

User's Manual

P/N 069671-001

TRAKKER[®] T2090 Application Simulator

 **intermec**

A **UNOVA** Company

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

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

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

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

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.

© 1999 by Intermec Technologies Corporation
All Rights Reserved

The word Intermec, the Intermec logo, JANUS, IRL, TRAKKER, Antares, Adara, Duratherm, EZBuilder, Precision Print, PrintSet, Virtual Wedge, and CrossBar are either trademarks or registered trademarks of Intermec.

Throughout this manual, trademarked names may be used. Rather than put a trademark (™ or ®) 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.

Contents

Before You Begin vii
 Warranty Information vii
 Cautions and Notes vii
 About this Manual vii
 Other Intermec Manuals ix

1

Getting Started

Introduction to the Application Simulator 1-3

Installing the Application Simulator 1-3
 Important Directories 1-3
 README.TXT File 1-4
 IMT209SIM Environment Variable 1-4
 Microsoft C/C++ Version Requirements 1-4
 Tips for Installing Microsoft Visual C/C++ 1-5

Using the Simulator With Other Products 1-5

Using the Simulator in the Development Process 1-5

How to Begin Simulating a T2090 Application 1-6

2

Working With the Application Simulator

How the Simulator Works 2-3
 Three Parts of the Simulator 2-3
 Example of Simulating Incoming Serial Communications 2-3

Starting the Serial Input Editor 2-4

Features That Are Simulated 2-4
 Text Display 2-4
 Keypad Input 2-5
 Bar Code Input 2-5
 Data Input Through COM1 2-6
 Data Output Through COM1 2-7
 Reader Command Syntax Checking 2-7

Features That Are Not Simulated 2-8

Overview of Modifications to PSK Functions 2-9
 Display Functions 2-9

im_cputs 2-9
im_puts 2-9
im_draw_underline 2-9
im_set_cursor_style 2-10
im_putchar 2-10
im_set_display_mode 2-10
Communications Functions 2-10
im_receive_buffer 2-10
im_transmit_buffer 2-11
im_receive_input 2-11
im_receive_field 2-11
Keyboard Mode Functions 2-11
im_set_kbmode 2-11
im_get_kbmode 2-11
Label Functions 2-12
im_get_label_symbology 2-12
im_get_label_symbologyid 2-12
Reader Command Function 2-12
im_command 2-12

3

INI File Parameter Descriptions

Why and How You Customize INI Files 3-3

Where to Store Customized INI Files 3-3

Parameter Descriptions 3-4

sample 3-4
label_preamble 3-4
label_preamble_string 3-4
label_postamble 3-5
label_postamble_string 3-5
label_symbology 3-6
label_symbologyid 3-6
serial_port_emulation 3-7
sim_directory 3-7

4

Customizing INI Files With the Editor

Introduction to the Simulator Editor 4-3

Starting the Simulator Editor 4-3

<i>No INI File Found</i>	4-4
<i>Introduction to the User Interface</i>	4-4
<i>Menu Bar</i>	4-4
<i>Tabs</i>	4-4
<i>Online Help</i>	4-5
<i>Creating a New INI File</i>	4-5
<i>Opening an Existing INI File</i>	4-6
<i>Setting INI Parameters</i>	4-6
<i>Saving Changes</i>	4-6
<i>Discarding Changes</i>	4-8
<i>Restoring the Default Values</i>	4-8
<i>Printing INI Files</i>	4-9
<i>Exiting the Simulator Editor</i>	4-9

5

Troubleshooting

<i>Problems Operating the Application Simulator</i>	5-3
<i>Linking to the Wrong IMT209ML.LIB Library</i>	5-3
<i>T2090 Application Fails to Launch</i>	5-3
<i>T2090 Application Locks Up</i>	5-3
<i>Problems Simulating Bar Code Input With a Wedge</i>	5-4
<i>Error and Status Messages</i>	5-4

A

Adding the Simulator to the Microsoft Visual C/C++ Tools Menu

<i>Adding the Simulator to the Tools Menu</i>	A-3
---	-----

I

Index

Before You Begin

This section introduces you to standard warranty provisions, cautions and notes, 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 1-800-755-5505, and in Canada call 1-800-688-7043. Otherwise, refer to the Worldwide Sales & Service list shipped with this manual for the address and telephone number of your Intermec sales organization.

Cautions and Notes

The cautions and notes 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 alerte d'une procédure de fonctionnement, d'une méthode, d'un état ou d'un rapport qui doit être strictement respecté pour empêcher l'endommagement ou la destruction de l'équipement, ou l'altération ou la perte de données.



***Note:** Notes are statements that either provide extra information about a topic or contain special instructions for handling a particular condition or set of circumstances.*

About This Manual

This manual describes how to use the TRAKKER T2090 Application Simulator, which helps you create, test, and debug applications for TRAKKER T2090 computers.

Use this manual in conjunction with the *TRAKKER T2090 Hand-Held Batch Computer User's Manual*, which describes the PSK library functions that the 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 C, and know how to use Microsoft Visual C/C++ and Microsoft CodeView for DOS debugger. They have already read the *TRAKKER T2090 Hand-Held Batch Computer User's Manual*, so they understand how to create programs for TRAKKER T2090 computers.

How This Manual Is Organized

This manual is organized as follows:

Chapter	What You'll Find
1	<i>Getting Started</i> Introduces the Simulator, explains how to install the software, describes other software products you can use with the Simulator, and explains how to begin simulating applications.
2	<i>Working With the Application Simulator</i> Describes how the Simulator works, explains which T2090 features are simulated and which are not, and describes the PSK functions that have been modified.
3	<i>INI File Parameter Descriptions</i> Lists and describes the parameters in the initialization (INI) file.
4	<i>Customizing INI Files With the Editor</i> Explains how to use the Simulator Editor to customize the INI file.
5	<i>Troubleshooting</i> Lists and describes the status and error messages you may see.
A	<i>Adding the Simulator to the Microsoft Visual C/C++ Tools Menu</i> Describes how to manually add the Simulate T2090 command to the Microsoft Visual C/C++ Tools menu.

Terminology

You should be aware of how these terms are used in this manual:

Term	Meaning
<i>Target_Directory</i>	This term refers to the drive and path you selected during the software installation.
working directory	This term refers to the directory from which you are running the T2090 application.

Conventions for Input From a Keyboard or Keypad

You should be aware of these formatting conventions for representing input from a keyboard or keypad.

Convention	Meaning
Bold	Keys that you press on a PC keyboard are shown in bold . For example, “press Enter ” means you press the key labeled “Enter” on the keyboard. The first letter of a key name is always capitalized.
Ctrl-Alt-Del	When a series of keys are shown with a dash between them, you must press and hold the keys in the order shown and then release them all. For example, to boot a PC, you press and hold Ctrl , press and hold Alt , press and hold Del , and then release the keys.

Conventions for Commands

You should be aware of these formatting conventions for entering commands.

Convention	Meaning
<i>Courier</i>	Commands are shown in Courier exactly as you must type them. For example: <code>dir</code>
<i>Italics</i>	Variables are shown in italics. You must enter a real value for the variable.

Other Intermec Manuals

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

Manual Title	Part No.
<i>TRAKKER T2090 Hand-Held Batch Computer User's Manual</i>	067212
<i>Model 200 Controller System Manual</i>	063439
<i>Model 200 Controller Technical Reference Manual</i>	064398

Read the online README file provided with the software. This file contains important information that was not available when this manual was printed.

You may need additional information when working with the T2090 computer in a data collection system. Please visit our Web site at www.intermec.com for a list of available manuals or to access many of our current manuals in PDF format. To order printed versions of the Intermec manuals, contact your local Intermec representative or distributor.

1

Getting Started

This chapter introduces the TRAKKER T2090 Application Simulator, describes its installation, explains how to use the Simulator with other products as part of your development process, and describes three ways to begin simulating applications.

Introduction to the Application Simulator

The TRAKKER® T2090 Application Simulator helps you create, debug, test, and run on your PC most applications for TRAKKER T2090 computers. The Application Simulator also lets you use Microsoft Visual C/C++ and Microsoft Codeview for DOS to develop TRAKKER T2090 applications.

You can also access the Application Simulator from Intermec's EZBuilder application development tool or from the Microsoft Visual C/C++ toolbar.

T2090 applications created with version 2.1 of the TRAKKER T2090 Programmer's Software Kit (PSK) can be run on PCs. The functions that issue T2090-specific interrupts, which can disrupt or crash a PC, have been modified. When the T2090 application detects that it is running on a PC, the PSK functions operate in a way that will not disrupt the PC. For details, see "Overview of Modifications to PSK Functions" in Chapter 2.

The Application Simulator also lets you customize the simulation by setting parameters in an initialization file. For details on the T2090 features that are simulated, see "Features That Are Simulated" in Chapter 2.

Installing the Application Simulator

The TRAKKER T2090 Application Simulator is automatically installed when you install the TRAKKER T2090 PSK v2.1 or EZBuilder v2.1.

The default location for the software is C:\INTERMEC, but you can select another drive and directory during installation. In this manual, the term *Target_Directory* is used to indicate the drive and path you selected. Any directory information appearing after *Target_Directory* is required.

For example, *Target_Directory* \IMT209\SIM can refer to C:\INTERMEC\IMT209\SIM or F:\DEVELOPMENT\IMT209\SIM, depending on the location you selected during installation.

Important Directories

The installation creates and fills these directories on your PC:

- *Target_Directory*\IMT209\DOC
- *Target_Directory*\IMT209\DRIVER
- *Target_Directory*\IMT209\EXAMPLES
- *Target_Directory*\IMT209\EZBUILD

- *Target_Directory*\IMT209\INCLUDE
- *Target_Directory*\IMT209\LIB
- *Target_Directory*\IMT209\REMDISK
- *Target_Directory*\IMT209\SAMPLE
- *Target_Directory*\IMT209\SIM

README.TXT File

Read the README file. This file may contain information that was not available when this manual was printed.

IMT209SIM Environment Variable

The installation adds the IMT209SIM environment variable to your AUTOEXEC.BAT file. This environment variable points to the location of the T2090 Application Simulator software, which is stored by default in the *Target_Directory*\IMT209\SIM directory.

Microsoft C/C++ Version Requirements

Applications built for the T2090 require Microsoft Visual C/C++ Professional Edition v1.0 or v1.5x, which can create 16-bit DOS applications. If you are using a different version, you must also install either v1.0 or v1.5x.

The Microsoft Visual C/C++ Professional Edition v4.x package includes a disk for v1.52. However, the Microsoft Visual C/C++ Professional Edition v5.0 does not contain the 16-bit version of C.

You can order Microsoft Visual C/C++ Enterprise Edition v1.52 from Intermec (P/N 590224).

Microsoft Version	16-Bit Support	
v1.0	✓	
v1.5x	✓	
v4.x	✓	Includes a disk for v1.52
v5.x	✗	Order Microsoft Visual C/C++ Enterprise Edition v1.52 from Intermec (P/N 590224)

Tips for Installing Microsoft Visual C/C++

If you installed Microsoft Visual C/C++ **before** you installed the Application Simulator on your PC, two commands were automatically added to the Tools menu of Microsoft Visual C/C++:

Simulate T2090 Choose this command to simulate the T2090 application.

T2090 Serial Input Editor Choose this command to start the Serial Input Editor.

If you installed Microsoft Visual C/C++ **after** you installed the Application Simulator, you can manually add the commands to the Tools menu. For instructions, see Appendix A, “Adding the Simulator to the Microsoft Visual C/C++ Tools Menu.”

Using the Simulator With Other Products

The Application Simulator was designed to be used with the Microsoft Visual C/C++ Professional Edition (version 1.0 or 1.5x) application development software. Or you can use Intermec’s EZBuilder v2.1 application development tool to create T2090 applications and then run the Application Simulator to test the applications.

You can use Microsoft Codeview for DOS to debug T2090 applications.

You can install an Intermec Wedge on your PC to simulate bar code input. For help configuring a wedge, see “Bar Code Input” in Chapter 2.

Using the Simulator in the Development Process

The Application Simulator can be an integral part of your application development process. For example, you can follow these steps:

1. Start Windows.
2. Start Microsoft Visual C/C++.
3. Open your T2090 project or create a new project.
4. Build the application.
5. Select Simulate T2090 from the Tools menu to simulate the application. Or select Codeview for MS-DOS to run and debug the application.
6. Exit the debugger or the simulated application.

How to Begin Simulating a T2090 Application

You can simulate a T2090 application using one of these methods:

- Choose TRAKKER Simulator from the Tools menu of EZBuilder.
- Choose Microsoft Codeview for DOS from the Tools menu of EZBuilder.
- Choose Simulate T2090 from the Tools menu of Microsoft Visual C/C++.
- Choose Codeview for MS-DOS from the Tools menu of Microsoft Visual C/C++.

Working With the Application Simulator

This chapter describes how the Simulator works and how various T2090 features and PSK functions are simulated.

How the Simulator Works

When a T2090 application begins running, it checks the operating system to determine if it is running on a T2090 or being simulated on a PC. If the application is being simulated on a PC, it performs two additional tasks:

- The application attempts to read the INI file, which is defined in the next section. If the application cannot find the file in the working directory, it checks the *Target_Directory*\TOOLS directory. If it cannot find the file there, it uses the default INI settings.
- Some PSK functions in the application change the way they execute so that they will not cause problems on the PC. For details, see “Overview of Modifications to PSK Functions” later in this chapter.

Three Parts of the Simulator

The Application Simulator consists of three parts:

Serial Input Editor The Serial Input Editor is a Windows-based tool for sending data into the T2090 application that is being simulated on your PC. For help using the Serial Input Editor, see “Starting the Serial Input Editor” and “Data Input Through COM1” later in this chapter.

INI file The initialization (INI) file specifies how the T2090 application simulates T2090 features such as bar code label input. IMT209SM.INI is the default INI file. To learn about the parameters in the INI file, see Chapter 3.

Simulator Editor The Simulator Editor is a Windows-based tool for setting the parameters stored in the IMT209SM.INI file. For help using the Simulator Editor, see Chapter 4.

Example of Simulating Incoming Serial Communications

This example tests how the TRAKKER T2090 application handles incoming serial communications.

1. Create a TRAKKER T2090 application that receives data through COM1.
2. Enable the `serial_port_emulation` parameter in the IMT209SM.INI file.
3. Create a text file called ORDERS.TXT and fill it with several lines of data to be input to the application. Each line can be up to 1024 characters long and must terminate with a linefeed (LF) character.

4. Run the Serial Input Editor. Specify the complete path of the ORDERS.TXT file and click the Send File button. The contents of the file are copied into a buffer, from which the T2090 application can read the data one line at a time.
5. Begin simulating the TRAKKER T2090 application. When the application executes the im_receive_buffer PSK function, the function reads a line of data from the buffer.
6. Observe how the T2090 application responds to the incoming data.

Starting the Serial Input Editor

You can start the Serial Input Editor using one of these methods:

- Choose T2090 Serial Input Editor from the Tools menu of EZBuilder.
- Choose T2090 Serial Input Editor from the Tools menu of Microsoft Visual C/C++.

For help using the Serial Input Editor, click the Help button.

Features That Are Simulated

This section describes the features of the TRAKKER T2090 computer's performance that can be simulated:

- Text Display
- Keypad Input
- Bar Code Input
- Data Input Through COM1
- Data Output Through COM1
- Reader Command Syntax Checking

Text Display

A simulated application creates a display on your PC that mimics the screen on the T2090. The simulated application limits character echoing to the lines and columns based on the display size you specified by calling the im_set_display_mode function.

Some display characteristics are represented on the PC in color, as shown in the next table.

Display Characteristics	Background Color	Foreground Color
Normal	Black	Light gray
Reverse video	Light gray	Black
Underline	Black	Magenta
Underline and reverse video	Magenta	Black



Note: A simulated application does not limit the output of standard C display functions (such as `printf`) within the T2090 display size (8x20).

Keypad Input

A simulated application simulates the T2090 keypad input automatically as you type on the PC keyboard.

There are only two special keys on the T2090 that are not available on a standard PC keyboard: the Backlight key and the Num/Alpha key. You do not need these keys while simulating a T2090 application.

Bar Code Input

You can customize how bar code input is simulated by setting these parameters in the IMT209SM.INI file:

- label_postamble
- label_postamble_string
- label_preamble
- label_preamble_string
- label_symbology
- label_symbologyid

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

- By pressing **Ctrl-G** and typing the data on your PC keyboard.
- By configuring an Intermec Wedge and scanning actual bar code labels.

To simulate bar code input with a keyboard

1. Press **Ctrl-G**.
2. Type the data you want entered into the application as if it were bar code input from a scanner or wand.
3. Press **Enter** to terminate the simulated bar code input.

To simulate bar code input with an Intermec wedge

1. Attach an Intermec Wedge to your PC.
2. Set the first preamble characters to **Ctrl-G**.
See your wedge or scanner documentation for help setting the preamble. Scan the SO character for **Ctrl** and the lowercase g character for **G**.
3. Use the wedge to scan bar code labels. The bar code data is entered into the TRAKKER T2090 application.
4. Scan Enter or have a Return in the postamble to terminate the bar code input.

Data Input Through COM1

A simulated application can simulate Frame mode serial communications input through the T2090's COM1 port.

To simulate data input to the application through COM1 with a data file

1. Enable the serial_port_emulation parameter in the INI file.
2. Create a text file that contains data for the application. A line of data is either:
 - A data string that ends in a <CR><LF>
 - A data string that is 1024 bytes long (without a <CR><LF>)

Any PSK function that reads data from COM1 (im_receive_buffer, im_receive_field, or im_receive_input) can read a line of data from the buffer. With each subsequent call, the functions continue reading data where the last function left off in the buffer until the last line has been read. Then the simulated application deletes the contents of the buffer.

3. Start the Serial Input Editor.
4. Make sure the SIM Directory field specifies the correct location of the T2090 Application Simulator software. The default directory is *Target_Directory\IMT209\SIM*.
5. At the Serial Input Editor screen, specify the path and filename of the file in the Outgoing File field and click the Send File button.

6. Run the T2090 application on the PC. The application checks the operating system of the computer and detects that it is not running on a T2090.
7. Observe how the T2090 application responds to the incoming data.

To simulate data input to the application through COM1 manually

1. Enable the serial_port_emulation parameter in the INI file.
2. Start the Serial Input Editor.
3. Run the T2090 application on the PC. The application checks the operating system of the computer and detects that it is not running on a T2090.
4. The application executes a function to read data from COM1.
5. At the Serial Input Editor screen, enter the data into the Outgoing Data field and click the Send Data button.
6. Observe how the T2090 application responds to the incoming data.

Data Output Through COM1

A simulated application can simulate Frame mode serial communications output through the T2090's COM1 port.

To simulate data output from the application through COM1

1. Run the T2090 application. The application checks the operating system of the computer and detects that it is not running on a T2090.
2. The application executes a function (such as im_transmit_buffer) to send data to COM1. Because the application is not running on a T2090, the data is not sent to COM1. Instead, the application creates the COMP1.TRX file in the directory and copies the data into that file.
3. Observe how the T2090 application operates.
4. Test the application on a TRAKKER T2090 computer to make sure that the application is handling the input and output communications protocols correctly.

Reader Command Syntax Checking

If the simulated application includes the im_command PSK function, which executes one or more reader commands, the application simply checks the syntax of the reader command, returning IM_SUCCESS if the syntax is correct or IM_PARSE_ERROR if the syntax is flawed.

Features That Are Not Simulated

This table describes the T2090 features that are not simulated:

Feature	Comments
Reader Command Execution	A simulated application does not simulate the execution of reader commands. As described earlier in this chapter, the application simply checks the syntax of the im_command PSK function.
File Transfer	A simulated application does not simulate file transfers to or from the T2090 application.
Character Mode	A simulated application does not simulate the Character mode serial communications. The application only simulates Frame mode serial communications.
Label Time Stamps	A simulated application does not append a time stamp to simulated bar code label data.
Double-Byte Characters	A simulated application does not simulate double-byte characters
Standard C Display Functions	A simulated application does not limit the output of standard C display functions (such as printf) within the T2090 display size (8x20).
Speed and Performance	A simulated application does not simulate the speed or performance of a T2090. The application will run as fast as the PC can execute it.
Contrast Level	A simulated application does not simulate the contrast set for the T2090.

Overview of Modifications to PSK Functions

Some TRAKKER T2090 PSK functions have been modified so that they operate differently if they detect that they are not being executed on a T2090 computer. This section describes those changes.

For more details about PSK functions, see Chapter 4, “Application Functions,” in the *TRAKKER T2090 Hand-Held Batch Computer User’s Manual*.

Display Functions

These PSK functions control the T2090 display.

im_cputs

The function displays a string on the screen with its attribute selected by the user. A string can be displayed as normal, reverse video, underlined, or underlined and reverse video.

When executing on a PC, this function changes the PC display to indicate the attributes, as shown in this table:

Display Characteristics	Background Color	Foreground Color
Normal	Black	Light gray
Reverse video	Light gray	Black
Underline	Black	Magenta
Underline and reverse video	Magenta	Black

im_puts

This function performs the same operation as *im_cputs*, but also appends a CR/LF to the end of the string.

im_draw_underline

This function assists *im_cputs*, *im_puts*, *im_putchar*, and other display functions to underline a string.

When executing on a PC, this function changes the PC display to indicate the underline attributes, as shown in this table:

Display Characteristics	Background Color	Foreground Color
Underline	Black	Magenta
Underline and reverse video	Magenta	Black

im_set_cursor_style

When executing on a T2090 or PC, this function sets the style of the cursor to a single underline or hidden.

im_putchar

This function places a character at the current cursor position and changes the line attribute to normal, reverse video, underline, underline and reverse video, or unchanged.

When executing on a PC, this function changes the PC display to indicate the attributes, as shown in this table:

Display Characteristics	Background Color	Foreground Color
Normal	Black	Light gray
Reverse video	Light gray	Black
Underline	Black	Magenta
Underline and reverse video	Magenta	Black

im_set_display_mode

When executing on a T2090, this function sets the font size for characters to be displayed as normal (5x7 pixels), large (5x14 pixels), and special (10x14 pixels).

When executing on a PC, this function does not change the font size. Instead, the function causes the display size to be limited as follows:

Font	Lines x Columns
Normal	8 x 20
Large	4 x 20
Special	4 x 10

Communications Functions

These PSK functions control communications with the T2090.

im_receive_buffer

This function receives the contents of a data buffer from COM1.

When executing on a PC, this function reads data from the Serial Input Editor and handles it as serial data in Frame mode incoming from COM1. The serial_port_emulation parameter in the IMT209SM.INI file must be enabled.

im_transmit_buffer

This function transmits the contents of a data buffer through COM1.

When executing on a PC, this function creates the COMP1.TRX file in the current directory and copies the contents of the data buffer into the file. The serial_port_emulation in the IMT209SM.INI file must be enabled.

im_receive_input

This function receives input from the specified source(s) and places it into the received buffer. The source can be bar code labels, the keypad, or COM1.

When executing on a PC, this function receives the input as follows:

- Bar code label input can be simulated as input from keyboard or from a wedge. For details, see “Bar Code Input” earlier in this chapter.
- Keypad input is simulated when you type on the PC keyboard.
- COM1 input is simulated as described for im_receive_buffer.

im_receive_field

This function manages an input field area on the screen. You can specify display attributes for the field and control the length of the input data.

When executing on a PC, this function receives the bar code label, keypad, and COM1 input as described for im_receive_input.

Keyboard Mode Functions

These PSK functions control the keyboard mode for the T2090.

im_set_kbmode

This function sets the operational mode (alpha or numeric) for the T2090 keypad.

When executing on a PC, this function does not set the mode because there is no alpha or numeric mode for a PC keyboard. This function only saves the value in the KeyMode variable.

im_get_kbmode

This function checks the mode of the T2090 keypad and returns a constant that indicates if the keypad is in alpha or numeric mode.

When executing on a PC, this function returns the value stored in the KeyMode variable, which was set by im_set_kbmode.

Label Functions

These PSK functions control how bar code labels are handled by the T2090.

im_get_label_symbology

This function gets and returns the identification of the symbology from the most recent label read. This function normally is used after calling `im_receive_input` or `im_receive_field`.

When executing on a PC, this function returns the value of the `label_symbology` parameter in the `IMT209SM.INI` file.

im_get_label_symbologyid

This function gets and returns the identification of an AIM symbology ID string, such as `JA0`, from most recent label read. This function normally is used after calling `im_receive_input` or `im_receive_field`.

When executing on a PC, this function returns the value of the `label_symbologyid` parameter in the `IMT209SM.INI` file.

Reader Command Function

This PSK function controls how reader commands are executed on the T2090.

im_command

This function sends reader commands to the T2090.

When executing on a PC, this function simply verifies that the function is syntactically correct, but it does not execute the reader commands themselves. The function returns `IM_SUCCESS` if the syntax is correct or `IM_PARSE_ERROR` if the syntax is flawed.

INI File Parameter Descriptions

This chapter explains why you should customize your INI file, describes the parameters in the INI file, and explains which PSK library functions will receive the INI parameters as return values and out parameters.

Why and How You Customize INI Files

The INI parameters control how the T2090 application simulates T2090 features (such as bar code input) when executing on a PC. You can customize the parameters to test your TRAKKER T2090 applications against specific conditions.



Note: *Customizing the INI file is optional. You do not have to customize the INI file if you are satisfied with the default values in IMT209SM.INI.*

You can customize INI parameters using the Simulator Editor or any ASCII text editor:

- The Simulator Editor provides a graphic environment for changing your INI files. Instead of typing the settings, you select them from menus and dialog boxes. For help using the Simulator Editor, see Chapter 4.
- If you create and edit INI files with an ASCII text editor, use a copy of the default INI file to make sure you conform to the formatting conventions.

Where to Store Customized INI Files

The INI file for the TRAKKER T2090 Application Simulator must be named IMT209SM.INI. You cannot change the name to indicate that the file has been customized. But you can store different copies of the file in different directories.

If you understand where a simulated application looks for the INI file, you can use the information to manage different versions of the INI file. The application first checks the working directory (which is the same directory where the simulated application is stored). If it does not find the file there, it checks *Target_Directory*\Tools. If it does not find the file there either, it uses the default INI settings.

Here is one way to use location to manage different versions of the INI file:

1. Keep a master copy of IMT209SM.INI in the Tools directory.
2. Store each application in its own directory. If the application requires a customized INI file, copy the master file to the directory and edit it.
3. When you run an application on a PC, it may find a customized IMT209SM.INI file in the working directory. If it does not, it will use the master copy in the Tools directory.

Parameter Descriptions

This section contains descriptions of the parameters in the INI file, in alphabetical order. The first description, *sample*, illustrates the type of information presented for each parameter and is not a valid parameter.

sample

Purpose	The purpose of the parameter.
Default	The default value for the parameter.
Values	The values you can set for the parameter.
Function	The 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. Some parameters do not have a PSK function associated with them.
Notes	Optional information about the parameter.

label_preamble

Purpose	Lets you add a user-defined string to the front of the label data. The user-defined string is specified in the <i>label_preamble_string</i> parameter.
Default	0 - disabled
Values	0 - disabled 1 - enabled
Function	<i>im_receive_field</i> <i>im_receive_input</i>

label_preamble_string

Purpose	Specifies the user-defined string to be added to the front of the label data if the <i>label_preamble</i> parameter is enabled.
Default	blank
Values	Any ASCII text
Function	<i>im_receive_field</i> <i>im_receive_input</i>

label_postamble

Purpose	Lets you append a user-defined string to the end of the label data. The user-defined string is specified in the <code>label_postamble_string</code> parameter.
Default	0 - disabled
Values	0 - disabled 1 - enabled
Function	<code>im_receive_field</code> <code>im_receive_input</code>

label_postamble_string

Purpose	Specifies the user-defined string that can be appended to the end of the label data if the <code>label_postamble</code> parameter is enabled.
Default	blank
Values	Any ASCII text
Function	<code>im_receive_field</code> <code>im_receive_input</code>

label_symbology

Purpose Specifies the symbology of the last simulated scanned label.

Default 1 - Code 39

Values 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

Function im_get_label_symbology
im_receive_input

Notes The value of this INI parameter is returned to the application in the symbology parameter of the function calls that use this INI parameter.

When testing an application, you can simulate the act of scanning a label by pressing **Ctrl-G**.

label_symbologyid

Purpose Specifies the AIM symbology ID of the simulated bar code input.

Default |A1

Values Enter up to 4 ASCII characters.

Function im_get_label_symbologyid

Notes The value of this INI parameter is returned to the application in the symbologyId parameter of the function calls that use this INI parameter.

serial_port_emulation

Purpose	Specifies whether the serial port operations are simulated with the Serial Input Editor and the COMP1.TRX file.
Default	1 - enabled
Values	0 - disabled 1 - enabled
Function	im_receive_buffer, im_receive_field, im_receive_input, im_transmit_buffer

sim_directory

Purpose	Specifies the directory (<i>Target_Directory</i> \IMT209\SIM) where the T2090 Application Simulator software is located.
Default	C:\INTERMEC\IMT209\SIM
Values	Enter the complete path name for the IMT209\SIM directory in this format: <i>drive:\path\IMT209\SIM</i>
Function	none
Notes	This INI parameter and the IMT209SIM environment variable both point to the location of the Application Simulator software. When this parameter is set, it overrides the value of the environment variable. This parameter was created because some Microsoft NT configurations could not recognize the IMT209SIM environment variable. To disable this parameter, delete it from the INI file.

4

Customizing INI Files With the Editor

This chapter introduces the Simulator Editor and describes how to use it to customize the initialization (INI) file for the TRAKKER T2090 Application Simulator.

Introduction to the Simulator Editor

The Simulator Editor is a Windows-based tool for viewing and setting the parameters stored in the IMT209SM.INI file. The INI file contains parameters that specify how certain features are simulated. For descriptions of the parameters, see Chapter 3, “INI File Parameter Descriptions.”



Note: *Because IMT209SM.INI is a text file, you can use any text editor to change the values of the parameters. You do not need to use the Simulator Editor.*

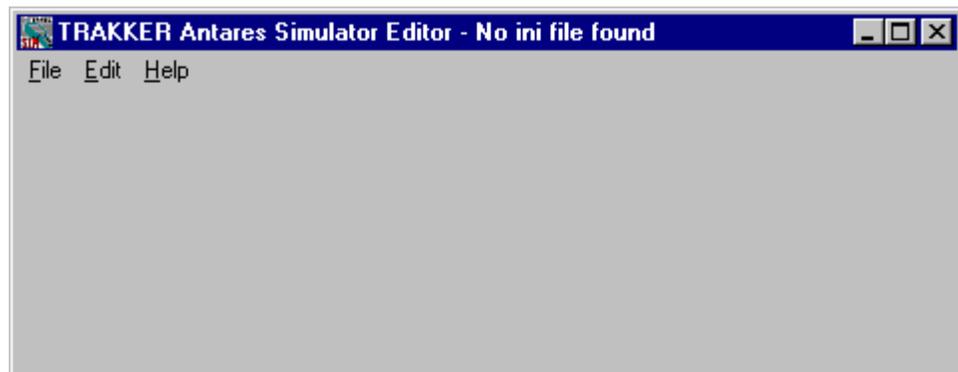
The Editor can be used with Intermec’s two application simulators:

- TRAKKER Antares Application Simulator, which simulates applications for the TRAKKER Antares family of computers, including the T2425 hand-held terminal and the T248X stationary terminal.
- TRAKKER T2090 Application Simulator, which simulates applications for the T2090 computer.

Starting the Simulator Editor

Start the Simulator Editor from the TRAKKER T2090 Development Tools group on the Windows desktop or on the Start menu.

The Editor window appears:



No INI File Found

When you start the Editor, it attempts to open the IMT24SIM.INI file in the working directory. The IMT24SIM.INI file is the default INI file for the TRAKKER Antares Application Simulator.

Because you are working with the TRAKKER T2090 Application Simulator and the IMT209SM.INI file, the Editor probably will not find the IMT24SIM.INI file. The Editor displays the message, No ini file found, in the title bar of the window.

You can open the IMT209SM.INI file by choosing Open from the File menu. Or you can create a new IMT209SM.INI file by choosing New from the File menu and selecting T209x when prompted to specify the type of INI file.

Introduction to the User Interface

This utility conforms to standard Microsoft Windows conventions, so it should be easy to familiarize yourself with the user interface.

Menu Bar

The main menu bar contains the File, Edit, and Help commands:

File Lets you open, create, and save INI files. Also lets you restore the parameters to their defaults, print INI files, and exit the Editor.

Edit Lets you select a category of parameters (such as Communications and Labels), causing the corresponding tab section to become active in the Editor window.

Help For details, see the "Online Help" section later in this chapter.

Tabs

The parameters are divided into categories, and each category has its own tab in the Editor window. To view the parameters in each category, click the corresponding tab or choose the category name from the Edit menu.

You set the value for each parameter by:

- selecting radio buttons.
- checking boxes.
- typing into the fields.
- choosing items from pull-down lists.

You can set reset a parameter to its default value by clicking the Default button next to the parameter. You can reset all parameters to their defaults by choosing Restore defaults from the File menu.

Online Help

You can access the online help from the Help menu on the Editor's menu bar. The online help contains all the information in this chapter and describes how to use the Editor with either application simulator.

Creating a New INI File

You can create and customize new INI files with the Editor. In each new file, all the parameters are set to their default values.

To create a new INI file

1. From the File menu, choose New.
2. The Select File Type dialog box appears. Select the T209x file type and click the OK button. The Editor creates an untitled file with its parameters set to their defaults.
3. Customize the parameters if necessary.
4. From the File menu, choose Save.
5. Specify IMT209SM.INI as the file name. If necessary, select the path for the new file. Click the Save button.

Note: You must use the *IMT209SM.INI* file name.

6. Click the OK button when the Editor prompts you to save all the values in the file.

Opening an Existing INI File

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

To open an existing INI file

1. From the bottom of the File menu, choose the INI file from the list of five most recently opened files.

Or:

1. From the File menu, choose Open. The Open dialog box appears.
2. Select the file name. If necessary, you can browse for the file.
3. Click the Open button. The file opens, you return to the main menu, and the filename is displayed in the title bar. You can begin customizing the file.

Setting INI Parameters

The INI parameters for IMT209SM.INI are organized into two categories: Communications and Labels. For descriptions of the parameters, see Chapter 3. For help customizing the parameters, see the next procedure.

To set a parameter

1. Click the Communications or Label tab to choose the type of parameter you want to edit. The tab you select moves forward in the Editor window.
2. To set the value for each parameter, select the button, type in the field, or choose an item from the pull-down list.

For help deciding which value to set for each parameter, see Chapter 3.

Saving Changes

You can save the changes to an INI file:

- into the current INI file and continue working.
- into the current INI file when you exit the Editor.
- into a new INI file.
- into an existing INI file.

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

To save the changes into the current INI file and continue working

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

```
Saved all values in file.
```

2. Click the OK button.

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 this message:

```
The filename file has been changed.  
Do you want to save the changes?
```

2. Click the Yes button. The changes are saved into the current INI file, the Editor shuts down, and you return to the Windows desktop.

To save the changes into a new INI file

1. From the File menu, choose Save As. The Save As dialog box appears.
2. In the File name field, type IMT209SM.INI. If necessary, select a directory from the Directories list box.
3. Click the Save button. The Editor creates the new file, saves the changes, and displays this message:

```
Saved all values in file.
```

4. Click the OK button.

To save the changes into an existing INI file

1. From the File menu, choose Save As. The Save As dialog box appears.
2. Select the name of the file where you want to save the changes. You may need to browse for the correct directory.
3. Click the Save button. The Editor displays this message:

```
This file already exists. Replace existing file?
```

4. Click the Yes button. The Editor saves the changes into the file and displays this message:

```
Saved all values in file.
```

4. Click the OK button.

Discarding Changes

You can discard changes when you exit the Editor.

To discard the changes

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

```
The filename file has been changed.  
Do you want to save the changes?
```

2. Click the No button. The file is closed and the changes are not saved.

Or:

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

```
The filename file has been changed.  
Do you want to save the changes?
```

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

Restoring the Default Values

You can reset all the parameters in the current INI file to their default settings at any time while the Editor is running.

To restore one parameter to its default

- Click the Default button next to the parameter.

To restore all parameters to their 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. Click the Yes button. The Editor resets the parameters to their defaults and displays the message:

```
All defaults have been restored.
```

3. Click the OK button.

Printing INI Files

You can print the INI file from the Editor or with any ASCII text editor.

Note: If you have not set up the printer for the Editor, choose *Print Setup* from the *File* menu and select the printer options as you would for any Windows application.

To print an INI file

1. From the *File* menu, choose *Print*.
2. Click the *OK* button at the *Print* dialog box.

Exiting the Simulator 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

- 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 a message similar to this one:

```
The filename file has been changed.  
Do you want to save the changes?
```

Click the *Yes* button if you want to save the changes. Click the *No* button if you do not want to save the changes.

5

Troubleshooting

This chapter describes problems and error messages you may encounter while using the Application Simulator, Simulator Editor, and the Serial Input Editor.

Problems Operating the Application Simulator

This section describes problems you may encounter when using the TRAKKER T2090 Application Simulator.

Linking to the Wrong IMT209ML.LIB Library

You may experience unpredictable results if the Application Simulator does not link to version 2.1 or higher of the IMT209ML.LIB library.

Check the Options/Projects/Linker and Options/Library Files Path in the Microsoft Visual C/C++ Professional Edition to make sure you have the right library.

If you are using EZBuilder, also check the Build Options PSK path and included library list.

T2090 Application Fails to Launch

The T2090 application may fail to launch when you select the Simulate T2090 command from the Tools menu of Microsoft Visual C\C++. Here are three possible solutions:

- Make sure the IMT29SIM.BAT file is in the *Target_Directory*\IMT209\SIM directory. This batch file helps launch the T2090 application from the working directory on the PC.
- The IMT29SIM.BAT file may be corrupted. Delete the batch file. Create and save a new batch file in the *Target_Directory*\IMT209\SIM directory. The file should contain these two characters only:
%1
- Make sure the T2090 application is in the working directory on the PC.

T2090 Application Locks Up

The T2090 application may lock up or crash on the PC while you are using the Application Simulator to run it. Here are three possible solutions:

- The application may have errors. Debug the application, correct the errors, and try the Application Simulator again.
- The PC memory may be corrupted. Run a memory check with available software utilities to determine if the memory has been corrupted.

Problems Simulating Bar Code Input With a Wedge

If you are having difficulty using an Intermec Wedge to provide bar code input while you run a TRAKKER T2090 application, you may have set the wedge preamble incorrectly.

Make sure the wedge preamble is **Ctrl-G**. When 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 scan the SO character for **Ctrl** and the lowercase **g** character for **G**.

If you scan an uppercase **G**, the wedge preamble will be set to **Ctrl-Shift-G**, which will not cause the Application Simulator to treat the data as bar code label input.

Error and Status Messages

This table describes the error and status messages you may see when using or installing the Simulator Editor and the Serial Input Editor. Follow the instructions in the Suggested Action column to recover from the error.

Message	Description	Suggested Action
File Not Found	The file you specified at the Serial Input Editor screen cannot be found.	Make sure you have the correct file name. Enter the name again with the complete path, including the drive.
No INI File Found	The Simulator Editor did not find the default INI file for the TRAKKER Antares Application Simulator in the working directory. This is expected because you are using the TRAKKER T2090 Application Simulator.	Open the IMT209SM.INI file by choosing Open from the File menu.
Parameter <i>name</i> is invalid.	The parameter is not valid for the T2090 Application Simulator.	Delete the parameter from the IMT209SIM.INI file and restart the application.
The <i>filename</i> file has been changed. Do you want to save the changes?	You have made changes that have not been saved yet.	Click the Yes button to save the changes, or click the No button to discard the changes.
<i>Filename</i> file not found. Please verify the correct file name was given.	You specified a file that the application cannot find.	Make sure you entered the file name correctly. Make sure the file exists.

Message	Description	Suggested Action
File name must be imt209sm.ini.	You entered an invalid name for the INI file.	The INI file for the TRAKKER T2090 Application Simulator must be named IMT209SM.INI. If you specify any other name for the INI file, the simulated application will not recognize the file.
Save all values in <i>filename</i> .	This message confirms that all the parameters will be saved in the specified file.	Click the OK button.
All parameters will be reset to their default values. Do you want to proceed?	You have chosen Restore Defaults from the File menu. All parameters in the IMT209SM.INI file will be reset to their default values.	Click the Yes button to restore the defaults, or click the No button to cancel the request.
All defaults have been restored.	This message confirms that all the parameters in IMT209SM.INI have been returned to their default values.	Click the OK button.



***Adding the Simulator to the
Microsoft Visual C/C++ Tools Menu***



This appendix shows how to add commands to the Tools menu in Microsoft Visual C/C++.

Adding the Simulator to the Tools Menu

If Microsoft Visual C/C++ was installed before you installed the PSK and Application Simulator on your PC, two commands were automatically added to the Tools menu of Microsoft Visual C/C++:

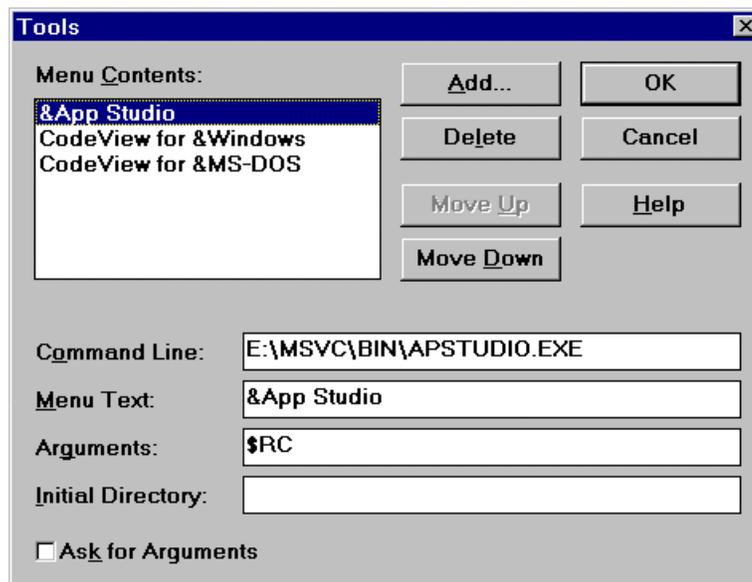
Simulate T2090 Choose this command to simulate the application.

T2090 Serial Input Editor Choose this command to start the Serial Input Editor.

If the commands were not added to the Tools menu, you can add them manually at any time by following the instructions in the next two procedures.

To add the Simulate T2090 command to the Tools menu

1. Start Microsoft Visual C/C++, and select Tools from the Options menu. The Tools dialog box appears.

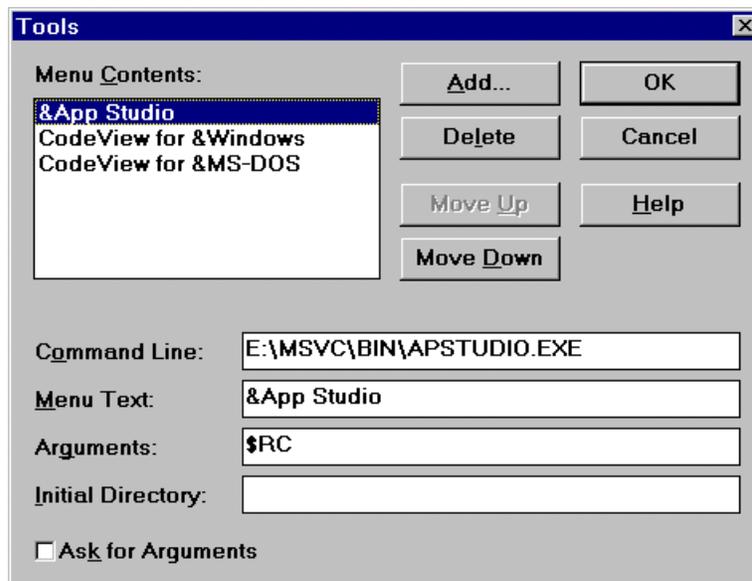


2. Choose Add to access the Add Tool dialog box.
3. Select the IMT29SIM.BAT file and choose OK to return to the Tools dialog box. The default location for this file is *Target_Directory\IMT209\SIM*.
4. In the Menu Text field, add this string: `Simulate T2090`
5. In the Arguments field, add this string: `$Target`

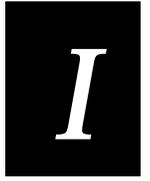
6. Choose Add to add a new command to the Tools menu. The Add Tool dialog box appears.
7. Choose OK.
8. From the main menu, choose Tools. The Simulate T2090 command appears on the Tools menu.

To add the T2090 Serial Input Editor command to the Tools menu

1. Start Microsoft Visual C/C++, and select Tools from the Options menu. The Tools dialog box appears.



2. Choose Add to access the Add Tool dialog box.
3. Select the SERIALED.EXE file and choose OK to return to the Tools dialog box. The default location for this file is *Target_Directory*\IMT209\SIM.
4. In the Menu Text field, add this string: T2090 Serial Input Editor
5. Choose Add to add a new command to the Tools menu. The Add Tool dialog box appears.
6. Choose OK.
7. From the main menu, choose Tools. The T2090 Serial Input Editor command appears on the Tools menu.



Index

A

Application Simulator
 compatibility with other products, 1-5
 how to begin simulating applications, 1-6
 installation, 1-3
 introduction, 1-3
 technical description, 2-3
 TRAKKER Antares Application Simulator, 4-3
 ASCII text editor, 3-3

B

bar code input
 choosing the symbology, 2-5, 3-6
 with a keyboard, 2-6
 with a wedge, 1-5, 2-6, 5-4

C

Character mode, not simulated, 2-8
 colors, simulating display characteristics, 2-5, 2-9, 2-10
 COM1, 2-6, 2-7
 communications
 how to simulate input through COM1, 2-6
 how to simulate output through COM1, 2-7
 serial communications example, 2-3
 COMP1.RCV, 2-10
 COMP1.TRX, 2-11, 3-7
 compatibility with other products, 1-5
 contrast level, not simulated, 2-8
 creating a new INI file, 4-5
 Ctrl-G, 2-6, 5-4
 current file, 4-6

D

data I/O example, 2-3
 defaults
 printing a list of, 4-9
 restoring, 4-8
 development process, 1-5
 display characteristics, shown in color, 2-5, 2-9, 2-10
 display size, how to set, 2-4

E

editing INI parameters, 3-3
 Editor, *See* INI File Editor or Serial Input Editor
 error messages
 listed and described, 5-4
 No INI File Found, 4-4
 example, serial communications, 2-3
 EZBuilder, 1-5

F

features that are not simulated, 2-8
 features that are simulated
 bar code input, 2-5
 data input through COM1, 2-6
 data output through COM1, 2-7
 keypad input, 2-5
 overview, 2-4
 reader command syntac checking, 2-7
 text display, 2-4
 file transfer, not simulated, 2-8
 formatting sample input data, 2-6
 Frame mode serial communications, 2-6

H

help, *See* online help

I

IMT209SIM environment variable, 3-7
 IMT209SM.INI file, 2-3, 3-3, 5-5
 IMT29SIM.BAT file, 5-3
 INI parameters, *See* parameters
 initialization file, *See* INI file
 input data, formatting, 2-6
 installation, 1-3
 Intermec wedge, *See* wedge

K

key sequence for bar code input, 2-6
 keyboard, simulated bar code input, 2-6

L

label time stamp, not simulated, 2-8
 label_postamble, 3-5
 label_postamble_string, 3-5
 label_preamble, 3-4
 label_preamble_string, 3-4
 label_symbologyid, 3-6

M

Microsoft Codeview for DOS, 1-4, 1-5
 Microsoft Visual C/C++, 1-4, 1-5, A-3
 Microsoft Windows 95, 1-5

N

No INI File Found, 4-4

O

online help
 accessing Serial Input Editor help, 2-4
 accessing Simulator Editor help, 4-5

P

parameters

- how to edit, 3-3
- label_postamble, 2-5, 3-5
- label_postamble_string, 2-5, 3-5
- label_preamble, 2-5, 3-4
- label_preamble_string, 2-5, 3-4
- label_symbology, 2-5, 3-6
- label_symbologyid, 2-5, 3-6
- reasons for customizing, 3-3
- restoring the default values, 4-8
- serial_port_emulation, 2-3, 2-10, 2-11, 3-7
- sim_directory, 3-7

printf, not simulated, 2-8

printing, 4-9

programmer input mode, 2-4

PSK functions

- changes made to support simulation, 2-3, 2-9
- im_command, 2-8, 2-12
- im_cputs, 2-9
- im_draw_underline, 2-9
- im_get_kbmode, 2-11
- im_get_label_symbology, 2-12
- im_get_label_symbologyid, 2-12, 3-6, 3-7
- im_putchar, 2-10
- im_puts, 2-9
- im_receive_buffer, 2-10
- im_receive_field, 2-11, 3-4, 3-5
- im_receive_input, 2-11, 3-4, 3-5
- im_set_input_mode, 2-4
- im_set_kbmode, 2-11
- im_transmit_buffer, 2-11

R

reader commands, 2-7, 2-12

README file, 1-4

restoring defaults, 4-8

S

sample input data, formatting, 2-6

saving INI file, 4-6

Select Type dialog box, 4-5

Serial Input Editor

- defined, 2-3
- serial_port_emulation INI parameter, 3-7
- starting, 2-4
- used for simulated data input, 2-6

serial_port_emulation, 3-7

sim_directory, 3-7

simulation, how it works, 2-3

Simulator Editor

- creating a new INI file, 4-5
- defined, 2-3
- discarding changes, 4-8
- exiting, 4-9
- No INI File Found, 4-4
- online help, 4-5
- opening an INI file, 4-6
- printing an INI file, 4-9
- restoring the defaults, 4-8
- saving changes, 4-6
- Select Type dialog box, 4-5
- starting, 4-3
- two application simulators, 4-3
- user interface, overview, 4-4
- speed and performance, not simulated, 2-8
- starting the Simulator Editor, 4-3
- symbology available, 3-6

T

T2090 computer, 1-3

Target_Directory, defined, viii, 1-3

Tools menu, adding Simulator commands, 1-5, A-3

TRAKKER Antares Application Simulator, 4-3

TRAKKER T2090 Application Simulator, 4-3

TRAKKER T2090 computer, 1-3

TRAKKER T2090 PSK v2.1, 1-5

troubleshooting

- application does not launch, 5-3
- application locks up, 5-3
- using a wedge for bar code input, 5-4
- wrong IMT209ML.LIB, 5-3

W

wedge

- simulating bar code input, 1-5, 2-6
- troubleshooting, 5-4

Windows 95, 1-5

working directory, defined, viii