

**User's Manual**

P/N 062778-001

# JANUS<sup>®</sup> Application Simulator

 **ntermec**

A **UNOVA** Company

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

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

Canadian technical and service 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.

The information contained herein is proprietary and is provided solely for the purpose of allowing customers to operate and/or service Intermec manufactured equipment and is not to be released, reproduced, or used for any other purpose without written permission of Intermec.

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

© 1995 by Intermec Corporation  
All Rights Reserved

The word Intermec, the Intermec logo, JANUS, IRL, Duratherm, Virtual Wedge, and CrossBar are trademarks of Intermec Corporation.

Throughout this manual, trademarked names may be used. Rather than put a trademark (<sup>TM</sup>) symbol in every occurrence of a trademarked name, we state that we are using the names only in an editorial fashion, and to the benefit of the trademark owner, with no intention of infringement.

## ***Contributors***

Author Karen Granger Ross

Editor Craig Thompson

Technical Illustrator George Wilson

Technical Reviewers Roy Chrisop  
Roy Law  
Yong-Qin Lu



# Contents

**Before You Begin xi**  
    Warranty Information xi  
    Cautions xi  
    About This Manual xii  
    Other Intermec Manuals xiv

## 1

---

### **Getting Started**

**Introduction to the JANUS Application Simulator 1-3**

**Hardware and Software Requirements 1-4**

**Installing the JANUS Application Simulator 1-5**

**Learning About the JANUS Application Simulator 1-6**  
    How the JANUS Application Simulator Works 1-6  
    If You Are Not Using JANUS PSK Library Functions 1-8  
    Integrating the Simulator Into Your Development Process 1-9  
    Using the Simulator With Other Products 1-9  
    JANUS Features That Are Simulated 1-10  
        Text Display 1-10  
        Bar Code Input 1-10  
        Power Management 1-12  
        Communications Input and Output 1-12  
    JANUS Features That Are Not Simulated 1-13  
        Character Height 1-13  
        Viewport 1-14  
        Contrast Level 1-14  
        Special JANUS Key Sequences 1-14  
        Speed and Performance 1-14

## 2

---

### ***Running and Debugging an Application***

*Deciding Where to Start the Simulator TSR 2-3*

*Starting the Simulator TSR 2-3*

*Running and Debugging a JANUS Application 2-4*

*Understanding the Limitations of the Simulator TSR 2-5*

*Exiting the Simulator TSR 2-6*

## 3

---

### ***Learning About the INI File***

*How to Customize the Parameters 3-3*

*How the Parameters Are Organized in This Chapter 3-3*

***Overview of Parameters 3-5***

*portn\_read\_file 3-6*

*portn\_write\_file 3-8*

*rx\_check\_status\_return 3-9*

*protocol\_ext\_status\_return 3-9*

*cancel\_rx\_buff\_return 3-10*

*cancel\_tx\_buff\_return 3-10*

*UART\_status 3-11*

*expanded\_keyboard\_status 3-12*

*kb\_insert\_string\_status 3-13*

*kb\_insert\_string\_capacity 3-14*

*flush\_exp\_keyboard\_status 3-15*

*warmboot\_status 3-16*

*power\_status\_line 3-17*

*power\_status\_battery 3-18*

*power\_status\_backup 3-18*

*power\_status\_batt\_life 3-19*

*power\_status\_return 3-19*

*display\_contrast\_level 3-20*



- display\_size\_mode* 3-21
- display\_video\_mode* 3-22
- display\_scroll\_mode* 3-23
- display\_viewport\_lock* 3-24
- display\_x\_pos* 3-25
- display\_y\_pos* 3-26
- display\_backlight* 3-27
- display\_char\_height* 3-28
- appl\_break\_status* 3-29
- follow\_cursor* 3-29
- control\_key* 3-30
- keyclick* 3-31
- numeric\_keypad\_switching* 3-32
- numeric\_keypad\_status* 3-33
- label\_symbology* 3-34
- sim\_wand\_key* 3-35
- clear\_abort\_callback\_return* 3-36
- command\_return* 3-36
- link\_comm\_return* 3-37
- serial\_protocol\_control\_return* 3-37
- standby\_wait\_return* 3-38
- unlink\_comm\_return* 3-38

***Communications Standard Return Values 3-39***

***Reader Wedge Standard Return Values 3-40***

# 4

---

## ***Customizing INI Files With the Editor***

***Why Customize INI Files? 4-3***

***Starting the Editor 4-5***

***Using Online Help 4-6***

***Creating a New INI File 4-7***

***Opening an Existing INI File 4-8***

***Setting Parameters 4-9***

*Setting Communications Parameters 4-10*

*Naming portn\_read\_file and portn\_write\_file 4-11*

*Specifying a Path for portn\_read\_file and portn\_write\_file 4-11*

*Setting Keyboard Parameters 4-12*

*Setting Power Management Parameters 4-13*

*Setting Display Parameters 4-14*

*Setting Keypad Parameters 4-15*

*Setting Reader Wedge Parameters 4-16*

***Saving Changes 4-17***

***Discarding Changes 4-20***

***Restoring the Default Values 4-20***

***Updating the Simulator TSR With the Current INI File 4-21***

***Setting Up the Printer 4-22***

***Printing INI Files 4-22***

***Exiting the Editor 4-23***

**5**

---

***Troubleshooting***

*Running an Application Without the Simulator TSR 5-3*

*Simulating Bar Code Input With an Intermec Wedge 5-3*

*Error and Status Messages for the Simulator TSR 5-3*

*Error and Status Messages for the Editor 5-4*

**I**

---

***Index***



## ***Before You Begin***

---

This section introduces you to standard warranty provisions, 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 sales organization. In the U.S. call (800) 755-5505, and in Canada call (800) 688-7043. Otherwise, refer to the Worldwide Sales & Service list shipped with this manual for the address and telephone number of your Intermec sales organization.

---

### ***Cautions***

The cautions in this manual use the following format.



#### **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 avertit 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 Manual**

This manual describes how to install, customize, and use the JANUS Application Simulator. You should see the online README file for important information that was not available when this manual was printed.

Use this manual in conjunction with the *JANUS PSK Reference Manual*, which describes the PSK library functions and JANUS-specific system interrupts that the JANUS Application Simulator captures and simulates.

## **Intended Audience**

This manual is intended for experienced PC programmers who already understand return values, know how to program in their chosen language, and know how to use their application development and debugging software. They have already read the *JANUS PSK Reference Manual* so they understand how to create programs for the JANUS.

## **How This Manual Is Organized**

The *JANUS Application Simulator User's Manual* is divided into five chapters:

<b>Chapter</b>	<b>What you'll find</b>
1	<i>Getting Started</i> Introduces the JANUS Application Simulator and explains how to install the software. Also describes which JANUS features the Simulator can make your development PC mimic.
2	<i>Running and Debugging an Application</i> Explains how to start and exit the Simulator TSR, and gives hints for debugging a JANUS application.
3	<i>Learning About the INI File</i> Lists the parameters in the initialization file.
4	<i>Customizing INI Files With the Editor</i> Describes how to use the Editor to customize the parameters in the initialization file.
5	<i>Troubleshooting</i> Describes how to recover if you run a JANUS application on your PC without first starting the Simulator TSR. Also lists and describes the status and error messages you may see.



### ***Terms and Conventions***

In this manual, the JANUS 2010 and JANUS 2020 are referred to as “the JANUS” or “the reader.” Also, the Intermec-specific library functions described in the *JANUS PSK Reference Manual* are referred to as “PSK functions.”

This manual uses the following conventions to emphasize keyboard input, screen messages, and command syntax.

#### ***Keyboard Input***

Keystrokes are emphasized in **bold**. For example, “press **Enter**” means you press the key labeled “Enter” on your keyboard.

If you must press more than one key at the same time, the keys are connected by a dash in the text. For example, press **Ctrl-Alt-Del** to perform a warm boot on a PC. When the keys are connected by a dash, it is important that you press and hold the keys in the order they are listed in the text.

#### ***Commands***

Command syntax is shown in the text as you should type it. For example, to see a listing of directories on the PC, type this command and press **Enter**:

```
dir
```

If a command line includes keywords in italics, the italics indicates that you must enter a real value in place of the keyword. For example, this command directs you to type the *janussim* command, specify the name of the INI file, and press the **Enter** key:

```
janussim filename.ini Enter
```

## ***JANUS Application Simulator User's Manual***

---

### ***Other Intermec Manuals***

You may need to refer to the JANUS manuals listed below while using the JANUS Application Simulator. To order manuals, contact your local Intermec representative or distributor.

<b>Manual</b>	<b>Intermec Part No.</b>
<i>JANUS Programmer's Software Kit Reference Manual</i>	062133
<i>JANUS 2010 Hand-Held Data Collection Computer User's Manual</i>	058426
<i>JANUS 2020 Hand-Held Data Collection Computer User's Manual</i>	059951

Also, you should see the online README file provided with the software. This README file contains important information that was not available when this manual was printed, such as operating guidelines or software development packages that are compatible with the JANUS Application Simulator.

*1*

# ***Getting Started***



*This chapter introduces the JANUS Application Simulator, explains how to install the software, and describes how to make your PC mimic a JANUS reader.*

## ***Introduction to the JANUS Application Simulator***

---

Applications created for JANUS readers may contain JANUS PSK library functions and JANUS-specific system interrupts that—when run on a PC instead of a reader—cause a PC to lock up.

With the JANUS Application Simulator, however, you can run JANUS applications on a PC. The Simulator captures those functions and interrupts before they can disrupt the PC. The Simulator then uses those functions and interrupts to make the PC mimic a JANUS reader, allowing you to run and debug any JANUS application on your PC.

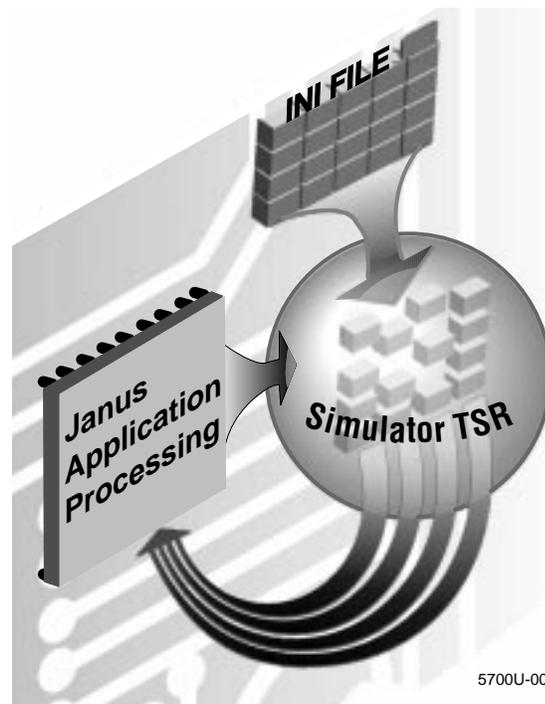
The JANUS Application Simulator operates as shown in this illustration.

Briefly, while a JANUS application is processing on a PC, it may issue a JANUS-specific system interrupt.

The Simulator TSR captures the interrupt, uses values from the INI file to assemble a response, and then sends the response to the JANUS application.

The JANUS application accepts the response and continues processing.

For a detailed technical description, see “How the JANUS Application Simulator Works” later in this chapter.



5700U-002

## ***JANUS Application Simulator User's Manual***

With the JANUS Application Simulator, you can debug the logic and syntax of a JANUS application, but not the user interface or performance:

- The Simulator helps you debug logic and coding errors by allowing you to use development tools and debuggers common in the software industry. For example, if you are using Microsoft Visual C++ to create JANUS applications, you can use Microsoft Codeview for DOS to test the applications. The Simulator runs in the background, transparent to your development software.
- The Simulator does not help you test the user interface or performance of the application. You can test those characteristics of the application far better on a JANUS reader than on any simulation program. You should always test your application by running it on a JANUS reader after you have finished debugging the logic.

Because JANUS applications are DOS applications, you must use a DOS-application debugger, not a Windows-application debugger. For example, you can use Codeview for DOS, but not Codeview for Windows.

Use the JANUS Application Simulator and this manual in conjunction with the programming software and documentation provided with your reader.

## ***Hardware and Software Requirements***

---

To run the JANUS Application Simulator, you need a PC with this hardware:

- Intel 386 or 486 CPU
- 3.5 inch floppy disk drive
- 1.2 MB available on the hard drive
- VGA monitor
- Mouse or other pointing device

You must have already installed this software on your PC:

- MS-DOS 5.0 or higher
- MS-Windows 3.0 or higher
- Application development software, such as Microsoft Visual C++ 1.5 Professional Edition

*Note:* Because JANUS applications are DOS applications, you must use DOS-application debugging software, such as Codeview for DOS. You cannot use Windows-application debugging software, such as Codeview for Windows.

## ***Installing the JANUS Application Simulator***

---

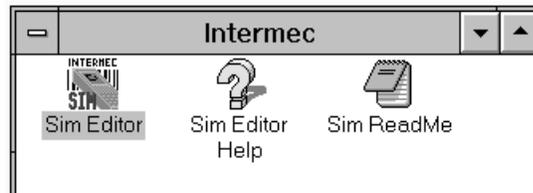
The Simulator includes a Windows-based installation program to help you install the software. During the installation, you choose the directory in which to install the software.

### **To start the installation**

1. Insert the JANUS Application Simulator disk into your floppy drive.
2. Start Windows.
3. From the Program Manager File menu, choose Run.
4. Start the installation with either of these commands:
  - If the disk is in Drive A, type this command: `a:setup`
  - If the disk is in Drive B, type this command: `b:setup`
5. Choose the OK button.
6. Follow the instructions displayed by the installation program.

The installation program creates an Intermec group on your Windows desktop.

The group contains the icon for the Editor, the Editor's help files, and the README file.



Read the README file before you use the Simulator. The README file may contain important information that was not available when this manual was printed, such as operating guidelines or the names of software development packages that are compatible with the Simulator.

For help using the Editor and its online help, see Chapter 4, "Customizing INI Files With the Editor."

## ***Learning About the JANUS Application Simulator***

---

To learn more about the JANUS Application Simulator, read these sections:

- How the JANUS Application Simulator Works
- If You Are Not Using the JANUS PSK
- Using the Simulator During Development
- Using the Simulator With Other Products
- JANUS Features That Are Simulated
- JANUS Features That Are Not Simulated

---

### ***How the JANUS Application Simulator Works***

The JANUS Application Simulator consists of three parts:

**Simulator TSR** The Simulator terminate and stay resident (TSR) program runs in the background on your PC. The Simulator TSR captures JANUS-specific system interrupts and makes your PC mimic a JANUS reader. For instructions, see Chapter 2.

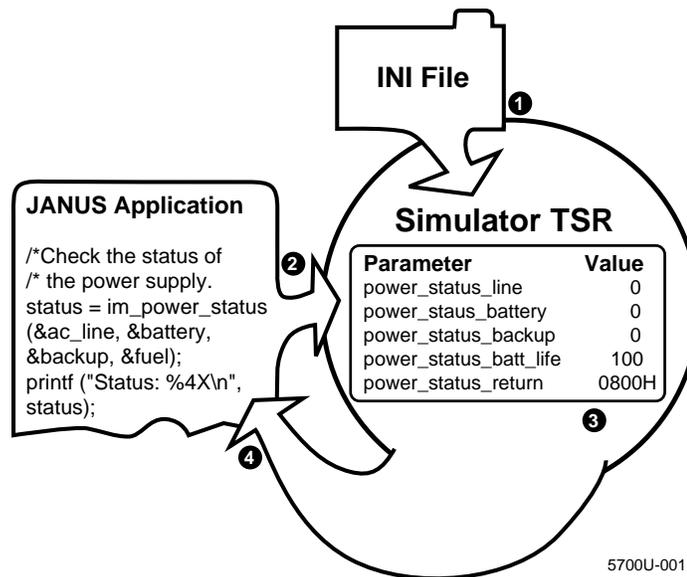
**INI File** The initialization (INI) file specifies how the Simulator TSR simulates JANUS features such as bar code input and battery status. JANUSSIM.INI is the default INI file. For a description of the parameters in the file, see Chapter 3.

**Editor** The Editor is a Windows-based tool for setting the parameters stored in the JANUSSIM.INI file. For instructions, see Chapter 4.

The Simulator TSR uses the parameters in the INI file as out parameters and return values for any JANUS PSK library functions and JANUS-specific system interrupts that a JANUS application contains.

For example, consider the JANUS battery. The status of the battery is very important because it may be the reader's only power source. JANUS applications may check the battery's status to make sure that sufficient power is available for the application. There is no battery to check on the PC. Therefore, the Simulator TSR must simulate the JANUS battery.

The following illustration shows how the Simulator TSR simulates a JANUS battery when a JANUS application issues the `im_power_status` function.



- ❶ When you start the Simulator TSR, it reads the parameters from the INI file, parses the parameter names, and saves the values into variables in memory.
- ❷ The JANUS application executes on the PC. To check the status of the power supply, the JANUS application issues the `im_power_status` function, which causes the INT 5EH interrupt to be sent to the Simulator TSR.
- ❸ The Simulator TSR uses the variables related to the interrupt and returns their values in the appropriate parameters.
- ❹ The Simulator TSR passes the values to the application. The application accepts the information as the status and return values of the `im_power_status` function call. The application continues executing.

---

## ***If You Are Not Using JANUS PSK Library Functions***

If you are programming with JANUS-specific system interrupts instead of JANUS PSK library functions, you can use the JANUS Application Simulator and this manual. Simply ignore all mention of the PSK functions.

The Simulator TSR services all the JANUS-specific system interrupts listed in the PSK reference manual. Some interrupts, such as INT 7CH, do not use the values stored in the INI parameters. This table lists the interrupts that do use the values stored in INI parameters:

<b>Interrupt</b>	<b>Related INI Parameters</b>	
INT 14H	portn_read_file portn_write_file rx_check_status_return protocol_ext_status_return	cancel_rx_buff_return cancel_tx_buff_return UART_status
INT 16H	expanded_keyboard_status kb_insert_string_status kb_insert_string_capacity	flush_exp_keyboard_status warmboot_status
INT 5EH	power_status_line power_status_battery power_status_backup	power_status_batt_life power_status_return
INT 79H	display_contrast_level display_size_mode display_video_mode display_scroll_mode display_viewport_lock	display_x_pos display_y_pos display_backlight display_char_height
INT 7EH	appl_break_status follow_cursor control_key	keyclick numberpad_switching numberpad_status
INT 7DH	label_symbology clear_abort_callback_return command_return link_comm_return serial_protocol_control_return	standby_wait_return unlink_comm_return portn_read_file portn_write_file

See Chapter 3 to learn more about INI parameters.

---

## ***Integrating the Simulator Into Your Development Process***

The JANUS Application Simulator can be an integral part of your development process. For example, if you use Visual C++ to develop applications, you can follow these steps:

1. From MS-DOS, start the Simulator TSR with this command:
2. Start Windows.
3. Run the Visual C++ compiler. Create or edit a JANUS application.
4. Run and debug the JANUS application.
5. Repeat Steps 3 and 4 as necessary.
6. Shut down Windows and return to MS-DOS.
7. Stop the Simulator TSR with this command:

```
janussim -d
```

---

## ***Using the Simulator With Other Products***

You can use the JANUS Application Simulator with other products, including application development software such as Microsoft Visual C++ 1.5 Professional Edition.

You can install an Intermec Wedge on your PC to simulate bar code input. For help configuring a wedge, see "Bar Code Input" later in this chapter.

The Simulator TSR should not interfere with the normal operation of other software on your PC. Therefore, you can start the Simulator TSR in your AUTOEXEC.BAT file and leave it running if you have sufficient RAM available.

You can use the JANUS Application Simulator to run JANUS applications that use the JANUS PSK library functions and JANUS-specific system interrupts, as described in Chapter 3.

You cannot use the JANUS Application Simulator to run JANUS programs such as the configuration application (IC.EXE) and the IRL Desktop (IRLDESK.EXE) on your PC. Those programs use low-level interrupts that the JANUS Application Simulator does not support.

---

## **JANUS Features That Are Simulated**

The Simulator TSR can reproduce these JANUS features:

- Text display
- Bar code input
- Power management
- Communications input and output

For more information, read the following sections.

### ***Text Display***

If the application calls the `im_set_input_mode` function to select the programmer input mode, the Simulator TSR limits character echoing to the lines and columns based on the display size you specified by calling the `im_set_display_mode` function or by setting the `display_size_mode` parameter in the INI file.

The Simulator TSR has no control of the application's output; however, the JANUS reader wraps output in some display modes. See your JANUS documentation for details.

*Note: All human factors and usability analysis of your JANUS applications should be performed on a JANUS reader, not with the Simulator TSR.*

### ***Bar Code Input***

You can simulate bar code input using either of these two methods:

- By pressing a special key sequence that lets you type "bar code" data.
- By configuring an Intermec Wedge and wand using actual bar code labels.

These methods are described in the following paragraphs.

#### **To simulate bar code input with a keyboard**

You can simulate bar code input by pressing the key sequence specified by the `sim_wand_key` in the INI file. After you press the key sequence, the Simulator TSR captures all keyboard input and presents it to the JANUS application as if it were bar code input from a wand. The simulated bar code input is terminated when you press **Enter**.

*Note:* You specify the symbology of the simulated bar code label with the `label_symbology` parameter in the INI file.

By default, the key sequence is **Ctrl-G**. You can change the key sequence by editing the `sim_wand_key` parameter. The key sequence consists of one or more control keys (**Ctrl**, **Alt**, **Shift**) and a character key (**A** to **Z**). Here are sample key sequences:

- **Alt-L**
- **Alt-Shift-B**
- **Ctrl-Shift-C**
- **Ctrl-Alt-Shift-W**

*Note:* The INI file displays a slash in the key sequence (`Ctrl/G`) instead of a hyphen. By convention, this manual uses a hyphen to indicate that you must press multiple keys simultaneously.

#### **To simulate bar code input with an Intermec Wedge**

You can also simulate bar code input by attaching an Intermec Wedge to your PC. After you set the first preamble characters to match the `sim_wand_key` value, you can use the wedge to enter bar code data into the JANUS application. The bar code input is terminated when you scan Enter or have a Return in the postamble.

See your wedge documentation for help setting the preamble. To decide how to scan the `sim_wand_key` value as the data for the preamble, consider this example:

Your `sim_wand_key` is **Ctrl-G**, so you must set the preamble to **Ctrl-G**. Because the wedge is in Set Preamble mode, you cannot scan the BEL character, even though it represents **Ctrl-G** in the full ASCII chart.

Instead, you must consult the PC/Workstation Keyboard Mapping table to learn which characters to scan for **Ctrl** and **G**. You learn that you must scan the SO character for **Ctrl** and the lowercase g character for **G**. (If you scan SO and uppercase G, the preamble is set to **Ctrl-Shift-G**.)

### **Power Management**

The initialization file contains parameters that help simulate the power management features of the JANUS reader. For example, the `power_status_line` parameter specifies if the simulated JANUS is connected to an AC power source.

If your JANUS application calls the `im_power_status` function or the INT 5EH interrupt, the Simulator TSR passes the preset AC power status to the JANUS application.

### **Communications Input and Output**

The Simulator TSR can simulate input and output data through the JANUS communications ports, COM1 to COM4.

#### **To simulate data input to the application**

1. Type sample data in an ASCII file. The format of the data in the file depends on the function you use to receive the data. For example:

**im\_receive\_byte** The `im_receive_byte` function reads one byte of data from the file each time it is called. Carriage return (CR) or line feed (LF) characters are returned as data to the JANUS application.

**im\_receive\_buffer** The `im_receive_buffer` function reads a buffer of data each time it is called. Therefore, your RCV file should contain a data string terminated by a CR/LF character. The CR/LF indicates the end of the buffer. If there is no CR/LF, the function reads up to 256 bytes of data.

With each subsequent call, either function would continue reading data where it left off until it reached an EOF. If the function was called after reaching an EOF, it would start reading data at the top of the file again.

2. Specify the ASCII file's path and filename in the `port $n$ _read_file` parameter in the INI file, where  $n$  is the COM port from which the application expects to read the data.
3. Load the TSR and run the application.
4. Test if the application read the data correctly from the ASCII file.

**To simulate data output from the application**

1. Specify the ASCII file's path and filename in the `port $n$ _write_file` parameter in the INI file, where  $n$  is the COM port to which the application expects to write the data.
2. Load the TSR and run the application.
3. The application creates the ASCII file and writes data to it.
4. Test if the application wrote the data correctly to the ASCII file.

*Note: You must test the application on a JANUS reader to make sure that the application is handling the input and output communications protocols correctly.*

---

**JANUS Features That Are Not Simulated**

The Simulator TSR cannot reproduce these JANUS features:

- Character height
- Viewport
- Contrast level
- Special JANUS key sequences
- Speed and performance

For more information, read the following sections.

**Character Height**

The Simulator TSR does not simulate the character height set for the JANUS reader. The characters appear in the standard font and size for the PC.

You can, however, store a character height value in the `display_char_height` parameter in the INI file. If your JANUS application calls the `im_get_display_mode` function or the INT 79H interrupt, the Simulator passes the preset character height to the JANUS application.

### **Viewport**

The JANUS viewport is not simulated in this release, but out parameters and return values for related function calls and interrupts are supported:

- You can specify if the viewport is locked (unmoving) with the `display_viewport_lock` parameter in the INI file.
- You can specify the horizontal and vertical offset for the viewport with the `display_x_pos` and `display_y_pos` parameters in the INI file.
- You can specify if the viewport will follow the cursor with the `follow_cursor` parameter in the INI file.

### **Contrast Level**

The Simulator TSR does not simulate the level of contrast set for the reader. The simulated display appears at the contrast level set for your PC.

You can, however, store a contrast value in the `display_contrast_level` parameter in the INI file. If your JANUS application calls the `im_get_contrast` function or the INT 79H interrupt, the Simulator TSR passes the preset contrast level to the JANUS application.

### **Special JANUS Key Sequences**

The JANUS keypad contains fewer keys than a standard PC-AT keyboard, but you can produce all 102 PC-AT keys with the reader by pressing a variety of key combinations. The special key sequences are listed in your JANUS user's manual.

When using the Simulator TSR to run JANUS applications on a PC, you do not use special key sequences because your PC keyboard contains all 102 PC-AT keys. For example, to enter the **F11** key on a reader, press . During a simulation, simply press the **F11** key on the PC keyboard.

*Note: The Simulator does not let you test how easily users can press the key sequences that your JANUS application expects. You should test the application's user interface by running it on a reader after you finish debugging the logic.*

### **Speed and Performance**

The Simulator TSR does not simulate the speed or performance of a JANUS reader. Your JANUS applications runs as fast as your PC can execute them.

2

## ***Running and Debugging an Application***



*This chapter describes how to use the Simulator TSR when you run and debug a JANUS application with application development software.*

## ***Deciding Where to Start the Simulator TSR***

---

The Simulator TSR is a terminate and stay resident program. TSRs are small programs that are loaded into DOS memory and run in the background. Therefore, it is important that you start the TSR from MS-DOS.

It is recommended that you start the Simulator TSR from MS-DOS without Windows running. That way, you can be sure that the Simulator TSR remains in memory until you exit the Simulator TSR or shut down your PC.

You can start the Simulator TSR if you are in an MS-DOS session from Windows or an MS-DOS shell spawned from your application development software. However, when you exit the session or shell, the Simulator TSR is removed from memory.

## ***Starting the Simulator TSR***

---

You can add a command to your AUTOEXEC.BAT file to start the TSR whenever you reboot, or you can perform the following procedure.

### **To start the Simulator**

1. At the MS-DOS prompt, change to the directory where the Simulator files are stored. For example:

```
cd c:\janussim
```

2. Start the Simulator TSR with either of these commands:

- To start the TSR with the default INI file, JANUSSIM.INI, type this command at the DOS prompt and press **Enter**:

```
janussim
```

- To start the TSR with a customized INI file, type this command at the DOS prompt and press **Enter**:

```
janussim filename.ini
```

Where *filename.ini* is the name of your customized INI file.

## ***JANUS Application Simulator User's Manual***

3. This message may appear:

Environment variable JANUSSIM not found

If so, type this command at the MS-DOS prompt and press **Enter**, or add this command to your AUTOEXEC.BAT file and reboot your PC:

```
set janussim=path\
```

Where *path* is the drive and directory where the JANUS Application Simulator software is installed. The *path* must have a terminating backslash. For example:

```
set janussim=c:\janussim\
```

4. When the Simulator TSR is loaded into memory, it displays this message:

The Simulator has been loaded!

You can now use your application development software to run and debug JANUS applications. For help, see the next section.

## ***Running and Debugging a JANUS Application***

---

Once you start the Simulator TSR, your next task is to start Windows and use your application development software to run and debug the JANUS application.

Your application development software executes as usual because the Simulator TSR does not affect it. You can use standard debugging techniques. Here are some suggestions:

- If you are editing JANUS application with the Microsoft Visual C++ editor, you can select the Tools menu and choose Codeview for DOS. The software displays the source code for your application in a DOS session. You can set break points, set watching, and step through the application using PSK library calls as well as standard library calls.
- JANUS applications are DOS applications, so make sure you use a DOS-application debugger, such as Codeview for DOS. You cannot use a Windows-application debugger, such as Codeview for Windows.
- Follow the instructions in the documentation provided with your application development software.

## ***Understanding the Limitations of the Simulator TSR***

---

Read these notes to understand the limitations of the Simulator TSR:

- Be sure to start the Simulator TSR before you run a JANUS application on your development PC.

If you run a JANUS application on your PC without first starting the Simulator TSR, the application may lock up your PC. If this happens, reboot your PC immediately.

- The JANUS Application Simulator does not help you test the application's user interface or performance. You can test those characteristics far better on a JANUS reader than on any simulation program. You should always test your application by running it on a JANUS reader after you have finished debugging the logic.
- An application containing an erroneous input combination will fail on the JANUS reader, but will not be detected by the Simulator TSR. Specifically, if a JANUS application performs the following two steps, it generates an error condition when it is run on a JANUS reader:
  1. The application calls the `im_set_input_mode` function and sets the input mode to `IM_PROGRAMMER` or `IM_DESKTOP`.
  2. The application calls a `stdlib` function (such as `getch` or `gets`) or `iostream` input function (such as `cin`) to read input from the keyboard.

The JANUS reader does not receive the keyboard input from the `stdlib` or `iostream` function. Instead the application hangs because the reader waits for input that it cannot receive.

However, if you run this application on your PC with the Simulator TSR, the PC receives the keyboard input. The Simulator TSR does not detect that this combination causes an error condition on the reader.

Make sure your JANUS applications do not combine a non-Reader Wedge input mode with a `stdlib` or `iostream` keyboard input function. Also, make sure you test your applications on a reader after you finish debugging them.

## ***Exiting the Simulator TSR***

---

To unload the Simulator TSR from memory, type this command at the DOS prompt and press **Enter**:

```
janussim -d
```

This message appears:

Simulator has been unloaded.

3

## ***Learning About the INI File***



*This chapter describes the parameters in the JANUS Application Simulator's initialization file. It also explains which JANUS PSK library functions and JANUS-specific system interrupts will receive the INI parameters as return values and out parameters.*

## ***How to Customize the Parameters***

---

You can customize the parameters in the initialization file using the Editor or an ASCII text editor:

- The Editor is more of a configuration tool than an editor. It helps you create INI files and customize their parameters. For help using the Editor, see Chapter 4.
- You can create and edit INI files with any text editor. Use a copy of the JANUSSIM.INI file to make sure you conform to the formatting conventions. Read this chapter to learn the values for each parameter.

## ***How the Parameters Are Organized in This Chapter***

---

The parameters in this chapter are listed in the order they follow in the INI file. For reference, here is the JANUSSIM.INI file with all the parameters set to their default values:

```
;
;
; Filename: JANUSSIM.INI
;
port1_read_file = comport1.rcv
port2_read_file = comport2.rcv
port3_read_file = comport3.rcv
port4_read_file = comport4.rcv
port1_write_file = comport1.trx
port2_write_file = comport2.trx
port3_write_file = comport3.trx
port4_write_file = comport4.trx
rx_check_status_return = 0600H - Comm Operation successful
protocol_ext_status_return = 0600H - Comm Operation successful
cancel_rx_buff_return = 0600H - Comm Operation successful
cancel_tx_buff_return = 4603H - Comm util no PH for config
UART_status = 37H
```

## ***JANUS Application Simulator User's Manual***

expanded\_keyboard\_status = 0 - command complete  
kb\_insert\_string\_status = 0 - success  
kb\_insert\_string\_capacity = 255  
flush\_exp\_keyboard\_status = 0 - success  
warmboot\_status = 0 - warmboot disabled  
power\_status\_line = 1 - AC line connected  
power\_status\_battery = 2 - critical battery  
power\_status\_backup = 1 - backup low  
power\_status\_batt\_life = 93  
power\_status\_return = 0800H - Power management success  
display\_contrast\_level = 16  
display\_size\_mode = 0 - 80 x 25  
display\_video\_mode = 3 - std video mode 3 (80x25)  
display\_scroll\_mode = 1 - scroll at 16  
display\_viewport\_lock = 0 - locked  
display\_x\_pos = 0  
display\_y\_pos = 0  
display\_backlight = 0 - backlight off  
display\_char\_height = 0 - standard height characters  
appl\_break\_status = 0 - break not been made  
follow\_cursor = 1 - follow cursor mode on  
control\_key = 0 - control key disabled  
keyclick = 0 - keyclick off  
numeric\_keypad\_switching = 0FFH - numeric keypad switching permitted  
numeric\_keypad\_status = 0FFH - number pad enabled  
label\_symbology = 1 - Code 39  
sim\_wand\_key = CTRL/G  
clear\_abort\_callback\_return = 0500H - Reader Wedge success  
command\_return = 0500H - Reader Wedge success  
link\_comm\_return = 0500H - Reader Wedge success  
serial\_protocol\_control\_return = 0500H - Reader Wedge success  
standby\_wait\_return = 0500H - Reader Wedge success  
unlink\_comm\_return = 0500H - Reader Wedge success

**Note:** *This example JANUSSIM.INI file shows the text message for each value as it appears in the INI file, in the software, and on the JANUS. However, the rest of this manual shows messages without abbreviations to make them more readable.*

## Overview of Parameters

---

This section contains the following information about each parameter in the JANUS Application Simulator's initialization file:

---

### ***parameter name***

<b>Description</b>	The purpose of the parameter.
<b>Default</b>	The default value for the parameter.
<b>Values</b>	The values you can set for the parameter.
<b>Function</b>	<p>The JANUS PSK library function that receives the parameter as a return value or out parameter. An out parameter specifies a value that is returned by the function.</p> <p>In this manual, the JANUS PSK library functions are called by their C library function names. The QuickBasic library function names are similar, except they use capitalization instead of underscores to make the names readable.</p>
<b>Interrupt</b>	The JANUS-specific system interrupt that triggers the Simulator TSR. Developers who are not using the JANUS PSK can use the interrupts in their JANUS applications.
<b>Notes</b>	<p>More information about the parameter.</p> <p>For example, this section may specify the name of the PSK function's out parameter that will contain the value of this INI parameter. The JANUS PSK documentation describes each PSK function and lists its out parameters and return values.</p>

The Notes section is optional.

---

## ***portn\_read\_file***

<b>Description</b>	Names the ASCII file that contains data to be read by the JANUS application as if it were received on a COM port. The <i>n</i> in <i>portn_read_file</i> is the port number (1 to 4).
<b>Default</b>	<i>comportn.rcv</i> , where <i>n</i> is the COM port number (1 to 4)
<b>Values</b>	Any filename
<b>Function</b>	<i>im_receive_buffer</i> <i>im_receive_buffer_no_wait</i> <i>im_receive_buffer_noprot</i> <i>im_receive_byte</i> <i>im_receive_input</i> <i>im_irl_v</i> <i>im_irl_y</i>
<b>Interrupt</b>	INT 14H
<b>Notes</b>	<p>Do not set this parameter to <i>comn.rcv</i>. Your PC will expect data from its COM<i>n</i> port.</p> <p>You must test the application on a JANUS reader to make sure that the application is handling the input communications protocols correctly.</p> <p>If you use the Editor to specify a new file for the <i>portn_read_file</i> parameter, the Editor lets you select an RCV file in any directory or on any drive that is accessible from your PC. However, the Editor saves only the filename to the INI file.</p> <p>If the RCV file will not be in the current directory when you run the JANUS application on your PC (or in a directory listed in the PATH statement in your AUTOEXEC.BAT file), you must edit the INI file to add the file's path to the RCV filename.</p>

For example:

```
port1_read_file = c:\payroll\temp\data1.rcv
port2_read_file = d:\sales\test2.rcv
port3_read_file = d:\sales\test3.rcv
port4_read_file = f:\mis\beta\sample4.rcv
```

The PSK function you use to read data from the RCV file affects how you format the data in the file. For example:

**im\_receive\_byte** This function reads one byte of data from the file each time it is called. Carriage return (CR) or line feed (LF) characters are returned as data to the JANUS application.

With each subsequent call, `im_receive_byte` continues reading data where it left off in the file until it reaches an end-of-file (EOF) character. If `im_receive_byte` is called again after reaching an EOF, the function starts reading data at the beginning of the file again.

**im\_receive\_buffer** This function reads a buffer of data each time it is called. Therefore, your RCV file should contain a data string terminated by a CR/LF character. The CR/LF indicates the end of the buffer. If there is no CR/LF, the function reads up to 256 bytes of data.

With each subsequent call, `im_receive_buffer` continues reading data where it left off in the file until it reaches an EOF. If `im_receive_buffer` is called again after reaching an EOF, the function starts reading data at the beginning of the file again.

**im\_receive\_input** This function reads a line at a time, similar to `im_receive_buffer`. However, because `im_receive_input` accepts input from multiple sources, when the function reaches the EOF, it does not start reading data at the beginning of the file again. This practice allows keyboard and wand input.

---

## ***portn\_write\_file***

<b>Description</b>	Specifies the file that will receive data the JANUS application writes to a COM port. The <i>n</i> in <i>portn_write_file</i> is the port number (1 to 4).
<b>Default</b>	comport <i>n</i> .trx, where <i>n</i> is the COM port number (1 to 4)
<b>Values</b>	Any filename
<b>Function</b>	im_transmit_buffer im_transmit_buffer_no_wait im_transmit_buffer_noprot im_transmit_byte
<b>Interrupt</b>	INT 14H
<b>Notes</b>	Do not set this parameter to com <i>n</i> .trx. Your PC will try to send data to its COM <i>n</i> port.

You must test the application on a JANUS reader to make sure that the application is handling the output communications protocols correctly.

If you use the Editor to specify a new file for the *portn\_write\_file* parameter, the Editor lets you select a TRX file in any directory or on any drive that is accessible from your PC. However, the Editor saves only the filename to the INI file.

If the TRX file will not be in the current directory when you run the JANUS application on your PC (or in a directory listed in the PATH statement in your AUTOEXEC.BAT file), you must edit the INI file to add the file's path to the TRX filename. For example:

```
port1_write_file = c:\payroll\temp\data1.trx
port2_write_file = d:\sales\test2.trx
port3_write_file = d:\sales\test3.trx
port4_write_file = f:\mis\beta\sample4.trx
```

---

***rx\_check\_status\_return***

<b>Description</b>	Specifies the return value for the simulated <code>im_rx_check_status</code> function.
<b>Default</b>	0600H – Communication operation successful
<b>Values</b>	See the “Communications Standard Return Values List” on page 3-39.
<b>Function</b>	<code>im_rx_check_status</code>
<b>Interrupt</b>	INT 14H

---

***protocol\_ext\_status\_return***

<b>Description</b>	Specifies the return value for the simulated <code>im_protocol_extended_status</code> function.
<b>Default</b>	0600H – Communication operation successful
<b>Values</b>	See the “Communications Standard Return Values List” on page 3-39.
<b>Function</b>	<code>im_protocol_extended_status</code>
<b>Interrupt</b>	INT 14H

---

### ***cancel\_rx\_buff\_return***

<b>Description</b>	Specifies the return value for the simulated im_cancel_rx_buffer function.
<b>Default</b>	0600H – Communication operation successful
<b>Values</b>	See the “Communications Standard Return Values List” on page 3-39.
<b>Function</b>	im_cancel_rx_buffer
<b>Interrupt</b>	INT 14H

---

### ***cancel\_tx\_buff\_return***

<b>Description</b>	Specifies the return value for the simulated im_cancel_tx_buffer function.
<b>Default</b>	0600H – Communication operation successful
<b>Values</b>	See the “Communications Standard Return Values List” on page 3-39.
<b>Function</b>	im_cancel_tx_buffer
<b>Interrupt</b>	INT 14H

---

## ***UART\_status***

<b>Description</b>	Specifies the return value for the simulated DOS UART function.
<b>Default</b>	30H – Bit 4 clear to send and Bit 5 data set ready
<b>Values</b>	01H – Change in clear to send 02H – Change in data set ready 30H – Bit 4 clear to send and Bit 5 data set ready 04H – Trailing edge ring detector 08H – Change in receive line 10H – Clear to send 20H – Data set ready 40H – Ring indicator 80H – Receive line detected
<b>Function</b>	None
<b>Interrupt</b>	INT 14H

---

## ***expanded\_keyboard\_status***

**Description** Specifies if the expanded keyboard is enabled or disabled.

**Default** 0 – Success

**Values** 0 – Success  
1 – Expanded keyboard buffer is enabled  
2 – Expanded keyboard buffer is disabled  
3 – Expanded buffer cannot be disabled

**Function** im\_get\_expanded\_keyboard

**Interrupt** INT 16H

**Notes** Intermec recommends that you do not use PSK expanded keyboard functions in your JANUS application.

The im\_get\_expanded\_keyboard function always returns 1000H (Expanded keyboard buffer success) as its return value.

If you select 2 for the expanded\_keyboard\_status parameter, im\_get\_expanded\_keyboard returns IM\_DISABLE as its first parameter.

If you select 0, 1, or 3 for the expanded\_keyboard\_status parameter, im\_get\_expanded\_keyboard returns IM\_ENABLE as its first parameter.

---

## ***kb\_insert\_string\_status***

<b>Description</b>	Specifies if the string was inserted into the expanded keyboard buffer.
<b>Default</b>	0 – Success
<b>Values</b>	0 – Success 1 – Expanded keyboard buffer not installed 2 – Keyboard buffer is full (entire string rejected) 3 – Invalid mode value was passed
<b>Function</b>	im_kb_insert_string im_kb_insert_string_keycodes
<b>Interrupt</b>	INT 16H
<b>Notes</b>	<p>Intermec recommends that you do not use PSK expanded keyboard functions in your JANUS application.</p> <p>This value is returned at the interrupt level:</p> <ul style="list-style-type: none"><li>• If you select 0 as the return value, the im_kb_insert_string function returns 1000H (Expanded keyboard buffer success) to the application.</li><li>• If you select 1 as the return value, the im_kb_insert_string function returns 9004H (Expanded keyboard buffer not installed) to the application.</li><li>• If you select 2 as the return value, the im_kb_insert_string function returns 9003H (Keyboard buffer is full (entire string rejected)) to the application.</li><li>• If you select 3 as the return value, the im_kb_insert_string function returns 9005H (Invalid mode value was passed) to the application.</li></ul>

---

## ***kb\_insert\_string\_capacity***

<b>Description</b>	Specifies the remaining number of characters that can be inserted into the expanded keyboard buffer.
<b>Default</b>	255
<b>Values</b>	Numeric value from 0 to 256
<b>Function</b>	im_kb_insert_string im_kb_insert_keycodes
<b>Interrupt</b>	INT 16H
<b>Notes</b>	Intermec recommends that you do not use PSK expanded keyboard functions in your JANUS application.  The value of this INI parameter is returned to the JANUS application in the *remaining_capacity parameter of the im_kb_insert_string and im_kb_insert_keycodes function calls.

---

***flush\_exp\_keyboard\_status***

<b>Description</b>	Specifies the status after deleting all characters from the expanded keyboard buffer.
<b>Default</b>	0 – Success
<b>Values</b>	0 – Success 1 – Failure
<b>Function</b>	im_flush_expanded_keyboard
<b>Interrupt</b>	INT 16H
<b>Notes</b>	Intermec recommends that you do not use PSK expanded keyboard functions in your JANUS application.

This value is returned at the interrupt level:

- If you select 0 as the return value, the im\_flush\_expanded\_keyboard function returns 1000H (Expanded keyboard buffer success) to the application.
- If you select 1 as the return value, the im\_flush\_expanded\_keyboard function returns 9006H (Expanded keyboard flush not successful) to the application.

---

## **warmboot\_status**

<b>Description</b>	Specifies the return value of the <code>im_get_warmboot</code> function, which lets users press <b>Ctrl-Alt-Del</b> to warm boot the JANUS.
<b>Default</b>	0 – Warmboot disabled
<b>Values</b>	0 – Warmboot disabled 1 – Warmboot enabled
<b>Function</b>	<code>im_get_warm_boot</code>
<b>Interrupt</b>	INT 16H
<b>Notes</b>	<p>The value of this INI parameter is returned to the JANUS application in the <code>*warm_boot</code> parameter of the <code>im_get_warm_boot</code> function call.</p> <p>The value held in memory by the Simulator TSR can be modified by the <code>im_set_warm_boot</code> function.</p>

---

## ***power\_status\_line***

<b>Description</b>	Specifies the line status of the simulated im_power_status function.
<b>Default</b>	0 – AC line not connected
<b>Values</b>	0 – AC line not connected 1 – AC line connected 255 – Unknown AC line
<b>Function</b>	im_power_status
<b>Interrupt</b>	INT 5EH
<b>Notes</b>	The value of this INI parameter is returned to the JANUS application in the *line_status parameter of the im_power_status function call.

---

### ***power\_status\_battery***

<b>Description</b>	Specifies the simulated condition of the battery.
<b>Default</b>	0 – High battery
<b>Values</b>	0 – High battery 1 – Low 2 – Critical 3 – Charging 255 – Unknown status
<b>Function</b>	im_power_status
<b>Interrupt</b>	INT 5EH
<b>Notes</b>	The value of this INI parameter is returned to the JANUS application in the *battery_status parameter of the im_power_status function call.

---

### ***power\_status\_backup***

<b>Description</b>	Specifies the simulated condition of the backup battery.
<b>Default</b>	0 – Backup OK
<b>Values</b>	0 – Backup OK 1 – Backup low
<b>Function</b>	im_power_status
<b>Interrupt</b>	INT 5EH
<b>Notes</b>	The value of this INI parameter is returned to the JANUS application in the *backup_status parameter of the im_power_status function call.

---

***power\_status\_batt\_life***

<b>Description</b>	Specifies the simulated percentage of the battery's full charge.
<b>Default</b>	100
<b>Values</b>	Numeric value from 0 to 100
<b>Function</b>	im_power_status
<b>Interrupt</b>	INT 5EH
<b>Notes</b>	The value of this INI parameter is returned to the JANUS application in the *fuel_gauge parameter of the im_power_status function call.

---

***power\_status\_return***

<b>Description</b>	Specifies the return value for the im_power_status function.
<b>Default</b>	0800H – Power management success
<b>Values</b>	0800H – Power management success 8821H – Cannot change state 8822H – Invalid state manager command 8842H – Invalid IPM interface command 8843H – Parameter out of range 8844H – Semaphore maximum value exceeded
<b>Function</b>	im_power_status
<b>Interrupt</b>	INT FEH

---

## ***display\_contrast\_level***

<b>Description</b>	Specifies the contrast level for the JANUS display.
<b>Default</b>	16
<b>Values</b>	Numeric value from 0 to 31
<b>Function</b>	im_get_contrast
<b>Interrupt</b>	INT 79H
<b>Notes</b>	<p>The value of this INI parameter is returned to the JANUS application in the *display_contrast_level parameter of the im_get_contrast function call.</p> <p>The value held in memory by the Simulator TSR can be modified by the im_set_contrast function.</p>

---

***display\_size\_mode***

<b>Description</b>	Specifies the size mode of the JANUS display.
<b>Default</b>	0 – 80 x 25
<b>Values</b>	0 – 80 x 25 1 – 20 x 16 2 – 20 x 8 3 – 10 x 16 4 – 10 x 8
<b>Function</b>	im_get_display_mode
<b>Interrupt</b>	INT 79H
<b>Notes</b>	<p>The value of this INI parameter is returned to the JANUS application in the *size_mode parameter of the im_get_display_mode function call.</p> <p>The value held in memory by the Simulator TSR can be modified by the im_set_display_mode function.</p>

---

## ***display\_video\_mode***

<b>Description</b>	Specifies the video mode of the JANUS display.
<b>Default</b>	3 – Standard video mode 3 (80 x 25)
<b>Values</b>	0 – Standard video mode 0 (40 x 25) 1 – Standard video mode 1 (40 x 25) 2 – Standard video mode 2 (80 x 25) 3 – Standard video mode 3 (80 x 25) 4 – Video mode 4 (graphics) 5 – Video mode 5 (graphics) 6 – Video mode 6 (graphics)
<b>Function</b>	im_get_display_mode
<b>Interrupt</b>	INT 79H
<b>Notes</b>	The value of this INI parameter is returned to the JANUS application in the *video_mode parameter of the im_get_display_mode function call.  The value held in memory by the Simulator TSR can be modified by the im_set_display_mode function.

---

***display\_scroll\_mode***

<b>Description</b>	Specifies the line number at which the JANUS display will scroll when the display size is set to 80 x 25.
<b>Default</b>	1 – Scroll at 16
<b>Values</b>	0 – Scroll at line 25 1 – Scroll at 16 2 – Scroll at line 8
<b>Function</b>	im_get_display_mode
<b>Interrupt</b>	INT 79H
<b>Notes</b>	<p>The value of this INI parameter is returned to the JANUS application in the *scroll_mode parameter of the im_get_display_mode function call.</p> <p>This parameter is in effect only when the display size is set to 80 x 25.</p> <p>The value held in memory by the Simulator TSR can be modified by the im_set_display_mode function.</p>

---

## ***display\_viewport\_lock***

<b>Description</b>	Specifies if the JANUS viewport is locked or unlocked.
<b>Default</b>	0 – Locked
<b>Values</b>	0 – Locked 1 – Unlocked
<b>Function</b>	im_get_viewport_lock
<b>Interrupt</b>	INT 79H
<b>Notes</b>	<p>If you choose 0 for this parameter, 1 - IM_LOCKED is returned in the *viewport_lock parameter of the im_get_viewport_lock function call.</p> <p>If you choose 1 for this parameter, 0 - IM_UNLOCKED is returned in the *viewport_lock parameter of the im_get_viewport_lock function call.</p> <p>The value held in memory by the Simulator TSR can be modified by the im_set_viewport_lock function.</p>

---

## ***display\_x\_pos***

<b>Description</b>	Specifies the horizontal offset for the left side of the simulated viewport.
<b>Default</b>	0
<b>Values</b>	Numeric value from 0 to 60
<b>Function</b>	im_viewport_getxy
<b>Interrupt</b>	INT 79H
<b>Notes</b>	<p>The value of this INI parameter is returned to the JANUS application in the *col parameter of the im_viewport_getxy function call.</p> <p>The Simulator TSR does not simulate viewport movement, but it does return the X and Y values to the im_viewport_getxy function:</p> <ul style="list-style-type: none"><li>• The X and Y parameters represent the position where the upper left corner of the JANUS viewport appear on the PC screen.</li><li>• You cannot set X to a value larger than 60. If you could, the viewport would be positioned beyond the left edge of the PC screen.</li></ul>

You can change the X value in memory by calling:

- im\_viewport\_end
- im\_viewport\_home
- im\_viewport\_move
- im\_viewport\_setxy
- im\_viewport\_to\_cursor
- im\_viewport\_page\_up
- im\_viewport\_page\_down

---

## ***display\_y\_pos***

**Description** Specifies the vertical offset for the left side of the simulated viewport.

**Default** 0

**Values** Numeric value from 0 to 9

**Function** im\_viewport\_getxy

**Interrupt** INT 79H

**Notes:** The value of this INI parameter is returned to the JANUS application in the \*row parameter of the im\_viewport\_getxy function call.

The Simulator TSR does not simulate viewport movement, but it does return the X and Y values to the im\_viewport\_getxy function:

- The X and Y parameters represent the position where the upper left corner of the JANUS viewport appear on the PC screen.
- You cannot set X to a value larger than 60. If you could, the viewport would be positioned beyond the left edge of the PC screen.

You can change the Y value in memory by calling:

- im\_viewport\_end
- im\_viewport\_home
- im\_viewport\_move
- im\_viewport\_setxy
- im\_viewport\_to\_cursor
- im\_viewport\_page\_up
- im\_viewport\_page\_down

---

## ***display\_backlight***

<b>Description</b>	Specifies the starting state of the JANUS backlight.
<b>Default</b>	0 – Backlight off
<b>Values</b>	0 – Backlight off 1 – Backlight on
<b>Function</b>	im_backlight_toggle
<b>Interrupt</b>	INT 79H
<b>Notes</b>	You can change the backlight status by calling these functions: <ul style="list-style-type: none"><li>• im_backlight_on</li><li>• im_backlight_off</li><li>• im_backlight_toggle</li></ul>

---

## ***display\_char\_height***

<b>Description</b>	Specifies if the characters are displayed at standard height or double height.
<b>Default</b>	0 – Standard height characters
<b>Values</b>	0 – Standard height characters 1 – Double height characters
<b>Function</b>	im_get_display_mode
<b>Interrupt</b>	INT 79H
<b>Notes</b>	<p>The value of this INI parameter is returned to the JANUS application in the *char_ht parameter of the im_get_display_mode function call.</p> <p>This parameter does not affect the characters displayed on the development PC.</p> <p>The value held in memory by the Simulator TSR can be modified by the im_set_display_mode function.</p>

---

## ***appl\_break\_status***

<b>Description</b>	Specifies if the user entered the application break key sequence to interrupt applications running on the JANUS.
<b>Default</b>	0 – Request for break has not been made
<b>Values</b>	0 – Request for break has not been made 0FFH – Request for break has been made
<b>Function</b>	im_appl_break_status
<b>Interrupt</b>	INT 7EH

---

## ***follow\_cursor***

<b>Description</b>	Specifies if the viewport will follow the cursor as it moves around the simulated display.
<b>Default</b>	1 – Follow cursor mode on
<b>Values</b>	0 – Follow cursor mode off 1 – Follow cursor mode on
<b>Function</b>	im_get_follow_cursor
<b>Interrupt</b>	INT 7EH
<b>Notes</b>	<p>The value of this INI parameter is returned to the JANUS application in the *follow_cursor parameter of the im_get_follow_cursor function call.</p> <p>The value held in memory by the Simulator TSR can be modified by the im_set_follow_cursor function.</p>

---

## ***control\_key***

<b>Description</b>	Specifies if the Control key is enabled or disabled for the simulated <code>im_get_control_key</code> function.
<b>Default</b>	0 – Control key disabled
<b>Values</b>	0 – Control key disabled 0FFH – Control key enabled
<b>Function</b>	<code>im_get_control_key</code>
<b>Interrupt</b>	INT 7EH
<b>Notes</b>	<p>The value of this INI parameter is returned to the JANUS application in the <code>*control_key</code> parameter of the <code>im_get_control_key</code> function call.</p> <p>The <code>im_get_control_key</code> function does not enable or disable the control key on the development PC.</p> <p>The value held in memory by the Simulator TSR can be modified by the <code>im_set_control_key</code> function.</p>

---

**keyclick**

<b>Description</b>	Specifies if the JANUS emits a click each time a key is pressed.
<b>Default</b>	0 – Keyclick off
<b>Values</b>	0 – Keyclick off 0FFH – Keyclick on
<b>Function</b>	im_get_keyclick
<b>Interrupt</b>	INT 7EH
<b>Notes</b>	<p>The value of this INI parameter is returned to the JANUS application in the *keyclick parameter of the im_get_keyclick function call.</p> <p>The value held in memory by the Simulator TSR can be modified by the im_set_keyclick function.</p>

---

## ***numeric\_keypad\_switching***

<b>Description</b>	Specifies if the user can switch the numeric keypad from numeric to function keys.
<b>Default</b>	0FFH – Numeric keypad switching permitted
<b>Values</b>	0 – Numeric keypad switching prohibited 0FFH – Numeric keypad switching permitted
<b>Function</b>	None
<b>Interrupt</b>	INT 7EH
<b>Notes</b>	<p>This parameter is provided for interrupt level support only; there is no JANUS PSK library function associated with this parameter.</p> <p>The value held in memory by the Simulator TSR can be modified by the <code>im_number_pad_on</code> and <code>im_number_pad_off</code> functions.</p>

---

***numeric\_keypad\_status***

<b>Description</b>	Specifies if the numeric keypad is in numeric or function key mode.
<b>Default</b>	0 – Numeric keypad disabled
<b>Values</b>	0 – Numeric keypad disabled 0FFH – Numeric keypad enabled
<b>Function</b>	None
<b>Interrupt</b>	INT 7EH
<b>Notes</b>	<p>This parameter is provided for interrupt level support only; there is no JANUS PSK library function associated with this parameter.</p> <p>The value held in memory by the Simulator TSR can be modified by the <code>im_number_pad_on</code> and <code>im_number_pad_off</code> functions.</p>

---

## ***label\_symbology***

<b>Description</b>	Specifies the symbology of the last simulated scanned label.
<b>Default</b>	1 – Code 39
<b>Values</b>	0 – Unknown 1 – Code 39 2 – Code 93 3 – Code 49 4 – Interleaved 2 of 5 (I 2 of 5) 5 – Codabar 6 – UPC/EAN 7 – Code 128 8 – Code 16K 9 – Plessey 10 – Code 11 11 – MSI
<b>Function</b>	im_get_label_symbology im_receive_input im_irl_a im_irl_n, im_irl_y
<b>Interrupt</b>	INT 7DH
<b>Notes</b>	<p>The value of this INI parameter is returned to the JANUS application in the *symbology parameter of the function calls that use this INI parameter.</p> <p>When testing a JANUS application, you can simulate the act of scanning a label by pressing the key sequence specified in the sim_wand_key parameter. The label_symbology parameter specifies the symbology for the simulated label.</p>

---

## ***sim\_wand\_key***

<b>Description</b>	Specifies the key sequence that causes the JANUS application to accept subsequent keyboard input as if it were wand input.
<b>Default</b>	<b>Ctrl-G</b>
<b>Values</b>	A key combination that includes one or more control keys ( <b>Ctrl</b> , <b>Alt</b> , <b>Shift</b> ) and a character key ( <b>A</b> to <b>Z</b> ).
<b>Function</b>	None
<b>Interrupt</b>	None
<b>Notes</b>	The user presses this key sequence and types data that the application accepts as input from a wand. The user presses <b>Enter</b> to indicate the end of the simulated wand input.

You can set the `sim_wand_key` parameter equal to one or more control keys (**Ctrl**, **Alt**, **Shift**) and a character key (**A** to **Z**). Here are some sample key sequences:

- **Alt-L**
- **Alt-Shift-B**
- **Ctrl-Shift-C**
- **Ctrl-Alt-Shift-W**

The INI file displays a slash in the key sequence (**Ctrl/G**) instead of a hyphen. By convention, this manual uses a hyphen to indicate that you must press multiple keys simultaneously.

---

### ***clear\_abort\_callback\_return***

<b>Description</b>	Specifies the return value upon clearing the abort callback routine.
<b>Default</b>	0500H – Reader Wedge success
<b>Values</b>	See the “Reader Wedge Standard Return Values List” on page 3-40.
<b>Function</b>	im_clear_abort_callback
<b>Interrupt</b>	INT 7DH

---

### ***command\_return***

<b>Description</b>	Specifies the return value after issuing a configuration command.
<b>Default</b>	0500H – Reader Wedge success
<b>Values</b>	See the “Reader Wedge Standard Return Values List” on page 3-40.
<b>Function</b>	im_command
<b>Interrupt</b>	INT 7DH

---

## ***link\_comm\_return***

<b>Description</b>	Specifies the return value after the application attempts to link a Reader Wedge and a designated COM port.
<b>Default</b>	0500H – Reader Wedge success
<b>Values</b>	See the “Reader Wedge Standard Return Values List” on page 3-40.
<b>Function</b>	im_link_comm
<b>Interrupt</b>	INT 7DH
<b>Notes</b>	Once a JANUS application calls the im_link_comm PSK function, other PSK functions that require a link or unlink will operate as if the link were successfully established—even if you select a value for the link_comm_return parameter that indicates the im_link_comm function failed.

---

## ***serial\_protocol\_control\_return***

<b>Description</b>	Specifies the return value for the simulated im_serial_protocol_control function.
<b>Default</b>	0500H – Reader Wedge success
<b>Values</b>	See the “Reader Wedge Standard Return Values List” on page 3-40.
<b>Function</b>	im_serial_protocol_control
<b>Interrupt</b>	INT 7DH

---

### ***standby\_wait\_return***

<b>Description</b>	Specifies the return value for the simulated im_standby_wait function.
<b>Default</b>	0500H – Reader Wedge success
<b>Values</b>	See the “Reader Wedge Standard Return Values List” on page 3-40.
<b>Function</b>	im_standby_wait
<b>Interrupt</b>	INT 7DH

---

### ***unlink\_comm\_return***

<b>Description</b>	Specifies the return value for the simulated im_unlink_comm function.
<b>Default</b>	0500H – Reader Wedge success
<b>Values</b>	See the “Reader Wedge Standard Return Values List” on page 3-40.
<b>Function</b>	im_unlink_comm
<b>Interrupt</b>	INT 7DH

## ***Communications Standard Return Values***

---

Communications standard return values are sent to a JANUS application to indicate the result of a communications function call, such as `im_rx_check_status`.

When you customize the INI file for the Simulator TSR, you can set these four parameters to a communications return value:

- `rx_check_status_return`
- `cancel_rx_buff_return`
- `protocol_ext_status_return`
- `cancel_tx_buff_return`

---

### ***Communications Standard Return Values List***

<b>Code</b>	<b>Description</b>
0600H	Communication operation successful
0601H	Communication buffer done
4600H	Warning buffer canceled
4601H	Communication timeout warning
4602H	Communication had no client to cancel
4603H	No protocol handler to configure
4604H	Protocol handler is not yet loaded
4605H	Protocol handler already receiving
4607H	Protocol handler not active for link request
8600H	Invalid configuration
8601H	Buffer length 0 error
8602H	Communication port is in use
8603H	Protocol error
8604H	Communication port error
8605H	Communication port is busy
8606H	Communication request is not supported
8607H	Protocol handler already loaded

## ***JANUS Application Simulator User's Manual***

<b>Code</b>	<b>Description</b>
8608H	Protocol buffer error
8609H	Unknown service request
860AH	No data available
860BH	Communication utility is not loaded
860CH	Communication utility resume/suspend failure
860DH	Communication utility INT 14 is already in use
860EH	Incompatible revision between protocol handler and communication utility
860FH	Non supported communication service
8610H	Invalid EOF character
8611H	Protocol handler is not active
8612H	Buffer must be 256 bytes or larger

## ***Reader Wedge Standard Return Values***

---

The Reader Wedge standard return values are codes that are sent to a JANUS application to indicate the result of a Reader Wedge function call, such as `link_comm_return`.

When you customize the INI file for the Simulator TSR, you can set these six parameters to a Reader Wedge return value:

- `clear_abort_callback_return`
- `command_return`
- `link_comm_return`
- `serial_protocol_control_return`
- `standby_wait_return`
- `unlink_comm_return`

---

**Reader Wedge Standard Return Values List**

<b>Code</b>	<b>Description</b>
0500H	Reader Wedge success
8501H	Input request error
4502H	Input timeout
4503H	No input data
8504H	Illegal reader wedge mode
0505H	No reader commands parsed
0506H	Valid reader commands parsed
8507H	Reader commands parsing error
8508H	Invalid configuration error
0509H	Accumulating reader command
050AH	Reader command override
050BH	Reader command enter accumulate
050CH	Reader command exit accumulate
050DH	Accumulate multiread labels
050EH	The application should be notified
850FH	Error in reader command edit
0510H	An ENTER reader command parsed
8511H	Unable to allocate memory for input
8512H	Protocol handler is not linked
0513H	Protected field has been parsed
8514H	Parse error on exit accumulate
8515H	Application break detected
8516H	Invalid reader wedge request
0517H	Keycode label has been parsed
0518H	Label not accepted (application is not requesting input)
0519H	Available buffers for comms link
051AH	No comms status to report
851BH	Invalid protocol for input request
051CH	Xmit buffer busy or Comm error
051DH	Exit Accumulate with Comms error



# 4

## ***Customizing INI Files With the Editor***



*This chapter explains why you customize the initialization (INI) file and describes how to do so with the Windows-based Editor.*

## ***Why Customize INI Files?***

---

*Note: Customizing the INI file is an optional process. You do not have to customize the INI file if you are satisfied with the default values in JANUSSIM.INI.*

The INI parameters control how the Simulator TSR simulates a JANUS reader executing an application. You can customize the parameters so the Simulator TSR mimics the conditions against which you want to test your JANUS applications.

For example, if your JANUS application checks the battery's status, you could use an INI file to mimic a low power condition and test how your application responds. Follow these steps:

1. Create a LOWBAT.INI file and customize these parameters to simulate a low power condition:
  - power\_status\_line = 0 - AC line not connected
  - power\_status\_battery = 2 - critical battery
  - power\_status\_backup = 1 - backup low
  - power\_status\_batt\_life = 10
  - power\_status\_return = 0800H - Power management success
2. Start the Simulator TSR by typing this command at the DOS prompt:  
`janussim lowbat.ini Enter`
3. Run your JANUS application. When the application issues the `im_power_status` function call or the 5EH interrupt, it will find that the reader is not plugged into an AC power source, the battery is critically low, and the backup battery is low.
4. Test how the application responds to this low power condition.

## ***JANUS Application Simulator User's Manual***

Here is the LOWBAT.INI file with the customized parameters shaded for emphasis:

```
;
; Filename: LOWBAT.INI
;
port1_read_file = comport1.rcv
port2_read_file = comport2.rcv
port3_read_file = comport3.rcv
port4_read_file = comport4.rcv
port1_write_file = comport1.trx
port2_write_file = comport2.trx
port3_write_file = comport3.trx
port4_write_file = comport4.trx
rx_check_status_return = 0600H - Comm Operation successful
protocol_ext_status_return = 0600H - Comm Operation successful
cancel_rx_buff_return = 0600H - Comm Operation successful
cancel_tx_buff_return = 0600H - Comm Operation successful
UART_status = 30H
expanded_keyboard_status = 0 - command complete
kb_insert_string_status = 0 - success
kb_insert_string_capacity = 255
flush_exp_keyboard_status = 0 - success
warmboot_status = 0 - warmboot disabled
power_status_line = 0 - AC line not connected
power_status_battery = 2 - critical battery
power_status_backup = 1 - backup low
power_status_batt_life = 10
power_status_return = 0800H - Power management success
display_contrast_level = 16
display_size_mode = 0 - 80 x 25
display_video_mode = 3 - std video mode 3 (80x25)
display_scroll_mode = 1 - scroll at 16
display_viewport_lock = 0 - locked
display_x_pos = 0
display_y_pos = 0
display_backlight = 0 - backlight off
display_char_height = 0 - standard height characters
appl_break_status = 0 - break not been made
follow_cursor = 1 - follow cursor mode on
control_key = 0 - control key disabled
keyclick = 0 - keyclick off
```

```
numeric_keypad_switching = 0FFH - numeric keypad switching permitted
numeric_keypad_status = 0 - numeric keypad disabled
label_symbology = 1 - Code 39
sim_wand_key = CTRL/G
clear_abort_callback_return = 0500H - Reader Wedge success
command_return = 0500H - Reader Wedge success
link_comm_return = 0500H - Reader Wedge success
serial_parse_return = 0500H - Reader Wedge success
serial_protocol_control_return = 0500H - Reader Wedge success
standby_wait_return = 0500H - Reader Wedge success
unlink_comm_return = 0500H - Reader Wedge success
```

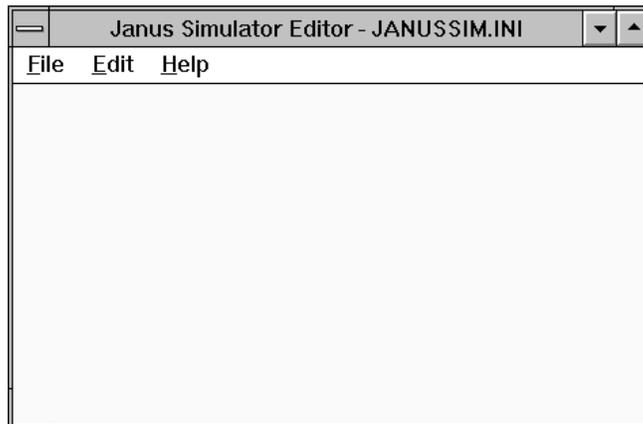
## Starting the Editor

---

To start the Editor, double-click on the Editor icon in the Simulator group on your Windows desktop. The Editor window appears:

The title bar contains the JANUSSIM.INI filename because the default initialization file is automatically opened when you start the Editor.

You can begin editing the parameters in the JANUSSIM.INI file, create a new INI file, or open an existing INI file.



If the Simulator TSR is running while you use the Editor, you can load the current INI file from the Editor into the Simulator TSR. You might do this to load a new version of the same INI file that the Simulator TSR is using, or to load a different INI file into the Simulator TSR. For help, see “Updating the Simulator TSR With the Current INI File” later in this chapter.

## ***Using Online Help***

---

The online help explains how to use the Editor and describes the parameters in the initialization file.

### **To access online help**

You can use one of these methods for accessing the Editor's online help:

- From any Editor screen that has a Help button, choose Help. A help screen appears, containing information about the Editor window.
- From the Help drop-down menu, choose Contents. An online help screen displays a list of topics. Double-click on any item in the Contents list for more information.
- From the Intermec group on your Windows desktop, you can double-click on the Sim Editor Help icon. An online help screen displays the Contents. Double-click on any item in the Contents list for more information.

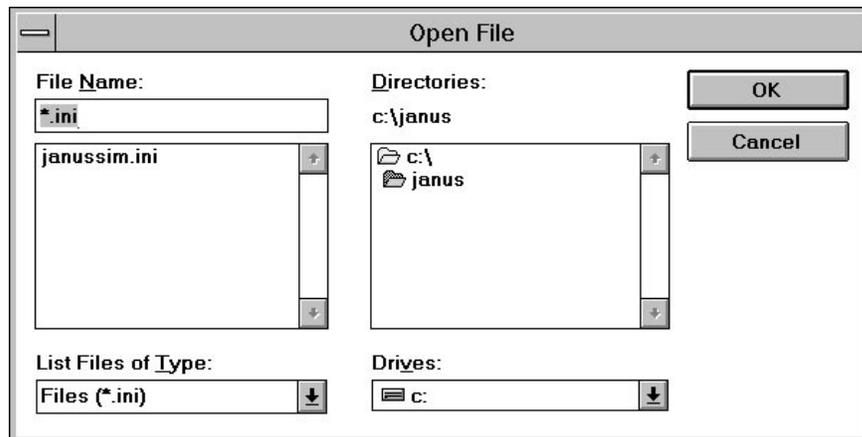
## Creating a New INI File

---

You can create and customize new initialization files with the Editor. Each new file is a duplicate of the JANUSSIM.INI file with all the parameters set to their default values.

### To create a new INI file

1. From the File menu, choose Open. The Open File window appears:



2. In the File Name field, enter the name of the new file you are creating.
3. Make sure the current directory, as shown in the Directories list box, is the directory where you want to save the new file. If not, select a different directory.
4. Choose OK.

The new INI file is created, and the parameters are set to the defaults. You return to the main menu, and the new filename appears in the title bar.

You can begin customizing the parameters immediately:

- For help choosing new values for the parameters, see Chapter 3.
- For help setting the parameters, see “Setting Parameters” later in this chapter.

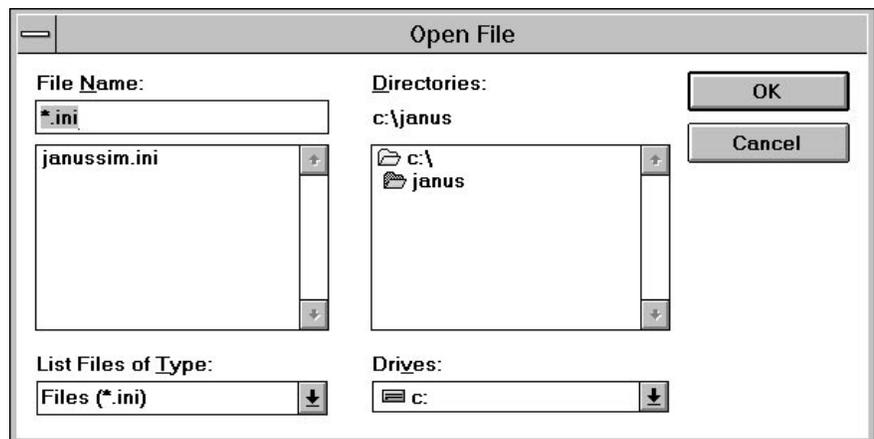
## Opening an Existing INI File

---

You can open an existing initialization file to view, edit, or print the file.

### To open an existing INI file

1. From the File menu, choose Open. The Open File window appears:



2. From the File Name list box, select the name of the file you want to open.

If the filename does not appear in the list box, make sure the current directory, as shown in the Directories list box, is the one where the file is stored. If not, select a different directory.

3. Choose OK.

The file is opened. You return to the main menu, and the filename appears in the title bar.

You can begin customizing the parameters immediately:

- For help choosing new values for the parameters, see Chapter 3.
- For help setting the parameters, read the following sections.

## Setting Parameters

---

The Editor groups the parameters into six topics:

<b>Communications</b>	cancel_rx_buff_return cancel_tx_buff_return portn_read_file portn_write_file	protocol_ext_status_return rx_check_status_return UART_status
<b>Keyboard</b>	expanded_keyboard_status flush_exp_keyboard_status kb_insert_string_capacity	kb_insert_string_status warmboot_status
<b>Power Management</b>	power_status_backup power_status_batt_life power_status_battery	power_status_line power_status_return
<b>Display</b>	display_backlight display_char_height display_contrast_level display_scroll_mode display_size_mode	display_video_mode display_viewport_lock display_x_pos display_y_pos
<b>Keypad</b>	appl_break_status follow_cursor control_key	keyclick numeric_keypad_switching numeric_keypad_status
<b>Reader Wedge</b>	label_symbology sim_wand_key clear_abort_callback_return command_return	link_comm_return serial_protocol_control_return standby_wait_return unlink_comm_return

For descriptions of the parameters, see Chapter 3. For help customizing the parameters, see the following sections.

---

## **Setting Communications Parameters**

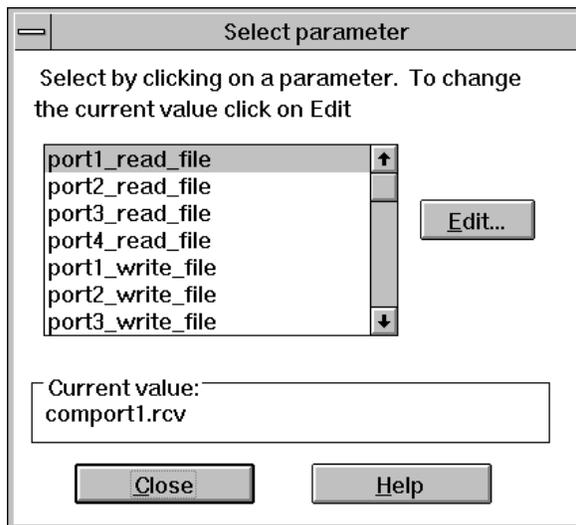
You can view or set the values for the communications parameters.

### **To set a communications parameter**

1. From the Edit menu, choose Communications. This Select parameter window appears:

The communications parameters are contained in the list box.

2. To view the current value for a parameter, select the parameter. The current value appears below the list box.
3. To customize a parameter, either double-click on it or select it and choose Edit.



Another window appears, displaying the values you can choose for the parameter. At that window, select a value and choose OK. You return to this Select parameter window.

4. Choose Close to return to the main menu.

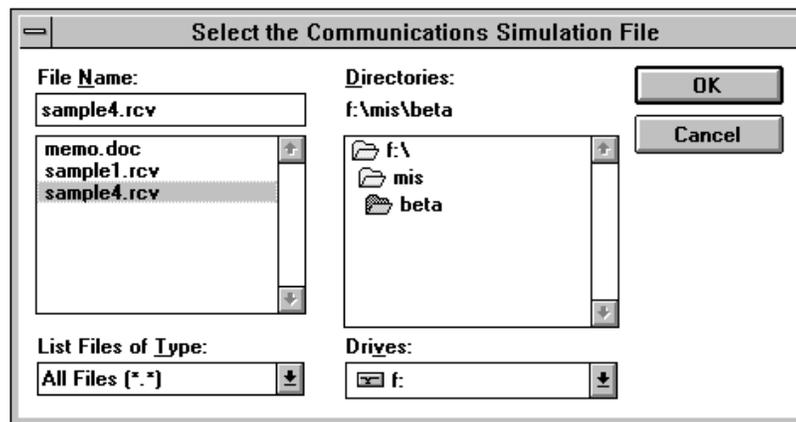
### ***Naming portn\_read\_file and portn\_write\_file***

The `portn_read_file` and `portn_write_file` parameters identify the ASCII text files that the Simulator TSR uses to simulate reading from and writing to the JANUS COM ports (1 to 4).

If you specify new filenames for these parameters, do **not** use `comn.rcv` and `comn.trx`. Those filenames will cause your PC to associate input and output with its communication ports.

### ***Specifying a Path for portn\_read\_file and portn\_write\_file***

If you use the Editor to select a file for the `portn_read_file` or `portn_write_file` parameters, the Editor lets you select a file in any directory or on any drive that is accessible from your PC. In this example, a user selects a file in a directory on Drive F:



However, the Editor saves only the filename to the INI file, not the path. For example, if you check the INI file after saving this change, you see:

```
port4_read_file = sample4.rcv
```

If the file will not be in the current directory when you run the JANUS application on your PC (or in a directory listed in the PATH statement in your AUTOEXEC.BAT file), you must edit the INI file and add the file's path to the filename. For example:

```
port4_read_file = f:\mis\beta\sample4.rcv
```

---

## **Setting Keyboard Parameters**

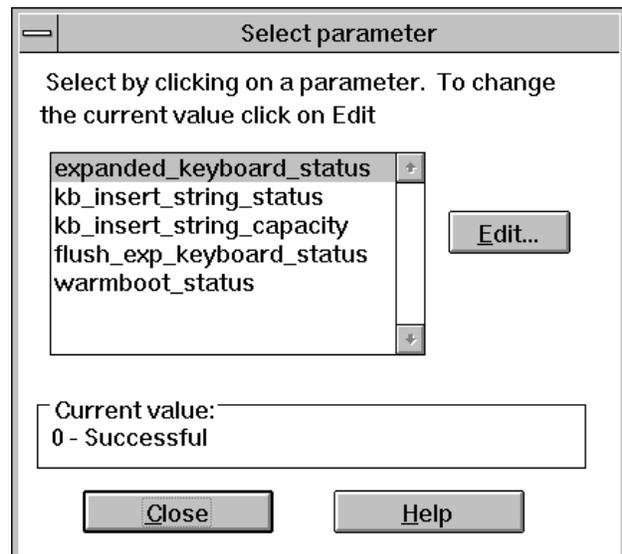
You can view or set the values for the keyboard parameters.

### **To set a keyboard parameter**

1. From the Edit menu, choose Keyboard. This Select parameter window appears:

The keyboard parameters are contained in the list box.

2. To view the current value for a parameter, select the parameter. The current value appears below the list box.
3. To customize a parameter, either double-click on it or select it and choose Edit.



Another window appears, displaying the values you can choose for the parameter. At that window, select a value and choose OK. You return to this Select parameter window.

4. Choose Close to return to the main menu.

## Setting Power Management Parameters

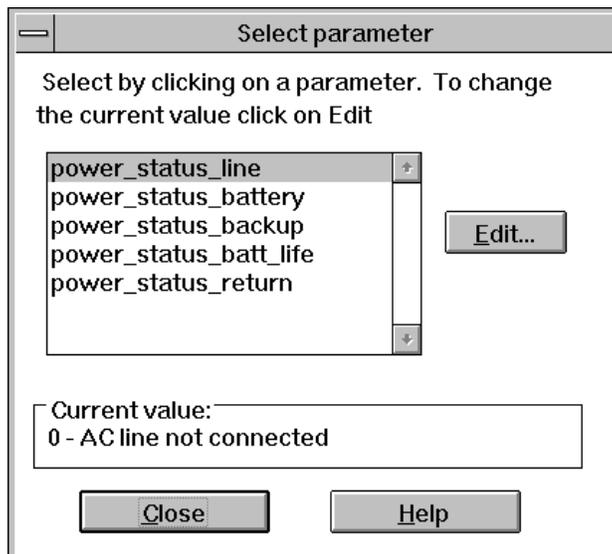
You can view or set the values for the power management parameters.

### To set a power management parameter

1. From the Edit menu, choose Power Management. This Select parameter window appears:

The power management parameters are contained in the list box.

2. To view the current value for a parameter, select the parameter. The current value appears below the list box.
3. To customize a parameter, either double-click on it or select it and choose Edit.



Another window appears, displaying the values you can choose for the parameter. At that window, select a value and choose OK. You return to this Select parameter window.

4. Choose Close to return to the main menu.

---

## **Setting Display Parameters**

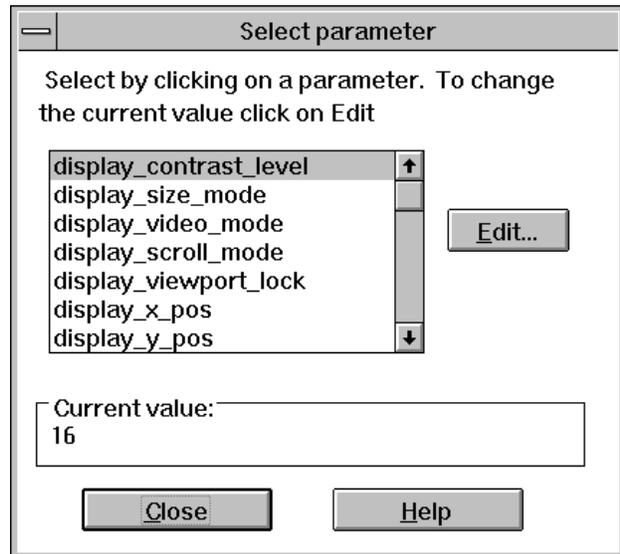
You can view or set the values for the display parameters.

### **To set a display parameter**

1. From the Edit menu, choose Display. This Select parameter window appears:

The display parameters are contained in the list box.

2. To view the current value for a parameter, select the parameter. The current value appears below the list box.
3. To customize a parameter, either double-click on it or select it and choose Edit.



Another window appears, displaying the values you can choose for the parameter. At that window, select a value and choose OK. You return to this Select parameter window.

4. Choose Close to return to the main menu.

## Setting Keypad Parameters

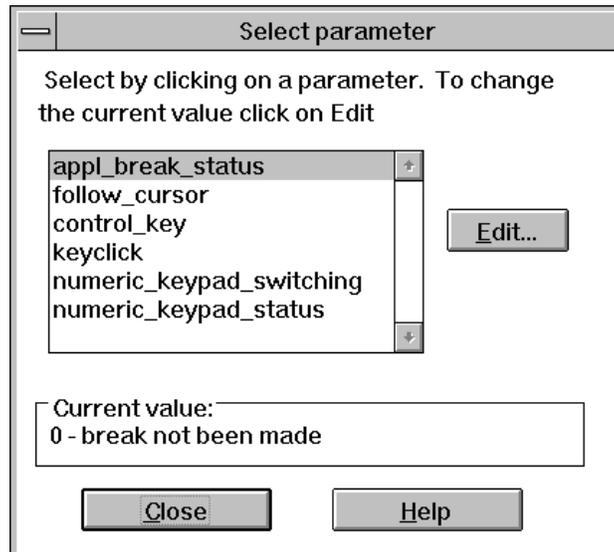
You can view or set the values for the keypad parameters.

### To set a keypad parameter

1. From the Edit menu, choose Keypad. This Select parameter window appears:

The keypad parameters are contained in the list box.

2. To view the current value for a parameter, select the parameter. The current value appears below the list box.
3. To customize a parameter, either double-click on it or select it and choose Edit.



Another window appears, displaying the values you can choose for the parameter. At that window, select a value and choose OK. You return to this Select parameter window.

4. Choose Close to return to the main menu.

---

## Setting Reader Wedge Parameters

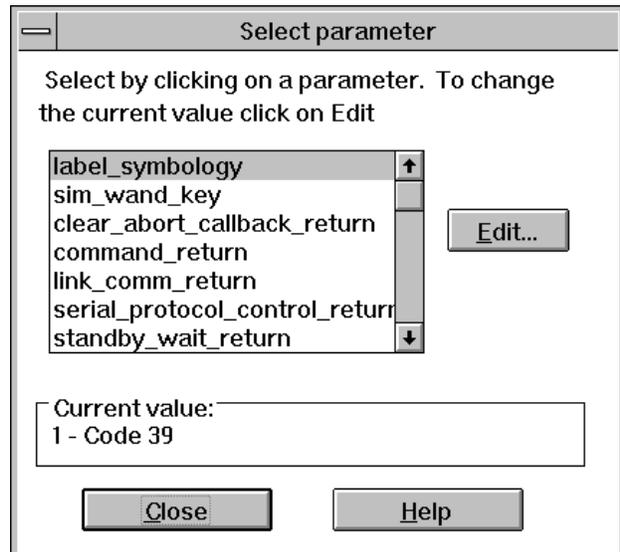
You can view or set the values for the Reader Wedge parameters.

### To set a Reader Wedge parameter

1. From the Edit menu, choose Reader Wedge. This Select parameter window appears:

The Reader Wedge parameters are contained in the list box.

2. To view the current value for a parameter, select the parameter. The current value appears below the list box.
3. To customize a parameter, either double-click on it or select it and choose Edit.



Another window appears, displaying the values you can choose for the parameter. At that window, select a value and choose OK. You return to this Select parameter window.

4. Choose Close to return to the main menu.

## Saving Changes

---

You can save the changes you make to an INI file in these ways:

- Save the changes to the current INI file.
- Save the changes to the current INI file when you exit the Editor.
- Save the changes into a new INI file.
- Save the changes into an existing INI file.

*Note: The current file is the file that is currently open. The current file's name is displayed in the title bar of the Editor window.*

### To save the changes into the current INI file

1. From the File menu, choose Save. The Editor saves the file and displays the message:

File: *name* Saved

For example, File: LOWBAT.INI Saved

2. Choose OK.

### To save the changes into the current INI file when you exit the Editor

1. From the File menu, choose Exit.

If you made changes to the INI file that you have not saved yet, the Editor displays the message:

Do you want to save changes to file *name*?

2. To save the changes into the current INI file, choose the Save button. The changes are saved, the Editor shuts down, and you return to your Windows desktop.

You have other choices:

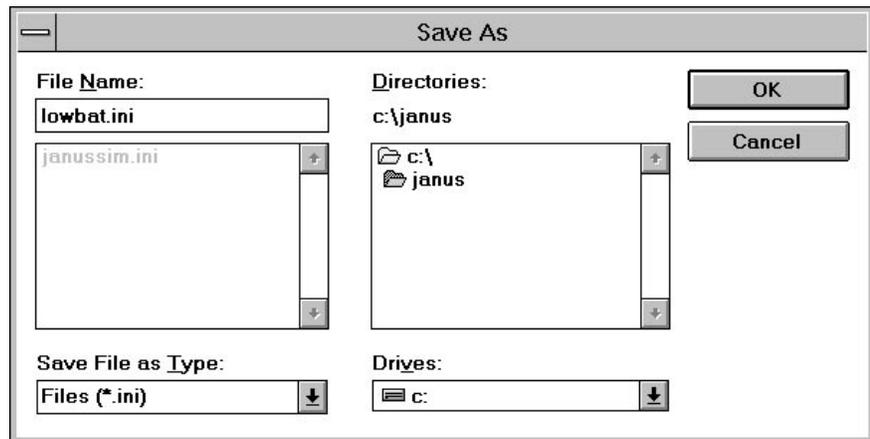
- To discard the changes, choose the No Save button. The changes are not saved, the Editor shuts down, and you return to your Windows desktop.

## JANUS Application Simulator User's Manual

- To avoid saving or discarding, choose the Cancel button. You return to the Editor main menu. You can save the changes to another INI file or perform some other function.

### To save the changes in a new INI file

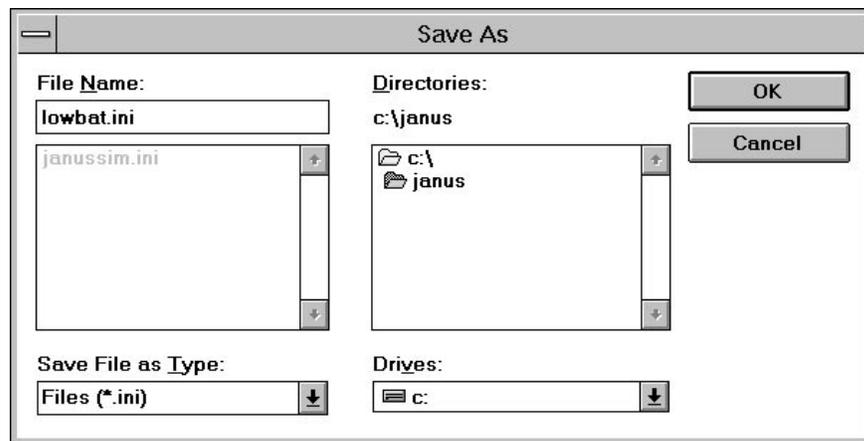
1. From the File menu, choose Save As. The Save As window appears:



2. Make sure the current directory, as shown in the Directories list box, is the one where you want to store the new file. If not, choose another directory.
3. In the File Name field, enter the new filename.
4. Choose OK to save the changes in the new file you specified. The Editor creates the new file, saves the changes, and displays this message:  
File: *name* Saved
5. Choose OK. You return to the main menu.

**To save the changes in an existing INI file**

1. From the File menu, choose Save As. The Save As window appears:



2. From the File Name list box, select the name of the file where you want to save the changes.

If the filename does not appear in the list box, make sure the current directory, as shown in the Directories list box, is the one where the file is stored. If not, select a different directory.

3. Choose OK to save the changes in the file you selected. The Editor displays this message:

File already exists. Replace existing file?

Choose Yes to overwrite the contents of the file with the changes you have made. You can also choose No to cancel this operation and return to the main menu.

If you choose Yes, the Editor saves the changes into the file and displays this message:

File: *name* Saved

4. Choose OK. You return to the main menu.

## ***Discarding Changes***

---

You can discard changes when you exit the Editor.

### **To discard the changes**

1. From the File menu, choose Exit. If you made changes to the INI file that you have not saved yet, the Editor displays the message:

Do you want to save changes to file *name*?

2. To discard the changes, choose the No Save button. The changes are not saved, the Editor shuts down, and you return to your Windows desktop.

## ***Restoring the Default Values***

---

Sometimes you will want to reset all the parameters in the current INI file to their default settings. You can do this at any time when the Editor is running. For a description of the parameters and their defaults, see Chapter 3.

### **To restore the defaults**

1. From the File menu, choose Restore Defaults. The Editor displays this message:

All parameters will be reset to their default values. Do you want to proceed?

2. Choose Restore to restore the defaults. The Editor resets the parameters to their defaults and displays the message:

All defaults have been restored.

3. Choose OK.

## **Updating the Simulator TSR With the Current INI File**

The Simulator TSR can be running while you use the Editor to create or edit initialization files. If the TSR is running, you can load a new or changed INI file into the Simulator TSR without exiting the Editor or stopping the Simulator TSR.

For example, if you start the Simulator TSR with the default INI file, JANUSSIM.INI, you can use the Editor to:

- Create a new INI file called LOWBAT.INI and load the new file into the Simulator TSR as it runs in the background.
- Edit the default INI file, JANUSSIM.INI, and load the changed file into the Simulator TSR as it runs in the background.

### **To update the Simulator TSR with a new or changed INI file**

1. Make sure the Simulator TSR is running.
2. Start the Editor. Either create or edit any INI file.
2. From the File menu, choose Update Simulator.
3. If the Simulator TSR is running, a status message appears:

Simulator has been updated.

Choose OK. You return to the main menu. The Simulator TSR has been updated with the current INI file.

4. If the Simulator TSR is not running, an error message appears:

The Simulator TSR is not running. Please exit the Editor, shut down Windows, and start the TSR.

Follow the instructions in the message. To start the TSR with a specific INI file, enter this command at the MS-DOS prompt:

```
janussim filename.ini
```

Where *filename.ini* is the name of the INI file.

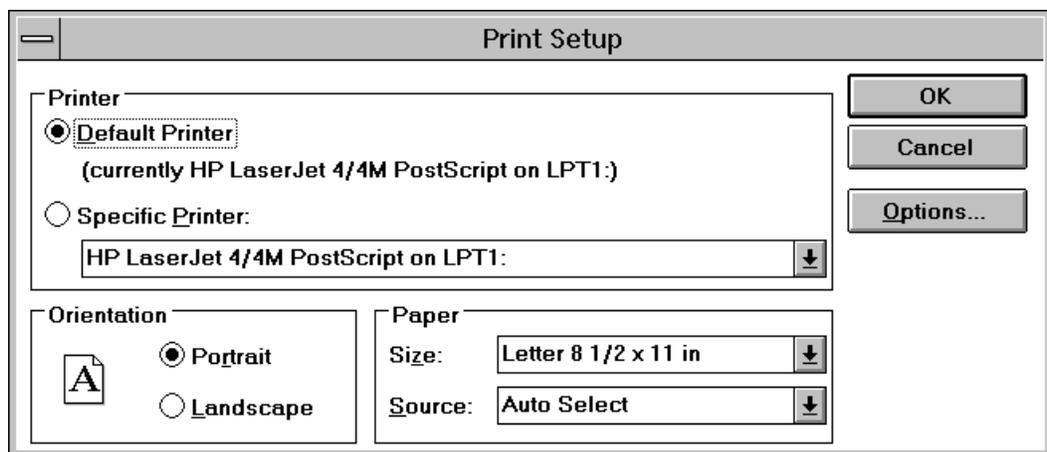
## Setting Up the Printer

---

You can set up the printer to which the Editor prints the INI file.

### To set up the printer

1. From the File menu, choose Printer Setup. The standard Windows Print Setup window appears:



2. Set up your printer as you would for any Windows application.
3. Choose OK. You return to the main menu.

## Printing INI Files

---

You can print initialization files with the Editor or with any ASCII text editor. Printing INI files is a good way to keep track of contents of the INI files, especially if you are using multiple files.

### To print an INI file

1. From the File menu, choose Print.

2. The Editor displays the message:

Printing document.

To cancel the print job, you can choose Cancel on the message box.

## ***Exiting the Editor***

---

When you exit the Editor, you shut down the Editor and close the current INI file. If you changed the current file and have not saved those changes yet, the Editor prompts you to save or discard the changes.

### **To exit the Editor**

1. From the File menu, choose Exit.

If you saved all changes to the current INI file, the Editor simply shuts down and you return to your Windows desktop.

If you made changes to the INI file that you have not saved yet, the Editor displays the message:

Do you want to save changes to file *name*?

Choose one of the following:

- To save the changes into the current INI file, choose the Save button. The changes are saved, the Editor shuts down, and you return to your Windows desktop.
- To discard the changes, choose the No Save button. The changes are not saved, the Editor shuts down, and you return to your Windows desktop.
- To avoid saving or discarding, choose the Cancel button. You return to the Editor main menu. This gives you the chance to save the changes to another INI file or perform some other function.



# 5

## ***Troubleshooting***



## ***Running an Application Without the Simulator TSR***

---

Start the Simulator TSR before you run a JANUS application on your PC.

If you run a JANUS application on your PC without first starting the Simulator TSR, the JANUS application may lock up your PC. If this happens, reboot your PC immediately.

## ***Simulating Bar Code Input With an Intermec Wedge***

---

If you are having difficulty using an Intermec Wedge to provide bar code input while you run a JANUS application, you may have set the wedge preamble incorrectly.

For help setting the preamble to match the value of the `sim_wand_key` parameter, see “Bar Code Input” in Chapter 1.

## ***Error and Status Messages for the Simulator TSR***

---

The following table describes the error messages you may see when using the Simulator TSR. Follow the instructions in the Suggested Action column to recover from the error.

<b>Message</b>	<b>Description</b>	<b>Suggested Action</b>
Insufficient memory.	This is a DOS error. You tried to load the TSR without sufficient conventional memory.	The TSR requires 50K of RAM. Free up some RAM and reissue the command to load the TSR.
Initialization file does not exist.	This is a DOS error. You tried to load the TSR, and DOS could not locate the INI file.	Make sure the INI file exists and is mentioned in your AUTOEXEC.BAT file as follows: the PATH statement should include the directory, or the JANUSSIM environment variable should point to the directory.
Initialization file invalid.	This is a DOS error. You tried to load the TSR with an invalid INI file.	Verify the file’s type and contents. Recreate the file with the Editor.

## ***Error and Status Messages for the Editor***

---

The following table describes the error messages and status messages you may see when using the Editor. Follow the instructions in the Suggested Action column to recover from the error.

<b>Message</b>	<b>Description</b>	<b>Suggested Action</b>
All defaults have been restored.	You chose Restore Defaults from the File menu, and the Editor reset the parameters to their default values.	No action required.
All parameters will return to default, do you wish to proceed?	You chose the Restore Defaults command from the File menu.	Choose OK to restore the defaults, or choose Cancel to keep the defaults.
Do you want to replace the existing <i>filename</i> ?	You tried to save the current changes in an existing file.	Choose Replace to overwrite the contents of the file with the new information. Choose Cancel to cancel the operation.
Do you want to save changes to <i>filename</i> ?	You chose to exit the Editor without saving changes you made to the current INI file.	Choose Save to save the changes in the current INI file and continue exiting.
	The current file is the file that is currently open. The file's name appears in the title bar of the Editor window.	Choose No Save to discard the changes and continue exiting. Choose Cancel to cancel the exit operation.

Message	Description	Suggested Action
Environment variable JANUSSIM not found.	You started the Editor, which could not locate the JANUSSIM environment variable in your AUTOEXEC.BAT file.	<p>Add this command to your AUTOEXEC.BAT file:</p> <pre>set janussim=path\</pre> <p>Where <i>path</i> is the drive and directory where the JANUS Application Simulator software is installed.</p> <p>The path must have a terminating backslash. For example:</p> <pre>set janussim=c:\janussim\</pre>
File already exists. Replace existing file?	You specified an existing file during the Save As operation.	Choose Yes to overwrite the file with the changes you have made with the Editor. Choose No to cancel the operation.
File name is of improper type! File not opened.	You tried to open an INI file that is not the correct type, or you specified the wrong filename.	<p>Choose OK. Try to open the file again, to make sure you typed the name correctly.</p> <p>If the problem persists, make sure the file has the correct file type. You may have to recreate the file with the Editor to ensure that it has the correct file type.</p>
File: <i>filename</i> Saved.	The Editor has saved the file.	No action required.
Number is out of range. Min = <i>n</i> , Max = <i>n</i> .	You entered a numeric value that is out of range for the parameter.	Enter a numeric value that is between the parameter's minimum and maximum values.
Printing Document	You chose the Print command from the File menu.	No action required.

## ***JANUS Application Simulator User's Manual***

<b>Message</b>	<b>Description</b>	<b>Suggested Action</b>
Simulator has been loaded with <i>filename</i> .	You chose the Load Simulator command from the File menu, and the Editor loaded the changes in the specified INI file.	No action required.
The Simulator TSR is not running. Please exit the Editor, shut down Windows, and start the TSR.	You chose the Load Simulator command from the File menu, but the Simulator TSR is not currently running.	Save the changes to the INI file, exit the Editor, exit Windows, and start the TSR.
This file already exists. Replace existing file?	You specified an existing file during the Save As operation.	Choose Yes to overwrite the file with the changes you have made with the Editor. Choose No to cancel the operation.



***Index***



**A**

ASCII data file, 3-6  
 ASCII text editor, 3-3  
 audience for this manual, xii  
 AUTOEXEC.BAT file  
   JANUSSIM environment variable, 2-4, 5-3,  
   5-5  
   loading TSR at startup, 1-9, 2-3  
   PATH to data files, 3-6, 3-8, 4-11

**B**

bar code input  
   choosing the symbology, 1-11, 3-34  
   with a keyboard, 1-10  
   with a Wedge, 1-9, 1-11  
 battery  
   backup, 3-18  
   example, 1-6  
   percent of charge available, 3-19  
   power management, 1-12  
   status, 3-18  
 break points, 2-4

**C**

cautions, xi  
 COM ports, 1-12, 3-6, 3-8, 4-11  
 communications protocols, 1-13, 3-6, 3-8  
 compatibility, 1-4, 1-9, 2-4  
 Configuration Manager, 1-9  
 Ctrl-G, 1-11, 3-35  
 current file, 4-17, 5-4

**D**

data input/output  
   communications protocols, 1-13, 3-6, 3-8  
   format of the data file, 3-7  
 debuggers, DOS vs. Windows, 1-4, 1-5, 2-4  
 debugging

communications protocols, 1-13, 3-6, 3-8  
 hints, 2-4  
 logic and syntax, 1-4  
 setting break points, 2-4  
 undetected reader error condition, 2-5  
 user interface and performance, 1-4, 1-10,  
   1-14, 2-5  
 default INI parameter values, 3-3  
 development process, 1-9, 2-4  
 display characteristics  
   backlight, 3-27  
   character height, 1-13, 3-28  
   contrast level, 1-14, 3-20  
   scroll mode, 3-23  
   size, 1-10, 3-21  
   text wrapping, 1-10  
   video mode, 3-22  
   viewport, 1-14, 3-24  
   x and y coordinates, 3-25, 3-26  
   x and y coordinates, 1-14  
 documentation, xiv  
 DOS-application debugger, 1-4, 2-4

**E**

Editor  
   discarding changes, 4-20  
   exiting, 4-23  
   opening an INI file, 4-8  
   printing an INI file, 4-22  
   saving changes, 4-17  
   starting, 4-5  
 environment variables  
   JANUSSIM, 2-4, 5-3, 5-5  
   PATH, 3-6, 3-8, 4-11  
 error messages, 2-5, 5-3 to 5-6  
 examples  
   im\_power\_status function, 1-6  
   low battery INI file, 4-3  
 exiting the Editor, 4-23  
 exiting the TSR, 2-6

## ***JANUS Application Simulator User's Manual***

### **F**

- features, not simulated
  - character height, 1-13
  - communications protocols, 1-13, 3-6, 3-8
  - contrast level, 1-14
  - special JANUS key sequences, 1-14
  - speed and performance, 1-14
  - viewport, 1-14
- features, simulated
  - bar code input, 1-9, 1-10
  - data input/output, 1-12, 1-13, 4-11
  - power management, 1-12
  - text display, 1-10

### **H**

- hardware requirements, 1-4
- help, *See* online help
- human factors testing, 1-4, 1-10, 1-14, 2-5

### **I**

- IC.EXE, 1-9
- INI file
  - creating, 4-7
  - default values, 3-3
  - discarding changes, 4-20
  - editing, 4-8
  - editing the parameters, *See* parameters
  - JANUSSIM.INI file, 1-6, 3-3, 4-3, 4-5, 4-21
  - loading values into the TSR, 2-3, 4-21
  - printing, 4-22
  - restoring the default values, 4-20
  - saving changes, 4-17
- INI parameters, *See* parameters
- initialization file, *See* INI file
- input data
  - naming the data file, 4-11
  - naming the file, 3-6
  - specifying the path, 3-6, 4-11
- input mode, 1-10, 2-5
- installation, 1-5
- Intermec group on Windows desktop, 1-5
- Intermec Wedge, *See* Wedge
- interrupts, JANUS-specific, 1-8
- iostream functions, 2-5

IRLDESK.EXE, 1-9

### **J**

- JANUS Application Simulator
  - audience for this manual, xii
  - compatibility, 1-9
  - installation, 1-5
  - introduction, 1-3
  - requirements, 1-4
  - technical description, 1-6
  - terms and conventions for this manual, xiii
  - warranty information, xi
- JANUSSIM environment variable, 2-4, 5-3, 5-5
- JANUSSIM.INI file, 1-6, 3-3, 4-3, 4-5, 4-21
- JANUS-specific system interrupts, *See* interrupts

### **K**

- key sequence for bar code input, 1-10, 3-34, 3-35
- keyboard input
  - not received on reader, 2-5
  - simulated bar code input, 1-10
- keyboard, standard PC-AT, 1-14

### **L**

- label symbology, *See* bar code input
- loading the TSR, 2-3
- locked up PC, 2-5, 5-3
- locked viewport, 1-14

### **M**

- manually specifying path for data files, 3-6, 3-8
- Microsoft Visual C++, 2-4
- MS-DOS
  - session or shell, 2-3
  - version required, 1-4

### **N**

- not simulated features, *See* features, not simulated
- numeric keypad, 3-32

### **O**

- online help
  - accessing Editor help screens, 4-6

- Sim Editor Help icon, 1-5
- out parameter, 3-5
- output data
  - naming the data file, 4-11
  - naming the file, 3-8
  - specifying the path, 3-8, 4-11

## P

### parameters

- default values, 3-3
- for Communications
  - cancel\_rx\_buff\_return, 3-10, 3-39
  - cancel\_tx\_buff\_return, 3-10, 3-39
  - portn\_read\_file, 1-12, 3-6
  - portn\_write\_file, 1-13, 3-8
  - protocol\_ext\_status\_return, 3-9, 3-39
  - rx\_check\_status\_return, 3-9, 3-39
  - UART\_status, 3-11
- for Display
  - display\_backlight, 3-27
  - display\_char\_height, 1-13, 3-28
  - display\_contrast\_level, 1-14, 3-20
  - display\_scroll\_mode, 3-23
  - display\_size\_mode, 1-10, 3-21
  - display\_video\_mode, 3-22
  - display\_viewport\_lock, 1-14, 3-24
  - display\_x\_pos, 1-14, 3-25
  - display\_y\_pos, 1-14, 3-26
  - follow\_cursor, 1-14
- for Keyboard
  - expanded\_keyboard\_status, 3-12
  - flush\_exp\_keyboard\_status, 3-15
  - kb\_insert\_string\_capacity, 3-14
  - kb\_insert\_string\_status, 3-13
  - warmboot\_status, 3-16
- for Keypad
  - appl\_break\_status, 3-29
  - control\_key, 3-30
  - follow\_cursor, 3-29
  - keyclick, 3-31
  - numeric\_keypad\_status, 3-33
  - numeric\_keypad\_switching, 3-32
- for Power Management
  - power\_status\_backup, 3-18, 4-3
  - power\_status\_batt\_life, 3-19, 4-3

- power\_status\_battery, 3-18, 4-3
- power\_status\_line, 1-12, 3-17, 4-3
- power\_status\_return, 3-19, 4-3
- for Reader Wedge
  - clear\_abort\_callback\_return, 3-36, 3-40
  - command\_return, 3-36, 3-40
  - label\_symbology, 1-11, 3-34
  - link\_comm\_return, 3-37, 3-40
  - serial\_protocol\_control\_return, 3-37, 3-40
  - sim\_wand\_key, 1-10, 3-35, 5-3
  - standby\_wait\_return, 3-38, 3-40
  - unlink\_comm\_return, 3-38, 3-40
- grouped according to purpose, 4-9
- loading values into the TSR, 4-21
- reasons for customizing, 4-3
- restoring the default values, 4-20
- PATH environment variable, 3-6, 3-8, 4-11
- preamble for Wedge, 1-11, 5-3
- printer setup, 4-22
- printing, 4-22
- PSK library functions
  - documentation, xiv
  - im\_get\_contrast, 1-14, 3-21
  - im\_get\_control\_key, 3-30
  - im\_get\_display\_mode, 1-13, 3-21, 3-22, 3-23, 3-28
  - im\_get\_follow\_cursor, 3-29
  - im\_get\_keyclick, 3-31
  - im\_get\_lebel\_symbology, 3-34
  - im\_get\_warm\_boot, 3-16
  - im\_irl\_a, 3-34
  - im\_irl\_n, 3-34
  - im\_irl\_v, 3-6
  - im\_irl\_y, 3-6, 3-34
  - im\_kb\_insert\_keycodes, 3-14
  - im\_kb\_insert\_string, 3-13
  - im\_kb\_insert\_string\_keycodes, 3-13
  - im\_link\_comm, 3-37
  - im\_power\_status, 1-7, 1-12, 3-17, 3-18, 3-19, 4-3
  - im\_receive\_buffer, 1-12, 3-6, 3-7
  - im\_receive\_buffer\_no\_wait, 3-6
  - im\_receive\_buffer\_noprot, 3-6
  - im\_receive\_byte, 1-12, 3-6, 3-7
  - im\_receive\_input, 3-6, 3-34

## ***JANUS Application Simulator User's Manual***

- im\_set\_input\_mode, 1-10, 2-5
- im\_transmit\_buffer, 3-8
- im\_transmit\_buffer\_no\_wait, 3-8
- im\_transmit\_buffer\_noprot, 3-8
- im\_transmit\_byte, 3-8
- im\_viewport\_getxy, 3-26
- naming conventions, 3-5
- related interrupts, 1-8

### **R**

- RAM for TSR, 1-9, 5-3
- README file, 1-5
- reboot, 3-16
- related documentation, xiv
- restoring defaults, 4-20
- return values
  - communication standard, 3-39
  - Reader Wedge standard, 3-40
- running JANUS applications on a PC, 1-3, 2-5, 5-3

### **S**

- Sim Editor Help icon, 1-5
- Sim Editor icon, 1-5
- Sim Readme icon, 1-5
- simulated features, *See* features, simulated
- Simulator TSR, *See* TSR
- software requirements, 1-4
- starting the Editor, 4-5
- starting the TSR, 2-3
- stdlib functions, 2-5
- symbolologies available, 3-34

### **T**

- terminate and stay ready program, *See* TSR
- terms and conventions for this manual, xiii
- testing
  - communications protocols, 1-13, 3-6, 3-8

- logic and syntax, 1-4
- setting break points, 2-4
- undetected reader error condition, 2-5
- user interface and performance, 1-4, 1-10, 1-14, 2-5
- troubleshooting, 2-5, 5-3 to 5-6
- TSR
  - definition, 2-3
  - exiting, 2-6
  - memory requirements, 5-3
  - not loaded into memory, 5-6
  - starting, 2-3
  - unexpected termination, 2-3

### **U**

- undetected reader error, 2-5
- unexpected termination of TSR, 2-3
- unloading the TSR, 2-6
- unsupported interrupts, 1-9
- unsupported software
  - Configuration Manager, 1-9
  - IRL Desktop, 1-9

### **V**

- viewport
  - cursor position, 3-29
  - lock, 3-24
  - related INI parameters, 1-14

### **W**

- wand input, *See* bar code input
- warm boot, 3-16
- warranty information, xi
- Wedge
  - setting the preamble, 1-11, 5-3
  - simulating bar code input, 1-9, 1-11
  - troubleshooting, 5-3
- Windows-application debugger, 1-4