

**iRMX<sup>®</sup> I**  
**Interactive Configuration**  
**Utility Reference Manual**



---

**iRMX<sup>®</sup> I**  
**Interactive Configuration**  
**Utility Reference Manual**

Order Number: 462932-001

Intel Corporation  
3065 Bowers Avenue  
Santa Clara, California 95051

Copyright © 1980, 1989, Intel Corporation, All Rights Reserved

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 after 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	iLBX	iPSC	Plug-A-Bubble
BITBUS	i <sub>m</sub>	iRMX	PROMPT
COMMputer	iMDDX	iSBC	Promware
CREDIT	iMMX	iSBX	QUEST
Data Pipeline	Insite	iSDM	QueX
Genius	int <sub>e</sub> l	iSSB	Ripplemode
↑	Intel376	iSXM	RMX/80
i	Intel386	Library Manager	RUPI
I <sup>2</sup> ICE	int <sub>e</sub> iBOS	MCS	Seamless
ICE	Intelelevision	Megachassis	SLD
iCEL	int <sub>e</sub> ligent Identifier	MICROMAINFRAME	UPI
iCS	int <sub>e</sub> ligent Programming	MULTIBUS	VLSiCEL
iDBP	Intellec	MULTICHANNEL	376
iDIS	Intellink	MULTIMODULE	386
	iOSP	OpenNET	386SX
	iPDS	ONCE	
	iPSB		

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, PC/XT, and PC/AT are registered trademarks of International Business Machines. Soft-Scope is a registered trademark of Concurrent Sciences.

Copyright © 1980, 1989, Intel Corporation. All Rights Reserved.

<b>REV.</b>	<b>REVISION HISTORY</b>	<b>DATE</b>
-001	Original Issue.	03/89



## INTRODUCTION

This manual describes the iRMX® I Interactive Configuration Utility (ICU) screens and parameters.

## READER LEVEL

The manual is written for programmers who are already familiar with the monitor and keyboard from which you run the ICU. It is also helpful if you are familiar with the following:

- The iRMX® I Operating System
- The PL/M-86 programming language.
- LINK86 and LOC86

## CONVENTIONS

This manual uses the following conventions:

- The term "iRMX I" refers to the iRMX I (iRMX 86) Operating System.
- The term "iRMX II" refers to the iRMX II.3 Operating System.
- Information appearing as **bold** characters when shown in keyboard examples must be entered or coded exactly as shown. You may, however, mix lower and uppercase characters when entering text.
- All numbers, unless otherwise indicated, are assumed to be decimal. Hexadecimal numbers include the "H" radix character, for example, 0FFH.
- The values shown in the example screens are for illustration purposes only. The actual values that you see depend on the definition file used as input to the ICU and your changes to the definition file.



# CONTENTS

---

## Chapter 1. Hardware-Related Parameters

1.1 Introduction .....	1-1
1.2 Hardware Screens.....	1-2
0 = 8086 CPU.....	1-2
1 = 8088 CPU.....	1-3
2 = 80286 CPU.....	1-3
3 = 80186 CPU.....	1-3
4 = 386™ CPU.....	1-3
1.3 Interrupt Screen.....	1-9
1.4 Slave Interrupt Levels Screen.....	1-11
1.5 80186 Initialization Screen.....	1-12
1.6 MULTIBUS® II Hardware Screen.....	1-17

---

## Chapter 2. Memory Parameters

2.1 Introduction .....	2-1
2.2 Memory Prompts.....	2-1
2.2.1 RAM.....	2-2
2.2.1.1 Upper Memory Address Considerations.....	2-2
2.2.1.2 80186 Considerations.....	2-2
2.2.1.3 Communications Board Considerations.....	2-2
2.2.1.4 RAM Driver Considerations.....	2-3
2.2.2 ROM.....	2-3
2.2.2.1 Monitor Considerations.....	2-3

---

## Chapter 3. Sub-Systems Parameters

3.1 Introduction .....	3-1
3.2 Sub-Systems Screen.....	3-1
3.2.1 Universal Development Interface.....	3-2
3.2.2 Human Interface.....	3-3
3.2.3 Application Loader.....	3-3
3.2.4 Remote File Access.....	3-3
3.2.5 Extended I/O System.....	3-4
3.2.6 Basic I/O System.....	3-4
3.2.7 System Debugger.....	3-4
3.2.8 Dynamic Debugger.....	3-5
3.2.9 Terminal Handler.....	3-5

---

## Chapter 4. Human Interface Parameters

4.1 Introduction.....	4-1
4.2 Background Information.....	4-2
4.2.1 Resident and Non-Resident Configuration, Users, and Initial Programs.....	4-2
4.2.2 System Device, Directories, Logical Names, and Prefixes.....	4-4
4.3 The Screens .....	4-4
4.3.1 Human Interface Screen .....	4-5
4.3.2 Human Interface Jobs Screen .....	4-8
4.3.3 Logical Names.....	4-10
4.3.4 Resident User Parameters.....	4-11
4.3.5 Prefixes .....	4-13

---

## Chapter 5. Application Loader Parameters

5.1 Application Loader Screen .....	5-1
-------------------------------------	-----

---

## Chapter 6. Remote File Access Parameters

6.1 Introduction.....	6-1
6.2 Remote File Access Screen .....	6-1
6.3 Query Screen.....	6-3
6.4 Remote File Server Screen.....	6-4

---

## Chapter 7. Extended I/O System Parameters

7.1 Introduction.....	7-1
7.2 EIOS Screen .....	7-2
7.3 Automatic Device Recognition Screen.....	7-5
7.4 Logical Names Screen .....	7-8
7.5 I/O Users Screen.....	7-9
7.6 Query Screen.....	7-10
7.7 I/O Jobs Screen .....	7-11

---

## Chapter 8. Basic I/O System Parameters

8.1 Introduction.....	8-1
8.2 BIOS Screen .....	8-1
8.3 BIOS System Calls Screen .....	8-8
8.3.1 Non-File System Calls.....	8-8
8.3.2 Physical File System Calls .....	8-9
8.3.3 Stream File System Calls.....	8-10
8.3.4 Named File System Calls.....	8-11

---

## Chapter 9. Device Driver Parameters

Mass Storage Controller Parameters .....	MSC-1
Mass Storage Controller Driver Screen .....	MSC-2
Unit Information Screen .....	MSC-4
Multiple Units on the Same Winchester Disk .....	MSC-10
Device-Unit Information Screen .....	MSC-12
<hr/>	
8274 Terminal Driver Parameters .....	8274-1
8274 Terminal Driver Screen .....	8274-2
Terminal Driver Unit Information Screen .....	8274-5
Terminal Driver Device-Unit Information Screen .....	8274-11
<hr/>	
8251A Driver Parameters .....	8251A-1
8251A Terminal Driver Screen .....	8251A-1
8251A Terminal Driver Unit Information Screen .....	8251A-5
8251A Terminal Driver Device-Unit Information Screen .....	8251A-12
<hr/>	
82530 Terminal Driver Parameters .....	82530-1
82530 Terminal Driver Screen .....	82530-2
82530 Unit Information Screen .....	82530-4
82530 Device Unit Information Screen .....	82530-10
<hr/>	
Terminal Communications Controller Driver Parameters .....	TCC-1
Terminal Communications Controller Driver Screen .....	TCC-2
Terminal Communications Controller Unit Information Screen .....	TCC-4
Terminal Communications Controller Device Unit Information Screen .....	TCC-10
<hr/>	
iSBC® 286/10(A) Line Printer Driver .....	LP286-1
iSBC® 286/10(A) Line Printer Driver Screen .....	LP286-1
iSBC® 286/10(A) Line Printer Device-Unit Information Screen .....	LP286-3
<hr/>	
Line Printer--iSBX™ 350 .....	LP350-1
Line Printer--iSBX™ 350 Driver Screen .....	LP350-1
Line Printer--iSBX™ 350 Driver Device Unit Information Screen .....	LP350-4
<hr/>	
iSBC® 220 Driver Parameters .....	220-1
iSBC® 220 Driver Screen .....	220-1
iSBC® 220 Unit Information Screen .....	220-3
iSBC® 220 Device-Unit Information Screen .....	220-8

---

**Chapter 9. Device Driver Parameters (continued)**

iSBX™ 218A Driver Parameters.....	218A-1
iSBX™ 218A Driver Screen.....	218A-1
iSBX™ 218A Unit Information.....	218A-3
iSBX™ 218A Device-Unit Information.....	218A-6
<hr/>	
iSBC® 208 Driver Parameters .....	208-1
iSBC® 208 Driver Screen.....	208-1
iSBC® 208 Unit Information Screen.....	208-4
iSBC® 208 Device-Unit Information Screen.....	208-8
<hr/>	
iSBC® 254 Driver Parameters .....	254-1
iSBC® 254 Driver Screen.....	254-1
iSBC® 254 Unit Information Screen.....	254-3
iSBC® 254 Device-Unit Information Screen.....	254-6
<hr/>	
Terminal Handler Driver Device-Unit Information .....	Terminal Handler-1
<hr/>	
iSBC® 534 Driver Parameters .....	534-1
iSBC® 534 Driver Screen.....	534-1
iSBC® 534 Unit Information Screen.....	534-3
iSBC® 534 Device Unit Information Screen .....	534-9
<hr/>	
iSBC® 544A Driver Parameters .....	544A-1
iSBC® 544A Driver Screen.....	544A-1
iSBC® 544A Unit Information Screen .....	544A-4
iSBC® 544A Device Unit Information Screen.....	544A-11
<hr/>	
Ram Driver Parameters .....	RAM-1
Ram Disk Driver Screen.....	RAM-2
Ram Driver Unit Information Screen.....	RAM-3
Ram Driver Device-Unit Information Screen .....	RAM-5
<hr/>	
SCSI Driver Parameters .....	SCSI-1
SCSI Driver Unit Information Screen.....	SCSI-5
SCSI Device-Unit Information Screen.....	SCSI-10

---

**Chapter 9. Device Driver Parameters (continued)**

iSBC® 264 Driver Parameters .....	264-1
iSBC® 264 Driver Screen .....	264-1
iSBC® 264 Unit Information Screen .....	264-3
iSBC® 264 Device Unit Information Screen .....	264-5

---

User Devices .....	User-1
UDS Device Drivers .....	User-6

---

**Chapter 10. System Debugger Parameters**

10.1 Introduction .....	10-1
10.2 The System Debugger Parameter .....	10-1

---

**Chapter 11. Terminal Handler and Dynamic Debugger Parameters**

11.1 Introduction .....	11-1
11.2 Dynamic Debugger/Terminal Handler Screen .....	11-2

---

**Chapter 12. Nucleus Parameters**

12.1 Introduction .....	12-1
12.2 Nucleus Screen .....	12-1
12.3 System Call Parameters .....	12-10
12.3.1 Object System Calls .....	12-10
12.3.2 Job and Task System Calls Screen .....	12-11
12.3.3 Exchange System Calls Screen .....	12-12
12.3.4 Free Space System Calls Screen .....	12-13
12.3.5 Interrupt System Calls Screen .....	12-14
12.3.6 Extension System Calls Screen .....	12-15
12.3.7 Exception System Calls Screen .....	12-16

---

**Chapter 13. User Job Parameters**

13.1 Introduction .....	13-1
13.2 First-Level Jobs .....	13-1
13.3 User Job Screen .....	13-2
13.4 Query Screen .....	13-7
13.5 User Modules .....	13-8

---

**Chapter 14. ROM Code Parameters**

14.1 Introduction.....	14-1
------------------------	------

---

**Chapter 15. Includes and Libraries Parameters**

15.1 Introduction.....	15-1
------------------------	------

---

**Chapter 16. Generate File Names Parameters**

16.1 Generate File Names Screen .....	16-1
---------------------------------------	------

---

**Chapter 17. Build File Comments Parameter**

17.1 Comments for Build Files Screen.....	17-1
---	------

---

**TABLES**

3-1. Subsystem Dependencies.....	3-2
8-1. Non-File System Calls .....	8-9
8-2. Physical File System Calls.....	8-10
8-3. Stream File System Calls .....	8-11
8-4. Named File System Calls .....	8-12
MSC-1. iSBC® 214/215G Controller Information for Winchester Drives.....	Msc-6
MSC-2. iSBC® 214/215G/iSBX™ 218A Controller Information for Diskette and Tape Drives.....	Msc-6
MSC-3. iSBC® 214/215G DUIB Information for Winchester Drives .....	Msc-14
MSC-4. iSBC® 214/215G/iSBX™ 218A DUIB Information for Flexible Disk and Tape Drives.....	Msc-15
MSC-5. 5.25-Inch Disk Characteristics .....	Msc-16
MSC-6. 8-Inch Disk Characteristics .....	Msc-16
218A-1. iSBX™ 218A Controller Unit Information .....	218A-4
218A-2. iSBX™ 218A Controller DUIB Information.....	218A-7
254-1. iSBC® 254 Controller Information .....	254-5
SCSI-1. SCSI Unit Information Names .....	SCSI-6
SCSI-2. SCSI DUIB Information.....	SCSI-11
SCSI-3. Example SCSI Device-Unit Numbers .....	SCSI-15
12-1. System-Level and Job-Level Parameter Validation .....	12-3
12-2. Object System Calls .....	12-11
12-3. Job and Task System Calls .....	12-12
12-4. Exchange System Calls.....	12-13
12-5. Free Space System Calls .....	12-14
12-6. Interrupt System Calls.....	12-15
12-7. Extension System Calls .....	12-16

## CONTENTS

---

### FIGURES

User-1 Device Numbering.....	User-5
13-1. Initial Job Tree .....	13-2



# 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, Interrupts, 80186 Initialization, and MULTIBUS® II hardware. 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.

Note that values shown on the screens are sample values only. The exact values on each screen change according to the definition file used and your own changes.

### Hardware Parameters

(BUS)	(CPU)	(PPU)	(IF)	Start on page 1-2
(BBC)	(OSP)	(OTU)	(OPU)	
(BP)	(OPS)	(TP)	(CIL)	
(CN)	(CI)	(CF)	(TPS)	
(NPX)	(NIL)			

### Interrupts

(MP)	(LSS)	(MPS)	(LSP)	Start on page 1-9
(PIL)	(LSI)	(SIL)	(OIL)	
(PLI)				

### 80186 Initialization

(UCS)	(MCR)	(UCW)	(PCS)	Start on page 1-12
(UCR)	(PCA)	(LCS)	(PCM)	
(LCW)	(LPW)	(LCR)	(LPR)	
(MCS)	(UPW)	(MCA)	(UPR)	
(MCW)	(PLA)			

### MBII Hardware

(MDP)	(MDS)	(MDL)	Start on page 1-17
-------	-------	-------	-----------------------

## HARDWARE-RELATED PARAMETERS

### 1.2 HARDWARE SCREENS

This section describes the hardware configurable parameters available through the ICU. For most of these parameter lines, using the correct definition file as input to the ICU will set the default values for you. The default values shown in this book are those that appear if the 28612.DEF file is used as input to the Interactive Configuration Utility.

Hardware		
(BUS)	System Bus Type [1 - MBI / 2 - MBII]	1
(CPU)	Processor used in the system [0,1,2,3,4]	2
(OSP)	80130 Component Used [Yes/No]	NO
(TT)	Timer Type [1,2,3]	1
(TBA)	Timer Base Port Address [0-0FFFFH]	0DOH
(TPS)	Timer Port Separation [0-0FFH]	02H
(TCN)	Timer Counter Number [0,1,2]	0
(CIL)	Clock Interrupt Level [0-7]	0
(CIN)	Clock Interval [0-0FFFFH msec]	10
(CF)	Clock Frequency [0-0FFFFH khz]	1229
(GC)	Global Clock [546/CSM/86C38/None]	NONE
(GCN)	Global Clock Name [0, 1-12 characters]	
(NPX)	Numeric Processor Extension [Yes/No]	YES
(NIL)	NPX Interrupt Level [Encoded]	08H
(IF)	Initialize On-board Functions [1,2,3/No]	2
(RMB)	Remote Boot Location [0-NONE, 40H-0FFFFH]	0H
Enter Changes [Abbreviation ?/- new_value] :		

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

Use this parameter to specify the bus type that will be used in your system. If your system is based on the 16-bit MULTIBUS (or MULTIBUS I) bus architecture, specify "1". If your system is based on the 32-bit MULTIBUS II bus architecture, specify "2".

(CPU) Processor used in the system [0,1,2,3,4]	2
--	---

Use this parameter to specify the type of processor used in your system. Your response affects how the Nucleus will be initialized. Your choices are: (0) = 8086, (1) = 8088, (2) = 80286, (3) = 80186, and (4) = 386™.

#### 0 = 8086 CPU

Use the "8086" response to the "Processor used in the system" parameter if your system includes a processor board based on an 8086 CPU. (For example, this includes the iSBC® 86/05, iSBC 86/12A, iSBC 86/14, iSBC 86/30, or iSBC 86/35 processor boards.)

**1 = 8088 CPU**

Use the "8088" response to the "Processor used in the system" parameter if your system includes a processor board based on an 8088 CPU. (For example, this includes the iSBC 88/40 and iSBC 88/45 processor boards.)

**2 = 80286 CPU**

Use the "80286" response to the "(CPU) Processor used in the system" parameter if your system includes a processor board based on an 80286 CPU.

**3 = 80186 CPU**

Use the "80186" response to the "Processor used in the system" parameter if your system includes a processor board based on an 80186 or 80188 CPU. (For example, this includes an iSBC 186/03(A), an iSBC 186/51, or an iSBC 188/48 processor board.) If you respond "80186" to the "(CPU) Processor used in the system" parameter, the ICU adds the "80186 Initialization" screen. In iRMX® 86 Compatibility Mode, an 8259A or 80130 PIC is the master PIC and the 80186 PIC is one of several possible slaves. (iRMX 86 Compatibility Mode is discussed in the iAPX 186 Data Sheet.) If you are using an iSBC 186/03(A), an iSBC 186/51, or an iSBC 188/48 processor board, your board also includes an 80130 component. On these boards you must use the 80130 component as the master PIC with the 80186 PIC as one of several slaves. The 80130 component is discussed in greater detail later in this chapter.

**4 = 386™ CPU**

Use the "80386" response to the "(CPU) Processor used in the system" parameter if your system includes a processor board based on a 386 CPU.

(OSP) 80130 Component Used [Yes/No]	No
-------------------------------------	----

Use this parameter to specify that your system includes an 80130 component. The 80130 component is designed to work with the 8086, 8088, 80186, and the 80188 microprocessors.

The default value is "No". If your system includes an 80130 component, you can change the default value to "Yes".

## HARDWARE-RELATED PARAMETERS

(TT) Timer Type [1,2,3]	1
-------------------------	---

Use this parameter to specify the type of interval timer used in your system. The choices are:

Value	Timer Type
1	8253 or 8254
2	80130
3	80186

(TBA) Timer Base Port Address [0-0FFFFH]	0D0H
--	------

Use this parameter to specify the base port address of your Programmable Interval Timer (PIT). The timer component provides timing for your Operating System. The base port address is the address of the lowest timer port of your PIT. The ICU will calculate the addresses of the other ports using this address and the value specified in the "(TPS) Timer Port Separation" parameter.

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

Use this parameter to specify the distance between counter addresses. 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. When using Intel processor boards you should not change the default value. Refer to the individual drivers for information on how to configure timers for the Terminal Drivers, and how to configure the 8253 or 8254 PIT for the 8251A Terminal Handler.

(TCN) Timer Counter Number [0,1,2]	0
------------------------------------	---

Use this parameter to specify which of your Programmable Interval Timer (PIT) counters is to be used by the Nucleus. The counter used by the Nucleus must not be used by other software.

The 8253, the 8254, and the 80130 PITs each contain three counters. The Nucleus needs one of these counters (usually counter number zero) and the Terminal Handler or one of the terminal drivers can use a different counter (usually counter number two). If you are using an Intel processor board such as the iSBC 86/12A board or the iSBC 86/30 board, counter zero is factory configured to a 1.23 MHz clock. If you are using the 80130 Component, the Nucleus uses counter number two. When using Intel processor boards such as the iSBC 86/12A board and the iSBC 86/30 board, you should not change the default value.

(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.) It is recommended that you do not alter the default for this parameter if you are using Intel processor boards such as the iSBC 86/12A board or the iSBC 86/30 board. If you are using the iSBC 186/03(A) board or one of the Intel iAPX 286-based boards, it is recommended that you do not change the default value of zero.

(GIN) Clock Interval [0-65535 msec]	10
-------------------------------------	----

Use this parameter to specify the standard clock interval for the iRMX Operating System. 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 systems.

(CF) Clock Frequency [0-65535 khz]	1229
------------------------------------	------

Use this parameter to specify the frequency of the clock input to the timer. The frequency is measured in kilohertz.

When using Intel processor boards that include separate 8253/8254 timer chips use the corresponding clock frequency 1228.8 kilohertz entered as 1229 decimal for this parameter.

Other boards, such as the iSBC 186/03(A) and iSBC 186/51 boards, do not have separate 8253/8254 timers. Instead, they use the CPU clock for their timer input. For these boards, enter a value for this parameter that is one-fourth the CPU clock frequency. Note that if the CPU clock frequency changes (such as in an upgraded version of the board), you must change this parameter accordingly.

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

$count = clock\ interval \times clock\ frequency$
---

Refer to the individual boards for information on how to configure timers for the Terminal Driver and how to configure the 8253 or 8254 PIT with the Intel Terminal Handler.

For those using the 80130 Timer, specify one-fourth of the actual clock frequency.

## HARDWARE-RELATED PARAMETERS

(GC) Global Clock [546/CSM/86C38/None]	None
--	------

Use this parameter to specify if your system contains a battery-powered clock. This type of clock is referred to as a global clock. Use the following to determine the correct response.

<u>Value</u>	<u>Type of Board</u>
546/549	iSBC 546/549 Terminal Communications Controller
CSM	All MULTIBUS II systems
86C383	iSBC 86C38 CPU board

### NOTE

The global clocks on the iSBC 546 and iSBC 549 boards are jumpered for either the iRMX I Operating System or the iRMX II Operating System. A global clock jumpered for an iRMX I system will not run an iRMX II system without changing jumpers.

(GCN) Global Clock Name [0, 1-12 characters]
--

Use this parameter to specify the name of the Global Clock that you are configuring into your system. The name should be defined in the Terminal Communications Controller device-unit name for the battery backed-up Global Clock. If you are not including a global clock in your Multibus I system, do not specify a name.

(NPX) Numeric Processor Extension [Yes/No]	Yes
--	-----

You must specify whether your system includes an 8087, 80287, or 80387 Numeric Processor Extension. The Numeric Processor Extensions (NPX) are coprocessors that perform arithmetic operations on a variety of numeric data types. If any task contains or will contain floating-point instructions, these instructions require the numeric coprocessor for execution and you must specify "Yes" to the "Numeric Processor Extension" parameter line.

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. 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.

If your NPX is an 8087, specifying that your system contains an NPX causes the Nucleus to set up a system interrupt handler for the NPX and associate it with a specific PIC level.

If your NPX is an 8087, no task that uses the NPX should have a priority high enough to mask the interrupt that you specify for the NPX. Master PIC level zero is recommended for the 8087 NPX. The 80287 and 387™ does not use an encoded interrupt level, because the component communicates with the 80286 without using an interrupt line.

For further information about the 8087 NPX, 80287 NPX ,or 80387 NPX refer to the *Microsystem Components Handbook Volume I*. Refer to the *iRMX® I Nucleus User's Guide* for additional information concerning the relationship between interrupt levels and priorities.

(NIL) NPX Interrupt Level [Encoded] 08H

Use this parameter to specify the interrupt level connected to the 8087 NPX interrupt pin for 8086-based systems. The value that you specify must be encoded. Table 1-2 lists the encoded values for both master and slave interrupt levels. (This parameter line does not appear if you specified "80286" to the "(CPU) Processor used in the system" parameter.)

Table 1-1. Encoded Interrupt Levels

Master		Slave		Master		Slave	
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

The following example discusses how to use the table. Assume that you choose to use slave level 1 on master level 1 for your NPX. Look at the row that lists master and slave levels associated with master level 1. As you can see, the encoded values for slave levels zero through seven are listed as 0010-0017H. The correct encoded value to slave level 1 on master level 1 is 0011H.

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

Use this parameter to specify if you wish to have initialization of certain on-board functions performed (e.g., NMI Mask, 8274 channel A loop back). Table 1-1 lists the functions that are initialized.

If you use a value of 1 and your response to the "(BUS) System Bus Type [MBI/MBII]" parameter was MBI, the system performs initialization for either the iSBC 286/10(A) boards.

If you use the default of 2 and your response to the "(BUS) System Bus Type [MBI/MBII]" parameter was MBI, the system performs initialization for either the iSBC 286/12 or iSBC 386/12 boards.

If you use a value of 3 and your response to the "(BUS) System Bus Type [MBI/MBII]" parameter was MBI, the system performs initialization for either the iSBC 386/2X/3X boards.

## HARDWARE-RELATED PARAMETERS

A "No" response indicates that you are using an 80286 CPU but that you do not want the Operating System to initialize the 8255A Programmable Peripheral Interface (PPI) on your board. The 8255A allows you to use the parallel port for a line printer.

**Table 1-2. On-Board Functions that are Initialized**

Function	iSBC <sup>®</sup> 286/10(A) iSBC <sup>®</sup> 286/12 iSBC <sup>®</sup> 386/12	iSBC <sup>®</sup> 386/2X/3X
NMI	Enabled	Not Available
Gate 2 (timer 2)	Not available	Not Available
Override (lock bus)	Disabled	Not Available
LED 2	Not Available	Not Available
Non-Volatile RAM	Enabled	Not Available
LED 0	Off	Not Available
LED 1	Off	Not Available
8274 Channel A Loopback	Disabled	Not Available
Bus Drive	Disabled	Not Available
Line Printer Data Strobe	Disabled	Not Available
Cache	Not available	Enabled

**Note:** Refer to your board's hardware reference manual for details about these hardware-specific functions.

(RMB) Remote Boot Location [0=None, 40H-0FFFFH] 0H

Use this parameter to specify whether or not this system is to be booted by another system on a network (remote boot). If it is not going to be bootstrap loaded remotely, specify 0. If it is going to be bootstrap loaded remotely, specify the memory location used when configuring the iSBC 552A device driver in the first stage of the iRMX Bootstrap Loader. Note that this location is specified to the nearest 16 byte paragraph.

A value of 106H is the default value assumed in the Bootstrap Loader.

### 1.3 INTERRUPT SCREEN

Depending on what values you specify for several parameter lines on the "Hardware" screen, the ICU displays a variety of "Interrupts" screens. The following screen shows the default values from the file 28612.DEF.

```
(INT)      Interrupts

(MPT) Master PIC Type [1,2,3]           1
(MPP) Master PIC Base Port Address [0FFFFH] 0C0H
(MPS) Master PIC Port Separation [0-0FFH]  02H
(LSI) 80130 Level-Sensitive Interrupts [0-0FFH] 0H
(LSP) Local Slave PICs [0-0FFH]          0H
(IS)  Interrupt Slaves [Yes/No]          YES

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

```
(MPT) Master PIC Type [1,2,3]           1
```

Use this parameter to specify the type of Master PIC used on your processor board. The values are:

Value	Type of PIC
1	8259A
2	80130
3	80186

```
(MPP) Master PIC Base Port Address [0-0FFFFH] 0C0H
```

Use this parameter to specify the base port address of your Master Programmable Interrupt Controller (PIC). The base port address is the only address that needs to be configured for the PIC. When using Intel processor boards such as the iSBC 86/12A board and the iSBC 86/30 board, you should not change the default value.

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

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

Use this parameter to specify the interval between each PIC port. The ICU configures each additional PIC port address for you. Starting with the base port address, the ICU ascertains the address of the next port by adding your response to the "Master PIC Port Separation" parameter line to the previous address. When using Intel processor boards such as the iSBC 86/12A board and the iSBC 86/30 board, you should not change the default value.

## HARDWARE-RELATED PARAMETERS

(LSI) 80130 Level Sensitive Interrupts [0-0FFH] 0H

Use this prompt to specify any level-sensitive interrupts you wish to configure on your 80130 PIC.

There are two ways of sensing an active interrupt request: a level-sensitive input or an edge-sensitive input. The 80130 component can implement both methods on a single device. This means that a single 80130 may have one port configured as level-sensitive and the others as edge triggered. Note that the system clock must be on an edge-triggered master interrupt.

Enter this value as a bit-encoded entry where each bit represents one of the 80130 interrupt lines; bit zero represents interrupt level 0 of the 80130. Each one in the bit-encoded parameter specifies a level-sensitive interrupt.

(LSP) Local Slave PICS [0-0FFH] 0H

Use this parameter to specify which slaves attached to the 80130 Master PIC are local (rather than communicate over the MULTIBUS interface).

### NOTE

If you are using the iSBC 186/03(A) or iSBC 188/48 boards, your response should be (0,1,2,3,4,5,6,7). If you are using the iSBC 186/51 board, your response should be (0,1,2,4,5,6,7). This programs the 80130 component to provide a signal used by on-board circuitry during the Interrupt Acknowledge cycles. This is true even if the level has no slave PICs attached. However, don't specify a local slave PIC for a level if there is a slave PIC on another board in your application.

(IS) Interrupt Slaves [Yes/No] YES

Use this parameter to specify whether or not there are slave PICs connected to the master PIC. Specifying "YES" causes the "Slave Interrupt Levels" screen to be displayed.

## 1.4 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, PIC type, Level_Sensitive, Port, Separation
          [0-7],      [1,2,3]      [Yes/No]      [0-0FFFFH] [0-0FFH]

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

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

If you need to add an additional interrupt level to those you already specified, enter a new line on the screen using the rules for editing repetitive screens defined in the *Guide to the iRMX® I Interactive Configuration Utility*. See the *Component Data Catalog* for more information on PICs and edge- and level-triggered modes.

## HARDWARE-RELATED PARAMETERS

### 1.5 80186 INITIALIZATION SCREEN

This screen allows you to configure the 80186 or 80188 logic that provides programmable chip-select generation for memories and peripherals. Refer to the *iAPX 186 High Integration 16-Bit Microprocessor Data Sheet* for more information about these parameters.

(A186)	80186 Initialization	
(UGS)	Upper CS Size [0400H-040000H]	00000400H
(UCW)	Upper CS Wait States [0,1,2,3]	0000H
(UCR)	Upper CS Wait for Ready [Yes/No]	Yes
(LGS)	Lower CS Size [0,0400H-040000H]	00000000H
(LCW)	Lower CS Wait States [0,1,2,3]	0000H
(LCR)	Lower CS Wait for Ready [Yes/No]	No
(MGS)	Midrange CS Size [0,02000H-080000H]	00000000H
(MGA)	Midrange CS Base Address [0-0FE000H]	00000000H
(MCW)	Midrange CS Wait States [0,1,2,3]	0000H
(MCR)	Midrange CS Wait for Ready [Yes/No]	No
(PCS)	Peripheral CS Active [Yes/No]	Yes
(PCA)	Periph. CS Base Address [0-0FC00H]	0000H
(PCM)	Periph. CS Mapped to Memory [Yes/No]	No
(LPW)	Lower Periph. CS Wait States [0,1,2,3]	0002H
(LPR)	Lower Periph. CS Wait for Ready [Yes/No]	Yes
(UPW)	Upper Periph. CS Wait States [0,1,2,3]	0002H
(UPR)	Upper Periph. CS Wait for Ready [Yes/No]	Yes
(PLA)	Peripheral CS 5,6 Latch A1,A2 [Yes/No]	No
Enter [Abbreviation = new_value / Abbreviation ? / H ]		
:		

(UCS) Upper CS Size [0400H-040000H]	00000400H
-------------------------------------	-----------

You must specify the size of the upper memory chip select line. The value you specify must be 1K (400H), 2K (800H), 4K (1000H), 8K (2000H), 16K (4000H), 32K (8000H), 64K (10000H), 128K (20000H), or 256K (40000H). If you are using the iSBC 186/03(A) or the iSBC 186/51 processor board, it is recommended that you use the default value. If you are using the iSBC 188/48 processor board, use the 18848.DEF file as input to the ICU. Its default value is 1000H, which is appropriate for that board.

The upper limit defined by this chip select line is always 0FFFFH. The lower limit is ascertained by the ICU as the upper limit less the value specified for this parameter line.

(UCW) Upper CS Wait States [0,1,2,3]	0000H
--------------------------------------	-------

You must specify the number of wait states for all accesses to the upper memory chip select line. The value you select can be from zero to three. If you are using the iSBC 186/03(A), the iSBC 186/51, or the iSBC 188/48 processor board, it is recommended that you use the default value.

(UCR) Upper CS Wait for Ready [Yes/No]	Yes
--	-----

You must select whether or not the 80186 should ignore external READY for the upper memory chip select line. If you specify "Yes", the 80186 will wait for the number of wait states specified or will wait for an external READY condition. If you specify "No", the 80186 will wait for the number of wait states specified but will **not** wait for an external READY condition. If you are using the iSBC 186/03(A), the iSBC 186/51, or the iSBC 188/48 processor board, it is recommended that you use the default value.

(LCS) Lower CS Size [0,0400H-040000]	00000000H
--------------------------------------	-----------

In response to the "Lower CS Size" parameter line, you must specify a value of zero or the size of the lower memory chip select line. The value of zero indicates that you do not intend to program the lower memory chip select line. Any non-zero value you specify must be 1K (400H), 2K (800H), 4K (1000H), 8K (2000H), 16K (4000H), 32K (8000H), 64K (10000H), 128K (20000H), or 256K (40000H). If you are using the iSBC 186/03(A), the iSBC 186/51, or the iSBC 188/48 processor board, it is recommended that you use the default value.

The lower limit defined by this chip select line is always 00000H. The upper limit is ascertained by the ICU as the lower limit plus the value specified for this parameter line.

(LCW) Lower CS Wait States [0,1,2,3]	0000H
--------------------------------------	-------

If you specified a non-zero value for the "Lower CS Size" parameter line, you must specify the number of wait states for all accesses to the lower memory chip select line. The value you select can be from zero to three. If you are using the iSBC 186/03(A), the iSBC 186/51, or the iSBC 188/48 processor board, it is recommended that you use the default value.

## HARDWARE-RELATED PARAMETERS

(LCR) Lower CS Wait for Ready [Yes/No]	No
--	----

If you specified a non-zero value for the "Lower CS Size" parameter line, you must select whether or not the 80186 should ignore external READY for the lower memory chip select line. If you specify "Yes", the 80186 will wait for the number of wait states specified or will wait for an external READY condition. If you specify "No", the 80186 will wait for the number of wait states specified but will **not** wait for an external READY condition. If you are using the iSBC 186/03(A), the iSBC 186/51, or the iSBC 188/48 processor board, it is recommended that you use the default value.

(MCS) Midrange CS Size [0,02000H-080000H]	00000000H
---	-----------

In response to the "Midrange CS Size" parameter line, you must specify a value of zero or the size of the midrange memory chip select line. The value of zero indicates that you do not intend on programming the midrange memory chip select line. Any non-zero value you specify must be 8K (2000H), 16K (4000H), 32K (8000H), 64K (10000H), 128K (20000H), 256K (40000H), or 512K (80000H). If you are using the iSBC 186/03(A), the iSBC 186/51, or the iSBC 188/48 processor board, it is recommended that you use the default value.

The 80186 provides four midrange memory chip select lines. Your response to this parameter sets the total size of the memory block defined by the four midrange select lines. The size of any one midrange memory chip select line is one-fourth of the total. The lower limit defined by this chip select line is defined by the "Midrange CS Base Address". The upper limit is ascertained by the ICU as the lower limit plus the value specified for this parameter line.

(MCA) Midrange CS Base Address [0-0FE000H]	00000000H
--	-----------

If you specify a non-zero value for the "Midrange CS Size" parameter line, you must specify the base address of the midrange memory chip select lines. Otherwise, specify a value of zero. If you are using the iSBC 186/03(A), the iSBC 186/51, or the iSBC 188/48 processor board, it is recommended that you use the default value.

You must set the base address at any integer multiple of the size of the total memory block selected. For example, if you specified a total block size of 32K for the previous parameter (MCS), you must select a base address of 10000H or 18000H but not 14000H.

If you specify MCS=080000H for the previous parameter line, you must also specify the base address to be 00000H and the "Lower CS Size" parameter to be zero.

(MCW) Midrange CS Wait States [0,1,2,3]	0000H
---	-------

If you specified a non-zero value for the "Midrange CS Size" parameter line, you must specify the number of wait states for all accesses to the midrange memory chip select lines. Otherwise, specify a value of zero. The value you select can be from zero to three. If you are using the iSBC 186/03(A), the iSBC 186/51, or the iSBC 188/48 processor board, it is recommended that you use the default value.

(MGR) Midrange CS Wait for Ready [Yes/No]	No
---	----

If you specified a non-zero value for the "Midrange CS Size" parameter line, you must select whether or not the 80186 should ignore external READY for the midrange memory chip select lines. Otherwise, specify a value of zero. If you specify "Yes", the 80186 will wait for the number of wait states specified or will wait for an external READY condition. If you specify "No", the 80186 will wait for the number of wait states specified but will not wait for an external READY condition. If you are using the iSBC 186/03(A), the iSBC 186/51, or the iSBC 188/48 processor board, it is recommended that you use the default value.

(PGS) Peripheral CS Active [Yes/No]	Yes
-------------------------------------	-----

You must specify "Yes" to the "Peripheral CS Active" parameter if you are going to use at least one peripheral chip select line. Otherwise, respond with a "No". The 80186 can generate chip selects for up to seven peripheral devices on peripheral chip select lines zero through six. If you are using the iSBC 186/03(A), the iSBC 186/51, or the iSBC 188/48 processor board, it is recommended that you use the default value.

(PGA) Periph. CS Base Address [0-0FC00H]	0000H
--	-------

If you specified "Yes" to the "Peripheral CS Active" parameter, you must specify the base address of the peripheral chip selects. These chip selects are active for seven contiguous blocks of 128 bytes above the peripheral chip select base address. The base address can only be a multiple of 1K bytes. That is, the least significant 10 bits of the base address must be zero. If you are using the iSBC 186/03(A), the iSBC 186/51, or the iSBC 188/48 processor board, it is recommended that you use the default value.

(PCM) Periph. CS Mapped to Memory [Yes/No]	No
--	----

If you specified "Yes" to the "Peripheral CS Active" parameter, you must specify whether the base address is located in memory or in I/O space. A "Yes" response to this parameter indicates that the base address is located in memory; a "No" response indicates that the base address is located in I/O space. If you are using the iSBC 186/03(A), the iSBC 186/51, or the iSBC 188/48 processor board, it is recommended that you use the default value.

## HARDWARE-RELATED PARAMETERS

(LPW) Lower Periph. CS Wait States [0,1,2,3] 0002H

If you specified "Yes" to the "Peripheral CS Active" parameter, you must specify the number of wait states for all accesses to the peripheral chip selects 0-3. The value you select can be from zero to three. If you are using the iSBC 186/03(A), the iSBC 186/51, or the iSBC 188/48 processor board, it is recommended that you use the default value.

(LPR) Lower Periph. CS Wait for Ready [Yes/No] Yes

If you specified "Yes" to the "Peripheral CS Active" parameter, you must select whether or not the 80186 should ignore external READY for the peripheral chip selects 0-3. If you specify "Yes", the 80186 will wait for the number of wait states specified or will wait for an external READY condition. If you specify "No", the 80186 will wait for the number of wait states specified but will **not** wait for an external READY condition. If you are using the iSBC 186/03(A), the iSBC 186/51, or the iSBC 188/48 processor board, it is recommended that you use the default value.

(UPW) Upper Periph. CS Wait States [0,1,2,3] 0002H

If you specified "Yes" to the "Peripheral CS Active" parameter, you must specify the number of wait states for all accesses to the peripheral chip selects 4-6. The value you select can be from zero to three. If you are using the iSBC 186/03(A), the iSBC 186/51, or the iSBC 188/48 processor board, it is recommended that you use the default value.

(UPR) Upper Periph. CS Wait for Ready [Yes/No] Yes

If you specified "Yes" to the "Peripheral CS Active" parameter, you must select whether or not the 80186 should ignore external READY for the peripheral chip selects 4-6. If you specify "Yes", the 80186 will wait for the number of wait states specified or will wait for an external READY condition. If you specify "No", the 80186 will wait for the number of wait states specified but will **not** wait for an external READY condition. If you are using the iSBC 186/03(A), the iSBC 186/51, or the iSBC 188/48 processor board, it is recommended that you use the default value.

(PLA) Peripheral CS 5,6 Latch A1,A2 [Yes/No] No

If you specified "Yes" to the "Peripheral CS Active" parameter, you must specify "Yes" or "No" to this parameter line. Specify "Yes" if peripheral chip selects five and six are to provide latched address bits A1, A2. Specify "No" if they are not.

## 1.6 MULTIBUS® II HARDWARE SCREEN

The MULTIBUS II hardware screen enables you to supply information about your MULTIBUS II boards. To get to MBII hardware screen shown below, select the MBII architecture (BUS = 2) from the Hardware Screen. As with the MBI architecture, you will see some intermediate screens before the MBII hardware screen is displayed.

```

MBII Hardware

(MDP) Message Device Base Port [0-0FFFFH]    0000H
(MDS) Message Device Port Separation [0-0FFH] 0004H
(MDL) Message Interrupt Level [0-7]          0002H

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

```

(MDP) Message Device Base Port [0-0FFFFH]    0000H
    
```

Use this parameter to specify the message device's base port I/O address. The message device is the message interrupt controller (MIC) chip. The MIC chip facilitates interprocessor communication in a MULTIBUS II system by generating and receiving MBII signals for its resident processor.

### NOTE

You should not change the value of this parameter from its default unless you are using a custom-built board.

```

(MDS) Message Device Port Separation [0-0FFH] 0004H
    
```

Use this parameter to specify the interval (in bytes) between I/O ports on the message device (the MIC chip) on the MULTIBUS II board. The MIC chip facilitates interprocessor communication in a MULTIBUS II system by generating and receiving iRMX I MBII signals for its resident processor. The ICU determines the address of the next port by adding the value of this parameter to the previous address.

### NOTE

Do not change the value of this parameter from its default value if you are using an Intel-supplied processor board. Change the value only if you are using a custom-built processor board.

## HARDWARE-RELATED PARAMETERS

(MDL) Message Interrupt Level [0-7]	0002H
-------------------------------------	-------

Use this parameter to specify the master PIC interrupt level associated with the message device (the MIC chip) on the MULTIBUS II board. (Note that this is not an encoded interrupt level.) This interrupt level is used for interboard communication via message-based signals.

It is recommended that you do not change this value from its default of 0002H.

# MEMORY PARAMETERS 2

---

## 2.1 INTRODUCTION

This chapter discusses how to respond to the two prompts that appear on the "Memory" screen.

## 2.2 MEMORY PROMPTS

The following screen shows the values from the 28612.DEF definition file. Note that values shown on the screens are sample values only. The exact values on each screen change according to the definition file used and your own changes.

```
(RAMEM) RAM Memory

      Memory =      low      ,      high
                [0-0FFFFH]  [0-0FFFFH]
[ 1] Memory =      0120H    ,      0DFFFH
[ 2] Memory =

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

By responding to the prompts on this screen, you can define the distinct contiguous blocks of RAM and/or ROM that you want the Operating System to manage. The second stage of the ICU (discussed in the *Guide to the iRMX® I Interactive Configuration Utility*) locates system code in these blocks of memory.

You must enter both the start and end addresses of each block of memory. The ICU interprets each address as the base portion of a 20-bit address where the offset is zero. The first number in the pair must be smaller than the second, and the specified block must not overlap any previously specified block. All numbers must be greater than 40H. The ICU allows a maximum of 20 blocks of RAM and 20 blocks of ROM.

## MEMORY PARAMETERS

Most users can use the default values the first time the system is configured. However, the remainder of this chapter explains which users might need to change the default values the first time.

The reasons why you might change the default values after you have generated your first system are explained in the *Guide to the iRMX® I Interactive Configuration Utility*.

### 2.2.1 RAM

All of the Intel-supplied definition files define RAM-based systems. These systems use the RAM specified on the "Memory" screen for two purposes. First, the RAM is used to store system code. Second, the RAM is used to provide free space to the Operating System.

If you are not using one of the supplied definition files or you have made changes to one of the supplied definition files (described in the *Guide to the iRMX® I Interactive Configuration Utility*), you may need to change the default value for RAM. Four possible situations in which you might want to change the default value are described in the following sections.

#### 2.2.1.1 Upper Memory Address Considerations

If your system does not contain the same amount of memory as specified by the upper RAM address, you can change this upper address. The value you use should reflect the maximum memory your system can address. However, if the value you use is greater than the maximum memory your system can address, the Nucleus performs a memory scan and sets this limit to the memory actually present in your system.

#### 2.2.1.2 80186 Considerations

If you are using the 80186 processor in your application and are not using the iSDM 86 monitor, the upper 1K of memory locations (locations 0FFC00H) to 0FFFFFFH) may need to be left free for the 80186 initialization. This requires that you not declare this block of ROM, even though it exists in the system.

If your system includes the iSDM 86 monitor, the monitor will initialize the 80186 processor.

#### 2.2.1.3 Communications Board Considerations

Communication boards, such as the iSBC 544 and the iSBC 188/48 boards (when configured as communication 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" screen.

A default value on the "iSBC 544 Driver" screen allows the on-board memory to start at 0E000H; the default for the "iSBC 188/48 Driver" screen starts the on-board memory at 0E400H. Using these default values dictates that the upper RAM address on the "Memory" screen should be 0DFFFH if you use the iSBC 544 driver (or 0E3FFH if you include only the iSBC 188/48 driver). If you change the "Memory Address Base" parameter lines on the "iSBC 544 Driver" and "iSBC 188/48 Driver" screens, you must also reflect that change on the "Memory" screen. Refer to the individual driver sections for more information about the iSBC 544 and iSBC 188/48 drivers.

#### 2.2.1.4 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" screen.

### 2.2.2 ROM

```
(ROMEM) ROM Memory

      Memory = low,          high
              [0-0FFFFH]  [0-0FFFFH]
[ 1] Memory =

Enter Changes [ Number = new_value / ^D Number / ? 1 / H]
:
```

If your application system includes any code burned into ROM, you need to specify the ROM addresses for this code. However, until you are ready to test a PROM-based system in RAM (explained in the *Guide to the iRMX® I Interactive Configuration Utility*) you do not need to use the "ROM" prompt. An example is given in the *Guide to the iRMX® I Interactive Configuration Utility*, you do not need to use the "ROM" prompt.

#### 2.2.2.1 Monitor Considerations

The code for your monitor or stand-alone Bootstrap Loader is probably burned into PROM. Since the Operating System should not have access to this memory, do not include the location of your monitor code (or stand-alone Bootstrap Loader) within a declared block of memory.



# SUB-SYSTEMS PARAMETERS

# 3

## 3.1 INTRODUCTION

This chapter discusses how to respond to the parameters that appear on the Sub-Systems screen. The iRMX I Operating System consists of the Nucleus, which must be included in your application system, 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 couple of broad impacts. First, specific capabilities are added to your system. Second, adding a subsystem increases your memory requirements.

```
(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
(DDb) Dynamic Debugger [Yes/No]               NO
(THD) Terminal Handler [Yes/No]               NO

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

The software that you create runs in an iRMX application system using the facilities of the Nucleus and any needed subsystem. Each optional subsystem requires at least one other subsystem or the Nucleus to be configured in the Operating System. Table 3-1 shows the dependency of one 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)					
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 iRMX® Operating Systems* for additional information on subsystems.

### 3.2.1 Universal Development Interface

The Universal Development Interface (UDI) is an optional interface to 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 I Operating System. Refer to the *iRMX® Universal Development Interface User's Guide* for more 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 running under the supervision of the Nucleus. 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 *iRMX® 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 can load absolute code into fixed locations, relocatable code into dynamically allocated memory locations, and it can load files containing overlays.

The Application Loader is a layer that requires the Basic I/O System and may require the Extended I/O System. (Your response to the "Load Job Type" parameter line on the "Application Loader" screen determines the need for the Extended I/O System.) Refer to the *iRMX® I 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 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®-Net Software User's Guide* for more information about the iRMX-NET network.

### 3.2.5 Extended I/O System

The Extended I/O System is an optional subsystem that adds high-level, synchronous file access capabilities for software running under the supervision of the Nucleus. It requires the capabilities of the Basic I/O System. It also 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 *iRMX® 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 for software running under the supervision of the Nucleus. 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 *iRMX® 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 System Debugger (SDB) extends the use of the monitor so that you can interactively examine data structures handled by the iRMX Operating Systems. You can use both the System Debugger and the Dynamic Debugger in the same system.

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

### 3.2.8 Dynamic Debugger

The Dynamic Debugger, also called the iRMX I Debugger, is an optional subsystem that provides a facility for debugging and monitoring software running under the supervision of the Nucleus. The Dynamic Debugger allows a software engineer to dynamically examine the data structures handled by the iRMX I Operating System. This capability permits you to easily debug a multitasking operation. The Dynamic Debugger supplies its own Terminal Handler. Your application software can make use of the Dynamic Debugger's Terminal Handler, or you can include a separate version (or versions) of the Terminal Handler in your system configuration for application use. Refer to the *iRMX® I Dynamic Debugger Reference Manual* for more information.

If you plan on using any of the features of the iRMX I Debugger, specify "Yes" to the "Dynamic Debugger" parameter line. You can use both the System Debugger and the Dynamic Debugger in the same system.

### 3.2.9 Terminal Handler

The Terminal Handler is an optional subsystem that provides a real-time interface between your terminal and other software running under the supervision of the Nucleus. It is intended for use in systems that do not include the Basic I/O System. Refer to the *iRMX® I Terminal Handler Reference Manual* for additional details.

If you plan on using the Terminal Handler, specify "Yes" to the "Terminal Handler" parameter line. If you change the default value of the "(DDB) Dynamic Debugger [Yes/No]" to "Yes", the "(TH) Terminal Handler [Yes/No]" prompt will change to "Req".



# HUMAN INTERFACE PARAMETERS

# 4

## 4.1 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. The page that is referenced contains a discussion of the first parameter line contained on that screen. The remaining parameter lines are discussed in the order they appear on the screen.

### Human Interface

(ICL)	(CNM)	(SYS)	(RIP)	Start on page 4-5
(UXC)	(SS)	(PMI)	(PMA)	
(DTN)	(RU)			

### HI Jobs

(MIN)	(MAX)	(NPX)	Start on page 4-9
-------	-------	-------	----------------------

### HI Logical Names

Start on  
page 4-10

### Resident/Recovery User

(TN)	(TDN)	(MTP)	(UID)	(MIN)	Start on page 4-11
(MAX)	(IPP)	(DEF)			

### Prefixes

Start on  
page 4-13

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.

### 4.2 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 people and an iRMX 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® Basic I/O System User's Guide*. An iRMX application system can incorporate one or more users, and the configuration considerations are different for single-user systems than for multiple-user systems.

#### 4.2.1 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 in iRMX named files rather than in memory with the Operating System (thus the term "non-resident user"). Non-resident configuration of the Human Interface is discussed in the *iRMX® I Interactive Configuration Utility User's Guide*.

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 user 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 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, each must load their own private copy of that program. 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 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. The disadvantage of doing this is that if your system device or your non-resident configuration files become damaged, none of your users will be able to access the system.

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 and doesn't have to expend the memory required to support the SUPER command. 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.

### 4.2.2 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 command.

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:SYSTEM/DATAFILE`. That would allow you to use `:DATA:` in place of `:SD:SYSTEM/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® Human Interface*.

### 4.3 THE SCREENS

If you elect to include the Human Interface in your application system, the ICU automatically includes in your system, 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.

### 4.3.1 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 characters]
:SD:SYSTEM
(RIP) Resident Initial Program [IntelCLI/1-45 characters]
INTELCLI
(UXC) User Extension for Intel CLI [1-45 Chars]

(SS)  Initial Program Stack Size [0-0FFFFH]       02400H
(PMI) Human Interface Pool Minimum [0-0FFFFH]     0260H
(PMA) Human Interface Pool Maximum [0-0FFFFH]     0FFFFH
(DTN) Default Terminal Name [1-6 Chars]           VT100
(RU)  Resident User [Yes/Recovery/None]            NONE

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 into 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.

## HUMAN INTERFACE PARAMETERS

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

This parameter line 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 characters]	:SD:SYSTEM
--	------------

This parameter line 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:SYSTEM specifies :SD: as the system device and SYSTEM 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 [Default/1-45 characters]	INTELCLI
--	----------

This parameter line lets you specify the resident initial program. You can respond in either of two ways. The default response, which is "Default", 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 linked to the application during system generation. An alternate resident initial program can replace the Human Interface CLI if you take the following steps:

- 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 linked (using LINK86) 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 parameters. For more information on user extensions, see the *iRMX® 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 [0-0FFFFH]      0260H
(PMA) Human Interface Pool Maximum [0-0FFFFH]      0FFFFH

These parameter lines let you specify the minimum and maximum sizes, in paragraphs, of the memory pool of the Human Interface. Unless you plan to include in your system 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 0FFFFH. 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
- SDB (If included in the system)
- Basic I/O System
- Application Loader
- Extended I/O System (including I/O jobs)
- Universal Development Interface (if included in system)
- First-level user jobs
- Human Interface

By setting PMA to 0FFFFH, 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, and 02400H is usually sufficient.

If you include in your system a first-level user job that borrows memory, you might not want to set PMA to 0FFFFH. 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.

## HUMAN INTERFACE PARAMETERS

(DTN) Default Terminal Name [1-6 Chars]	VT100
---	-------

This parameter provides a default terminal name for the entire system. It is used only if no terminal 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 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® I Interactive Configuration Utility*.

(RU) Resident User [Yes/Recovery/None]	NONE
--	------

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.

### 4.3.2 Human Interface Jobs Screen

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

```
(HIJOBS)           HI Jobs

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

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

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

When an operator enters a Human Interface command, the Operating System invokes an Application Loader system call (either A\$LOAD\$I/O\$JOB or S\$LOAD\$I/O\$JOB) 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 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 0FFFFH allows HI jobs to borrow memory. Refer to the *iRMX<sup>®</sup> Application Loader User's Guide* for more information about the pool\$lower\$bound and pool\$upper\$bound parameters.

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

This parameter lets you specify whether any of the HI jobs that are created contain floating-point instructions and therefore use an 8087, 80287 or 387™ Numeric Processor Extension (NPX).

### 4.3.3 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:UTILS
[ 3] Name = LANG          , :SD:LANG
[ 4] Name = ICU           , :SD:RMX86/ICU
[ 5] Name = RMX           , :SD:RMX86
[ 6] Name =

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

You can edit this screen by the methods outlined for repetitive screens in *Guide to the iRMX® I 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.

### 4.3.4 Resident 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 [1-12 Characters]  T0
(MTP) Maximum Task Priority [0-OFFH]       192
(UID) User ID Number [0-OFFFFH]          OFFFFH
(MIN) Minimum Memory Required [0-OFFFFH]   02000H
(MAX) Maximum Memory Required [0-OFFFFH]   OFFFFH
(IPP) Initial-Program Pathname [Resident/1-45 Characters]
                                           RESIDENT
(DEF) Default Directory [1-45 characters]
                                           :SD:
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 cannot be opened. For a more detailed description of the :CONFIG:TERMINALS files and a list of the recognized terminal names, see the *Guide to the iRMX I Interactive Configuration Utility*.

```
(TDN) Terminal Device Name [1-12 Characters]  T0
```

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-OFFH]       192
```

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

## HUMAN INTERFACE PARAMETERS

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

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 default 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® Basic I/O System User's Guide* for a discussion of users, user IDs, WORLD, and the system manager.

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

During the initialization phase of your application, the Operating System creates an I/O job for the resident user's terminal and also for each terminal (if any) that has an associated non-resident user that you define during non-resident configuration. These parameters let you specify the amount of memory that the I/O job for the resident user's terminal is to have. The default 02000H 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 0FFFFH, then set MAX to 0FFFFH, 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 the Human Interface has for them.

(IPP) Initial-Program Pathname [Resident/1-45 Characters]	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 characters]           :SD:
```

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: 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.

### 4.3.5 Prefixes

Prefixes are logical names of directories that the Human Interface searches through when an operator enters a command with a file name that does not contain a directory. 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 = :ICU:
[ 6] Prefix = :$:
[ 7] Prefix =

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

You can edit this screen by the methods outlined for repetitive screens in the *Guide to the iRMX® I Interactive Configuration Utility*. 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 96 prefixes.



# APPLICATION LOADER PARAMETERS

# 5

## 5.1 APPLICATION LOADER SCREEN

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

```
(APPL) Application Loader

(IBS) Internal Buffer Size [0-0FFFFH]          0400H
(RBS) Read Buffer Size [0-0FFFFH]            01000H
(LJT) Load Job Type [None/Async/Sync]        SYNC
(DMP) Default Memory Pool Size [0-0FFFFH]    0800H
(CT) Code Type [Abs/Pic/Ltl/Ovr]            OVR
(CC) Compute Checksum during load [Yes/No]    NO

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

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

Use this parameter to specify the size of the Application Loader's internal buffer. LINK86 and LOC86 usually write records that are up to 1024 bytes long.

### NOTE

LINK86 creates records longer than 1024 bytes when linking certain FORTRAN programs. When you try to load a program which contains a record length greater than your response to this parameter, the Operating System returns an E\$REC\$LENGTH exception code. Doubling the size of your response to this parameter should correct this problem.

If you write a program that produces object code (like a custom locator) and you intend to use the Application Loader to load the output of this program, you can set this parameter to a value appropriate to your program. However, if you specify a value smaller than 1024, the Application Loader cannot load output from LINK86 and LOC86.

## APPLICATION LOADER PARAMETERS

(RBS) Read Buffer Size [0 - 0FFFFH]	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.

(LJT) Load Job Type [None/Async/Sync]	SYNC
---------------------------------------	------

This parameter lets you specify which of the A\$LOAD, A\$LOAD\$IO\$JOB, and S\$LOAD\$IO\$JOB system calls are to be included in your system. The values and their meanings are as follows:

- None Indicates only the A\$LOAD system call.
- Async Indicates only the A\$LOAD and A\$LOAD\$IO\$JOB system calls.
- Sync Indicates all three of the A\$LOAD, A\$LOAD\$IO\$JOB, and S\$LOAD\$IO\$JOB system calls.

In general, if you want your tasks to have control over how the loading of I/O jobs is to be synchronized with other system activities, specify Async. However, if you want the system to manage synchronization for you, specify Sync.

If you have already indicated to the ICU that your system will incorporate the Human Interface, this screen displays "SYNC" in place of the default value for this parameter. This means that all three loading system calls are required. That is, the only acceptable value for this parameter is Sync, and you cannot change it.

(DMP) Default Memory Pool Size [0 - 0FFFFH]	0800H
---	-------

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.

There are three ways to specify the minimum and maximum memory pool sizes of jobs created by S\$LOAD\$IO\$JOB and A\$LOAD\$IO\$JOB:

- Explicit specification of the "pool\$lower\$bound" and "pool\$upper\$bound" parameters in calls to S\$LOAD\$IO\$JOB and A\$LOAD\$IO\$JOB. Refer to the *iRMX® I Application Loader System Calls Reference Manual* for more information about these parameters. Note that these parameters are not the same as the pool\$min and pool\$max parameters of CREATE\$JOB and CREATE\$IO\$JOB.

- Use of the MEMPOOL control of LINK86 when loading load-time locatable code (LTL) or position-independent code (PIC). Refer to the *iAPX 86,88 Family Utilities User's Guide* for more information about the MEMPOOL control.
- The DMP parameter of the Application Loader screen.

Using the MEMPOOL offers the greatest flexibility of the three options, because you can use it to assign memory pool sizes on a program-by-program basis. Assigning memory pool sizes in the system calls is also flexible. Least flexible is the use of the DMP configuration parameter, but there is one situation in which you should use this option.

If you use LOC86 to produce absolute code that is to be loaded later, you cannot use the MEMPOOL control of LINK86. Moreover, LOC86 does not have such a control, so with absolute code you cannot specify the memory requirements within the code itself.

If you load the absolute code via Human Interface commands, the Human Interface makes the call to A\$LOAD\$IO\$JOB or S\$LOAD\$IO\$JOB for you, so you must specify during configuration the size of the memory pool that is requested. One way to do this would be by the MIN and MAX parameters on the "HI Jobs" screen. However, Intel recommends that instead you select the default values for those parameters and use this, the DMP parameter of the "Application Loader" screen.

If the situation just described applies to your system, assign a DMP value that is an adequate memory pool size for the absolute code you intend to load. The default value of 0800H paragraphs (32768 bytes) is adequate for most situations. However, if your absolutely located programs create segments or use buffered I/O via the Extended I/O System or the UDI, you must select a larger DMP value.

To compute this value, add to the default value the number of paragraphs of memory used for segments and buffers.

When the Application Loader loads code, it compares the DMP value with, if applicable, the MIN value from the "Human Interface Jobs" screen, or with, if applicable, the value obtained from the code itself. It uses the larger of the two compared values as the memory pool size for the loaded code.

(CT) Code Type [Abs/Pic/Ltl/Ovr]	OVR
----------------------------------	-----

Use this parameter to specify which kinds of code (absolute, position-independent (PIC), load-time locatable (LTL), or overlay) the Application Loader can load. In this way, you can insure that the Application Loader is configured to load the code that your application requires. The four possible responses and their meanings are as follows:

- Abs: Your Application Loader can load only absolute code.
- Pic: Your Application Loader can load absolute code or PIC (Position-Independent Code).

## APPLICATION LOADER PARAMETERS

- **Ltl:** Your Application Loader can load absolute, PIC, or LTL code.
- **Ovr** Your Application Loader can load absolute, PIC, LTL, or overlay code.

If you have already indicated to the ICU that your system will incorporate the Human Interface, this screen displays "OVR." This means that your Application Loader must be prepared to load any of the four kinds of code. That is, the only acceptable value for this parameter is OVR, and you cannot change it.

(CC) Compute Checksum During Load [Yes/No]	NO
--	----

Use this parameter to indicate if you wish the Application Loader to compute checksums during program loads.

Computing checksums during program loads will ensure the integrity of the program, but will decrease the speed with which the Application Loader performs. Many modern device controllers calculate the checksum. In systems using such devices, the Application Loader performs a redundant operation.

## REMOTE FILE ACCESS PARAMETERS

### 6.1 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 I Operating System. If you have iRMX-NET and you want your system to access remote files (be a file consumer), answer "Yes" to the "RFA" parameter on the "Subsystems" screen. 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®-Net Software User's Guide* for more information about the iRMX-NET software.

### 6.2 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 [0-0FFH]          082H
(NOR) Number of Outstanding RFD system calls [0-0FFH] 0AH
(NOS) Number of Outstanding RFD status calls [0-0FFH] 01H
(JEI) Job Exit Interval [0-0FFFFFFFH]      01H
(LI) Logoff Interval [0-0FFFFFFFH]        014H
(CBT) Configuration Base Time [0-0FFFFFFF)  0H

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

## REMOTE FILE ACCESS PARAMETERS

(ITP) I/O Task Priority [0-0FFH]

082H

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®-Net Software User's Guide*. The default value is recommended for most systems.

(NOR) Number of Outstanding RFD system calls [0-0FFH]

0AH

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 [0-0FFH]

01H

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 [0-0FFFFFFFH]

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

(LI) Logoff Interval [0-0FFFFFFFH]

014H

This parameter sets the 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 [0-0FFFFFFF]

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

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

### 6.3 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".

## REMOTE FILE ACCESS PARAMETERS

### 6.4 REMOTE FILE SERVER 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 = 03H 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 Characters]    DEFAULT
(NFR) Number of Fragmentation Buffers [0-0FFFFH]   03H
(MBF) Maximum Number of Buffers per File [0-0FFH]  0FFH

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

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

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

```
(NFR) Number of Fragmentation Buffers [0-0FFFFH]   03H
```

#### 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" command on the Device-Unit Information screens.

(MBF) Maximum Number of Buffers per File [0-0FFH]	0FFH
---	------

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 0FFH 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 0FFH. If the applications are I/O intensive (copying files, for example), set the MBF parameter to 0. If the applications are mixed, choose 0FFH.



# EXTENDED I/O SYSTEM PARAMETERS

# 7

## 7.1 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 page 7-2
(PMI)	(PMA)	(CD)		

### Automatic Boot Device Recognition Parameters

(DLN)	(DPN)	(DFD)	(DO)	Start on page 7-5
-------	-------	-------	------	----------------------

### Logical Names Parameters

Start on  
page 7-8

### I/O Users Parameters

Start on  
page 7-9

### I/O Jobs Parameters

(IJD)	(DU)	(PMI)	(PMA)	Start on page 7-11
(AEH)	(EM)	(PV)	(TP)	
(TSA)	(DSB)	(SSA)	(SSI)	
(NPX)				

## EXTENDED I/O SYSTEM PARAMETERS

### 7.2 EIOS SCREEN

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

```
(EIOS)          EIOS

(ASC) All Sys Calls in EIOS [Yes/No]          REQ
(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 [0-0FFFFH]            0180H
(PMA) EIOS Pool Maximum [0-0FFFFH]            0180H
(CD) Configuration Directory [1-45 Chars]

                                           :SD:RMX86/CONFIG

Enter [ Abbreviation - new_value / Abbreviation ? / H ]
```

```
(ASC) All System Calls in EIOS [Yes/No]          REQ
```

Use this parameter to specify whether all, or only some, of the EIOS system calls are to be included in your application system. The possible values are "Yes" (the default) and "No." If you select "No," the ICU gives you the opportunity to select individually the system calls that you want in your configuration by displaying, as the next screen, a menu of EIOS system calls. Any EIOS system calls that are not represented on that screen are required.

If you have already indicated to the ICU that your system will incorporate the Universal Development Interface (UDI) or the Human Interface, this screen displays "REQ" in place of the default value for this parameter. This means that your application must include all EIOS system calls. That is, the only acceptable value for this parameter is REQ, and you cannot change it.

```
(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® 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.

## EXTENDED I/O SYSTEM PARAMETERS

(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 [0-0FFFFH]	0180H
(PMA) EIOS Pool Maximum [0-0FFFFH]	0180H

These parameters specify the minimum and maximum sizes, in 16-byte paragraphs, of the EIOS memory pool. 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.

Follow these guidelines when setting the PMI and PMA values:

- If your system has no I/O jobs, accept the default value 0180H (equivalent to 6144 bytes) for both parameters.

- If your system does not include either the Human Interface or first-level jobs that borrow memory and it does include I/O jobs, set PMA to 0FFFFH and leave PMI at 0180H. This allows the EIOS to borrow memory from the root job as needed.
- If your system includes the Human Interface or has a first-level job that borrows memory, set PMI equal to PMA, to prevent the EIOS from borrowing memory needed by other jobs. If your system has I/O jobs, you should add their combined memory requirements of the default 0180H value for both PMI and PMA.

```
(CD) Configuration Directory [1-45 Chars] :SD:RMX86/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:.

### 7.3 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]     W0
(DFD) Default System Device File Driver [Phys/Str/Named/Remote] NAMED
(DO)  Default System Device Owner's ID [0-0FFFFH]          OH

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

## EXTENDED I/O SYSTEM PARAMETERS

(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] W0

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<sup>®</sup> Extended I/O System User's Guide* for more information about file drivers.

(D0) 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.

## 7.4 LOGICAL NAMES SCREEN

The following screen lets you 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, OH
[2] Logical Name = STREAM , STREAM      , STREAM  , OH
[3] Logical Name = LP     , LP          , PHYSICAL, OH
[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® I 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 96 logical names on these screens, including the three that are already defined there.

## 7.5 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 96 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],      [0-0FFFFH]
[1] 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 *iRMX® Interactive Configuration Utility User's Guide*. 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.

### 7.6 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.

## 7.7 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.

```

(IJOB)      I/O Jobs

(IJD) I/O Job Default Prefix [1-12 Chars]
(DU)  Default User [1-12 Chars]
(PMI) Pool Minimum [20H - 0FFFFH]          0260H
(PMA) Pool Maximum [20H - 0FFFFH]          0FFFFH
(AEH) Address of Exception Handler [CS:IP]  0000:0000H
(EM)  Exception Mode [Never/Prog/Environ/All] NEVER
(PV)  Parameter Validation [Yes/No]        YES
(TP)  Task Priority [0-255]                 155
(TSA) Task Start Address [CS:IP]           0000:0000H
(DSB) Data Segment Base [0-0FFFFH]        0H
(SSA) Stack Segment Address [SS:SP]       0000:0000H
(SSS) 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.

## EXTENDED I/O SYSTEM PARAMETERS

(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 - 0FFFFH] 0260H  
(PMA) Pool Maximum [20H - 0FFFFH] 0FFFFH

These parameters let you specify the minimum and maximum sizes, in 16-byte paragraphs, of this I/O job's memory pool. 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.)

If your system includes the Human Interface, you should set the minimum pool size equal to the maximum pool size, to prevent this job from using memory needed by the Human Interface.

(AEH) Address of Exception Handler [CS:IP] 0000:0000H

Use this parameter to specify the start address of the I/O job's exception handler. A value of 0:0 indicates that the I/O job uses the default system exception handler. A base address of 900H and offset address of 384H should be entered as 900H:384H.

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 a handler other than the default system exception handler, this parameter allows you to specify the start address of this other exception handler.

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

If you specified a non-zero value for the "Address of Exception Handler" 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 Nucleus Parameters chapter. 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 Start Address [CS:IP]	0000:0000H
----------------------------------	------------

Use this parameter to specify the start address of this job's initialization task. A response to this parameter indicates that you have decided how to use your system's RAM and/or ROM (as defined on you "Memory" screen). Your response also indicates that you have already compiled, linked, and located your application code, or that you know where LOC86 will locate your last system module.

## EXTENDED I/O SYSTEM PARAMETERS

Deciding where you want to locate your I/O job(s) is a decision that should be based on a number of factors. These factors affect how you respond to the "Memory" screen and are discussed in the Memory Parameters chapter.

If you have already compiled, linked and located your application code, LOC86 created a memory map that lists the base and offset address of the first instruction of this I/O job's initialization task. Use these addresses as your response to this parameter. The *Guide to the iRMX® I Interactive Configuration Utility* describes how to link and locate your jobs.

If you have not compiled, linked, and located your job, you can still respond to this parameter. This approach requires that you know where LOC86 has and will continued to locate your root job. Because the root job is the last system module to be located, the memory map for the root job provides the highest address of any system module. You can find the memory map for the root job in the file ROT.MP2 (after you have invoked the ICU SUBMIT file). The *Guide to the iRMX® I Interactive Configuration Utility* describes how to read memory maps, as well as a sample configuration.

(DSB) Data Segment Base [0-0FFFFH]

Use this parameter to specify the base value of the initialization task's data segment. Before you answer this prompt, either link and locate the I/O job or make a best guess. It is recommended that you link and locate the job. Refer to the *Guide to the iRMX® I Interactive Configuration Utility* for more information about linking and locating an application job.

The value you specify for this parameter is directly related to the size control that you specify when using the PL/M-86 compiler. The size control can be either LARGE, MEDIUM, COMPACT, or SMALL. If you choose PL/M-86 LARGE model procedures, you must set the data segment base parameter to zero. (A value of zero indicates that the task itself assigns the data segment.)

If you choose either PL/M-86 MEDIUM or COMPACT model procedures, you can obtain the base address of the data segment from the locate map. You **cannot** use the PL/M-86 SMALL model procedures for an initialization task. This model is only for Universal Development Interface (UDI) level applications (the iRMX I Operating System does not support applications compiled in SMALL).

It is recommended that you use the same PL/M-86 size control for all of your PL/M-86 jobs, and that any assembly language modules be compatible with this control.

The locate map produced by LOC86 lists both the base and offset addresses for the job. You must specify the base address of that module's DGROUP as the data segment for the I/O job. Obtain the base address from the locate map produced by LOC86. (DGROUP includes the data, stack, and memory segments/classes for the MEDIUM model and the data segments/class for the COMPACT model. The constant segment/class is included in CGROUP if the ROM compiler control is used.)

(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. If you choose PL/M-86 MEDIUM model procedures, you can obtain the base address of the stack segment from the locate map produced by LOC86.

(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-86 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 LOC86 command. Refer to the *Guide to the iRMX® I Interactive Configuration Utility* for more information on the LOC86 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 offsprung 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 the Hardware-Related Parameters chapter for more information about numeric processor extensions.



## 8.1 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 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 a later chapter.

## 8.2 BIOS SCREEN

The following screen shows the values from the 28612.DEF definition file. Note that values shown on the screens are sample values only. The exact values on each screen change according to the definition file used and your own changes.

```

(BIOS)      BIOS

(ASC) All Sys Calls in BIOS [Yes/No]          REQ
(ADP) Attach Device Task Priority [1-0FFH]     081H
(TF)  Timing Facilities Required [Yes/No]      REQ
(TTP) Timer Task Priority [0-0FFH]             081H
(CON) Connection Job Delete Priority [0-0FFH]   082H
(ACE) Ability to Create Existing Files [Yes/No] REQ
(SMI) System Manager ID [Yes/No]              REQ
(CUT) Common Update Timeout [0-0FFFFH]        03E8H
(CST) Control-Sequence Translation [Yes/No]    YES
(OSC) Terminal OSC Controls [Yes/No]          REQ
(TS)  Tape Support [Yes/No]                   YES
(PMI) BIOS Pool Minimum [0-0FFFFH]            01000H
(PMA) BIOS Pool Maximum [0-0FFFFH]            01000H

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

## BASIC I/O SYSTEM PARAMETERS

(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 does not include any subsystems other than the Extended I/O System, you can change the "Yes" value for this parameter to a "No". Making this change allows you to specify the Basic I/O System system calls you want to include in your application system.

If you specify "No" to this parameter line, the ICU allows you to make changes on four additional screens that are system-call related. These screens are discussed later in this chapter. Note, however, that the ICU always includes the following Basic I/O System system calls: A\$PHYSICAL\$ATTACH\$DEVICE and CREATE\$USER.

(ADP) Attach Device Task Priority [1-0FFH]	081H
--	------

This parameter line lets you specify the priority of the attach-device task. The values allowed for this parameter range from 1 to 0FFH. The default value of 81H is equivalent to 129 decimal. The default value is the highest priority allowed for a non-interrupt task. If you change the priority, Intel recommends 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 82H (130 decimal).

(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 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 should also exclude the GET\$TIME, SET\$TIME, GET\$GLOBAL\$TIME, and SET\$GLOBAL\$TIME system calls from the Basic I/O System.

(TTP) Timer Task Priority [0-0FFH] 081H

If you specified that your system includes timing facilities, you must specify the priority of the timer task. The default value is in decimal.

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 0001H. Although the ICU allows values from 0H to 0FFH, 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 [0-0FFH] 082H

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

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<sup>®</sup> Basic I/O System Calls Reference Manual* for more information about these options.)

## BASIC I/O SYSTEM PARAMETERS

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 take the following steps during configuration:

- Specify "Yes" to the "Ability to Create Existing Files" parameter.
- Include the A\$TRUNCATE system call in your system.
- Include the A\$DELETE\$FILE system call in your system.

If you intend on 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
----------------------------------	-----

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 [0-0FFFFH]	03E8H
--	-------

If you specified "Yes" to the "(TF) Timing Facilities Required" parameter, you must specify the time interval, in tens of milliseconds, that the Basic I/O System waits before updating devices with data buffered in memory. The default value of 03E8H (1000 decimal) represents 10 seconds. You must use a value that is a multiple of whole seconds; that is, 100 (1 second), 200 (2 seconds), 300 (3 seconds), and so on. A value of 0FFFFH 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 be shut down 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 I 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.

## BASIC I/O SYSTEM PARAMETERS

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 a later chapter for additional information on how to respond to terminal driver unit information screens. Refer to the *iRMX® 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
----------------------------	-----

Leave this parameter at its default setting in order to configure tape support in your system.

(PMI) BIOS Pool Minimum [0-0FFFFH]	01000H
(PMA) BIOS Pool Maximum [0-0FFFFH]	01000H

This parameter line lets 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 01000H is equivalent to 4096 decimal bytes.

The "BIOS Pool Minimum" and "BIOS Pool Maximum" parameters define the size of your Basic I/O System memory pool. Intel recommends that you set these prompts to the same value, to prevent the Basic I/O System from attempting to borrow memory later. This approach is particularly important if your application system includes the Human Interface and you set the Human Interface maximum memory pool size to 0FFFFH, 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 terminal 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 534 terminal driver, add 260H paragraphs to your memory pool.
- If your system includes an iSBC 544 terminal driver, add 250H paragraphs to the memory pool for each driver.
- If your system includes an iSBC 188/48 terminal driver, add 600H paragraphs to the memory pool for each driver.
- If your system includes an 8274 or iSBX 354 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:

$$\text{paragraphs} = 49\text{H} + \text{round-up}(\text{number of units}/2) \times 10\text{CH}$$

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:

$$\text{paragraphs} = 64\text{H} + (\text{round-up}((1\text{CH} + \text{dev}\$\text{gran})/10\text{H}) \times (\text{num}\$\text{bufs}))$$

## BASIC I/O SYSTEM PARAMETERS

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.

### 8.3 BIOS SYSTEM CALLS 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.

#### 8.3.1 Non-File System Calls

The "Non-File System Calls" screen lets you select the BIOS system calls that are not required by any particular file type.

```
(NOFSC)      Non-File System Calls

(IU)  Inspect User          REQ (DU)  Delete User          YES
(SDU) Set Default User     REQ (GDU) Get Default User   REQ
(SDP) Set Default Prefix   REQ (GDP) Get Default Prefix  REQ
(PDD) Physical Detach Device REQ (ST)  Set Time           YES
(GT)  Get Time             YES (WIO) Wait I/O          REQ
(ENG) Encrypt              YES (GGT) Get Global Time   REQ
(SGT) Set Global Time      REQ

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

Table 8.1 shows several configurations that are possible with this screen.

**Table 8-1. Non-File System Calls**

System Call	Configuration	
	EIOS	Application Loader
Inspect User	Required	Required
Delete User	Optional	Optional
Set Default User	Required	Required
Get Default User	Required	Required
Get Default Prefix	Required	Required
Set Default Prefix	Required	Required
Physical Detach Device	Required	Required
Set Time	Optional	Optional
Get Time	Optional	Optional
Set Global Time	Required	Required
Get Global Time	Required	Required
Wait I/O	Required	Required
Encrypt	Optional	Optional

### 8.3.2 Physical File System Calls

The "Physical File Sys Calls" screen lets you select the BIOS system calls that could be needed to support physical files.

```

Physical Files Sys Calls

(CF) Create File          YES (AF) Attach File      YES
(OP) Open                 YES (SE) Seek             YES
(RE) Read                 YES (WR) Write           YES
(SP) Special              YES (CL) Close          YES
(GCS) Get Con. Status    YES (GFS) Get File Status YES
(GPC) Get Path Component YES (DC) Delete Connection YES
(UPD) Update              YES

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

## BASIC I/O SYSTEM PARAMETERS

Table 8.2 shows several configurations that are possible with this screen.

**Table 8-2. Physical File System Calls**

System Call	Configuration	
	EIOS	Application Loader
Create File	Required	Required
Attach File	Required	Required
Open	Required	Required
Seek	Required	Required
Read	Required	Required
Write	Required	Required
Special	Required	Required
Close	Required	Required
Get Connection Status	Required	Required
Get File Status	Required	Required
Get Path Component	Optional	Optional
Delete Connection	Required	Required
Update	Optional	Optional

### 8.3.3 Stream File System Calls

The "Stream File Sys Calls" screen lets you select the BIOS system calls that could be needed to support stream files.

Stream File Sys Calls			
(CF) Create File	YES	(AF) Attach File	YES
(OP) Open	YES	(RE) Read	YES
(WR) Write	YES	(SP) Special	YES
(CL) Close	YES	(GCS) Get Con. Status	YES
(GFS) Get File Status	YES	(GPC) Get Path Component	YES
(DC) Delete Connection	YES	(DF) Delete File	YES
Enter Changes [Abbreviation ?/= new_value] :			

Table 8-3 shows several configurations that are possible with this screen.

Table 8-3. Stream File System Calls

System Call	Configuration	
	EIOS	Application Loader
Create File	Required	Required
Attach File	Required	Required
Open	Required	Required
Read	Required	Required
Write	Required	Required
Special	Required	Required
Close	Required	Required
Get Connection Status	Required	Required
Get File Status	Required	Required
Get Path Component	Optional	Optional
Delete Connection	Required	Required
Delete File	Required	Optional

### 8.3.4 Named File System Calls

The "Named File Sys Calls" screen lets you select the BIOS system calls that could be needed to support the named files.

```

Named File Sys Calls

(CF) Create File          YES (AF) Attach File      YES
(CD) Create Directory    YES (CA) Change Access    YES
(RF) Rename File        YES (OP) Open              YES
(SE) Seek                YES (RE) Read              YES
(WR) Write               YES (SP) Special           YES
(CL) Close               YES (GCS) Get Conn. Status YES
(GFS) Get File Status   YES (GDE) Get Directory Entry YES
(GPC) Get Path Component YES (DC) Delete Connection YES
(TR) Truncate           YES (DF) Delete File       YES
(GED) Get Ext. Data     YES (SED) Set Ext. Data   YES
(UPD) Update            YES

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

## BASIC I/O SYSTEM PARAMETERS

Table 8-4 shows several configurations that are possible with this system.

**Table 8-4. Named File System Calls**

System Call	Configuration	
	EIOS	Application Loader
Create File	Required	Required
Attach File	Required	Required
Create Directory	Required	Required
Change Access	Required	Required
Rename File	Required	Required
Open	Required	Required
Seek	Required	Required
Read	Required	Required
Write	Required	Required
Special	Required	Required
Close	Required	Required
Get Connection Status	Required	Required
Get File Status	Required	Required
Get Directory Entry	Optional	Optional
Get Path Component	Optional	Optional
Delete Connection	Required	Required
Truncate	Required	Required
Delete File	Required	Required
Get Extension Data	Optional	Optional
Set Extension Data	Optional	Optional
Update	Optional	Optional

## 9.1 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.

## 9.2 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 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 definition (.def extension) 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. Note that values shown on the screens are sample values only. The exact values on each screen change according to the definition file used and your own changes.

```

(IDEVS)          Intel Device Drivers

(S14) Mass Storage Controller Driver [Yes/No]  YES
(T74) 8274 Terminal Driver [Yes/No]          YES
(T51) 8251A Terminal Driver [Yes/No]         NO
(T30) 82530 Terminal Driver [Yes/No]         NO
(TCC) Terminal Comm Controller [Yes/No]      YES
(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] YES (S54) iSBC 254 [Yes/No] YES
(T34) iSBC 534 [Yes/No] NO (T44) iSBC 544A [Yes/No] YES
(RAM) RAM Disk Driver [Yes/No] NO (SCS) SCSI Driver [Yes/No] NO
(S64) iSBC 264 [Yes/No] NO (THD) Term. Handler [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 device drivers.

## 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. This driver also supports the iSBC 221 board which is similar to the iSBC 214 controller and supports both the ST506 and ESDI interfaces. The Driver also supports 5.25-inch flexible disk drives including the 1.2 megabyte quad-density type. The iSBC 221 controller also supports 125 Megabyte tape drives.

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 not include this memory in the "Memory" screen. 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 *iRMX® 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. Note that the values shown on the screens are sample values only. The exact values on each screen change according to the definition file used and your own changes.

## 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 MSC 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 [400H-0FFFFH] 01180H

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 Level	Code	with Slave Attached Level	Code	Master Level	Code	with Slave Attached 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 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" screen. Refer to the *iSBC 215 Generic Winchester Disk Controller Hardware Reference Manual*, the *iSBC® 214 Multi-Peripheral Controller Hardware Reference Manual*, or the *iSBC® 221 Peripheral Controller User's Guide* for information about the wakeup port address. Refer to Memory Parameters chapter and the *Guide to the iRMX® I Interactive Configuration Utility* for additional information about planning your memory usage.

(IPA) I/O Processor Block Address [0-0FFFFH] 01180H

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 in the megabyte of memory. 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.

## MASS STORAGE CONTROLLER

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 and iSBC 221 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 iRMX® 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 the Extended I/O System Parameters chapter 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 *iRMX® 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_214WTA
(MR) Maximum Retries [0-0FFFFH]		0H
(CS) Cylinder Size [0-0FFFFH]		0H
(NC) Number of Cylinders [0-0FFFFH]		0H
(NFH) Number of Heads/Fixed Disk [0-0FFH]		0H
(NRH) Number of Heads/Removable Disk [0-0FFH]		0H
(NS) Number of Sectors/Track [0-0FFFFH]		0H
(NAC) Number of Alternate Cylinders [0-0FFH]		0H
(SSN) Starting Sector Number [0-0FFFFFFFH]		0H
(BTI) Bad Track Information [Yes/No]		NO
(HLT) Head Load Time [0-0FFH]		0H
(SR) Step Rate [0-0FFH]		0H

(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 *ASM86 Language Reference Manual* for rules regarding this name.

Table MSC-1 lists examples of names that are recommended for Winchester disk drives. Table MSC-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.

# MASS STORAGE CONTROLLER

**Table MSC-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 MSC-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
uinfo_221qf	30	80	2	15	50	4

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 *iRMX® 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*, the *iSBC® 221 Peripheral Controller User's Guide*, 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.

## MASS STORAGE CONTROLLER

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]	0H
-------------------------------------	----

**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 MSC-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 MSC-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]	0H
---	----

**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]

0H

**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.

## MASS STORAGE CONTROLLER

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

(SSN) Starting Sector Number [0-0FFFFFFFH]	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. 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® 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 the Extended I/O Parameters chapter in this manual for information about automatic boot device recognition. Refer to the *iRMX® 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]           WTA0
(PFD) Physical File Driver Required [Yes/No]   YES
(NFD) Named File Driver Required [Yes/No]     NO
(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]                 07E00H
(DSZ) Device Size [0-0FFFFFFFH]              03C00000H
(UN) Unit Number on this Device [0-0FFH]     0CH
(UIN) Unit Info Name [1-16 Chars]            UINFO_214WTA
(RUT) Request Update Timeout [0-0FFFFH]     064H
(NB) Number of Buffers [nonrandom = 0/rand = 1-0FFFFH] 08H
(CUP) Common Update [Yes/No]                 NO
(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]           WTA0
```

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 the Extended I/O Parameters chapter).

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 *iRMX® 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.

## MASS STORAGE CONTROLLER

The ICU allows you to enter as many as 14 characters for the name. Refer to the *ASM86 Language Reference Manual* for rules regarding this name. Tables MSC-3 and MSC-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 MSC-3. iSBC® 214/215G DUIB Information for Winchester Drives**

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

DUIB Information						
Device-Unit Name(s)	Density	Sides	Gran	Device Size	Unit-Info Name	Num Buffers
WFO	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
WMFO	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]	NO

**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.

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 MSC-4 lists some of the values possible. Tables MSC-5 and MSC-6 list additional values.

Table MSC-5. 5.25-Inch Disk Characteristics

Gran	Density	Sectors per Track	<u>Device Size</u>			
			<u>One Sided</u> 40 Tracks	80 Tracks	<u>Two Sided</u> 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 MSC-6. 8-Inch Disk Characteristics

Gran	Density	Sectors per Track	<u>Device Size</u>	
			<u>One Sided</u> 77 Tracks	<u>Two Sided</u> 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 MSC-6 and MSC-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]

07E04H

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 *iRMX® 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 MSC-5 and MSC-6 show how the granularity relates to other parameters.

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

## MASS STORAGE CONTROLLER

(DSZ) Device Size [0-0FFFFFFFH]

03C00000H

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 MSC-3 shows the device sizes of several Winchester disks. Tables MSC-5 and MSC-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:

$$\text{DSZ} = \text{Granularity} * (\text{sectors per track}) * (\# \text{ heads}) \\ * (\# \text{ cylinders} - \# \text{ alternate cylinders})$$

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

### Uniform flexible diskettes

$$\text{DSZ} = \text{Granularity} * (\text{sectors per track}) * (2 \text{ heads}) * (\# \text{ cylinders})$$

### Standard flexible diskettes

$$\text{Size first track} = (128 \text{ bytes/sector}) * (16 \text{ sectors}) \\ \text{Size other tracks} = \text{Granularity} * (\text{sectors per track}) * (\# \text{ tracks})$$

$$\text{DSZ} = (\text{size first track}) + (\text{size other tracks})$$

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

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

0CH

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:

<u>Unit Number</u>	<u>Drive Type</u>
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_214WTA
-----------------------------------	--------------

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]	0FFFFH
---	--------

**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 Basic I/O Parameters chapter. 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".

## MASS STORAGE CONTROLLER

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

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

**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 MSC-3 shows how this value relates to other parameters for Winchester drives. Table MSC-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 Basic I/O Parameters chapter.

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]	NO
------------------------------	----

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 the Basic I/O Parameters chapter 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 the Generate File Names Parameters chapter for additional information on generating 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 *iRMX® 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. Note that the values shown on the screens are sample values only. The exact values on each screen change according to the definition file used and your own changes.

## 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 [1,2,3]                      (ATT) 1       (BTT) 1
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. [0-0FFFFFFFH]              (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. [0-0FFFFFFFH]              (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 [1,2,3] (ATT) 1 (BTT) 1

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. [0-0FFFFFFFH]	(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. [0-0FFFFFFFH]	(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]            UINFO_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 *ASM86 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 *iRMX® 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 *iRMX® 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 *iRMX® 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 *iRMX® 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 *iRMX® 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 *iRMX® 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 *iRMX® 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 *iRMX® 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 *iRMX® 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 *iRMX® 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 *iRMX® 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 *iRMX® 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 *iRMX® 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] T1
(UN) Unit Number on this Device [0-0FFH] 0H
(UIN) Unit Info Name [1-16 Chars] UINFO_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]	T1
-------------------------------------	----

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 *ASM86 Language Reference* 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 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 80286/386™-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 *iRMX® 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 28612.DEF file. Note that the values shown on the screens are sample values only. The exact values on each screen change according to the definition file used and your own changes.

## 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 [0-0FFFFFFFFFH]       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 [0-0FFFFFFFFFH]      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] 074H

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 [0-0FFFFH]	082H
------------------------------------	------

This parameter specifies the address of the 8251A control/status port. Choose the address that matches your board.

(IRP) 8254 Inrate Port [0-0FFFFH]	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.

(IGP) 8254 Input Control Port [0-0FFFFH]	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 [0-0FFFFFFFH]	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 [0-0FFFFH]	0H
------------------------------------	----

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 [0-0FFFFH]	0H
---	----

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 [0-0FFFFFFFH] 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 *ASM86 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 *iRMX® 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 *iRMX® 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 *iRMX® 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 *iRMX® Device Drivers User's Guide*.

(OGC) 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 *iRMX® 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 *iRMX® 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 *iRMX® 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 *iRMX® 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 *iRMX® 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 *iRMX® 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 *iRMX® 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 *iRMX® 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 *iRMX® 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 *iRMX® 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 *ASM86 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] OH

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 *iRMX® Device Drivers User's Guide* for further information about these tables.

The 82530 terminal driver is not part of a Start-up system supplied by Intel. The default values displayed in the following screens are the ICU defaults. Note that the values shown on the screens are sample values only. The exact values on each screen change according to the definition file used and your own changes.

## 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

                                Channel A          Channel B
Data Port [0-0FFFFH]           (D1A) 0D3H         (D1B) 0D1H
Status Port [0-0FFFFH]         (S1A) 0D2H         (S1B) 0D0H
Inrate Freq. [0-0FFFFFFFH]     (I1A) 04B0000H   (I1B) 04B0000H

Data Port [0-0FFFFH]           (D2A) 0D7H         (D2B) 0D5H
Status Port [0-0FFFFH]         (S2A) 0D6H         (S2B) 0D4H
Inrate Freq. [0-0FFFFFFFH]     (I2A) 04B0000H   (I2B) 04B0000H

Data Port [0-0FFFFH]           (D3A) 0D9H         (D3B) 0DBH
Status Port [0-0FFFFH]         (S3A) 0D8H         (S3B) 0DAH
Inrate Freq. [0-0FFFFFFFH]     (I3A) 04B0000H   (I3B) 04B0000H

Data Port [0-0FFFFH]           (D4A) 0DDH         (D4B) 0DFH
Status Port [0-0FFFFH]         (S4A) 0DCH         (S4B) 0DEH
Inrate Freq. [0-0FFFFFFFH]     (I4A) 04B0000H   (I4B) 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] (D1A) 0D3H

This parameter specifies the address of channel A of the 82530 data port.

(CAS) Channel A Status Port [0-0FFFFH] (S1A) 0D2H

This parameter specifies the address of channel A of the 82530 control/status port.

(AIF) Channel A Inrate Freq. [0-0FFFFFFFH] 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] (D1B) 0D1H

This parameter specifies the address of channel B of the 82530 data port.

(CBS) Channel B Status Port [0-0FFFFH] (S1B) 0D0H

This parameter specifies the address of channel B of the 82530 control/status port.

(BIF) Channel B Inrate Freq. [0-0FFFFFFFH] 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 *ASM86 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 *iRMX® Device Drivers User's Guide*.

(EGH) 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 *iRMX® 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 *iRMX® 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 *iRMX® Device Drivers User's Guide*.

(OGC) 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 *iRMX® 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 *iRMX® 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 *iRMX® 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 *iRMX® Device Drivers User's Guide*.

(TRM) Terminal Type [GRT/HardCopy]	GRT
------------------------------------	-----

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 *iRMX® 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 *iRMX® 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 *iRMX® 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 *iRMX® 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 *iRMX® Device Drivers User's Guide*.

## 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-OFFH]	OH
--	----

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

(MB) Max Buffers [0-OFFH]	OH
---------------------------	----

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 *ASM86 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.

## 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, iSBC 548, or iSBC 549 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 *iRMX® 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

## 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]          546
(IL)  Interrupt Level [Encoded Level]    038H
(MA)  Memory Address Base [0-0FFFFFFH]  0E0000H
(PA)  Port Address [0-0FFFFFFH]         08A6H

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

```
(DEV) Device Name [1-16 Chars]          546
```

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]	0E0000H
---------------------------------------	---------

The Memory Address Base is the lowest address in dual port RAM that is visible to other boards on the MULTIBUS. This parameter line lets you specify the base address in 16-byte paragraphs of the dual port RAM that matches the jumper configuration for the appropriate Terminal Communications Controller Board.

When you are configuring the jumpers on the board, be aware that the firmware on the communications controller board needs at least 48 K-bytes of MULTIBUS dual port RAM.

The user can compute the lowest visible dual port RAM on the board by using the following formula:

$$\text{Address} = (\text{256K boundary start address}) + (\text{number of the block selected} * 64\text{K}) + 16\text{K}$$

(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 546/547/548/549 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 I 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

### 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]           546
(NAM) Unit Info Name [1-16 Chars]       UINFO_546
(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 *ASM86 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 *iRMX® Device Drivers User's Guide*.

## TERMINAL COMMUNICATIONS CONTROLLER

(EGH) 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 *iRMX® 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 *iRMX® 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 *iRMX® 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 *iRMX® 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 *iRMX® 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 *iRMX® 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 *iRMX® Device Drivers User's Guide*.

## TERMINAL COMMUNICATIONS CONTROLLER

(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 *iRMX® Device Drivers User's Guide*.

(MG) 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 *iRMX® 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 *iRMX® 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 *iRMX® Device Drivers User's Guide*.

## TERMINAL COMMUNICATIONS CONTROLLER

(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 *iRMX® 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]	546
(NAM) Device-Unit Name [1-14 Chars]	T546_0
(UN) Unit Number on this Device [0-OFFH]	0H
(UIN) Unit Info Name [1-16 Chars]	UINFO_546
(MB) Max Buffers [0-OFFH]	0H
Enter [ Abbreviation = new_value / Abbreviation ? / H ]	
:	

(DEV) Device Name [1-16 Chars]	546
--------------------------------	-----

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]	T546_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 *ASM86 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_546
-----------------------------------	-----------

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.

## TERMINAL COMMUNICATIONS CONTROLLER

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

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(A) Line Printer Driver

The line printer driver for the iSBC 286/10(A), iSBC 286/12, and iSBC 386/12 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 *iRMX® 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. Note that the values shown on the screens are sample values only. The exact values on each screen change according to the definition file used and your own changes.

## 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]	NO
(ITO)	Interrupt Time Out [0-0FFFFH]	0200H

(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), iSBC 286/12, or iSBC 386/12 board. These addresses must match the jumper setting on the board. Intel recommends that you do not change the factory-configured jumper settings.

(TAB) Printer Expands Tabs [Yes/No] NO

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 *ASM86 Language Reference Manual* for rules regarding this name.

(MB) Max Buffers [0-0FFH]

OH

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 *iRMX® 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. Note that the values shown on the screens are sample values only. The exact values on each screen change according to the definition file used and your own changes.

## 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-OFFH]                                OH

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 *ASM86 Language Reference Manual* for rules regarding this name.

```
(MB) Max Buffers [0-OFFH]                                OH
```

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® 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 *iRMX® 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. Note that the values shown on the screens are sample values only. The exact values on each screen change according to the definition file used and your own changes.

## 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]  01180H
(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 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" screen. Refer to the iSBC 220 SMD Disk Controller Hardware Reference Manual for information about the wakeup address. Refer to Memory Parameters chapter in this book and the *iRMX® I Interactive Configuration Utility Reference* manual for additional information about planning your memory usage.

(IPA) I/O Processor Block Address [0-0FFFFFFH]	01180H
--	--------

This parameter defines the starting address of memory required by the I/O Processor Blocks and I/O buffers. This memory can be anywhere in the megabyte of supported RAM. Do not include this memory in the memory you define on the "Memory" screen. 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

$$1152 + (N * \text{device granularity}) * U$$

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 iRMX® 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 an SMD disk. Refer to the Extended I/O System Parameters in this manual for information about automatic boot device recognition. Refer to the *iRMX® 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 [0-0FFFFFFFH] 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 *ASM86 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 *iRMX® 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 an 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 an 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 [0-0FFFFFFFH]	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 iRMX® 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 an SMD. Refer to the Extended I/O System Parameters chapter in this manual for information about automatic boot device recognition. Refer to the *iRMX® 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 [0-0FFFFFFFH]                       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]                             OFFH

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 the Extended I/O System Parameters chapter).

The ICU allows you to enter from one to fourteen characters. Refer to the *ASM86 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 *iRMX® Basic I/O System Calls Reference Manual* for more information about granularity.

(DSZ) Device Size [0-0FFFFFFFH]	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.

$$\text{DSZ} = [\text{total cylinders} - \text{alternate cylinders}] * \text{number of heads} \\ * \text{sectors per track} * \text{device granularity}$$

(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.

<u>Unit Numbers</u>	<u>on Drive</u>
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 the Basic I/O System Parameters chapter in this manual. 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]
-----------------------------------

08H
-----

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.

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 the Basic I/O System Parameters chapter 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 *iRMX® 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 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]              0FH
(HUT) Head Unload Time [0-0FFH]            0FOH

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 *ASM86 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 218A-1.

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 218A-1. 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 *iRMX® 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 218A-1 shows some of the typical values for this parameter. Table 218A-1 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 218A-1 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]	0FH
-------------------------------	-----

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 [0-0FFFFFFFH]                        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]                              OFFH

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 the Extended I/O System Parameters chapter).

The ICU allows you to enter from one to fourteen characters. Refer to the *ASM86 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 218A-2.

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 218A-2 shows how your response to this prompt is related to other parameters.

**Table 218A-2. 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
PMFO	5.25	Double	Double	512	325632	uinfo 218mf	8
PMFDY0	5.25	Double	Double	512	653312	uinfo_218mfdy	8

(SDS) Single or Double Sided Disks [Single/Double]	DOUBLE
--	--------

This parameter specifies the number of recording surfaces for this flexible disk. Table 218A-2 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.

(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 218A-2 shows how this value relates to other parameters. Refer to the *iRMX® Basic I/O System Calls Reference Manual* for more information about granularity.

(DSZ) Device Size [0-0FFFFFFFH]	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:

$$\text{DSZ} = \text{total cylinders} * \text{number of heads} * \text{sectors per track} * \text{device granularity}$$

(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 the Basic I/O Parameters chapter. 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 218A-2 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 the Basic I/O Parameters chapter.

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 the Basic I/O Parameters chapter 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 *iRMX® Device Drivers User's Guide* for more information about these tables.

The default values you see displayed in the following sections are the default values supplied by the ICU when the iSBC 208 is selected. Note that the values shown on the screens are sample values only. The exact values on each screen change according to the definition file used and your own changes.

## 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]          208
(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]      0H
(BBA) Boundary Buffer Address [0-0FFFFFFFH] 01600H

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

```
(DEV) Device Name [1-16 Chars]          208
```

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]	0H
------------------------------------	----

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 [0-0FFFFFFFH] 01600H
--

This parameter specifies the address of a 1K buffer that is in the megabyte of supported 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" 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<sup>®</sup> 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]           208
(NAM) Unit Info Name [1-16 Chars]       UINFO_208F
(MR) Maximum Retries [0-0FFFFH]        09H
(CS) Cylinder Size [0-0FFFFH]          01AH
(NT) Number of Tracks per Side [0-0FFFFH] 04DH
(NS) Number of Sectors/Track [0-0FFFFH] 01AH
(SR) Step Rate [0-0FFH]                 08H
(HLT) Head Load Time [0-0FFH]          028H
(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]	208
--------------------------------	-----

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_208
-----------------------------------	-----------

This parameter specifies a unique unit information name for this device. The first character must be an alphabetic character. Refer to the *ASM86 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 208-1.

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 208-1. iSBC® 208 Controller Unit Information**

Unit-Info Name	Cylinder Size	D U I B Information			Step Rate	Head Load Time
		No. Tracks per Side	No. Sectors per Track			
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 *iRMX® 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 208-2 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 208-1 and 208-2 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 208-2. Flexible Disk Unit Information

Sector Size	Density	Sectors per Track		Density	Sectors per Track	
		8-inch	5.25-inch		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<sup>®</sup> 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]                208
(NAM) Device-Unit Name [1-14 Chars]           AFO
(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] STANDARD
(GRA) Granularity [0-0FFFFH]                 080H
(DSZ) Device Size [0-0FFFFFFFH]              03E900H
(UN)  Unit Number on this Device [0-0FFH]     0H
(UIN) Unit Info Name [1-16 Chars]            UINFO_208F
(RUT) Request Update Timeout [0-0FFFFH]      096H
(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]                208
```

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]           AFO
```

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 the Extended I/O System Parameters chapter).

The ICU allows you to enter from one to fourteen characters. Refer to the *ASM86 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 208-3.

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]	SINGLE
--	--------

This parameter specifies the recording density of this flexible disk. The recording density is either single or double. Table 208-3 shows how your response to this prompt is related to other parameters.

**Table 208-3. iSBC® 208 Controller DUIB Information**

Device-Unit Name(s)	Size	Density	DUIB Information			Device Size	Unit-Info Name	Num Buffers
			Sides	Gran.				
AF0	8	Single	Single	128	256256	uinfo_208f	6	
AFD0	8	Double	Single	256	509184	uinfo_208f	6	
AFDD0	8	Double	Double	256	1021696	uinfo_208fd	6	
AFDX0	8	Double	Double	1024	1255168	uinfo_208fdx	3	
AMF0	5.25	Double	Double	512	325632	uinfo_208mf	6	
AMFDY0	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 208-3 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]	080H
------------------------------	------

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 80H is equivalent to 128 decimal. Table 208-3 shows how this value relates to other parameters. Refer to the *iRMX® Basic I/O System User's Guide* for more information about granularity.

(DSZ) Device Size [0-0FFFFFFFH]	03E900H
---------------------------------	---------

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:

$$\text{DSZ} = \text{number of cylinders} * \text{number of heads} * \text{sectors per track} * \text{device granularity}$$

Table 208-3 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]	UINFO_208F
-----------------------------------	------------

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]	096H
---	------

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 96H is equivalent to 150 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 the Basic I/O System Parameters chapter. 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 208-3 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 the Basic I/O System Parameters chapter.

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 the Basic I/O System Parameters chapter 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.

## iSBC® 254 Driver Parameters

The iSBC 254 driver supports both the iSBC 254 and iSBC 254S bubble memory controllers. Three screens define the interface between the iSBC 254 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 *iRMX® Device Drivers User's Guide* for further information about these tables.

## iSBC® 254 Driver Screen

The ICU uses the information from the following screen to create a device information table for the iSBC 254 driver. If your system includes more than one iSBC 254 controller, you can set your system up in three ways:

- You can specify a unique interrupt level for each controller.
- You can specify the total number of iSBC 254 controllers on one information screen.
- You can specify multiple units on the same interrupt level.

```
(D254)          iSBC 254 Driver
(DEV) Device Name [1-16 Chars]           254
(IL)  Interrupt Level [Encoded Level]     038H
(ITP) Interrupt Task Priority [0-255]     130
Enter Changes [ Abbreviation ?/- new_value ]
:
```

```
(DEV) Device Name [1-16 Chars]           254
```

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
--------------------------------------	------

Use this parameter to specify the encoded interrupt level for the iSBC 254 driver. If your system includes more than one board and each board is considered a separate unit, then specify a unique interrupt level for each board. 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 [0-0FFH]	082H
--	------

Use this parameter to specify the initial priority of the device's interrupt task. The default value 082H is equivalent to 130 decimal. The actual priority of the iRMX I interrupt task changes because the Nucleus adjusts an interrupt task's priority according to the interrupt level that it services.

## Query Screen

After you have completed the "iSBC 254 Driver" screen, the query screen is displayed. It contains only one line:

Do you have any units for this device?
--

Respond to this prompt with a "Yes" if you have not already defined a unit information table for this driver. Respond to this prompt with a "No" if the following statements are true:

- You are defining more than one device information table for this driver (that is, you have more than one controller for this driver).
- You already have a unit information table that describes the additional controller.

## iSBC® 254 Unit Information Screen

The ICU uses the information from the following screen to create a unit information table for the iSBC 254 driver.

```

(U254)  iSBC 254 Unit Information

(DEV) Device Name [1-16 Chars]           254
(NAM) Unit Info Name [1-16 Chars]       UINFO_254
(MR) Maximum Retries [0-0FFFFH]        09H
(NB) Number of Boards [0-0FFFFH]        1
(BPA) Base Port Address [0-0FFFFH]      0880H
(NPB) Number of Pages per Board [0-0FFFFH] 0800H

Enter Changes [ Abbreviation ?/= new_value]
;

```

```

(DEV) Device Name [1-16 Chars]           254

```

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]       UINFO_254

```

Use this parameter to specify a unique name for this unit information table. The first character must be an alphabetic character. Refer to the *ASM86 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
---------------------------------	-----

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 *iRMX® Basic I/O System Calls Reference Manual* for related information on the E\$IO\$SOFT exception code. Refer to the *iSBC 254 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]	1
----------------------------------	---

Use this parameter to specify the number of boards that are in your system connected to the same interrupt line. Multiple boards on the same interrupt level must have contiguous base addresses, 20H apart.

If your system has more than one iSBC 254 board, you can configure each board as a separate device (each with its own interrupt level and own DUIB) or you can configure as many as four boards as a single unit. If you configure several boards as a single unit, all the boards must be connected to the same interrupt line, and each board should have the same number of bubble devices (7110 components).

(BPA) Base Port Address of Board [0-0FFFFH]	0880H
---	-------

Use this parameter to specify the base address of the I/O port which matches the jumper configuration of the iSBC 254 board. If you have more than one iSBC 254 board connected to the same interrupt line, you must provide the driver with the port address of only the first iSBC 254 board. The port address of additional iSBC 254 boards must be 20H greater than the previous board.

(NPB) Number of Pages per Board [0-0FFFFH]	0800H
--	-------

Use this parameter to specify the number of pages on each iSBC 254 board. Intel recommends that you do not change the default value of 0800H.

You must use the default value of 0800H if your board has just one bubble device (7110 component). If your board has two bubble devices, it is recommended that you keep the number of pages per board constant and increase the device granularity. The relationships between various responses to iSBC 254 parameter lines are shown in Table 9-13.

Table iSBC® 254-1. iSBC® 254 Controller Information

Boards/Unit	Bubble Devices/Board	Pages/Board	Granularity	Device Size
1	1	800H	40H (64 dec)	20000H
1	2	800H	80H (128 dec)	40000H
1	4	800H	100H (256 dec)	80000H
2	1	800H	40H	40000H
2	2	800H	80H	80000H
2	4	800H	100H	100000H
3	1	800H	40H	60000H
3	2	800H	80H	100000H
3	4	800H	100H	180000H
4	1	800H	40H	80000H
4	2	800H	80H	100000H
4	4	800H	100H	200000H

If your bubble device consists of more than one board defined as a single unit, then each board must have the same granularity. If each board does not have the same granularity, then define each board as a separate unit.

## Query Screen

After you have completed the "iSBC 264 Unit Information" screen, the query screen is displayed. It contains only one line:

Do you have any more units for this device??

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

If you define an additional unit on the same interrupt level with the same unit number, use the same base port address for each unit. Having units with different base port addresses requires you to format the second unit separate from the first unit. This action would effectively delete any previous information stored on the board with the larger base port address.

## iSBC<sup>®</sup> 254 Device-Unit Information Screen

The ICU uses the information from the following screen to create a device-unit information block (DUIB) for the iSBC<sup>®</sup> 254 driver.

```
(I254) iSBC 254 Device-Unit Information

(DEV) Device Name [1-16 Chars]                254
(NAM) Device-Unit Name [1-12 chars]           B0
(PFD) Physical File Driver Required [Yes/No]   YES
(NFD) Named File Driver Required [Yes/No]     YES
(GRA) Granularity [0-0FFFFH]                 0100H
(DSZ) Device Size [0-0FFFFFFFH]              0080000H
(UN) Unit Number on this Device [0-0FFH]      0H
(UIN) Unit Info Name [1-16 Chars]            UINFO_254
(RUT) Request Unit Timeout [0-0FFFFH]        0064H
(NB) No. of Buffers [nonrandom = 0/rand = 1/0FFFFH] 0008H
(CUP) Common Update [Yes/No]                 YES
(MB) Max Buffers [0-0FFH]                    00FFH

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

```
(NAM) Device-Unit Name [1-12 chars]           B0
```

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 in the Logical Names screen (see the Extended I/O System Parameters chapter). 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 *iRMX<sup>®</sup> Bootstrap Loader Reference Manual* for detailed information). The Intel-supplied Bootstrap Loader file and/or PROMs use the name "B0" for this device.

The ICU allows you to enter from one to twelve characters. Refer to the *ASM86 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 ATTACH\$DEVICE 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

The 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 ATTACH\$DEVICE 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
------------------------------	-------

Use this parameter to specify the minimum number of bytes that the device reads or writes in one operation. This value is determined by the number of bubble devices (7110 components) on each board. The default value 0100H is equivalent to 256 decimal (4 bubble devices per board). Each board in a unit must have the same granularity. Refer to the *iRMX® Basic I/O System User's Guide* for more information about granularity.

(DSZ) Device Size (0- 0FFFFFFFH]	0080000H
----------------------------------	----------

Use this parameter to specify the device storage capacity in bytes. The default value 0080000H is equivalent to 524288 decimal. The device size varies with the granularity and the number of boards per unit (see Table 9-13).

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

Use this parameter to specify the unit number for this device-unit. If your system includes two boards with different granularities but sharing the same interrupt level, then configure each board as a separate unit.

(UIN) Unit Info Name [1-16 chars]	UINFO_254
-----------------------------------	-----------

Use this parameter to specify the 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" before you run the second stage of the ICU.
- Each DUIB can only point 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
------

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 0064H is equivalent to 100 decimal.

The values 9 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 the Basic I/O System Parameters chapter. 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 until the shorter of the two intervals and updates the attached files on the device. Thus, if you use the common update features 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] 08H
--

Use this parameter to specify 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 256 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 to 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 response to the "(GRA) Granularity" and the "(NB) Number of Buffers" parameter lines is in the Basic I/O Parameters chapter.

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. First, 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
------------------------------	-----

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 the Basic I/O System Parameters chapter 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 (00FFH) 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.

Do you have any more DUIBs for this device?
---

Respond "Yes" to this prompt "Do you have any more device-unit information blocks for this device?" if you plan to use the iSBC 254 controller with two device units 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 description file before you generate your configuration files.



## Terminal Handler Driver Device-Unit Information

The Terminal Handler driver supports any terminal handler in your system, including the Dynamic Debugger's terminal handler. This driver is included primarily for compatibility with previous releases of the iRMX Operating System. You should use this driver if your system includes only the Nucleus and the Basic I/O System and you wish to invoke BIOS system calls to a terminal handler or the dynamic debugger's terminal handler.

```
(DTHD)      Terminal Handler 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.

## Terminal Handler

```
(ITHD)      Terminal Handler Driver Device-Unit Information

(DEV) Device Name [1-16 Chars]
(NAM) Device-Unit Name [1-12 Chars]
(MB)  Max Buffers [0-OFFH]                OH

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) Device-Unit Name [1-12 Chars]
```

This parameter specifies a unique name for this unit information table. The first character must be an alphabetic character. Refer to the *ASM86 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.

```
(MB) Max Buffers [0-OFFH]                OH
```

Use this parameter to specify 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.

## 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 *iRMX® 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
(OGC) 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 *ASM86 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 *iRMX® 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 *iRMX® 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 *iRMX® 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 *iRMX® Device Drivers User's Guide*.

(OGC) 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 *iRMX® 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 *iRMX® 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 *iRMX® 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 *iRMX® Device Drivers User's Guide*.

(TRM) Terminal Type [GRT/HardCopy]	GRT
------------------------------------	-----

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 *iRMX® 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 *iRMX® 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 *iRMX® 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 *iRMX® 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 *iRMX® 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 *ASM86 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 the Generate File Names chapter 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 *iRMX® 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. Note that the values shown on the screens are sample values only. The exact values on each screen change according to the definition file used and your own changes.

## 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]  0E0000H
(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]

0E0000H

This parameter specifies the base address in 16-byte paragraphs of the dual port RAM that matches the jumper configuration of the iSBC 544A board. 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 (first divide by 16 to convert from bytes to paragraphs). Use the same approach on the remaining boards. For example, starting with a base address of 0E0000H and a dual port memory size of 400H paragraphs (your response to the "(MS) Dual Port Memory Size" parameter), the addresses for the second, third, and fourth boards should be 0E400H, 0E800H, and 0EC00H 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" screen includes addresses that represent iSBC 544A on-board memory. For example, if the base address of your iSBC 544A controller board's memory address is 0E00H, then you must specify 0DFFFH on the "Memory" 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_544
(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 *ASM86 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 *iRMX® 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 *iRMX® 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 *iRMX® 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 *iRMX® 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 *iRMX® 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 *iRMX® 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 *iRMX® 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 *iRMX® 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 *iRMX® 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 *iRMX® 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 *iRMX® 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 *iRMX® 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 *iRMX® 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_544
(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 *ASM86 Language Reference Manual* for rules regarding this name.

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

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]	OH
---------------------------	----

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 the Generate File Names Parameters chapter for additional information on generating configuration files).

## Ram Driver Parameters

The RAM driver allows you to use a portion of the 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 1M byte address space. The maximum allowable size of the RAM disk is 1M byte 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 command LOCDATA to contain Human Interface commands (see the *Operator's Guide To The iRMX® 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 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 command. LOCDATA creates an object module (which contains an image of the RAM disk) and adds the module to a boot-loadable file. This process allows the second 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 *iRMX® 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]          080000H
(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 *ASM86 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.

## RAM

(BMA) Base Memory Address [0-0FFFFFFH]	080000H
--	---------

This parameter specifies the physical address at which the RAM disk starts. This address may be anywhere within the 1M 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 - BFFFFH 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 LOCDATA command to create a bootloadable file containing a RAM disk image. Refer to the *Operator's Guide to the iRMX® Human Interface* for detailed steps of how to use the 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 [0-0FFFFFFFH]                      03FFFF0H
(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]                            OFFH

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.

## RAM

(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 the Extended I/O System Parameters chapter).

The ICU allows you to enter from one to fourteen characters. Refer to the *ASM86 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 *iRMX<sup>®</sup> Basic I/O System User's Guide* for more information about granularity.

(DSZ) Device Size [0-0FFFFFFFFH]	03FFF0H
----------------------------------	---------

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 [0-0FH]	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 the Basic I/O System Parameters chapter. 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 the chapter on the Basic I/O System.

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 the Chapter on the Basic I/O System 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 28612.DEF description file. Note that the values shown on the screens are sample values only. The exact values on each screen change according to the definition file used and your own changes.

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:

<u>Command</u>	<u>Opcode</u>
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.)

## SCSI

(DSCSI) SCSI Driver	
(DEV) Device Name [1-16 Chars]	
(IL) Interrupt Level [Encoded Level]	018H
(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 [1,2]	2
(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]

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] 018H

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 "018H" (0000 0000 0001 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 Level	Code	with Slave Attached Level	Code	Master Level	Code	with Slave Attached 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".

## SCSI

(DCT) DMA Controller Type [1,2]	2
---------------------------------	---

This parameter specifies the type of direct memory access (DMA) controller used in your system. A value of 1 means you are using an 80186; a value of 2 means you are using an 82258.

(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".

(DGR) 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]
(NAM) Unit Info Name [1-16 Chars]
(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]     0H
(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]
```

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]
```

Use this parameter to specify a unique name for this unit information table. The first character must be an alphabetic character. Refer to the *ASM86 Language Reference Manual* for rules regarding this name. Table SCSI-1 lists examples of recommended names for the SCSI driver.

## SCSI

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 SCSI-1. 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, 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
<p>Note: * The Xebec 1420 is now superseded by the Xebec 1421. The Xebec 1420/1421 controller must be ordered with the Intel option to get the correct firmware.</p>					

(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 *iRMX® 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]
--------------------------------------

0H
----

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]
(NAM) Device-Unit Name [1-12 Chars]
(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]                          0100H
(DSZ) Device Size [0-0FFFFFFFH]                       OECAC00H
(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]                             OFFH

Enter [ Abbreviation = newvalue / 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) 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]

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 *iRMX® 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 *ASM86 Language Reference Manual* for rules regarding this name. Table SCSI-2 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 SCSI-2. SCSI DUID Information**

Device-Unit Name	Granularity	Device Size	Device Type	Unit Info Name
S0	200H	00000400H	Generic	UINFO SCSI*
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

\*(generic SCSI/SASI)

(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.

## SCSI

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]	0100H
------------------------------	-------

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 "0100H" is equivalent to 256 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 *iRMX<sup>®</sup> Basic I/O System User's Guide* for more information about granularity.

(DSZ) Device Size [0-0FFFFFFFFH]	0ECAC00H
----------------------------------	----------

Use this parameter to specify the device storage capacity in bytes. Calculate this value as follows:

$$\text{DSZ} = [(\text{total cylinders}) - (\# \text{ alternate cylinders})] * (\# \text{ heads}) \\ * (\text{sectors per track}) * \text{device granularity}$$

The default value of "0ECACOOH" 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 iRMX<sup>®</sup> 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 SCSI-3 for examples). This byte can identify up to eight controllers (0-7) and eight devices per controller (also numbered 0-7).

**Table SCSI-3. 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]

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]	06H
--	-----

Use this parameter to specify 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 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 the Basic I/O Parameters chapter.

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 the Basic I/O Parameters chapter 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.

## SCSI

### 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® 264 Driver Parameters

The iSBC 264 driver supports the iSBC 264 magnetic bubble memory board. 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 *iRMX® 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). Note that the values shown on the screens are sample values only. The exact values on each screen change according to the definition file used and your own changes.

## 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 *ASM86 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 *iRMX® 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 [0-0FFFFFFFH]                     0200000H
(UN)  Unit Number on this Device [0-OFFH]           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-OFFH]                          OFFH

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 the Extended I/O Parameters chapter). 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 *iRMX® Bootstrap Loader Reference Manual* for detailed information).

The ICU allows you to enter from one to fourteen characters. Refer to the *ASM86 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:

<u>Bubbles in Parallel</u>	<u>Granularity</u>
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 *iRMX® Basic I/O System User's Guide* for more information about granularity.

(DSZ) Device Size [0-0FFFFFFFH]	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 the Basic I/O Parameters chapter. 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 Basic I/O Parameters chapter.

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.

(GUP) 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 the Basic I/O Parameters chapter for information about common update timeout (updating attached files at fixed intervals).

(MB) Max Buffers [0-OFFH]
---------------------------

OFFH
------

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.



## 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 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 *iRMX® 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.

## USER DEVICES

```
(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]                 OH
(NDU) Number of User Defined Device-Units [0-0FFH]           OH

          Terminal Device and Unit Information Names [1-16 Chars]
(NO1) NONE      (NO2) NONE      (NO3) NONE
(NO4) NONE      (NO5) NONE      (NO6) NONE
(NO7) NONE      (NO8) NONE      (NO9) NONE
(NO10) NONE     (NO11) NONE     (NO12) NONE
(NO13) NONE     (NO14) NONE     (NO15) NONE
(NO16) NONE     (NO17) NONE     (NO18) 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/M86 or ASM86 then LINK86 and LOC86 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 *iRMX® 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 *iRMX 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.

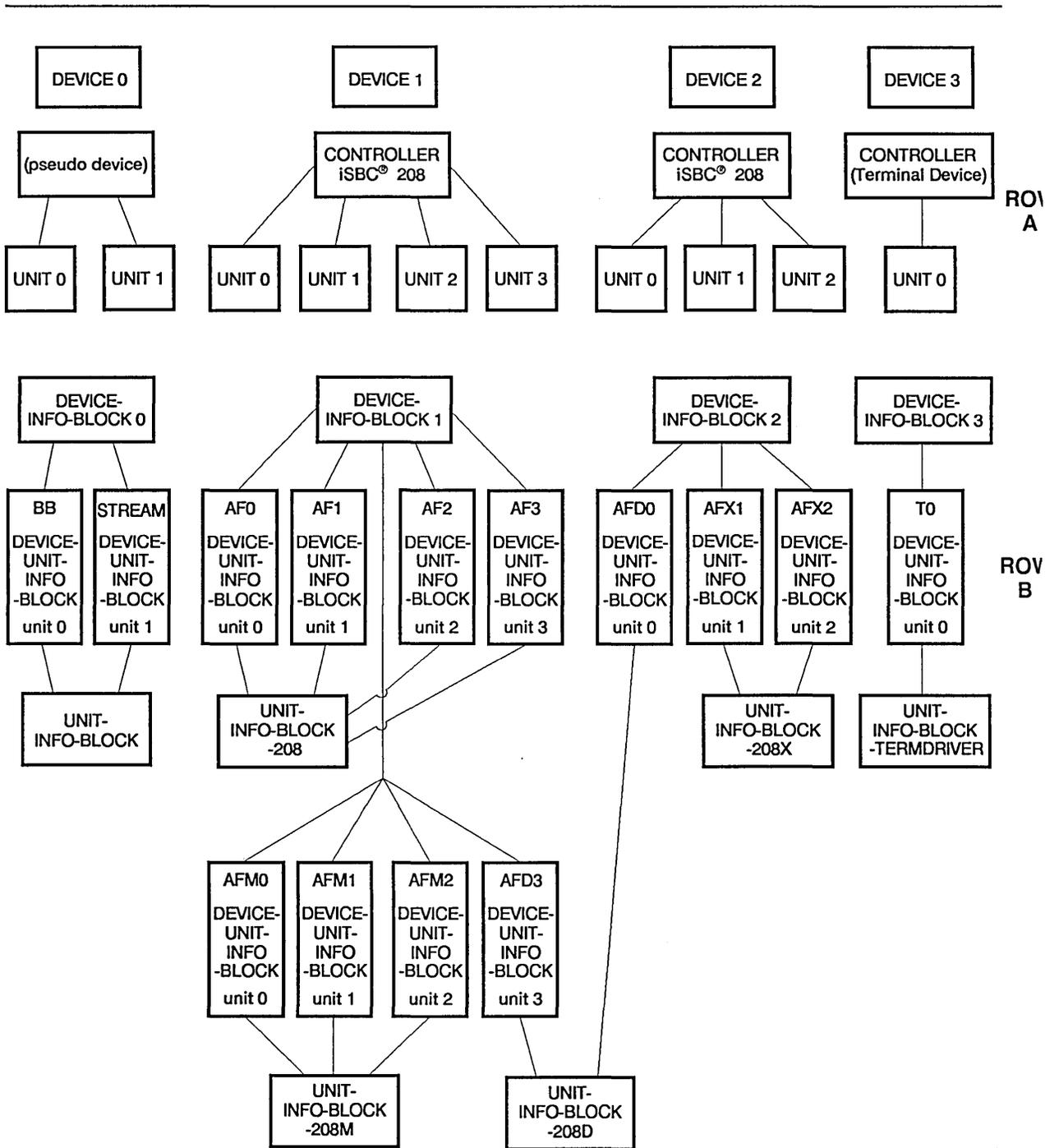
**(ND) Number of User Defined Devices [0-0FFH]**

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.

## USER DEVICES

(NDU) Number of User Defined Device-Units [0-0FFH]

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 USER-1 shows the device numbering of a sample system.



W-1052

Figure USER-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.

## SYSTEM DEBUGGER PARAMETERS

### 10.1 INTRODUCTION

The iRMX System Debugger (SDB) is one of the debugging aids supplied by the iRMX I 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.

### 10.2 THE SYSTEM DEBUGGER PARAMETER

The SDB screen is as follows:

```
(SDB)      System Debugger
(SLV) SDB Interrupt Level [Encoded Level/NONE - 0FFH] 018H
```

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® I Hardware and Software Installation Guide* provides further information.



# TERMINAL HANDLER AND DYNAMIC DEBUGGER PARAMETERS

# 11

## 11.1 INTRODUCTION

The Terminal Handler provides a real-time, asynchronous I/O interface between one terminal and tasks in an iRMX I application system. The Terminal Handler is configurable for both input and output or only for output.

When using the ICU, if you select the Terminal Handler on the Subsystems screen, you are presented with a Terminal Handler screen. After you edit that screen, the ICU asks you if you want another Terminal Handler. In this way, you can configure as many independent Terminal Handlers as you need for your terminals. Each terminal requires its own Terminal Handler, and each terminal/Terminal Handler pair requires its own 8251A USART for I/O and serial/parallel conversion.

There are two Terminal Handlers. One, which is described in the *iRMX® I Terminal Handler Reference Manual*, stands alone. The other is embedded in the Dynamic Debugger, and is essentially identical, for the points of view of programmers and terminal operators, to the stand-alone version. The only significant difference is that the Dynamic Debugger version includes an interface between the Terminal Handler and the Dynamic Debugger. One of the parameters on the Terminal Handler screen enables you to specify, for each Terminal Handler you configure, whether it is to be the one that is associated with the Dynamic Debugger. In every application system that includes the Dynamic Debugger, one of the Terminal Handlers that is configured into that application must be the Dynamic Debugger's Terminal Handler. If you fail to so identify one of the Terminal Handlers in your configuration, your system will not generate successfully.

Note that the Terminal Handler is not the same as a terminal driver. The BIOS has several terminal drivers: the 8251A USART terminal driver, the 8274 terminal driver, the 82530 Serial Communications Controller (SCC) terminal driver, the iSBC 534 driver, the iSBC 544 driver, the iSBX 270 driver, and the iSBC 188/48 driver. Each of these terminal drivers allows your system to perform I/O to a terminal via the BIOS procedural interface, that is, by A\$READ and A\$WRITE system calls. This is quite different than the mailbox interface that tasks use when interfacing via the Terminal Handler.

## 11.2 DYNAMIC DEBUGGER/TERMINAL HANDLER SCREEN

The ICU Terminal Driver screen refers to a special driver that was included in an earlier release of the iRMX I software in order to support the Human Interface. Its purpose was to allow tasks to use I/O system calls to communicate via the Terminal Handler. However, this driver is not compatible with the Human Interface, so we recommend that you use the Terminal Handler driver only in systems having the BIOS but not the Human Interface.

This chapter describes how to select appropriate parameter values on the Terminal Handler screen, which is as follows:

```

Dynamic Debugger/Terminal Handler

(CF)   Clock Frequency [0-0FFFFH khz]           04CDH
(BR)   Baud Rate [0-0FFFFH]                     2580H
(UBP)  USART Base Port [0-0FFFFH]               0D8H
(UPS)  USART Port Separation [0-0FFH]           002H
(DDT)  Dynamic Debugger Terminal Handler [Yes/No] NO
(IOV)  Input/Output Version [Yes/No]            YES
(IMN)  Input Mailbox Name [1-12 characters]      RQTHNORMIN
(OMN)  Output Mailbox Name [1-12 characters]    RQTHNORMOUT
(IIL)  Input Interrupt Level [Encoded]          068H
(OIL)  Output Interrupt Level [Encoded]         078H
(RM)   Rubout Mode [Echo/Replace]              REPLACE
(BC)   Blanking Character [0-0FFH]             020H
(SIC)  Size of Characters [7,8]                 7
(CC)   Control-C Required [Yes/No]             NO
(UC)   Name of Control-C Object File [0-32 chars] NONE

Enter Changes [Abbreviation ?/ - new_value] :
```

```

(CF)   Clock Frequency [0-0FFFFH khz]           04CDH
```

Use this parameter to specify the clock frequency, in kHz, for the input to the timer used by the Terminal Handler. The timer, which can be on- or off-board, can be either one of the timers of a 8253 or 8254 Programmable Interval Timer component or later or timer number 2 on the 80130 component, in case one is present on your CPU board. Usually, the Terminal Handler can use the same time component that the Nucleus uses. If the Nucleus is using the same 8253 or 8254 timer, specify the same clock frequency as that of the Nucleus, (on the Hardware screen).

The default value 04CDH is equivalent to 1229 kHz. If your Terminal Handler uses the timer on the 80310 component, the value you supply for this parameter is divided by two before being used.

## TERMINAL HANDLER/DEBUGGER PARAMETERS

(BR)	Baud Rate [0-0FFFFH]	2580H
------	----------------------	-------

Use this parameter to specify the baud rate at which the terminal is to operate. The valid values are 19200, 9600, 4800, 1200, 600, 300, 150, and 110. The default value 2580 H is equal to 9600.

(UBP)	USART Base Port [0-0FFFFH]	0D8H
-------	----------------------------	------

Use this parameter to specify the base address of the first USART port. The USART is an 8251A component that communicates serially with the terminal and in parallel with other internal components. That is, it acts as a serial-parallel converter as it does I/O for the Terminal Handler.

The default value 00D8H is required if your CPU board is an Intel iSBC board.

(UPS)	USART Port Separation [0-0FFH]	02H
-------	--------------------------------	-----

Use this parameter to specify the number of bytes comprising the interval between consecutive ports on the USART for the Terminal Handler. The ICU uses the value you specify for this parameter with the value of the previous parameter to determine the addresses of all ports for the USART.

(DDT)	Dynamic Debugger Terminal Handler [Yes/No]	NO
-------	--	----

Use this parameter to specify whether this Terminal Handler will be used for debugging with the Dynamic Debugger, that is, whether this is to be the Terminal Handler that is embedded in the Dynamic Debugger.

If you:

- select the Dynamic Debugger on the Subsystems screen, and
- have the value of this parameter set to YES,

then you cannot change the value of this parameter to NO without first returning to the Dynamic Debugger parameter to zero. Conversely, if you want Dynamic Debugger in your system you must respond "Yes" to the (DDB) Dynamic Debugger [Yes/No] parameter and "Yes" to the parameter.

## TERMINAL HANDLER/DEBUGGER PARAMETERS

(IOV)	Input/Output Version [Yes/No]	YES
-------	-------------------------------	-----

Use this parameter to specify whether you want this Terminal Handler to be configured for both input and output or only for output. The default YES means configure for both input and output.

(IMN)	Input Mailbox Name [1-12 characters]	RQTHNORMIN
-------	--------------------------------------	------------

Use this parameter to specify the name of the mailbox that this Terminal Handler uses for input. Only one Terminal Handler can have an input mailbox with the default name RQTHNORMIN. If you use a Terminal Handler with the BIOS Terminal Handler Driver, you must use the default name. The name you specify must be limited to 12 characters, where upper and lower case are not equivalent.

(OMN)	Output Mailbox Name [1-12 characters]	RQTHNORMOUT
-------	---------------------------------------	-------------

Use this parameter to specify the name of the mailbox that this Terminal Handler uses for output. Only one Terminal Handler can have an output mailbox with the default name RQTHNORMOUT. If you use a Terminal Handler with the BIOS Terminal Handler Driver, you must use the default name. The name you specify must be limited to 12 characters, where upper and lower case are not equivalent.

(IIL)	Input Interrupt Level [Encoded]	068H
-------	---------------------------------	------

Use this parameter to specify the interrupt level that is to be used for input with this Terminal Handler. The input interrupt level must be higher in priority (lower in number) than the output interrupt level.

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

Master Level	Master Code	with Slave Attached Level	with Slave Attached Code	Master Level	Master Code	with Slave Attached Level	with Slave 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

The default 068H specifies master level 6.

## TERMINAL HANDLER/DEBUGGER PARAMETERS

(OIL) Output Interrupt Level [Encoded]	078H
--	------

Use this parameter to specify the interrupt level that is to be used for output with this Terminal Handler. The output interrupt level must be lower in priority (higher in number) than the input interrupt level. The priorities of your non-interrupt tasks should be lower (numerically higher) than the priority of the output interrupt task of this Terminal Handler.

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

The default 078H specifies master level 7, which causes the output interrupt task of the Terminal Handler to have priority 130. Therefore, if you accept the default value of this parameter, your non-interrupt tasks should have priorities lower (numerically higher) than 130.

(RM) Rubout Mode [Echo/Replace]	REPLACE
---------------------------------	---------

Use this parameter to specify how this Terminal Handler is to modify the terminal's display screen when an operator presses the RUBOUT key. (Regardless of the value you give for this parameter, the internal effect of the RUBOUT key is to delete the last character in the Terminal Handler's input buffer. Therefore, this parameter concerns only the external effects of pressing RUBOUT.)

If you select "Echo" as the value of this parameter, then each time the operator presses RUBOUT, a previously-entered character is displayed on the terminal's screen. If RUBOUT is pressed multiple times, previously entered characters are echoed in the reverse order from that in which they were originally entered. For example, if the operator enters the sequence A,B,C,RUBOUT, RUBOUT, RUBOUT, then the screen shows ABCCBA (and the internal key buffer is cleared of all evidence that any of these keys were pressed).

## TERMINAL HANDLER/DEBUGGER PARAMETERS

If you accept the default "Replace" for this parameter, then each time the operator presses RUBOUT, the most recently-entered character on the screen is replaced by the blanking character (and the cursor, if it is visible, moves backward one space). The next parameter on this screen defines the replacement character. When the blanking character is the blank character (space), as it usually is, then when the operator presses RUBOUT the last visible character on the screen disappears. Therefore, in this case, if you enter the sequence A,B,C,RUBOUT, RUBOUT, RUBOUT, the string ABC appears on the screen and then disappears again (and the cursor returns to where it was before the A key was pressed). Note that, in this case, the screen reflects the contents of the Terminal Handler's input buffer.

(BC) Blanking Character [0-0FFH]	020H
----------------------------------	------

Use this parameter to specify the ASCII value of the character that is to be used as the blanking character if the Rubout Mode parameter is set to "REPLACE". See the description of the Rubout Mode parameter for an explanation of the blanking character. The default 020H specifies the blank (space) character.

The ICU ignores this parameter if the Rubout Mode parameter is set to "Echo".

(SIC) Size of Characters [7,8]	7
--------------------------------	---

Use this parameter to specify the number of significant bits in the binary representation of each character. The default 7 is appropriate for ASCII character representation.

(CC) Control-C Required [Yes/No]	NO
----------------------------------	----

Use this parameter to specify whether your application system contains a procedure, names RQ\$ABORT\$AP, that is to be invoked automatically whenever an operator at the terminal enters a Control-C. If you accept the default NO, then there is no effect when a Control-C is entered at the terminal. If you specify the value YES, then you must provide a Control-C procedure that suits the needs of your application.

The *iRMX® Human Interface User's Guide* provides information about writing Control-C procedures.

(UC)	Name of Control-C Object File [0-32 chars.]	NONE
------	---	------

Use this parameter to specify the name of the object file or library containing your RQ\$ABORT\$AP procedure. See the description of the previous parameter for an explanation of RQ\$ABORT\$AP. The ICU ignores this parameter if the value of the CONTROL-C Required parameter is NO. On the other hand, if the CONTROL-C Required parameter is YES, then you must supply the appropriate name for this parameter; otherwise, your system will not generate properly.



## 12.1 INTRODUCTION

This chapter discusses how to respond to the prompts that appear on the "Nucleus" screen. The defaults shown are from the 28612.DEF. Note that values shown on the screens are sample values only. The exact values on each screen change according to the definition file used and your own changes.

## 12.2 NUCLEUS SCREEN

The following screen provides the ICU with information about the Nucleus.

```

(NUC)      Nucleus

(ASC) All Sys Calls [Yes/No]                YES
(PV)  Parameter Validation [Yes/No]         REQ
(ROD) Root Object Directory Size [0-0FF0H]  032H
(MTS) Minimum Transfer Size [0-0FFFFH]     040H
(DEH) Default Exception Handler [Yes/No/Deb/Use] YES
(NEH) Name of Ex Handler Object Module [1-32 chars]

(EM)  Exception Mode [Never/Program/Environ/All] NEVER
(SRR) Start Root Job from Reset [Yes/No]    NO
(BEA) BIST Entry Address [0:0=None/CS:IP]    0000:0000H
(CLM) Configure LBX Memory [Yes/No]         NO
(CPM) Configure PSB Memory [Yes/No]         NO
(PSP) PSB Start Page [0-0FFFFH]            0H
(RIE) Report Initialization Errors [Yes/No] YES
(RR)  Round Robin Scheduling [Yes/No]       YES
(TP)  Threshold Priority [0-0FEH]           08CH
(TS)  Time Slice [0-0FFH]                  05H
(IS)  Installation System [Yes/No]         NO

Enter [ Abbreviation = new value / Abbreviation ? / H ]
:

```

## NUCLEUS PARAMETERS

(ASC) All Sys Calls [Yes/No]	YES]
------------------------------	------

The default value for the "All Sys Calls" parameter line is "Yes". This means that the system you defined on the "Sub-System" screen requires all the Nucleus System calls. Without changing your responses on the "Sub-System" screen, you cannot change the default value on the "Nucleus" screen.

If your "Sub-System" screen does not include any subsystems other than the Basic I/O System, you can change the "Yes" value for this parameter to a "No". Making this change allows you to specify exactly which Nucleus system calls you want to include in your application system.

If you specify "No" to the "All Sys Calls" parameter line, the ICU allows you to make changes on seven additional screens that are system-call related. These screens are discussed later in this chapter. Note, however, that the ICU always includes the following Nucleus system calls: CREAT\$JOB, SUSPEND\$TASK, RESUME\$TASK, GET\$TASK\$TOKENS, SIGNAL\$EXCEPTION, and END\$INIT\$TASK. If you specified "MBII" as the value of the "(Bus) System Bus Type" parameter, two calls, GET\$INTERCONNECT and SET\$INTERCONNECT, are added to the previous list.

(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® I Nucleus System Calls Reference Manual* and the *iRMX® 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

## NUCLEUS PARAMETERS

(ROD) Root Object Directory Size [0 - 3840]	032H
---	------

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.

(MTS) Minimum Transfer Size [0 - OFF0H]	040H
---	------

You must specify the size of the minimum amount of memory that the Nucleus can transfer from a parent job to one of its offspring. The default value of the minimum transfer size is 64 16-byte paragraphs. If your application programs consistently request memory in larger than 64-paragraph blocks, you should increase the minimum transfer size. This increase cuts down on system overhead involved with transferring memory.

The value that you specify is used by the Nucleus in the following manner. If the memory to be borrowed is greater than the minimum transfer size, the Nucleus transfers the actual amount of the request from parent to child. If the memory to be borrowed is less than the minimum transfer size, the Nucleus transfers the amount of memory specified by the minimum transfer size.

You should specify a minimum transfer size large enough to handle a fairly large request or a multiple of smaller requests. Other than the risk of specifying a value that is too large (causing memory fragmentation), a fairly large request can be beneficial. The benefit is a more efficient use of memory. Since the difference between the actual request for memory and the minimum transfer size becomes part of the job's unallocated memory pool, future requests for memory can use the job's unallocated memory rather than needing a transfer of memory from a parent's memory pool.

(DEH) Default Exception Handler [Yes/No/Job/DEB] YES

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 "DEB", your system will use the iRMX System Debugger as the system default exception handler. This exception handler causes all hardware exceptions to be handled by the monitor, not the iRMX I 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® I Nucleus User's Guide*.

(NEH) Name of EX Handler Object Module [1-32 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 lines. 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 32 characters) of the object module which contains the RQSYSEX procedure. This module will be linked in when the system is built..

(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.

## NUCLEUS PARAMETERS

Programmer Control passes to the exception handler on programmer errors only.  
Environment Control passes to the exception handler on environmental errors only.  
All Control passes to the exception handler on all exceptional conditions.

(SRR) Start Root Job From Reset [Yes/No]	NO
--	----

MULTIBUS II support is provided for compatibility with the iRMX 86 Release 7.0 product only. MULTIBUS customers should ignore all MULTIBUS II-specific parameters.

If you answer "Yes" to the "(SRR) Start Root Job From Reset" parameter line, you will be configuring the Root Job (instead of the monitor software or the Bootstrap Loader) to start running upon system reset. A "No" response indicates that either your system is in ROM and you want control turned over to your monitor on resetting your system or that your system is in RAM.

If you specified "MBII" to the "(BUS) System Bus Type" parameter line (on the Hardware screen) and "Yes" to the "(SRR) Start Root Job From Reset" parameter line, the ICU will display several more parameter lines that can configure the Root Job to perform the following operations:

- Invoke the MBII BIST diagnostics on the host processor board.
- Automatically configure the starting and ending addresses of the iPSB and/or iLBX II memory board interfaces.

The following four parameters explain these options.

(BEA) BIST Entry Address [0:0=None]/CS:1P]	0000:0000H
--	------------

This parameter is Multibus II specific and indicates the entry point for the BIST (Built-In Self Test) diagnostic software. If you answered "MBII" to the "(BUS) System Bus Type" parameter line and "Yes" to the "(SRR) Start Root Job From Reset" parameter line, specify the entry point address of the BIST that resides on the host processor board. The address is entered in the format base:offset, for example, a base address of F000H and an offset address of 0H is entered as F000H:0H.

If you do not want to have the BIST invoked on system reset, enter "None".

(CLM) Configure LBX Memory [Yes/No]

NO

If you answered "MBII" to the "(BUS) System Bus Type" parameter line and "Yes" to the "(SRR) Start Root Job From Reset" parameter line, specify whether you want the Root Job to automatically configure the starting and ending addresses of all iLBX II memory boards during system initialization.

Answer "Yes" if you want the Root Job to configure iLBX II memory boards; answer "No" if you do not. To have the Root Job configure the addresses of all iLBX II and iPSB memory boards to the same addresses, answer "Yes" to this and the "(CPM) Configure PSB Memory" parameter lines.

The Root Job configures the starting and ending addresses of all iLBX II memory boards, starting at address zero for the first board encountered and continuing sequentially and consecutively for the remaining iLBX II memory boards. The Root Job's search begins with iLBX II cardslot 0.

(CPM) Configure PSB Memory [Yes/No]

NO

If you answered "MBII" to the "(BUS) System Bus Type" parameter line and "Yes" to the "(SRR) Start Root Job From Reset" parameter line, specify whether you want the Root Job to configure the starting addresses of all memory boards attached to the iPSB Parallel System Bus automatically during power-on or system reset.

One of two configurations is possible with this parameter:

- Configure only the memory boards attached to the iPSB bus.
- Configure only memory boards that share an iPSB and iLBX II bus interface.

Answer "Yes" if you want the Root Job to configure the addresses of iPSB memory boards; answer "No" if you do not. To have the Root Job configure the addresses of iLBX II and iPSB memory boards, answer "Yes" to this and the "(CLM) Configure iLBX Memory" parameter lines. In this case, the iLBX II and iPSB interfaces of memory boards available on the iLBX II bus are initialized to the same address. To have the Root Job configure the addresses of memory boards attached to the iPSB bus only, answer "Yes" to this parameter line and "No" to the "(CLM) Configure LBX Memory" parameter line.

The Root Job configures the starting addresses of all memory boards attached to the iPSB Parallel System Bus, starting with the first memory board encountered and at the address you specify for the "(PSP) PSB Start Page" parameter line. Answering "Yes" to the "(CPM) Configure PSB Memory" allows you to respond to the "PSP" PSB Start Page" parameter line.

## NUCLEUS PARAMETERS

(PSP) PSB Start Page [0 - 0FFFFH]

0H

This parameter line is displayed if you assigned the following values to the indicated parameter lines:

- "MBII" to the "(BUS) System Bus Type" parameter line
- "Yes" to the "(SRR) Start Root Job From Reset" parameter line
- "No" to the "(CLM) Configure LBX Memory" parameter line
- "Yes" to the "(CPM) Configure PSB Memory" parameter line

Use this parameter line to specify a value that represents the starting address of the first memory board attached to the iPSB bus. The value specifies the starting address in 64-byte increments as shown:

Value	Starting Address
0	0H (0K)
1	1000H (64K)
2	2000H (128K)
3	3000H (192K)
4	4000H (256K)
	•
	•
	•

The Root Job assigns the indicated address to the first memory board it encounters. The Root Job then determines the board's memory size by reading the memory board's appropriate interconnect register. The Root Job then adds the board's memory size to the previous starting address and uses the result as the starting address of the next memory board. This process continues until no more memory boards are found.

(RIE) Report Initialization Errors [Yes/No]

YES

Use this parameter to have initialization errors for all iRMX I 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® I Nucleus User's Guide*.

(RR) Round Robin Scheduling [Yes/No]	YES
--------------------------------------	-----

This parameter enables round-robin scheduling. See the *Introduction to the iRMX® Operating Systems* for an explanation of round-robin scheduling.

(TP) Threshold Priority [0-0FEH]	08CH
----------------------------------	------

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 08CH (140 decimal). This threshold value of 08CH is compatible with the HI, EIOS, and BIOS task priorities defined in the standard ICU definition files. Entering 0FFH disables round-robin scheduling.

(TS) Time Slice [0-0FFH]	05H
--------------------------	-----

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 05H. 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 0FFH in the "RRP" parameter, this parameter is ignored.

(IS) Installation System [Yes/No]	NO
-----------------------------------	----

Use this parameter to specify whether the iSDM monitor should be added to the system. If you specify "Yes", the system will enter the monitor after the boot file has been loaded and the message:

Insert the start-Up System Command Diskette and type <Return>

has been displayed.

## 12.3 SYSTEM CALL PARAMETERS

There are 57 fully-supported Nucleus System Calls. Depending on configuration choices that you make, such as operating system layers and debuggers, you can choose to exclude some system calls to save system memory. If you set the "ASC All System Calls" parameter to "NO", the ICU presents seven more screens.

When you select a system call that is not marked "YES" the ICU does not check to see how reasonable your choice is. For example, if you choose to include the system call DELETE\$SEGMENT in your system, the ICU does not check to determine if you have also included CREATE\$SEGMENT.

### 12.3.1 Object System Calls

The object-related Nucleus system calls are shown on the following screen.

(OBSC)	Object System Calls	
(GT)	Get Type	YES
(ED)	Enable Deletion	YES
(ED)	Disable Deletion	YES
(FD)	Force Delete	YES
(CO)	Catalog Object	YES
(LO)	Lookup Object	NO
(UO)	Uncatalog Object	YES
(SOS)	Set O.S. Extension	YES

Enter [ Abbreviation = new\_value / Abbreviation ? / H ]

Table 12-2. Show several configuration that are possible with this screen.

Table 12-2. Object System Calls

Configuration		
System Call	Dynamic Debugger	BIOS
Get Type	Required	Required
Enable Deletion	Required	Required
Disable Deletion	Required	Required
Force Delete	Optional	Required
Catalog Object	Required	Required
Lookup Object	Required	Required
Uncatalog Object	Optional	Required
Set O. S. Extension	Optional	Required

### 12.3.2 Job and Task System Calls Screen

The Job- and task-related Nucleus system calls are shown on the following screen.

```

(JTSC)      Job and Task System Calls

(DJ) Delete Job           YES
(OFF) Offspring          YES
(CT) Create Task         YES
(DT) Delete Task         YES
(SLP) Sleep              YES
(GP) Get Priority         YES
(SP) Set Priority         YES

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

## NUCLEUS PARAMETERS

Table 12-3. shows several configurations that are possible with this screen.

**Table 12-3. Job and Task System Calls**

Configuration		
System Call	Dynamic Debugger	BIOS
Delete Job	Optional	Optional
Offspring	Required	Optional
Create Task	Required	Required
Delete Task	Optional	Required
Sleep	Required	Required
Get Priority	Required	Optional
Set Priority	Required	Optional

### 12.3.3 Exchange System Calls Screen

The exchange-related Nucleus system calls are shown on the following screen.

(ECGSC) Exchange System Calls		
(MBX) Create, Send and Receive Mailbox		YES
(DMB) Delete Mailbox		NO
(SEM) Create, Send and Receive Semaphore		YES
(DSM) Delete Semaphore		YES
(REG) Create Region and Send Control		YES
(RC) Receive Control		YES
(AC) Accept Control		YES
(DR) Delete Region		YES
(PRT) Create, Send and Receive Port		YES
(DP) Delete Port		YES
Enter [ Abbreviation = new_value / Abbreviation ? / H ]		

**Table 12-4. Exchange System Calls**

Configuration		
System Call	Dynamic Debugger	BIOS
Create, Send and Receive Mailbox	Required	Required
Delete Mailbox	Optional	Required
Create, Send and Receive Semaphore	Optional	Required
Delete Semaphore	Optional	Required
Create Region and Send Control	Optional	Required
Receive Control	Optional	Required
Accept Control	Optional	Optional
Delete Region	Optional	Required
Create, Send and Receive Port	Optional	Required
Delete Port	Optional	Required

### 12.3.4 Free Space System Calls Screen

The free space-related Nucleus system calls are shown on the following screen.

```

(FSSC)      Free Space System Calls

(GS) Create Segment      YES
(DS) Delete Segment     YES
(GS) Get Size           YES
(GPA) Get Pool Attributes YES
(SPM) Set Pool Minimum  YES

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

Table 12-5 show several configuration that are possible with this screen.

Table 12-5. Free Space System Calls

System Call	Configuration	
	Dynamic Debugger	BIOS
Create Segment	Required	Required
Delete Segment	Required	Required
Get Size	Required	Optional
Get Pool Attributes	Optional	Optional
Set Pool Minimum	Optional	Optional
Get Priority	Required	Optional
Set Priority	Required	Optional

### 12.3.5 Interrupt System Calls Screen

The interrupt-related Nucleus system calls are shown on the following screen.

```

(INTSC)      Interrupt System Calls

(SI)  Set Interrupt           YES
(RI)  Reset Interrupt        YES
(ENI) Enter Interrupt        YES
(EXI) Exit Interrupt         YES
(SWI) Signal and Wait Interrupt YES
(ENA) Enable                 YES
(DSA) Disable                YES
(GL)  Get Level              YES

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

Table 12-6 shows several configurations that are possible with this screen.

**Table 12-6. Interrupt System Calls**

Configuration		
System Call	Dynamic Debugger	BIOS
Set Interrupt	Required	Required
Reset Interrupt	Optional	Required
Enter Interrupt	Optional	Optional
Exit Interrupt	Required	Optional
Signal and Wait Interrupt	Required	Required
Enable	Required	Optional
Disable	Required	Optional
Get Level	Optional	Required

### 12.3.6 Extension System Calls Screen

The extension-related Nucleus system calls are shown on the following screen.

```

(EXTSC)      Extension System Calls

(CEC) Create Extension and Composite      YES
(DE)  Delete Extension                    YES
(DC)  Delete Composite                    YES
(IC)  Inspect Composite                    YES
(AC)  Alter Composite                      YES

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

The extension-related Nucleus system calls are shown on the following screen.

## NUCLEUS PARAMETERS

Table 12-7. Extension System Calls

Configuration		
System Call	Dynamic Debugger	BIOS
Create Extension and Composite	Optional	Required
Delete Extension	Optional	Optional
Delete Composite	Optional	Required
Inspect Composite	Optional	Required
Alter Composite	Optional	Optional

### 12.3.7 Exception System Calls Screen

The exception-related Nucleus system calls are shown on the following screen.

```
(EXSC)      Exception System Calls

(SEH) Set Exception Handler          YES
(GEH) Get Exception Handler         YES

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

## 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.

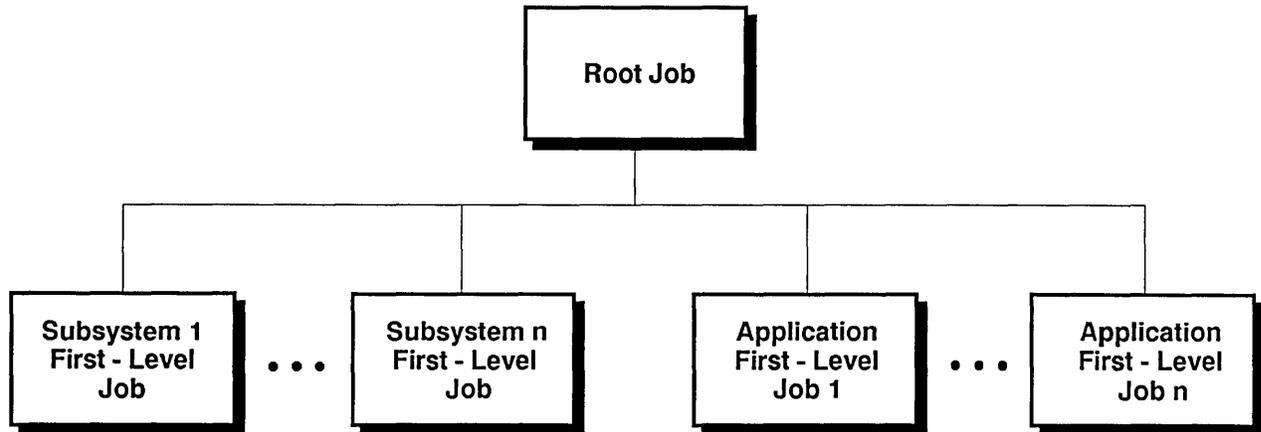
Do you need any/more User Jobs?

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.

## 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 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. Each subsystem: 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 12-1 illustrates an initial job tree.



W-1053

---

**Figure 13-1. Initial Job Tree**

---

First-level jobs can spawn a number of offspring jobs, beginning the dynamic tree structure of the system. The *iRMX® I Nucleus System Calls Reference Manual* describes how to create new tasks and jobs.

### 13.3 USER JOB SCREEN

You control the type and number of jobs that are created during initialization. Each "User Jobs" screen represents a different user job. The maximum number of user jobs the Nucleus can create during initialization is 32. 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 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 characters]	
(ODS) Object Directory Size [0-3840]	0
(PMI) Pool Minimum [20H - 0FFFFH]	60H
(PMA) Pool Maximum [20H - 0FFFFH]	0FFFFH
(MOB) Maximum Objects [1 - 0FFFFH]	0FFFFH
(MTK) Maximum Tasks [1 - 0FFFFH]	0FFFFH
(MPR) Maximum Priority [0 - 255]	129
(EHS) Exception Handler Address [CS:IP]	0000:0000H
(EM) Exception Mode [Never/Prog/Environ/All]	NEVER
(PV) Parameter Validation [Yes/No]	YES
(TP) Task Priority [0-255]	155
(TSA) Task Start Address [CS:IP]	0000:0000H
(DSB) Data Segment Base [0 - 0FFFFH]	0300H
(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 characters]

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.

## USER JOBS PARAMETERS

(PMI) Pool Minimum	[20H - 0FFFFH]	0060H
(PMA) Pool Maximum	[20H - 0FFFFH]	0FFFFH

These parameters specify the minimum allowable size of the user job's memory pool, 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 - 0FFFFH]	0FFFFH
-----------------------	--------------	--------

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 - 0FFFFH]	0FFFFH
---------------------	--------------	--------

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 Address	[CS:IP]	0000:0000H
---------------------------------	---------	------------

This parameter specifies the start address of the user job's exception handler. A value of 0H:0H indicates that the job uses the default system exception handler. A base address of 900H and an offset address of 384H should be entered as 900H:384H.

User jobs 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 a handler other than the default system exception handler, this parameter allows you to specify the start address of the other exception handler.

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

If you specified a non-zero value for the "Address of Exception Handler" 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 error 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 "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 the Nucleus Parameters chapter. 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.

## USER JOBS PARAMETERS

(TSA) Task Start Address [CS:IP]

0000:0000H

This parameter specifies the start address of this job's initialization task. A response to this parameter indicates that you have decided how to use your system RAM and/or ROM (as defined on your "Memory" screen). Your response also indicates that you have already compiled, linked, and located your application code, or that you know where LOC86 will locate your last system module.

Deciding where you want to locate your user job(s) is a decision that should be based on a number of factors. These factors affect how you respond to the "Memory" screen and are discussed in the Memory Parameters chapter.

If you have already compiled, linked and located your application code, LOC86 created a memory map that lists the base and offset address of the first instruction of this user job's initialization task. Use these addresses as your response to this parameter. The *Guide to the iRMX® I Interactive Configuration Utility* describes how to link and locate your user jobs.

If you have not compiled, linked, and located your job, you can still respond to this parameter. This approach requires that you know where LOC86 has and will continue to locate your root job. Because the root job is the last system module to be located, the memory map for the root job provides the highest address of any system module. You can find the memory map for the root job in the file ROOT.MP2. The *Guide to the iRMX® I Interactive Configuration Utility* presents a sample configuration that discusses how to read MAP files.

(DSB) Data Segment Base [0 - 0FFFFH]

0300H

Use this parameter to specify the base value of the initialization task's data segment. Before you answer this prompt, link and locate the user job. Refer to the *Guide to the iRMX® I Interactive Configuration Utility* for information on linking and locating user jobs.

The value you specify for this parameter line is directly related to the size control that you specify with the PL/M-86 compiler call. The size control can be LARGE, MEDIUM, or COMPACT. If you choose PL/M-86 LARGE model procedures, you must set the data segment base parameter to zero. (A value of zero indicates that the task itself assigns the data segment.)

If you choose either PL/M-86 MEDIUM or COMPACT model procedures, you can obtain the base address of the data segment from the locate map. It is recommended that you use the same PL/M-86 size control for all of your PL/M-86 jobs, and that any assembly language modules be compatible with this control.

The locate map produced by LOC86 lists both the base and offset addresses for the job. You must specify the base address of that module's DGROUP as the data segment for the user job. Obtain the base address from the locate map produced by LOC86. DGROUP includes the data, stack, and memory segments/classes for the MEDIUM model and the data segment/class for the COMPACT model. The constant segment/class is included in CGROUP if the ROM compiler control is used.

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

This parameter specifies 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 base address 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 specifies 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-86 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 LOC86 command. Refer to the "Locating Application Jobs" section of the *Guide To The iRMX® 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.

## USER JOBS PARAMETERS

### 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 them.

### 13.5 USER MODULES

After you respond "No" to the prompt "Do you have 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 of the linked and located object code for the first-level application jobs defined on the "I/O Jobs" screens. There should be a unique pathname for each user job that will be located in RAM. If the user job is to be located in ROM, do not specify a pathname for that user module on this screen.

Pathnames specified on this screen are added to the library which will contain your system RAM code. You specify the file name of this library using the "(RAF) RAM Code File Name" parameter in the "Generate File Names" screen.

Use the "D" command to delete entries made on this screen. You can also use the "I" command on this screen. For purposes of configuring the system, the order of your user module pathnames is not significant.

## ROM CODE PARAMETERS

## 14.1 INTRODUCTION

This chapter discusses how to respond to the parameters that appear on the "ROM code" screen. Unless you intend on programming at least one of your system modules into PROM, you will not find the information in this chapter particularly useful. If you are a configuring a RAM-based system, the responses to each of the parameters listed on the following screen should be "No".

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, plan the size and location of your RAM and PROM memory blocks, test your PROM-based system in RAM, and program your code into PROM. Appendix C of the *Guide to the iRMX® Interactive Configuration Utility* illustrates how to program your Nucleus code into PROM. Refer to this appendix for more information.

```
(ROM) ROM code

(UIR) UDI in ROM [Yes/No]          NO
(HIR) Human Interface in ROM [Yes/No] NO
(ALR) Application Loader in ROM [Yes/No] NO
(EIR) Extended I/O System in ROM [Yes/No] NO
(BIR) Basic I/O System in ROM [Yes/No] NO
(THR) Terminal Handler in ROM [Yes/No] NO
(SIR) SDB in ROM [Yes/No]          NO
(NIR) Nucleus in ROM [Yes/No]      NO
(RIR) Root Job in ROM [Yes/No]     NO

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

## ROM CODE PARAMETERS

(UIR)	UDI in ROM [Yes/No]	NO
-------	---------------------	----

Specify "Yes" to this parameter line if you wish to execute the Universal Development Interface code from ROM (PROM). The default "No" indicates that you wish to load the UDI from secondary storage into RAM.

(HIR)	Human Interface in ROM [Yes/No]	NO
-------	---------------------------------	----

Specify "Yes" to this parameter line if you wish to execute the Human Interface code from ROM (PROM). The default "No" indicates that you wish to load the Human Interface from secondary storage into RAM.

(ALR)	Application Loader in ROM [Yes/No]	NO
-------	------------------------------------	----

Specify "Yes" to this parameter line if you wish to execute the Application Loader code from ROM (PROM). The default "No" indicates that you wish to load the Application Loader from secondary storage into RAM.

(EIR)	Extended I/O System in ROM [Yes/No]	NO
-------	-------------------------------------	----

Specify "Yes" to this parameter line if you wish to execute the Extended I/O System (EIOS) code from ROM (PROM). The default "No" indicates that you wish to load the EIOS from secondary storage into RAM.

(BIR)	Basic I/O System in ROM [Yes/No]	NO
-------	----------------------------------	----

Specify "Yes" to this parameter line if you wish to execute the Basic I/O System (BIOS) code from ROM (PROM). The default "No" indicates that you wish to load the BIOS from secondary storage into RAM.

If you anticipate that you will be adding additional device drivers to your configuration in the near future, you may find it advantageous to keep your BIOS code in RAM.

(THR)	Terminal Handler(s) in ROM [Yes/No]	NO
-------	-------------------------------------	----

Specify "Yes" to this parameter line if you wish to execute the Terminal Handlers' code from ROM (PROM). The default "No" indicates that you wish to load the Terminal Handler from secondary storage into RAM.

(SIR) SDB in ROM [Yes/No]	NO
---------------------------	----

Specify "Yes" to this parameter line if you wish to execute the System Debugger code from ROM (PROM). The default "No" indicates that you wish to load the System Debugger from secondary storage into RAM.

(NIR) Nucleus in ROM [Yes/No]	NO
-------------------------------	----

Specify "Yes" to this parameter line if the Nucleus code is to execute from ROM (PROM). The default "No" indicates that you wish to load the Nucleus from secondary storage into RAM.

(RIR) Root Job in Rom [Yes/No]	NO
--------------------------------	----

Specify "Yes" to this parameter line if you wish to execute the Root Job from ROM (PROM). The default "No" indicates that you wish to load the Root Job from secondary storage into RAM.

If your Root Job is in PROM, you need to set the "(SRR) Start Root Job from Reset" parameter to "Yes" in the "Nucleus" screen. This parameter is described in the Nucleus Parameters chapter.



# INCLUDES AND LIBRARIES PARAMETERS

# 15

## 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. Note that values shown on the screens are sample values only. The exact values on each screen change according to the definition file used and your own changes.

```
(INCL) Includes and Libraries [1-45 Characters]

(UDF) UDI Includes and Libs           /RMX86/UDI/
(HIF) Human Interface Includes and Libs /RMX86/HI/
(EIF) Extended I/O System Includes and Libs /RMX86/EIOS/
(ALF) Application Loader Includes and Libs /RMX86/LOADER/
(BIF) Basic I/O System Includes and Libs /RMX86/IOS/
(SDF) System Debugger Includes and Libs /RMX86/SDB/
(THF) Terminal Handler and Dynamic Debugger Includes and Libs
                                           /RMX86/TH/
(NUF) Nucleus and Root Job Includes and Libs /RMX86/NUCLEUS/
(ILF) Interface Libraries               /RMX86/LIB/
(DTF) Development Tools Prefix          :LANG:

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

To generate a SUBMIT file that will assemble, bind, and build 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.



# GENERATE FILE NAMES PARAMETERS

# 16

## 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           /BOOT86/RMX86.ROM
(RAF) RAM Code File Name           /BOOT86/28612

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

```
(ROF) ROM Code File Name           /BOOT86/RMX86.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           /BOOT86/28612
```

This parameter specifies the pathname of the system's bootloadable file. The default file name used in this example is 28612, 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.



## 17.1 COMMENTS FOR BUILD FILES SCREEN

The "Comments for Build File" screen has been incorporated into the ICU for your convenience. It allows you to add comments 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 any way. 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 an iRMX I System based on the 28612.def file'
```

The screen has the following format:

```
(COMNT)          Comments for Build File
each line = 1-55 Characters - IN QUOTES
[ 1] =
Enter Changes [ Number= new_value / ^D Number / ? / H ]
:
```



- 386 processor 1-3
- 80186 initialization screen 1-12
- 80186 processor 1-3
- 80286 processor 1-3
- 8086 processor 1-2
- 8088 processor 1-3
- 8251A Driver Parameters 8251A-1
  - 8251A Terminal Driver Screen 8251A-1
  - Unit Information Screen 8251A-5
  - 8251A Terminal Driver Device-Unit Information 8251A-12
- 82530 Terminal Driver Parameters 82530-1
  - 82530 Terminal Driver Screen 82530-2
  - Unit Information Screen 82530-4
  - Device Unit Information Screen 82530-10
- 8274 Terminal Driver Parameters 8274-1
  - 8274 Terminal Driver Screen 8274-2
  - Terminal Driver Unit Information 8274-5
  - Terminal Driver Device-Unit Information Screen 8274-11

## A

- Application Loader 3-3
- Application Loader screen 5-1
- Automatic boot device recognition 7-2
- Automatic device recognition screen 7-5

## B

- Basic I/O Screen 8-1
- BIOS Screen 8-1
- BIOS System Calls Screen 8-8
- BIOS
  - named file system calls 8-11
  - non-file system calls 8-8
  - physical file system calls 8-9
  - stream file system calls 8-10
- Bootstrap Loader 7-3

## INDEX

### C

Comments for build files screen 17-1  
configuring first level jobs 13-1

### D

Default exception handler 12-5  
Dynamic-Debugger/Terminal Handler Screen 11-2

### E

EIOS  
    buffers 7-4  
    memory pool 7-4  
Exception handler, default 12-5  
Extended I/O system parameters 7-1

### F

File driver 7-6

### G

generate file names screen 16-1

### H

hardware screens 1-2

### I

I/O jobs screen 7-11  
I/O users screen 7-9  
includes and libraries screen 15-1  
initial program, definition 4-2  
Interrupt level of the System Debugger 10-1  
interrupt screen 1-9  
iRMX-NET 6-1  
iSBC 208 Driver Parameters 208-1  
    Device-Unit Information Screen 208-8  
    Driver Screen 208-1  
    Unit Information Screen 208-4  
iSBC 220 Driver Parameters 220-1  
    Driver Screen 220-1  
    Unit Information Screen 220-3  
    Device-Unit Information Screen 220-8

**I (continued)**

- iSBC 254 Driver Parameters 254-1
  - Driver Screen 254-1
  - Unit Information Screen 254-3
  - Device-Unit Information Screen 254-6
- iSBC 264
  - Device Unit Information Screen 264-5
  - Driver Parameters 264-1
  - Driver Screen 264-1
  - Unit Information Screen 264-3
- iSBC 286/10(A) Line Printer
  - Driver Screen LP286-1
  - Device Unit Information Screen LP286-3
- iSBC 534 Driver Parameters
  - Driver Screen 534-1
  - Unit Information Screen 534-3
  - Device Unit Information Screen 534-9
- iSBC 544A Driver Parameters 544A-1
  - Driver Screen 544A-1
  - Unit Information Screen 544A-4
- iSBX 218A Driver Parameters
  - Driver Screen 218A-1
  - Unit Information 218A-3
  - Device-Unit Information 218A-6
- iSBX 350-Line Printer
  - Driver Screen LP350-1
  - Device Unit Information LP350-4

**L**

- Logical names screen 7-8

## INDEX

### M

Mass Storage Controller  
  alternate cylinders MSC-9  
  bad track information MSC-11  
  buffers MSC-20  
  common update MSC-21  
  cylinder size MSC-7  
  device size MSC-18  
  Device-Unit Information screen MSC-12  
  granularity MSC-17  
  heads MSC-8  
  interrupt level MSC-2  
  max buffers MSC-21  
  number of cylinders MSC-8  
  priority MSC-3  
  request update timeout MSC-19  
  retries MSC-7  
  sectors per track MSC-9  
  step rate MSC-11  
  Unit Information screen MSC-4  
  unit number MSC-18  
  Driver screen MSC-2  
memory prompts 2-1  
MULTIBUS II hardware screen 1-17

### N

Non-resident user 7-5  
Nucleus parameters 12-1  
Nucleus screen 12-1

### O

Object directories 7-4

### P

Physical device 7-6  
Priority of tasks 7-4, 13

### Q

Query screen 7-10

**R**

- RAM
  - 80186 considerations 2-2
  - communications boards 2-2
  - driver 2-3
  - upper memory address 2-2
- RAM Driver Parameters RAM-1
  - Disk Driver Screen RAM-2
  - Driver Unit Information Screen RAM-3
  - Driver Device-Unit Information Screen RAM-5
- RAM-based, definition 2-2
- Remote File
  - Access Screen 6-1
  - Server Screen 6-4
- Resident programs and users 4-2
- resident user, definition 4-2
- ROM code screen 14-1
- ROM, monitor considerations 2-3
- Root job object directory 12-4

**S**

- SCSI Driver Parameters SCSI-1
  - Driver Unit Information Screen SCSI-5
  - Device-Unit Information Screen SCSI-10
- slave interrupt screen 1-11
- sub-systems screen 3-1
  - Basic I/O System 3-4
  - Dynamic Debugger 3-5
  - Extended I/O System 3-4
  - Human Interface 3-3
- Remote File Access 3-3
- System Debugger 3-4
- Terminal Handler 3-5
- Universal Development System 3-2
  - subsystem dependencies 3-2
  - Subsystems Screen
- System Debugger parameters 10-1
- System device 7-6
- System-level and job-level parameter validation 12-3

## **INDEX**

### **T**

Terminal Communications Controller

Driver Parameters TCC-1

Driver Screen TCC-2

Unit Information Screen TCC-4

Terminal Handler Driver

Device-Unit Information Terminal Handler-1

### **U**

user job screen 13-2

user modules screen 13-8

### REQUEST FOR READER'S COMMENTS

Intel's Technical Publications Departments attempt to provide publications that meet the needs of all Intel product users. This form lets you participate directly in the publication process. Your comment will help us correct and improve our publications. Please take a few minutes to respond.

Please restrict your comments to the usability, accuracy, organization, and completeness of this publication. If you have any comments on the product that this publication describes, please contact your Intel representative.

- 1. Please describe any errors you found in this publication (include page number).

---

---

---

---

- 2. Does this publication cover the information you expected or required? Please make suggestion for improvement.

---

---

---

---

- 3. Is this the right type of publication for your needs? Is it at the right level? What other types of publications are needed?

---

---

---

---

- 4. Did you have any difficulty understanding descriptions or wording? Where?

---

---

---

- 5. Please rate this publication on a scale of 1 to 5 (5 being the best rating). \_\_\_\_\_

NAME \_\_\_\_\_ DATE \_\_\_\_\_

TITLE \_\_\_\_\_

COMPANY NAME/DEPARTMENT \_\_\_\_\_

ADDRESS \_\_\_\_\_ PHONE ( ) \_\_\_\_\_

CITY \_\_\_\_\_ STATE \_\_\_\_\_ ZIP CODE \_\_\_\_\_  
(COUNTRY)

Please check here if you require a written reply

**'D LIKE YOUR COMMENTS . . .**

document is one of a series describing Intel products. Your comments on the back of this form will us produce better manuals. Each reply will be carefully reviewed by the responsible person. All ments and suggestions become the property of Intel Corporation.

u are in the United States, use the preprinted address provided on this form to return your ments. No postage is required. If you are not in the United States, return your comments to the Intel s office in your country. For your convenience, international sales office addresses are printed on ast page of this document.



**NO POSTAGE  
NECESSARY  
IF MAILED  
IN THE  
UNITED STATES**

**BUSINESS REPLY MAIL**  
FIRST CLASS PERMIT NO. 79 HILLSBORO, OR



POSTAGE WILL BE PAID BY ADDRESSEE

**Intel Corporation  
OMSO Technical Publications, MS: HF3-72  
5200 N.E. Elam Young Parkway  
Hillsboro, OR 97124-9978**



## 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

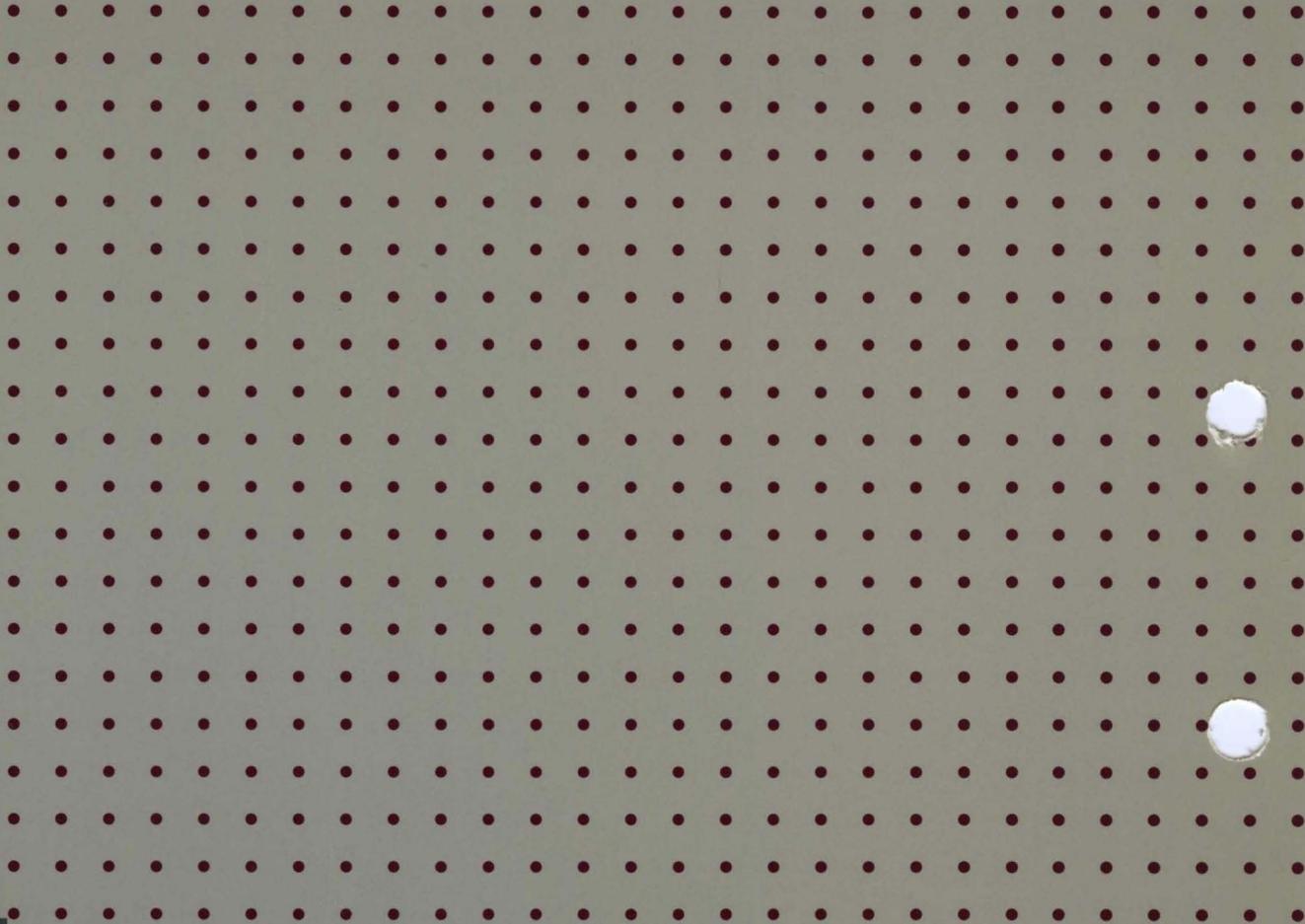
SPAIN  
Intel Iberia  
Calle Zurbaran 28-IZQDA  
28010 Madrid

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





INTEL CORPORATION  
3065 Bowers Avenue  
Santa Clara, California 95051  
(408) 987-8080