EXTENDED iRMX® II.3 OPERATING SYSTEM DOCUMENTATION

VOLUME 5 INTERACTIVE CONFIGURATION UTILITY REFERENCE

Order Number: 461848-001

Intel Corporation
3065 Bowers Avenue
Santa Clara, California 95051

In locations outside the United States, obtain additional copies of Intel documentation by contacting your local Intel sales office. For your convenience, international sales office addresses are located directly before the reader reply card in the back of the manual.

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

Intel Corporation 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. Intel Corporation assumes no responsibility for any errors that may appear in this document. Intel Corporation makes no commitment to update or to keep current the information contained in this document.

Intel Corporation assumes no responsibility for the use of any circuitry other than circuitry embodied in an Intel product. No other circuit patent licenses are implied.

Intel software products are copyrighted by and shall remain the property of Intel Corporation. Use, duplication or disclosure is subject to restrictions stated in Intel's software license, or as defined in ASPR 7-104.9 (a) (9).

No part of this document may be copied or reproduced in any form or by any means without prior written consent of Intel Corporation.

The following are trademarks of Intel Corporation and its affiliates and may be used only to identify Intel products:

Above
BITBUS
COMMputer
CREDIT
Data Pipeline
Genius
△i
i
I^2ICE
ICE
iCEL
iCS
iDBP
iDIS

iLBX
i_m
iMDDX
iMMX
Insite
int$_e$l
int$_e$lBOS
Intelevision
int$_e$ligent Identifier
int$_e$ligent Programming
Intellec
Intellink
iOSP
iPDS
iPSB

iPSC
iRMX
iSBC
iSBX
iSDM
iSSB
iSXM
Library Manager
MCS
Megachassis
MICROMAINFRAME
MULTIBUS
MULTICHANNEL
MULTIMODULE

OpenNET
ONCE
Plug-A-Bubble
PROMPT
Promware
QUEST
QueX
Ripplemode
RMX/80
RUPI
Seamless
SLD
UPI
VLSiCEL

XENIX, MS-DOS, Multiplan, and Microsoft are trademarks of Microsoft Corporation. UNIX is a trademark of Bell Laboratories. Ethernet is a trademark of Xerox Corporation. Centronics is a trademark of Centronics Data Computer Corporation. Chassis Trak is a trademark of General Devices Company, Inc. VAX and VMS are trademarks of Digital Equipment Corporation. Smartmodem 1200 and Hayes are trademarks of Hayes Microcomputer Products, Inc. IBM is a registered trademark of International Business Machines. Soft-Scope is a registered trademark of Concurrent Sciences.

VOLUME PREFACE

MANUALS IN THIS VOLUME

This volume (Volume 5, *Extended iRMX® II Interactive Configuration Utility Reference*) contains the *Extended iRMX II Interactive Configuration Reference* manual.

EXTENDED iRMX® II INTERACTIVE CONFIGURATION UTILITY REFERENCE MANUAL

This manual describes all the configuration parameters for every feature the Interactive Configuration Utility (ICU) supports. For example, this manual explains all the configuration parameters needed to define the device drivers for Intel Single Board Computers (iSBC) supported by the Operating System. Each screen is displayed with the values found in the definition file 28612.def or 286100A.def supplied by Intel. In the case of device drivers that are not part of the Intel-supplied start-up system, the screens are displayed with the ICU default values.

RELATED PUBLICATIONS

- *iAPX 286 Utilities User's Guide for iRMX® 286 Systems*, Order Number: 122658
- *iAPX 286 System Builder User's Guide for iRMX® Systems,* Order Number: 122662
- *iRMX Networking Software User's Guide,* Order Number: 122323

VOLUME CONTENTS

Manuals are listed in the order they appear in the volumes. For a synopsis of each manual, refer to the *Introduction to the Extended iRMX® II Operating System*.

REV.	REVISION HISTORY	DATE
-001	Original Issue.	01/88

EXTENDED iRMX®II INTERACTIVE CONFIGURATION UTILITY REFERENCE MANUAL

Intel Corporation
3065 Bowers Avenue
Santa Clara, California 95051

PREFACE

INTRODUCTION

This manual describes the Interactive Configuration Utility (ICU) and explains its use. It does not explain each screen displayed by the ICU. For a description of the ICU screens and their parameters, refer to the *Extended iRMX II Interactive Configuration Utility Reference Manual.*

READER LEVEL

The manual assumes that you are familiar with the monitor and keyboard from which you run the ICU. It is also helpful if you are familiar with the following:

- The Extended iRMX II Operating System
- PL/M-286
- BND286 and BLD286

CONVENTIONS

The following conventions are used throughout this manual:

- Information appearing as UPPERCASE characters when shown in keyboard examples must be entered or coded exactly as shown. You may, however, mix lower and uppercase characters when entering the text.
- Fields appearing as lowercase characters within angle brackets (< >) when shown in keyboard examples indicate variable information. You must enter an appropriate value or symbol for variable fields.
- The term "iRMX II" refers to the Extended iRMX II.3 Operating System.
- The term "iRMX I" refers to the iRMX I (iRMX 86) Operating System.
- All numbers, unless otherwise stated, are assumed to be decimal. Hexadecimal numbers include the "H" radix character (for example, 0FFH).

CONTENTS

CHAPTER 1 PAGE
HARDWARE-RELATED PARAMETERS

CHAPTER 2 PAGE
MEMORY PARAMETERS

CHAPTER 3 PAGE
SUB-SYSTEM PARAMETERS

CHAPTER 4 PAGE
OS EXTENSION PARAMETERS

CHAPTER 5 PAGE
HUMAN INTERFACE PARAMETERS

CHAPTER 6 PAGE
APPLICATION LOADER PARAMETERS

CHAPTER 7 PAGE
REMOTE FILES ACCESS PARAMETERS

CHAPTER 8 PAGE
EXTENDED I/O SYSTEM PARAMETERS

CHAPTER 9 PAGE
BASIC I/O SYSTEM PARAMETERS

CHAPTER 10 PAGE
DEVICE DRIVER PARAMETERS

CHAPTER 10 (continued) **PAGE**

CHAPTER 10 (continued) **PAGE**

Terminal Communications Controller

iSBC® 286/10 Line Printer

Line Printer - iSBX™ 350

iSBC® 264 Driver

iSBX™ 251 Driver

RAM Driver

SCSI Driver

iSBC 186/224A

iSBC® 186/410

User Devices

CHAPTER 11 PAGE
SYSTEM DEBUGGER PARAMETERS

CHAPTER 12 PAGE
NUCLEUS PARAMETERS

CHAPTER 13 PAGE
USER JOB PARAMETERS

CHAPTER 14 PAGE
ROM CODE PARAMETERS

CHAPTER 15 PAGE
INCLUDES AND LIBRARIES PARAMETERS

CHAPTER 16 PAGE
GENERATE FILE NAMES PARAMETERS

CHAPTER 17 PAGE
BUILD FILE COMMENTS

TABLES

TABLES **PAGE**

FIGURES

FIGURES **PAGE**

CHAPTER 1
HARDWARE-RELATED PARAMETERS

1.1 INTRODUCTION

This chapter discusses how to respond to the prompts that appear on the hardware-related screens. The prompts are grouped into four categories: Hardware, MULTIBUS® II hardware, Interrupts and Slave Interrupts. To the right of each group of parameters listed below, is the starting page for that group. The remaining parameters follow in the same order as they appear in the list.

Hardware Parameters

(TP)	(CIL)	(CN)	(CIN)		Start on
(CF)	(TPS)	(NPX)	(IF)		page 1-2
(BIP)					

MULTIBUS® II Hardware

(MDP)	(MDS)	(MDL)	(MCO)	(MCT)	Start on
(DDP)	(BEA)	(CLM)	(CPM)	(PSP)	page 1-4
(AIB)	(ACI)	(ACO)	(ABW)		

Interrupt Parameters

(MP)	(MPS)	(IS)			Start on
					page 1-7

Slave Interrupt Levels

	Start on
	page 1-8

1.2 HARDWARE SCREEN

The first screen displayed when you enter "Change" mode without requesting a specific screen is the "Hardware" screen shown here. The default values displayed are values from the 28612.def definition file.

```
(HARD)          Hardware

(BUS) System Bus Type [ 1 = MBI / 2 = MBII]          1
(TP)  8254 Timer Port [0-OFFFFH]                     ODOH
(CIL) Clock Interrupt Level [0-7]                    0
(CN)  Timer Counter Number [0,1,2]                   0
(CIN) Clock Interval [0-65535 msec]                  10
(CF)  Clock Frequency [0-65535 kHz]                  1229
(TPS) Timer Port Separation [0-OFFH]                 02H
(NPX) Numeric Processor Extension [Yes/No]           YES
(IF)  Initialize On-board Functions [1,2,3,4/No]     1
(BIP) Board Initialization Procedure [1-45 Chars]

Enter  [ Abbreviation = new_value / Abbreviation ? / H ]
:
```

```
(BUS) System Bus Type [ 1 = MBI / 2 = MBII]          1
```

This parameter specifies the type of system bus that your system contains. If you enter a 2 for this parameter, the MULTIBUS II hardware screen will be shown.

```
(TP)  8254 Timer Port [0-OFFFFH]                         ODOH
```

This parameter specifies the base port address of your 8254 timer. The 8254 component provides timing for your Operating System. The base port address is the lowest valued port of the 8254 timer and is the only address that you need to configure.

```
(CIL) Clock Interrupt Level [0-7]                        0
```

You must specify the interrupt line on the master PIC to which the timer is connected. (Note that this is not an encoded interrupt level.) The default value for this parameter is set at 0.

```
(CN)  Timer Counter Number [0,1,2]                     0
```

This parameter specifies the number of the Programmable Interval Timer (PIT) counter used by the Nucleus. This counter cannot be used by any other software.

Three counters are associated with the 8254 PITs. The Nucleus needs one of these counters, usually counter zero. Counter 1 is used by the iSDM monitor. One of the terminal drivers can use the remaining counter, usually counter number two. If you are using an Intel board such as the iSBC 286/12, you should not change the default value unless you have a specific need to do so.

```
 (CIN)  Clock Interval [0-65535 msec]              10
```

This parameter specifies the standard clock interval for the iRMX II Operating System in units of one millisecond. This interval is normally 10 milliseconds. Unless an application requires a different value, it is highly recommended that you use this standard value. This will ensure that programs using timed wait operations will be portable between iRMX II systems.

```
 (CF)  Clock Frequency [0-65535 kHz]             1229
```

This parameter specifies the frequency of the clock input to the timer. The frequency is measured in kilohertz.

All Intel processor boards use the corresponding clock frequency 1228.8 kilohertz (entered as 1229) for this parameter.

The Nucleus loads a down count value into the timer register based on the formula:

count = clock interval (CIN) * clock frequency (CF)

```
 (TPS) Timer Port Separation [0-0FFH]              02H
```

This parameter specifies the timer port separation of your 8254 PIT. Starting with the base port address, the ICU determines the address of the next port by adding your response to this parameter line to the previous address. All Intel processor boards use a value of 02H for this parameter. Refer to Chapter 10 for information on how to configure timers for the Terminal Drivers.

```
 (NPX) Numeric Processor Extension [Yes/No]          YES
```

You must specify whether your system includes an 80287/80387 Numeric Processor Extension. The 80287/80387 Numeric Processor Extension (NPX) is a coprocessor that performs arithmetic operations on a variety of numeric data types. If any task in your application contains or will contain floating-point instructions, these instructions require the 80287/80387 component for execution. Therefore, you must specify "Yes" to the "Numeric Processor Extension" parameter line.

If no NPX is specified, your application cannot use floating-point instructions even if the NPX is actually in the system.

The "User Jobs", "I/O Jobs", and the "HI Jobs" screens also have "Numeric Processor Extension" parameter lines. If you respond "Yes" to any of these parameter lines, you must also specify "Yes" on the "Hardware" screen. If you specify "No" on the "Hardware" screen and "Yes" on any of the other screens, the ICU issues an error message telling you to change the value on the "Hardware" screen. You should also specify "Yes" to the NPX parameter in the "Hardware" screen if you respond "No" to the other three screens but anticipate having tasks that require floating-point arithmetic.

For further information about the 80287/80387 NPX, refer to the *Microsystem Components Handbook* Volume I.

```
 (IF)  Initialize On-board Functions [1,2,3,4/No]          1
```

This parameter indicates the type of board used in your system. Your response determines the board the Nucleus will initialize. Set this parameter to "No" if your system is a MULTIBUS II system.

Response	Board Initialized
1	iSBC 286/10(A), iSBC 286/12
2	iSBC 286/100A
3	iSBC 386/20
4	iSBC 386/1xx
"No"	Indicates that you do not want the Operating System to initialize your board. If you enter "No" for this parameter, you must supply a procedure name in the "(BIP) Board Initialization Procedure" parameter. Do not specify a custom initialization routine for MULTIBUS II systems.

```
 (BIP) Board Initialization Procedure [1-45 Chars]
```

Use this parameter only if you responded "No" to the "IF" parameter. This parameter specifies your custom initialization routine. The name you enter here is the file name of the object module. Your initialization routine must contain the PUBLIC names INIT_BOARD1 and INIT_BOARD2. The first is called before the initial memory scan. The second is called after the memory scan.

1.3 MULTIBUS® II HARDWARE SCREENS

This screen defines the MULTIBUS II hardware environment for your system. This screen appears only if you specify a "2" for the System Bus Type parameter of the Hardware screen. The defaults shown for this screen are from the 386100.def definition file.

```
(MBII)      Multibus II Hardware

(MDP) Message Device Base Port Address [0-0FFFFH]                 0H
(MDS) Message Device Port Separation [0-0FFH]                     04H
(MDL) Message Interrupt Level [0-7]                               2
(MCO) Message Device Duty Cycle for One Cycle DMA [0-0FFH]        04FH
(MCT) Message Device Duty Cycle for Two Cycle DMA [0-0FFH]        089H
(DDP) Message Device ADMA Data Port [0-0FFFFH]                    0H
(BEA) BIST Entry Address [0:0 = none/CS:IP]                   0000:0000H
(CLM) Configure iLBX Memory [Yes/No]                              NO
(CPM) Configure iPSB Memory [Yes/No]                              NO
(PSP) iPSB Starting Paragraph [0-0FFFFH]                          0H
(AIB) ADMA Base Port Address [0-0FFFFH]                           200H
(ACI) ADMA Channel for Input [0-0FFFFH]                           02H
(ACO) ADMA Channel for Output [0-0FFFFH]                          03H
(ABW) 16/32 bitness for ADMA [16/32]                              16

Enter  [ Abbreviation = new_value / Abbreviation ? / H ]
:
```

```
(MDP) Message Device Base Port Address [0-0FFFFH]              0H
```

This parameter specifies the message device's (MPC) lowest I/O port address. It is the only port address needed for the MPC.

```
(MDS) Message Device Port Separation [0-0FFH]                    04H
```

This parameter specifies the separation between port addresses of the message device's (MPC) registers. The ICU computes the addresses of each successive MPC register by adding the value you specify here to the current value, starting with the Message Device Base Port Address.

```
(MDL) Message Interrupt Level [0-7]                              2
```

This parameter indicates the interrupt level that the message device is attached to on the master PIC. Only an unencoded master level may be specified.

```
(MCO) One Cycle DMA [0-0FFH]                                   4FH
```

This parameter indicates the MPC duty cycle for the local bus. Only the iSBC 386/1xx boards support one-cycle Direct Memory Access (DMA). Note that you must also specify a "32" on the "(ABW) 16/32 Bitness for ADMA" parameter to get one-cycle DMA support.

```
(MCT) Two Cycle DMA [0-0FFH]                                   89H
```

This parameter indicates the MPC duty cycle for the local bus. All Intel boards that contain an MPC support two-cycle Direct Memory Access (DMA)

```
(DDP) Message Device ADMA Data Port [0-0FFFFH]              0H
```

This parameter specifies the DMA data port address (the default value is 0H).

Appropriate values by board type are as follows:

Board	Value
iSBC 286/100A board	0E0H
iSBC 386/1xx board	0H

```
(BEA) BIST Entry Address [0:0 = none/CS:IP]           0000:0000H
```

This parameter specifies the address of the Built-in Self Test (BIST) entry point. The BIST tests provide a system-wide "GO" or "NO GO" condition. These tests are supported for either MULTIBUS I or MULTIBUS II systems. Enter '0' if BIST support is not desired.

```
(CLM) Configure iLBX Memory [Yes/No]                          NO
```

This parameter indicates whether the Nucleus should or should not automatically configure the start and stop addresses of memory boards found on the Local Bus Extension (iLBX) bus.

A 'yes' is only valid in a configuration where the application system resides in ROM.

```
(CPM) Configure iPSB Memory [Yes/No]                          NO
```

This parameter indicates whether the Nucleus should or should not automatically configure the start and stop addresses of memory boards found on the iPSB bus.

A 'yes' is only valid in a configuration where the application system resides in ROM.

```
(PSP) iPSB Starting Paragraph [0-0FFFFH]                      0H
```

This parameter specifies the starting address, in units of 64K bytes, to assign to the first memory board found on the iPSB bus.

A 'yes' is only valid in a configuration where the application system resides in ROM.

```
(AIB) ADMA Base Port Address [0-0FFFFH]                    0200H
```

This parameter specifies the I/O base port address of the message passing DMA device.

```
(ACI) ADMA Channel for Input [0-0FFFFH]                        02H
```

This parameter indicates the ADMA channel used for input.

```
(ACO) ADMA Channel for Output [0-0FFFFH]                       03H
```

This parameter indicates the ADMA channel used for message-passing output.

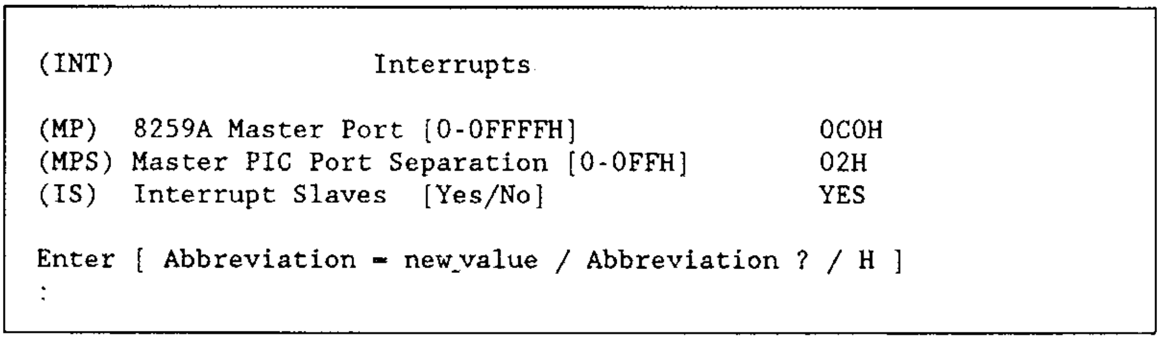
```
(ABW) 16/32 bitness for ADMA [16/32]                            16
```

This parameter specifies whether the ADMA devices memory accesses will be performed on a 16- or 32-bit local bus. Note that a response of 32, for iSBC 386/1xx boards only, is required to provide one-cycle Direct Memory Access (DMA).

1.4 INTERRUPT SCREENS

There are two interrupt screens. One defines the Master Interrupt Level and one defines the Slave Interrupt Levels. These screens are shown in the following sections.

1.4.1 Master Level Interrupt Screen



```
(INT)                  Interrupts

(MP)  8259A Master Port [0-0FFFFH]              0C0H
(MPS) Master PIC Port Separation [0-0FFH]       02H
(IS)  Interrupt Slaves  [Yes/No]                YES

Enter [ Abbreviation = new_value / Abbreviation ? / H ]
:
```

```
(MP)  8259A Master Port [0-0FFFFH]              0C0H
```

This parameter specifies the base port address of your 8259A Master PIC. The base port address is the lowest valued port of the 8259A PIC and is the only address that needs to be configured for the 8259A PIC. When using Intel processor boards such as the iSBC 286/12 board and the supplied definition file for the board, you should not change the default value.

The ICU determines the 8259A PIC addresses based on your responses to the "8259A Master Port" and "Master PIC Port Separation" parameter lines.

```
(MPS) Master PIC Port Separation [0-0FFH]          02H
```

This parameter specifies the interval between each 8259A PIC port. The ICU configures each additional 8259A PIC port address for you. Starting with the base port address, the ICU computes the address of the next port by adding your response to the "Master PIC Port Separation" parameter line to the previous address.

All Intel processor boards use a value of 02H for this parameter.

```
(IS)  Interrupt Slaves  [Yes/No]                   YES
```

This parameter indicates that there are slave 8259A PICs connected to the master 8259A PIC. A response of "Yes" causes the "Slave Interrupt Levels" screen to be displayed.

1.4.2 Slave Interrupt Levels Screen

This repetitive screen lets you enter the interrupt levels on your master PIC which are connected to slave PICs, if the slave is level-sensitive, the port and the port separation of the slave PICs.

```
(SLAVE)        Slave Interrupt Levels

Slave -  Slave_number, Level_Sensitive,   Port,      Separation
              [0-7],        [Yes/No]     [0-0FFFFH]    [0-0FFH]

[1] Slave =        7 ,         NO ,         0C4H   ,      02H
[2] Slave =


Enter Changes [ Number= new_value / ^D Number / ? / H ]
:
```

This screen lets you enter the interrupt levels on your master 8259A PIC that are connected to slave 8259A PICs; also, which slave is level-sensitive, the port and port separation of the slave 8259A PICs.

If you need to add an additional interrupt level to those you have already specified, enter a new line on the screen using the rules for editing repetitive screens defined in the *Extended iRMX II Guide To The Interactive Configuration Utility*.

There are two ways of sensing an active interrupt request: a level-triggered mode or an edge-triggered mode. The 8259A component gives the user the capability for either method. However, since the system clock must be on an edge-sensitive master line, the ICU configures the 8259A master interrupt controller only in edge-triggered mode. Refer to the *Component Data Catalog* for additional information on the 8259A component and edge-triggered and level-triggered modes.

CHAPTER 2
MEMORY PARAMETERS

2.1 INTRODUCTION

This chapter discusses how to respond to the prompts that appear on the two memory related screens. These two screens - "Memory for System" and "Memory for Free Space Manager" define the memory blocks available to the iRMX II Operating System for the system's free space and the iRMX II system code. The following sections display the screens and their default values.

2.2 MEMORY FOR SYSTEM SCREEN

The "Memory for System" screen and its default values are displayed here. The values shown here are part of the Intel supplied Start-up System.

```
(MEMS)        Memory for System

    SYS    = low [0-0FFFFFFH], high [0-0FFFFFFH]
[1] SYS    =          02000H    ,         059FFFH
[2] SYS    =

Enter Changes [ Number = new_value / ^D Number / ? / H ]
:
```

This screen defines the memory blocks reserved for the Operating System code and data segments. You must enter both the start and end addresses of each contiguous block of memory you want to reserve. The addresses you enter are the 24-bit physical memory locations the ICU will reserve. When entering addresses for SYS, be sure that the lower limit is not less than 1000H. The actual address is determined by the device drivers in your system, the type of monitor, and the location of the Bootstrap Loader. The following sections discuss these considerations.When configuring your system for the first time, you may want to pad the memory you allocate for the system. This allows you enough memory for any changes you make to the definition files or any device drivers you may add during development. Once you have a final system, you can minimize the system memory by changing the values on this screen. Follow these steps to determine the final values for the memory parameters.

1. Run the ICU.
2. Invoke the SUBMIT file.
3. Read the MP2 file generated by the System Builder (BLD286) to find the amount of memory the system requires.
4. Rerun the ICU entering the size obtained in step 3 as the "SYS" parameter. The rest of the memory is available to the free space manager.
5. Regenerate the system.

Chapter 3 of the *Guide To The Extended iRMX II Interactive Configuration Utility* explains this process in more detail.

The memory you define for "SYS" is ROM if your system is ROM-based and RAM if your system is RAM-based. If your system is ROM-based, you must also define RAM memory on the "ROM Code" screen.

2.2.1 Bootstrap Loader Considerations

When reserving memory locations, remember that the second stage of the Bootstrap Loader resides in memory locations 0B8000H - 0BFFFFH, the third stage resides in memory locations 0BC000H - 0BFFFFH by default. If you define this memory range for the system, you will cause the system to overwrite the Bootstrap Loader. You may, however, use this memory for the free space manager as the memory is no longer needed after the system is loaded.

2.2.2 Sub-System Considerations

When calculating the memory required by the system, you must consider the memory required by the subsystems as well as that required by the descriptor tables. The size of the Global Descriptor Table (GDT) and the Local Descriptor Table (LDT) depends on the number of entries you configure them to contain. Each entry is eight bytes and each table has a maximum of 32K entries. The Intel-supplied, Multi-user System defines both the GDT and the LDT as 32K. The Interrupt Descriptor Table (IDT) requires 1K bytes of memory.

The memory requirements for each of the sybsystems are

Sub-system	Approximate Memory Required
Nucleus	34K
BIOS	95K
EIOS	19K
Application Loader	12K
Human Interface	36K
CLI	56K (Release 2 CLI)
UDI	11K
SDB	25K

The size of the Basic I/O System varies depending on the number of device drivers included in the system. The figure listed above is the size of the BIOS in the Intel-supplied start-up system.

2.2.3 Mass Storage Controller (MSC) Considerations

If your application includes an MSC driver with a start I/O address of 100H, it is recommended that the lower limit of the "SYS" parameter not be less than 1070H. The MSC driver requires 68 bytes of memory. Be sure to take these 68 bytes into consideration when your are configuring your system memory. For the exact memory locations and the method for calculating them, see the discussion of the "Wakeup I/O Port Address" and "I/O Processor Block Address" parameters in Chapter 10. Do not include that memory space in the memory you define on the "Memory for System" or "Memory for Free Space Manager" screens. If your application does not include this driver, you may define contiguous memory from 1000H.

2.2.4 Communication Board Considerations

Communication boards, such as the iSBC 544A and the Terminal Communications Controller boards (specifically the iSBC 188/48/56 and the iSBC 546/547/548 boards), have on-board dual-port memory which cannot be managed by the Operating System. If your system includes such a communication board, do **not** include the board's on-board, dual-port memory in the memory declared on the "Memory for System" screen.

The default value on the "iSBC 544A Driver" screen allows the on-board memory to start at FE0000H; the default for the "Terminal Communications Controller Driver" screen starts the on-board memory at F90000H. Using these default values dictates that the upper address on the "Memory for System" screen should be FDFFFFH if you use the iSBC 544A driver (or F8FFFFH if you include only the Terminal Communications Controller). If you change the "Memory Address Base" parameter lines on the "iSBC 544A Driver" and "Terminal Communications Controller Driver" screens, you must also reflect that change on the "Memory for System" and "Memory for Free Space Manager" screens. Refer to Chapter 10 for more information about the iSBC 544A and Terminal Communications Controller drivers.

2.2.5 RAM Driver Considerations

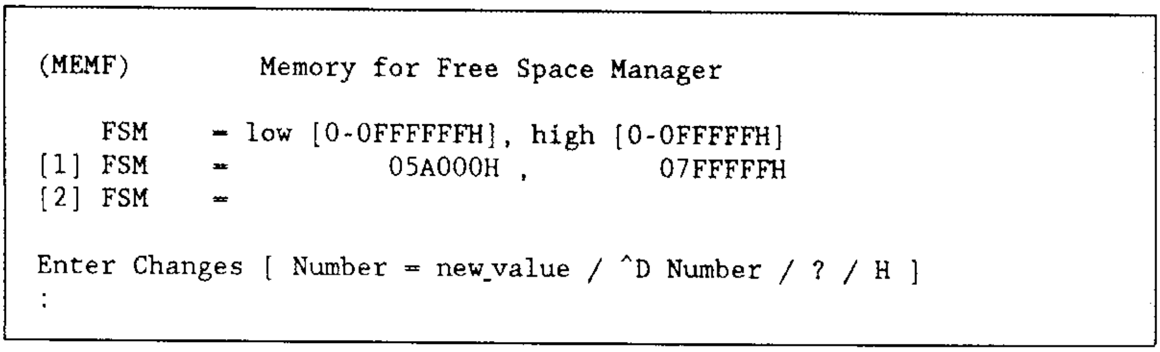
If your system includes the RAM Driver, an Intel Device screen allows you to specify the base memory address for the driver. Like the memory for a communications board, the memory that you specify for the RAM Driver cannot be included in the memory that you specify on the "Memory for System" or "Memory for Free Space Manager" screens.

2.2.6 Monitor Considerations

If you have a RAM-based system, you must consider the memory required by the monitor and the Bootstrap Loader when calculating the upper limit of memory. The Operating System does not have access to this memory. Therefore, do not include the location (either the top 32K for an SDM monitor or 64K for a monitor with SCT PROMs) allocated to monitor and Bootstrap Loader code in your "Memory for System" or "Memory for Free Space Manager" screens.

2.3 MEMORY FOR FREE SPACE MANAGER SCREEN

The "Memory for Free Space Manager" screen and its default values are displayed here. The values shown here are the default values in the 28612.def definition file.



```
(MEMF)          Memory for Free Space Manager

    FSM     = low [0-0FFFFFFH], high [0-0FFFFFH]
[1] FSM     =          05A000H ,         07FFFFFH
[2] FSM     =

Enter Changes [ Number = new_value / ^D Number / ? / H ]
:
```

This screen defines the memory blocks available to the Operating System for the system's free space. You must enter both the start and end addresses of each contiguous block of memory you want to reserve. The addresses you enter are the 24-bit physical memory locations the ICU will reserve. When entering addresses for the FSM, be sure to take into consideration all the locations you reserved for system memory.

If you change the default definition file, add device drivers or user jobs, calculate the correct values for this parameter in the following manner.

1. Run the ICU.
2. Invoke the SUBMIT file.
3. Read the MP2 file generated by the System Builder (BLD286) to find the amount of memory the system requires.
4. Rerun the ICU entering the size obtained in step 3 as the "SYS" parameter. The rest of the memory is available to the free space manager.
5. Regenerate the system.

The memory you define for the FSM is RAM memory regardless of whether or not your system is RAM-based or ROM-based.

CHAPTER 3
SUB-SYSTEM PARAMETERS

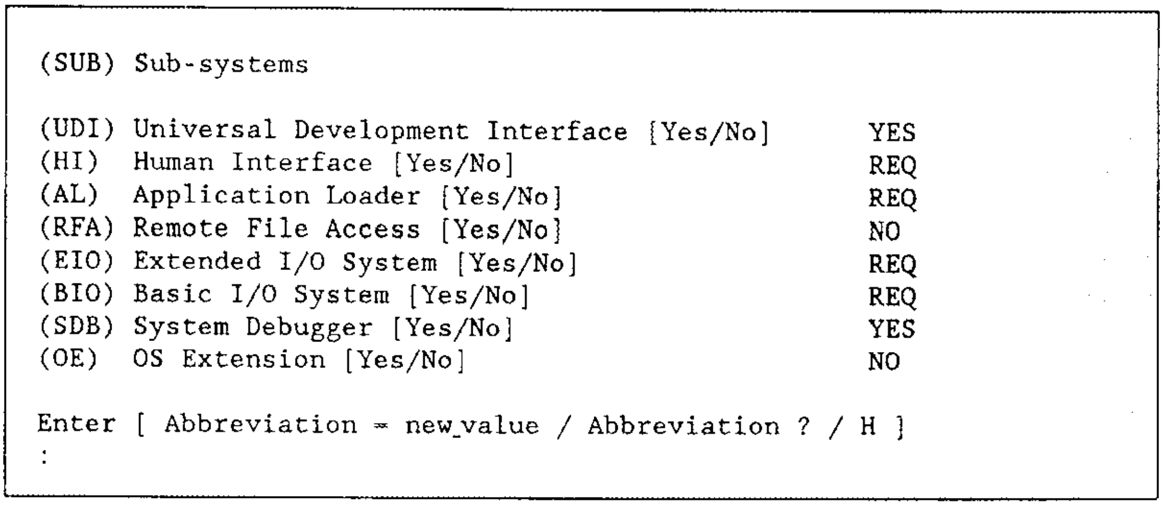
3.1 INTRODUCTION

This chapter discusses how to respond to the parameters that appear on the "Sub-systems" screen. The iRMX II Operating System consists of the Nucleus, which is required, and a number of subsystems which are optional.

3.2 SUB-SYSTEMS SCREEN

Your responses on the following "Sub-systems" screen determine which subsystems will be included in your system. A "Yes" response to a parameter line on this screen has a number of implications. First, specific capabilities are added to your system. Second, adding a subsystem increases your memory requirements.

The screen shown here illustrates how the ICU responds to a prompt of "Yes" for the UDI subsystem. This response requires that most of the other subsystems be included.



```
(SUB) Sub-systems

(UDI) Universal Development Interface [Yes/No]      YES
(HI)  Human Interface [Yes/No]                      REQ
(AL)  Application Loader [Yes/No]                   REQ
(RFA) Remote File Access [Yes/No]                   NO
(EIO) Extended I/O System [Yes/No]                  REQ
(BIO) Basic I/O System [Yes/No]                     REQ
(SDB) System Debugger [Yes/No]                      YES
(OE)  OS Extension [Yes/No]                         NO

Enter [ Abbreviation = new_value / Abbreviation ? / H ]
:
```

The system that you build uses the Nucleus and any needed subsystem. Each optional subsystem requires at least one other subsystem (which may be the Nucleus). Table 3-1 shows the dependency of one configurable subsystem to another.

Table 3-1. Subsystem Dependencies

Subsystem	Supporting Subsystems				
	HI	AL	EIO	BIO	SDB
Universal Development Interface (UDI)	Req	Req	Req	Req	
Human Interface (HI)		Req	Req	Req	
Application Loader (AL)			Req*	Req	
Remote File Access				Req	
Extended I/O System (EIO)				Req	
Basic I/O System (BIO)					
System Debugger (SDB)					
OS Extension (OE)					
Note: * See section on Application Loader for explanation.					

If you specify "Yes" to a subsystem that requires another subsystem, the ICU displays "REQ" for each required subsystem. Refer to the *Introduction To The Extended iRMX II Operating System* for additional information on subsystems.

3.2.1 Universal Development Interface

The Universal Development Interface (UDI) is an optional sub-system of the Operating System that provides a standard set of system calls. These system calls allow you to write applications that can run on any operating system supporting the UDI. Also, UDI provides a standard, flexible, protocol which allows language translators, language run-time packages, and other software development tools to run on the iRMX II Operating System. Refer to the *Extended iRMX II Universal Development User's Guide* for detailed information about UDI.

3.2.2 Human Interface

The Human Interface is an optional subsystem that adds an interactive interface between users and software. This is a layer of the Operating System that builds on (and requires) the capabilities of the Application Loader, the Extended I/O System, and the Basic I/O System. Refer to the *Extended iRMX II Human Interface User's Guide* for detailed information about the Human Interface.

If you plan to use any of the Human Interface system calls, commands, or other features of the Human Interface, you must specify "Yes" to the "Human Interface" parameter line. However, there is no need to specify "Yes" if you have already specified "Yes" to the UDI parameter line.

3.2.3 Application Loader

The Application Loader is an optional subsystem that adds the capabilities to load object files into memory from secondary storage under the control of the Operating System. It also provides system calls that load programs in systems that do not have enough memory to accommodate all programs at one time. The Application Loader can be used to load programs that contain overlays.

The Application Loader is a sub-system that requires the Basic I/O System and may require the Extended I/O System. (Your response to the "All System Calls" parameter line on the "Application Loader" screen determines the need for the Extended I/O System.) Refer to the *Extended iRMX II Application Loader User's Guide* for more information about the Application Loader.

If you plan to use any of the features of the Application Loader, you must specify "Yes" to the "Application Loader" parameter line. There is no need to specify "Yes", however, if you have already specified "Yes" to either the UDI or Human Interface parameter lines.

3.2.4 Remote File Access

Remote File Access applies to iRMX Networking Software operating in a file consumer mode. The iRMX Networking Software is an optional package purchased separately from the iRMX II Operating System. If you have iRMX-NET and you want your system to access remote files (be a file consumer), answer "yes" to this parameter. If you have iRMX-NET and you want your system to supply files to remote systems (be a file server), answer "no" to this parameter. If you want your system to do both (be both a file server and a file consumer), answer "yes" to this parameter.

The Remote File Access requires the Basic I/O System; the Extended I/O System is optional. Refer to the *Extended iRMX Networking Software User's Guide* for more information about the iRMX-NET software.

3.2.5 Extended I/O System

The Extended I/O System is an optional subsystem that adds high-level, synchronous file access capabilities. It requires the capabilities of the Basic I/O System. It allows a user to access system I/O devices without having to write procedures to specify particular devices with particular device names. Refer to the *Extended iRMX II Extended I/O System User's Guide* for more information.

If you plan to use any of the features of the Extended I/O System, you must specify "Yes" to the "Extended I/O System" parameter line. There is no need to specify "Yes", however, if you have already specified "Yes" to the UDI, or Human Interface parameter lines.

3.2.6 Basic I/O System

The Basic I/O System is an optional subsystem that provides asynchronous file access capabilities. The Basic I/O System provides an extensive facility for device-independent I/O. It supplies all file drivers and a number of device drivers. Refer to the *Extended iRMX II Basic I/O System User's Guide* for details about the Basic I/O System.

If you plan to use any of the Basic I/O System system calls, any of the file access capabilities, or any other feature of the Basic I/O System, you must specify "Yes" to the "Basic I/O System" parameter line. However, there is no need to specify "Yes" if you have already specified "Yes" to the UDI, Human Interface, Application Loader, Remote File Access, or Extended I/O System parameter lines.

3.2.7 System Debugger

The System Debugger extends the capabilities of the iSDM Monitor. The iRMX II System Debugger (SDB) extends the use of the monitor so that you can interactively examine data structures handled by the iRMX II Operating System.

If you include the SDB you must also include iSDM.

For more information on the monitor, consult the following manuals: the *iSDM System Debug Monitor User's Guide*, or the *Extended iRMX II System Debugger Reference Manual.*

3.2.8 OS Extension

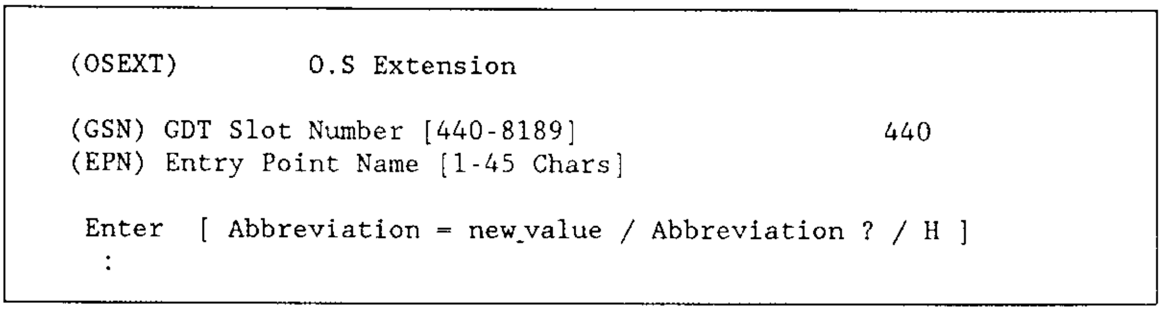
Use this parameter to specify whether or not your system includes OS extensions. OS extensions are a feature of the iRMX II Operating System that let you modify the Operating System to include custom objects. With this feature you can create an operating system that precisely meets your needs. Changing the default to "Yes" indicates that the system will include OS Extensions. It will also ensure that the "OS Extension" Screen is displayed. For more information on OS Extensions, see the *Extended iRMX II Nucleus User's Guide*.

CHAPTER 4
OS EXTENSION PARAMETERS

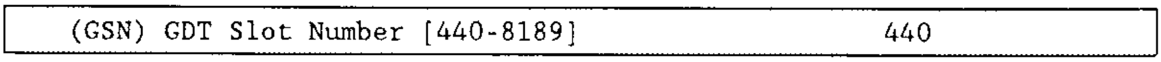
4.1 INTRODUCTION

This chapter explains the OS extension parameters and the way you may respond to the prompts that appear on the screen. Even if you are familiar with the iRMX I Operating System, you should read this chapter carefully as this screen is a new feature in the iRMX II Operating System.

4.2 OS EXTENSION SCREEN



```
(OSEXT)        O.S Extension

(GSN) GDT Slot Number [440-8189]                     440
(EPN) Entry Point Name [1-45 Chars]

 Enter  [ Abbreviation = new_value / Abbreviation ? / H ]
  :
```



```
(GSN) GDT Slot Number [440-8189]                     440
```

OS Extensions are entered through call-gates whose GDT location must be established during configuration. This parameter allows you to define the actual GDT location that will be reserved for a call-gate. This prevents your call-gates from colliding with any current or future call-gates used by the Operating System. Be careful when responding to this parameter not to enter slot numbers that you have reserved for other OS extensions or that you have configured out on the "Nucleus" Screen. The ICU does not check the logic of your entries.

```
 (EPN) Entry Point Name [1-45 Chars]
```

The name you enter for this parameter defines the PUBLIC name of the entry procedure from which the call-gate enters the OS extension. When binding your OS extension, you must include a NOPUBLIC EXCEPT(xxxx) control, where xxxx is the name you enter in the "EPN" parameter. Do not enter the same entry point name for different OS extensions. Each name must be unique. The pathname of the object file (obtained from compiling and binding the OS extension) containing this procedure must also be specified on the "User Module" Screen.

When you configure an OS extension, the ICU creates a file called USER.GAT which contains the gate's PUBLIC name. Therefore, you should bind USER.GAT with the procedure calling the OS extension and its interface in order to invoke the appropriate call-gate.

4.3 QUERY SCREEN

After you have completed the "OS Extension" screen, the query screen is displayed. It contains only one line:

```
Do you need any/more OS Extensions ?
```

Respond "Yes" if you have more OS extensions. The "OS Extension" screen is then redisplayed. Otherwise, respond "No" or enter a carriage return. The only limit on the number of OS extensions is the number of GDT slots available (GDT maximum is 8K slots).

CHAPTER 5
HUMAN INTERFACE PARAMETERS

5.1 INTRODUCTION

This chapter describes how to select appropriate parameter values on the five Human Interface screens. If you are using this chapter to understand a parameter line, turn to the page number indicated to the right of the screen that contains your parameter line abbreviation.

Human Interface				
(ICL)	(CNM)	(SYS)	(RIP)	Start on
(UXC)	(SS)	(PMI)	(PMA)	page 5-5
(DTN)	(RU)			
HI Jobs				
(MIN)	(MAX)	(NPX)		Start on
				page 5-8
Resident/Recovery User				
(TN)	(TDN)	(MTP)	(UID)	Start on
(MIN)	(MAX)	(IPP)	(DEF)	page 5-9
Prefixes				
				Start on
				page 5-11
HI Logical Names				
				Start on
				page 5-12

There are a number of terms with which you should be familiar before configuring the Human Interface. To make the configuration process easier, this chapter begins with an overview of the basic Human Interface concepts. Following the overview, the chapter discusses the related parameter lines, in the manner of previous chapters.

5.2 BACKGROUND INFORMATION

This section introduces three families of terms and concepts: terminals, operators, and users. Using the Human Interface requires that these concepts be understood.

The key to understanding this family of concepts is the **terminal**, which is the hardware that acts as the interface between the operator and an iRMX II application system. Whenever operators log onto the Human Interface from terminals, they are assigned IDs. These IDs identify **users**, which are a means of controlling and determining the access rights to files, as described in the *iRMX II Basic I/O System User's Guide*. An iRMX II application system can incorporate one or more users, and the configuration considerations are different for single-user systems than for multi-user systems.

5.2.1 Resident and Non-Resident Configuration, Users, and Initial Programs

Human Interface configuration consists of two parts: resident and non-resident configuration. **Resident Human Interface configuration** is the process of defining the parts of the Human Interface subsystem that are always present in memory while the Operating System is running. **Non-resident Human Interface configuration** is the process of defining other elements of the overall Human Interface configuration that can reside in secondary storage files.

Users of the Human Interface subsystem are normally called resident, recovery resident, or non-resident users. You specify the configuration information for the **resident user** and the **recovery resident user** on the "Resident/Recovery User" screen while running the ICU. This information becomes part of your final system and resides in memory along with the rest of the Operating System (thus, the term "resident user"). There are two types of resident users. One is a recovery resident user which gains control only if an initialization error occurs. The second is a resident user which occupies one of the system terminals and is created before the nonresident users. The Operating System can contain information about only one resident user. If you need to have multiple users, one of them can be the resident user; the rest must be non-resident users (or all users can be non-resident). One of the parameter lines on this screen calls for the device name of a terminal which is this user's console. A **non-resident user** is a user whose configuration information resides on an iRMX II named volume rather than in memory with the Operating System (thus, the term "non-resident user"). Non-resident configuration of the Human Interface is discussed in *Guide To The iRMX II Interactive Configuration Utility*.

An initial program is associated with each user. A user's initial program is the program that starts running immediately after the Human Interface creates the user's interactive job. Most users have the Human Interface-supplied Command Line Interpreter (CLI) as their initial program.

Only one resident initial program can be present in a system. Therefore, all users who are configured to use the resident initial program will have the same initial program. Either the default CLI or a user-supplied program can be the resident initial program.

The only important difference between resident and non-resident initial programs is that the resident program can be shared among any number of users, whereas each non-resident initial program can belong to only one user. That is, only one copy of the resident initial program is required, but if multiple users are to have the same non-resident initial program, a copy of that program must be loaded for each user. Which type of initial program is chosen for a user is independent of the type of user. That is, a non-resident user can use the resident initial program, or a resident user can use a non-resident initial program.

If you have a single-user system, the only disadvantage of configuring your single user as a resident user is the inconvenience of changing parameters associated with a resident user. To change a configuration parameter for a resident user, you must use the ICU to reconfigure the entire system. This takes longer to do than merely changing a value in a non-resident user configuration file and rebooting or restarting your system.

If you have a multi-user system, you can configure your system in any of the following ways:

- All users are non-resident
- One resident user and one or more non-resident users

If your user parameters change frequently, you probably want to configure all your users as non-resident. It is recommended that you configure a recovery resident user which is invoked only if the non-resident configuration files are damaged, or for some reason the first user could not be initialized. This default option provides all of the advantages of the other two options.

If one of your users is not going to change, you might want to make that user the resident user and the remaining users non-resident. The advantage of doing this is that if the non-resident configuration files are somehow corrupted, the system can still be bootstrap loaded and initialized. The resident user can then access the system and correct the problems.

If one or more people using your system perform system manager functions frequently, you might want to configure the system manager (user ID 0) as the resident user. Doing this gives you two advantages. First is the advantage described in the previous paragraph. Second, your system manager has immediate access to system manager functions without invoking the SUPER command of the Human Interface. The disadvantage of having a system manager as the resident user is that, if you are concerned about security, you must permit access to the system manager terminal only to the person who is the system manager. This eliminates one terminal, and the memory required to support it, from access by the general user population.

5.2.2 System Device, Directories, Logical Names, and Prefixes

If you configure the Extended I/O System to use the Automatic Boot Device Recognition feature and your application system is bootstrap loaded, the device from which it loads is automatically the **system device**. Otherwise, the system device is another device that is specified during the Extended I/O System configuration. A special directory on the system device is the **system directory**. You specify names for both the system device and the system directory in the "Human Interface" screen. The system directory contains the code for Intel-supplied Human Interface commands that you can enter at the terminals.

As part of configuration, you must specify a default directory for each user. A user's **default directory** is the directory to which he is automatically "attached" on system initialization or dynamic logon. It has the following traits, it is the directory to which the user will be attached if ATTACHFILE :HOME: <CR> is entered at the terminal, and it is the directory used if a partial pathname is used in a commmand.

One of the Human Interface configuration screens is devoted to logical names. A **logical name** is an alias for a pathname. For example, you could assign the logical name DATA to the pathname :SD:SYS286/DATAFILE. That would allow you to use :DATA: in place of :SD:SYS286/DATAFILE, both in programs and at terminals. Note that logical names, when used, must be surrounded by a pair of colons. The logical name WORK is present by default in every logical name list.

Another of the Human Interface configuration screens requires you to specify a list of prefixes. A **prefix** is a logical name of a directory. The order in which you list the prefixes is important since the Human Interface searches the directories they stand for, in list order, whenever a program (command) is invoked from a terminal with a partial pathname. Normally the list of prefixes should order the directories from most used to the least used.

The prefix list must contain :SYSTEM:, which is the logical name of the system directory. The default prefix list should also contain :PROG:, :UTILS:, :LANG:, and :$:. These directories are described in the *Operator's Guide To The iRMX II Human Interface.*

5.3 THE SCREENS

If you elect to include the Human Interface in your application system, the ICU automatically includes all system calls of the Human Interface, Application Loader, Extended I/O System, Basic I/O System, and Nucleus. The ICU obtains information about the Human Interface from the five screens explained in this section.

5.3.1 Human Interface Screen

The ICU obtains information about the Human Interface as a whole from the following screen.

```
(HI)            Human Interface

(ICL) Initial Command Line Size [0-65535]          256
(CNM) Command Name Length [1-255]                  64
(SYS) System Directory [1-45 Chars]
                                                 :SD:SYS286
(RIP) Resident Initial Program [IntelCLI/1-45 Chars]
                                                   INTELCLI
(UXC) User Extension for Intel CLI [1-45 chars]

(SS)  Initial Program Stack Size [0-0FFFFH]        02400H
(PMI) Human Interface Pool Minimum [0-0FFFFFH]     0260H
(PMA) Human Interface Pool Maximum [0-0FFFFFH]     0FFFFFH
(DTN) Default Terminal Name [1-6 characters]       ANY
(RU)  Resident User [Yes/Recovery only/None]       RECOVERY

Enter  [ Abbreviation = new_value / Abbreviation ? / H ]
:
```

```
    (ICL) Initial Command Line Size [0-65535]          256
```

This parameter lets you specify the initial size of the command line buffer. Whenever a Human Interface operator enters characters at a terminal to invoke a command, the CLI (if it is the resident initial program) places the command in a buffer called the command line buffer.

The size of this buffer should be the greatest number of characters that will be in any one command line. Care should be taken in selecting this value. Specifying a value that is too large will waste memory. Specifying a value that is too small can affect performance.

If your application system does not have the Human Interface CLI (INTELCLI) as its resident initial program, the ICU ignores this parameter.

```
     (CNM) Command Name Length [1-255]                  64
```

This parameter lets you specify the number of characters in the longest name among the command pathnames that the Human Interface CLI is to handle.

If your application system does not have the Human Interface CLI as its resident initial program, the ICU ignores this parameter.

```
    (SYS) System Directory [1-45 chars]
                                                    :SD:SYS286
```

This parameter lets you specify the pathname of the system directory on the system device. Your response, which is limited to 45 characters, must be the pathname of the system directory. The default system directory pathname :SD:SYS286 specifies :SD: as the system device and SYS286 as the system directory on that device. (You specify the logical name :SD: on the "Automatic Boot Device Recognition" screen, described later.)

```
   (RIP) Resident Initial Program [IntelCLI/1-45 characters]
                                                    INTELCLI
```

This parameter lets you specify the resident initial program. You can respond in either of two ways. The default response, which is "INTELCLI", indicates that the resident initial program is to be the Human Interface CLI program.

The other way to respond is to give the pathname of an object code file that is to be the alternate resident initial program replacing the Human Interface CLI. The object code is bound and built into the system during system generation. If you wish to use the iRMX II.1 Operating System as your resident CLI, enter the pathname /RMX286/HI/HR1CLI.LNK in this parameter.

The initial program must meet the following criteria:

- the entry point must have the public label "H$CLI$INIT
- the CLI must either initialize its own data segment value or use no data segment at all
- the CLI must invoke the EXIT$IO$JOB system call to terminate its execution

```
     (UXC)  User Extension for Intel CLI [1-45 chars]
```

This parameter specifies the pathname of the user extension module that is to be bound (using BND286) with the Human Interface CLI. By adding user extensions the Human Interface CLI can be extended to include custom features.

If your application does not include user extensions, ignore this parameter. For more information on user extensions, see the *iRMX II Human Interface User's Guide*.

```
    (SS)  Initial Program Stack Size [0-0FFFFH]       02400H
```

This parameter lets you specify, in bytes, the size of the initial program's stack. If your system includes the Intel-supplied CLI as the initial program, use the default stack size. If you are using the default CLI with user extensions, you should add the requirements of the user extension to the default value. If you are using a custom initial program, you will have to calculate the stack size based on the requirements of your initial program.

```
    (PMI) Human Interface Pool Minimum [0-0FFFFFH]    0260H
    (PMA) Human Interface Pool Maximum [0-0FFFFFH]  0FFFFFH
```

These parameter lines let you specify the minimum and maximum sizes, in 16-byte paragraphs, of the Human Interface memory. (Memory pools in the iRMX II Operating System can be up to 16M bytes.)

Unless your system includes a first-level user job with a large range of memory pool requirements, you should set the Human Interface's maximum memory pool size (PMA) to the default value 0FFFFFH. This value allows the Human Interface to claim all memory that remains after all other first-level jobs have been initialized, because the order in which the root task initializes jobs is:

Nucleus
Basic I/O System
Extended I/O System (including I/O jobs)
First-level user jobs
Human Interface

By setting PMA to 0FFFFFH, you allow the Human Interface to borrow memory from the root job when necessary. Moreover, you make it possible to add more memory to your system without reconfiguring the Operating System.

To ensure that initialization is performed rapidly, you should set the minimum memory pool size (PMI) to a value large enough to allow the Human Interface to initialize itself without borrowing memory. The default value 0260H is always large enough for this purpose.

If your system includes a first-level user job that borrows memory, you might not want to set PMA to 0FFFFFH. Instead, you might want to set it to a lower value to ensure that the first-level job has access to additional memory.

By setting PMA equal to PMI, you ensure that no borrowing occurs.

```
    (DTN)  Default Terminal Name [1-6 characters]       ANY
```

This parameter provides a default terminal name for the entire system. It is used only if no terminal name is supplied as an entry in the :CONFIG:TERMINALS file. A response of "ANY" applies to all terminal types. However, it has limited functionality, providing only those line-editing features that were available in the first release of the iRMX II Human Interface. If you want to take full advantage of the line-editing features supplied in the second and third releases, enter the terminal name of a specific terminal, such as VT100. For a more detailed description of the :CONFIG:TERMINALS file, see the *Guide To The iRMX II Interactive Configuration Utility.*

```
    (RU)  Resident User [Yes/Recovery/None]         RECOVERY
```

This parameter provides three options for defining the resident user. You can respond: "Yes", "Recovery", or "None". A response of "Yes" means you want a resident user that comes up as the first HI user. A "Recovery" response means that a resident user will be present only if initialization errors occur. "Recovery" should be entered if there is any chance that some of your files may be damaged. The resident user invoked upon recovery is the resident initial program you defined in the "RIP" parameter on the "Human Interface" screen.

A response of "None" means the Human Interface will not create a resident user.

5.3.2 Human Interface Jobs Screen

The ICU obtains information about Human Interface jobs from the following screen.

```
  (HIJOB)         HI Jobs

  (MIN) Jobs Minimum Memory [0-0FFFFFH]                 0200H
  (MAX) Jobs Maximum Memory [0-0FFFFFH]                 0FFFFFH
  (NPX) Numeric Processor Extension Used [Yes/No]       NO

  Enter [ Abbreviation = new_value / Abbreviation ? / H ] :
  :
```

```
     (MIN) Jobs Minimum Memory [0-0FFFFFH ]              0200H
     (MAX) Jobs Maximum Memory [0-0FFFFFH ]             0FFFFFH
```

When an operator enters a Human Interface command, the Operating System invokes an Application Loader system call to load the command's code into memory from secondary storage, create an I/O job for the command, and execute the command. The MIN and MAX parameters specify the pool$lower$bound and pool$upper$bound parameters in 16-byte paragraphs of I/O jobs created during processing of commands.

MIN defines the minimum amount of memory with which a Human Interface job can be created. MAX defines the maximum amount of memory that a Human Interface job can have. Entering a value of zero means that the Application Loader will assign these values. Entering 0FFFFFH allows HI jobs to borrow memory. Refer to the *iRMX II Application Loader System Calls Reference Manual* for more information about the pool$lower$bound and pool$upper$bound parameters.

```
     (NPX) Numeric Processor Extension Used [Yes/No]      NO
```

This parameter lets you specify whether any of the HI jobs that are created contain floating-point instructions and therefore use an 80287/80387 Numeric Processor Extension (NPX). If you respond to this parameter with "Yes", you must also have entered "Yes" on the "Hardware" screen or you will receive an error message.

5.3.3 Resident/Recovery User Parameters

The ICU obtains information about the resident user from the following screen.

```
(RES)            Resident/Recovery User

(TN)  Terminal Name [1-6 Chars]                        ANY
(TDN) Terminal Device Name [1-14 Chars]                TO
(MTP) Maximum Task Priority [0-255]                    192
(UID) User ID Number [0-OFFFFH]                        OH
(MIN) Minimum Memory Required [0-OFFFFFH]              02000H
(MAX) Maximum Memory Required [0-OFFFFFH]              OFFFFFH
(IPP) Initial-Program Pathname [Resident/1-45 Chars]
                                                       RESIDENT
(DEF) Default Directory [1-45 Chars]
                                                 :SD:USER/SUPER

Enter  [ Abbreviation = new_value / Abbreviation ? / H ]
:
```

```
      (TN)  Terminal Name [1-6 Chars]                    ANY
```

This parameter provides the name of a terminal that will be used as the Resident/Recovery user. If this screen defines a resident user, this parameter defines the terminal name for the resident user. However, if this screen defines a recovery resident user, the terminal name entered for this parameter is used only if an initialization error occurs and the :CONFIG:TERMINALS file can not be opened. For a more detailed description of the :CONFIG:TERMINALS file and a list of the recognized terminal names, see the *Guide To The iRMX II Interactive Configuration Utility.*

```
     (TDN) Terminal Device Name [1-14 Chars]        TO
```

This parameter lets you specify the physical device name of the resident user's terminal. This name must be the device unit name specified for this terminal during Basic I/O System configuration.

```
     (MTP) Maximum Task Priority [0-255]                   192
```

This parameter lets you specify the highest (numerically lowest) allowable priority among the tasks that will run on behalf of the resident user.

```
     (UID) User ID Number [0-0FFFFH]                        0H
```

This parameter specifies the user ID of the Resident/Recovery user. By specifying a value for this parameter, you ensure that a user object will be created with this user ID during system initialization. The value 0FFFFH (equivalent to 65535) represents WORLD, a special ID for which a user object is created even if you don't request it. The value 0 represents the system manager, which has no special characteristics unless you request a system manager when editing the BIOS screen of the ICU. The values 8000H through 0FFFEH (equivalent to 32768 to 65534) are reserved for use by Intel. Refer to the *iRMX II Basic I/O System User's Guide* for a discussion of users, user IDs, WORLD, and the system manager.

```
     (MIN) Minimum Memory Required [0-0FFFFFH]          02000H
     (MAX) Maximum Memory Required [0-0FFFFFH]         0FFFFFH
```

During the initialization phase of your application, the Operating System creates an HI job for the resident user's terminal and for each terminal (if any) that has an associated non-resident user that you define during non-resident user configuration. These parameters let you specify the amount of memory, in 16-byte paragraphs, that the HI job for the Resident/Recovery user's terminal is to have. The default 2000H paragraphs is equivalent to 128K (131,072 decimal) bytes, but the amount you specify depends both on the demands that will be placed on the resident user and on whether your system has only a resident user or a resident user and non-resident users.

If the Resident/Recovery user is the only user and you have set the Human Interface PMA parameter to a value other than 0FFFFFH, then set MAX to 0FFFFFH, in order to give the resident user access to all Human Interface memory. Also set MIN to a value less than the PMA value, so that the Human Interface can successfully initialize the Resident/Recovery user's I/O job. If there are non-resident users in addition to the resident user, they must all share the memory that the Human Interface has for them.

```
     (IPP) Initial-Program Pathname [Resident/1-45 Chars]
                                                        RESIDENT
```

This parameter lets you specify the initial program for the resident user. The default value "Resident" specifies the Human Interface CLI program. Any other response must be the pathname of the object code file that will be bound to the application system during configuration as the initial program for the resident user.

```
(DEF) Default Directory [1-45 Chars]
                                                     :SD:USER/SUPER
```

This parameter lets you specify the pathname of the default directory for the Resident/Recovery user. This directory will be assigned the logical name :HOME:. When code running on behalf of this user encounters a file name without a directory, the default directory that you specify here is assumed. The default value :SD:USER/SUPER for this parameter specifies a special directory on the system device. If you want to specify a different directory, or if you are changing the name of the system device (this is not recommended), make the appropriate adjustment to this parameter.

5.3.4 Prefixes

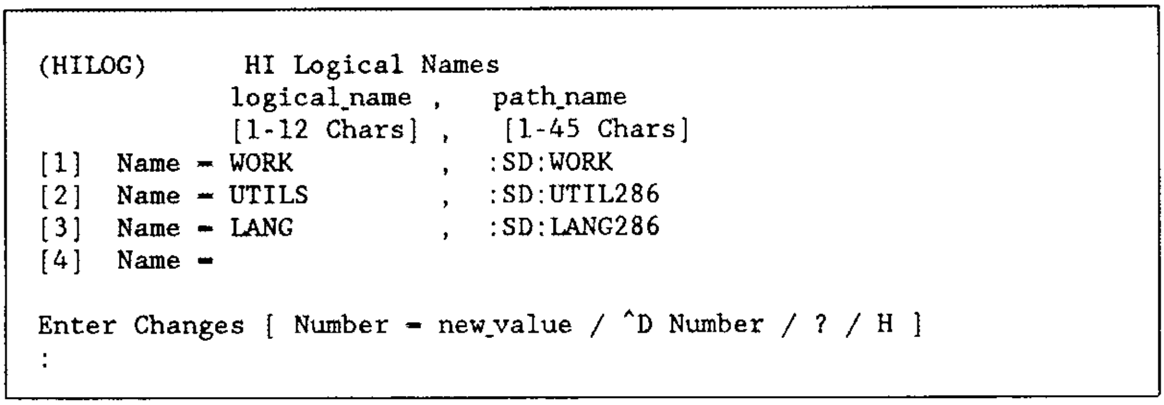
Prefixes are logical names for directories that the Human Interface searches through when an operator enters a Human Interface command without specifying a complete pathname. The ICU obtains information about prefixes from the following screen:

```
(PREF)          Prefixes

    Prefix = 1-45 characters
[1] Prefix = :PROG:
[2] Prefix = :UTILS:
[3] Prefix = :SYSTEM:
[4] Prefix = :LANG:
[5] Prefix = :$:
[6] Prefix =

Enter Changes [ Number = new_value / ^D Number / ? / H ]
:
```

You can edit this screen using the methods outlined for repetitive screens in *Guide To The iRMX II Interactive Configuration Utility* manual. For each prefix that you add, be sure to define it as a logical name via the "Logical Names Screen" screen. The prefix list can contain up to 20 prefixes.

5.3.5 Logical Names

A logical name is a nickname for a pathname and, once defined, can be used in place of its associated pathname. The ICU obtains information about logical names from the following screen:



```
(HILOG)       HI Logical Names
             logical_name ,    path_name
             [1-12 Chars] ,    [1-45 Chars]
[1]   Name - WORK         ,   :SD:WORK
[2]   Name - UTILS        ,   :SD:UTIL286
[3]   Name - LANG         ,   :SD:LANG286
[4]   Name -

Enter Changes [ Number - new_value / ^D Number / ? / H ]
:
```

You can edit this screen using the methods outlined for repetitive screens in *Guide To The iRMX II Interactive Configuration Utility*. For each logical name that you add, be sure to include the logical name, the separating comma, and the defining pathname.

:CONFIG:, :PROG:, and :SYSTEM: are automatically assigned as logical names. You must not assign them again using this screen.

CHAPTER 6
APPLICATION LOADER PARAMETERS

6.1 APPLICATION LOADER SCREEN

This chapter discusses how to select appropriate parameter values on the Application Loader Screen.

```
(APPL)          Application Loader

(ASC) All System Calls [Yes / No =only RQ$A$LOAD]        REQ
(DMP) Default Memory Pool Size [0 - 0FFFFFH]              0500H
(RBS) Read Buffer Size [0 - 0FFFFH]                       01000H

Enter  [ Abbreviation = new_value / Abbreviation ? / H ]
:
```

```
(ASC)  All System Calls [Yes/No =only RQ$A$LOAD]          REQ
```

This parameter allows you two options: "Yes" or "No". A response of "Yes" means that your system will include all the Application Loader system calls. If your system includes all the Application Loader system calls, the Extended I/O System must also be configured. A response of "No" means the system will include only the RQ$A$LOAD system call. In this case, the Application Loader does not require the EIOS. If your system includes the Human Interface, the default "REQ" appears on the screen.

```
(DMP)  Default Memory Pool Size [0 - OFFFFFH]                0500H
```

This parameter allows you to specify, in 16-byte paragraphs, the minimum memory pool that will be allocated to any I/O jobs created by the Application Loader. This parameter is used when no other minimum memory pool has been specified. The upper limit is 16M bytes.

```
(RBS) Read Buffer Size [0 - OFFFFH]                01000H
```

This parameter lets you specify the size, in bytes, of the buffer the Application Loader uses when loading data from secondary storage. Your choice will affect loading performance in two ways. First, the Application Loader loads faster if it has a large read buffer. Second, the Application Loader can load faster if its buffer is an integral multiple of the granularity of the device from which it is loading. The default value specifies a 4,096 byte read buffer.

CHAPTER 7
REMOTE FILES ACCESS PARAMETERS

7.1 INTRODUCTION

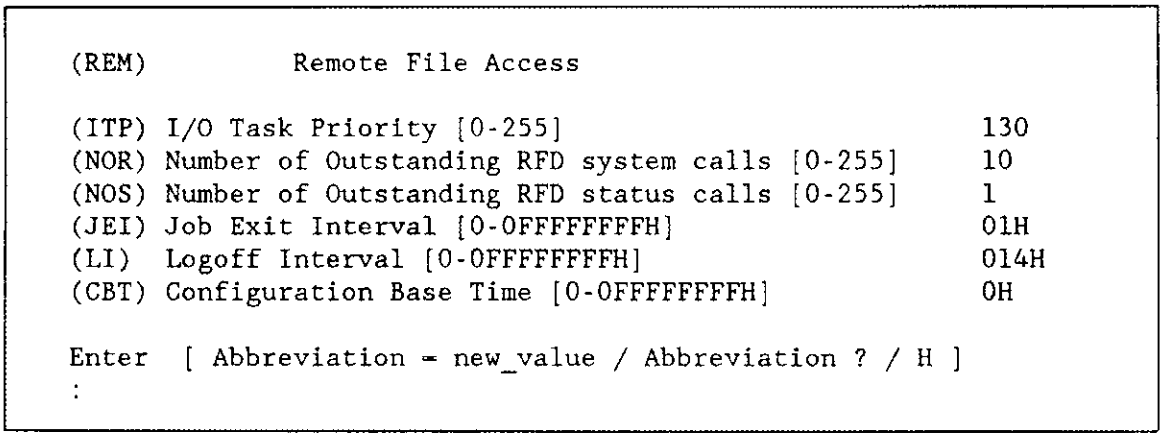
This chapter discusses how to respond to the prompts that appear on both the "Remote File Access" screen and the related "Remote Server" screen. This configuration option requires the iRMX networking software.

Remote File Access applies to iRMX Networking Software operating in a file consumer mode. The iRMX Networking Software is an optional package purchased separately from the iRMX II Operating System. If you have iRMX-NET and you want your system to access remote files (be a file consumer), answer "yes" to this parameter. If you have iRMX-NET and you want your system to supply files to remote systems (be a file server), answer "no" to this parameter. If you want your system to do both (be both a file server and a file consumer), answer "yes" to this parameter.

The Remote File Access requires the Basic I/O System; the Extended I/O System is optional. Refer to the *iRMX Networking Software User's Guide* for more information about the iRMX-NET software.

7.2 REMOTE FILE ACCESS SCREEN

This section lists the "Remote File Access" screen and describes each of the parameters on the screen.



```
(REM)         Remote File Access

(ITP) I/O Task Priority [0-255]                          130
(NOR) Number of Outstanding RFD system calls [0-255]     10
(NOS) Number of Outstanding RFD status calls [0-255]     1
(JEI) Job Exit Interval [0-0FFFFFFFFH]                   01H
(LI)  Logoff Interval [0-0FFFFFFFFH]                     014H
(CBT) Configuration Base Time [0-0FFFFFFFFH]             0H

Enter  [ Abbreviation = new_value / Abbreviation ? / H ]
:
```

```
(ITP) I/O Task Priority [0-255]                          130
```

Use this parameter to set the priority of the two RFD (Remote File Driver) service tasks created. For optimum performance, the priority of the RFD service task must be lower (numerically greater) than the initial task priority of the iRMX-NET job. The iRMX-NET job is referred to as the iRMX-NET object file in the *iRMX Networking Software User's Guide*. The default value is recommended for most systems.

```
(NOR) Number of Outstanding RFD system calls [0-255]     10
```

This parameter indicates the average number of simultaneous outstanding RFD system calls. The RFD uses this information for efficient management of system resources. The default value is recommended for most systems.

```
(NOS) Number of Outstanding RFD status calls [0-255]     1
```

This parameter indicates the average number of simultaneous outstanding RFD A$GET$CONNECTION$STATUS, A$GET$FILE$STATUS, and A$GET$EXTENSION$DATA system calls. The RFD uses this estimate to manage system resources efficiently. Unless you have a specific reason, do not change the default parameter.

```
    (JEI) Job Exit Interval [0-0FFFFFFFFH]                 01H
```

This parameter sets the length of time that the RFD waits after an individual job detaches its last connection to files residing on a remote file server. If this interval expires before the job creates subsequent connections to remote files, the RFD deletes all the resources it allocated for the job. If the job creates subsequent remote connections before the time interval expires, the job has access to the same RFD resources, without forcing the RFD to recreate them.

Because the creation of RFD resources incurs overhead, the proper setting of this parameter should reduce overhead by eliminating unnecessary sequences of deleting and recreating resources.

Specify this parameter in units of Nucleus clock intervals multiplied by 1024.

```
    (LI) Logoff Interval [0-0FFFFFFFFH]                   014H
```

This parameter sets the length of time that the RFD waits after all jobs for a single user have detached their last connection to files residing on a remote file server. If this interval expires before the user creates subsequent connections to remote files, the RFD deletes all the resources it allocated for the user. If the user creates subsequent remote connections before the time interval expires, the user has access to the same RFD resources, without forcing the RFD to recreate them.

Because the creation of RFD resources incurs overhead, the proper setting of this parameter should reduce overhead by eliminating unnecessary sequences of deleting and recreating resources.

Specify a value for this parameter in units of Nucleus clock intervals multiplied by 1024.

```
    (CBT) Configuration Base Time [-0FFFFFFFFH]              0H
```

Most systems calculate the date and time by maintaining a count of the number of seconds since some fixed point in time. The RFD uses midnight, January 1, 1978 as its fixed point. If your system uses a different fixed point, set this parameter to indicate the number of seconds from the RFD fixed point to the fixed point used by your system. RFD uses this value to adjust the creation, access, and modification times returned by the A$GET$FILE$STATUS system call.

The iRMX II Operating System uses midnight, January 1, 1978 for its fixed point in date/time calculations. Therefore, you should leave this parameter set at its default value of zero.

7.3 QUERY SCREEN

After you have completed the "Remote File Access" screen, the query screen is displayed. It contains only one line:

```
   Do you have any/more Remote Servers ?
```

Answering this prompt with a "Yes" will take you to the Remote File Server screen. Answering this prompt with a "No" will take you to the next major screen. If no servers require special parameters, specify "No".

7.4 REMOTE FILE SERVERS SCREEN

This section lists the "Remote File Server" screen and describes each of the parameters on that screen. The "Remote File Server" screen defines one remote file server at a time. After you have completed the screen, a query screen is displayed asking if you have any more remote file servers. A response of "Yes" causes the "Remote File Server" screen to be redisplayed. A response of "No" takes you to the next major screen.

When a user accesses files on a remote server, the Extended I/O System allocates buffers for the files according to the values specified on this screen. If you do not define a "Remote File Server" screen for a particular server, the RFD assumes the default values. Originally, the defaults are set at NFR=3 and MBF=255. You can change the defaults by filling out a "Remote File Server" screen and assigning a server name of DEFAULT. The reason to fill out a Remote File Server screen is if a particular Remote File Server system requires different parameters than the defaults.

```
   (REMFS)        Remote File Server

   (NAM) Name of Server [Default / 1-14 Chars]          DEFAULT
   (NFR) Number of Fragmentation Buffers [0-65535]      3
   (MBF) Maximum number of Buffers per File [0-255]     255

   Enter  [ Abbreviation = new_value / Abbreviation ? / H ]
   :
```

```
   (NAM) Name of Server  [Default/1-14 Chars]     DEFAULT
```

This parameter specifies a unique name for a particular Remote File Server. This name is set by a Remote File Server system using the iRMX-NET SETNAME command.

```
    (NFR) Number of Fragmentation Buffers [0-65535]              3
```

NOTE

iRMX-NET 2.0 ignores this parameter.

This parameter sets the number of Basic I/O System buffers that the RFD allocates to this remote file server for blocking and deblocking I/O requests. This value is analogous to the "(NB) Number of Buffers" parameter on the Device-Unit Information screens.

```
    (MBF) Maximum number of Buffers per File [0-255]           255
```

This parameter specifies the number of buffers that the Extended I/O System allocates when performing I/O operations for files on this remote server. It is analogous to the "(MB) Max Buffers" parameter on the Device-Unit Information screens. The Extended I/O System buffers should be 1000H bytes long, so set the "(IBS) Internal Buffer Size" parameter on the EIOS screen accordingly.

The default value of 255 allows the S$OPEN system call to specify the actual number of Extended I/O System buffers.

If the applications that use remote files are computationally intense (such as compilers and assemblers), set the MBF parameter to 255. If the applications are I/O intensive (copying files, for example), set the MBF parameter to 0. If the applications are mixed, choose 255.

CHAPTER 8
EXTENDED I/O SYSTEM PARAMETERS

8.1 INTRODUCTION

Five screens define the Extended I/O System parameters. They are the "EIOS" Screen, "Automatic Boot Device Recognition Screen", "Logical Names" Screen, "I/O Users" Screen, and "I/O Jobs" Screen. This chapter discusses how to respond to the prompts that appear on these screens. If you are using this chapter to understand a parameter line, turn to the page number indicated to the right of the screen that contains the parameter line abbreviation.

EIOS Parameters

(ABR)	(IBS)	(DDS)	(ITP)	Start on
(PMI)	(PMA)	(CD)		page 8-2

Automatic Boot Device Recognition Parameters

(DLN)	(DPN)	(DFD)	(DO)	Start on
				page 8-4

Logical Names Parameters

Start on
page 8-6

I/O Users Parameters

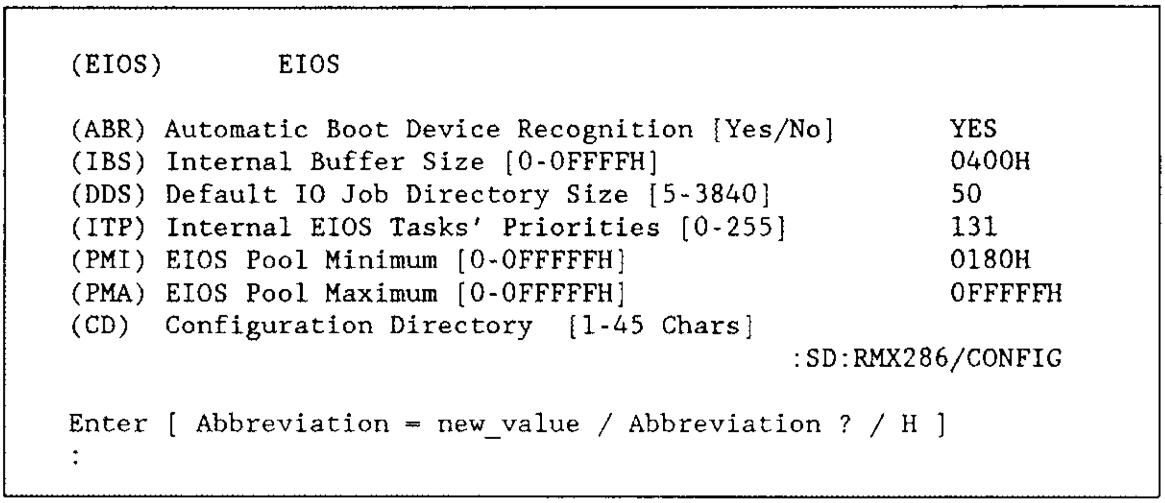
Start on
page 8-7

I/O Jobs Parameters

				Start on
(IJD)	(DU)	(PMI)	(PMA)	page 8-8
(EHS)	(EM)	(PV)	(TP)	
(TSA)	(VAR)	(SSA)	(SSI)	
(NPX)				

8.2 EIOS SCREEN

The ICU obtains general information about the Extended I/O System from the following screen.



```
(EIOS)      EIOS

(ABR) Automatic Boot Device Recognition [Yes/No]       YES
(IBS) Internal Buffer Size [0-0FFFFH]                  0400H
(DDS) Default IO Job Directory Size [5-3840]           50
(ITP) Internal EIOS Tasks' Priorities [0-255]          131
(PMI) EIOS Pool Minimum [0-0FFFFFH]                    0180H
(PMA) EIOS Pool Maximum [0-0FFFFFH]                    0FFFFFH
(CD)  Configuration Directory  [1-45 Chars]
                                             :SD:RMX286/CONFIG

Enter [ Abbreviation = new_value / Abbreviation ? / H ]
:
```

```
(ABR) Automatic Boot Device Recognition [Yes/No]           YES
```

This parameter lets you specify whether your application system is to incorporate the Automatic Boot Device Recognition (ABDR) feature. If you load your system using the Bootstrap Loader with this feature, your system can ascertain the device from which it was loaded (the bootstrap device) and automatically assign the bootstrap device as your system device.

A response of "Yes" to this parameter causes the "Automatic Boot Device Recognition" screen to be displayed (see following screen). If you specify "No" to this parameter, the next screen displayed will be the "Logical Names" screen. If you specify "No" to this parameter and need a system device, you must configure the system device on the EIOS "Logical Names" screen.

When you load the system using the Bootstrap Loader, the following series of events occurs:

1. The Nucleus receives the boot device name from the Bootstrap Loader. This is the device name you supplied via a %DEVICE macro when configuring the Bootstrap Loader. (Configuring the Bootstrap Loader is usually performed in conjunction with configuring your monitor and is a separate process from configuring your application system.)
2. The Nucleus attempts to catalog the device unit name (physical name) of the boot device under the name RQBOOTED in the object directory of the root job.

3. The EIOS looks up RQBOOTED in the root job's directory and tries to attach the physical name it finds in RQBOOTED under the logical name you specify for the "(DLN) Default System Device Logical Name" parameter on the Automatic Boot Device Recognition.

For systems booted on flexible disk drives, the disk must have a "standard" format. See the *iRMX II Device Drivers User's Guide* for information on the "standard" disk format.

If you did not configure a device unit during BIOS configuration with the name the EIOS finds under RQBOOTED, then no system device is attached. Therefore, if you select Automatic Boot Device Recognition, configure the Bootstrap Loader using at least one of the names you used when configuring your device drivers.

Even though the device names must be the same, the device name you specify in the device driver configuration does not need to exactly describe the device you wish to boot from. It only needs to have valid values. You may see an example of this naming by examining the values associated with the "W0" device defined in the MSC Driver configuration screens of all Intel-supplied definition files. "W0" in this case is a "generic" device name, capable of establishing any formatted Winchester drive as the system device.

Even if you select the Automatic Boot Device Recognition feature, there is a possibility that the system will not be able to ascertain the bootstrap device. When you download the system from a Series IV by means of the iSDM monitor, the Bootstrap Loader does not pass the Nucleus the name of the bootstrap device. Since the Nucleus does not have a device unit name to catalog in the root job's directory, the EIOS will not be able to look up the physical name and device recognition will not be possible.

```
    (IBS) Internal Buffer Size [0-0FFFFH]                     0400H
```

This parameter lets you specify the size, in bytes, of the buffers the EIOS uses when performing I/O operations. When your application system opens a connection to a file, the EIOS creates buffers equal in size to the largest integral multiple of the device granularity that does not exceed the value you specify for this parameter.

You may specify any value in the indicated range. However, if you have selected the Human Interface, you must not specify a value larger than the default 0400H, which is equivalent to 1024 bytes.

```
    (DDS) Default IO Job Directory Size [5-3840]                50
```

This parameter lets you specify the capacity (number of entries) of the object directories for the EIOS job and all I/O jobs. The default value of 50 entries is large enough for most systems. However, because I/O jobs can communicate only via the object directory of a common ancestor job, your system might require this value to be larger if extensive communication between jobs is desired.

If you have configured the Human Interface into your system, DDS should include at least 25 entries for system use only.

```
    (ITP) Internal EIOS Tasks Priorities [0-255]            131
```

The EIOS creates tasks that perform housekeeping functions. This parameter lets you specify the priority of those (internal) tasks. By carefully selecting the value of this parameter, you can help ensure the maximum performance of your application system. In particular, the value you specify should not be lower (numerically higher) than the priority of any internal BIOS task. The default value is 131 decimal.

```
    (PMI) EIOS Pool Minimum [0-0FFFFFH]                        0180H
    (PMA) EIOS Pool Maximum [0-0FFFFFH]                      0FFFFFH
```

These parameters specify the minimum and maximum sizes, in 16-byte paragraphs, of the EIOS memory pool. Memory pools can be up to 16M bytes in size. The minimum value serves also as the initial size of the pool. If the maximum value exceeds the minimum value, the EIOS can borrow memory from the root job memory pool. To prevent borrowing, set the minimum and maximum values equal to each other. The minimum value may not exceed the maximum value.

```
    (CD) Configuration Directory  [1-45 Chars]
                                                     :SD:RMX286/CONFIG
```

This parameter provides the pathname for the directory which contains non-resident user configuration information. Each non-resident user must be defined before gaining access to the system. The files used to define these users are :CONFIG:TERMINALS, :CONFIG:UDF, and :CONFIG:USER/user-name. The names of these files never change and are located on the System device in the directory defined by the logical name :CONFIG:. The pathname you specify here is for the directory used to configure and verify terminals and non-resident users. The Human Interface catalogs this directory as :CONFIG:.

8.3 AUTOMATIC DEVICE RECOGNITION SCREEN

If you responded "Yes" to the "(ABR) Automatic Boot Device Recognition" parameter line in the "EIOS" screen, the ICU displays this screen. This screen lets you define one of your devices as the default system device in case the EIOS is unable to find an entry in the root job's directory under the name RQBOOTED. The absence of this name indicates that the system was not loaded by the Bootstrap Loader.

The following screen makes it possible to enter the information required by the Automatic Boot Device Recognition feature.

```
(ABDR)          Automatic Boot Device Recognition

(DLN) Default System Device Logical Name [1-12 Chars]              SD
(DPN) Default System Device Physical Name [1-12 Chars]             WO
(DFD) Default System Device File Driver [Phys/Str/Named/Remote] NAMED
(DO)  Default System Device Owner's ID [0-0FFFFH]                  0H

Enter [ Abbreviation = new_value / Abbreviation ? / H ]
:
```

```
    (DLN) Default System Device Logical Name [1-12 Chars]        SD
```

This parameter line lets you specify the logical name of the system device. See the description of the Automatic Boot Device Recognition parameter for more information.

Intel recommends that you do not change this parameter.

If you do change the value of this parameter, limit your new_value to 12 characters, each of which must have an ASCII value in the 20H to 7EH range and none of which can be a colon (3AH), a slash (2FH), or an up-arrow or circumflex (both 5EH, depending on your terminal).

NOTE

Do not use the name you specify for this parameter as the logical name of a physical device in the "Logical Names" screen. For the system device, the association between logical name and device is established in this screen, not the "Logical Names" screen.

```
    (DPN) Default System Device Physical Name [1-12 Chars]       WO
```

This parameter lets you specify the physical (device unit) name. The Operating System makes this device the system device in the event that the EIOS is unable to find a physical name cataloged under the name RQBOOTED in the object directory of the root job. See the description of the Automatic Boot Device Recognition parameter for more information.

NOTE

Do not use the name you specify for this parameter as the name of a physical device in the "Logical Names" screen. For the system device, the association between logical name and device is established in this screen, not the "Logical Names" screen.

```
(DFD) Default System Device File Driver [Phys/Str/Named/Remote] NAMED
```

This parameter lets you specify the file driver (Physical, Stream, Named, or Remote) for the system device. See the description of the Automatic Boot Device Recognition parameter for more information about Automatic Boot Device Recognition. See the *iRMX II Extended I/O System User's Guide* for more information about file drivers.

```
    (DO)  Default System Device Owner's ID [0-0FFFFH]             0H
```

This parameter lets you specify the owner ID of the system device. The default value for this parameter is 0H, signifying that the system manager (user ID 0) is the owner of the device.

8.4 LOGICAL NAMES SCREEN

The following screen lets your define some initial logical names for your devices. During system operation, you can change these names or add more names, but these are available when the Operating System is initialized.

```
(LOGN)      Logical Names

Logical Name = logical_name,  device_name,  file_driver, owners-id
               [1-12 Chars],   [1-14 Chars],   [P/S/N/R],  [0-0FFFFH]
[1] Logical Name = BB       , BB              ,    PHYSICAL,    0H
[2] Logical Name = STREAM   , STREAM          ,    STREAM  ,    0H
[3] Logical Name = LP       , LP              ,    PHYSICAL,    0H
[4] Logical Name =

Enter Changes [Number = new_value / ^D Number / ? / H ]
:
```

You can edit this screen by the methods outlined for repetitive screens in Chapter 1 of the *Guide to the iRMX II Interactive Configuration Utility*. For each logical name that you add to the logical name list, be sure to include the logical name, the device-unit name, the file driver that supports the device, and the user ID of the owner of the device connection for the device, as follows:

- Limit the logical name to 12 characters, each of which must have an ASCII value in the 20H to 7EH range and none of which may be a colon (3AH), a slash (2FH), or an up-arrow or circumflex (both 5EH, depending on your terminal).
- For the device name, use the device unit name of the device as it is (or will be) known to the ICU for device configuration purposes. Limit the name to 14 characters.
- Specify the file driver as Physical, Stream, Named, or Remote, depending on the file structure of the device and how the device will be accessed by the EIOS.
- Specify an I/O user ID value. This value need not be defined on the "I/O Users" screen of the EIOS.

You may specify as many as 20 logical names on this screen, including the two that are already defined there.

8.5 I/O USERS SCREEN

The EIOS uses a type of object called a user object to determine access rights to files. Each I/O user (user object) must have both a user name (any string up to 12 characters in length) and from one to five IDs (any 16-bit value). You may specify up to 20 I/O users by entering their user names and user IDs. The first ID that you enter for each user is the Owner ID. Subsequent IDs, separated by commas from the Owner ID, define additional access rights of the I/O user.

Every I/O job must have a default user object before any of its tasks can successfully use system calls that check access rights. When creating an I/O job, the EIOS catalogs the I/O job's default user in the object directory of that I/O job (under the name R?IOUSER). The ICU must have information about the I/O users that are default users for the initial I/O jobs of the system. The ICU obtains information about initial I/O jobs via the "I/O Jobs" screen (described later in this chapter). It obtains information about I/O users via the following screen:

```
(IOUS)          I/O Users

    I/O User =    user name,        Owner-ID[,ID,ID,ID,ID]
                 [1-12 Chars],      [0-OFFFFH]
[1] I/O User = WORLD        , OFFFFH, OFFFFH, OFFFFH, OFFFFH
[2] I/O User =

Enter Changes [ Number= new_value / ^D Number / ? / H ]
:
```

You can edit this screen by the methods outlined for repetitive screens in Chapter 1 of the *Guide to the iRMX II Interactive Configuration Utility.* For each I/O user that you add to the I/O user list, be sure to include the name, which can have up to 12 characters, and the owner ID, which is a 16-bit value. You may also include up to four additional IDs as well, provided that they are separated by commas.

8.6 QUERY SCREEN

After you have completed the "I/O Jobs" screen, the query screen is displayed. It contains only one line:

```
Do you need any/more I/O Jobs?
```

Respond "Yes" if you have another I/O user job. A response of "Yes" causes the "I/O Jobs" screen to be displayed again. The default value is "No". There is no limit on the number of I/O jobs you can create.

The order in which you define your I/O user jobs is also the order in which the EIOS task initializes the jobs.

8.7 I/O JOBS SCREEN

An application system, immediately after initialization, includes at least one application job. Each initial application job is either a first-level job or an initial I/O job. An initial I/O job is a child of the EIOS job and, as an I/O job, can invoke EIOS system calls. I/O jobs should be initial I/O jobs if they must be resident in memory (rather than having to be loaded from secondary storage), if they must begin running at system startup time, or if they must be independent of the Human Interface.

I/O job initialization occurs asynchronously to the rest of the system initialization. The ICU obtains information about each of your initial I/O jobs by means of the following screen. The order in which you specify your initial I/O jobs is the order in which the Operating System creates them.

```
(IOJOB)     I/O Jobs

(IJD) I/O Job Default Prefix  [1-12 Chars]
(DU)  Default User [1-12 Chars]
(PMI) Pool Minimum [20H - 0FFFFFH]                 0260H
(PMA) Pool Maximum [20H - 0FFFFFH]                 0FFFFFH
(EHS) Exception Handler Entry Point [1-31 Chars]

(EM)  Exception Mode [Never/Prog/Environ/All]      NEVER
(PV)  Parameter Validation [Yes/No]                YES
(TP)  Task Priority [0-255]                        155
(TSA) Task Entry Point  [1-31 Chars]

(VAR) Public Variable Name [1-31 Chars]

(SSA) Stack Segment Address [SS:SP]                0000:0000H
(SSI)  Stack Size [0-0FFFFH]                       0300H
(NPX) Numeric Processor Extension Used [Yes/No]   NO

Enter  [ Abbreviation = new_value / Abbreviation ? / H ]
:
```

```
(IJD) I/O Job Default Prefix [1-12 Chars]
```

This parameter specifies the logical name that is to be the default prefix for this I/O job. This is the prefix that is assumed when a task in this I/O job makes a system call that requires a device name, without specifying a device. The value you specify can be either the logical name of the system device or any of the logical names defined via the "Logical Names" screen of the EIOS.

```
(DU)  Default User [1-12 Chars]
```

This parameter lets you specify the default user object for this I/O job. This is the user object that is assumed when a task in this job, without specifying a user object, makes a system call that requires a user object as a parameter. The value you specify can be any of the user object names defined via the "I/O User" screen.

```
    (PMI) Pool Minimum [20H - 0FFFFFH]                 0260H
    (PMA) Pool Maximum [20H - 0FFFFFH]               0FFFFFH
```

These parameters let you specify the minimum and maximum sizes, in 16-byte paragraphs, of this I/O job's memory pool. Memory pools can be up to 16M bytes in size. The minimum value serves also as the initial size of the pool. If the maximum value exceeds the minimum value, then this job may borrow memory from the EIOS memory pool, with the maximum pool size limiting the amount of memory that can be borrowed at any given time. To prevent borrowing, set both the minimum and maximum values equal to the desired constant memory pool size. Whether you allow borrowing or not, the minimum value must not exceed the maximum value. The default maximum means that there is no upper limit on how much memory can be borrowed.

If the "Stack Segment Address" parameter has a value of 0:0, set the "Pool Minimum" parameter line value to 32 (decimal) plus the value specified on the "Stack Size" parameter line. (The value specified on the "Stack Size" parameter must first be converted from the number of bytes to the number of 16-byte paragraphs.)

```
    (EHS) Exception Handler Entry Point  [1-31 Chars]
```

An I/O job can inherit the exception handler of its parent job. In this case, the exception handler that would be inherited is the default system exception handler that you specified for the Nucleus. If you want to use the default exception handler, leave this parameter line blank. If you want to use a handler other than the default system exception handler, this parameter allows you to specify the PUBLIC name of the other exception handler.

The name you enter for this parameter defines the PUBLIC name of the exception handler procedure. When binding your I/O job, you must include a NOPUBLICS EXCEPT (xxx) control, where xxx is the entry point of the I/O Job. Do not use the letters "RQ" to begin PUBLIC names.

```
    (EM)  Exception Mode [Never/Prog/Environ/All]        NEVER
```

If you specified a PUBLIC name for the "(EHS) Exception Handler Entry Point" parameter line (indicating that you do not want the default system exception handler), you must specify the exception mode of the exception handler specified for this I/O job.

Choose one of the following four options:

Never — Control never passes to the exception handler.

Programmer — Control passes to the exception handler on programmer errors only.

Environment — Control passes to the exception handler on environmental conditions only.

All — Control passes to the exception handler on all exceptional conditions.

```
    (PV)  Parameter Validation [Yes/No]               YES
```

This parameter line lets you specify a "Yes" if you want the Nucleus to perform parameter validation for all Nucleus system calls made by tasks in this I/O job. Specify a "No" if you do not want the Nucleus to validate parameters for tasks in this I/O job.

If your I/O job includes tasks that invoke Basic I/O System system calls, respond to this parameter line with a "Yes".

Explanations for not including parameter validation at either a system-level or at an I/O job level are given in Chapter 12. Note that the ICU ignores your response to this parameter line if you have not included system-level parameter validation.

```
    (TP)  Task Priority [0-255]                          155
```

This parameter line lets you specify the priority of this job's initialization task. A value of zero assigns the initialization task a priority equal to the priority of its parent.

When created, each I/O job contains only a single task. That single task creates or starts the creation of all other objects required by the I/O job. Thus, this task is referred to as the initialization task for its job, even though it may perform other functions as well.

A task's priority is an integer value between 0 and 255, inclusive. The lower the priority number, the higher the priority of the task. Unless a task is involved in processing interrupts, its priority should be between 129 and 255. The value of this parameter must not be numerically smaller than the maximum priority for the job.

```
    (TSA) Task Entry Point  [1-31 Chars]
```

The name you enter for this parameter is the PUBLIC name of your I/O job's entry procedure. When binding your I/O job, you must include a NOPUBLICS EXCEPT(xxx) control, where xxx is the entry point of your task. The pathname of the object file (produced by compiling and binding your I/O job) containing this procedure must also be specified on the "User Modules" screen.

```
    (VAR)  Public Variable Name [1-31 chars]
```

This parameter lets you enter the PUBLIC name of any one of your variables. The Nucleus uses this name to obtain the selector of the I/O job's data segment. When binding your I/O job, you must include a NOPUBLIC EXCEPT(xxxx) control, where xxxx is the entry point of the I/O job.

```
    (SSA) Stack Segment Address [SS:SP]             0000:0000H
```

This parameter line lets you specify the address of the initialization task's stack. A value of 0:0 causes the Nucleus to allocate a stack segment to the task and initialize the SS register to the selector of this segment and the SP register to the value of the stack size prompt.

It is recommended that you specify 0:0 for this parameter. This permits dynamic stack allocation and deallocation.

```
    (SSI)  Stack Size [0-0FFFFH]                         0300H
```

This parameter line lets you specify the size in bytes of the initialization task's stack segment. This stack size must be at least 16 bytes. The Nucleus increases specified values that are not multiples of 16 up to the next higher multiple of 16. The size should be at least 300 (decimal) bytes if the initialization task is going to make Nucleus system calls. The value you specify depends on which model of segmentation you use.

PL/M-286 Medium Models

Procedures compiled using the MEDIUM model require statically allocated stacks. Thus, for these procedures, you must specify the stack segment address for the I/O job. The size you specify for the "Stack Size" parameter line should be the same size you specified in the SEGSIZE(STACK(...) control of the BND286 command. Refer to the "Binding and Building Application Jobs" section of Chapter 3 in the *Guide to the iRMX II Interactive Configuration Utility* for more information on the BND286 command.

```
    (NPX) Numeric Processor Extension Used [Yes/No]       NO
```

You must respond to this parameter with a "Yes" if the I/O jobs initial task contains floating-point instructions. You do not have to respond with a "Yes", however, if an offspring and not this I/O job contains a task that uses floating-point instructions.

If you respond "Yes" to this parameter, the priority of the initial task can not be high enough to mask the interrupt level that you specified for the "NPX Interrupt Level" parameter line in the "Hardware" screen. Refer to Chapter 1 for more information about numeric processor extensions.

CHAPTER 9
BASIC I/O SYSTEM PARAMETERS

9.1 INTRODUCTION

This chapter discusses how to respond to the prompts that appear on the BIOS (Basic I/O System) screen. There are two screens for defining the Basic I/O System directly - the "BIOS" screen and the "BIOS System Calls" screen. There are also screens for defining the device drivers supported by the BIOS. The device driver screens are defined in Chapter 10.

9.2 BIOS SCREEN

The ICU obtains general information about the BIOS from the "BIOS" screen. The following screen shows the values from the 28612.def definition file.

```
(BIOS)          BIOS

(ASC) All Sys Calls in BIOS [Yes/No]                REQ
(ADP) Attach Device Task Priority [1-255]           129
(TF)  Timing Facilities Required [Yes/No]           REQ
(TTP) Timer Task Priority [0-255]                   129
(CON) Connection Job Delete Priority [0-255]        130
(ACE) Ability to Create Existing Files [Yes/No]     REQ
(SMI) System Manager ID [Yes/No]                    REQ
(CUT) Common Update Timeout [0-65535]               1000
(CST) Control-Sequence Translation [Yes/No]         YES
(OSC) Terminal OSC Controls [Yes/No]                REQ
(TS)  Tape Support [Yes/No]                         YES
(PMI) BIOS Pool Minimum [0-0FFFFFH]                 0D00H
(PMA) BIOS Pool Maximum [0-0FFFFFH]                 0FFFFFH
(GC)  Global Clock [546/CSM/None]                   NONE
(GCN) Global Clock Name [1-13 Chars]                T546_CC

Enter  [ Abbreviation = new_value / Abbreviation ? / H ]
:
```

```
(ASC) All Sys Calls in BIOS [Yes/No]                REQ
```

The default value indicates that the system you defined on the "Sub-system" screen requires all the Basic I/O System system calls. Without changing your responses on the "Sub-system" screen, you cannot change the default value on the "BIOS" screen.

If your "Sub-system" screen includes either the EIOS or the Application Loader, a "REQ" will appear as the default value for this parameter. These subsystems require all the BIOS system calls.

If you specify "No" to this parameter line, the ICU displays the "BIOS System Calls" screen. This screen allows you to select the system calls you wish to include in your system. Note, however, that the ICU always includes the following Basic I/O System system calls: A$PHYSICAL$ATTACH$DEVICE, A$PHYSICAL$DETACH$DEVICE, CREATE$USER, DELETE$USER, GET$DEFAULT$USER, INSPECT$USER, SET$DEFAULT$USER, ENCRYPT, and WAIT$IO.

```
    (ADP) Attach Device Task Priority [1-255]         129
```

This parameter line lets you specify the priority of the attach-device task. The values allowed for this parameter range from 1 to 255. The default value is the highest priority allowed for a non-interrupt task. If you change the priority, it is recommended that you select a priority that is higher (numerically lower) than the priorities of user tasks which perform I/O and lower than the priorities of high-priority tasks that do not use the Basic or Extended I/O Systems. If you have no tasks that run at high priorities, leave this parameter set to the default.

The attach-device task is one of a number of Basic I/O System tasks that perform the services of the BIOS. The Basic I/O System invokes the attach-device task whenever tasks call A$PHYSICAL$ATTACH$DEVICE. The attach-device task receives all requests to attach devices. For each request, it creates another task which actually attaches the device and services the device until the device is detached. Although the priority of the attach-device task is configurable, the priority of the second task is always 130.

```
    (TF)  Timing Facilities Required [Yes/No]         REQ
```

This parameter line lets you specify whether you want your I/O system to include timing facilities. If your system includes the Human Interface, a "REQ" will appear as the default parameter value.

If you respond "Yes", timing facilities will be included in your I/O system. If you respond "No", the Basic I/O System fills in all time fields with a zero value and saves the overhead of maintaining a timer. A "No" response also means that any value you specify to the "Timer Task Priority" and "Common Update Timeout" parameters will be ignored by the system. (These particular parameters are described in this chapter.) If you exclude timing facilities, you automatically exclude the GET$TIME and SET$TIME system calls from the Basic I/O System.

```
    (TTP) Timer Task Priority [0-255]                 129
```

If you specified that your system includes timing facilities, you must specify the priority of the timer task.

The priority you set affects the performance of the time-of-day clock for the Basic I/O System. If you set the priority of the timer task too low, the timer task might not get to run as often as it needs and the clock will run slow. If you set the priority too high, the timer task might take machine cycles away from high priority tasks.

It is recommended that you use the default value initially and adjust the priority up or down later as your situation dictates. Adjust the priority in increments of 1. Although the ICU allows values from 0 to 255, it is recommended that your response be between the value of your lowest interrupt task and your highest non-interrupt task.

```
    (CON) Connection Job Delete Priority [0-255]      130
```

This parameter line lets you specify the priority of the Basic I/O System task which deletes file and device connections. The priority of the connection deletion task should be high enough to allow job deletion to proceed without delay. The default is the same as the default priority of the device service tasks. This value is sufficient for most applications.

Before the Nucleus can delete a job, all objects contained in the job must be deleted. The Nucleus can delete most objects by itself, but it sends composite objects (such as connections) to a mailbox where the type manager (in this case, the Basic I/O System) must delete them. The Basic I/O System contains a task that waits at the deletion mailbox to delete connections. The "Connection Job Delete Priority" parameter specifies the priority of this task.

```
    (ACE) Ability to Create Existing Files [Yes/No]      REQ
```

If your system includes the UDI sub-system, a "REQ" appears as the default parameter value.

Specify "Yes" if you want to respond "FALSE" to the "must$create" parameter of the A$CREATE$FILE system call. Specify "No" if you want the Operating System to return an error condition whenever the creation of an existing file with a duplicate pathname is attempted.

The A$CREATE$FILE system call allows you to specify what you want it to do in situations where an attempt has been made to create a new file with the pathname of an existing file. The "must$create" parameter of the A$CREATE$FILE system call allows you to specify one of two things in the situation described. (Refer to the *iRMX II Basic I/O System Calls Reference Manual* for more information about these options.)

To allow the Operating System to create a temporary file (for an existing directory or devicename) when you specify FALSE for the "must$create" parameter, you must specify "Yes" to the "Ability to Create Existing Files" parameter.

If you intend running compilers or editors on your system via the Universal Development Interface (UDI), you must respond "Yes" to the "(ACE) Ability to Create Existing Files" parameter.

```
    (SMI) System Manager ID [Yes/No]                          REQ
```

If your system includes the Human Interface, a "REQ" appears as the default parameter value.

Specify "Yes" if you want your user ID 0 to have system manager attributes. These attributes include the following:

- Read access to all data files.
- Change access to all files.

Specify "No" if you do not have the Human Interface in your system and you wish to prevent user 0 from having the capabilities normally associated with the system manager. (Changing a "Yes" response to a "No" response does not require that you change any other response in the ICU.)

```
    (CUT) Common Update Timeout [0-65535]             1000
```

If you specified "Yes" to the "(TF) Timing Facilities Required" parameter, you must specify the time interval, in milliseconds, that the Basic I/O System waits before updating devices with data buffered in memory. The default value is 1000 decimal. You must use a value that is a multiple of whole seconds; that is, 100, 200, 300. A value of 65535 indicates that the common update-timeout feature will not be used.

You can specify update-timeout values in two places: in the "BIOS" screen and in the "Device-Unit Information" screen for the individual device-units. Your response to the "(RUT) Request Update Timeout" parameter on a "Device-Unit Information" screen applies only to an individual device-unit. That value specifies a time interval which, if no device activity occurs, the Basic I/O System waits before automatically updating the device-unit with information stored in its buffers.

Your response to the "(CUT) Common Update Timeout" (entered on the "BIOS" screen) differs from the individual request timeout values in two ways. First, it applies to all designated devices in your system. Second, it is a fixed timeout value that the Basic I/O System uses regardless of the activity on any device-unit. Therefore, the common update-time value specifies a time interval which the Basic I/O System waits, regardless of device activity, before automatically updating all attached files on designated device-units. This process ensures that all open files on designated devices are updated on a regular basis. You designate which devices will use this feature by specifying "True" to the "Common Update" parameter on the appropriate "Device-Unit Information" screens.

When deciding on a common update-timeout value, you will be trading off system performance against the data integrity you gain should your system be shut down. The larger the common update-timeout value, the less often the Basic I/O System updates the designated devices. This implies higher performance. It also implies a greater amount of data loss should your system fail during the middle of an operating session.

If you use the common update-timeout feature in conjunction with the per-unit update timeout feature, you should generally specify a value for the "(CUT) Common Update Timeout" parameter that is larger than the per-unit timeout values you specify for the individual "(RUT) Request Update Timeout" parameters on the "Device-Unit Information" screen. However, if you set the per-unit timeout values to 0FFFFH (no updating), you will make your performance/data-integrity tradeoff based solely on the common update-timeout value.

```
    (CST) Control-Sequence Translation [Yes/No]       YES
```

Specify "Yes" if you want the translation capabilities supported by the iRMX II Terminal Support Code. Otherwise, specify "No".

The Basic I/O System supports control-sequence translation through a group of modules that make up the Terminal Support Code. If you specify "Yes" to the "Control-Sequence Translation" parameter, the ICU includes the translation code (between 2K and 3K bytes of code) into the Basic I/O System. If you specify "No", the ICU omits this code.

You should answer "Yes" to this parameter if you require the ability to write programs that are independent of individual terminals. If you answer "No" to the "Control-Sequence Translation" parameter, some of your programs might run on only one kind of terminal, depending on the kinds of I/O your programs perform.

If you answer "Yes" to the "Control-Sequence Translation" parameter, you should also specify the Operating System Command (OSC) control sequences in the unit information screen for your terminal driver (your response to the "OSC Controls" parameter on the terminal driver unit information screen must be "Both", "Input", or "Output"). Refer to Chapter 10 for additional information on how to respond to terminal driver unit information screens. Refer to the *iRMX II Device Drivers User's Guide* for additional information on OSC controls.

```
    (OSC) Terminal OSC Controls [Yes/No]            REQ
```

Specify "Yes" if any of your terminals need Operating System Command (OSC) controls. That is, if you want to specify "Both", "Input", or "Output" on a terminal driver unit information screen, you first need to specify "Yes" to this BIOS parameter line. The default value is "REQ" if your system includes the Human Interface.

Specify "No" if your system does not include the Human Interface or if you do not have any terminals that require OSC controls.

```
    (TS)  Tape Support [Yes/No]                          YES
```

Specify "Yes" if you intend to use a magnetic tape cartridge controller (iSBX 217C on a iSBC 215G or an iSBC 214 controller). The tape-related configuration screens are grouped with the configuration screens for the MSC device drivers.

```
    (PMI) BIOS Pool Minimum [0-0FFFFFH]                 0D00H
    (PMA) BIOS Pool Maximum [0-0FFFFFH]                0FFFFFH
```

These parameter lines let you specify both the minimum and maximum allowable size of the Basic I/O System's memory pool (in 16-byte paragraphs). The default value for pool minimum, 0D00H, is equivalent to 53248 decimal bytes. The default value for pool maximum is equivalent to 16M bytes, the full capacity of the iRMX II Operating System.

The "BIOS Pool Minimum" and "BIOS Pool Maximum" parameters define the size of your Basic I/O System memory pool. It is recommended that you set the minimum to a value that reflects the memory needed by the BIOS for its buffers and device drivers (see below for guidelines on calculating this figure). This approach is particularly important if your application system includes the Human Interface and you set the Human Interface maximum memory pool size to 0FFFFFH, allowing it to use all remaining memory after system initialization. If the Basic I/O System does not claim its full memory pool at initialization, the jobs initialized after the Basic I/O System (the Extended I/O System, Application Loader, UDI, first-level jobs, and Human Interface) will claim the memory the Basic I/O System needs later.

The minimum memory pool requirement of the Basic I/O System is 1C0H 16-byte paragraphs of memory. However, the minimum memory pool size is not large enough to include the buffers needed to support terminals or mass storage devices. These buffers (which you specify for each Intel device driver) must be included in the BIOS minimum memory pool size.

To add support for terminals and other device drivers, use the 1C0H value as a starting point and increase the memory pool size using the following guidelines:

- If your system includes the 8251A terminal driver, add 11CH paragraphs to your memory pool.
- If your system includes an iSBC 544A terminal driver, add 250H paragraphs to the memory pool for each driver.
- If your system includes a Terminal Communications Controller driver, add 600H paragraphs to the memory pool for each driver.
- If your system includes an 8274 terminal driver, add 153H paragraphs to the memory pool for each driver.
- If your system includes an 82530 terminal driver, use the following formula to calculate the addition to the memory pool:

  paragraphs = 49H + round-up(number of units/2) x 10CH

  where round-up is a function that rounds any fractional number up to the next greater integer.
- For each attached terminal, add an additional 40H paragraph.

- For other device drivers, you must calculate the memory pool requirements for each device-unit in your system. To do that, calculate the memory requirements for each DUIB (Device Unit Information Block) in your system. Then compare the numbers for the DUIBs that correspond to the same device-unit (have the same device and unit numbers).

  Take the largest number and add it to the Basic I/O System's memory pool requirement. Do this for each device-unit in your configuration.

  To calculate the memory pool requirements of a DUIB, use the following formula:

  paragraphs = 64H + (round-up((1CH + dev$gran)/10H) x (num$bufs))

  Where:

  paragraphs — Memory requirements of the DUIB in 16-byte paragraphs.

  round-up — Function which rounds any fractional number up to the next greater integer. For example:

  round-up(1.3) = 2

  dev$gran — The device granularity you specify in the DUIB screen.

  num$bufs — The number of buffers you specify in the DUIB screen.

```
(GC)  Global Clock [546/CSM/None]                NONE
```

This parameter specifies whether or not your system has an on-board, hardware time-of-day clock. These clocks are battery-powered and continue to keep time even when the system is turned off, Intel refers to these clocks as "global clocks". The global clock maintains the current date and time for the entire system, and is accessed during system initialization or upon operator request.

For the second and third releases of the iRMX II Operating System, only the iSBC 546 global clock is supported.

```
(GCN)  Global Clock Name [1-13 chars}         T546_CC
```

This parameter specifies a name that uniquely identifies the global clock name to the Basic I/O System. Unless you have a specific reason do not change the default.

9.3 BIOS SYSTEM CALL SCREEN

If you specified "No" to the "All Sys Calls in BIOS" parameter on the "BIOS" screen, the ICU displays the "BIOS System Calls" screen. The screen consists of three parameter lines and lets you choose the type of system calls you wish to include in your system.

```
(BCALL)        BIOS System Calls

(PSC) Physical File System Calls [Yes/No]              YES
(SSC) Stream File System Calls [Yes/No]                YES
(NSC) Named File System Calls [Yes/No]                 YES

Enter  [ Abbreviation = new_value / Abbreviation ? / H ]
:
```

```
(PSC)  Physical File System Calls [Yes/No]             YES
```

A response of "Yes" to this parameter means you wish to select all the BIOS system calls needed to support physical files. The following list includes the BIOS system calls that comprise the physical file group.

Create File
Open
Read
Special
Get Connection Status
Get Path Component
Update

Attach File
Seek
Write
Close
Get File Status
Delete Connection

```
(SSC)  Stream File System Calls [Yes/No]               YES
```

A response of "Yes" to this parameter means you wish to include all the system calls that support stream files. The following list includes the BIOS system calls that comprise the stream file group.

Create File
Open
Write
Close
Get File Status
Delete Connection

Attach File
Read
Special
Get Connection Status
Get Path Component
Delete File

```
   (NSC)  Named File System Calls [Yes/No]              YES
```

A response of "Yes" to this parameter means you wish to include all the system calls that support named files. The following list includes the BIOS system calls that comprise the named file group. If your system includes Remote File Access, you must respond "Yes" to this parameter.

Create File
Create Directory
Rename File
Seek
Write
Close
Get Default User
Get File Status
Get Path Component
Truncate
Get Extension Data
Update

Attach File
Change Access
Open
Read
Special
Get Connection Status
Set Default User
Get Directory Entry
Delete Connection
Delete File
Set Extension Data

NOTE

Configuring the "BIOS System Calls" screen allows you to choose the type of BIOS system calls you need to support your files. However, the ICU always includes the following Basic I/O System system calls automatically when you configure BIOS into your system: A$PHYSICAL$ATTACH$DEVICE, A$PHYSICAL$DETACH$DEVICE, CREATE$USER, DELETE$USER, GET$DEFAULT$USER, INSPECT$USER, SET$DEFAULT$USER, ENCRYPT and WAIT$IO.

CHAPTER 10
DEVICE DRIVER PARAMETERS

10.1 INTRODUCTION

This chapter discusses how to respond to the prompts that appear on the Intel Device Driver screens. The first screen, the "Intel Device Drivers" screen displays a list of the Intel supported devices in the order they are discussed in this chapter. You will also notice that this chapter has small tabs on the side to help you find the explanation of each device easily. If you are using this chapter to understand the configuration options for each device, look up the device on the "Intel Device Drivers" screen and then find its tab on the side of the manual.

10.2 INTEL-SUPPLIED DRIVERS SCREEN

The following screen allows you to choose the Intel-supplied drivers to include in your application system. This screen appears only in the following circumstances:

- When you are starting a new configuration file (not modifying an existing one)
- When you use the F command to search for a driver that is not configured into your system
- After you add a new Intel-supplied driver or modify your existing drivers
- When you type "f idevs <CR>".
- When you start with an existing def definition file.

When you choose a driver from this screen, the ICU lets you add a new driver, even if a driver of that kind is already configured into your system. To change the parameters of a driver you have already added to your system, either select an entry from this screen and go directly to the specific driver or use the F command and specify the name of the driver. If you want to add more than one driver of the same type, insert one driver. Upon completion, the ICU will display a query screen asking if you have more drivers to enter. If so, respond "Yes". The values shown here are the default values supplied in the 28612.def file.

```
(IDEVS)      Intel Device Drivers

(S14) Mass Storage Controller Driver [Yes/No]   YES
(T74) 8274 Terminal Driver  [Yes/No]            NO
(T51) 8251A Terminal Driver [Yes/No]            NO
(T30) 82530 Terminal Driver [Yes/No]            NO
(TCC) Terminal Comm Controller [Yes/No]         NO
(L86) Line Printer - iSBC 286/10 [Yes/No]       YES
(L50) Line Printer - iSBX 350 [Yes/No]          NO
(S20) iSBC 220        [Yes/No]  NO   (X18) iSBX 218A       [Yes/No]  NO
(S08) iSBC 208        [Yes/No]  NO   (T34) iSBC 534        [Yes/No]  NO
(T44) iSBC 544A       [Yes/No]  YES  (S64) iSBC 264        [Yes/No]  NO
(X51) iSBX 251        [Yes/No]  NO   (RAM) RAM Disk Driver [Yes/No]  NO
(SCS) SCSI Driver     [Yes/No]  NO   (S24) iSBC 186/224A   [Yes/No]  NO
(S10) iSBC 186/410    [Yes/No]  NO   (G79) iSBX 279        [Yes/No]  NO

  Enter [ Abbreviation = new_value / Abbreviation ? / H ]

  :
```

The following sections of this chapter describe the parameters associated with each of the Intel-supplied drivers.

NOTE

If you select the iSBX 279 driver, several screens will appear. The documentation that came with your package describes how to use these screens.

Mass Storage Controller Parameters

The Mass Storage Controller (MSC) driver supports the following device and controller combinations:

- 5 1/4-inch or 8-inch Winchester disk drives connected to an iSBC 215G controller.
- Tape drives connected to an iSBX 217C board that is mounted on an iSBC 215G controller (5 1/4-inch or 8-inch drives).
- Flexible diskette drives connected to an iSBX 218A board that is mounted on an iSBC 215G controller.
- 5.25-inch Winchester disk drives, streaming tape drives, and 5.25-inch flexible diskette drives that are connected to the iSBC 214 controller, a single board that combines all the features of the iSBC 215G, iSBX 217C, and iSBX 218(A) boards for 5.25-inch peripherals.

Regardless of the kind of drive you are configuring (hard disk, tape, or flexible diskette), the ICU screens you see when you configure the drive are exactly alike. The ICU knows which kind of drive you are configuring by the unit number you assign the drive (on the unit information screen).

Including the MSC driver in your application requires 68 bytes of memory, 6 bytes for the MSC driver and 62 bytes for the I/O Processor Block. Therefore, when including this device you must set the "SYS" parameter on the "Memory for System" and "Memory for Free Space Manager" screens to exclude the memory required by the MSC driver. For more information on how to calculate these memory locations see the explanation of the "Wakeup I/O Port" and "I/O Processor Block Address" parameters.

Three screens define the interface between the MSC random access device driver and the I/O system. These screens relate to the three device configuration tables: the device information table, the unit information table, and the Device Unit Information Block (DUIB). Refer to the *Extended iRMX II Device Drivers User's Guide* for further information about these tables.

The values shown on the screens in this section are the values you see when you invoke the ICU using the 28612.def file.

Mass Storage Controller Driver Screen

The ICU uses the information from the following screen to create a device information table for the MSC driver. If your system includes more than one iSBC 214 or iSBC 215G controller, you must specify a unique interrupt level and wakeup I/O port address for each controller.

```
(D214)        Mass Storage Controller Driver

(DEV) Device Name [1-16 Chars]                    214_A
(IL)  Interrupt Level [Encoded Level]             058H
(ITP) Interrupt Task Priority [0-255]             130
(WIP) Wakeup I/O Port [0-0FFFFH]                  0100H
(IPA) I/O Processor Block Address [0-0FFFFFH]     01200H

Enter [ Abbreviation = new_value / Abbreviation ? / H ]
:
```

```
(DEV) Device Name [1-16 Chars]                        214_A
```

This parameter provides a means of associating the device and all the units and DUIBs which belong to it. The ICU stores all the screens by type, not by device. This means that all device driver screens, all unit information screens, and all DUIB screens are stored together. The ICU determines which unit information and device-unit information screen relate to a particular device driver by the device name you enter in this parameter.

Thus, if you want to delete a device and all its related units, simply enter the name you specify in the "(DEV) Device Name" parameter, in this case 214_A, rather than delete each unit separately. To delete all units related to the "214_A" device, type "^d 214_A <CR>".

```
(IL)  Interrupt Level [Encoded Level]             058H
```

This parameter specifies the encoded interrupt level used by the iSBC 214/215G controller. The interrupt task uses this value to associate the interrupt task with the correct interrupt level. The default value 058H (0000 0000 0101 1000 binary) specifies master interrupt level 5.

The possible values for this field are encoded as follows (where bit 0 is the low-order bit):

Master		with Slave Attached		Master		with Slave Attached	
Level	Code	Level	Code	Level	Code	Level	Code
0	0008H	0-7	0000-0007H	4	0048H	0-7	0040-0047H
1	0018H	0-7	0010-0017H	5	0058H	0-7	0050-0057H
2	0028H	0-7	0020-0027H	6	0068H	0-7	0060-0067H
3	0038H	0-7	0030-0037H	7	0078H	0-7	0070-0077H

```
(ITP) Interrupt Task Priority [0-255]            130
```

This parameter specifies the initial priority of the device's interrupt task. The default value is 130 decimal. The actual priority of the iRMX II interrupt task changes because the Nucleus adjusts an interrupt task's priority according to the interrupt level that it services.

```
(WIP)  Wakeup I/O Port [0-0FFFFH]              0100H
```

This parameter specifies the I/O port address used to communicate with the controller. This wakeup address must match the jumper setting on the iSBC 214/215G board. Intel recommends that you do not change the default value of "0100H".

The value you enter for this parameter determines the location of six bytes of memory you must reserve for the MSC driver. These memory locations are used to communicate with the controller. To determine the starting address of this memory, multiply this parameter by 10H. Intel reserves six bytes of memory at this location.

The memory-mapped address associated with this parameter cannot be part of the memory you declared on the "Memory for System" or "Memory for Free Space Manager" screens. Refer to the *iSBC 215 Generic Winchester Disk Controller Hardware Reference Manual* or the *iSBC 214 Multi-Peripheral Controller Hardware Reference Manual* for information about the wakeup port address. Refer to Chapter 2 and the *Guide to the iRMX II Interactive Configuration Utility* for additional information about planning your memory usage.

```
(IPA) I/O Processor Block Address [0-0FFFFFH]          01200H
```

This parameter defines the starting address of the 60 bytes of memory required by the I/O Processor Block. These 60 bytes can be anywhere within the lower megabyte. Be sure to take these 60 bytes into consideration when you are configuring your system memory. Do not include these 60 bytes in the memory you define on the "Memory for System" or "Memory for Free Space Manager" screens. Also be sure that the memory you define for the I/O Processor Block does not overlap any other reserved memory location.

Entering a zero in this parameter means the 60 bytes of memory required by the I/O Processor Block will be contiguous to the wakeup address defined in the "Wakeup I/O Port" parameter. For example, if you enter 100H in the "Wakeup I/O Port" parameter, the I/O Processor Block will begin at location 1006H.

Query Screen

After you have completed the "Mass Storage Controller Driver" screen, the query screen is displayed. It contains only one line:

```
  Do you want any/more Mass Storage Controller DEVICEs ?
```

Respond "Yes" to this prompt if you wish to add another device driver of this type.

Mass Storage Controller Unit Information Screen

Use this screen to define a unit information table for each unique iSBC 214/215G/iSBX 217C/218A unit in your system. In addition, use this screen to define a "generic" Winchester disk. A "generic" unit allows you to use a version of the Operating System that does not need to know the specific characteristics of your Winchester disk. This means that you can run the same configuration on systems that contain different kinds of Winchester disks. This general feature is called automatic device characteristics recognition. Refer to the *Operator's Guide to the Extended iRMX II Human Interface* for additional information about automatic device characteristics recognition.

Automatic device characteristics recognition also allows you to configure a version of the Bootstrap Loader without knowing the specific characteristics of a Winchester disk. Refer to Chapter 8 in this manual for information about automatic boot device recognition. If you use the automatic boot device recognition feature, the drive from which you wish to load the Operating System needs the values shown on the following screen. Refer to the *Extended iRMX II Bootstrap Loader Reference Manual* for information about configuring the Bootstrap Loader.

```
(U214)   Mass Storage Controller Unit Information

(DEV) Device Name [1-16 Chars]                        214_A
(NAM) Unit Info Name [1-16 Chars]                     UINFO_214W
(MR)  Maximum Retries [0-0FFFFH]                      09H
(CS)  Cylinder Size [0-0FFFFH]                        0H
(NC)  Number of Cylinders [0-0FFFFH]                  01H
(NFH) Number of Heads/Fixed Disk [0-0FFH]             01H
(NRH) Number of Heads/Removable Disk [0-0FFH]         0H
(NS)  Number of Sectors/Track [0-0FFFFH]              0CH
(NAC) Number of Alternate Cylinders [0-0FFH]          0H
(SSN) Starting Sector Number [0-0FFFFFFFFH]           0H
(BTI) Bad Track Information [Yes/No]                  NO
(HLT) Head Load Time [0-0FFH]                         0H
(SR)  Step Rate [0-0FFH]                              0H

Enter [ Abbreviation = new_value / Abbreviation ? / H ]
:
```

```
(DEV)  Device Name [1-16 Chars]          214_A
```

This parameter must be the same name you entered for the "DEV" parameter on the "Mass Storage Controller Driver" screen. This name provides the logical ICU connection between the driver and all of its units. Thus, when a device is deleted it is possible to delete all the related units at the same time.

```
(NAM) Unit Info Name [1-16 Chars]         UINFO_214W
```

This parameter specifies a unique name for this unit information table. The first character must be an alphabetic character. Refer to the *ASM286 Language Reference Manual* for rules regarding this name.

Table 10-1 lists examples of names that are recommended for Winchester disk drives. Table 10-2 lists examples of names that are recommended for flexible disk drives. A DUIB uses the "Unit Info Name" to point to this particular unit information table. When developing your initial systems, you can create unit information tables that are never pointed to by a DUIB. There is no harm in this process during the development stages. However, when you configure your final system, eliminate all unused unit information tables to save memory.

Table 10-1. iSBC® 214/215G Controller Information for Winchester Drives

Unit Information						
Unit-Info Name	Device Type	Cylinder Size	Number Cylinder	Fixed Head /Platter	Sectors /Track	Alter-nates
uinfo_214w	"generic"	0	1	1	12	0
uinfo_214iw	Priam 3450	60	525	5	12	10
uinfo_214cm	CMI 5412	36	306	4	9	10
uinfo_214cmb	CMI 5619	54	306	6	9	10
uinfo_214qma	Quantum Q540	72	512	8	9	10
uinfo_214mma	Maxtor XT-1140	135	918	15	9	18
uinfo_214mmb	Maxtor XT-1085	72	1024	8	9	20
uinfo_214tma	Toshiba MK56FB	90	830	10	9	10
uinfo_214sma	Seagate SR251	54	820	6	9	10

Table 10-2. iSBC® 214/215G/iSBX™ 218A Controller Information for Diskette and Tape Drives

Unit Information						
Unit-Info Name	Cylinder Size	Number Cylinders	Heads/ Removable	Sectors/ Track	Head Load Time	Step Rate
uinfo_214f	26	77	1	26	40	8
uinfo_214fd	52	77	2	26	40	4
uinfo_214fdx	16	77	2	8	40	4
uinfo_214mfdx*	16	40	2	8	15	20
uinfo_214mfdy*	16	80	2	8	15	6
uinfo_217wt	0	0	0	0	N/A	N/A

Note: * This information applies only to an iSBX 218A MULTIMODULE board.

```
(MR)  Maximum Retries [0-0FFFFH]                  09H
```

For Winchester disk drives and flexible disk drives, this parameter specifies the maximum number of times the Basic I/O System will retry an operation before returning an E$IO$SOFT exception code. Refer to the *Extended iRMX II Basic I/O System User's Guide* for related information on the E$IO$SOFT exception code. Refer to the *iSBC 214 Multi-Peripheral Controller Hardware Reference Manual*, the *iSBC 215 Generic Winchester Disk Controller Hardware Reference Manual*, or the *iSBX 218 Flexible Disk Controller Hardware Reference Manual* for information on soft error conditions that can cause E$IO$SOFT errors.

For tape drives, this parameter is not relevant. Set the parameter to "0H".

```
(CS)  Cylinder Size [0-0FFFFH]                    0H
```

For Winchester disk drives and flexible disk drives, this parameter specifies the number of sectors per cylinder and whether the controller should perform automatic and concurrent seek operations. If you specify "0H", the I/O System will not perform either automatic seek operations or concurrent seek operations.

Automatic seek operations allow the random access software to ascertain the need to perform a seek operation before a requested read or write operation. The random access software is able to perform this function because it maintains the current location of the read/write head and can ascertain (based on the response to the "Cylinder Size" parameter) whether it should perform a read or write request within the current cylinder. If the request is not in the current cylinder, the random access software performs a seek operation.

The value you specify affects how or if automatic seek operations are performed. If you specify "0H", no automatic seek operations are performed by the random access software. Specify a zero value only if the device can perform seek functions independent of the random access software.

If you specify "01H", the random access software will perform automatic seek operations before each read or write request. If you specify a non-zero value other than 01H, the value you should enter is the cylinder size for this device (the number of sectors per track times the number of heads per disk).

For example, for a Winchester drive, if your response to the "Number of Sectors/Track" parameter is 12 decimal and your response to the "Number of Heads/Fixed Disks" is 6, then cylinder size is 72. For a flexible disk drive, if your response to the "Number of Sectors/Track" parameter is 26 decimal and your response to the "Single and Double Sided Disks" parameter is double, then cylinder size is 52.

If you specify a non-zero value for the "Cylinder Size" parameter, the random access software can concurrently perform seeks on more than one device. If you specify a zero value for the "Cylinder Size" parameter, the random access software does not attempt seek functions (including concurrent seeks). You must specify a zero value if you are defining multiple units on the same disk. (See the section "(SSN) Starting Sector Number" for information on how to define multiple units on a Winchester disk.)

For tape drives, this parameter is not relevant. Set it to "0H".

```
(NC)  Number of Cylinders [0-0FFFFH]                    01H
```

For Winchester disk drives, this parameter specifies the number of cylinders (tracks per surface) on the drive. In general, the number you specify should total the number of data cylinders and alternate cylinders on your drive. The exception to this guideline is using the default value "01H". This value (for a "generic" Winchester) designates an unknown drive type that is already formatted. Table 10-1 shows some typical values for this parameter.

For flexible disk drives, this parameter specifies the number of tracks per side of an 8- or 5.25-inch disk. Table 10-2 shows some of the typical values for this parameter.

For tape drives, this parameter is not relevant, set it to zero.

```
(NFH) Number of Heads/Fixed Disk [0-0FFH]               01H
```

For Winchester disk drives, this parameter specifies the number of heads on your Winchester's fixed platters. Your vendor manual should contain this number. Should you need to determine this number, multiply the number of platters by the number of surfaces per platter. For example, suppose your Winchester has three fixed platters and each platter has two surfaces. Assuming one read/write head per usable surface, the response to the "Number of Heads/Fixed Disk" parameter should be six.

For flexible disk drives and tape drives, this parameter is not relevant. Set the parameter to "0H".

```
(NRH) Number of Heads/Removable Disk [0-0FFH]            0H
```

For Winchester disk drives, this parameter is not relevant. Set the parameter to "0H".

For flexible disk drives, this parameter specifies the number of heads on your flexible disk drive. For single-sided drives enter "01H". For double-sided drives enter "02H".

For tape drives, this parameter is not relevant. Set this parameter to "0H".

```
(NS)  Number of Sectors/Track [0-0FFFFH]               0CH
```

For Winchester disk drives and flexible disk drives, the number of sectors per track varies with the sector size and the hardware vendor. For flexible disk drives, the number of sectors per track varies with the sector size, density, and overall size.

If you are using the automatic device characteristics recognition feature, specify a value that represents the maximum sectors-per-track value of all the disks accessed by the controller. If the value you use for this parameter is not as large as the actual number of sectors per track, you will experience reliability problems when accessing the disk.

For tape drives, this parameter is not relevant. Set the parameter to "0H".

```
(NAC) Number of Alternate Cylinders [0-0FFH]           0H
```

For Winchester disk drives, this parameter specifies the number of alternate cylinders on your Winchester device. When assigning a number for this parameter, use a value that represents one to two percent of the total number of available cylinders on the unit.

When you format a Winchester device, the device driver detects bad tracks. When it detects bad tracks, it assigns one alternate track for each bad track it detects. These alternate tracks are located on the inside cylinders of the disk.

During normal operation, when the controller accesses a track that has been marked as defective, it automatically invokes a seek to the assigned alternate track. It uses the alternate track as if it were the original data track. This operation is automatic and is invisible to the user, except for the additional time needed to complete the disk operation.

For a device with multiple recording surfaces, the number of alternate tracks on each surface are equal. Therefore, devices with multiple platters set aside entire cylinders for alternate tracks.

In addition to containing alternate tracks, one of the alternate cylinders is a diagnostic cylinder. Consequently, on an error-free disk, you must assign a minimum of one alternate cylinder. A Winchester disk in an Intel System 300 Series Microcomputer System uses an additional four tracks for permanent bad-track information. As a result, for an error-free disk in this environment, you must assign a minimum of two alternate cylinders.

When an alternate track is assigned, the Operating System returns an E$IO$ALT$ASSIGNED warning message.

If your disk is not error-free and you have not specified enough alternate cylinders, the Operating System can return an exception code when you format the disk. Specifically, the Basic I/O System returns an E$IO$NO$SPARES exception code if the driver cannot find enough alternate tracks.

For flexible disk drives and tape drives, this parameter is not relevant. Set the parameter to "0H".

```
(SSN) Starting Sector Number [0-0FFFFFFFFH]             0H
```

For Winchester disk drives, this parameter specifies the starting sector number of this unit. Change the default value only if you are creating multiple units on the same disk.

Each track is divided into equal-sized sectors and each sector is assigned a number. Sector numbering starts with the number 0 on side 0, cylinder 0 and continues consecutively around that track. Numbering continues on cylinder 0 before proceeding to surface 0, cylinder 1.

For flexible disk drives and tape drives, this parameter is not relevant. Set the parameter to "0H".

Multiple Units on the Same Winchester Disk

You can perform a number of steps during configuration to define multiple units. Before performing these steps, however, you should consider the following alternative and note the possible difficulties.

If the reason for setting up multiple units on the same disk is to set aside a scratch area on your Winchester, Intel recommends that you create a file large enough to provide this capability rather than define multiple units on your Winchester. Defining your Winchester as more than one logical unit can have several drawbacks. First, having multiple units on the same physical device prevents the random access software from performing concurrent seeks on the device. Second, if for some reason one of the multiple units needs to be reformatted, all units on the device must be reformatted. Third, multiple units could put an unusual strain on your Winchester drive (ruining the seek optimization).

Define the following values for multiple units:

- A separate device-unit number that you can assign to the same physical device-unit.
- A unit number that accommodates multiple units. The driver checks only the four least-significant bits of the unit number. You can use the upper bits to identify multiple units on the same disk.
- A device size for the device-unit that must be less than the device size for the entire device. You must be careful not to allow the areas used by different units to overlap.
- A starting sector number that allows you to define a unique starting point for each device-unit. To avoid difficulties with formatting, each unit should start and end on a track boundary.

Take care when formatting multiple units on the same disk. You should heed the following guidelines:

- Do not format the units by invoking the FORMAT command from two terminals at the same time.
- Format the units sequentially during the same initial attach device of the base unit.

```
(BTI) Bad Track Information [Yes/No]                    NO
```

For Winchester disk drives, this parameter specifies whether the disk controller, when it formats the disk, uses the factory-generated bad-track information stored on the disk. Specify "Yes" if the disk is an Intel-supplied hard disk or a disk that contains valid bad-track information. Specify "No" if you are not using Intel-supplied system products or you do not want the disk controller to use the factory-generated bad-track information during the process of formatting the disk.

For flexible disk drives and tape drives, this parameter is not relevant. Set the parameter to "No".

```
(HLT)  Head Load Time [0-0FFH]                          0H
```

For Winchester disk drives and tape drives, this parameter is not relevant. Set the parameter to "0H".

For flexible disk drives, this parameter specifies the time interval (in milliseconds) that the controller waits after loading the head but before initiating a read or write operation. The value used in the iRMX II Multi-user System (028H) is equivalent to 40 decimal milliseconds. This head load time includes a provision for head settling time.

```
(SR)  Step Rate [0-0FFH]                                  0H
```

For Winchester disk drives and tape drives, this parameter is not relevant. Set the parameter to "0H".

For flexible disk drives, this parameter specifies the time interval (in milliseconds) between step pulses as they relate to track-to-track access time.

Query Screen

After you have completed the unit information screen, the query screen is displayed. It contains only one line:

```
Do you want any/more Mass Storage Controller UNITs ?
```

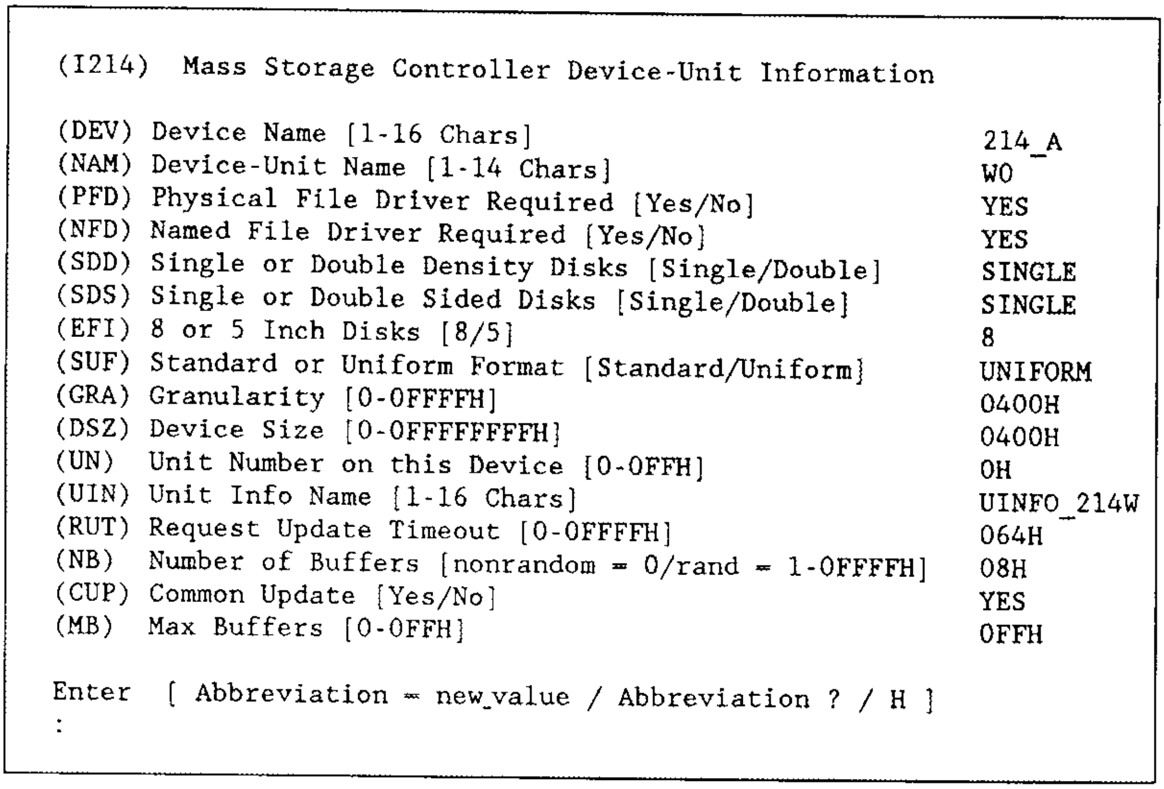
Respond "Yes" to this prompt if you need another unit information table for this device.

Mass Storage Controller Device-Unit Information Screen

Use this screen to define a device unit information block (DUIB) for each unique device-unit in your system (Winchester drives, flexible disk drives, and tape drives). The value you choose for the unit number tells the driver whether the device-unit is a Winchester, flexible disk, or tape drive.

In addition, use this screen to define a "generic" Winchester disk. A "generic" device-unit allows you to use a version of the Operating System that does not need to know the specific characteristics of your Winchester disk. This means that you can run the same configuration on systems that contain different kinds of Winchester disks. This general feature is called automatic device characteristics recognition. Refer to the *Operator's Guide to the iRMX II Human Interface* for additional information about automatic device characteristics recognition.

Automatic device characteristics recognition also allows you to configure a version of the Bootstrap Loader without knowing the specific characteristics of a Winchester disk. Refer to Chapter 8 in this manual for information about automatic boot device recognition. Refer to the *Extended iRMX II Bootstrap Loader Reference Manual* for information about configuring the Bootstrap Loader.



```
(I214)  Mass Storage Controller Device-Unit Information

(DEV) Device Name [1-16 Chars]                               214_A
(NAM) Device-Unit Name [1-14 Chars]                          W0
(PFD) Physical File Driver Required [Yes/No]                 YES
(NFD) Named File Driver Required [Yes/No]                    YES
(SDD) Single or Double Density Disks [Single/Double]         SINGLE
(SDS) Single or Double Sided Disks [Single/Double]           SINGLE
(EFI) 8 or 5 Inch Disks [8/5]                                8
(SUF) Standard or Uniform Format [Standard/Uniform]          UNIFORM
(GRA) Granularity [0-0FFFFH]                                 0400H
(DSZ) Device Size [0-0FFFFFFFFH]                             0400H
(UN)  Unit Number on this Device [0-0FFH]                    0H
(UIN) Unit Info Name [1-16 Chars]                            UINFO_214W
(RUT) Request Update Timeout [0-0FFFFH]                      064H
(NB)  Number of Buffers [nonrandom = 0/rand = 1-0FFFFH]      08H
(CUP) Common Update [Yes/No]                                 YES
(MB)  Max Buffers [0-0FFH]                                   0FFH

Enter  [ Abbreviation = new_value / Abbreviation ? / H ]
:
```

```
(DEV)  Device Name [1-16 Chars]                    214_A
```

The name you enter for this parameter must be the same name you entered in the "(DEV) Device Name" parameter on the "Mass Storage Controller Driver" screen. This name provides the logical ICU connection between a driver and all of its DUIBs. Thus, when a device is deleted it is possible to delete all the related units and DUIBs at the same time.

```
(NAM) Device-Unit Name [1-14 Chars]                    W0
```

This parameter specifies a name that uniquely identifies this device-unit to the I/O System. If you want the Extended I/O System to logically attach this device during initialization, the name you specify for this parameter must also be one of the device names you specified on the "Logical Names" screen (see Chapter 8).

To support auto boot device recognition for your Winchester and flexible diskette drives, specify the same name in one of the %DEVICE macros when you configure this device in your Bootstrap Loader (see the *Extended iRMX II Bootstrap Loader Reference Manual* for detailed information). The default name used by the Bootstrap Loader for a Winchester drive is W0 or W1. The default names used for flexible disk drives are wf0 and wf1.

The ICU allows you to enter as many as 14 characters for the name. Refer to the *ASM286 Language Reference Manual* for rules regarding this name. Tables 10-3 and 10-4 list examples of device-unit names that are recommended for Winchester, tape, and flexible disk drives.

The name you specify for this parameter is the physical name you specify when invoking the Human Interface ATTACHDEVICE command, the BIOS A$PHYSICAL$ATTACH$DEVICE system call, or the EIOS LOGICAL$ATTACH$DEVICE system call.

Table 10-3. iSBC® 214/215G DUIB Information for Winchester Drives

D U I B Information						
Model	Device-Unit Name(s)	Size	Gran	Device Size	Unit-Info Name	Num Buffers
Generic Winchester	w0	8	1024	400H	uinfo_214w	8
PRIAM3450	iw0	8	1024	1E2D00H	uinfo_214iw	8
PRIAM8050	iwb0	8	1024	3C6900H	uinfo_214iwb	8
CMI5412	cm0	5	1024	0A68000H	uinfo_214cm	8
CMI5619	cmb0	5	1024	0F9C000H	uinfo_214cmb	8
or						
Fujitsu M2235	cmb0	5	1024	0F9C000H	uinfo_214cmb	8
Quantum Q540	qma0	5	1024	234C000H	uinfo_214qma	8
Maxtor XT-1140	mma0	5	1024	76A7000H	uinfo_214mma	8
Maxtor XT-1085	mmb0	5	1024	4698000H	uinfo_214mmb	8
Toshiba MK56FB	tma0	5	1024	4812000H	uinfo_214tma	8
Seagate SR251	sma0	5	1024	2AB7000H	uinfo_214sma	8

Table 10-4. iSBC® 214/215G/iSBX™ 218A DUIB Information for Flexible Disk and Tape Drives

	D U I B Information					
Device-Unit Name(s)	Density	Sides	Gran	Device Size	Unit-Info Name	Num Buffers
WF0	Single	Single	128	3E900H	uinfo_214f	6
WFD0	Double	Single	256	7C500H	uinfo_214f	6
WFDD0	Double	Double	256	F9700H	uinfo_214fd	6
WFDX0	Double	Double	1024	132700H	uinfo_214fdx	3
WMF0	Double	Double	512	4F800H	uinfo_214mf	6
WMFDY0	Double	Double	512	9F800H	uinfo_214mfdy	6
WTA0	N/A	N/A	32256	1400000H	uinfo_214wt	0

```
(PFD) Physical File Driver Required [Yes/No]              YES
(NFD) Named File Driver Required [Yes/No]                 YES
```

For Winchester disk drives and flexible disk drives, the driver supports both named and physical file drivers. These parameter lines let you specify "Yes" to at least one of these parameters.

Two good reasons exist for not changing either "Yes" default value. First, the Human Interface ATTACHDEVICE command, the EIOS LOGICAL$ATTACH$DEVICE system call, and the BIOS A$PHYSICAL$ATTACH$DEVICE system call require you to select the file type at run time. Second, there is no code savings if you specify "Yes" to one parameter and "No" to the other.

For flexible disk drives, the ICU ignores your response to the "Named File Driver Required" parameter if you specify a "Uniform" format (refer to the section on "Standard or Uniform Format").

For tape drives, specify "Yes" to the "Physical File Driver Required" parameter and "No" to the "Named File Driver Required" parameter.

```
(SDD) Single or Double Density Disks [Single/Double]  SINGLE
```

For Winchester disk drives and tape drives, this parameter is not relevant. Set this parameter to "single".

For flexible disk drives, this parameter specifies the recording density of this flexible disk. The recording density is either single or double. Table 10-4 lists some of the values possible. Tables 10-5 and 10-6 list additional values.

Table 10-5. 5.25-Inch Disk Characteristics

			Device Size			
		Sectors	One Sided		Two Sided	
Gran	Density	per Track	40 Tracks	80 Tracks	40 Tracks	80 Tracks
128	Single	16	81920	163840	163840	327680
256	Single	9	91904	184064	184064	368384
512	Single	4	81920	163840	163840	327680
1024	Single	2	81920	163840	163840	327680
256	Double	16	1617921	325632	325632	653312
512	Double	8	1617921	325632	325632	653312
1024	Double	4	1617921	325632	325632	653312

Table 10-6. 8-Inch Disk Characteristics

			Device Size	
		Sectors	One Sided	Two Sided
Gran	Density	per Track	77 Tracks	77 Tracks
128	Single	26	256256	512512
256	Single	15	295168	590848
512	Single	8	314880	630272
1024	Single	4	315392	630784
256	Double	26	509184	1021696
512	Double	15	587264	1177600
1024	Double	8	626688	1255168

```
(SDS) Single or Double Sided Disks [Single/Double]  SINGLE
```

For Winchester disk drives and tape drives, this parameter is not relevant. Leave the parameter at its default setting.

For flexible disk drives, this parameter specifies the number of recording surfaces on the flexible disk. Tables 10-6 and 10-7 show how your response to this prompt is related to other parameters.

```
(EFI) 8 or 5 Inch Disks [8/5]                          8
```

For Winchester disk drives and flexible disk drives, this parameter specifies the size of the disk used in your system.

For tape drives, this parameter is not relevant. Leave the parameter at its default setting.

```
(SUF) Standard or Uniform Format [Standard/Uniform]  UNIFORM
```

For Winchester disk drives and tape drives, this parameter must be left in the default setting.

For flexible disk drives, this parameter specifies the format that the controller should expect on track zero. Choose the option "Standard" if you want track zero to be single density with 128-byte sectors (regardless of how the remaining tracks are formatted). Choose the option "Uniform" if you want all tracks on a disk to have the same format.

Normally, when you use the FORMAT command to format a disk as a named volume, the command formats track zero with a fixed density (single density) and a fixed sector size (128 bytes). This is the "standard" format. Intel recommends that you use this format.

If you wish to read a disk that is in uniform format, you should use the "Standard/Uniform Format" parameter to designate a "Uniform" format and attach the drive as a physical file. If you choose the "Uniform" option, you should also respond "No" to the "Named File Driver Required" parameter. Note that automatic device characteristics recognition does not work unless track zero is single density with 128-byte sectors. Booting a system from a flexible disk drive is supported only when using the "standard" format.

```
(GRA) Granularity [0-0FFFFH]                         0400H
```

This parameter specifies the minimum number of bytes that the device reads or writes in one operation. This value is also called device granularity. For Winchester disk drives and flexible disk drives, this value is sector size.

Refer to the *Extended iRMX II Basic I/O System User's Guide* for more information about granularity.

For Winchester disk drives, the vendor of your drive may recommend optimum values for this parameter. Refer to vendor documentation for additional information.

For flexible disk drives, the granularity should equal the volume granularity. (You specify volume granularity when you format diskettes.) Tables 10-5 and 10-6 show how the granularity relates to other parameters.

For tape drives, Intel recommends the value "7E00H".

```
(DSZ) Device Size [0-0FFFFFFFFH]                    0400H
```

This parameter specifies the device storage capacity in bytes.

For Winchester disk drives and flexible disk drives, the device size varies with the granularity and sectors per track. Table 10-3 shows the device sizes of several Winchester disks. Tables 10-5 and 10-6 show the relationships between these factors for flexible disk drives. For generic DUIBs, set this value to device granularity.

For Winchester disk drives, use the following formula to calculate device size:

DSZ = Granularity * (sectors per track) * (# heads)
* (# cylinders - # alternate cylinders)

For flexible diskette drives, use these formulas to calculate device size:

Uniform flexible diskettes

DSZ = Granularity * (sectors per track) * (2 heads) * (# cylinders)

Standard flexible diskettes

Size first track = (128 bytes/sector) * (16 sectors)
Size other tracks = Granularity * (sectors per track) * (# tracks)

DSZ = (size first track) + (size other tracks)

For tape drives, set to manufacturer's specifications for capacity.

```
(UN)  Unit Number on this Device [0-0FFH]                OH
```

This parameter specifies the unit number of this device-unit. This number identifies one of 15 possible units on this device. The unit numbers for the device begin with zero and increase sequentially. The unit number you choose tells the driver whether the unit is a Winchester disk drive, flexible disk drive, or tape drive. The driver assumes that unit numbers are associated with drive types as follows:

Unit Number	Drive Type
0-3	Fixed disks on Winchester drives 1 through 4
4-7	Removable disks on Winchester drives 1 through 4
8-11	Flexible disk drives 1 through 4
12-15	Tape drives 1 through 4

```
(UIN) Unit Info Name [1-16 Chars]            UINFO_214W
```

This parameter specifies the "Unit Info Name" of a unit information table that fills the needs of this DUIB. The following rules apply to the name you choose:

- You must create a unit information table with this "unit info name".
- Each DUIB can point only to one unit information table.
- A particular unit information table can be pointed to by more than one DUIB.

For example, if you used the name "uinfo_217wta" when you defined the unit information screen for a tape drive (NAM=uinfo_217wta), you must use the same name for this parameter (UIN=uinfo_217wta).

```
(RUT) Request Update Timeout [0-0FFFFH]              064H
```

For Winchester disk drives and flexible disk drives, this parameter specifies the number of clock ticks that the driver should wait (during a pause in activity) before updating the attached files on the device.

The values "0" and "0FFFFH" do not indicate time intervals. A value of "0" indicates that the driver will never leave any data buffered in memory (eliminating the need to update). A value of "0FFFFH" indicates that updates on this device will occur only when a file is detached. However, the ICU treats your response to this parameter totally separate from the response made to the "Common Update" parameter (you can specify both).

The update capability provided by this parameter differs from the common update capability described in Chapter 9. Unlike the ability to update at fixed periods that is provided by the common update feature, the update capability referenced by the "Request Update Timeout" parameter allows the driver to update based on pauses in activity. If there is a pause in activity on this device, the driver determines how soon the common update would occur and compares that time interval to the request update timeout value. The driver then waits the shorter of the two intervals and updates the attached files on the device. Thus, if you use the common update feature, the time interval you specify for the "Request Update Timeout" should be shorter than the time interval specified for the "Common Update Timeout".

For tape drives, this parameter is not relevant. Set the parameter to "0FFFFH".

```
(NB) Number of Buffers [nonrandom = 0/rand = 1-0FFFFH]  08H
```

For Winchester disk drives and flexible disk drives, this parameter specifies the number of buffers this device has for blocking and deblocking I/O requests.

The value "8" is appropriate for a unit that has a granularity of 1024 bytes. Table 10-3 shows how this value relates to other parameters for Winchester drives. Table 10-4 shows how the value relates for flexible disk drives. Because this is a device which supports random access, do not specify a value of zero.

The number of buffers prompt determines the number of Basic I/O System buffers the device uses for I/O. These buffers are general-purpose cache buffers that the Basic I/O System uses to increase I/O performance. The size of each buffer is 32 bytes greater than the device granularity you specified for this device (your response to the "(GRA) Granularity" parameter). The buffers are associated with a device-unit, not with a particular file. The Basic I/O System assigns these buffers to the device-unit when it attaches the device-unit.

Because the Operating System uses memory from the BIOS memory pool to create these buffers, you should increase the values you specified on the "BIOS" screen for both the minimum and maximum memory pool sizes if you increase the default value for the "Number of Buffers" parameter. An explanation of how to calculate memory usage based on your responses to the "(GRA) Granularity" and the "(NB) Number of Buffers" parameter lines is in Chapter 9.

Choosing an optimum number of buffers depends on many factors: how your application works; the speed of your components, your CPU, your memory, and your peripherals; the performance characteristics of your peripherals; and the number of users. Plus, you must trade off optimum performance against the amount of memory you use (because the more buffers you choose, the more memory you need).

Because there are so many factors, the only way that you can obtain the optimum value for your system is by testing different values and fine-tuning your configuration. Intel recommends that you use at least two buffers for this device.

For tape drives, leave this parameter set to "0H".

```
(CUP) Common Update [Yes/No]                         YES
```

For Winchester disk drives and flexible disk drives, specify "Yes" for this parameter if you want the driver to update the attached files on this device at the fixed interval you specified on the "BIOS" screen. Specify "No" if you do not want to update attached files on this device at fixed intervals but prefer to update attached files based only on the time interval you specified for the "Request Update Timeout" parameter. You should not specify "No" for this parameter and "0FFFFH" for the "Request Update Timeout" parameter. See Chapter 9 for information about common update timeout (updating attached files at fixed intervals).

For tape drives, this parameter is not relevant. Set the parameter to "No".

```
(MB)  Max Buffers [0-0FFH]                        0FFH
```

This parameter specifies the maximum number of buffers that the Extended I/O System can allocate for this device's I/O. The default value for this parameter (0FFH) allows the S$OPEN system call to specify the actual number of Extended I/O System buffers. The Operating System takes memory required for these buffers from the calling job's memory pool, so by setting this parameter to "0FFH" you allow the calling job to select the number of buffers based on its own memory pool size. Intel recommends that you use the default value.

Query Screen

After you have completed the DUIB information screen, the query screen is displayed. It contains only one line:

```
Do you want any/more Mass Storage Controller DUIBs ?
```

Respond "Yes" to this prompt if you plan to use this controller with two devices that have different characteristics, or devices that have the same characteristics, but different unit numbers.

While developing your initial systems, you can create as many device-unit information blocks as you want. The number of DUIBs can exceed the number of devices on your system. The particular DUIB associated with the device depends on the physical name you use when attaching it. Once you know that you will never need a particular DUIB, save memory by deleting it from your definition file before you generate your configuration files (refer to Chapter 16 for additional information on generating configuration files).

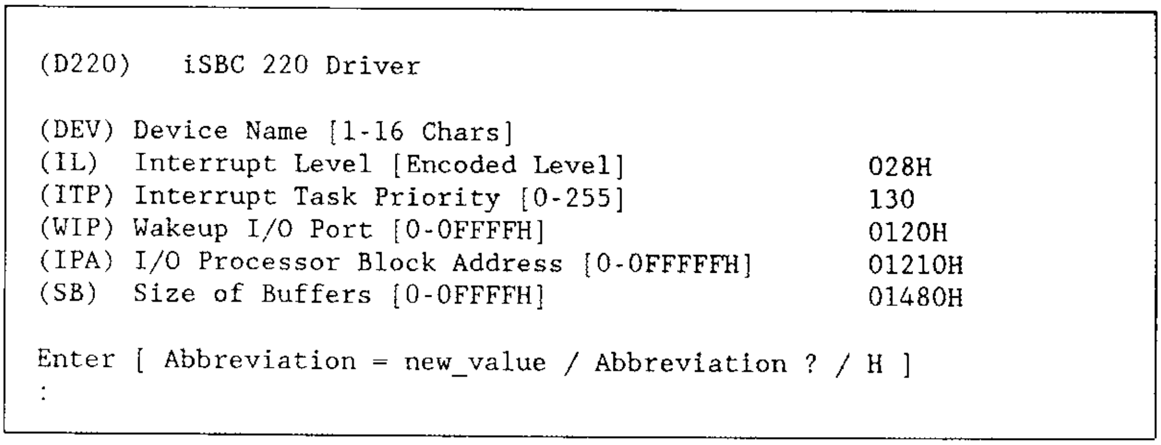
iSBC® 220 Driver Parameters

The iSBC 220 controller supports Storage Module Device disks. Three screens define the interface between the iSBC 220 random access device driver and the I/O system. These screens relate to the three device configuration tables: the device information table, the unit information table, and the device unit information block (DUIB). Refer to the *Extended iRMX II Device Drivers User's Guide* for more information about these tables.

The iSBC 220 driver is not part of the Intel-supplied Start-up system. The default values displayed here are the ICU default values.

iSBC® 220 Driver Screen

The ICU uses the information from the following screen to create a device information table for the iSBC 220 driver. If your system includes more than one iSBC 220 controller, you must specify a unique interrupt level and wakeup I/O port address for each controller.



```
(D220)    iSBC 220 Driver

(DEV) Device Name [1-16 Chars]
(IL)  Interrupt Level [Encoded Level]                028H
(ITP) Interrupt Task Priority [0-255]                130
(WIP) Wakeup I/O Port [0-0FFFFH]                     0120H
(IPA) I/O Processor Block Address [0-0FFFFFH]        01210H
(SB)  Size of Buffers [0-0FFFFH]                     01480H

Enter [ Abbreviation = new_value / Abbreviation ? / H ]
:
```

```
(DEV) Device Name [1-16 Chars]
```

This parameter provides a means of associating the device and all the units and DUIBs which belong to it. The ICU stores all the screens by type, not by device. This means that all device driver screens are stored together, all unit information screens and all DUIB screens. The ICU "knows" which unit information and device-unit information screen relate to a particular device driver by the device name you enter in the "DEV" parameter.

```
(IL)  Interrupt Level [Encoded Level]              028H
```

This parameter specifies the encoded interrupt level for the iSBC 220 driver. The interrupt task uses this value to associate the interrupt task with the correct interrupt level. The default value 0028H (0000 0000 0010 1000 binary) specifies master interrupt level 2.

The possible values for this field are encoded as follows (where bit 0 is the low-order bit):

Master Level	Code	Slave Level	Attached Code	Master Level	Code	Slave Level	Attached Code
0	0008H	0-7	0000-0007H	4	0048H	0-7	0040-0047H
1	0018H	0-7	0010-0017H	5	0058H	0-7	0050-0057H
2	0028H	0-7	0020-0027H	6	0068H	0-7	0060-0067H
3	0038H	0-7	0030-0037H	7	0078H	0-7	0070-0077H

```
(ITP) Interrupt Task Priority [0-255]              130
```

This parameter specifies the initial priority of the device's interrupt task. The actual priority of the iRMX II interrupt task changes because the Nucleus adjusts an interrupt task's priority according to the interrupt level that it services.

```
(WIP)  Wakeup I/O Port [0-0FFFFH]                0120H
```

This parameter specifies the I/O port address used to communicate with the controller. This I/O port address, called the wakeup port, must match the appropriate switch setting on your iSBC 220 board.

Additionally, this value is used to define the memory locations used to initiate communication with the controller. To determine the starting address of this memory, multiply this parameter by 10H. Intel reserves six bytes of memory at this location.

The memory mapped address associated with this parameter can not be part of the memory you declared on the "Memory for System" screen. Refer to the iSBC 220 SMD Disk Controller Hardware Reference Manual for information about the wakeup address. Refer to Chapter 2 and the *Extended iRMX II Interactive Configuration Utility Reference* manual for additional information about planning your memory usage.

```
(IPA) I/O Processor Block Address [0-0FFFFFH]        01210H
```

This parameter defines the starting address of memory required by the I/O Processor Blocks and I/O buffers. This memory can be anywhere within the lower megabyte. Do not include this memory in the memory you define on the "Memory for System" or "Memory for Free Space Manager" screens. Also be sure that the memory you define for the I/O Processor Board does not overlap any other reserved memory location.

```
    (SB) Size of Buffers [0-0FFFFH]                        01480H
```

This parameter defines the size of the I/O processor blocks and the I/O buffers. The formula for calculating this figure is

1152 + (N * device granularity)* U

where

N is the number of transfer buffers.

U is the number of units configured for this board.

Query Screen

After you have completed the "iSBC 220 Driver" screen, the query screen is displayed. It contains only one line:

```
    Do you want any/more iSBC 220 DEVICEs ?
```

Respond to this prompt with a "Yes" if you wish to add another device driver of this type.

iSBC® 220 Unit Information Screen

The ICU uses the information from the following screen to create a unit information table for the iSBC 220 driver.

Use the "iSBC 220 Unit Information" screen to define a unit information table for each unique unit in your system. In addition, use this screen to define a "generic" storage module device. A "generic" unit allows you to use a version of the Operating System that does not need to know the specific characteristics of your storage module device (SMD) disk. This means that you can run the same configuration on systems that contain different kinds of SMD disks. This general feature is called automatic device characteristics recognition. Refer to the *Operator's Guide to the Extended iRMX II Human Interface* for additional information about automatic device characteristics recognition.

If you use the automatic device characteristics recognition feature, the only default value you might want to change is the number of sectors per track. Refer to the section labeled "(NS) Number of Sectors/Track" for detailed information.

Automatic device characteristics recognition also allows you to configure a version of the Bootstrap Loader without knowing the specific characteristics of a SMD disk. Refer to Chapter 8 in this manual for information about automatic boot device recognition. Refer to the *Extended iRMX II Bootstrap Loader Reference Manual* for information about configuring the Bootstrap Loader. If you use the automatic boot device recognition feature, only the drive from which you wish to load the Bootstrap Loader needs the default values shown in following screen.

```
(U220)     iSBC 220 Unit Information

(DEV) Device Name [1-16 Chars]
(NAM) Unit Info Name [1-16 Chars]
(MR)  Maximum Retries [0-0FFFFH]                          09H
(CS)  Cylinder Size [0-0FFFFH]                            07EH
(NC)  Number of Cylinders [0-0FFFFH]                      024DH
(NFH) Number of Heads/Fixed Disk [0-0FFH]                 07H
(NRH) Number of Heads/Removable Disk [0-0FFH]             0H
(NS)  Number of Sectors/Track [0-0FFFFH]                  012H
(NAC) Number of Alternate Cylinders [0-0FFH]              0BH
(SSN) Starting Sector Number [0-0FFFFFFFFH]               0H
(BTI) Bad Track Information [Yes/No]                      YES

Enter [ Abbreviation = new_value / Abbreviation ? / H ]
:
```

```
(DEV)  Device Name [1-16 Chars]
```

The name you enter for this parameter must be the same name you entered for the "DEV" parameter on the "iSBC 220 Driver" screen. This name provides the logical ICU connection between the driver and all of its units. Thus, when a device is deleted it is possible to delete all the related units at the same time.

```
(NAM) Unit Info Name [1-16 Chars]
```

This parameter specifies a unique name for this unit information table. The first character must be an alphabetic character. Refer to the *ASM286 Language Reference Manual* for rules regarding this name.

A DUIB uses the "Unit Info Name" to point to this particular unit information table. When developing your initial systems, you can create unit information tables that are never pointed to by a DUIB. There is no harm in this process during the development stages. However, when you configure your final system, eliminate all unused unit information tables to save memory.

```
    (MR)  Maximum Retries [0-0FFFFH]                    09H
```

This parameter specifies the maximum number of times the Basic I/O System should retry an operation before returning an E$IO$SOFT exception code. Refer to the *Extended iRMX II Basic I/O System User's Guide* for related information on the E$IO$SOFT exception code. Refer to the *iSBC 220 SMD Disk Controller Hardware Reference Manual* for information on soft error conditions that can cause E$IO$SOFT errors. The default value of 9 is recommended.

```
    (CS)  Cylinder Size [0-0FFFFH]                      07EH
```

This parameter specifies the number of sectors per cylinder and whether the controller should perform automatic and concurrent seek operations. If you specify "0H", the I/O System will not perform either automatic seek operations or concurrent seek operations. If you specify a value other than "0H", the I/O System can perform both automatic seek operations and concurrent seek operations.

Automatic seek operations allow the random access software to ascertain the need to perform a seek operation before a requested read or write operation. The random access software is able to perform this function because it maintains the current location of the read/write head and can ascertain (based on the response to the "Cylinder Size" parameter) whether it should perform a read or write request within the current cylinder. If the request is not in the current cylinder, the random access software performs a seek operation.

The value you specify affects how or if automatic seek operations are performed. If you specify "0H", no automatic seek operations are performed by the random access software. Only specify a zero value if the device can perform seek functions independent of the random access software. If you specify a value of "01H", the random access software will perform automatic seek operations before each read or write request. If you specify a non-zero value other than "01H", the value you should enter is the cylinder size for this device (the number of sectors per track times the number of heads per disk). For example, if your response to the "Number of Sectors/Track" parameter is 12 decimal and your response to the "Number of Heads/Fixed Disks" is 6, then cylinder size should be 72.

If you specify a non-zero value for the "Cylinder Size" parameter, the random access software can concurrently perform seeks on more than one device. If you specify a zero value for the "Cylinder Size" parameter, the random access software does not attempt seek functions (including concurrent seeks).

```
    (NC)  Number of Cylinders [0-0FFFFH]          024DH
```

This parameter specifies the number of cylinders (tracks per surface) on the drive accessed by your iSBC 220 SMD disk controller. In general, the number you specify should total the number of data cylinders and alternate cylinders on your drive. The exception to this guideline is using a value of "1". This value (for a "generic" SMD) designates an unknown drive type that is already formatted. The default value 024DH is equivalent to 589 decimal.

```
    (NFH) Number of Heads/Fixed Disk [0-0FFH]       07H
```

This parameter specifies the number of heads on your SMD's fixed platters. To determine this number, multiply the number of platters by the number of surfaces per fixed platter. For example, if your SMD has three fixed platters and each platter has two surfaces, then assuming one read/write head per usable surface the response to the "Number of Heads/Fixed Disk" parameter should be six.

```
    (NRH) Number of Heads/Removable Disk [0-0FFH]   0H
```

This parameter specifies the number of heads on the removable platters. To determine this number, multiply the number of platters by the number of surfaces per platter.

```
    (NS)  Number of Sectors/Track [0-0FFFFH]        012H
```

This parameter specifies the number of sectors per track. The default value 012H is equivalent to 18 decimal. The number of sectors per track on a SMD varies with the sector size and the hardware vendor.

If you are using the automatic device characteristics feature, specify a value that represents the maximum sectors-per-track value of all the disks accessed by the controller. If the value you use for this parameter is not as large as the number of sectors per track, you will experience reliability problems when accessing the disk.

```
    (NAC) Number of Alternate Cylinders [0-00FFH]   0BH
```

This parameter specifies the number of alternate cylinders on your SMD device.

When you format a SMD disk, the device driver can detect bad tracks. When it detects bad tracks, it assigns one alternate track for each bad track it detects. These alternate tracks are located on the inside cylinders of the disk.

During normal operation, when the controller accesses a track that has been marked as defective, it automatically invokes a seek to the assigned alternate track. It uses the alternate track as if it were the original data track. This operation is automatic and is invisible to the user, except for the additional time needed to complete the disk operation.

For a device with multiple recording surfaces, the number of alternate tracks on each surface are equal. Therefore, devices with multiple platters set aside entire cylinders for alternate tracks. When assigning a number for this parameter, use a value that represents one to two percent of the total number of available cylinders on the unit.

```
   (SSN) Starting Sector Number [0-0FFFFFFFFH]       0H
```

This parameter specifies the starting sector number for this unit. Change the default only if you are creating multiple units on the same disk.

```
   (BTI) Bad Track Information [Yes/No]              YES
```

This parameter specifies whether the disk controller, when it formats the disk, uses the factory-generated bad-track information stored on the disk.

Specify "Yes" if the disk contains valid bad-track information. Specify "No" if you do not want the disk controller to use the factory-generated bad-track information during the process of formatting the disk.

See the FORMAT command for information on how to put Bad Track Information on a disk.

Query Screen

After you have completed the unit information screen, the query screen is displayed. It contains only one line:

```
   Do you want any/more iSBC 220 UNITs ?
```

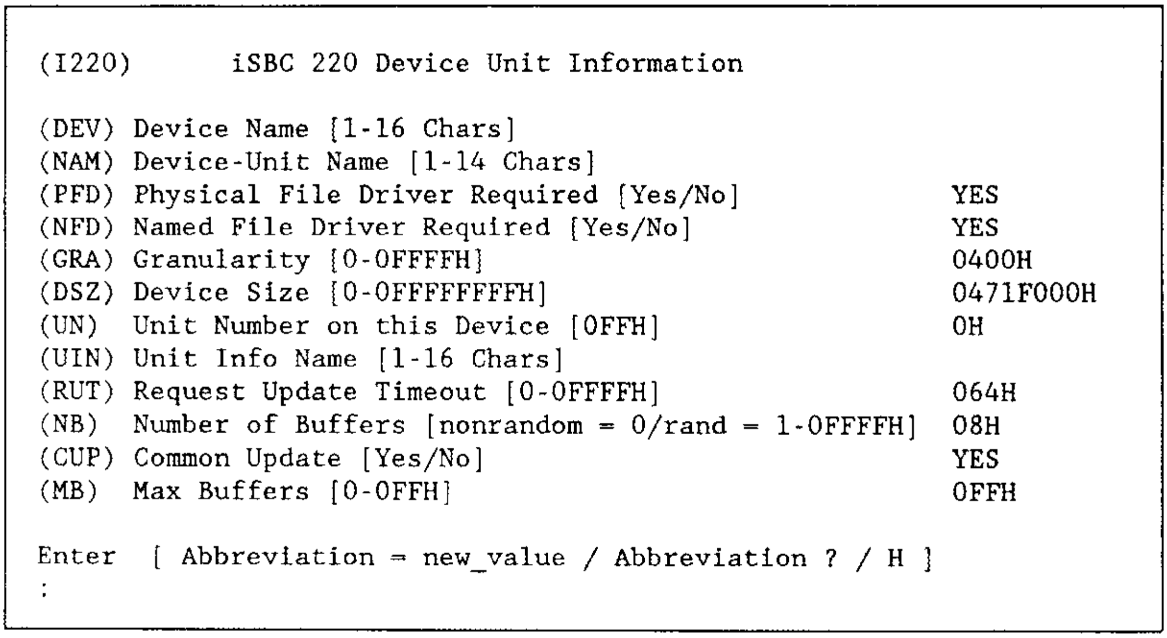
Respond to this prompt with a "Yes" if you need another unit information table for this device. Otherwise, respond to this prompt with a "No".

iSBC® 220 Device-Unit Information Screen

The ICU uses the information from the following screen to create a device unit information block (DUIB) for the iSBC 220 driver.

Use the "iSBC 220 Device-Unit Information" screen to define a device-unit information block for each unique unit in your system. In addition, use this screen to define a "generic" storage module device. A "generic" unit allows you to use a version of the Operating System that does not need to know the specific characteristics of your SMD. This means that you can run the same configuration on systems that contain different kinds of storage module device disks. This general feature is called automatic device characteristics recognition. Refer to the *Operator's Guide to the Extended iRMX II Human Interface* for additional information about automatic device characteristics recognition.

Automatic device characteristics recognition also allows you to configure a version of the Bootstrap Loader without knowing the specific characteristics of a SMD. Refer to Chapter 8 in this manual for information about automatic boot device recognition. Refer to the *Extended iRMX II Bootstrap Loader Reference Manual* for information about configuring the Bootstrap Loader. If you use the automatic boot device recognition feature, only the drive from which you wish to load the Bootstrap Loader needs to the default values shown in the following screen.



```
(I220)      iSBC 220 Device Unit Information

(DEV) Device Name [1-16 Chars]
(NAM) Device-Unit Name [1-14 Chars]
(PFD) Physical File Driver Required [Yes/No]                YES
(NFD) Named File Driver Required [Yes/No]                   YES
(GRA) Granularity [0-0FFFFH]                                0400H
(DSZ) Device Size [0-0FFFFFFFFH]                            0471F000H
(UN)  Unit Number on this Device [0FFH]                     0H
(UIN) Unit Info Name [1-16 Chars]
(RUT) Request Update Timeout [0-0FFFFH]                     064H
(NB)  Number of Buffers [nonrandom = 0/rand = 1-0FFFFH]     08H
(CUP) Common Update [Yes/No]                                YES
(MB)  Max Buffers [0-0FFH]                                  0FFH

Enter  [ Abbreviation = new_value / Abbreviation ? / H ]
:
```

```
(DEV)  Device Name [1-16 Chars]
```

The name you enter for this parameter must be the same name you entered in the "DEV" parameter on the "iSBC 220 Driver" screen. This name provides the logical ICU connection between a driver and all of its DUIBs. Thus, when a device is deleted it is possible to delete all the related units and DUIBs at the same time.

```
(NAM) Device-Unit Name [1-14 Chars]
```

This parameter specifies a unique name that identifies this device-unit to the I/O System. If you want the Extended I/O System to logically attach this device during initialization, the name you specify for this parameter must also be one of the device names you specified on the Logical Names screen (see Chapter 8).

The ICU allows you to enter from one to fourteen characters. Refer to the *ASM286 Language Reference Manual* for rules regarding this name.

The name you specify for this parameter is the physical name you specify when invoking the Human Interface ATTACHDEVICE command, the BIOS A$PHYSICAL$ATTACH$DEVICE system call, or the EIOS LOGICAL$ATTACH$DEVICE system call.

```
    (PFD) Physical File Driver Required [Yes/No]     YES
    (NFD) Named File Driver Required [Yes/No]        YES
```

This driver supports both named and physical file drivers. These parameter lines let you specify "Yes" to at least one of these parameters.

Two good reasons exist for not changing either "Yes" default value. First, the Human Interface ATTACHDEVICE command, the EIOS LOGICAL$ATTACH$DEVICE system call, and the BIOS A$PHYSICAL$ATTACH$DEVICE system call require you to select the file type at run time. Second, there is no code savings if you specify "Yes" to one parameter and "No" to the other.

```
    (GRA) Granularity [0-0FFFFH]                     0400H
```

This parameter specifies the minimum number of bytes that the device reads or writes in one operation. This value is also called device granularity. Device granularity determines sector size and multiples of device granularity define volume and file granularity. Sector size is related to device size in the following manner: sector size multiplied by sectors per track multiplied by tracks per side equals device size.

The vendor of your drive may recommend optimum values for this parameter. Refer to vendor documentation for additional information.

The default value 0400H is equivalent to 1024 decimal. Refer to the *Extended iRMX II Basic I/O System Calls Reference Manual* for more information about granularity.

```
    (DSZ) Device Size [0-0FFFFFFFFH]                 0471F000H
```

This parameter specifies the device storage capacity in bytes. The device size varies with the granularity and sectors per track. Use the following formula to compute DSZ.

DSZ = [total cylinders - alternate cylinders] * number of heads
* sectors per track * device granularity

```
    (UN)  Unit Number on this Device [0-0FFH]        0H
```

This parameter specifies the unit number of this device-unit. This number identifies one of 8 possible units on this device. The unit numbers for the device begin with zero and increase sequentially.

Unit Numbers	on Drive
0-3 are fixed disks	1-4
4-7 are removable disks	1-4

Even though the ICU allows numbers from 0 to FFH, the greatest meaningful value that you can enter is seven.

```
    (UIN) Unit Info Name [1-16 Chars]
```

This parameter specifies the "Unit Info Name" of a unit information table that fills the needs of this DUIB. The following rules apply to the name you choose:

- You must create a unit information table with this "unit info name".
- Each DUIB can point to only one unit information table.
- A particular unit information table can be pointed to by more than one DUIB.

```
    (RUT) Request Update Timeout [0-0FFFFH]         064H
```

This parameter specifies the number of clock ticks that the driver should wait (during a pause in activity) before updating the attached files on the device. The default value 064H is equivalent to 100 decimal.

The values 0 and 0FFFFH do not indicate time intervals. A value of 0 indicates that the driver will never leave any data buffered in memory (eliminating the need to update). A value of 0FFFFH indicates that updates on this device will occur only when a file is detached. However, the ICU treats your response to this parameter totally separate from the response made to the "Common Update" parameter (you can specify both).

The update capability provided by this parameter differs from the common update capability described in Chapter 9. Unlike the ability to update at fixed periods that is provided by the common update feature, the update capability (referred to by the "Request Update Timeout" parameter) allows the driver to update based on pauses in activity. If there is a pause in activity on this device, the driver determines how soon the common update would occur and compares that time interval to the request update timeout value. The driver then waits the shorter of the two intervals and updates the attached files on the device. Thus, if you use the common update feature, the time interval you specify for the "Request Update Timeout" should be shorter than time interval specified for the "Common Update Timeout".

```
    (NB)  Number of Buffers [ 1-0FFFFH]                06H
```

This parameter specifies the number of buffers this device has for blocking and deblocking I/O requests. The default value of 06H is appropriate for a unit that has a granularity of 512 bytes. Since this is a device that supports random access, do not specify a value of zero.

The number of buffers prompt determines the number of Basic I/O System buffers the device uses for I/O. These buffers are general-purpose cache buffers that the Basic I/O System uses to increase I/O performance. The size of each buffer is 32 bytes greater than the device granularity you specified for this device (your response to the "(GRA) Granularity" parameter). The buffers are associated with a device-unit, not with a particular file. The Basic I/O System assigns these buffers to the device-unit when it attaches the device-unit.

Because the Operating System uses memory from the BIOS memory pool to create these buffers, you should increase the values you specified on the "BIOS" screen for both the minimum and maximum memory pool sizes if you increase the default value for the "Number of Buffers" parameter. An explanation of how to calculate memory usage based on your responses to the "(GRA) Granularity" and the "(NB) Number of Buffers" parameter lines is in Chapter 9.

Choosing an optimum number of buffers depends on many factors: how your application works; the speed of your components, your CPU, your memory, and your peripherals; the performance characteristics of your peripherals; and the number of users. Plus, you must trade off optimum performance against the amount of memory you use (because the more buffers you choose, the more memory you need).

Because there are so many factors, the only way that you can obtain the optimum value for your system is by testing different values and fine-tuning your configuration. Intel recommends that you use at least two buffers for this device.

```
    (CUP) Common Update [Yes/No]                       YES
```

Specify "Yes" if you want the driver to update the attached files on this device at the fixed interval you specified on the BIOS screen. Specify "No" if you do not want to update attached files on this device at fixed intervals, but prefer to update attached files based only on the time interval you specified for the "Requested Update Timeout" parameter. You should not specify "No" for this parameter and 0FFFFH for the "Request Update Timeout" parameter. See Chapter 9 for information about common update timeout (updating attached files at fixed intervals).

```
    (MB) Max Buffers [0-0FFH]                          0FFH
```

This parameter specifies the maximum number of buffers that the Extended I/O System can allocate for this device's I/O. The default value for this device (0FFH) enables the S$OPEN system call to specify the actual number of Extended I/O System buffers. The operating system takes memory required for these buffers from the calling job's memory pool, so by setting this parameter to 0FFH you allow the calling job to select the number of buffers based on its own memory pool size. It is recommended that you use the default value.

Query Screen

After you have completed the DUIB information screen, the query screen is displayed. It contains only one line:

```
    Do you want any/more iSBC 220 DUIBs ?
```

Respond "Yes" to this prompt if you plan to use this controller with two devices that have different characteristics, or devices that have the same characteristics, but different unit numbers.

While developing your initial systems, you can create as many device-unit information blocks as you want. The number of DUIBs can exceed the number of devices on your system. The particular DUIB associated with the device depends on the physical name you use when attaching it. Once you know that you will not need a particular DUIB, save memory by deleting it from your definition file before you generate your configuration files.

iSBX™ 218A Driver Parameters

The iSBX 218A flexible disk driver supports 5.25-inch disks (single- or double-sided, single- or double-density). Three screens define the interface between the iSBX 218A random access device driver and the I/O system. These screens relate to the three device configuration tables: the device information table, the unit information table, and the device unit information block (DUIB). Refer to the *Extended iRMX II Device Drivers User's Guide* for further information about these tables.

iSBX™ 218A Driver Screen

The ICU uses the information from the following screen to create a device information table for the iSBX 218A driver. If your system includes more than one iSBX 218A MULTIMODULE board, you must specify a unique interrupt level and port address for each MULTIMODULE board.

```
(D218)      iSBX 218A Driver

(DEV) Device Name [1-16 Chars]
(IL)  Interrupt Level [Encoded Level]                074H
(ITP) Interrupt Task Priority [0-255]                130
(PA)  Port Address [0-0FFFFH]                        080H
(MDV) Motor Delay Value [0-0FFFFH]                   01AH

Enter  [ Abbreviation = new_value / Abbreviation ? / H ]
:
```

```
   (DEV) Device Name [1-16 Chars]
```

This parameter provides a means of associating the device and all the units and DUIBs which belong to it. The ICU stores all the screens by type, not by device. This means that all device driver screens are stored together, all unit information screens and all DUIB screens. The ICU "knows" which unit information and device-unit information screen relate to a particular device driver by the device name you enter in the "DEV" parameter.

```
   (IL)  Interrupt Level [Encoded Level]           074H
```

This parameter specifies the encoded interrupt level for the iSBX 218A driver. The interrupt task uses this value to associate the interrupt task with the correct interrupt level. The default value 074H (0000 0000 0111 0100 binary) specifies slave interrupt level 4.

The possible values for this field are encoded as follows (where bit 0 is the low-order bit):

Master Level	Code	Slave Level	Attached Code	Master Level	Code	Slave Level	Attached Code
0	0008H	0-7	0000-0007H	4	0048H	0-7	0040-0047H
1	0018H	0-7	0010-0017H	5	0058H	0-7	0050-0057H
2	0028H	0-7	0020-0027H	6	0068H	0-7	0060-0067H
3	0038H	0-7	0030-0037H	7	0078H	0-7	0070-0077H

```
    (ITP) Interrupt Task Priority [0-255]            130
```

This parameter specifies the initial priority of the device's interrupt task. The actual priority of the iRMX II interrupt task changes because the Nucleus adjusts an interrupt task's priority according to the interrupt level that it services. The default value is 130 decimal.

```
    (PA)  Port Address [0-0FFFFH]                    0080H
```

This parameter specifies the address of the I/O port through which the Operating System communicates with the board. Set this value to match the address of the iSBX connector on your processor board.

```
    (MDV) Motor Delay Value [0-0FFFFH]               01AH
```

To support 5.25-inch disks, you must specify the number of system clock intervals the controller should wait after turning on a minifloppy motor but before accessing the drive. Typical values are from 50 (decimal) to 100 (decimal) clock ticks.

Query Screen

After you have completed the "iSBX 218A Driver" screen, the query screen is displayed. It contains only one line:

```
    Do you want any/more iSBX 218A DEVICEs ?
```

Respond to this prompt with a "Yes" if you wish to add another device driver of this type.

iSBX™ 218A Unit Information

The ICU uses the information from the following screen to create a unit information table for the iSBX 218A driver.

```
(U218)     iSBX 218A Unit Information

(DEV) Device Name [1-16 Chars]
(NAM) Unit Info Name [1-16 Chars]
(MR)  Maximum Retries [0-0FFFFH]                     09H
(NT)  Number of Tracks per Side [0-0FFFFH]           028H
(NS)  Number of Sectors/Track [0-0FFFFH]             08H
(SR)  Step Rate [0-0FFH]                             014H
(HLT) Head Load Time [0-0FFH]                        FH
(HUT) Head Unload Time [0-0FFH]                      0F0H

Enter [ Abbreviation = new_value / Abbreviation ? / H ]
:
```

NOTE

The values listed for Step Rate (SR) and Head Load Time (HLT) assume that your system uses 60 Hz A.C. electricity. If you are using 50 Hz A.C. power, you must increase both the Step Rate value and the Head Load Time value by 25%.

```
    (DEV)  Device Name [1-16 Chars]
```

The name you enter for this parameter must be the same name you entered in the "DEV" parameter on the driver screen. This name provides the logical ICU connection between a driver and all of its units. Thus, when a device is deleted it is possible to delete all the related units at the same time.

```
    (NAM) Unit Info Name [1-16 Chars]
```

This parameter specifies a unique unit information name for this device. The first character must be an alphabetic character. Refer to the *ASM286 Language Reference Manual* for rules regarding this name. This name should help identify this iSBX 218A unit information table versus any another unit information table. Examples of names that are recommended for the iSBX 218A Controller are listed in Table 10-7.

A DUIB uses the "Unit Info Name" to point to this particular unit information table. When developing your initial systems, you can create unit information tables that are never pointed to by a DUIB. There is no harm in this process during the development stages. However, when you configure your final system, eliminate all unused unit information tables to save memory.

Table 10-7. iSBX™ 218A Controller Unit Information

UNIT INFORMATION					
Unit Info Name	No Tracks per Side	Heads/ Removable	Sectors/ Track	Head Load Time	Step Rate
uinfo 218mf	40	2	8	15	20
uinfo 218mfdy	80	2	8	15	6

```
    (MR)  Maximum Retries [0-0FFFFH]                  09H
```

This parameter specifies the maximum number of times the Basic I/O System should retry an operation before returning an E$IO$SOFT exception code. Refer to the *Extended iRMX II Basic I/O System Calls Reference Manual* for related information on the E$IO$SOFT exception code. The default value of 9 is recommended.

```
    (NT)  Number of Tracks per Side [0-0FFFFH]        028H
```

This parameter specifies the number of tracks per side of an 5.25-inch disk. The default value 028H is equivalent to 40 decimal. Table 10-7 shows some of the typical values for this parameter. Table 10-7 does not include all possible values.

```
    (NS)  Number of Sectors/Track [0-0FFFFH]          08H
```

This parameter specifies the number of sectors per track. The number of sectors per track varies with the sector size, the density, and the drives overall size. Table 10-7 lists some of the values possible.

```
    (SR)  Step Rate [0-0FFH]                          014H
```

This parameter specifies the time interval (in milliseconds) between step pulses as they relate to track-to-track access time.

```
    (HLT)  Head Load Time [0-0FFH]                    FH
```

This parameter specifies the time interval (in milliseconds) that the controller waits after loading the head but before initiating a read or write operation. The default value of 0FH is equivalent to 15 decimal. Note that head load time includes a provision for head settling time.

```
    (HUT)  Head Unload Time [0-0FFH]                    0F0H
```

This parameter specifies the time interval (in milliseconds) that the controller must wait before unloading the head. During this interval, the controller could initiate another read or write operation if the head is in the right position. The default value 0F0H is equivalent to 240 decimal.

Query Screen

After you have completed the "iSBX 218A Unit Information" screen, the query screen is displayed. It contains only one line:

```
    Do you want any/more iSBX 218A UNITs  ?
```

Respond to this prompt with a "Yes" if you need another unit information table for this device. Otherwise, respond to this prompt with a "No".

iSBX™ 218A Device-Unit Information

The ICU uses the information from the following screen to create a device unit information block (DUIB) for the iSBX 218A driver.

```
(I218)    iSBX 218A DUIB Information

(DEV) Device Name [1-16 Chars]
(NAM) Device-Unit Name [1-14 Chars]
(PFD) Physical File Driver Required [Yes/No]                YES
(NFD) Named File Driver Required [Yes/No]                   YES
(SDD) Single or Double Density Disks [Single/Double]        DOUBLE
(SDS) Single or Double Sided Disks [Single/Double]          DOUBLE
(EFI) 8 or 5 Inch Disks [8/5]                               5
(SUF) Standard or Uniform Format [Standard/Uniform]         STANDARD
(GRA) Granularity [0-0FFFFH]                                0200H
(DSZ) Device Size [0-0FFFFFFFFH]                            04F8000H
(UN)  Unit Number on this Device [0-0FFH]                   0H
(UIN) Unit Info Name [1-16 Chars]
(RUT) Request Update Timeout [0-0FFFFH]                     064H
(NB)  Number of Buffers [nonrandom = 0/rand = 1-0FFFFH]     06H
(CUP) Common Update [Yes/No]                                YES
(MB)  Max Buffers [0-0FFH]                                  0FFH

Enter [ Abbreviation = new_value / Abbreviation ? / H ]
:
```

```
(DEV)  Device Name [1-16 Chars]
```

The name you enter for this parameter must be the same name you entered in the "DEV" parameter on the driver screen. This name provides the logical ICU connection between a driver and all of its DUIBs. Thus, when a device is deleted it is possible to delete all the related units and DUIBs at the same time.

```
(NAM) Device-Unit Name [1-14 Chars]
```

This parameter specifies a name that uniquely identifies this device-unit to the I/O System. If you want the Extended I/O System to logically attach this device during initialization, the name you specify for this parameter must also be one of the device names you specified on the "Logical Names" screen (see Chapter 8).

The ICU allows you to enter from one to fourteen characters. Refer to the *ASM286 Language Reference Manual* for rules regarding this name. Examples of device-unit names that are recommended for the iSBX 218A Controller are listed in Table 10-8.

The name you specify for this parameter is the physical name you specify when invoking the Human Interface ATTACHDEVICE command, the BIOS A$PHYSICAL$ATTACH$DEVICE system call, or the EIOS LOGICAL$ATTACH$DEVICE system call.

```
    (PFD) Physical File Driver Required [Yes/No]     YES
    (NFD) Named File Driver Required [Yes/No]        YES
```

This driver supports both named and physical file drivers. These parameter lines let you specify "Yes" to at least one of these parameters.

Two good reasons exist for not changing either "Yes" default value. First, the Human Interface ATTACHDEVICE command, the EIOS LOGICAL$ATTACH$DEVICE system call, and the BIOS A$PHYSICAL$ATTACH$DEVICE system call require you to select the file type at run time. Second, there is no code savings if you specify "Yes" to one parameter and "No" to the other.

Note that the ICU ignores your response to the "Named File Driver Required" parameter if you specify a "Uniform" format (see section on Standard or Uniform Format).

```
    (SDD) Single or Double Density Disks [Single/Double]  DOUBLE
```

This parameter specifies the recording density of this flexible disk. The recording density is either single or double. Table 10-8 shows how your response to this prompt is related to other parameters.

Table 10-8. iSBX™ 218A Controller DUIB Information

D U I B INFORMATION							
Device-unit Name(s)	Size	Density	Sides Size	Gran.	Device Name	Unit-Info Buffers	Num
PMF0 PMFDY0	5.25 5.25	Double Double	Double Double	512 512	325632 653312	uinfo_218mf uinfo_218mfdy	8 8

```
    (SDS) Single or Double Sided Disks [Single/Double]  DOUBLE
```

This parameter specifies the number of recording surfaces for this flexible disk. Table 10-8 shows how your response to this prompt is related to other parameters.

```
    (EFI) 8 or 5 Inch Disks [8/5]                        5
```

This parameter specifies the size of the flexible disk. This driver does not support 8-inch drives.

```
   (EFI) 8 or 5 Inch Disks [8/5]                          5
```

This parameter specifies the size of the flexible disk. This driver does not support 8-inch drives.

```
   (SUF) Standard or Uniform Format [Standard/Uniform]  STANDARD
```

This parameter specifies the format that the controller should expect on track zero. Choose the option "Standard" if you want track zero to be single density with 128-byte sectors (regardless of how the remaining tracks are formatted). Choose the option "Uniform" if you want all tracks on a disk to have the same format.

Normally, when you use the FORMAT command to format a disk as a named volume, the command formats track zero with a fixed density (single density) and a fixed sector size (128 bytes). This is the "standard" format. It is recommended that you use this format.

If you wish to read a disk that has a uniform format, you should use the "Standard/Uniform Format" parameter to designate a "Uniform" format and attach the drive as a physical file. Note that automatic device characteristics recognition does not work unless track zero is single density with 128-byte sectors.

Note that booting a system from a flexible disk drive is supported only with the standard format.

```
   (GRA) Granularity [0-0FFFFH]                        0200H
```

This parameter specifies the minimum number of bytes that the device reads or writes in one operation. This value is also called device granularity. Device granularity determines sector size and multiples of device granularity define volume and file granularities.

The vendor of your drive may recommend optimum values for this parameter. Refer to vendor documentation for additional information.

The default value 0200H is equivalent to 512 decimal. Table 10-8 shows how this value relates to other parameters. Refer to the *Extended iRMX II Basic I/O System Calls Reference Manual* for more information about granularity.

```
    (DSZ) Device Size [0-0FFFFFFFFH]                04F800H
```

This parameter specifies the device storage capacity in bytes. The device size varies with the granularity and sectors per track. For flexible disk drives, use the following formula:

DSZ = total cylinders * number of heads * sectors per track
* device granularity

```
    (UN)  Unit Number on this Device [0-0FFH]       0H
```

This parameter specifies the unit number of the device-unit. This number identifies one of four possible units on this device. The unit numbers for the device begin with zero and increase sequentially. The typical unit numbers are zero through three. Even though the ICU allows numbers from 0 to FFH, the greatest meaningful value that you can enter is three.

```
    (UIN) Unit Info Name [1-16 Chars]
```

This parameter specifies the "Unit Info Name" of a unit information table that fills the needs of this DUIB. The following rules apply to the name you choose:

- You must create a unit information table with this "unit info name".
- Each DUIB can point to only one unit information table.
- A particular unit information table can be pointed to by more than one DUIB.

```
    (RUT) Request Update Timeout [0-0FFFFH]        064H
```

This parameter specifies the number of clock ticks that the driver should wait (during a pause in activity) before updating the attached files on the device. The default value 064H is equivalent to 100 decimal.

The values 0 and 0FFFFH do not indicate time intervals. A value of 0 indicates that the driver will never leave any data buffered in memory (eliminating the need to update). A value of 0FFFFH indicates that updates on this device will occur only when a file is detached. However, the ICU treats your response to this parameter totally separate from the response made to the "Common Update" parameter (you can specify both).

The update capability provided by this parameter differs from the common update capability described in Chapter 9. Unlike the ability to update at fixed periods that is provided by the common update feature, the update capability (referred to by the "Request Update Timeout") parameter allows the driver to update based on pauses in activity. If there is a pause in activity on this device, the driver determines how soon the common update would occur and compares that time interval to the request update timeout value. The driver then waits the shorter of the two intervals and updates the attached files on the device. Thus, if you use the common update feature, the time interval you specify for the "Request Update Timeout" should be shorter than time interval specified for the "Common Update Timeout".

```
    (NB)  Number of Buffers [nonrandom = 0/rand = 1-0FFFFH]   06H
```

This parameter specifies the number of buffers this device has for blocking and deblocking I/O requests. The default value of 6 is appropriate for a unit that has a granularity of 512 bytes. Table 10-8 shows how this value relates to other parameters. Since this is a device which supports random access, do not specify a value of zero.

The number of buffers prompt determines the number of Basic I/O System buffers the device uses for I/O. These buffers are general-purpose cache buffers that the Basic I/O System uses to increase I/O performance. The size of each buffer is 32 bytes greater than the device granularity you specified for this device (your response to the "Granularity" parameter). The buffers are associated with a device-unit, not with a particular file. The Basic I/O System assigns these buffers to the device-unit when it attaches the device-unit.

Because the Operating System uses memory from the BIOS memory pool to create these buffers, you should increase the values you specified on the "BIOS" screen for both the minimum and maximum memory pool sizes if you increase the default value for the "Number of Buffers" parameter. An explanation of how to calculate memory usage based on your responses to the "Granularity" and the "Number of Buffers" parameter lines is in Chapter 9.

Choosing an optimum number of buffers depends on many factors: how your application works; the speed of your components, your CPU, your memory, and your peripherals; the performance characteristics of your peripherals; and the number of users. Plus, you must trade off optimum performance against the amount of memory you use. The more buffers you choose, the more memory you need.

Because there are so many factors, the only way that you can obtain the optimum value for your system is by testing different values and fine-tuning your configuration. It is recommended that you use at least two buffers for this device.

```
    (CUP) Common Update [Yes/No]                      YES
```

Specify "Yes" if you want the driver to update the attached files on this device at the fixed interval you specified on the "BIOS" screen. Specify "No" if you do not want to update attached files on this device at fixed intervals but prefer to update attached files based only on the time interval you specified for the "Request Update Timeout" parameter. You should not specify "No" for this parameter and 0FFFFH for the "Request Update Timeout" parameter. See Chapter 9 for information about common update timeout (updating attached files at fixed intervals).

```
    (MB)  Max Buffers [0-0FFH]                        0FFH
```

This parameter specifies the maximum number of buffers that the Extended I/O System can allocate for this device's I/O. The default value for this device (0FFH) allows the S$OPEN system call to specify the actual number of Extended I/O System buffers. The Operating System takes memory required for these buffers from the calling job's memory pool, so by setting this parameter to 0FFH you allow the calling job to select the number of buffers based on its own memory pool size. It is recommended that you use the default value.

Query Screen

After you have completed the "iSBX 218A DUIB Information" screen, the query screen is displayed. It contains only one line:

```
    Do you want any/more iSBX 218A DUIBs ?
```

Respond "Yes" to this prompt, if you plan to use the iSBX 218A MULTIMODULE board with device-units that have different characteristics or if your system contains two devices with the same characteristics, but different unit numbers.

While developing your initial systems, you can create as many device-unit information blocks as you want. The number of DUIBs can exceed the number of devices on your system.

iSBC® 208 Driver Parameters

The iSBC 208 is a flexible diskette controller. It supports 5.25-inch and 8-inch disks (single- or double-sided, single- or double-density). Three screens define the interface between the iSBC 208 random access device driver and the I/O system. These screens relate to the three device configuration tables: the device information table, the unit information table, and the device unit information block (DUIB). Refer to the *Extended iRMX II Device Drivers User's Guide* for more information about these tables.

The iSBC 208 is not supplied in the 28612.def definition file. The default values you see displayed in the following sections are the default values supplied by the ICU.

iSBC® 208 Driver Screen

The ICU uses the information from the following screen to create a device information table for the iSBC 208 driver. If your system includes more than one iSBC 208 controller, you must specify a unique interrupt level and port address for each controller.

```
(D208)        iSBC 208 Driver

(DEV) Device Name [1-16 Chars]
(IL)  Interrupt Level [Encoded Level]            048H
(ITP) Interrupt Task Priority [0-255]            130
(PA)  Port Address [0-0FFFFH]                    0180H
(MDV) Motor Delay Value [0-0FFFFH]               050H
(BBA) Boundary Buffer Address [0-0FFFFFFFFH]     01600H

Enter  [ Abbreviation = new_value / Abbreviation ? / H ]
:
```

```
   (DEV) Device Name [1-16 Chars]
```

This parameter provides a means of associating the device and all the units and DUIBs which belong to it. The ICU stores all the screens by type, not by device. This means that all device driver screens are stored together, all unit information screens and all DUIB screens. The ICU "knows" which unit information and device-unit information screen relate to a particular device driver by the device name you enter in the "DEV" parameter.

```
(IL)  Interrupt Level [Encoded Level]          0048H
```

This parameter specifies the encoded interrupt level for the iSBC 208 driver. The interrupt task uses this value to associate the interrupt task with the correct interrupt level. The default value 0048H (0000 0000 0100 1000 binary) specifies master interrupt level 4. The possible values for this field are encoded as follows (where bit 0 is the low-order bit):

Master Level	Code	Slave Level	Attached Code	Master Level	Code	Slave Level	Attached Code
0	0008H	0-7	000-0007H	4	0048H	0-7	0040-0047H
1	0018H	0-7	010-0017H	5	0058H	0-7	0050-0057H
2	0028H	0-7	020-0027H	6	0068H	0-7	0060-0067H
3	0038H	0-7	030-0037H	7	0078H	0-7	0070-0077H

```
(ITP) Interrupt Task Priority [0-255]          130
```

This parameter specifies the initial priority of the device's interrupt task. The default value is 130 decimal. The actual priority of the interrupt task changes because the Nucleus adjusts an interrupt task's priority according to the interrupt level that it services.

```
(PA)  Port Address [0-0FFFFH]                  0180H
```

This parameter specifies the address of the I/O port through which the Operating System communicates with the board. Set this value to match the switch setting on the iSBC 208 board.

```
(MDV) Motor Delay Value [0-0FFFFH]             050H
```

If you are using your iSBC 208 controller to support 5.25-inch disks, you must specify the number of system clock intervals the controller should wait after turning on a minifloppy motor but before accessing the drive. This value must be 032H.

Note that there is no need for this value when controlling an 8-inch disk drive.

```
    (BBA) Boundary Buffer Address [0-0FFFFFFFFH]     01600H
```

This parameter specifies the address of a 1K buffer that is in the low megabyte of memory. The iSBC 208 driver cannot directly access addresses across a megabyte boundary. Therefore, whenever a read or write request is made that causes the iSBC 208 driver to access a memory location across a megabyte boundary, the buffer specified in this parameter is used instead of the actual memory. This may cause the request to be broken into a number of sub-requests depending on whether or not whole sectors of information were requested before or after the memory on the boundary. One sector of information is read from or written to this buffer at a time. These reads are performed in on one track at a time and in multiples of device granularity.

Do not include memory in the memory you define on the "Memory for System" screen or "Memory for Free Space Manager" screen or any other I/O controller.

Query Screen

After you have completed the "iSBC 208 Driver" screen, the query screen is displayed. It contains only one line:

```
    Do you want any/more iSBC 208 DEVICEs ?
```

Respond to this prompt with a "Yes" if you wish to add another device driver of this type.

iSBC® 208 Unit Information Screen

The ICU uses the information from the following screen to create a unit information table for the iSBC 208 driver.

```
(U208)    iSBC 208 Unit Information

(DEV) Device Name [1-16 Chars]
(NAM) Unit Info Name [1-16 Chars]
(MR)  Maximum Retries [0-0FFFFH]                     09H
(CS)  Cylinder Size [0-0FFFFH]                       010H
(NT)  Number of Tracks per Side [0-0FFFFH]           028H
(NS)  Number of Sectors/Track [0-0FFFFH]             08H
(SR)  Step Rate [0-0FFH]                             0AH
(HLT) Head Load Time [0-0FFH]                        023H
(HUT) Head Unload Time [0-0FFH]                      0F0H

Enter [ Abbreviation = new_value / Abbreviation ? / H ]
:
```

NOTE

The values listed for Step Rate (SR) and Head Load Time (HLT) assume that your system uses 60 Hz A.C. electricity. If you are using 50 Hz A.C. power, you must increase both the Step Rate value and the Head Load Time value by 25%.

```
    (DEV)  Device Name [1-16 Chars]
```

The name you enter for this parameter must be the same name you entered in the "DEV" parameter on the driver screen. This name provides the logical ICU connection between a driver and all of its units. Thus, when a device is deleted it is possible to delete all the related units at the same time.

```
    (NAM) Unit Info Name [1-16 Chars]
```

This parameter specifies a unique unit information name for this device. The first character must be an alphabetic character. Refer to the *ASM286 Language Reference Manual* for rules regarding this name. This name should help identify this iSBC 208 unit information table versus any other unit information table. Examples of names that are recommended for the iSBC 208 Controller are listed in Table 10-9.

A DUIB uses the "Unit Info Name" to point to this particular unit information table. When developing your initial systems, you can create unit information tables that are never pointed to by a DUIB. There is no harm in this process during the development stages. However, when you configure your final system, eliminate all unused unit information tables to save memory.

Table 10-9. iSBC® 208 Controller Unit Information

D U I B Information					
Unit-Info Name	Cylinder Size	No. Tracks per Side	No. Sectors per Track	Step Rate	Head Load Time
uinfo 208f	26	77	26	8	40
uinfo 208fd	52	77	26	4	40
uinfo 208fdx	16	77	8	4	40
uinfo 208mf	16	40	8	20	15
uinfo 208mfdy	16	80	8	6	15

```
    (MR)  Maximum Retries [0-0FFFFH]            09H
```

This parameter specifies the maximum number of times the Basic I/O System should retry an operation before returning an E$IO$SOFT exception code. Refer to the *Extended iRMX II Basic I/O System User's Guide* for related information on the E$IO$SOFT exception code. Refer to the *iSBC 208 Flexible Diskette Controller Hardware Reference Manual* for related information on conditions (see result phase status registers) that can cause E$IO$SOFT errors. The default value of 9 is recommended.

```
    (CS)  Cylinder Size [0-0FFFFH]              01AH
```

This parameter specifies the number of sectors per cylinder and whether the controller should perform automatic and concurrent seek operations. If you specify 0H, the I/O System will not perform either automatic seek operations or concurrent seek operations. If you specify a value other than 0H, the I/O System can perform both automatic seek operations and concurrent seek operations. Use the default value 01AH if you want the I/O System to perform concurrent seeks on single-sided 8-inch disks. Use the value 034H if you want the I/O System to perform concurrent seeks on double-sided 8-inch disks.

Automatic seek operations allow the random access software to ascertain the need to perform a seek operation before a requested read or write operation. The random access software is able to perform this function because it maintains the current location of the file pointer and can ascertain (based on the response to the "Cylinder Size" parameter) whether it should perform a read or write request within the current cylinder. If the request is not in the current cylinder, the random access software performs a seek operation.

The value you specify affects how or if automatic seek operations are performed. If you specify 0H, no automatic seek operations are performed by the random access software. Only specify a zero value if the device can perform seek functions independent of the random access software. If you specify a value of 01H, the random access software will perform automatic seek operations before each read or write request. If you specify a non-zero value other than 01H, the value you should enter is the cylinder size for this device (the number of sectors per track times the number of sides per disk). For example, if your response to the "Number of Sectors/Track" parameter is 26 decimal and your response to the "Single or Double Sided Disks" is double, then cylinder size should be 52.

If you specify a non-zero value for the "Cylinder Size" parameter, the random access software can concurrently perform seeks on more than one device unit. If you specify a zero value for the "Cylinder Size" parameter, the random access software does not attempt seek functions (including concurrent seeks).

If you are using your iSBC 208 controller to support 5.25-inch disks, you must set this value to "010H".

```
   (NT)  Number of Tracks per Side [0-0FFFFH]   04DH
```

This parameter specifies the number of tracks per side of an 8- or 5.25-inch disk. The default value 04DH is equivalent to 77 decimal. Table 10-10 shows some of the typical values for this parameter.

If you are using your iSBC 208 controller to support 5.25-inch disks, you must set this value to "028H".

```
   (NS)  Number of Sectors/Track [0-0FFFFH]     01AH
```

This parameter specifies the number of sectors per track. The number of sectors per track varies with the sector size, the density, and the overall size of the disk. The default value 01AH is equivalent to 26 decimal. Tables 10-9 and 10-10 list some of the values possible.

If you are using your iSBC 208 controller to support 5.25-inch disks, you must set this value to "08H".

Table 10-10. Flexible Disk Unit Information

	Sectors per Track			Sectors per Track		
Sector Size	Density	8-inch	5.25-inch	Density	8-inch	5.25-inch
128	Single	26	16	Double	26	16
256	Single	15	9	Double	26	16
512	Single	8	4	Double	15	8
1024	Single	4	2	Double	8	4

```
    (SR)  Step Rate [0-0FFH]                    08H
```

This parameter specifies the time interval (in milliseconds) between step pulses as they relate to track-to-track access time. If you are using your iSBC 208 controller to support 5.25-inch disks, you must set this value to "0AH".

```
    (HLT)  Head Load Time [0-0FFH]              028H
```

This parameter specifies the time interval (in milliseconds) the controller waits after loading the head but before initiating a read or write operation. The default value of 028H is equivalent to 40 decimal. Note that head load time includes a provision for head settling time.

If you are using your iSBC 208 controller to support 5.25-inch disks, you must set this value to "023H".

```
    (HUT)  Head Unload Time [0-0FFH]            0F0H
```

This parameter specifies the time interval (in milliseconds) that the controller must wait before unloading the head. During this interval, the controller could initiate another read or write operation if the head is in the right position. The default value 0F0H is equivalent to 240 decimal.

Query Screen

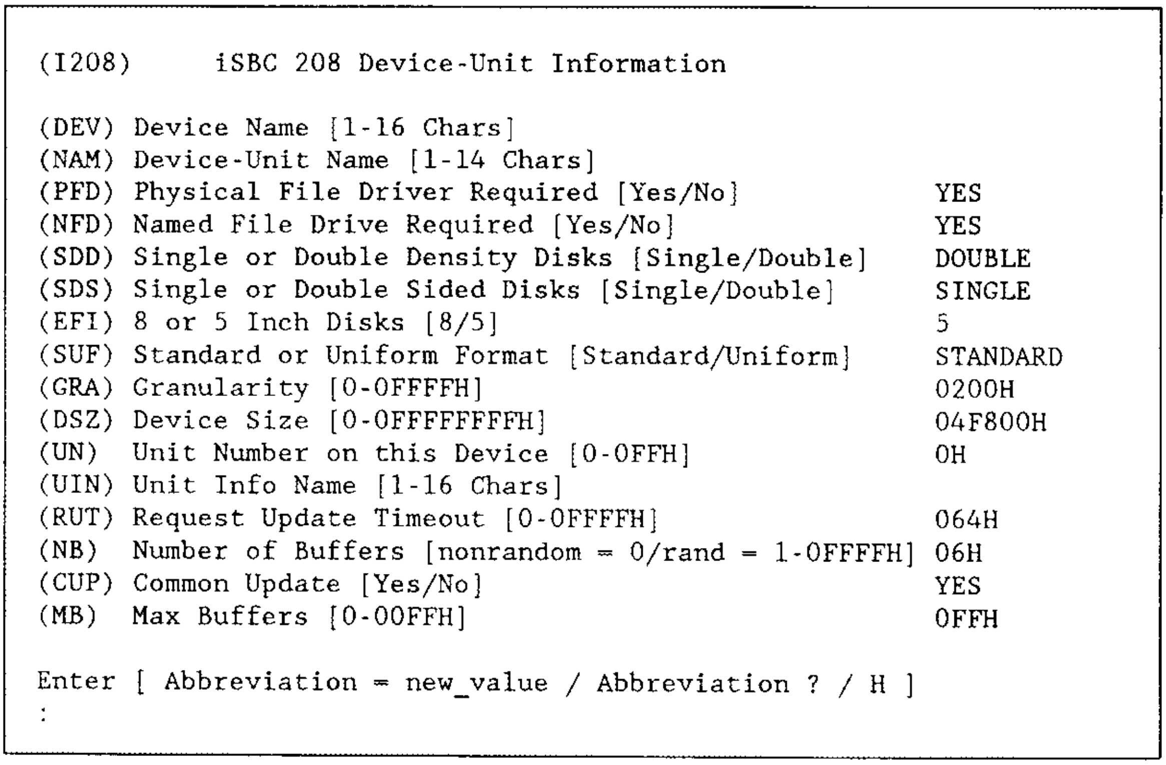
After you have completed the "iSBC 208 Unit Information" screen, the query screen is displayed. It contains only one line:

```
    Do you want any/more iSBC 208 UNITs  ?
```

Respond to this prompt with a "Yes" if you need another unit information table for this device. Otherwise, respond to this prompt with a "No".

iSBC® 208 Device-Unit Information Screen

The ICU uses the information from the following screen to create a device unit information block (DUIB) for the iSBC 208 driver.



```
(I208)      iSBC 208 Device-Unit Information

(DEV) Device Name [1-16 Chars]
(NAM) Device-Unit Name [1-14 Chars]
(PFD) Physical File Driver Required [Yes/No]              YES
(NFD) Named File Drive Required [Yes/No]                  YES
(SDD) Single or Double Density Disks [Single/Double]      DOUBLE
(SDS) Single or Double Sided Disks [Single/Double]        SINGLE
(EFI) 8 or 5 Inch Disks [8/5]                             5
(SUF) Standard or Uniform Format [Standard/Uniform]       STANDARD
(GRA) Granularity [0-0FFFFH]                              0200H
(DSZ) Device Size [0-0FFFFFFFFH]                          04F800H
(UN)  Unit Number on this Device [0-0FFH]                 0H
(UIN) Unit Info Name [1-16 Chars]
(RUT) Request Update Timeout [0-0FFFFH]                   064H
(NB)  Number of Buffers [nonrandom = 0/rand = 1-0FFFFH] 06H
(CUP) Common Update [Yes/No]                              YES
(MB)  Max Buffers [0-00FFH]                               0FFH

Enter [ Abbreviation = new_value / Abbreviation ? / H ]
:
```

(DEV) Device Name [1-16 Chars]

The name you enter for this parameter must be the same name you entered in the "DEV" parameter on the driver screen. This name provides the logical ICU connection between a driver and all of its DUIBs. Thus, when a device is deleted it is possible to delete all the related units and DUIBs at the same time.

(NAM) Device-Unit Name [1-14 Chars]

This parameter specifies a name that uniquely identifies this device-unit to the I/O System. If you want the Extended I/O System to logically attach this device during initialization, the name you specify for this parameter must also be one of the device names you specified on the "Logical Names" screen (see Chapter 8).

The ICU allows you to enter from one to fourteen characters. Refer to the *ASM286 Language Reference Manual* for rules regarding this name. Examples of device-unit names that are recommended for the iSBC 208 Controller are listed in Table 10-11.

The name you specify for this parameter is the physical name you specify when invoking the Human Interface ATTACHDEVICE command, the BIOS A$PHYSICAL$ATTACH$DEVICE system call, or the EIOS LOGICAL$ATTACH$DEVICE system call.

```
   (PFD) Physical File Driver Required [Yes/No]          YES
   (NFD) Named File Driver Required [Yes/No]             YES
```

This driver supports both named and physical file drivers. These parameter lines let you specify "Yes" to at least one of these parameters.

Two good reasons exist for not changing either "Yes" default value. First, the Human Interface ATTACHDEVICE command, the EIOS LOGICAL$ATTACH$DEVICE system call, and the BIOS A$PHYSICAL$ATTACH$DEVICE system call require you to select the file type at run time. Second, there is no code savings if you specify "Yes" to one parameter and "No" to the other.

Note that the ICU ignores your response to the "Named File Driver Required" parameter if you specify a "Uniform" format (see section on Standard or Uniform Format).

```
   (SDD) Single or Double Density Disks [Single/Double]  DOUBLE
```

This parameter specifies the recording density of this flexible disk. The recording density is either single or double. Table 10-11 shows how your response to this prompt is related to other parameters.

Table 10-11. iSBC® 208 Controller DUIB Information

D U I B Information							
Device-Unit Name(s)	Size	Density	Sides	Gran.	Device Size	Unit-Info Name	Num Buffers
AFO	8	Single	Single	128	256256	uinfo_208f	6
AFDO	8	Double	Single	256	509184	uinfo_208f	6
AFDDO	8	Double	Double	256	1021696	uinfo_208fd	6
AFDXO	8	Double	Double	1024	1255168	uinfo_208fdx	3
AMFO	5.25	Double	Double	512	325632	uinfo_208mf	6
AMFDYO	5.25	Double	Double	512	653312	uinfo_208mfdy	6

```
    (SDS) Single or Double Sided Disks [Single/Double]   SINGLE
```

This parameter specifies the number of recording surfaces for this flexible disk. Table 10-11 shows how your response to this prompt is related to other parameters.

```
    (EFI) 8 or 5 Inch Disks [8/5]                        8
```

This parameter specifies the size of the flexible disk. If you are using the iSBC 208 controller to support 5.25-inch disks, you must set this value to "5".

```
    (SUF) Standard or Uniform Format [Standard/Uniform]   STANDARD
```

This parameter specifies the format that the controller should expect on track zero. Choose the option "Standard" if you want track zero to be single density with 128-byte sectors (regardless of how the remaining tracks are formatted). Choose the option "Uniform" if you want all tracks on a disk to have the same format.

Normally, when you use the FORMAT command to format a disk as a named volume, the command formats track zero with a fixed density (single density) and a fixed sector size (128 bytes). This is the "standard" format. It is recommended that you use this format.

If you wish to read a disk that is in uniform format, you should use the "Standard/Uniform Format" parameter to designate a "Uniform" format and attach the drive as a physical file. If you choose the "Uniform" option, you should also respond "No" to the "Named File Driver Required" parameter. Note that automatic device characteristics recognition does not work unless track zero is single density with 128-byte sectors.

Note that booting from a flexible disk drive is only supported with the standard format.

```
    (GRA) Granularity [0-0FFFFH]                          0100H
```

This parameter specifies the minimum number of bytes that the device reads or writes in one operation. This value is also called device granularity. Device granularity determines sector size and multiples of device granularity define volume and file granularity. Sector size is related to device size in the following manner: sector size multiplied by sectors per track multiplied by tracks per size equals device size.

The vendor of your drive may recommend optimum values for this parameter. Refer to vendor documentation for additional information. If you are using the iSBC 208 controller to support 5.25-inch disks, you must set this value to "0200H".

The default value 0100H is equivalent to 256 decimal. Table 10-11 shows how this value relates to other parameters. Refer to the *Extended iRMX II Basic I/O System User's Guide* for more information about granularity.

```
  (DSZ) Device Size [0-0FFFFFFFFH]                    07C500H
```

This parameter specifies the device storage capacity in bytes. The device size varies with the granularity and sectors per track. Use the following formula to compute the device size:

DSZ = number of cylinders * number of heads * sectors per track
* device granularity

Table 10-11 shows the relationships between these factors.

If you are using the iSBC 208 controller to support 5.25-inch disks, you must set this value to "0200H".

```
  (UN)  Unit Number on this Device [0-0FFH]           0H
```

This parameter specifies the unit number of this device-unit. This number identifies one of four possible units on this device. The unit numbers for the device begin with zero and increase sequentially. The typical unit numbers are zero through three. Even though the ICU allows numbers from 0 to FFH, the greatest meaningful value that you can enter is three.

```
  (UIN) Unit Info Name [1-16 Chars]
```

This parameter specifies the "Unit Info Name" of a unit information table that fills the needs of this DUIB. The following rules apply to the name you choose:

- You must create a unit information table with this "unit info name".
- Each DUIB can point only to one unit information table.
- A particular unit information table can be pointed to by more than one DUIB.

```
  (RUT) Request Update Timeout [0-0FFFFH]             064H
```

This parameter specifies the number of clock ticks the driver should wait (during a pause in activity) before updating the attached files on the device. The default value 064H is equivalent to 100 decimal. If you are using the iSBC 208 controller to support 5.25-inch disks, you must use the default value.

The values 0 and 0FFFFH do not indicate time intervals. A value of 0 indicates that the driver will never leave any data buffered in memory (eliminating the need to update). A value of 0FFFFH indicates that updates on this device will occur only when a file is detached.

The update capability provided by this parameter differs from the common update capability described in Chapter 9. Unlike the ability to update at fixed periods that is provided by the common update feature, the update capability (referred to by the "Request Update Timeout" parameter)allows the driver to update based on pauses in activity. If there is a pause in activity on this device, the driver determines how soon the common update would occur and compares that time interval to the request update timeout value. The driver then waits the shorter of the two intervals and updates the attached files on the device. Thus, if you use the common update feature, the time interval you specify for the "Request Update Timeout" should be shorter than the time interval specified for the "Common Update Timeout".

```
   (NB)  Number of Buffers [nonrandom = 0/rand = 1-0FFFFH]     06H
```

This parameter specifies the number of buffers this device has for blocking and deblocking I/O requests. The default value of 6 is appropriate for a unit that has a granularity of 512 bytes. Table 10-11 shows how this value relates to other parameters. Since this is a device that supports random access, do not specify a value of zero.

The number of buffers prompt determines the number of Basic I/O System buffers the device uses for I/O. These buffers are general-purpose cache buffers that the Basic I/O System uses to increase I/O performance. The size of each buffer is 32 bytes greater than the device granularity you specified for this device (your response to the "(GRA) Granularity" parameter). The buffers are associated with a device-unit, not with a particular file. The Basic I/O System assigns these buffers to the device-unit when it attaches the device-unit.

Since the Operating System uses memory from the BIOS memory pool to create these buffers, you should increase the values you specified on the "BIOS" screen for both the minimum and maximum memory pool sizes if you increase the default value for the "Number of Buffers" parameter. An explanation of how to calculate memory usage based on your responses to the "(GRA) Granularity" and the "(NB) Number of Buffers" parameter lines is in Chapter 9.

Choosing an optimum number of buffers depends on many factors: how your application works; the speed of your components, your CPU, your memory, and your peripherals; the performance characteristics of your peripherals; and the number of users. Plus, you must trade off optimum performance against the amount of memory you use (because the more buffers you choose, the more memory you need).

Because there are so many factors, the only way that you can obtain the optimum value for your system is by testing different values and fine-tuning your configuration. Intel recommends that you use at least two buffers for this device.

```
    (CUP) Common Update [Yes/No]                          YES
```

Specify "Yes" if you want the driver to update the attached files on this device at the fixed interval you specified on the BIOS screen. Specify "No" if you do not want to update attached files on this device at fixed intervals but prefer to update attached files based only on the time interval you specified for the "Request Update Timeout" parameter. You should not specify "No" for this parameter and 0FFFFH for the "Request Update Timeout" parameter. See Chapter 9 for information about common update timeout (updating attached files at fixed intervals).

```
    (MB)  Max Buffers [0-0FFH]                            0FFH
```

This parameter specifies the maximum number of buffers that the Extended I/O System can allocate for this device's I/O. The default value for this device (0FFH) allows the S$OPEN system call to specify the actual number of Extended I/O System buffers.

The Operating System takes memory required for these buffers from the calling job's memory pool, so by setting this parameter to 0FFH you allow the calling job to select the number of buffers based on its own memory pool size. It is recommended that you use the default value.

Query Screen

After you have completed the "iSBC 208 Device-Unit Information" screen, the query screen is displayed. It contains only one line:

```
    Do you want any/more iSBC 208 DUIBs ?
```

Respond "Yes" to this prompt, if you plan to use the iSBC 208 board with device-units that have different characteristics or if your system contains two devices with the same characteristics, but different unit numbers.

While developing your initial systems, you can create as many device-unit information blocks as you want. The number of DUIBs can exceed the number of devices on your system. The particular DUIB associated with the device depends on the physical name you use when attaching it. Once you know that you will never need a particular DUIB, save memory by deleting it from your definition file before you generate your configuration files.

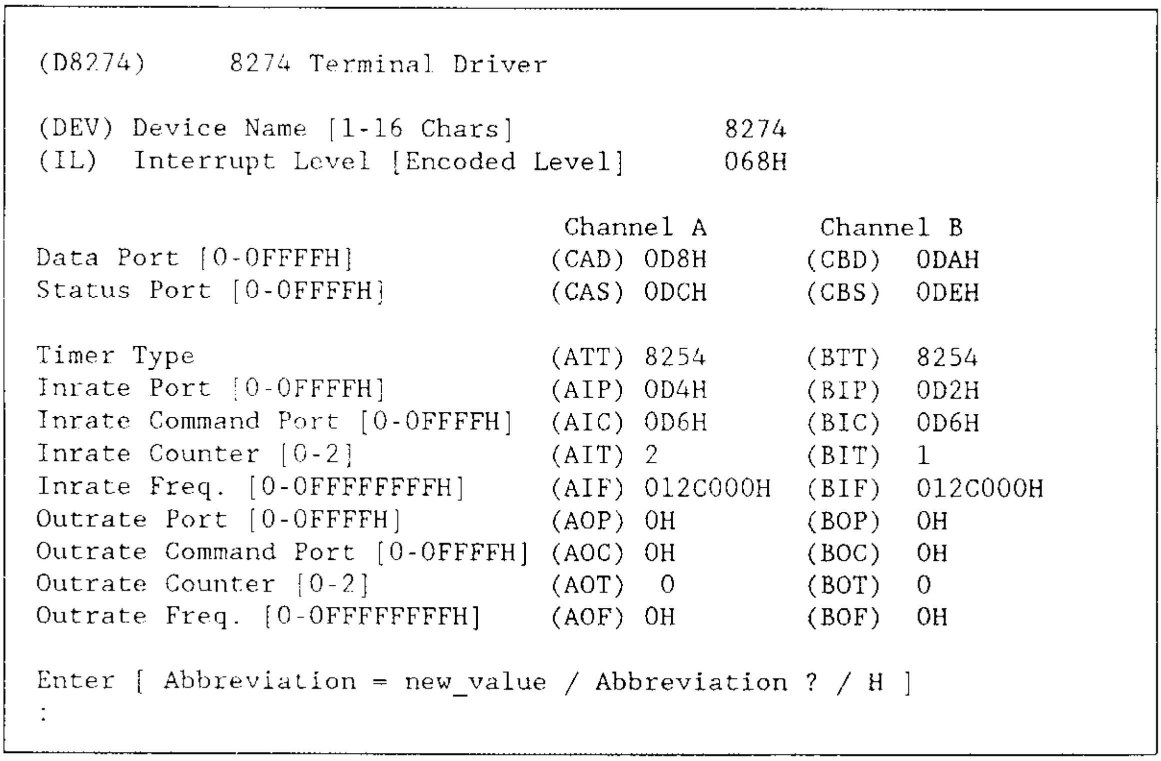
8274 Terminal Driver Parameters

The 8274 terminal driver is a terminal support driver which allows I/O to a terminal device via either of two on-board serial ports of an iSBC 286/10(A), iSBC 286/12 board. Three screens define the interface between the 8274 terminal driver and the I/O system. These screens relate to the three device configuration tables: the device information table, the unit information table, and the device unit information block (DUIB). Refer to the *Extended iRMX II Device Drivers User's Guide* for information about these tables.

The values shown on the screens in this section are the values you see if you invoke the ICU with the 28612.def file.

8274 Terminal Driver Screen

The ICU uses the information from the following screen to create a device information table for the terminal driver. If your system includes more than one controller for the 8274 terminal driver, you must specify unique port addresses and a unique interrupt level for each controller. However, each 8274 Multi-Protocol Serial Controller supports two channels on the same interrupt.



```
(D8274)      8274 Terminal Driver

(DEV) Device Name [1-16 Chars]             8274
(IL)  Interrupt Level [Encoded Level]      068H

                                  Channel A       Channel B
Data Port [0-0FFFFH]             (CAD) 0D8H      (CBD)  0DAH
Status Port [0-0FFFFH]           (CAS) 0DCH      (CBS)  0DEH

Timer Type                       (ATT) 8254      (BTT)  8254
Inrate Port [0-0FFFFH]           (AIP) 0D4H      (BIP)  0D2H
Inrate Command Port [0-0FFFFH]   (AIC) 0D6H      (BIC)  0D6H
Inrate Counter [0-2]             (AIT) 2         (BIT)  1
Inrate Freq. [0-0FFFFFFFFH]      (AIF) 012C000H  (BIF)  012C000H
Outrate Port [0-0FFFFH]          (AOP) 0H        (BOP)  0H
Outrate Command Port [0-0FFFFH]  (AOC) 0H        (BOC)  0H
Outrate Counter [0-2]            (AOT)  0        (BOT)  0
Outrate Freq. [0-0FFFFFFFFH]     (AOF) 0H        (BOF)  0H

Enter [ Abbreviation = new_value / Abbreviation ? / H ]
:
   (DEV) Device Name [1-16 Chars]                    8274
```

This parameter provides a means of associating the device and all the units and DUIBs which belong to it. The ICU stores all the screens by type, not by device. This means that all device driver screens are stored together, all unit information screens and all DUIB screens. The ICU "knows" which unit information and device-unit information screen relate to a particular device driver by the device name you enter in the "DEV" parameter.

```
   (IL)  Interrupt Level [Encoded Level]             068H
```

Use this parameter to specify the encoded interrupt level for the 8274 driver. The interrupt task uses this value to associate the interrupt task with the correct interrupt level. The default value 068H (0000 0000 0110 1000 binary) specifies master interrupt level 6.

The possible values for this field are encoded as follows (where bit 0 is the low-order bit):

Master Level	Code	Slave Level	Attached Code	Master Level	Code	Slave Level	Attached Code
0	0008H	0-7	0000-0007H	4	0048H	0-7	0040-0047H
1	0018H	0-7	0010-0017H	5	0058H	0-7	0050-0057H
2	0028H	0-7	0020-0027H	6	0068H	0-7	0060-0067H
3	0038H	0-7	0030-0037H	7	0078H	0-7	0070-0077H

```
   Data Port [0-0FFFFH]       (CAD)     0D8H  (CBD)    0DAH
```

These parameters specify the addresses of channel A (CAD) and channel B (CBD) of the 8274 data ports. Choose the addresses that match your board.

```
   Status Port [0-0FFFFH]             (CAS)     0DCH  (CBS)  0DEH
```

These parameters specify the addresses of Channel A (CAS) and Channel B (CBS) of the 8274 control/status port. Choose the addresses that match your board.

```
   Timer Type                         (ATT)     8254  (BTT)  8254
```

These parameters specify the type of Programmable Interval Timer (PIT) used by this driver for Channel A (ATT) and Channel B (BTT) (both input and output). The ICU supports only the 8254 timer.

```
Inrate Port [0-0FFFFH]          (AIP)    0D4H  (BIP)  0D2H
```

This parameter specifies the addresses for Channel A (AIP) and Channel B's (BIP) PIT counter register ports (that is, for the counter used for the input rate).

```
Inrate Command Port [0-0FFFFH] (AIC)    0D6H  (BIC)  0D6H
```

These parameters specify the addresses for Channel A's (AIC) and Channel B's (BIC) PIT command port for the timer used for the input rate.

```
Inrate Counter [0-2]            (AIT)       2  (BIT)  1
```

These parameters specify channel A's (AIT) and channel B's (BIT) PIT counter number for the counter used for the input rate.

```
Inrate Freq. [0-0FFFFFFFFH]     (AIF)  012C000H (BIF)  012C000H
```

These parameters specify the timer clock frequency input into Channel A's and Channel B's 8254 PIT. For a 1.2288 MHz counter-input frequency, this field is 1228800 decimal (012C000H).

```
Outrate Port [0-0FFFFH]         (AOP)         0H (BOP)  0H
```

These parameters specify the addresses of Channel A's and Channel B's PIT counter register for the counter used for the output rate (if it is different from the input rate counter).

```
Outrate Command Port [0-0FFFFH](AOC)         0H (BOC)  0H
```

This parameter specifies the address of the PIT mode-control port for the timer used for the output rate (if it is different from the input rate counter).

```
Outrate Counter [0-2]           (AOT)         0 (BOT)  0
```

These parameters specify channel A's (AOT) and channel B's (BOT) PIT counter number for the counter used for the output rate (if it is different from the input rate counter).

```
    Outrate Freq. [0-0FFFFFFFFH]    (AOF)         0H (BOF)  0H
```

If one timer is used for both input and output, you should set these parameters to zero. Otherwise, these parameters let you specify the output timer clock frequency for channel A's (AOF) and channel B's (BOF) 8254 PIT.

Query Screen

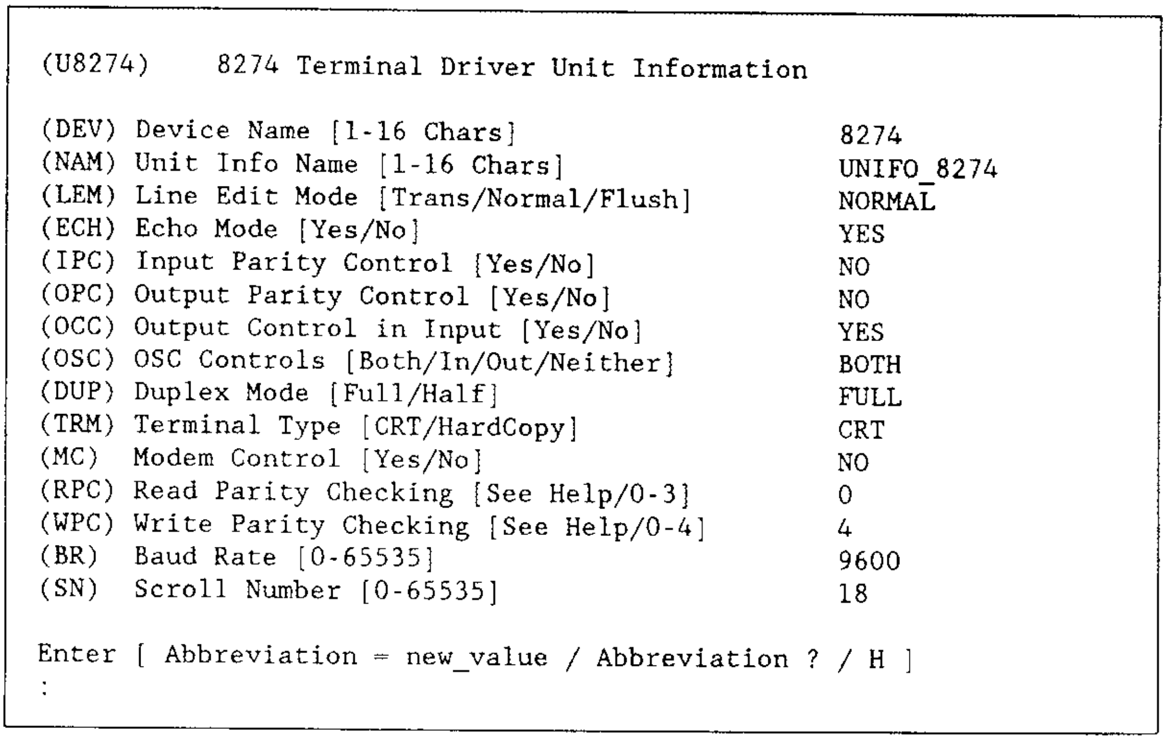
After you have completed the "8274 Driver" screen, the query screen is displayed. It contains only one line:

```
    Do you want any/more 8274 Terminal Driver DEVICEs ?
```

Respond "Yes" to this prompt if you wish to add another device driver of this type. Otherwise, respond with a "No".

Terminal Driver Unit Information Screen

The ICU uses the information from the following screen to create a unit information table for the terminal driver.



```
(U8274)    8274 Terminal Driver Unit Information

(DEV) Device Name [1-16 Chars]                        8274
(NAM) Unit Info Name [1-16 Chars]                     UNIFO_8274
(LEM) Line Edit Mode [Trans/Normal/Flush]             NORMAL
(ECH) Echo Mode [Yes/No]                              YES
(IPC) Input Parity Control [Yes/No]                   NO
(OPC) Output Parity Control [Yes/No]                  NO
(OCC) Output Control in Input [Yes/No]                YES
(OSC) OSC Controls [Both/In/Out/Neither]              BOTH
(DUP) Duplex Mode [Full/Half]                         FULL
(TRM) Terminal Type [CRT/HardCopy]                    CRT
(MC)  Modem Control [Yes/No]                          NO
(RPC) Read Parity Checking [See Help/0-3]             0
(WPC) Write Parity Checking [See Help/0-4]            4
(BR)  Baud Rate [0-65535]                             9600
(SN)  Scroll Number [0-65535]                         18

Enter [ Abbreviation = new_value / Abbreviation ? / H ]
:
```

```
(DEV)  Device Name [1-16 Chars]                8274
```

The name you enter for this parameter must be the same name you entered in the "DEV" parameter on the driver screen. This name provides the logical ICU connection between a driver and all of its units. Thus, when a device is deleted it is possible to delete all the related units at the same time.

```
(NAM) Unit Info Name [1-16 Chars]              UNIFO_8274
```

This parameter specifies a unique name for this unit information table. The first character must be an alphabetic character. Refer to the *ASM286 Language Reference Manual* for rules regarding this name.

A DUIB uses the "Unit Info Name" to point to this particular unit information table. When developing your initial systems, you can create unit information tables that are never pointed to by a DUIB. There is no harm in this process during the development stages. However, when you configure your final system, eliminate all unused unit information tables to save memory.

```
    (LEM) Line Edit Mode [Trans/Normal/Flush]       NORMAL
```

This parameter specifies the initial default line editing mode. You must choose from the following three options:

Transparent — Console input is transparent (not line-edited). The Terminal Support Code transmits input to the requesting task exactly as entered at the terminal. Before being transmitted, the Terminal Support Code accumulates data in a buffer until an operator enters the requested number of characters.

Normal — Console input is line-edited. Edited data accumulates in a buffer until an operator enters a line terminator (carriage return).

Flush — Console input is not line-edited and the Terminal Support Code transmits input to the requesting task exactly as entered at the terminal. Before being transmitted, the Terminal Support Code accumulates data in a buffer until it receives an input request. At that time, it transmits the contents of the buffer (or the number of characters requested, if the buffer contains more than that number) to the requesting task. If any characters remain in the buffer, the Terminal Support Code saves the characters for the next input request.

You can alter the value you specify for this parameter at run time if you invoke any of the SPECIAL system calls or send an OSC sequence. OSC sequences are described in the *Extended iRMX II Device Drivers User's Guide*.

```
    (ECH) Echo Mode [Yes/No]                          YES
```

Specify "Yes" if you want characters entered into the terminal to be "echoed" to the terminal's display screen. Otherwise, specify "No".

You can alter the value you specify for this parameter at run time if you invoke any of the SPECIAL system calls or send an OSC sequence. OSC sequences are described in the *Extended iRMX II Device Drivers User's Guide*.

```
    (IPC) Input Parity Control [Yes/No]             NO
```

Specify "Yes" if you want the system to change the parity bit (bit 7) of characters entered into the terminal to zero. Specify "No" if you do not want the terminal driver to change bit 7 of the input characters.

You can alter the value you specify for this parameter at run time if you invoke any of the SPECIAL system calls or send an OSC sequence. OSC sequences are described in the *Extended iRMX II Device Drivers User's Guide*.

```
    (OPC) Output Parity Control [Yes/No]            NO
```

Specify "Yes" if you want the system to change the parity bit (bit 7) of characters being output to the terminal to zero. Specify "No" if you want bit 7 in the output characters to remain unchanged.

You can alter the value you specify for this parameter at run time if you invoke any of the SPECIAL system calls or send an OSC sequence. OSC sequences are described in the *Extended iRMX II Device Drivers User's Guide*.

```
    (OCC) Output Control in Input [Yes/No]          YES
```

Specify "Yes" if you want the Terminal Support Code to accept output control characters in the input stream. Specify "No" if you want the Terminal Support Code to ignore output control characters. Control characters are described in the *Extended iRMX II Device Drivers User's Guide*.

You can alter the value you specify for this parameter at run time if you invoke any of the SPECIAL system calls or send an OSC sequence. OSC sequences are described in the *Extended iRMX II Device Drivers User's Guide*.

```
    (OSC) OSC Controls [Both/In/Out/Neither]        BOTH
```

This parameter specifies whether the device driver should act upon Operating System Command (OSC) controls when they appear in either an input or an output stream. Choose one of the following options as the initial default value for the device driver:

Both — Act upon OSC control sequences in either input or output stream (from either terminal or program).

Input — Act upon OSC control sequences in input stream only (from terminal and not from program).

Output — Act upon OSC control sequences in output stream only (from program and not from terminal).

Neither   Do not act upon OSC control sequences.

The OSC control sequence, used in communicating from a program or a terminal to the Operating System, is described in the *Extended iRMX II Device Drivers User's Guide*. You can alter the value you specify for this parameter at run time if you invoke any of the SPECIAL system calls or send an OSC sequence.

```
    (DUP) Duplex Mode [Full/Half]                    FULL
```

This parameter specifies the line protocol mode for a terminal. Choose either full-duplex or half-duplex.

When line protocol mode is full-duplex, the terminal driver concurrently handles input to and output from the terminal. If you specified both echo mode and full-duplex, the terminal driver echoes each character. When the line protocol mode is half-duplex, there can be input to and output from the terminal, but not concurrently. If you specify both echo mode and half-duplex, the terminal, not the terminal driver, must echo each character.

You can alter the value you specify for this parameter at run time if you invoke any of the SPECIAL system calls or send an OSC sequence. OSC sequences are described in the *Extended iRMX II Device Drivers User's Guide*.

```
    (TRM) Terminal Type [CRT/HardCopy]              CRT
```

This parameter specifies how your terminal supports the rubout function. Respond "CRT" if your terminal can backspace and leave a blank character on the screen for each character "rubbed out." Respond "Hard Copy" if your terminal cannot backspace and leave a blank character on previously-displayed characters.

You can alter the value you specify for this parameter at run time if you invoke any of the SPECIAL system calls or send an OSC sequence. OSC sequences are described in the *Extended iRMX II Device Drivers User's Guide*.

```
    (MC)  Modem Control [Yes/No]                     NO
```

The 8274 Driver does not support Modem Control, do not change the default setting.

```
    (RPC) Read Parity Checking [See Help/0-3]    0
```

This parameter specifies how the hardware in your system will handle read parity checking. A response of 0 or 1 specifies 8-bit data with no parity checking. A response of 2 or 3 specifies 7-bit data with parity checking. In this section, "parity bit" refers to bit seven. Specify one of the following four values:

0 Ignore parity checking and set the input parity bit (bit 7) to zero.

1 Ignore parity checking and do not change the parity bit.

2 The driver expects even parity on input and sets the parity bit to 0 on meeting this condition.

The driver sets the parity bit is to 1 if odd parity is received, a framing error occurs (receive stop bit is zero), or an overrun error is encountered (a new character has been received before the interrupt routine for character processing has completed).

3 Driver expects odd parity. If a parity error, a framing error, or an overrun error is encountered, the parity bit is set to 1. Otherwise, the parity bit is set to zero.

Note that because this parameter specifies how the hardware deals with parity, a response of zero here negates a "No" response to the software parity checking parameter "(IPC) Input Parity Checking".

If you set "(RPC) Read Parity Checking" to two or three and you set "(IPC) Input Parity Checking" to "Yes", your application will not be able to detect a transmission error.

You can alter the value you specify for this parameter at run time if you invoke any of the SPECIAL system calls or send an OSC sequence. OSC sequences are described in the *Extended iRMX II Device Drivers User's Guide.*

```
    (WPC) Write Parity Checking [See Help/0-4]   4
```

This parameter specifies how the hardware in your system will handle write parity checking. A response of 0, 1, or 4 specifies 8-bit data with no parity checking. A response of 2 or 3 specifies 7-bit data with parity checking. In this section, "parity bit" refers to bit seven. Specify one of the following five values:

0 Set the output parity bit to zero.

1 Set the output parity bit to one.

2 Set the output parity bit to one if the total number of 1's in the character is odd. Set the parity bit to zero if the total number of 1's is even (even parity). This option should be used if the driver is using even parity checking for input.

3 Set the output parity bit to 0 if the total number of 1's in the character is odd (odd parity). Set the parity bit to 1 if the total number of 1's is even. This option should be used if the driver is using odd parity checking for input.

4 Do not change the output parity bit.

Note that because this parameter specifies how the hardware deals with parity, a response of zero here negates a "No" response to the software parity checking parameter "(OPC) Output Parity Checking".

You can alter the value you specify for this parameter at run time if you invoke any of the SPECIAL system calls or send an OSC sequence. OSC sequences are described in the *Extended iRMX II Device Drivers User's Guide.*

```
    (BR)  Baud Rate [0-65535]                     9600
```

This parameter specifies the initial baud rate of this terminal. Specify a value of one if you want the controller to ascertain the initial baud rate automatically. The default value is 9600 decimal.

You can alter the value you specify for this parameter at run time if you invoke any of the SPECIAL system calls or send an OSC sequence. OSC sequences are described in the *Extended iRMX II Device Drivers User's Guide.*

```
    (SN)  Scroll Number [0-65535]                 18
```

This parameter specifies the number of lines to scroll when an operator enters the scrolling output control character (CONTROL-W is the default). Typical values should range from 10 to 24. The default value is 18 decimal.

You can alter the value you specify for this parameter at run time if you invoke any of the SPECIAL system calls or send an OSC sequence. OSC sequences are described in the *Extended iRMX II Device Drivers User's Guide.*

Query Screen

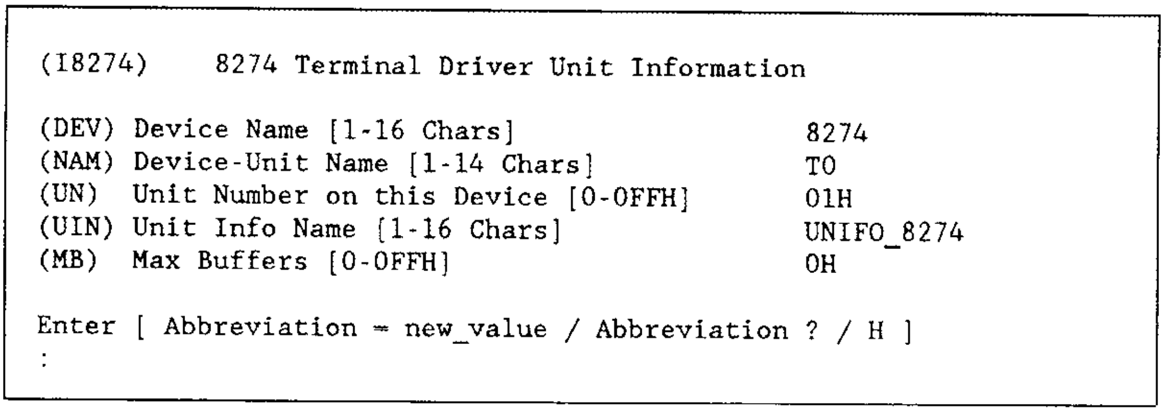
After you have completed the "8274 Terminal Driver Unit Information" screen, the query screen is displayed. It contains only one line:

```
    Do you want any/more 8274 Terminal Driver UNITs  ?
```

Respond to this prompt with a "Yes" if you need another unit information table for this device. Otherwise, respond to this prompt with a "No".

Terminal Driver Device-Unit Information Screen

The ICU uses the information from the following screen to create a device unit information block (DUIB) for the 8274 terminal driver.



```
(I8274)    8274 Terminal Driver Unit Information

(DEV) Device Name [1-16 Chars]                  8274
(NAM) Device-Unit Name [1-14 Chars]             TO
(UN)  Unit Number on this Device [0-0FFH]       01H
(UIN) Unit Info Name [1-16 Chars]               UNIFO_8274
(MB)  Max Buffers [0-0FFH]                      0H

Enter [ Abbreviation = new_value / Abbreviation ? / H ]
:
```

```
    (DEV)  Device Name [1-16 Chars]                8274
```

The name you enter for this parameter must be the same name you entered in the "DEV" parameter on the driver screen. This name provides the logical ICU connection between the driver and all of its DUIBs. Thus, when a device is deleted it is possible to delete all the related units and DUIBs at the same time.

```
    (NAM) Device-Unit Name [1-14 Chars]          TO
```

This parameter specifies a name that uniquely identifies the device-unit for the I/O System. The name you specify for this parameter is the physical name you specify when invoking the Human Interface ATTACHDEVICE command, the BIOS A$PHYSICAL$ATTACH$DEVICE system call, or the EIOS LOGICAL$ATTACH$DEVICE system call.

The ICU allows you to enter from one to fourteen characters. Refer to the *ASM286 Language Reference* for rules regarding this name.

```
    (UN)  Unit Number on this Device [0-0FFH]    01H
```

This parameter specifies the unit number of this device-unit. The unit numbers for the device should be either zero or one: unit 0 corresponds to channel B and unit 1 corresponds to channel A.

Each of the two possible serial channels is a single unit.

```
    (UIN) Unit Info Name [1-16 Chars]              UNIFO_8274
```

This parameter specifies the "Unit Info Name" of a unit information table that fills the needs of this DUIB. The following rules apply to the name you choose:

- You must create a unit information table with this "unit info name".
- Each DUIB can point to only one unit information table.
- A particular unit information table can be pointed to by more than one DUIB.

```
    (MB)  Max Buffers [0-0FFH]                     0H
```

This parameter specifies the maximum number of buffers that the Extended I/O System can allocate for this device's I/O. Unless you are planning on using this terminal for output only, do not change the default value. For terminals used for output only, you may want to specify a small non-zero value. Note, however, that the EIOS will delay output until the buffer is full or you close the connection to the device.

Query Screen

After you have completed the "8274 Terminal Driver Device-Unit Information" screen, the query screen is displayed. It contains only one line:

```
    Do you want any/more 8274 Terminal Driver DUIBs ?
```

Respond "Yes" to this prompt if you plan to use the 8274 Terminal Driver with two terminals that have different characteristics or if you have two devices with the same characteristics, but different unit numbers.

While developing your initial systems, you can create as many device-unit information blocks as you want. The number of DUIBs can exceed the number of devices on your system. The particular DUIB associated with the device depends on the physical name you use when attaching it. Once you know that you will never need a particular DUIB, save memory by deleting it from your definition file before you generate your configuration files.

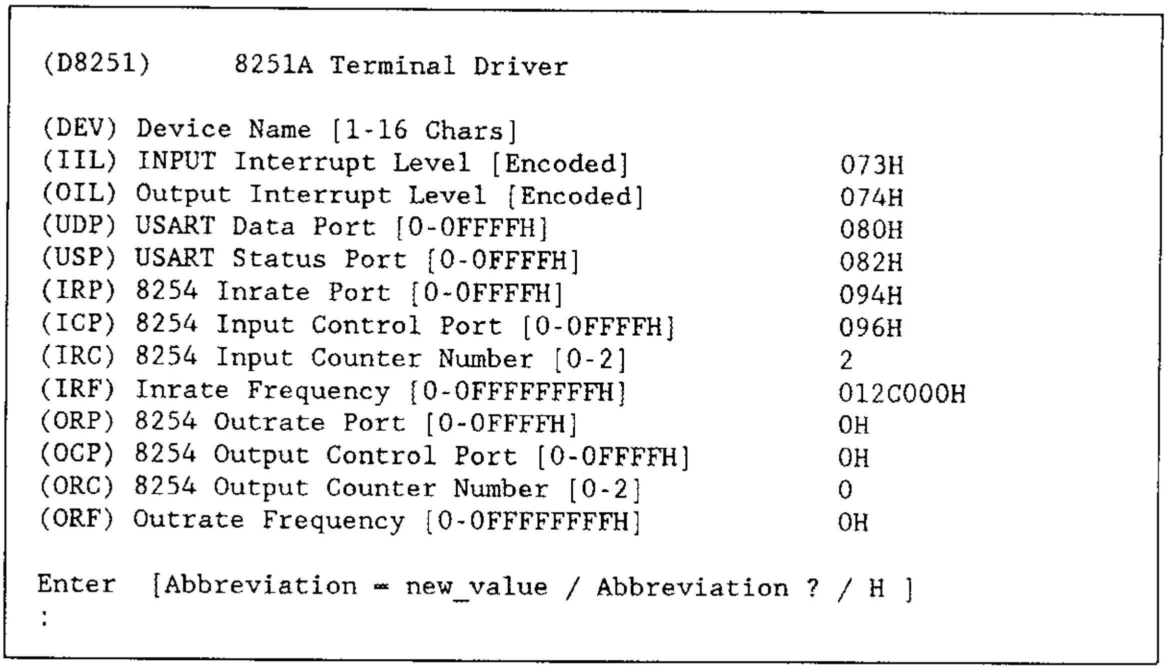
8251A Driver Parameters

The 8251A Terminal Driver is a terminal driver that allows I/O to a terminal device via the on-board serial port of an iSBC 386/20 or any other 80286/80386-based board which includes an iSBX 351 MULTIMODULE. Three screens define the interface between the terminal driver and the I/O system. These screens relate to the three device configuration tables: the device information table, the unit information table, and the device unit information block (DUIB). Refer to the *Extended iRMX II Device Drivers User's Guide* for further information about these tables.

The values shown on the screens in this section are the values you see if you invoke the ICU using the 38620.DEF file.

8251A Terminal Driver Screen

The ICU uses the information from the following screen to create a device information table for the terminal driver. If your system includes more than one 8251A controller, you must specify a unique set of interrupt levels for each controller.



```
(D8251)     8251A Terminal Driver

(DEV) Device Name [1-16 Chars]
(IIL) INPUT Interrupt Level [Encoded]               073H
(OIL) Output Interrupt Level [Encoded]              074H
(UDP) USART Data Port [0-0FFFFH]                    080H
(USP) USART Status Port [0-0FFFFH]                  082H
(IRP) 8254 Inrate Port [0-0FFFFH]                   094H
(ICP) 8254 Input Control Port [0-0FFFFH]            096H
(IRC) 8254 Input Counter Number [0-2]               2
(IRF) Inrate Frequency [0-0FFFFFFFFH]               012C000H
(ORP) 8254 Outrate Port [0-0FFFFH]                  0H
(OCP) 8254 Output Control Port [0-0FFFFH]           0H
(ORC) 8254 Output Counter Number [0-2]              0
(ORF) Outrate Frequency [0-0FFFFFFFFH]              0H

Enter  [Abbreviation = new_value / Abbreviation ? / H ]
:
    (DEV) Device Name [1-16 Chars]
```

This parameter provides a means of associating the device and all the units and DUIBs which belong to it. The ICU stores all the screens by type, not by device. This means that all device driver screens are stored together, all unit information screens and all DUIB screens. The ICU "knows" which unit information and device-unit information screen relate to a particular device driver by the device name you enter in the "DEV" parameter.

```
    (IIL) Input Interrupt Level [Encoded]          073H
```

This parameter specifies the encoded interrupt level for the receiver ready interrupt. The interrupt task uses this value to associate the interrupt task with the correct interrupt level. The default value 073H (0000 0000 0111 0011 binary) specifies slave interrupt level 3.

The possible values for this field are encoded as follows (where bit 0 is the low-order bit):

Master Level	Code	Slave Level	Attached Code	Master Level	Code	Slave Level	Attached Code
0	0008H	0-7	0000-0007H	4	0048H	0-7	0040-0047H
1	0018H	0-7	0010-0017H	5	0058H	0-7	0050-0057H
2	0028H	0-7	0020-0027H	6	0068H	0-7	0060-0067H
3	0038H	0-7	0030-0037H	7	0078H	0-7	0070-0077H

```
    (OIL) Output Interrupt Level [Encoded]         0747H
```

This parameter specifies the encoded interrupt level for the transmitter ready interrupt. The interrupt task uses this value to associate itself with the correct interrupt level. The possible values for this field are as follows (where bit 0 is the low-order bit):

Master Level	Code	Slave Level	Attached Code	Master Level	Code	Slave Level	Attached Code
0	0008H	0-7	0000-0007H	4	0048H	0-7	0040-0047H
1	0018H	0-7	0010-0017H	5	0058H	0-7	0050-0057H
2	0028H	0-7	0020-0027H	6	0068H	0-7	0060-0067H
3	0038H	0-7	0030-0037H	7	0078H	0-7	0070-0077H

```
    (UDP) USART Data Port [0-0FFFFH]               080H
```

This parameter specifies the address of the 8251A data port. Choose the address that matches your board.

```
(USP) USART Status Port [0-OFFFFH]            082H
```

This parameter specifies the address of the 8251A control/status port. Choose the address that matches your board.

```
(IRP) 8254 Inrate Port [0-OFFFFH]             094H
```

This parameter specifies the port address for the 8254A PIT counter register used for the input rate. Choose the address that matches your board.

```
(ICP) 8254 Input Control Port [0-OFFFFH]      096H
```

This parameter specifies the port address for the 8254 PIT mode control used for the input rate. Choose the address that matches your board.

```
(IRC) 8254 Input Counter Number [0-2]         2
```

This parameter specifies the number of the 8254 PIT counter used for the input rate.

```
(IRF) Inrate Frequency [0-OFFFFFFFFH]         012C000H
```

This parameter specifies the counter input frequency. The default value 012C000H (1228800 decimal) indicates a 1.2288 MHz counter-input frequency.

```
(ORP) 8254 Outrate Port [0-OFFFFH]            OH
```

This parameter specifies the address of the 8254 PIT counter register for the counter used for the output rate. If one timer is used for both input and output, set this parameter to zero.

```
(OCP) 8254 Output Control Port [0-OFFFFH]     OH
```

This parameter specifies the address of the 8254 PIT control port for the timer used for the output rate. If one timer is used for both input and output, set this parameter to zero.

```
(ORC) 8254 Output Counter Number [0-2]        0
```

This parameter specifies the 8254 PIT counter number for the counter used for the output rate. If one timer is used for both input and output, set this parameter to zero.

```
    (ORF) Outrate Frequency [0-OFFFFFFFFH]          OH
```

This parameter specifies the counter output frequency. If one timer is used for both input and output, you should set this parameter to zero.

Query Screen

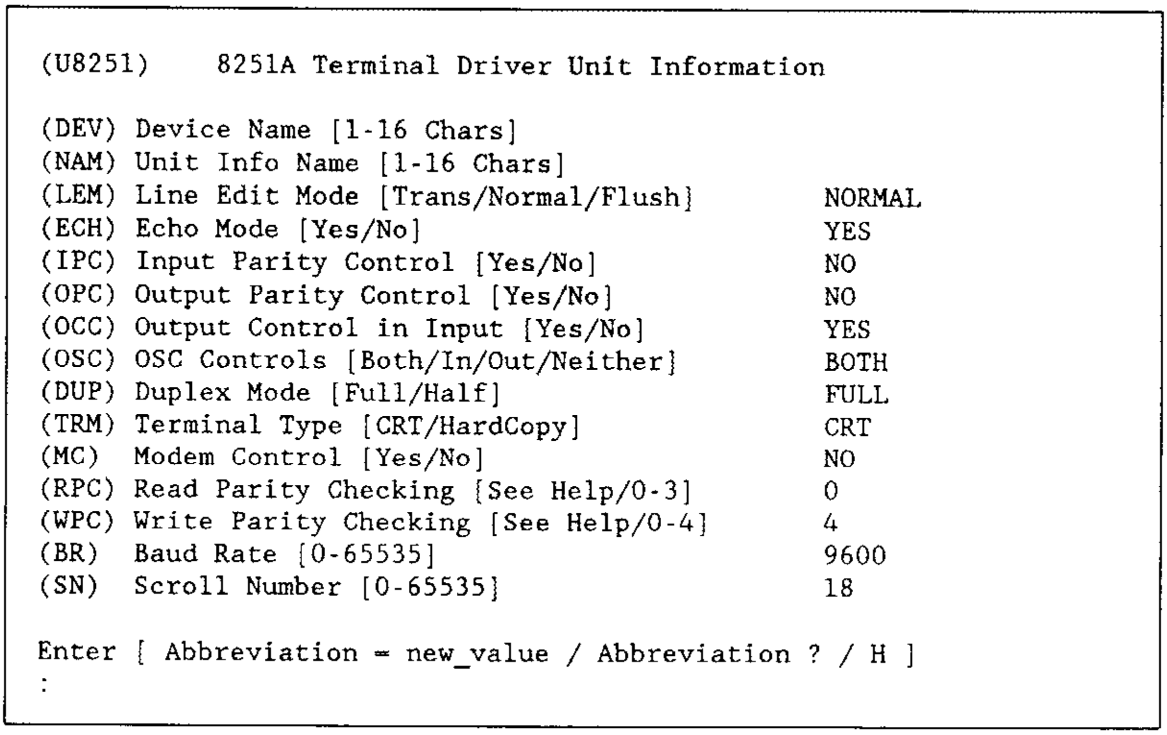
After you have completed the "8251A Driver" screen, the query screen is displayed. It contains only one line:

```
    Do you want any/more 8251A Terminal Driver DEVICEs ?
```

Respond "Yes" if you wish to add additional device drivers of the same type. Otherwise, respond with a "No".

8251A Terminal Driver Unit Information Screen

The ICU uses the information from the following screen to create a unit information table for the 8251A driver.



```
(U8251)    8251A Terminal Driver Unit Information

(DEV) Device Name [1-16 Chars]
(NAM) Unit Info Name [1-16 Chars]
(LEM) Line Edit Mode [Trans/Normal/Flush]           NORMAL
(ECH) Echo Mode [Yes/No]                            YES
(IPC) Input Parity Control [Yes/No]                 NO
(OPC) Output Parity Control [Yes/No]                NO
(OCC) Output Control in Input [Yes/No]              YES
(OSC) OSC Controls [Both/In/Out/Neither]            BOTH
(DUP) Duplex Mode [Full/Half]                       FULL
(TRM) Terminal Type [CRT/HardCopy]                  CRT
(MC)  Modem Control [Yes/No]                        NO
(RPC) Read Parity Checking [See Help/0-3]           0
(WPC) Write Parity Checking [See Help/0-4]          4
(BR)  Baud Rate [0-65535]                           9600
(SN)  Scroll Number [0-65535]                       18

Enter [ Abbreviation = new_value / Abbreviation ? / H ]
:
```

(DEV) Device Name [1-16 Chars]

The name you enter for this parameter must be the same name you entered in the "DEV" parameter on the "8251A Driver" screen. This name provides the logical ICU connection between the driver and all of its units. Thus, when a device is deleted it is possible to delete all the related units at the same time.

(NAM) Unit Info Name [1-16 Chars]

This parameter specifies a unique name for this unit information table. The first character must be an alphabetic character. Refer to the *ASM286 Language Reference Manual* for rules regarding this name.

A DUIB uses the "Unit Info Name" to point to this particular unit information table. When developing your initial systems, you can create unit information tables that are never pointed to by a DUIB. There is no harm in this process during the development stages. However, when you configure your final system, eliminate all unused unit information tables to save memory.

```
    (LEM) Line Edit Mode [Trans/Normal/Flush]       NORMAL
```

This parameter specifies the initial default line editing mode. You must choose from the following three options:

Transparent — Console input is transparent (not line-edited). The Terminal Support Code transmits input to the requesting task exactly as entered at the terminal. Before being transmitted, the Terminal Support Code accumulates data in a buffer until an operator enters the requested number of characters.

Normal — Console input is line-edited. Edited data accumulates in a buffer until an operator enters a carriage return.

Flush — Console input is not line-edited and the Terminal Support Code transmits input to the requesting task exactly as entered at the terminal. Before being transmitted, the Terminal Support Code accumulates data in a buffer until it receives an input request. At that time, it transmits the contents of the buffer (or the number of characters requested, if the buffer contains more than that number) to the requesting task. If any characters remain in the buffer, the Terminal Support Code saves the characters for the next input request.

You can alter the value you specify for this parameter at run time if you invoke any of the SPECIAL system calls or send an OSC sequence. OSC sequences are described in the *Extended iRMX II Device Drivers User's Guide.*

```
    (ECH) Echo Mode [Yes/No]                        YES
```

Specify "Yes" if you want characters entered into the terminal to be "echoed" to the terminal's display screen. Otherwise, specify "No".

You can alter the value you specify for this parameter at run time if you invoke any of the SPECIAL system calls or send an OSC sequence. OSC sequences are described in the *Extended iRMX II Device Drivers User's Guide.*

```
    (IPC) Input Parity Control [Yes/No]              NO
```

Specify "Yes" if you want the system to change the parity bit (bit 7) of characters entered into the terminal to zero. Specify "No" if you do not want the terminal driver to change bit 7 of the input character.

You can alter the value you specify for this parameter at run time if you invoke any of the SPECIAL system calls or send an OSC sequence. OSC sequences are described in the *Extended iRMX II Device Drivers User's Guide*.

```
    (OPC) Output Parity Control [Yes/No]             NO
```

Specify "Yes" if you want the system to change the parity bit (bit 7) of characters being output to the terminal to zero. Specify "No" if you want bit 7 in the output characters to remain unchanged.

You can alter the value you specify for this parameter at run time if you invoke any of the SPECIAL system calls or send an OSC sequence. OSC sequences are described in the *Extended iRMX II Device Drivers User's Guide*.

```
    (OCC) Output Control in Input [Yes/No]           YES
```

Specify "Yes" if you want the Terminal Support Code to accept output control characters in the input stream. Specify "No" if you want the Terminal Support Code to ignore output control characters. Control characters are described in the *Extended iRMX II Device Drivers User's Guide*.

You can alter the value you specify for this parameter at run time if you invoke any of the SPECIAL system calls or send an OSC sequence. OSC sequences are described in the *Extended iRMX II Device Drivers User's Guide*.

```
    (OSC) OSC Controls [Both/In/Out/Neither]         BOTH
```

This parameter specifies whether the device driver should act upon Operating System Command (OSC) controls when they appear in either an input or an output stream. Choose one of the following options as the initial default value for the device driver:

Both — Act upon OSC control sequences in either input or output stream (from either terminal or program).

Input — Act upon OSC control sequences in input stream only (from terminal and not from program).

Output — Act upon OSC control sequences in output stream only (from program and not from terminal).

Neither Do not act upon OSC control sequences.

The OSC control sequence, used in communicating from a program or a terminal to the Operating System, is described in the *Extended iRMX II Device Drivers User's Guide*. You can alter the value you specify for this parameter at run time if you invoke any of the SPECIAL system calls or send an OSC sequence.

```
    (DUP) Duplex Mode [Full/Half]                    FULL
```

This parameter specifies the line protocol mode for a terminal. Choose either full-duplex or half-duplex.

When line protocol mode is full-duplex, the terminal driver concurrently handles input to and output from the terminal. If you specified both echo mode and full-duplex, the terminal driver echoes each character. When the line protocol mode is half-duplex, there can be input to and output from the terminal, but not concurrently. If you specify both echo mode and half-duplex, the terminal, not the terminal driver, must echo each character.

You can alter the value you specify for this parameter at run time if you invoke any of the SPECIAL system calls or send an OSC sequence. OSC sequences are described in the *Extended iRMX II Device Drivers User's Guide.*

```
    (TRM) Terminal Type [CRT/HardCopy]               CRT
```

This parameter specifies how your terminal supports the rubout function. Respond "CRT" if your terminal can backspace and leave a blank character on the screen for each character "rubbed out." Respond "Hard Copy" if you terminal cannot backspace and leave a blank character on previously displayed characters.

You can alter the value you specify for this parameter at run time if you invoke any of the SPECIAL system calls or send an OSC sequence. OSC sequences are described in the *Extended iRMX II Device Drivers User's Guide.*

```
    (MC)  Modem Control [Yes/No]                     NO
```

This driver does not support modem control, leave the parameter set to "No".

```
    (RPC) Read Parity Checking [See Help/0-3]     0
```

This parameter specifies how the hardware in your system will handle read parity checking. A response of 0 or 1 specifies 8-bit data with no parity checking. A response of 2 or 3 specifies 7-bit data with parity checking. In this section, "parity bit" refers to bit seven. Specify one of the following four values:

0 Ignore parity checking and set the input parity bit (bit 7) to zero.

1 Ignore parity checking and do not change the parity bit.

2 The driver expects even parity on input and sets the parity bit to 0 on meeting this condition.

The driver sets the parity bit is to 1 if odd parity is received, a framing error occurs (receive stop bit is zero), or an overrun error is encountered (a new character has been received before the interrupt routine for character processing has completed).

3 Driver expects odd parity. If a parity error, a framing error, or an overrun error is encountered, the parity bit is set to 1. Otherwise, the parity bit is set to zero.

Note that because this parameter specifies how the hardware deals with parity, a response of zero here negates a "No" response to the software parity checking parameter "(IPC) Input Parity Checking".

If you set "(RPC) Read Parity Checking" to two or three and you set "(IPC) Input Parity Checking" to "Yes", your application will not be able to detect a transmission error.

You can alter the value you specify for this parameter at run time if you invoke any of the SPECIAL system calls or send an OSC sequence. OSC sequences are described in the *Extended iRMX II Device Drivers User's Guide.*

Note that a response of zero or one is only meaningful if your response to the "(IPC) Input Parity Control" parameter is "No". Likewise, a response of two or three to this parameter implies that you responded "Yes" to the "(IPC)" parameter.You can alter the value you specify for this parameter at run time if you invoke any of the SPECIAL system calls or send an OSC sequence. OSC sequences are described in the *Extended iRMX II Device Drivers User's Guide.*

```
    (WPC) Write Parity Checking [See Help/0-4]    4
```

This parameter specifies how the hardware in your system will handle write parity checking. A response of 0, 1, or 4 specifies 8-bit data with no parity checking. A response of 2 or 3 specifies 7-bit data with parity checking. In this section, "parity bit" refers to bit seven. Specify one of the following five values:

0 Set the output parity bit to zero.

1 Set the output parity bit to one.

2 Set the output parity bit to one if the total number of 1's in the character is odd. Set the parity bit to zero if the total number of 1's is even (even parity). This option should be used if the driver is using even parity checking for input.

3 Set the output parity bit to 0 if the total number of 1's in the character is odd (odd parity). Set the parity bit to 1 if the total number of 1's is even. This option should be used if the driver is using odd parity checking for input.

4 Do not change the output parity bit.

Note that because this parameter specifies how the hardware deals with parity, a response of zero here negates a "No" response to the software parity checking parameter "(OPC) Output Parity Checking".

You can alter the value you specify for this parameter at run time if you invoke any of the SPECIAL system calls or send an OSC sequence. OSC sequences are described in the *Extended iRMX II Device Drivers User's Guide*.

```
    (BR)  Baud Rate [0-65535]                         9600
```

This parameter specifies the initial baud rate of this terminal. Specify a value of one if you want the controller to ascertain the initial baud rate automatically. The default value is 9600 decimal.

You can alter the value you specify for this parameter at run time if you invoke any of the SPECIAL system calls or send an OSC sequence. OSC sequences are described in the *Extended iRMX II Device Drivers User's Guide*.

```
    (SN)  Scroll Number [0-65535]                   18
```

This parameter specifies the number of lines to scroll when the operator enters the scrolling output control character (Control-W is the default). Typical values should range from 10 to 24. The default value is 18 decimal.

You can alter the value you specify for this parameter at run time if you invoke any of the SPECIAL system calls or send an OSC sequence. OSC sequences are described in the *Extended iRMX II Device Drivers User's Guide.*

Query Screen

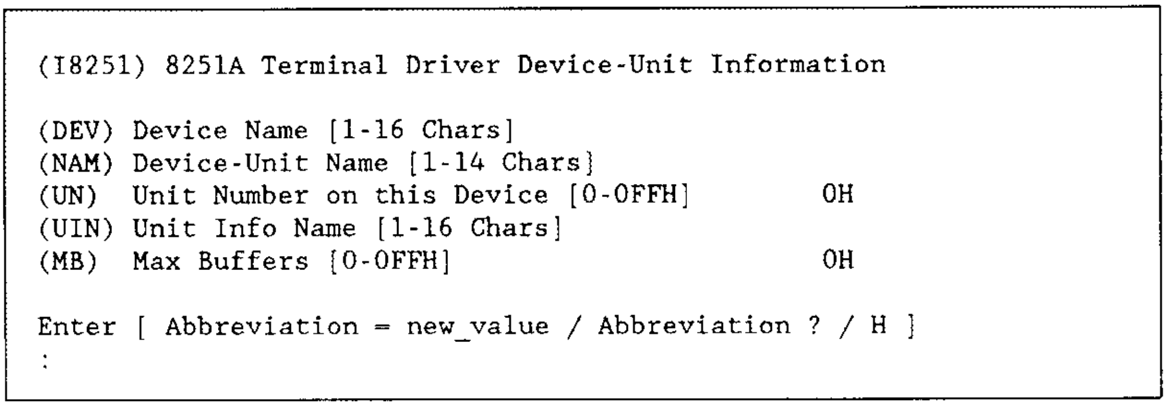
After you have completed the "8251A Terminal Driver Unit Information" screen, the query screen is displayed. It contains only one line:

```
Do you have any/more 8251A Terminal Driver UNITs  ?
```

Respond to this prompt with a "Yes" if you need another unit information table for this device. Otherwise, respond to this prompt with a "No".

8251A Terminal Driver Device-Unit Information Screen

The ICU uses the information from the following screen to create a device unit information block (DUIB) for the 8251A terminal driver.



```
(I8251) 8251A Terminal Driver Device-Unit Information

(DEV) Device Name [1-16 Chars]
(NAM) Device-Unit Name [1-14 Chars]
(UN)  Unit Number on this Device [0-0FFH]        0H
(UIN) Unit Info Name [1-16 Chars]
(MB)  Max Buffers [0-0FFH]                       0H

Enter [ Abbreviation = new_value / Abbreviation ? / H ]
:
```

```
    (DEV)  Device Name [1-16 Chars]
```

The name you enter for this parameter must be the same name you entered in the "DEV" parameter on the "8251A Driver" screen. This name provides the logical ICU connection between the driver and all of its DUIBs. Thus, when a device is deleted it is possible to delete all the related units and DUIBs at the same time.

```
    (NAM) Device-Unit Name [1-14 chars]
```

This parameter specifies a name that uniquely identifies the device-unit for the I/O System. The name you specify for this parameter is the physical name you specify when invoking the Human Interface ATTACHDEVICE command, the BIOS A$PHYSICAL$ATTACH$DEVICE system call, or the EIOS LOGICAL$ATTACH$DEVICE system call.

The ICU allows you to enter from one to fourteen characters. Refer to the *ASM286 Language Reference Manual* for rules regarding this name.

```
    (UN)  Unit Number on this Device [0-0FFH]     0H
```

This parameter specifies the unit number of the device-unit associated with this device-unit information block. Since this driver supports only one device-unit per drive, the default value of 0H is the only value you need (unless you need to define the same device-unit with different characteristics).

```
    (UIN) Unit Info Name [1-16 Chars]
```

This parameter specifies the "Unit Info Name" of a unit information table that fills the needs of this DUIB. The following rules apply to the name you choose:

- You must create a unit information table with this "unit info name".
- Each DUIBs can point to only one unit information table.
- A particular unit information table can be pointed to by more than one DUIB.

```
    (MB)  Max Buffers [0-0FFH]                      0H
```

This parameter specifies the maximum number of buffers that the Extended I/O System can allocate for this device's I/O. Unless you are planning on using this terminal for output only, do not change the default value. For terminals used for output only, you may want to specify a small non-zero value. Note, however, that the EIOS will delay output until the buffer is full or you close the connection to the device.

Query Screen

After you have completed the "8251A Terminal Driver Device-Unit Information" screen, the query screen is displayed. It contains only one line:

```
    Do you want any/more 8251A Terminal Driver DUIBs ?
```

Respond "Yes" to this prompt if you plan to use the 8251A Terminal Driver with a terminal that has variable characteristics.

While developing your initial systems, you can create as many device-unit information blocks as you want. The number of DUIBs can exceed the number of devices on your system. The particular DUIB associated with the device depends on the physical name you use when attaching it. Once you know that you will never need a particular DUIB, save memory by deleting it from your definition file before you generate your configuration files.

82530 Terminal Driver Parameters

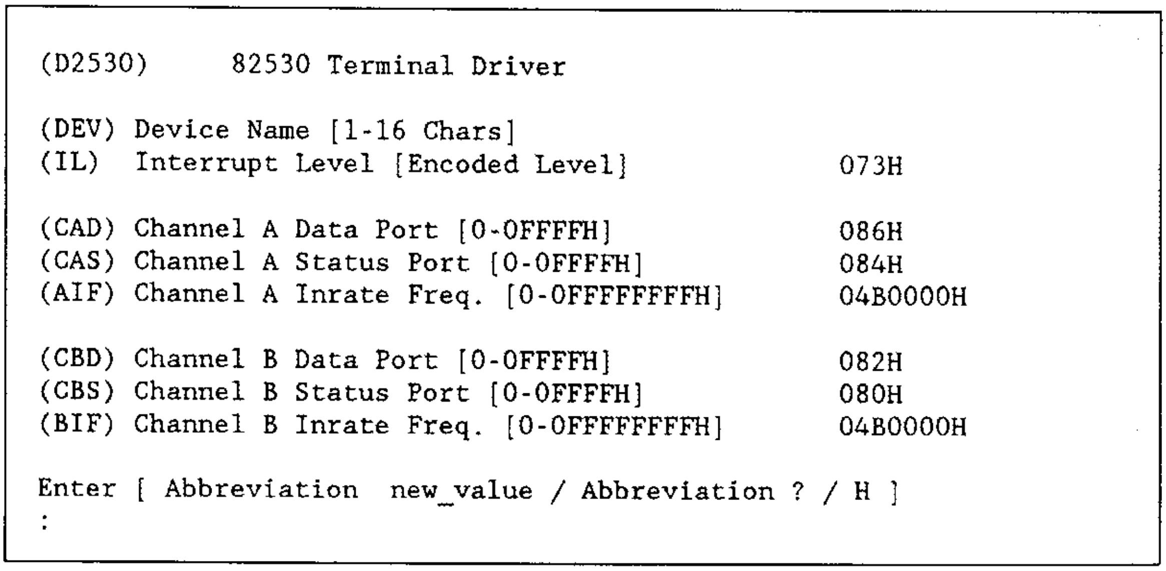
The 82530 terminal driver is a terminal driver that allows I/O to a terminal device via the on-board serial port of an iSBX 354 MULTIMODULE board. The 82530 Terminal Driver provides two channels of terminal I/O in conjunction with an 82530 SCC. Both channels use a single interrupt level.

Three screens define the interface between the terminal driver and the I/O system. These screens relate to the three device configuration tables: the device information table, the unit information table, and the device unit information block (DUIB). Refer to the *Extended iRMX II Device Drivers User's Guide* for further information about these tables.

The 82530 terminal driver is not part of the Start-up system supplied by Intel. The default values displayed in the following screens are the ICU defaults.

82530 Terminal Driver Screen

The ICU uses the information from the following screen to create a device information table for the terminal driver. If your system includes an iSBX 354 MULTIMODULE board, you must specify a unique interrupt level and port address for each MULTIMODULE board.



```
(D2530)     82530 Terminal Driver

(DEV) Device Name [1-16 Chars]
(IL)  Interrupt Level [Encoded Level]             073H

(CAD) Channel A Data Port [0-0FFFFH]              086H
(CAS) Channel A Status Port [0-0FFFFH]            084H
(AIF) Channel A Inrate Freq. [0-0FFFFFFFFH]       04B0000H

(CBD) Channel B Data Port [0-0FFFFH]              082H
(CBS) Channel B Status Port [0-0FFFFH]            080H
(BIF) Channel B Inrate Freq. [0-0FFFFFFFFH]       04B0000H

Enter [ Abbreviation  new_value / Abbreviation ? / H ]
:
    (DEV) Device Name [1-16 Chars]
```

This parameter provides a means of associating the device and all the units and DUIBs which belong to it. The ICU stores all the screens by type, not by device. This means that all device driver screens are stored together, all unit information screens and all DUIB screens. The ICU "knows" which unit information and device-unit information screen relate to a particular device driver by the device name you enter in the "DEV" parameter.

```
    (IL)  Interrupt Level [Encoded Level]          073H
```

This parameter specifies the encoded interrupt level for the output buffer full interrupt. The interrupt task uses this value to associate the interrupt task with the correct interrupt level. The default value 0073H (0000 0000 0111 0011 binary) specifies slave interrupt level 3.

The possible values for this field are encoded as follows (where bit 0 is the low-order bit):

Master Level	Code	Slave Level	Attached Code	Master Level	Code	Slave Level	Attached Code
0	0008H	0-7	0000-0007H	4	0048H	0-7	0040-0047H
1	0018H	0-7	0010-0017H	5	0058H	0-7	0050-0057H
2	0028H	0-7	0020-0027H	6	0068H	0-7	0060-0067H
3	0038H	0-7	0030-0037H	7	0078H	0-7	0070-0077H

```
    (CAD) Channel A Data Port [0-0FFFFH]           086H
```

This parameter specifies the address of channel A of the 82530 data port.

```
    (CAS) Channel A Status Port [0-0FFFFH]         084H
```

This parameter specifies the address of channel A of the 82530 control/status port.

```
    (AIF) Channel A Inrate Freq. [0-0FFFFFFFFH]    04B0000H
```

This parameter specifies the input frequency of the 82530 timer. The default value (004B0000H) is equivalent to 4915200 decimal.

```
    (CBD) Channel B Data Port [0-0FFFFH]           082H
```

This parameter specifies the address of channel B of the 82530 data port.

```
    (CBS) Channel B Status Port [0-0FFFFH]         080H
```

This parameter specifies the address of channel B of the 82530 control/status port.

```
    (BIF) Channel B Inrate Freq. [0-0FFFFFFFFH]   04B0000H
```

This parameter specifies the input frequency of the 82530 timer. The default value (004B0000H) is equivalent to 4915200 decimal.

Query Screen

After you have completed the "82530 Terminal Driver" screen, the query screen is displayed. It contains only one line:

```
    Do you want any/more 82530 Terminal Driver DEVICEs ?
```

Respond "Yes" to this prompt if you wish to add another device driver of this type. Otherwise, respond with a "No".

82530 Unit Information Screen

The ICU uses the information from the following screen to create a unit information table for the 82530 terminal driver.

```
(U2530)      82530 Terminal Driver Unit Information

(DEV) Device Name [1-16 Chars]
(NAM) Unit Info Name [1-16 Chars]
(LEM) Line Edit Mode [Trans/Normal/Flush]          NORMAL
(ECH) Echo Mode [Yes/No]                           YES
(IPC) Input Parity Control [Yes/No]                NO
(OPC) Output Parity Control [Yes/No]               NO
(OCC) Output Control in Input [Yes/No]             YES
(OSC) OSC Controls [Both/In/Out/Neither]           BOTH
(DUP) Duplex Mode [Full/Half]                      FULL
(TRM) Terminal Type [CRT/HardCopy]                 CRT
(MC)  Modem Control [Yes/No]                       NO
(RPC) Read Parity Checking [See Help/0-3]          0
(WPC) Write Parity Checking [See Help/0-4]         4
(BR)  Baud Rate [0-65535]                          9600
(SN)  Scroll Number [0-65535]                      18

Enter [ Abbreviation = new_value / Abbreviation ? / H ]
:
```

(DEV) Device Name [1-16 Chars]

The name you enter for this parameter must be the same name you entered in the "DEV" parameter on the driver screen. This name provides the logical ICU connection between the driver and all of its units. Thus, when a device is deleted it is possible to delete all the related units at the same time.

(NAM) Unit Info Name [1-16 Chars]

This parameter specifies a unique name for this unit information table. The first character must be an alphabetic character. Refer to the *ASM286 Language Reference Manual* for rules regarding this name.

A DUIB uses the "Unit Info Name" to point to this particular unit information table. When developing your initial systems, you can create unit information tables that are never pointed to by a DUIB. There is no harm in this process during the development stages. However, when you configure your final system, eliminate all unused unit information tables to save memory.

```
    (LEM) Line Edit Mode [Trans/Normal/Flush}      NORMAL
```

This parameter specifies the initial default line editing mode. You must choose from the following three options:

Transparent

Console input is transparent (not line-edited). The Terminal Support Code transmits input to the requesting task exactly as entered at the terminal. Before being transmitted, the Terminal Support Code accumulates data in a buffer until an operator enters the requested number of characters.

Normal

Console input is used for line editing. Edited data accumulates in a buffer until an operator enters a carriage return.

Flush

Console input is not line-edited and the Terminal Support Code transmits input to the requesting task exactly as entered at the terminal. Before being transmitted, the Terminal Support Code accumulates data in a buffer until it receives an input request. At that time, it transmits the contents of the buffer (or the number of characters requested, if the buffer contains more than that number) to the requesting task. If any characters remain in the buffer, the Terminal Support Code saves the characters for the next input request.

You can alter the value you specify for this parameter at run time if you invoke any of the SPECIAL system calls or send an OSC sequence. OSC sequences are described in the *Extended iRMX II Device Drivers User's Guide.*

```
    (ECH) Echo Mode [Yes/No]                       YES
```

Specify "Yes" if you want characters entered into the terminal to be "echoed" to the terminal's display screen. Otherwise, specify "No".

You can alter the value you specify for this parameter at run time if you invoke any of the SPECIAL system calls or send an OSC sequence. OSC sequences are described in the *Extended iRMX II Device Drivers User's Guide.*

```
    (IPC) Input Parity Control [Yes/No]            NO
```

Specify "Yes" if you want the system to change the parity bit (bit 7) of characters entered into the terminal to zero. Specify "No" if you do not want the terminal driver to change bit 7 of the input character.

You can alter the value you specify for this parameter at run time if you invoke any of the SPECIAL system calls or send an OSC sequence. OSC sequences are described in the *Extended iRMX II Device Drivers User's Guide.*

```
    (OPC) Output Parity Control [Yes/No]           NO
```

Specify "Yes" if you want the system to change the parity bit (bit 7) of characters being output to the terminal to zero. Specify "No" if you want bit 7 in the output characters to remain unchanged.

You can alter the value you specify for this parameter at run time if you invoke any of the SPECIAL system calls or send an OSC sequence. OSC sequences are described in the *Extended iRMX II Device Drivers User's Guide.*

```
    (OCC) Output Control in Input [Yes/No]         YES
```

Specify "Yes" if you want the Terminal Support Code to accept output control characters in the input stream. Specify "No" if you want the Terminal Support Code to ignore output control characters. Control characters are described in the *Extended iRMX II Device Drivers User's Guide.*

You can alter the value you specify for this parameter at run time if you invoke any of the SPECIAL system calls or send an OSC sequence. OSC sequences are described in the *Extended iRMX II Device Drivers User's Guide.*

```
    (OSC) OSC Controls [Both/In/Out/Neither]       BOTH
```

This parameter specifies whether the device driver should act upon Operating System Command (OSC) controls when they appear in either an input or an output stream. Choose one of the following options as the initial default value for the device driver:

Both — Act upon OSC control sequences in either input or output stream (from either terminal or program).

Input — Act upon OSC control sequences in input stream only (from terminal and not from program).

Output — Act upon OSC control sequences in output stream only (from program and not from terminal).

Neither Do not act upon OSC control sequences.

The OSC control sequence used in communicating from a program or a terminal to the Operating System is described in the *Extended iRMX II Device Drivers User's Guide*. You can alter the value you specify for this parameter at run time if you invoke any of the SPECIAL system calls or send an OSC sequence.

```
   (DUP) Duplex Mode [Full/Half]                   FULL
```

This parameter specifies the line protocol mode for a terminal. Choose either full-duplex or half-duplex.

When line protocol mode is full-duplex, the terminal driver concurrently handles input to and output from the terminal. If you specified both echo mode and full-duplex, the terminal driver echoes each character. When the line protocol mode is half-duplex, there can be input to and output from the terminal, but not concurrently. If you specify both echo mode and half-duplex, the terminal, not the terminal driver, must echo each character.

You can alter the value you specify for this parameter at run time if you invoke any of the SPECIAL system calls or send an OSC sequence. OSC sequences are described in the *Extended iRMX II Device Drivers User's Guide*.

```
   (TRM) Terminal Type [CRT/HardCopy]              CRT
```

This parameter specifies how your terminal supports the rubout function. Respond "CRT" if your terminal can backspace and leave a blank character on the screen for each character "rubbed out". Respond "Hard Copy" if you terminal cannot backspace and leave a blank character on previously-displayed characters.

You can alter the value you specify for this parameter at run time if you invoke any of the SPECIAL system calls or send an OSC sequence. OSC sequences are described in the *Extended iRMX II Device Drivers User's Guide*.

```
   (MC)  Modem Control [Yes/No]                    NO
```

Specify "Yes" to establish an initial modem-based link between a task and a terminal. Specify "No" if your terminal is not connected to a modem.

If you specify "Yes" to this parameter, set the "OSC Controls" parameter to either "Both" or "Out".

You can alter the value you specify for this parameter at run time if you invoke any of the SPECIAL system calls or send an OSC sequence.

```
    (RPC) Read Parity Checking [See Help/0-3]      0
```

This parameter specifies how the hardware in your system will handle read parity checking. A response of 0 or 1 specifies 8-bit data with no parity checking. A response of 2 or 3 specifies 7-bit data with parity checking. In this section, "parity bit" refers to bit seven. Specify one of the following four values:

0 Ignore parity checking and set the input parity bit (bit 7) to zero.

1 Ignore parity checking and do not change the parity bit.

2 The driver expects even parity on input and sets the parity bit to 0 on meeting this condition.

The driver sets the parity bit to 1 if odd parity is received, a framing error occurs (receive stop bit is zero), or an overrun error is encountered (a new character has been received before the interrupt routine for character processing has completed).

3 Driver expects odd parity. If a parity error, a framing error, or an overrun error is encountered, the parity bit is set to 1. Otherwise, the parity bit is set to zero.

Note that because this parameter specifies how the hardware deals with parity, a response of zero here negates a "No" response to the software parity checking parameter "(IPC) Input Parity Checking".

If you set "(RPC) Read Parity Checking" to two or three and you set "(IPC) Input Parity Checking" to "Yes", your application will not be able to detect a transmission error.

You can alter the value you specify for this parameter at run time if you invoke any of the SPECIAL system calls or send an OSC sequence. OSC sequences are described in the *Extended iRMX II Device Drivers User's Guide*.

```
    (WPC) Write Parity Checking [See Help/0-4]   4
```

This parameter specifies how the hardware in your system will handle write parity checking. A response of 0, 1, or 4 specifies 8-bit data with no parity checking. A response of 2 or 3 specifies 7-bit data with parity checking. In this section, "parity bit" refers to bit seven. Specify one of the following five values:

0 Set the output parity bit to zero.

1 Set the output parity bit to one.

2 Set the output parity bit to one if the total number of 1's in the character is odd. Set the parity bit to zero if the total number of 1's is even (even parity). This option should be used if the driver is using even parity checking for input.

3 Set the output parity bit to 0 if the total number of 1's in the character is odd (odd parity). Set the parity bit to 1 if the total number of 1's is even. This option should be used if the driver is using odd parity checking for input.

4 Do not change the output parity bit.

Note that because this parameter specifies how the hardware deals with parity, a response of zero here negates a "No" response to the software parity checking parameter "(OPC) Output Parity Checking".

You can alter the value you specify for this parameter at run time if you invoke any of the SPECIAL system calls or send an OSC sequence. OSC sequences are described in the *Extended iRMX II Device Drivers User's Guide.*

```
    (BR)  Baud Rate [0-65535]                       9600
```

This parameter specifies the initial baud rate of this terminal. Specify a value of one if you want the controller to ascertain the initial baud rate automatically. The default value is 9600 decimal.

You can alter the value you specify for this parameter at run time if you invoke any of the SPECIAL system calls or send an OSC sequence. OSC sequences are described in the *Extended iRMX II Device Drivers User's Guide.*

```
    (SN)  Scroll Number [0-65535]                   18
```

This parameter specifies the number of lines to scroll when an operator enters the scrolling output control character (Control-W is the default). Typically the value you specify should be between 10 and 24. The default value is 18 decimal.

You can alter the value you specify for this parameter at run time if you invoke any of the SPECIAL system calls or send an OSC sequence. OSC sequences are described in the *Extended iRMX II Device Drivers User's.*

Query Screen

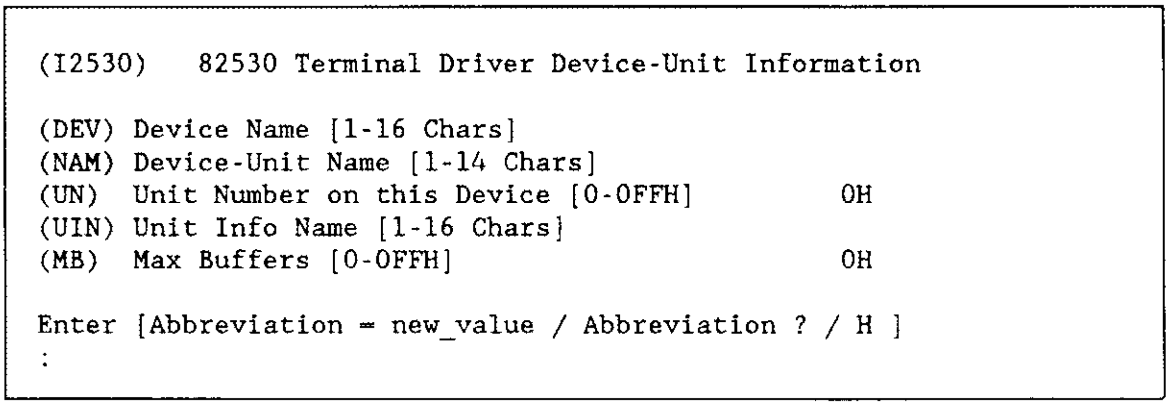
After you have completed the "82530 Terminal Driver Unit Information" screen, the query screen is displayed. It contains only one line:

```
    Do you want any/more 82530 Terminal Driver UNITs  ?
```

Respond to this prompt with a "Yes" if you need another unit information table for this device. Otherwise, respond to this prompt with a "No".

82530 Device Unit Information Screen

The ICU uses the information from the following screen to create a device unit information block (DUIB) for the 82530 driver.



```
(I2530)   82530 Terminal Driver Device-Unit Information

(DEV) Device Name [1-16 Chars]
(NAM) Device-Unit Name [1-14 Chars]
(UN)  Unit Number on this Device [0-0FFH]         0H
(UIN) Unit Info Name [1-16 Chars]
(MB)  Max Buffers [0-0FFH]                        0H

Enter [Abbreviation = new_value / Abbreviation ? / H ]
:
```

```
   (DEV)  Device Name [1-16 Chars]
```

The name you enter for this parameter must be the same name you entered in the "DEV" parameter on the driver screen. This name provides the logical ICU connection between the driver and all of its DUIBs. Thus, when a device is deleted it is possible to delete all the related units and DUIBs at the same time.

```
   (NAM) Device-Unit Name [1-14 Chars]
```

This parameter specifies a name that uniquely identifies the device-unit for the I/O System. The name you specify for this parameter is the physical name you specify when invoking the Human Interface ATTACHDEVICE command, the BIOS A$PHYSICAL$ATTACH$DEVICE system call, or the EIOS LOGICAL$ATTACH$DEVICE system call.

The ICU allows you to enter from one to fourteen characters. Refer to the *ASM286 Language Reference Manual* for rules regarding this name.

```
   (UN)  Unit Number on this Device [0-0FFH]     0H
```

This parameter specifies the unit number of the device-unit. The iSBX 354 MULTIMODULE board can support two units per MULTIMODULE board. The unit numbers for the device begin with zero and increase sequentially. Because each 82530 serial communications controller (SCC) has two channels, number the units as follows:

unit 0 = SCC channel A   unit 1 = SCC channel B

```
    (UIN) Unit Info Name [1-16 Chars]
```

This parameter specifies the "Unit Info Name" of a unit information table that fills the needs of this DUIB. The following rules apply to the name you choose:

- You must create a unit information table with this "unit info name."
- Each DUIB can point to only one unit information table.
- A particular unit information table can be pointed to by more than one DUIB.

```
    (MB)  Max Buffers [0-0FFH]                       0H
```

This parameter specifies the maximum number of buffers that the Extended I/O System can allocate for this device's I/O. Unless you are planning on using this terminal for output only, do not change the default value. For terminals used for output only, you may want to specify a small non-zero value. Note, however, that the EIOS will delay output until the buffer is full or you close the connection to the device.

Query Screen

After you have completed the "82530 Terminal Driver Device-Unit Information" screen, the query screen is displayed. It contains only one line:

```
    Do you want any/more 82530 Terminal Driver DUIBs ?
```

Respond "Yes" to this prompt, if you plan to use the 82530 driver with terminals that have different characteristics or if your system has two devices with the same characteristics, but different unit numbers.

While developing your initial systems, you can create as many device-unit information blocks as you want. The number of DUIBs can exceed the number of devices on your system. The particular DUIB associated with the device depends on the physical name you use when attaching it. Once you know that you will never need a particular DUIB, save memory by deleting it from your definition file before you generate your configuration files.

iSBC® 534 Driver Parameters

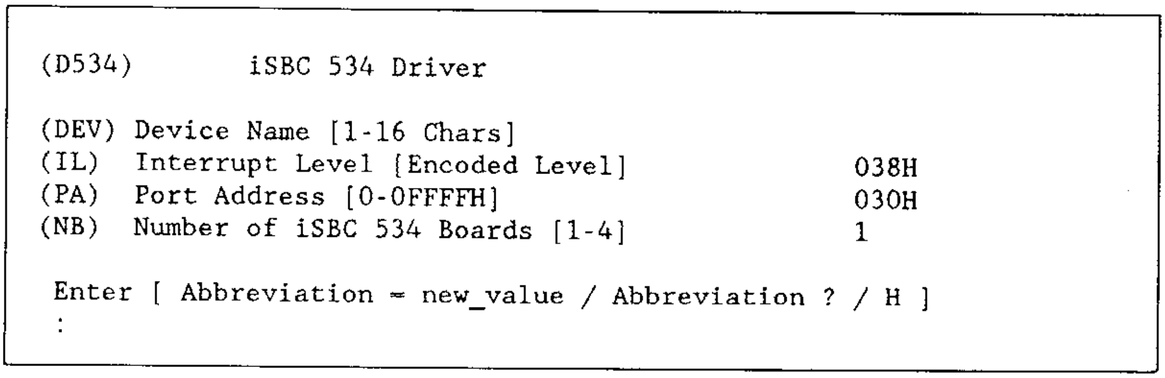
The iSBC 534 terminal driver is a custom device driver which allows I/O to a terminal device via the iSBC 534 communications board. Three screens define the interface between the iSBC 534 terminal support driver and the I/O system. These screens relate to the three device configuration tables: the device information table, the unit information table, and the device unit information block (DUIB). Refer to the *Extended iRMX II Device Drivers User's Guide* for more information about these tables.

The iSBC 534 terminal driver is not part of the Start-up system supplied by Intel. The values you see displayed in the following sections are the ICU default values.

iSBC® 534 Driver Screen

The ICU uses the information from the following screen to create a device information table for the iSBC 534 driver. If your system includes more than one iSBC 534 controller, you can set your system up in two ways:

- You can specify a unique interrupt level and port address for each controller.
- You can specify the total number of iSBC 534 controllers on the device information screen (driver screen).



```
(D534)       iSBC 534 Driver

(DEV) Device Name [1-16 Chars]
(IL)  Interrupt Level [Encoded Level]           038H
(PA)  Port Address [0-0FFFFH]                   030H
(NB)  Number of iSBC 534 Boards [1-4]           1

 Enter [ Abbreviation = new_value / Abbreviation ? / H ]
 :
```

```
   (DEV) Device Name [1-16 Chars]
```

This parameter provides a means of associating the device and all the units and DUIBs which belong to it. The ICU stores all the screens by type, not by device. This means that all device driver screens are stored together, all unit information screens and all DUIB screens. The ICU "knows" which unit information and device-unit information screen relate to a particular device driver by the device name you enter in the "DEV" parameter.

```
   (IL)  Interrupt Level [Encoded Level]           038H
```

This parameter specifies the encoded interrupt level for the iSBC 534 driver. The interrupt task uses this value to associate the interrupt task with the correct interrupt level. The default value 038H (0000 0000 0011 1000 binary) specifies master interrupt level 3.

The possible values for this field are encoded as follows (where bit 0 is the low-order bit):

Master Level	Code	Slave Level	Attached Code	Master Level	Code	Slave Level	Attached Code
0	0008H	0-7	0000-0007H	4	0048H	0-7	0040-0047H
1	0018H	0-7	0010-0017H	5	0058H	0-7	0050-0057H
2	0028H	0-7	0020-0027H	6	0068H	0-7	0060-0067H
3	0038H	0-7	0030-0037H	7	0078H	0-7	0070-0077H

```
   (PA)  Port Address [0-0FFFFH]                   030H
```

This parameter specifies the I/O wakeup address of the Terminal Communication Controller. Set this value to match the jumper configuration of the iSBC 534 board. If you have more than one iSBC 534 board connected to the same interrupt line, specify the port address of the first iSBC 534 board.

```
   (NB)  Number of iSBC 534 Boards [1-4]           1
```

This parameter specifies the number of iSBC 534 boards that share the same interrupt line. The port addresses of these iSBC 534 boards must be consecutive and the interval between addresses must be 16.

Query Screen

After you have completed the "iSBC 534 Terminal Driver" screen, the query screen is displayed. It contains only one line:

```
   Do you want any/more iSBC 534 DEVICEs ?
```

Respond to this prompt with a "Yes" if you want to add more device drivers of this type.

iSBC® 534 Unit Information Screen

The ICU uses the information from the following screen to create a unit information table for the iSBC 534 driver.

```
(U534) iSBC 534 Driver Unit Information

(DEV) Device Name [1-16 Chars]
(NAM) Unit Info Name [1-16 Chars]
(LEM) Line Edit Mode [Trans/Normal/Flush]               NORMAL
(ECH) Echo Mode [Yes/No]                                YES
(IPC) Input Parity Control [Yes/No]                     NO
(OPC) Output Parity Control [Yes/No]                    NO
(OCC) Output Control in Input [Yes/No]                  YES
(OSC) OSC Controls [Both/In/Out/Neither]                BOTH
(DUP) Duplex Mode [Full/Half]                           FULL
(TRM) Terminal Type [CRT/HardCopy]                      CRT
(MC)  Modem Control [Yes/No]                            NO
(RPC) Read Parity Checking [See Help/0-3]               0
(WPC) Write Parity Checking [See Help/0-4]              4
(BR)  Baud Rate [0-65535]                               9600
(SN)  Scroll Number [0-65535]                           18

Enter [ Abbreviation = new_value / Abbreviation ? / H ]
:
```

```
(DEV)  Device Name [1-16 Chars]
```

The name you enter for this parameter must be the same name you entered in the "DEV" parameter on the driver screen. This name provides the logical ICU connection between a driver and all of its units. Thus, when a device is deleted it is possible to delete all the related units at the same time.

```
(NAM) Unit Info Name [1-16 Chars]
```

This parameter specifies a unique name for this unit information table. The first character must be an alphabetic character. Refer to the *ASM286 Language Reference Manual* for rules regarding this name.

A DUIB uses the "Unit Info Name" to point to this particular unit information table. When developing your initial systems, you can create unit information tables that are never pointed to by a DUIB. There is no harm in this process during the development stages. However, when you configure your final system, eliminate all unused unit information tables to save memory.

```
    (LEM) Line Edit Mode [Trans/Normal/Flush]       NORMAL
```

This parameter specifies the initial default line editing mode. You must choose from the following three options:

Transparent — Console input is transparent (not line-edited). The Terminal Support Code transmits input to the requesting task exactly as entered at the terminal. Before being transmitted, the Terminal Support Code accumulates data in a buffer until an operator enters the requested number of characters.

Normal — Console input is line-edited. Edited data accumulates in a buffer until an operator enters a carriage return or other line terminator.

Flush — Console input is not line-edited and the Terminal Support Code transmits input to the requesting task exactly as entered at the terminal. Before being transmitted, the Terminal Support Code accumulates data in a buffer until it receives an input request. At that time, it transmits the contents of the buffer (or the number of characters requested, if the buffer contains more than that number) to the requesting task. If any characters remain in the buffer, the Terminal Support Code saves the characters for the next input request.

You can alter the value you specify for this parameter at run time if you invoke any of the SPECIAL system calls or send an OSC sequence. OSC sequences are described in the *Extended iRMX II Device Drivers User's Guide.*

```
    (ECH) Echo Mode [Yes/No]                        YES
```

Specify "Yes" if you want characters entered into the terminal to be "echoed" to the terminal's display screen. Otherwise, specify "No".

You can alter the value you specify for this parameter at run time if you invoke any of the SPECIAL system calls or send an OSC sequence. OSC sequences are described in the *Extended iRMX II Device Drivers User's Guide.*

```
    (IPC) Input Parity Control [Yes/No]             NO
```

Specify "Yes" if you want the system to change the parity bit (bit 7) of characters entered into the terminal to zero. Specify "No" if you do not want the terminal driver to change bit 7 of the input character.

You can alter the value you specify for this parameter at run time if you invoke any of the SPECIAL system calls or send an OSC sequence. OSC sequences are described in the *Extended iRMX II Device Drivers User's Guide.*

```
    (OPC) Output Parity Control [Yes/No]            NO
```

Specify "Yes" if you want the system to change the parity bit (bit 7) of characters being output to the terminal to zero. Specify "No" if you want bit 7 in the output characters to remain unchanged.

You can alter the value you specify for this parameter at run time if you invoke any of the SPECIAL system calls or send an OSC sequence. OSC sequences are described in the *Extended iRMX II Device Drivers User's Guide.*

```
    (OCC) Output Control in Input [Yes/No]          YES
```

Specify "Yes" if you want the Terminal Support Code to accept output control characters in the input stream. Specify "No" if you want the Terminal Support Code to ignore output control characters. Control characters are described in the *Extended iRMX II Device Drivers User's Guide.*

You can alter the value you specify for this parameter at run time if you invoke any of the SPECIAL system calls or send an OSC sequence. OSC sequences are described in the *Extended iRMX II Device Drivers User's Guide.*

```
    (OSC) OSC Controls [Both/In/Out/Neither]        BOTH
```

This parameter specifies whether the device driver should act upon Operating System Command (OSC) controls when they appear in either an input or an output stream. Choose one of the following options as the initial default value for the device driver:

Both — Act upon OSC control sequences in either input or output stream (from either terminal or program).

Input — Act upon OSC control sequences in input stream only (from terminal and not from program).

Output — Act upon OSC control sequences in output stream only (from program and not from terminal).

Neither Do not act upon OSC control sequences.

The OSC control sequence, used in communicating from a program or a terminal to an operating system, is described in the *Extended iRMX II Device Drivers User's Guide*. You can alter the value you specify for this parameter at run time if you invoke any of the SPECIAL system calls or send an OSC sequence.

```
    (DUP) Duplex Mode [Full/Half]                      FULL
```

This parameter specifies the line protocol mode for a terminal. Choose either full-duplex or half-duplex.

When line protocol mode is full-duplex, the terminal driver concurrently handles input to and output from the terminal. If you specified both echo mode and full-duplex, the terminal driver echoes each character. When the line protocol mode is half-duplex, there can be input to and output from the terminal, but not concurrently. If you specify both echo mode and half-duplex, the terminal, not the terminal driver, must echo each character.

You can alter the value you specify for this parameter at run time if you invoke any of the SPECIAL system calls or send an OSC sequence. OSC sequences are described in the *Extended iRMX II Device Drivers User's Guide.*

```
    (TRM) Terminal Type [CRT/HardCopy]                 CRT
```

This parameter specifies how your terminal supports the rubout function. Respond "CRT" if your terminal can backspace and leave a blank character on the screen for each character "rubbed out." Respond "Hard Copy" if your terminal cannot backspace and leave a blank character on previously-displayed characters.

You can alter the value you specify for this parameter at run time if you invoke any of the SPECIAL system calls or send an OSC sequence. OSC sequences are described in the *Extended iRMX II Device Drivers User's Guide.*

```
    (MC)  Modem Control [Yes/No]                       NO
```

Specify "Yes" to establish an initial modem-based link between a task and a terminal. Specify "No" if your terminal is not connected to a modem.

If you specify "Yes" to this parameter, set the "OSC Controls" parameter to either "Both" or "Out".

You can alter the value you specify for this parameter at run time if you invoke any of the SPECIAL system calls or send an OSC sequence.

```
    (RPC) Read Parity Checking [See Help/0-3]       0
```

This parameter specifies how the hardware in your system will handle read parity checking. A response of 0 or 1 specifies 8-bit data with no parity checking. A response of 2 or 3 specifies 7-bit data with parity checking. In this section, "parity bit" refers to bit seven. Specify one of the following four values:

0 Ignore parity checking and set the input parity bit (bit 7) to zero.

1 Ignore parity checking and do not change the parity bit.

2 The driver expects even parity on input and sets the parity bit to 0 on meeting this condition.

The driver sets the parity bit is to 1 if odd parity is received, a framing error occurs (receive stop bit is zero), or an overrun error is encountered (a new character has been received before the interrupt routine for character processing has completed).

Driver expects odd parity. If a parity error, a framing error, or an overrun error is encountered, the parity bit is set to 1. Otherwise, the parity bit is set to zero.

Note that because this parameter specifies how the hardware deals with parity, a response of zero here negates a "No" response to the software parity checking parameter "(IPC) Input Parity Checking".

If you set "(RPC) Read Parity Checking" to two or three and you set "(IPC) Input Parity Checking" to "Yes", your application will not be able to detect a transmission error.

You can alter the value you specify for this parameter at run time if you invoke any of the SPECIAL system calls or send an OSC sequence. OSC sequences are described in the *Extended iRMX II Device Drivers User's Guide.*

```
    (WPC) Write Parity Checking [See Help/0-4]     4
```

This parameter specifies how the hardware in your system will handle write parity checking. A response of 0, 1, or 4 specifies 8-bit data with no parity checking. A response of 2 or 3 specifies 7-bit data with parity checking. In this section, "parity bit" refers to bit seven. Specify one of the following five values:

0 Set the output parity bit to zero.

1 Set the output parity bit to one.

2 Set the output parity bit to one if the total number of 1's in the character is odd. Set the parity bit to zero if the total number of 1's is even (even parity). This option should be used if the driver is using even parity checking for input.

3 Set the output parity bit to 0 if the total number of 1's in the character is odd (odd parity). Set the parity bit to 1 if the total number of 1's is even. This option should be used if the driver is using odd parity checking for input.

4 Do not change the output parity bit.

Note that because this parameter specifies how the hardware deals with parity, a response of zero here negates a "No" response to the software parity checking parameter "(OPC) Output Parity Checking".

You can alter the value you specify for this parameter at run time if you invoke any of the SPECIAL system calls or send an OSC sequence. OSC sequences are described in the *Extended iRMX II Device Drivers User's Guide*.

```
   (BR)  Baud Rate [0-65535]                        9600
```

This parameter specifies the initial baud rate of this terminal. Specify a value of one if you want the controller to ascertain the initial baud rate automatically. The default value is 9600 decimal.

You can alter the value you specify for this parameter at run time if you invoke any of the SPECIAL system calls or send an OSC sequence. OSC sequences are described in the *Extended iRMX II Device Drivers User's Guide*.

```
   (SN)  Scroll Number [0-65535]                    18
```

This parameter specifies the number of lines to scroll when an operator enters the scrolling output control character (Control-W is the default). Typical values should range from 10 to 24. The default value is 18 decimal.

You can alter the value you specify for this parameter at run time if you invoke any of the SPECIAL system calls or send an OSC sequence. OSC sequences are described in the *Extended iRMX II Device Drivers User's Guide*.

Query Screen

After you have completed the "iSBC 534 Terminal Driver Unit Information" screen, the query screen is displayed. It contains only one line:

```
   Do you want any/more iSBC 534 UNITs ?
```

Respond "Yes" to this prompt if you need to define another unit information table for this device. Otherwise, respond with a "No".

iSBC® 534 Device Unit Information Screen

The ICU uses the information from the following screen to create a device unit information block (DUIB) for the iSBC 534 driver.

```
(I534)     iSBC 534 Driver Device-Unit Information

(DEV) Device Name [1-16 Chars]
(NAM) Device-Unit Name [1-14 Chars]
(UN)  Unit Number on this Device [0-0FFH]          0H
(UIN) Unit Info Name [1-16 Chars]
(MB)  Max Buffers [0-0FFH]                         0H

Enter [Abbreviation = new_value / Abbreviation ? / H ]
:
```

```
(DEV)  Device Name [1-16 Chars]
```

The name you enter for this parameter must be the same name you entered in the "DEV" parameter on the "iSBC 534 Driver" screen. This name provides the logical ICU connection between a driver and all of its DUIBs. Thus, when a device is deleted it is possible to delete all the related units and DUIBs at the same time.

```
(NAM) Device-Unit Name [1-14 Chars]
```

This parameter specifies a name that uniquely identifies the device-unit for the I/O System. The name you specify for this parameter is the physical name you specify when invoking the Human Interface ATTACHDEVICE command, the BIOS A$PHYSICAL$ATTACH$DEVICE system call, or the EIOS LOGICAL$ATTACH$DEVICE system call.

The ICU allows you to enter from one to fourteen characters. Refer to the *ASM286 Language Reference Manual* for rules regarding this name.

```
(UN)  Unit Number on this Device [0-0FFH]      0H
```

This parameter specifies the unit number of this device-unit. The unit numbers for the device begin with zero and increase sequentially.

Each of the four possible serial lines is a single unit. If several boards share a single interrupt line, their lines are treated as separate units of a single device.

```
    (UIN) Unit Info Name [1-16 Chars]
```

This parameter specifies the "Unit Info Name" of a unit information table that fills the needs of this DUIB. The following rules apply to the name you choose:

- You must create a unit information table with this "unit info name".
- Each DUIBs can point to only one unit information table.
- A particular unit information table can be pointed to by more than one DUIB.

```
    (MB)  Max Buffers [0-0FFH]                          0H
```

This parameter specifies the maximum number of buffers that the Extended I/O System can allocate for this device's I/O. Unless you are planning on using this terminal for output only, do not change the default value. For terminals used for output only, you may want to specify a small non-zero value. Note, however, that the EIOS will delay output until the buffer is full or you close the connection to the device.

Query Screen

After you have completed the "iSBC 534 Terminal Driver Device-Unit Information" screen, the query screen is displayed. It contains only one line:

```
    Do you want any/more iSBC 534 Driver DUIBs ?
```

Respond "Yes" to this prompt if you plan to use the iSBC 534 controller with two terminals that have different characteristics or if you have two devices with the same characteristics, but different unit numbers.

While developing your initial systems, you can create as many device-unit information blocks as you want. The number of DUIBs can exceed the number of devices on your system. The particular DUIB associated with the device depends on the physical name you use when attaching it. Once you know that you will never need a particular DUIB, save memory by deleting it from your definition file before you generate your configuration files (refer to Chapter 16 for additional information on generating configuration files).

iSBC® 544A Driver Parameters

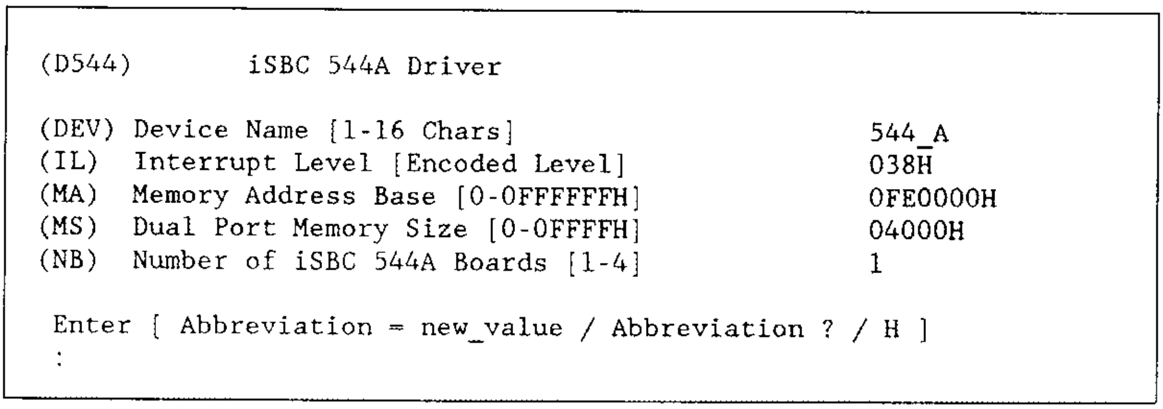
The iSBC 544A terminal driver is a custom device driver which allows I/O to a terminal device via the iSBC 544A intelligent communications board. Three screens define the interface between the iSBC 544A terminal support driver and the I/O system. These screens relate to the three device configuration tables: the device information table, the unit information table, and the device unit information block (DUIB). Refer to the *Extended iRMX II Device Drivers User's Guide* for further information about these tables.

The values shown on the screens in this section are the values you see when you invoke the ICU with the 28612.def file.

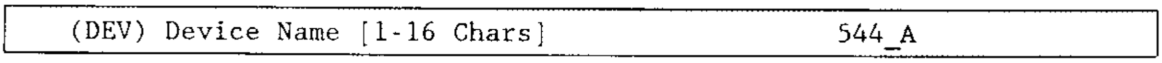
iSBC® 544A Driver Screen

The ICU uses the information from the following screen to create a device information table for the iSBC 544A driver. If your system includes more than one iSBC 544A controller, you can set your system up in two ways:

- You can specify a unique interrupt level and memory address base for each controller.
- You can specify the total number of iSBC 544A controllers on the device information screen (driver screen).



```
(D544)       iSBC 544A Driver

(DEV) Device Name [1-16 Chars]                            544_A
(IL)  Interrupt Level [Encoded Level]                     038H
(MA)  Memory Address Base [0-0FFFFFFH]                    0FE0000H
(MS)  Dual Port Memory Size [0-0FFFFH]                    04000H
(NB)  Number of iSBC 544A Boards [1-4]                    1

 Enter [ Abbreviation = new_value / Abbreviation ? / H ]
 :
```



```
   (DEV) Device Name [1-16 Chars]                        544_A
```

This parameter provides a means of associating the device and all the units and DUIBs which belong to it. The ICU stores all the screens by type, not by device. This means that all device driver screens are stored together, all unit information screens and all DUIB screens. The ICU "knows" which unit information and device-unit information screen relate to a particular device driver by the device name you enter in the "DEV" parameter.

```
(IL)  Interrupt Level [Encoded Level]           038H
```

This parameter specifies the encoded interrupt level for the iSBC 544A driver. The interrupt task uses this value to associate the interrupt task with the correct interrupt level. The default value 038H (0000 0000 0011 1000 binary) specifies master interrupt level 3.

The possible values for this field are encoded as follows (where bit 0 is the low-order bit):

Master Level	Code	Slave Level	Attached Code	Master Level	Code	Slave Level	Attached Code
0	0008H	0-7	0000-0007H	4	0048H	0-7	0040-0047H
1	0018H	0-7	0010-0017H	5	0058H	0-7	0050-0057H
2	0028H	0-7	0020-0027H	6	0068H	0-7	0060-0067H
3	0038H	0-7	0030-0037H	7	0078H	0-7	0070-0077H

```
(MA)  Memory Address Base [0-0FFFFFFH]          0FE0000H
```

This parameter specifies the 24-bit physical address of the dual port RAM that matches the jumper configuration of the iSBC 544A board. This address can be anywhere within the system's 16M byte address space. If you have more than one iSBC 544A board connected to the same interrupt line, specify the base address of the first iSBC 544A board.

The driver assumes that your dual-port memory is contiguous. Therefore, if you have multiple boards, set the jumpers on the second board to indicate a base address greater than the first board. The value you add to the base address to determine the second address is the dual port memory board size. Use the same approach on the remaining boards. For example, starting with a base address of 0FE0000H and a dual port memory size of 4000H paragraphs (your response to the "(MS) Dual Port Memory Size" parameter), the addresses for the second, third, and fourth boards should be 0FE4000H, 0FE8000H, and 0FEC000H respectively.

In addition, you must prevent the Operating System from assigning this dual-port memory. To prevent this automatic assignment, ensure that none of the blocks of memory you define in the "Memory For System" or "Memory for Free Space Manager" screens include addresses that represent iSBC 544A on-board memory. For example, if the base address of your iSBC 544A controller board's memory address is 0FE0000H, then you must specify 0FDFFFFH on the "Memory For System" screen as an ending address.

```
    (MS)  Dual Port Memory Size [0-0FFFFH]            04000H
```

This parameter specifies the size of each board's dual port memory. This value should match the appropriate jumpers on your iSBC 544A board. Do not change the default value.

The iSBC 544A board features a two-part access arrangement in which a bus master board can access the on-board dynamic RAM via the MULTIBUS system bus. If there are multiple boards on one line, they must have consecutive memory addresses, separated by the dual port memory board size.

```
    (NB)  Number of iSBC 544A Boards [1-4]            1
```

This parameter specifies the number of iSBC 544A boards that share the same interrupt line. The addresses of these iSBC 544A boards must be consecutive and the interval between addresses must be the value you specified for the "(MS) Dual Port Memory Size" parameter.

Query Screen

After you have completed the "544A Terminal Driver" screen, the query screen is displayed. It contains only one line:

```
    Do you want any/more iSBC 544A DEVICEs ?
```

Respond to this prompt with a "Yes" if you want to add more device drivers of this type.

iSBC® 544A Unit Information Screen

The ICU uses the information from the following screen to create a unit information table for the iSBC 544A driver.

```
(U544)  iSBC 544A Unit Information

(DEV) Device Name [1-16 Chars]                          544_A
(NAM) Unit Info Name [1-16 Chars]                       UINFO_544A
(LEM) Line Edit Mode [Trans/Normal/Flush]               NORMAL
(ECH) Echo Mode [Yes/No]                                YES
(IPC) Input Parity Control [Yes/No]                     NO
(OPC) Output Parity Control [Yes/No]                    NO
(OCC) Output Control in Input [Yes/No]                  YES
(OSC) OSC Controls [Both/In/Out/Neither]                BOTH
(DUP) Duplex Mode [Full/Half]                           FULL
(TRM) Terminal Type [CRT/HardCopy]                      CRT
(MC)  Modem Control [Yes/No]                            NO
(RPC) Read Parity Checking [See Help/0-3]               0
(WPC) Write Parity Checking [See Help/0-4]              4
(BR)  Baud Rate [0-65535]                               9600
(SN)  Scroll Number [0-65535]                           18

Enter [ Abbreviation = new_value / Abbreviation ? / H ]
:
```

```
(DEV)  Device Name [1-16 Chars]                  544_A
```

The name you enter for this parameter must be the same name you entered in the "DEV" parameter on the driver screen. This name provides the logical ICU connection between a driver and all of its units. Thus, when a device is deleted it is possible to delete all the related units at the same time.

```
(NAM) Unit Info Name [1-16 Chars]                UINFO_544A
```

This parameter specifies a unique name for this unit information table. The first character must be an alphabetic character. Refer to the *ASM286 Language Reference Manual* for rules regarding this name.

A DUIB uses the "Unit Info Name" to point to this particular unit information table. When developing your initial systems, you can create unit information tables that are never pointed to by a DUIB. There is no harm in this process during the development stages. However, when you configure your final system, eliminate all unused unit information tables to save memory.

```
    (LEM) Line Edit Mode [Trans/Normal/Flush}        NORMAL
```

This parameter specifies the initial default line editing mode. You must choose from the following three options:

Transparent
Console input is transparent (not line-edited). The Terminal Support Code transmits input to the requesting task exactly as entered at the terminal. Before being transmitted, the Terminal Support Code accumulates data in a buffer until an operator enters the requested number of characters.

Normal
Console input is line-edited. Edited data accumulates in a buffer until an operator enters a carriage return.

Flush
Console input is not line-edited and the Terminal Support Code transmits input to the requesting task exactly as entered at the terminal. Before being transmitted, the Terminal Support Code accumulates data in a buffer until it receives an input request. At that time, it transmits the contents of the buffer (or the number of characters requested, if the buffer contains more than that number) to the requesting task. If any characters remain in the buffer, the Terminal Support Code saves the characters for the next input request.

You can alter the value you specify for this parameter at run time if you invoke any of the SPECIAL system calls or send an OSC sequence. OSC sequences are described in the *Extended iRMX II Device Drivers User's Guide*.

```
    (ECH) Echo Mode [Yes/No]                          YES
```

Specify "Yes" if you want characters entered into the terminal to be "echoed" to the terminal's display screen. Otherwise, specify "No".

You can alter the value you specify for this parameter at run time if you invoke any of the SPECIAL system calls or send an OSC sequence. OSC sequences are described in the *Extended iRMX II Device Drivers User's Guide.*

```
    (IPC) Input Parity Control [Yes/No]              NO
```

Specify "Yes" if you want the system to change the parity bit (bit 7) of characters entered into the terminal to zero. Specify "No" if you do not want the system to change bit 7 of the input character.

You can alter the value you specify for this parameter at run time if you invoke any of the SPECIAL system calls or send an OSC sequence. OSC sequences are described in the *Extended iRMX II Device Drivers User's Guide.*

```
    (OPC) Output Parity Control [Yes/No]             NO
```

Specify "Yes" if you want the system to change the parity bit (bit 7) of characters being output to the terminal to zero. Specify "No" if you want bit 7 in the output characters to remain unchanged.

You can alter the value you specify for this parameter at run time if you invoke any of the SPECIAL system calls or send an OSC sequence. OSC sequences are described in the *Extended iRMX II Device Drivers User's Guide.*

```
    (OCC) Output Control in Input [Yes/No]         YES
```

Specify "Yes" if you want the Terminal Support Code to accept output control characters in the input stream. Specify "No" if you want the Terminal Support Code to ignore output control characters. Control characters are described in the *Extended iRMX II Device Drivers User's Guide*.

You can alter the value you specify for this parameter at run time if you invoke any of the SPECIAL system calls or send an OSC sequence. OSC sequences are described in the *Extended iRMX II Device Drivers User's Guide.*

```
    (OSC) OSC Controls [Both/In/Out/Neither]         BOTH
```

This parameter specifies whether the device driver should act upon Operating System Command (OSC) controls when they appear in either an input or an output stream. Choose one of the following options as the initial default value for the device driver:

Both — Act upon OSC control sequences in either input or output stream (from either terminal or program).

Input — Act upon OSC control sequences in input stream only (from terminal and not from program).

Output — Act upon OSC control sequences in output stream only (from program and not from terminal).

Neither Do not act upon OSC control sequences.

The OSC control sequence, used in communicating from a program or a terminal to the Operating System, is described in the *Extended iRMX II Device Drivers User's Guide*. You can alter the value you specify for this parameter at run time if you invoke any of the SPECIAL system call or send an OSC sequence.

```
    (DUP) Duplex Mode [Full/Half]                        FULL
```

This parameter specifies the line protocol mode for a terminal. Choose either full-duplex or half-duplex.

When line protocol mode is full-duplex, the terminal driver concurrently handles input to and output from the terminal. If you specified both echo mode and full-duplex, the terminal driver echoes each character. When the line protocol mode is half-duplex, there can be input to and output from the terminal, but not concurrently. If you specify both echo mode and half-duplex, the terminal, not the terminal driver, must echo each character.

You can alter the value you specify for this parameter at run time if you invoke any of the SPECIAL system calls or send an OSC sequence. OSC sequences are described in the *Extended iRMX II Device Drivers User's Guide*.

```
    (TRM) Terminal Type [CRT/HardCopy]                   CRT
```

This parameter specifies how your terminal supports the rubout function. Respond "CRT" if your terminal can backspace and leave a blank character on the screen for each character "rubbed out." Respond "Hard Copy" if your terminal cannot backspace and leave a blank character on previously-displayed characters.

You can alter the value you specify for this parameter at run time if you invoke any of the SPECIAL system calls or send an OSC sequence. OSC sequences are described in the *Extended iRMX II Device Drivers User's Guide*.

```
    (MC)  Modem Control [Yes/No]                         NO
```

Specify "Yes" to establish an initial modem-based link between a task and a terminal. Specify "No" if your terminal is not connected to a modem.

If you specify "Yes" to this parameter, set the "OSC Controls" parameter to either "Both" or "Out".

You can alter the value you specify for this parameter at run time if you invoke any of the SPECIAL system calls or send an OSC sequence.

```
    (RPC) Read Parity Checking [See Help/0-3]        0
```

This parameter specifies how the hardware in your system will handle read parity checking. A response of 0 or 1 specifies 8-bit data with no parity checking. A response of 2 or 3 specifies 7-bit data with parity checking. In this section, "parity bit" refers to bit seven. Specify one of the following four values:

0 Ignore parity checking and set the input parity bit (bit 7) to zero.

1 Ignore parity checking and do not change the parity bit.

2 The driver expects even parity on input and sets the parity bit to 0 on meeting this condition.

The driver sets the parity bit is to 1 if odd parity is received, a framing error occurs (receive stop bit is zero), or an overrun error is encountered (a new character has been received before the interrupt routine for character processing has completed).

3 Driver expects odd parity. If a parity error, a framing error, or an overrun error is encountered, the parity bit is set to 1. Otherwise, the parity bit is set to zero.

Note that because this parameter specifies how the hardware deals with parity, a response of zero here negates a "No" response to the software parity checking parameter "(IPC) Input Parity Checking".

If you set "(RPC) Read Parity Checking" to two or three and you set "(IPC) Input Parity Checking" to "Yes", your application will not be able to detect a transmission error.

You can alter the value you specify for this parameter at run time if you invoke any of the SPECIAL system calls or send an OSC sequence. OSC sequences are described in the *Extended iRMX II Device Drivers User's Guide*.

```
    (WPC) Write Parity Checking [See Help/0-4]       4
```

This parameter specifies how the hardware in your system will handle write parity checking. A response of 0, 1, or 4 specifies 8-bit data with no parity checking. A response of 2 or 3 specifies 7-bit data with parity checking. In this section, "parity bit" refers to bit seven. Specify one of the following five values:

0 Set the output parity bit to zero.

1 Set the output parity bit to one.

2 Set the output parity bit to one if the total number of 1's in the character is odd. Set the parity bit to zero if the total number of 1's is even (even parity). This option should be used if the driver is using even parity checking for input.

3 Set the output parity bit to 0 if the total number of 1's in the character is odd (odd parity). Set the parity bit to 1 if the total number of 1's is even. This option should be used if the driver is using odd parity checking for input.

4 Do not change the output parity bit.

Note that because this parameter specifies how the hardware deals with parity, a response of zero here negates a "No" response to the software parity checking parameter "(OPC) Output Parity Checking".

You can alter the value you specify for this parameter at run time if you invoke any of the SPECIAL system calls or send an OSC sequence. OSC sequences are described in the *Extended iRMX II Device Drivers User's Guide.*

```
   (BR)  Baud Rate [0-65535]                          9600
```

This parameter specifies the initial baud rate of this terminal. Specify a value of one if you want the controller to ascertain the initial baud rate automatically. The default value is 9600 decimal.

You can alter the value you specify for this parameter at run time if you invoke any of the SPECIAL system calls or send an OSC sequence. OSC sequences are described in the *Extended iRMX II Device Drivers User's Guide.*

```
   (SN)  Scroll Number [0-65535]                      18
```

This parameter specifies the number of lines to scroll when an operator enters the scrolling output control character (CONTROL-W is the default). Typical values should be from 10 to 24. The default value is 18 decimal.

You can alter the value you specify for this parameter at run time if you invoke any of the SPECIAL system calls or send an OSC sequence. OSC sequences are described in the *Extended iRMX II Device Drivers User's Guide.*

Query Screen

After you have completed the "iSBC 544A Terminal Driver Unit Information" screen, the query screen is displayed. It contains only one line:

```
   Do you want any/more iSBC 544A UNITs ?
```

Respond "Yes" to this prompt if you need to define another unit information table for this device. Otherwise, respond with a "No".

iSBC® 544A Device Unit Information Screen

The ICU uses the information from the following screen to create a device unit information block (DUIB) for the iSBC 544A driver.

```
(I544)    iSBC 544A Device-Unit Information

(DEV) Device Name [1-16 Chars]                          544_A
(NAM) Device-Unit Name [1-14 Chars]                     T2
(UN)  Unit Number on this Device [0-0FFH]               0H
(UIN) Unit Info Name [1-16 Chars]                       UINFO_544A
(MB)  Max Buffers [0-0FFH]                              0H

Enter [ Abbreviation = new_value / Abbreviation ? / H ]
:
```

```
   (DEV)  Device Name [1-16 Chars]                     544_A
```

The name you enter for this parameter must be the same name you entered in the "DEV" parameter on the "iSBC 544A Driver" screen. This name provides the logical ICU connection between a driver and all of its DUIBs. Thus, when a device is deleted it is possible to delete all the related units and DUIBs at the same time.

```
   (NAM) Device-Unit Name [1-14 chars]                 T2
```

This parameter specifies a name that uniquely identifies the device-unit for the I/O System. The name you specify for this parameter is the physical name you specify when invoking the Human Interface ATTACHDEVICE command, the BIOS A$PHYSICAL$ATTACH$DEVICE system call, or the EIOS LOGICAL$ATTACH$DEVICE system call.

The ICU allows you to enter from one to fourteen characters. Refer to the *ASM286 Language Reference Manual* for rules regarding this name.

```
   (UN)  Unit Number on this Device [0-0FFH]           0H
```

This parameter specifies the unit number of this device-unit. The unit numbers for the device begin with zero and increase sequentially.

Each of the four possible serial lines per board is a single unit. If several boards share a single interrupt line, their lines are treated as separate units of a single device.

```
(UIN) Unit Info Name [1-16 Chars]                    UINFO_544A
```

This parameter specifies the "Unit Info Name" of a unit information table that fills the needs of this DUIB. The following rules apply to the name you choose:

- You must create a unit information table with this "unit info name".
- Each DUIBs can point to only one unit information table.
- A particular unit information table can be pointed to by more than one DUIB.

```
(MB)  Max Buffers [0-0FFH]                           0H
```

This parameter specifies the maximum number of buffers that the Extended I/O System can allocate for this device's I/O. Unless you are planning on using this terminal for output only, do not change the default value. For terminals used for output only, you may want to specify a small non-zero value. Note, however, that the EIOS will delay output until the buffer is full or you close the connection to the device.

Query Screen

After you have completed the "544A Terminal Driver DUIB Information" screen, the query screen is displayed. It contains only one line:

```
Do you want any/more iSBC 544A DUIBs ?
```

Respond "Yes" to this prompt if you plan to use the iSBC 544A controller with two terminals that have different characteristics. Another reason to respond "Yes" to this prompt is if you have two devices with the same characteristics, but different unit numbers.

While developing your initial systems, you can create as many device-unit information blocks as you want. The number of DUIBs can exceed the number of devices on your system. The particular DUIB associated with the device depends on the physical name you use when attaching it. Once you know that you will never need a particular DUIB, save memory by deleting it from your definition file before you generate your configuration files (refer to Chapter 16 for additional information on generating configuration files).

Terminal Communications Controller Driver Parameters

The Terminal Communications Controller device driver controls intelligent terminal controllers that can manage buffered input and output. It may consist of one of the following boards:

- iSBC 188/48 or iSBC 188/56 communication board--manages an 8-channel controller.
- iSBC 546 communication board--manages a serial line printer, a clock and a 4-channel controller.
- iSBC 547 or iSBC 548 communication board--manages an 8-channel controller.

There are three screens that define the interface between the Terminal Communications Controller driver and the I/O system. These screens relate to the three device configuration tables: the device information table, the unit information table, and the device unit information block (DUIB). Refer to the *Extended iRMX II Device Drivers User's Guide* for further information about these tables.

The values shown on the screens in this section are the values you see when you invoke the ICU using the 28612.DEF file.

CAUTION

Application programs that need to know when transmission errors occur may no longer function properly with the Terminal Communications Controller board. Whenever the Terminal Communications Controller firmware receives a character with a parity error, the firmware discards the character. If the firmware receives a character with a framing error, the firmware replaces the character with an eight bit null character (00H). The Terminal Communications Controller firmware does not inform the device driver that it performed these actions.

Terminal Communications Controller Driver Screen

The ICU uses the information from the following screen to create a device information table for the Terminal Communications Controller driver. If your system includes more than one Terminal Communications Controller, you must specify a unique interrupt level and memory address base for each controller.

```
(D8848)  Terminal Communications Controller Driver

(DEV) Device Name [1-16 Chars]                        18848_A
(IL)  Interrupt Level [Encoded Level]                 038H
(MA)  Memory Address Base [0-0FFFFFFH]                0F94000H
(PA)  Port Address  [0-0FFFFH]                        08A6H

Enter [ Abbreviation = new_value / Abbreviation ? / H ]
:
```

```
   (DEV) Device Name [1-16 Chars]                      18848_A
```

This parameter provides a means of associating the device and all the units and DUIBs which belong to it. The ICU stores all the screens by type, not by device. This means that all device driver screens are stored together, all unit information screens and all DUIB screens. The ICU "knows" which unit information and device-unit information screen relate to a particular device driver by the device name you enter in the "DEV" parameter.

```
   (IL)  Interrupt Level [Encoded Level]               038H
```

This parameter specifies the encoded interrupt level for the Terminal Communications Controller driver. The interrupt task uses this value to associate itself with the correct interrupt level. The default value 038H (0000 0000 0011 1000 binary) specifies master interrupt level 3.

The possible values for this field are encoded as follows (where bit 0 is the low-order bit):

Master Level	Code	Slave Level	Attached Code	Master Level	Code	Slave Level	Attached Code
0	0008H	0-7	0000-0007H	4	0048H	0-7	0040-0047H
1	0018H	0-7	0010-0017H	5	0058H	0-7	0050-0057H
2	0028H	0-7	0020-0027H	6	0068H	0-7	0060-0067H
3	0038H	0-7	0030-0037H	7	0078H	0-7	0070-0077H

```
   (MA)  Memory Address Base [0-0FFFFFFH]              0F94000H
```

The Memory Address Base is the 24-bit address of the dual port RAM that matches the jumper configuration of the Terminal Communication Controller.

When you are configuring the jumpers on the board, be aware that the firmware on the communications controller board needs at least 48K-bytes of MULTIBUS dual port RAM.

For the 188/48/56 boards, add 4000H to the jumper-specified value. For example, F94000H when the board is jumpered for F90000H.

```
    (PA)  Port Address [0-0FFFFH]                        08A6H
```

This parameter specifies the I/O wakeup address of the Terminal Communications Controller. This value must match the appropriate jumpers on your Terminal Communications Controller board. Refer to the *iSBC 188/48 Advanced Communicating Computer Hardware Reference Manual, iSBC 188/56 Advanced Communications Computer Hardware Reference Manual*, or the *iSBC 546/547/548 High Performance Terminal Controller Hardware Reference Manual* for more information. Intel recommends that you do not change the default value (08A6H).

The Terminal Communications Controller board features a two-part access arrangement in which a bus master board can access the on-board dynamic RAM via the MULTIBUS system bus.

Query Screen

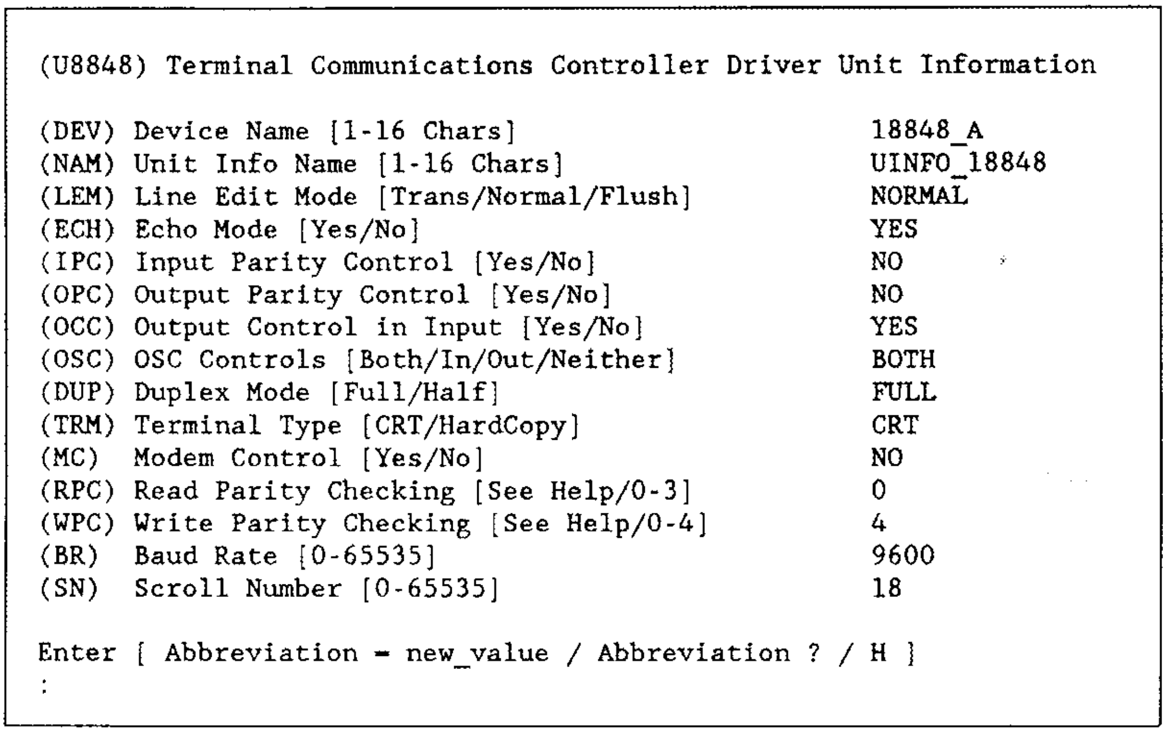
After you have completed the "Terminal Communications Controller Driver" screen, the query screen is displayed. It contains only one line:

```
   Do you want any/more Terminal Communications Controller DEVICEs?
```

Respond to this prompt with a "Yes" if you want to add more device drivers of this type.

Terminal Communications Controller Unit Information Screen

The ICU uses the information from the following screen to create a unit information table for the Terminal Communications Controller driver.



```
(U8848) Terminal Communications Controller Driver Unit Information

(DEV) Device Name [1-16 Chars]                          18848_A
(NAM) Unit Info Name [1-16 Chars]                       UINFO_18848
(LEM) Line Edit Mode [Trans/Normal/Flush]               NORMAL
(ECH) Echo Mode [Yes/No]                                YES
(IPC) Input Parity Control [Yes/No]                     NO
(OPC) Output Parity Control [Yes/No]                    NO
(OCC) Output Control in Input [Yes/No]                  YES
(OSC) OSC Controls [Both/In/Out/Neither]                BOTH
(DUP) Duplex Mode [Full/Half]                           FULL
(TRM) Terminal Type [CRT/HardCopy]                      CRT
(MC)  Modem Control [Yes/No]                            NO
(RPC) Read Parity Checking [See Help/0-3]               0
(WPC) Write Parity Checking [See Help/0-4]              4
(BR)  Baud Rate [0-65535]                               9600
(SN)  Scroll Number [0-65535]                           18

Enter [ Abbreviation = new_value / Abbreviation ? / H ]
:
```

```
  (DEV)  Device Name [1-16 Chars]                       18848_A
```

The name you enter for this parameter must be the same name you entered in the "DEV" parameter on the driver screen. This name provides the logical ICU connection between a driver and all of its units. Thus, when a device is deleted it is possible to delete all the related units at the same time.

```
  (NAM) Unit Info Name [1-16 Chars]                     UNIFO_18848
```

This parameter specifies a unique name for this unit information table. The first character must be an alphabetic character. Refer to the *ASM286 Language Reference Manual* for rules regarding this name.

A DUIB uses the "Unit Info Name" to point to this particular unit information table. When developing your initial systems, you can create unit information tables that are never pointed to by a DUIB. There is no harm in this process during the development stages. However, when you configure your final system, eliminate all unused unit information tables to save memory.

```
    (LEM) Line Edit Mode [Trans/Normal/Flush]          NORMAL
```

This parameter specifies the initial default line editing mode. You must choose from the following three options:

Transparent
: Console input is transparent (not line-edited). The Terminal Support Code transmits input to the requesting task exactly as entered at the terminal. Before being transmitted, the Terminal Support Code accumulates data in a buffer until an operator enters the requested number of characters.

Normal
: Console input is line-edited. Edited data accumulates in a buffer until an operator enters a carriage return.

Flush
: Console input is not line-edited and the Terminal Support Code transmits input to the requesting task exactly as entered at the terminal. Before being transmitted, the Terminal Support Code accumulates data in a buffer until it receives an input request. At that time, it transmits the contents of the buffer (or the number of characters requested, if the buffer contains more than that number) to the requesting task. If any characters remain in the buffer, the Terminal Support Code saves the characters for the next input request.

You can alter the value you specify for this parameter at run time if you invoke any of the SPECIAL system calls or send an OSC sequence. OSC sequences are described in the *Extended iRMX II Device Drivers User's Guide*.

```
    (ECH) Echo Mode [Yes/No]                           YES
```

Specify "Yes" if you want characters entered into the terminal to be "echoed" to the terminal's display screen. Otherwise, specify "No".

You can alter the value you specify for this parameter at run time if you invoke any of the SPECIAL system calls or send an OSC sequence. OSC sequences are described in the *Extended iRMX II Device Drivers User's Guide*.

```
    (IPC) Input Parity Control [Yes/No]                NO
```

Specify "Yes" if you want the system to change the parity bit (bit 7) of characters entered into the terminal to zero. Specify "No" if you do not want the system to change bit 7 of the input character.

You can alter the value you specify for this parameter at run time if you invoke any of the SPECIAL system calls or send an OSC sequence. OSC sequences are described in the *Extended iRMX II Device Drivers User's Guide.*

```
    (OPC) Output Parity Control [Yes/No]               NO
```

Specify "Yes" if you want the system to change the parity bit (bit 7) of characters being output to the terminal to zero. Specify "No" if you want bit 7 in the output characters to remain unchanged.

You can alter the value you specify for this parameter at run time if you invoke any of the SPECIAL system calls or send an OSC sequence. OSC sequences are described in the *Extended iRMX II Device Drivers User's Guide.*

```
    (OCC) Output Control in Input [Yes/No]             YES
```

Specify "Yes" if you want the Terminal Support Code to accept output control characters in the input stream. Specify "No" if you want the Terminal Support Code to ignore output control characters. Control characters are described in the *Extended iRMX II Device Drivers User's Guide.*

You can alter the value you specify for this parameter at run time if you invoke any of the SPECIAL system calls or send an OSC sequence. OSC sequences are described in the *Extended iRMX II Device Drivers User's Guide.*

```
    (OSC) OSC Controls [Both/In/Out/Neither]           BOTH
```

This parameter specifies whether the device driver should act upon Operating System Command (OSC) controls when they appear in either an input or an output stream. Choose one of the following options as the initial default value for the device driver:

Both — Act upon OSC control sequences in either input or output stream (from either terminal or program).

Input — Act upon OSC control sequences in input stream only (from terminal and not from program).

Output — Act upon OSC control sequences in output stream only (from program and not from terminal).

Neither	Do not act upon OSC control sequences.

The OSC control sequence, used in communicating from a program or a terminal to an operating system, is described in the *Extended iRMX II Device Drivers User's Guide*. You can alter the value you specify for this parameter at run time if you invoke any of the SPECIAL system calls or send an OSC sequence.

```
    (DUP) Duplex Mode [Full/Half]                         FULL
```

This parameter specifies the line protocol mode for a terminal. Choose either full-duplex or half-duplex.

When line protocol mode is full-duplex, the terminal driver concurrently handles input to and output from the terminal. If you specified both echo mode and full-duplex, the terminal driver echoes each character. When the line protocol mode is half-duplex, there can be input to and output from the terminal, but not concurrently. If you specify both echo mode and half-duplex, the terminal, not the terminal driver, must echo each character.

You can alter the value you specify for this parameter at run time if you invoke any of the SPECIAL system calls or send an OSC sequence. OSC sequences are described in the *Extended iRMX II Device Drivers User's Guide.*

```
    (TRM) Terminal Type [CRT/HardCopy]                    CRT
```

This parameter specifies how your terminal supports the rubout function. Respond "CRT" if your terminal can backspace and leave a blank character on the screen for each character "rubbed out." Respond "Hard Copy" if your terminal cannot backspace and leave a blank character on previously-displayed characters.

You can alter the value you specify for this parameter at run time if you invoke any of the SPECIAL system calls or send an OSC sequence. OSC sequences are described in the *Extended iRMX II Device Drivers User's Guide.*

```
    (MC)  Modem Control [Yes/No]                          NO
```

Specify "Yes" to establish an initial modem-based link between a task and a terminal. Specify "No" if your terminal is not connected to a modem.

If you specify "Yes" to this parameter, set the "OSC Controls" parameter to either "Both" or "Out".

You can alter the value you specify for this parameter at run time if you invoke any of the SPECIAL system calls or send an OSC sequence.

```
    (RPC) Read Parity Checking [See Help/0-3]         0
```

This parameter specifies how the hardware in your system will handle read parity checking. A response of 0 or 1 specifies 8-bit data with no parity checking. A response of 2 or 3 specifies 7-bit data with parity checking. In this section, "parity bit" refers to bit seven. Specify one of the following four values:

0 Ignore parity checking and set the input parity bit (bit 7) to zero.

1 Ignore parity checking and do not change the parity bit.

2 The driver expects even parity on input and sets the parity bit to 0 on meeting this condition.

The driver sets the parity bit is to 1 if odd parity is received, a framing error occurs (receive stop bit is zero), or an overrun error is encountered (a new character has been received before the interrupt routine for character processing has completed).

3 Driver expects odd parity. If a parity error, a framing error, or an overrun error is encountered, the parity bit is set to 1. Otherwise, the parity bit is set to zero.

Note that because this parameter specifies how the hardware deals with parity, a response of zero here negates a "No" response to the software parity checking parameter "(IPC) Input Parity Checking".

If you set "(RPC) Read Parity Checking" to two or three and you set "(IPC) Input Parity Checking" to "Yes", your application will not be able to detect a transmission error.

You can alter the value you specify for this parameter at run time if you invoke any of the SPECIAL system calls or send an OSC sequence. OSC sequences are described in the *Extended iRMX II Device Drivers User's Guide*.

```
    (WPC) Write Parity Checking [See Help/0-4]        4
```

This parameter specifies how the hardware in your system will handle write parity checking. A response of 0, 1, or 4 specifies 8-bit data with no parity checking. A response of 2 or 3 specifies 7-bit data with parity checking. In this section, "parity bit" refers to bit seven. Specify one of the following five values:

0 Set the output parity bit to zero.

1 Set the output parity bit to one.

2 Set the output parity bit to one if the total number of 1's in the character is odd. Set the parity bit to zero if the total number of 1's is even (even parity). This option should be used if the driver is using even parity checking for input.

3 Set the output parity bit to 0 if the total number of 1's in the character is odd (odd parity). Set the parity bit to 1 if the total number of 1's is even. This option should be used if the driver is using odd parity checking for input.

4 Do not change the output parity bit.

Note that because this parameter specifies how the hardware deals with parity, a response of zero here negates a "No" response to the software parity checking parameter "(OPC) Output Parity Checking".

You can alter the value you specify for this parameter at run time if you invoke any of the SPECIAL system calls or send an OSC sequence. OSC sequences are described in the *Extended iRMX II Device Drivers User's Guide.*

```
    (BR)  Baud Rate [0-65535]                            9600
```

This parameter specifies the initial baud rate of this terminal. Specify a value of one if you want the controller to ascertain the initial baud rate automatically. The default value is 9600 decimal.

You can alter the value you specify for this parameter at run time if you invoke any of the SPECIAL system calls or send an OSC sequence. OSC sequences are described in the *Extended iRMX II Device Drivers User's Guide.*

```
    (SN)  Scroll Number [0-65535]                        8
```

This parameter specifies the number of lines to scroll when an operator enters the scrolling output control character (CONTROL-W is the default). Typical values should be from 10 to 24. The default value is 18 decimal.

You can alter the value you specify for this parameter at run time if you invoke any of the SPECIAL system calls or send an OSC sequence. OSC sequences are described in the *Extended iRMX II Device Drivers User's Guide.*

Query Screen

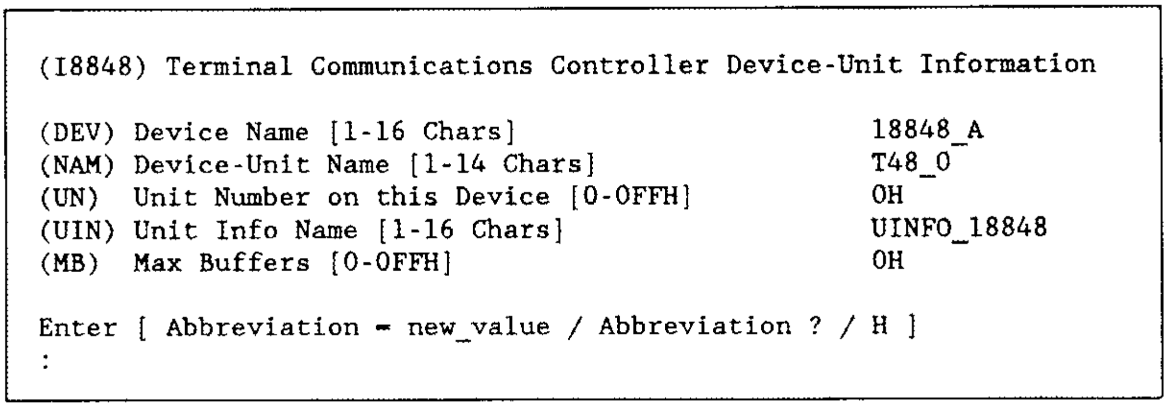
After you have completed the "Terminal Communications Controller Unit Information" screen, the query screen is displayed. It contains only one line:

```
    Do you want any/more Terminal Communications Controller UNITs ?
```

Respond "Yes" to this prompt if you need to define another unit information table for this device. Otherwise, respond with a "No".

Terminal Communications Controller Device Unit Information Screen

The ICU uses the information from the following screen to create a device unit information block (DUIB) for the Terminal Communications Controller driver.



```
(I8848) Terminal Communications Controller Device-Unit Information

(DEV) Device Name [1-16 Chars]                    18848_A
(NAM) Device-Unit Name [1-14 Chars]               T48_0
(UN)  Unit Number on this Device [0-0FFH]         0H
(UIN) Unit Info Name [1-16 Chars]                 UINFO_18848
(MB)  Max Buffers [0-0FFH]                        0H

Enter [ Abbreviation = new_value / Abbreviation ? / H ]
:
```

```
(DEV)  Device Name [1-16 Chars]                 18848_A
```

The name you enter for this parameter must be the same name you entered in the "DEV" parameter on the driver screen. This name provides the logical ICU connection between the driver and all of its DUIBs. Thus, when a device is deleted it is possible to delete all the related units and DUIBs at the same time.

```
(NAM) Device-Unit Name [1-14 Chars]             T48_0
```

This parameter specifies a name that uniquely identifies the device-unit for the I/O System. The name you specify for this parameter is the physical name you specify when invoking the Human Interface ATTACHDEVICE command, the BIOS A$PHYSICAL$ATTACH$DEVICE system call, or the EIOS LOGICAL$ATTACH$DEVICE system call.

The ICU allows you to enter from 1 to 14 characters. Refer to the *ASM286 Language Reference Manual* for rules regarding this name.

```
(UN)  Unit Number on this Device [0-0FFH]       0H
```

This parameter specifies the unit number of this device-unit.

If your Terminal Communications Controller consists of the iSBC 188/48 and iSBC 188/56, unit numbers zero to seven (0-7) specify the device-unit as on-board the Terminal Communications Controller. Unit numbers eight and nine (8-9) specify channels A and B respectively, of an iSBX 354 MULTIMODULE board when it is installed in socket 1. Unit numbers 10 and 11 are on an iSBX 354 MULTIMODULE board installed in iSBX socket 2.

If your Terminal Communications Controller consists of the iSBC 546/547, unit numbers zero to three (0-3) specify the device-unit as on-board the Terminal Communications Controller. Unit number 4 specifies the line printer channel and unit number 5 specifies the Global Clock.

```
    (UIN) Unit Info Name [1-16 Chars]                    UINFO_18848
```

This parameter specifies the "Unit Info Name" of a unit information table that fills the needs of this DUIB. The following rules apply to the name you choose:

- You must create a unit information table with this "unit info name".
- Each DUIB can point to only one unit information table.
- A particular unit information table can be pointed to by more than one DUIB.

```
    (MB)  Max Buffers [0-0FFH]                           0H
```

This parameter specifies the maximum number of buffers that the Extended I/O System can allocate for this device's I/O. Unless you are planning on using a terminal for output only, do not change the default value. For terminals used for output only, you may want to specify a small non-zero value. Note, however, that the EIOS will delay output until the buffer is full or you close the connection to the device.

Query Screen

After you have completed the "Terminal Communications Controller DUIB Information" screen, the query screen is displayed. It contains only one line:

```
    Do you want any/more Terminal Communications Controller DUIBs ?
```

If you plan to use the Terminal Communications Controller with two terminals that have different characteristics, respond "Yes".

While developing your initial systems, you can create as many device-unit information blocks as you want. The number of DUIBs can exceed the number of devices on your system. The particular DUIB associated with the device depends on the physical name you use when attaching it. Once you know that you will never need a particular DUIB, save memory by deleting it from your definition file before you generate your configuration files.

iSBC® 286/10 Line Printer Driver

The line printer driver for the iSBC 286/10(A) and iSBC 286/12 boards interface the Basic I/O System physical file driver to the 8255 parallel I/O port. Two screens define the interface between this common device driver and the I/O system. These screens relate to two device configuration tables: the device information table and the device unit information block (DUIB). Refer to the *Extended iRMX II Device Drivers User's Guide* for further information about these tables.

The values shown on the screens in this section are the values you see when you invoke the ICU with the 28612.DEF file.

iSBC® 286/10(A) Line Printer Driver Screen

The ICU uses the information from the following screen to create a device information table for the line printer driver. If your system includes more than one line printer controller, you must specify unique port addresses and a unique interrupt level for each controller.

```
(D286)    Line Printer for iSBC 286/10

(DEV) Device Name [1-16 Chars]                         LP286
(IL)  Interrupt Level [Encoded Level]                  077H
(ITP) Interrupt Task Priority [0-255]                  130
(POA) 8255A Port A Address [0-0FFFFH]                  0C8H
(POB) 8255A Port B Address [0-0FFFFH]                  0CAH
(POC) 8255A Port C Address [0-0FFFFH]                  0CCH
(CON) 8255A Control Port Address [0-0FFFFH]            0CEH
(TAB) Printer Expands Tabs [Yes/No]                    YES
(ITO) Interrupt Time Out [0-0FFFFH]                    0200H

Enter [ Abbreviation = new_value / Abbreviation ? / H ]
:
```

```
   (DEV) Device Name [1-16 Chars]                   LP286
```

This parameter provides a means of associating the device and all the units and DUIBs which belong to it. The ICU stores all the screens by type, not by device. This means that all device driver screens are stored together, all unit information screens and all DUIB screens. The ICU "knows" which unit information and device-unit information screen relate to a particular device driver by the device name you enter in the "DEV" parameter.

```
    (IL)  Interrupt Level [Encoded Level]             077H
```

This parameter specifies the encoded interrupt level for the line printer driver. The interrupt task uses this value to associate the interrupt task with the correct interrupt level. The default value 077H (0000 0000 0111 0111 binary) specifies slave interrupt level 7. Intel recommends that you do not change the default value.

The possible values for this field are encoded as follows (where bit 0 is the low-order bit):

Master Level	Code	Slave Level	Attached Code	Master Level	Code	Slave Level	Attached Code
0	0008H	0-7	0000-0007H	4	0048H	0-7	0040-0047H
1	0018H	0-7	0010-0017H	5	0058H	0-7	0050-0057H
2	0028H	0-7	0020-0027H	6	0068H	0-7	0060-0067H
3	0038H	0-7	0030-0037H	7	0078H	0-7	0070-0077H

```
    (ITP) Interrupt Task Priority [0-255]             130
```

This parameter specifies the initial priority of the device's interrupt task. The default value is 130 decimal. The actual priority of the interrupt task changes because the Nucleus adjusts an interrupt task's priority according to the interrupt level that it services.

```
     (POA) 8255A Port A Address [0-0FFFFH]            0C8H
     (POB) 8255A Port B Address [0-0FFFFH]            0CAH
     (POC) 8255A Port C Address [0-0FFFFH]            0CCH
     (CON) 8255A Control Port Address [0-0FFFFH]      0CEH
```

These parameter lines specify the addresses of the 8255A Programmable Interface ports on your iSBC 286/10(A) or iSBC 286/12 board. These addresses must match the jumper setting on the iSBC 286-based board. Intel recommends that you do not change the factory-configured jumper settings.

```
    (TAB) Printer Expands Tabs [Yes/No]               YES
```

Specify "Yes" if you want the driver to force all sequential tabs into a single blank character. If you want the driver to pass tab characters unchanged to the line printer, specify "No".

```
    (ITO) Interrupt Time Out [0-0FFFFH]                 0200H
```

This parameter specifies the length of time (in system clock ticks) the system should wait for an interrupt from the line printer before resuming control. If, for some reason the line printer is not connected, and there is a value in this parameter, the system waits to receive an interrupt for the indicated time before it resumes control.

A value of 0FFFFH for this parameter, causes the system to wait indefinitely. If the system does not receive a response from the line printer and no value has been entered for this parameter, the system may "hang". The default value of 200H indicates the system will wait 512 (decimal) clock ticks, each of which is 10 milliseconds.

Query Screen

After you have completed the iSBC 286/10(A) driver screen, the query screen is displayed. It contains only one line:

```
    Do you want any/more Line Printer for iSBC 286/10 DEVICEs ?
```

Respond to this prompt with a "Yes" if you want to add more device drivers of this type.

iSBC® 286/10(A) Line Printer Device-Unit Information Screen

The ICU uses the information from the following screen to create a device unit information block (DUIB) for the iSBC 286/10(A) line printer driver.

```
(I286) Line Printer for iSBC 286/10 Device-Unit Information

(DEV) Device Name [1-16 Chars]                       LP286
(NAM) Device-Unit Name [1-14 Chars]                  LP
(MB)  Max Buffers [0-0FFH]                           0H

Enter [ Abbreviation = new_value / Abbreviation ? / H ]
:
```

```
    (DEV)  Device Name [1-16 Chars]                   LP286
```

The name you enter for this parameter must be the same name you entered in the "DEV" parameter on the iSBC 286/10(A) driver screen. This name provides the logical ICU connection between the driver and all of its DUIBs. Thus, when a device is deleted it is possible to delete all the related units and DUIBs at the same time.

```
    (NAM) Device-Unit Name [1-14 Chars]                LP
```

This parameter specifies a name that uniquely identifies this device-unit to the I/O System. The name you specify for this parameter is the physical name you specify when invoking the Human Interface ATTACHDEVICE command, the BIOS A$PHYSICAL$ATTACH$DEVICE system call, or the EIOS LOGICAL$ATTACH$DEVICE system call.

The ICU allows you to enter from one to fourteen characters. Refer to the *ASM286 Language Reference Manual* for rules regarding this name.

```
    (MB)  Max Buffers [0-0FFH]                          0H
```

This parameter specifies the maximum number of buffers that the Extended I/O System can allocate for this device's I/O. You may want to specify a small non-zero value for this parameter. Note, however, that the EIOS will delay output until the buffer is full or you close the connection to the device.

Query Screen

After you have completed the "Line Printer for iSBC 286/10(A) Device Unit Information" screen, the query screen is displayed. It contains only one line:

```
    Do you want any/more Line Printer for iSBC 286/10 DUIBs ?
```

Respond "Yes" to this prompt if you plan to use the iSBC 286/10(A) Line Printer Driver with two line printers that have different characteristics.

While developing your initial systems, you can create as many device-unit information blocks as you want. The number of DUIBs can exceed the number of devices on your system. The particular DUIB associated with the device depends on the physical name you use when attaching it. Once you know that you will never need a particular DUIB, save memory by deleting it from your definition file before you generate your configuration files.

Line Printer--iSBX™ 350

This line printer driver interfaces the Basic I/O System physical file driver to the iSBX 350 MULTIMODULE. Two screens define the interface between this common device driver and the I/O system. These screens relate to two device configuration tables: the device information table and the device unit information block (DUIB). Refer to the *Extended iRMX II Device Drivers User's Guide* for further information about these tables.

The values shown on the screens in this section are the values you see when you invoke the ICU using the 28612.DEF file.

Line Printer--iSBX™ 350 Driver Screen

The ICU uses the information from the following screen to create a device information table for the line printer driver. If your system includes more than one iSBX 350 controller, you must specify unique port addresses and a unique interrupt level for each controller.

```
(D350)        Line Printer - iSBX 350

(DEV)  Device Name [1-16 Chars]
(IL)  Interrupt Level [Encoded Level]                  073H
(ITP) Interrupt Task Priority [0-255]                  130
(POA) 8255A Port A Address [0-0FFFFH]                  080H
(POB) 8255A Port B Address [0-0FFFFH]                  082H
(POC) 8255A Port C Address [0-0FFFFH]                  084H
(CON) 8255A Control Port Address [0-0FFFFH]            086H
(TAB) Printer Expands Tabs [Yes/No]                    YES
(ITO) Interrupt Time Out [0-0FFFFH]                    0200H

Enter [ Abbreviation = new_value / Abbreviation ? / H ]
:
```

```
   (DEV) Device Name [1-16 Chars]
```

This parameter provides a means of associating the device and all the units and DUIBs which belong to it. The ICU stores all the screens by type, not by device. This means that all device driver screens are stored together, all unit information screens and all DUIB screens. The ICU "knows" which unit information and device-unit information screen relate to a particular device driver by the device name you enter in the "DEV" parameter.

```
    (IL)  Interrupt Level [Encoded Level]          073H
```

This parameter specifies the encoded interrupt level for the iSBX 350 driver. The interrupt task uses this value to associate the interrupt task with the correct interrupt level. The default value 073H (0000 0000 0111 0011 binary) specifies slave interrupt level 3.

The possible values for this field are encoded as follows (where bit 0 is the low-order bit):

Master Level	Code	Slave Level	Attached Code	Master Level	Code	Slave Level	Attached Code
0	0008H	0-7	0000-0007H	4	0048H	0-7	0040-0047H
1	0018H	0-7	0010-0017H	5	0058H	0-7	0050-0057H
2	0028H	0-7	0020-0027H	6	0068H	0-7	0060-0067H
3	0038H	0-7	0030-0037H	7	0078H	0-7	0070-0077H

```
    (ITP) Interrupt Task Priority [0-255]          130
```

This parameter specifies the initial priority of the device's interrupt task. The default value is 130 decimal. The actual priority of the interrupt task changes because the Nucleus adjusts an interrupt task's priority according to the interrupt level that it services.

```
    (POA) 8255A Port A Address [0-0FFFFH]           080H
    (POB) 8255A Port B Address [0-0FFFFH]           082H
    (POC) 8255A Port C Address [0-0FFFFH]           084H
    (CON) 8255A Control Port Address [0-0FFFFH]     086H
```

These parameter lines specify the addresses of the 8255A Programmable Interface ports on your 80286/80386-based board. These addresses are determined by the iSBX socket that the module is placed in. Intel recommends that you do not change the factory-configured jumper settings.

```
    (TAB) Printer Expands Tabs [Yes/No]            YES
```

Specify "Yes" if you want the driver to force all sequential tabs into a single blank character. If you want the driver to pass tab characters unchanged to the iSBX 350, specify "No".

```
    (ITO) Interrupt Time Out [0-0FFFFH]                 0200H
```

This parameter specifies the length of time (in system clock ticks) the system should wait for an interrupt from the line printer before resuming control. If, for some reason the line printer is not connected, and there is a value in this parameter, the system waits to receive an interrupt for the indicated time before it resumes control.

A value of 0FFFFH for this parameter, causes the system to wait indefinitely. If the system does not receive a response from the line printer and no value has been entered for this parameter, the system may "hang". The default value of 200H indicates the system will wait 512 (decimal) clock ticks each of which is 10 milliseconds.

Query Screen

After you have completed the driver screen, the query screen is displayed It contains only one line:

```
    Do you want any/more Line Printer - iSBX 350 DEVICEs ?
```

Respond to this prompt with a "Yes" if you want to add more device drivers of this type.

Line Printer--iSBX™ 350 Driver Device Unit Information Screen

The ICU uses the information from the following screen to create a device unit information block (DUIB) for the iSBX 350 driver.

```
(I350)  Line Printer - iSBX 350 Device-Unit Information

(DEV) Device Name [1-16 Chars]
(NAM) Device-Unit Name [1-14 Chars]
(MB)  Max Buffers [0-0FFH]                        0H

Enter [ Abbreviation = new_value / Abbreviation ? / H ]
:
```

```
    (DEV)  Device Name [1-16 Chars]
```

The name you enter for this parameter must be the same name you entered in the "DEV" parameter on the "iSBX 350 Driver" screen. This name provides the logical ICU connection between the driver and all of its DUIBs. Thus, when a device is deleted it is possible to delete all the related units and DUIBs at the same time.

```
    (NAM) Device-Unit Name [1-14 Chars]
```

This parameter specifies a name that uniquely identifies this device-unit to the I/O System. The name you specify for this parameter is the physical name you specify when invoking the Human Interface ATTACHDEVICE command, the BIOS A$PHYSICAL$ATTACH$DEVICE system call, or the EIOS LOGICAL$ATTACH$DEVICE system call.

The ICU allows you to enter from one to fourteen characters. Refer to the *ASM286 Language Reference Manual* for rules regarding this name.

```
    (MB)  Max Buffers [0-0FFH]                         0H
```

This parameter specifies the maximum number of buffers that the Extended I/O System can allocate for this device's I/O.

Query Screen

After you have completed the DUIB Information screen, the query screen is displayed. It contains only one line:

```
    Do you want any/more Line Printer - iSBX 350 DUIBs ?
```

Respond "Yes" to this prompt if you plan to use the iSBX 350 driver with a printer that has different characteristics.

iSBC® 264 Driver Parameters

The iSBC 264 driver supports the iSBC 264 magnetic bubble memory board. Each board supports up to 2M bytes of memory. The iSBC 264 boards can be configured in three different ways:

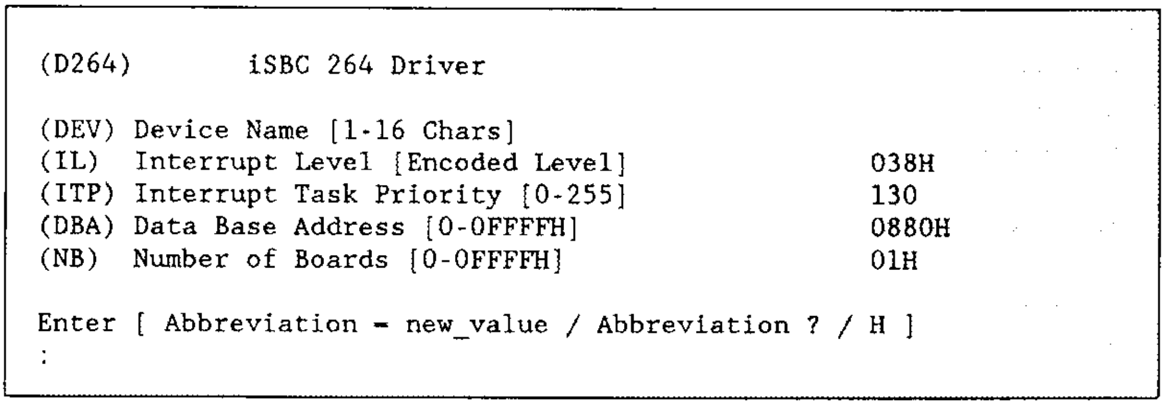
- Configure each iSBC 264 driver as a complete device with one unit.
- Configure several iSBC 264 boards as though they are units of one imaginary device.
- Configure several iSBC 264 boards as though they are one unit of a device.

It is also possible to combine these methods. For more information on how to do this, refer to the *Extended iRMX II Device Drivers User's Guide*.

Three screens define the interface between the iSBC 264 random access device driver and the I/O system. These screens relate to the three device configuration tables: the device information table, the unit information table, and the device unit information block (DUIB).

iSBC® 264 Driver Screen

The ICU uses the information from the following screen to create a device information table for the iSBC 264 driver. If your system includes more than one iSBC 264 controller, you must specify a unique interrupt level and port addresses for each controller. There is one exception to needing a unique interrupt level for each board. If you configure several boards as a single unit, only one interrupt level is needed.



```
(D264)       iSBC 264 Driver

(DEV) Device Name [1-16 Chars]
(IL)  Interrupt Level [Encoded Level]          038H
(ITP) Interrupt Task Priority [0-255]          130
(DBA) Data Base Address [0-0FFFFH]             0880H
(NB)  Number of Boards [0-0FFFFH]              01H

Enter [ Abbreviation = new_value / Abbreviation ? / H ]
:
    (DEV) Device Name [1-16 Chars]
```

This parameter provides a means of associating the device and all the units and DUIBs which belong to it. The ICU stores all the screens by type, not by device. This means that all device driver screens are stored together, all unit information screens and all DUIB screens. The ICU "knows" which unit information and device-unit information screen relate to a particular device driver by the device name you enter in the "DEV" parameter.

```
    (IL)  Interrupt Level [Encoded Level]           038H
```

This parameter specifies the encoded interrupt level for the iSBC 264A driver. The interrupt task uses this value to associate the interrupt task with the correct interrupt level. The default value 038H (0000 0000 0011 1000 binary) specifies master interrupt level 3.

The possible values for this field are encoded as follows (where bit 0 is the low-order bit):

Master Level	Code	Slave Level	Attached Code	Master Level	Code	Slave Level	Attached Code
0	0008H	0-7	0000-0007H	4	0048H	0-7	0040-0047H
1	0018H	0-7	0010-0017H	5	0058H	0-7	0050-0057H
2	0028H	0-7	0020-0027H	6	0068H	0-7	0060-0067H
3	0038H	0-7	0030-0037H	7	0078H	0-7	0070-0077H

```
    (ITP) Interrupt Task Priority [0-255]           130
```

This parameter specifies the initial priority of the device's interrupt task. The default value is 130 decimal. The actual priority of the interrupt task changes because the Nucleus adjusts an interrupt task's priority according to the interrupt level that it services.

```
    (DBA) Device Base Address [0-0FFFFH]            0880H
```

This parameter specifies the lowest hardware I/O address of the iSBC 264 boards configured for this device. All other iSBC 264 boards that are to be part of this device must have consecutive I/O addresses of 20H greater than the preceding address.

```
    (NB) Number of Boards [0-0FFFFH]                01H
```

This parameter specifies the number of iSBC 264 boards that share the same interrupt level.

Query Screen

After you have completed the "iSBC 264 Driver" screen, the query screen is displayed. It contains only one line:

```
   Do you want any/more iSBC 264 DEVICEs ?
```

Respond to this prompt with a "Yes" if you want to add more device drivers of this type.

iSBC® 264 Unit Information Screen

The ICU uses the information from the following screen to create a unit information table for the iSBC 264 driver.

```
  (U264)    iSBC 264 Unit Information

  (DEV) Device Name [1-16 Chars]
  (NAM) Unit Info Name [1-16 Chars]
  (MR)  Maximum Retries [0-0FFFFH]                 09H
  (NB)  Number of Boards [0-0FFFFH]                01H
  (UBA) Unit Base Address [0-0FFFFH]               0880H
  (BS)  Board Size [0-0FFFFH]                      02000H

   Enter [ Abbreviation = new_value / Abbreviation ? / H ]
   :
```

```
   (DEV)  Device Name [1-16 Chars]
```

The name you enter for this parameter must be the same name you entered in the "DEV" parameter on the driver screen. This name provides the logical ICU connection between the driver and all of its units. Thus, when a device is deleted it is possible to delete all the related units at the same time.

```
   (NAM) Unit Info Name [1-16 Chars]
```

This parameter specifies a unique name for this unit information table. The first character must be an alphabetic character. Refer to the *ASM286 Language Reference Manual* for rules regarding this name.

A DUIB uses the "Unit Info Name" to point to this particular unit information table. When developing your initial systems, you can create unit information tables that are never pointed to by a DUIB. There is no harm in this process during the development stages. However, when you configure your final system, eliminate all unused unit information tables to save memory.

```
    (MR)  Maximum Retries [0-0FFFFH]                  09H
```

This parameter specifies the maximum number of times the Basic I/O System should retry an operation before returning an E$IO$SOFT exception code. Refer to the *Extended iRMX II Basic I/O System Calls Reference Manual* for related information on the E$IO$SOFT exception condition. Refer to the *iSBC 264 Bubble Memory Controller Hardware Reference Manual* for information on conditions that can cause E$IO$SOFT errors (see status register 7220). The default value of 9 is recommended.

```
    (NB) Number of Boards [0-0FFFFH]                  01H
```

This parameter specifies the number of iSBC 264 boards used in this unit.

```
    (UBA) Unit Base Address [0-0FFFFH]                880H
```

This parameter specifies the base address of the first iSBC 264 board of the unit. All other iSBC 264 boards must have a consecutive base addresses of 20H greater than the preceding address.

```
    (BS) Board Size [0-0FFFFH]                        02000H
```

This parameter specifies the number of pages (a page is the same as granularity) on the iSBC 264 board. Since more than one board may be configured as one unit, the device driver needs to know the boundaries of each board. If multiple boards are configured in the unit, only the last board may be less than this parameter.

Query Screen

After you have completed the "iSBC 264 Unit Information" screen, the query screen is displayed. It contains only one line:

```
   Do you want any/more iSBC 264 UNITs ?
```

Respond to this prompt with a "Yes" if you need another unit information table for this device. Otherwise, respond to this prompt with a "No".

iSBC® 264 Device Unit Information Screen

The ICU uses the information from the following screen to create a device unit information block (DUIB) for the iSBC 264 driver.

```
(I264)    iSBC 264 Device-Unit Information

(DEV) Device Name [1-16 Chars]
(NAM) Device-Unit Name [1-14 Chars]
(PFD) Physical File Driver Required [Yes/No]                YES
(NFD) Named File Driver Required [Yes/No]                   YES
(GRA) Granularity [0-0FFFFH]                                0100H
(DSZ) Device Size [0-0FFFFFFFFH]                            0200000H
(UN)  Unit Number on this Device [0-0FFH]                   0H
(UIN) Unit Info Name [1-16 Chars]
(RUT) Request Update Timeout [0-0FFFFH]                     064H
(NB)  Number of Buffers [nonrandom = 0/rand = 1-0FFFFH]     08H
(CUP) Common Update [Yes/No]                                YES
(MB)  Max Buffers [0-0FFH]                                  0FFH

Enter [ Abbreviation = new_value / Abbreviation ? / H ]
:
```

(DEV) Device Name [1-16 Chars]

The name you enter for this parameter must be the same name you entered in the "DEV" parameter on the driver screen. This name provides the logical ICU connection between the driver and all of its DUIBs. Thus, when a device is deleted it is possible to delete all the related units and DUIBs at the same time.

(NAM) Device-Unit Name [1-14 Chars]

This parameter specifies a name that uniquely identifies this device-unit to the I/O System. If you want the Extended I/O System to logically attach this device during initialization, the name you specify for this parameter must also be one of the device names you specified on the "Logical Names" screen (see Chapter 8). To support Auto Boot device Recognition, specify the same name in one of the %DEVICE macros when you configure this device in your Bootstrap Loader (see the *Extended iRMX II Bootstrap Loader Reference Manual* for detailed information).

The ICU allows you to enter from one to fourteen characters. Refer to the *ASM286 Language Reference Manual* for rules regarding this name.

The name you specify for this parameter is the physical name you specify when invoking the Human Interface ATTACHDEVICE command, the BIOS A$PHYSICAL$ATTACH$DEVICE system call, or the EIOS LOGICAL$ATTACH$DEVICE EIOS system call.

```
    (PFD) Physical File Driver Required [Yes/No]     YES
    (NFD) Named File Driver Required [Yes/No]        YES
```

This driver supports both named and physical file drivers. These parameter lines let you specify "Yes" to at least one of these parameters. However, there are two good reasons for not changing either "Yes" default value. First, the Human Interface ATTACHDEVICE command, the EIOS LOGICAL$ATTACH$DEVICE system call, and the BIOS A$PHYSICAL$ATTACH$DEVICE system call require you to select the file type at run time. Second, there is no code savings if you specify "Yes" to one parameter and "No" to the other.

```
    (GRA) Granularity [0-0FFFFH]                     0100H
```

This parameter specifies the minimum number of bytes that the device reads or writes in one operation. This value is determined by the number of bubble devices operated in parallel as follows:

Bubbles in Parallel	Granularity
1	64 (040H)
2	128 (080H)
4	256 (100H)

The default value 0100H (256 decimal) represents 4 bubble devices per board.

The manufacturer of your drive may recommend values for this parameter. Refer to vendor documentation for additional information. Refer to the *Extended iRMX II Basic I/O System User's Guide* for more information about granularity.

```
    (DSZ) Device Size [0-0FFFFFFFFH]                 0200000H
```

This parameter specifies the device storage capacity in bytes. For the iSBC 264, this parameter should be entered as increments of 512K bytes.

```
    (UN)  Unit Number on this Device [0-0FFH]        01H
```

This parameter specifies the unit number of this device-unit. This number identifies the only possible unit on this device.

```
    (UIN) Unit Info Name [1-16 Chars]
```

This parameter specifies the "Unit Info Name" of a unit information table that fills the needs of this DUIB. The following rules apply to the name you choose:

- You must create a unit information table with this "unit info name".
- Each DUIB can point to only one unit information table.
- A particular unit information table can be pointed to by more than one DUIB.

```
    (RUT) Request Update Timeout [0-0FFFFH]          064H
```

This parameter specifies the number of clock ticks that the driver should wait (during a pause in activity) before updating the attached files on the device. The default value 064H is equivalent to 100 decimal.

The values 0 and 0FFFFH do not indicate time intervals. A value of 0 indicates that the driver will never leave any data buffered in memory (eliminating the need to update). A value of 0FFFFH indicates that updates on this device will occur only when a file is detached.

The update capability provided by this parameter differs from the common update capability described in Chapter 9. Unlike the ability to update at fixed periods that is provided by the common update feature, the update capability (referred to by the "Request Update Timeout" parameter) allows the driver to update based on pauses in activity. If there is a pause in activity on this device, the driver determines how soon the common update would occur and compares that time interval to the request update timeout value. The driver then waits the shorter of the two intervals and updates the attached files on the device. Thus, if you use the common update feature, the time interval you specify for the "Request Update Timeout" should be shorter than time interval specified for the "Common Update Timeout".

```
    (NB)  Number of Buffers [nonrandom - 0/rand - 1-0FFFFH]     08H
```

This parameter specifies the number of buffers this device has for blocking and deblocking I/O requests. The default value of 8 is appropriate for a unit that has a granularity of 1024 bytes. Since this is a device which supports random access, do not specify a value of zero.

The number of buffers prompt determines the number of Basic I/O System buffers the device uses for I/O. These buffers are general-purpose cache buffers that the Basic I/O System uses to increase I/O performance. The size of each buffer is 32 bytes greater than the device granularity you specified for this device (your response to the "(GRA) Granularity" parameter). The buffers are associated with a device-unit, not with a particular file. The Basic I/O System assigns these buffers to the device-unit when it attaches the device-unit.

Since the Operating System uses memory from the BIOS memory pool to create these buffers, you should increase the values you specified on the "BIOS" screen for both the minimum and maximum memory pool sizes if you increase the default value for the "Number of Buffers" parameter. An explanation of how to calculate memory usage based on your responses to the "(GRA) Granularity" and the "(NB) Number of Buffers" parameter lines is in Chapter 9.

Choosing an optimum number of buffers depends on many factors: how your application works; the speed of your components, your CPU, your memory, and your peripherals; the performance characteristics of your peripherals; and the number of users. Plus, you must trade off optimum performance against the amount of memory you use (because the more buffers you choose, the more memory you need).

Because there are so many factors, the only way that you can obtain the optimum value for your system is by testing different values and fine-tuning your configuration. Intel recommends that you use at least two buffers for this device.

```
    (CUP) Common Update [Yes/No]                        YES
```

Specify "Yes" if you want the driver to update the attached files on this device at the fixed interval you specified on the "BIOS" screen. Specify "No" if you do not want to update attached files on this device at fixed intervals but prefer to update attached files based only on the time interval you specified for the "Request Update Timeout" parameter.

You should not specify "Yes" for this parameter and 0FFFFH for the "Request Update Timeout" parameter. See Chapter 9 for information about common update timeout (updating attached files at fixed intervals).

```
    (MB)  Max Buffers [0-0FFH]                          0FFH
```

This parameter specifies the maximum number of buffers that the Extended I/O System can allocate for this device's I/O. Intel recommends that you use the default value.

Query Screen

After you have completed the "iSBC 264 Device-Unit Information" screen, the query screen is displayed. It contains only one line:

```
    Do you want any/more iSBC 264 DUIBs ?
```

Respond "Yes" to this prompt if you plan to use the iSBC 264 board with two bubble devices that have different characteristics.

While developing your initial systems, you can create as many device-unit information blocks as you want. The number of DUIBs can exceed the number of devices on your system. The particular DUIB associated with the device depends on the physical name you use when attaching it. Once you know that you will never need a particular DUIB, save memory by deleting it from your definition file before you generate your configuration files.

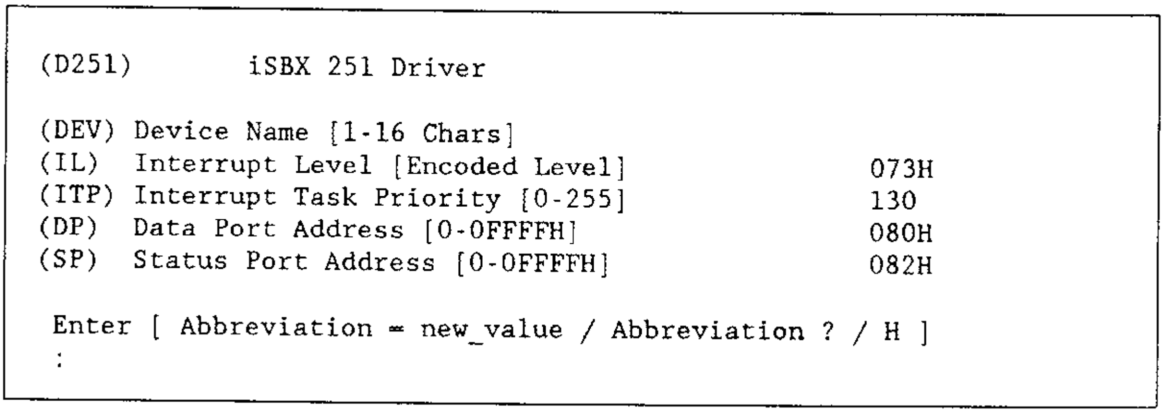
iSBX™ 251 Driver Parameters

The iSBX 251 driver supports the iSBX 251 magnetic bubble memory MULTIMODULE board. This board is supported at 6mHz only. Three screens define the interface between the iSBX 251 random access device driver and the I/O system. These screens relate to the three device configuration tables: the device information table, the unit information table, and the device unit information block (DUIB). Refer to the *Extended iRMX II Device Drivers User's Guide* for further information about these tables.

iSBX™ 251 Driver Screen

The ICU uses the information from the following screen to create a device information table for the iSBX 251 driver. If your system includes more than one iSBX 251 controller, you must specify a unique interrupt level and port addresses for each controller.

The iSBX 251 driver is not part of the Start-up system supplied by Intel. The default values displayed in the following screens are those supplied by the ICU.



```
(D251)       iSBX 251 Driver

(DEV) Device Name [1-16 Chars]
(IL)  Interrupt Level [Encoded Level]               073H
(ITP) Interrupt Task Priority [0-255]               130
(DP)  Data Port Address [0-0FFFFH]                  080H
(SP)  Status Port Address [0-0FFFFH]                082H

 Enter [ Abbreviation = new_value / Abbreviation ? / H ]
 :
```

```
(DEV) Device Name [1-16 Chars]
```

This parameter provides a means of associating the device and all the units and DUIBs which belong to it. The ICU stores all the screens by type, not by device. This means that all device driver screens are stored together, all unit information screens and all DUIB screens. The ICU "knows" which unit information and device-unit information screen relate to a particular device driver by the device name you enter in the "DEV" parameter.

```
    (IL)  Interrupt Level [Encoded Level]            073H
```

This parameter specifies the encoded interrupt level for the iSBX 251 driver. The interrupt task uses this value to associate the interrupt task with the correct interrupt level. The default value 073H (0000 0000 0111 0011 binary) specifies slave interrupt level 3.

The possible values for this field are encoded as follows (where bit 0 is the low-order bit):

Master		Slave Attached		Master		Slave Attached	
Level	Code	Level	Code	Level	Code	Level	Code
0	0008H	0-7	0000-0007H	4	0048H	0-7	0040-0047H
1	0018H	0-7	0010-0017H	5	0058H	0-7	0050-0057H
2	0028H	0-7	0020-0027H	6	0068H	0-7	0060-0067H
3	0038H	0-7	0030-0037H	7	0078H	0-7	0070-0077H

```
    (ITP) Interrupt Task Priority [0-255]            130
```

This parameter specifies the initial priority of the device's interrupt task. The default value is 130 decimal. The actual priority of the interrupt task changes because the Nucleus adjusts an interrupt task's priority according to the interrupt level that it services.

```
    (DP)  Data Port Address [0-0FFFFH]               080H
```

This parameter specifies the address of the data port. The device driver uses this data port for bidirectional data communication. The address of the data port you specify for this parameter must match the hardware configuration of your CPU board's iSBX connector.

```
    (SP)  Status Port Address [0-0FFFFH]             082H
```

This parameter specifies the address of the status port. The address of the status port you specify for this parameter must match the hardware configuration of your CPU board's iSBX connector.

The device driver uses the status port to ascertain the state of the board.

Query Screen

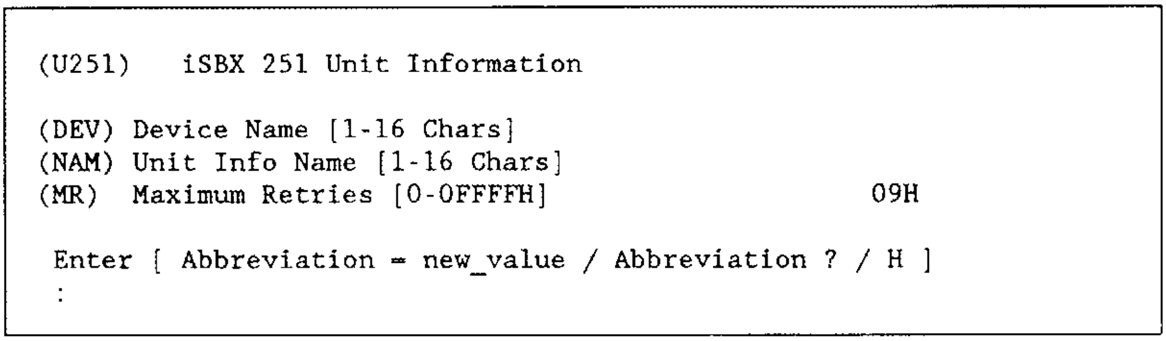
After you have completed the "iSBX 251 Driver" screen, the query screen is displayed. It contains only one line:

```
   Do you want any/more iSBX 251 DEVICEs ?
```

Respond to this prompt with a "Yes" if you want to add more device drivers of this type.

iSBX™ 251 Unit Information Screen

The ICU uses the information from the following screen to create a unit information table for the iSBX 251 driver.



```
(U251)   iSBX 251 Unit Information

(DEV) Device Name [1-16 Chars]
(NAM) Unit Info Name [1-16 Chars]
(MR)  Maximum Retries [0-0FFFFH]                       09H

 Enter [ Abbreviation = new_value / Abbreviation ? / H ]
  :
```

```
    (DEV)  Device Name [1-16 Chars]
```

This parameter provides a means of associating the device and all the units and DUIBs which belong to it. The ICU stores all the screens by type, not by device. This means that all device driver screens are stored together, all unit information screens and all DUIB screens. The ICU "knows" which unit information and device-unit information screen relate to a particular device driver by the device name you enter in the "DEV" parameter.

```
    (NAM) Unit Info Name [1-16 Chars]
```

This parameter specifies a unique name for this unit information table. The first character must be an alphabetic character. Refer to the *ASM286 Language Reference Manual* for rules regarding this name.

A DUIB uses the "Unit Info Name" to point to this particular unit information table. When developing your initial systems, you can create unit information tables that are never pointed to by a DUIB. There is no harm in this process during the development stages. However, when you configure your final system, eliminate all unused unit information tables to save memory.

```
    (MR)  Maximum Retries [0-0FFFFH]                    09H
```

This parameter specifies the maximum number of times the Basic I/O System should retry an operation before returning an E$IO$SOFT exception code. Refer to the *Extended iRMX II Basic I/O System Calls Reference Manual* for related information on the E$IO$SOFT exception condition. Refer to the *iSBX 251 Bubble Memory Controller Hardware Reference Manual* for information on conditions that can cause E$IO$SOFT errors (see status register 7220). The default value of 9 is recommended.

Query Screen

After you have completed the "iSBX 251 Driver Unit Information" screen, the query screen is displayed. It contains only one line:

```
    Do you have any/more iSBX 251 UNITs ?
```

Respond to this prompt with a "Yes" if you need another unit information table for this device. Otherwise, respond to this prompt with a "No".

iSBX™ 251 Device Unit Information Screen

The ICU uses the information from the following screen to create a device unit information block (DUIB) for the iSBX 251 driver.

```
(I251)   iSBX 251 Device-Unit Information

(DEV) Device Name [1-16 Chars]
(NAM) Device-Unit Name [1-14 Chars]
(PFD) Physical File Driver Required [Yes/No]       YES
(NFD) Named File Driver Required [Yes/No]          YES
(GRA) Granularity [0-0FFFFH]                       040H
(DSZ) Device Size [0-0FFFFFFFFH]                   020000H
(UN)  Unit Number on this Device [0-0FFH]          0H
(UIN) Unit Info Name [1-16 Chars]
(RUT) Request Update Timeout [0-0FFFFH]            064H
(NB)  Number of Buffers [ 1-0FFFFH ]               04H
(CUP) Common Update [Yes/No]                       YES
(MB)  Max Buffers [0-0FFH]                         0FFH

Enter [ Abbreviation = new_value / Abbreviation ? / H ]
:
    (DEV)  Device Name [1-16 Characters]
```

The name you enter for this parameter must be the same name you entered in the "DEV" parameter on the driver screen. This name provides the logical ICU connection between the driver and all of its DUIBs. Thus, when a device is deleted it is possible to delete all the related units and DUIBs at the same time.

```
    (NAM) Device-Unit Name [1-14 chars]
```

This parameter specifies a name that uniquely identifies this device-unit to the I/O System. If you want the Extended I/O System to logically attach this device during initialization, the name you specify for this parameter must also be one of the device names you specified on the "Logical Names" screen (see Chapter 8).

The ICU allows you to enter from one to fourteen characters. Refer to the *ASM286 Language Reference Manual* for rules regarding this name.

The name you specify for this parameter is the physical name you specify when invoking the Human Interface ATTACHDEVICE command, the BIOS A$PHYSICAL$ATTACH$DEVICE system call, or the EIOS LOGICAL$ATTACH$DEVICE EIOS system call.

```
    (PFD) Physical File Driver Required [Yes/No]    YES
    (NFD) Named File Driver Required [Yes/No]       YES
```

This driver supports both named and physical file drivers. These parameter lines let you specify "Yes" to at least one of these parameters. However, there are two good reasons for not changing either "Yes" default value. First, the Human Interface ATTACHDEVICE command, the EIOS LOGICAL$ATTACH$DEVICE system call, and the BIOS A$PHYSICAL$ATTACH$DEVICE system call require you to select the file type at run time. Second, there is no code savings if you specify "Yes" to one parameter and "No" to the other.

```
    (GRA) Granularity [0-0FFFFH]                    040H
```

This parameter specifies the minimum number of bytes that the device reads or writes in one operation. A value of 20H or 40H is recommended.

The manufacturer of your drive may recommend values for this parameter. Refer to vendor documentation for additional information. Refer to the *Extended iRMX II Basic I/O System User's Guide* for more information about granularity.

```
    (DSZ) Device Size [0-0FFFFFFFFH]                    020000H
```

This parameter specifies the device storage capacity in bytes. The default value 020000H is equivalent to 131072 decimal.

```
    (UN)  Unit Number on this Device [0-0FFH]         0H
```

This parameter specifies the unit number of this device-unit. This number identifies the only possible unit on this device.

```
    (UIN) Unit Info Name [1-16 Chars]
```

This parameter specifies the "Unit Info Name" of a unit information table that fills the needs of this DUIB. The following rules apply to the name you choose:

- You must create a unit information table with this "unit info name."
- Each DUIB can point to only one unit information table.
- A particular unit information table can be pointed to by more than one DUIB.

```
    (RUT) Request Update Timeout [0-0FFFFH]          064H
```

This parameter specifies the number of clock ticks that the driver should wait (during a pause in activity) before updating the attached files on the device. The default value 064H is equivalent to 100 decimal.

The values 0 and 0FFFFH do not indicate time intervals. A value of 0 indicates that the driver will never leave any data buffered in memory (eliminating the need to update). A value of 0FFFFH indicates that updates on this device will occur only when a file is detached.

The update capability provided by this parameter differs from the common update capability described in Chapter 9. Unlike the ability to update at fixed periods that is provided by the common update feature, the update capability (referred to by the "Request Update Timeout" parameter) allows the driver to update based on pauses in activity. If there is a pause in activity on this device, the driver determines how soon the common update would occur and compares that time interval to the request update timeout value. The driver then waits the shorter of the two intervals and updates the attached files on the device. Thus, if you use the common update feature, the time interval you specify for the "Request Update Timeout" should be shorter than time interval specified for the "Common Update Timeout."

```
     (NB)  Number of Buffers [nonrandom - 0/rand - 1-0FFFFH]    04H
```

This parameter specifies the number of buffers this device has for blocking and deblocking I/O requests. The default value of 4 is appropriate for a unit that has a granularity of 64 bytes. Since this is a device which supports random access, do not specify a value of zero.

The number of buffers prompt determines the number of Basic I/O System buffers the device uses for I/O. These buffers are general-purpose cache buffers that the Basic I/O System uses to increase I/O performance. The size of each buffer is 32 bytes greater than the device granularity you specified for this device (your response to the "(GRA) Granularity" parameter). The buffers are associated with a device-unit, not with a particular file. The Basic I/O System assigns these buffers to the device-unit when it attaches the device-unit.

Since the Operating System uses memory from the BIOS memory pool to create these buffers, you should increase the values you specified on the "BIOS" screen for both the minimum and maximum memory pool sizes if you increase the default value for the "Number of Buffers" parameter. An explanation of how to calculate memory usage based on your responses to the "(GRA) Granularity" and the "(NB) Number of Buffers" parameter lines is in Chapter 9.

Choosing an optimum number of buffers depends on many factors: how your application works; the speed of your components, your CPU, your memory, and your peripherals; the performance characteristics of your peripherals; and the number of users. Plus, you must trade off optimum performance against the amount of memory you use (because the more buffers you choose, the more memory you need).

Because there are so many factors, the only way that you can obtain the optimum value for your system is by testing different values and fine-tuning your configuration. Intel recommends that you use at least two buffers for this device.

```
     (CUP) Common Update [Yes/No]                              YES
```

Specify "Yes" if you want the driver to update the attached files on this device at the fixed interval you specified on the "BIOS" screen. Specify "No" if you do not want to update attached files on this device at fixed intervals but prefer to update attached files based only on the time interval you specified for the "Request Update Timeout" parameter. You should not specify "No" for this parameter and 0FFFFH for the "Request Update Timeout" parameter. See Chapter 9 for information about common update timeout (updating attached files at fixed intervals).

```
    (MB)  Max Buffers [0-0FFH}                        0FFH
```

This parameter specifies the maximum number of buffers that the Extended I/O System can allocate for this device's I/O. The default value for this device (0FFH) allows the S$OPEN system call to specify the actual number of Extended I/O System buffers. The Operating System takes memory required for these buffers from the calling job's memory pool, so by setting this parameter to 0FFH you allow the calling job to select the number of buffers based on its own memory pool size.

For tape drives this value is zero.

Query Screen

After you have completed the "iSBX 251 Driver DUIB Information" screen, the query screen is displayed. It contains only one line:

```
    Do you want any/more iSBX 251 DUIBs ?
```

Respond "Yes" to this prompt if you plan to use the iSBX 251 MULTIMODULE board with two bubble devices that have different characteristics.

While developing your initial systems, you can create as many device-unit information blocks as you want. The number of DUIBs can exceed the number of devices on your system. The particular DUIB associated with the device depends on the physical name you use when attaching it. Once you know that you will never need a particular DUIB, save memory by deleting it from your definition file before you generate your configuration files.

Ram Driver Parameters

The RAM driver allows you to use a portion of the 80286/80386 microprocessor memory address space as a RAM disk (that is, an area of memory treated as a secondary storage device by the system). The RAM driver supports a RAM disk with up to 16 units located in any contiguous memory within the Operating System's 16M byte address space. The maximum allowable size of the RAM disk is 16M bytes minus the size of the system it is configured into.

You can use the RAM driver in two different ways: it can allow the I/O system to use an area of RAM as a named or physical volume; it can be used in conjunction with the Human Interface commands ADDLOC and LOCDATA to contain Human Interface commands (see the *Operator's Guide To The Extended iRMX II Human Interface* for more information). Two screens define the interface between the RAM custom device driver and the I/O system. The RAM driver supports the READ, WRITE, and SPECIAL functions.

If you intend to the use the driver to emulate a disk storage device, you can attach and format the RAM device after booting the system. You can then access the device as if it were a disk (until the system is reset). If you intend to use the driver to store Human Interface commands, you want these commands available in the RAM disk when you boot your system. To do this you need to use the LOCDATA and ADDLOC commands. LOCDATA creates an object module (which contains an image of the RAM disk) and ADDLOC adds the module to an boot-loadable file. This process allows the third stage of the Bootstrap Loader to load the RAM-disk.

These screens relate to the three device configuration tables: the device information table, the unit information table and the device unit information block (DUIB). Refer to the *Extended iRMX II Device Drivers User's Guide* for further information about these tables.

Ram Disk Driver Screen

The "RAM Disk Driver" screen contains only one parameter line. The ICU uses the device name you enter on this screen to establish a communication link between the driver and its units.

```
(DRAM)       RAM Disk Driver

(DEV) Device Name [1-16 Chars]

Enter [ Abbreviation = new value / Abbreviation ? / H ]
:
```

```
   (DEV) Device Name [1-16 Chars]
```

This parameter provides a means of associating the device and all the units and DUIBs which belong to it. The ICU stores all the screens by type, not by device. This means that all device driver screens are stored together, all unit information screens and all DUIB screens. The ICU "knows" which unit information and device-unit information screen relate to a particular device driver by the device name you enter in the "DEV" parameter.

Query Screen

After you have completed the "RAM Disk Devices" screen, the query screen is displayed. It contains only one line:

```
   Do you have any/more RAM Disk Driver DEVICEs ?
```

Respond to this prompt with a "Yes" if you want to add more device drivers of this type.

Ram Driver Unit Information Screen

The ICU uses the information from the following screen to create a unit information table for the RAM driver.

```
(URAM)    RAM Disk Driver Unit Information

(DEV) Device Name [1-16 Chars]
(NAM) Unit Info Name [1-16 Chars]
(BMA) Base Memory Address [0-0FFFFFFH]              0100000H
(WP)  Write Protected [Yes/No]                      NO

Enter [ Abbreviation = new value / Abbreviation ? / H ] :
:
```

```
    (DEV)  Device Name [1-16 Chars]
```

The name you enter for this parameter must be the same name you entered in the "DEV" parameter on the driver screen. This name provides the logical ICU connection between a driver and all of its units. Thus, when a device is deleted it is possible to delete all the related units at the same time.

```
    (NAM) Unit Info Name [1-16 Chars]
```

This parameter specifies a unique unit information name for this device. The first character must be an alphabetic character. Refer to the *ASM286 Language Reference Manual* for rules regarding this name. This name should help identify this unit information table versus any other unit information table.

A DUIB uses the "Unit Info Name" to point to this particular unit information table. When developing your initial systems, you can create unit information tables that are never pointed to by a DUIB. There is no harm in this process during the development stages. However, when you configure your final system, eliminate all unused unit information tables to save memory.

```
    (BMA) Base Memory Address [0-0FFFFFFH]          0100000H
```

This parameter specifies the physical address at which the RAM disk starts. This address may be anywhere within the 16M byte address space available to the system; however, it must be on a WORD boundary.

Your response to this parameter determines the starting address for the RAM disk. Your response to the "Device Size" parameter (on the "Device Unit" screen) determines the size of the RAM disk. This block of memory can not be declared on the "Memory for System" or Memory for Free Space Manager" screens. (The Memory screens declare memory that can be used by the system to locate system modules or to be used as free space.)

You can locate the device anywhere in addressable memory, but if the Bootstrap Loader loads the RAM-disk, do not locate the device in memory locations 0B000H-0BFFFFH which are used by the Bootstrap Loader.

Although the Bootstrap Loader may reside in PROM, it requires RAM addresses B8000H - BBFFFH for the second stage and 0BC000H - 0BFFFFH to store its third stage. It uses this data area during the bootstrap load process. This RAM area cannot be used for other purposes until the bootstrap load process completes. If any part of the Operating System is located in the Bootstrap Loader third-stage area, thereby causing the Bootstrap Loader to load code over itself, the load process will fail. Therefore, during system configuration, you must ensure that none of the modules eventually loaded by the Bootstrap Loader have addresses that conflict with the Bootstrap Loader third-stage area. After the load process completes, you can use the third-stage area for dynamic memory (or for the RAM driver if it emulates a disk storage device).

```
    (WP)  Write Protected [Yes/No]                      NO
```

Use this parameter to write protect your RAM driver. Since you cannot format a write protected device, respond with a "Yes" to this parameter if you plan on using the ADDLOC and LOCDATA commands to create a bootloadable file containing a RAM disk image. Refer to the *Operator's Guide to the Extended iRMX II Human Interface* for detailed steps of how to use the ADDLOC and LOCDATA commands.

You must specify "Yes" for this particular usage if you locate the RAM driver in PROM.

Query Screen

After you have completed the "RAM Disk Driver Unit Information" screen, the query screen is displayed. It contains only one line:

```
    Do you want any/more RAM Disk Driver UNITs  ?
```

Respond to this prompt with a "Yes" if you need another unit information table for this device. Otherwise, respond to this prompt with a "No".

Ram Driver Device-Unit Information Screen

The ICU uses the information from the following screen to create a device unit information block (DUIB) for the RAM driver.

```
(IRAM)   RAM Disk Device-Unit Information

(DEV) Device Name [1-16 Chars]
(NAM) Device-Unit Name [1-14 Chars]
(PFD) Physical File Driver Required [Yes/No]                YES
(NFD) Name File Driver Required [Yes/No]                    YES
(GRA) Granularity [0-0FFFFH]                                0200H
(DSZ) Device Size [0-0FFFFFFFFH]                            0100000H
(UN)  Unit Number of this Device [0-0FFH]                   0H
(UIN) Unit Info Name [1-16 Chars]
(RUT) Request Update Timeout [0-0FFFFH]                     0H
(NB)  Number of Buffers [nonrandom = 0/rand = 1-0ffffH]     02H
(CUP) Common Update [Yes/No]                                YES
(MB)  Max Buffers [0-0FFH]                                  0FFH

Enter [ Abbreviation = new value / Abbreviation ? / H ]
:
```

```
    (DEV)  Device Name [1-16 Chars]
```

The name you enter for this parameter must be the same name you entered in the "DEV" parameter on the driver screen. This name provides the logical ICU connection between the driver and all of its DUIBs. Thus, when a device is deleted it is possible to delete all the related units and DUIBs at the same time.

```
    (NAM) Device-Unit Name [1-14 Chars]
```

This parameter specifies a unique name that identifies this device-unit to the I/O System. If you want the Extended I/O System to logically attach this device during initialization, the name you specify for this parameter must also be one of the device names you specified on the "Logical Names" screen (see Chapter 8).

The ICU allows you to enter from one to fourteen characters. Refer to the *ASM286 Language Reference Manual* for rules regarding this name.

The name you specify for this parameter is the physical name you specify when invoking the Human Interface ATTACHDEVICE command, the BIOS A$PHYSICAL$ATTACH$DEVICE BIOS system call or the EIOS LOGICAL$ATTACH$DEVICE system call.

```
    (PFD) Physical File Driver Required [Yes/No]     YES
    (NFD) Named File Driver Required [Yes/No]        YES
```

This driver supports both named and physical file drivers. These parameter lines let you specify "Yes" to at least one of these parameters.

Two good reasons exist for not changing either "Yes" default value. First, the Human Interface ATTACHDEVICE command, the EIOS LOGICAL$ATTACH$DEVICE system call, and the BIOS A$PHYSICAL$ATTACH$DEVICE system call require you to select the file type at run time. Second, there is no code savings if you specify "Yes" to one parameter and "No" to the other.

```
    (GRA) Granularity [0-0FFFFH]                     0200H
```

This parameter specifies the minimum number of bytes of information that the device reads or writes in one operation. This value is also called device granularity and must be an even number.

The default value 0200H is equivalent to 512 decimal. Refer to the *Extended iRMX II Basic I/O System User's Guide* for more information about granularity.

```
    (DSZ) Device Size [0-0FFFFFFFFH]                 0100000H
```

This parameter specifies the device storage capacity in bytes. Device size must be a multiple of the device granularity. Device granularity can be any integral multiple of 16. The best values are 128, 256, 512, and 1024 bytes. All I/O requests are multiples of device granularity. (I/O requests are always multiples of device granularity if "number of buffers" is non-zero.)

```
    (UN)  Unit Number on this Device [0-0FFH]        0H
```

This parameter specifies the unit number of the device-unit. This device may have up to 16 units.

```
    (UIN) Unit Info Name [1-16 Chars]
```

This parameter specifies the "Unit Info Name" of a unit information table that fills the needs of this DUIB. The following rules apply to the name you choose:

- You must create a unit information table with this "unit info name".
- Each DUIB can point to only one unit information table.
- A particular unit information table can be pointed to by more than one DUIB.

```
    (RUT) Request Update Timeout [0-0FFFFH]          0H
```

This parameter specifies the number of clock ticks that the driver should wait (during a pause in activity) before updating the attached files on the device.

The values 0 and 0FFFFH do not indicate time intervals. A value of 0 indicates that the driver will never leave any data buffered in memory (eliminating the need to update). A value of 0FFFFH indicates that updates on this device will occur only when a file is detached. However, the ICU treats your response to this parameter totally separate from the response made to the "Common Update" parameter (you can specify both).

The update capability provided by this parameter differs from the common update capability described in Chapter 9. Unlike the ability to update at fixed periods that is provided by the common update feature, the update capability referenced by the "Request Update Timeout" parameter allows the driver to update based on pauses in activity. If there is a pause in activity on this device, the driver determines how soon the common update would occur and compares that time interval to the request update timeout value. The driver then waits the shorter of the two intervals and updates the attached files on the device. Thus, if you use the Common Update feature, the time interval you specify for the "Request Update Timeout" should be shorter than time interval specified for the "Common Update Timeout."

```
    (NB)  Number of Buffers [nonrandom = 0/rand = 1-0FFFFH]   02H
```

This parameter specifies the number of buffers this device has for blocking and deblocking I/O requests. The default value of 02H is appropriate for a unit that has a granularity of 512 bytes. Do not specify a value of zero since I/O requests are always multiples of device granularity.

Because the Operating System uses memory from the BIOS memory pool to create these buffers, you should increase the values you specified on the "BIOS" screen for both the minimum and maximum memory pool sizes if you increase the default value for the "Number of Buffers" parameter. An explanation of how to calculate memory usage based on your responses to the "(GRA) Granularity" and the "(NB) Number of Buffers" parameter lines is in Chapter 9.

The number of buffers prompt determines the number of Basic I/O System buffers the device uses for I/O. These buffers are general-purpose cache buffers that the Basic I/O System uses to increase I/O performance. The size of each buffer is 32 bytes greater than the device granularity you specified for this device (your response to the "(GRA) Granularity" parameter). The buffers are associated with a device-unit, not with a particular file. The Basic I/O System assigns these buffers to the device-unit when it attaches the device-unit.

Choosing an optimum number of buffers depends on many factors: how your application works; the speed of your components, your CPU, your memory, and your peripherals; the performance characteristics of your peripherals; and the number of users. Plus, you must trade off optimum performance against the amount of memory you use (because the more buffers you choose, the more memory you need).

Because there are so many factors, the only way that you can obtain the optimum value for your system is by testing different values and fine-tuning your configuration. Intel recommends that you use at least two buffers for this device.

```
    (CUP) Common Update [Yes/No]                     YES
```

Specify "Yes" if you want the driver to update the attached files on this device at the fixed interval you specified on the "BIOS" screen. Specify "No" if you do not want to update attached files on this device at fixed intervals but prefer to update attached files based only on the time interval you specified for the "Request Update Timeout" parameter. You should not specify "No" for this parameter and 0FFFFH for the "Request Update Timeout" parameter. See Chapter 9 for information about common update timeout (updating attached files at fixed intervals).

```
    (MB)  Max Buffers [0-0FFH]                       0FFH
```

This parameter specifies the maximum number of buffers that the Extended I/O System can allocate for this device's I/O. The default value for this device (0FFH) allows the S$OPEN system call to specify the actual number of Extended I/O System buffers. The memory required for these buffers is taken from the calling job's memory pool, so by setting this parameter to 0FFH you allow the calling job to select the number of buffers based on its own memory pool size.

Query Screen

After you have completed the "RAM Disk Device-Unit Information" screen, the query screen is displayed. It contains only one line:

```
   Do you want any/more RAM Disk Driver DUIBs ?
```

Respond "Yes" to this prompt if you need more than one DUIB to adequately define the RAM driver.

While developing your initial systems, you can create as many device-unit information blocks as you want. The number of DUIBs can exceed the number of devices on your system. The particular DUIB associated with the device depends on the physical name you use when attaching it. Once you know that you will never need a particular DUIB, save memory by deleting it from your definition file before you generate your configuration files.

SCSI Driver Parameters

The SCSI driver supports both 5.25- and 8-inch Winchester controllers and 5.25-inch flexible disk controllers that meet the Small Computer System Interface specifications described in the ANSI document ANSC X3T9.2/82-2. Three screens define the interface between the SCSI random access device driver and the I/O system. These screens relate to the three device configuration tables: the device information table, the unit information table, and the device unit information block (DUIB).

The values shown on the screens in this section are the same as values you would see if you invoked the ICU using the 286100A.def description file.

The Small Computer System Interface (SCSI) driver uses the same three tables to define the interface between the driver and the I/O system as do other drivers. Many of the parameters that normally appear on screens for other drivers also appear on the screens for the SCSI driver. There is one basic difference, however. Not only do the screens ask for information that pertains to a specific controller, to unique devices, and to groups of devices that share common characteristics, but the screens must also define the interface on the host CPU board.

The SCSI driver supports only SCSI or SASI boards that utilize the following commands:

Command	Opcode
test_unit_ready	00H
format_drive	04H
request_sense	03H
read	08H
write	0AH
seek	0BH

The SCSI or SASI boards must also support a single initialization command, such as mode select (15H), or no initialization command at all. In general, this driver supports SCSI boards that adhere to the ANSI SCSI specification. This driver also supports SASI boards that utilize the needed commands.

SCSI Driver Screen

The ICU uses the information from the following screen to create a device information table for the SCSI host adapter. The Operating System supports only one device information table for the SCSI host adapter. (This is different from other drivers which need multiple device information tables to support multiple controllers.)

```
(DSCSI)      SCSI Driver

(DEV) Device Name [1-16 Chars]                      SCSI
(IL)  Interrupt Level [Encoded Level]               048H
(ITP) Interrupt Task Priority [0-255]               130
(POA) 8255A Port A Address [0-0FFFFH]               0C8H
(POB) 8255A Port B Address [0-0FFFFH]               0CAH
(POC) 8255A Port C Address [0-0FFFFH]               0CCH
(CON) 8255A Control Port Address [0-0FFFFH]         0CEH
(HID) Host ID [0-0FFH]                              0H
(ARB) SCSI Bus Arbitration [Yes/No]                 NO
(DCT) DMA Controller Type                           ADMA
(DCP) DMA Controller Base Port [0-0FFFFH]           0200H
(DPS) DMA Port Separation [0-0FFH]                  02H
(DCH) DMA Channel [0-3]                             0
(DCR) DMA Control Register Address [0-FFFFH]        0D1H

Enter  [ Abbreviation = newvalue / Abbreviation ? / H ]
:
```

```
(DEV) Device Name [1-16 Chars]                      SCSI
```

This parameter provides a means of associating the device and all the units and DUIBs that belong to it. The ICU stores all the screens by type, not by device. This means that all device driver screens, all unit information screens, and all DUIB screens are stored together. The ICU determines which unit information and device-unit information screen relate to a particular device driver by the name you enter in this parameter.

```
(IL)  Interrupt Level [Encoded Level]               048H
```

Use this parameter to specify the encoded interrupt level for the SCSI driver. The interrupt task uses this value to associate the interrupt task with the correct interrupt level. The default value "048H" (0000 0000 0100 1000 binary) specifies master interrupt level 4. If you need to change this value, you should specify a value that is numerically greater (lower priority) than the level you specified for the parameter "(CIL) Clock Interrupt Level" on the "Hardware" screen.

The possible values for this field are encoded as follows (where bit 0 is the low-order bit):

Master		with Slave Attached		Master		with Slave Attached	
Level	Code	Level	Code	Level	Code	Level	Code
0	0008H	0-7	0000-0007H	4	0048H	0-7	0040-0047H
1	0018H	0-7	0010-0017H	5	0058H	0-7	0050-0057H
2	0028H	0-7	0020-0027H	6	0068H	0-7	0060-0067H
3	0038H	0-7	0030-0037H	7	0078H	0-7	0070-0077H

```
(ITP) Interrupt Task Priority [0-255]         130
```

Use this parameter to specify the initial priority of the device's interrupt task. The actual priority of the interrupt task changes because the Nucleus adjusts an interrupt task's priority according to the interrupt level it services.

```
(POA) 8255A Port A Address [0-0FFFFH]         0C8H
(POB) 8255A Port B Address [0-0FFFFH]         0CAH
(POC) 8255A Port C Address [0-0FFFFH]         0CCH
(CON) 8255A Control Port Address [0-0FFFFH]   0CEH
```

These parameter lines enable you to specify the addresses of the 8255A Programmable Interface ports on your processor board. These addresses must match the fixed settings on your processor board.

```
(HID) Host ID [0-0FFH]                        0H
```

Use this parameter to specify the SCSI Device ID that you want assigned to the host processor. The possible values for this parameter are 080H, 040H, 020H, 010H, 08H, 04H, 02H, 01H, and 0H.

The host and each controller have individual device IDs. You specify the device ID for the host through this parameter. The ICU determines the IDs for each controller from your response to the "Device-Unit Number" parameter in the "SCSI Device-Unit Information" screen.

```
(ARB) SCSI Bus Arbitration [Yes/No]           NO
```

This parameter is not currently supported. Do not change the default setting of "NO".

```
(DCT) DMA Controller Type                          ADMA
```

This parameter specifies the type of direct memory access (DMA) controller used in your system. The only valid entry for this parameter at this time is "ADMA". Do not change the default. Other DMA controllers will be supported in future releases.

```
(DCP) DMA Controller Base Port [0-0FFFFH]          0200H
```

This parameter specifies the base I/O port address of your system's DMA controller.

```
(DPS) DMA Port Separation [0-0FFH]                 02H
```

Set this parameter to indicate the separation (in bytes) between the I/O ports on your system's DMA controller. The value "02H" is appropriate for most 16-bit Intel boards.

```
(DCH) DMA Channel [0-3]                            0
```

The DMA channel is the channel through which the processor communicates with the DMA controller used in your system.

Set this parameter to indicate the DMA channel associated with the I/O port specified in the "(DCP) DMA Controller Base Port" parameter. An appropriate value for the iSBC 286/100A board is "0".

```
(DCR) DMA Control Register Address [0-FFFFH]    0D1H
```

The DMA control register allows dynamic multiplexing of the ADMA channels 0 and 1 for the various sources on the iSBC 286/100A board. Use this parameter to specify the I/O address of the DMA control register. This parameter applies only to the iSBC 286/100A board and should be "0D1H".

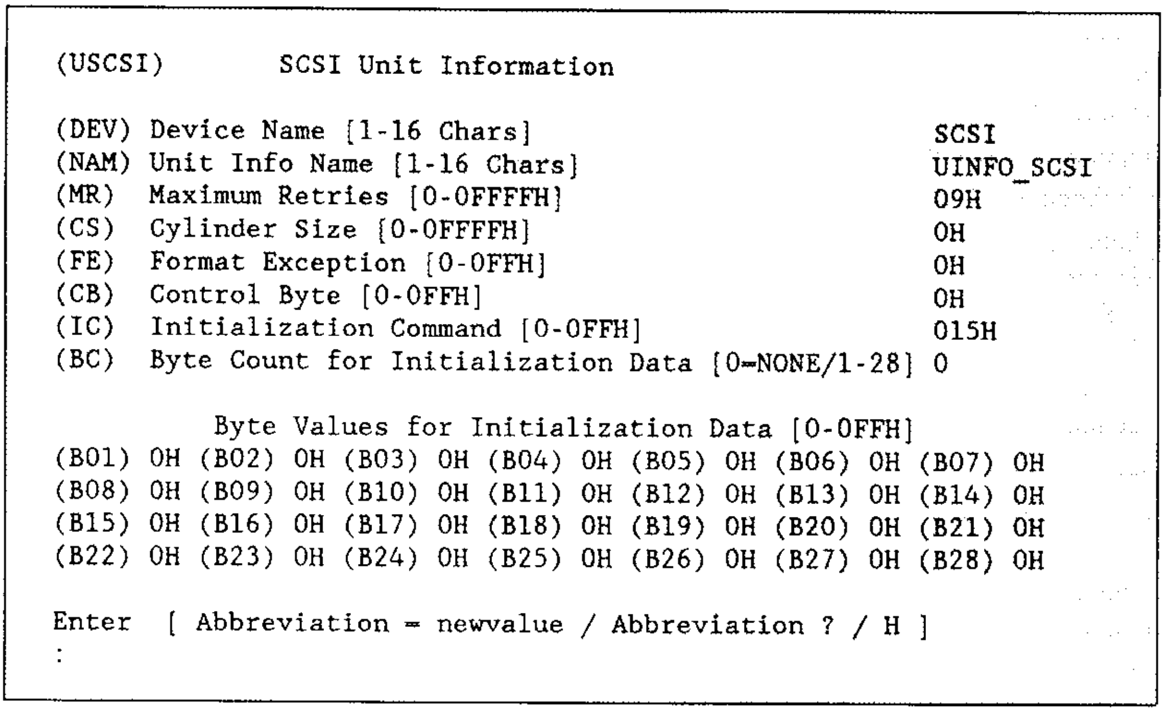
Query Screen

```
Do you want any/more SCSI DEVICEs?
```

Respond to this prompt with a "Yes" if you want to add more device drivers of this type.

SCSI Driver Unit Information Screen

The ICU uses the information from the following screen to create a unit information table for the SCSI driver.



```
(USCSI)       SCSI Unit Information

(DEV) Device Name [1-16 Chars]                               SCSI
(NAM) Unit Info Name [1-16 Chars]                            UINFO_SCSI
(MR)  Maximum Retries [0-0FFFFH]                             09H
(CS)  Cylinder Size [0-0FFFFH]                               0H
(FE)  Format Exception [0-0FFH]                              0H
(CB)  Control Byte [0-0FFH]                                  0H
(IC)  Initialization Command [0-0FFH]                        015H
(BC)  Byte Count for Initialization Data [0=NONE/1-28] 0

        Byte Values for Initialization Data [0-0FFH]
(B01) 0H (B02) 0H (B03) 0H (B04) 0H (B05) 0H (B06) 0H (B07) 0H
(B08) 0H (B09) 0H (B10) 0H (B11) 0H (B12) 0H (B13) 0H (B14) 0H
(B15) 0H (B16) 0H (B17) 0H (B18) 0H (B19) 0H (B20) 0H (B21) 0H
(B22) 0H (B23) 0H (B24) 0H (B25) 0H (B26) 0H (B27) 0H (B28) 0H

Enter  [ Abbreviation = newvalue / Abbreviation ? / H ]
:
```

```
(DEV) Device Name [1-16 Chars]                               SCSI
```

This parameter provides a means of associating the device and all the units and DUIBs that belong to it. The ICU "knows" which unit information and device-unit information screen relate to a particular device driver by the name you enter in the "(DEV) Device Name" parameter.

```
(NAM) Unit Info Name [1-16 Chars]                            UINFO_SCSI
```

Use this parameter to specify a unique name for this unit information table. The first character must be an alphabetic character. Refer to the *ASM286 Language Reference Manual* for rules regarding this name. Table 10-12 lists examples of recommended names for the SCSI driver.

A DUIB uses the "Unit Info Name" to point to this particular unit information table. When developing your initial systems, you can create unit information tables that are never pointed to by a DUIB. There is no harm in this process during the development stages. However, when you configure your final system, eliminate all unused unit information tables to save memory.

Table 10-12. SCSI Unit Information Names

Controller	Drive	Unit Info Name	Control Byte	Init. Command	Initialization Data
Generic	Generic	UINFO_SCSI	0000H	0015H	None
Adaptec ACB-4000	CMI 5619	UINFO_AT	0000H	0015H	00H, 00H, 00H, 08H, 00H, 00H, 00H, 00H, 00H, 00H, 02H, 00H, 01H, 01H, 32H, 06H, 00H, B4H, 00H, B4H, 00H, 02H
Xebec 1410	CMI 5619	UINFO_X1410A	0004H	000CH	01H, 32H, 06H, 00H, B4H, 00H, 00H, 0BH
Xebec 1410	Quantum Q540	UINFO_X1410B	0000H	000CH	02H, 00H, 08H, 02H, 00H, 00H, 0BH
Xebec 1420*	CMI 5619 or Fujitsu M2235	UINFO_X1420A	0000H	0011H	01H, 32H, 06H, 10H, 02H, 01H, 32H, 00H, B4H, 0BH
Xebec 1420*	Quantum Q540	UINFO_X1420B	0000H	0011H	02H, 00H, 08H, 11H, 02H 02H, 00H, 00H, F0H, 0BH
Xebec 1420*	Maxtor XT-1140	UINFO_X1420C	0000H	0011H	03H, 96H, 0FH, 11H, 02H 03H, 96H, 01H, C6H, 0BH
Xebec 1420*	Teac F55B floppy	UINFO_X1420MF	0000H	0011H	00H, 28H, 02H, 90H, 03H 0FH, 50H, 0FH, 14H, 00H

Note: * The Xebec 1420 is now superceded by the Xebec 1421. The Xebec 1420/1421 controller must be ordered with the Intel option to get the correct firmware.

```
(MR)  Maximum Retries [0-0FFFFH]                     09H
```

Use this parameter to specify the maximum number of times the Basic I/O System should retry an operation before returning an E$IO$SOFT exception code. Refer to the *Extended iRMX II Basic I/O System Calls* manual for related information on the E$IO$SOFT exception code.

```
(CS)  Cylinder Size [0-0FFFFH]                      0H
```

Use this parameter to specify the number of sectors per cylinder and whether the controller should perform automatic and concurrent seek operations. If you specify "0H", the I/O System will not perform either automatic or concurrent seek operations.

Automatic seek operations enable the random access software to ascertain the need to perform a seek operation before a requested read or write operation. The random access software is able to perform this function because it maintains the current location of the read/write head and can ascertain (based on the response to the "Cylinder Size" parameter) whether it should perform a read or write request within the current cylinder. If the request is not in the current cylinder, the random access software performs a seek operation.

The value you specify affects how or if seek operations are performed. If you specify "0H", the random access software does not perform automatic seek operations, the device driver for the unit does not call the SEEK$COMPLETE procedure, and the controller ignores all seek requests. Specify a zero value if the device can perform seek functions independent of the random access software. (In general, most SCSI controllers can perform seek functions independent of random access software.) If you specify a non-zero value, the value you should enter is the cylinder size for this device (the number of sectors per track multiplied by the number of heads per drive).

NOTE

Some SCSI controllers do not support concurrent seeks and some SCSI controllers cannot queue a read/write request while a drive is seeking. Set this parameter to zero in these cases.

```
(FE)  Format Exception [0-0FFH]                     0H
```

Use this parameter to specify the value of the error code returned by the SCSI command "Test Unit Ready" when a SCSI drive is not formatted. The SCSI driver will ignore this error code when it attempts to attach the SCSI drive immediately before an attempt to format it. A value of zero indicates that your SCSI controller does not return an error code when you attempt to attach an unformatted drive.

```
(CB)  Control Byte [0-0FFH]                         0H
```

Use this parameter to specify the contents of the last byte in all commands. This byte is called the "control byte" in the ANSI SCSI document. The bits in this byte have the following meaning:

Bits 7-6 Vendor Unique--set these bits as specified by the manufacturer of the SCSI controller

Bits 5-0 Reserved--set to zero

```
(IC)  Initialization Command [0-0FFH]                    015H
```

Use this parameter to specify the hexadecimal value of the mode select command (or an initialization command that the manufacturer of the SCSI controller has used to replace the mode select command). If your SCSI controller is device specific and does not require any initialization, use the value "0H".

```
(BC)  Byte Count for Initialization Data [0=NONE/1-28] 0
```

Use this parameter to specify both unit and device parameters. If your SCSI controller is device-specific and does not require any initialization, specify "0"; otherwise, enter the number of bytes (1-28) for the initialization data.

This parameter list enables you to specify an extent descriptor list. Each extent descriptor allows you to specify the number of sectors on the unit (called Number of Blocks in the SCSI specification), the device-unit granularity (called Block Size in the SCSI specification), and density code for a device-unit. You can add additional vendor-unique parameters to the extent descriptor list.

The first three bytes (B01 through B03) define an extent descriptor. They are defined as follows (SASI drivers also use this parameter but the following descriptions do not apply):

Byte B01 Media Type--set to "0H" for hard disks

Byte B02 Reserved--set to "0H"

Byte B03 Length in bytes of the extent descriptor list. The default value "08H" defines one extent descriptor and should be the typical value for hard disks supported by this driver.

The extent descriptor list follows bytes B01 through B03. Each group of eight bytes defines both unit and device characteristics. Each byte in this group is defined as follows:

Byte B04 Density Code--set to "0H" for hard disks

Byte B05 Reserved--set this value to "0H"

Byte B06 Number of sectors--this byte and Byte B07 define the number of sectors on this unit. Use this byte as the most significant byte in the word that defines the number of sectors.

Byte B07 Number of sectors--this byte and Byte B06 define the number of sectors on this unit. Use this byte as the least significant byte in the word that defines the number of sectors.

Byte B08 Granularity--this byte and Bytes B09, B10, and B11 define the granularity of a particular device-unit. This byte is the most significant byte in the two words that define granularity. The value defined by these two words has to equal the granularity value you specify on the device-unit screen.

Byte B09 Granularity--this byte and Bytes B08, B10, and B11 define the granularity of a particular device-unit. This byte is the second most significant byte in the two words that define granularity. The value defined by these two words has to equal the granularity value you specify on the device-unit screen.

Byte B10 Granularity--this byte and Bytes B08, B09, and B11 define the granularity of a particular device-unit. This byte is the third most significant byte in the two words that define granularity. The value defined by these two words has to equal the granularity value you specify on the device-unit screen.

Byte B11 Granularity--this byte and Bytes B08, B09, and B10 define the granularity of a particular device-unit. This byte is the least significant byte in the two words that define granularity. The value defined by these two words has to equal the granularity value you specify on the device-unit screen.

Byte B12 - B28 Vendor Unique--enter these bytes in accordance with the vendor's hardware reference manual and the particular device unit characteristics. The number of bytes is vendor unique.

Query Screen

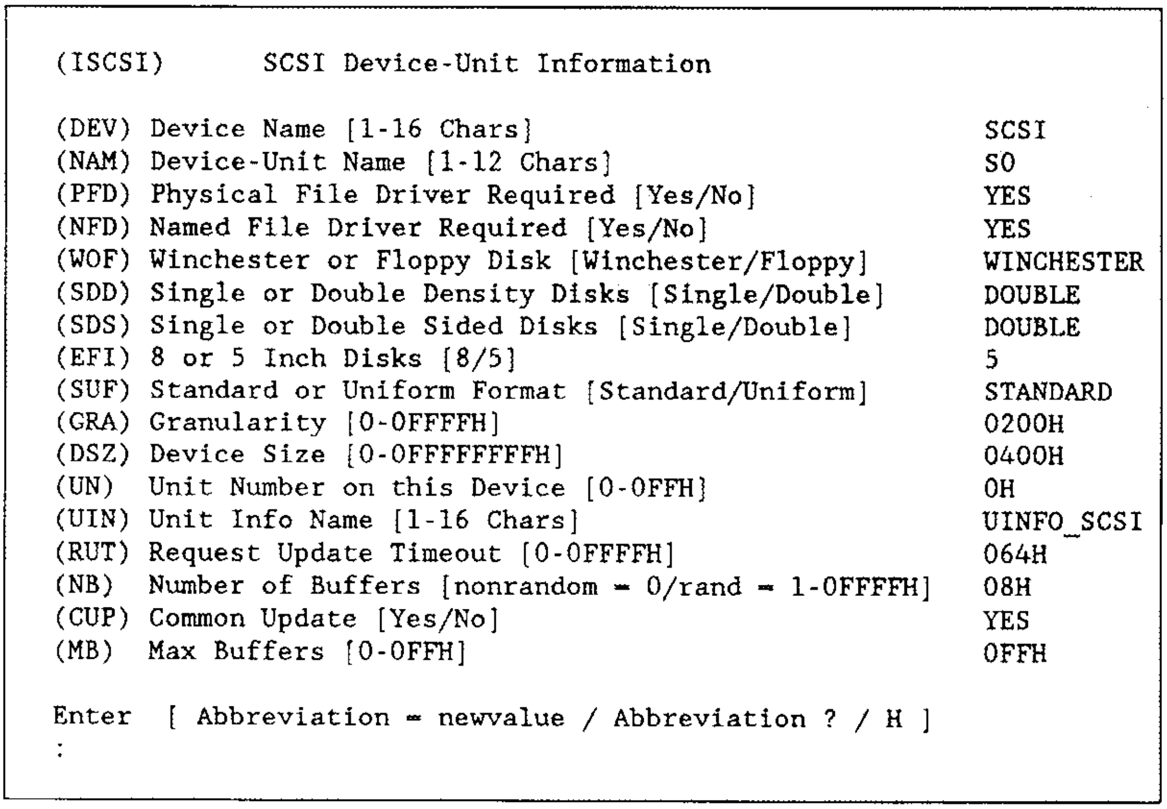
After you have completed the "SCSI Unit Information" screen, the query screen is displayed. It contains only one line:

```
Do you want any/more SCSI UNITs?
```

Respond "Yes" to this prompt if you need another unit information table for this device.

SCSI Device-Unit Information Screen

The ICU uses the information from the following screen to create a device unit information block (DUIB) for the SCSI driver.



```
(ISCSI)      SCSI Device-Unit Information

(DEV) Device Name [1-16 Chars]                                  SCSI
(NAM) Device-Unit Name [1-12 Chars]                             S0
(PFD) Physical File Driver Required [Yes/No]                    YES
(NFD) Named File Driver Required [Yes/No]                       YES
(WOF) Winchester or Floppy Disk [Winchester/Floppy]             WINCHESTER
(SDD) Single or Double Density Disks [Single/Double]            DOUBLE
(SDS) Single or Double Sided Disks [Single/Double]              DOUBLE
(EFI) 8 or 5 Inch Disks [8/5]                                   5
(SUF) Standard or Uniform Format [Standard/Uniform]             STANDARD
(GRA) Granularity [0-0FFFFH]                                    0200H
(DSZ) Device Size [0-0FFFFFFFFH]                                0400H
(UN)  Unit Number on this Device [0-0FFH]                       0H
(UIN) Unit Info Name [1-16 Chars]                               UINFO_SCSI
(RUT) Request Update Timeout [0-0FFFFH]                         064H
(NB)  Number of Buffers [nonrandom = 0/rand = 1-0FFFFH]         08H
(CUP) Common Update [Yes/No]                                    YES
(MB)  Max Buffers [0-0FFH]                                      0FFH

Enter  [ Abbreviation = newvalue / Abbreviation ? / H ]
:
```

```
(DEV) Device Name [1-16 Chars]                                  SCSI
```

The name you enter for this parameter must be the same name you entered for the "(DEV) Device Name" parameter on the "SCSI Driver" screen. This name provides the logical ICU connection between a driver and all of its DUIBs. Thus, when you delete a device you can delete all the related units and DUIBs at the same time.

```
(NAM) Device-Unit Name [1-12 Chars]                              S0
```

Use this parameter to specify a name that uniquely identifies this device-unit to the I/O System. If you want the Extended I/O System to logically attach this device during initialization, the name you specify for this parameter must also be one of the device names you specified on the "HI Logical Names" screen. To support auto boot device recognition, specify the same name in one of the %DEVICE macros when you configure this device in your bootstrap loader (see the *Extended iRMX II Bootstrap Loader Reference Manual* for detailed information). The default name supplied by the Bootstrap Loader is S0 for SCSI devices.

The ICU allows you to enter from one to twelve characters. Refer to the *ASM286 Language Reference Manual* for rules regarding this name. Table 10-13 lists examples of recommended names for the SCSI Controller.

The name you specify for this parameter is the physical name you specify when invoking the Human Interface ATTACHDEVICE command, the BIOS A$PHYSICAL$ATTACH$DEVICE system call, or the EIOS LOGICAL$ATTACH$DEVICE system call.

Table 10-13. SCSI DUIB Information

D U I B Information				
Device-Unit Name	Granularity	Device Size	Device Type	Unit Info Name
S0	200H	00000400H	Generic	UINFO_SCSI (generic scsi/sasi)
SAT0	200H	00F17400H	CMI 5619	UINFO_AT
SX1410A0	200H	00EB1200H	CMI 5619	UINFO_X1410A
SX1410B0	200H	02144E00H	Quantum Q540	UINFO_X1410B
SX1420A0	200H	00EB1400H	CMI 5619 or Fujitsu M2235	UINFO_X1420A
SX1420B0	200H	02145000H	Quantum Q540	UINFO_X1420B
SX1420C0	200H	070EEA00H	Maxtor XT-1140	UINFO_X1420C
SMF0	200H	0004F800H	Teac F55B floppy	UINFO_X1420MF

```
(PFD) Physical File Driver Required [Yes/No]                    YES
(NFD) Named File Driver Required [Yes/No]                       YES
```

This driver supports both named and physical file drivers. These parameter lines enable you to specify "Yes" to at least one of these parameters.

Two good reasons exist for not changing either "Yes" default value. First, the Human Interface ATTACHDEVICE command, the EIOS LOGICAL$ATTACH$DEVICE system call, and the BIOS A$PHYSICAL$ATTACH$DEVICE system call require you to select the file type at run time. Second, there is no code savings if you specify "Yes" to one parameter and "No" to the other.

```
(WOF) Winchester or Floppy Disk [Winchester/Floppy]       WINCHESTER
```

Use this parameter to indicate the type of disk drive that is connected to your SCSI controller, Winchester (hard disk drive) or Floppy (flexible disk drive).

```
(SDD) Single or Double Density Disks [Single/Double]      DOUBLE
```

Use this parameter to indicate the recording density of the flexible disk connected to your SCSI controller. Ignore this parameter for Winchester disk drives.

```
(SDS) Single or Double Sided Disks [Single/Double]        DOUBLE
```

Use this parameter to specify the number of recording surfaces on the flexible disk connected to your SCSI controller. Ignore this parameter for Winchester disk drives.

```
(EFI) 8 or 5 Inch Disks [8/5]                             5
```

Use this parameter to specify the size of the Winchester disk connected to your SCSI controller, 8 (for 8-inch disk) or 5 (for 5.25-inch disk). Ignore this parameter for flexible disks.

```
(SUF) Standard or Uniform Format [Standard/Uniform]       STANDARD
```

For Winchester drives, set this parameter to "Uniform".

For flexible disk drives, use this parameter to specify the format the controller should expect on track zero. Choose the option "Standard" if you want track zero to be single density with 128-byte sectors (regardless of how the remaining tracks are formatted). Choose the option "Uniform" if you want all tracks on a disk to have the same format.

Normally, when you use the FORMAT command to format a disk as a named volume, the command formats track zero with a fixed density (single density) and a fixed sector size (128 bytes). This is the "standard" format. Intel recommends that you use this format. The bootstrap loader supports booting only from "standard" format diskettes.

If you wish to read a disk that has a uniform format, you should use the "Standard/Uniform Format" parameter to designate a "Uniform" format. Note that automatic device characteristics recognition does not work unless track zero is single-density with 128-byte sectors.

```
(GRA) Granularity [0-0FFFFH]                                0200H
```

Use this parameter to specify the minimum number of bytes that the device reads or writes in one operation. This value is also called device granularity. Device granularity determines sector size, and multiples of device granularity define volume and file granularity.

The vendor of your drive may recommend optimum values for this parameter. Further, this value may be either software or hardware selectable. Refer to vendor documentation for additional information.

The default value "0200H" is equivalent to 512 decimal. Do not change the default value if you intend to use the automatic device characteristics recognition features. Unless you specified "0" for the parameter "(BC) Byte Count for Initialization Data [None/CR]" on the "SCSI Unit Information" screen, the granularity you specify for this parameter has to be equal to the granularity you specified in the "(BC) Byte Count for Initialization Data" parameter line. Refer to the *Extended iRMX II Basic I/O System User's Guide* for more information about granularity.

```
(DSZ) Device Size [0-0FFFFFFFFH]                            0400H
```

Use this parameter to specify the device storage capacity in bytes. Calculate this value as follows:

DSZ = [(total cylinders) - (# alternate cylinders)] * (# heads)
* (sectors per track) * device granularity

The default value of "0400H" specifies a "generic" SCSI/SASI Device Unit Information Block that takes advantage of the automatic device characteristics recognition feature of the Operating System. With this feature, whenever you attach a unit, the Operating System compares the information in the DUIB with the information written on track 0 of the unit itself (this information is placed there during formatting). If the information doesn't match, the Operating System searches for and uses another DUIB that matches the characteristics of the device. If no matching DUIB is found, a temporary DUIB is created and used. This feature allows you to connect and use different SCSI/SASI devices without reconfiguring the entire system. However, to format those devices, you must set up specific DUIBs that match the characteristics of the devices. Refer to the *Operator's Guide to the Extended iRMX II Human Interface* for additional information about automatic device characteristics recognition.

The 400H size of the "generic" DUIB is smaller than any actual device. This guarantees that the "generic" DUIB will not match the characteristics of any unit. Therefore, the Operating System, when attaching the unit, will search for (and possibly create) a matching DUIB.

If you set up "generic" DUIBs, be careful when setting up DUIBs for actual devices, particularly when setting up multiple DUIBs for the same unit. Unlike other kinds of drivers, the SCSI driver supports different kinds of controller boards. This enables you to set up multiple DUIBs for a single unit that has the same characteristics for the drive but specifies different controller boards. This can present problems when using the "generic" DUIB to attach the unit.

When the "generic" DUIB is used, the Operating System searches for a DUIB whose device characteristics match those written on the drive. If multiple DUIBs have matching characteristics, it uses the first one found. If that DUIB does not match the actual controller board used, (that is the DUIB points to the wrong unit information table), the Operating System will attempt to initialize the controller with the wrong initialization commands. The Operating System will be unable to attach the device, and the invalid initialization could damage the data on the drive.

To avoid these problems, whenever you set up multiple DUIBs for the same unit, where the drive is the same and the controller board differs, use a slightly smaller device size for each of the DUIBs (differences of a single byte or a single sector are enough). Then, when you use a "generic" DUIB to attach the device, the Operating System will be able to find and use the DUIB that actually matches the controller being used.

NOTE

After formatting the drive, do not change the device size in the DUIBs for actual devices; otherwise, the Operating System will not be able to locate the DUIB matching the controller.

```
 (UN)  Unit Number on this Device [0-0FFH]                 0H
```

Use this parameter to specify the controller ID number and the unit number of this device-unit. This field of information is different for SCSI drivers from the information you would normally enter on other device drivers. The byte of information you specify must use bits 0 through 2 to identify the device-unit number and bits 3 through 5 to identify the controller SCSI bus id (see Table 10-14 for examples). This byte can identify up to eight controllers (0-7) and eight devices per controller (also numbered 0-7).

Table 10-14. Example SCSI Device-Unit Numbers

Encoded Controller ID	Device-Unit Number	Reserved Bits	Controller Id Bits	Device-Unit Number Bits	Hex Values
		7 6	5 4 3	2 1 0	
00	00	0 0	0 0 0	0 0 0	0000H
01	00	0 0	0 0 1	0 0 0	0008H
01	01	0 0	0 0 1	0 0 1	0009H
02	00	0 0	0 1 0	0 0 0	0010H
02	01	0 0	0 1 0	0 0 1	0011H
02	02	0 0	0 1 0	0 1 0	0012H
07	00	0 0	1 1 1	0 0 0	0038H

```
(UIN)  Unit Info Name [1-16 Chars]                    UINFO_SCSI
```

Use this parameter to specify the "Unit Info Name" of a unit information table that fills the needs of this DUIB. The following rules apply to the name you choose:

- You must create a unit information table having this "unit info name."
- Each DUIB can point to only one unit information table.
- A particular unit information table can be pointed to by more than one DUIB.

```
(RUT) Request Update Timeout [0-0FFFFH]               064H
```

Use this parameter to specify the number of clock ticks that the driver should wait (during a pause in activity) before updating the attached files on the device. The default value "064H" is equivalent to 100 decimal.

The values "0" and "0FFFFH" do not indicate time intervals. A value of "0" indicates that the driver will never leave any data buffered in memory (eliminating the need to update). A value of "0FFFFH" indicates that updates on this device will occur only when a file is detached. However, the ICU treats your response to this parameter totally separate from the response made to the "Common Update" parameter (you can specify both).

The update capability provided by this parameter differs from the common update capability. Unlike the ability to update at fixed periods that is provided by the common update feature, the update capability (referred to by the "Request Update Timeout" parameter) allows the driver to update based on pauses in activity. If there is a pause in activity on this device, the driver determines how soon the common update would occur and compares that time interval to the request update timeout value. The driver then waits the shorter of the two intervals and updates the attached files on the device. Thus, if you use the common update feature, the time interval you specify for the "Request Update Timeout" should be shorter than time interval specified for the "Common Update Timeout".

```
(NB)  Number of Buffers [nonrandom = 0/rand = 1-0FFFFH]   08H
```

Use this parameter to specify the number of buffers this device has for blocking and deblocking I/O requests. The default value of "08H" is appropriate for a unit that has a granularity of 1024 bytes. Since this is a device which supports random access, do not specify a value of zero.

The number of buffers prompt determines the number of Basic I/O System buffers the device uses for I/O. These buffers are general-purpose cache buffers the Basic I/O System uses to increase I/O performance. The size of each buffer is 32 bytes greater than the device granularity you specified for this device (your response to the "(GRA) Granularity" parameter). The buffers are associated with a device-unit, not with a particular file. The Basic I/O System assigns these buffers to the device-unit when it attaches the device-unit.

Because the Operating System uses memory from the BIOS memory pool to create these buffers, you should increase the values you specified on the "BIOS" screen for both the minimum and maximum memory pool sizes if you increase the default value for the "Number of Buffers" parameter. An explanation of how to calculate memory usage based on your responses to the "(GRA) Granularity" and the "(NB) Number of Buffers" parameter lines is in Chapter 9.

Choosing an optimum number of buffers depends on many factors: how your application works; the speed of your components, CPU, memory, and peripherals; the performance characteristics of your peripherals; and the number of users. Plus, you must trade off optimum performance against the amount of memory you use (because the more buffers you choose, the more memory you need).

Because there are so many factors, the only way you can obtain the optimum value for your system is by testing different values and fine-tuning your configuration. Intel recommends that you use at least two buffers for this device.

```
(CUP) Common Update [Yes/No]                            YES
```

Use this parameter to specify "Yes" if you want the driver to update the attached files on this device at the fixed interval you specified on the "BIOS" screen. Specify "No" if you do not want to update attached files on this device at fixed intervals but prefer to update attached files based only on the time interval you specified for the "Request Update Timeout" parameter. You should not specify "No" for this parameter and "0FFFFH" for the "Request Update Timeout" parameter. See Chapter 9 for information about common update timeout (updating attached files at fixed intervals).

```
(MB)  Max Buffers [0-0FFH]                                    0FFH
```

Use this parameter to specify the maximum number of buffers that the Extended I/O System can allocate for this device's I/O. The default value for this device (0FFH) allows the S$OPEN system call to specify the actual number of Extended I/O System buffers. The Operating System takes memory required for these buffers from the calling job's memory pool, so by setting this parameter to "0FFH" you allow the calling job to select the number of buffers based on its own memory pool size. Intel recommends that you use the default value.

QUERY SCREEN

After you complete the "SCSI Driver Unit Information" screen, the query screen is displayed. It contains the following line:

```
Do you want any/more SCSI DUIBs ?
```

Respond "Yes" to this prompt if you plan to use the SCSI driver with two devices having different characteristics or if you have two devices with the same characteristics, but different unit numbers.

While developing your initial systems, you can create as many device-unit information blocks as you want. The number of DUIBs can exceed the number of devices on your system. The particular DUIB associated with the device depends on the physical name you use when attaching it. Once you know that you will never need a particular DUIB, save memory by deleting it from your definition file before you generate your configuration files.

iSBC 186/224A DRIVER PARAMETERS

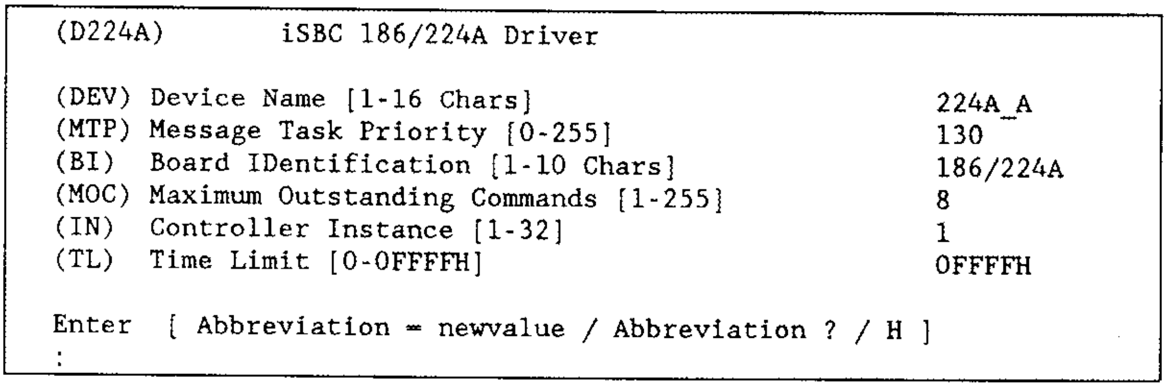
The iSBC 186/224A is a multi-peripheral controller, that supports up to four Winchester disk drives, four 5.25-inch flexible disk drives, and four streaming tape drives. The iSBC 186/224A device driver communicates with the iSBC 186/224A controller via the MULTIBUS II message-passing transport protocol of the parallel system bus (iPSB). Refer to the *Extended iRMX II Nucleus User's Guide* for more information on this protocol.

The values shown on the screens in this section are the same values you would see if you invoked the ICU using the 286100A.def or 386100.def description file.

Three screens define the interface between the MULTIBUS II iSBC 186/224A random access driver and the I/O system. These screens relate to the three device configuration tables: the device information table, the unit information table, and the device unit information block (DUIB). Refer to the *Extended iRMX II Device Drivers User's Guide* for further information about these tables.

iSBC 186/224A DRIVER SCREEN

The ICU uses the information from the following screen to create a device information table for the iSBC 186/224A driver. If your system includes more than one iSBC 186/224A controller, use this screen to complete a unique device information table for each iSBC 186/224A controller in your system. Specify a unique device name and instance number for each iSBC 186/224A controller.



```
(D224A)       iSBC 186/224A Driver

(DEV) Device Name [1-16 Chars]                          224A_A
(MTP) Message Task Priority [0-255]                     130
(BI)  Board IDentification [1-10 Chars]                 186/224A
(MOC) Maximum Outstanding Commands [1-255]              8
(IN)  Controller Instance [1-32]                        1
(TL)  Time Limit [0-0FFFFH]                             0FFFFH

Enter  [ Abbreviation = newvalue / Abbreviation ? / H ]
:
```

```
(DEV) Device Name [1-16 Chars]                        224A_A
```

This parameter provides a means of associating the device and all the units and DUIBs that belong to it. The ICU stores all the screens by type, not by device. This means that all device driver screens, all unit information screens, and all device-unit information (DUIB) screens are stored together. The ICU determines which unit information and device-unit information screen relate to a particular controller by the name you enter in the "(DEV) Device Name" parameter.

```
(MTP) Message Task Priority [0-255]                   130
```

This parameter specifies the priority of the task that receives messages from the controller. The default value is 130 decimal, which is approximately the same priority specified for other I/O tasks. To determine this setting, refer to the *Extended iRMX II Nucleus User's Guide.*

```
(BI)  Board IDentification [1-10 Chars]               186/224A
```

This parameter specifies the 10-character board name stored in registers two through eleven of the header record in the iSBC 186/224A board's interconnect space. For the iSBC 186/224A controller, the ID is the string "186/224A". The driver automatically pads this field with ASCII NULL (0) characters.

```
(MOC) Maximum Outstanding Commands [1-255]            8
```

This parameter limits the number of simultaneous controller commands the driver can have outstanding on the controller. This limit is determined by the controller. Refer to your vendor documentation for this value.

In the "(NCOM) Nucleus Communication Service" screen, you must increase the "(MST) Max No. of Simultaneous Transactions" and "(MSM) Max No. of Simultaneous Messages" values you specify by the number you enter for this parameter.

```
(IN)  Controller Instance [1-32]                      1
```

This parameter distinguishes two or more iSBC 186/224A controllers having identical board IDs in the same system. During initialization the driver calculates the instance by scanning the MULTIBUS II slots in ascending order and sequentially assigning numbers to each iSBC 186/224A controller found. For example, 1 is assigned to the iSBC 186/224A board in the lowest-numbered slot, and 2 to the iSBC 186/224A board in the next-lowest-numbered slot.

```
(TL)  Time Limit [0-0FFFFH]                                 0FFFFH
```

This parameter specifies the number of clock ticks that the driver's message task should wait for a message from the iSBC 186/224A controller. Use the default value of "0FFFFH" that means wait forever, or set this parameter to a value greater than the longest response time the controller would take to handle a command to the device (for example, the command to rewind a tape).

QUERY SCREEN

After you complete the "iSBC 186/224A Driver" screen, the query screen is displayed. It contains the following line:

```
Do you want any/more iSBC 186/224A DEVICEs ?
```

Respond "Yes" to this prompt if you want to configure more iSBC 186/224A boards in your system.

iSBC 186/224A UNIT INFORMATION SCREEN

The ICU uses the information from the following screen to create a unit information table for the iSBC 186/224A driver.

```
(U224A)   iSBC 186/224A Driver Unit Information

(DEV) Device Name [1-16 Chars]                              224A_A
(NAM) Unit Info Name [1-16 Chars]                           UINFO_224AMMA
(AR)  Automatic Retries [Yes/No]                            YES
(NC)  Number of Cylinders [0-0FFFFH]                        0396H
(NH)  Number of Heads [0-0FFH]                              0FH
(NS)  Number of Sectors/Track [0-0FFH]                      09H
(NAC) Number of Alternate Cylinders [0-0FFH]                12H
(BTI) Bad Track Information [Yes/No]                        YES
(SR)  Step Rate [0-0FFH]                                    0H
(RWC) Reduce Write Current Cylinder [0-0FFFFH]              01D0H
(WPC) Write Precompensation Cylinder [0-0FFFFH]             01D0H
(RD)  Reserved Drive [Yes/No]                               NO

Enter  [ Abbreviation = newvalue / Abbreviation ? / H ]
:
```

```
(DEV) Device Name [1-16 Chars]                    224A_A
```

This parameter provides a means of associating the device and all the units and DUIBs that belong to it. The ICU "knows" which unit information and device-unit information screen relate to a particular device driver by the name you enter in the "(DEV) Device Name" parameter.

```
(NAM) Unit Info Name [1-16 Chars]                 UINFO_224AMMA
```

This parameter specifies a unique name for this unit information table. The first character must be an alphabetic character. Refer to the *ASM286 Language Reference Manual* for rules regarding this name.

A DUIB uses the "Unit Info Name" to point to this particular unit information table. You can create unit information tables that are never pointed to by a DUIB. There is no harm in this process during the development stages. However, when you configure your final system, eliminate all unused information tables to save memory.

Table 10-1. iSBC® 186/224A Controller Information for Winchester Drives

Unit Information					
Unit-Info Name	Device Type	Number Cylinders	Number Heads	Sectors /Track	Alternate Cylinders
UINFO_224AW	"generic"	1	1	9	0
UINFO_224ACM	CMI 5412	306	4	9	10
UINFO_224ACMB	CMI 5619 Fujitsu M2235	306	6	9	10
UINFO_224AQMA	Quantum 540	512	8	9	10
UINFO_224ATMA	Toshiba MK56FB	830	10	9	10
UINFO_224AMMA	Maxtor XT-1140	918	15	9	18
UINFO_224AMMB	Maxtor XT-1085	1024	8	9	20
UINFO_224ASMA	Seagate SR251	820	6	9	10

Table 10-2. iSBC® 186/224A Controller Information for Diskette and Tape Drives

Unit Information				
Unit-Info Name	Device Type	Number Cylinders	Sectors /Track	Step Rate
UINFO_224AF	generic	77	26	2
UINFO_224AMFDX	Shugart 450 compatible drives	40	8	2
UINFO_224AMFDY	Shugart 460 compatible drives	80	8	0
UINFO_224AWT	QIC-02 streaming tape	0	0	0

```
(AR)  Automatic Retries [Yes/No]                    YES
```

This parameter determines whether the driver should retry the I/O operation if the operation resulted in a soft error. If you respond "Yes" to enable automatic retries, the iSBC 186/224A firmware performs up to two retries before returning an E$IO$HARD exception code. If you respond "No", the firmware performs no retries. Refer to the *Extended iRMX II Basic I/O System Calls* manual for related information on the E$IO$HARD exception code.

```
(NC)  Number of Cylinders [0-0FFFFH]                    0396H
```

For Winchester disk drives, this parameter specifies the total number of cylinders (tracks per surface) on the Winchester or flexible disk. In general, the number you specify should be the total of the number of data cylinders, including the diagnostic cylinder and the bad sector information (highest cylinder - 1) on the drive. The exception to this guideline is using the value "1". This value causes the driver to read the device characteristics from the volume label. Table 10-15 shows some typical values for this parameter.

For flexible diskette drives, this parameter specifies the number of tracks per side of a 5.25-inch disk. Table 10-16 shows some typical values for this parameter.

For tape drives, this parameter is not relevant. Set this value to "0H".

```
(NH)  Number of Heads [0-0FFH]                          0FH
```

For Winchester disk drives, this parameter specifies the number of fixed heads on your Winchester's platters. To obtain this value, see the vendor documentation for your device. Should you need to determine this number, multiply the number of platters by the number of surfaces per platter. For example, suppose your Winchester has three fixed platters and each platter has two surfaces. Assuming one read/write head per usable surface, the response to this parameter should be six.

For flexible disk and tape drives, this parameter is not relevant. Set this value to "0H".

```
(NS)  Number of Sectors/Track [0-0FFH]                  09H
```

For Winchester disk drives and flexible disk drives, this parameter specifies the number of sectors per track. Tables 10-1 and 10-2 show some typical values. For Winchester drives, the number of sectors per track varies with the sector size and the hardware vendor. For flexible disk drives, the number of sectors per track varies with the sector size, density, and overall size.

If you are using the automatic device characteristics recognition feature, specify a value that represents the maximum sectors-per-track value of all the disks accessed by the controller. If the value you use for this parameter is not as large as the actual number of sectors per track, you will experience reliability problems when accessing the disk.

For tape drives, this parameter is not relevant. Set the parameter to "0H".

```
(NAC) Number of Alternate Cylinders [0-0FFH]          12H
```

For Winchester disk drives, this parameter specifies the number of alternate cylinders on your Winchester device. Use a value that represents one to two percent of the total number of available cylinders on the unit.

When you format a Winchester device, the device driver detects bad tracks and assigns one alternate track for each bad track it detects. These alternate tracks are located on the innermost cylinders of the disk.

During normal operation, when the controller accesses a track marked as defective, it automatically invokes a seek to the assigned alternate track. It uses the alternate track as if it were the original data track. This operation is automatic and is invisible to the user, except for the additional time needed to complete the disk operation.

For a device with multiple recording surfaces, the number of alternate tracks on each surface are equal. Therefore, devices with multiple platters set aside entire cylinders for alternate tracks.

One of the alternate cylinders is a diagnostic cylinder. Consequently, on an error-free disk, you must assign a minimum of one alternate cylinder. A Winchester disk in an Intel System 3XX/5XX Series Microcomputer System uses an additional four tracks for permanent bad-track information. Thus, for an error-free disk in this environment, you must assign a minimum of two alternate cylinders.

When an alternate track is assigned, the Operating System returns an E$IO$ALT$ASSIGNED warning message.

If your disk is not error free and you have not specified enough alternate cylinders, the Basic I/O System returns an E$IO$NO$SPARES exception code if the driver cannot find enough alternate tracks.

For flexible diskette drives and tape drives, this parameter is not relevant. Set the value to "0H".

```
(BTI) Bad Track Information [Yes/No]                    YES
```

For Winchester disk drives, this parameter specifies whether the disk controller, when it formats the disk, uses the factory-generated bad-track information stored on the disk. Specify "Yes" if the disk is an Intel-supplied hard disk or a disk containing valid bad-track information. Specify "No" if you are not using Intel-supplied system products or you do not want the disk controller to use the factory-generated bad-track information when formatting the disk.

For information on how to put Bad Track Information on a disk, see the FORMAT command in the *Extended iRMX II Operator's Guide to the Human Interface*.

For flexible diskette drives and tape drives, this parameter is not relevant. Set the value to "No".

```
(SR)  Step Rate [0-0FFH]                                 0H
```

For flexible diskette drives, this parameter specifies the encoded time interval between step pulses as they relate to track-to-track accesses. Refer to the *iSBC 186/224A Peripheral Controller Board User's Guide* for valid step rates.

NOTE

Head load time is not programmable for the iSBC 186/224A controller. Also, the media-change-detect and motor-control options for floppy drives are not user-configurable. The device driver always sets up the controller for media-change polling, even if the drive cannot generate a Ready signal. The driver always sets up the controller for normal motor operation (that is, the motor is turned off after a period of inactivity), even if the drive supports head load/unload capability.

For Winchester disk and tape drives, this parameter is not relevant. Set the value to "0H".

```
(RWC) Reduce Write Current Cylinder [0-0FFFFH]            01D0H
```

For Winchester disk and flexible diskette drives, this parameter specifies the starting cylinder number that reduced write current (RWC) will be enabled for. RWC will be enabled for all cylinders greater than or equal to this cylinder number. Refer to the vendor's reference manual for your drive for more information on the use of RWC. To disable RWC, this value should be set greater than the total number of cylinders (including alternates). Typically this will be set equal to half the total number of cylinders.

For tape drives, this parameter is not relevant. Set the value to "0H".

```
(WPC) Write Precompensation Cylinder [0-0FFFFH]           01D0H
```

For Winchester disk and flexible diskette drives, this parameter specifies the starting cylinder number that write precompensation (WPC) is enabled for. WPC is enabled for all cylinders greater than or equal to this cylinder number. Refer to the vendor's reference manual for the drive for more information on the use of WPC. To disable WPC, this value should be set greater than the total number of cylinders (including alternates). Typically this value will be set equal to half the total number of cylinders.

For tape drives, this parameter is not relevant. Set this value to "0H".

```
(RD)  Reserved Drive [Yes/No]                             NO
```

This parameter specifies whether this drive should be reserved for exclusive use by one host in a multiple-host system. If you respond "Yes", the driver reserves the drive as it performs the Human Interface ATTACHDEVICE command. No other host can access this drive until the attached host detaches it. If the drive is reserved by one host, other hosts will get an error if they try to access it.

If you want this drive to be simultaneously shared by multiple hosts (for example, if this is a read-only disk), specify "No". Also, specify "No" if this is a single-host system.

QUERY SCREEN

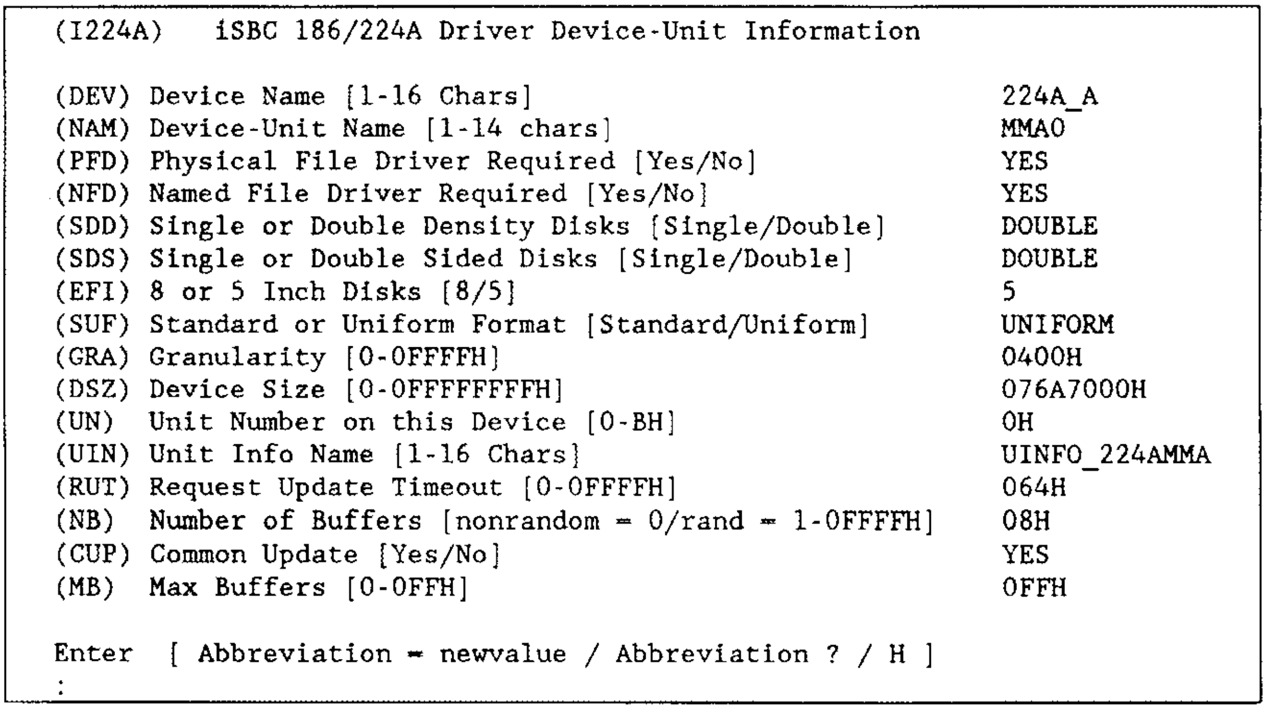
After you complete the "iSBC 186/224A Driver Unit Information" screen, the query screen is displayed. It contains the following line:

```
Do you want any/more iSBC 186/224A UNITs ?
```

Respond "Yes" to this prompt if you need another unit information table for this device.

iSBC 186/224A DRIVER DEVICE-UNIT INFORMATION SCREEN

The ICU uses the information from the following screen to create a device-unit information block (DUIB) for the iSBC 186/224A driver.



```
(I224A)   iSBC 186/224A Driver Device-Unit Information

(DEV) Device Name [1-16 Chars]                             224A_A
(NAM) Device-Unit Name [1-14 chars]                        MMA0
(PFD) Physical File Driver Required [Yes/No]               YES
(NFD) Named File Driver Required [Yes/No]                  YES
(SDD) Single or Double Density Disks [Single/Double]       DOUBLE
(SDS) Single or Double Sided Disks [Single/Double]         DOUBLE
(EFI) 8 or 5 Inch Disks [8/5]                              5
(SUF) Standard or Uniform Format [Standard/Uniform]        UNIFORM
(GRA) Granularity [0-0FFFFH]                               0400H
(DSZ) Device Size [0-0FFFFFFFFH]                           076A7000H
(UN)  Unit Number on this Device [0-BH]                    0H
(UIN) Unit Info Name [1-16 Chars]                          UINFO_224AMMA
(RUT) Request Update Timeout [0-0FFFFH]                    064H
(NB)  Number of Buffers [nonrandom = 0/rand = 1-0FFFFH]    08H
(CUP) Common Update [Yes/No]                               YES
(MB)  Max Buffers [0-0FFH]                                 0FFH

Enter  [ Abbreviation = newvalue / Abbreviation ? / H ]
:
```

```
(DEV) Device Name [1-16 Chars]                             224A_A
```

The name you enter for this parameter must be the same name you entered for the "(DEV) Device Name" parameter on the "iSBC 186/224A Driver" screen. This name provides the logical ICU connection between a driver and all of its DUIBs. Thus, when you delete a device you can delete all the related units and DUIBs at the same time.

```
(NAM) Device-Unit Name [1-14 chars]                    MMA0
```

Use this parameter to specify a name that uniquely identifies this device-unit to the I/O System. The name you specify is the physical name you specify when invoking the Human Interface ATTACHDEVICE command, the BIOS A$PHYSICAL$ATTACH$DEVICE system call, or the EIOS LOGICAL$ATTACH$DEVICE system call. If you want the Extended I/O System to logically attach this device during initialization, the name you specify for this parameter must also be one of the device names you specified on the "HI Logical Names" screen. To support auto boot device recognition, specify the same name in one of the %DEVICE macros when you configure this device in your bootstrap loader (see the *Extended iRMX II Bootstrap Loader Reference Manual* for detailed information).

The ICU allows you to enter from one to fourteen characters. Refer to the *ASM286 Language Reference Manual* for rules regarding this name.

Table 10-17. iSBC® 186/224A DUIB Information for Winchester Drives

DUIB Information						
Model	Device-Unit Name	Size	Gran	Device Size	Unit-Info Name	Num Buffer
Generic Winchester	W0	5	1024	400H	UINFO_224AW	8
CMI5412	CM0	5	1024	0A68000H	UINFO_224ACM	8
CMI5619	CMB0	5	1024	0F9C000H	UINFO_224ACMB	8
or						
Fujitsu M2235	CMB0	5	1024	0F9C000H	UINFO_224ACMB	8
Quantum Q540	QMA0	5	1024	234C000H	UINFO_224AQMA	8
Toshiba MK56FB	TMA0	5	1024	4812000H	UINFO_224ATMA	8
Maxtor XT-1140	MMA0	5	1024	76A7000H	UINFO_224AMMA	8
Maxtor XT-1085	MMB0	5	1024	4698000H	UINFO_224AMMB	8
Seagate	SMA0	5	1024	2AB7000H	UINFO_224ASMA	8

Table 10-18. iSBC® 186/224A DUIB Information for Flexible Disk and Tape Drives

DUIB Information						
Device-Unit Name	Density	Sides	Gran	Device Size	Unit-Info Name	Num Buffer
WF0	Single	Single	128	03E900H	UINFO_224AF	6
WMF0	Double	Double	512	04F800H	UINFO_224AMFDX	6
WMFDY0	Double	Double	512	09F800H	UINFO_224AMFDY	6
WTA0	Single	Single	32256	1400000H	UINFO_224AWT	0

```
(PFD) Physical File Driver Required [Yes/No]              YES
(NFD) Named File Driver Required [Yes/No]                 YES
```

For Winchester disk drives and flexible disk drives, the driver supports both named and physical file drivers. These parameter lines enable you to specify "Yes" to at least one of these parameters.

Two good reasons exist for not changing either "Yes" default value. First, the Human Interface ATTACHDEVICE command, the EIOS LOGICAL$ATTACH$DEVICE system call, and the BIOS A$PHYSICAL$ATTACH$DEVICE system call require you to select the file type at run time. Second, there is no code savings if you specify "Yes" to one parameter and "No" to the other.

For tape drives, specify "Yes" to the "(PFD) Physical File Driver Required" parameter and "No" to the "(NFD) Named File Driver Required" parameter.

```
(SDD) Single or Double Density Disks [Single/Double]      DOUBLE
```

For Winchester disk and tape drives, this parameter is not relevant. Leave this parameter set to the default value "Double".

For flexible diskette drives, this parameter specifies the recording density of the flexible disk. The recording density is either single or double. Table 10-19 lists some of the possible values.

```
(SDS) Single or Double Sided Disks [Single/Double]        DOUBLE
```

For Winchester disk and tape drives, this parameter is not relevant. Leave the parameter at its default setting.

For flexible diskette drives, this parameter specifies the number of recording surfaces on the flexible disk. Table 10-19 shows how your response to this prompt is related to other parameters.

```
(EFI) 8 or 5 Inch Disks [8/5]                             5
```

For Winchester disk and flexible disk drives, this parameter specifies the size of the disk connected to your iSBC 186/224A controller, 8 (for 8-inch disk) or 5 (for 5.25-inch disk). Currently, the iSBC 186/224A supports only 5.25-inch diskettes. Do not change the default setting of "5".

For tape drives, this parameter is not relevant. Leave the parameter at its default setting.

```
(SUF) Standard or Uniform Format [Standard/Uniform]        UNIFORM
```

For tape drives, this parameter is not relevant. Leave the parameter at its default setting.

For Winchester disk drives, set this parameter to "Uniform".

For flexible disk drives, use this parameter to specify the format the controller should expect on track zero. Choose the option "Standard" if you want track zero to be single density with 128-byte sectors (regardless of how the remaining tracks are formatted). Choose the option "Uniform" if you want all tracks (including track 0) on a disk to formatted with the specified granularity.

If you attach the flexible disk drive with a DUIB specifying "standard" format then use the Human Interface FORMAT command to format a disk as a named volume, the command formats track zero with a fixed density (single density) and a fixed sector size (128 bytes). Intel recommends that you use this format. The bootstrap loader supports booting only from "standard" format diskettes.

If you wish to read a disk that has a uniform format, you should use the "Standard/Uniform Format" parameter to designate a "Uniform" format and attach the drive as a physical file. Note that automatic device characteristics recognition does not work unless track zero is single-density with 128-byte sectors.

```
(GRA) Granularity [0-0FFFFH]                               0400H
```

Use this parameter to specify the minimum number of bytes that the device reads or writes in one operation. This value is also called device granularity. For Winchester disk and flexible disk drives, this parameter is the physical sector size. For tape drives this parameter is the record size.

Refer to the *Extended iRMX II Basic I/O System User's Guide* for more information about granularity.

For Winchester disk drives, the vendor of your drive may recommend optimum values for this parameter. Refer to vendor documentation for additional information.

For flexible diskette drives, you can specify one of these granularities: 128, 256, 512, 1024. (You specify volume granularity when you format diskettes.) Table 10-18 shows how the granularity relates to other parameters.

For tape drives, this parameter must be a multiple of 512. Intel recommends a value of "7E00H".

```
(DSZ) Device Size [0-0FFFFFFFFH]                                   076A7000H
```

This parameter specifies the device storage capacity in bytes.

For Winchester and flexible diskette drives, the device size varies with the granularity and sectors per track. Table 10-17 shows the device sizes of several Winchester disks. Table 10-18 shows the relationships between these factors for flexible diskette drives. For generic DUIBs, set this value to the device granularity.

For Winchester disk drives, use the following formula to calculate device size:

DSZ = Granularity * (sectors per track) * (# heads)
* (# cylinders - # alternate cylinders)

For flexible diskette drives, use these formulas to calculate device size:

Uniform flexible diskettes

DSZ = Granularity * (sectors per track) * (2 heads) * (# cylinders)

Standard flexible diskettes

Size first track = (128 bytes/sector) * (16 sectors)
Size other tracks = Granularity * (sectors per track) * (# tracks)

DSZ = (size first track) + (size other tracks)

For tape drives, set to manufacturer's specifications for capacity.

Table 10-19. 5.25-Inch Disk Characteristics

			Device Size			
		Sectors	One Sided		Two Sided	
Gran	Density	per Track	40 Tracks	80 Tracks	40 Tracks	80 Tracks
256	Double	16	1617921	325632	325632	653312
512	Double	8	1617921	325632	325632	653312
1024	Double	4	1617921	325632	325632	653312

```
(UN)  Unit Number on this Device [0-BH]                          0H
```

This parameter specifies the unit number of this device-unit. The unit numbers for each type of device must be set within the following ranges:

Winchester disk drives:	0-3
Flexible disk drives:	4-7
Tape drives:	8-B

```
(UIN) Unit Info Name [1-16 Chars]                       UINFO_224AMMA
```

This parameter specifies the "Unit Info Name" of a unit information table that fills the needs of this DUIB. The following rules apply to the name you choose:

- You must create a unit information table having this "unit info name."
- Each DUIB can point to only one unit information table.
- A particular unit information table can be pointed to by more than one DUIB.

For example, if you used the name "UINFO_224AMMA" when you defined the "Driver Unit Information" screen (NAM = UINFO_224AMMA), you must use the same name for this parameter.

```
(RUT) Request Update Timeout [0-0FFFFH]                          064H
```

For Winchester disk and flexible diskette drives, this parameter specifies the number of clock ticks the driver should wait (during a pause in activity) before updating the attached files on the device.

The values "0H" and "0FFFFH" do not indicate time intervals. A value of "0H" indicates that the driver will never leave any data buffered in memory (eliminating the need to update). A value of "0FFFFH" indicates that updates on this device will occur only when a file is detached. However, the ICU treats your response to this parameter totally separate from the response made to the "Common Update" parameter (you can specify both).

The update capability provided by this parameter differs from the common update capability described in Chapter 9. Unlike the ability to update at fixed periods that is provided by the common update feature, the update capability referenced by the "Request Update Timeout" parameter enables the driver to update based on pauses in activity. If there is a pause in activity on this device, the driver determines how soon the common update would occur and compares that time interval to the request update timeout value. The driver then waits the shorter of the two intervals and updates the attached files on the device. Thus, if you use the common update feature, the time interval you specify for the "Request Update Timeout" should be shorter than the time interval specified for the "Common Update Timeout".

For tape drives, this parameter is not relevant. Set the parameter to "0FFFFH".

```
(NB)  Number of Buffers [nonrandom - 0/rand - 1-0FFFFH]     08H
```

For Winchester disk drives and flexible disk drives, this parameter specifies the number of buffers this device has for blocking and deblocking I/O requests.

The value of "08H" is appropriate for a unit that has a granularity of 1024 bytes. Table 10-17 shows how this value relates to other parameters for Winchester drives. Table 10-18 shows how the value relates for flexible disk drives. Because this is a device which supports random access, do not specify a value of "0H".

The "Number of Buffers" prompt determines the number of Basic I/O System buffers the device uses for I/O. These buffers are general-purpose cache buffers that the Basic I/O System uses to increase I/O performance. The size of each buffer is 32 bytes greater than the device granularity you specified for this device (your response to the "(GRA) Granularity" parameter). The buffers are associated with a device-unit, not with a particular file. The Basic I/O System assigns these buffers to the device-unit when it attaches the device-unit.

Because the Operating System uses memory from the BIOS memory pool to create these buffers, you should increase the values you specified on the "BIOS" screen for both the minimum and maximum memory pool sizes if you increase the default value for the "Number of Buffers" parameter. An explanation of how to calculate memory usage based on your responses to the "(GRA) Granularity" and the "(NB) Number of Buffers" parameter lines is in Chapter 9.

Choosing an optimum number of buffers depends on many factors: how your application works; the speed of your components, your CPU, your memory, and your peripherals; the performance characteristics of your peripherals; and the number of users. Plus, you must trade off optimum performance against the amount of memory you use (because the more buffers you choose, the more memory you need).

Because there are so many factors, the only way that you can obtain the optimum value for your system is by testing different values and fine-tuning your configuration. Intel recommends that you use at least two buffers for this device.

For tape drives, this parameter must be set to "0H".

```
(CUP) Common Update [Yes/No]                          YES
```

For Winchester disk drives and flexible disk drives, specify "Yes" for this parameter if you want the driver to update the attached files on this device at the fixed interval you specified on the "BIOS" screen. Specify "No" if you do not want to update attached files on this device at fixed intervals but prefer to update attached files based only on the time interval you specified for the "Request Update Timeout" parameter. You should not specify "No" for this parameter and "0FFFFH" for the "Request Update Timeout" parameter. See Chapter 9 for information about common update timeout (updating attached files at fixed intervals).

For tape drives, this parameter is not relevant. Set the parameter to "No."

```
(MB)  Max Buffers [0-0FFH]                          0FFH
```

This parameter specifies the maximum number of buffers that the Extended I/O System can allocate for this device's I/O. The default value for this parameter (0FFH) allows the S$OPEN system call to specify the actual number of Extended I/O System buffers. The Operating System takes memory required for these buffers from the calling job's memory pool, so by setting this parameter to "0FFH" you allow the calling job to select the number of buffers based on its own memory pool size. Intel recommends that you use the default value.

QUERY SCREEN

After you have completed the DUIB information screen, the query screen is displayed. It contains the following line:

```
Do you want any/more iSBC 186/224A DUIBs ?
```

Respond "Yes" to this prompt if you plan to use this controller with two peripheral devices that have different characteristics, or peripheral devices that have the same characteristics, but different unit numbers.

While developing your initial systems, you can create as many device-unit information blocks as you want. The number of DUIBs can exceed the number of devices on your system. The particular DUIB associated with the device depends on the physical name you use when attaching it. Once you know that you will never need a particular DUIB, save memory by deleting it from your definition file before you generate your configuration files (refer to Chapter 16 for additional information on generating configuration files).

iSBC® 186/410 DRIVER PARAMETERS

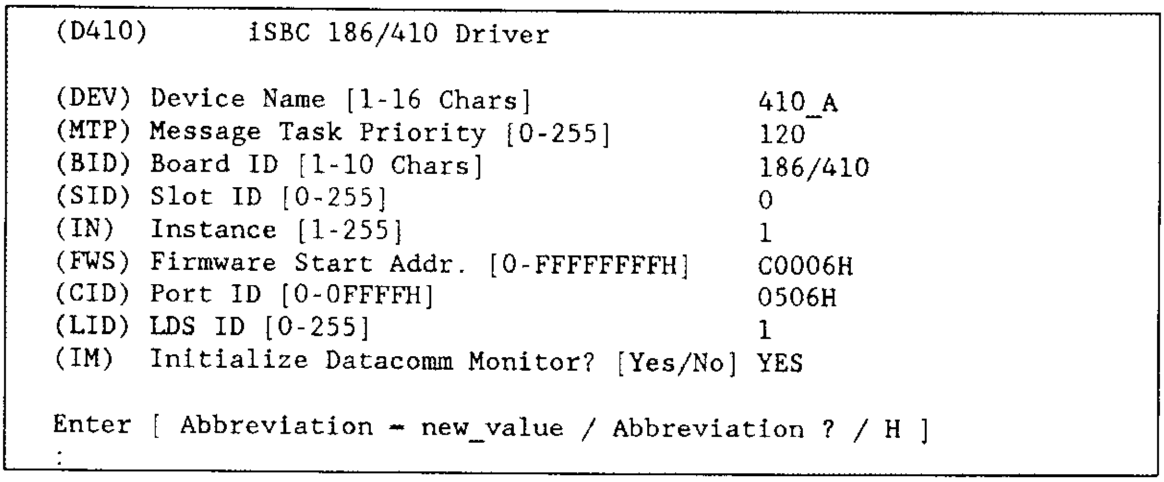
Three screens define the interface between the iSBC 186/410 driver and the I/O System. These screens relate to the three device configuration tables: the device information table, the unit information table, and the device unit information block (DUIB). Refer to the *Extended iRMX II Device Drivers User's Guide* for further information about these tables.

The values shown on the screens in this section are the values you see when you invoke the ICU with the 286100A.def file.

iSBC® 186/410 DRIVER SCREEN

The ICU uses the information from the following screen to create a device information table for the iSBC 186/410 driver. If your system includes more than one iSBC 186/410 controller, use this screen to complete a unique device information table for each iSBC 186/224A controller in your system. You must make the following specifications:

- The number of iSBC 186/410 boards your system will contain in the "(IN) Instance" parameter on the following screen
- One device information table for each device
- One unit information table for each serial line you want to access on each device



```
(D410)      iSBC 186/410 Driver

(DEV) Device Name [1-16 Chars]                 410_A
(MTP) Message Task Priority [0-255]            120
(BID) Board ID [1-10 Chars]                    186/410
(SID) Slot ID [0-255]                          0
(IN)  Instance [1-255]                         1
(FWS) Firmware Start Addr. [0-FFFFFFFFH]       C0006H
(CID) Port ID [0-0FFFFH]                       0506H
(LID) LDS ID [0-255]                           1
(IM)  Initialize Datacomm Monitor? [Yes/No] YES

Enter [ Abbreviation = new_value / Abbreviation ? / H ]
:
```

```
(DEV) Device Name [1-16 Chars]              410_A
```

This parameter provides a means of associating the device and all the units and DUIBs which belong to it. The ICU stores all the screens by type, not by device. This means that all device driver screens, all unit information screens, and all DUIB screens are stored together. The ICU determines which unit information and device-unit information screen relate to a particular device driver by the name you enter in the "(DEV) Device Name" parameter.

```
(MTP) Message Task Priority [0-255]         120
```

This parameter specifies the priority of the task which receives messages from the controller.

```
(BID) Board ID [1-10 Characters]            186/410
```

This parameter specifies the 10-character board name. For the iSBC 186/410 controller, it is the string "186/410". The driver automatically pads this field with ASCII NULL (0) characters.

The board name is found in registers 2-11 of the header record in the iSBC 186/410 board's interconnect space. If your system contains multiple iSBC 186/410 controllers, you can set the "(SID) Slot ID" parameter to "0". The iSBC 186/410 driver then uses the board name and the instance (specified in the "(IN) Instance" parameter) to scan the MULTIBUS II backplane for a particular iSBC 186/410 controller.

```
(SID) Slot ID [0-255]                       0
```

This parameter specifies the MULTIBUS II slot number (as described in your system manual) in which the board resides. A non-zero value for slot ID indicates to the driver that it should expect the device in this slot. A value of zero for this parameter indicates that the driver should locate the board by scanning the MULTIBUS II slots until it finds the board with the specified board ID and instance number.

```
(IN)  Instance [1-255]                        1
```

This parameter, which is significant only if the slot ID is zero, identifies a particular iSBC 186/410 controller in a system containing multiple iSBC 186/410 boards. During initialization the driver calculates the instance by scanning the MULTIBUS II slots in ascending order and sequentially assigning numbers to each iSBC 186/410 controller found. For example, 1 is assigned to the iSBC 186/410 board in the lowest-numbered slot, and 2 to the iSBC 186/410 board in the next-lowest-numbered slot. This method of identifying the board (as opposed to specifying a non-zero slot ID) provides slot independence.

```
(FWS) Firmware Start Addr. [0-FFFFFFFFH]     C0006H
```

This parameter specifies the execution start address of the controller firmware. For the iSBC 186/410 controller, you must use the value "C0006H".

```
(CID) Port ID [0-0FFFFH]                      0506H
```

This parameter specifies the MULTIBUS II port on the iSBC 186/410 controller used to exchange messages with requesting agents. For the iSBC 186/410 board, do not change the default value "0506H".

```
(LID) LDS ID [0-255]                          1
```

For the iSBC 186/410 controller, this value must be 1.

```
(IM)  Initialize Datacomm Monitor? [Yes/No] YES
```

This parameter controls the initialization sequence the driver should perform on the controller. If your system contains one processor board (host), set this parameter to "Yes". If your system contains multiple hosts sharing the iSBC 186/410 controller, set this parameter to "Yes" for only one host. Set this parameter to "No" for the remaining hosts. When the iSBC 186/410 controller is attached, it will be initialized by the host having the "(IM) Initialize Datacomm Monitor" parameter set to "yes".

QUERY SCREEN

After you have completed the "iSBC 186/410 Driver" screen, the query screen is displayed. It contains the following line:

```
Do you want any/more iSBC 186/410 DEVICEs ?
```

Respond "Yes" to this prompt if you want to add more iSBC 186/410 controllers.

iSBC® 186/410 UNIT INFORMATION SCREEN

The ICU uses the information from the following screen to create a unit information table for the iSBC 186/410 driver. Configuring an iSBC 186/410 controller is different from configuring other serial controllers in that you must complete at least one "iSBC 186/410 Unit Information" screen for each of the controller's six serial lines you want to access. In this screen you will define a set of unit characteristics for one serial line.

NOTE

> Some parameters in the "(NCOM) Nucleus Communication Service" screen depend on the number of units for each iSBC 186/410 device. You must increase the value of the "(MST) Max No. of Simultaneous Transactions" parameter by (6 * number of units on the device). Also, increase the "(MSM) Max No. of Simultaneous Messages" parameter value by (15 * number of units on the device). The iRMX II default ICU definition files have "(MSM)" and "(MST)" set for one iSBC 186/410 controller configured with six units.

```
(U410)   iSBC 186/410 Driver Unit Information

(DEV) Device Name [1-16 Chars]                                    410_A
(NAM) Unit Info Name [1-16 Chars]                                 410_A
(LEM) Line Edit Mode [Trans/Normal/Flush]                         NORMAL
(ECH) Echo Mode [Yes/No]                                          YES
(IPC) Input Parity Control [Yes/No]                               NO
(OPC) Output Parity Control [Yes/No]                              NO
(OCC) Output Control in Input [Yes/No]                            YES
(OSC) OSC Controls [Both/In/Out/Neither]                          BOTH
(DUP) Duplex Mode [Full/Half]                                     FULL
(TRM) Terminal Type [CRT/HardCopy]                                CRT
(MC)  Modem Control [Yes/No]                                      NO

(RPC) Read Parity [See Help]    0      (WPC) Write Parity [See Help]    4
(IR)  In Baud Rate [0-65535]    9600   (OR)  Out Baud Rate [0-65535]    0
(SN)  Scroll Number [0-65535]   18     (CL)  Character Length [5-8]     8
(SB)  Stop Bits  [See Help]     0      (REH) Rx Errs [See Help]         3
(XE)  Tx/Rx always enabled      YES    (LN)  Line Number [0-255]        1
(TM)  Tandem Mode [Yes/No]      NO     (MRC) Min Rx Count [1-254]       10
(TO)  Timeout [0-254]           1      (IT)  Idle Time [0-255]          1

Enter [ Abbreviation = new_value / Abbreviation ? / H ]
:
```

```
(DEV) Device Name [1-16 Chars]                              410_A
```

This parameter provides a means of associating the device and all the units and DUIBs which belong to it. The ICU "knows" which unit information and device-unit information screen relate to a particular device by the name you enter in the "(DEV) Device Name" parameter.

```
(NAM) Unit Info Name [1-16 Chars]                           410_A
```

This parameter specifies a unique name for this unit information table. The first character must be an alphabetic character. Refer to the *ASM286 Language Reference Manual* for rules regarding this name.

A DUIB uses the "Unit Info Name" to point to this particular unit information table. When developing your initial systems, you can create unit information tables that are never pointed to by a DUIB. There is no harm in this process during the development stages. However, when you configure your final system, eliminate all unused information tables to save memory.

```
(LEM) Line Edit Mode [Trans/Normal/Flush]             NORMAL
```

This parameter specifies the initial default line editing mode. You must choose from the following three options:

Transparent

Console input is transparent (not line-edited). The Terminal Support Code transmits input to the requesting task exactly as entered at the terminal. Before the Terminal Support Code transmits the input, it accumulates data in a buffer until an operator enters the requested number of characters.

Normal

Console input is line-edited. Edited data accumulates in a buffer until an operator enters a line terminator.

Flush

Console input is not line-edited and the Terminal Support Code transmits input to the requesting task exactly as entered at the terminal. Before the Terminal Support Code transmits the input, it accumulates data in a buffer until it receives an input request. At that time, it transmits the contents of the buffer (or the number of characters requested, if the buffer contains more than that number) to the requesting task. If any characters remain in the buffer, the Terminal Support Code saves the characters for the next input request.

You can alter the value you specify for this parameter at run time if you invoke any of the EIOS or BIOS SPECIAL system calls or send an OSC sequence. OSC sequences are described in the *Extended iRMX II Device Drivers User's Guide*.

```
(ECH) Echo Mode [Yes/No]                              YES
```

Specify "Yes" if you want characters entered into the terminal to be echoed to the terminal's display screen. Otherwise, specify "No".

You can alter the value you specify for this parameter at run time if you invoke any of the EIOS or BIOS SPECIAL system calls or send an OSC sequence. OSC sequences are described in the *Extended iRMX II Device Drivers User's Guide*.

```
(IPC) Input Parity Control [Yes/No]                   NO
```

Specify "Yes" if you want the system to change the parity bit (bit 7) of characters entered into the terminal to zero. Specify "No" if you do not want the system to change bit 7 of the input character.

You can alter the value you specify for this parameter at run time if you invoke any of the EIOS or BIOS SPECIAL system calls or send an OSC sequence. OSC sequences are described in the *Extended iRMX II Device Drivers User's Guide.*

```
(OPC) Output Parity Control [Yes/No]                NO
```

Specify "Yes" if you want the system to change the parity bit (bit 7) of characters being output to the terminal to zero. Specify "No" if you want bit 7 in the output characters to remain unchanged.

You can alter the value you specify for this parameter at run time if you invoke any of the EIOS or BIOS SPECIAL system calls or send an OSC sequence. OSC sequences are described in the *Extended iRMX II Device Drivers User's Guide.*

```
(OCC) Output Control in Input [Yes/No]              YES
```

Specify "Yes" if you want the Terminal Support Code to accept output control characters in the input stream. Specify "No" if you want the Terminal Support Code to ignore output control characters. Control characters are described in the *Extended iRMX II Device Drivers User's Guide.*

You can alter the value you specify for this parameter at run time if you invoke any of the EIOS or BIOS SPECIAL system calls or send an OSC sequence. OSC sequences are described in the *Extended iRMX II Device Drivers User's Guide.*

```
(OSC) OSC Controls [Both/In/Out/Neither]            BOTH
```

This parameter specifies whether the device driver should act upon Operating System Command (OSC) controls when they appear in either the input or the output stream. Choose one of the following options as the initial default value for the device driver:

Both — Act upon OSC control sequences in either input or output stream (from either terminal or program).

Input — Act upon OSC control sequences in input stream only (from terminal and not from program).

Output — Act upon OSC control sequences in output stream only (from program and not from terminal).

Neither — Do not act upon OSC control sequences.

The OSC control sequence, used in communicating from a program or a terminal to the Operating System, is described in the *Extended iRMX II Device Drivers User's Guide*. You can alter the value you specify for this parameter at run time if you invoke any of the EIOS or BIOS SPECIAL system calls or send an OSC sequence.

```
(DUP) Duplex Mode [Full/Half]                         FULL
```

This parameter specifies the line protocol mode for a terminal. Choose either full-duplex or half-duplex.

When line protocol mode is full-duplex, the terminal driver concurrently handles input to and output from the terminal. If you specified both echo mode and full-duplex, the terminal driver echoes each character. When the line protocol mode is half-duplex, there can be input to and output from the terminal, but not concurrently. If you specify both echo mode and half-duplex, the terminal, not the terminal driver, must echo each character.

You can alter the value you specify for this parameter at run time if you invoke any of the EIOS or BIOS SPECIAL system calls or send an OSC sequence. OSC sequences are described in the *Extended iRMX II Device Drivers User's Guide*.

```
(TRM) Terminal Type [CRT/HardCopy]                    CRT
```

This parameter specifies how your terminal supports the rubout function. Respond "CRT" if your terminal can backspace and leave a blank character on the screen for each character "rubbed out." Respond "Hard Copy" if your terminal cannot backspace and leave a blank character on previously displayed characters.

You can alter the value you specify for this parameter at run time if you invoke any of the EIOS or BIOS SPECIAL system calls or send an OSC sequence. OSC sequences are described in the *Extended iRMX II Device Drivers User's Guide*.

```
(MC)  Modem Control [Yes/No]                          NO
```

Specify "Yes" to establish an initial modem-based link between a task and a terminal. Specify "No" if your terminal is not connected to a modem.

If you specify "Yes" to this parameter, set the "OSC Controls" parameter to either "Both" or "Out".

You can alter the value you specify for this parameter at run time if you invoke any of the EIOS or BIOS SPECIAL system calls or send an OSC sequence.

```
 (RPC) Read Parity [See Help]                          0
```

This parameter specifies how the hardware in your system will handle parity checking on input. Specify one of the following four values:

0 Ignore parity checking and set the input parity bit (bit 7) to zero.

1 Ignore parity checking and do not change the parity bit (bit 7).

A response of "0" or "1" specifies 8-bit data with no parity.

2 The driver expects even parity on input and sets the parity bit (bit 7) to 0 on meeting this condition.

3 Driver expects odd parity and sets the parity bit (bit 7) to 0 on meeting this condition.

A response of "2" or "3" specifies 5-, 6-, or 7-bit data with parity. The following error conditions can occur: incorrect parity received, a framing error (receive stop bit is zero), or an overrun error (a new character has been received before the interrupt routine for character processing has completed). The driver handles the erroneous characters according to the mode you specify in the "(REH) RX Errs" parameter.

NOTE

Because this parameter specifies how the hardware deals with parity, a response of "0" voids a "No" response to the software parity checking parameter "(IPC) Input Parity Checking".

If you set "(RPC) Read Parity Checking" to "2" or "3" and you set "(IPC) Input Parity Checking" to "Yes", your application will not be able to detect a transmission error.

You can alter the value you specify for this parameter at run time if you invoke any of the EIOS or BIOS SPECIAL system calls or send an OSC sequence. OSC sequences are described in the *Extended iRMX II Device Drivers User's Guide*.

```
 (WPC) Write Parity [See Help]                         4
```

This parameter specifies how the hardware in your system will handle parity on output. Specify one of the following five values:

0 Set the output parity bit (bit 7) to zero.

1 Set the output parity bit (bit 7) to one.

Setting this parameter to "0" or "1", makes the LINK$PARAMETER field of the Terminal Support Code's UNIT$DATA structure invalid; therefore, the I/O characteristics specified in the LINK$PARAMETER will not be in effect. The I/O characteristics for this channel will be 8 bits per character and 1 stop bit (in hardware) with no parity. You cannot use the "(SB) Stop Bits" parameter to set the number of stop bits. Both transmitter and receiver are always enabled.

2-3 Setting this parameter to "2" or "3" specifies 5-, 6-, or 7-bit data with parity. If you specify "2" or "3", you must set the "(RPC) Read Parity" parameter to the same value. You can specify the number of stop bits in the "(SB) Stop Bits" parameter. You can specify whether this unit's transmitter and receiver are always enabled in the "(XE) Tx/Rx Always Enabled" parameter.

Setting this value to "2" specifies even parity; setting it to "3" specifies odd parity.

4 Do not change the output parity bit. There are 5,6, or 7 bits per character with no parity. You can specify the number of stop bits in the "(SB) Stop Bits" parameter and enabling the transmitter and receiver in the "(XE) Tx/Rx Always Enabled" parameter.

NOTE

Because this parameter specifies how the hardware deals with parity, a response of "0" voids a "No" response to the software parity checking parameter "(OPC) Output Parity Checking".

You can alter the value you specify for this parameter at run time if you invoke any of the EIOS or BIOS SPECIAL system calls or send an OSC sequence. OSC sequences are described in the *Extended iRMX II Device Drivers User's Guide*.

```
(IR)  In Baud Rate [0-65535]                        9600
```

This parameter specifies the initial input baud rate of this terminal. If you specify an invalid value, the driver automatically converts this value to the next higher valid baud rate. Specify a value of one if you want the controller to ascertain the initial baud rate automatically. For the iSBC 186/410 controller, the output baud rate value you specify for the "(OR) Out Baud Rate" parameter must be the same value you specify for this parameter.

```
(OR)  Out Baud Rate [0-65535]                 0
```

This parameter specifies the initial output baud rate of this terminal. If you specify an invalid value, the driver automatically converts this value to the next higher valid baud rate. For the iSBC 186/410 controller, set this parameter equal to the input baud rate you specify for the "(IR) In Baud Rate" parameter. If you specify "0" or "1", the Operating System assumes the output baud rate is the value you specified for the input baud rate.

```
(SN)  Scroll Number [0-65535]                 18
```

This parameter specifies the number of lines to scroll when an operator enters the scrolling output control character (Control-W is the default). Typical values should be from 10 to 24. The default value is 18 decimal.

You can alter the value you specify for this parameter at run time if you invoke any of the EIOS or BIOS SPECIAL system calls or send an OSC sequence. OSC sequences are described in the *Extended iRMX II Device Drivers User's Guide.*

```
(CL)  Character Length [5-8]                  8
```

This parameter specifies the length of the character in bits. Valid values are 5, 6, 7, or 8. This parameter will be ignored if your response to the "(WPC) Write Parity" parameter on this screen is either "0" or "1".

```
(SB)  Stop Bits  [See Help]                   0
```

This parameter specifies the stop bits. Enter a 0, 1 or 2 corresponding to 1, 1.5, or 2 stop bits respectively. This parameter will be ignored if your response to the "(WPC) Write Parity" parameter on this screen is either "0" or "1".

```
(REH) Rx Errs [See Help]                      3
```

This parameter specifies the method of handling characters received with an error. Values and their meanings are as follows:

Value	Meaning
0	Characters received with an error are replaced by an ASCII NULL (0) character.
1	Characters received with an error are discarded.
2	Characters received with an error are prefixed with "0FFH 0H".
3	Characters received with an error have their most significant bit set to one.

Note that this parameter is ignored if your response to the "(WPC) Write Parity" parameter on this screen is either "0" or "1".

```
(XE)  Tx/Rx always enabled                              YES
```

This parameter specifies whether or not this unit's transmitter and receiver are always enabled. Enter "Yes" to have this unit's transmitter and receiver always enabled. To enable this unit's transmitter only when CTS is active and to enable this unit's receiver only when CD is active, enter "No".

Note that this parameter is ignored if your response to the "(WPC) Write Parity" parameter on this screen is either "0" or "1".

```
(LN)  Line Number [0-255]                               1
```

This parameter specifies the number of the serial line on the controller corresponding to this unit. Valid line numbers are zero through five. Because the line number for this unit is a field in the unit information table, you must create a separate unit information table for each serial line you configure.

```
(TM)  Tandem Mode [Yes/No]                              NO
```

This parameter specifies whether tandem mode (input flow control) should be enabled or disabled at attach time. Specify "Yes" to enable tandem mode. Specify "No" to disable tandem mode. You can alter the value you specify for this parameter at run time if you invoke any of the EIOS or BIOS SPECIAL system calls or send an OSC sequence. OSC sequences are described in the *Extended iRMX II Device Drivers User's Guide*.

```
(MRC) Min Rx Count [1-254]                              10
```

This parameter specifies the minimum number of characters that should be received by the controller before a message is sent to the host. A large value for this parameter minimizes the number of messages sent to the host.

```
(TO)  Timeout [0-254]                                   1
```

This parameter and the "(IT) Idle Time" parameter can be set to ensure your iSBC 186/410 controller will not wait indefinitely to receive the minimum number of characters you specified in the "(MRC)" parameter before sending the host a message. This parameter specifies a time in 10-millisecond units. When this time limit has elapsed, the iSBC 186/410 controller adjusts the minimum number of characters to one. If at least one character has been received, a message is sent to the host.

```
(IT)  Idle Time [0-255]                                 1
```

This parameter and the "(TO) Timeout" parameter can be set to ensure your iSBC 186/410 controller will not wait indefinitely to receive the minimum number of characters you specified in the "(MRC)" parameter before sending the host a message. This parameter specifies the time (in 10-millisecond units) that the channel should be idle, before the iSBC 186/410 controller adjusts the "(MRC) Min Rx Count" value to 1. If the channel remains idle for the specified time and at least one character has been received, a message is sent to the host.

If you specify both "(TO) Timeout" and "(IT) Idle Time" the iSBC 186/410 controller will wait for both conditions to be satisfied before sending the host a message.

QUERY SCREEN

After you have completed the "iSBC 186/410 Driver Unit Information" screen, the query screen is displayed. It contains only one line:

```
Do you want any/more iSBC 186/410 UNITs ?
```

Respond "Yes" to this prompt if you need to define another unit information table for this device. If you are configuring more than one serial line, you must create a unique unit information table for each line.

iSBC® 186/410 DEVICE-UNIT INFORMATION SCREEN

The ICU uses the information from the following screen to create a device unit information block (DUIB) for the iSBC 186/410 driver.

```
(I410)   iSBC 186/410 Driver Device-Unit Information

(DEV) Device Name [1-16 Chars]               410_A
(NAM) Device-Unit Name [1-14 Chars]          T410_0
(UN)  Unit Number on this Device [0-0FFH]    01H
(UIN) Unit Info Name [1-16 Chars]            UINFO_410_0
(MB)  Max Buffers [0-0FFH]                   0H

Enter [ Abbreviation = new_value / Abbreviation ? / H ]
:
```

```
(DEV) Device Name [1-16 Chars]               410_A
```

The name you enter for this parameter must be the same name you entered in the "(DEV) Device Name" parameter on the "iSBC 186/410 Driver" screen. This name provides the logical ICU connection between a device and all of its DUIBs. Thus, when you delete a device you can delete all the related units and DUIBs at the same time.

```
(NAM) Device-Unit Name [1-14 Chars]          T410_0
```

This parameter specifies a name that uniquely identifies this device-unit to the I/O System. The name you specify for this parameter is the physical name you specify when invoking the Human Interface ATTACHDEVICE command, the BIOS A$PHYSICAL$ATTACH$DEVICE system call, or the EIOS LOGICAL$ATTACH$DEVICE system call. If you want the Extended I/O System to logically attach this device during initialization, specify this name as one of the device names in the "HI Logical Names" screen.

The ICU allows you to enter from one to fourteen characters. Refer to the *ASM286 Language Reference Manual* for rules regarding this name.

```
(UN)  Unit Number on this Device [0-0FFH]    01H
```

This parameter specifies the unit number of this device-unit. The unit numbers for the device begin with zero and increase sequentially. Each of the six serial lines on the board is a single unit.

```
(UIN) Unit Info Name [1-16 Chars]            UINFO_410_0
```

This parameter specifies the "Unit Info Name" of a unit information table that fills the needs of this DUIB. The following rules apply to the name you choose:

- You must create a unit information table having this "unit info name."
- Each DUIB can point to only one unit information table.
- To access a serial line with different parameters at different times, you must create one unit information table pointed to by one DUIB for each set of parameters.

```
(MB)  Max Buffers [0-0FFH]                    0H
```

This parameter specifies the maximum number of buffers that the Extended I/O System can allocate for this device's I/O. If you plan to use this terminal for input and output, do not change the default value. If this will be an output only terminal, you can specify a small non-zero value, but note that the EIOS will delay output until either the buffer is full or you close the connection to the device.

QUERY SCREEN

After you complete the "iSBC 186/410 Driver Unit Information" screen, the query screen is displayed. It contains the following line:

```
Do you want any/more iSBC 186/410 DUIBs ?
```

Respond "Yes" to this prompt if you plan to use the iSBC 186/410 driver with a single device-unit having more than one set of default characteristics or with two device-units having the same characteristics, but different unit numbers.

If you want to access a serial line with multiple sets of default characteristics, you must create multiple unit information table/DUIB pairs (one pair for each set of characteristics you want this line to contain). Default characteristics are those contained in the unit information table/DUIB pair used when the device is attached. Regardless of the chosen default characteristics, you can change Terminal Support Code connection mode and terminal mode parameters after you attach a line by using EIOS or BIOS SPECIAL system calls or OSC sequences.

While developing your initial systems, you can create as many device-unit information blocks as you want. The number of DUIBs can exceed the number of device-units on your system. The particular DUIB associated with the device-unit depends on the physical name you use when attaching it. Once you know that you will never need a particular DUIB, save memory by deleting it from your definition file before you generate your configuration files.

User Devices

The previous pages in this chapter discuss the Intel-supplied device drivers that this ICU supports. The ICU also provides two methods for including user-written device drivers in your system. The first method is to use the User Device Support Utility (UDS) to modify the ICU so that it supports your device driver automatically. The second method is to add your driver as a user-written driver, without first modifying the ICU. If you are upgrading from either an iRMX I Operating System or iRMX II.1, you should be familiar with the second method and may wish to continue using it as before. However, if you are a new user or are adding a new user-written driver, Intel recommends you use the UDS.

The UDS is a program that allows you to modify the ICU to automatically support user-written device drivers. With the UDS you can add screens to the ICU so that configuring your device driver is simply a matter of running the ICU and answering the appropriate questions. The UDS utility lets you add device information, unit information, and device-unit information. You can then configure the new screens into your system using the ICU. For more information about the UDS, refer to the *Extended iRMX II Device Drivers User's Guide*.

This section discusses the screens needed for both methods. To add a user-written device driver without modifying the ICU, respond "Yes" to the prompt "Do you have any/more device drivers not supported by this ICU ?" shown on the query screen. Otherwise, respond with a carriage return or a "No".

When you respond "Yes" to the query, the ICU displays the following screen.

```
(USERD)         User Devices
(OPN) Random Access Object Code Path Name [1-45 Chars/NONE]
                                                                       NONE
(TOP) Terminal Object Code Path Name [1-45 Chars/NONE]
                                                                       NONE
(DPN) DUIB Source Code Path Name [1-45 Chars/NONE]
                                                                       NONE
(DUP) Random Access Device and Unit Source Code Path Name [1-45 Chars/NONE]
                                                                       NONE
(TUP) Terminal Device and Unit Source Code Path Name [1-45 Chars/NONE]
                                                                       NONE
(ND)  Number of User Defined Devices [0-0FFH]              0H
(NDU) Number of User Defined Device-Units [0-0FFH]         0H

            Terminal Device and Unit Information Names [1-16 Chars]
(N01) NONE     (N02) NONE     (N03) NONE
(N04) NONE     (N05) NONE     (N06) NONE
(N07) NONE     (N08) NONE     (N09) NONE
(N10) NONE     (N11) NONE     (N12) NONE
(N13) NONE     (N14) NONE     (N15) NONE
(N16) NONE     (N17) NONE     (N18) NONE

Enter [ Abbreviation = new_value / Abbreviation ? / H ]
:
```

```
(OPN) Object Code Path Name [1-45 Chars/NONE]
```

This parameter lets you specify the pathname for the object code (the result of running PL/M286 or ASM286 then BND286 and placing the object modules in a library) of your user-written device drivers. Entering the word "None" indicates that the object code for this device is provided by Intel in the I/O system driver library.

The *Extended iRMX II Device Drivers User's Guide* describes the steps necessary to bind a user-written device driver to the I/O system. These steps apply to both user-written code and Intel-supplied code. Unless the documentation that describes the Intel-supplied code indicates that some of these steps have already been taken for you, take all the steps in the order described.

```
(DPN) DUIB Source Code Path Name [1-45 Chars]
```

This parameter specifies the pathname for the include file containing user-written Device Unit Information Blocks (DUIBs). If you are adding an Intel-supplied user-written device driver such as the iSBX 275 or the iSBC 186/78A device driver, use the pathname described in the appropriate reference documentation.

Custom-written device drivers (those not supported by this ICU) must have user-written source DUIBs, device information tables, and unit information tables in order for the driver to properly interface with the Operating System. The ICU assembles these modules along with the DUIBs, device information tables, and unit information tables for ICU-supported device drivers.

External declarations for code you have written to support this device driver must be contained in the device information include file.

The source code for any user-written DUIBs must contain the correct device numbers and device-unit numbers for your devices. The *Extended iRMX II Device Drivers User's Guide* describes how you can ascertain the correct numbers for your system.

```
    (DUP) Device and Unit Source Code Path Name [1-45 Chars]
```

This parameter specifies the pathname for the include file contains the user-written device information tables or unit information tables. If you are adding an Intel-supplied user-written device driver such as the iSBX 275 or the iSBC 186/78A device driver, use the pathname described in the appropriate reference documentation.

Custom-written device drivers (those not supported by this ICU) must have user-written source DUIBs, device information tables, and unit information tables in order for the driver to properly interface with the Operating System. The ICU assembles these modules along with the DUIBs, device information tables, and unit information tables for ICU-supported device drivers.

External declarations for code you have written to support this device driver must be contained in the device information include file.

```
    (ND)  Number of User Defined Devices [0-0FFH]     01H
```

This parameter specifies the number of user-written devices that are in your system. The value you specify is the number of different device numbers that you used in the DUIBs that you defined in your DUIB include file. Row A of Figure 10-1 shows the device numbering of a sample system.

```
    (NDU) Number of User Defined Device-Units [0-0FFH]    01H
```

This parameter specifies the number of user-written device-units that are in your system. The value you specify is the number of different device-unit numbers that you used in the DUIBs that you defined in your DUIB include file. Row B of Figure 10-1 shows the device numbering of a sample system.

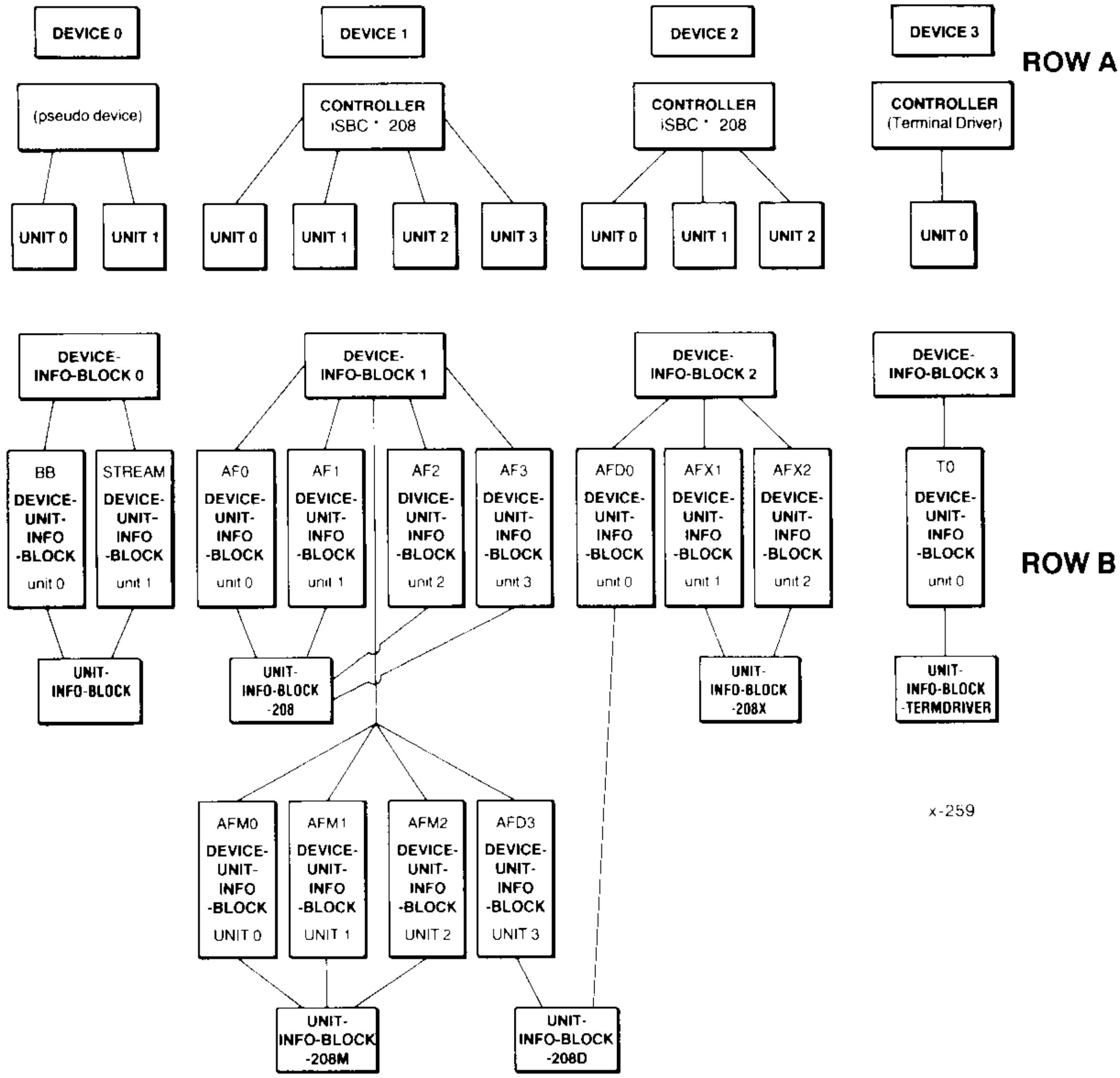
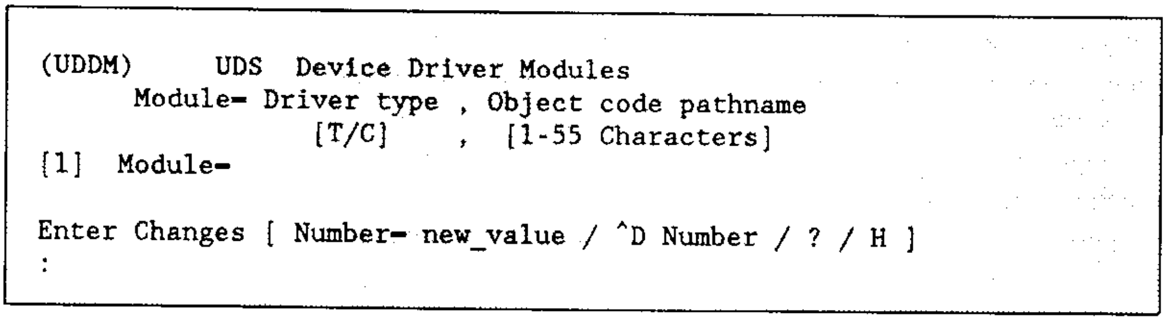

DEVICE 0
DEVICE 1
DEVICE 2
DEVICE 3
ROW A
(pseudo device)
CONTROLLER iSBC * 208
CONTROLLER iSBC * 208
CONTROLLER (Terminal Driver)
UNIT 0
UNIT 1
UNIT 0
UNIT 1
UNIT 2
UNIT 3
UNIT 0
UNIT 1
UNIT 2
UNIT 0
DEVICE-INFO-BLOCK 0
DEVICE-INFO-BLOCK 1
DEVICE-INFO-BLOCK 2
DEVICE-INFO-BLOCK 3
BB DEVICE-UNIT-INFO-BLOCK unit 0
STREAM DEVICE-UNIT-INFO-BLOCK unit 1
AF0 DEVICE-UNIT-INFO-BLOCK unit 0
AF1 DEVICE-UNIT-INFO-BLOCK unit 1
AF2 DIVICE-UNIT-INFO-BLOCK unit 2
AF3 DEVICE-UNIT-INFO-BLOCK unit 3
AFD0 DEVICE-UNIT-INFO-BLOCK unit 0
AFX1 DEVICE-UNIT-INFO-BLOCK unit 1
AFX2 DEVICE-UNIT-INFO-BLOCK unit 2
T0 DEVICE-UNIT-INFO-BLOCK unit 0
ROW B
UNIT-INFO-BLOCK
UNIT-INFO-BLOCK-208
UNIT-INFO-BLOCK-208X
UNIT-INFO-BLOCK-TERMDRIVER
AFM0 DEVICE-UNIT-INFO-BLOCK UNIT 0
AFM1 DEVICE-UNIT-INFO-BLOCK UNIT 1
AFM2 DEVICE-UNIT-INFO-BLOCK UNIT 2
AFD3 DEVICE-UNIT-INFO-BLOCK UNIT 3
UNIT-INFO-BLOCK-208M
UNIT-INFO-BLOCK-208D
x-259

Figure 10-1. Device Numbering

UDS Device Drivers

The "UDS Device Driver Modules" screen is displayed after the "User Devices" screen. If you are not adding user-written devices with the "User Devices" screen, the "UDS Device Driver Modules" screen appears when you respond "No" to the prompt "Do you have any/more device drivers not supported by this ICU ?"



```
(UDDM)      UDS  Device Driver Modules
      Module= Driver type , Object code pathname
                 [T/C]     ,  [1-55 Characters]
[1]  Module=

Enter Changes [ Number= new_value / ^D Number / ? / H ]
:
```

This screen lets you specify the pathnames for the object code of the User device drivers added by the UDS utility.

For purposes of configuring the system, the order of your UDS module pathnames is not significant.

CHAPTER 11
SYSTEM DEBUGGER PARAMETERS

11.1 INTRODUCTION

The iRMX II System Debugger (SDB) is one of the debugging aids supplied by the iRMX II Operating System. It extends the capabilities of the iSDM System Debug Monitor. To use this feature of the Operating System, respond "Yes" to the System Debugger Parameter on the "Sub-systems" Screen.

This chapter describes how to select the appropriate parameter value for the single parameter of the System Debugger screen of the ICU.

11.2 THE SYSTEM DEBUGGER PARAMETER

The SDB screen is as follows:

```
(SDB)        System Debugger

(SLV) SDB Interrupt Level [Encoded Level/NONE = 0FFH]  018H

Enter  [ Abbreviation = new_value / Abbreviation ? / H ]
:
```

Use the "SDB Interrupt Level" parameter to specify the interrupt level, if any, that your system uses to invoke the SDB.

There are two ways to invoke the SDB. One way, which does not involve interrupts, is the Human Interface DEBUG command. If you plan to invoke the DEBUG command, use the default value, 018H, for this parameter.

The other way to invoke the SDB is to press a button that is physically attached to an interrupt level on your CPU board. If you plan to invoke the SDB by pressing an interrupt button, you must specify the interrupt level associated with the interrupt button. If the desired level is "NMI" your response to this parameter should be "0FFH", and the "(NMI) Non-maskable Interrupt" parameter on the Nucleus Screen must be set to "SDB".

The *iRMX II Hardware and Software Installation Guide* provides further information.

CHAPTER 12
NUCLEUS PARAMETERS

12.1 INTRODUCTION

This chapter discusses how to respond to the prompts that appear on the "Nucleus" and "Nucleus Comm Service" screens. The "Nucleus" screen parameters deal with configuration decisions that affect only the local iRMX II Operating System and the defaults shown are from the 28612.def definition file. The "Nucleus Communication Service" screen deals with configuration decisions that relate to MULTIBUS II systems only and the defaults shown are from the 386100.def definition file.

12.2 NUCLEUS SCREEN

The following screen provides the ICU with information about the Nucleus.

```
(NUC)       Nucleus

(NGE) Number of GDT Entries [440 - 8190]                    2000
(NIE) Number of IDT Entries [0-256]                         128
(PV)  Parameter Validation [Yes/No]                         REQ
(ROD) Root Object Directory Size [0 - 3840]                 50
(DEH) Default Exception Handler [Yes/No/Job/SDB/User]       SDB
(NMI) NMI Exception Handler [Yes/No/Ignore/Job/SDM/User]    IGNORE
(NEB) NMI Enable Byte [0-255]                               4
(STK) Exception Handler for Stack Exception, Bad TSS and
              Double Fault  [Yes/No/Job/SDB]                JOB
(NEH) Name of Ex Handler Object Module [1-55 Chars]

(EM)  Exception Mode [Never/Program/Environ/All]            NEVER
(LSE) Low GDT/LDT Slot Excluded from FSM [440-8189/NONE=0]  0
(HSE) High GDT/LDT Slot Excluded from FSM [440-8189/NONE=0] 0
(RRP) Round Robin Priority Threshold [0-255]                140
(RRT) Round Robin Time Quota [0-255]                        5
(RIE) Report Initialization Errors [Yes/No]                 YES
(MCE) Maximum Data Chain Elements [0-0FFFFH]                080H
(CS)  Nucleus Communication Service [Yes/No]                NO

Enter  [ Abbreviation = new_value / Abbreviation ? / H ]
:
```

```
(NGE)  Number of GDT Entries [440 - 8189]          2000
```

This parameter allows you to define the number of objects in your system. In the iRMX II Operating System each object must have a descriptor in the global descriptor table (GDT). Therefore, the maximum number of objects available is 8K, the maximum number of slots available in the GDT. Slots 0-439 are reserved for Intel.

```
(NIE)  Number of IDT Entries [0-256]                128
```

This parameter defines the number of entries in the IDT. The maximum number of entries is 256. If your application system has only a master 8259A PIC and uses no software interrupts, you need a maximum of 40H (64 decimal) entries. If your system includes slave PICs and uses no software interrupts, you need a maximum of 80H (128 decimal) entries.

```
 (PV)  Parameter Validation [Yes/No]                    REQ
```

Parameter validation consists of an optional set of routines that validate system call parameters. Whenever a task invokes a system call, these routines ensure that:

- the value of each object parameter is reasonable
- the system uses correct object types whenever it expects to manipulate an object

Parameter validation provides a very important safeguard while developing software. If you do not include parameter validation, erroneous parameters go undetected and can cause some undefined and possibly catastrophic results.

The response that you make to the "Parameter Validation" parameter on the "Nucleus" screen indicates whether you want system-level parameter validation. The term "system-level" is appropriate since you must enable parameter validation at this level (on the "Nucleus" screen) if you wish to enable parameter validation in any I/O job or any user job (first-level job).

Once you enable system-level parameter validation, you can selectively enable parameter validation on an I/O-job or user-job basis. If a job is to be created during system initialization, use the "User Jobs" screen or the "I/O Jobs" screen to specify parameter validation for that particular job. For all other situations, use either the Nucleus RQE$CREATE$JOB system call for user jobs or the Extended I/O System RQE$CREATE$IO$JOB system call for I/O jobs. Refer to the *iRMX II Nucleus System Calls Reference Manual* and the *iRMX II Extended I/O System Calls Reference Manual* for additional information. Table 12-1 shows the relationship between system-level and job-level validation support in terms of code savings and performance.

To remove parameter validation from your system, specify "No" to the "Parameter Validation" prompt on the "Nucleus" screen, the "User Jobs" and "I/O Jobs" screens. Specifying "No" on either the "User Jobs" or the "I/O Jobs" screen eliminates the capability to use parameter validation on a job created during initialization. However, a reduction in the code size does not occur until you specify "No" to the system-level parameter validation prompt on the "Nucleus" screen.

If your system software includes the Basic I/O System, you must include parameter validation. If your software does not include the Basic I/O System, it is recommended that you include parameter validation during development stages. Then, once you have debugged your code, you can remove parameter validation to improve performance and reduce code size.

Table 12-1. System-Level and Job-Level Parameter Validation

System-level parameter Validation	Included		Excluded	
Job-level parameter Validation	Included	Excluded	Included	Excluded
Is parameter validation performed for this job?	yes	no	no	no
Does the system realize a code savings?	no	no	yes	yes
Does the job realize a performance improvement?	no	yes	yes	yes

```
 (ROD) Root Object Directory Size [0 - 3840]           50
```

This parameter specifies the maximum number of entries in the root job's object directory. The value of zero is an acceptable value if you do not catalog entries in the root job's object directory and your application does not include any subsystems. Use the default value if your application requires a subsystem and you do not catalog entries in the root job's object directory. If you intend to catalog entries in the root job's object directory, add the maximum number of entries that you would catalog to the default size and specify the sum as the root job's object directory size.

When you calculate the maximum number of entries in the root job's object directory, keep in mind the needs of the Human Interface and the Extended I/O System. Both the Extended I/O System (EIOS) and the Human Interface catalog objects in the Root Job's object directory. The EIOS catalogs the logical name of each logical name you specify on the "Logical Names" screen in the root job's object directory. The Human Interface catalogs system-wide logical names in the root job's object directory. You specify these logical names on the "Prefix" screen. Also, if your system includes the Human Interface and you use ATTACHDEVICE to create logical names for devices, the Operating System catalogs the logical names in the root job's object directory.

```
 (DEH) Default Exception Handler [Yes/No/Job/SDB/User]    SDB
```

This parameter specifies one of the five options for the "Default Exception Handler" parameter line. Your options are as follows:

- If you specify "Yes", a system default exception handler is included in your system automatically. This exception handler deletes any task that causes an exceptional condition to occur.

- If you specify "No", your system will use the alternative system exception handler. This exception handler suspends rather than deletes any task that causes an exceptional condition to occur.
- If you specify "Job", the job whose task caused the exception is deleted. Control is passed to the SDB and not to the user job's exception handler when a hardware exception occurs.
- If you specify "SDB", your system will use the iRMX II System Debugger as the system default exception handler. This exception handler causes all hardware exceptions to be handled by the monitor, **not** the iRMX II Operating system. Therefore, it is not possible to regain control of these exceptions dynamically using SET$EXCEPTION$HANDLER. If your definition of a user job includes an exception handler, control is transferred to the System Debugger. If you enter "SDB" in this parameter and did not configure SDB on the "Sub-system" screen, the ICU displays a warning message.
- If you specify "User", your system expects a user-supplied exception handler to function as the system default exception handler. If you choose this option, you must create your own exception handler and designate it to be a PUBLIC procedure having the name RQSYSEX. Users wanting to write their own exception handlers should refer to the discussion on user exception handlers in the *iRMX II Nucleus User's Guide*.

```
 (NMI) Exception Handler [Yes/No/Ignore/Job/SDM/User]   IGNORE
```

This parameter specifies one of the six options for the NMI (Non-maskable Interrupt) Exception Handler. Your options are as follows:

- If you specify "Yes", a system NMI exception handler is automatically included in your system. Any task that causes an NMI exception is deleted.
- If you specify "No", the system uses an NMI exception handler that suspends rather than deletes any task that causes an NMI exception.
- If you enter "Ignore", the system ignores NMI exceptions. This can be useful for handling memory ECC errors.
- If you specify "Job", the system supplies an NMI exception handler to delete the job whose task causes an NMI exception.
- If you specify "SDM", the system breaks to the monitor when an NMI exception occurs.
- If you specify "User", the system expects a user-supplied exception handler to handle NMI exceptions. If you choose this option, you must create your own NMI exception handler and enter the object file name in the "Name of Exception Handler Object Module" parameter.

```
(NEB) NMI Enable Byte [0-255]                                   4
```

This parameter applies only to MULTIBUS II systems. it is ignored for MULTIBUS I systems.

If the NMI Exception Handler parameter is **not set** to "ignore" and the Initialize On-Board Functions parameter on the Hardware screen is set to "4", this value is written to the "NMI Enable" interconnect register. See the hardware reference manual for your processor board for the meanings of the bit encoding for your board.

```
(STK) Exception Handler for Stack Exception, Bad TSS and
             Double Fault [Yes/No/Job/SDB]                      JOB
```

This parameter specifies one of the following four options for the stack fault, double fault and bad TSS exception handler.

- If you specify "Yes", a system exception handler is automatically included in your system that deletes any task that causes such an exception.
- If you specify "No", your system includes an exception handler that suspends rather than deletes any task causing a stack fault, a double fault, or a bad Task State Segment (TSS) exception.
- If you specify "Job", the system supplies an exception handler to delete the job whose task caused the stack fault, double fault or bad TSS exception.
- If you specify "SDB", your system will break to the monitor if a stack fault, double fault or bad TSS exception occurs.

```
(NEH) Name of Ex Handler Object Module [1-55 Chars]
```

You must specify the name of the exception handler object module if you specified "User" for the "Default Exception Handler" or the "NMI Exception Handler" parameter line. If your response to the "DEH" or "NMI" parameter lines is not "User", ignore this parameter line.

The name that you specify should be the pathname (limit of 55 characters) of the object module output by BND286.

```
(EM) Exception Mode [Never/Program/Environ/All]             NEVER
```

This parameter specifies the exception mode of the default system exception handler. Regardless of how you responded to the "Default Exception Handler" parameter line, you must choose one of the following four options:

Never — Control never passes to the exception handler.

Programmer — Control passes to the exception handler on programmer errors only.

Environment — Control passes to the exception handler on environmental conditions only.

All — Control passes to the exception handler on all exceptional conditions.

```
(LSE) Low GDT/LDT Slot Excluded from FSM [440-8189/NONE=0]   0
(HSE) High GDT/LDT Slot Excluded from FSM [440-8189/NONE=0]   0
```

These parameters may be ignored if your bootloadable application requires less than 200 segments. In such a case, BLD286 automatically puts your segments into the system's available slots.

If your bootloadable application requires more than 200 segments, which is unlikely if you use the default PACKING control of BND286, you must enter a value for the "LSE/HSE" parameters to reserve more GDT/LDT slots. If the values you enter here fall within the range of 0-NGE (the number of GDT entries specified in the "NGE" parameter), your system will have fewer objects for the Free Space Manager. Be sure that "HSE" is greater than "LSE".

The slots you reserve with these parameters are not available as gates for any OS extensions you may have.

```
(RRP) Round Robin Priority Threshold [0-255]              140
```

This parameter determines the priority below which tasks will be assigned round-robin scheduling. Only tasks with a priority lower (with a numerically higher number) than the priority entered here will have round-robin scheduling. To use round-robin, the recommended threshold is 140. This threshold value of 140 is compatible with the HI, EIOS, and BIOS task priorities defined in the standard ICU definition files. Entering 255 disables round-robin scheduling.

```
(RRT) Round Robin Time Quota [0-255]                        5
```

This parameter determines the time, in clock ticks, each task is allowed to run before the Nucleus begins rescheduling tasks with round-robin priority (by default, 1 clock tick = 10 milliseconds). For example, assume you enter the default value of 5. After the task has been running for 5 clock ticks (50 milliseconds), the Nucleus checks to see if another task of the same priority is ready and waiting to run. If so, the Nucleus transfers control to the waiting task. If you entered 255 in the "RRP" parameter, this parameter is ignored.

```
 (RIE) Report Initialization Errors [Yes/No]          YES
```

Use this parameter to have initialization errors for all iRMX II layers displayed at the monitor console. If errors occur during system initialization, a hexadecimal code and a mnemonic are displayed at the monitor console indicating the layer that caused the initialization error. This initialization error reporting is either selected for all subsystems or for none of the subsystems.

If you respond "Yes" to this parameter, your application system must contain the monitor. Responding "Yes" for a system without a monitor causes the system to execute random code when initialization errors occur.

If your system does not include a monitor, you can find the exception code returned by the unsuccessful system call in both the AX register and the first WORD of the Nucleus data segment (NUCDAT). The code designating the layer which failed initialization can be found in the second WORD of the Nucleus data segment (NUCDAT). For a list of the error numbers see the *iRMX II Nucleus User's Guide.*

```
 (MCE) Maximum Data Chain Elements [0-0FFFFH]               080H
```

This parameter specifies the maximum number of elements allowed in a single DMA data chain. For information on data chains see the *iRMX II Nucleus User's Guide.*

MULTIBUS I systems do not support data chaining, set this parameter to zero.

```
 (CS)  Nucleus Communication Service [Yes / No]               NO
```

This parameter specifies whether or not the Nucleus Communication Service is to be configured. An MBII system bus type is required for this service. The system bus type is specified at the (HARD) hardware screen.

12.3 NUCLEUS COMMUNICATION SERVICE SCREEN

The following screen provides the ICU with information about how the Nucleus Communications Service is to handle message passing.

```
(NCOM)      Nucleus Communication Service

(PMT) Message Task Priority [0-255]                          128
(PDT) Deletion Task Priority [0-255]                         128
(DPT) Default Number of Port Transactions [0-255]            16
(DHI) Default Host ID [0 = None/1-254]                       0
(VBP) Validate Buffer Parameters [Yes/No]                    YES
(MST) Max No. of Simultaneous Transactions [0-0FFFFH]        040H
(MSM) Max No. of Simultaneous Messages [0-0FFFFH]            080H
(RFT) Receive Fragment Failsafe Timeout [0-0FFFFH]           0A00H
(NTM) Number of Trace Messages [0 - 125]                     0

Enter [ Abbreviation - new_value / Abbreviation ? / H ]
:
```

```
(PMT) Message Task Priority [0-255]                          128
```

This parameter specifies the priority of the Nucleus Communications Service message task. This task runs when a **solicited** message is received. Tasks that deal with I/O typically have priorities set around 128 through 131.

```
(PDT) Deletion Task Priority [0-255]                         128
```

This parameter specifies the Port Deletion Task priority. The management of ports (creation and deletion) should typically have a higher priority than your own applications running on the processor board.

```
(DPT) Default Number of Port Transactions [0-255]            16
```

This parameter specifies a default number of simultaneous port transactions. The value you enter on this line is used when an RQ$CREATE$PORT system call is issued with a zero in the num$trans field.

```
(DHI) Default Host ID [0 = None/1 - 254]                       0
```

Each board in a MULTIBUS II system that needs to pass messages must have a message address (Host ID). This message address can be assigned by you (values 1 through 254); or, if you do not specify a default, the Central Services Module (CSM) will assign each board an ID that is the same as the board's cardcage slot number.

```
(VBP) Validate Buffer Parameters [Yes/No]                      YES
```

This parameter specifies whether or not buffer parameter validation should be performed. Whenever a system call that contains a POINTER to a buffer is issued the Nucleus can check to ensure that the POINTER is valid and that the buffer is large enough to complete the transaction. If the parameter validation fails, an error is returned.

Parameter validation imposes a substantial amount of system overhead in cases, such as data chain buffers, where many POINTERs must be checked.

Note: if you set parameter validation (PV) to NO on the Nucleus Screen you should set this parameter to NO.

```
(MST) Max No. of Simultaneous Transactions [0 - 0FFFFH]        040H
```

This parameter specifies the maximum number of simultaneous transactions in the system. Each transaction requires a buffer in which state information for the transaction is kept. This buffer space is taken from the root job's memory pool. When all of the buffer space is used, an error is returned. The number of transactions permitted should be tuned so as to get a minimal number of errors while not using up buffer space needlessly.

```
(MSM) Max No. of Simultaneous Messages [0 - 0FFFFH]            080H
```

This parameter specifies the maximum number of simultaneous messages in the system. Each message sent, either solicited or unsolicited, has a control part that is placed into a buffer space taken from the root job's memory pool. When all of the buffer space is used, an error is returned. The number of simultaneous messages permitted should be chosen so as to get a minimal number of errors while not using up buffer space needlessly.

```
(RFT) Receive Fragment Failsafe Timeout [0 - 0FFFFH]           0A00H
```

This parameter specifies the Receive Fragment system calls timeout value. This value is specified in 10 millisecond clock ticks.

If an RQ$RECEIVE$FRAGMENT call is issued, this parameter specifies how long the calling task will wait for a message fragment. In the event of a message transmission failure, the callers's task will not wait indefinitely. A value of 0FFFFH means that the task will wait forever.

```
 (NTM) Number of Trace Messages [0 - 125]                        0
```

This parameter specifies the number of trace messages permitted on this board for debugging purposes. Each trace message requires 32 bytes of memory. This memory is taken from the root job. Note that for each trace message specified here, the system allocates 32 bytes for an input buffer and 32 bytes for an output buffer. Therefore, each trace message specified here requires 64 bytes of the root job's memory.

CHAPTER 13 USER JOB PARAMETERS

13.1 INTRODUCTION

This chapter discusses how to respond to the parameters that appear on the "User Jobs" and "User Modules" screens. To get to these screens you must first respond "Yes" to the "Do you need any/more User Jobs? " prompt which appears after the "Nucleus" screen. The following screen shows this prompt.

```
    Do you need any/more User Jobs?
    :
```

A "Yes" response to the "Do you need any/more User Jobs?" prompt indicates that you have at least one application job (also called first-level jobs) that the Nucleus needs to create during initialization. The configuration process affects two types of application jobs: the first-level (user) job and the I/O job. This chapter discusses jobs in general and first-level (user) jobs in particular. Refer to Chapter 8 for a description of I/O jobs.

13.2 FIRST-LEVEL JOBS

The jobs in a system form a hierarchy. A task in one job can create other jobs. Tasks in the new jobs can create still other jobs, and so forth. The jobs which contain tasks that create other jobs are called parent jobs, and the jobs they create are their offspring.

Task and job creation is a dynamic process. However, when you configure a system, you specify an initial system which is created automatically when the system starts executing. (The *Guide to the iRMX II Interactive Configuration Utility* discusses the initialization process.) The job tree for an initial system consists of an ultimate parent job called the root job and a number of its offspring called first-level jobs. Intel supplies the root job and the first-level jobs required for the subsystems and software components included in your application system. The Basic I/O System, the Extended I/O System, and the Human Interface require their own first-level job. The remainder of any first-level jobs are jobs that you provide. These user-provided first-level jobs are called user jobs. Figure 13-1 illustrates an initial job tree.

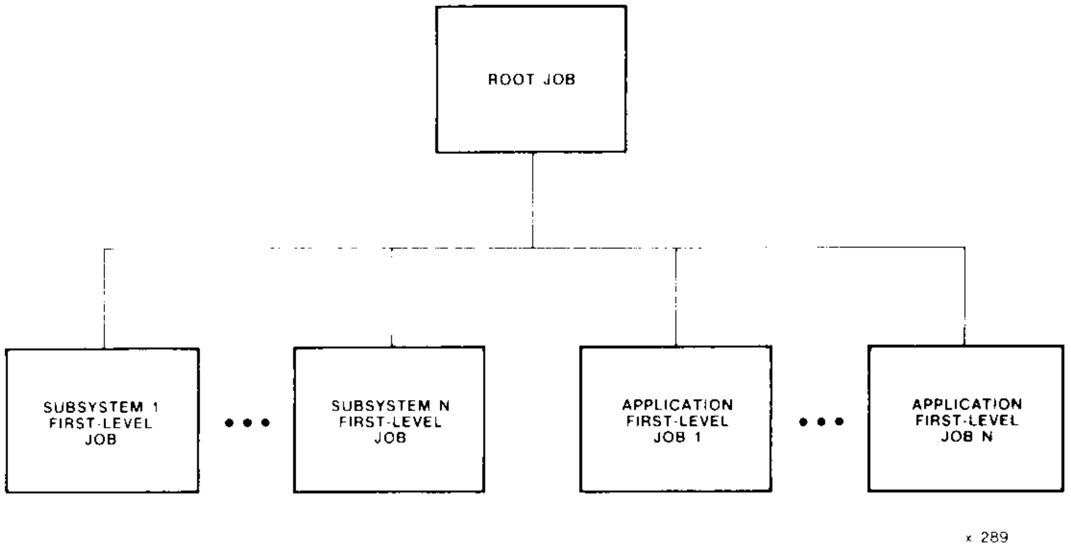

ROOT JOB
SUBSYSTEM 1 FIRST-LEVEL JOB
SUBSYSTEM N FIRST-LEVEL JOB
APPLICATION FIRST-LEVEL JOB 1
APPLICATION FIRST-LEVEL JOB N
x 289

Figure 13-1. Initial Job Tree

First-level jobs can create a number of offspring jobs, beginning the dynamic tree structure of the system. The *iRMX II Nucleus System Calls Reference Manual* describes how to create new tasks and jobs.

13.3 USER JOBS SCREEN

You control the type and number of jobs that are created during initialization. Each "User Jobs" screen represents a different user job. If you do not configure the Human Interface, at least one user job (or one I/O job) must be created during initialization. User jobs are created after all other first-level jobs and all I/O jobs, but before the Human Interface and any of its jobs. The order in which you specify user jobs is the order in which the Nucleus creates the jobs.

The parameters which define any job created with the Nucleus system call RQE$CREATE$JOB and RQ$CREATE$JOB are the same parameters that define a user job. These parameters are discussed in the following sections.

```
(USERJ)      User Jobs

(NAM) Job Name [0-14 Chars]
(ODS) Object Directory Size [0-3840]                0
(PMI) Pool Minimum [20H - 0FFFFFH]                  060H
(PMA) Pool Maximum [20H - 0FFFFFH]                  0FFFFFH
(MOB) Maximum Objects [1 - 0FFFFH]                  0FFFFH
(MTK) Maximum Tasks [1 - 0FFFFH]                    0FFFFH
(MPR) Maximum Priority [0 - 255]                    129
(EHS) Exception Handler Entry Point [1-31 Chars]

(EM)  Exception Mode [Never/Prog/Environ/All]       NEVER
(PV)  Parameter Validation [Yes/No]                 YES
(TP)  Task Priority [0-255]                         155
(TSA) Task Entry Point [1-31 Chars]

(VAR) Public Variable Name [1-31 Chars]

(SSA) Stack Segment Address [SS:SP]                 0000:0000H
(SSI)  Stack Size [0-0FFFFH]                        0300H
(NPX) Numeric Processor Extension Used [Yes/No]     NO

Enter  [ Abbreviation = new_value / Abbreviation ? / H ]
:
```

```
(NAM) Job Name [0-14 Chars]
```

This parameter specifies a name that will identify a particular user job screen. This name is an aid while you run the ICU and does not become part of the job.

```
(ODS) Object Directory Size [0-3840]                0
```

This parameter specifies the maximum allowable number of entries in this user job's object directory. A value of zero indicates that no directory is to be created.

Each user job has its own object directory. An entry in an object directory consists of a token for an object and the object name. The number of objects in the object directory is limited to the object directory size.

```
    (PMI) Pool Minimum [20H - OFFFFFH]                   060H
    (PMA) Pool Maximum [20H - OFFFFFH]                OFFFFFH
```

These parameters let you define the minimum and maximum allowable size of the user job's memory pool. The maximum memory pool size is 16M bytes, in 16-byte paragraphs. The minimum pool value determines the initial size of the memory pool and the limit to which the memory pool can be reduced. If the "Stack Segment Address" parameter has a base value of zero, set the "Pool Minimum" parameter line value to 32 (decimal) plus the value specified on the "Stack Size" parameter line. (The value specified on the "Stack Size" parameter must first be converted from the number of bytes to the number of 16-byte paragraphs.)

The value of the maximum pool size must be greater than or equal to the minimum pool size. If your system includes the Human Interface, the value you specify for the "(PMI) Pool Minimum" parameter should be the same value you specify for the "(PMA) Pool Maximum" parameter.

```
    (MOB) Maximum Objects [1 - OFFFFH]                  OFFFFH
```

This parameter specifies the maximum number of objects that can exist simultaneously in this user job. A value of 0FFFFH indicates that there is no limit to the number of objects that tasks in this user job can create.

```
    (MTK) Maximum Tasks [1 - OFFFFH]                    OFFFFH
```

This parameter specifies the maximum number of tasks that can exist simultaneously in this user job. A value of 0FFFFH indicates that there is no limit to the number of tasks that tasks in this user job can create.

```
    (MPR) Maximum Priority [0 - 255]                       129
```

This parameter specifies the maximum allowable priority of tasks in a user job. Specify a value in the range of 0 to 255 decimal. A value of zero indicates that the priority of the root task is the maximum allowable.

```
    (EHS) Exception Handler Entry Point [1-31 Chars]
```

User jobs can inherit the exception handler of the parent job. In this case, the exception handler that would be inherited is the default system exception handler that you specified for the Nucleus. If you want to use the default exception handler, leave this parameter line blank. If you want to use a handler other than the default system exception handler, this parameter allows you to specify the PUBLIC name of the other exception handler.

The name you enter for this parameter defines the PUBLIC name of the exception handler procedure. When binding your user job, you must include a NOPUBLICS EXCEPT (xxx) control, where xxx is the name you enter in the this parameter line. The pathname of the object files (obtained from compiling and binding the user job code and exception handler) containing these procedures must also be specified on the "User Module" screen. Do not use the letters "RQ" to begin PUBLIC names.

```
    (EM)   Exception Mode [Never/Prog/Environ/All]        NEVER
```

If you specified a PUBLIC name for the "Exception Handler Entry Point" parameter line (indicating that you do not want the default system exception handler), you must specify the exception mode of the exception handler specified for this user job.

Choose one of the following four options:

Never — Control never passes to the exception handler.

Programmer — Control passes to the exception handler on programmer errors only.

Environment — Control passes to the exception handler on environmental conditions only.

All — Control passes to the exception handler on all exceptional conditions.

```
    (PV)   Parameter Validation [Yes/No]                  YES
```

Specify a "Yes" to this parameter line if you want the Nucleus to perform parameter validation for all Nucleus system calls made by tasks in this user job. Specify a "No" if you do not want the Nucleus to validate parameters for tasks in this user job.

If your user job includes tasks that invoke Basic I/O System system calls, respond to this parameter line with a "Yes".

Explanations for not including parameter validation at either a system-level or at a user job level are given in Chapter 13. Note that the ICU ignores your response to this parameter line if you have not included system-level parameter validation.

```
    (TP)   Task Priority [0-255]                          155
```

This parameter specifies the priority of this job's initialization task. A value of zero assigns the initialization task a priority equal to the value you specified for the "Maximum Priority" parameter on this screen.

When created, each user job contains only a single task. That single task creates or starts the creation of all other objects required by the user job. Thus, this task is referred to as the initialization task for its job, even though it may perform other functions as well.

A task's priority is an integer value between 0 and 255, inclusive. The lower the priority number, the higher the priority of the task. Unless a task is involved in processing interrupts, its priority should be between 129 and 255. The value of this parameter must not be numerically smaller than the maximum priority for the job.

```
    (TSA) Task Entry Point [1-31 Chars]
```

The name you enter for this parameter defines the PUBLIC name of your user job's entry procedure. When binding your user job, you must include a NOPUBLICS EXCEPT(xxx) control, where xxx is the name you enter in this parameter line. The pathname of the object file (produced by compiling and binding your user job) containing this procedure must also be specified on the "User Module" screen.

```
    (VAR) Public Variable Name [1-31 Chars]
```

The name you enter for this parameter defines the PUBLIC name of any one of your variables. The Nucleus uses this name to obtain the selector of the User Job's data segment. When binding your user job, you must include a NOPUBLIC EXCEPT(xxxx) control, where xxxx is the name you enter in the "VAR" parameter.

```
    (SSA) Stack Segment Address [SS:SP]          0000:0000H
```

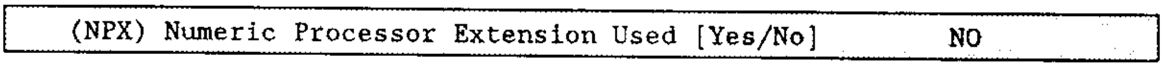
This parameter line lets you specify the address of the initialization task's stack. A value of 0:0 causes the Nucleus to allocate a stack segment to the task and initialize the SS register to the selector of this segment and the SP register to the value of the stack size prompt. It is recommended that you specify 0:0 for this parameter. This permits dynamic stack allocation and deallocation.

```
    (SSI)  Stack Size [0-0FFFFH]                       0300H
```

This parameter line lets you specify the size in bytes of the initialization task's stack segment. This stack size must be at least 16 bytes. The Nucleus increases specified values that are not multiples of 16 up to the next higher multiple of 16. The size should be at least 300 (decimal) bytes if the initialization task is going to make Nucleus system calls. The value you specify depends on which model of segmentation you use.

PL/M-286 Medium Models

Procedures compiled using the MEDIUM model require statically allocated stacks. Thus, for these procedures, you must specify the stack segment address for the user job. The size you specify for the "Stack Size" parameter line should be the same size you specified in the SEGSIZE(STACK(...) control of the BND286 command. Refer to the "Binding and Building Application Jobs" section of Chapter 3 in the *Guide to the iRMX II Interactive Configuration Utility*.

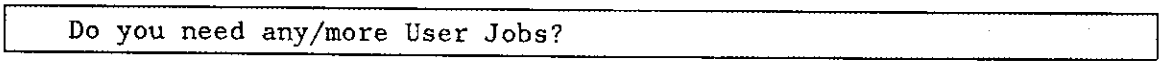


```
(NPX) Numeric Processor Extension Used [Yes/No]          NO
```

You must respond to this parameter with a "Yes" if the user job's initial task contains floating-point instructions. You do not have to respond with a "Yes", however, if an offspring and not this user job contains a task that uses floating-point instructions. You should not respond "Yes" to this parameter if you have entered "No" to the "NPX" parameter on the "Hardware" screen.

13.4 QUERY SCREEN

After you have completed the "User Jobs" screen, a query screen is displayed. It contains only one line:



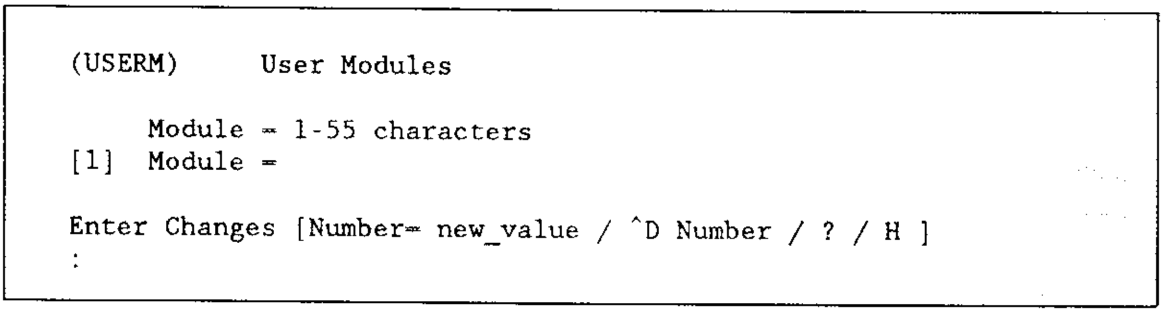
```
Do you need any/more User Jobs?
```

Respond "Yes" if you have another user job. A response of "Yes" causes the "User Jobs" screen to be redisplayed. The default value is "No". There is no limit on the number of user jobs you can create.

The order in which you define your user jobs is also the order in which the root task initializes the jobs.

13.5 USER MODULES SCREEN

After you respond "No" to the prompt "Do you need any/more User Jobs?", the ICU displays the following screen.



```
(USERM)      User Modules

     Module = 1-55 characters
[1]  Module =

Enter Changes [Number= new_value / ^D Number / ? / H ]
:
```

This screen lets you specify the pathnames for the object code that has been bound (using BND286) for the application jobs defined on the "I/O Jobs" screen, the "User Jobs" screen or for the OS extensions you defined on the "OS Extension" screen.

For purposes of configuring the system, the order of your user module pathnames is not significant.

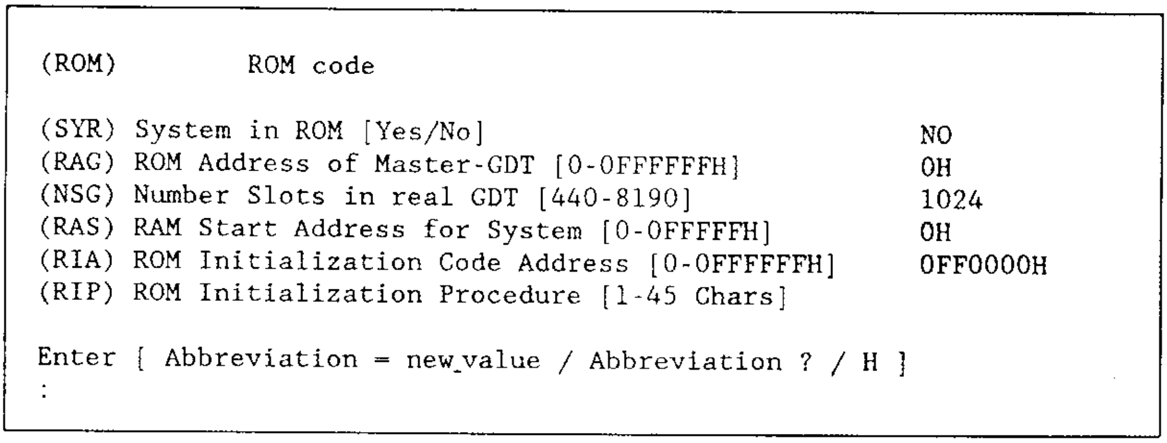
CHAPTER 14
ROM CODE PARAMETERS

14.1 INTRODUCTION

This chapter discusses how to respond to the parameters that appear on the "ROM Code" screen. This screen is useful only if you are burning your entire system into PROM. If you are configuring a RAM-based system, leave the ROM code parameters in their default state and the ICU will ignore them.

Configuring a PROM-based system allows you to write-protect your stable code, allows your system to load quicker than a RAM-based system, and may cost less than a RAM-based system. Before you can execute code from PROM, you must first test and debug your system in RAM, ascertain the start and stop addresses for your code and data segments, test your PROM-based system in RAM, and program your code into PROM. These steps are described in greater detail later in this chapter.

14.2 ROM CODE SCREEN



```
(ROM)        ROM code

(SYR) System in ROM [Yes/No]                          NO
(RAG) ROM Address of Master-GDT [0-0FFFFFFFH]         0H
(NSG) Number Slots in real GDT [440-8190]             1024
(RAS) RAM Start Address for System [0-0FFFFFFH]       0H
(RIA) ROM Initialization Code Address [0-0FFFFFFFH]   0FF0000H
(RIP) ROM Initialization Procedure [1-45 Chars]

Enter [ Abbreviation = new_value / Abbreviation ? / H ]
:
```

```
(SYR) System in ROM [Yes/No]                        NO
```

This parameter informs the ICU whether or not your application is to reside in ROM. If you answer "Yes" to this parameter, your system will reside in ROM. If you answer "No", your system will reside in RAM and the ICU will ignore the rest of the parameters on this screen. If you are familiar with the iRMX I Operating System, note that only the entire system can be placed in ROM, not each layer.

```
  (RAG) ROM Address of Master-GDT [0-0FFFFFFH]          0H
```

This parameter describes the 24-bit physical address of the master GDT as it is to be burned into ROM. The master GDT is part of the output of BLD286 and describes all the data, code and stack segments. However, you cannot use a ROM-based GDT when running your program. To use the GDT, you must first copy it into RAM. For that reason you must also respond to the "NSG" parameter.

In a RAM-based system, this parameter is ignored.

```
  (NSG) Number Slots in Real GDT [440-8190]          1024
```

This parameter defines the number of GDT slots in the RAM-based GDT created at run time. If you have a ROM-based system, you must copy the GDT from ROM to RAM, expanding it to the value given in this parameter line, before you can use it. In most cases, only the minimum number of slots are burned into ROM as ROM space is limited. In a typical iRMX II system, only the bottom 300-400 GDT entries have initial values. The remaining slots are initialized by the free space manager when the system is executed. Therefore, rather than burn empty GDT slots into ROM simply because they may be required later by the free space manager, you only burn those GDT slots that are needed for system initialization. The Nucleus copies all the read/write segments and descriptor tables from ROM to RAM.

In a RAM-based system, this parameter is ignored.

```
  (RAS) RAM Start Address for System [0-0FFFFFFH]           0H
```

Parts of a ROM-based system must be copied into RAM before it can be executed. This parameter gives the start address, in bytes, where the Nucleus is to begin copying the iRMX II Operating System into RAM.

In a RAM-based system, this parameter is ignored.

```
(RIA) ROM Initialization Code Address [0-0FFFFFFH]     0FF0000H
```

This parameter defines the 24-bit physical address of the special initialization routine required by ROM-based systems. When an 80286 microprocessor is initialized, it is in Real Address Mode and can only access up to 1M of memory. The initialization routine starts running in real mode and then switches the processor to Protected Virtual Address Mode (PVAM) which supports a 16M byte address space. The initialization routine switches the CPU board's address decoding circuitry to access all 16M bytes of memory (instead of the 1M byte allowed in real mode).

Intel 80286-based Single Board Computers (iSBC 286/10, iSBC 286/10A, iSBC 286/12, and iSBC 286/100A) all utilize the same on-board EPROM address decoding scheme in which the EPROM sites reside at the highest available physical memory address. For example, if the 80286-based Single Board Computer is configured for 27512 EPROMs, the address space reserved for these EPROMs is from 0C000H to 0FFFFH in real mode, and from 0FC0000H to 0FFFFFFH in PVAM. Therefore, the address of the EPROM code will effectively jump from the low megabyte of memory to the high megabyte of memory when the 80286-based Single Board Computer is switched to operate in PVAM. Therefore, the address specified for "ROM Initialization Code Address" must be the address of the ROM-initialization code when in protected mode. This causes no problems while in the real mode portion of the initialization routine because both the 80286-based Single Board Computer decoding circuitry and the 80286 microprocessor running in real mode ignore the upper four (4) bits of the ROM Initialization Code Address.

In a RAM-based system, this parameter is ignored.

```
(RIP) ROM Initialization Procedure [1-45 Chars]
```

This parameter defines an object file name containing the board initialization procedure INIT_ROM_BOARD. This routine is called immediately after the processor has been switched to protected mode. It performs board-specific initializations, for example, output to specific ports to switch the board to a 16M-byte addressability mode.

In a RAM-based system, this parameter is ignored.

14.3 CONFIGURING A ROM-BASED SYSTEM

Before you program any system into ROM (PROM), you should first be confident that your code is fully debugged. You must also know the size of all the code and data segments. In a RAM-based system, the amount of memory needed by the code and data segments is reserved for the system on the "Memory for System" screen. The remaining memory is available for the free space manager. However, in a ROM-based system, it is necessary to copy the GDT, LDT, IDT and all writable segments into RAM. Therefore, you should exclude the amount of memory required by the descriptor tables and the writable segments from the memory required for the free space manager. The actual area to be used for the system RAM is defined by the "RAM Start Address" parameter and the size of the RAM segments. Ensure that this area is not reserved for the system or the free space manager.

To determine the size of your code and data segments, follow these guidelines.

1. Bind your application using BND286.
2. Run the ICU to create the generation SUBMIT file.
3. Invoke the SUBMIT file.
4. Read the memory map (MP2), created during the build phase, to find the start address for the GDT in ROM, the amount of ROM your system uses, and the amount of RAM your system requires. Calculate the RAM memory required by adding the size of the IDT, the final GDT (the number of GDT entries multiplied by 8), the final LDT, which is the same as the final GDT, and the sum of all the data segments with the WRITABLE attribute. To this number add 2700 bytes which are used as a work area during system startup.
5. Configure out the size of the memory calculated in step 4 from the memory defined on the "Memory for System" screen.
6. Rerun the ICU to create the final generation SUBMIT file.
7. Invoke the final generation SUBMIT file.

The final file generated after rerunning the ICU with the correct parameters does not include the start address of your system, that is the initial JMP at address 0FFFFF0H. You must burn that into ROM separately.

If you specified that the initialization code resides somewhere in the top 64K bytes (not below address 0FF0000H), then you can burn a short jump to that address. Otherwise, you need a FAR jump, which may cause a problem. The 80286 processor resets the high 4 address bits. This means you can only jump to the low megabyte, but your ROM is usually in the high megabyte. Many boards such as the iSBC 286/12 and iSBC 386/20 solve this problem by setting the 4 address bits to one until a specific OUT command is issued. This allows you to perform a FAR jump to anywhere in the high megabyte.

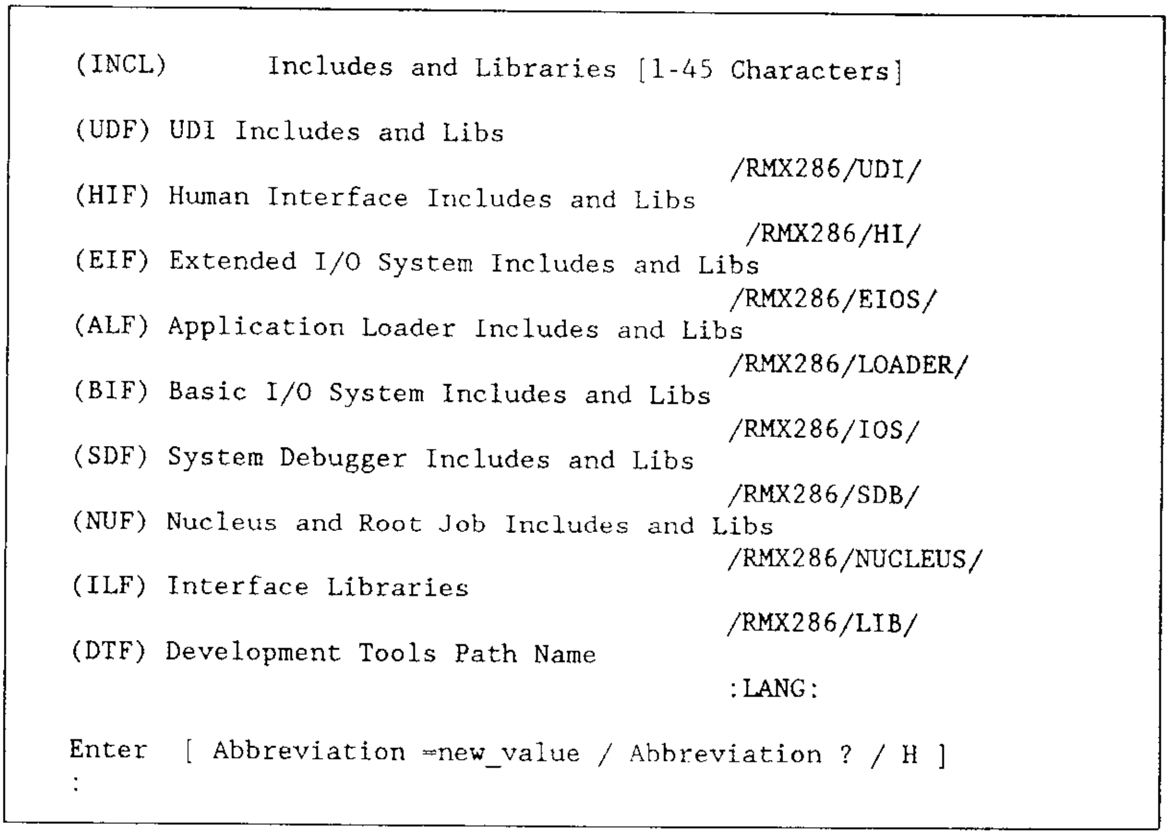
The initialization routine resides in the segment NUCDAT.CODE_ROM, and its entry point is at offset 12H. To obtain the entry point address, add the entry routine offset (12H) to the address of NUCDAT.CODE_ROM (which you specified during ICU configuration). For example, if the address of NUCDAT.CODE_ROM is 0FC0000H, the address of the ROM-initialization routine would be 0FC0012H. At location 0FFFFF0H in your PROMs burn a FAR jump to the address you derived for the ROM-initialization routine.

Appendixes C and D of the *Guide To The Extended iRMX II Interactive Configuration Utility* provides an example of how to program your code into PROM. It lists both the hardware and software necessary to do this. Refer to these appendixes for further information.

CHAPTER 15
INCLUDES AND LIBRARIES PARAMETERS

15.1 INTRODUCTION

This chapter discusses how to respond to the parameters that appear on the "Includes and Libraries" screen. The following "Includes and Libraries" screen shows the default values that appear if you invoke the ICU using the 28612.def file. It is recommended that you use these values if you are developing your code on an iRMX II-based system and have installed the Operating System as instructed in the *iRMX II Hardware And Software Installation Guide*.



```
(INCL)      Includes and Libraries [1-45 Characters]

(UDF) UDI Includes and Libs
                                        /RMX286/UDI/
(HIF) Human Interface Includes and Libs
                                        /RMX286/HI/
(EIF) Extended I/O System Includes and Libs
                                        /RMX286/EIOS/
(ALF) Application Loader Includes and Libs
                                        /RMX286/LOADER/
(BIF) Basic I/O System Includes and Libs
                                        /RMX286/IOS/
(SDF) System Debugger Includes and Libs
                                        /RMX286/SDB/
(NUF) Nucleus and Root Job Includes and Libs
                                        /RMX286/NUCLEUS/
(ILF) Interface Libraries
                                        /RMX286/LIB/
(DTF) Development Tools Path Name
                                        :LANG:

Enter  [ Abbreviation =new_value / Abbreviation ? / H ]
:
```

To generate a SUBMIT file that will assemble, bind, and build your configuration files, the ICU must know the pathnames for the libraries and INCLUDE files that it needs during the generation phase. The parameters that appear on the "Includes and Libraries" screen let you specify a unique pathname for each subsystem.

```
    (UDF) UDI Includes and Libs                            /RMX286/UDI/
```

This parameter specifies the pathname of the directory that contains the Universal Development Interface files. The name must be a logical name surrounded by colons or a pathname ending in a slash.

```
    (HIF) Human Interface Includes and Libs                 /RMX286/HI/
```

This parameter specifies the pathname of the directory that contains the Human Interface files. The name must be a logical name surrounded by colons or a pathname ending in a slash.

```
    (EIF) Extended I/O System Includes and Libs           /RMX286/EIOS/
```

This parameter specifies the pathname of the directory that contains the Extended I/O System files. The name must be a logical name surrounded by colons or a pathname ending in a slash.

```
    (ALF) Application Loader Includes and Libs          /RMX286/LOADER/
```

This parameter specifies the pathname of the directory that contains the Application Loader files. The name must be a logical name surrounded by colons or a pathname ending in a slash.

```
    (BIF) Basic I/O System Includes and Libs            /RMX286/IOS/
```

This parameter specifies the pathname of the directory that contains the Basic I/O System and device driver files. The name must be a logical name surrounded by colons or a pathname ending in a slash.

```
    (SDF) System Debugger Includes and Libs            /RMX286/SDB/
```

This parameter specifies the pathname of the directory that contains the System Debugger files. The name must be a logical name surrounded by colons or a pathname ending in a slash.

```
    (NUF) Nucleus and Root Job Includes and Libs      /RMX286/NUCLEUS/
```

This parameter specifies the pathname of the directory that contains the iRMX II Nucleus and root job files. The name must be a logical name surrounded by colons or a pathname ending in a slash.

```
    (ILF) Interface Libraries                         /RMX286/LIB/
```

This parameter specifies the pathname of the directory that contains the Interface Library files. The name must be a logical name surrounded by colons or a pathname ending in a slash.

```
    (DTF) Development Tools Path Names                       :LANG:
```

This parameter specifies the pathname of the directory that contains the Development Tools files. The name must be a logical name surrounded by colons or a pathname ending in a slash.

CHAPTER 16
GENERATE FILE NAMES PARAMETERS

16.1 GENERATE FILE NAMES SCREEN

This chapter discusses how to respond to the parameters that appear on the "Generate File Names" screen.

```
(GEN)          Generate File Names

File Name [1-55 Characters]

(ROF) ROM Code File Name
                                        /BOOT/RMX286.ROM
(RAF) RAM Code File Name
                                        /BOOT/28612.286

Enter  [ Abbreviation = new_value / Abbreviation ? / H ]

:
```

```
(ROF) ROM Code File Name                     /BOOT/RMX286.ROM
```

This parameter specifies the pathname of the directory (prefix) that will contain your system PROM code. The ICU places the object files for the entire system into this directory. If you enter a pathname in this parameter, and you entered "No" for the "System in ROM" parameter on the "ROM Code" screen, this parameter is ignored.

```
(RAF) RAM Code File Name                     /BOOT/28612.286
```

This parameter specifies the pathname of the system's bootloadable file. The default system file name is rmx86.286, although if you are using the 28612.def definition file, the default file name is 28612.286, as shown here. The file name you specify here contains the object files for your system RAM code. If you enter a pathname for this parameter, be sure you have responded "No" to the "System in ROM" parameter on the "ROM Code" screen. The bootloadable file must have a "286" extension in order to work properly with the bootstrap loader.

CHAPTER 17
BUILD FILE COMMENTS

17.1 COMMENTS FOR BUILD FILE SCREEN

The "Comments for Build File" screen has been incorporated into the ICU for your convenience. It allows you to add comments to yourself such as the name of the build file, important variable locations, and the release supported. These comments are added to the build file, but they are not recognized or used by the ICU in anyway. This is an optional screen and does not have to be filled in. All comments must fit on a single line.

An example of a possible comment is:

'Defines a 286.12 Multi Access System'

The screen has the following format.

```
(COMNT)        Comments for Build File

  each line  = 1-55 Characters - IN QUOTES

[ 1] = 'iRMX II R3.0 for System 286/310 Microcomputers   '
[ 2] =

Enter Changes [ Number= new_value / ^D Number / ? / H ]
:
```

INTERNATIONAL SALES OFFICES

INTEL CORPORATION
3065 Bowers Avenue
Santa Clara, California 95051

BELGIUM
Intel Corporation SA
Rue des Cottages 65
B-1180 Brussels

DENMARK
Intel Denmark A/S
Glentevej 61-3rd Floor
dk-2400 Copenhagen

ENGLAND
Intel Corporation (U.K.) LTD.
Piper's Way
Swindon, Wiltshire SN3 1RJ

FINLAND
Intel Finland OY
Ruosilante 2
00390 Helsinki

FRANCE
Intel Paris
1 Rue Edison-BP 303
78054 St.-Quentin-en-Yvelines Cedex

ISRAEL
Intel Semiconductors LTD.
Atidim Industrial Park
Neve Sharet
P.O. Box 43202
Tel-Aviv 61430

ITALY
Intel Corporation S.P.A.
Milandfiori, Palazzo E/4
20090 Assago (Milano)

JAPAN
Intel Japan K.K.
Flower-Hill Shin-machi
1-23-9, Shinmachi
Setagaya-ku, Tokyo 15

NETHERLANDS
Intel Semiconductor (Netherland B.V.)
Alexanderpoort Building
Marten Meesweg 93
3068 Rotterdam

NORWAY
Intel Norway A/S
P.O. Box 92
Hvamveien 4
N-2013, Skjetten

SPAIN
Intel Iberia
Calle Zurbaran 28-IZQDA
28010 Madrid

SWEDEN
Intel Sweden A.B.
Dalvaegen 24
S-171 36 Solna

SWITZERLAND
Intel Semiconductor A.G.
Talackerstrasse 17
8125 Glattbrugg
CH-8065 Zurich

WEST GERMANY
Intel Semiconductor G.N.B.H.
Seidlestrasse 27
D-8000 Munchen