



Getting Results Guide

Doc ID EMULAT-GR002A-EN-P



Contacting Technical Support Telephone—1-440-646-7800 Rockwell Software Technical Support Fax—1-440-646-7801 World Wide Web-www.software.rockwell.com Copyright Notice © 2004 Rockwell Automation Inc. All rights reserved Printed in the United States of America Portions copyrighted by Allen-Bradley Company, LLC, a Rockwell Automation company. This manual and any accompanying Rockwell Software products are copyrighted by Rockwell Automation Inc. Any reproduction and/or distribution without prior written consent from Rockwell Automation Inc. is strictly prohibited. Please refer to the license agreement for details. Trademark Notices The Rockwell Software logo, RSLinx, RSLogix 5, RSLogix 500, RSTrend, RSView32, RSView, A.I. Series, and Advanced Interface (A.I.) Series are trademarks of Rockwell Automation Inc. Data Highway Plus, DH+, PLC-5, SLC, and SLC 500 are trademarks of the Allen-Bradley Company, LLC, a Rockwell Automation company. Microsoft, Windows NT, Windows 2000, Windows XP, Windows ME, and Windows 98 are trademarks of the Microsoft Corporation. Pentium is a registered trademark of the Intel Corporation. Adobe and Acrobat are trademarks of Adobe Systems Incorporated. All other trademarks are the property of their respective holders and are hereby acknowledged. Warranty This Rockwell Software product is warranted in accord with the product license. The product's performance will be affected by system configuration, the application being performed, operator

control and other related factors.

The product's implementation may vary among users.

This manual is as up-to-date as possible at the time of printing; however, the accompanying software may have changed since that time. Rockwell Software reserves the right to change any information contained in this manual or the software at anytime without prior notice.

The instructions in this manual do not claim to cover all the details or variations in the equipment, procedure, or process described, nor to provide directions for meeting every possible contingency during installation, operation, or maintenance.

Contents

Preface	
Purpose of this document	iii
Intended audience	iii
How does the getting results guide fit in with other Rockwell Software product documentation?	
Document conventions	iv
Feedback	iv

Chapter 1

nstalling RSLogix Emulate software1	
Introduction	1
System requirements	1
Installing RSLogix Emulate software	2
Starting RSLogix Emulate software	4
Troubleshooting installation	4

Chapter 2

Exploring RSLogix Emulate	5
What RSLogix Emulate does	5
What happens when RSLogix Emulate runs	6
Navigating RSLogix Emulate	7
Starting your work	8
Set up communications for RSLogix Emulate	8
Load a ladder logic project into RSLogix Emulate	9
Editing ladder logic during emulation	10

Chapter 3

Using debug files	11
About debug files	

Creating debug files	.13
Selecting debug files	13
Uploading and downloading with debug files	. 13

Chapter 4

Using breakpoints1	
Breakpoint strategies	15
Setting breakpoints	
Editing breakpoints	
Enabling and disabling breakpoints	
Deleting breakpoints	

Preface

Purpose of this document

This getting results guide provides you with information on how to install and navigate the RSLogix Emulate software. It explains how to access and navigate the online help, and how to effectively use the RSLogix Emulate software.

Intended audience

We assume that you are familiar with:

- IBM-compliant personal computers
- Microsoft® Windows® operating systems
- RSLinxTM communications software
- Ladder logic programming for Allen-Bradley PLC-5 and/or SLC-500 processors

How does the getting results guide fit in with other Rockwell Software product documentation?

The Getting Results with RSLogix Emulate guide can be considered the entry point into Rockwell Software's documentation set for this product. Other components of the documentation set include electronic release notes and online help.

The documentation set contains pertinent, easily accessible product information and ships with the software product. This set ships with the software product, and is designed to free you from tedious paper shuffling and reduce information overload.

Online help

The online help includes all overview, procedural, screen, and reference information for the product. The help contains overview topics, step-by-step procedures, and screen element descriptions (for example, text boxes, dropdown lists, and option buttons). All of the help is context-sensitive with the application and provides you with immediate access to application tasks and screen element descriptions.

Document conventions

The conventions used throughout this document for the user interface comply with those recommended by Microsoft. If you are not familiar with the Microsoft Windows user interface, we recommend that you read the documentation supplied with the operating system you are using before attempting to use this software.

Feedback

Please use the feedback form packaged with your software to report errors or let us know what information you would like to see added in future editions of this document.

Installing RSLogix Emulate software

Introduction

This chapter explains how to install and start RSLogix Emulate software. This chapter includes information on the following:

- system requirements
- installation procedures
- starting the software

System requirements

To effectively use RSLogix Emulate, your personal computer must meet the following hardware and software requirements:

Hardware requirements

- a Pentium[™] or Pentium-compatible microprocessor
- 256 MB of RAM
- 10 MB of available hard disk space
- a CD-ROM drive
- a 3.5-inch, 1.4 MB disk drive

Software requirements

- The operating system must be Microsoft Windows NT[®] 4.0 with Service Pack 6 or greater, Microsoft Windows 2000[®], Microsoft Windows XP[®], Microsoft Windows ME[®], or Microsoft Windows 98[®].
- RSLinx[™] communication software, version 2.00.118 or later.

Installing RSLogix Emulate software

Important

If you are running Microsoft Windows NT, Microsoft Windows 2000, or Microsoft Windows XP and performing the tasks that you will read about in this chapter, you must have system administrator privileges and your user account must be a member of the local administrator user group. For more information, contact your system administrator.

To install RSLogix Emulate software, perform the following steps:

1. Insert the RSLogix Emulate CD-ROM into the CD-ROM drive.

If autorun is:	Then:
enabled	The Setup program starts automatically. Proceed to step 3.
disabled	 Perform the following steps: a. Click Start, then click Run. The Run dialog box appears. b. In the Open field, type <i>x</i>:\autorun, where <i>x</i> is the letter of the drive containing the RSLogix Emulate CD-ROM. c. Click OK. The Setup program starts.

- **2.** RSLogix Emulate software must be activated before it can run on your computer.
 - **a.** If you have never installed RSLogix Emulate software on your computer, click **Install FactoryTalk Activation**.
 - **b.** Follow the instructions that appear on the screen. If you need help activating the software, click the **Help** button that appears in the activation software.



If you receive a message saying that activation was found on your machine, you do not need to install FactoryTalk Activation.

- You should read the release notes for the software before proceeding with the installation. Click View RSLogix Emulate 5 (or 500)
 Documentation, and then click Read RSLogix Emulate 5 (or 500)
 Release Notes. When you have finished reading the release notes, close the release notes document, and then click Back.
- 4. Click Install RSLogix Emulate 5 (or Install RSLogix Emulate 500).
- 5. Follow the instructions that appear on the screen.
 - a. On the Welcome dialog box Read the RSLogix Emulate introductory information, and then click Next.
 - b. On the Software License Agreement dialog box Read the entire Software License Agreement. Click I accept the terms of the licensing agreement to accept and continue installation, or click I do not accept the terms of the licensing agreement to decline and exit the installation. (If you do not accept the terms of the agreement, you will not be able to install the software.) Click Next.
 - **c.** On the Customer Information dialog box Type your name, the name of your company, the serial number of your RSLogix Emulate software, and then click Next.

 Tip
 You can find the serial number on the product box label.

 Image: I

- d. On the Setup Type dialog box Choose whether you want a Complete or Custom setup. A Complete setup installs all software components and places them in the default folder. A Custom setup allows you to choose software components to install and/or a custom folder for installation. Click Next.
- e. If you chose a Complete setup, skip to step f.. If you chose a Custom setup, the Custom Setup dialog appears. Choose the program components you want to install by right-clicking on them. Choose the installation folder by clicking the **Change** button. When you have completed your selections, click **Next**.
- f. On the Ready to Install the Program dialog box Click Install to install the software.
- **g.** The software installs on your computer. When the installation is complete, click **Finish**.
- **6.** When you are finished installing the software, remove the RSLogix Emulate CD-ROM from the CD-ROM drive and store it in a safe place.

Starting RSLogix Emulate software

To start RSLogix Emulate software: Click **Start**, and then select **Programs** > **Rockwell Software** > **RSLogix Emulate** (5 or 500) (folder) > **RSLogix Emulate** (5 or 500) (executable).



We assume that you used the default names for the directory and program group. If you did not use the default names, substitute the actual names that you specified for the default names shown.

Troubleshooting installation

If RSLogix Emulate does not start up or run properly, keep the following in mind:

- Do you have the correct version of RSLinx installed? The software requires RSLinx version 2.00.118 or later.
- Does your computer have enough memory? Running RSLogix Emulate requires a minimum of 256 MB of RAM (or more, depending on how many other applications you wish to run at the same time).

^{Chapter} Exploring RSLogix Emulate

What RSLogix Emulate does

RSLogix Emulate is a troubleshooting and debugging tool that can emulate most operations of Allen-Bradley PLC-5 and SLC-500 family processors. It executes your ladder logic programs in your computer, updating your programs' data tables, allowing you to approximate what is going to happen when you download your programs to physical PLC-5 processors.



While RSLogix Emulate is useful for verifying the integrity of your program, you should not rely solely on RSLogix Emulate for final program checkout. Variations in processor function (and variations based on processor series and revision) dictate that final checkout must be performed with the actual processor and I/O for your system.

When a PLC or SLC runs a ladder logic program, the following occurs:

- **1.** The inputs from your physical I/O generate a response in your processor's data table.
- 2. The processor executes your ladder logic, using the inputs from your I/O and other data table values to generate new values for your outputs (and other data table values).
- **3.** The output values in your data table are applied to your outputs in physical I/O.

This sequence of events is called one program scan. Depending on the processor and the program, 10 to 50 program scans occur each second.

When you use Rockwell Software products to program your processor, the ladder logic and data table are stored in your computer. RSLogix Emulate executes the stored ladder logic and updates the data table as if the processor was running.

Simulating I/O

There's something missing when RSLogix Emulate runs a processor's program: the I/O. Since your computer will not be connected to your I/O modules, RSLogix Emulate uses debug files to simulate the inputs for your ladder logic program.

Debug files are sections of ladder logic that you program to simulate your process' inputs. The ladder logic in a debug file can be edited like any other program file. For more information about using debug files, see page 11.

RSLogix Emulate has some powerful troubleshooting features. You can:

- scan your ladder logic continuously, one rung at a time, or one program scan at a time
- restrict the program scan to a block of rungs that you specify
- set breakpoints in your ladder logic to stop execution when a specified event occurs (e.g., when an important value goes out of range), freezing the conditions that caused the breakpoint to trigger

Channel 0 emulation

RSLogix Emulate also allows you to emulate Channel 0 on your processor (for those projects that use Enhanced PLC-5s or the SLC-5/03 and -5/04). This feature is handy for troubleshooting programs that interface with serial devices (e.g., bar code readers, scales, terminals, etc.).

As a debugging tool, you can feed trend data into RSLogix Emulate to "replay" your data through your ladder logic.

What happens when RSLogix Emulate runs

RSLogix Emulate uses your computer's CPU to scan the rungs in your ladder program. The rungs in your program read inputs from and write outputs to the data table stored offline with your ladder logic project. The offline data table is also active during emulation.

RSLogix Emulate reads the ladder program and data table into your computer's memory before executing the ladder logic. The files on your disk are not affected by the emulator.

The basic order of events during emulation is no different from when a processor runs a project. RSLogix Emulate scans your rungs, pauses to update the output and input image tables, and scans your rungs again. (During normal operation, RSLogix Emulate repeats the program scan until you tell it to stop. There are other scanning modes you can use.)

Since there is no real I/O, the emulated ladder logic takes cues only from the state of the data table. To generate responses in your ladder program, you will need to change the value of the desired I/O bits, storage bits, or storage words acting as inputs to your ladder program.

Тір

P

Remember that the data table is always active, no matter which rungs are actually being scanned. It is not unusual to see inputs and outputs highlighting and values changing (like in an ADD instruction) on rungs that are not being scanned. If a dormant rung contains addresses that are being changed by an actively scanned rung elsewhere in the program, you will see the changing values on the dormant rungs.

Navigating RSLogix Emulate

These are the basic controls for RSLogix Emulate:



- When you have a file in memory, it appears in the file selector's pull-down menu. Select the file you want to act on from the file selector.
- Click on the Load File or Unload File icon to load a ladder logic file into or out of memory.
- Use the mode selector buttons to change the "processor" mode of a program in memory. You can set the following modes:
 - **Program mode**, which halts execution of the program (you can't edit the program in memory)
 - Run mode, which runs the program as if it was in a processor
 - **One-rung mode**, which runs the program by scanning one rung at a time
 - **Single-scan mode**, which runs through one complete program scan, then stops execution

The software's other features (breakpoints, Channel 0 emulation) are accessible from the pull-down menus.

Starting your work

If all you want to do is see if your logic runs, there's very little you need to do with RSLogix Emulate. If you need to simulate your I/O with debug files or set up a Channel 0 device, you will need to do some more work.

Here are the steps to get going with RSLogix Emulate:

- **1.** Set up communications for RSLogix Emulate. See "Set up communications for RSLogix Emulate" on page 8.
- **2.** Load your ladder logic project into RSLogix Emulate. See "Load a ladder logic project into RSLogix Emulate" on page 9.
- **3.** Start your ladder logic editing application, and go online with RSLogix Emulate. See the documentation for your ladder logic editing software.
- 4. Click the Run button in the RSLogix Emulate window.

You can then monitor data table and ladder logic files for your process as if it was running in an actual processor.

Set up communications for RSLogix Emulate

If you have set up your communications to your physical processors in your communications software, you'll find setting up communications for RSLogix Emulate a simple procedure. Basically, you'll set up communications between RSLogix Emulate and your communications software exactly as if RSLogix Emulate is a hardware device.

Setting up communications with RSLinx

Here's how to set up communications between RSLinx and RSLogix Emulate:

- 1. Open RSLinx.
- **2.** In RSLinx, click **Communications > Configure Drivers**. RSLinx displays the Configure Drivers dialog.
- **3.** From the **Available Driver Types** list, select PLC-5 (DH+) Emulator or SLC-500 (DH485) Emulator, and click the **Add New** button. RSLinx displays a dialog for naming the driver you selected. Click **OK** to continue.

- **4.** RSLinx displays a dialog for configuring the driver. Type a station number for RSLogix Emulate in the **Station Number** field. Remember, RSLogix Emulate is being treated as if it were a hardware device, so it needs a unique station number. You can also type a name for RSLogix Emulate in the **Station Name** field.
- **5.** When you have configured the driver, click **OK**. The driver configuration dialog closes.

When RSLogix Emulate is running with programs in memory, you can use the Who Active or RSWho functions and see the emulator as if it was a PLC-5 on a Data Highway Plus network or an SLC-500 on a DH-485 network. Other applications that use RSLinx can also access the programs as if they were on a communications network.

Load a ladder logic project into RSLogix Emulate

To bring a ladder logic project into RSLogix Emulate's memory:

- In RSLogix Emulate, click File > Open. RSLogix Emulate displays the Open dialog.
- 2. Highlight the file you want to open, and click **OK**.
- **3.** RSLogix Emulate displays a configuration dialog for the ladder logic file. In this dialog, specify the following:

This field or setting:	Sets:
File #	the ladder file from which scanning is started. During a program scan, the main file, any subroutines called by the main file, and any nested subroutine files are executed.
Start Rung	the first rung in the main file to be scanned. A rung number of 0 starts the scan from the beginning.
End Rung	the last rung to be scanned. The rung number for this field IS scanned. In order to scan one rung, make the start and end numbers identical. Setting the end rung to -1 scans to the end of the file.
Station #	the "Data Highway Plus/Data Highway 485" station number for the emulated processor.
Restore Mode on Start	whether the emulated program will retain whatever mode it had when RSLogix Emulate is closed. If this box is checked, the program will reopen in its last mode.

4. Click the **OK** button. RSLogix Emulate reads the file into memory, and holds it there.

Tip 💡	With this same dialog, you can specify debug file options. Debug files are not required for RSLogix Emulate to function, but are used to simulate real I/O. For more information about using debug files, see page 11.
Тір	Once a file is opened in RSLogix Emulate, that file remains in the
@	file list. Whenever RSLogix Emulate is opened, that file will also be taken into memory. You can close the file which tells RSL ogix
	Emulate to not load it again unless you tell it to load the file.

Editing ladder logic during emulation

You cannot edit ladder logic while online with the emulator. (You can change data values, but you cannot edit logic rungs.) If you see something in your ladder logic you want to change:

If you are using this ladder logic programming software:	You can change the ladder program by:
RSLogix 5 or RSLogix 500	Edit offline and download to the emulator as if you were downloading to an actual processor. (You cannot, however, edit logic while online with the emulator.)
A.I. Series software	1. Go into offline editing in your ladder logic editor, make your changes, and save your changes to disk.
	 Reload the project into RSLogix Emulate by clicking the Reload icon. Note that you will have to stop emulating by clicking the HLT button before the software will allow you to reload the project.

Using debug files

RSLogix Emulate software does not simulate the actions of I/O. If you need to simulate the actions of I/O to increase the accuracy of your emulations, you will need to create debug files.

About debug files

Debug files are ladder logic. Your program uses ladder logic to control a process; a debug file uses ladder logic to simulate the process.

Anytime your program acts on a specific sequence of events (e.g., parts coming down an assembly line), you can simulate the process by programming ladder logic in a debug file. The following is an example.

Let's suppose you have a process that monitors a part's progress down an assembly line. There is a bit in your process that controls the conveyor motor. You have proximity switches that trigger operations when a part reaches specific areas of the assembly line. A debug file can easily simulate the part moving down the assembly line. Here is an example of how you can do this.



The first rung is conditioned on the start button for the process. The process starts, and TR1 starts timing. The preset should be approximately the time to get a part to the first station.

When TR1 is done timing, the PROX_1 proximity switch turns on. Since the proximity switch is on, the process at station 1 should activate.

By expanding on this concept, you could emulate the movement of a part down the entire assembly line.

Emulating block transfer reads

By using debug files, you can emulate the activity of Block Transfer Read (BTR) instructions.

BTR instructions take a specific number of words from an I/O module, and place those words in an area of the processor's data table. Since there is no real I/O while emulating a process, BTR instructions have no data to read.

To emulate a BTR, place a Sequencer Output (SQO) instruction in a debug file. Program the SQO so that the destination address of the SQO is the same as the BTR's Data File address. Enter the test data into the source words of the SQO. (You might want to enter a few values that are out of range for your process to test how the ladder logic program handles the erroneous data.)

When the SQO is executed by the emulator, it passes the data to the BTR's Data File. Your process can then use the data. You can execute the SQO automatically by triggering it off the done bit (.DN) of a free running timer in the same debug file.

Emulating many inputs

You can use the SQO instruction to emulate inputs. Program an SQO with a destination address in the input data table file. This way, you can send entire bit patterns to the input words instead of changing the individual bits.



Remember that the inputs will remain in their last states while emulating. If you want to reset your inputs, you will need to reset them manually. You can use an FFL instruction for this purpose.

Emulating fault routines

The status and fault bits in the processor are used by RSLogix Emulate. If you have a fault routine or shutdown sequence in your ladder program, you can test it by generating a fault or shutdown signal in a debug file.

Limit your debug files

If you try to program a single debug file to simulate the inputs for your entire process, you could end up with a debug file that is larger than your entire ladder logic program! Try to break down your process into pieces and create debug files to handle each piece. As you debug the pieces of your ladder program, you can combine your debug files to test a larger portion of your ladder program.

Creating debug files

You create debug files with your ladder logic editing application. Rockwell Software applications for editing ladder logic all support creating debug files. For information about creating debug files with your ladder logic editing software, please see the documentation for that software package.

Selecting debug files

You can select which debug file you want to use with the emulation session. You can select the debug file when you first open the ladder logic file in RSLogix Emulate, or you can define it later. You can also change the debug file as RSLogix Emulate scans the ladder logic. You can have multiple debug files defined in your ladder logic program, but you can use only one at a time.

To specify which debug file you want to use:

- Click Settings > Emulation. The software displays a configuration dialog.
- 2. In the Debug File # field, type the number of the debug file.
- **3.** In the **Start Rung** field, type the number of the first rung to scan in the debug file.
- **4.** In the **End Rung** field, type the number of the last rung to scan in the debug file.
- 5. Click OK. The software uses the new settings for the debug file.

Uploading and downloading with debug files

With all of these debug files in your project, you may wonder what happens when you download to your processors. What happens to the debug files? The programming software filters out debug files. Debug files are **not** downloaded to your processors. When you upload from a processor into a project that has debug files, the debug files are deleted. You should save debug files in a separate ladder logic project, or save the debug files as libraries to load back into your project. See the documentation for your ladder logic editing software for information on saving projects to different file names and using program libraries.

Chapter Using breakpoints

In ladder logic programs, error conditions often appear and disappear before you can decipher what happened (let alone troubleshoot the problem). To assist you in troubleshooting your program, RSLogix Emulate allows you to set breakpoints in your ladder logic. Breakpoints are functions that stop program execution when certain conditions are met. This is called "hitting" a breakpoint. When you set a breakpoint, you are telling RSLogix Emulate, "When these conditions are met, stop the process so I can take a look around."

Breakpoint strategies

If you find yourself wishing that you could "take a snapshot of the processor's data table the next time that happens," you can probably use breakpoints. When your program encounters a breakpoint, the data table is frozen in the state that caused the breakpoint to hit.

The strategy for using breakpoints is simple. If you have an error condition that has a short duration, or an impossible condition that insists on occurring, breakpoints can help you debug the problem.

Setting the right conditions

When using breakpoints, you should strive to make the conditions that trigger the breakpoint as specific as possible. In many cases, there won't be a single rung in your program that will define the desired breakpoint conditions. In these cases, write down all of the specifics that will trigger the breakpoint, and program a temporary rung that collectively defines the breakpoint condition.

For example, let's assume your process has the following conditions:

- the process generally runs hot and needs cooling
- an analog input module is reporting the process' temperature
- a cooling fan helps control the running temperature, and turns on when the input temperature reaches 180 degrees
- raw materials are fed into the process by a conveyor which turns off when the process temperature reaches 175 degrees

The problem is that the fan never comes on. The temperature is reading correctly. The conveyor stops when the temperature reaches 175 degrees, but the fan never comes on.

In this case, you would add a rung to your program to trigger a breakpoint, checking for the following:

- the process temperature is greater than or equal to 180 degrees
- the conveyor is not running

The instant the breakpoint conditions are met, the breakpoint would stop the software, and you could examine the input conditions for the cooling fan to determine which input condition is preventing the fan from starting.

Manipulating and storing breakpoints

The software stores breakpoints in a file separate from the ladder logic and data table. This file is not downloaded to the processor.



If you make changes to your ladder logic that cause the rungs to renumber, you may need to redefine your breakpoints.

Setting breakpoints

Breakpoints are defined by file number, rung number, whether the rung should be true or false, and the number of times the rung is true or false (Count).

To set a breakpoint:

- Click Ladder > Breakpoints. The software displays the Breakpoints dialog.
- 1. Click Add New.
- **2.** In the **File** field, type the number of the ladder file in which you want to set the breakpoint.
- **3.** In the **Rung** field, type the number of the rung for which you want to set the breakpoint.

4. In the **Count** field, type the number of scans you want to occur with the rung in the state you select. For instance, if you want to "hit" the breakpoint when the rung is true for 40 scans, type 40 in the Count field.



This doesn't mean the rung must scan with the same condition consecutively. For instance, if you set a TRUE breakpoint with a count of 40 on a rung that goes true once per minute, the breakpoint would hit after 40 minutes.

5. From the list box under the **Count** field, select the state you want to examine with the breakpoint. For instance, if you want to count when the rung is true, select TRUE.

TRUE	when the rung is true
FALSE	when the rung is false
ANY	when the rung is scanned, regardless of the state of the rung

Here is an example:

Rungs are considered TRUE when there is logical continuity from the beginning of the rung to at least one output. The examples below show how some rungs are assessed by RSLogix Emulate. The gray squares indicate which instructions evaluate as true.



6. Click Save. The software moves the breakpoint into the breakpoint list.

Editing breakpoints

To edit a breakpoint:

- 1. Click Ladder > Breakpoint.
- **1.** Click the breakpoint you want to edit.
- **2.** Click **Edit**. The software moves the parameters for the breakpoint into the fields at the right of the dialog.
- **3.** Edit the breakpoint's parameters. When you are through editing, click **Save** to save the changes.

Enabling and disabling breakpoints

You can disable breakpoints so that they are still in place but will not stop program execution. To do this, click the breakpoint you want to disable, and click **Toggle**. The active/inactive state of the breakpoint is shown in the state column in the breakpoint list: **A** indicates an active breakpoint, and **I** indicates an inactive breakpoint.

Deleting breakpoints

To delete a breakpoint, click the breakpoint you want to delete, then click **Delete**. The breakpoint is deleted.

Index

B

Block transfers emulating • 12 Breakpoints • 6, 15 deleting • 19 editing • 18 enabling and disabling • 18 setting • 16 strategies for using • 15

С

Channel 0 emulation • 6 Communications • 8

D

Data table • 7 Debug files • 6, 10, 11 creating • 13 described • 11 selecting which to execute • 13 uploading and downloading to physical processors • 13 downloading • 10

Е

editing ladder logic • 10 End rung • 9

F

Fault routines emulating • 12 File number to start scanning • 9

Η

Help with Emulate • iv

I

I/O simulating • 6 Icons • 7 Inputs emulating • 12 Installing RSLogix Emulate • 1, 2

L

Ladder logic loading into RSLogix Emulate • 9 ladder logic editing during emulation • 10 Load file • 7

M

Modes • 7

0

One-rung mode • 7 Online help • iv

Р

Program mode • 7 Program scan • 5

R

reloading ladder logic • 10 Restore mode on start • 9 RSLinx • 1, 4, 8 RSLogix Emulate configuring emulation • 9 description of function • 5 installing the software • 1, 2 loading ladder logic into • 9 navigating • 7 runtime behavior • 6 starting the software • 4 troubleshooting installation • 4 RSTrend • 6 Run mode • 7

S

Scan • 5, 6 Single-scan mode • 7 Start rung • 9 Starting RSLogix Emulate • 4 Station number of emulated processor • 9 Support ID number • 3

Т

Trend data feeding into RSLogix Emulate • 6 Troubleshooting RSLogix Emulate installation • 4

U

Unload file • 7