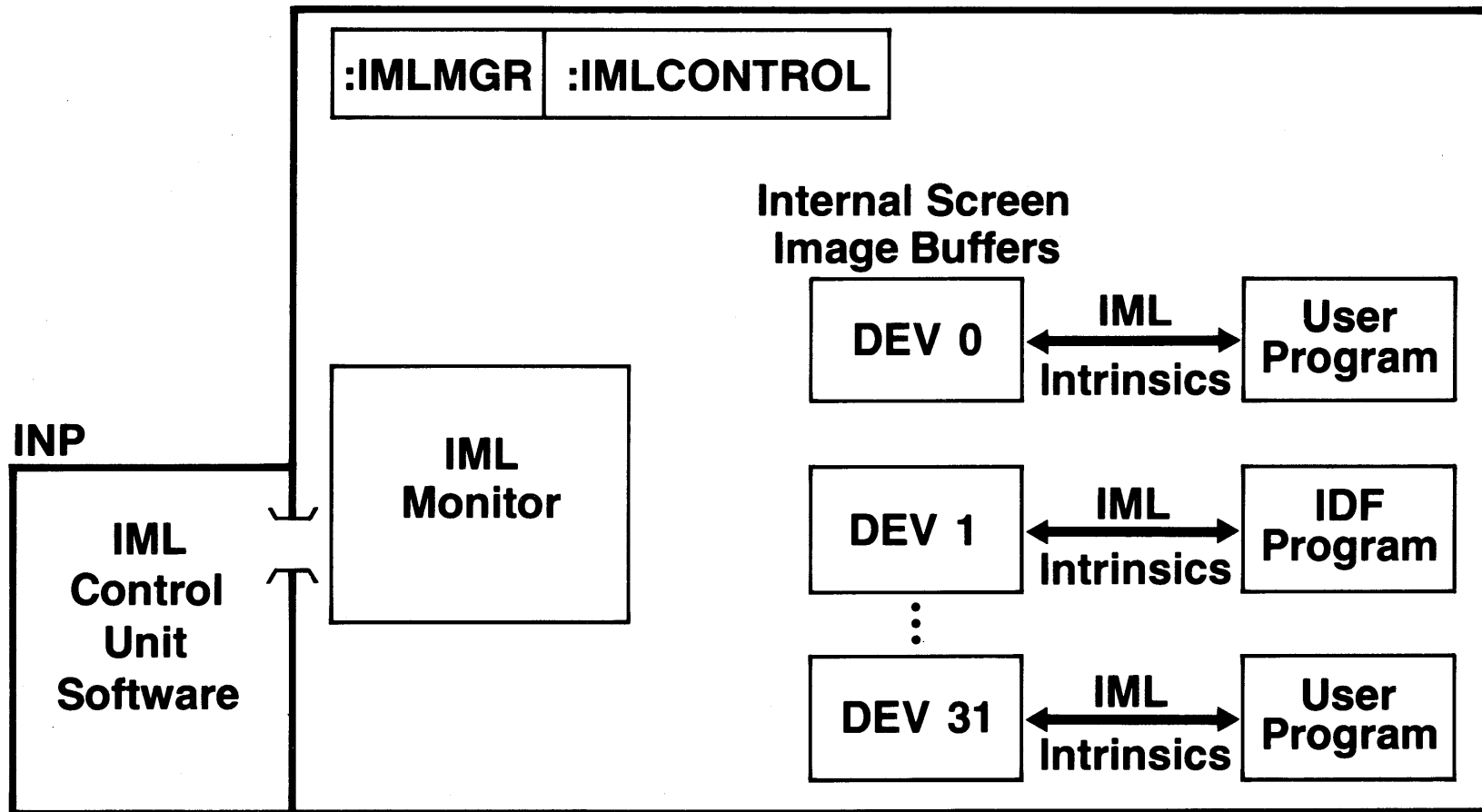
- Internal screen image review
- IML intrinsics
- Timeout parameters
- Result code checking
- Debugging with a “dumpscreen” procedure

Programming Overview

programming overview

Review

HP 3000



Review

- o The buffer containing the internal screen image is an MPE extra data segment. The internal screen image is a character by character representation of a 3270 screen image.
- o The internal screen image buffer (extra data segment) also contains control information.
- o There can be as many as 32 internal screen image buffers per IML line, one for each possible emulated 3270 terminal device.
- o IML application programs access the internal screen image via the IML intrinsics.

programming overview

IML/3000 Intrinsic

IML/3000 Intrinsic

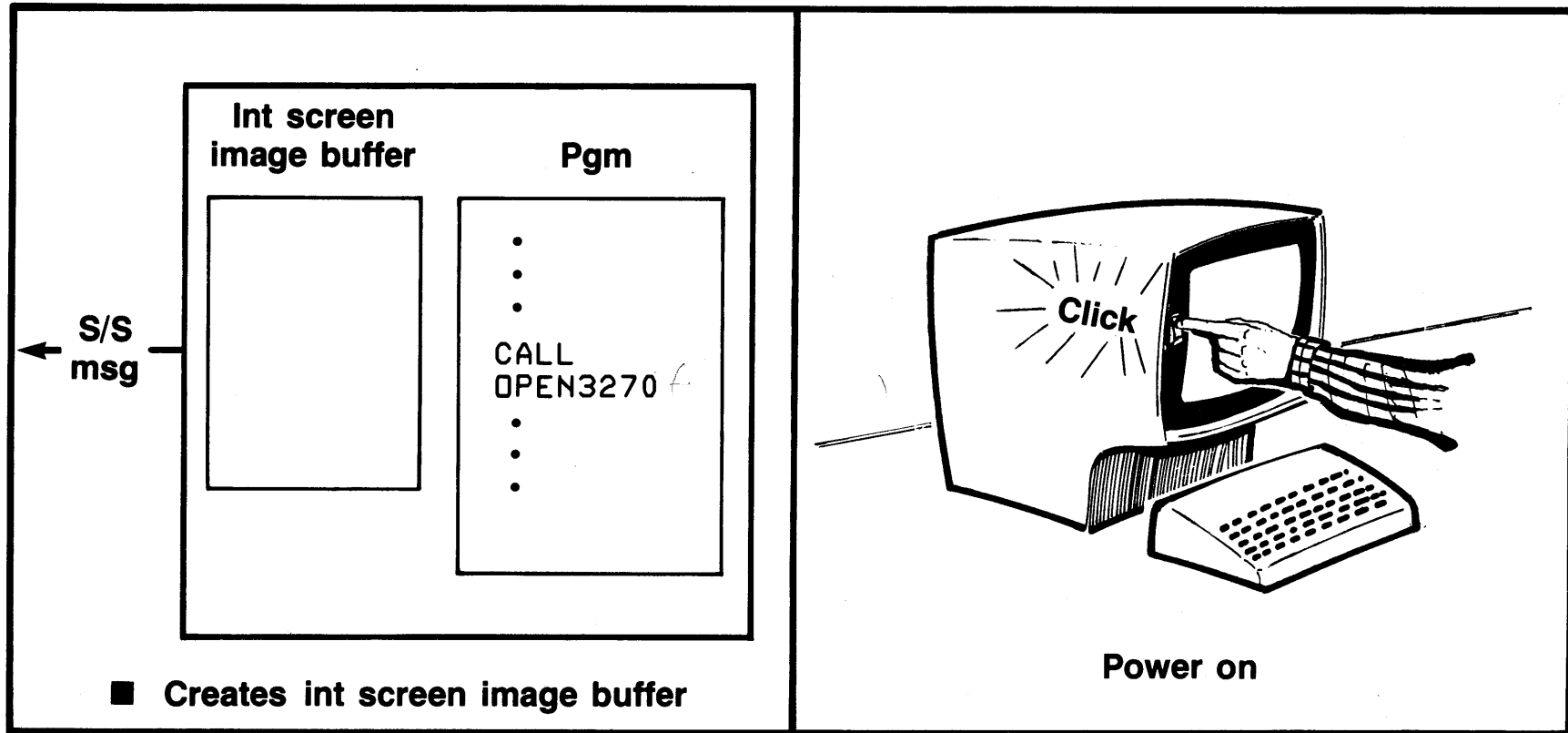
- o The IML intrinsics are procedures directly callable from BASIC, COBOL, COBOL II, FORTRAN, and SPL programs.
- o The IML intrinsics provide access to the IML internal screen image.
- o There are 17 IML intrinsics. These intrinsics allow your program to:
 - * initiate and terminate IML communication
 - OPEN3270 CLOSE3270
 - * read data from and write data into the internal screen image buffer.
 - READFIELD READSCREEN ATTRLIST WRITEFIELD
 - STREAM3270 SCREENATTR FIELDATTR RESET3270
 - * transmit data to the host system, and wait for data to be sent back
 - TRAN3270 RECV3270 IOWAIT IODONTWAIT
 - ABORT3270
 - * invoke useful programming assistance routines
 - VERS3270 ERR3270

programming overview

OPEN3270

IML/3000

3270



OPEN3270

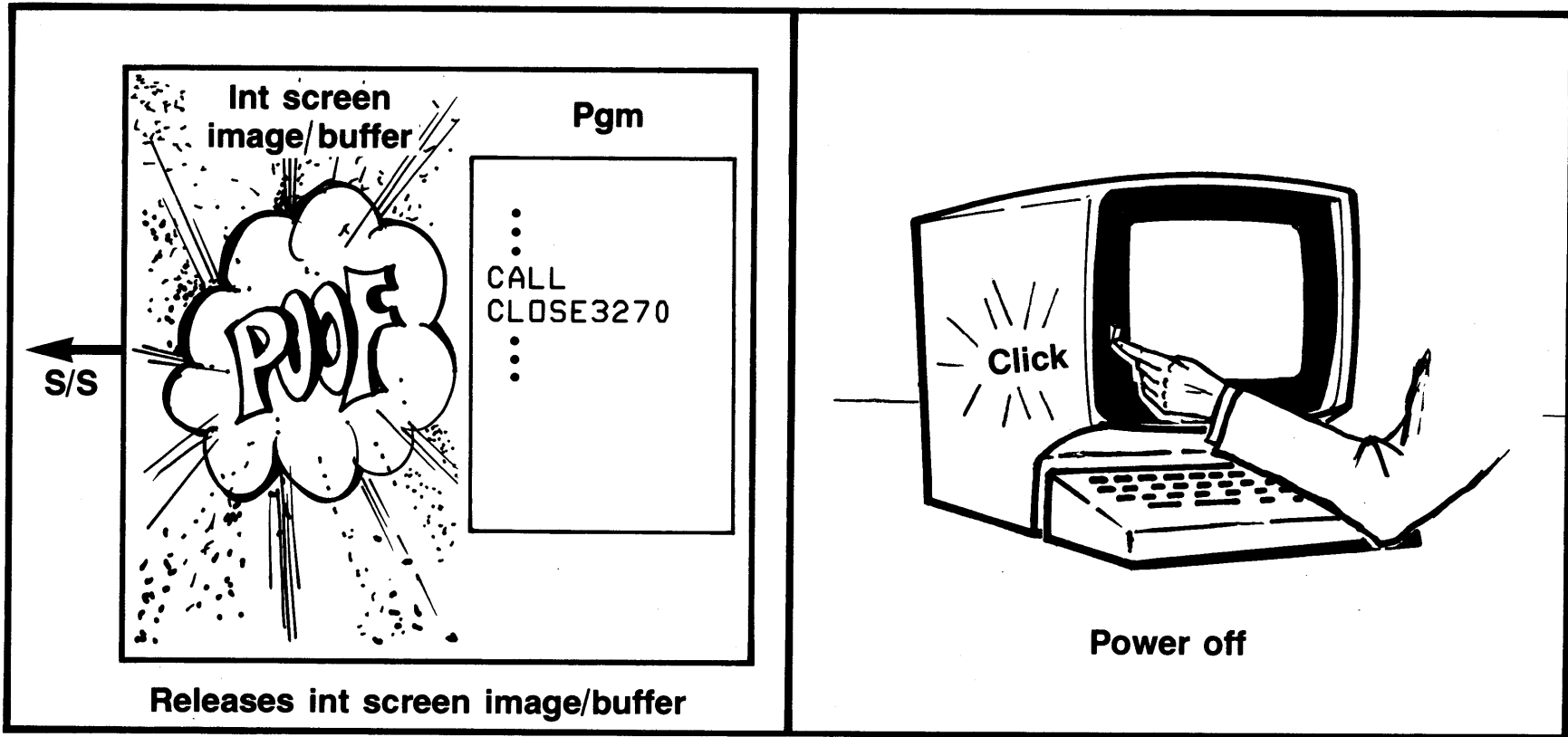
- o initiates IML activity for a program on the 3000
- o is equivalent to powering on a 3270 terminal
- o will only be successful if the IML subsystem has been started (the "control unit" must be "powered on")
- o is specific and exclusive for one of the 32 possible devices. Only one program at a time (including IDF) may have a particular device open. A single program may call OPEN3270 more than once (it may have several devices open concurrently).
- o sends a status/sense message of DEVICE END to the host system, just like a real 3270.
- o does not guarantee that the host system is active nor does it provide an automatic "logon" to the host system.
- o creates an internal screen image buffer for a specified device number.

programming overview

CLOSE3270

IML/3000

3270



CLOSE3270

- o terminates IML activity for a program on the 3000.
- o is equivalent to powering off a 3270 terminal.
- o frees the device number and makes it available for use by others.
- o releases the internal screen image buffer (extra data segment).

- o sends a status/sense message of INTERVENTION REQUIRED to the host on a specific poll, just like a real 3270.
- o does not automatically log you off the host system. Your program should do this before you call CLOSE3270.

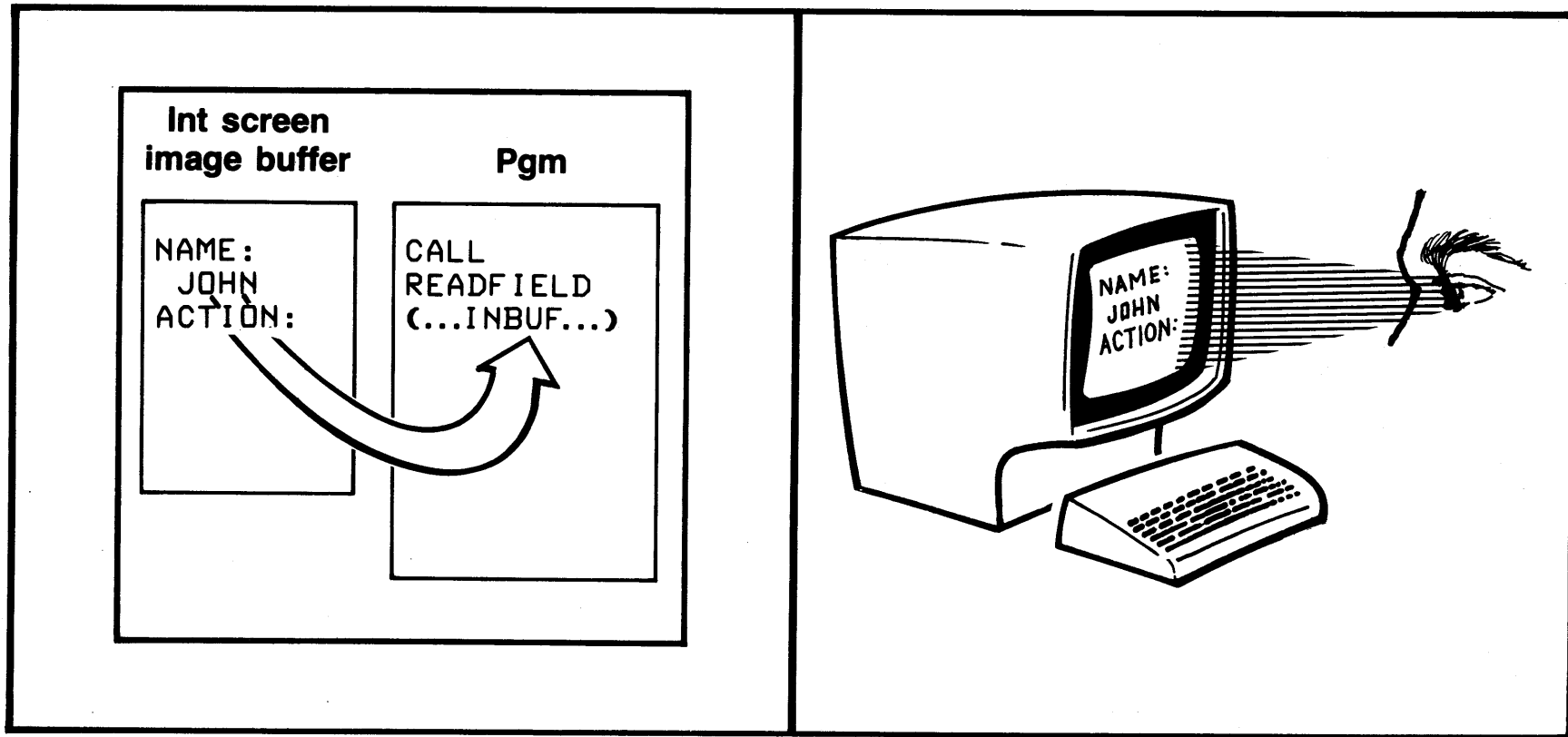
If your program ends without calling CLOSE3270 (for example if your program aborts), MPE will close your IML devices for you.

programming overview

READFIELD, READSCREEN ATTRLIST, FIELDATTR

IML/3000

3270



READFIELD READSCREEN
ATTRLIST FIELDATTR

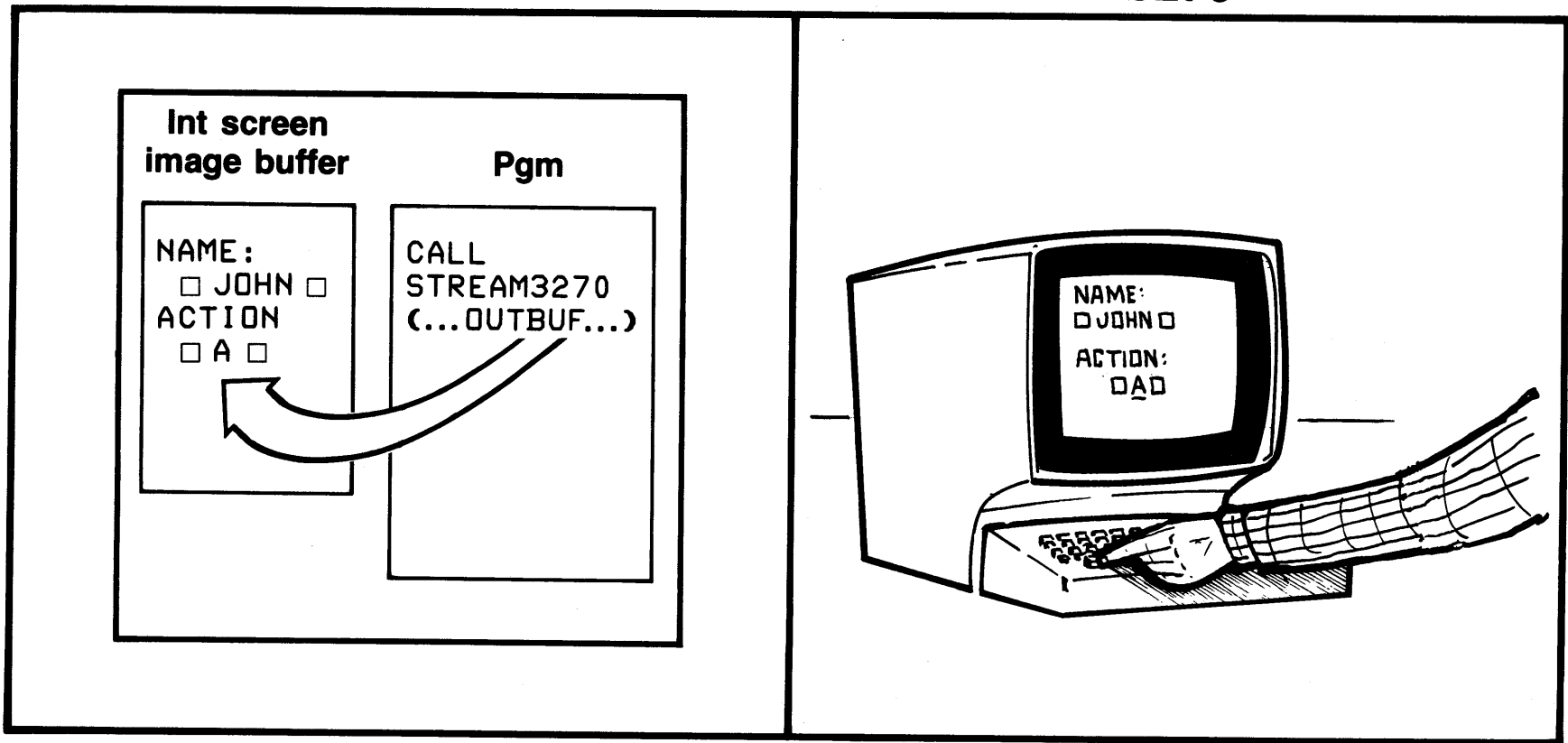
These intrinsics allow your program to read data from the internal screen image, analogously to the way a terminal operator reads data from a 3270 display screen.

- o READFIELD reads the data from one field at a time.
- o READSCREEN reads data from any portion of the screen at a time.
- o FIELDATTR gives you the attributes of a particular field: protected, unprotected, high intensity, non-display, etc.
- o ATTRLIST returns attribute byte locations. It is usually used with READSCREEN.

Note that no I/O to the host system takes place when these intrinsics are called. Data is moved from one portion of 3000 main memory - the internal screen image buffer - to another your program's data area (stack).

programming overview

STREAM3270 WRITEFIELD RESET3270
IML/3000 **3270**



STREAM3270 WRITEFIELD RESET3270

These intrinsics are equivalent to typing at a 3270 terminal.

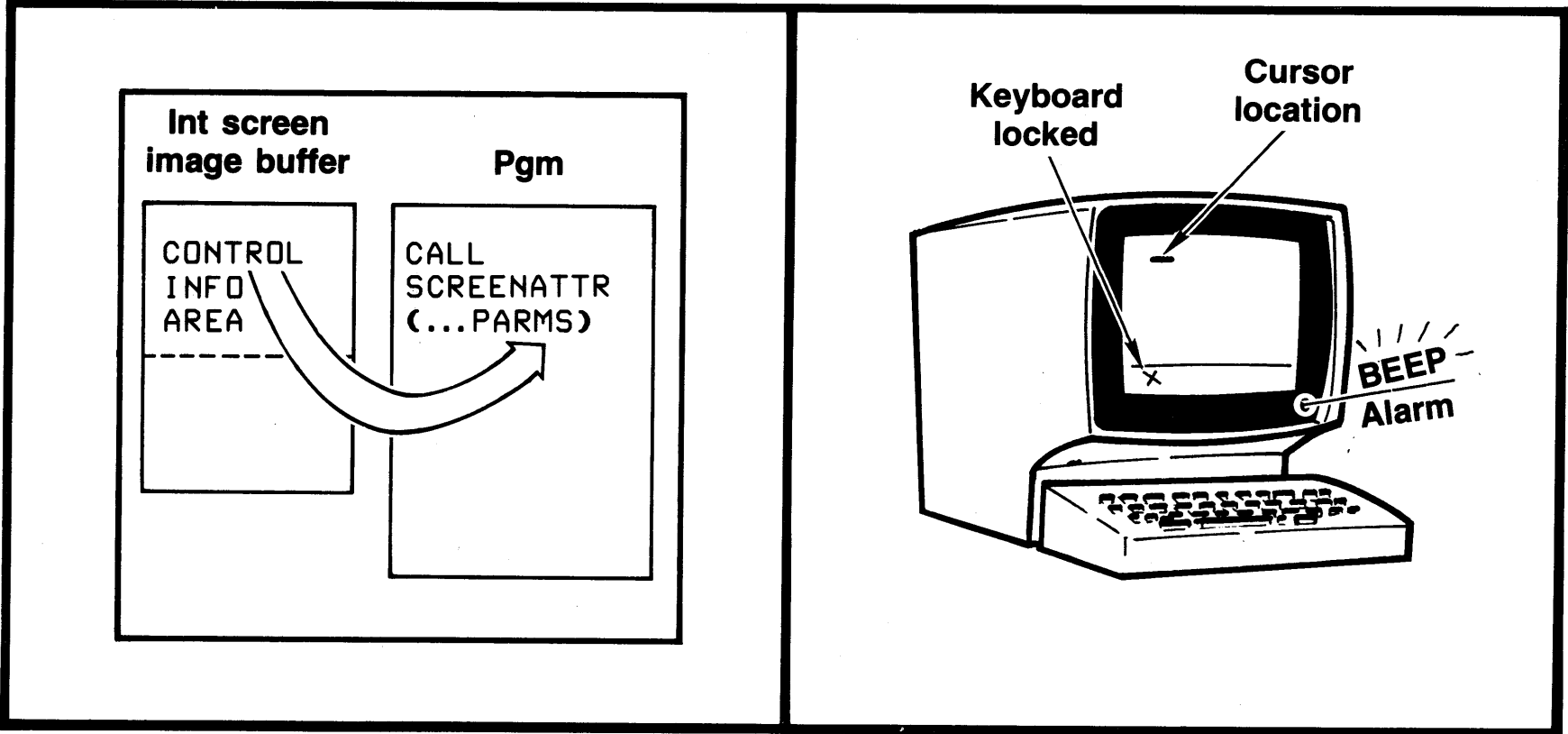
- o WRITEFIELD writes into a specific field.
- o STREAM3270 simulates typing a series of keystrokes on a 3270 keyboard. *manipulate cursor / move stream of char around*
updates buffer in VP 2000
- o RESET3270 is equivalent to pressing the RESET key. *or reset key once*
← unlocks keyboard
- o You cannot write into your internal screen image when your internal screen image's "keyboard" is locked. You can unlock the keyboard only with RESET3270 or the reset keystroke of STREAM3270.
- o Just like on a real 3270, you cannot enter data into a protected field in the internal screen image, nor can you write data over an attribute byte.
- o The fields that you enter data into are marked internally as "modified" (the MDT bit is turned on for that field) just as on a real 3270.
- o Note that no transmission to or from the host system occurs as a result of calling STREAM3270, WRITEFIELD or RESET3270.

programming overview

SCREENATTR

IML/3000

3270



SCREENATTR

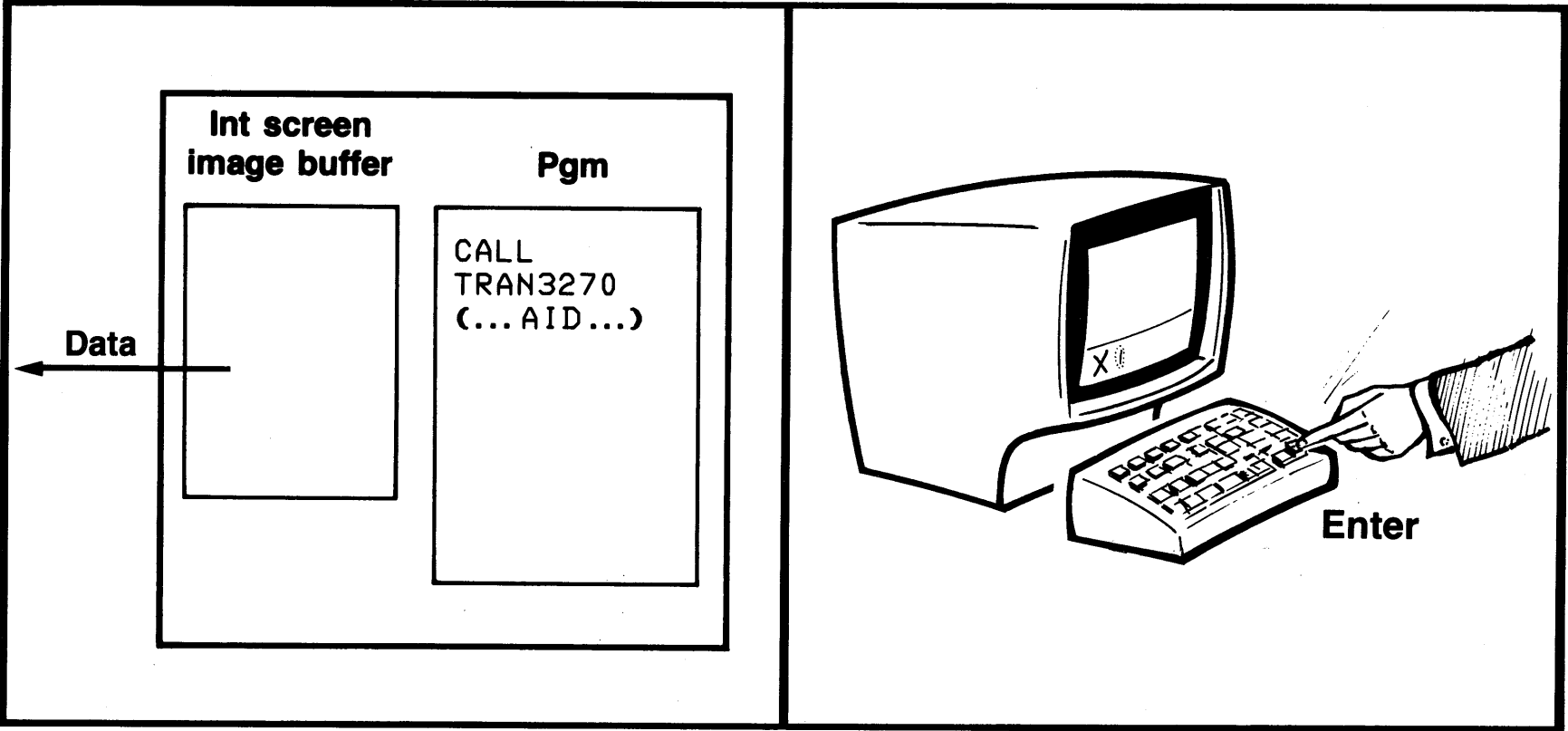
- o gives you information about the current internal screen image, and about the last write the host made to you.
- o it will tell you:
 - * where the cursor is
 - * how many fields are in the screen
 - * whether or not the (logical) keyboard is locked
 - * whether the host sounded the alarm
 - * whether the host set the start print bit
 - * what the printout format isThese last two are significant only for printer device emulation.

programming overview

TRAN3270

IML/3000

3270



TRAN3270

- o is like pressing ENTER (or some other AID such as PA1 OR PF4) at a 3270 terminal.
- o causes data to be sent to the host system the next time this device is polled.
- o locks the internal screen image's "keyboard" ("input inhibited" condition).
- o under normal wait I/O, does not return control until the data is transmitted.

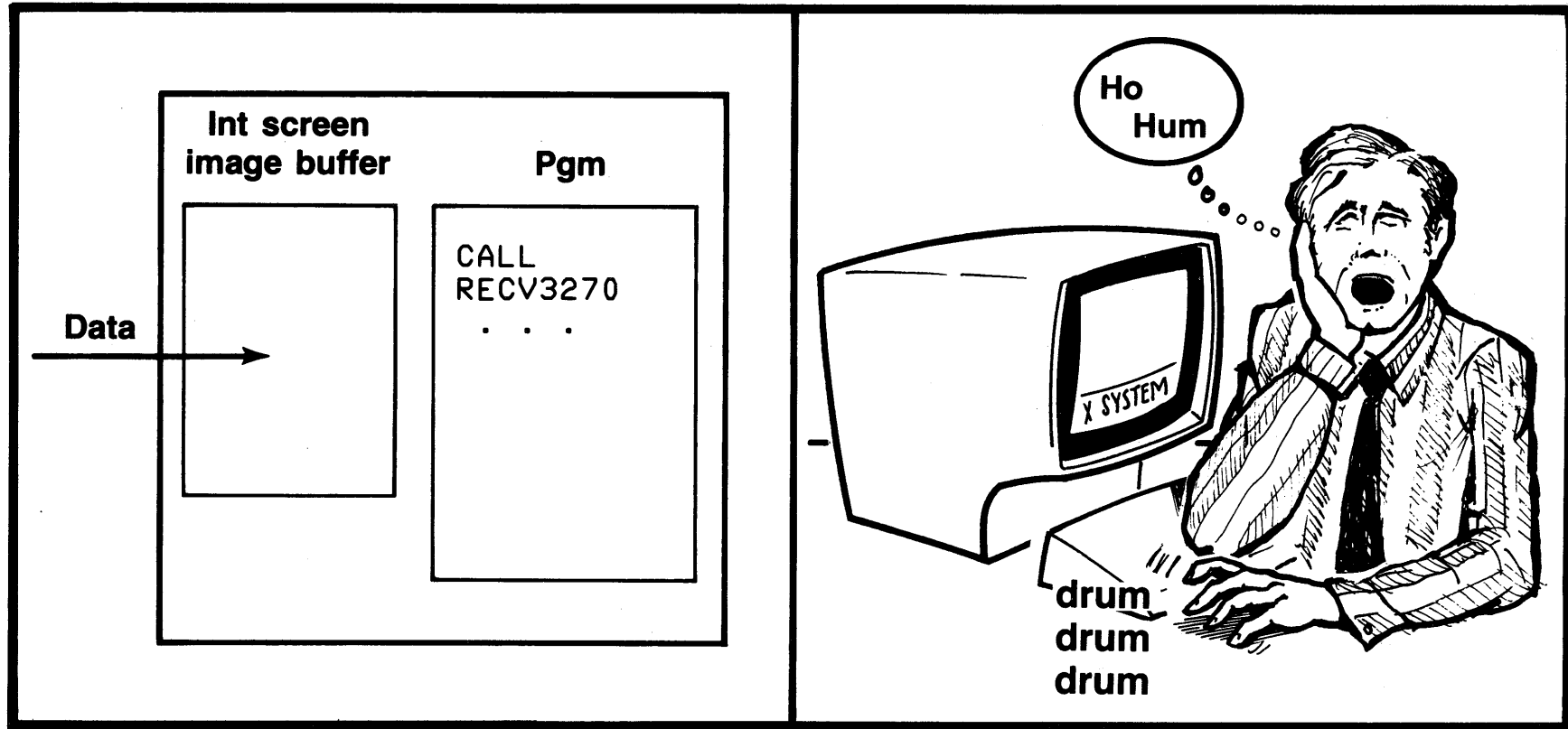
Under no-wait I/O, one of the intrinsics IOWAIT, IODONTWAIT, or ABORT3270 must be called to insure I/O completion.

programming overview

RECV3270

IML/3000

3270

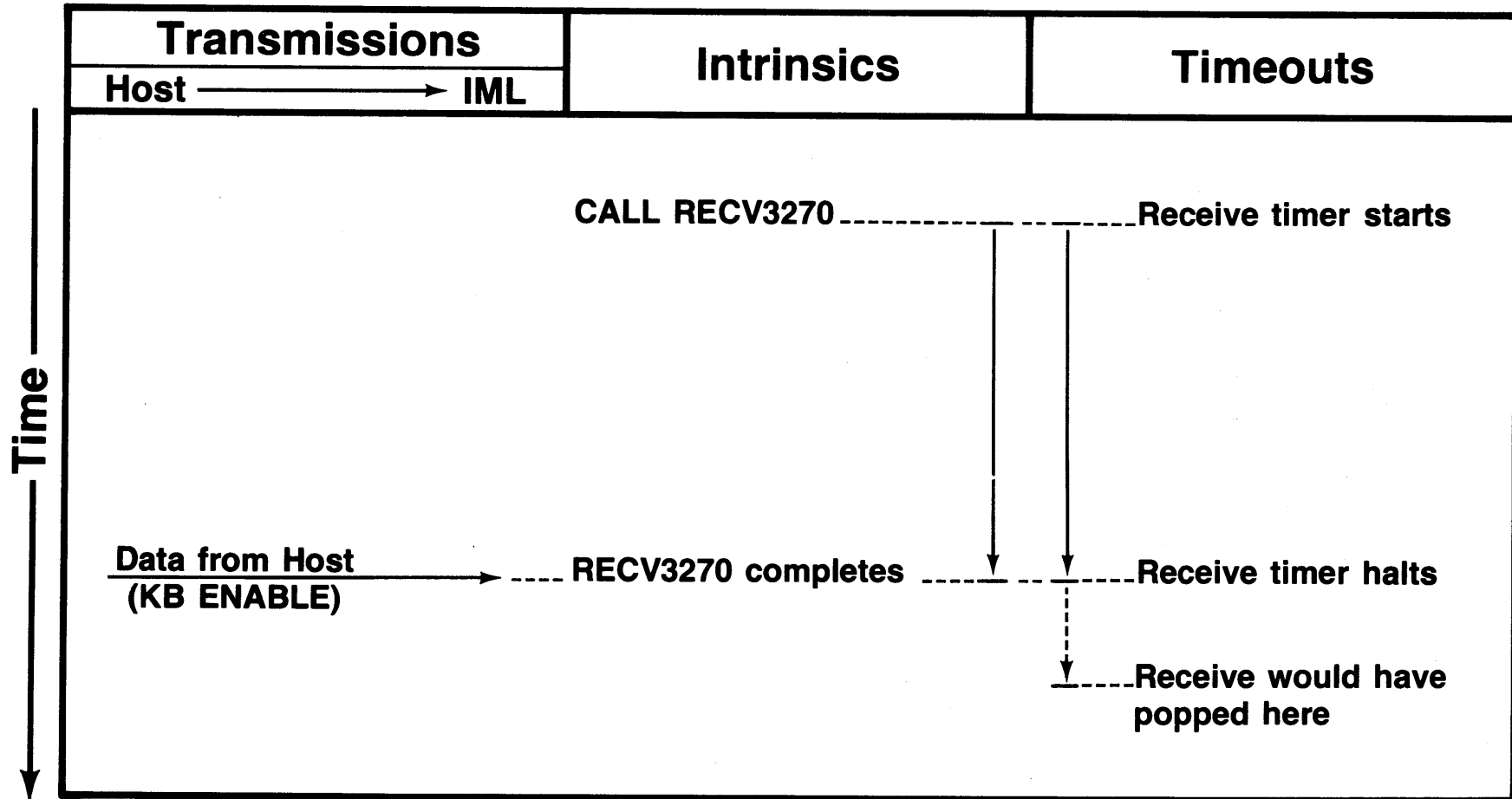


RECV3270

- o is like a terminal operator sitting in front of a 3270 terminal waiting for a data transmission from the host (similar to "input inhibited" and "system" conditions).
- o normally, will complete when data has been returned from the host system.
- o optionally, can be made to complete when the host system unlocks the keyboard.
- o uses the TIMEOUT parameter in OPEN3270 to set two timers. These timers, if used, insure that your program does not hang indefinitely if the host system does not send you any data.

Under no-wait I/O, one of the intrinsics IOWAIT, IODONTWAIT, or ABORT3270, must be called to insure I/O completion.

Using Timeout Parameters — Case 1



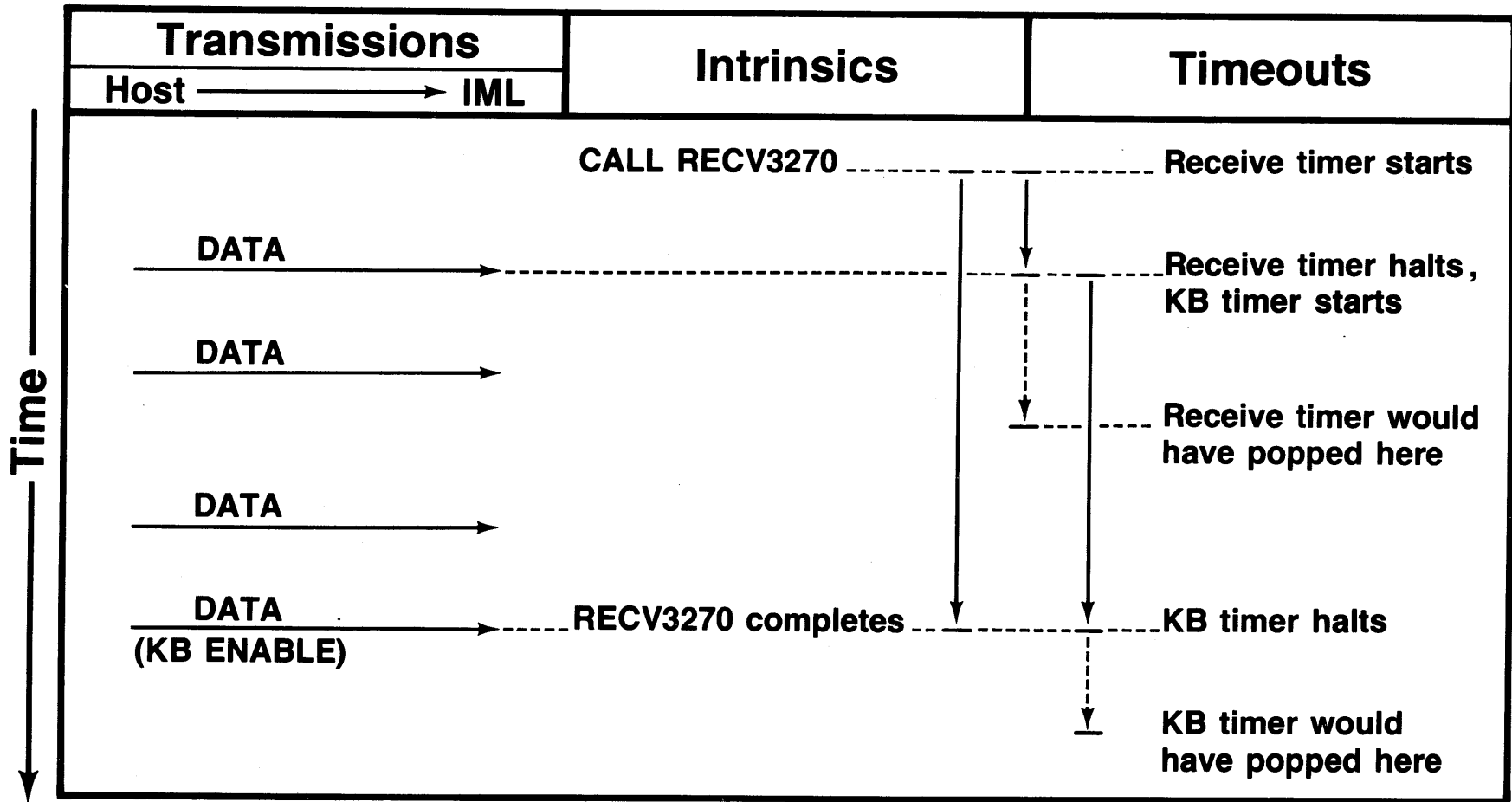
Using Timeout Parameters - Case 1

The completion of a call to RECV3270 can be affected by two optional timeouts: the Receive Timeout and the Keyboard Enable Timeout. The Receive Timeout forces completion of the call if no data has been received from the host after a specified amount of time has passed. The Keyboard Enable Timeout delays completion of the call until the host enables the Keyboard, or until the Keyboard Enable timer pops, whichever occurs first.

Case 1 - Host sends data in one select sequence, and unlocks the keyboard within that sequence.

- o Use a Receive Timeout to insure your program will not suspend indefinitely if the host system doesn't respond. Recommended Value: long enough so that slow host response won't cause it to run out, usually on the order of 300 seconds (5 minutes).
- o Keyboard enable timer makes no difference in this case.

Using Timeout Parameters — Case 2

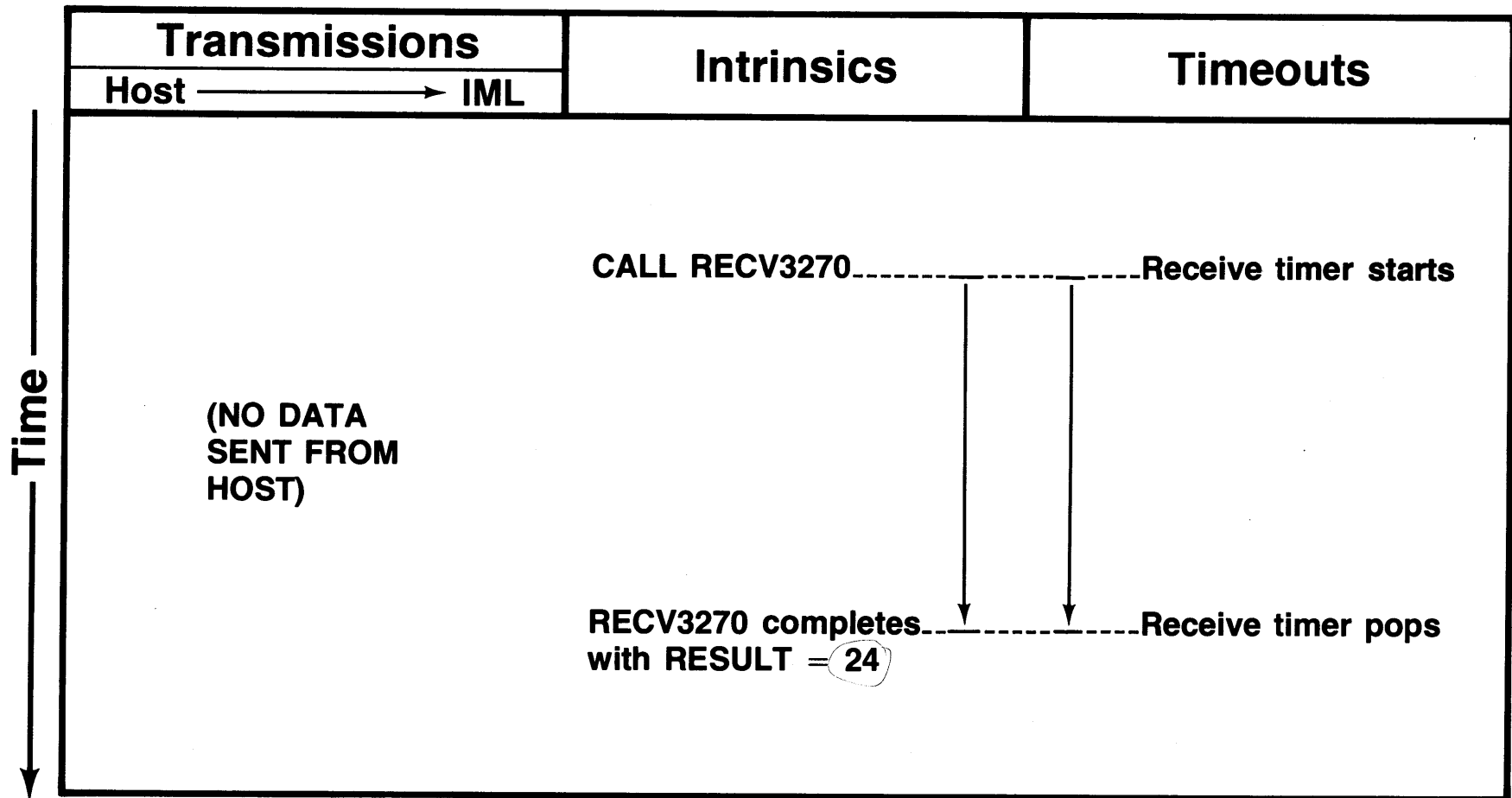


Using Timeout Parameters - Case 2

CASE 2 - Host sends data in one or more select sequences and unlocks the keyboard in the last select sequence.

- o Use a non-zero receive timeout to insure your program won't hang if the host system doesn't send anything.
- o Use a non-zero keyboard enable timeout to cause your program to wait until the host system enables your keyboard, or until the keyboard enable timer pops, whichever comes first. Recommended Value: usually around 180-300 seconds (3-5 minutes).

Using Timeout Parameters — Case 3

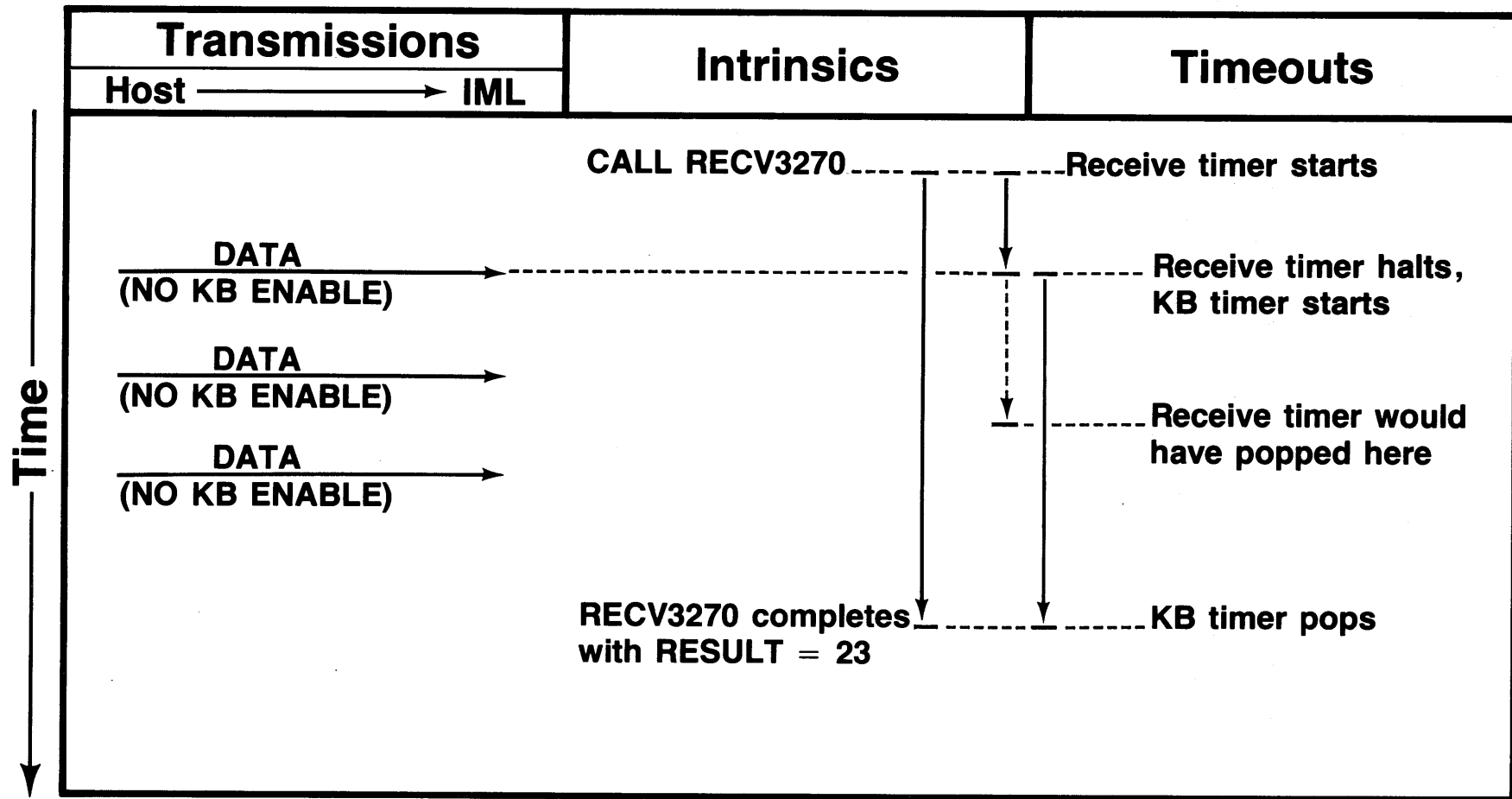


Using Timeout Parameters - Case 3

CASE 3 - Host doesn't send data back.

If the host system does not send any data back to your program and you have the Receive Data Timeout set, the RECV3270 call will complete when the timer pops. Your program will get a non-zero result code and can then take appropriate action. (Perhaps the host system crashed....).

Using Timeout Parameters — Case 4



Using Timeout Parameters - Case 4

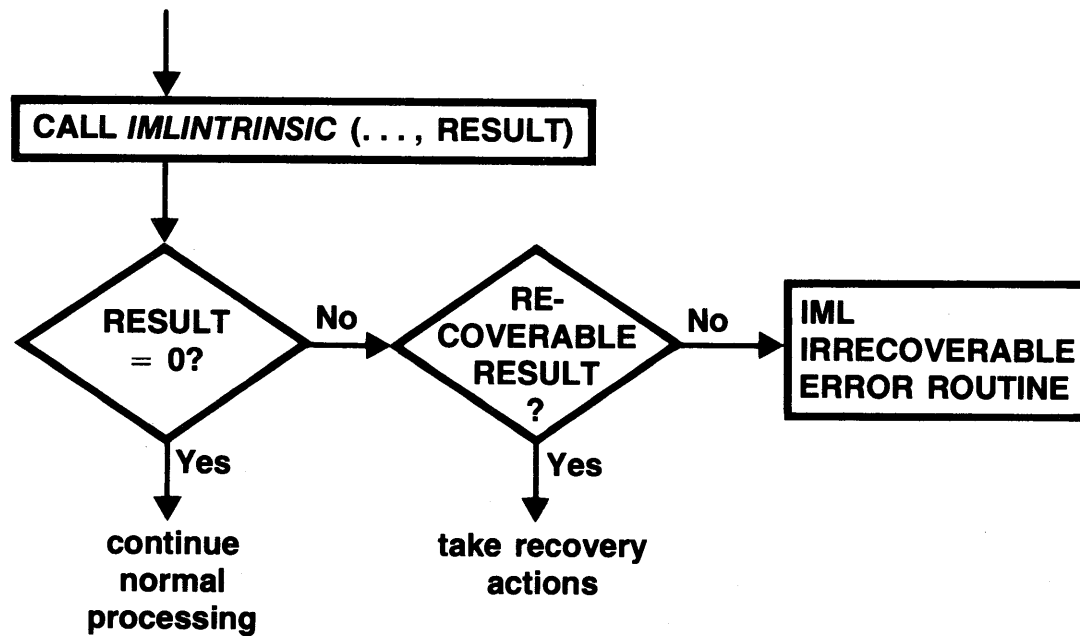
CASE 4 - Host doesn't unlock your (logical) keyboard.

If the host doesn't unlock your keyboard, and you have set the keyboard enable timer, the RECV3270 call will complete when the keyboard enable timer pops. Your program will get a non-zero result code from the RECV3270 and can take appropriate action. (Perhaps the host sent you an error message).

Timer Values

- o Specify how long these timers are in your OPEN3270 call.
- o Specify these timers to be long enough so the host has plenty of time to respond normally. In most cases, a value on the order of a few minutes for these timers should be adequate.

Result Code Checking

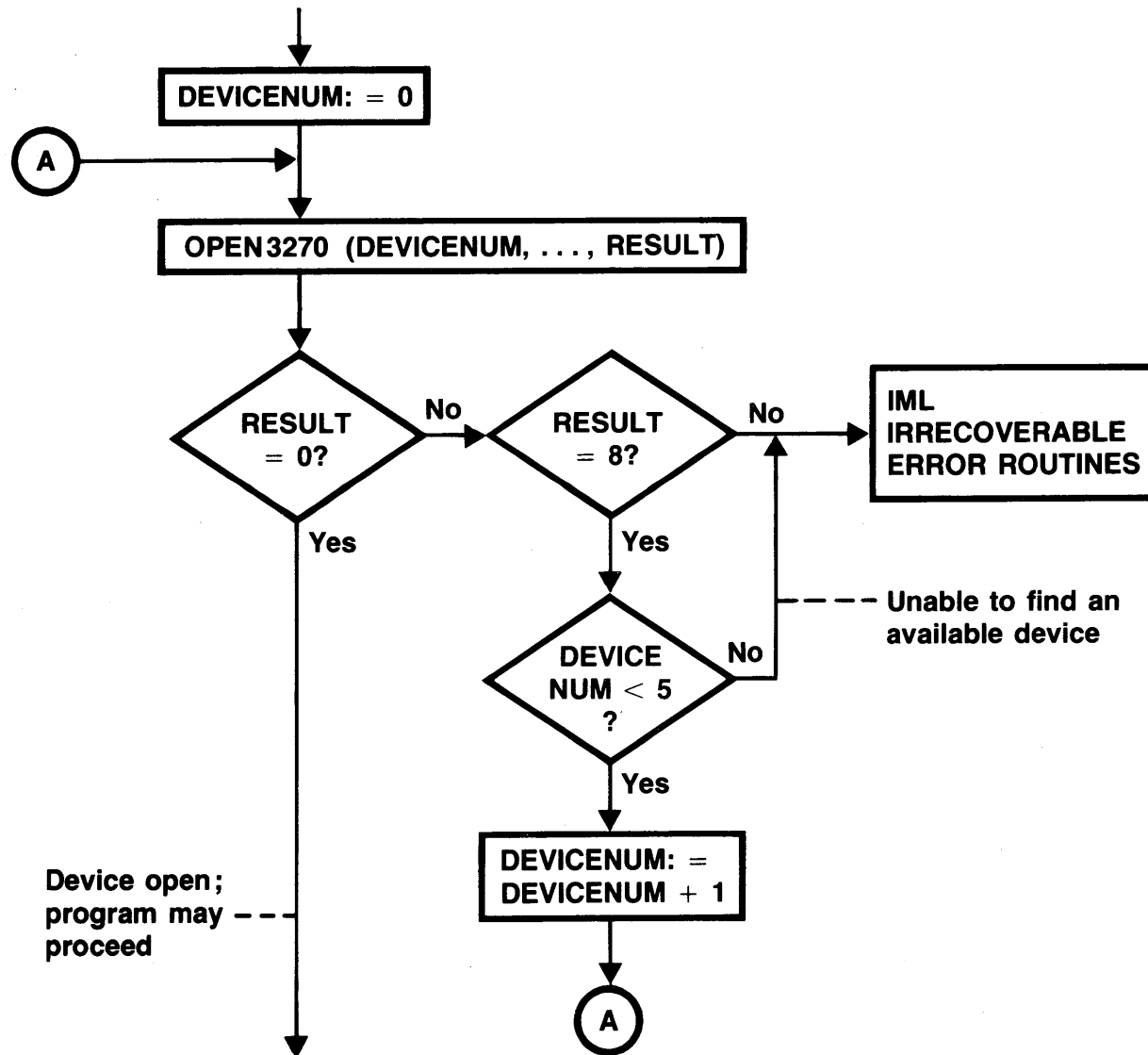


Result Code Checking

Most of the IML intrinsics return a RESULT code which indicates what happened with the intrinsic call. The different RESULT codes have the same meaning across all of the IML intrinsics, i.e., a RESULT=9 has the same meaning no matter which IML intrinsic it came from.

In some cases a non-zero RESULT may indicate a recoverable error. A production program should check for recoverable errors, and try to recover from these. Whether or not a particular RESULT is recoverable may depend on which intrinsic was called, and where your program is in its processing logic.

Recoverable Result Code: Example 1

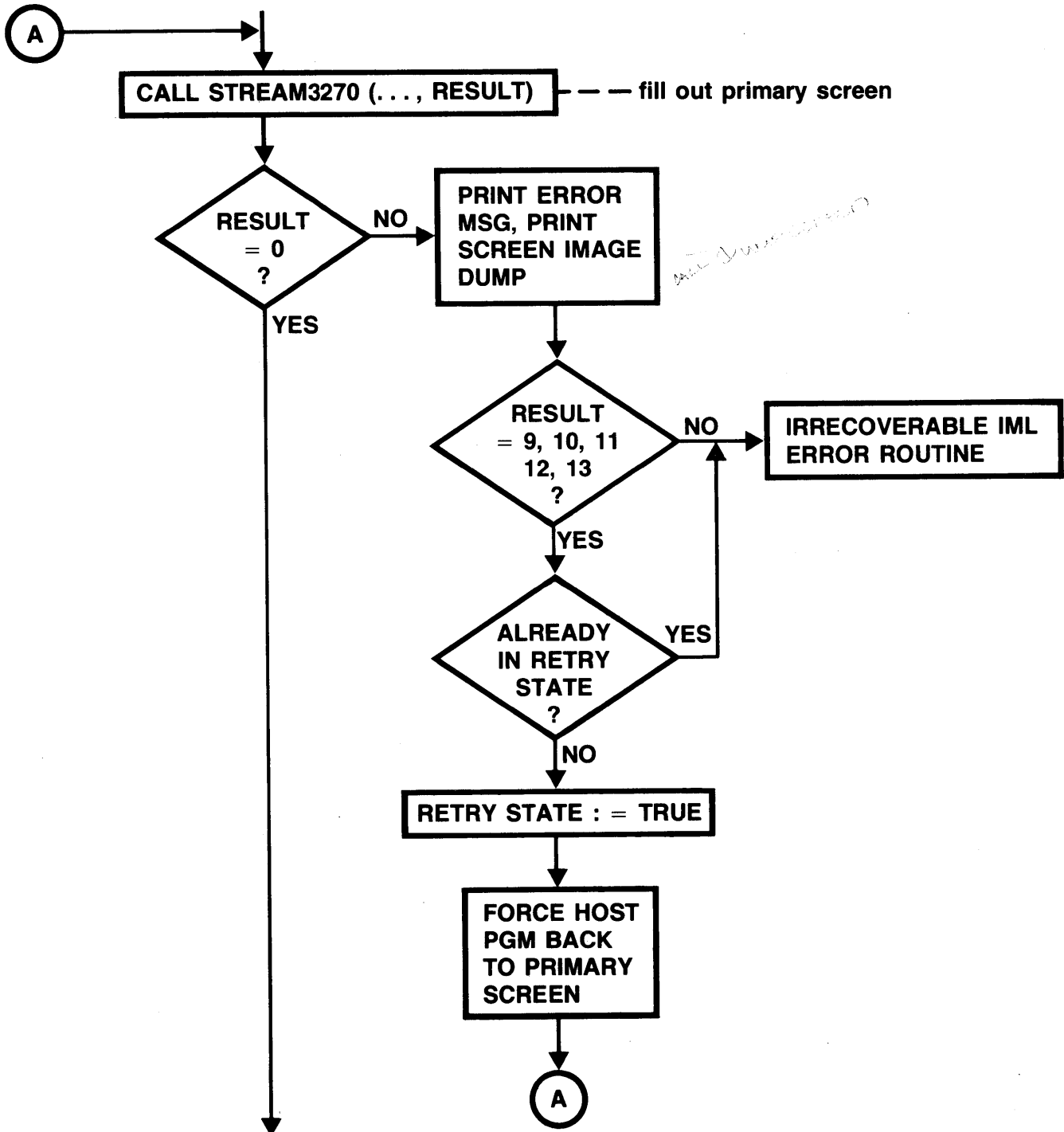


Recoverable Result Code: Example 1

Assume that you are writing an accounting program that is allowed access to IML device numbers 0 thru 5. Other programs also have access to these devices, so your program will need to be able to search for a currently unused device in order to begin 3270 device emulation.

You may code a RESULT code check and retry into your routine which calls OPEN3270.

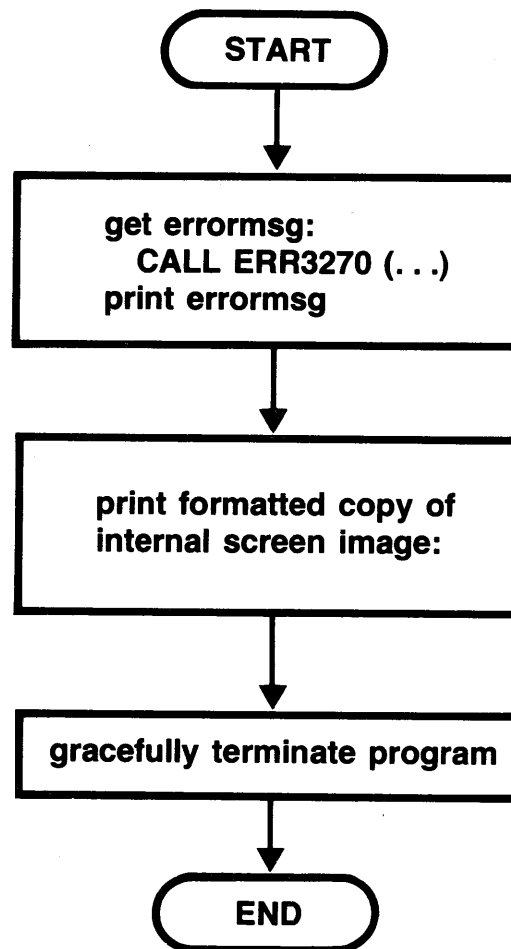
Recoverable Result Code: Example 2



Recoverable Result Code: Example 2

Assume a distributed data base application setup between the host system and the 3000. Assume the host application is programmed so that if an terminal user in the middle of a transaction presses PA1, the transaction is aborted, and the host applications goes back to its primary or main menu. An IML application program could use this feature of the host application to force it back to a "common ground", thereby re-establishing synchronization between programs. This could be useful in the event something unexpected happens, like a host system operator warning message.

IML Irrecoverable Error Routine

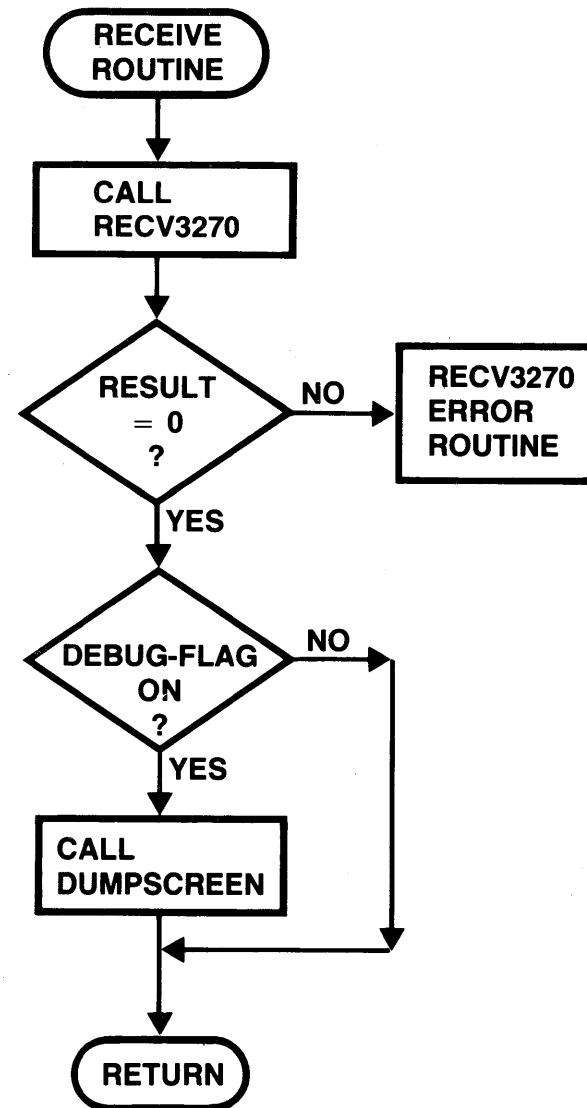
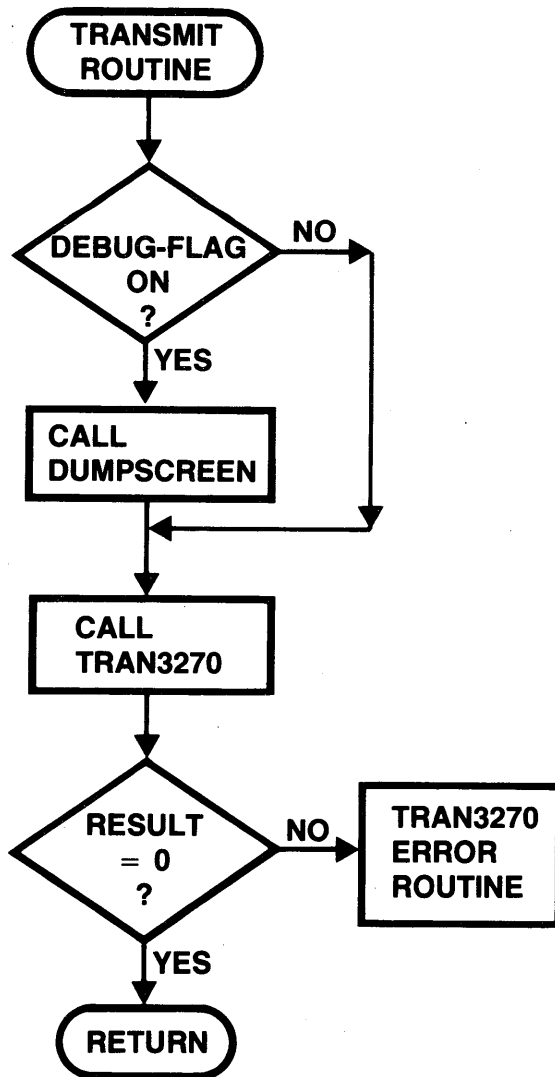


IML Irrecoverable Error Routine

- o document the error
 - * ERR3270 intrinsic
 - Accesses IML/3000 message catalog
 - Call it to retrieve message associated with error(s)
 - * Use a screen image dump routine to get formatted copy of current screen image.
 - * These two items will often be sufficient to allow debugging.
- o exiting IML/3000
 - * Simple Programs:
 - Just end program
 - MPE will close any files and IML devices.
 - * More complicated programs with multiple son processes or IMAGE/3000 database access:
 - Probably advisable to terminate any son processes gracefully and close any IMAGE databases before terminating program.

Close 2270 -

Debugging with a "Dumpscreen" Procedure



Debugging with a "Dumpscreen" Procedure

When an IML intrinsic returns an irrecoverable RESULT code, it is a tremendous help in the debugging process if you can see what the internal screen image looked like when the error occurred. A routine to do this should be built into every IML application program. This routine, when called, should read the internal screen image, and write it to a file (usually, to the lineprinter) in an easily readable, formatted form.

One example of such a routine is built into IDF. When a LOGIML file equation is issued before running IDF, formatted copies of the screen images processed by IDF are sent to the LOGIML file.

Before writing an IML application program, ask your System Manager or IML manager if such a routine is available on your system and how to access it.

One example of where such a routine would be useful is in your program's irrecoverable IML error handling routine.

Another example of how this routine would be useful is shown in the slide.

programming iml/3000

Sample IML/3000 Application

“Ross Corporation parts database” example

- **Application overview**
- **Information needed about host program**
- **The IML/3000 application program**

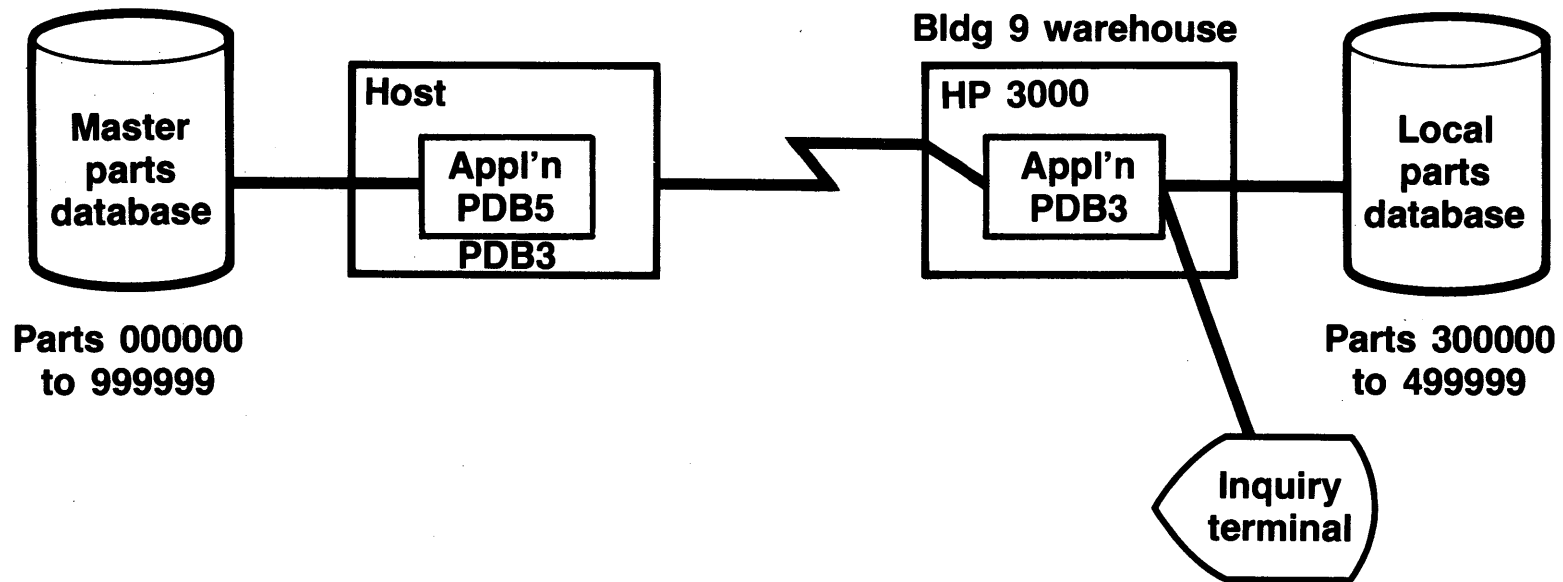
programming iml/3000

Sample IML/3000 Application

sample iml/3000 application

Application Overview

Ross Corporation parts database system



Application Overview

Ross Corporation has a Master Parts Database on their Host system. Bldg 9 is one of Ross's warehouses. This center works mostly with parts in the number range 300000 to 499999, but sometimes has to inquire about parts whose numbers are outside this range.

In the past, bldg 9 had four 3270 type terminals connected to their host processor. Every inquiry made from Bldg 9 was processed by the host system CPU, even for inquiries in the range 300000 to 499999.

In an effort to offload some of these transactions from their host system, Ross Corp. acquired an HP 3000 computer with V/3000, IMAGE/3000, IML/3000, and MRJE/3000 and distributed the parts data base over both machines. The HP 3000 has local control over all parts with numbers from 30000 to 49999. The host system has primary control over all the other part numbers in the database. Since consolidated reports have to be generated on the host machine once a week a batch MRJE job is used to transmit a copy of the HP 3000 database the host system.

The HP 3000 application program that accesses the host parts database on-line via IML is called PDB3 (for Parts DataBase 3000) and is written in COBOL II/3000 (ANS74 Standard COBOL). PDB3, on the 3000, accesses the same host application program that bldg 9 workers used to use with their 3270 terminals. The host master parts database system is called PDBS (Parts DataBase System), and the host application program is the Part Stock Update program.

sample iml/3000 application

Information Needed About Host Program

- **Host screen definitions**
- **Host program logic and screen sequence**
- **Additional host program information**

sample iml/3000 application

Information Needed About Host Program

In order to write an IML application program that will communicate with the host system certain kinds of information about the host system and host application program are needed.

information needed about the host program

Host Screen Definitions

Screen ID: LOGON
Screen #001

Screen ID: LOGON (ERROR MSG) 002

	0	1	2	3	4	5	6	7																														
0																																						
1																																						
2																																						
3																																						
4																																						
5																																						
6																																						
7																																						
8																																						
9																																						
10																																						
11																																						
12																																						
13																																						
14																																						
15																																						
16																																						
17																																						
18	□																																					
19	□																																					
20																																						
21																																						
22																																						
23																																						

information needed about the host program

Host Screen Definitions

Screen ID: LOGON (ERROR MSG)
Screen #002

Screen ID: SELECT PGM 100

	0					1					2					3					4					5					6					7				
	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9
0																																								
1																																								
2																																								
3																																								
4																																								
5																																								
6																																								
7																																								
8																																								
9																																								
10																																								
11																																								
12																																								
13																																								
14																																								
15																																								
16																																								
17																																								
18																																								
19																																								
20																																								
21																																								
22																																								
23																																								

information needed about the host program

Host Screen Definitions

Screen ID: SELECT PGM
Screen #100

Screen ID: PART STOCK UPDATE 200

0	1	2	3	4	5	6	7	8	9
0									
1									
2									
3									
4									
5									
6									
7									
8									
9									
10									
11									
12									
13									
14									
15									
16									
17									
18									
19									
20									
21									
22									
23									

PART STOCK UPDATE

PART NUMBER

QUANTITY

ACTIONS

PF1 - REPORT QUANTITY ON HAND

PF2 - ADD STOCK

PF3 - WITHDRAW STOCK

PF12 - EXIT

information needed about the host program

Host Screen Definitions

Screen ID: PART STOCK UPDATE
Screen #200

Screen ID: PART STOCK UPDATE ERR1 201

0	1	2	3	4	5	6	7	8	9
0	1	2	3	4	5	6	7	8	9
1									
2									
3									
4									
5									
6									
7									
8									
9									
10									
11									
12									
13									
14									
15									
16									
17									
18									
19									
20									
21									
22									
23									

PART STOCK UPDATE

PART NUMBER

QUANTITY

ACTIONS

PF1 - REPORT QUANTITY ON HAND

PF2 - ADD STOCK

PF3 - WITHDRAW STOCK

PF12 - EXIT

** PLEASE ENTER PART NUMBER **

information needed about the host program

Host Screen Definitions

Screen ID: PART STOCK UPDATE ERR1
Screen #201

Screen ID: PART STOCK UPDATE ERR2 202

0	1									2									3									4									5									6									7								
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9														
0																																																															
1																																																															
2																																																															
3																																																															
4																																																															
5																																																															
6																																																															
7																																																															
8																																																															
9																																																															
10																																																															
11																																																															
12																																																															
13																																																															
14																																																															
15																																																															
16																																																															
17																																																															
18	0	XX PLEASE ENTER QUANTITY XX																																																													
19	0																																																														
20																																																															
21																																																															
22																																																															
23																																																															

information needed about the host program

Host Screen Definitions

Screen ID: PART STOCK UPDATE ERR2

Screen #202

Screen ID: PART STOCK UPDATE ERR3 203

0	1	2	3	4	5	6	7																							
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	
0																														
1																														
2																														
3																														
4																														
5																														
6																														
7																														
8																														
9																														
10																														
11																														
12																														
13																														
14																														
15																														
16																														
17																														
18																														
19																														
20																														
21																														
22																														
23																														

information needed about the host program

Host Screen Definitions

Screen ID: PART STOCK UPDATE ERR3
Screen #203

information needed about the host program

Host Screen Definitions

Screen ID: REPORT QOH
Screen #300

Screen ID: ADD STOCK 400

	0	1									2									3									4									5									6									7								
	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9				
0																																																																
1																																																																
2																																																																
3																																																																
4																																																																
5																																																																
6																																																																
7																																																																
8																																																																
9																																																																
10																																																																
11																																																																
12																																																																
13																																																																
14																																																																
15																																																																
16																																																																
17																																																																
18																																																																
19																																																																
20																																																																
21																																																																
22																																																																
23																																																																

information needed about the host program

Host Screen Definitions

Screen ID: ADD STOCK
Screen #400

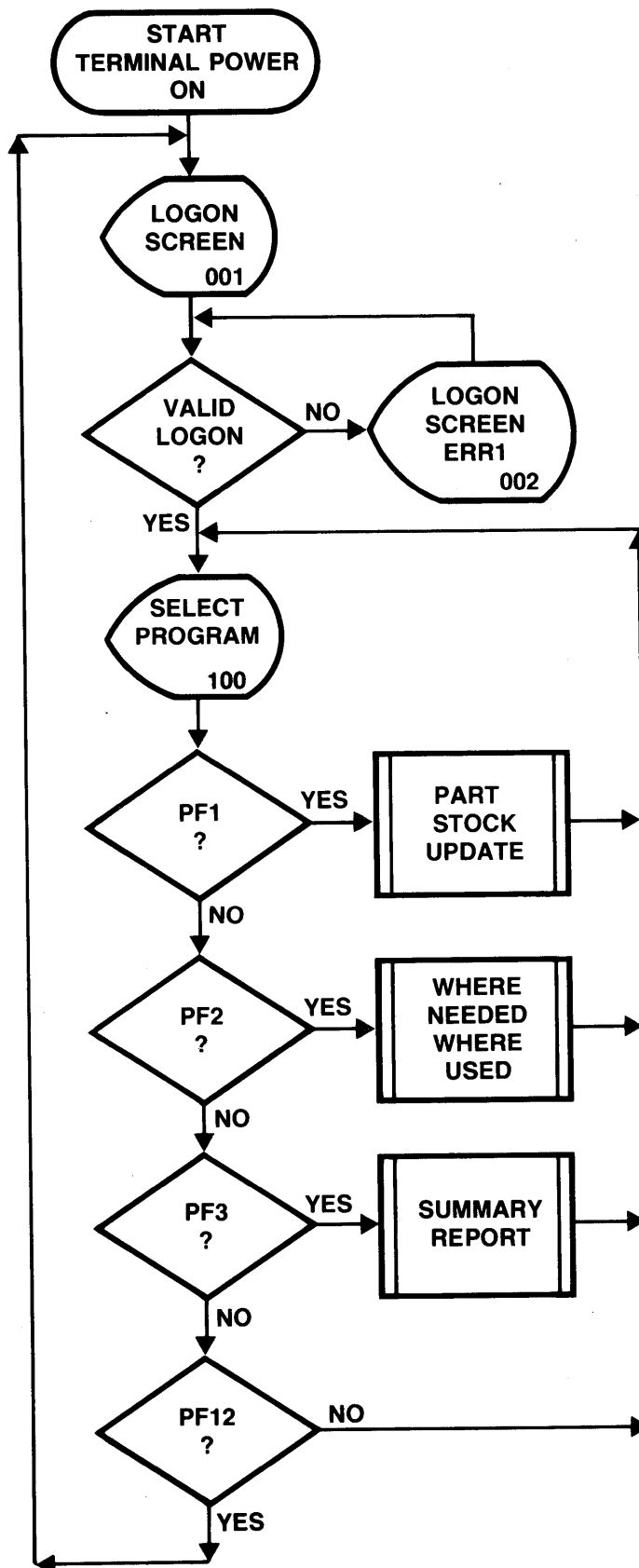
Screen ID: WITHDRAW STOCK 500

	0										1										2										3										4										5										6										7																			
	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9																				
0																																																																																										
1																																																																																										
2	WITHDRAW STOCK																																																																																									
3																																																																																										
4																																																																																										
5	PART NUMBER																																																																																									
6																																																																																										
7	OLD STOCK ON HAND																																																																																									
8	WITHDRAW QUANTITY																																																																																									
9																																																																																										
10	NEW STOCK ON HAND																																																																																									
11																																																																																										
12																																																																																										
13	ACTIONS																																																																																									
14																																																																																										
15	ENTER - VERIFY WITHDRAWAL																																																																																									
16	ANY PF KEY - REJECT WITHDRAWAL																																																																																									
17																																																																																										
18																																																																																										
19																																																																																										
20																																																																																										
21																																																																																										
22																																																																																										
23																																																																																										

information needed about the host program

Host Screen Definitions

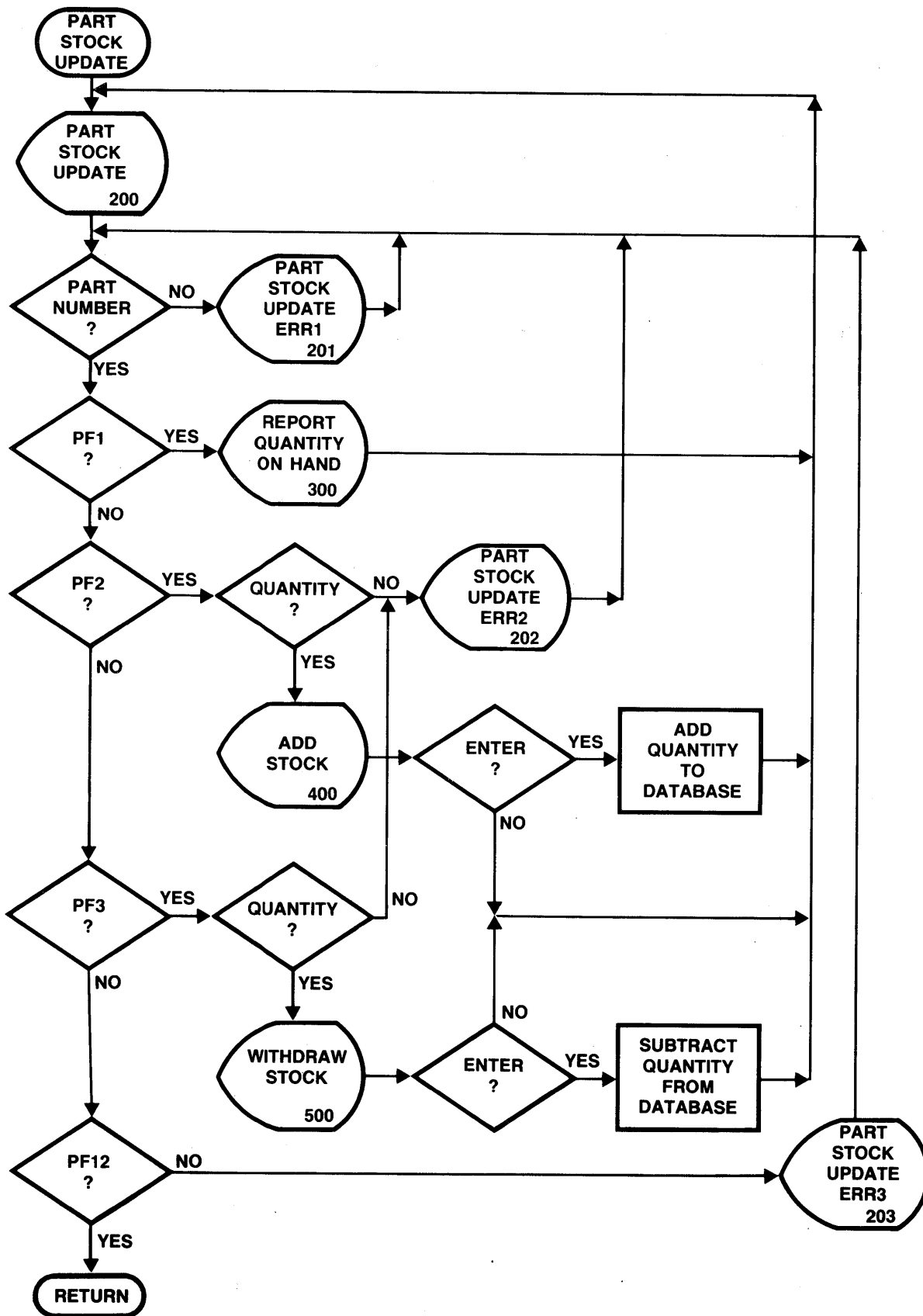
Screen ID: WITHDRAW STOCK
Screen #500



information needed about the host program

Host Program Logic:
Logging On and Program Selection

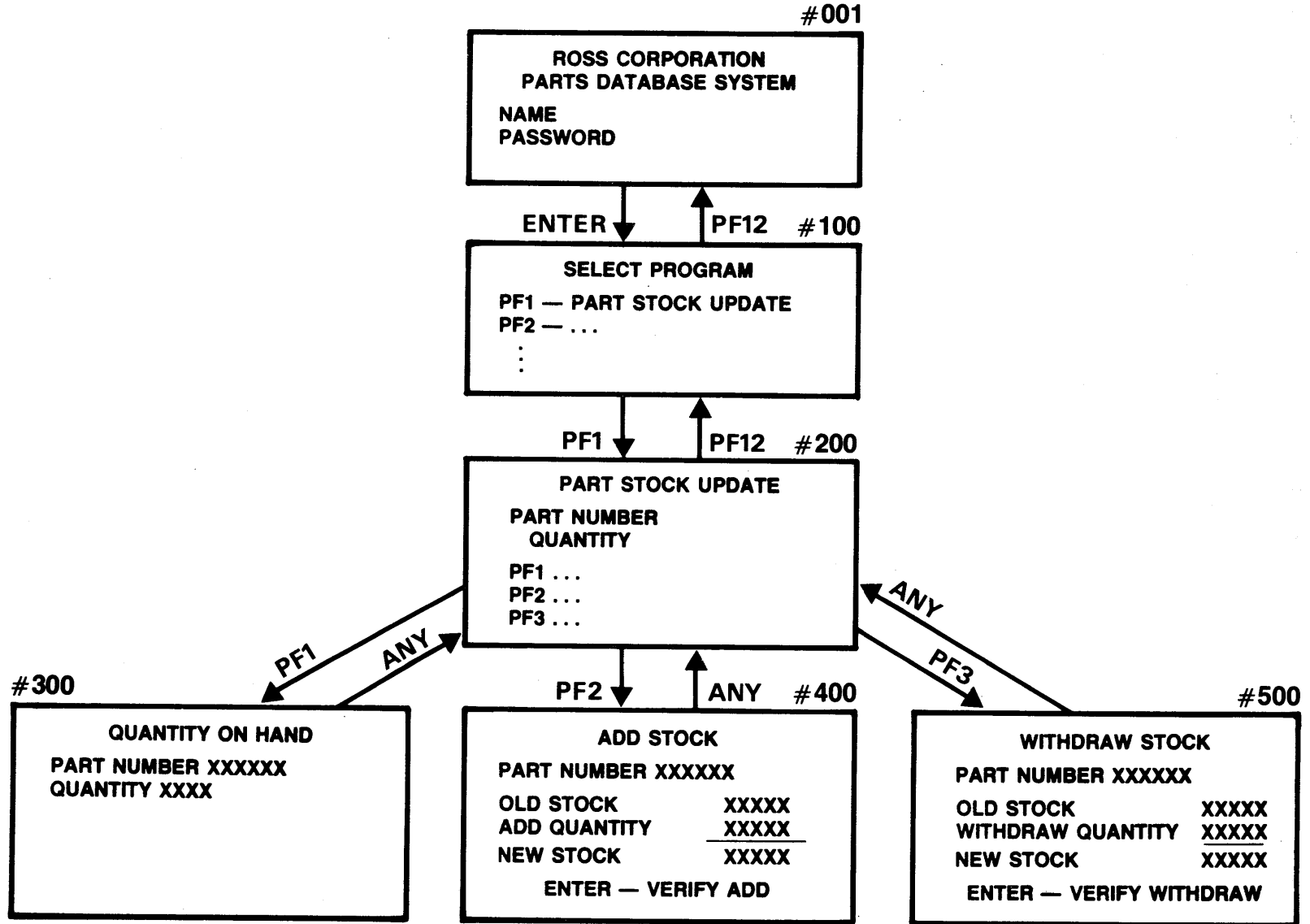
- o This is the logic that PDBS, the host system, goes through when a 3270 terminal is first powered on.
- o PDB3, the 3000 IML application program, wants to get to the PART STOCK UPDATE block, which is the host application program that bldg 9 was using before.
- o The next slide describes the logic for the PART STOCK UPDATE block.



information needed about the host program

Host Program Logic:
Part Stock Update Host Application

Screen Sequence Summary



information needed about the host program

Screen Sequence Summary

information needed about host program

Additional Host Information

- **Logon procedures**
- **Timeouts**
- **Device numbers that may be used**
- **Sample questionnaire supplied in textbook**

information needed about the host program

IML/3000: HOST APPLICATION QUESTIONNAIRE

1. HP 3000 APPLICATION: PDB 3 / PDBIML
2. HOST APPLICATION: PDBS / PART STOCK UPDATE
3. IML CONFIGURATION FILENAME: IML CONF. PUB. SYS
4. 3270 DEVICE NUMBERS THAT MAY BE USED FOR THIS APPLICATION:
DEVICE NUM: 8, 9, 10, 11, 12
DEVTYPE : 3278 (devtype 2)
BUFFSIZE : 1920
5. HOST SYSTEM LOGON USERID: JOHN DOE
PASSWORD: PASSWORD

6. Does Host system automatically transmit a logon screen when terminal is powered on (i.e. after OPEN3270 call)?

YES NO

7. Longest acceptable Host response time before assuming a Host or line failure? (Use this value for the receive data timeout in the OPEN3270 call. Use a value of 0 to disable this timeout.)

180 seconds.

8. For this application, does the Host system normally unlock the keyboard when ready to accept data?

YES NO

If the answer is NO, skip to item 11 and enter a value of 0.

9. If the answer to question 8 is YES, does the Host always send all of its data for a screen and unlock the keyboard within a single select sequence? (Ask your communications expert about this. If you are not sure, answer NO.)

YES NO

If the answer is YES, skip to item 11 and enter a value of 0.

10. If the answer to questions 8 and 9 are (in order) YES and NO, what is the longest acceptable time between receiving the first select sequence and receiving the keyboard unlock? Enter this value in item 11.

11. Keyboard unlock timeout for OPEN3270 call:

120 seconds.

information needed about the host program

12. When a terminal is powered on (when an IML program calls OPEN3270), does the host system automatically logoff any old leftover previous sessions on that terminal?

YES NO

13. If the answer to item 12 is NO, consider how your IML application can distinguish whether or not you are in the host system logon screen, and if not, how your program can force the host system to get you there.

CHECK BUFFER POSITION 906 FOR "NAME".

IF TRUE, THEN WE ARE IN

SCREEN #001 OR SCREEN #002.

IF FALSE, KEEP ENTERING PF12

UNTIL WE GO BACK UP THE

MENU TREE BACK TO SCREEN #001.

HP top (0,0)
IBM (1,1)

BUFFER POSITION (0)
(179,
POSITION (19)

- ① Dilco logo
- ② Press Enter
- ③ 2 screens

OFFON - LOG OFF ROSCOE

PF12 - gives menu w/ Dilco logo

ERR = 6
FAIL TO PASS SECURITY, CONFIG, ETC
= 8
3270 BUSY

- ① OPEN 3270
- ② RECV 3270 LOGIN SCREEN
- ③ TRAN 3270 w/ AID value and ENTER
- ④ RECV 3270 MENU SCREEN
- ⑤ WRITE 2270 (ROW 22 COL 1, FIELD 1)
- ⑥ RECV 3270 WAIT MSG
- ⑦ RECV 3270 ROSCOE KEY
- ⑧ WRITE 3270 MSC.I.JLM + ENTER
- ⑨ RECV 3270 WELCOME
- ⑩ READFIELD FOR WELCOME MSG
- ⑪ RECV 3270
- ⑫ WRITE (OFFON + ENTER)
- ⑬ RECV 3270
- ⑭ TRAN 3270 PF12 + ENTER
- ⑮ RECV 3270
- ⑯ RECV 3270 - MENU

97
1700/1760

READFIELD - SIGNOFF 1204 A - IS SIGNOFF GO 15.

sample iml/3000 application

The IML/3000 Application Program

- **PDB3 program logic**
- **Source code highlights**
- **PDBIML source code**

sample iml/3000 application

The IML/3000 Application Program

the iml/3000 application program

PDB3 Program Logic

PDB3 has 4 modules

- PDBMAIN**
- PDBTERM**
- PDBDB**
- PDBIML**

the iml/3000 application program

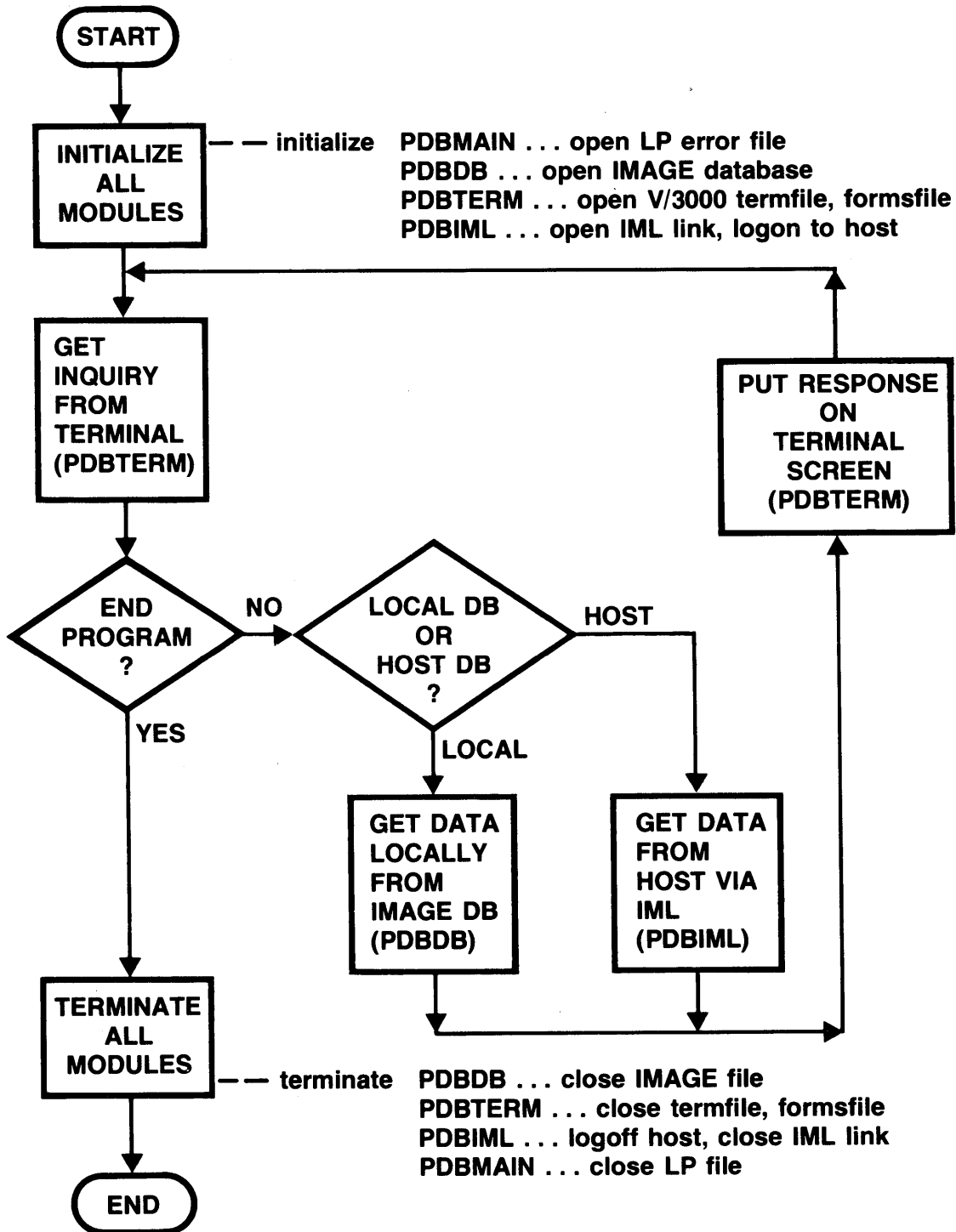
PDB3 Program Logic

PDB3 is divided into four major modules:

1. PDBMAIN - main routine
2. PDTERM - terminal I/O and forms handling using V/3000
3. PDBDB - local database access using IMAGE
4. PDBIML - host database access using IML

Each of these modules is a separately compiled COBOL routine.

PDBMAIN Logic

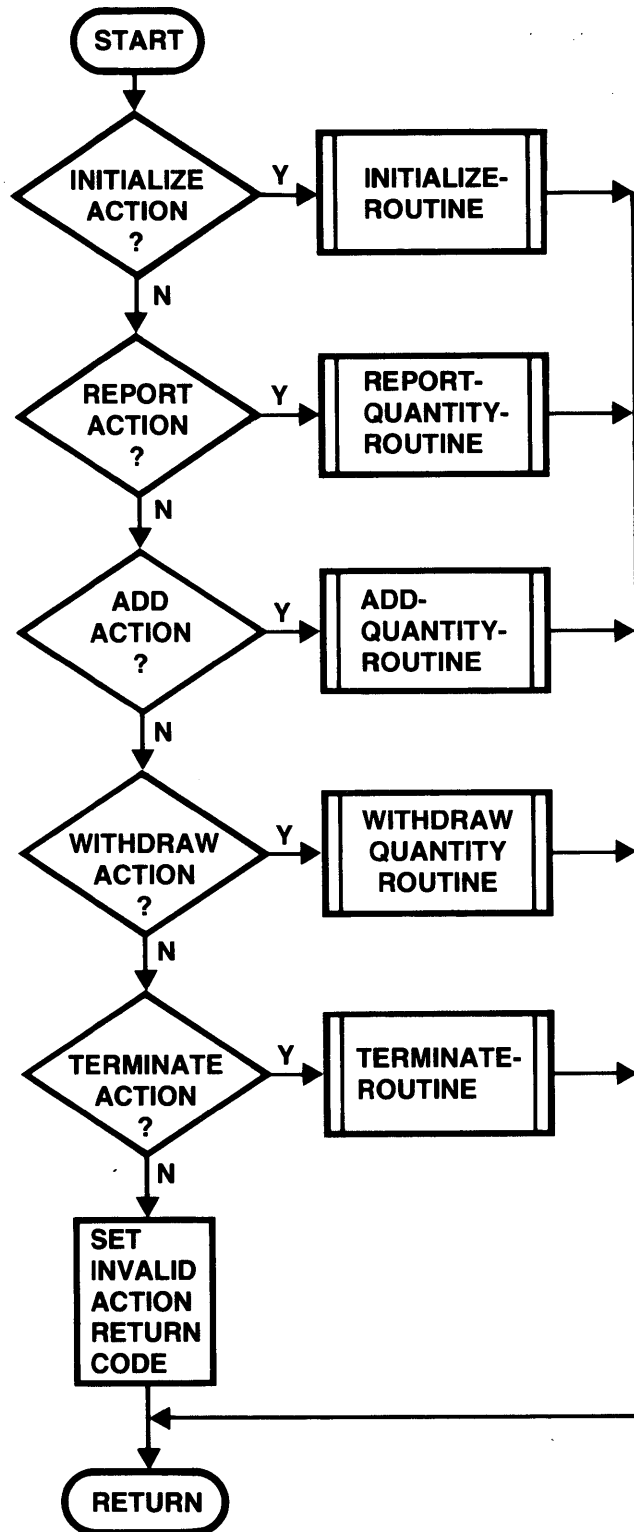


the iml/3000 application program

PDBMAIN Logic

- o PDBMAIN is the main routine. It calls PDBTERM, PDBDB, and PDBIML in a single process (no son processes).
- o PDBDB, PDBTERM, and PDBIML are COBOL subroutines.

PDBIML Logic



the iml/3000 application program

PDBIML Logic

- o There are three major tasks PDBIML will be asked to do:
 - * Open the IML link to host system, log on to host, and call up the proper host application program.
 - * Execute transactions on the host program, and retrieve data.
 - * Terminate the host application program, logoff the host and release the IML link.
- o The flowchart for PDBIML is shown in the slide.
 - * Details of error handling not shown here for clarity.
- o Constraints:
 - * PDBIML should never terminate the entire program if an error occurs. Instead, PDBIML should return an error code to PDBMAIN. This way, PDBMAIN can "clean up" before terminating.
 - * \$STDLIST is a terminal under control of V/3000. This affects how IML error messages are handled.

the iml/3000 application program

Source Code Highlights

- **Refer to source listing**

Source Code Highlights

The source listing for PDBIML follows this section. The line numbers referred to below are the compiler line numbers on the far left side of the source listing.

LINE NOS.	COMMENTS
00001	PDBIML is a subprogram. It will be compiled into the same USL file as the PDB3 main module PDBMAIN and the other PDB3 subprogram modules PDBDB and PDBTERM. This USL file can then be prepared into a program file for execution.
00021-00055	These are the IML variable declarations. Only the variables used by PDBIML are actually declared, thus saving stack storage space.
00077-00090	PDBIML-LINK-RECORD and PART-RECORD are the two parameters passed to PDBIML. Note that PDBIML does not have any FD file descriptors.
00091-00141	This is the main or driving part of PDBIML. See the comments in the code.
00134-00138	Resetting the RETRY-FLAG. See the comments immediately above.

00141 Line 141 contains the only exit from
PDBIML. It has a label because some of the
error handling code later on does a direct
branch to this statement if a
non-recoverable error is discovered.
Note that a COBOL EXIT PROGRAM statement
is merely a subroutine return. Control
will be passed back to PDBMAIN, the main
program that called PDBIML.

00142-00178 INITIALIZE-ROUTINE.

00155 LOCATION is used in PDBIML only by
DUMPSCREEN. If an error occurs a
DUMPSCREEN screen image trace will be
attempted. The DUMPSCREEN printout will
include the value of LOCATION - this way we
can identify where PDBIML was when
DUMPSCREEN was called.

00156 The calling program, PDBMAIN, can set
SCREENTRACE-ON to be true. This will cause
a complete DUMPSCREEN screen trace of every
I/O with the host system. Normally,
SCREENTRACE-ON will be set true only for
debugging purposes. This statement opens
the DUMPSCREEN file for future use.

00179-00213 REPORT-ROUTINE. Note how LOCATION is
updated, the various IML variables are
updated, and the appropriate IML intrinsic
paragraphs are performed. Compare this
code with host screens #200 and #300 to see
how the IML variables and intrinsics "fill
out" the 3270 menu and "press" the "enter"
or "pf" key to send and receive data from
the host system.

00214-00257 ADD-ROUTINE. Techniques used here are similar to those used in REPORT-ROUTINE. Comments in the code describe what is happening.

00258-00301 WITHDRAW-ROUTINE. Similar to ADD-ROUTINE.

00302-00327 TERMINATE-ROUTINE. This routine causes PDBIML to logoff the host system, then it closes the IML terminal file, and, if necessary, closes the DUMPSCREEN file.

00328-00400 IMLERR-ROUTINE. This is the main routine to handle unexpected IML errors. An unexpected IML error could be anything from a line failure to an unexpected host system operator warning message appearing unexpectedly in PDBIML's 3270 screen image. This routine needs to provide the following information: what is the error number, what is the IML error message, and if possible, a DUMPSCREEN screen trace. Finally, if the error is a recoverable error, this routine will attempt automatic recovery by resynchronizing the host program with PDBIML, and re-trying the original action code. The comments in the source code describe what is happening.

00348-00349 If the session device is under control of V/3000, as is assumed here, it would not be a good idea to write non V/3000 formatted error messages to the terminal. In PDBIML, error messages are written to the system console. In other IML application programs that have this same problem, it may be desirable to have a special error reporting file directed to a lineprinter or to disc.

00364 Traceable error codes for PDBIML are defined on line 27. The error code meanings are defined in the IML Reference Manual in Chapter 6. The ones used here include: 10, 11, 12, 13, 17, 21, 23, 24, and 38.

00388 Recoverable error codes are defined on line 30. Among the ones included here are: 9, 10, 11, 12, and 13.

00401-00441 GET-TO-SCREEN-200-ROUTINE. This routine is used to (re)synchronize the host program with PDBIML.

00442-00469 LOGOFF-HOST-ROUTINE. This routine is performed by GET-TO-SCREEN-200-ROUTINE and by TERMINATE-ROUTINE. PDBIML has a very simplistic case where a logoff can be forced on the host system no matter what host screen the 3270 is in by entering PF12 enough times in a row. Programs running on other host systems will probably require different logic to force a logoff from the host system.

00458 Buffer position 906 is row 11 column 26. Compare this with host screens #1 and #2. Note that we are logged off if we get either screen. We can logon from either of these screens as well.

00470-00504 DUMPSCREEN routines.

00505-00545 OPEN3270-ROUTINE.

00518

PDBIML has been ALLOW'd in the IML configuration file to use device numbers 8 through 12. This statement keeps trying different device numbers until it finds an available IML device number, it runs out of eligible device numbers, or a fatal error occurs.

00546-00586

IML INTRINSIC CALLS. Note how all the IML intrinsic calls are put in one location in PDBIML. All of the input IML variables are set either via MOVE statements or via previous IML intrinsic calls before each intrinsic paragraph is performed. For many of the intrinsics the same error checking will apply each time they are called, so the result code checking is included here as well.

00572-00581

IML-RECV3270 and IML-TRAN3270. Note how, if SCREENTRACE-ON is true, a DUMPSCREEN screen trace will be taken just before each transmission to the host, and just after each receive back from the host.

Additional Considerations

- Logging off the host system
- Writing host and HP 3000 applications together

Additional Considerations

Logging off the host system

Be aware that when your IML program calls CLOSE3270 (or when MPE closes your IML device for you if your program aborts), this does not necessarily mean you are logged off from the host system. This has two implications:

- o For your program's normal termination, code into your program the logic necessary to logoff the host before calling CLOSE3270.
- o When your program first attempts to logon to the host system, your program should check and see if the previous user of this IML device number logged off. If the previous user left his/her host session active, be prepared to deal with this.

Hints on writing host and HP 3000 applications together

- o Whenever possible, use unformatted screens.
- o If possible, define an area on the screen to contain program to program control information, with the rest of the unformatted buffer used for data. Each different screen from the host program could identify itself in the first, say, eight bytes of the buffer. This way the 3000 program could easily check what host screen it was in and decide what to do from there.
- o If transferring multiple records from a file in either direction, use a control field to indicate how big each record is, and what record number you are currently transferring. Transfer more than one record at a time. Use all 1920 bytes of the screen buffer to optimize line time.
- o Remember that 3270s, and hence IML, do not handle transparent data. This means you will need conversion routines on both ends if you want to send binary or packed decimal data, or data that contains control characters.

CS trace: supplement

Reading CS Trace File

CS trace: supplement

Reading CS Trace File

CS trace: supplement

Reading Trace File: Line Information Display

```
*****
* BEGIN TRACING FOR DEVICE 45 *
*****

*****
*-L-I-N-E---I-N-F-O-R-M-A-T-I-O-N---D-I-S-P-L-A-Y*
*****
* LINE NUMBER: 5          LOGICAL DEV. NUMBER: 45 *
* DEV. TYPE: 17          SUBTYPE: 1   VER: A.03.50 *
*          0123456789012345 *
* COPTIONS: 0101100100000011 *
* AOPTIONS: 0000000100011111 *
* DOPTIONS: 0000000100000000 *
* NUMBUFFERS: 249          BUFFSIZE: 1934 (BYTES) *
* INSPEED: 1200           OUTSPEED: 1200 *
* MISCARRAY:             RECEIVE TIMEOUT: 20 SECS. *
*                          LOCAL TIMEOUT: 60 SECS. *
*                          CONNECT TIMEOUT: 900 SECS. *
*                          RESPONSE TIMEOUT: 500 HSECS. *
*                          LINE BID TIMEOUT: 60 SECS. *
*                          NO. ERROR RETRIES: 7 *
*                          CLEAR-TO-SEND DELAY: 00.0 SECS. *
*                          DATA-SET-READY DELAY: DISABLED. *
*                          TRANSMISSION MODE: HALF DUPLEX. *
*                          MMSTAT TRACE FACILITY: ENABLED. *
*                          POLL LOOP DELAY: 0 MSECS. *
*                          POLL REPEAT: 0 *
*                          POLL ENTRY DELAY: 0 MSECS. *
* DRIVERNAME: IOINP0 *
* DOWNLOAD FILE: CSDIMLO.PUB.SYS *
```

(A)

(B)

(C)

Reading Trace File: Line Information Display

o Gives information about line configuration

A: MPE logical device number of IML pseudo device

B: Size of CS buffer(s) that store data to/from INP
- note that size shown in slide is 1934--accomodates 1920
character screen plus protocol and control characters
- for IML, this is not the same size as the line buffer
on the INP

C: Ignore the INSPEED and OUTSPEED values; line speed is
taken from the modem clock.

Reading Trace File: Line Information Display (continued)

```
* CTRACEINFO:  ENTRIES=3          MASK=011111 ← (D)
*              TYPE OF TRACE = ALL, NOWRAP ← (E)
* POLLIST:     ENTRIES=0          INDEX=-1
* PHONELIST:   ENTRIES=0          INDEX=0
* IDLIST:      ENTRIES=0          INDEX=0
* SUPLIST:     GROUPS=1          DESC=00000000
* { GRP 1 STATIONS= 3            DESC=00000001
*   * POLL ID = C1C1
*   * SEL ID = 6161
*   * GEN POLL ID= C1C17F7F
*   * STA 1 COMPONENTS= 1        TYPE=0
*   * (G) { CMP 0 POLL= 4040
*   *     SEL= 4040
*   *   * STA 2 COMPONENTS= 1        TYPE=0
*   *   * (H) { CMP 0 POLL= C1C1
*   *     SEL= C1C1
*   *   * STA 3 COMPONENTS= 1        TYPE=0
*   *   * (I) { CMP 0 POLL= C2C2
*   *     SEL= C2C2
*   * (J) { ERRORCODE: RECOVERABLE=0  IRRECOVERABLE=0
*   *     MSGSENT: 13170          MSGRCV: 0
*   *     RECOVERERRORS: 26        IRRECOVERERRORS: 0
*   * *****
```

Reading Trace File:
Line Information Display (continued)

o Information provided in display:

- D: Mask. Corresponds to "mask" parameter in TRACE command. Mask used in this trace is default of octal 37.
- E: Type of trace. Also correspond to parameters in TRACE command.
- F-I: Pollist. There are one group (control unit) and three stations (3270 devices).
 - F: Group polling ID: C1C1 (hex EBCDIC AA)
Group selection ID: 6161 (hex EBCDIC //)
Gen poll ID: C1C17F7F (hex EBCDIC AA"")
 - G: Station 1 has 1 component. (Components are numbered starting with 0).
For component 0:
Poll ID: 4040 (hex EBCDIC blank blank)
Select ID: 4040 (hex EBCDIC blank blank)
 - H-I: Similar to G.
 - J: Errorcode is the last recoverable and last irrecoverable CS error that occurred.
Msgsent, Msgrecv is cumulative number of text blocks sent/received up to this point in time.
Recovererrors, irrecovererrors is cumulative number of recoverable/irrecoverable CS errors up to this point in time

Reading Trace File: Trace Entry Types

- **POPR**
- **PRCT**
- **PRTX**
- **PEDT**
- **PSTX**
- **PSCT**
- **PCMP**

Reading Trace File: Trace Entry Types

- o POPR. Operation. This type of entry is created each time the physical driver is asked to perform an operation. A brief statement in English identifies the type of operation.
- o PRCT. Receive Control Sequence. This type of entry is created each time a control character sequence is received from the host. The trace entry contains a byte-for-byte copy of what was received.
- o PRTX. Receive Text. This type of entry is created every time a text message is received from the host. The entry contains a byte for byte copy of what was received.
- o PEDT. Editor. Created when the INP scans the PRCT or PRTX entry and decides what it is. The text in this entry is an edited representation of what was received. Use the PRCT or PRTX entries to see what was actually on the line.
- o PSTX. Send text. Created every time the driver sends a text message to the host. Contains a byte for byte copy of what was sent.
- o PSCT. Send Control Sequence. Generated each time the driver sends a control sequence to the host. Contains a byte for byte copy of what was received.
- o PCMP: User Request Completed. Created each time a user request is completed. The entry is a summary of text messages, errors, and other events that have occurred.

CS trace: supplement

Reading Trace File: Trace File Entries Example 1

```
POPR 5      9201.574 POPR REQUEST ID=NONE
              RECEIVE CONTROL SEG
              OUT RFR=% 033626 LENGTH=0
              IN RFR=% 033626 LENGTH=2048

PRCT 6      9201.734 PRCT REQUEST ID=NONE
              067.062 301.301 100.100 055.377 377.377 377.377
              EOT SYN A A ENQ
              377.377 377.377 377.377 377.377 377.377 377.377
              377.377 377.377 377.377 377.377

PEDT 7      9201.734 PEDT REQUEST ID=NONE
              RECV POLL
              000.002 377.377
              NUL STX

POPR 8      9201.834 POPR REQUEST ID=%027364(!2EF4)
              SEND TEXT THEN RECEIVE RESPONSE
              TIMFOUT= 5.000
              OUT RFR=% 033626 LENGTH=27
              IN RFR=% 033626 LENGTH=2048

PSTX 9      9201.884 PSTX REQUEST ID=%027364(!2EF4)
              125.062 062.062 062.002 301.100 175.301 343.021
              SYN SYN SYN SYN STX A A T DC1
              301.120 323.326 307.326 325.100 301.331 361.360
              A & L O G O N A R I O
              360.360 141.301 302.303 304.305
              0 0 / A B C D E

PSTX 10     9201.884 PSTX REQUEST ID=%027364(!2EF4)
              364.003 237.210 377.000
              4 ETX h NUL
```

Reading Trace File: Trace File Entries
Example 1

- POPR-5: RECEIVE CONTROL SEQ. Control sequence from host will be received in next operation.
- PRCT-6:
 - * Control sequence from host, in character and in octal (would be in hex if HEX parameter specified when CSDUMP was run).
 - * EOT SYN = beginning of message
 - * A A space space = address of device polled
 - * ENQ = BSC protocol character
 - * 377 = pad character
- PEDT-7: RECV POLL. INP recognizes at this point that the previous entry is a poll from the host.
- POPR-8: SEND TEXT THEN RECEIVE RESPONSE. IML will send text to host and receive host response.
- PSTX-9 & 10: Text sent by IML to host:
 - * SYN = BSC synchronization character
 - * STX = BSC protocol (Start-of-Text)
 - * A blank = address of device (control unit 1 device 0)
 - * ' = AID (ENTER)
 - * AT = cursor address
 - * DC1 = Set Buffer Address order
 - * A& = buffer address
 - * LOGON AR1000/ABCDE4 = data from modified field
 - * ETX = BSC protocol character (End-of-Text)
 - * %237.210 = block check characters

CS trace: supplement

Reading Trace File: Trace File Entries Example 1 (continued)

PRTX 11 9202.464 PRTX REQUEST ID=%027364(!2EF4)
002.047 361.301 021.135 177.035 100.021 301.117
STX ESC 1 A DC1) " GS DC1 A !
035.100 027.003 077.326 377.377 377.377 377.377
GS ETX SUB 0

PEDT 12 9202.474 PEDT REQUEST ID=%027364(!2EF4)
RECV TEXT ERROR
000.012 054.001 047.361 301.021 135.177 035.100
NUL SOH ESC 1 A DC1) " GS
021.301 117.035 100.027 377.377
DC1 A ! GS

POPR 13 9202.474 POPR REQUEST ID=%027364(!2EF4)
SEND CONTROL SEQ THEN RECEIVE RESPONSE
SEND SEQ=NAK TIMEOUT= 20.000
OUT BFR=% 033626 LENGTH=27
IN BFR=% 033626 LENGTH=2048

PSCT 14 9202.474 PSCT REQUEST ID=%027364(!2EF4)
125.062 062.062 062.062 075.377
SYN SYN SYN SYN SYN NAK

Reading Trace File: Trace File Entries
Example 1 (continued)

- o PRTX-11: Host's response to IML text:
 - * STX = Start of Text
 - * ESC 1 = 3270 Write command
 - * A = Write Control Character
 - * DC1)" = Set Buffer Address and address characters
 - * GS blank = Start Field and attribute byte
 - * DC1 A! = Set Buffer Address and address characters
 - * GS blank = Start Field and attribute byte
 - * %027 = EBCDIC IL character. This character is not normally used for a 3270
 - * ETX = End of Text
 - * %077.326 = Block Check Characters
- o PEDT-12: RECV TEXT ERROR. The Editor detected a transmission error. (Most likely a bit error when the IL character was received above!)
- o POPR-13: SEND CONTROL SEQ THEN RECEIVE RESPONSE. SEND SEQ=NAK. Here, the INP is indicating it will "NAK" the host transmission and await a response.
- o PSCT-14: The control sequence containing the NAK is sent to the host.

CS trace: supplement

Reading Trace File: Trace File Entries Example 2

```
PSCT 12      9203.844 PSCT REQUEST ID=%031424(!3314)
              125.062 062.062 062.020 141.377
              SYN SYN SYN SYN DLE /

PRTX 0       9204.004 PRTX REQUEST ID=%031424(!3314)
              067.062 067.062 141.141 100.100 055.377 377.377
              EOT SYN EOT SYN / / ENG
              377.377 377.377 377.377 377.377 377.377 377.377

              377.377 377.377 377.377 377.377

PEDT 1       9204.014 PEDT REQUEST ID=%031424(!3314)
              RECV SELECT
              000.002 377.377
              NUL STX

POPR 2       9204.014 POPR REQUEST ID=%031424(!3314)
              SEND CONTROL SEQ THEN RECEIVE RESPONSE
              SEND SEQ=ACK0 TIMEOUT= 20.000
              OUT BFR=% 033626 LENGTH=0
              IN BUFR=% 033626 LENGTH=2048

PSCT 3       9204.014 PSCT REQUEST ID=%031424(!3314)
              125.062 062.062 062.020 160.377
              SYN SYN SYN SYN DLE
```

Reading Trace File: Trace File Entries
Example 2

- o PSCT-12: The INP gives positive acknowledgement to the receipt of a previous data transmission.
* DLE / = bisync ACK1 in EBCDIC
- o PRTX-0: Receive text entry. Turns out host sends a select sequence.
* EOT SYN EOT SYN = leading EOT and SYN characters
* // blank blank ENQ = Select sequence, for control unit 1 device 0.
- o PEDT-1: Editor recognizes the PRTX data is a select sequence.
- o POPR-2: SEND CONTROL SEQ THEN RECEIVE RESPONSE SEND SEQ=ACK0
The INP will send an ACK0 to the host.
- o PSCT-3: Here we see the ACK0:
* DLE %160 = bisync ACK0 in EBCDIC

CS trace: supplement

Reading Trace File: Trace File Entries Example 2 (continued)

```
PRTX 4      9204.184 PRTX REQUEST ID=%031424(!3314)
              002.047 361.301 021.301 120.035 310.327 301.342
              STX ESC 1  A  DC1 A  &  GS  H  P  A  S
              342.346 326.331 304.100 325.326 343.100 301.344
              S  W  O  R  D  N  O  T  A  U
              343.310 326.331 311.351 305.304
              T  H  O  R  I  Z  E  D

PRTX 5      9204.244 PRTX REQUEST ID=%031424(!3314)
              100.306 326.331 100.344 342.305 331.311 304.035
              F  O  R  U  S  E  R  I  D  GS
              100.021 302.140 035.310 331.305 305.325 343.305
              DC1 B  -  GS  H  R  E  E  N  T  E
              331.100 140.035 100.021 303.360
              R  -  GS  DC1 C  0

PRTX 6      9204.294 PRTX REQUEST ID=%031424(!3314)
              035.310 035.114 023.003 060.332 377.377 377.377
              GS  H  GS  <  DC3 ETX
              377.377 377.377 377.377 377.377 377.377 377.377

              377.377 377.377 377.377 377.377
```

Reading Trace File: Trace File Entries
Example 2 (continued)

- o PRTX-4,5,6: Text and control information received from host:
 - * STX = Start of Text
 - * ESC l = 3270 Write command
 - * A = write control character
 - * DC1 A& = Set Buffer Address, address characters
 - * GS H = Start Field, attribute byte
 - * PASSWORD NOT AUTHORIZED FOR USERID = text
 - * GS blank = start field, attribute byte
 - * DC1 B- = set buffer address, address characters
 - * GS H = start field, attribute byte
 - * REENTER - = text
 - * GS blank = start field, attribute byte
 - * DC1 C0 = set buffer address, address
 - * GS H = start field, attribute byte
 - * GS < = start field, attribute byte
 - * DC3 = insert cursor order
 - * ETX = end of text
 - * %060.332

CS trace: supplement

Reading Trace File: Trace File Entries Example 2 (continued)

```

PEDT 7      9204.294 PEDT REQUEST ID=%031424(!3314)
                RECV TEXT
                000.020 000.001 047.361 301.021 301.120 035.310
                NUL DLE NUL SOH ESC 1  A  DC1 A  &  GS  H
                327.301 342.342 346.326 331.304 100.325 326.343
                P  A  S  S  W  O  R  D          N  O  T
                100.301 344.343 310.326
                A  U  T  H  O

POPR 8      9204.424 POPR REQUEST ID=%031424(!3314)
                SEND CONTROL SEQ THEN RECEIVE RESPONSE
                SEND SEQ=ACK1      TIMEOUT=  20.000
                OUT BFR=%  033626 LENGTH=0
                IN  BFR=%  033626 LENGTH=2048

PSCT 9      9204.424 PSCT REQUEST ID=%031424(!3314)
                125.062 062.062 062.020 141.377
                SYN SYN SYN SYN DLE /
```

Reading Trace File: Trace File Entries
Example 2 (continued)

- o PEDT-7: Editor recognizes previous PRTX entries as text.
- o POPR-8: INP will send ACK1 and await response.
- o PSCT-9: The INP sends ACK1 (DLE /).


```

00001 00001000          $CONTROL SUBPROGRAM
00002 00002000          IDENTIFICATION DIVISION.
00003 00003000          PROGRAM-ID. PDBIML.
00004 00004000          AUTHOR. IML CUSTOMER TRAINING COURSE.
00005 00005000          DATE-WRITTEN. NOVEMBER 1, 1980.
00006 00006000          REMARKS. PDBIML IS A SUBROUTINE TO BE CALLED BY MAIN
00007 00007000          PROGRAM PDB3, AND IS PART OF THE ROSS CORPORATION
00008 00008000          PARTS DATABASE SYSTEM EXAMPLE IN THE IML/3000
00009 00009000          CUSTOMER TRAINING COURSE.
00010 00010000
00011 00011000          ENVIRONMENT DIVISION.
00012 00012000
00013 00013000          CONFIGURATION SECTION.
00014 00014000          SOURCE-COMPUTER. HP-3000.
00015 00015000          OBJECT-COMPUTER. HP-3000.
00016 00016000
00017 00017000          DATA DIVISION.
00018 00018000
00019 00019000          WORKING-STORAGE SECTION.
00020 00020000
00021 00021000          01 IML-VARIABLES.
00022 00022000             05 TERMINALID          PIC S9(4) COMP SYNC.
00023 00023000             05 RESULT              PIC S9(4) COMP SYNC.
00024 00024000                 88 RESULT-OK          VALUE 0.
00025 00025000                 88 RESULT568          VALUES 5, 6, 8.
00026 00026000             05 ERRORCODE              PIC S9(4) COMP SYNC.
00027 00027000                 88 TRACEABLE-ERRORCODE VALUES 10, 11, 12,
00028 00028000                                         13, 17, 21,
00029 00029000                                         23, 24, 38.
00030 00030000                 88 RECOVERABLE-ERRORCODE VALUES 9, 10, 11,
00031 00031000                                         12, 13.
00032 00032000             05 AID                    PIC S9(4) COMP SYNC.
00033 00033000             05 CURSORROW              PIC S9(4) COMP SYNC.
00034 00034000             05 CURSORCOLUMN          PIC S9(4) COMP SYNC.
00035 00035000             05 MAXINBUFLEN           PIC S9(4) COMP SYNC.
00036 00036000             05 ACTINBUFLEN           PIC S9(4) COMP SYNC.
00037 00037000             05 MSGLEN                PIC S9(4) COMP SYNC.
00038 00038000             05 OUTBUFLEN            PIC S9(4) COMP SYNC.
00039 00039000             05 DEVICENUM             PIC S9(4) COMP SYNC.
00040 00040000             05 FIELDNUM         PIC S9(4) COMP SYNC.
00041 00041000             05 OFFSET               PIC S9(4) COMP SYNC.
00042 00042000             05 DEVTYPE              PIC S9(4) COMP SYNC.
00043 00043000             05 FFINDEX              PIC S9(4) COMP SYNC.
00044 00044000             05 BUFSIZE              PIC S9(4) COMP SYNC.
00045 00045000             05 TIMEOUT.
00046 00046000                 10 RECEIVE-DATA-TIMEOUT PIC S9(4) COMP SYNC
00047 00047000                                         VALUE 180.
00048 00048000                 10 KBUNLOCK-TIMEOUT  PIC S9(4) COMP SYNC
00049 00049000                                         VALUE 120.
00050 00050000             05 FLAGS               PIC S9(4) COMP SYNC VALUE 0.
00051 00051000             05 CONFILE              PIC X(27)
00052 00052000                                         VALUE "IMLCONF.PUB.SYS ".
00053 00053000             05 OUTBUF              PIC X(80).
00054 00054000             05 INBUF               PIC X(80).
00055 00055000             05 MSGBUF              PIC X(144).

```

00056	00057000	01	DUMPSCREEN-VARIABLES.		
00057	00058000	05	LOCATION	PIC S9(4) COMP SYNC.	
00058	00059000	05	WORK.		
00059	00060000	10	FILLER	PIC S9(4) COMP SYNC	
00060	00061000			OCCURS 16 TIMES.	
00061	00062000				
00062	00063000	01	LOCAL-VARIABLES.		
00063	00064000	05	INVALID-ACTION-CODE	PIC 9999 VALUE 1.	
00064	00065000	05	OPEN-FAIL-CODE	PIC 9999 VALUE 2.	
00065	00066000	05	FATAL-ERROR-CODE	PIC 9999 VALUE 3.	
00066	00067000	05	ENTER-AID	PIC S9(4) COMP SYNC VALUE 39.	
00067	00068000	05	PF1-AID	PIC S9(4) COMP SYNC VALUE 49.	
00068	00069000	05	PF2-AID	PIC S9(4) COMP SYNC VALUE 50.	
00069	00070000	05	PF3-AID	PIC S9(4) COMP SYNC VALUE 51.	
00070	00071000	05	PF12-AID	PIC S9(4) COMP SYNC VALUE 52.	
00071	00072000	05	MIN-DEVICENUM	PIC S9(4) COMP SYNC VALUE 8.	
00072	00073000	05	MAX-DEVICENUM	PIC S9(4) COMP SYNC VALUE 12.	
00073	00074000	05	RETRY-FLAG	PIC X VALUE "F".	
00074	00075000	05	COMPARE-FIELD	PIC X(4).	
00075	00076000				
00076	00077000		LINKAGE SECTION.		
00077	00078000	01	PDBIML-LINK-RECORD.		
00078	00079000	05	ACTION-CODE	PIC X.	
00079	00080000	88	INITIALIZE-ACTION		VALUE "I".
00080	00081000	88	REPORT-ACTION		VALUE "R".
00081	00082000	88	ADD-ACTION		VALUE "A".
00082	00083000	88	WITHDRAW-ACTION		VALUE "W".
00083	00084000	88	TERMINATE-ACTION		VALUE "T".
00084	00085000	05	SCREENTRACE-SWITCH	PIC X.	
00085	00086000	88	SCREENTRACE-ON		VALUE "T".
00086	00087000	05	MAIN-RETURN-CODE	PIC 9999.	
00087	00088000				
00088	00089000	01	PART-RECORD.		
00089	00090000	05	PART-NUMBER	PIC 9(6) USAGE DISPLAY.	
00090	00091000	05	PART-QUANTITY	PIC 9(4) USAGE DISPLAY.	

```

00091 00092000 / * * * * *
00092 00093000 *
00093 00094000 *           P R O C E D U R E   D I V I S I O N
00094 00095000 *
00095 00096000 * * * * *
00096 00097000 * PROCEDURE DIVISION
00097 00098000 *   USING PDBIML-LINK-RECORD, PART-RECORD.
00098 00099000 *
00099 00100000 * PERFORM-ACTION-ROUTINE.
00100 00101000 *
00101 00102000 * STEP 1.  INITIALIZE MAIN RETURN CODE.
00102 00103000 *
00103 00104000 *       MOVE 0 TO MAIN-RETURN-CODE.
00104 00105000 *
00105 00106000 * STEP 2.  BRANCH TABLE.  PERFORM THE REQUESTED ACTION.
00106 00107000 *
00107 00108000 *       IF INITIALIZE-ACTION      THEN PERFORM INITIALIZE-ROUTINE
00108 00109000 *
00109 00110000 *       ELSE
00110 00111000 *
00111 00112000 *       IF REPORT-ACTION          THEN PERFORM REPORT-ROUTINE
00112 00113000 *
00113 00114000 *       ELSE
00114 00115000 *
00115 00116000 *       IF ADD-ACTION             THEN PERFORM ADD-ROUTINE
00116 00117000 *
00117 00118000 *       ELSE
00118 00119000 *
00119 00120000 *       IF WITHDRAW-ACTION       THEN PERFORM WITHDRAW-ROUTINE
00120 00121000 *
00121 00122000 *       ELSE
00122 00123000 *
00123 00124000 *       IF TERMINATE-ACTION      THEN PERFORM TERMINATE-ROUTINE
00124 00125000 *
00125 00126000 *       ELSE
00126 00127000 *
00127 00128000 *       MOVE INVALID-ACTION-CODE TO MAIN-RETURN-CODE.
00128 00129000 *
00129 00130000 * STEP 3.  WE HAVE SUCCESSFULLY PERFORMED THE REQUESTED
00130 00131000 * ACTION.  IF THIS WAS A RETRY FOR A RECOVERABLE
00131 00132000 * ERROR, WE NEED TO RESET THE RETRY FLAGS AND
00132 00133000 * PUT OUT RETRY SUCCESSFUL MSG.
00133 00134000 *
00134 00135000 *       IF RETRY-FLAG = "T"
00135 00136000 *           THEN
00136 00137000 *               DISPLAY "*** PDBIML RECOVERABLE ERROR RETRY ",
00137 00138000 *                   "SUCCESSFUL" UPON CONSOLE
00138 00139000 *                   MOVE "F" TO RETRY-FLAG.
00139 00140000 *
00140 00141000 * PROGRAM-EXIT.
00141 00142000 *       EXIT PROGRAM.

```

```

00142 00143000 / * * * * *
00143 00144000 *
00144 00145000 *           I N I T I A L I Z E - R O U T I N E           *
00145 00146000 *
00146 00147000 * * * * *
00147 00148000 *   INITIALIZE-ROUTINE.
00148 00149000 *
00149 00150000 *   OPEN IML TERMINAL, LOGON TO HOST, SELECT PDBS APPLICATION
00150 00151000 *   ON HOST, GET TO PART STOCK UPDATE MENU.
00151 00152000 *
00152 00153000 *   STEP 1.  OPEN THE SCREENTRACE FILE IF NECESSARY, THEN
00153 00154000 *   OPEN THE IML TERMINAL.
00154 00155000 *
00155 00156000 *   MOVE 101 TO LOCATION.
00156 00157000 *   IF SCREENTRACE-ON THEN PERFORM DUMPSCREENOPEN-ROUTINE.
00157 00158000 *   PERFORM OPEN3270-ROUTINE.
00158 00159000 *
00159 00160000 *   STEP 2.  RECEIVE IML BANNER SCREEN.
00160 00161000 *
00161 00162000 *   MOVE 102 TO LOCATION.
00162 00163000 *   PERFORM IML-RECV3270.
00163 00164000 *
00164 00165000 *   STEP 3.  RECEIVE FIRST SCREEN (SENT AUTOMATICALLY BY
00165 00166000 *   HOST WHEN TERMINAL IS "POWERED ON").
00166 00167000 *
00167 00168000 *   MOVE 103 TO LOCATION.
00168 00169000 *   PERFORM IML-RECV3270.
00169 00170000 *
00170 00171000 *   STEP 4.  THIS MAY OR MAY NOT BE THE LOGON SCREEN, WE
00171 00172000 *   ARE NOT SURE AT THIS POINT.  WE NEED TO FORCE
00172 00173000 *   OUR WAY TO THE LOGON SCREEN, LOGON, AND SELECT
00173 00174000 *   THE PDBS APPLICATION ON THE HOST.  THIS WILL
00174 00175000 *   PUT US IN HOST SCREEN #200, FROM WHICH WE BEGIN
00175 00176000 *   ALL OF THE OTHER ACTIONS.
00176 00177000 *
00177 00178000 *   MOVE 104 TO LOCATION.
00178 00179000 *   PERFORM GET-TO-SCREEN-200-ROUTINE.

```

```
00179 00180000 / * * * * *
00180 00181000 *
00181 00182000 *           R E P O R T - R O U T I N E           *
00182 00183000 *
00183 00184000 * * * * *
00184 00185000 REPORT-ROUTINE.
00185 00186000
00186 00187000 * REPORT QUANTITY ON HAND FOR A GIVEN PART NUMBER.
00187 00188000 *
00188 00189000 * STEP 1.  FILL OUT PART STOCK UPDATE SCREEN W/ PART NUMBER,
00189 00190000 *           AND ASK FOR "REPORT QUANTITY ON HAND" BY
00190 00191000 *           ENTERING PF1.
00191 00192000
00192 00193000         MOVE          201  TO  LOCATION.
00193 00194000         MOVE           1  TO  FIELDNUM.
00194 00195000         MOVE           0  TO  OFFSET.
00195 00196000         MOVE PART-NUMBER TO  OUTBUF.
00196 00197000         MOVE           6  TO  OUTBUFLN.
00197 00198000         PERFORM IML-WRITEFIELD.
00198 00199000         MOVE PF1-AID TO AID.
00199 00200000         PERFORM IML-TRAN3270.
00200 00201000         PERFORM IML-RECV3270.
00201 00202000
00202 00203000 * STEP 2.  SHOULD BE IN HOST SCREEN #300.  READ QUANTITY,
00203 00204000 *           THEN RETURN TO PART STOCK UPDATE SCREEN W/ ENTER.
00204 00205000
00205 00206000         MOVE 202 TO LOCATION.
00206 00207000         MOVE   3 TO FIELDNUM.
00207 00208000         MOVE   0 TO OFFSET.
00208 00209000         MOVE   4 TO MAXINBUFLN.
00209 00210000         PERFORM IML-READFIELD.
00210 00211000         MOVE INBUF TO PART-QUANTITY.
00211 00212000         MOVE ENTER-AID TO AID.
00212 00213000         PERFORM IML-TRAN3270.
00213 00214000         PERFORM IML-RECV3270.
```

```

00214 00215000 / * * * * *
00215 00216000 *
00216 00217000 *           A D D - R O U T I N E           *
00217 00218000 *
00218 00219000 * * * * *
00219 00220000 ADD-ROUTINE.
00220 00221000
00221 00222000 * ADD QUANTITY FROM PART-RECORD TO QUANTITY ON HAND IN
00222 00223000 * HOST SCREEN.
00223 00224000 *
00224 00225000 * STEP 1.  FILL OUT PART STOCK UPDATE SCREEN WITH PART NUMBER
00225 00226000 * AND QUANTITY, AND ENTER A PF2 TO GET TO ADD SCREEN
00226 00227000 * (HOST SCREEN #400).
00227 00228000
00228 00229000 MOVE          301 TO LOCATION.
00229 00230000 MOVE           1 TO FIELDNUM.
00230 00231000 MOVE           0 TO OFFSET.
00231 00232000 MOVE PART-NUMBER TO OUTBUF.
00232 00233000 MOVE           6 TO OUTBUFLEN.
00233 00234000 PERFORM IML-WRITEFIELD.
00234 00235000
00235 00236000 MOVE          3 TO FIELDNUM.
00236 00237000 MOVE           0 TO OFFSET.
00237 00238000 MOVE PART-QUANTITY TO OUTBUF.
00238 00239000 MOVE           4 TO OUTBUFLEN.
00239 00240000 PERFORM IML-WRITEFIELD.
00240 00241000
00241 00242000 MOVE PF2-AID TO AID.
00242 00243000 PERFORM IML-TRAN3270.
00243 00244000 PERFORM IML-PECV3270.
00244 00245000
00245 00246000 * STEP 2.  READ BACK NEW STOCK ON HAND QUANTITY, AND RETURN
00246 00247000 * TO PART STOCK UPDATE SCREEN WITH "ENTER".
00247 00248000
00248 00249000 MOVE 302 TO LOCATION.
00249 00250000 MOVE 7 TO FIELDNUM.
00250 00251000 MOVE 0 TO OFFSET.
00251 00252000 MOVE 4 TO MAXINBUFLEN.
00252 00253000 PERFORM IML-READFIELD.
00253 00254000 MOVE INBUF TO PART-QUANTITY.
00254 00255000
00255 00256000 MOVE ENTER-AID TO AID.
00256 00257000 PERFORM IML-TRAN3270.
00257 00258000 PERFORM IML-RECV3270.

```

```

00258 00259000 / * * * * *
00259 00260000 *
00260 00261000 *           W I T H D R A W - R O U T I N E
00261 00262000 *
00262 00263000 * * * * *
00263 00264000 * WITHDRAW-ROUTINE.
00264 00265000
00265 00266000 * WITHDRAW PART-QUANTITY IN PART-RECORD FROM QUANTITY ON HAND
00266 00267000 * IN HOST SCREEN.
00267 00268000 *
00268 00269000 * STEP 1.  FILL OUT PART STOCK UPDATE SCREEN WITH PART NUMBER
00269 00270000 * AND QUANTITY, AND ENTER A PF3 TO GET TO WITHDRAW
00270 00271000 * SCREEN (HOST SCREEN #500).
00271 00272000
00272 00273000     MOVE          401 TO LOCATION.
00273 00274000     MOVE           1 TO FIELDNUM.
00274 00275000     MOVE           0 TO OFFSET.
00275 00276000     MOVE PART-NUMBER TO OUTBUF.
00276 00277000     MOVE           6 TO OUTBUFLN.
00277 00278000     PERFORM IML-WRITEFIELD.
00278 00279000
00279 00280000     MOVE           3 TO FIELDNUM.
00280 00281000     MOVE           0 TO OFFSET.
00281 00282000     MOVE PART-QUANTITY TO OUTBUF.
00282 00283000     MOVE           4 TO OUTBUFLN.
00283 00284000     PERFORM IML-WRITEFIELD.
00284 00285000
00285 00286000     MOVE PF3-AID TO AID.
00286 00287000     PERFORM IML-TRAN3270.
00287 00288000     PERFORM IML-RECV3270.
00288 00289000
00289 00290000 * STEP 2.  READ BACK NEW STOCK ON HAND QUANTITY, AND RETURN
00290 00291000 * TO PART STOCK UPDATE SCREEN WITH "ENTER".
00291 00292000
00292 00293000     MOVE  402 TO LOCATION.
00293 00294000     MOVE   7 TO FIELDNUM.
00294 00295000     MOVE   0 TO OFFSET.
00295 00296000     MOVE   4 TO MAXINBUFLN.
00296 00297000     PERFORM IML-READFIELD.
00297 00298000     MOVE INBUF TO PART-QUANTITY.
00298 00299000
00299 00300000     MOVE ENTER-AID TO AID.
00300 00301000     PERFORM IML-TRAN3270.
00301 00302000     PERFORM IML-RECV3270.

```



```

00302 00303000 / * * * * *
00303 00304000 *
00304 00305000 *           T E R M I N A T E - R O U T I N E           *
00305 00306000 *
00306 00307000 * * * * *
00307 00308000 *   T E R M I N A T E - R O U T I N E .
00308 00309000
00309 00310000 *   * LOGOFF THE HOST, CLOSE THE IML TERMINAL ("POWER IT OFF").
00310 00311000 *
00311 00312000 *   * STEP 1. LOGOFF THE HOST. HOST PROGRAM IS CODED SO THAT
00312 00313000 *           IF WE KEEP ENTERING PF12, SOONER OR LATER WE WILL
00313 00314000 *           BE BACK TO SCREEN #001, THE LOGON SCREEN. THIS,
00314 00315000 *           OF COURSE, MEANS WE HAVE SUCCESSFULLY LOGGED OFF.
00315 00316000
00316 00317000 *           MOVE 501 TO LOCATION.
00317 00318000 *           PERFORM LOGOFF-HOST-ROUTINE.
00318 00319000
00319 00320000 *   * STEP 2. CLOSE THE IML TERMINAL.
00320 00321000
00321 00322000 *           MOVE 502 TO LOCATION.
00322 00323000 *           PERFORM IML-CLOSE3270.
00323 00324000
00324 00325000 *   * STEP 3. CLOSE IF NECESSARY THE SCREEN TRACE FILE.
00325 00326000
00326 00327000 *           IF SCREENTRACE-ON
00327 00328000 *           THEN PERFORM DUMPSCREENCLOSE-ROUTINE.

```

```

00328 00329000 / * * * * *
00329 00330000 *
00330 00331000 *           I M L E R R - R O U T I N E           *
00331 00332000 *
00332 00333000 * * * * *
00333 00334000 * IMLERR-ROUTINE.
00334 00335000
00335 00336000 * DISPLAY ERROR MSG ON THE CONSOLE AND PRINT SCREENTRACE IF
00336 00337000 * POSSIBLE. ALSO IF POSSIBLE TRY TO RECOVER FROM THE ERROR
00337 00338000 * BY RETRYING THE ACTION-CODE.
00338 00339000 *
00339 00340000 * STEP 1. SAVE THE RESULT CODE.
00340 00341000
00341 00342000         MOVE RESULT TO ERRORCODE.
00342 00343000
00343 00344000 * STEP 2. PUT OUT FRROR MESSAGE. CAN'T WRITE ERROR MESSAGE
00344 00345000 * TO $STDLIST BECAUSE $STDLIST IS TERMINAL, WHICH IS
00345 00346000 * UNDER CONTROL OF V/3000. BESIDES, TERMINAL USER
00346 00347000 * PROBABLY WOULDN'T KNOW WHAT TO DO ANYWAY.
00347 00348000
00348 00349000         DISPLAY "*** PDBIML IML ERROR DETECTED, RESULT CODE=",
00349 00350000         ERRORCODE UPON CONSOLE.
00350 00351000         PERFORM IML-ERP3270.
00351 00352000         IF RESULT NOT EQUAL 0
00352 00353000             THEN
00353 00354000                 DISPLAY "*** ERR3270 ERROR DETECTED, UNABLE TO ",
00354 00355000                 "RETRIEVE ERROP MSG" UPON CONSOLE
00355 00356000                 DISPLAY "*** ERR3270 RESULT CODE=", RESULT,
00356 00357000                 UPON CONSOLE
00357 00358000             ELSE
00358 00359000                 DISPLAY MSGBUF UPON CONSOLE.
00359 00360000
00360 00361000 * STEP 3. FOR CERTAIN ERROR CODES, IT WILL BE IMPOSSIBLE TO
00361 00362000 * GET A SCREENTRACE. FOR THESE CASES, PUT OUT
00362 00363000 * AN ERROR MESSAGE...
00363 00364000
00364 00365000         IF NOT TRACEABLE-ERRORCODE
00365 00366000             THEN
00366 00367000                 DISPLAY "*** SCREENTRACE NOT ATTEMPTED"
00367 00368000                 UPON CONSOLE
00368 00369000
00369 00370000 * STEP 4. ... OTHERWISE TRY TO GET A SCREENTRACE. IF WE'RE NO
00370 00371000 * ALREADY SCREENTRACING (SCREENTRACE-ON IS TRUE) THEN
00371 00372000 * WE HAVE TO OPEN THE SCREENTRACE FILE.
00372 00373000 * USE THE NEGATIVE OF "LOCATION" SO WE CAN
00373 00374000 * DISTINGUISH THIS SCREENTRACE FROM A NON-ERROR
00374 00375000 * GENERATED SCREENTRACE.
00375 00376000
00376 00377000         ELSE
00377 00378000             IF NOT SCREENTRACE-ON
00378 00379000                 THEN PERFORM DUMPSCREENOPEN-ROUTINE
00379 00380000             COMPUTE LOCATION = - LOCATION
00380 00381000             PERFORM DUMPSCREEN-ROUTINE
00381 00382000             IF NOT SCREENTRACE-ON
00382 00383000                 THEN PERFORM DUMPSCREENCLOSE-ROUTINE.

```

```
00383 00385000      * STEP 5.  IF THE ERROR IS A RECOVERABLE ONE AND WE ARE NOT
00384 00386000      *
00385 00387000      * ALREADY TRYING TO RECOVER FROM AN ERROR, TRY
00386 00388000      * TO RECOVER BY FORCING THE HOST PROGRAM BACK TO
00387 00389000      * SCREEN #200 AND RE-TRYING THE ORIGINAL ACTION AGAIN.
00388 00390000      IF RECOVERABLE-ERRORCODE AND RETRY-FLAG NOT EQUAL "T"
00389 00391000      THEN
00390 00392000          DISPLAY "*** RETRY WILL BE ATTEMPTED FOR ACTION ",
00391 00393000          ACTION-CODE UPON CONSOLE
00392 00394000          MOVE "T" TO RETRY-FLAG
00393 00395000          PERFORM GET-TO-SCREEN-200-ROUTINE
00394 00396000          DISPLAY "*** RETRY ATTEMPT PROCEEDING" UPON CONSOLE
00395 00397000          GO TO PERFORM-ACTION-ROUTINE.
00396 00398000
00397 00399000      * STEP 6.  THE ERROR IS NON-RECOVERABLE.
00398 00400000
00399 00401000          MOVE FATAL-ERROR-CODE TO MAIN-RETURN-CODE.
00400 00402000          GO TO PROGRAM-EXIT.
```

```

00401 00403000 / * * * * *
00402 00404000 *
00403 00405000 *   G E T - T O - S C R E E N - 2 0 0 - R O U T I N E   *
00404 00406000 *
00405 00407000 * * * * *
00406 00408000 * FOR SOME REASON WE AREN'T SURE WHAT HOST SCREEN WE'RE IN,
00407 00409000 * FORCE LOGOFF THE HOST SYSTEM, LOGON, AND SELECT PDDBS
00408 00410000 * APPLICATION. WE SHOULD FINISH THIS ROUTINE IN HOST SCREEN
00409 00411000 * #200.
00410 00412000
00411 00413000   GET-TO-SCREEN-200-ROUTINE.
00412 00414000
00413 00415000 * STEP 1.  FORCE LOGOFF THE HOST SYSTEM.
00414 00416000   PERFORM LOGOFF-HOST-ROUTINE.
00415 00417000
00416 00418000
00417 00419000 * STEP 2.  WE ARE IN SCREEN #001.  LOGON BY "TYPING" IN A
00418 00420000 *   USERID AND PASSWORD, AND "PRESSING ENTER".
00419 00421000
00420 00422000   MOVE           1   TO FIELDNUM.
00421 00423000   MOVE           0   TO OFFSET.
00422 00424000   MOVE "JOHN DOE" TO OUTBUF.
00423 00425000   MOVE           8   TO OUTBUFLN.
00424 00426000   PERFORM IML-WRITEFIELD.
00425 00427000
00426 00428000   MOVE           3   TO FIELDNUM.
00427 00429000   MOVE           0   TO OFFSET.
00428 00430000   MOVE "PASSWORD" TO OUTBUF.
00429 00431000   MOVE           8   TO OUTBUFLN.
00430 00432000   PERFORM IML-WRITEFIELD.
00431 00433000
00432 00434000   MOVE ENTER-AID TO AID.
00433 00435000   PERFORM IML-TRAN3270.
00434 00436000   PERFORM IML-RECV3270.
00435 00437000
00436 00438000 * STEP 3.  SHOULD BE IN HOST SCREEN #100 - SELECT PROGRAM.
00437 00439000 *   SELECT "PART STOCK UPDATE" WITH PF1.
00438 00440000
00439 00441000   MOVE PF1-AID TO AID.
00440 00442000   PERFORM IML-TRAN3270.
00441 00443000   PERFORM IML-RECV3270.

```

```

00442 00444000 / * * * * *
00443 00445000 *
00444 00446000 *           L O G O F F - H O S T - R O U T I N E           *
00445 00447000 *
00446 00448000 * * * * *
00447 00449000 LOGOFF-HOST-ROUTINE.
00448 00450000
00449 00451000 * THE HOST APPLICATION PROGRAM IS CODED IN SUCH A FASHION
00450 00452000 * THAT IF YOU KEEP ENTERING PF12 ENOUGH TIMES IN A ROW, IT WILL
00451 00453000 * KEEP MOVING BACK THRU ITS MENU TREE STRUCTURE UNTIL IT
00452 00454000 * LOGS YOU OFF, AND LEAVES THE LOGON SCREEN ON THE TERMINAL.
00453 00455000 * THIS ROUTINE WILL BE USED BOTH TO TERMINATE PDBIML, AND ALSO
00454 00456000 * TO TRY TO GET BACK TO A RECOGNIZABLE SCREEN IN CASE OF A
00455 00457000 * RECOVERABLE ERROR.
00456 00458000
00457 00459000     MOVE PF12-AID TO AID.
00458 00460000     MOVE     906 TO OFFSET.
00459 00461000     MOVE     4 TO MAXINBUFLN.
00460 00462000     PERFORM IML-READSCREEN.
00461 00463000     MOVE     INBUF TO COMPARE-FIELD.
00462 00464000
00463 00465000     PERFORM PF12-ROUTINE UNTIL COMPARE-FIELD = "NAME".
00464 00466000
00465 00467000 PF12-ROUTINE.
00466 00468000     PERFORM IML-TRAN3270.
00467 00469000     PERFORM IML-RECV3270.
00468 00470000     PERFORM IML-READSCREEN.
00469 00471000     MOVE     INBUF TO COMPARE-FIELD.

```

```
00470 00472000 / * * * * *
00471 00473000 *
00472 00474000 *           D U M P S C R E E N O P E N - R O U T I N E
00473 00475000 *
00474 00476000 * * * * *
00475 00477000 * DUMPSCREENOPEN-ROUTINE.
00476 00478000 *   CALL "DUMPSCREENOPEN" USING WORK, RESULT.
00477 00479000 *   IF RESULT NOT EQUAL 0
00478 00480000 *       THEN
00479 00481000 *           DISPLAY "*** DUMPSCREENOPEN ERROR", RESULT,
00480 00482000 *               UPON CONSOLE
00481 00483000 *           MOVE FATAL-ERROR-CODE TO MAIN-RETURN-CODE
00482 00484000 *           GO TO PROGRAM-EXIT.
00483 00485000 * * * * *
00484 00486000 *
00485 00487000 *           D U M P S C R E E N - R O U T I N E
00486 00488000 *
00487 00489000 * * * * *
00488 00490000 * DUMPSCREEN-ROUTINE.
00489 00491000 *   CALL "DUMPSCREEN" USING WORK, TERMINALID, LOCATION,
00490 00492000 *               RESULT.
00491 00493000 *   IF RESULT NOT EQUAL 0
00492 00494000 *       THEN
00493 00495000 *           DISPLAY "*** DUMPSCREEN ERROR", RESULT,
00494 00496000 *               UPON CONSOLE
00495 00497000 *           MOVE FATAL-ERROR-CODE TO MAIN-RETURN-CODE
00496 00498000 *           GO TO PROGRAM-EXIT.
00497 00499000 * * * * *
00498 00500000 *
00499 00501000 *           D U M P S C R E E N C L O S E - R O U T I N E
00500 00502000 *
00501 00503000 * * * * *
00502 00504000 * DUMPSCREENCLOSE-ROUTINE.
00503 00505000 *   CALL "DUMPSCREENCLOSE" USING WORK, RESULT.
00504 00506000 * DON'T CARE ABOUT THE RESULT IN THIS CASE.
```

```

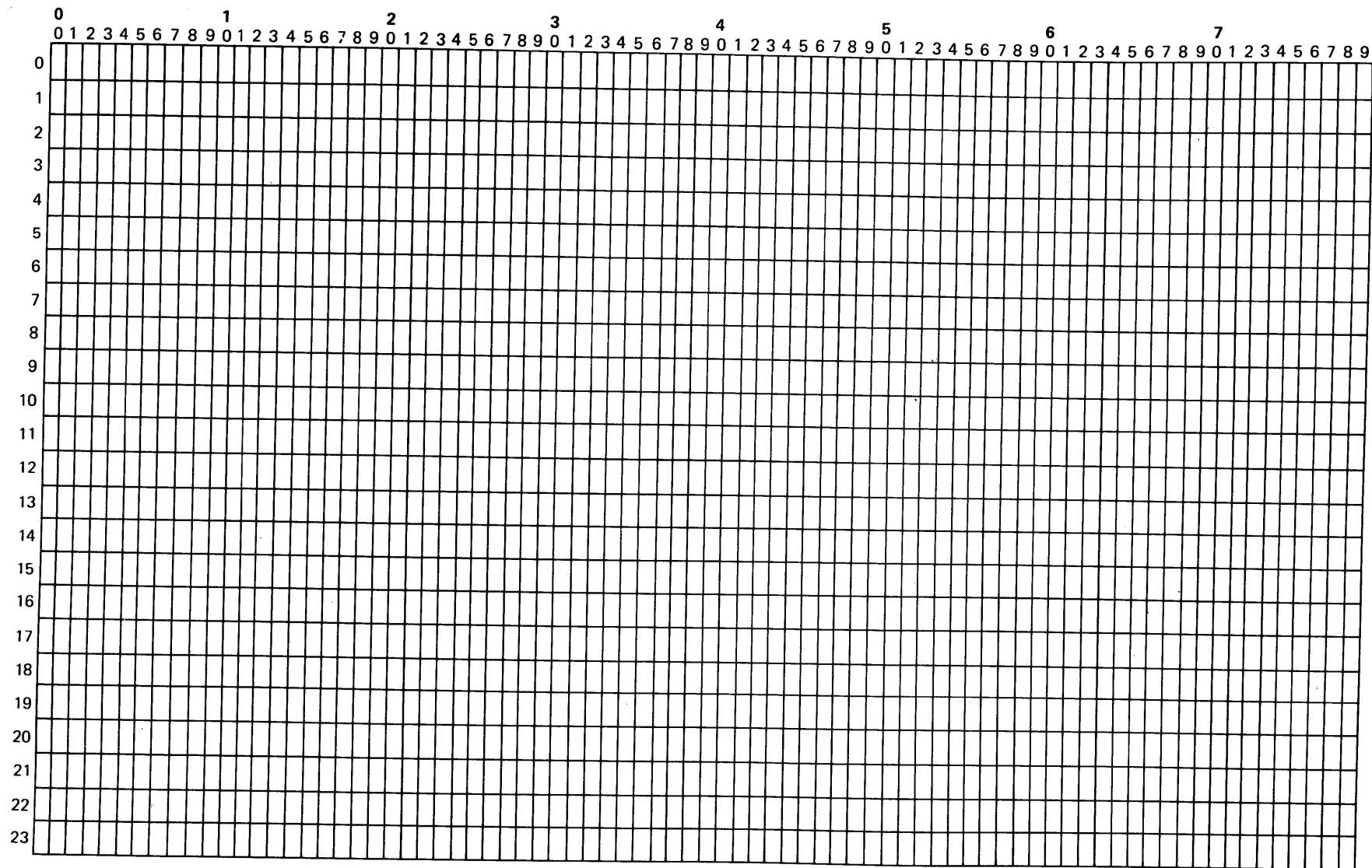
00505 00507000 / * * * * *
00506 00508000 *
00507 00509000 *           O P E N 3 2 7 0 - R O U T I N E           *
00508 00510000 *
00509 00511000 * * * * *
00510 00512000 OPEN3270-ROUTINE.
00511 00513000
00512 00514000 * KEEP TRYING TO OPEN EACH OF THE ALLOWED DEVICENUMS UNTIL ONE
00513 00515000 * SUCCEEDS. THE ALLOWED DEVICENUMS ARE EVERY DEVICENUM BETWEEN
00514 00516000 * MIN-DEVICENUM AND MAX-DEVICE, INCLUSIVE.
00515 00517000 *
00516 00518000 * STEP 1. PERFORM OPEN3270 CALL.
00517 00519000
00518 00520000     PERFORM IML-OPEN3270
00519 00521000     VARYING DEVICENUM FROM MIN-DEVICENUM BY 1
00520 00522000     UNTIL (DEVICENUM > MAX-DEVICENUM)
00521 00523000     OR (NOT RESULT568).
00522 00524000
00523 00525000 * STEP 2. CHECK IF NO AVAILABLE DEVICE NUMBERS
00524 00526000
00525 00527000     IF DEVICENUM > MAX-DEVICENUM
00526 00528000     THEN
00527 00529000         DISPLAY "*** PDBIML OPEN3270 ERROR -",
00528 00530000         " NO AVAILABLE IML DEVICES" UPON CONSOLE
00529 00531000         MOVE OPEN-FAIL-CODE TO MAIN-RETURN-CODE
00530 00532000         GO TO PROGRAM-EXIT.
00531 00533000
00532 00534000 * STEP 3. CHECK IF RESULT IS OKAY (=0).
00533 00535000
00534 00536000     IF NOT RESULT-OK
00535 00537000     THEN
00536 00538000         DISPLAY "*** PDBIML OPEN3270 ERROR ",
00537 00539000         RESULT UPON CONSOLE
00538 00540000         MOVE RESULT TO ERRORCODE
00539 00541000         PERFORM IML-ERR3270
00540 00542000         MOVE OPEN-FAIL-CODE TO MAIN-RETURN-CODE
00541 00543000         GO TO PROGRAM-EXIT.
00542 00544000
00543 00545000 * STEP 4. ONLY POSSIBILITY LEFT IS RESULT WAS OKAY.
00544 00546000 * NOTHING LEFT TO DO, SO END PARAGRAPH AND RETURN TO
00545 00547000 * CALLER.

```

```

00546 00548000 / * * * * *
00547 00549000 *
00548 00550000 *           I M L   I N T R I N S I C   C A L L S           *
00549 00551000 *
00550 00552000 * * * * *
00551 00553000 IML-OPEN3270.
00552 00554000     CALL "COPEN3270" USING DEVICENUM, CONFILE, FLAGS,
00553 00555000     TERMINALID, DEVTYPE, FFINDEX, BUFSIZE,
00554 00556000     TIMEOUT, RESULT.
00555 00557000
00556 00558000 IML-CLOSE3270.
00557 00559000     CALL "CCLOSE3270" USING TERMINALID, RESULT.
00558 00560000
00559 00561000 IML-ERR3270.
00560 00562000     CALL "CERR3270" USING ERRORCODE, MSGBUF, MSGLEN, RESULT.
00561 00571000
00562 00572000 IML-READFIELD.
00563 00573000     CALL "CREADFIELD" USING TERMINALID, FIELDNUM, OFFSET,
00564 00574000     MAXINBUFLEN, INBUF, ACTINBUFLEN, RESULT.
00565 00575000     IF NOT RESULT-OK THEN PERFORM IMLERR-ROUTINE.
00566 00576000
00567 00577000 IML-READSCREEN.
00568 00578000     CALL "CREADSCREEN" USING TERMINALID, OFFSET, MAXINBUFLEN,
00569 00579000     INBUF, ACTINBUFLEN, RESULT.
00570 00580000     IF NOT RESULT-OK THEN PERFORM IMLERR-ROUTINE.
00571 00581000
00572 00582000 IML-RECV3270.
00573 00583000     CALL "CRECV3270" USING TERMINALID, RESULT.
00574 00584000     IF NOT RESULT-OK THEN PERFORM IMLERR-ROUTINE.
00575 00585000     IF SCREENTRACE-ON THEN PERFORM DUMPSCREEN-ROUTINE.
00576 00586000
00577 00587000 IML-TRAN3270.
00578 00588000     IF SCREENTRACE-ON THEN PERFORM DUMPSCREEN-ROUTINE.
00579 00589000     CALL "CTRAN3270" USING TERMINALID, AID, CURSORROW,
00580 00590000     CURSORCOLUMN, RESULT.
00581 00591000     IF NOT RESULT-OK THEN PERFORM IMLERR-ROUTINE.
00582 00592000
00583 00593000 IML-WRITEFIELD.
00584 00594000     CALL "CWRITEFIELD" USING TERMINALID, FIELDNUM, OFFSET,
00585 00595000     OUTBUF, OUTBUFLEN, RESULT.
00586 00596000     IF NOT RESULT-OK THEN PERFORM IMLERR-ROUTINE.

```

IML/3000: HOST APPLICATION QUESTIONNAIRE

1. HP 3000 APPLICATION: _____

2. HOST APPLICATION: _____

3. IML CONFIGURATION FILENAME: _____

4. 3270 DEVICE NUMBERS THAT MAY BE USED FOR THIS APPLICATION:

DEVICE NUM:
DEVTYPE :
BUFSIZE :

5. HOST SYSTEM LOGON USERID: _____

PASSWORD: _____

6. Does Host system automatically transmit a logon screen when terminal is powered on (i.e. after OPEN3270 call)?
YES NO

7. Longest acceptable Host response time before assuming a Host or line failure? (Use this value for the receive data timeout in the OPEN3270 call. Use a value of 0 to disable this timeout.)
_____ seconds.

8. For this application, does the Host system normally unlock the keyboard when ready to accept data?
YES NO

If the answer is NO, skip to item 11 and enter a value of 0.

9. If the answer to question 8 is YES, does the Host always send all of its data for a screen and unlock the keyboard within a single select sequence? (Ask your communications expert about this. If you are not sure, answer NO.)
YES NO

If the answer is YES, skip to item 11 and enter a value of 0.

10. If the answer to questions 8 and 9 are (in order) YES and NO, what is the longest acceptable time between receiving the first select sequence and receiving the keyboard unlock? Enter this value in item 11.

(over)

11. Keyboard unlock timeout for OPEN3270 call:

_____ seconds.

12. When a terminal is powered on (when an IML program calls OPEN3270), does the host system automatically logoff any old leftover previous sessions on that terminal?

YES NO

13. If the answer to item 12 is NO, consider how your IML application can distinguish whether or not you are in the host system logon screen, and if not, how your program can force the host system to get you there.

