iRMX®
Installation and Startup

RadiSys Corporation
5445 NE Dawson Creek Drive
Hillsboro, OR   97124
(503) 615-1100
FAX: (503) 615-1150
www.radisys.com
07-0683-01
December 1999

December 1999

Quick Contents

Notational Conventions

Most of the references to system calls in the text and graphics use C syntax instead of PL/M (for example, the system call **send_message** instead of **send$message**). If you are working in C, you must use the C header files, *rmx_c.h*, *udi_c.h* and *rmx_err.h*. If you are working in PL/M, you must use dollar signs ($) and use the *rmxplm.ext* and *error.lit* header files.

This manual uses the following conventions:

- Syntax strings, data types, and data structures are provided for PL/M and C respectively.
- All numbers are decimal unless otherwise stated. Hexadecimal numbers include the `H` radix character (for example, `0FFH`). Binary numbers include the `B` radix character (for example, `11011000B`).
- Bit 0 is the low-order bit. If a bit is set to 1, the associated description is true unless otherwise stated.
- `Data structures and syntax strings appear in this font.`
- r? denotes an invisible file.
- **System call names and command names appear in this font.**
- PL/M data types such as BYTE and SELECTOR, and iRMX data types such as STRING and SOCKET are capitalized. All C data types are lower case except those that represent data structures.
- Whenever this manual describes I/O operations, it assumes that tasks use BIOS calls (such as **rq_a_read**, **rq_a_write**, and **rq_a_special**). Although not mentioned, tasks can also use the equivalent EIOS calls (such as **rq_s_read**, **rq_s_write**, and **rq_s_special**) or UDI calls (**dq_read** or **dq_write**) to do the same operations.

Note

Notes indicate important information.

CAUTION

Cautions indicate situations which may damage hardware or data.

Contents

1 Introduction

2 Installing on iRMX development/target systems that are PC-compatible Platforms with no DOS

3 Installing on iRMX development/target systems that are PC-compatible Platforms with DOS

4 Installing on iRMX Development/Target Systems that are Multibus II Platforms

5 Installing the iRMX III OS on Multibus I Systems

6 Installing on Windows NT systems used as iRMX development systems

7 DOXRMX Specifics

8 iRMX for PCs Specifics

9 Getting Acquainted with the Operating System

A Installed Directories

B Limitations

C Configuration Requirements for PC Platforms

D Changing Memory Size During the iRMX III OS Installation

E Hardware Modifications for the iRMX III OS Installation

Tables

Table 5-1. BSL Device Name, Diskette, and Controller Information 69
Table 5-2. Interleave Factors 73
Table 5-3. Device Names and Numbers of Files for ST506 and ESDI Hard Disks 74
Table 5-4. Device Names and Numbers of Files for Generic SCSI Hard Disks on an SBC 221S Controller 74
Table 5-5. Device Names and Numbers of Files for SCSI Hard Disks (386/12S and 486/12S only) 75
Table 5-6. Directories Renamed in a Multibus I Installation 78
Table 5-7. Terminal Names Defined in the *termcap* File 80
Table 5-8. Files Generated by mksys for CPU Boards 83

Figures

CHOOSING YOUR INSTALLATION

This section helps you decide which iRMX installation procedure to perform. This chapter is in the section:

Introduction 1

Welcome to the iRMX® Family of Operating Systems!

The iRMX® Operating System runs on Intel386™, Intel486™, Pentium®, Pentium Pro, Pentium II, Celeron, and Pentium III microprocessor-based single board computers and other microcomputers based on these microprocessors. There are three versions of the iRMX Operating System (OS): iRMX for PCs, DOSRMX, and the iRMX III OS. With any iRMX OS, you can develop applications that require preemptive, priority-based multitasking and real-time response.

This manual:

- Guides you through the installation procedure appropriate for you
- Shows you how to get help on using the OS
- Shows you how to boot the system and enter simple commands
- Shows you how to get the most from the documentation

Package Inventory

Before installing an iRMX OS, check the contents of your package against the contents listed in the Customer Letter. Make sure that you have all the CDs, tapes, diskettes, documentation, and other materials. If your package is not complete, contact RadiSys Customer Support.

Software

The iRMX Software Package includes:

- iRMX III OS software, iNA 960, iRMX-NET, TCP/IP and NFS, demonstration software, and the Aedit text editor on a CD ROM.
- DOSRMX Load Diskette
- iRMX for PCs Boot Diskette
- SCSI Support Load Diskette

These tools and languages are also provided with the OS:

- iC-386

- Support files for developing applications using third-party compilers.
- ASM386 and utilities (BLD386, BND386, LIB386, and MAP386)
- PL/M-386
- Soft-Scope Debugger

Documentation

The following operating system documentation is included in this package:

- *Installation and Startup* (this manual)
- *Introducing the iRMX Operating Systems(in PDF form on the CD)*
- *System Concepts(in PDF form on the CD)*
- *Programming Concepts for DOS(in PDF form on the CD)*
- *Driver Programming Concept(in PDF form on the CD)s*
- *Network User's Guide and Reference(in PDF form on the CD)*
- *TCP/IP for the iRMX Operating System(in PDF form on the CD)*
- *System Call Reference(in PDF form on the CD)*
- *Command Reference(in PDF form on the CD)*
- *C Library Reference(in PDF form on the CD)*
- *Programming Techniques(in PDF form on the CD)*
- *AEDIT Text Editor(in PDF form on the CD)*
- *System Debugger Reference(in PDF form on the CD)*
- *System Configuration and Administration(in PDF form on the CD)*
- *MSA for the iRMX Operating System(in PDF form on the CD)*
- *ICU User's Guide and Quick Reference(in PDF form on the CD)*
- *iC-386 Compiler User's Guide(in PDF form on the CD)*
- *Soft-Scope Debugger User’s Guide(in PDF form on the CD)*
- *Intel386 Family Utilities User's Guide(in PDF form on the CD)*
- *ASM386 Macro Assembler Operating Instructions(in PDF form on the CD)*
- *ASM386 Assembly Language Reference(in PDF form on the CD)*
- *PL/M-386 Programmer's Guide(in PDF form on the CD)*

Registration

Please complete the Product Registration card included with your package. Sending in your completed registration card enables you to receive telephone support during the term of the product warranty period.

To register the software, fill out the software registration card and fax it or mail it to the address/phone number listed in the Dear Customer Letter. Your registration number is listed on the registration certificate.

Note

European customers must affix appropriate postage if mailing the registration. Alternatively, you may fill in the registration card and fax it to the USA fax number given on the card.

Important “Getting Started” Files

After performing the installation you might want to examine these files:

readme.txt describes known problems and workarounds, and important last-minute information that did not make it into the manuals.

rnotes.txt describes incorporation information and lists problems fixed since the last release.

features.txt describes major and minor changes in this release of the operating system.

You can find each of these files in the top-level directory *rmx386* after the installation is complete.

Deciding Which Installation Procedure to Perform

Chapters 2 through 6 present different iRMX OS installation procedures. The one you use depends on the platform on which you are installing the software. If the target platform is a PC Architecture system, you can install the iRMX III .2.3 software on either a DOS partition or an iRMX partition. Make your choice using the following guidelines:

Install the iRMX OS on an iRMX named partition if either of these apply:

- You have no need for DOS and/or the DOS file system on your application system
- You need added security through file access permissions

If your requirements match these criteria, Chapter 2 will guide you through the installation process.

Install the iRMX OS on a DOS partition if any of these apply:

- You need to maintain existing DOS files
- You are not changing partition size or adding an iRMX partition
- You desire to have DOS running concurrently with iRMX as a low priority task

If your requirements match these criteria, Chapter 3 will guide you through the installation process.

Chapter 2 Use the install procedure in this chapter if these conditions are true:

- The platform you are installing software on is a PC system or a PC-compatible board in a Multibus I or II system. PC-compatible boards in a Multibus system include:

  SBC PCP4 (Multibus I)
  SBC 486DX66 (Multibus II)
  SBC p5090[V] (Multibus II)
  SBC p5150[ISE] (Multibus II)
  SBC p5200 (Multibus II)

- You want the iRMX OS to run independent of DOS (installation of iRMX for PCs).

Chapter 3 Use the install procedure in this chapter if these conditions are true:

- The platform you are installing software on is a PC system or a PC-compatible board in a Multibus I or II system. PC-compatible boards in a Multibus system include:

  SBC PCP4 (Multibus I)
  SBC 486DX66 (Multibus II)
  SBC p5090[V] (Multibus II)
  SBC p5150[ISE] (Multibus II)
  SBC p5200 (Multibus II)

- You want the iRMX OS to run concurrently with DOS (installation of DOSRMX.

Chapter 4 Use the install procedure in this chapter if the platform you are installing software on is a Multibus II board that cannot or will not be run as a PC.

These boards include:

CPU Boards	**CPU + I/O Server Boards**
SBC 386/120	SBC 386/258
SBC 486/150	SBC P5090ISE
SBC 486/133SE	SBC P5120ISE
SBC 486/166SE	
MIX 486DX66	
SBC 486DX66	
SBC P5090CPU	
SBC P5150CPU	
SBC P5200CPU	

See also: *Introducing the iRMX Operating Systems* for information on iRMX for PCs, DOSRMX, and the iRMX III OS versions of the iRMX OS.

Chapter 5 Use the install procedure in this chapter if the platform you are installing software on is a Multibus I board that cannot be run as a PC.

These boards include:

CPU Boards	**Controller Boards**
SBC 386/12(S)	SBC 214
SBC 486/12(S)	SBC 221(S)
SBC 386/12S	SCSI controller on SBC 386/12S and 486/12S boards
SBC 486/12S	SCSI controller on SBC 386/12S and 486/12S boards

SBC 386/3x

Chapter 6 Use the install procedure in this chapter if you are installing the iRMX III.2.3 Development Software on a Windows NT 4.0 system on which you will be doing cross development of iRMX applications and systems.

What To Do If You Need More Help

If you have problems installing the iRMX OS or have a question, call Customer Support. Use the remainder of this chapter to learn about support services terms and conditions, customer support offerings, and software licensing information.

Support Services Terms & Conditions (World-Wide Services)

1. **TO OBTAIN SERVICE**
   To be eligible for service under these terms and conditions, an authorized representative of your company must sign and return the Product and Support Registration Certificate.

2. **PHONE SERVICE**
   Please refer to Customer Support Offerings in this chapter for specific product detail. Technical phone service is available by calling one of the numbers listed in the Dear Customer Letter.

3. **EXCLUDED SERVICES**
   The services provided specifically exclude: (i) service for Software not used on PCs based on the Intel386, Intel486 or Pentium, Pentium Pro, Pentium II, Celeron, and Pentium III microprocessor or on non-PC hardware not designated by the product documentation as a supported configuration; (ii) hardware maintenance; (iii) problem assistance for non-iRMX software products; (iv) service or support for source code and (v) service for software or hardware that has been discontinued by RadiSys. You may request application assistance or engineering time not covered under this Agreement. Customer Support may, at its option, issue a quotation for this service on a best reasonable efforts basis.

4. **RESTRICTIONS/LICENSES**
   All services provided are subject to these terms and conditions. The Software is provided under the terms and conditions of the RadiSys Developer Software License Agreement.

5. **TERM OF THE AGREEMENT**
The period of software service begins upon RadiSys's receipt of the unaltered, executed RadiSys Product and Support Registration Certificate and shall continue for a ninety (90) day period.

6. **CONTINGENCIES**
   RadiSys shall not be liable for its failure to perform any obligation under this Agreement due to unforeseen circumstances or causes beyond RadiSys' reasonable control, including but not limited to, acts of God, war, riot, embargoes, acts of civil or military authorities, fire, flood, accident, strikes, inability to secure transportation, facilities, fuel, energy, labor or materials. Time for performance will be extended by Force Majeure.

7. **LIMITED WARRANTY**
   EXCEPT AS SET FORTH HEREIN, RADISYS DISCLAIMS THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. ALL OTHER WARRANTIES, EXPRESS OR IMPLIED, ARE EXCLUDED AND SHALL NOT APPLY TO THE SERVICES PROVIDED UNDER THIS AGREEMENT. INTEL NEITHER ASSUMES NOR AUTHORIZES ANY OTHER PERSON TO ASSUME ANY OTHER LIABILITIES IN CONNECTION WITH THE SERVICE OF THE SOFTWARE.

8. **LIMITATION OF LIABILITY**
   RADISYS SHALL NOT BE RESPONSIBLE FOR ANY LOSS OF PROFITS, LOSS OF DATA, LOSS OF USE, INCIDENTAL, CONSEQUENTIAL, OR SPECIAL DAMAGES WHETHER UNDER THIS AGREEMENT OR OTHERWISE. IN NO EVENT WILL RADISYS' TOTAL LIABILITY EXCEED THE SUM PAID TO RADISYS FOR THE SERVICES PROVIDED HEREUNDER.

9. **CONTROLLING LAW**
   This Agreement shall be governed by, subject to and construed according to the laws of the State of Oregon. For purposes of applying Oregon law, this Agreement shall be deemed to have been entered into and wholly performed in Oregon.

REFER THE PHONE NUMBER AND ADDRESS IN THE DEAR CUSTOMER LETTER AT WHICH YOU MAY CONTACT CUSTOMER SUPPORT IF YOU HAVE ANY QUESTIONS ABOUT THIS AGREEMENT.

Customer Support Offerings

Refer to the RadiSys Website at www.radisys.com for the latest information on a wide variety of customer support offerings

Training

RadiSys can offer an iRMX OS Training workshop built to your needs. To find out more, see our Website at www.radisys.com .

INtime and iRMX Users Group (IRUG)

IRUG is an international organization, formed in 1984, that promotes the sharing of information and experience of hundreds of members worldwide relative to the use of the iRMX and INtime OSs in a wide variety of applications. IRUG hosts an annual conference, bringing members, RadiSys management and technical staff together for two days of intensive technical seminars. IRUG maintains the iRMX Program Library with hundreds of useful programs for the iRMX I, II, III, iRMX for PCs, and DOSRMX operating systems.

Check out IRUG on its Website at www.irug.com.

Software Licensing Information

As a registered licensee of the iRMX OS developer kit, you can begin development of your application right away. However, when you are ready to distribute your application, please follow the instructions below. Because the iRMX OS is an OEM[1] software product, RadiSys assumes that you will be making copies of the operating system to distribute with your product. RadiSys requires you to pay a royalty fee for each copy you make. This section explains how you can make copies of the software and describes how and when to use RadiSys Incorporation Permit Labels and Licenses.

[1] OEM (Original Equipment Manufacturer): Someone who purchases products, integrates them, adds their own value, and then sells the integrated product to an end user.

How to Make Copies of the Software

Every run-time copy of the iRMX OS development kit you make must include the iRMX OS run-time and requires a royalty fee

First, you must sign a Software Redistribution License agreement, available from the iRMX Incorporations Administrator. Contact the iRMX Incorporations Administrator at www.radisys.com to obtain the agreement. After signing the agreement, you can make run-time copies. The files which can be included in your product are listed in *rnotes.txt*. This file may be found, once the iRMX Operating System is installed, in the */rmx386* directory.

Note

The developer kit contains other software that cannot be copied, such as the compilers, a debugger, and parts of the networking software. Contact the iRMX Incorporations Administrator if you need to make copies of these items.

Incorporation Administration and Permit Labels

Prior to redistributing the iRMX run-time software, you must sign a Software Redistribution License Agreement and purchase an incorporations permit label for each iRMX run-time copy that you make. Both of these are available from the iRMX Incorporations Administrator. Contact the iRMX Incorporations Administrator at www.radisys.com to obtain the agreement.

Affix the labels to a part of the product *shipped to your end customer*. The labels may be affixed to the chassis of a system, a diskette, or a manual. The end customer's site must keep one label for each run-time copy of the iRMX operating system. The sole exclusion to this rule is copies made for backup purposes.

□ □ □

iRMX INSTALLATION PROCEDURES

This section describes the iRMX OS installation procedures. These are the chapters in this section:

Installing on iRMX development/ target systems that are PC-compatible Platforms with no DOS 2

Before performing these installation steps, ensure that your computer meets the minimum requirements necessary to complete the installation.

Hardware Requirements

- A PC based on an Intel386 processor (or above) or a PC-compatible Multibus board
- At least 4 Mbytes of total memory (conventional and extended). If a SCSI device is to be used in the system, a minimum of 8 Mbytes is required.
- An ATA hard disk drive
- An ATAPI IDE CDROM drive
- A 1.44 Mbyte 3.5-inch diskette drive.
- 52 Mbytes of free hard disk space, depending upon the product configuration you select during installation.
- EGA, VGA, or Super VGA video controller.
- Optionally, a supported SCSI controller, if you are installing onto a SCSI disk.

Software Requirements

If you wish to boot from a Primary iRMX Partition, you must insure that this partition is entirely within the first 8 Gbytes on the disk. In order to make sure that all PC BIOS bootloaders can load from the partition, it is also wise to confine it to within the first 1024 cylinders of the disk.

Install Procedure

To perform the installation, follow the steps in this section

➠ **Note**

Once the iRMX OS has been installed, be sure to review the *readme.txt, rnotes.txt*, and *features.txt* files in the */rmx386* directory. These files contain the latest iRMX OS product information. You can also view updated information on the RadiSys Corporation Website at www.radisys.com.

Step 1. Boot from the PC Boot Disk

A. Insert Boot Diskette

Insert the PC *Boot Disk* in diskette drive A:. Make sure that the diskette is not write-protected.

B. Reboot System

Reboot the system using <Ctrl-Alt-Del> or the system's reset button.

C. Boot the iRMX III OS

With the iRMX for PCs Third Stage visible on the console, either hit the Enter key or wait until the 30 second timeout expires, thus causing the iRMX III OS to boot.

D. Logon as Install

With the iRMX OS waiting at the Logon Prompt on the console, type install<enter> with a password of <enter>.

Step 1A. Loading the SCSI Drivers

If you require access to a SCSI device, you need to load the SCSI drivers at this point. Otherwise skip to Step 2.

A. Insert PC Drivers Diskette And Attach It

Insert the *PC Drivers* diskette in drive A:. Attach it as a DOS volume:

```
attachdevice A as :f: dos
```

B. Identify the Required SCSI Driver

Supported driver names are:

- 1542: Adaptec 1542 and similar ISA adapters.
- 78XX: Adaptec 78xx series PCI SCSI controllers.
- SYMBIOS: NCR/Symbios/LSI logic 53C8xx series PCI SCSI controllers

C. Load the Required Driver

Load the driver using the following command, and the name selected in step B:

submit :f:loadscsi(:f:,*name*)

where *name* is replaced with one of 1542, 78XX or SYMBIOS.

When the PCI server and driver has loaded, new DUIBS for the SCSI devices are available. Hard disk DUIB names have the following format:

GSCW5_A*n*

where *n* is the SCSI unit number.

Hard disk partition DUIB names have the following format:

GSCW5_A*n*M*p*

where *n* is the SCSI unit number, and *p* is the partition number.

There are two CDROM DUIB names, CDA0 and CDA1. CDA0 is for a CDROM drive attached as SCSI unit 5, and CDA1 is for a CDROM drive attached as SCSI unit 6.

Step 2. Hard Drive Preparation (Existing Primary DOS Partition)

➯ **Note**

Perform this step only if you are installing the iRMX III OS on an existing Primary DOS partition.

With the iRMX OS waiting at the Logon Prompt on the console, type install<enter> with a password of <enter>.

This step assumes you are installing the iRMX OS onto an existing Primary DOS partition, and that the partition on which you are installing the iRMX OS has enough space.

First, attach the target Primary DOS partition as **:w:**.

If are using the resident ATAPI driver, the DUIB to use in this attachdevice operation will be one of the following:

hda1	1st Partition on the C drive
hda2	2nd Partition on the C drive
hda3	3rd Partition on the C drive
hda4	4th Partition on the C drive

(It is assumed here that the C: drive is the primary drive on the first IDE controller).

If you have loaded SCSI drivers, the disk DUIB names are encoded as follows:

GSCW5_A*u*M*p*

where *u* is the SCSI unit number, and *p* is the master partition number. For example, the first partition on the drive at SCSI unit 2 would be GSCW5_A2M1.

For example, if the Primary DOS partition on which you wish to install the iRMX OS is the 2nd partition on an IDE hard disk, type

```
ad hda2 as w
```

Next, if a previous version of the iRMX OS resides on the target DOS partition, save your custom versions of the following files in the /rmx386/save subdirectory by entering the appropriate commands from the iRMX console as follows:

```
createdir :w:386/save<enter>
copy :w:rmx386/config/terminal to
       :w:rmx386/save/terminal
copy :w:rmx386/config/termcap to :w: rmx386/save/termcap
copy :w:rmx386/config/udf to :w: rmx386/save/udf
copy :w:rmx386/config/cdf to :w: rmx386/save/cdf
copy :w:rmx386/config/r'?'init to :w: rmx386/save/rinit
copy :w:rmx386/config/rmx.ini to :w: rmx386/save/rmx.ini
copy :w:rmx386/config/loadinfo to
       :w:rmx386/save/loadinfo
copy :w:rmx386/config/alias.csd to
       :w:rmx386/save/c_alias.csd
copy :w:user/super/prog/r'?'logon to
       :w:rmx386/save/s_rlogon
copy :w:user/super/prog/r'?'logoff to
       :w:rmx386/save/s_rlogof
copy :w:user/world/prog/r'?'logon to
       :w:rmx386/save/w_rlogon
copy :w:user/world/prog/r'?'logoff to
       :w:rmx386/save/w_rlogof
```

No other disk preparation is needed, so go to Step 6

Step 3. Hard Drive Preparation (New Primary DOS Partition)

➠ **Note**

Perform this step only if you are creating a new Primary DOS Partition and installing the iRMX III OS on it.

This step assumes you are creating a new Primary DOS Partition on which you wish to install the iRMX OS. It assumes that you will create a partition large enough to contain the installed OS.

➠ **Note**

Do NOT create a partition larger than 2Gbytes if you want to ensure compatibility with the Microsoft disk partitioning scheme.

First use the RDISK utility to create the new Primary DOS Partition. Invoke RDISK from the iRMX command line as follows:

```
rdisk hda0
```

If you have loaded SCSI drivers, the DUIB names for the whole disk (as opposed to paritions) are encoded as follows:

GSCW5_A*u*

where *u* is the SCSI unit number.

Follow the menu provided by RDISK to assign a new Primary DOS Partition and to give it no more than 510 Mbytes of disk space.

Next, attach the target Primary DOS partition just created as :w:. The DUIB to use in this attachdevice operation will be one of the following:

hda1	First partition on the C drive
hda2	Second partition on the C drive
hda3	Third partition on the C drive
hda4	Fourth partition on the C drive

If you are using a SCSI driver, the DUIB names for partitions are encoded as follows:

GSCW5_A*u*M*p*

where *u* is the SCSI unit number, and *p* is the master partition number, starting at 1.

For example, if the Primary DOS partition you created and on which you wish to install the iRMX OS is the second partition on an IDE hard disk, type

```
ad hda2 as w dos
```

Next, format the new DOS partition as follows:

```
Format :w:DOS_part
```

Now go to Step 6

Step 4. Hard Drive Preparation (Existing Primary iRMX Partition)

➨ **Note**
Perform this step only if you are installing the iRMX III OS on a existing Primary iRMX Partition.

This step assumes you are installing the iRMX OS onto an existing Primary iRMX Partition, and that the partition on which you are installing the iRMX OS has enough space.

First, attach the target Primary iRMX partition as :w:. The DUIB to use in this attachdevice operation will be one of the following:

hda1	1st Partition on the C drive
hda2	2nd Partition on the C drive
hda3	3rd Partition on the C drive
hda4	4th Partition on the C drive

If you have loaded SCSI driver, the DUIB names for partitions are encoded as follows:

GSCW5_A*u*M*p*

where *u* is the SCSI unit number and *p* is the master partition number.

For example, if the Primary iRMX partition on which you wish to install the iRMX OS is the 2nd partition on an IDE hard disk, type

ad hda2 as w

Next, if a previous version of the iRMX OS resides on the target iRMX partition, save your custom versions of the following files in the /rmx386/save subdirectory by entering the appropriate commands from the iRMX console as follows:

```
createdir :w:386/save<enter>
copy :w:rmx386/config/terminal to :w: rmx386/save/terminal
copy :w:rmx386/config/termcap to :w: rmx386/save/termcap
copy :w:rmx386/config/udf to :w: rmx386/save/udf
copy :w:rmx386/config/cdf to :w: rmx386/save/cdf
copy :w:rmx386/config/r’?’init to :w: rmx386/save/rinit
copy :w:rmx386/config/rmx.ini to :w: rmx386/save/rmx.ini
copy :w:rmx386/config/loadinfo to :w: rmx386/save/loadinfo
copy :w:rmx386/config/alias.csd to :w: rmx386/save/c_alias.csd
copy :w:user/super/prog/r’?’logon to :w: rmx386/save/s_rlogon
copy :w:user/super/prog/r’?’logoff to :w: rmx386/save/s_rlogof
copy :w:user/world/prog/r’?’logon to :w: rmx386/save/w_rlogon
copy :w:user/world/prog/r’?’logoff to :w: rmx386/save/w_rlogof
```

Now, use the iRMX FORMAT command to update the iRMX Second Stage Bootstrap Loader on the target partition as follows:

```
Format :w: bs pcboot
```

If you wish to boot from this partition, use the iRMX RDISK command to make sure it is the ACTIVE partition as follows:

```
Rdisk hda0
```

Follow the menu of options presented by the RDISK command to select the iRMX partition as the ACTIVE partition.

No other disk preparation is needed, so go to Step 6

Step 5. Hard Drive Preparation (New Primary iRMX Partition)

Note
Perform this step only if you are creating a new Primary iRMX Partition and installing the iRMX III OS on it.

This step assumes you are creating a new Primary iRMX Partition on which you wish to install the iRMX OS. It assumes that you will create a partition large enough to contain the installed OS.

First use the RDISK utility to create the new Primary iRMX Partition. Invoke RDISK from the iRMX command line as follows:

```
rdisk hda0
```

Follow the menu provided by RDISK to assign a new Primary iRMX Partition. It can be of any size.

Next, attach the target Primary iRMX partition just created as :w:. The DUIB to use in this attachdevice operation will be one of the following:

hda1	First partition on the C drive
hda2	Second partition on the C drive
hda3	Third partition on the C drive
hda4	Fourth partition on the C drive

If you have loaded SCSI driver, the DUIB names for partitions are encoded as follows:

GSCW5_A*u*M*p*

where *u* is the SCSI unit number and *p* is the master partition number.

For example, if the Primary DOS partition you created and on which you wish to install the iRMX OS is the second partition on an IDE hard disk, type:

```
ad hda2 as w named
```

Next, format the new iRMX partition as follows:

```
Format :w: RMX2.3 il=1 named48 pcboot<enter>
```

The FORMAT command will evaluate the partition size and lay down either a named32 or named48 file system and calculate the optimum number of files that can be supported by the selected file system.

If you wish to boot from this partition, use the iRMX RDISK command to make sure it is the ACTIVE partition as follows:

```
rdisk hda0
```

Follow the menu of options presented by the RDISK command to select the iRMX partition as the ACTIVE partition.

Step 6. Install the iRMX OS from the CDROM onto the prepared Primary Partition

A. Log off.

Type `logoff<enter>` and press Return.

B. Logon as Finish

With the iRMX OS waiting at the Logon Prompt on the console, type finish<enter> with a password of <enter>.

C. Attach the CDROM

From the iRMX prompt, attach the iRMX Installation CDROM as **:cd:** as follows:

- If the CD ROM drive is the master on the primary IDE controller, type:
  ```
  ad cda as cd cdrom
  ```
- If the CD ROM drive is the slave on the primary IDE controller, type:
  ```
  ad cdb as cd cdrom
  ```
- If the CD ROM drive is the master on the secondary IDE controller, type:
  ```
  ad cdc as cd cdrom
  ```
- If the CD ROM drive is the slave on the secondary IDE controller, type:
  ```
  ad cdd as cd cdrom
  ```

If the CDROM drive is a SCSI device, there are two DUIBs available:

- If the CDROM drive is at SCSI unit 5, type:
  ```
  ad cda0 as cd cdrom
  ```
- If the CDROM drive is at SCSI unit 6, type:
  ```
  ad cda1 as cd cdrom
  ```

Enter the Setup Screen in the System BIOS during system boot to determine how your CD ROM drive is configured.

D. Install the Software onto the prepared partition

With the target hard disk partition already attached as **:w:** from the previous steps, invoke the iRMX submit file RMXINSTL.CSD on the CDROM as follows:

```
esubmit :cd:rmxinstl(:w:,:cd:) over :w:install.out e
```

E. Copy the SCSI configuration file to the parameters file

If you are going to boot from a SCSI drive, you need to copy the SCSI parameters to the parameters file (RMX.INI) on the hard drive:

```
copy :A:name.ini after :W:rmx386/config/rmx.ini
```

where *name* is the name you used to load the SCSI drivers in step 1A.

F. Reset the System

Remove the iRMX Installation CDROM and Boot Disk from their drives and reset the system. If booting from an iRMX partition, go to Step 7A. If loading from a DOS partition, go to Step 7B.

Step 7. Booting the newly installed iRMX OS

A. Booting the newly installed iRMX OS from an iRMX partition (IDE)

If you are booting from a SCSI drive, go to step 7B now.

When the system reboots, you will see the iRMX for PCs Third Stage user interface on the system console. To boot the default PC image, change the Boot File Name to /boot32/pc and press <Enter>. Once the iRMX OS has booted and you have logged on, you can copy the /boot32/pc image to /boot32/rmx so that autoboot can be achieved. Likewise, you can generate your own custom application system and change its name to /boot32/rmx so that it will autoboot.

Go to step 8.

B. Booting the newly installed iRMX OS from an iRMX partition (SCSI)

When the system reboots, you will see the iRMX for PCs Third Stage user interface on the system console.

To boot the PC SCSI image, change the *Boot File Name* to /boot32/pcs.

To enter the name of the DUIB where you installed the boot system, change the *Boot device* field.

Press <Enter> to boot the system.

Once the iRMX OS has booted and you have logged on, you can copy the /boot32/pcs image to /boot32/rmx so that autoboot can be achieved. Likewise, you can generate your own custom application system and change its name to /boot32/rmx so that it will autoboot.

To make sure that the bootloader will find your DUIB without intervention, you will need to regenerate your image and change the parameters of the C DUIB to match your boot partition DUIB. The SCSI disk partition DUIBS are found in the PCHD screens of the ICU. By default, the C DUIB looks for SCSI ID 0, partition 1.

Step 8. Where to Go Next

The installation process is complete. Here are other areas of the manual that you should investigate:

Chapter 7 describes system-level operations specific to DOSRMX.

Chapter 8 helps you get acquainted with the iRMX OS.

Chapter 9 shows you where to go beyond this manual.

Appendix B describes what is new in this release of the OS.

Appendix C describes iRMX limitations.

Note

Chapter 9 describes how to install/access the online iRMX manuals and the Windows Help file for the iRMX system calls supplied with this release. Installation of both these features is performed separately from the installation procedure in this chapter.

❑ ❑ ❑

Installing on iRMX development/ target systems that are PC-compatible Platforms with DOS 3

Before performing these installation steps, ensure that your computer meets the minimum requirements necessary to complete the installation.

Hardware Requirements

- A PC based on an Intel386 processor (or above) or a PC-compatible Multibus board
- At least 4 Mbytes of total memory (conventional and extended).
- A 1.44 Mbyte 3.5-inch diskette drive.
- 52 Mbytes of free hard disk space, depending upon the product configuration you select during installation.
- EGA, VGA, or Super VGA video controller.

Software Requirements

- DOS 5.0 or later.

Install Procedure

To perform the installation, follow the steps in this section

✏ **Note**

Once the iRMX OS has been installed, be sure to review the *readme.txt, rnotes.txt,* and *features.txt* files in the */rmx386* directory. These files contain the latest iRMX OS product information. You can also view updated information on the RadiSys Corporation Website at www.radisys.com.

Step 1. Boot DOS from the C: drive

Step 2. Set up the DOS Environment to be compatible with the iRMX III.2.3 (DOSRMX) Operating System.

Before DOSRMX can be run on the PC, you must modify the C:\CONFIG.SYS and C:\AUTOEXEC.BAT files to insure that a proper, compatible DOS environment is first established.

A. Config.Sys changes

- Make sure there are no memory management drivers or TSRs loaded such as EMM.SYS
- Set FILES to 64
- Set BUFFERS to 64

B. Autoexec.bat changes

- Make sure SMARTDRV.EXE is not loaded/started
- Make sure there are no memory management drivers or TSRs loaded such as EMM.SYS
- Add <drive>:\DOSRMX and <drive>:\INTEL\BIN to the PATH
- Add line “TYPE <drive>:\DOSRMX\STARTUP.TXT”
- Add line “SET :INCLUDE:=<drive>:\INTEL\INCLUDE

  where <drive> is the DOS drive on which you installed the iRMX III.2.3 Software

C. Reboot DOS

Step 3. Load DOSRMX from the DOSRMX Load Disk

A. Insert DOSRMX Load Diskette

Insert the DOSRMX Startup *Disk* in diskette drive A:. Make sure that the diskette
is not write-protected.

B. Modify DOS batch file A:\rmx.bat as needed

The batch file **a:\rmx.bat** contains a command to load a floppy TSR (floppy.com). On some older systems that use an AMI BIOS, this TSR is not needed and may even be incompatible. If the invocation of the rmx.bat batch file results in a floppy read error, comment out (REM) the line which invokes the floppy TSR in the **a:\rmx.bat** batch file.

C. Load the iRMX III.2.3 Operating System

Type **a:rmx <Enter>**.

D. Logon as Install

With the iRMX OS waiting at the Logon Prompt on the console, type install<Enter> with a password of <Enter>.

Step 4. Hard Drive Preparation (Existing DOS Partition)

This step assumes you are installing the iRMX OS onto an existing DOS partition, and that the partition on which you are installing the iRMX OS has enough space. It also assumes that DOS is running as an iRMX task, and that the system CD is accessible as a DOS drive.

First, attach the target DOS partition as **:w:**. The DUIB to use in this attachdevice operation will correspond to the DOS drive ID of the target partition as follows:

c_dos	C: DOS partition
d_dos	D: DOS partition
e_dos	E: DOS partition
f_dos	F: DOS partition

For example, if the DOS partition on which you wish to install the iRMX OS is the DOS e: drive, type

```
ad e_dos as w edos
```

Next, if a previous version of the iRMX OS resides on the target DOS partition, save your custom versions of the following files in the /rmx386/save subdirectory by entering the appropriate commands from the iRMX console as follows:

```
createdir :w:386/save<enter>
copy :w:rmx386/config/terminal to :w: rmx386/save/terminal
copy :w:rmx386/config/termcap to :w: rmx386/save/termcap
copy :w:rmx386/config/udf to :w: rmx386/save/udf
copy :w:rmx386/config/cdf to :w: rmx386/save/cdf
copy :w:rmx386/config/r’?’init to :w: rmx386/save/rinit
copy :w:rmx386/config/rmx.ini to :w: rmx386/save/rmx.ini
copy :w:rmx386/config/loadinfo to :w: rmx386/save/loadinfo
copy :w:rmx386/config/alias.csd to :w: rmx386/save/c_alias.csd
copy :w:user/super/prog/r’?’logon to :w: rmx386/save/s_rlogon
copy :w:user/super/prog/r’?’logoff to :w: rmx386/save/s_rlogof
copy :w:user/world/prog/r’?’logon to :w: rmx386/save/w_rlogon
copy :w:user/world/prog/r’?’logoff to :w: rmx386/save/w_rlogof
```

Step 5. Install the iRMX OS from the CDROM onto the prepared Partition

A. Log off

Type `logoff<enter>` and press Return.

B. Logon as Finish

With the iRMX OS waiting at the Logon Prompt on the console, type finish<Enter> with a password of <Enter>.

C. Attach the CDROM

From the iRMX prompt, attach the iRMX Installation CD as **:cd:** using the DUIB that corresponds to the drive ID with which the CDROM is accessible from DOS as follows:

If the CD ROM drive has a DOS drive ID of e:

```
ad e_dos as cd edos
```

If the CD ROM drive has a DOS drive ID of f:

```
ad f_dos as cd edos
```

etc.

D. Install the Software onto the prepared partition

With the target hard disk partition already attached as **:w:** from the previous steps, invoke the iRMX submit file RMXINSTL.CSD on the CDROM as follows:

```
submit :cd:rmxinstl(:w:,:cd:) over :w:install.out e
```

E. Reset the System

Remove the iRMX Installation CDROM and Boot Disk from their drives and reset the system

Step 6. Loading the newly installed iRMX III.2.3 Operating System from a DOS partition via DOS

To load the iRMX III.2.3 OS (DOSRMX), simply type

```
rmx <drive>
```

where <drive> is the DOS drive on which you installed the iRMX III.2.3 Software

Step 7. Where to Go Next

The installation process is complete. Here are other areas of the manual that you should investigate:

Chapter 7	describes system-level operations specific to DOSRMX.
Chapter 8	helps you get acquainted with the iRMX OS.
Chapter 9	shows you where to go beyond this manual.
Appendix B	describes what is new in this release of the OS.
Appendix C	describes iRMX limitations.

➡ **Note**

Chapter 9 describes how to install/access the online iRMX manuals and the Windows Help file for the iRMX system calls supplied with this release. Installation of both these features is performed separately from the installation procedure in this chapter.

❑ ❑ ❑

Installing on iRMX Development/ Target Systems that are Multibus II Platforms[1] 4

Use this procedure if the platform on which you are installing is a Multibus II board and you will not be running it as a PC.

The standard Multibus II installation assumes you have an I/O Server board compatible with the System 520 system.

Use the instructions in this section to make a first-time installation of the OS, or to install over a previous version. During the installation you will:

- Run the system diagnostics.
- Bootstrap load the system from a diskette.
- Log on as the system manager (Super).
- Format the computer's hard disk, either partially or completely.
- Install the OS software and support products.
- Generate a boot system and reboot the Multibus II system.
- Establish the default boot system.
- Merge user files with the new software (if you installed over a previous version of the OS).

Before performing these installation steps, ensure that your computer meets the minimum requirements necessary to complete the installation.

Minimum System Requirements

A minimum of 75 Mbytes of space on your hard disk is required to install the iRMX software. A minimum of 4 Mbytes of RAM is required to boot the iRMX OS.

[1] MSA bootable platform

Validated Multibus II Boards

See also: *MSA for the iRMX Operating System* for additional information on configuring multiple processor boards

The iRMX OS has been validated on these Multibus II boards:

CPU Boards	**CPU + I/O Server Boards**
SBC 386/120	SBC 386/258
SBC 386/258	SBC 486/166SE
SBC 486/150	SBC P5090
SBC 486/166SE	SBC P5150
MIX 486/DX4	SBC P5090ISE
	SBC P5150ISE
	SBC P5200

Hardware Requirements

- A MSA-bootable Multibus II system with one of the following CPU boards in slot 0 (the system boot master)
    - SBC 386/258
    - SBC 486/166SE
    - SBC P5090 / P5090/ISE and related products
    - SBC P5200

Preparing for the Installation

1. If you are not installing the OS on a new Multibus II system, back up all current files on tape. In a later step you will format the Multibus II system's hard disk. At that point you choose whether to do a complete or partial format. Choose the partial format to retain all current files on your disk. However, first back up the files to protect these files.
2. If you are using an Intel Multibus II system, write down the type of the hard disk in your system. This information should be listed on documents delivered with the hardware. If there are two hard disks, choose the device type of the disk on which you want to make the installation. If you are building a custom system, the supported device types are listed in Table 4-1.
3. Determine the type of the I/O Server board in your system. If you do not know what type of I/O Server board is in your system, refer to the back panel.

4. Read your Multibus II system's hardware installation manual to be sure that the equipment is set up correctly and that the terminal is connected with the correct cable.

5. If you are going to install from the CDROM, add a SCSI CDROM drive to your Multibus II system and configure it to have SCSI host ID 5.

6. Choose which diskette(s) your system requires from this list, based on the I/O Server board in your Multibus II system:

   - SBC 386/258 Start-up Boot Diskette (for SBC 386/258 and 386/258D boards)
   - SBC 486/166SE Start-up Boot Diskette (for SBC 486/166SE and 486/133SE boards)
   - SBC P5150ISE Start-up Boot Diskette (SBC P5120ISE, SBC P5090ISE, SBC P5150, and SBC P5090 boards)
   - SBC P51200 Start-up Boot Diskette (SBC P5200 board)
   - You will also need the iRMX Installation CDROM

7. Check that all the diskettes are write-protected. (Boot diskettes for the SBC P5* boards are write-enabled.)

The installation process described in this chapter can take two hours or more, depending on the type of disk formatting performed and the amount of software being installed.

Step 1. Testing the System Hardware

1. If you are using the SBX 279A module for your system console, make sure the SBX scan bit in your I/O Server board's firmware is enabled.

   See also: SBX scan, *Firmware User's Guide for MSA Firmware*

2. Turn on the system console and the Multibus II system.

3. When you see asterisks or other characters displayed on the screen, enter:

   ```
   u
   ```

   The u will not show on the screen.

4. The system prompts you for a selection from this menu. Select Run System Diagnostics by typing 1 at the Enter number ? prompt:

   ```
   1 Run System Diagnostics
   2 Go to Operator Interface
   ```

```
    3 Go to Bootphase

Enter number ?  1
```

The 1 will not show on the screen. If you do not make a choice in 9 seconds, the system defaults to running the diagnostics (selection 1).

5. Test results look like Figure 4-1. Make a note of which Multibus II system slots hold the I/O Server board and the optional processor board. You need this information for later steps.

6. At this prompt, type a `y` to invoke the MTH:

```
Do you want to enter additional commands? Enter y or n [n]?
y <CR>
```

If the hardware is working correctly, the status column displays PASS for all the tests, and the display is similar to Figure 4-1. If one of the tests fails, invoke the MTH to choose further tests by pressing <Ctrl-C>. To display the MTH help menu, enter `h`. If you cannot find and fix the problem, call Intel's Customer Support.

See also: Using the MTH Commands, *Firmware User's Guide for MSA Firmware*

The following display is only an example. If your system hardware configuration differs, so will the display.

```
System Configuration and BIST Status
   Slot  Product Code   Vendor   Rev   Class SSR   SLR   Test   Status
    0    386/258        Intel    01H   F2H   00H   A3H    18     PASS
    2    386/120        Intel    01H   F1H   00H   23H    21     PASS
    6    186/410        Intel    03H   E4H   00H   23H    14     PASS
    7    186/530        Intel    03H   E5H   00H   23H    14     PASS
SYSTEM CONFIGURATION AND BIST STATUS.............................PASS

iPSB Tests
  Slot   2 Test 128   Unsolicited Message Receive                PASS
  Slot   2 Test 129   Unsolicited Message Send                   PASS
  Slot   2 Test 130   Solicited Message Receive 2-C              PASS
  Slot   2 Test 131   Solicited Message Send 2-C                 PASS
   _                             _                                  _
   _                             _                                  _
iPSB TESTS...................................................... PASS

Pre-configured Systems Tests
  Slot   0 Test   1   Init Routine                               PASS
  Slot   0 Test  12   Data & Parity Dynamic RAMs                 PASS
  Slot   0 Test  20   RAM, Refresh                               PASS
  Slot   0 Test  22   Display On-Board RAM Size
On-Board RAM Size is     4 Megabytest
                                                                 PASS
  Slot   0 Test  24   PSB Interconnect Access                    PASS
  Slot   2 Test   1   Init Routine                               PASS
  Slot   2 Test  12   Data & Parity Dynamic RAMs                 PASS
  Slot   2 Test  22   Display On-Board RAM Size
On-Board RAM Size is     4 Megabytes
                                                                 PASS
PRE-CONFIGURED SYSTEMS TESTS.................................... PASS

Do you want to enter additional commands?  Enter y or n  [n]  ?
```

Figure 4-1. Sample Board and System Test Report

Step 2. Booting From the Start-up Boot Diskette

In this step, you instruct the MTH to change a boot parameter that makes the system boot from software on diskette (in the drive with SCSI ID 0). The instructions show the prompt as MTH [0], but the number in brackets might not be zero for your system. The number indicates which system slot holds the I/O Server board.

1. Insert the Start-up Boot Diskette for your I/O Server board into the diskette drive.

2. Enter the **mp** command to change a bootstrap parameter:

   ```
   MTH [0] mp <CR>
   ```

3. At this display, set the `bl_unit` parameter. (The parameter value is zero, not the letter O.):

   ```
   Modify Boot Parameters for slot   0:
   Modify parameters: "#" deletes, "." quits, <CR> advances
   new parameter

   bl_unit = 0 <CR>
   ```

4. Enter the device name for the boot device (diskette drive):

   If you are using the TEAC 5.25 inch drive with the embedded SCSI controller, enter:

   ```
   rq_sd = t55_0 <CR>
   ```

   For the TEAC 3.5 inch drive with the embedded SCSI controller, enter:

   ```
   rq_sd = t235_0 <CR>
   ```

5. If your I/O server board is an SBC 386/258, SBC 486/166SE, or SBC P5120ISE proceed to item 6.

   If you are using an SBC 486/133SE, you must set the `bl_config_file` parameter.

   For an SBC 486/133SE, enter:

   ```
   bl_config_file = /msa/config/bps.133 <CR>
   ```

   For an SBC P5090ISE, enter:

   ```
   bl_config_file = /msa/config/bpsise.90 <CR>
   ```

   For an SBC P5120, enter:

   ```
   bl_config_file = /msa/config/bps.120 <CR>
   ```

For an SBC P5090, enter:

```
bl_config_file = /msa/config/bps.90 <CR>
```

For an SBC 386/258D, enter:

```
bl_config_file = /msa/config/bps.d <CR>
```

6. Press the <CR> key in response to each of the next two prompts:

```
new parameter
<CR>
save changes ([y] or n)
<CR>
```

7. Enter the **b** (Bphase) command. This tells the MTH to invoke the boot phase:

```
MTH [0] b <CR>
```

The screen clears, messages indicating boot status are displayed, and then the OS signs on. The boot diskette portion of the OS is loaded onto the I/O Server board.

Note

A logon banner should appear within two minutes after you invoke the boot phase. If after several minutes nothing appears on the screen, you may not have shut the diskette drive door, or you entered the `bl_unit` parameter wrong. Reset the Multibus II system and start over with Step 1.

Step 3. Logging On and Setting the Date and Time

The console displays this bannerhead:

```
*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*

                  iRMX* Operating System

                        Release 2.3

* iRMX is a Registered Trademark of RadiSys Corporation

*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*

 Logon:
```

Log on as the system manager, Super. The system manager has access to all files used during the installation process, and can change access rights to all iRMX files.

1. Enter the Super logon at the prompt:

   Logon: **super <CR>**

2. The system prompts you for a password. For now, enter the default password for Super, `passme`. The password must be in lower-case letters. It is not shown on the screen.

   Password: **passme <CR>**

   The OS signs on with this message:

```
iRMX HI CLI, Vx.y: USER = 0
Copyright years Intel Corporation
All rights reserved
-
```

3. When you see a prompt for the date, enter the date in one of these formats. (If the displayed date is already correct, enter `e` instead.)

   month/day/year   (for example, `01/15/1994`)
   day month year   (for example, `15 JAN 1994`)
   day month year   (for example, `15 JANUARY 1994`)

   Example:

   DATE: **01/15/1994 <CR>**

   If your system does not include a global clock, you will see the message:

```
E$EXIST, Global Clock not present - will query for local
date
```

4. The system prompts you for the time, which is based on a 24-hour clock. Enter the time in hours:minutes:seconds format, using either one or two digits for each field. If you omit the minutes and seconds, the system sets them to zero. (If the displayed time is already correct, enter `e` instead.)

   For example, at 7:36 AM and 40 seconds, enter `07:36:40` or at 2:00 PM, enter `14:00`.

   Example:

   ```
   TIME:  7:36:40 <CR>
   ```

   If you enter either the date or the time in the wrong format, you get a message to re-enter it.

   If your system does not include a global clock, you will see the message:

   ```
   E$EXIST, Global Clock not present - will query for local
   time
   ```

Two **submit** files named *:prog:alias.csd* and *:config:alias.csd* are executed. These files define aliases for command names. Use the **alias** command to see the complete list of aliases. Do not modify the aliases in *:config:alias.csd* because the iRMX software uses them as defined.

The OS defines one logical name, *:tape:*. In these steps, the command that formats the hard disk requires this logical name when you are installing the OS from tape.

Step 4. Determining Disk Parameters

The **submit** commands in these steps require parameters specific to the hardware in your system. In this step, the format command determines some the parameters required based on your peripheral controller board and hard disk. You may override these if you wish. For an installation over a previous release, you need only the device name parameter, described in Item 2.

1. The **submit** commands in the next steps include parameters that are specific to the type of hard disk. Use Table 4-1, to determine the parameters for your disk drive. You will need the device name, and optionally number of files.
2. Refer to Table 4-1 to locate the type of hard disk on which you will install the software. When you locate the type of hard disk, make a note of the device names and number of files for that disk type. If your system includes multiple

hard disks, choose the appropriate device name based on which disk you will use. Device names for the first hard disk (or for systems with only one hard disk) end in 2. Device names for a second hard disk end in 3.

Note

The `files` parameter used in the **submit** commands is the total number of files that can be created on the hard disk. If the number of files needs to be changed later, you will have to back up your files and reformat the hard disk. The recommended numbers listed in Table 4-1 are based on 125 files per Mbyte of disk capacity. You may choose a different number. The maximum number of files that the OS can handle is 65,528.

Table 4-1. Device Names and Numbers of Files for SCSI Hard Disks

Hard Disk Model Number	**Device Name**	**SCSI ID Number (*n*)**	**Recommended Number of Files**
Maxtor XT-3170S	M3170_*n*	2, 3	22000
Maxtor XT-4170S	M4170_*n*	2, 3	22000
Maxtor XT-4380S	M4380_*n*	2, 3	48000
Maxtor XT-8380S	M8380_*n*	2, 3	48000
Maxtor XT-8760S	M8760_*n*	2, 3	65528
Hewlett-Packard 97536	HP97536_*n*	2, 3	40000
Siemens Megafile 2300	S2300_*n*	2, 3	48000
Micropolis 1578	MP1578_*n*	2, 3	39000
Quantum Pro 80S	Q280_*n*	2, 3	11000
Quantum Pro 40S	Q280_*n*	2, 3	5500
Generic SCSI 512 byte granularity	GSCW5_*n*	2, 3	allow **format** to determine
Generic SCSI 1024-byte granularity	GSCW_*n*	2, 3	allow **format** to determine
Generic SCSI[1] partitioned with a 512-byte granularity	GSCW5_*n*M*x*E*y*	2, 3	allow **format** to determine

[1] Generic SCSI partitioned with a 512-byte granularity are supported by the SBC 486/133, 486/166, and P5* family boards only.

3. Replace the *n* in the device name in the second column in Table 4-1 with the appropriate SCSI ID Number from the third column. For example, the device name for the Maxtor XT-4380S drive with SCSI ID 2 is M4380_2. Use the Generic SCSI DUIBs for hard disks not listed in the table.

   See also: Your hard disk drive reference manual for the appropriate SCSI ID number;
   Appendix F, *Command Reference* for information on partitioned SCSI hard disk drives

4. If you wish to create multiple partitions on your hard disk, you must take the following action:

⚠ **CAUTION**

Do not perform this option if you want to keep the files on your hard disk. The Full format option deletes all files. Proceed to the Partial format option if you want to save existing files.

a. Enter the following **submit** command:

```
submit :tape:part_form (device, format-options)
```

where *device* is the device name for the SCSI hard drive you want to partition. If you require extra options for format, specify them a the second parameter.

b. Use the **rdisk** command to partition the SCSI device.

```
rdisk device
```

See also: Partitioning Example for the iRMX OS, *Command Reference,* Appendix F

Step 5a. Formatting the Disk and Installing Software From Tape

(If you are intending to install from CDROM, proceed to step 5b)

There are two options to format the hard disk: a Full format or a Partial format (track 0 only).

Full format Formats the entire hard disk.

Choose this option if you have a new Multibus II system with nothing on the hard disk. (All Intel microcomputers are shipped with unformatted hard disks.) You may also choose this option for a system whose software you want to completely erase.

Partial format Formats only track 0 of the hard disk with the second stage of the iRMX bootstrap loader.

Choose this option if your disk was formatted with any earlier release of the iRMX OS. This option does not delete any current files, but it does rename some of them.

Either option also installs the OS, along with any included support software.

Enter your commands after the hyphen (-) prompt.

Full Format Option

CAUTION

Do not perform this option if you want to keep the files on your hard disk. The Full format option deletes all files. Proceed to the Partial format option if you want to save existing files.

1. Insert the OS tape.

2. To format the hard disk, invoke a **submit** command with the proper parameters for the disk you want to format. Substitute a device name, and any format options you require.

   ```
   submit :tape:form_disk(device, format-options) <CR>
   ```

   Note

   If you performed partitioning in Step 4, make sure you specify the appropriate partitioned DUIB name from Table 4-1.

   This command is valid for a Maxtor XT-4380S disk that has a SCSI ID of 2:

   ```
   submit :tape:form_disk(m4380_2, full)  <CR>
   ```

   The OS formats the hard disk. Then it displays a summary of alternate tracks assigned for bad tracks found during the format.

```
The following tracks have been assigned an alternate:

     {disk information}
```

   Following formatting, the **submit** file sets up a directory structure on the hard disk. It invokes the **restore** command and copies all the files from the tape to the hard disk. Enter `Yes` <CR> when switching tapes. It establishes World permission rights for files. Then it switches the system device from the diskette drive to the hard disk, so that the OS reads future commands from the hard disk. The process can take up to 45 minutes.

3. When you receive the logon prompt, remove the Start-up Commands diskette and OS tape from their drives.

Partial Format Option

If you performed the Full format option, skip these instructions and go to Step 6.

⚠ **CAUTION**

If you have performed this installation once, and want to repeat the Partial format option on a disk, read the instructions for saving current files before a Partial format. If you do not follow those instructions, repeating the Partial format option can overwrite files you want to save.

What Happens to Current Files When Installing Over a Previous Release

Some directories from previous releases of the OS are used in the iRMX OS. The current iRMX directories and files are renamed so that you may choose what to do with them. Any files you have created will not be deleted during an installation over a previous release.

To save the current iRMX files in those directories, the **submit** file renames directories as shown in Table 4-2. Following an installation over a previous release, current iRMX files reside in the **_old* directories specified in Table 4-2.

Table 4-2. Directories Renamed in a Multibus II Installation

Previous Directory/File	Renamed To
/user	/user_old
/bsl	/bsl_old
/msa32	/msa32_old
/boot32	/boot32_old
/rboot32	/rboot32_old
/net	/net_old
/system/rmx86	/system/rmx86_old
/system/rmx86.386	/system/rmx86_old.386
/msa/config/bps	/msa/config/bps_old
/rmx386/config/terminals	/rmx386/config/terminals_old
/rmx386/config/udf	/rmx/config/udf_old
/rmx386/config/user/super	/rmx386/config/user/super_old
/rmx386/config/user/world	/rmx386/config/user/world_old

Renaming Files With Existing _old Extensions

If your hard disk has any of the target files listed in the Renamed To column in Table 4-6, the **submit** file renames the current files from the Previous Directory/File column in Table 4-2 to have an extension of *_dd.#* where # is the lowest number that doesn't overwrite an existing file.

Performing a Partial Format

1. Using the device (hard disk) name you selected in Step 4, enter a **submit** command to format track 0 of the hard disk:

   ```
   submit :tape:form_trk0(device_name) <CR>
   ```

   For example, use this command for a Maxtor Model XT-8760S hard disk with a SCSI ID of 2:

   ```
   submit :tape:form_trk0(m8760_2) <CR>
   ```

   The **submit** file formats track 0 of the hard disk. It renames directories/files to preserve current files.

   After formatting, the **submit** file sets up a directory structure on the hard disk. It invokes the **restore** command and copies all the files from the tape to the hard disk. It establishes World permission rights for files. Then it switches the system device from the diskette drive to the hard disk, so that the OS reads future commands from the hard disk.

2. When you receive the logon prompt, remove the Start-up Commands diskette and OS tapes from their drives. Store them in their covers.

Step 5b. Formatting the Disk and Installing Software From CDROM

There are two options to format the hard disk: a Full format or a Partial format (track 0 only).

Full format	Formats the entire hard disk. Choose this option if you have a new Multibus II system with nothing on the hard disk. (All Intel microcomputers are shipped with unformatted hard disks.) You may also choose this option for a system whose software you want to completely erase.
Partial format	Formats only track 0 of the hard disk with the second stage of the iRMX bootstrap loader. Choose this option if your disk was formatted with any earlier release of the iRMX OS. This option does not delete any current files, but it does rename some of them.

Either option also installs the OS, along with any included support software.

Enter your commands after the hyphen (-) prompt.

Full Format Option

⚠ **CAUTION**

Do not perform this option if you want to keep the files on your hard disk. The Full format option deletes all files. Proceed to the Partial format option if you want to save existing files.

1. Insert the OS tape.
2. To format the hard disk, invoke a **submit** command with the proper parameters for the disk you want to format. Substitute a device name, the interleave factor, and the number of files:

```
submit :cdrom:form_disk(device, format-options) <CR>
```

Note

If you performed partitioning in Step 4, make sure you specify the appropriate partitioned DUIB name from Table 4-1.

This command is valid for a generic SCSI disk that has a SCSI ID of 2:

```
submit :cdrom:form_disk(gscw5_2)  <CR>
```

The OS formats the hard disk. Then it displays a summary of alternate tracks assigned for bad tracks found during the format.

```
The following tracks have been assigned an alternate:

     {disk information}
```

Following formatting, the **submit** file sets up a directory structure on the hard disk. It invokes the **restore** command and copies all the files from the tape to the hard disk. Enter `Yes` <CR> when switching tapes. It establishes World permission rights for files. Then it switches the system device from the diskette drive to the hard disk, so that the OS reads future commands from the hard disk. The process can take up to 45 minutes.

3. When you receive the logon prompt, remove the Start-up Commands diskette and OS CDROM from their drives.

Partial Format Option

If you performed the Full format option, skip these instructions and go to Step 6.

> **CAUTION**
> If you have performed this installation once, and want to repeat the Partial format option on a disk, read the instructions for saving current files before a Partial format. If you do not follow those instructions, repeating the Partial format option can overwrite files you want to save.

What Happens to Current Files When Installing Over a Previous Release

Some directories from previous releases of the OS are used in the iRMX OS. The current iRMX directories and files are renamed so that you may choose what to do with them. Any files you have created will not be deleted during an installation over a previous release.

To save the current iRMX files in those directories, the **submit** file renames directories as shown in Table 4-2. Following an installation over a previous release, current iRMX files reside in the *_*old* directories specified in Table 4-2.

Table 4-3. Directories Renamed in a Multibus II Installation

Previous Directory/File	Renamed To
/user	/user_old
/bsl	/bsl_old
/msa32	/msa32_old
/boot32	/boot32_old
/rboot32	/rboot32_old
/net	/net_old
/system/rmx86	/system/rmx86_old
/system/rmx86.386	/system/rmx86_old.386
/msa/config/bps	/msa/config/bps_old
/rmx386/config/terminals	/rmx386/config/terminals_old
/rmx386/config/udf	/rmx/config/udf_old
/rmx386/config/user/super	/rmx386/config/user/super_old
/rmx386/config/user/world	/rmx386/config/user/world_old

Renaming Files With Existing _old Extensions

If your hard disk has any of the target files listed in the Renamed To column in Table 4-6, the **submit** file renames the current files from the Previous Directory/File column in Table 4-2 to have an extension of *_dd.#* where # is the lowest number that doesn't overwrite an existing file.

Performing a Partial Format

1. Using the device (hard disk) name you selected in Step 4, enter a **submit** command to format track 0 of the hard disk:

   ```
   submit :cdrom:form_trk0(device_name) <CR>
   ```

   For example, use this command for a Maxtor Model XT-8760S hard disk with a SCSI ID of 2:

   ```
   submit :cdrom:form_trk0(m8760_2) <CR>
   ```

   The **submit** file formats track 0 of the hard disk. It renames directories/files to preserve current files.

   After formatting, the **submit** file sets up a directory structure on the hard disk. It invokes the **restore** command and copies all the files from the CDROM to the hard disk. It establishes World permission rights for files. Then it switches the system device from the diskette drive to the hard disk, so that the OS reads future commands from the hard disk.

2. When you receive the logon prompt, remove the Start-up Commands diskette and OS CDROM from their drives. Store them in their covers.

Step 6. Logging onto the System

Enter the Super logon and the default password (in lower case) as follows:

```
Logon: super <CR>
Password: passme <CR>
```

Step 7. Setting Up Your Terminal

1. Two sample versions of *:config:terminals* are supplied. Both are installed in the *:config:default* directory. The one you select depends on what device is used for the system console. If your system uses an SBX 279A board, select the *terminals.279* file. If your system uses a serial terminal, select the *terminals.arc* file. For example, to select the *terminals.arc* file, enter:

   ```
   copy :config:default/terminals.arc over :config:terminals <CR>
   ```

   Use the **skim** command to examine both of these files. In general, the physical device name `t279_x` is used to communicate with the SBX 279A driver and the physical device name `atcs_con_0` is used to communicate with the serial terminal.

   See also: Configuring Users and Terminals and ATCS Driver, *System Configuration and Administration*

2. Select the *terminals* file for your system. The command described below selects terminal attributes, so you can use your terminal with the Aedit text editor and the CLI. You should put this command in your *:prog:r?logon* file. Until you do so, you must enter the command each time you log on.

   Select the appropriate terminal name from Table 4-4.

Table 4-4. Terminal Names Defined in the *termcap* File

Terminal Type	Description
ANY	Default ANSI terminal for terminals not listed
1510E	Hazeltine 1510 with escape lead-in
1510T	Hazeltine 1510 with tilde lead-in
ADM3A	Lear Seigler ADM-3A
AT386	PC system console
RMXPC or PC	PC system console
QVT102	Qume QVT102, in QVT102 mode
RGI	SBX 279 Graphic Subsystem
S120	PC system console
TV910P	Televideo 910 Plus
TV950	Televideo 950
VT100	DEC VT100, VT101 (also for Wyse 75 and Wyse 85)
VT102	DEC VT102
VT52	DEC VT52
WYSE50 or WY50	Wyse 30, Wyse 50
XTERM	X-Windows terminal
ZENTEC	Zentec Zephyr and Cobra

Enter this command using the terminal name:

```
- set terminal = terminal_name <CR>
```

Step 8. Modifying BPS Files for Non-default I/O Server Boards and Devices

If your I/O server board in slot 0 is an SBC 486/133SE, P5090ISE, P5120, or P5090; or your system device is anything other than GSCW5_2 you need to perform some modifications to the BPS files. If none of these conditions apply to you, skip this step and proceed to Step 9.

Modifications for 486/133SE Board

If your I/O server board in slot 0 or 1 is a 486/133SE, edit the file */msa/config/bps*.

1. Invoke the Aedit editor:

```
aedit /msa/config/bps
```

2. In the `bnam` parameters of `rq_pci_a` and `rq_atcs_con`, replace the string `486/166SE` with `486/133SE`.

3. Quit the Aedit editor while saving the file using these commands:

```
<ESC>
q
e
```

See also: Boot Scenarios, SPS Parameters, *MSA for the iRMX Operating System*

Modifications for SBC P5090ISE, P5150, and P5090 Boards

If your I/O server board in slot 0 or 1 is an SBC P5090ISE, P5120, or P5090, edit the file */msa/config/bps*.

1. Invoke the Aedit editor:

```
aedit /msa/config/bps
```

2. In the `bnam` parameters of `rq_pci_a` and `rq_atcs_con`, replace the string `P5120ISE` with the appropriate string: `P5090ISE, SBCP5120, or SBCP5090`.

3. Quit the Aedit editor while saving the file using these commands:

```
<ESC>
q
e
```

See also: Boot Scenarios, SPS Parameters, *MSA for the iRMX Operating System*

Modifications for Hard Drive Type

If your system device is anything other than GSCW5_2 (generic SCSI with a 512-byte granularity) replace the string GSCW5_2 with the device name. For example, if you have a SCSI drive with 1024-byte granularity, replace the string GSCW5_n with GSCW_n.

Step 9. Updating Your Software

For information on updating your iRMX software, please refer to the Release Notes.

Updating the software and generating a new version may cause the size of the OS to exceed the amount of memory allocated for it in the ICU definition file. If you get an error message, you will need to increase the amount of memory.

See also: Changing the Memory Size, Appendix E

Step 10. Generating a Default Operating System for the I/O Server Board

In this step, you make a new version of the OS using the **mksys** command. This version will independently boot the I/O Server board in your Multibus system, so it will act also as a processor board.

⚠ **CAUTION**

For any one definition file, do not invoke the **mksys** command again after you have successfully used it once to generate the OS. Because **mksys** does not use your modified file for input, it will overwrite the local copy of your definition file. You will lose any modifications that you have made to the file. Instead of using **mksys**, invoke the ICU using the definition file as the input parameter.

1. Table 4-5 shows the files used and generated by the **mksys** command for different processor boards. It also shows the board name to use for your processor board in these commands.

   Find the board name that corresponds to the processor board in your Multibus system. For instance, the SBC 386/258 board has the board name 386258.

   See also: Multibus II Standard Definition Files, *ICU User's Guide and Quick Reference*

Table 4-5. Files Generated by mksys for I/O Server Boards as CPU Boards

I/O Server Board	Board Name	ICU Definition File	Generation Directory	Boot File Pathname	OS Name
SBC 386/258	386258	258s.bck	:home:258s	/msa32/boot/258s	258s
SBC 386/258D	386258	258s.bck	:home:258s	/msa32/boot/258s	258s
SBC 486/133SE	486133	433s.bck	:home:433s	/msa32/boot/433s	433s
SBC 486/166SE	486133	433s.bck	:home:433s	/msa32/boot/433s	433s
SBC P5120ISE	P5120ISE	p90s.bck	:home:p90s	/msa32/boot/p90s	p90s
SBC P5090ISE	P5090ISE	p90s.bck	:home:p90s	/msa32/boot/p90s	p90s
SBC P5150	SBCP5150	p90s.bck	:home:p90s	/msa32/boot/p90s	p90s
SBC P5090	SBCP5090	p90s.bck	:home:p90s	/msa32/boot/p90s	p90s
SBC P5200	SBCP5200	P200s.bck	:home:p200s	/msa32/boot/p200s	P200s

2. This command may take as long as 45 minutes to execute. Sometimes the screen scrolls faster than you can read it. This is because the **mksys** command rapidly shows the ICU screens as it executes. The scrolling is not a cause for concern.

   Enter this command substituting the appropriate OS name for your processor board:

   ```
   - mksys os_name <CR>
   ```

   Various ICU prompts are displayed on the screen. These prompts are answered by a **submit** file. Do nothing until you see an END **submit** message.

   The **mksys** command creates a directory under your *:home:* directory. It attaches that new directory as your default prefix. It creates a local copy of the appropriate ICU definition file. Then it generates on disk an updated version of the OS suited to the processor board appropriate for the specified OS in the command.

   The command ends with a line similar to this:

   ```
   -END submit :config:cmd/mksys.CSD
   ```

3. Following the END **submit** message, use either the Aedit editor or the following **skim** command to discover if errors occurred in the process.

   While the **mksys** command was running, it recorded its progress in an output file. Use the **skim** command to read this output file to find out if any errors occurred during the generation process. Replace `os_name` with the same name you used in the **mksys** command:

```
- skim os_name.out <CR>
```

Press the space bar at the `more?` prompt to page through the file.

If the *os_name.out* file contains any error messages, some problem occurred during the **mksys** generation. Use an example file to identify the errors.

See also: Generation output file, *ICU User's Guide and Quick Reference*

If you find this error message, use the ICU to change the amount of memory reserved for the iRMX OS, as described in Appendix E.

```
ERROR 192:  NO SPACE FOR SEGMENT - BASE NOT SET
```

See also: Changing memory size, Appendix E

If you find a different error, call Intel Customer Support.

See also: Boot Scenarios, *MSA for the iRMX Operating System*

Step 11. Testing by Booting from the Hard Disk

To test the previous installation, you must shut down the OS, then reset and reboot the Multibus system.

1. Shut down the OS with the **sh** command (an alias for **shutdown w=0**):

   ```
   - sh <CR>
   ```

2. When you see a `Shutdown Completed` message, reset the system. Turn the key switch on the front panel of the computer to Reset and then back to Unlock.

3. When the first character appears on the screen, type a u. The power-up tests will run again.

   ```
   u
   ```

4. The system prompts you for a selection from this menu. Select `Go to Operator Interface` by typing 2 at the Enter number ? prompt:

   ```
   1 Run system Diagnostics
   2 Go to Operator Interface
   3 Go to Bootphase

   Enter number ?  2
   ```

   The 2 will not show on the screen. If you do not type a choice in 9 seconds, the system defaults to running the diagnostics (selection 1). The diagnostics can be interrupted by pressing <Ctrl-C>.

5. Enter the **mp** command to change a bootstrap parameter:

```
MTH [0] mp <CR>
```

6. At the following display, enter a new parameter:

```
Modify Boot Parameters for slot   0:
Modify parameters: "#" deletes, "." quits, <CR> advances
new parameter

bl_target_file = /msa32/boot/os_name <CR>
```

7. Press the <CR> key in response to each of the next two prompts:

```
new parameter
      <CR>
save changes ([y] or n)
      <CR>
```

8. Enter the **b** (Bphase) command. This tells the MTH to invoke the boot phase:

```
MTH [0] b <CR>
```

The OS boots on the I/O Server board.

9. Enter the super logon and the default password:

```
[0] Logon: super <CR>
Password: passme <CR>
```

See also: Boot Scenarios, *MSA for the iRMX Operating System*

Step 12. Copying the New Version Over the Start-up Version

Assuming the new OS is running, use this procedure to copy the new bootable image over the image first used to boot the system. If the OS is not running, call your Intel Customer Support representative.

This step allows you to choose option 3 of the MTH menu to automatically boot the OS, rather than setting boot parameters each time you start the Multibus system.

Use the **copy** command to copy over the *rmx* file:

```
- copy /msa32/boot/os_name over /msa32/boot/rmx <CR>
```

Use an appropriate value for `os_name` from the right-most column of Table 4-5.

Step 13. Taking Precautions for System Security

Use this step to protect the security of your OS. By securing the OS, unauthorized people cannot log on as the system manager and make changes to the OS. To prevent unauthorized OS changes, the system manager should implement password protection.

A password can be up to eight characters long. It may consist of any printable ASCII characters, and is case-sensitive.

1. Change the system manager password using the **password** command:

   ```
   - password <CR>
   ```

2. You are prompted with a series of commands. Enter `c` to change the password:

   ```
   Enter the command:  c <CR>
   ```

3. You receive these prompts. Respond to each. The password entries that you make will not be shown on the screen:

   ```
   Enter your user name - super <CR>
   Enter the old password - passme <CR>
   Enter the new password - new_password <CR>
   Repeat the new password - new_password <CR>
   ```

4. To exit, enter `e` as follows:

   ```
   Enter the command:  e <CR>
   ```

There is another logon called World that does not have the powers of the system manager. The World default password is <CR> (press <CR> at the password prompt). Do not change the password for World. If you want to remove the ability of the user World to access your system, delete the file *:config:user/world*, and use the **password** command to remove the World user from *:config:udf*.

Do not routinely log on as the system manager. Instead, log on either as World or as some other user that you define.

See also: World user, Super user, *Command Reference*

Log on as Super only when system files need manipulation. This reduces the chance of deleting or modifying a system file by mistake.

See also: Logging On and Off, Chapter 9

When you want to log off as Super, issue the **logoff** command. To become Super while logged on as another user, enter the command **super**, then the Super password. Then to get back to the ordinary user logon, enter the **exit** command.

Step 14. Merging the User and Configuration Files

✏ **Notes**

Perform this step only if you have installed over a previous version of the OS, using the Partial Format option. If you formatted the whole disk, the installation is finished.

If you are configuring multiple processor boards, refer to *MSA for the iRMX Operating System* for additional instructions.

⚠ **CAUTION**

The default *r?logon* file submits file *:config:alias.csd* in addition to *:prog:alias.csd*. The file *:config:alias.csd* contains aliases used by the OS. Installation instructions and example code use these aliases as defined. Do not modify the existing aliases in *:config:alias.csd*; adding new aliases is acceptable.

User and Alias Files

The **submit** file for the installation renamed the existing */user* directory to */user_old*.

1. Invoke these commands to restore user files to their original directory structure as well as put the boot system in Super's *:home:* directory. Replace `os_name` in the last command below with the same OS name used in previous commands.

```
- rename /user to /user_new <CR>
- rename /user_old to /user <CR>
- rename /user_new/super/os_name
    to /user/super/os_name <CR>
```

You can now delete the */user_new* directory

2. The structure of the *:prog:r?logon* file changed slightly from previous iRMX II releases (up through release 3.0). In addition to submitting *:prog:alias.csd* as in the past, the default *r?logon* file for each user now also submits *:config:alias.csd*.

   To allow users access to the aliases in *:config:alias.csd*, add this line to the *:prog:r?logon* file for each user:

   ```
   submit :config:alias
   ```

3. Some of the aliases in *:config:alias.csd* are the same as aliases defined in the old *:prog:alias.csd* file. An alias defined in *:config:alias.csd* should not also be defined in *:prog:alias.csd*.

   From this list, delete duplicate aliases in file *:prog:alias.csd*. The 32-bit languages are invoked using the **run86** utility.

   ```
   alias ad=attachdevice
   alias af=attachfile
   alias crdir=createdir
   alias dd=detachdevice
   alias df=detachfile
   alias flim=floppyimage
   alias install=submit :config:cmd/instal(#0)
   alias tinstall=submit :config:cmd/tinstall(#0)
   alias mksys=submit :config:cmd/mksys(#0)
   alias plm386=run86 -fixplm :lang:plm386
   alias asm386=run86 :lang:asm386
   alias bnd386=run86 :lang:bnd386
   alias bld386=run86 :lang:bld386
   alias lib386=run86 :lang:lib386
   alias map386=run86 :lang:map386
   alias ic386=run86  :lang:ic386
   ```

Alias Additions

4. Use the Aedit text editor to add new user aliases to your *:prog:alias.csd* file. These are defined in the file *:config:default/alias.csd*.

   ```
   alias s=submit
   alias m=skim
   alias trv=traverse
   ```

5. Aliases specific to Multibus II systems are in the *:config:alias.csd* and *:prog:alias.csd* files. Use the Aedit text editor to add new aliases to this file.

   ```
       alias agents = ic -c agents
       alias agentreset = ic -c reset #0 local
       alias armfpi = ic -c fpi arm
       alias coldreset = ic -c reset 0 cold
       alias disarmfpi = ic -c fpi disarm
       alias icread = ic -c get #0 #1 #2
       alias icwrite = ic -c set #0 #1 #2
       alias monitor = ic -c reset -p monitor #0 local
       alias myslot = ic -c myslot
       alias nmi = ic -c nmi #0 software
       alias nmiforce = ic -c nmi -e #0 software
       alias offline = ic -c kill #0
       alias reboot = ic -c reset -p bootstrap #0 local
       alias warmreset = ic -c reset 0 warm
   ```

Configuration Files

These instructions help you decide how to merge your old iRMX II or iRMX III configuration files with the newly installed files in the *:config:* directory. After you have merged these files to your satisfaction, you may remove the old versions of the files.

The new system configuration files in the *:config:* directory are compatible with previous releases of the OS. This release can boot and use the old versions with no problems. For some of the files, however, if you simply overwrite the new version with your previous version, some important enhancements are lost.

6. The *:config:termcap* file is the iRMX HI CLI terminal attribute file. The information in this file provides terminal attribute definitions for the CLI and for Aedit. Issuing the CLI's **set** command allows Aedit to operate without creating a specific *aedit.mac* file, as was necessary in the past.

   Compare the new *termcap* file with changes you made to your previous version. Decide which file provides a better starting point for merging your changes; adding your changes to the new file may be easier. If *:lang:aedit.mac* or

:home:aedit.mac exist, they need to contain only utility macros, not terminal attributes. Also, *:lang:useful:mac* contains some valuable macros. If you are using the SBX 279A board, try the file */rmx386/rgi/rgi.mac*.

7. The *:config:terminals* file is the HI user terminal configuration file. The format of this file allows a Network Server system to provide remote system device services to a diskless remote file consumer. The *terminals* file allows multiple remote consumers to use one copy of the file for each of their unique terminal definitions.

   Read the description of *:config:terminals* in the *System Configuration and Administration* manual for more information. The original form of the file should work fine for the local system. To restore the original iRMX II version, enter:

   ```
   - copy /rmx286/config/terminals over /rmx386/config <CR>
   ```

   To restore the old iRMX III version, enter:

   ```
   - copy /rmx386/config/terminals_old over /rmx386/config/terminals
   <CR>
   ```

   The physical device names of terminals controlled by the Multibus II ATCS driver have been expanded.

   See also: ATCS Driver in *System Configuration and Administration* for the list of new device names

8. The *:config:udf* file is the user definition file. No changes have been made to this file's format.

   If you defined users in a previous version of the iRMX II UDF file, restore it with the command:

   ```
   - copy /rmx286/config/udf over /rmx386/config <CR>
   ```

   For a previous version of the iRMX III UDF file:

   ```
   - copy /rmx386/config/udf_old over /rmx386/config/udf <CR>
   ```

9. The *:config:cdf* file is the Administration Unit definition file for network consumer systems. No changes have been made to this file's format.

   If you defined any networking Administration Units (subnets) in a previous version of the iRMX II CDF file, restore it with the command:

   ```
   - copy /rmx286/config/cdf over /rmx386/config <CR>
   ```

   For a previous version of the iRMX III CDF file:

   ```
   - copy /rmx386/config/cdf_old over /rmx386/config/cdf <CR>
   ```

10. All users that you created in an iRMX II system are still valid in your iRMX III system. The process of adding and configuring users has not changed. To add all of your iRMX II users to your iRMX III system, enter this command:

    ```
    - copy /rmx286/config/user/* to /rmx386/config/user <CR>
    ```

    For a previous version of the iRMX III users, enter:

    ```
    - copy /rmx386/config/user/*_old to /rmx386/config/user/* <CR>
    ```

11. Modify the *:config:r?init* file. The *:config:r?init* file is the iRMX System Initialization file. This submit file is executed when the HI is initialized. Its output goes to the first entry in *:config:terminals*. No CLI commands, such as **background** or **alias**, can be used, because no CLI job is created when the file is executed.

    A default version of the file is provided. It contains a description of how to set it up to load one of the SBX 279(A) menus. To select the default version, enter:

    ```
        copy :config:default/rinit over :config:r'?'init
    ```

    Modify the file as needed.

12. Modify the *logon.msg* file as needed. The *:config:logon.msg* file is the logon message file.

Where to Go Next

If you want to turn off the Multibus system, first issue the **sh** command (as Super). You may leave the system running, but you should first log off as Super by issuing the **logoff** command.

To learn more about your new OS, refer to these manuals:

Topic	**Manual**
Basic operations, such as logging on, entering commands, etc.	Chapter 9 of this manual
Features of the iRMX Operating Systems	*Introducing the iRMX Operating Systems*

➨ **Note**

Chapter 9 describes the installation process for online iRMX manuals and the Windows Help file for the iRMX system calls supplied with this release. Installation of both these features is performed separately from the installation procedure in this chapter.

□ □ □

Installing the iRMX III OS on Multibus I Systems 5

Use this procedure if the platform on which you are installing is a Multibus I board that cannot run as a PC.

Use the instructions in this chapter to make a first-time installation of the OS, or to install over a previous version. During the installation you will:

- Run the board diagnostics.
- Bootstrap load the system from a diskette.
- Log on as the system manager.
- Format the computer's hard disk, either partially or completely.
- Install the OS software and support products.
- Reboot the Multibus system.
- Generate a boot system.
- Establish the default boot system.
- Merge user files with the new software if you installed over a previous version of the OS.

Minimum System Requirements

The installation procedure in this chapter assumes you have an Intel System 310, System 310AP, System 320, or compatible Multibus system.

You need a minimum of 75 Mbytes of space on your hard disk to install the iRMX III software. You need a minimum of 4 Mbytes of RAM to boot the iRMX III OS.

Multibus I Boards

The iRMX III OS supports these Multibus I boards:

CPU Boards	**Controller Boards**
SBC 386/12(S)	SBC 214
SBC 486/12(S)	SBC 221(S)
SBC 386/12S	SCSI controller on SBC 386/12S and 486/12S boards
SBC 486/12S	SCSI controller on SBC 386/12S and 486/12S boards
SBC 386/3x	

Preparing for the Installation

1. If you are not installing the OS on a new Multibus system, back up all current files on tape, especially before formatting the Multibus system's hard disk. In a later step you will format the hard disk. At that point you choose whether to do a complete or partial format. Choosing the partial format allows you to retain all current files on your disk.

2. Write down the option code of the hard disk in your Multibus system if you are using an Intel Multibus system. You can find the option code information on documents delivered with the hardware, or on a label on the bottom cover of the Multibus system. If there are two hard disks, choose the option code of the disk on which you want to make the installation. If you are building a custom system, the supported device types are listed in Tables 5-3 through 5-5.

3. Determine the type of the processor board and peripheral controller board in your Multibus system. If you are using an Intel Multibus system and do not know what type of boards you have, refer to the label on the bottom of the cover of the Multibus system.

4. Read your Multibus system's hardware installation manual to be sure that the equipment is set up correctly and that the terminal is connected with the correct cable.

5. See Appendix F, Hardware Modifications, to determine if you need to make any changes to your hardware so that it can run the iRMX III OS.
6. Gather the materials you need for the product(s) you will install. For low-density boot diskettes, you need separate boot and commands diskettes. For high-density diskettes, the commands are included on the boot diskette.
7. Choose which diskette(s) your system requires from this list, based on the processor board in your Multibus system:

   - SBC 386/12 and 486/12 ST506 and ESDI Start-up Boot Diskette, plus the iRMX III Multibus I Start-up Commands Diskette (low-density diskettes)
   - SBC 386/2x/3x ST506 and ESDI Start-up Boot Diskette, plus the iRMX III Multibus I Start-up Commands Diskette (low-density diskettes)
   - SBC 386/12S and 486/12S SCSI Start-up Boot Diskette (high-density diskette that includes the Multibus I Start-up Commands)

   You will also need these tapes or diskettes:

   - OS tapes
   - Tapes or diskettes for products to be installed that are not included in the iRMX III OS software package

   If you are missing an item for a product you have purchased, contact your sales representative.
8. Check that all the diskettes and tapes are write-protected.

The installation process described in this chapter can take two hours or more, depending on the type of disk format and the amount of software being installed.

Step 1. Running the Board Diagnostics

Multibus I processor boards use System Confidence Test (SCT) board diagnostics. The SCT is supplied in firmware as part of an Intel Multibus system. It tests all parts of the system.

Running the System Confidence Test

1. Turn on the power for the system console and the Multibus system. In about five seconds, the system console displays a series of x's, which is the prompt.
2. Within ten seconds after you see the prompt, type the uppercase letter U repeatedly, until the diagnostics begin to execute (the U does not show on the screen).
3. If you see this prompt, enter n <CR> (after ten seconds, n is assumed):

   ```
   Break to DMON-386 monitor (y or [n]) ?  n <CR>
   ```

   If you see this prompt, enter y <CR> (after ten seconds, automatic booting begins):

   ```
   Exit to iSDM after testing? Enter y or n [n]  y <CR>
   ```

   The diagnostics execute tests and display a test status report on the system console. When the tests are complete, the SDM monitor is invoked. It displays an entry address (xxxx:yyyy) and a dot prompt (.) as shown:

   ```
   Interrupt 3 at <xxxx:yyyy>
   .
   ```

See also: The manual for your Multibus system for information about the test results

Step 2. Booting from the Start-up Boot Diskette

Two procedures are described here, one for a low-density diskette and one for a high-density diskette.

To Boot from a Low-Density Diskette

1. Insert the boot diskette for your processor board.
2. At the dot prompt, enter this to boot from a low-density diskette controlled by the SBC 214 or SBC 221(S), controller board (where WF0 ends in a 0, not the letter O):

   ```
   .b :wf0: <CR>
   ```

   When bootstrap loading is complete, the system console displays this message:

   ```
   Insert the Start-up Commands Diskette and type G <RETURN>
   Interrupt 3 at <xxxx:yyyy>
   .
   ```

3. Remove the boot diskette.
4. Insert the commands diskette for your system.
5. Enter g (the SDM monitor's **go** command):

   ```
   .g <CR>
   ```

To Boot from a High-Density Diskette

Table 5-1 lists the device names for supported diskettes and controllers. Find the appropriate device name for your system.

Table 5-1. BSL Device Name, Diskette, and Controller Information

BSL Device Name	Granularity	Density	Controller
wf0	iRMX standard	low	SBC 214, 221(S), and 215G
sqf0	uniform	high	SCSI to floppy adapter
t55_0	uniform	high	SCSI to embedded SCSI Teac 5.25-inch diskette
t235_0	uniform	high	SCSI to embedded SCSI Teac 3.5-inch diskette

1. Insert the boot diskette for your processor board.
2. At the dot prompt, boot from a high-density diskette (controlled by an on-board SCSI interface) by entering:

   ```
   .b :sqf0: <CR>
   ```

   or

   ```
   .b :t55_0: <CR>
   ```

   or

   ```
   .b :t235_0: <CR>
   ```

   To boot from a diskette controlled by an SBC 221(S) or SBC 214, enter:

   ```
   .b :wf0: <CR>
   ```

Step 3. Logging On and Setting the Date and Time

The console displays this bannerhead:

```
*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*

            iRMX* Ver.# Operating System

* iRMX is a Registered Trademark of Intel Corporation

*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*

 Logon:
```

You now log on as the system manager, Super. The system manager has access to all files used during the installation process, and can change access rights to all iRMX files.

1. Enter the Super logon at the prompt:

   ```
   Logon: super <CR>
   ```

2. The system prompts you for a password. For now, enter the default password for Super, passme. The password must be in lower-case letters. It is not shown on the screen.

   ```
   Password: passme <CR>
   ```

   The OS signs on with this message:

```
 iRMX HI CLI, Vx.y: USER = 0
 Copyright years Intel Corporation
 All rights reserved
 -
```

3. When you see a prompt for the date, enter the date in one of these formats. If the displayed date is already correct, enter `e` instead.

month/day/year	(for example, `01/15/1999`)
day month year	(for example, `15 JAN 1999`)
day month year	(for example, `15 JANUARY 1999`)

   Example:

   ```
   DATE:  01/15/1999 <CR>
   ```

   If your system does not include a global clock, you will see the message:

   ```
   E$EXIST, Global Clock not present - will query for
   local date
   ```

4. The system prompts you for the time, which is based on a 24-hour clock. Enter the time in hours:minutes:seconds format, using either one or two digits for each field. If you omit the minutes and seconds, the system sets them to 0. If the displayed time is already correct, enter `e` instead.

   For example, at 7:36 AM and 40 seconds, enter `07:36:40` or at 2:00 PM, enter `14:00`.

   Example:

   ```
   TIME:  7:36:40 <CR>
   ```

   If you enter either the date or the time in the wrong format, you get a message to re-enter it.

   If your system does not include a global clock, you will see the message:

```
E$EXIST, Global Clock not present - will query for local time
```

Two **submit** files named *:prog:alias.csd* and *:config:alias.csd* are executed.

These files define aliases for command names. To see the complete list of aliases, use the **alias** command. Do not modify the aliases in the *:config:alias.csd* file because the iRMX software uses them as defined.

The OS defines the logical name *:tape:*. In these steps, the command that formats the hard disk requires this logical name.

Step 4. Determining Disk Parameters

The **submit** commands in these steps require parameters specific to the hardware in your system. In this step, you determine the parameters based on your peripheral controller board and hard disk. For an installation over a previous release, you need only the device name parameter, described in Item 2.

1. If you are not installing over a previous release, use Table 5-2 to determine the parameters for your disk drive. You will need the device name, number of files, and interleave factor.

Table 5-2. Interleave Factors

Controller Board	Interleave Factor for 5.25- and 3.5-inch Hard Drives	Interleave Factor for 8-inch Hard Drives
SBC 214 or 221(S)[1]	1	--
SBC 215G	4	5
SCSI	1	--

[1] Device names for the first hard disk (or for systems with only one hard disk) end in 0. Device names for a second hard disk end in 1.

2. Refer to Tables 5-3 through 5-5 to locate the type of hard disk on which you will install the software. Table 5-4 applies only if you are using a 221S controller board.

   When you locate the type of hard disk, make a note of the device names and number of files for that disk type. If your system includes multiple hard disks, choose the appropriate device name based on which disk you will use.

Note

The `files` parameter used in the submit commands is the total number of files that can be created on the hard disk. If the number of files needs to be changed later, you will have to back up your files and reformat the hard disk. The recommended numbers listed in Tables 5-3 through 5-5 are approximations based on 125 files per megabyte of disk capacity. You may choose a different number. The maximum number of files that the OS can handle is 65,528.

Table 5-3. Device Names and Numbers of Files for ST506 and ESDI Hard Disks

Hard Disk Model Number	System 310 Designation	System 310AP Option Code	System 320 Option Code	Device Name	Recommended Number of Files
Maxtor XT-1140	140 Mbyte 5.25-inch	GK3	T1GF4 T1GF9	mma0 mma1	18000
Maxtor XT-1085	86 Mbyte 5.25-inch	GK8	none none	mmb0 mmb1	11000
Maxtor XT-4175E	none none	none none	T1GF10 T1GF12	mmc0 mmc1	22000
Maxtor XT-4380E	none none	none none	T1GF11 T1GF13	mmd0 mmd1	48000
Maxtor XT8760E	none none	none none	T1GF16 T1GF17	mme0 mme1	65528
Seagate ST-251	43 Mbyte 5.25-inch	GK12	none	sma0 sma1	5400
Toshiba MK56FB	85 Mbyte 5.25-inch	none T1GF8	T1GF3 tma1	tma0	11000

Table 5-4. Device Names and Numbers of Files for Generic SCSI Hard Disks on an SBC 221S Controller

Generic SCSI Size	Device Name	SCSI ID Number (*n*)	Recommended Number of Files
500 Mbyte or larger with 512-byte granularity	GW500M_*n*	0, 1	62500
700 Mbyte or larger with 512-byte granularity	GW700M_*n*	0, 1	65528
1 Gbyte or larger	GW1G_*n*	0, 1	65528
2 Gbyte or larger	GW2G_*n*	0, 1	65528
4 Gbyte or larger	GW4G_*n*	0, 1	65528

Table 5-5. Device Names and Numbers of Files for SCSI Hard Disks (386/12S and 486/12S only)

Hard Disk Model Number	Device Name	SCSI ID Number (*n*)	Recommended Number of Files
Maxtor XT-3170S	M3170_*n*	2, 3	22000
Maxtor XT-4170S	M4170_*n*	2, 3	22000
Maxtor XT-4380S	M4380_*n*	2, 3	48000
Maxtor XT-8380S	M8380_*n*	2, 3	48000
Maxtor XT-8760S	M8760_*n*	2, 3	65528
Hewlett-Packard 97536	HP97536_*n*	2, 3	40000
Siemens Megafile 2300	S2300_*n*	2, 3	48000
Micropolis 1578	MP1578_*n*	2, 3	39000
Quantum Pro 80	Q280_*n*	2, 3	11000
Generic SCSI 512 byte granularity	GSCW5_*n*	2, 3	depends on drive size
Generic SCSI 1024 byte granularity	GSCW_*n*	2, 3	depends on drive size

3. Replace the *n* in the device name in the second column in Tables 5-4 and 5-5 with the appropriate SCSI ID Number from the third column. For example, the device name for the Maxtor XT-4380S drive with SCSI ID 2 is M4380_2. Use the Generic SCSI DUIBs for hard disks not listed in the table.

   See also: Your hard disk drive reference manual for the appropriate SCSI ID number;
   Appendix F, *Command Reference* for information on partitioned SCSI hard disk drives

Step 5. Formatting the Disk and Installing Software From Tape

You must now format the hard disk. There are two options, a full format or a partial format (track 0 only).

Full format Formats the entire hard disk.

Choose this option if you have a new Multibus system with nothing on the hard disk. All Intel Multibus systems are shipped with unformatted hard disks. You may also choose this option for a system whose software you want to completely erase.

Partial format Formats only track 0 of the hard disk with the second stage of the iRMX bootstrap loader.

Choose this option if your disk was formatted with any earlier release of the iRMX OS. This option does not delete any current files, but it does rename some of them.

Either option also installs the OS, along with any included support software. Enter your commands after the hyphen (-) prompt.

Full Format Option

⚠ **CAUTION**

Do not perform this option if you want to keep the files on your hard disk. The Full format option deletes all files. Proceed to the Partial format option if you want to save existing files.

1. Insert the first OS tape.
2. To format the hard disk, enter a **submit** command with the proper parameters for the disk you want to format. Substitute a device name, the interleave factor, and the number of files appropriate for your system:

   ```
   submit :tape:form_disk(device, interleave, files) <CR>
   ```

   This command is valid for a Maxtor XT-4380E disk that is the first hard disk in the system, and is controlled by an SBC 221 board:

   ```
   submit :tape:form_disk(mmd0, 1, 48000)  <CR>
   ```

The OS formats the hard disk, which can take 30 minutes or longer. Then it displays a summary of alternate tracks assigned for bad tracks found during the format.

```
The following tracks have been assigned an alternate:

     {disk information}
```

Following formatting, the **submit** file sets up a directory structure on the hard disk. It invokes the **restore** command and copies all the files from the tape to the hard disk. It establishes World permission rights for files. Then it switches the system device from the diskette drive to the hard disk, so that the OS reads future commands from the hard disk. This process can take up to 45 minutes.

3. When you receive the logon prompt, remove the *Start-up Commands* diskette and OS tape from their drives. Store them in their covers.

Partial Format Option

If you performed the Full format option, skip these instructions and go to Step 6.

⚠ **CAUTION**

If you have performed this installation once, and want to repeat the partial format option on a disk, read the instructions for saving current files before a partial format. If you do not follow those instructions, repeating the partial format option can overwrite files you want to save.

What Happens to Current Files When Installing Over a Previous Release

The iRMX III OS uses some directories from previous releases. The current iRMX directories and files are renamed so that you may choose what to do with them. Files you have created will not be deleted when installing over a previous release.

To save the current iRMX files in those directories, the **submit** file renames directories as shown in Table 5-6. Following an installation over the previous release, current iRMX files reside in the **_old* directories/files specified in Table 5-6.

Table 5-6. Directories Renamed in a Multibus I Installation

Previous Directory/File	Renamed To
/user	/user_old
/msa32	/msa32_old
/boot32	/boot32_old
/rboot32	/rboot32_old
/net	/net_old
/system/rmx86	/system/rmx86_old
/system/rmx86.386	/system/rmx86_old.386
/msa/config/bps	/msa/config/bps_old
/rmx386/config/terminals	/rmx386/config/terminals_old
/rmx386/config/udf	/rmx/config/udf_old
/rmx386/config/user/super	/rmx386/config/user/super_old
/rmx386/config/user/world	/rmx386/config/user/world_old

Renaming Files With Existing _old Extensions

If your hard disk has any of the target files listed in the Renamed To column in Table 5-6, the **submit** file renames the current files from the Previous Directory/File column in Table 5-6 to have an extension of _dd.# where # is the lowest number that doesn't overwrite an existing file.

Performing a Partial Format

1. Using the device name you selected in Step 4, enter this **submit** command to format track 0 of the hard disk:

   ```
   submit :tape:form_trk0(device_name) <CR>
   ```

   Use this command for a Maxtor Model XT-1140 hard disk:

   ```
   submit :tape:form_trk0(mma0) <CR>
   ```

   The **submit** file formats track 0 of the hard disk. It renames directories and files to preserve current files.

   After formatting, the **submit** file creates a directory structure on the hard disk. It invokes the **restore** command and copies all files from the tapes to the hard disk. Enter `Yes` <CR> after switching tapes. It establishes World permission rights for files and then it switches the system device from the diskette drive to the hard disk (so that the OS reads future commands from the hard disk).

2. When the logon prompt appears, remove the Start-up Commands Diskette and OS tapes from their drives.

Step 6. Logging On and Copying the Start-up Boot System

1. Enter the Super logon and the default password (in lower case) as:

```
Logon: super <CR>
Password: passme <CR>
```

2. When prompted for the date and time, enter the letter `e`:

```
DATE:  e <CR>
TIME:  e <CR>
```

The start-up boot system is now copied from the diskette to the hard disk. The Multibus system can now be rebooted from the hard disk if a failure occurs.

Step 7. Setting Up Your Terminal

The command described below selects terminal attributes, so you can use your terminal with the Aedit text editor and the Command Line Interpreter (CLI). Put this command in your *:prog:r?logon* file. Until you do so, you must enter the command each time you log on.

1. Select the appropriate terminal name from Table 5-7.

Table 5-7. Terminal Names Defined in the *termcap* File

Terminal Type	Description
ANY	Default ANSI terminal for terminals not listed
1510E	Hazeltine 1510 with escape lead-in
1510T	Hazeltine 1510 with tilde lead-in
ADM3A	Lear Seigler ADM-3A
AT386	PC system console
RMXPC or PC	PC system console
QVT102	Qume QVT102, in QVT102 mode
RGI	SBX 279 Graphic Subsystem
S120	PC system console
TV910P	Televideo 910 Plus
TV950	Televideo 950
VT100	DEC VT100, VT101 (also for Wyse 75 and Wyse 85)
VT102	DEC VT102
VT52	DEC VT52
WYSE50 or WY50	Wyse 30, Wyse 50
XTERM	X-Windows terminal
ZENTEC	Zentec Zephyr and Cobra

2. Enter this command using this terminal name:

```
- set terminal = terminal_name <CR>
```

Step 8. Updating Your Software

For information on updating your iRMX III software, please see the Release Notes.

Updating the software and generating a new version could cause the size of the OS to exceed the amount of memory allocated for it in the ICU definition file. If you get an error message to this effect, you must increase the amount of memory available to the OS.

See also: Changing the Memory Size, Appendix E

Step 9. Generating a Default OS

In this step, you make a new version of the OS using the **mksys** command. This version will independently boot the processor board in your Multibus system.

A. Additional Step for Non-SCSI Boards

➾ **Note**

Perform part A of this step only if you are running a non-SCSI board. Otherwise, skip to part B.

The installed system comes with the following iRMX OSs pre-generated:

- /boot32/38612.386
- /boot32/38620.386

Before you generate your new version of the operating system, you will need to reboot the system from one of the pre-generated operating systems. Use the following list to see which OS to reboot with:

CPU Board	Boot File Pathname
SBC 386/12	/boot32/38612
SBC 386/12S	/boot32/48612S (if using SCSI hard disk)
SBC 486/12	/boot32/38612
SBC 486/12S	/boot32/48612S (if using SCSI hard disk)
SBC 3862x	/boot32/38620
SBC 3863x	/boot32/38620

Note

Before generating your new version of the OS with **mksys**, you should rename the pre-generated OS to something which will not be over written during the OS generation process. For example:

```
rename /boot32/38612 to /boot32/i38612
rename /boot32/38612.386 to /boot32/i38612.386
```

You must rename both the third stage bootstrap loader (*38612* in this example) and the iRMX OS (*38612.386* in this example).

To reboot the system with the pre-generated version of the OS, follow these steps:

1. Shut down the OS with the **sh** command (an alias for **shutdown w=0**):

   ```
   - sh <CR>
   ```

2. When you see a `SHUTDOWN COMPLETED` message, reset the system. Press the RESET button or turn the RESET switch on the front panel of the computer.

3. The SCT runs again. Answer its prompts as you did in Step 2, to invoke the SDM monitor.

4. When you see the monitor prompt (.), enter this command to bootstrap load the OS. Use the boot file pathname from the previous page:

   ```
   .b /boot32/board_name <CR>
   ```

   If the system does not boot, call your Intel Customer Support representative.

B. Making a New Version of the OS

⚠ **CAUTION**

For any one definition file, do not invoke the **mksys** command again after you have successfully used it once to generate the OS. Because **mksys** does not use your modified file for input, it will overwrite the local copy of your definition file. You will lose any modifications that you have made to the file. Instead of using **mksys**, invoke the ICU using the definition filename as the input parameter.

Table 5-8 shows the files used and generated by the **mksys** command for different processor boards. It also shows the board name to use for your processor board in these commands.

See also: Multibus I Standard Definition files, *ICU User's Guide and Quick Reference*

Table 5-8. Files Generated by mksys for CPU Boards

CPU Board	Board Name	ICU Definition File	Generation Directory	Boot File Pathname
SBC 386/12	38612	38612.bck	:home:38612	/boot32/38612.386
SBC 386/12S	38612S	38612S.bck	:home:38612S	/boot32/38612S.386
SBC 486/12	48612	48612.bck	:home:48612	/boot32/48612.386
SBC 486/12S	48612S	48612S.bck	:home:48612S	/boot32/48612S.386
SBC 386/2x	38620	38620.bck	:home:38620	/boot32/38620.386
SBC 386/3x	38620	38620.bck	:home:38620	/boot32/38620.386

To generate a default OS:

1. From Table 5-8, find the board name that corresponds to your processor board. For instance, the SBC 386/3x board has board name 38620.

2. The **mksys** command takes approximately 15 minutes to execute. Sometimes the screen scrolls faster than you can read it. This is because the **mksys** command rapidly shows the ICU screens as it executes. The scrolling is not a cause for concern.

   Enter this command substituting `board_name` for your processor board:

   ```
   - mksys board_name <CR>
   ```

   Various ICU prompts are displayed on the screen; they are answered by a **submit** file. Do nothing until you see an `END submit` message.

   The **mksys** command creates a directory under your *:home:* directory. It attaches that new directory as your default prefix. It creates a local copy of the appropriate ICU definition file. Then it generates on disk an updated version of the OS suited to the processor board specified in the command.

   The command ends with a line similar to this:

   ```
   -END submit :config:cmd/mksys.CSD
   ```

3. Following the `END submit` message, use either the Aedit editor or the **skim** command to discover if errors occurred in the process.

   See also: **skim** command, *Command Reference*

   While the **mksys** command was running, it recorded its progress in an output file. Use the **skim** command to read this output file to find out if any errors occurred during the generation process. Replace `board_name` with the same name you used in the **mksys** command:

   ```
   - skim board_name.out <CR>
   ```

   Press the space bar at the `more?` prompt to page through the file.

   If the output file contains any error messages, a problem occurred during the **mksys** generation. Use an example file to identify the errors.

   See also: Generation output file, *ICU User's Guide and Quick Reference*

If you receive this error message, use the ICU to change the amount of memory reserved for the iRMX III OS.

```
ERROR 192:  NO SPACE FOR SEGMENT - BASE NOT SET
```

See also: Changing memory size, Appendix E

If you receive a different error, call Intel Customer Support.

Note

The **mksys** command copies a third stage bootstrap loader for you. If you create a different boot system using the ICU rather than **mksys**, you must remember to copy the correct third stage from the bootstrap loader directory to the */boot32* path. For example:

```
copy /bsl/38612.32 to /boot32/38612 <CR>
```

Third Stage files in the iRMX III system have the suffix *.32* to indicate that they load 32-bit code.

Step 10. Testing by Booting from the Hard Disk

To test the previous installation, you must shut down the OS, then reset and reboot the Multibus system.

1. Shut down the OS with the **sh** command (an alias for **shutdown w=0**):

```
- sh <CR>
```

2. When you see a `SHUTDOWN COMPLETED` message, reset the system. Press the RESET button or turn the RESET switch on the front panel of the computer.

3. The SCT runs again. Answer its prompts as you did in Step 2, to invoke the SDM monitor.

⟹ **Note**

> If you interrupt the SCT with a <Ctrl-C> and proceed directly to the boot phase, on some configurations a spurious parity error can cause a break to the SDM monitor. To avoid this problem let the SCT run to completion. However, if this spurious break to SDM does occur, simply type g <CR> at the SDM prompt and the boot will continue.

4. When you see the monitor prompt (.), enter this command to bootstrap load the OS. Use the same board name that you used in previous commands:

```
.b /boot32/board_name <CR>
```

   If the system does not boot, call your Intel Customer Support representative.

5. When prompted for the date and time, enter the letter `e`:

```
DATE:  e <CR>
TIME:  e <CR>
```

6. At the Logon prompt, enter the Super logon and the default password:

```
Logon: super <CR>
Password: passme <CR>
```

Step 11. Copying the New Version over the Start-up Version

➾ **Note**

If you do not perform this step, or you copy incorrect file names, the system will not boot automatically the next time you start it. If that occurs, repeat Step 10 to boot the system, then perform this step.

Assuming the new OS is running, use this procedure to copy the new bootable image over the image first used to boot the system. If the OS is not running, call your Intel Customer Support representative.

This step allows you to automatically boot the OS, rather than setting boot parameters each time you start the Multibus system.

Issue these **copy** commands, using the same board name as in previous commands:

```
- copy /boot32/board_name over /system/rmx86
- copy /boot32/board_name.386 over /system/rmx86.386
```

Appropriate values for `board_name` are found in the second column of Table 5-8.

The file */system/rmx86* is the default BSL third-stage pathname for the bootstrap loader on Multibus I systems. After executing the above commands, the file contains the BSL third-stage for your processor board. The file */system/rmx86.386* contains the OS system for your processor board. From now on, you may let the system boot on its own, or you may enter `b` at the monitor dot prompt. The OS boots using the target BSL third-stage file */system/rmx86* and the target OS file */system/rmx86.386*.

Step 12. Taking Precautions for System Security

Use this step to protect the security of your system. You want to make sure that unauthorized people cannot log on as the system manager because a system manager can make changes to the OS. Memorize the new password you enter below. You must know the password to perform system maintenance.

A password can be up to eight characters long. It may consist of any printable ASCII characters, and is case-sensitive.

1. Change the system manager password using the **password** command:

```
- password <CR>
```

2. You are prompted with a series of commands. Select `c` to change the password:

```
Enter the command:  c <CR>
```

3. You receive these prompts. Respond to each. The password entries that you make will not be shown on the screen:

```
Enter your user name - super <CR>
Enter the old password - passme <CR>
Enter the new password - new_password <CR>
Repeat the new password - new_password <CR>
```

4. To exit, enter `e` as follows:

```
Enter the command:  e <CR>
```

There is another logon called World that does not have the powers of the system manager. The World default password is <CR> (press <CR> at the password prompt). Do not change the password for World. If you want to remove the ability of the user World to access your system, delete the file *:config:user/world*, and use the **password** command to remove the World user from *:config:udf*.

Do not routinely log on as the system manager. Instead, log on either as World or as some other user that you define. Log on as Super only to manipulate system files. This reduces the chance of deleting or modifying a system file by mistake.

To log off as Super, use the **logoff** command. To become Super while logged on as another user, enter the command **super**, then the Super password. To return to the ordinary user logon, enter the **exit** command.)

Step 13. Merging the User and Configuration Files

Note

Perform this step only if you have installed over a previous version of the OS, using the Partial format option. If you formatted the whole disk, the installation is finished.

CAUTION

The default *r?logon* file submits the file *:config:alias.csd* in addition to *:prog:alias.csd*. The file *:config:alias.csd* contains aliases used by the OS. Installation instructions and example code use these aliases as defined. Do not modify the existing aliases in *:config:alias.csd*; adding new aliases is acceptable.

User and Alias Files

The **submit** file for the installation renamed the existing */user* directory to */user_old*.

1. Enter these commands to restore user files to their original directory structure, and put the boot system in Super's *:home:* directory. Replace `board_name` in the last command below with the same board name you used in previous commands.

```
- rename /user to /user_new <CR>
- rename /user_old to /user <CR>
- rename /user_new/super/board_name
   to /user/super/board_name <CR>
```

   You can now delete the */user_new* directory.

2. The default *r?logon* file for each user submits *:config:alias.csd* in addition to submitting *:prog:alias.csd*.

   To allow users access to the aliases in *:config:alias.csd*, add this line to the *:prog:r?logon* file for each user:

```
submit :config:alias
```

3. Some of the aliases in *:config:alias.csd* are the same as aliases defined in the old *:prog:alias.csd* file. An alias defined in *:config:alias.csd* should not also be defined in *:prog:alias.csd*.

   From this list, delete duplicate aliases in file *:prog:alias.csd*. Invoke the 32-bit languages with the **run86** utility.

   ```
   alias ad=attachdevice
   alias af=attachfile
   alias crdir=createdir
   alias dd=detachdevice
   alias df=detachfile
   alias flim=floppyimage
   alias install=submit :config:cmd/instal(#0)
   alias tinstall=submit :config:cmd/tinstall(#0)
   alias mksys=submit :config:cmd/mksys(#0)
   alias plm386=run86 -fixplm :lang:plm386
   alias asm386=run86 :lang:asm386
   alias bnd386=run86 :lang:bnd386
   alias bld386=run86 :lang:bld386
   alias lib386=run86 :lang:lib386
   alias map386=run86
   alias ic386=run86  :lang:ic386
   ```

Alias Additions

4. You may want to add the new user aliases listed below to your *:prog:alias.csd* file. These are defined in the file *:config:default/alias.csd*.

   ```
   alias s=submit
   alias m=skim
   alias trv=traverse
   ```

Configuration Files

These instructions help you decide how to merge your old iRMX II or iRMX III configuration files with the newly installed files in the *:config:* directory. After you have merged these files to your satisfaction, you may remove the old versions of the files.

The new system configuration files in the *:config:* directory are compatible with previous releases of the OS. This release can boot and use the old versions with no problems. For some of the files, however, if you simply overwrite the new version with your previous version, some important enhancements are lost.

5. The *:config:termcap* file is the iRMX HI CLI terminal attribute file. The information in this file provides terminal attribute definitions for the CLI and for Aedit. Using the CLI's **set** command allows Aedit to operate without creating a specific *aedit.mac* file, as was necessary in the past.

   Compare the new *termcap* file with changes you made to your previous version. Decide whichever file provides a better starting point for merging your changes; adding your changes to the new file may be easier. If *:lang:aedit.mac* or *:home:aedit.mac* exist, they need to contain only utility macros, not terminal attributes. Also, *:lang:useful:mac* contains some valuable macros.

6. The *:config:terminals* file is the Human Interface (HI) user terminal configuration file. The format of this file allows a Network Server system to provide remote system device services to a diskless remote file consumer. The *terminals* file allows multiple remote consumers to use one copy of the file for each of their unique terminal definitions.

   Read the description of *:config:terminals* in the *System Configuration and Administration* manual for more information. The original form of the file will work fine for the local system. To restore the original iRMX II version, enter:

   ```
   - copy /rmx286/config/terminals over :config:<CR>
   ```

   To restore the old iRMX III version, enter:

   ```
   - copy /rmx386/config/terminals_old over :config:terminals <CR>
   ```

7. The *:config:udf* file is the user definition file. No changes have been made to this file's format.

   If you defined users in a previous version of the iRMX II UDF file, restore it with this command:

   ```
   - copy /rmx286/config/udf over :config:<CR>
   ```

   For a previous version of the iRMX III UDF file:

   ```
   - copy /rmx386/config/udf_old over :config:udf <CR>
   ```

8. The *:config:cdf* file is the Administration Unit definition file for network consumer systems. No changes have been made to this file's format.

   If you defined any networking Administration Units (subnets) in a previous version of the iRMX II CDF file, restore it with the command:

   ```
   - copy /rmx286/config/cdf over :config:<CR>
   ```

   For a previous version of the iRMX III CDF file:

   ```
   - copy /rmx386/config/cdf_old over :config:cdf <CR>
   ```

9. All users that you created in an iRMX II system are still valid in your iRMX III system. The process of adding and configuring users has not changed. To add all of your iRMX II users to your iRMX III system, enter:

   ```
   copy /rmx286/config/user/* to :config:user <CR>
   ```

   For a previous version of the iRMX III OS users, enter:

   ```
   copy /rmx386/config/user/*_old to :config:user/* <CR>
   ```

10. The *:config:r?init* file is the iRMX System Initialization file. This submit file is executed when the HI is initialized. Its output goes to the first entry in *:config:terminals*. No CLI commands, such as **background** or **alias**, can be used, because no CLI job is created when the file is executed.

    A default version of the *config:r?init* file is provided. It contains a description of how to set the file up to load one of the SBX 279(A) menus. To select the default version, enter:

    ```
    copy :config:default/rinit over :config:r'?'init
    ```

    Modify the file as necessary.

11. The *:config:logon.msg* file is the logon message file. Modify the *logon.msg* file as needed.

Where to Go Next

If you want to turn off the Multibus system, first invoke the **sh** command (as Super). You may leave the system running, but you should first log off as Super by invoking the **logoff** command.

To learn more about your new OS, see these manuals:

Topic	**Manual**
Basic operations, such as logging on, entering commands, etc.	Chapter 9 of this manual
Features of the iRMX Operating Systems	*Introducing the iRMX Operating Systems*

➡ **Note**

Chapter 9 describes the installation process for online iRMX manuals and the Windows Help file for the iRMX system calls supplied with this release. Installation of both these features is performed separately from the installation procedure in this chapter.

□ □ □

Installing on Windows NT systems used as iRMX development systems 6

The iRMX III R2.3 Software includes the new capability of having a Windows NT 4.0 system act as an iRMX Cross Development System. This feature allows both iRMX bootimages and applications to be developed on the NT Host and then run on the iRMX target system. Components of the INtime Real Time Extension to Windows NT which is based on iRMX software are included to provide a communications mechanism between the NT Host and the iRMX target. This NTX ("NT eXtension") communications mechanism includes both UDP/IP and Serial (at up to 115KB) channel capabilities in the product and will include Multibus II transport as a third communications channel in the final release. Using these INtime components and the NTX interface on the NT Host, and the corresponding NTX interface software on an iRMX system. Applications can be remotely down loaded on an iRMX system and debugged using an NT-hosted version of Soft-Scope.

Before performing these installation steps, ensure that your computer meets the minimum requirements necessary to complete the installation.

Hardware Requirements

- A PC based on a Pentium or higher IntelArchitecture CPU capable of running Windows NT 4.0
- At least 20 Mbytes of system memory
- An ATAPI IDE CDROM drive
- A 1.44 Mbyte 3.5-inch diskette drive.
- 52 Mbytes of free hard disk space

Software Requirements

- Windows NT 4.0 with Service Pack 4 or later

Installing iRMX III R2.3 Software from the Installation CDROM

1. Insert the iRMX III R2.3 Installation CDROM into the NT system's CDROM drive

2. Start the Add/Remove Programs Applet in the NT Control Panel (Start/Settings/Control Panel/ Add/Remove Programs)
3. Click the Install button to start the Add portion of the Applet
4. Click the Next Button to get to the Browse Screen
5. Click the Browse Button to select the file InstlRMX.exe on the CDROM. Once the file has been selected, click the Open Button
6. Click the Finish Button to begin running the iRMX III R2.3 Installation program. Follow the directions displayed on the system console as the installation proceeds.

➡ **Note**

It is recommended that you use the default directory name (i,e, iRMXIII) as the root directory on the drive you choose to contain the iRMX R2.3 Software.

Once the iRMX R2.3 Software has been installed, remove the CDROM and allow the installation program to reset the NT system.

Generating an iRMX R2.3 Boot image

Selecting appropriate ICU definition file

Because of the complexity of many iRMX systems, we have provided a number of standard ICU (Interactive Configuration Utility) definition for standard PC systems plus Multibus I and Multibus II CPU boards. We recommend that you start with one of these standard systems and then customize it to fit your applications systems needs.

Creating a Generation Subdirectory

The iRMX III R2.3 Software installation places all the system generation files in various subdirectories under the \iRMXIII\sysgen directory. The installation utility also places into the NT Registry an association between :SD: (present in the iRMX definition files) and this \iRMXIII\sysgen directory. We recommend that you create a separate subdirectory under the ..\sysgen directory for each application system you require. By convention, if you were to use the default PC definition file PC.BCK as your starting point, you would also create a directory \iRMXIII\sysgen\pc in which to generate this new application image.

Running the ICU

Starting with the PC.BCK definition file listed above, do the following

1. Invoke the DOS console applet so you have a command line from which to invoke the ICU.
   ```
   (Start/Programs/Command Prompt)
   ```
2. Make the sysgen directory you current directory
   ```
   cd \iRMXIII\sysgen<Enter>
   ```
3. Create a generation subdirectory
   ```
   mkdir pc<Enter>
   ```
4. Make the generation subdirectory you current directory
   ```
   cd pc<Enter>
   ```
5. Invoke the ICU on the default PC definition file with a target output definition file in your working directory. Follow the instructions to restore the .BCK file into a working ICU definition file.
   ```
   ..\rmx386\icu\icu386 ..\rmx386\icu\pc.bck to pc.def<Enter>
   ```
6. Make any desired customization to the ICU definition file to meet the needs of your application system. Refer to the iRMX ICU User's Guide (ICU.pdf) in the \iRMXIII\Manuals directory for more information on the ICU.
7. Once your system is customized as needed, return to the Command Screen of the ICU and generate the generation and configuration files.
   ```
   g<Enter>
   ```

1. Exit the ICU saving you custom definition file
   ```
   e<Enter>
   ```

Invoking the Generation Batch File

Invoke the batch file listed on the command line when the ICU exited above, redirecting the output to a log file.

```
pc > pc.log<Enter>
```

Checking for Errors

Edit the log file to insure there are no errors in any of the generation actions. Warnings are to be expected on many of the BND386 invocations. Make sure the BLD386 invocation produces no errors.

Moving the new Bootimage to the Target System

Unless you gave a different pathname to the bootimage being generated by one of the standard definition files, you will find the bootimage generated above in the \iRMXIII\sysgen\boot32 directory for PC and Multibus I systems and in the \iRMXIII\sysgen\msa32\boot directory for Multibus II systems. Once your application system has been generated with no errors, move it to the iRMX target system either via floppy or via FTP, depending on what iRMX operating services you have running on the target system. Until you have tested the new bootimage, it is good practive to not overwrite working bootimages on the target system with new, untested bootimages.

Booting the new Bootimage on the Target System

Use the boot method appropriate to your iRMX target system to load and run the new bootimage generated above.

Generating an iRMX III R2.3 application

Using Intel/RadiSys OMF386 Tools

The \iRMXIII\Demo Directory contains a number of example programs that are compiled using the Intel/RadiSys OMF386 tool chain. Each demo includes a makefile.m makefile that you can use with the nmake utility from Microsoft to generate these sample programs. The include pathnames and logical names in the source files for these example programs are set up by the iRMX III R2.3 Software NT Installation program. The nmake utility comes with the Microsoft Developer Studio (MSVC 6.0).

Using Microsoft Developer Studio, Version 6 (MSVC 6.0)

The \iRMXIII\Project directory includes a flat model sample program that is compiled using the Microsoft Developer Studio (MSVC 6.0). Each subdirectory under the Projects directory is a separate workspace for MSVC.

Proper generation of iRMX applications using MSVC 6.0 require that the following settings be made in the Microsoft Developer Studio under the Project/Settings pulldown menu (Leave all other settings at their default values):

General Tab:

- Not Using MFC

C/C++ Tab:

General:

Debug Info:	C7 Compatible

Preprocessor:

Additional include directories:	<installed drive>:\irmxIII\include
Ignore Standard Include Paths	<checked>

Link Tab:

General	
Object/Library Modules:	cstrtf3m.obj, ciff3m.lib, netiff3m.lib, rmxiff3m.lib
Generate debug info	<checked>
Ignore all default libraries	<checked>
Customize	
Use Program Database	<UN checked>
Debug	
Debug Info	
Debug Info	<checked>
Microsoft Fromat	<checked>
Input	
Object/Library Modules:	cstrtf3m.obj, ciff3m.lib, netiff3m.lib, rmxiff3m.lib
Ignore all default libraries	<checked>
Additional library paths	<installed drive>:\irmxIII\lib
Output	
Stack Allocations:	
Reserve:	0x4000
Commit:	0x2000
Version Information:	
Major:	21076
Minor:	20052

Where to Go Next

To learn more about your new OS, refer to these manuals:

Topic	**Manual**
Basic operations, such as logging on, entering commands, etc.	Chapter 9 of this manual
Features of the iRMX Operating Systems	*Introducing the iRMX Operating Systems*
Browsing/Cross Debugging an application on an iRMX target system from Windows NT.	Chapter 6 *of System Configuration and Administration*

➨ **Note**

Chapter 9 describes the installation process for online iRMX manuals and the Windows Help file for the iRMX system calls supplied with this release. Installation of both these features is performed separately from the installation procedure in this chapter.

❑ ❑ ❑

iRMX GETTING STARTED CHAPTERS

This section contains information that lets you get started with your iRMX OS. These are the chapters in this section:

Chapter 7.	**DOSRMX Specifics**
Chapter 8.	**iRMX for PCs Specifics**
Chapter 9.	**Getting Acquainted with the Operating System**
Chapter 10.	**Where To Go From Here**

DOXRMX Specifics 7

Customizing the OS

You can specify which parts of the OS are loadable using the **sysload** command from the command line or modifying the *:config:loadinfo* file. Additionally, by modifying parameters in the *rmx.ini* file, you can enhance the performance of your application system.

After you install the OS, you may want to modify the default configuration to match your application. Some of the defaults you can change are:

- System memory configuration
- User IDs and memory requirements
- Terminal configuration
- Loadable jobs (including network jobs)
- Loadable device drivers
- Interrupt response

See also: Loadinfo file and rmx.ini parameters, *System Configuration and Administration*

Starting DOSRMX

There are three scenarios for starting DOSRMX depending on which partition it is installed.

- From a DOS partition
- From an iRMX partition
- From *config.sys* on a DOS partition (DOS 6.22 only)

Starting DOSRMX from a DOS Partition

After you install DOSRMX, you can boot it from DOS using these steps:

1. To load DOSRMX, use the *rmx.bat* file, followed by the drive letter where the software is installed. Do not follow the drive letter with a colon. If the iRMX OS is installed on drive C:, enter:

   ```
   C:\> \dosrmx\rmx c <CR>
   ```

A number of messages will be displayed. When these messages appear, the OS is running.

```
DOS is now running as a VM86 iRMX Task
Use Alt-SysRq key combination to switch to the iRMX console

C:\>
```

Note

If you receive one of the messages below, you have not removed all the lines in your *autoexec.bat* or *config.sys* file that consume extended memory or switch the processor to Protected Mode.

```
No extended memory present
Protected mode program already loaded
```

Starting DOSRMX from an iRMX Partition

To run DOSRMX if you have installed it on an iRMX partition, use these steps.

1. Start the system or press <Ctrl-Alt-Del> to warm boot the system.
2. Invoke the **rmxtsr** command at the DOS prompt to load the iRMX interface.
3. Invoke the **loadrmx** command. This example assumes that the boot file is *dosrmx* and that partition 2 on fixed disk 1 is the iRMX partition (c_rmx2).

   ```
   C:\>  loadrmx -n c:\dosrmx\dosrmx -s c_rmx2 -f n <CR>
   ```

   See also: **loadrmx**, *Command Reference*

4. Press <Alt-SysRq> to invoke DOSRMX from the iRMX partition.
5. Log on to DOSRMX. Enter `world` at the Logon: prompt and <CR> at the Password: prompt.

After DOSRMX is installed on an iRMX hard disk, you have two options for backing up the system:

- You can use iRMX-NET and the iRMX network redirector to map the iRMX partition, then continue to use your existing DOS tools to back up the system.

See also: Using iRMX-NET in a DOS Environment, *System Configuration and Administration, Network User's Guide and Reference*

- You can add an Adaptec 1542 SCSI controller as a secondary (not primary) disk controller, then use the Intel-supplied tape driver and the iRMX **backup** command to back up the system.

  See also: **backup**, *Command Reference*

Starting DOSRMX from Config.sys

You can load DOSRMX from the *config.sys* file provided the system is using DOS 6.22.

See also: Booting DOSRMX and iRMX for PCs Automatically, *System Configuration and Administration*

Loading from *config.sys* has these advantages:

- Loading HIMEM.SYS driver.
- Loading DOS into HMA memory (DOS=HIGH).
- Automatically invoking DOSRMX upon boot-up.

To load DOSRMX from the *config.sys* file, use these steps:

1. Modify your *config.sys* file to begin with these lines.

   ```
   buffers=40
   files=64
   lastdrive=Z
   install=C:\dos\himem.sys
   dos=high
   install=C:\dosrmx\rmxtsr.exe
   install=C:\dosrmx\loadrmx.exe -n C:\dosrmx\dosrmx -s c_dos -f d -w
   .
   ```

   These values for `buffers`, `files`, and `lastdrive` are recommended; you can change them to match the requirements of other software applications.

2. Reset (warm boot) your system. DOSRMX should be automatically invoked upon reboot.

If the OS has not booted, check for:

- Syntax errors in the *config.sys* file
- Driver conflicts

Logging On

Refer to Chapter 9 for instructions on how to log on the iRMX operating systems.

Using the Hot Keys

A number of Hot Keys control the mode of operation of the console.

Invoke the Hot Keys by entering an <Alt-*appropriate key*> sequence. Table 7-Table 7-1 provides a summary of the Hot Keys and their function.

Table 7-1. Hot Keys

Hot Keys	Function
<Alt-SysRq>	Toggle between DOS and iRMX prompts
<Alt +>*	Change iRMX console foreground color
<Alt ->*	Change iRMX console background color
<Ctrl-Alt-Del>	Restart system (DOS and iRMX OSs)
<Ctrl-Alt-Break>	Break to system debug monitor (SDM) from the iRMX prompt

* Use the '+' and '-' keys on the numeric keypad.

Toggling the Console Between DOS and iRMX Operation

You can toggle between the DOS and the iRMX prompts whenever the console is in text mode, but not when it is in graphics mode. Graphics mode is incompatible with the iRMX command line. To toggle the console, press <Alt-SysRq>.

The current display is saved before the change is made, unless the console has been in graphics mode or a DOS application has reset the video card. When you toggle back, the displayed data is the same as when you left.

If the console has been in graphics mode or if a DOS application has reset the video card, you can restore the iRMX prompt by changing to text mode and then pressing <Alt-SysRq> followed by <Enter>. This sets the console display to 80 columns by 25 lines.

At the DOS prompt, you may enter DOS commands at the keyboard; at the iRMX prompt, you may enter iRMX commands at the keyboard. When DOS owns the console, iRMX output to the console is saved in a buffer and is displayed when you switch back to iRMX. When the iRMX OS owns the console, DOS output to the console is discarded.

Changing the Console Colors

You can change the foreground and background colors of the iRMX console. On an EGA, VGA, or Super VGA display, the selectable colors are: white, black, blue, green, cyan, red, magenta and yellow, at both normal and high intensity. The default colors are normal intensity, white foreground on a black background.

You cannot select the same color for the foreground and the background.

To change the foreground color press

<Alt +>

using the plus key on the numeric keypad. Each time you press this key sequence, the foreground color changes to the next available color, eventually wrapping around to the initial color.

To change the background color press:

<Alt ->

using the minus key on the numeric keypad.

You can change colors by changing the `x` and `y` parameters in this line from the *:config:loadinfo* file.

```
sysload /rmx386/jobs/keybd.job(x,y,7a55)
```

See also: the *loadinfo* file for the line `sysload keybd.job(f,1,7a55)`

Breaking to the SDM Debug Monitor

You can invoke the system debug monitor from the iRMX prompt (not the DOS prompt) by pressing <Ctrl-Alt-Break>.

If you want to use SDM to debug an iRMX application while the console is in graphics mode, install an auxiliary MDA monochrome card attached to a monochrome monitor. Then, SDM will always use this monitor for its console output. (The iRMX OS, however, does not support the MDA monochrome card and the HI will not use it.) Use the **console** command to direct output and input for the terminal.

You can resume normal operation by entering the system debugger **g** command at the system console. This command exits SDM.

⇒ **Note**

A General Protection error will also cause a break to SDM, even if the DOS prompt is displayed. In this case, entering `g 284:1c` will exit SDM and return to the iRMX prompt. To reboot the system from SDM, enter `o64,fe` at the system console. Alternatively, you can use a power OFF-then-ON cycle.

Both the DOS and the iRMX OS are effectively suspended when control passes to SDM on the system console. You will know that you are in SDM when the system prompt changes to **..** (two dots).

Restarting the System

You can initiate a cold reset of the entire system by pressing <Ctrl-Alt-Del>. This resets the processor and should not be done unless necessary.

Networking Capabilities

DOSRMX gives DOS and the iRMX OS simultaneous access to network services. This support for both DOS and the iRMX OS provides a variety of capabilities:

- DOS and iRMX applications that communicate on the network run unchanged when they run on the same processor.
- DOS files can be accessed from a remote file consumer without a dedicated file server.
- DOS and iRMX OSs running on the same processor can share a single network controller.
- OpenNET networking support provides connections to computers running the DOS and UNIX operating systems.

- DOS networking applications can use the network controller in a Multibus I or Multibus II system.
- TCP/IP networking support provides access to other operating systems through TCP/IP, FTP, or TELNET.
- NFS networking support provides access to remote files on other operating systems.

Network Software Choices

PCs running DOSRMX can run this network software used by DOS:

- MS-Net client or server
- IBM PC LAN client or server
- Novell NetWare client
- Combinations of iRMX-NET client, MS-Net client, and NetWare on one computer
- DOS Virtual Terminal consumer

⚠ **CAUTION**

A system error may occur if MS-Net is active when DOSRMX is loaded with a network system job specified in *\rmx386\config\loadinfo* and both use the same network controller.

See also: Using iRMX-NET in a DOS Environment, *System Configuration and Administration*

Using iRMX-NET

After you have installed DOSRMX, you can connect your system to a network. Before you attempt to establish connection to a network, ensure that:

- The appropriate supported Intel network interface controller board for your system has been installed.
- The Ethernet cable is connected.
- DOSRMX is running.
- The appropriate iNA 960 job for your network interface controller (NIC) is loaded and running.
- For ISO networking, the iRMX-NET LAN is set up and ready to accept nodes.
- Additionally for TCP/IP networking:
  - The TCP/IP Kernel jobs and optionally the NFS jobs are loaded and running
  - The TCP/IP LAN configuration files are set up

See also: Network jobs, *System Configuration and Administration*;
Using the Network, Supported Hardware, *Network User's Guide and Reference;*
Installing and Starting TCP/IP, *TCP/IP for the iRMX Operating System*

Accessing DOS-formatted Diskettes from the iRMX Prompt

Note
You can access iRMX formatted diskettes only from the iRMX prompt.

Toggle to the iRMX prompt from DOS.

Before any diskette may be used by the iRMX OS, you must attach the diskette drive to the OS and assign it a logical name. Logical names are surrounded with colons to differentiate them from other device or file names.

See also: Creating and Using Logical Names, *Command Reference*

To attach DOS diskette drive A:, enter:

```
- ad A_DOS as :f: <CR>
```

See also: **attachdevice**, *Command Reference*

Drive A: has now been attached with the iRMX logical name *:f:*. Obtain a directory listing of the DOS-formatted diskette in drive A: by entering:

```
- dir :f: <CR>
```

The directory listing is displayed in regular iRMX fashion.

See also: **dir**, *Command Reference*

Toggle the console back to DOS operation and compare the DOS and iRMX directory listings of the DOS diskette.

Before proceeding to the next section you must detach the diskette drive.

1. Toggle to the iRMX prompt.
2. Detach the diskette drive by entering:

```
- dd :f: <CR>
```

See also: **detachdevice**, *Command Reference*

⚠ **CAUTION**
Always detach and reattach diskette drives when changing iRMX-formatted diskettes; otherwise data may be corrupted.

See also: Switching Diskettes, *Command Reference*

Using a DOS-formatted Hard Drive from the iRMX Prompt

If you booted with DOS as the system device, toggle to the iRMX prompt from DOS and enter:

```
- dir :sd: <CR>
```

If your primary DOS partition is not being used by the iRMX OS as the system device, toggle to the iRMX prompt from DOS and enter:

```
- ad c_dos as :w: <CR>
```

The DOS partition on Drive C: is now attached and you can access it as :w: from the iRMX OS. Use the logical name *:w:* just as any other iRMX logical name.

See also: Logical names, *Command Reference*

Using an iRMX-formatted Hard Drive from the DOS Prompt

If your system contains an iRMX-formatted drive or an iRMX-formatted partition, toggle to the iRMX prompt from DOS and attach that iRMX-formatted partition (or drive) as *:x:*. For example, if you have an iRMX partition on your first physical drive and that partition has not been attached, enter:

```
- ad c_rmx as :x: <CR>
```

If you receive the error:

```
c_rmx, device already attached
```

c_rmx is being used as your system device *:sd:*.

After establishing the iRMX logical name and loading the appropriate networking job, use the **offer** command to set *:x:* as a network public directory.

```
- offer :x: as w <CR>
```

See also: **offer** command, *Command Reference*

Once the partition has been offered by the iRMX OS, DOS can access it. If you are running MS-NET on DOS, toggle to DOS from the iRMX prompt and enter:

```
C:\> net use f: \\sysname\world <CR>
```

where `sysname` is the iRMX-NET network name of your system.

See also: Network names, *Network User's Guide and Reference*;
loadname, **setname** commands, *Command Reference*

You can now access the iRMX logical name *:x:* (or *:sd:*, if the device is your system device) from DOS as DOS logical drive F:. For example, enter:

```
C:\> dir f:<CR>
```

The directory listing of the iRMX partition is displayed in regular DOS fashion.

Initializing DOSRMX for Shutdown

Before turning the system power off, initialize DOSRMX for system shutdown. This initialization, which you can do if you are logged on as a Super user, flushes data to the disk and not to internal buffers.

1. Toggle to the iRMX prompt from DOS.
2. Execute the **shutdown** command as follows:

   ```
   - shutdown wait=0 <CR>
   ```

3. After this message appears, turn the system power off.

   ```
   ***  SHUTDOWN COMPLETED  ***
   ```

4. Toggle to the DOS prompt. You can turn your system power off now if desired.

See also: **shutdown**, *Command Reference*

□ □ □

iRMX for PCs Specifics 8

Switching Between DOS and iRMX for PCs

You must first perform the following steps if iRMX for PCs is installed on an iRMX partition and you want to change between the iRMX OS and DOS on a regular basis. If iRMX for PCs is installed on a DOS partition, *do not* perform these steps.

1. At the iRMX prompt, change to the iRMX directory *dosrmx*.

```
- cd /dosrmx <CR>
```

2. Attach the DOS partition.

```
- ad c_dos as :c: <CR>
```

3. Create a DOS *dosrmx* directory.

```
- createdir :c:dosrmx <CR>
```

4. Copy iRMX files as listed below:

```
- copy rdisk.exe to :c:/dosrmx <CR>
- copy bootrmx.exe to :c:/dosrmx <CR>
```

5. Append *c:\dosrmx* to the PATH line in your *autoexec.bat* file.

Customizing the OS

You can specify some custom configurations of the OS. You can load certain parts of the OS with the **sysload** command from the command line or by modifying the *:config:loadinfo* file. Additionally, by modifying parameters in the *rmx.ini* file, you can enhance the performance of your application system.

After you install the OS, you may want to modify the default configuration to match your application. Some of the defaults you can change are:

- System memory configuration
- User IDs and memory requirements
- Terminal configuration
- Loadable jobs (including network jobs)
- Loadable device and file drivers
- Interrupt response

See also: Loadinfo file and rmx.ini parameters, *System Configuration and Administration*;
Interrupt response, *\rmx386\readme.txt* file

Starting iRMX for PCs

There are four scenarios for starting iRMX for PCs, depending on the partition in which it is installed.

Table 8-1 is a matrix for selecting commands to boot DOS or iRMX for PCs.

Table 8-1. Boot Command Selection

	OS to Boot	
Partition iRMX for Pcs is Installed On	**DOS OS**	**iRMX for PCs OS**
DOS partition	*rundos*	*runrmx*
iRMX partition	*bootdos*	*bootrmx*

Note: The *rundos* file undoes the changes of the *runrmx* file and vice-versa.

Files Used for Booting iRMX for PCs

RUNRMX iRMX for PCs is installed on an DOS Partition

Invoke the *runrmx.bat* file to change the *config.sys* and *autoexec.bat* files to boot iRMX for PCs. Also invoke this file to undo the changes of *rundos.bat*.

```
C:\> \dosrmx\runrmx <drive with no colon> <CR>
```

`<drive with no colon>` is the DOS partition (drive) on which iRMX for PCs is installed. For example, if iRMX for PCs is installed on drive D:, the invocation is:

```
C:\> \dosrmx\runrmx d <CR>
```

If you have run the *runrmx.bat* file and need to undo the change, invoke *\dosrmx\rundos.bat*.

```
C:\> \dosrmx\rundos <CR>
```

Reset your system using <Ctrl-Alt-Del> for the change to take effect.

BOOTRMX iRMX for PCs is installed on an iRMX Partition

The *bootrmx* file activates the first iRMX partition and resets the system.

```
C:\> \dosrmx\bootrmx <CR>
```

Files Used for Booting DOS

RUNDOS iRMX for PCs installed on a DOS Partition

The *rundos* file copies *c:\config.dos* over *c:\config.sys* and *autoexec.dos* over *autoexec.bat*.

```
- rundos <:logical drive:> <CR>
```

<:`logical drive`:> is the iRMX logical name used to attach the drive. For example, if iRMX is installed on the C: drive, the invocation would be:

```
- rundos :sd: <CR>
```

If iRMX is installed on any drive but C:, you must attach the C: drive before invoking *rundos*, like this:

```
- ad c_dos as :c: <CR>
- rundos :c: <CR>
```

Reset your system using <Ctrl-Alt-Del> for the change to take effect.

See also: **attachdevice** command, *Command Reference*

BOOTDOS iRMX for PCs installed on an iRMX Partition

The *bootdos* file activates the primary DOS partition and resets the system.

```
- bootdos <CR>
```

Logging On

Refer to Chapter 9 for instructions on how to log on the iRMX operating systems.

Using the Hot Keys

A number of Hot Keys control the mode of operation of the console.

Invoke the Hot Keys by entering an <Alt-*appropriate key*> sequence. Table 8-2 provides a summary of the Hot Keys and their function.

Table 8-2. Hot Keys

Hot Keys	Function
<Alt +>*	Change iRMX console foreground color.
<Alt ->*	Change iRMX console background color.
<Ctrl-Alt-Break>	Break to system debug monitor from the iRMX prompt.
<Ctrl-Alt-Del>	Restart system. An error message is displayed if the shutdown command has not been used.

* Use the '+' and '-' keys on the numeric keypad.

Changing the Console Colors

Colors are selectable on EGA, VGA, or Super VGA displays. The default colors are normal intensity, white foreground on a black background. You cannot select the same color for the foreground and the background.

To change the foreground color press:

<Alt +>

using the plus key on the numeric keypad. Each time you press this key sequence, the foreground color changes to the next available color, eventually wrapping around to the initial color.

To change the background color press:

<Alt ->

using the minus key on the numeric keypad.

You can change colors by changing the *x* and *y* parameters in this line in the *:config:loadinfo* file.

```
sysload /rmx386/jobs/keybd.job(x,y,7a55)
```

See also: the *loadinfo* file for the line `sysload keybd.job(f,1,7a55)`

Breaking to the SDM Debug Monitor

You can invoke the system debug monitor (SDM) from the iRMX prompt (not the DOS prompt) by pressing <Ctrl-Alt-Break>.

If you want to use SDM to debug an iRMX application while the console is in graphics mode, install an auxiliary MDA monochrome card attached to a monochrome monitor. Then, SDM will always use this monitor for its console output. (The iRMX OS, however, does not support the MDA monochrome card and the HI will not use it.) Use the **console** command to direct output and input for the terminal.

You can resume normal operation by entering the system debugger **g** command at the system console. This command exits SDM.

Note
A General Protection error will also cause a break to SDM, even if the DOS prompt is displayed. In this case, entering `g 284:1c` will exit SDM and return to the iRMX prompt. To reboot the system from SDM, enter `o64,fe` at the system console. Alternatively, you can use a power OFF-then-ON cycle.

The iRMX OS is effectively suspended when control passes to the SDM monitor on the system console. You will know that you that you are in SDM when the system prompt changes to **..** (two dots).

Restarting the System

You can initiate a cold reset of the entire system by pressing <Ctrl-Alt-Del>. This resets the processor and should not be done unless necessary.

Networking Capabilities

iRMX for PCs enables access to OpenNET and TCP/IP network services. These networks provide connections to computers running the DOS, UNIX, and iRMX operating systems.

⚠ **CAUTION**

If you are booting iRMX for PCs from a DOS partition, ensure that DOS networking has not been loaded prior to the loading of iRMX for PCs.

PCs running iRMX for PCs can also run the iRMX network software listed below.

- ISO-based Protocols:
  - iNA 960 jobs for a programmatic interface
  - iRMX-NET consumer and server for remote file access

  See also: i*.job, *System Configuration and Administration*

- TCP/IP-based Protocols:
  - NFS client and server for remote file access
  - FTP client and server for remote file access
  - TELNET client and server for remote login
  - Socket library programmatic interface for user networking applications

  See also: Overview of TCP/IP, *TCP/IP for the iRMX Operating System*

Using iRMX-NET

After you have installed iRMX for PCs, you can connect your system to a network. Before you attempt to establish a connection to a network, ensure that:

- The appropriate supported Intel network interface controller board for your system has been installed.
- The Ethernet cable is connected.
- iRMX for PCs is running.
- The appropriate iNA 960 job for your network controller is loaded and running.
- For ISO networking, the iRMX-NET LAN is set up and ready to accept nodes.
- Additionally for TCP/IP networking:
    - The TCP/IP Kernel jobs and optionally the NFS jobs are loaded and running.
    - The TCP/IP LAN configuration files are set up.

See also: Network jobs, *System Configuration and Administration*;
Installing and Starting TCP/IP, *TCP/IP for the iRMX Operating System*

Configuring the Screen Saver

iRMX for PCs comes with a configurable screen saver. As with changing screen colors, the parameters for the screen saver are listed in the *loadinfo* file:

```
sysload /rmx386/jobs/keybd.job(x,y,FFFF)
```

where `x and y` configure screen colors and the value `FFFFH`, in units of 10 ms, configures the screen saver. The first of its two functions is the amount of time in which there is no output to the console. The screen saver activates after this time span. The second function serves as the blinking rate for the screen saver icon.

The screen saver is activated and similarly, de-activated based on output to the terminal console. If your program captures keystrokes and processes them, it should send them to the terminal console.

Note

The default value is FFFFH. This sets the screen saver never to be activated. If the screen saver will be activated, ensure that the time delay value is greater than 1770H (60 seconds).

Initializing iRMX for PCs for Shutdown

Before turning the system power off, initialize iRMX for PCs for system shutdown. This initialization, which you can do if you are logged on as a Super user, flushes data to the disk and not to internal buffers.

1. Execute the **shutdown** command as follows:

```
    - shutdown wait=0 <CR>
```

2. After this message appears, turn the system power off.

```
*** SHUTDOWN COMPLETED ***
```

See also: **shutdown**, *Command Reference*

❑ ❑ ❑

Getting Acquainted with the Operating System 9

This chapter tells you how to perform some basic operations with the iRMX OS. It covers these topics:

- Logging on and off
- Moving the cursor
- Managing files
- Using devices
- Using the on-line help facility

See also: The Human Interface (HI), The Command Line Interpreter (CLI), and Understanding the File System, *System Concepts*;
Chapter 1, *Command Reference*

Logging On and Off

Logging On to the Operating System

If your terminal is configured as a static terminal, you do not need to log on to the system. The system prompt is displayed and you may begin entering commands. If you use a dynamic terminal, you must log on to the system.

When the HI initializes a dynamic terminal, it displays a prompt for the user logon name, which is not case-sensitive. When you enter a name, the HI prompts for a password. The password is case-sensitive, and is not displayed when you enter it. The prompts look like this:

```
Logon:
Password:
```

If you have been assigned a user name and password by a system manager, enter these at the prompts. Otherwise, there are two user names defined for all iRMX systems. One is Super, the system manager, with user ID 0. The default password for Super is *passme*. Super has special privileges with regard to file access and command usage. The other user is World, with user ID 65535. World is the default user on iRMX systems, and requires no password. Simply press <CR> at the password prompt.

When you enter a valid user name and the correct password, the HI places you in a home working directory: for example, */user/world*. Then it displays the CLI sign-on message. This is the default message:

```
iRMX HI CLI Vx.y: user = <ID>
Copyright years Intel Corporation
All Rights Reserved
-submit commands
-
```

The sign-on message displays your user ID. When you log on to the system, the HI creates your interactive job and starts the CLI. The CLI executes the logon file, *:prog:r?logon*, which contains a set of **submit** commands. These commands are automatically invoked whenever you log on.

The *:prog:r?logon* file, and any file with a name that contains *r?*, is a hidden file. It does not show up in a normal directory listing. The *:prog:r?logon* file is unique for each user, though many user files can be very similar. You can customize your file by adding commands to it. After processing all the commands in the logon file, the CLI issues a dash (**-**) prompt. You can now enter commands or invoke application programs.

See also: Hidden files, *Command Reference*;
submit and **logoff** commands, *Command Reference*;
Logon, static, and dynamic terminals, *System Configuration and Administration*

Logging Off

Logging off of a dynamic terminal frees the terminal for other users, and frees the memory pool used by your interactive job.

Enter the **logoff** command to log off. Whenever you log off, the CLI searches for the logoff file, *:prog:r?logoff*, and invokes the commands in it. You can customize this file by adding commands to it.

Moving the Cursor

Table 9-Table 9.1 lists the line-editing and cursor-movement keys supported by the CLI, along with other characters that have meaning for the CLI and HI.

Bell Warning

On some systems, when you attempt an invalid action the CLI line-editor beeps the terminal bell. For example, it beeps if you:

- Type <Up-Arrow> to move the cursor up when you have not previously entered commands (there is no history buffer).
- Type <Down-Arrow> to move the cursor down when you are on the last line of the history buffer.
- Type <Right-Arrow> to move the cursor right beyond the line limit.

Terminal Characteristics

The CLI executes on a wide variety of terminals. It uses the *:config:termcap* file to identify terminals and define keys used for special functions, such as cursor movement and line-editing. Each terminal is assigned a terminal name in the file. To change terminal types, use the **set** command and specify a terminal name from the file. To add a terminal or change terminal characteristics, edit the *:config:termcap* file.

See also: Terminal definition file, *System Configuration and Administration*; **set** command, *Command Reference*

Table 9.1. Function Keys and Special Characters

Key	Function
<Right-Arrow>	Moves the cursor one place to the right.
<Left-Arrow>	Moves the cursor one place to the left.
<Up-Arrow>	Replaces the current command line with the previous command line.
<Down-Arrow>	Replaces the current command line with the next command line.
<Home>	Moves the cursor in the direction of the last arrow key used. After <Right-Arrow> or <Left-Arrow>, <Home> moves to the end or beginning of the line. After <Up-Arrow> or <Down-Arrow>, <Home> moves to the first or last command in the history buffer.
<Del> or <Backspace>	Deletes one character to the left of the cursor.
<DelCh> (<Ctrl-F>)	Deletes the character on which the cursor is positioned; usually configured to <Ctrl-F>.
<DelR> (<Ctrl-A>)	Deletes all characters to the right of and including the cursor location: usually configured to <Ctrl-A>.
<DelL> (<Ctrl-X>)	Deletes all characters to the left of the cursor; usually configured to <Ctrl-X>.
<Esc>	Executes the entire command line, regardless of the cursor position.
<CR>	Executes from the beginning of the command to the cursor. Everything to the right of the cursor is ignored.
<Ctrl-C>	Aborts the current foreground command and returns control to the CLI.
&	An ampersand designates the next line as a command continuation line.
;	A semicolon designates a comment in HI commands. Everything to the right of the semicolon is considered a comment.
' or "	Single or double quotes around a character remove the special meaning of the character. For example, to use an ampersand or semicolon in a command for any other purpose than line continuation or a comment, surround the character with quotes. The only special characters not affected by the quotes are slashes (/), which are pathname separators, and dollar signs ($). The second quote must match the first. To include a quote character inside a quoted string, specify the character twice or use the opposite type of character (single or double).

Managing Files

One of the main commands to manage files is the **copy** command. Use it for:

- Creating a new file
- Displaying the contents of files at the console screen
- Copying one or more files to new files
- Copying data from one file and overwriting data in another file
- Appending data from one or more files to the end of data in another file
- Concatenating multiple files into a single file

The examples that follow show you how to use these techniques. They also illustrate the general use of the `to`, `over`, and `after` parameters common to many iRMX commands. The syntax for the commands used in these sections is typically:

```
command infile [to|over|after] outfile options
```

Creating a Simple Data File

Use the **copy** command to create data files. You do this by redirecting keyboard input to a file. To create a file called *alpha* under your *:$:* working directory and write two lines of data into the file, enter:

```
-copy :ci: to alpha <CR>
aaaaa <CR>
bbbbb <CR>
<Ctrl-Z>
```

The HI responds:

```
:ci: copied to alpha
-
```

The command does not prompt you for the data lines; you simply begin entering data after you press <CR> at the end of the command line. The <Ctrl-Z> entry writes an end-of-file mark at the end of your input (but not into the file you are copying), to inform the **copy** command that there is no more data to be copied.

After you enter the last line of data, you must press <CR> before you enter <Ctrl-Z> to insert an end-of-file. Otherwise, the <Ctrl-Z> will be ignored.

If the file already exists, this message and query appear:

```
alpha, already exists, OVERWRITE?
```

Enter `y` to overwrite the file, or `n` to cancel the command.

Displaying the Contents of Files

To display a file's contents on the screen, enter:

```
-copy alpha <CR>
```

The HI responds:

```
aaaaa
bbbbb
alpha copied to :co:
-
```

This example uses the default preposition `to` and default output file *:co:*, which means that the command copies the output to the console screen.

To halt the scrolling of a displayed list, press these control keys:

<Ctrl-S> Stops the data from scrolling off the screen until you press <Ctrl-Q>.

<Ctrl-W> Puts the terminal into scrolling mode. In this mode, output stops after a single screen of data appears. Entering another <Ctrl-W> displays the next screen of data. <Ctrl-Q> exits this mode.

<Ctrl-T> Similar to <Ctrl-W> except output stops after each line.

<Ctrl-Q> Resumes scrolling of listed data until the end-of-file is reached or you enter <Ctrl-C>.

<Ctrl-C> Cancels listing of the data and returns control to the HI, which prompts for a new command.

Copying to New Files

You can copy multiple files with one **copy** command. The files are copied in the same sequence you specify in the input list and output list on the command line. To copy files *a1*, *a2*, and *a3* to files *b1*, *b2*, and *b3* respectively, enter this command:

```
-copy a1, a2, a3 to b1, b2, b3 <CR>
```

The HI responds:

```
a1 copied to b1
a2 copied to b2
a3 copied to b3
-
```

You can also use wildcards when copying files. If the files *a1*, *a2*, and *a3* are the only files in the directory that begin with the character *a*, you can use this command to perform the same operation:

```
-copy a* to b* <CR>
```

The HI responds:

```
a1 copied to b1
a2 copied to b2
a3 copied to b3
-
```

Replacing Existing Files

Use either the **rename** command or the **copy** command's `over` preposition to update existing files.

To copy new data over an existing file, first create a file named *temp*, then enter:

```
-copy temp over alpha <CR>
```

The HI responds:

```
temp copied over alpha
```

You can also use the `to` preposition, but you get a warning message:

```
-copy temp to alpha <CR>
```

The HI responds like this, and you reply with `y`:

```
alpha, already exists, OVERWRITE? y <CR>
temp copied to alpha
-
```

You now have two copies of the same new data; one in the *temp* file and one in the *alpha* file. If you use the `over` preposition in a **rename** command instead of the **copy** command, the *temp* file is deleted automatically when **rename** is executed.

Concatenating Files

Concatenation is combining a number of files by appending them in sequence into a single file. Use the **copy** command to concatenate files by specifying the `after` preposition or by specifying multiple input pathnames and a single output pathname. (If the output pathname is a directory, concatenation does not occur).

Use the `after` preposition as follows:

```
-copy b,c,d after a <CR>
```

The HI responds:

```
b copied after a
c copied after a
d copied after a
-
```

To concatenate all four files into a new file called *all*, enter:

```
-copy a,b,c,d to all <CR>
```

The HI responds:

```
a copied to all
b copied after all
c copied after all
d copied after all
-
```

In this last example, file *a* is copied to *all* and the remaining input files are automatically appended to the end of *all*.

You can also concatenate files displayed on the screen. To list files named *alpha*, *beta*, and *gamma* to the default file *:co:* (the output device, or screen), enter:

```
-copy alpha,beta,gamma <CR>
```

The HI responds:

```
aaaaa
aaaaa
alpha copied to :co:
bbbbb
bbbbb
beta copied after :co:
ggggg
gamma copied after :co:
-
```

Renaming Files

To change the name of file *alpha* to the new name *omega*, where *omega* does not already exist, enter:

```
-rename alpha to omega <CR>
```

The HI responds:

```
alpha renamed to omega
-
```

The *alpha* pathname is automatically deleted from the system when the **rename** command executes.

You can rename lists of files to new pathnames. When using the **rename** command, you must always have a one-for-one match of pathnames between the old file list and the new list of names.

You can also use the `over` preposition with **rename**. The primary purpose of `over` is to move data from one named file and overwrite the data in another existing file. This use of the `over` preposition matches the action of the `over` preposition in the **copy** command, with one important distinction: **rename** automatically deletes the input file when the command is executed.

Deleting Files

To delete files *alpha*, *beta*, and *key* from the system, enter this command:

```
-delete alpha,beta,key <CR>
```

The HI responds:

```
alpha, deleted
beta, deleted
key, does not exist
-
```

The error message for the *key* file tells you one of three things:

- There is a spelling error in the name of the file.
- The file does not exist.
- The file exists in a different directory than where you are currently working.

Using Devices

The list below shows which kinds of devices you can communicate with and the types of files supported on those devices.

Terminals Use terminals to communicate with the HI. You can also write programs that read from and write to terminals. Terminals are accessed as physical files.

Disks Disks provide permanent storage for programs and data. You can communicate with hard disks and diskette drives. During **format**, **backup**, **restore**, and **diskverify** operations, disks are treated as physical files. Other operations can access named files.

RAM disk A RAM disk provides an area of memory that acts like a secondary storage device in the system. The RAM disk creates a disk image within your memory and provides faster access to your files than a physical disk. It is similar to a physical disk in all aspects except that it does not provide permanent storage. If there is a power failure, the system fails for some reason, or you turn the power off, the data on the RAM disk is lost.

Tapes Tapes provide a quick and convenient method of backing up hard disks. You can use tape drives only for backing up and restoring files using the **backup** and **restore** commands. The OS does not support tape drives for any other uses.

See also: File types, *Introducing the iRMX Operating Systems*

Device Names

Except for devices attached by the system as a result of the system configuration, you must attach a device with the **attachdevice** command before using it. In the **attachdevice** command, you specify a physical device name that defines device characteristics. Device names are provided for standard configurations. In a custom configuration you may define new device types (and names) or remove standard ones. You can also use the **sysload** command with a loadable device driver to add new device types.

For terminal devices, you also specify physical device names in the **lock**, **unlock**, and **connect** commands.

See also: **sysload**, **attachdevice**, *Command Reference*

Switching Diskettes

When you remove an iRMX-formatted diskette from the drive and then access the drive device, any connections to files on that device are deleted. Any logical names that represent files on the volume are no longer valid. The names remain cataloged in the directories, but they do not represent valid connections.

See also: Creating Logical Names for Devices, *Command Reference*

To maintain your data in usable form, detach a diskette before you remove it, and then attach a new diskette before you use it. Use this procedure:

1. Before removing the first diskette, invoke the **detachfile** command to detach the files, or use the `force` parameter in the **detachdevice** command.
2. Remove the first diskette and insert the replacement diskette.
3. Invoke the **attachdevice** command (or the alias **ad**) to gain access to the new diskette.

⚠ **CAUTION**

If you do not follow this procedure, the OS assumes that the first diskette is still in the drive. The volume directory is written only during the **detachdevice** command and directory changes made to the volume are not written if you remove the diskette before the **detachdevice**. Furthermore, if you change diskettes without a **detachdevice**, access the new diskette (using **dir**, for example), then detach the device, the directory from the former diskette is written to the current volume, destroying the file system on it. This also applies to remote diskette volumes managed by the iRMX named file driver.

Uniform and Standard Granularity Diskettes

Modern diskette formats use *uniform granularity*. This means the diskette has a consistent number of bytes per sector on every track. *Standard granularity* is an older iRMX diskette format that uses 128 bytes per sector (single density) on track 0, regardless of the sector density used on the rest of the volume. Older releases of the OS were distributed on this format diskette, and you may still use standard granularity diskettes.

On a standard granularity device, the iRMX BIOS can recognize the type of physical device from information written on track 0 during the format operation. If the diskette does not match the physical device type specified in the **attachdevice** command, the BIOS attaches an appropriate device unit information block (DUIB) for the device. The BIOS cannot do this, however, when you switch diskettes in a 5.25-inch or 3.5-inch drive without detaching the device. If you use a Multibus I system with 8-inch drives, you can switch diskettes without a **detachdevice**, because the 8-inch drive interrupts the OS when the diskette is changed.

Because the device characteristics recognition feature does not apply to physical-file operations, the **format**, **backup**, and **restore** commands do not use this feature.

Using the Online Command Help

Use the command help at the iRMX prompt to obtain information about most iRMX commands. You can get help by typing:

```
help command_list [to|over|after outpath_list] [q] [p = num]
```

Where:

`command_list`
: One or more command names for which you want help. Separate command names with commas. Wildcards are permitted.

`to|over|after outpath_list`
: Writes the help file to the specified file(s) rather than to the screen. If you specify multiple help files and one output file, the output is appended.

`q(uery)` Prompts for permission to display each help file.

`p(agelength) = num`
: The maximum number of lines in the output page.

See also: **help** command, *Command Reference*

Using Online iRMX System Call Help

A Windows Help file is available for all system calls, **cq_** calls from the *Network User's Guide and Reference*, and all exception codes. The help program runs only under Windows, not the iRMX OS.

After installing the iRMX OS, the *rmxcall.hlp* file is located in the *\dosrmx* directory on the hard disk.

To install *rmxcall.hlp* on any PC running DOS/Windows without performing an iRMX installation procedure, copy the file from the iRMX installation CD to your hard disk. The file is in the *\install\common\dosrmx* directory on the CD.

Once you have the Help file on your hard disk, you can set up an icon for it in Windows by choosing **New.../Program Item** from the **Files** menu in the Program Manager. Or, double-click on *rmxcall.hlp* from the File Manager to start the Help file.

Using Online iRMX Manuals

➥ **Note**

If you installed the iRMX III.2.3 OS on a Windows NT 4.0 Cross Development system following the procedures in Chapter 6, the Online iRMX Manuals will already have been installed in the \iRMX\manuals directory.

This release of the OS provides a complete set of iRMX manuals online in Adobe Acrobat format. The manuals are on the installation CD in the *\manuals* directory. Also on the CD is a special version of the Acrobat Reader that allows full-text search across the iRMX manual set. After you install the Reader you can access the files (Portable Document Format files) directly from the CD or from a location on your hard disk, if you choose to copy them to the hard disk.

You can view the manuals online, search for text in them, and print any part (or all) of the manuals from the Reader.

Installing the Reader

The Acrobat Reader shipped with the iRMX CD supports the Acrobat Reader Search command. With this command you can perform full-text searches of collections of the iRMX documents. Even if you have already installed a version of the Acrobat Reader, you must install this new version to support full-text search.

Unlike the Find command of a word-processing program, which searches for words by reading every word on every page, the Search command searches full-text indexes included with the electronic manuals. The full-text indexes represent the entire iRMX online documentation set. The Search command uses the index to find words and terms quickly in the iRMX manual set without having to open the documents.

To install the Reader, perform these steps:

1. Insert the iRMX 2.3 CD into the CD drive.
2. From Windows, run ar302.exe from the *\manuals* directory on the CD. For example, if the CD is drive E:, in the Program Manager, choose **Run...** from the **File** menu, and enter this command:

   ```
   e:\manuals\ar203
   ```

3. Read the agreement that appears in the Electronic End User License Agreement Screen and press the Accept button.
4. From the Acrobat Installer screen:

   – Be sure the Acrobat Search check box is selected to get the full-search capability within the iRMX manual set. You will need approximately 4.7 Mbytes on your hard disk to perform the installation with the search capability activated.

   – Enter the path on your hard disk where you want the Reader to be installed, for example, c:*\userbob\acrosrch.*

5. Accept the Windows information displayed for Name, Organization, and Serial Number.
6. Wait for the installation to complete.

After the installation completes, you will have an Adobe Acrobat program group with two items: the Reader and a “readme” file.

Start the Reader and view the *help_r.pdf* and *help_s.pdf* files in the *help* subdirectory under the directory where you installed the Acrobat Reader. These files describe how to use the Reader and the search facility, respectively.

You can use this version of the Reader to view Acrobat-format documents (.PDF files) for any product, not just for the iRMX OS.

Accessing the Online Manuals

You can view the iRMX manuals directly from the CD or you can copy the files from CD to a directory of your choice on the hard drive.

By accessing the manuals directly off the CD you save hard disk space but can sacrifice access time.

By installing the files to your hard disk you can gain better performance when viewing manuals. To copy the files to your hard disk enter this command:

```
xcopy e:\manuals\*.* \d:\<new_directory> /s/e
```

where `e:` is the CD-ROM drive, `d:` is your hard disk, and `<new_directory>` is the destination path.

The *\manuals* directory contains **.pdf* files and a *.pdx* file. There is also a *readme.txt* file with the latest information about using and installing this tool. The subdirectories under `\manuals` contain parts of the index tool used by the search engine in the Acrobat Reader.

➠ **Note**

Read the *readme.txt* file to make sure you have the latest information. To start viewing iRMX manuals, start the Reader and open *Welcome.PDF*.

□ □ □

Where To Go From Here 10

You can take several paths through the documentation. The figures that follow offer suggestions.

Figure 10-1 on page 142 displays the recommended order for new users. New users means that you're new to the iRMX OS.

If you've developed application programs with the iRMX OS, browse through *Introducing the iRMX Operating Systems*. Figure 10-2 on page 143 displays the recommended order for experienced users.

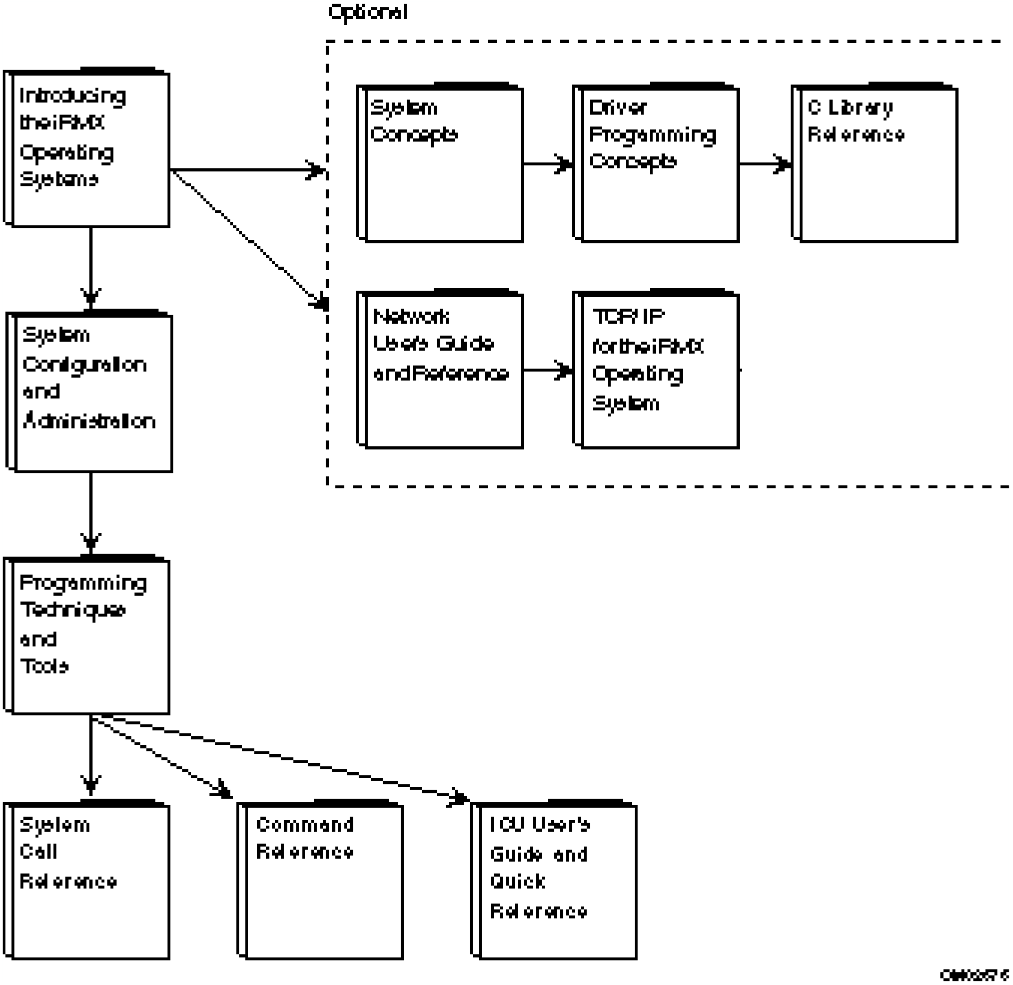
Optional
Introducing the iRMX Operating Systems
System Concepts
Driver Progamming Concepts
C Library Reference
Network User's Guide and Reference
TCP/IP for the iRMX Operating System
System Configuration and Administration
Progamming Techniques and Tools
System Call Reference
Command Reference
ICU User's Guide and Quick Reference

Figure 10-1. Recommended Documentation Roadmap for New Users

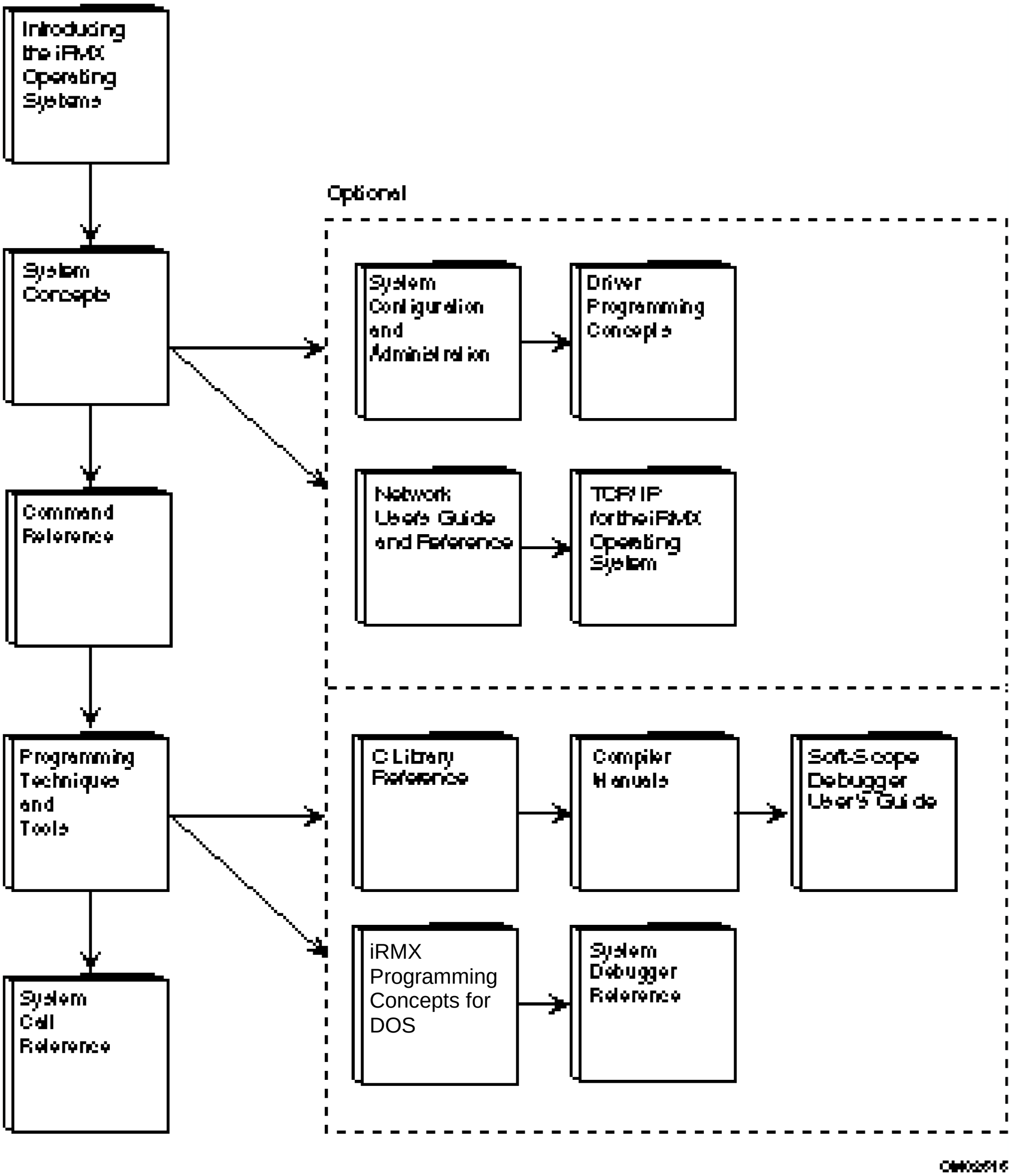
Introducing the iRMX Operating Systems
System Concepts
Command Reference
Programming Techniques and Tools
System Call Reference
Optional
System Configuration and Administration
Driver Programming Concepts
Network User's Guide and Reference
TCP/IP for the iRMX Operating System
C Library Reference
Compiler Manuals
Soft-Scope Debugger User's Guide
iRMX Programming Concepts for DOS
System Debugger Reference

Figure 10-2. Recommended Documentation Roadmap for Experienced Users

❑ ❑ ❑

iRMX INSTALLATION APPENDICES

This section contains information that lets you get started with your iRMX OS. These are the chapters in this section:

Appendix A.	**Installed Directories**
Appendix B	**Limitations**
Appendix C	**Configuration Requirements for PC Platforms**
Appendix D	**Changing Memory Size During the iRMX III OS Installation**
Appendix E	**Hardware Modifications for the iRMX III OS Installation**
Appendix F	**Error Messages**

Installed Directories A

This appendix shows the installed directory structure for all versions of the OS: DOSRMX, iRMX for PCs, and the iRMX III OS.

Directory Structure

Following are key directories resulting from the installation of iRMX OS. The column to the left gives the name of the directory (shown hierarchically) and the column to the right provides a brief description of the directory's contents.

Directories	**Comments**
:sd:	
boot32	Multibus I boot images
bsl	Multibus I boot generation and configuration files
dosrmx	DOS-specific files including Windows Help
ins_sys	iRMX MBI Installation system commands
intel	
bin	AEDIT, Intel386 utilities
include	C header files (include files)
arpa	C include files specific to TCP/IP protocols, e.g., FTP
net	C include files for networking not specific to TCP/IP
netinet	C include files specific to TCP/IP protocols, e.g., TCP
sys	System-related C include files
lib	C-start objects, C-library interface libraries
ndp387	Math coprocessor utilities
gen	Generation files
src	C-start objects for ASM
lang286	Compiler and utility executable files
msa	Default MSA files
boot	Default MSA boot image file
config	Default MSA BPS file
msa32	iRMX III MSA boot files
386258	BPS files for 386/258 and 386/258D
486133	BPS files for 486/133SE and 486/166SE
p5090	BPS files for P5xxx family of boards
boot	MSA boot images

```
:sd:
    net                   iNA 960 load files
    rboot32               Multibus I remote boot files
    rmx386                Readme text files
        clib              C run-time system libraries
        config            iRMX configuration files
        demo              Example files (includes source files)
            altsys
            asm
            c
            plm
            rom
            rte
            sscope
        drivers           Loadable device driver files
        early             New but unintegrated software
        eios              EIOS library files (used in generating boot images)
        help              Help files for iRMX commands
        hi                Human Interface (HI) library files
        icu               ICU files (screen master, template, registration, and .bck)
        inc               P/LM 32-bit external declaration and literal files
        inc16             P/LM 16-bit external declaration and literal files
        ios               I/O system library files (used in generating boot images)
        jobs              Loadable system jobs
        lib               Interface libraries for OS
        loader            Application Loader library files (used in generating boot images)
        nucleus           Nucleus library files (used in generating boot images)
        network           iNA 960 network library files (used in generating boot images)
        obsolete          Various files (source and binary) from iRMX III.2.2 which are now
                           obsolete
        sdb               SDB library files (used in generating boot images)
        sdm               SDM library files (used in generating boot images)
        udi               UDI library files (used in generating boot images)
        unsupprt          Unsupported software features
        vm86              VM86 library files (used in generating boot images)
    sys386                Commonly used Intel-provided command files
    system                Default Multibus I boot image file
    user          User directory
        super             Super user :home:directory
            prog          Super user :prog: directory
        world             World user :home:directory
            prog          World user :prog: directory
    util286               HI utility commands (16-bit)
    util386               HI utility commands (32-bit)
    work                  Temporary work files
```

Multibus I Minimum File Structure Needed

This directory structure is the minimum needed to boot the iRMX III OS from a Multibus I system.

Directories and *Files*

:sd:
.   sys386
.   system
.   .   *rmx86*
.   .   *rmx86.386*
.   rmx386
.   .   config
.   .   .   *udf*
.   .   .   *terminals*
.   .   .   *logon.msg*
.   .   .   *termcap*
.   .   .   user
.   .   .   .   *super*
.   user
.   .   super
.   .   .   prog
.   .   .   .   *r?logon*
.   work
.   util286
.   util386
.   lang286

Multibus II Minimum File Structure Needed

This directory structure is the minimum needed to boot the iRMX III OS from a Multibus II system.

Directories and *Files*
:sd:
. msa
. . *stage2.rmx*
. . boot
. . . *rmx*
. . . *rmxio*
. . config
. . . *bps*
. sys386
. util286
. util386
. lang286
. rmx386
. . config
. . . user
. . . . *super*
. . . *udf*
. . . *terminals*
. . . *logon.msg*
. . . *termcap*
. user
. . super
. . . prog
. . . . *r?logon*
. work

□ □ □

Limitations B

This appendix presents general and area-specific limitations for the DOSRMX version of the OS as well as other limitations that are common to all versions of the OS. Review this appendix to make sure your application and software configuration is compatible.

DOSRMX General Limitations

- The OS has been validated with DOS versions 5.0 through 6.22.
- The OS does not support any version of Windows.
- The OS does not support DPMI (DOS protected-mode interface) applications, such as those built with Phar Lap's DOS Extender. You cannot start DOSRMX with such an application running, nor can you start a DPMI application after starting DOSRMX.
- DOS applications that remotely control another PC or that manage the system timer are incompatible with the OS, and are not supported.

DOSRMX-specific Limitations

DOSRMX has limitations in these specific areas:

- Extended memory usage
- Expanded memory usage
- Ill-behaved DOS applications
- DOS application installation
- Real-time performance issues
- Networking and Communication
- DOS debuggers
- Software interrupts
- Hardware sharing

- Priority level
- Invisible files
- Last cylinder usage
- Printing
- EDOS File Driver
- RTE Calls

These limitations are described in the subsections that follow.

Extended Memory Usage

DOSRMX provides support for DOS extended memory; however, there are restrictions on what you can load into extended memory. These restrictions are due to the order of loading DOS drivers and loading the iRMX extended memory manager.

See also: Chapter 6 for more details

Expanded Memory Usage

If an application requires expanded memory, provide hardware expanded memory, such as an Intel Above Board, not expanded memory emulated in extended memory.

Ill-Behaved DOS Applications

DOS application programs which use or do the following are not supported:

- Reprogramming the Programmable Interrupt Controller (PIC)
- Accessing the timer in ways other than trying to re-program it to a different clock rate, or trying to calibrate programs by using the timer to measure elapsed time. Both of these attempts are handled by the operating system. Other access attempts will cause unpredictable results, such as breaking to the SDM monitor.
- Calling software interrupt handlers directly instead of using the INT instruction
- Relying on 64 Kbyte wrap-around for Code and Data segments
- For TSR's, not following DOS reentry protocols or chaining improperly to interrupt vectors

DOS Application Installation

Any DOS application that requires resetting the must be run prior to loading DOSRMX. Even if you have shut down the OS, reset the system before running such software.

Real-time Performance Issues

DOS applications are prevented from disabling iRMX interrupts indefinitely. Virtual interrupts are used to provide correct behavior for DOS without impacting the iRMX OS.

The DOS clock interrupt handler can take an indefinite amount of time, especially if there are many TSR's chained to it. The DOS handler needs to do the end-of-interrupt to the PIC within a millisecond, even if the call does not return for a longer duration.

Networking and Communication

DOS applications performing disk I/O or communication I/O, such as XMODEM data transfers or disk testing software, can fail if the applications are sensitive to timing and are competing with the iRMX OS for processor time.

Since the EDOS file driver removes *r?* from the beginning of filenames before sending them across the network, you may need to duplicate invisible iRMX named files that reside on an iRMX partition (that is, have one file named *r?logon* and another named *logon*) before you can access them via DOS networks.

iRMX filenames sent across DOS networks are truncated to eight characters and a three character extension even if the remote host is an iRMX partition and the named file driver is used.

You cannot start the DOS version of MS-NET software with the normal `net start rdr` command and then start DOSRMX with its network software. Instead, use the *pcnet* version of MS-NET software, with the iRMX *netrdr* job. You can start Novell software before starting DOSRMX.

See also: Using iRMX-NET in a DOS Environment, *System Configuration and Administration*

Booting the OS from files on a Novell server is not supported. If you start DOS and Novell networking, then attach to the Novell server as drive E:, for example, do not boot DOSRMX from drive E:. The length of filenames under iRMX is incompatible with filename length as managed on the Novell server.

DOS Debuggers

Some DOS debuggers support both Real Mode and Protected Mode. Use Real Mode for debugging DOS application programs.

Software Interrupts

Certain software interrupts are reserved and not available for use by DOS application programs.

Hardware Sharing

These iRMX-encoded hardware interrupt levels are reserved and not available for any other use:

- For PC compatible networking systems, PIC Slave Level 1 is reserved for a PCLINK2(A) board.
- For Multibus I networking systems, PIC Master Level 5 is reserved for the SBX 586.
- For Multibus II networking systems, PIC Slave Level 1 is reserved for the MPC.

Accessing Serial Devices

DOS applications can access any serial device not attached by the iRMX OS. However, once a serial device is attached by iRMX, it is not available to DOS. If you use a serial device from DOS, do not enter the device name in the terminal configuration file or attach it with the **attachdevice** command.

See also: Configuration of Local Terminals, *System Configuration and Administration*

Priority Level

DOS runs at priority 254: an iRMX task running at 255 will not execute. This also implies that if any iRMX task running at a priority higher (numerically lower) than 254 is in a tight polling loop, that task will prevent the DOS task from ever executing. Your iRMX application program must not include tight polling loops.

A low priority iRMX task (for example, priority=200) in a tight loop can indirectly keep a higher priority task (for example, priority=141) from running, if the higher-priority task needs I/O from the EDOS file driver or the ROM BIOS. This is because an I/O task that uses EDOS or the ROM BIOS has to wait for DOS to be safe before initiating I/O. If the low-priority iRMX task preempts DOS while it is in a "DOS-busy" state (during an INT 21 or INT 13), DOS never gets a chance to complete and become "safe" so the high-priority iRMX task can perform its I/O. Therefore, if you must poll in an iRMX task, have the task either periodically release the CPU using an **rq_sleep** call, or run the task round robin with DOS at priority 254.

When you create users with the Command Line Interface (CLI), set the maximum task priority at one less than the DOS priority (that is, 253) in the *:config:user* file. Then those users will share the processor with DOS. If you use **submit**, set the task priority to one less than DOS. If you use **background**, set the task priority to three less than DOS. If you use all three, set the task priority to five less than DOS.

See also: User attributes files, *System Configuration and Administration*

Invisible Files

In the iRMX OS, hidden files begin with *r?*. The ? (question mark) character is not valid in a DOS file name so iRMX filenames containing a ? are not accessible to DOS. The EDOS file driver uses the following convention:

- Files on the iRMX volume that begin with *r?* are created on the DOS volume without the *r?*, but with the DOS invisible attribute.
- Files on the DOS volume that have their invisible bits set, appear to the iRMX OS as beginning with *r?*.

For example, if you copy the file *r?login* from an iRMX partition to a DOS partition, the file will appear as *login* with the invisible attribute. DOS invisible files such as *msdos.sys* appear as *r?msdos.sys* when you list them from iRMX using the EDOS file driver.

On the iRMX volume, you can have two versions of the same file, one invisible and one visible, with the same name, for example *r?login* and *login*. On DOS, you can have only one version of a file, invisible or visible.

Last Cylinder Usage and RDISK

Rdisk and the DOSRMX hard disk device driver are ROM BIOS-based. Certain implementations of the ROM BIOS do not provide access to the last cylinder on the disk. If you have an existing partition table that includes the last cylinder in one of the partitions, and you invoke **rdisk**, it reports:

```
Invalid Partition in Partition Table
```

If the partition table still contains the last cylinder in one of the partitions and you attempt to write the partition table to the disk, **rdisk** reports the same error message and will not perform the write.

If you wish to create an iRMX partition on this disk, perform the following steps to remove the last cylinder and preserve the contents of the partition.

1. Back up the contents of the partition that contains the last cylinder.
2. Resize the partition so it does not contain the last cylinder.
3. Format the resized partition to modify the FAT (for DOS partitions) or bit map table (for iRMX partitions).
4. Restore the files back onto the partition.

Printing

There is no print spooler. Output sent to the :lp: device from one job can interrupt printing from another job or interrupt printing from a DOS/Windows application. The print output is mixed.

The <Print Scrn> key is not supported from the iRMX OS as it is in DOS. You cannot use the <Print Scrn> key to print the display from iRMX.

EDOS File Driver

These limitations or behaviors exist for the EDOS file driver:

- You cannot open more than 55 EDOS files at a time. Attempting to open more results in an E_IOMEM exception.
- An application that uses the EDOS file driver or I/O from the ROM BIOS can be blocked by a lower-priority task. Assume that a low priority iRMX task (priority = 200) is in a tight loop. If a higher-priority task (priority = 141) needs I/O from EDOS or the ROM BIOS, the I/O task has to wait for DOS to be safe before initiating I/O. If the low-priority task preempts the DOS task while DOS is in a “DOS-busy” state (i.e., in the middle of an INT 21 or INT 13), the DOS task never gets a chance to complete. Thus the low-priority task can block the high-priority task that needs DOS I/O. If you must poll in an iRMX task, have the task either periodically release the processor (using an **rq_sleep** call), or run the task round-robin with DOS at priority 254.
- The EDOS file driver treats various file access conditions differently than the named file driver. The following system calls can return unexpected condition codes with the EDOS file driver:
  - **rq_s_delete_file** may return E_FNEXIST (21H) instead of E_FACCESS (26H)
  - **rq_s_rename_file** may return E_FNEXIST (21H) instead of E_FACCESS (26H)
  - **rq_s_rename_file**, when renaming a data file to which World has read-only access, may return E_OK instead of E_FACCESS (26H)
  - **rq_s_get_path_component,** when a deleted file is specified, may return E_OK instead of E_FNEXIST (21H).
- iRMX for PCs/Windows EIOS buffer sizes are configurable in the *rmx.ini* file on a per-file driver basis. The default buffer size is 1K for the Named, EDOS, DOS, and Remote file drivers. Increasing the default size can provide an increase in the performance of sequential I/O, but will not enhance the performance of random I/O. Also, the memory tradeoff must be considered, since buffers are allocated on a per-connection basis system-wide.

Table C-1 below summarizes the results of an I/O benchmark which performs sequential writes to a single disk file. This benchmark sequentially writes out blocks of data (using **fwrite**) of size "Transfer Size" until the "File Size" is reached. The table relates EIOS buffer sizes to the time it takes to write the entire file. Times are in seconds. DOS timings for the same benchmark are included for completeness. Hardware was an Intel SYP403E (33MHz 486) with a Maxtor XT-8760S SCSI drive controlled by an Adaptec 1542B SCSI controller. When comparing the numbers, the lower the number, the better the I/O performance.

Table B-1. Sequential I/O Performance Benchmark

Transfer Size	File Size	1K Buffers	2K Buffers	4K Buffers	DOS Plus iRMX	DOS Only
128	256K	4	3	2	9	9
256	512K	9	6	5	17	17
512	1M	19	11	10	35	35
1K	2M	37	21	20	37	36

Note

The I/O benchmark was run on the Named, EDOS, and DOS file drivers with essentially the same results. The EDOS file driver is slightly slower due to the overhead of DOS system calls.

- On DOSRMX with short circuit iRMX-NET configurations, offering any local EDOS file driver devices as public directories when accessing from the DOS side of the system will result in a system hang.

RTE Calls

These iRMX system calls do not work properly when invoked as real-time extension (RTE) calls from DOS.

- **rq_sleep** can hang DOS.
- Using timeouts with mailbox and semaphore operations (**rq_receive_message**, **rq_receive_data**, and **rq_receive_units**) can also hang DOS.
- Using regions (**rq_accept_control** and **rq_send_control**) can temporarily raise the DOS task’s priority.

Message Passing Limitations

The PC-compatible Multibus II processor boards SBC 486DX33 and SBC 486DX66 have the following Nucleus Communication System message passing limitations.

`Data chains are not supported as message passing buffers.`

The DMA hardware expects solicited message buffers to meet address alignment restrictions: the physical address of the buffer must be DWORD aligned (two LSBs of address are 0). Alignment buffers are used to force alignment of misaligned data, one each for input and output. The size of these buffers is configurable, in the ICU adjust the values of DIB and DOB on the MBII screen, or in the *rmx.ini* file set these parameters to at least the size of the largest unaligned solicited message. Attempts to send/receive from/to unaligned buffers larger than the configured alignment buffer size results in an E_NUC_BAD_BUF exception. The iRMX Free Space Manager guarantees that all segments are created with 16-bit aligned physical addresses (four LSBs are 0), so any iRMX segment used as a buffer will work (e.g., buffers in a buffer pool).

Delete_task or Suspend_task Deadlock Problem

Deletion or suspension of an iRMX task that has a VM86 interrupt running in its context will cause a deadlock.

A deadlock will occur in the following scenario:

1. The current iRMX task (Task A) is interrupted by a DOS hardware interrupt.
2. An iRMX hardware interrupt (which could be the system clock interrupt) occurs before the DOS interrupt handler completes. This situation causes a task switch to iRMX Task B that leaves Task A ready but not running.
3. Task B deletes or suspends Task A.

To avoid this problem, do not use **suspend_task** except to suspend the calling task. The preferred method of task synchronization is to use mailboxes, ports, regions, and semaphores.

See also: Mailboxes, Ports, Regions, and Semaphores, *System Concepts*

❑ ❑ ❑

Configuration Requirements for PC Platforms C

This appendix lists configuration information for the following PC hardware:

- RadiSys PCL2(A) board
- Intel iSBC 486DX33 and 486DX66 Multibus II cards
- 4, 8, and 16 channel HOSTESS 550 terminal controller PC peripheral cards
- Digiboard terminal controller PC peripheral cards
- See also: The hardware reference manual for your particular board.

RadiSys PCL2(A) Board

Table C-1 lists the jumpers for the PCL2(A) board.

Table C-1. Intel PCL2(A) Network Interface Adapter Board

Network Interface Adapter	Jumper Changes For Networking
PCL2(A) NIA	
I/O Port 360H	Install E2-E3
Interrupt IRQ2 (iRMX Int 21H)	Install E11-E12

See also: Interrupt management, *System Concepts*

Intel SBC 486DX33 and 486DX66 Platforms

Table C-2 lists boards that can be used with SBC 486DX33 and 486DX66 platforms.

Table C-2. SBC 486DX33/486DX66 Platform

CPU Board	Jumper Changes
SBC 486DX33/486DX66	None
Peripheral Controller and Network Interface Adapter Board	
SBC 486/133SE Single Board Computer	None
MIX 386/560	None
MIX n86/020(A) with MIX 560	None
SBC 186/530	None
Serial Controller Board	
Up to four SBC 186/410	None
Up to four SBC 186/450	None
MIX n86/020(A) with up to three MIX 450	None

HOSTESS 550 Terminal Controller

DOSRMX supports 4, 8, and 16 channel HOSTESS 550 terminal controller boards. Table C-3 lists the jumper changes for the HOSTESS 550 eight port terminal controller board in order to have an I/O address of 280H and an interrupt level of 58H.

See also: The board documentation for the 4 and 16 channel switch settings

Table C-3. HOSTESS 550 Terminal Controller

HOSTESS 550 Terminal Controller	Switch Settings
Switch 1	5, 3, 1 OFF 8, 7, 6, 4, 2, ON
Switch 2	8, 7, 6, 4, 3, 2, 1 OFF 5 ON

Digiboard Asynchronous Communications Boards

Table C-4 lists supported Digiboard terminal controllers.

See also: The Digiboard manual for your board

Table C-4. Digiboard Switch Settings

Digiboard	Switch Settings
DigiBoard Digichannel PC/4 DigiBoard Digichannel PC/8 DigiBoard Digichannel PC/16	*Refer to the Digiboard manual with each Digiboard*

❑ ❑ ❑

Changing Memory Size During the iRMX III OS Installation D

Changing the Memory Size

When you install over a previous iRMX OS release and you use old definition files, the size of the operating system may increase beyond the amount of memory configured for it. If so, you will receive this error message:

```
ERROR 192:  NO SPACE FOR SEGMENT - BASE NOT SET
```

This example shows how to increase the memory size for the operating system by 8000H bytes.

➠ **Note**
The example in this appendix uses a specific definition file, *p90s.bck*. This example is appropriate for systems whose I/O Server board is an SBC P5090 board. If your system has a different processor board, substitute every reference in the definition file with the appropriate `os_name`. The current directory *:home:os_name* is assumed. (See the Generating a Default Operating System step in either Chapter 4 or 5.)

1. Examine the segment map portion of the builder's map file, */msa32/boot/p90s.mp2*. If a segment's base is not set, it will have a string of dashes instead of a physical address. To determine how much space you require, add up the segment lengths. The example uses 8000H.

2. Invoke the ICU:

```
- icu386 p90s.bck <CR>
```

See also: *ICU User's Guide and Quick Reference* for complete instructions on using the ICU and a list of definition files provided with the OS

3. Display the MEMS screen.
4. Add 8000H to the high address.
5. Display the MEMF screen.
6. Add 8000H to the low address.
7. Generate the system.
8. Save the changes made to the definition file and exit the ICU.

❑ ❑ ❑

Hardware Modifications for the iRMX III OS Installation E

This appendix contains tables of jumper configurations for your hardware so it can run the iRMX III OS (as defined by the ICU definition files). Each table lists all the changes you need to make to a particular processor board.

See also: The hardware reference manual for each board;
standard definition files, *ICU User's Guide and Quick Reference*

How To Use This Appendix

To use the information in this appendix, follow these steps:

1. Locate the section that describes your board:
    a. Processor Boards
    b. Peripheral Controller Boards
    c. Communications and I/O Controller Boards
    d. Network Controller Boards
    e. Non-Intel Controller Boards
2. Remove the jumpers listed in the Remove Jumper column of the jumpering table.
3. Install the jumpers listed in the Install Jumper column.
4. Make the changes to the switch settings listed in the switch table if your board has switches. The changes to switch settings are listed in a separate table for each board.
5. Install any required components.

➾ **Note**
The jumpering changes in this appendix describe the modifications that must be made to the factory-installed jumpers; not the default jumper configuration. Refer to the individual board's hardware reference manual for a list of the factory-installed jumpers.

Multibus I CPU Boards

This section lists the hardware modifications for these Intel Multibus I CPU boards:

SBC 386/12
SBC 386/12S
SBC 386/2x/3x
SBC 486/12S

SBC 386/12 CPU Board

No jumpering changes are required to run the iRMX III OS. This board supports a range of memory devices.

See also: The board's hardware reference manual for details on how to select memory types

You must use the standard definition files for the SBC 386/12 board as a starting point for building a custom version of the operating system to run on the SBC 386/12 board. These definition files select only the MSC device driver. They do not support SCSI peripherals.

SBC 386/12S CPU Board

Note

This release of the OS requires that you install firmware version V1.6, X127 or later on an SBC 386/12S board. A version of firmware that works with these SBCs is included on the Multibus Firmware diskette. Only install this firmware if it is newer than the firmware currently on your SBC.

When you boot the OS, this message may appear:

```
Break at xxxx:yyyyyyyy
..
```

If this occurs, enter `g <CR>` to boot the system. To avoid this error message, allow the firmware memory tests to complete after a system reset or power-up.

No jumpering changes are required to run the iRMX III OS. This board supports a range of EPROM devices. Refer to the board's hardware reference manual for details on how to select the EPROM type.

You can use the standard definition files *38612S.bck* and *38612Snet.bck* as a starting point for building a custom version of the operating system to run on the

SBC 386/12S board. These definition files select both the SCSI device driver and the MSC device driver (for ST506 and ESDI peripherals).

SBC 386/2x/3x CPU Boards

There are minimum changes to the default jumpers required to execute the iRMX III OS. After making the changes, you can run the iRMX III Start-up System for System 320 Microcomputers.

See also: The board's hardware reference manual for details on how to select memory types

This release of the iRMX III OS provides preconfigured EPROM images containing board diagnostics, the SDM monitor, and the iRMX Bootstrap Loader for this board.

You can use the standard definition files for the SBC 386/2x/3x boards as a starting point for building a custom version of the operating system.

Table E-1 SBC 386/2x/3x CPU Boards

Function	Remove Jumper	Description
Numeric Coprocessor	E11-E12	Remove this jumper when installing the Intel387™ math coprocessor

SBC 486/12S CPU Board

Note

This release of the OS requires that you install firmware version V1.6, X127 or later on an SBC 386/12S board. A version of firmware that works with these SBCs is included on the Multibus Firmware diskette. Only install this firmware if it is newer than the firmware currently on your SBC.

When you boot the OS, this message may appear:

```
Break at xxxx:yyyyyyyy
..
```

If this occurs, enter `g<CR>` to boot the system. To avoid this error message, allow the firmware memory tests to complete after a system reset or power-up.

No jumpering changes are required. This board supports a range of EPROM devices. Refer to the board's hardware reference manual for details on how to select the EPROM type.

You can use the standard definition files *48612S.bck* and *48612Snet.bck* as a starting point for building a custom version of the operating system to run on the SBC 486/12S board. These definition files select both the SCSI device driver and the MSC device driver (for ST506 and ESDI peripherals).

You can use the standard definition files *48612.bck*, *48612net.bck*, and *48612rsd.bck* as a starting point for building a custom version of the operating system to run on the SBC 486/12S board. These definition files select only the MSC device driver.

Multibus II CPU Boards

This section lists the hardware modifications for these Intel Multibus II CPU boards:

MIX 486?X33 (where ? is either S or D)
MIX 486DX66
MIX *n*86/020A (where *n* is either 3 or 4)
SBC 486/133SE
SBC 486/166SE

MIX 486?X33, MIX 486DX66, and MIX n86/020A CPU Boards

This section describes the minimum changes to the default jumpers. The jumpering is described in Table E-2.

You can use the standard definition files for the MIX486?x33, MIX 486DX66 and MIX n86/020a boards as a starting point for building a custom version of the OS.

Table E-2. MIX 486?X33, MIX 486DX66, and MIX n86/020A CPU Boards

		MIX Board	
Function	**Description**	**486?X33 and 486DX66 (Jumper Block 2)**	**n86/020A**
Bus arbitration mode	Explicit		
	Implicit (default)	J13-14	E2-E3
	Enhanced implicit	J14-16	E1-E2

The OS images generated from *486020a*.bck* files are not bootable on the MIX 386/020(A) processor boards, since *486020a*.bck* files include support for the numeric-coprocessor.

In the *386020a*.bck*, *486020a*.bck*, and *mix4.bck* files provided, the MPC duty cycle parameters (MCO, MCT, and MDC) in the MBII screen are set assuming the default bus arbitration mode of Implicit Mode. This mode is selected as the default arbitration mode for backward compatibility with the MIX 386/020 board. On the MIX 386/020A, MIX 486/020A, MIX 486SX33, MIX 486DX33, and MIX 486DX66 boards, the Enhance Implicit mode provides greater performance.

To select Enhanced Implicit mode on the MIX 386/020A and MIX 486/020A boards, change jumper E2-E3 to E1-E2, and change the MPC duty cycle parameters for the *386020a*.bck* and the *486020a*.bck* files to one of the Enhanced Implicit Mode values from Table E-3 (select one of 40% or 80% local bus bandwidth, depending on your application). For completeness, the Implicit Mode values are also provided.

Table E-3. Bus Arbitration Mode Selection for the MIX 386/020A, and 486/020A Boards

Bus Arbitration Mode	DMA Mode	Percent Local Bus Bandwidth	Duty Cycle Parameter
Implicit	One Cycle (MCO)	50	4FH
	Two Cycle (MCT)	50	94H
	Burst (MDC)	50	4BH
Implicit	One Cycle (MCO)	25	55H
	Two Cycle (MCT)	25	94H
	Burst (MDC)	25	50H
Enhanced Implicit	One Cycle (MCO)	85	1CH
	Two Cycle (MCT)	85	8AH
	Burst (MDC)	85	17H
Enhanced Implicit	One Cycle (MCO)	40	4CH
	Two Cycle (MCT)	40	8CH
	Burst (MDC)	40	49H

To select Enhanced Implicit mode on the MIX 486SX33, MIX 486DX33, and the MIX 486DX66 boards, change jumper J13-J14 to J14-J16 on jumper block J2, and change the MPC duty cycle parameters for the *mix4.bck* file to one of the Enhanced Implicit Mode values from Table E-4. For completeness, the Implicit Mode values are also provided.

Table E-4. Bus Arbitration Mode Selection for the MIX 486SX33, 486DX33, and 486DX66 Boards

Bus Arbitration Mode	DMA Mode	Percent Local Bus Bandwidth	Duty Cycle Parameter
Implicit	One Cycle (MCO)	50	52H
	Two Cycle (MCT)	50	97H
	Burst (MDC)	50	4AH
Enhanced Implicit	One Cycle (MCO)	85	4AH
	Two Cycle (MCT)	85	8CH
	Burst (MDC)	85	12H

See also: *MIX 386/020A and MIX 486/020A Baseboard User's Guide* for further explanation of these bus arbitration modes and duty cycles.

SBC 486/166SE and 486/133SE CPU Boards

This section describes the minimum changes to the default jumpers. The jumpering is described in Table E-5.

On some Maxtor SCSI hard drives, scanning does not detect SCSI Logical Units (LUNs) on a SCSI bus. Changing jumpers on the SBC 486/133SE and SBC 486/166SE boards overcomes this by scanning for SCSI LUN 0.

Table E-5. SBC 486/166SE and SBC 486/133SE CPU Boards

Function	Install Jumper
Detect SCSI logical units on a SCSI bus	E54-E58

SBC 386/258(D) Boards

This section describes the minimum changes to the default jumpers. The jumpering is described in Table E-6.

On some Maxtor SCSI hard drives, scanning does not detect SCSI Logical Units (LUNs) on a SCSI bus. Changing jumpers on the SBC 386/258 and SBC 386/258D boards overcomes this by scanning for SCSI LUN 0.

Table E-6. SBC 386/258(D) Boards

Function	Install Jumpers
Detect SCSI logical units on a SCSI bus	E20-E21 E28-E29

Peripheral Controller Boards

This section lists the hardware modifications you must make to these Intel Multibus I and SBX bus peripheral controllers:

Multibus I

SBC 208
SBC 214
SBC 215G
SBC 220
SBC 221

SBX Bus

SBX 217C
SBX 218A

SBC 208 Flexible Diskette Controller

The SBC 208 board can control either 5.25-inch or 8-inch flexible diskette drives. The drives support either single-sided or double-sided diskettes. In its default configuration, the SBC 208 board controls 8-inch diskette drives.

Table E-7. SBC 208 Board Jumper Changes

Function	Remove Jumper	Install Jumper	Description
Interrupt Level Selection		E79-E85	SBC 208 to Multibus INT4
I/O Address Size Decoding	E45-E49	E41-E45	Selects 16 bit address decoding
I/O Controller Address Selection	E48-E52 E61-E69	E44-E48 E53-E61	Selects port address 180H
Multibus Priority Selection			Default selects serial priority
	E77-E78		Selects parallel priority
5.25-inch Drive Selection	E18-E19	E4-E5 E6-E11 E17-E19	Selects 5.25-inch diskette drive
	E21-E22	E17-E22	Change these two jumpers if the 5.25-inch drive you are using is single-sided

SBC 214 Peripheral Controller

The SBC 214 board controls 5.25-inch ST506 style hard disks, diskette, and tape drives. The board does not support 8-inch devices.

Table E-8. SBC 214 Board Jumper Changes

	Remove	Install	
Function	**Jumper**	**Jumper**	**Description**
Multibus Priority Selection	no change	no change	Default setting selects parallel priority
	E101-E102	E100-E101 E95-E96	Selects serial priority

SBC 215G Hard Disk Controller

The SBC 215G board controls 5.25-inch or 8-inch ST506 hard disks. You must modify the default configuration to support any type of ST506 hard disks. The jumpering changes for mounting the SBX 217C Tape Controller module on the SBC 215G board assume that the SBX 218A is also mounted on the SBC 215G board.

Note
If this is not how you are configuring your SBC 215G board, you must refer to the hardware reference manual for additional jumpering instructions.

Table E-9. SBC 215G Board Jumper Changes

Function	Remove Jumper	Install Jumper	Description
Interrupt Level Selection			The default configuration connects the SBC 215G interrupt to Multibus INT5
I/O Address Size Decoding		W30, 2-19	Selects 16-bit addressing
Multibus Priority Selection	W28, 1-2		Selects parallel priority
	W23, 1-2	W23, 2-3	Selects serial priority
5.25-inch Drive Selection	W1, 1-2	W1, 2-3	Selects 5.25-inch hard disk drives
	W2, 1-2		
	W5, 1-2	W5, 2-3	
	W6, 1-2	W6, 2-3	
	W7, 1-2	W7, 2-3	
	W8, 1-2	W8, 2-3	
		W9, 1-2	
	W13, 2-3	W13, 1-2	
	W14, 1-3	W14, 1-2	
	W15, 1-2		
	W16, 1-3	W16, 1-2	
	W22, 1-3	W22, 1-2	
		W27, 1-2	
	W33, 1-3	W33, 1-2	
	W34, 1-2	W37, 1-2	
	W35, 1-2	W38, 1-2	

continued

Table E-9. SBC 215G Board Jumper Changes (continued)

Function	Remove Jumper	Install Jumper	Description
8-inch Drive Selection	W9, 1-2		Selects 8-inch Priam, model 3450-10, hard disk drives
	W10- 1-2		
	W17, 1-2		
	W22, 1-3	W22, 1-2	
	W26, 1-2		
	W27, 1-2		
	W28, 1-2		
	W33, 1-3	W33, 1-2	
	W34, 1-2	W37, 1-2	
	W35, 1-2	W38, 1-2	
SBX 218A Installation		W24, 1-2	Selects an SBX 218A module installed in SBX1 connector J4
		W4, 1-2	Selects an SBX 218A module installed in SBX1 connector J4
SBX 217C Installation	W3, 1-2		Selects an SBX 217C module installed in SBX2 connector J3
		W24, 1-2	Selects modules installed in both SBX1 and SBX2 connectors
Power connect		W20, 1-2	Connects ±12 volts from the Multibus to the SBC 215G

SBC 220 SMD Disk Controller

There are two versions of the SBC 220 board. One is configured by changing switch settings. The other is configured by changing jumpers. Both are described below.

Table E-10. SBC 220 Board Jumper Changes

Function	Remove Jumper	Install Jumper	Description
Select port address		E15-E16 E19-E20 E17-E18	Selects port address 120H
Select address decoding		W4-C - W4-2	Selects a 16-bit bus and 16-bit address decoding
Set Interrupt Level	W4-C - W4-5		Connect Multibus Interrupt level 2

Switch Settings for the SBC 220 Board	
Switch Setting	**Description**
SW1, 1-7 OFF 8 ON	Selects Port address 120H
S2, 1-2 ON	Selects a 16-bit bus and 16-bit address decoding
3-4 OFF 5 ON 6-10 OFF	Selects Port Address 120H

SBC 221 Peripheral Controller

The SBC 221 board controls 5.25-inch ST506 and ESDI-type hard disks, flexible diskettes (including high density 1.2 MByte diskettes), and tape devices. It does not support 8-inch devices. No jumper changes are required.

SBX 217C Cartridge Tape Controller

You can mount the SBX 217C module on the SBC 215G board and use it to control 1/4-inch cartridge tapes. The iRMX III OS does not support use of the SBX 217C module in any other mode.

Table E-11. SBX 217C Module Jumper Changes

Function	Remove Jumper	Install Jumper	Description

Configure the SBC 217C	E37-E38 E87-E88 E95-E96	E88-E95 E95-E96	Use a wire-wrap wire to connect these jumper posts.

SBX 218A Flexible Diskette Controller

The SBX 218A module controls either 5.25-inch or 8-inch flexible disk drives. The board can be mounted on an SBC 215G board. The BIOS device driver for the SBC 215G board controls the SBX 218A module in this configuration. The default jumpering for the SBX 218A module allows it to be mounted in the SBX1 connector (J4) of the SBC 215G board. This configuration controls 5.25-inch drives.

Table E-12. SBX 218A Module Jumper Changes

Function	Remove Jumper	Install Jumper	Description
8-inch Drive Selection	E5-E8	E1-E2	Make these changes to the jumpers on the SBX 218A when mounting it on the SBC 215G to control 8-inch drives
	E6-E7	E7-E8	
	E9-E10	E11-E12	
	E11-E14	E13-E14	
	E16-E17	E17-E18	
	E21-E23	E21-E22	
	E24-E25	E25-E26	
	E28-E29	E27-E28	
	E31-E32	E30-E31	
	E43-E44	E42-E43	
	E56-E57	E55-E56	
	E58-E59	E59-E60	
	E61-E62	E62-E63	
	E64-E65		
	E68-E69	E69-E70	

Communications and I/O Controller Boards

This section lists the hardware modifications you must make to these Intel Multibus I, Multibus II, and SBX bus communications and I/O controller boards:

Multibus I	**SBX Bus**	**Multibus II**
SBC 534	SBX 350	SBC 186/410
SBC/SXM 544A	SBX 351	
SBC 546	SBX 354	
SBC 547	SBX 279A	
SBC 548		
SBC 549		
SBC 188/48 SBC 188/56		

SBC 186/410 Serial Communications Controller

This SBC 186/410 Multibus II serial communications controller supports up to six serial I/O interfaces simultaneously. The six serial I/O interfaces are configured as two terminal devices and four interfaces that use the IBM implementation of RS-232C. No jumper changes are required.

SBC 534 Four Channel Communications Expansion Board

The SBC 534 board controls up to four RS-232C serial channels when the iRMX III OS is configured with the SBC 534 device driver. As shipped, the DIP header sockets on the SBC 534 board do not have a header. You must construct a header, using the instructions below, to connect a terminal to a serial channel on the SBC 534 board using a mass-terminated ribbon cable.

Table E-13. SBC 534 Board Configuration Information

Function	Remove Jumper	Install Jumper	Description
Interrupt Level Selection	E132-E140 E131-E140	E130-E137 E131-E138	SBC 534 interrupt to Multibus INT3
I/O Controller Address Selection	E118-E125	E123-E126	Selects port address 30H

To construct a header to insert in the serial port's 18 pin header socket (W1, W2, W3, or W4), you must either solder wires on a solder-style header or insert wires into a pin-and-socket-style header. Connect the header pins as follows:

DIP Header pin	**Description/Function**
4-5	Board DSR to Board DTR
6-7	Board RTS to Board CTS
8-10	Board RxD to Terminal TxD
9-11	Board TxD to Terminal RxD
12-13	Terminal RTS to Terminal CTS
14-15	Terminal DSR to Terminal DTR

SBC/SXM 544A Communications Controller

The SBC 544A board controls up to four RS-232C serial channels when the iRMX III OS is configured with the SBC 544A device driver. The SXM 544A is a preconfigured version of the SBC 544A.

As shipped, the DIP header sockets on the SBC 544A board do not have a header. You must construct a header, using the instructions that follow Table E-14, to connect a terminal to a serial channel on the SBC 544A board using a massterminated ribbon cable.

If you use the SXM 544A kit, no changes are required. If you use the SBC 544A board, you must:

1. Program the SBC 544A PROMs with the file */rmx386/proms/xf544*
2. Install the PROMs
3. Jumper the board
4. Install the headers

Table E-14. SBC 544A Board Jumper Changes

Function	Remove Jumper	Install Jumper	Description
Interrupt Level Selection	E80-E83	E80-E85	SBC 544A Multibus INT3
I/O Controller Address Selection	E40-E41 E45-E46	E41-E42	Selects 2732A EPROMs
		E51-E108 E53-E54 E55-E56 E57-E58 E59-E60 E61-E62 E63-E64 E65-E66 E72-E73	Selects base address 0E000H for dual-port RAM

To construct a header to insert in the serial port's 18 pin header socket (W1, W2, W3, or W4), you must either solder wires on a solder-style header or insert wires into a pin-and-socket-style header. Connect the header pins as follows:

DIP Header pin	Description / Function
2-3	Board DSR to Board DTR
4-5	Board RTS to Board CTS
6-12	Board RxD to Terminal TxD
7-13	Board TxD to Terminal RxD
14-15	Terminal RTS to Terminal CTS
16-17	Terminal DSR to Terminal DTR

If the terminal being used requires a DSR signal but does not provide a DTR signal for loopback purposes, the DIP header connections should be modified as follows:

DIP Header pin	Description / Function
15-16	Board DSR to Board DTR
3-17	Board DSR to Terminal DTR
4-5	Board RTS to Board CTS
6-12	Board RxD to Terminal TxD
7-13	Board TxD to Terminal RxD
14-15	Terminal RTS to Terminal CTS

SBC 546 Terminal Controller Board

The SBC 546 board is a four-channel serial I/O board that controls up to four terminals. It also has a Centronics-compatible line printer port and a clock calendar on it. The line printer is accessed as if it were a serial channel. After attaching the channel using the **attachdevice** command, use the iRMX **copy** command to copy a text file to the line printer.

The clock calendar is referred to as a Global Clock in iRMX OS documentation. It is battery-powered and, when jumpered correctly, runs regardless of whether power is applied to the system. When the iRMX III OS is configured to use the Global Clock on the SBC 546 board, the BIOS attaches it and the HI reads it (during initialization) to set the local time-of-day clock.

See also: **attachdevice**, *Command Reference*

Table E-15. SBC 546 Board Jumper Changes

Function	Install Jumper	Description
Enable Battery	E30-E31	Enables the battery-backup for the Global Clock

SBC 547 Terminal Controller Board

The SBC 547 board is an eight channel serial I/O board that controls up to eight terminals. The standard definition files for the SBC 386/2x/3x boards support up to three SBC 547 boards, plus one SBC 546 installed at the same time. No jumper changes are required.

SBC 548 Terminal Controller Board

The SBC 548 board is an eight channel serial I/O board that controls up to eight terminals. No jumper changes are required.

SBC 549 Terminal Controller Board

The SBC 549 board is a four-channel serial I/O board that controls up to four terminals. It also has a Centronics-compatible line printer port and a clock calendar on it. The line printer is accessed as if it were a serial channel. After attaching the channel using the **attachdevice** command, use the iRMX III **copy** command to copy a text file to the line printer.

The clock calendar is referred to as a Global Clock in the iRMX III OS documentation. It is battery-powered and, when jumpered correctly, runs regardless of whether power is applied to the system. When the iRMX III OS is configured to use the Global Clock on the SBC 549 board, the BIOS attaches it and the HI reads it (during initialization) to set the local time-of-day clock.

See also: **attachdevice**, **copy**, *Command Reference*

Table E-16. SBC 549 Board Jumper Changes

Function	Install Jumper	Description
Enable Battery	E30-E31	Enables the battery-backup for the Global Clock

SBC 188/48 and SBC 188/56 Communications Boards

The SBC 188/48 and SBC 188/56 communications boards control up to eight RS-232C serial channels when the iRMX III OS is configured with the TCC driver. The SBC 188/48 and 188/56 communications boards are configured for data terminal equipment (DTE) operation. This requires a null-modem cable to connect a terminal to the board. A null-modem cable swaps the receive data and transmit data signals (pins 2 and 3). To use a mass terminated ribbon cable to connect the SBX to a terminal, you must jumper the communication boards for data communications equipment (DCE) operation. If you received your SBC 188/48 or SBC 188/56 board as part of a System 300 Series Microcomputer, the conversion from DTE to DCE is already done. The jumpering instructions for both board configurations are described in Table E-17.

Table E-17. SBC 188/48 and SBC 188/56 Board Jumper Changes

Function	Remove Jumper	Install Jumper	Description
Interrupt selection	E251-E252		Disconnects PIC level 0 interrupt source.
	E335-E336		Disconnects PIC level 4 interrupt source.
		E259-E274	Connects SBX1 MINT0 to PIC level 0. Make this change if you are installing an SBX 354 module in SBX1.
		E276-E335	Connects SBX2 MINT0 to PIC level 0.
Channel 0 DCE	E202-E204 E203-E205 E206-E208 E207-E209 E210-E212 E211-E213 E214-E216 E217-E219	E190-E191 E192-E193 E194-E195 E202-E203 E204-E205 E206-E207 E208-E209 E210-E211 E212-E213 E214-E215 E218-E219	Configure Channel 0 for DCE operation.
Channel 1 DCE	E172-E174 E173-E175 E176-E178 E177-E179 E180-E182 E181-E183 E184-E186 E187-E189	E161-E162 E166-E167 E170-E171 E172-E173 E174-E175 E176-E177 E178-E179 E180-E181 E182-E183 E184-E185 E188-E189	Configure Channel 1 for DCE operation.

continued

Table E-17. SBC 188/48 and SBC 188/56 Board Jumper Changes (continued)

Function	Remove Jumper	Install Jumper	Description
Channel 2 DCE	E140-E142 E141-E143 E144-E146 E145-E147 E148-E150 E149-E151 E152-E154 E155-E157	E111-E112 E113-E114 E140-E141 E142-E143 E144-E145 E146-E147 E148-E149 E150-E151 E152-E153 E156-E157	Configure Channel 2 for DCE operation.
Channel 3 DCE	E119-E121 E120-E122 E123-E125 E124-E126 E127-E129 E128-E130 E131-E133 E134-E136	E115-E116 E117-E118 E119-E120 E121-E122 E123-E124 E125-E126 E127-E128 E129-E130 E131-E132 E135-E136	Configure Channel 3 for DCE operation.
Channel 4 DCE	E88-E90 E89-E91 E92-E94 E93-E95 E96-E98 E97-E99 E100-E102 E103-E105	E61-E62 E63-E64 E88-E89 E90-E91 E92-E93 E94-E95 E96-E97 E98-E99 E100-E101 E104-E105	Configure Channel 4 for DCE operation.

continued

Table E-17. SBC 188/48 and SBC 188/56 Board Jumper Changes (continued)

Function	Remove Jumper	Install Jumper	Description
Channel 5 DCE	E67-E69 E68-E70 E71-E73 E72-E74 E75-E77 E76-E78 E79-E81 E82-E84	E59-E60 E65-E66 E67-E68 E69-E70 E71-E72 E73-E74 E75-E76 E77-E78 E79-E80 E83-E84	Configure Channel 5 for DCE operation.
Channel 6 DCE	E38-E40 E39-E41 E42-E44 E43-E45 E46-E48 E47-E49 E50-E52 E53-E55	E1-E2 E3-E4 E38-E39 E40-E41 E42-E43 E44-E45 E46-E47 E48-E49 E50-E51 E54-E55	
Channel 7 DCE	E17-E19 E18-E20 E21-E23 E22-E24 E25-E27 E26-E28 E29-E31 E32-E34	E5-E6 E7-E8 E17-E18 E19-E20 E21-E22 E23-E24 E25-E26 E27-E28 E29-E30 E33-E34	Configure Channel 7 for DCE operation.

SBX 351 Serial I/O Module

The SBX 351 module controls one RS-232C serial channel when the iRMX III OS includes the 8251A device driver. As shipped, the SBX 351 module is configured for DTE operation. This requires a null-modem cable to connect a terminal to the module. A null-modem cable swaps the receive data and transmit data signals (pins 2 and 3). To use a mass-terminated ribbon cable to connect the module to a terminal, you must jumper the module for DCE operation.

To mount the SBX 351 module on a processor board, you must:

- Jumper the processor board to route the interrupt signal from the SBX connector MINTR0 pin (receive interrupt) to the appropriate interrupt of the PIC.
- Jumper the processor board to route the interrupt signal from the SBX connector MINTR1 pin (transmitter interrupt) to the appropriate interrupt of the PIC.

Table E-18. SBX 351 Module Jumper Changes

Function	Install Jumper	Description
Interrupt Level Selection	E33-E34 E35-E36	Connects 8251A RXRDY to MINTR0 Connects 8251A TXRDY to MINTR1

To construct a header to insert in the serial port's 18 pin header socket, U6, you must either solder wires on a solder-style header or insert wires into a pin-and-socket style header. Connect the header pins as follows:

DIP Header pin	Description/Function
3-13	Board RxD to Terminal TxD
4-14	Board TxD to Terminal RxD
5-6	Board RTS to Board CTS
7-8	Board DSR to Board DTR
9-10	Terminal DSR to Terminal DTR
11-12	Terminal RTS to Terminal CTS

SBX 354 Serial I/O Module

The SBX 354 module controls two RS-232C serial channels when the iRMX III OS includes the 82530 device driver. As shipped, the SBX 354 module is configured for DTE operation. This requires a null-modem cable to connect a terminal to the module. A null-modem cable swaps the receive data and transmit data signals (pins 2 and 3). To use a mass-terminated ribbon cable to connect the module to a terminal, you must jumper the module for DCE operation.

To mount the SBX 354 module on a processor board, you must jumper the processor board to route the interrupt signal from the SBX connector MINTR0 pin (receive interrupt) to the appropriate interrupt of the PIC.

Table E-19. SBX 354 Module Jumper Changes

Function	Remove Jumper	Install Jumper	Description
Channel A:		E13-E14	Connects DSR to +5 volts
		E9-E10	Selects DCE configuration
		E19-E20	
	E27-E29	E27-E28	
	E28-E30	E29-E30	
	E31-E33	E31-E32	
	E32-E34	E33-E34	
	E35-E37	E35-E36	
	E36-E38	E37-E38	
	E40-E42	E39-E40	
	E41-E43	E43-E44	
Channel B:		E15-E16	Connects DSR to +5 volts
		E11-E12	Selects DCE configuration
		E17-E18	
	E45-E47	E45-E46	
	E46-E48	E47-E48	
	E49-E51	E49-E50	
	E50-E52	E51-E52	
	E53-E55	E53-E54	
	E54-E56	E55-E56	
	E58-E60	E57-E58	
	E59-E61	E61-E62	

SBX 279A Graphics Module

The SBX 279A module controls a color graphics monitor, serial mouse, and an IBM PC/AT-compatible keyboard. To mount the module on a processor board, you must jumper the processor board to route the interrupt signal from the SBX connector MINTR0 pin to the appropriate input of the PIC. The PIC level is ICU-configurable. The submit file */rmx386/rgi/add279.csd* assists you in configuring the device driver. The device driver is compatible with both the SBX 279 and SBX 279A boards. No changes are required to the standard jumpering. Refer to the module's hardware reference manual for more information.

SBX 350 Parallel I/O Module

The SBX 350 module controls a Centronics interface line printer. To mount the SBX 350 module on a processor board, jumper the processor board to route the interrupt signal from the SBX connector MINTR0 pin to the appropriate PIC input for the SBX 350 line printer driver.

Table E-20. SBX 350 Module Jumper Changes

Function	Remove Jumper	Install Jumper	Description
INTA	E13-E14	E13-E18	Connect INTA (PC3) toMINTR0
PC7	E25-E26		
		SBX 350 IC Sockets	
Socket Number	**IC Type**	**Description**	
U3	SBC 902	1K Pullup Terminator Pack	
U4	7438	Open Collector Line Driver	
U5	SBC 902	1K Pullup Terminator Pack	
U6	SBC 902	1K Pullup Terminator Pack	

If you use the parallel port for the line printer, configure the printer interface connector to the Centronics interface standard as shown in Table E-21. You can use any line printer that recognizes the Centronics signal/pin standard.

Table E-21. Line Printer Pin Assignments

50-Pin Cable Connector[1]	50-Pin SBX Edge Connector[2]	30-Pin Connector[3]	Centronics-Standard Signal
24	23	1	Character strobe to printer
26	25	13	SLCT (Select)
28	27	12	Paper Out
30	29	10	Acknowledge from printer
34	33	9	Data Bit 7
36	35	8	Data Bit 6
38	37	7	Data Bit 5
40	39	6	Data Bit 4
42	41	5	Data Bit 3
44	43	4	Data Bit 2
46	45	3	Data Bit 1
48	47	2	Data Bit 0

[1] All odd-numbered pin numbers are connected to ground.

[2] All even-numbered pins are connected to ground.

[3] On the Centronics LP connector pin 16 is signal ground and pin 17 is chassis ground.

Networking Controller Boards

This section describes how to jumper these Intel network controller boards:

- SBX 586 network controller
- SXM 552A Ethernet communication engine

SBX 586 Network Controller

The SBX 586 module provides access to a LAN. No jumpering is required on the module. The iRMX III OS supports the SBX 586 mounted on the SBC 386/12, 386/12S, and 486/12S boards. This configuration is a COMMputer configuration.

See also: i*.job, *System Configuration and Administration*

The hardware configuration of the processor board required to use the SBX 586 module is:

- The SBX 586 module must be mounted on the SBX1 (80H) connector.
- The MINTR0 from the SBX 586 is connected to interrupt level 3 of the slave PIC connected to interrupt level 7 of the master PIC (encoded level 73H).
- Timer 2 of the on-board 8254 timer at port address D0H is connected to interrupt level 4 of the slave PIC connected to interrupt level 7 of the master PIC (encoded level 74H).
- 16-bit width of SBX Bus I/O.

See also: The hardware reference manual for your processor board for the specific jumper numbers

SXM 552A Ethernet Communication Engine

The SXM 552A Ethernet communication engine is a front-end processor for all Multibus systems. The intelligent controller is driven by the 82586 LAN coprocessor and supports an IEEE 802.3 network. The LAN controller executes iNA 960 software, which is in a COMMengine environment.

When you specify a COMMengine configuration, the OS uses these parameters:

- The Multibus interrupt line INT4 is used by the SXM 552A Ethernet communication engine.
- Interrupt level 48H (master level 4) of the 8259A programmable interrupt controller (PIC), on the processor board, is connected to Multibus INT4.
- Port address 8B4H is used for the SXM 552A Ethernet communication engine wakeup port.
- Memory addresses 1040H and 1088H are used for the start of the SXM 552A data queues.
- Multibus parallel priority bus resolution is used.

The IEEE 802.3 transceivers power up and power down without causing noise on the coaxial cable; therefore, the transceiver may be connected to or disconnected from a LAN communications controller board at any time.

See also: The appropriate hardware installation manual for specific directions on removal and replacement of boards in the system;
i*.job, *System Configuration and Administration*

Modifications to Non-Intel Controller Boards

These sections contain tables that you need to jumper some non-Intel controller boards. The jumpering changes cover only the changes that must be made to the factory-installed jumpers. If your board has been used in another application, refer to the controller board's installation guide and jumper your board as it was originally jumpered. Then proceed with the modifications described in the tables.

The controller boards are:

- Adaptec 1542A Intelligent Host Adapter (SCSI)
- Comptrol Hostess 550 Serial Controller
- NCR ADP-20

Adaptec 1542A Intelligent Host Adapter (SCSI)

The Adaptec 1542A Intelligent Host Adapter (SCSI) can be used on PC bus platforms with the iRMX III OS to control SCSI flexible diskette drives, tape drives, or hard disk drives. The jumper configurations in Table E-22 make this controller compatible with the SCSI devices in these ICU definition files:

Board	**Definition File**
Adaptec 1542A	pc.bck pcnet.bck pccprsd.bck pccp.bck

Table E-22. Adaptec 1542A Jumper Connections

J-1,8	J-1,11	J-6,1	J-7,1
J-11,1	J-14,2	J-15,2	J-16,3

SCSI ID	**Used for**
0, 1	floppies
2, 3, 4, 5	hard disks
6	tape devices

Comptrol Hostess 550 Serial Controller

The iRMX III OS supports 4, 8, and 16 channel HOSTESS 550 terminal controller boards. The Comptrol Hostess 550 Serial Controller is an 8 channel serial communications board. To be compatible with the default ICU definition files for PC Bus systems (base I/O address of 280H and an interrupt level of 5 (58H)), change the switch settings on the Comptrol Hostess 550 board.

See also: The board documentation for the 4 and 16 channel switch settings

Table E-23. HOSTESS 550 Terminal Controller

HOSTESS 550 Terminal Controller	Switch Settings
Switch 1	5, 3, 1 OFF 8, 7, 6, 4, 2, ON
Switch 2	8, 7, 6, 4, 3, 2, 1 OFF 5 ON

NCR ADP-20 Flexible Disk Controller

The NCR ADP-20 is a high-performance, general-purpose flexible disk drive controller for 3.5-inch or 5.25-inch drives. The controller provides a high-level interface between the SCSI bus and the SA-400 drive interface. The switch settings on the NCR ADP-20 flexible disk controller need to be changed to the settings in Table E-24. On a high-density TEAC drive, ensure the RY, I, and U1 jumpers are installed.

Table E-24. NCR ADP-20 Flexible Diskette Switch Settings

Switch Setting		Switch Setting
8 Position Switch	1-2 OFF	Selects 1 drive present
	3-4 OFF	Power-up diagnostic options
	5 OFF	Selects SCSI ID 0
	6-8 OFF	Ignore SCSI parity
4 Position Switch	1-3 ON	Not user selective
	4 OFF	Must be off

DigiBoard DigiCHANNEL PC/4, PC/8, and PC/16 Controllers

The DigiBoard PC/4, PC/8, and PC/16 boards are 4-channel, 8-channel, and 16-channel, respectively, serial communications boards. Each PC/*X* board contains jumpers and/or switches that set the I/O addresses of the board's status register and each asynchronous communication element (ACE). An ACE is one serial channel; if it is a PC/4 board, there will be four serial channels (ACE chips) on it.

See also: *pcxdrv.job, System Configuration and Administration*

In every case except the second PC/16 board, the hardware board number must be set to 0. If you have a second PC/16 board, use the jumpers to set its hardware board number to 1.

Table E-25. PC/4 I/O Addresses

Board	Status Port	Port 0	Port 1	Port 2	Port 3
0	188	130	138	140	148
1	288	150	158	160	168
2	208	1B0	1B8	1C0	1C8
3	308	1D0	1D8	1E0	1E8

Table E-26. PC/8 I/O Addresses

					Ports				
Board	**Status Port**	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
0	188	130	138	140	148	150	158	160	168
1	288	1B0	1B8	1C0	1C8	1D0	1D8	1E0	1E8
2	208	230	238	240	248	250	258	260	268
3	308	2B0	2B8	2C0	2C8	2D0	2D8	2E0	2E8

✏ **Note**

The PC/16 board uses PAL chips to set addresses instead of address switches. When ordering the PC/16 board, specify a board number of 0 or 1 and if you want the PICK OS PAL chips. These are available from Digiboard. Consult your PC/16 hardware reference manual for more information.

□ □ □

Error Messages F

Since the iRMX OS installation procedures cover different situations such as installing the operating system on a newly formatted disk or installing over a previous release, it is possible that you will receive error messages during installation. The error messages might appear in a log file or at the console terminal. This type of error message is generally a non-fatal message.

There are mistakes you might make in the installation that generate a fatal error message, such as attempting to install software while not logged on as Super. This type of error requires that you correct the situation and restart the installation.

Non-Fatal Error Messages

Many error messages are for information only, and do not reflect an installation failure. Such messages can be ignored. Some typical error messages and their causes are:

```
file_name, file does not exist
```

This error may appear multiple times in the log file.

```
0021:  E$FILE_NOT_EXIST, during submit execution
```

This error may be displayed on the screen when the **submit** file terminates.

The error messages above can occur because the current installation uses the convention of naming a previous version to *..._old*. The errors listed above are generated if a file to be renamed is not found, because it was not originally installed.

```
directory, file already exists
```

This error message can occur when installation software attempts to create a directory that already exists.

```
file does not exist no files found
```

This error message can occur when installation software attempts to assign access rights to a file that does not exist.

Fatal Error Messages

Some error messages state that the error is fatal, which means you cannot proceed with the operation you are doing. You can easily correct some fatal errors.

Handling Fatal Error Messages from the Tinstall Command

1. If you receive a fatal error message after issuing a **tinstall** command, stop the **submit** file by pressing;

   ```
   <Ctrl-C>
   ```

   If an error occurs, an error message is displayed, but the **submit** file does not stop. It continues executing the series of commands. The <Ctrl-C> stops the **submit** file so you can investigate the problem. It may be several minutes before you see a prompt on the screen, because the tape completely rewinds before the prompt appears.

2. Detach the tape drive by entering:

   ```
   dd :in_tape: <CR>
   ```

3. Correct the fault. The most likely problem is that you are not logged on as the system manager, Super. If this is the case, enter:

   ```
   Login: super <CR>
   Password: passme <CR>
   ```

4. Enter the **tinstall** command again.

Handling Fatal Error Messages from the Install Command

1. If you receive a fatal error message after issuing an **install** command, stop the execution of the **submit** file by pressing;

   ```
   <Ctrl-C>
   ```

   If an error occurs, an error message is displayed, but the **submit** file does not stop. It continues executing the series of commands. The <Ctrl-C> stops the **submit** file so you can investigate the problem.

2. Detach the diskette drive by entering:

   ```
   dd :f: <CR>
   ```

3. Correct the fault.

   If the error was caused by inserting the diskette incorrectly, re-insert the diskette.

   If you are not logged on as the system manager (Super), enter:

   ```
   Login: super <CR>
   Password: passme <CR>
   ```

4. Enter the **install** command again.

□ □ □

Index