Addendum

P/N 064541-002

4MB JANUS® Devices

Intermec

A UNOVA Company

Intermec® Corporation
6001 36th Avenue West
P.O. Box 4280
Everett, WA 98203-9280

U.S. service and technical support: 1-800-755-5505
U.S. media supplies ordering information: 1-800-227-9947

Canadian service and technical support: 1-800-688-7043
Canadian media supplies ordering information: 1-800-268-6936

Outside U.S. and Canada: Contact your local Intermec service supplier.

Information and specifications in this manual are subject to change without notice.

Manual Change Record

This page records the changes to this manual, which was originally released as version 001.

Version	Date	Description of Change
002	10/96	This manual was reformatted to 8.5 x 11.

Contents

3

Using DOS Extenders

4

Memory Mapping

I

Index

Before You Begin

This section introduces you to standard warranty provisions, safety precautions, warnings and cautions, document formatting conventions, and sources of additional product information.

Warranty Information

To receive a copy of the standard warranty provision for this product, contact your local Intermec support services organization. In the U.S. call 1-800-755-5505, and in Canada call 1-800-688-7043. Otherwise, refer to the Worldwide Sales & Service list that ships with this manual for the address and telephone number of your Intermec sales organization.

Safety Summary

Your safety is extremely important. Read and follow all warnings and cautions in this book before handling and operating Intermec equipment. You can be seriously injured, and equipment and data can be damaged if you do not follow the safety warnings and cautions.

Do not repair or adjust alone Do not repair or adjust energized equipment alone under any circumstances. Someone capable of providing first aid must always be present for your safety.

First aid Always obtain first aid or medical attention immediately after an injury. Never neglect an injury, no matter how slight it seems.

Resuscitation Begin resuscitation immediately if someone is injured and stops breathing. Any delay could result in death. To work on or near high voltage, you should be familiar with approved industrial first aid methods.

Energized equipment Never work on energized equipment unless authorized by a responsible authority. Energized electrical equipment is dangerous. Electrical shock from energized equipment can cause death. If you must perform authorized emergency work on energized equipment, be sure that you comply strictly with approved safety regulations.

Cautions

The cautions in this manual use this format.

CAUTION

Caution
A caution alerts you to an operating procedure, practice, condition, or statement that must be strictly observed to prevent equipment damage or destruction, or corruption or loss of data.

Conseil
Une précaution vous alerte d'une procédure de fonctionnement, d'une méthode, d'un état ou d'un rapport qui doit être strictement respecté pour empêcher l'endommagement ou la destruction de l'équipement, ou l'altération ou la perte de données.

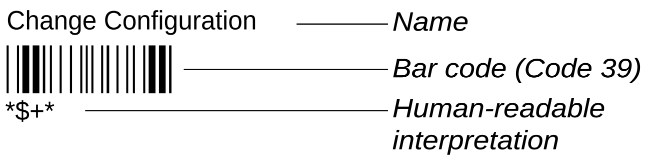
About This Addendum

This table summarizes the information in each chapter.

Chapter	What You Will Find
1	*Introduction* Describes the purpose and contents of the addendum. It also summarizes the enhancements and features of your 4MB JANUS device.
2	*Changes to JANUS Software* Explains the changes that have been made to JANUS 4.0 software.
3	*Using DOS Extenders* Describes DOS extenders and provides examples of how you can use them.
4	*Memory Mapping* Explains how the 4MB JANUS devices uses memory.
I	*Index* Provides an index to the addendum.

Format Conventions for Bar Codes

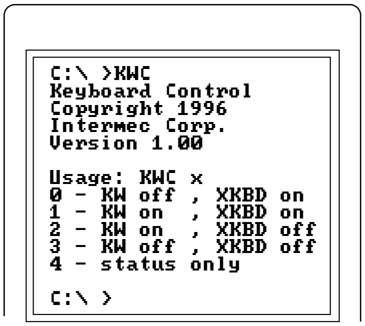
You can scan the bar codes listed in this manual to enter data or perform a command. Each bar code includes the name and human-readable interpretation. For example:



20X0A.009

Format Conventions for Software Screens and Messages

This manual includes illustrations that represent how the JANUS device displays software screens and messages. Here are two examples:

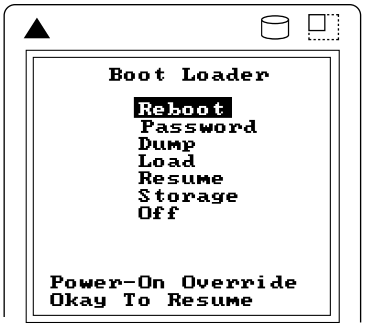


```
C:\ >KWC
Keyboard Control
Copyright 1996
Intermec Corp.
Version 1.00

Usage: KWC x
0 - KW off , XKBD on
1 - KW on  , XKBD on
2 - KW on  , XKBD off
3 - KW off , XKBD off
4 - status only

C:\ >
```

20X0A.004



```
       Boot Loader

         Reboot
          Password
         Dump
         Load
         Resume
         Storage
         Off


Power-On Override
Okay To Resume
```

20X0A.001

Format Conventions for Commands

This manual includes sample commands that are shown exactly as you should type them on your JANUS device. The manual also describes the syntax for many commands, defining each parameter in the command. This example illustrates the format conventions used for commands:

When you use the LOADADD command, follow this syntax:

```
loadadd [path\]filename [path\filename path\filename...]
```

where:

path is the drive and directory of the file(s) to include in the image file. If you do not include a *path*, the current directory is used.

filename is the name of the file or files to include in the image file and load to the JANUS device.

You can include multiple *path*\filename and *path*\*.* parameters in the command. The *path*\*.* parameter loads all the files in a directory. For example, type this command at the DOS prompt and press enter ↵ :

```
loadadd c:\janus\config.sys c:\atadrv\*.* c:\data\*.*
```

This table defines the conventions used in the example:

Convention	Description
`Special font`	Commands appear in this font. You enter the command exactly as it is shown.
Italic text	Italics indicate a variable, which you must replace with a real value, such as a number, filename, or keyword.
[ ]	Brackets enclose a parameter that you may omit from the command. Do not include the brackets in the command.
Required parameters	If a parameter is not enclosed in brackets [ ], the parameter is required. You **must** include the parameter in the command; otherwise, the command will not execute correctly.
where	This word introduces a list of the command’s parameters and explains the values you can specify for them.

Other Manuals

You may need to refer to the manuals listed below. To order additional manuals, contact your local Intermec representative or distributor.

Manual	Intermec Part No.
JANUS 2010 Hand-Held Computer User's Manual	058426
JANUS 2020 Hand-Held Computer User's Manual	059951
JANUS 2050 Vehicle-Mount Computer User's Manual	062874
0100 Access Point User's Manual	062367
Data Communications Reference Manual	044737
IRL Programming Reference Manual	048609
JANUS 2.4 GHz Installation Utility (4MB) User's Manual	064673
JANUS 2.4 GHz Terminal Emulation Quick Reference Guide	063682
JANUS 900 MHz Radio Frequency Quick Reference Guide	060207
JANUS 2010 and 2020 Optical Link Adapter Quick Reference Guide	058431
JANUS 2020 Battery Charger Quick Reference Guide	059955
JANUS 2020 Communications Dock Quick Reference Guide	059954
JANUS Application Simulator User's Manual	062778
JANUS PSK for Ada Reference Manual	062038
JANUS PSK for Basic Reference Manual	063191
JANUS PSK for C++ Reference Manual	062133
JANUS Terminal Emulation Quick Reference Guide	062178
PC-IRL Reference Manual	049212
RF System/9180 User's Manual	054292

1

Introduction

This chapter describes the purpose and contents of the addendum. It also summarizes the enhancements and features of your 4MB JANUS device.

How to Use This Addendum

Read this addendum before you begin using your 4MB JANUS™ device.

The hardware and software on 4MB JANUS devices has been upgraded to substantially improve efficiency and ease of use. These changes are not yet reflected in the user’s manuals, but they are described here in detail.

The information in this addendum applies to all 4MB JANUS devices, including the J2010 (all models), J2020 (all models), and J2050 (all models).

To learn about your 4MB JANUS device, use this addendum in conjunction with other JANUS documentation:

- This addendum contains the latest information about 4MB JANUS devices. If there are any conflicts between the information in the JANUS user’s manual and this addendum, use the information in this addendum.
- Check the README.DOC file on the Boot companion disk for any information about the new JANUS software that was not available when this addendum was printed.

New Software Features

The main features of JANUS 4.0 software are:

- Increased conventional memory.

  Reader Services and Configuration Manager are now loaded into upper memory. This change increases the amount of conventional memory available to run IRL or other applications to more than 512K.

  The addition of 3MB of extended memory means that you can now run much larger applications such as off-the-shelf database applications and terminal emulation packages.

- MS-DOS 6.2 upgrade from ROM DOS 5.0.

  MS-DOS uses a "shadowed BIOS," which allows 4MB JANUS devices to operate on DRAM rather than flash as on 1MB JANUS devices. This significantly increases program execution speed.

  Benchmark testing demonstrates that you can expect your existing applications will run 3 to 5 times faster on 4MB JANUS devices than on 1MB JANUS devices. Increased processing speed is most noticeable when performing operations that are very I/O intensive, such as saving files to disk or reading files from disk.

- New bar code wedge option.

  With the addition of the Direct Hardware Wedge, your 4MB JANUS device has two wedge options, one for maximum PC compatibility and one for maximum performance. The Virtual Wedge provides maximum performance, up to 10 times faster than the Direct Hardware Wedge, but cannot be used with applications (Attachmate, FTP, Zstem for DOS, PC-TCP for DOS, and others) that require keystroke interrupts when data is received. For those applications, the Direct Hardware Wedge is the best option to select.

  The default wedge setting on your 4MB JANUS device enables the Virtual Wedge and disables the Direct Hardware Wedge. Use the wedge configuration program (KWC.COM) to set wedge options.

Operational Changes With JANUS 4.0 Software

With this software release you will notice the following operational changes.

- The MS-DOS 6.2 Startup menu is displayed when you start or boot your 4MB JANUS device. The default configuration of the Startup menu allows you to select one of several PC Card configurations. You can modify the information presented on the Startup menu by changing the configuration block in the CONFIG.SYS file.
- MakeDisk and PutDisk utilities are supplied with 4MB JANUS devices.

  MakeDisk and PutDisk are two new utilities that you must use to change the contents of drives C or D. MakeDisk creates an image file containing the files you want on the drive, and PutDisk places the new image on the JANUS drive. MakeDisk supports the creation of subdirectories on drives C and D. *These utilities replace the MkImage and PutImage utilities that are supplied on 1MB units. The use of MkImage and PutImage is restricted to the 1MB JANUS devices, and will not work properly on 4MB JANUS devices.*

2

Changes to JANUS Software

This chapter describes the changes that have been made to the software provided with your 4MB JANUS device. Refer to your JANUS user's manual for information about JANUS software features.

Software Changes With 4MB JANUS Devices

Your 4MB JANUS device comes with updated software:

DOS Your 4MB JANUS device comes with MS-DOS 6.2 and supports many standard MS-DOS 6.2 commands. Use DOS commands and utilities to transfer files, create and run programs, create a RAM drive, and access files on PC cards. For help using MS-DOS commands, see any MS-DOS manual.

MakeDisk and PutDisk MakeDisk and PutDisk are two new utilities on your 4MB JANUS device that you can use to change the contents of drives C or D. These utilities replace the MkImage and PutImage utilities that are supplied on 1MB JANUS devices. MakeDisk creates an image file containing the files you want on the drive, and PutDisk places the new image on the JANUS drive. MakeDisk supports the creation of subdirectories on drives C and D. For more information, see "Using MakeDisk and PutDisk to Change Drives C or D" later in this chapter.

Direct Hardware Wedge The Direct Hardware Wedge is a new feature of JANUS 4.0 software that provides hardware level PC compatibility. It provides bar code data to PC applications that directly access the hardware. The existing Virtual Wedge is a software wedge that is ten times faster than going through the PC hardware. Use the Virtual Wedge for maximum performance. Use the Direct Hardware Wedge for maximum PC compatibility.

Using DOS Commands

Your JANUS device uses the MS-DOS operating system, and you can use DOS commands on the JANUS device just as you do on a PC. From the DOS prompt, you type a DOS command and press enter↵ to execute the command. For example:

```
dir
```

Or, you can create and scan bar code labels that contain DOS commands:

DIR Command

DIR

For help using DOS commands, see any DOS manual.

All DOS commands provided with the JANUS device are available on the MS-DOS Programs companion disk 2. The most commonly used commands are also stored on drive D. See the README.DOC for a list of the files on drive D.

Note: *The DOS commands available on drive D are a subset of the DOS commands that are available on the MS-DOS companion disk. You can add or replace DOS commands on drive D as needed.*

Your 4MB JANUS device supports these commands, but you cannot use them on drives C or D because they are ROM (read only memory) drives:

CHKDSK (You can analyze, but not fix, drives C and D with CHKDSK)

DISKCOMP

DISKCOPY

SCANDISK (You can analyze, but not fix, drives C and D with SCANDISK)

DEFRAG (DEFRAG will not work on drives C and D)

Defining the Startup Files

JANUS devices use two startup files to control how DOS uses hardware, memory, and files: AUTOEXEC.BAT and CONFIG.SYS. AUTOEXEC.BAT loads programs and defines paths. CONFIG.SYS loads device drivers and reserves memory for processing information. The commands in the startup files execute when you warm boot or cold boot the JANUS device.

You may modify the startup files for a variety of reasons:

- To change the default JANUS Startup menu.
- To support applications you will run on your JANUS device.
- To load drivers for your PC cards.
- To create physical RAM drives.
- To configure your JANUS device to operate in another language.

Because AUTOEXEC.BAT and CONFIG.SYS are stored on drives C or D, you must use Auto-Loader, binary file transfer (BFT), or MakeDisk and PutDisk to replace them. See your JANUS user's manual for more information on changing startup files.

The next sections illustrate what the two startup files may contain when your 4MB JANUS device arrives from the factory.

AUTOEXEC.BAT File

The AUTOEXEC.BAT file on your 4MB JANUS device should look like this one:

Command Line	Definition
`echo off`	The AUTOEXEC.BAT commands are not displayed on the screen as they are executed.
`cls`	Clears the screen.

`if not exist autoinst.bat goto T2`
`    call autoinst`
`    goto T3`
`:T2`
`if exist d:\autoinst.bat call`
`    d:\autoinst`
`:T3`
	These commands call the AUTOINST.BAT file, enabling you to update drives C or D with Auto-Loader.
	Do not remove these commands.
	Do not delete AUTOINST.BAT from drives C or D.

AUTOEXEC.BAT (continued)

Command Line	Definition
`set prompt=$p$g`	Sets the DOS prompt to display the current drive and directory, followed by the > symbol.
`set path=c:\;d:\;e:\;`	Directs DOS to look for commands and programs in the root directories of drives C, D, and E. Do not add drive G to the path, or else errors will occur when you do not have a PC card installed.
`set temp=e:\`	A temp directory is required for MORE.COM to work correctly on ROM drives.
`set im_errpath=e:\`	Tells the JANUS device to write the configuration error file JANUS.ERR to drive E. The JANUS device must execute this command before it calls LOADUMA.EXE.
`d:`	

`d:\ipm_4m.exe`
`REM d:\apm_4m.exe`
	IPM_4M and APM_4M work with card services to manage the power on the PC card drive when you suspend and resume the JANUS device.
	Note: *Do not load both at the same time, but you must load one.*

`if exist d:\loaduma.exe d:\loaduma`	Loads Reader Services, Configuration Manager TSR, and the decode and scanner utilities that let the JANUS device operate as a bar code reader.
`d:\im_disp.exe`	Loads software required for the display.
`d:\ic /0 e:\janus.ini`	Loads the JANUS.INI configuration file if it exists.

AUTOEXEC.BAT (continued)

Command Line

Definition

```
d:\kwc.com 0
```

Sets the bar code wedge options. The default configuration is 0. Configuration parameters are:

0 Virtual wedge and expanded keyboard enabled.

1 Direct Hardware wedge and expanded keyboard enabled.

2 Direct Hardware wedge enabled and expanded keyboard disabled.

3 Virtual Wedge enabled and expanded keyboard disabled.

4 Display status.

```
rfph 4
```

Loads the RF protocol handler for COM4 only if you are using a JANUS RF device.

```
if not exist c:\fta.exe goto
    DOS_PROMPT
e:
fta.exe checkhost; exit
```

Runs the FTSERVER batch file if it is on drives C or D. FTSERVER runs FTA.EXE, which provides binary file transfer (BFT) on the JANUS device. If a host is trying to initiate a BFT session, FTA continues running; if not, FTA stops running.

```
%IM_APPLICATION%
:DOS_PROMPT
```

After FTA terminates, %IM_APPLICATION% runs any application identified by the DOS environment variable IM_APPLICATION.

```
c:
```

Resets the JANUS device to drive C or D.

```
cls
```

Clears the screen.

CONFIG.SYS File

The CONFIG.SYS file on your 4MB JANUS device should look like this one:

Command Line	Definition

[menu]
menuitem=sram, Load PCCard
menuitem=ata, ATA PCCard
menuitem=flash, Flash PCCard
menuitem=io, I/O PCCard
menuitem=no, No PCCard
menucolor=15,0
menudefault=sram, 20
	Creates the Startup menu.

shell=command.com /e:2000 /p	Increases the size of the environment space to 2000 bytes. This line is required for Auto-Loader.

device=d:\himem.sys /testmem:off
dos=high
	Loads the DOS extended memory manager, HIMEM.SYS. You must load HIMEM before you load device drivers.

device=d:\power.exe /low	Loads APM power management.
device=d:\sramdisk.sys 256 512 /e	Creates the 256K RAM drive E.
install=d:\card_sr.exe	Loads software required for the PC card drives.

[sram]
device=d:\cs.exe /poll:1
device=d:\csalloc.exe
    d:\csalloc.ini
device=mtsram.ext
device=mtddrv.exe
	Loads software required for PC cards. CSALLOC is a DOS program that scans the system for available memory, I/O port, and interrupt request queue (IRQ) resources.

[ata]
device=d:\cs.exe /poll:1
device=d:\csalloc.exe
    d:\csalloc.ini
device=\atadrv.exe /s:2
device=mtddrv.exe
device=d:\cardid.exe
	Loads software required for ATA cards.

CONFIG.SYS (continued)

Command Line	Definition

[flash]
device=d:\cs.exe /poll:1
device=d:\csalloc.exe
    d:\csalloc.ini
device=d:\mti1.exe
device=d:\mti2p.exe
device=mtddrv.exe
device=d:\ftl.exe
	Loads software required for flash cards.

[io]
device=d:\cs.exe /poll:1
device=d:\csalloc.exe
    d:\csalloc.ini
device=mtsram.exe
device=mtddrv.exe
device=d:\cardid.exe
	Loads software required for SRAM and I/O cards.

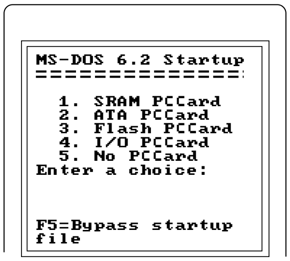
[no]	

[common]
device=d:\interlnk.exe /drives:7
    /noprinter /com:1 /auto
	Loads Interlnk as a resident device driver only if Intersvr is executing on a host computer that is connected to the JANUS device. Do not remove. Auto-Loader uses this line.

buffers=10	Sets the amount of memory that DOS reserves for data transferred to and from a disk.
files=50	Sets the number of files that can be open at one time. You need this command for IRL support.
stacks=9,256	Sets the amount of memory that DOS reserves to process hardware interrupts.

MS-DOS Startup Menu

The default JANUS startup menu is defined in the menu configuration block in the CONFIG.SYS file. It defines several different PC Card configurations that you can enable on your JANUS device. You can modify or disable the JANUS startup menu by removing or changing parameters in the menu configuration. Refer to any MS-DOS 6.2 manual for more information on setting up or changing the startup menu.



```
MS-DOS 6.2 Startup
==================

  1. SRAM PCCard
  2. ATA PCCard
  3. Flash PCCard
  4. I/O PCCard
  5. No PCCard
Enter a choice:


F5=Bypass startup
file
```

20X0A.002

Using the New I/O PC Card Drivers

Your 4MB JANUS device comes with two new PC card drivers, APM_4M.EXE and IPM_4M.EXE. APM_4M.EXE is loaded only for I/O cards, use IPM_4M.EXE for all other types of PC cards.

Preparing to Use I/O Cards

You can use I/O expansion cards in the PC card drive. To use an I/O card in the drive, you need to add the I/O card drivers and utilities to AUTOEXEC.BAT and CONFIG.SYS.

Note: *Before you use an I/O card in your JANUS device, read the documentation shipped with the PC card for any instructions on using your card. Some cards may need to be configured before they can be used in the PC card drive.*

Configuring the JANUS device to Use an I/O Card

Follow these steps to configure the JANUS device for an I/O card:

1. Edit the AUTOEXEC.BAT file, which is stored on the drive C.
2. Search for this line in the AUTOEXEC.BAT file:

   ```
   d:\apm_4m.exe
   ```

   If the line is remarked out, delete the REM characters. If you cannot find this line, add it to the AUTOEXEC.BAT file.
3. Remark out this line:

   ```
   d:\ipm_4m.exe
   ```

   Note: *Do not load both IPM_4M.EXE or APM_4M.EXE at the same time.*

4. Save your changes to the AUTOEXEC.BAT file.
5. Edit the CONFIG.SYS file on the JANUS drive C.
6. Search for this line in the CONFIG.SYS file:

   ```
   device=d:\cardid.exe d:\cardid.ini
   ```

   If the line is remarked out, delete the REM characters. If you cannot find this line, add it to the CONFIG.SYS file.
7. To save conventional memory, remark out any lines that load drivers for ATA, flash, or SRAM cards.
8. Save your changes to the CONFIG.SYS file.
9. Replace the AUTOEXEC.BAT and CONFIG.SYS files on the JANUS drive C.

Resetting the PC Card Drive for I/O Cards

By default, the PC card drive is not reset when you suspend or resume the JANUS device. Therefore, you can keep files open when you suspend and resume without losing data.

However, if you have an I/O card in the PC card drive, you must reset the drive when you resume the device. You can configure the device to reset the drive after a suspend/resume cycle by running the APM_4M.EXE program.

To load APM_4M.EXE into memory

- Type this command at the DOS prompt:

  ```
  d:\apm_4m
  ```

  Or scan this bar code:

  D:\APM_4M

  You can add this command to your AUTOEXEC.BAT file to start APM_4M.EXE whenever you boot the reader:

  ```
  d:\apm_4m.exe
  ```

  Note: *APM_4M.EXE is also available on the Application companion disk 3.*

Using MakeDisk and PutDisk to Change Drives C or D

You can use the new MakeDisk and PutDisk utilities to perform these tasks on your 4MB JANUS device:

- Adding or editing files on drives C or D.
- Replacing some files on drives C or D.
- Replacing all of the files on drives C or D.
- Deleting files from drives C or D.

Follow these steps, which the next sections describe in detail:

1. Create a working directory and fill it with files for drive C or D.
2. Create the new image file with MakeDisk.
3. Load the new image file onto the JANUS device with PutDisk.

***Note:** Due to its large size, MakeDisk is not loaded on drive D. You must copy self-expanding executable C_MAKE.EXE from the Boot Utilities companion disk to a drive on the host PC and run C_MAKE.EXE. After running C_MAKE.EXE, delete all of the extracted files except for MAKEDISK.EXE.*

CAUTION

Caution
Do not run any Intermec-provided JANUS application programs (such as IC.EXE) on your PC. Also, do not run any .EXE programs that use Intermec interrupt extensions or libraries on your PC. These programs will lock up your PC and may corrupt the PC BIOS.

Conseil
N'exécutez pas sur votre PC de programmes d'application JANUS fournis par Intermec (tels que IC.EXE). N'exécutez pas non plus sur votre PC de programmes .EXE qui utilisent des bibiothèques ou des extensions d'interruption car ces programmes bloqueront votre PC et pourraient corrompre le BIOS du PC.

Deciding Where to Run MakeDisk

You can run MakeDisk:

- on the RAM drive.
  If you want to make images larger than 256K on the RAM drive, you need to change your CONFIG.SYS file to enlarge the drive. Use the following to estimate the amount of space needed to create a RAM drive for running MakeDisk.

  Note: *On drive C you need to allocate space for your files, plus 15K, plus 512 bytes overhead per file, and 135K for the MS-DOS system files.*

  Note*: On drive D you can omit the 135K for the MS-DOS system files.*

- on a PC card inserted into the JANUS device.
- on a host computer with or without Interlnk.

  With Interlnk You can use Interlnk to copy files from drives C or D to the host's working directory and then copy MAKEDISK.EXE to a working directory on the host computer. You can then run MakeDisk on the host computer. For more information on using Interlnk, see your JANUS user's manual.

  Without Interlnk You can run MakeDisk on the host computer without Interlnk if you copy MAKEDISK.EXE to a working directory on the host computer. You can also use the DOS COPY commands to copy additional files from the companion disks instead of from drives C or D.

Creating and Filling the Working Source Directory

1. Create a working source directory where you will store the files you want on drives C or D.

   Since the use of subdirectories is fully supported by MakeDisk, you can create and populate subdirectories in the working source directory.

2. Create or copy files to the working directory.

   You can create or copy a maximum of 125 files to the drive C root directory or 128 files to drive D root directory. You can create or copy an infinite number of files to subdirectories below the root directory, provided they can fit in the allocated drive space.

   - If you are adding new files to drives C or D, copy all files currently on drives C or D as well as the new files to the working directory and its subdirectories.
   - If you are changing or replacing some of the existing files on drives C or D, copy all files currently on drives C or D to the working directory and its subdirectories. Edit the files you want to change or replace.
   - If you are replacing all files on drives C or D, copy the files you want on drives C or D to the working directory and its subdirectories.
   - If you are deleting files from drives C or D, copy all files currently on drives C or D to the working directory and its subdirectories. Delete the files you want to remove from drives C or D.

   You can use the DOS COPY command to copy files from the companion disks, or you can use Interlnk to copy files from the JANUS drives C or D.

Creating the New Image File

Create the image file by typing the command:

```
makedisk /s=[path] {/o=[output file] /d=[drvletter]}
```

where:

path specifies the directory that contains the files for the image file. This directory is called the source directory.

The *path* parameter is **required**. You may choose to include the drive letter if the source directory resides on another drive. For example, you could enter A:\JANUS\IMAGE as the path. If you do not specify the path correctly, the command fails.

output file specifies the full path and name of the image file to create. If you do not include this parameter, the image file is called DRIVE_C.IMG (DRIVE_D.IMG if you specified /d=d) and is placed in the current working directory.

The *output file* parameter is **optional**.

drvletter specifies the target JANUS drive, either C or D. The default is drive C.

The *drvletter* parameter is **optional**.

All files in the source directory are placed into the image file, including hidden and read-only files. If subdirectories exist, they are also incorporated in the image file.

If an error occurs while creating the image file, see “Troubleshooting,” later in this chapter.

Loading the New Image File

Note: *When you run PutDisk, all files on drives C or D are overwritten. Be sure to back up any files you want to keep before you continue.*

1. Connect the JANUS device to the host computer.

   Note: *If you use a communications dock and plan to use Interlnk during this procedure (as described in Step 3), you must connect the dock to the host computer with a 3-wire (2, 3, and 7) cable for Interlnk to operate properly.*

2. Connect a power supply to the communications dock or optical link adapter.

   Note: *For J2010 and J2020 devices only. This step is recommended, but is not required.*

   CAUTION

   Caution
   JANUS 2010 and 2020 devices should be powered by an AC power source when you use the PutDisk command in case the battery pack goes low while the flash memory is being overwritten. If the power goes down when using PutDisk, internal safeguards permit a resume of the flash programming process. In rare cases, drive C flash must be reloaded from the Boot Loader menu and you will lose data.

   Conseil
   Les périphériques JANUS 2010 et 2020 doivent être alimentés par une source de courant alternatif lors de l'exécution de la commande PutDisk étant donné le risque de décharge de la batterie pendant que la mémoire flash est écrasée. Si le courant est coupé pendant l'exécution de PutDisk, des mesures de sécurité internes font en sorte de remettre en opération le processus de programmation flash. Il peut arriver, quoique rarement, qu'il faille recharger la mémoire flash du lecteur C à partir du menu Chargeur-amorce (Boot Loader) et que des données soient perdues.

3. If you created the image file on the host computer and want to load it to the JANUS device from the host, start Interlnk now (with the host as server, and the JANUS device as client).

   Note: *You can simply reboot the JANUS device after the connected host has Interlnk installed.*

   You do not have to start Interlnk if you followed one of these strategies:

   - You created the image file on the JANUS device.
   - You created the image file on the host computer, copied the file to a PC card, and inserted the PC card into the JANUS device.

4. Type this command on the JANUS device:

   ```
   putdisk /d=[drvletter] /i=[image filename]
   ```

   where:

   drvletter is the target JANUS drive to be updated. The default is drive C.

   image filename is the name of the drives C or D image file created by MakeDisk. If you do not specify a path and filename, PutDisk looks for DRIVE_C.IMG.

   You may receive error messages in the following conditions:

   - If the specified file does not exist, PutDisk displays an error message and terminates.
   - If you specify a path without a file name, PutDisk also displays an error message.
   - PutDisk generates an error message if you attempt to put a DRIVE_C.IMG on drive D or a DRIVE_D.IMG on drive C.

   Note: *PutDisk will overwrite all files on drives C or D.*

For example, to copy the C-ATA.IMG image file from a PC card on the JANUS device to the JANUS drive C, type this command:

```
putdisk /i=g:\c-ata.img
```

PutDisk tries to determine if there is adequate power before replacing the drive image. If a power problem exists, PutDisk displays an error message and terminates.

You see messages on the JANUS device as PutDisk updates drives C or D, erases flash memory, and copies the DOS and new drives C or D image file to the JANUS device, and warm boots the JANUS device. If an error occurs, see “Troubleshooting,” later in this chapter.

Examples of Using MakeDisk and PutDisk

Following are three examples of how you can use MakeDisk and PutDisk to change the contents of drives C or D:

- In the first example, you run MakeDisk on the host, transfer the image file to the JANUS device on a PC card, and run PutDisk on the JANUS device to load the image file to drive C.
- In the second example, you run MakeDisk on the host, use Interlnk to make the image file appear as if it were on the JANUS device, and run PutDisk on the JANUS device.
- In the third example, you run MakeDisk on the host, transfer the image file to the JANUS device on a PC card, and run PutDisk on the JANUS device to load the image file to drive D.

Example 1

In this example, the resulting image file will be larger than 256K, so you must create the image file on the host computer if the JANUS RAM drive is configured at 256K (factory default). This example assumes that the host computer and the JANUS device can share a PC card.

1. Connect the JANUS device to a host computer using an appropriate serial connection.
2. Create a source directory on the host computer by typing this command at the host computer's DOS prompt:

   ```
   mkdir c:\drivec
   ```

3. Fill the source directory with all the files you want to include in the new image file:

   a. Copy the files from the current drive C to the source directory. Because the host computer and JANUS device can share a PC card, you can insert the card into the JANUS PC card drive and copy all files from drive C to drive G. Then insert the card into the host computer's PC card drive and copy all the files to the source directory.

   b. Add any other files and / or subdirectories to the working directory that you want on drive C. These files may include source read-only data files and applications for your JANUS device.

   c. Edit AUTOEXEC.BAT, CONFIG.SYS, and other files in the source directory as necessary.

4. Insert the JANUS Boot Utilities companion disk into a drive on the host PC.
5. Copy the self-expanding executable C_MAKE.EXE to drive C on the host PC by typing:

   ```
   copy a:\c_make.exe c:\
   ```

6 Run C_MAKE.EXE and delete all of the extracted files except for MAKEDISK.EXE.

7. Type this command at the host computer’s DOS prompt:

```
makedisk /s=c:\drivec /o=newdrv.img
```

An image file, NEWDRV.IMG, is created in the working directory of the host computer’s drive C.

Note: *Do not create the image file in the directory where the source files reside.*

8. Copy the image file from the host computer to the PC card and then insert the PC card into the JANUS PC card drive, drive G.

Note: *It is recommended that you provide an external power supply for the JANUS device before you execute PutDisk.*

9. At the DOS prompt on the JANUS device, type this command to load the new image file onto drive C.

```
putdisk /i=g:\newdrv.img
```

Example 2

In this example, you create the image file on the host computer, use Interlnk to make the image file appear as if it were on the JANUS device, and use PutDisk to load the image file to drive C.

1. Connect the JANUS device to a host computer using an appropriate serial connection.
2. Create a source directory on the host computer by typing this command at the host computer’s DOS prompt:

```
mkdir c:\drivec
```

3. Fill the source directory with all the files you want to include in the new image file:

   a. Copy the files from the current JANUS drive C to the source directory on the host PC. Use Interlnk with the host computer as the client and the JANUS device as the server.

   b. Add any other files to the source directory that you want on drive C. These files may include read-only data files and applications for your JANUS device.

   Note: *Do not create the image file in the directory where the source files reside.*

   c. Edit AUTOEXEC.BAT, CONFIG.SYS, and other files in the source directory as necessary.

4. Insert the JANUS Boot Utilities companion disk into a drive on the host PC.
5. Copy the self-expanding executable C_MAKE.EXE to drive C on the host PC by typing:

   ```
   copy a:\c_make.exe c:\
   ```

6 Run C_MAKE.EXE and delete all of the extracted files except for MAKEDISK.EXE.

7. Type this command at the host computer's DOS prompt:

   ```
   makedisk /s=c:\drivec /o=newdrv.img
   ```

   MakeDisk creates an image file, NEWDRV.IMG, in the working directory of the host computer's drive C.
8. Restart Interlnk. Make the JANUS device the client, and the host computer the server. The new image file, NEWDRV.IMG, appears on drive the H root directory.
9. At the DOS prompt on the JANUS device, type this command to load the image file to drive C.

   ```
   putdisk /i=h:\newdrv.img
   ```

Example 3

In this example, the resulting image file will update drive D. Since the image file will be larger than 256K, you must create the image file on the host computer if the JANUS RAM drive is configured at 256K (factory default). This example assumes that the host computer and you JANUS device can share a PC card.

1. Connect the JANUS device to a host computer using an appropriate serial connection.

2 Create a source directory on the host computer by typing this command at the host computer's DOS prompt:

   ```
   mkdir c:\drived
   ```

3. Fill the source directory with all the files you want to include in the new image file:
   a. Copy the files from the current drive D to the source directory. Because the host computer and your JANUS device can share a PC card, you can insert the card into the JANUS PC card drive and copy all files from drive D to drive G. Then insert the card into the host computer's PC card drive and copy all the files to the source directory.
   b. Add any other files and/or subdirectories to the source directory that you want on drive D. These files may include read-only data files and applications for your JANUS device.

c. Edit AUTOEXEC.BAT, CONFIG.SYS, and other files in the source directory as necessary.

4. Insert the JANUS Boot Utilities companion disk into a drive on the host PC.

5. Copy the self-expanding executable C_MAKE.EXE to drive C on the host PC by typing:

   ```
   copy a:\c_make.exe c:\
   ```

6 Run C_MAKE.EXE and delete all of the extracted files except for MAKEDISK.EXE.

7. Create the drive image by typing this command at the host computer's DOS prompt:

   ```
   makedisk /s=c:\drived /o=newdrvd.img /d=d
   ```

   An image file, NEWDRVD.IMG, is created in the working directory of the host computer's drive C.

   Note: *Do not create the image file in the directory where the source files reside.*

8. Copy the image file from the host computer to the PC card and then insert the PC card into the JANUS PC card drive (drive G).

   Note: *It is recommended that you provide an external power supply for the JANUS device before you execute PutDisk.*

9. At the DOS prompt on the JANUS device, type this command to load the new image file onto drive D.

   ```
   putdisk /i=g:\newdrvd.img /d=d
   ```

Changes to Reader Services

Reader Services programs are no longer loaded into conventional memory on your 4MB JANUS device. This frees conventional memory space while maintaining access to the Reader Services features.

Reader Services includes these programs:

Communications Utilities Communications Utilities are transmit and receive functions that you can call with PSK functions or software interrupts. Communications Utilities let your JANUS device transmit and receive buffers or bytes of data regardless of the communications protocol that you use.

Configuration Manager Configuration Manager maintains the current configuration file (such as JANUS.INI), ensures that your JANUS device operates according to that configuration, and lets you change the configuration file. Configuration Manager is a TSR (terminate and stay resident program) that loads into memory during a warm or cold boot.

Input Manager Using software interrupts and library functions, your JANUS applications can use Input Manager as an interface for all data I/O from the JANUS COM ports, wand, scanner, and keypad. Input Manager also handles all power management during data I/O. For help with software interrupts and library functions, see your JANUS PSK reference manual.

You can use Input Manager to customize how your JANUS device handles data from the keypad, wand, or scanner. The JANUS device can display all input, beep when you scan labels, and parse JANUS commands. For example, when you scan a bar code label, Input Manager can identify the symbology and determine if the label includes JANUS commands.

Virtual Wedge The Virtual Wedge decodes and manages bar code input, making your JANUS device functionally equivalent to a wedge reader connected to a PC. This program facilitates rapid porting of PC applications to your JANUS device. When you scan a bar code label, the Virtual Wedge inserts the data into the keyboard buffer as if it had been typed. The Virtual Wedge sends configuration commands to Configuration Manager and also recognizes and implements JANUS commands.

Direct Hardware Wedge The Direct Hardware Wedge is a new feature of JANUS 4.0 software that allows bar code data to emulate keystrokes on the keypad. For more information on the Virtual Wedge and Direct Hardware Wedge, see “Understanding the Bar Code Wedge,” in the following section.

Understanding the Bar Code Wedge

The Bar Code Wedge is a Reader Services feature that enables your JANUS device to decode bar codes, making it functionally equivalent to a wedge reader connected to a PC. Prior to JANUS 4.0 software, the only wedge option was the Virtual Wedge. With the 4.0 software release, the Reader Services wedge feature has a second option, the Direct Hardware Wedge.

When you scan a bar code with the Virtual Wedge Option enabled, data is inserted directly into the keyboard buffer as if it had been typed. The Direct Hardware Wedge provides hardware level PC compatibility and processes the scanned data directly as individual keystrokes.

The Direct Hardware Wedge provides bar code data to applications that directly access the hardware, it is the option to select when working with applications (for example, Attachmate, FTP, Zstem for DOS, PC-TCP for DOS, and others) that require keystroke interrupts to occur when data is received. The Virtual Wedge option, which is up to 10 times faster than going through the PC hardware, is the option you should select when you need maximum performance.

If you have developed advanced applications for your JANUS device, you need to customize how the reader and application handle data from the keypad, wand, or scanner. Refer to the JANUS Programmer’s Software Kit and reference manuals for instructions.

The Virtual Wedge is the default configuration. You can enable the Direct Hardware Wedge by using the KWC.COM program. For more information, see “Using the Wedge Configuration Program,” later in this chapter.

Enabling Direct Hardware Wedge Functions

When you enable the Direct Hardware Wedge function, you disable the portion of the Virtual Wedge that inserts data directly into the keyboard buffer. With the Direct Hardware Wedge function enabled, data is inserted into the hardware to be processed as actual keystrokes. The processing of Reader Services commands and configuration bar codes is not affected by the wedge option you select.

Intermec recommends that you enable the Direct Hardware Wedge if you experience any of the following problems using off-the-shelf PC software on your 4MB JANUS device.

- You scan a bar code and get a “read beep,” but the data isn’t received by your application.
- You scan a bar code and characters are dropped.
- Your application uses bar codes to manipulate screen functions and the screen doesn’t respond correctly.

Note: *If you attempt to run the wedge configuration program on a JANUS device running version 3.0 or earlier software, the Direct Hardware Wedge feature will not be available. The Direct Hardware Wedge feature requires a software modification and a new KSCPU chip.*

Using the Wedge Configuration Program

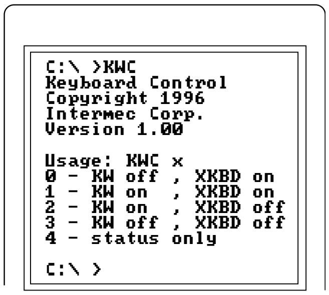
You can use the wedge configuration program (KWC.COM) to set wedge options on your 4MB JANUS device. To start KWC.COM, type this command at the DOS prompt and press enter↵.

```
kwc
```

Or, scan this bar code.

KWC

The main menu appears, displaying five configuration options.



```
C:\ >KWC
Keyboard Control
Copyright 1996
Intermec Corp.
Version 1.00

Usage: KWC x
0 - KW off , XKBD on
1 - KW on  , XKBD on
2 - KW on  , XKBD off
3 - KW off , XKBD off
4 - status only

C:\ >
```

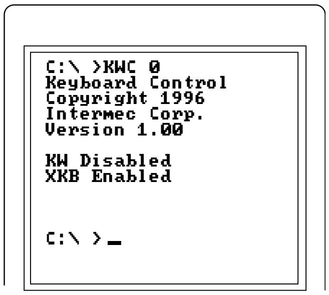
20X0A.004

Option 0 Option 0 is the default configuration. Select Option 0 to enable the Virtual Wedge and the expanded keyboard. To select this option, type the following command at the DOS prompt and press enter↵.

```
kwc 0
```

Or, scan this bar code.

KWC 0



```
C:\ >KWC 0
Keyboard Control
Copyright 1996
Intermec Corp.
Version 1.00

KW Disabled
XKB Enabled


C:\ >_
```

20X0A.005

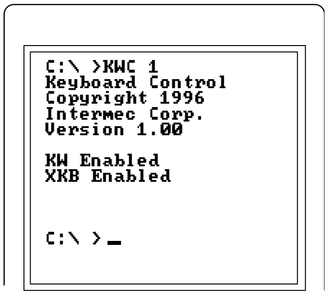
Note: *The AUTOEXEC.BAT on your 4MB JANUS device is set to Option 0 (Virtual Wedge enabled/extended keyboard enabled). To permanently change this option, you will need to modify your AUTOEXEC.BAT file and load a new image to drive C.*

Option 1 Select Option 1 to enable the Direct Hardware Wedge and the expanded keyboard. To select this option, type the following command at the DOS prompt and press enter ↵.

```
kwc 1
```

Or, scan this bar code.

KWC 1



```
C:\ >KWC 1
Keyboard Control
Copyright 1996
Intermec Corp.
Version 1.00

KW Enabled
XKB Enabled


C:\ >_
```

20X0A.006

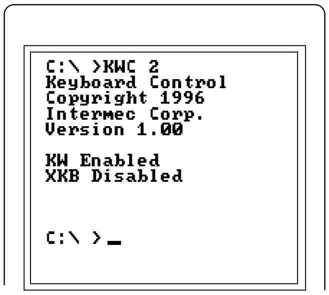
Note: *The following two options disable the expanded keyboard buffer. The expanded keyboard buffer allows the JANUS device to consistently process bar codes that are longer than 15 characters. Do not disable the expanded keyboard unless your application continues to experience compatibility problems.*

Option 2 Select Option 2 to enable the Direct Hardware Wedge and disable the expanded keyboard. To select this option, type the following command at the DOS prompt and press enter ↵.

```
kwc 2
```

Or, scan this bar code.

KWC 2



```
C:\ >KWC 2
Keyboard Control
Copyright 1996
Intermec Corp.
Version 1.00

KW Enabled
XKB Disabled


C:\ >_
```

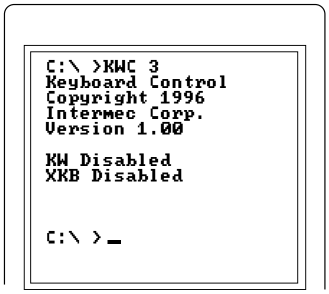
20X0A.007

Option 3 Select Option 3 to disable the Virtual Wedge and the expanded keyboard. To select this option, type the following command at the DOS prompt and press enter ↵.

```
kwc 3
```

Or, scan this bar code.

KWC 3



```
C:\ >KWC 3
Keyboard Control
Copyright 1996
Intermec Corp.
Version 1.00

KW Disabled
XKB Disabled


C:\ >_
```

20X0A.008

Option 4 Select Option 4 to display the current wedge configuration status. To select this option, type the following command at the DOS prompt and press enter ↵.

```
kwc 4
```

Or, scan this bar code.

KWC 4

Troubleshooting

This section describes how to solve problems you may have using the new MakeDisk and PutDisk utilities. It also describes informative messages that may be displayed on the Boot Loader menu when you turn on your 4MB JANUS device and a message you load PC card utilities.

PC Cards

This table describes a problem you may see when you use the new PC card utilities.

Problem	Solution
You try to load APM_4M.EXE or IPM_4M.EXE and you see the message, "CS PM already loaded."	You cannot load APM_4M and IPM_4M at the same time. If you see this message, one of the utilities is already loaded. Load APM_4M only for I/O cards; otherwise use IPM_4M. Check the AUTOEXEC.BAT file to see which utility is loaded. If you are loading the wrong one, edit the AUTOEXEC.BAT file or load the utility from the DOS prompt.

MakeDisk

This table lists messages that you may see when you use MakeDisk to create a new image file for drives C or D.

Message	Explanation/Solution
A source path has been specified that contains no files.	You cannot attempt to create an image file from a source directory that contains no files. MakeDisk will not let you create an empty drive image.
An invalid path and/or file name has been specified for the image file.	MakeDisk was unable to create the image file based on the "`/o=[path\filename]`" argument given. This error can be caused by either an invalid path name (for example, a path that doesn't exist or a "not ready" drive), or a *path\filename* that contains characters that DOS does not recognize as valid characters for filenames.
An invalid path name has been specified.	MakeDisk could not find the source directory path you specified on the command line. Please specify a valid subdirectory for the "`/s=[path]`" argument.

MakeDisk (continued)

Message	Explanation/Solution
• Corrupt BPB in suffix: Unable to create a drive image. • Corrupt suffix: Unable to create a drive image.	If either of these messages persist after further attempts, it may indicate a corrupted MAKEDISK.EXE file. Get a fresh copy of MAKEDISK.EXE from the Boot Utilities companion disk.
Drive full: Unable to write a complete image file to the output drive. Remove some files or specify a different output path and file name.	The drive where the image file is being written has insufficient free space available to contain the entire image file.
ERROR, unknown or incomplete command, or too many characters: [*bad command line argument shown here*].	You have entered a command line argument that MakeDisk does not recognize (as shown in the error message). Run MakeDisk again with the command line argument corrected.
Insufficient memory, unable to create a complete image file. Try again after removing device drivers, TSRs, or other applications that are occupying conventional memory.	There is not enough conventional DOS memory available on the computer running MakeDisk to complete the MakeDisk process. You may need to unload (terminate) some applications, device drivers, or TSRs in order to free the needed memory. Another approach is to reduce the number of nested subdirectories in the longest directory chain below your source directory.
Invalid drive letter specified. Please specify either drive C or D only.	MakeDisk will only create image files for drives C and D on your 4MB JANUS device.
Makedisk was unable to complete the drive image. Try again.	You may receive this message for several types of general faults such as: • One or more of the files in the source directory could not be accessed. This could indicate a corrupt file or a bad disk. • A network error occurred when DOS attempted to retrieve a remote file. • DOS was unable to read the source directory for any number of reasons.
Please specify a source path where your drive image can get its files from.	Enter a valid subdirectory name where your source files reside that will be used to create the drive image. (This is the `/s=[path]` argument).

MakeDisk (continued)

Message	Explanation/Solution
The root directory can only contain [*n*] files; your source files exceed this count. Place some of your root files in a subdirectory as an alternative.	Your source directory (first level) contains more than the maximum number of files and / or subdirectories permitted for the target drive's root directory. One approach to solve this is to place some of the first-level files in a subdirectory below your source directory.
The system has too many files open; unable to create a drive image. Increase the 'FILES=' setting in your CONFIG.SYS file or terminate other applications that may be running.	MakeDisk needs to open several files during the image file creation process, but is unable to do so. You may need to terminate other applications or close some files in order to complete the MakeDisk process.
Your source files would create a [*n*] byte image file which is [*n*] bytes in excess of drive [*C: or D:*]'s capacity.	The total size of the files and subdirectories in your source directory, including the overhead space required to emulate a DOS compatible drive, exceeds the storage capacity of the JANUS drive you have targeted.

PutDisk

This table lists messages that you may see when you use PutDisk to load a new image file to drives C or D.

Message	Explanation/Solution
Attempt to allocate memory failed, unable to proceed.	This indicates that there is not enough conventional DOS memory available for PutDisk to complete the flash update process. You may need to unload (terminate) some applications, device drivers, or TSR programs to free the needed memory.
Attempted to upgrade a ROM drive, unable to proceed.	PutDisk has determined that the target JANUS drive is comprised of ROM (read-only memory, which cannot be written to by PutDisk) instead of flash memory. Other kinds of system faults can lead PutDisk to make this determination in error, such as a flash programming voltage fault or address decoding problems.
Attempted upgrade in a RAM region, unable to proceed.	PutDisk has determined that the target JANUS drive is comprised of RAM (random access memory, which cannot support a non-volatile drive) instead of flash memory. Other kinds of system faults can lead PutDisk to make this determination in error, such as faulty flash memory or address decoding problems. You may need technical support to solve this type of error.
Cannot upgrade an unsupported drive.	Attempts to update a JANUS drive other than C or D are unsupported. Restart PutDisk, and specify the drive `/d=[drvletter]` as either C or D.
• Fault has occurred during attempt to erase flash memory. • Fault has occurred during attempt to write flash memory.	If either of these messages persist after retrying PutDisk with a fresh battery or charging source, it may indicate a flash memory hardware failure on the JANUS device.
Image checksum failed comparison with checksum of received data.	This may indicate that the flash memory update was incomplete. You will need to restart PutDisk using the same image file.
Image file error: Drive image file is larger than the drive's BPB capacity.	The size of the drive image file is larger than is declared valid by the formatting portion of the image file. You will have to reconstruct the drive image file by using the MakeDisk utility.

PutDisk (continued)

Message	Explanation/Solution
Image file error: Drive image file is larger than the flash media capacity.	The size of the image file is larger than the targeted drive’s flash memory on the JANUS device. You must reduce the total file content (total size) of the drive image file by rebuilding it with the MakeDisk utility using fewer or smaller files.
Image file error: Drive image file is too small to be valid.	You have attempted to write an image file to a JANUS drive that is smaller than the minimum permitted size for an image file. The minimum image file size is unique for each JANUS drive (for example, the image file may target a JANUS drive other than the one you attempted to write it to).
Image file error: Drive image type is not compatible with reader's DOS version.	You have attempted to write a drive image intended for a 1MB JANUS device to a 4MB JANUS device, or vice versa.
Image file error: Image is not compatible with the specified drive.	If the image file you attempted to write to drive C, was created for drive D, it must only be written to drive D.
Image file error: Received image data has exceeded the specified image file size.	This generally indicates a communications error or a DOS file I/O error. You should retry PutDisk to see if the problem persists. If the problem is related to the use of Intersvr/Interlnk or a remote drive that is networked using another method, you need to determine if a communications problem exists before proceeding.
• Image file error: Unrecognized drive image type. • Image file error: Drive image contains an invalid or unsupported BPB.	These messages indicate that the image file you attempted to write to a JANUS drive is either corrupt, or is an invalid image file format for the JANUS device. Build a fresh image file using the MakeDisk utility.
Image file not found.	PutDisk was unable to find the image file you specified on the command line as the `/i=[imagefile]` argument.
Power management fault has occurred on reader.	PutDisk attempts to cooperate with JANUS power management processes; however, rare circumstances may cause a temporary fault. The best remedy for this problem is to reboot the JANUS device after PutDisk completes. If a flash update was not completed due to this fault, restart PutDisk after rebooting.

PutDisk (continued)

Message	Explanation/Solution
Reader's programming voltage failed during flash operation.	This message will likely occur if the battery level is borderline. You will need to restart PutDisk after installing a fresh battery or connecting the JANUS device to a charging source. If the drive being updated was drive C, it is unlikely that you will be able to reboot to the MS-DOS prompt, however you may be able to continue by rerunning PutDisk from the current MS-DOS session (without rebooting). If this method fails, you will need to enter the boot screen and use LOADER.EXE to update the JANUS drive C.
Reader's programming voltage has failed operational limits.	The flash programming voltage source has exceeded the acceptable current limit. This may be due to other PC Cards that use flash programming voltage. Retry PutDisk after removing all PC Cards other than SRAM cards.
Specified image file path is not valid.	This error can be caused by either an invalid path name (for example, a path that doesn't exist or a "not ready" drive), or a path\filename that contains characters that DOS does not recognize as valid characters in filenames.
The reader’s battery is too low to proceed with flash operations.	You must install a fresh battery or connect the JANUS device to either a charger via the optical link adapter or a communications dock before proceeding with the flash update.
Unable to activate Reader's A20 signal.	This may indicate either a conflict with other software that controls the A20 signal, or a hardware problem on the JANUS device. Try using PutDisk again, after terminating any suspect applications or utility programs. Another alternative is to enter the boot screen and attempt the use of LOADER.EXE to update the drives C or D.
Unable to activate reader's programming voltage, cannot proceed.	This message indicates that flash memory cannot be updated due to flash programming voltage problems. If the problem persists after installing a fresh battery or connecting the JANUS device to a charging source, there may be a hardware problem that requires service.

PutDisk (continued)

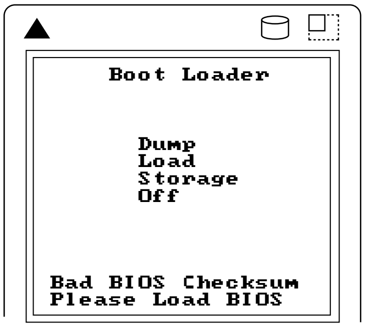
Message	Explanation/Solution
Unable to assert PM control due to concurrent PM host running on reader.	In this context, PM refers to protected mode which is used to access the extended memory regions assigned to the flash memory for JANUS drives. If another application is running that uses PM, it will likely introduce a conflict with PutDisk. You need to terminate (or uninstall) the application that is causing the conflict. Programs in this category include DOS extenders and EMM386 (which is not supported on the JANUS device).
Unable to engage reader's programming voltage, cannot proceed.	This may indicate a conflict with another process or device that uses flash programming voltage. Remove any PC Cards (other than SRAM cards) from the JANUS device.
Unable to resume flash drive upgrade; must be restarted.	PutDisk was unable to continue the flash update process that was interrupted with a power-down suspend / resume cycle. You need to restart PutDisk. First attempt to rerun PutDisk in the current MS-DOS session. If this fails, you will need to reboot to the MS-DOS prompt before restarting PutDisk. If the target drive was C and the JANUS device will no longer boot to the MS-DOS prompt, you will need to enter the boot screen and use LOADER.EXE to update drive C.
Upgrade failed, but original drive contents have been preserved.	This message may be displayed in conjunction with another specific fault message showing the cause of PutDisk interruption. It means that although the drive update was attempted, the original files on the target drive have not been altered and are therefore safe to use.

Boot Loader Menu

This table lists problems that may occur when you turn on your 4MB JANUS device.

Problem

When you turn on or cold boot the reader, the Boot Loader menu appears. The menu may not list all the Boot Loader commands, and the screen may include a status message. For example, the partial Boot Loader menu may look like this:



```
Boot Loader

Dump
Load
Storage
Off

Bad BIOS Checksum
Please Load BIOS
```

20X0A.003

Solution

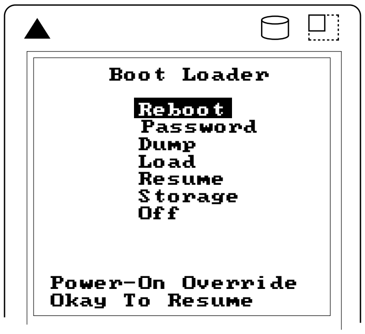
The bootstrap checks the system flash when doing a resume or cold boot. If the system flash is corrupt or there are bad checksum error messages, it brings up the Boot Loader menu. Read the status messages at the bottom of the menu.

If you see “Bad BIOS Checksum” or a message that indicates the system flash is corrupt, you could save the contents of conventional memory using the Dump menu selection, but you must reload the system flash.

Boot Loader Menu (continued)

Problem

Other Boot Loader menu messages may look like this:



```
Power-On Override
Okay To Resume
```

20X0A.001

Solution

If you see “Bad Saved State,” the device may have been in Storage mode. Press enter↵ to reboot. If you are prompted to exit Storage mode, press enter↵ again.

You may also see the “Bad Saved State” message if you press I/O to suspend the device while it is booting. Press enter↵ to reboot the device. You may be able to save the contents of the physical RAM drive.

Other self-explanatory messages may display on bottom of the Boot Loader menu.

For messages in the following group, choose Reboot on the Boot Loader menu and press enter↵ .

- “System Has Reset
  Please Reboot”
- “Bad CMOS Checksum, Errors
  Please Reboot”
- “DRAM Check Failed
  Please Reboot”

For the messages in the following group, choose Resume or Reboot on the Boot Loader menu and press enter↵ .

- “Keypad Fault Found
  Okay to Resume”
- “Power-On Override
  Okay to Resume”
- “Power-Off Timeout, Errors
  Okay to Resume”
- “Power-Off Forced
  Okay to Resume”

3

Using DOS Extenders

This chapter describes DOS extenders and provides examples of how you can use them with your applications.

Using DOS Extenders

DOS extenders are software products that allow the DOS-based programs you develop to access extended memory. Using DOS extenders you can now develop and run programs with large arrays or enormous amounts of code. With DOS extenders, you can use protected mode DOS versions of C functions and C++ functions to dynamically allocate large amounts of memory. While they are generally used with C or C++ programs, DOS extenders are not part of the standard C or C++ library.

Understanding the DPMI Server

The DOS Protected Mode Interface (DPMI) is an industry standard that allows DOS programs to access the advanced features of PCs in a well-behaved, hardware-independent fashion. DPMI functions are defined to:

- Manage descriptor tables.
- Perform mode switching.
- Allocate extended memory.
- Allocate DOS memory.
- Control the interrupt subsystem.
- Communicate with real-mode programs.

The DPMI transparently intercepts DOS and BIOS software interrupts in protected mode and reissues them in real mode.

Addressing DOS and BIOS Functions

In essence, DOS extenders provide DOS and BIOS functions in protected mode. Since your programs run in protected mode, any software interrupts your program generates (such as DOS calls made by the C run time library) occur in protected mode and are caught by the DOS extender. In most cases, the DOS extender passes the call to DOS which is running in real mode.

By providing protected mode surrogates for these functions, DOS extenders allow your program to perform as if it is running under a protected mode version of DOS. You can issue calls to BIOS functions in protected mode without worrying that the file I/O buffers are probably located in extended memory. The DOS extender handles the details of transferring data between conventional memory (where read mode DOS can access it) and extended memory (where your protected mode program probably allocated it).

DOS extenders act as a protected mode “wrapper” around real mode DOS and provide a completely compatible environment. DOS extenders don’t replace DOS, so they will work with whatever version of DOS you are currently using.

Use the following JANUS PSK libraries when you are developing programs using DOS extenders:

• Borland C/C++	im20libe.lib
• Microsoft C/C++	im20_mse.lib

Note: *Intermec has tested the 286|DOS Extender by Phar Lap and the PowerPack DOS extender from Borland and can recommend these products.*

Example - Using the Phar Lap DOS Extender

The following is an example of using the Phar Lap DOS extender.

Note: *If you are using Borland, add the following files to your project using the Integrated Development Environment. If you are using Microsoft Programmer's Work Bench, similar files are added to your project. Refer to the Phar Lap documentation for additional information.*

```
\run286\bc3\lib\c0pl.obj
filename.c
filename.def
\run286\bc3\lib\bcl286.lib
\run286\bc3\lib\phapi.lib
im20libe.lib
```

where:

filename.c specifies your source file.

filename.def contains the following lines:

```
name filename
protmode
exetype os2
stub "gorun286.exe"
exports BCC286_EXE
```

im20libe.lib is the PSK library used for DOS extenders.

The following Phar Lap files need to be in the DOS path on your JANUS device in order to run FILENAME.EXE.

```
run286.exe
doscalls.dll
```

If these files are loaded on drive E, you will have approximately 30K for your executable. You can allocate more disk space for drive E by modifying the CONFIG.SYS file on your JANUS device. This process is described in your JANUS user's manual.

Using PSK Functions With DOS Extenders

With a few exceptions, you use PSK functions the same way with DOS extenders as you do without them. The main difference is that DOS extenders use extended memory rather than conventional memory. The functions that you use differently are:

- im_receive_buffer_no_wait( )
- im_transmit_buffer_no_wait( )
- im_set_abort_callback( )

The buffers used by these functions must be allocated from conventional memory and are accessed by both real mode and protected mode addressing.

im_receive_buffer_no_wait

Purpose: This function receives data from the designated communications port and places the data into a user-defined record. It differs from im_receive_buffer in that the programmer must set up the data record and monitor the transfer status until transmission is complete. This function runs in the background after it is initiated; no checks are made.

Use this function when receiving multiple buffer transmissions in conjunction with im_rx_check_status.

Syntax:

```
#include <im20lib.h>
IM_STATUS im_receive_buffer_no_wait
    (IM_COM_PORT port_id,
    IM_COM_DATA_BUFFER far *data_struct);
```

IN Parameters: The *port_id* parameter identifies the communications port as follows:

IM_COM1	COM1
IM_COM4	COM4 (RF only)

The *data_struct* parameter is a far pointer to the data array for the received data. This buffer must hold at least 256 bytes.

OUT Parameters: None.

Return Value: For more information on return values, see, “Status Codes” in Appendix A of the *JANUS PSK for C/C++ Reference Manual*.

Notes: You must build and maintain a *data_struct* of the type IM_COM_DATA_BUFFER as described in IM20LIB.H.

You must periodically check the protocol_xfer_status element in the IM_COM_DATA_BUFFER record to see if all the data has been received. When protocol_xfer_status is no longer code 8602H (communications port in use), all data is received.

You must install a protocol handler to use this function.

This function differs from im_receive_buffer_noprot in that program execution continues after you call im_receive_buffer_no_wait. When im_receive_buffer_noprot returns from a function call, the data has been read, the time-out has expired, or an error was encountered.

Example

```
// Test of the transmit and receive functions

#include <alloc.h>
#include <conio.h>
#include <dos.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#include "im20lib.h"
#include "immsg.h"
#include "phapi.h"

void main
(
)
{
IM_COM_DATA_BUFFER * com_buff_struct;          // Protected mode address
```

Example (continued)

```
unsigned long        realmode_com_buff_struct; // Real mode address

IM_UCHAR           * data_buff;                // Protected mode address
unsigned long        realmode_data_buff;       // Real mode address

IM_USHORT            status;
IM_USHORT            wrk_pseg, wrk_psel;

    clrscr();
    printf("   Testing\nim_rcv_buff_no_wait\nim_xmit_buff_no_wait\n\n");
    // Allocate 300 bytes of conventional memory for the data buffer
    status = DosAllocRealSeg(300, &wrk_pseg, &wrk_psel);
    if (status != 0) {
       printf("Alloc failure #1!");
       exit(0);
    }

    // Construct selector and offset to make protected mode addressing
    data_buff = (char *) MK_FP(wrk_psel, 0);

    // Construct segment and offset to make real mode addressing
    realmode_data_buff = (unsigned long) MK_FP(wrk_pseg,0);

    // Allocate comm buffer structure
    status = DosAllocRealSeg(sizeof(IM_COM_DATA_BUFFER),
                             &wrk_pseg, &wrk_psel);
    if (status != 0) {
       printf("Alloc failure #2!");
       exit(0);
    }

    // Construct selector and offset to make protected mode addressing
    com_buff_struct = (IM_COM_DATA_BUFFER *) MK_FP(wrk_psel, 0);

    // Construct segment and offset to make real mode addressing
    realmode_com_buff_struct = (unsigned long) MK_FP(wrk_pseg,0);

// Set up comm buffer parameters to receive buffer
    com_buff_struct->command        = IM_CU_RECEIVE;
    com_buff_struct->user_length    = 300;
    com_buff_struct->comm_length    = 0;
    com_buff_struct->protocol_mode  = IM_USE_PROTOCOL;
    com_buff_struct->eom_char       = 0;
    com_buff_struct->protocol_xfer_status = IM_COMM_INUSE;
```

Example (continued)

```
// Use real mode address for data pointer parameter
    com_buff_struct->data_ptr       = (char far *) realmode_data_buff;
    // Receive buffer
    status = im_receive_buffer_no_wait
                (IM_COM1,
                (IM_COM_DATA_BUFFER far *)realmode_com_buff_struct);
    if (IM_ISSUCCESS(status)) {
        // Wait form transfer to be completed
        while (com_buff_struct->protocol_xfer_status == IM_COMM_INUSE);
        data_buff[com_buff_struct->comm_length] = 0; // null terminate
        printf("\nRecvd=>%s", data_buff );
    }
    else {
        printf("\nReceive error\nStatus = %4x", status);
    }
    // Set up comm buffer parameters to transmit buffer
    com_buff_struct->command        = IM_CU_TRANSMIT;
    com_buff_struct->user_length    = strlen(data_buff);
    com_buff_struct->comm_length    = 0;
    com_buff_struct->protocol_mode  = IM_USE_PROTOCOL;
    com_buff_struct->eom_char       = 0;
    com_buff_struct->protocol_xfer_status = IM_COMM_INUSE;

    // Use real mode address for data pointer parameter
    com_buff_struct->data_ptr       = (char far *) realmode_data_buff;

// Transmit buffer
    status = im_transmit_buffer_no_wait
                (IM_COM1,
                (IM_COM_DATA_BUFFER far *)realmode_com_buff_struct);
    if (IM_ISSUCCESS(status)) {
        // Wait form transfer to be completed
        while (com_buff_struct->protocol_xfer_status == IM_COMM_INUSE);
        printf("\nTransmit complete");
    }
    else {
        printf("\nTransmit error\nStatus = %4x", status);
    }
    printf("\n\nAll done!\n");}
```

im_transmit_buffer_no_wait

Purpose: This function transmits the contents of a buffer and passes control back to its client.

Syntax:

```
#include <im20lib.h>
IM_STATUS im_transmit_buffer_no_wait
    (IM_COM_PORT port_id,
    IM_COM_DATA_BUFFER far *data_struct);
```

IN Parameters: The *port_id* parameter identifies the communications port as follows:

IM_COM1	COM1
IM_COM4	COM4 (RF only)

The *data_struct* parameter is a far pointer to the data array that you want to transmit.

OUT Parameters: None.

Return Value: For more information on return values, see “Status Codes” in Appendix A of the *JANUS PSK for C/C++ Reference Manual.*

Notes: You must install a protocol handler to use this function.

This function differs from im_transmit_buffer in that the client must set the buffer structure and monitor the buffer status until transmission is complete.

Use this function for multiple buffer transmissions in conjunction with checking the status in *data_struct*.

You must build and maintain a *data_struct* of the type IM_COM_DATA_BUFFER as described in IM20LIB.H.

Example

```
See the example for im_receive_buffer_no_wait( ).
```

im_clear_abort_callback and im_set_abort_callback

The PSK im_clear_abort_callback function clears the notification (callback) mode of the Virtual Wedge. The im_set_abort_callback function sets up the callback address for notification of an exit program reader command. It determines which function is activated when /$ is received. For example, use this function when you are running a program and want to exit without using a warm boot. For more information on these commands, see Chapter 2, “C Language Support,” in the *JANUS PSK for C/C++ Reference Manual.*

The callback function must be in conventional memory in a Dynamic Link Library (DLL). The following example shows how you can do this. The example program consists of two files, ABORT.C and ABORTDLL.C.

Example

File 1 - ABORT.C

```
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
//
// Example of a program compiled with the Phar Lap 286|DOS-Extender
// and Borland C Version 3.1
//
// Test of the abort callback function along with the IRL A command
//
//
// The environment variable LIB must include at least the following:
// ================================================================
//
//    SET LIB=C:\RUN286\BC3\LIB;C:\BORLANDC\LIB
//
//
// The DLL with the callback is compiled as follows:
// ================================================
//
//    C:\>bcc286 -realdll abortdll.c abortdll.def
//
//    (this builds abortdll.lib)
//
//
// abort.c is compiled as follows:
// ==============================
//
//    C:\>bcc286 abort.c abortdll.lib im20libe.lib
//
//    C:\>bind286 abort
//
// Copy abort.exe and abortdll.dll to the Janus.
//
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
```

Example (continued)

```
#include <alloc.h>
#include <conio.h>
#include <dos.h>
#include <stdio.h>
#include <string.h>

#include "phapi.h"

#include "im20lib.h"

// external references to real-mode DLL (abortdll.c)

extern void far Abort_Callback(void);
extern int far abort_flag;

// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

void main
(
)

{
    IM_ORIGIN   source;
    IM_DECTYPE  symbol;

    IM_UCHAR    inString[300];

    int         done = 0;

    IM_USHORT   status;
IM_USHORT   cmd_count;

    IM_LENGTH_SPEC length_table[IM_LENGTH_SPEC_MAX] = {

         {IM_NO_LENGTH,0,0,0},
         {IM_NO_LENGTH,0,0,0},
         {IM_NO_LENGTH,0,0,0},
         {IM_NO_LENGTH,0,0,0},
         {IM_NO_LENGTH,0,0,0}
    };

    IM_UCHAR mask[10];

    mask[0] = '\0';             // null for no mask
//  strcpy(mask,"?##-####");    // Mask string for phone number
                                // (allow q for first character)

    clrscr();
```

Example (continued)

```
    // Set to programmer mode for IRL A instruction

    im_set_input_mode(IM_PROGRAMMER);

    // Setup abort callback (called when /$ is scanned)

    im_set_abort_callback(IM_ALL_SELECT,(void *)DosProtToReal(Abort_Callback));

    // Configure display and show amount of memory available

    im_set_display_mode(IM_SIZE_MODE_20X16,0,0,0);

    printf("\nAvail Mem = %lu\n\n", farcoreleft());

    // Loop until user types a "Q" or a "q" and display status of abort_flag

    while (!done) {
         printf("\nEnter>>");

status = im_irl_a(10000,length_table,mask,inString,&cmd_count,&symbol);

         printf("\nRecvd >>%s",inString);

         printf("\nStatus>>%x",status);

         if (abort_flag == 1) {

             printf("\nAbort flag set");
             abort_flag = 0;                 // clear flag
         }
         else {
             printf("\nAbort flag not set");

         }
         if (inString[0] == 'Q' || inString[0] == 'q')
             done = 1;
    }

    // Return Janus back to the way it was

    im_set_display_mode(IM_SIZE_MODE_80X25,
                        IM_STD_VIDEO_MODE_3,
                        IM_LCD_SCROLL_AT_16,
                        IM_STANDARD_CHAR_HEIGHT);
    im_clear_abort_callback();

    im_set_input_mode(IM_WEDGE);

    printf("\n\nAll done!\n");}
```

File 2 - ABORTDLL.C

```
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
//
// Test of the abort callback function
//
// Callback must be done in real mode, hence the need for this DLL.
//
// This DLL with the callback is compiled as follows:
// =================================================
//
//    C:\>bcc286 -realdll abortdll.c abortdll.def
//
//    (this builds abortdll.lib)
//
//
// abortdll.def consists of:
// ========================
//
//    LIBRARY abortdll
//    exetype dos4
//    description 'Real Mode Abort Callback routine'
//    exports
//      _Abort_Callback
//      _abort_flag
//
//
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

#include <dos.h>

// Global variables

int far abort_flag;
//***************************************************************************
//*
//* PURPOSE: callback function for abort program reader command detection
//*
//***************************************************************************


void _loadds Abort_Callback
(
    /* INPUTS */

unsigned short arg1,    // (arg1)
unsigned short arg2     // (arg2)

    /* IN/OUTS */
        /*  none */
```

Example (continued)

```
    /* OUTPUTS */
        /*  none */
)

{
    /* STATIC VARIABLES */
        /* none */

    /* AUTO VARIABLES */
        /* none */

    /* FUNCTION BODY */

    _enable();

    // set the abort flag (checked and cleared in abort.c)

    abort_flag = 1;

    /* VALUE RETURNED */
    /* none */
}
```

4

Memory Mapping

This chapter describes how your 4MB JANUS device uses memory.

Types of Memory Used in 4MB JANUS Devices

4MB JANUS devices use the types of memory described in this section. Memory in your 4MB JANUS device is similar to that of a PC in some ways, and different in other ways. One major difference is that your JANUS device uses memory to simulate disk drives. This memory map shows how the 4MB JANUS device uses memory.

Address	Contents	Description
FFFFFF E00000	No Physical Memory	End of 386SX Address Space
C00000	**APPLICATION DISK** **Drive D**	2MB Flash
880000	Unused Address Space No Physical Memory	(3.5MB)
800000	**USER FLASH** **Drive C**	512K Flash
400000	No Physical Memory	(4MB)
110000	**3MB DRAM** **Extended Memory** Contains 256K Drive E:	HIMEM is included in this 3MB
100000	**HIMEM, 64K** **MS-DOS 6.2 Loaded**	1MB Real Mode Boundary
FFFFF FFFF0	**RESET VECTOR** **(10H Bytes)**	Bootstrap Vector 10H ASIC BANKED
F2000	**BIOS & PM** (56K)	BIOS Flash, Shadowed

Memory Map (continued)

Address	Contents	Description
F0000	**INT13/19** (4K, 8K allowed)	
E0000	BIOS Extensions (64K)	BIOS Extensions, Shadowed
D0000	Scanner / Decodes (64K)	Upper memory block
C0000	PC Card Window (64K)	Can be used for optional upper memory
B8000	VIDEO Memory (32K)	Video SRAM, DRAM Masked Out
A0000	Reader Services and Configuration Manager Code and Data (95k)	32K BANKED Bootstrap ROM
9A800	BIOS Extension Data (22K)	End of 640K DRAM
16AA0	**Conventional Memory** (527K)	User Memory **Note:** *This area also contains any additional needed drivers.*
700	MS-DOS and PC card Data and Drivers (89K)	Includes MS-DOS data and IO.SYS, POWER, HIMEM, SRAMDISK, CS, CSALLOC, MTSRAM, MTDDRY, CARD_SR, and COMMAND.COM stub.
0	BIOS GDT MS-DOS Data Area BIOS Data Area Vector Table	Reserved for BIOS / MS-DOS (1.8K)

Conventional Memory (0 Through 640K)

Your 4MB JANUS device has 4MB of battery-backed dynamic RAM. The first 640K is conventional memory and is virtually the same as that of a PC. You can use this memory to run applications. The MS-DOS 6.2 operating system loads at 0 (zero) and works up, followed by the statements and drivers in the CONFIG.SYS file, then COMMAND.COM, TSR (terminate and stay resident) programs, and commands in the AUTOEXEC.BAT file just like a PC.

Upper Memory Area

The area between the 640K and 1024K is known as the upper memory area. The upper memory area is physically composed of a 256K flash ROM chip (system flash) and the video memory chip under the DRAM memory. Like a PC, this area contains the BIOS and video buffers. The upper memory area also contains PC card memory and Intermec JANUS driver software. The 384K upper memory area maps just above 640K conventional memory.

RAM Drive

Part of extended memory is configured as a RAM drive (drive E) on the 4MB JANUS device. The default CONFIG.SYS file loads HIMEM.SYS and uses the SRAMDISK.SYS device driver to create the RAM drive.

You can use the RAM drive to store files just like a disk drive on your PC. File access is usually faster on a RAM drive than a physical disk drive. Unlike a PC, the JANUS device saves any files on the RAM drive during a warm boot. However, the JANUS device clears any files on the RAM drive on a cold boot.

The RAM drive uses 256K of extended memory. If you are not using a PC card, Intermec recommends that you use the RAM drive for data files that you save or update frequently. The RAM drive is not as secure as a PC card since the drive is erased when you cold boot the JANUS device (unless you press Esc during memory test).

Extended Memory

The JANUS device is initially configured with 2.75MB of extended memory available, which is allocated for use by Intermec applications (IC.EXE and IRL). To remove the RAM drive and use the available extended memory, you need to change or delete the SRAMDISK statement in the CONFIG.SYS file. You can also use extended memory for overlay memory.

The JANUS device provides extended memory for control and access to RAM above 1MB. The extended memory conforms to XMS specification 2.0. The HIMEM.SYS driver initializes and manages extended memory. HIMEM.SYS looks at all the memory in the system and takes control of everything above 1024K. You must load all drivers in extended memory through the HIMEM.SYS interface. If a driver is not compatible with HIMEM.SYS, you cannot load it in high memory.

Some applications may require you to exclude memory from HIMEM.SYS management, leaving some extended memory available. If this is necessary, you should reconfigure the RAM drive (E) to use the available extended memory. DOS drivers must check in with HIMEM.SYS for their memory assignments within the XMS managed by HIMEM.SYS. Currently, the RAM drive uses the XMS interface through HIMEM.SYS.

Expanded Memory

When you install more than 1MB of memory on a PC, it can be used as expanded memory and is a way for the system to access memory beyond the 1MB barrier. To use expanded memory on a PC, you need an expanded memory board and an expanded memory manager.

Expanded memory is not available on 4MB JANUS devices.

High Memory Area (HMA)

The high memory area (HMA) is a 64K block of memory, starting 16 bytes below the 1024K mark, and is the first 64K of extended memory.

Since HMA can only hold one item, the first program that requests HMA uses it, regardless of the size of the program. The DOS=HIGH statement in CONFIG.SYS loads MS-DOS into high memory.

User Flash Memory

User flash memory contains the 512K ROM drive C. It is mapped to 800000H. Drive C is similar to the hard drive C on your PC and contains the MS-DOS startup files, CONFIG.SYS, and AUTOEXEC.BAT.

Since drive C is a ROM drive that resides in flash memory, you cannot write files to this drive as you would on your PC. You must use the Auto-Loader or MakeDisk and PutDisk to modify the startup files or change other files on drive C.

Application Flash Memory

Application flash memory is the 2MB ROM drive D. It is mapped to C00000H. DOS files and utilities are stored on drive D.

I

Index