505™ Programming Software for Windows®

505 WorkShop™

Version 3.50

By FasTrak SoftWorks, Inc.

1 - Introduction

Manual Design

Welcome to 505 WorkShop for Windows, a powerful Windows-based tool for programming programmable logic controllers (PLCs). Whether you are a novice or an experienced programmer, this manual has been constructed to help you begin using 505 WorkShop for Windows quickly. We at FasTrak have tried to assume little about you, the user, except that when you have a question regarding this software, you will want it answered by using this manual and the online help.

Chapter topics use names that point you quickly to the specific information you want to find. Titles of Chapters are listed in the following table:

Chapter Name	Description
Introduction	Outlines manual contents, Customer Support numbers, and necessary hardware and software to run 505 WorkShop for Windows.
Installation	Guides you through the installation procedures for the software and security.
505 WorkShop for Windows Basics	Walks you through the nitty gritty by describing 505 WorkShop for Windows features and helps you move through the Windows environment.
505 WorkShop for Windows Setup Overview	Provides specific guidelines in setting up and customizing the software.
PLC Configuration	Provides directions for configuring your processor to work with 505 WorkShop for Windows.
Programming	Gives a you solid understanding of the 505 WorkShop for Windows easy-to-use programming features.
Documentation	Shows you how to add description to your logic programs.
Alarms	Details the 505 WorkShop for Windows easy-to-use programming of Alarms features.
PID	Details the 505 WorkShop for Windows easy-to-use programming PID Loops features.
FTLogger/FTTrend	Shows you how to program the data Logger and Trender.
Appendices	Provides a quick reference to keyboard commands and troubleshooting guide.

Customer Support

From time to time you may find that this manual and the online help cannot answer your question. FasTrak Customer Support staff will gladly help you find the answers to your questions. To obtain assistance in the United States call:

FasTrak SoftWorks, Inc.	1-414-358-8088
Technical Support	1-414-358-8088
Technical Support Fax	1-414-358-8066

You can also send questions, comments and suggestions to:

Attention: Technical Service
FasTrak SoftWorks, Inc.
PO Box 240065
Milwaukee, WI 53224

or you can visit the FasTrak Home Page on the World Wide Web at:
http://www.fast-soft.com

For detailed information on Siemens 505 CPUs and Instructions set, refer to the following manuals:

Manual Name	Part Number
505 System Manual	PPX:505-8201-X
505 Programming Reference Manual	PPX:505-8204-X
ET200 Distributed I/O Manual	PPX:505-8206-X

Updating the Software License Key

Security of this WorkShop product is protected by use of a security key. This key may be a 25-pin connector that is provided when a new purchase is made of the product. The key may also be provided in the form of a USB (Universal Serial Bus) connector. For Site License Users, security is key-less in that it is built in to the product. For more information on site licenses, please contact your WorkShop distributor or sales representative.

For users with a security key, a license code is provided per each Maintenance Agreement by the manufacturer's of this product- FasTrak SoftWorks, Inc. Upon renewal of a Maintenance Agreement, a new license code may be obtained by contacting your distributor or sales representative.

505 WorkShop for Windows Overview

What is 505 WorkShop for Windows?

505 WorkShop for Windows is one of the world's most powerful and exciting programmable logic controller (PLC) programming software offerings. 505 WorkShop for Windows is the universal solution offering features that will save you time and money, such as:

- Symbolic Programming
- Cut, Copy, & Paste
- Instruction Toolbar and Mnemonics
- Multiple windows view and edit
- Flexible program setup
- Write, read, and force addresses from the Data Window
- Multiple documentation options
- Generous Online Help

Package Contents

Your 505 WorkShop for Windows package includes the items listed below. If any of these items are missing or damaged, please contact FasTrak's' Customer Service.

- One (1) 505 WorkShop with Com Profibus v5.0 software CD
- 505 WorkShop for Windows User's Guide (this manual)
- FasTrak-Key to attach to your computer's parallel port
- Com Profibus Manual
- 505 Programming Reference Manual

System Requirements

Hardware Requirements

To install 505 WorkShop for Windows on your computer, you need the following hardware:

- A personal computer with an Intel Pentium 100 processor or higher
- 32 Mb or more of RAM
- An 800 X 600 VGA monitor with at least 256 colors
- 100 Mb free disk space on your hard drive
- A mouse is recommended, but not required

505 WorkShop for Windows may not function properly on systems that are not 100% Intel compatible. Certain other hardware components and peripherals can create incompatibility problems. Please refer to Appendix B, *Troubleshooting Hardware Compatibility*, for a list of devices that cause compatibility problems.

Software Requirements

You also need the following software loaded on your computer before you install 505 WorkShop for Windows:

- Windows 95, 98, NT, 2000, or XP

Parallel Port Compatibility

Connecting the FasTrak-Key to your computer's parallel printer port or USB is required to use 505 WorkShop for Windows, except when in the demo mode. Procedures for installing the FasTrak-Key and 505 WorkShop for Windows are outlined in the next chapter. If your parallel port is not 100% IBM compatible, you may experience problems with the FasTrak-Key which will prevent 505 WorkShop for Windows from functioning properly.

In most cases, you can correct the parallel interface incompatibilities by replacing the parallel port with a 100% IBM compatible port or by adding a second printer adapter card.

NOTE FasTrak is not responsible for problems that result from using an incompatible parallel interface.

2 - Installation

Installing 505 WorkShop

Before you begin installation, you should review the *System Requirements* section in the Introduction chapter.

To install 505 WorkShop, turn your computer on and start Windows. You may need to provide a user name and password to log in to a computer network. If you are unsure, contact your company's System Administrator or IT representative. Follow these steps to install the software:

1. Insert the 505 WorkShop Disk in your computer's CD-ROM drive.
2. The CD should autostart, if not click the Windows **Start** button. Then Click **Run**, and type x:\setup.exe, where x is the letter for the CD-ROM drive.
3. Follow the instructions that appear on the screen.
4. After clicking Next, a 505 WorkShop Access Level Read/Load Only Install box appears. From the install window entitled 505 WorkShop Access Level, you can decide if you want Full-function, Read-only, or Load-only install. See *505 WorkShop Access Level*, below, for details.
5. After making your selection, click **Next**. Installation begins and a message appears telling you that the 505 WorkShop Installation Utility is loading.

505 WorkShop Access Level

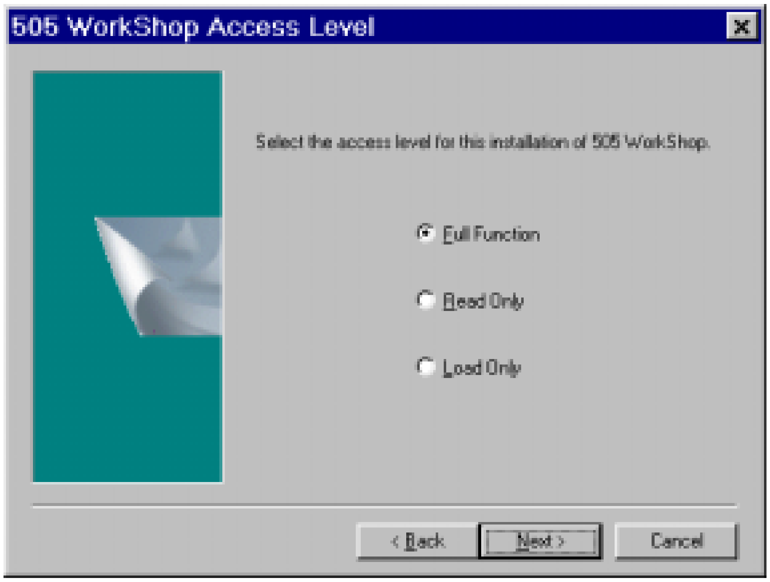


Figure 2.0

Full-function install allows full access to all features of 505 WorkShop.

Read-only install will not allow the user to access the following:

- **File Menu operations** - New, Open (Online), Save, Save As, Import, Export, and Merge.
- **Edit Menu** - Undo, Cut, Copy, Paste, Clear, Delete, Insert, Append, and Select All.
- **View Menu** - Ladder Editor, Special Function Editor, PID Loops, Alarms, Documentation Window, Symbol Library, and Data Window are all view only.
- **Program Menu** - New Network, Select Instruction, and Validate and Enter.
- **Configuration Editing** - Memory configuration editing, set PLC Scan Time, 505 I/O Edit, Profibus I/O Edit, Watchdog Timer Edit, and Time of Day edit.
- **Utilities Menu** - PLC Operations modes (PG/RUN etc.), Profibus Operations, Diagnostics on base, Task Codes per scan, Port Lockout, Password, Clear U Memory, and Clear Memory.

Load-only install will not allow the user to access the following:

- **File Menu operations** - New, Save, Save As, Import, Export, and Merge.
- **Edit Menu** - Undo, Cut, Copy, Paste, Clear, Delete, Insert, Append, and Select All.
- **View Menu** - Ladder Editor, Special Function Editor, PID Loops, Alarms, Documentation Window, Symbol Library, and Data Window are all view only.
- **Program Menu** - New Network, Select Instruction, and Validate and Enter.
- **Configuration Editing** - Memory configuration editing, set PLC Scan Time, 505 I/O Edit, Profibus I/O Edit, Watchdog Timer Edit, and Time of Day edit.
- **Utilities Menu** - Diagnostics on bases, Task Codes per scan, Port Lockout, Clear U Memory, and Password.

Installing the FasTrak-Key

What is the FasTrak-Key?

The 505 WorkShop software is copy protected with a device called the FasTrak-Key included in your shipment. To use 505 WorkShop online and save logic programs, connect the FasTrak-Key to a parallel printer port (LPT1-LPT3) on your computer.

NOTE Installation of the FasTrak-Key must be completed before you can run 505 WorkShop. If a FasTrak-Key is not detected, you can enter 505 WorkShop Demo mode only.

The FasTrak-Key will not interfere with normal port data transmissions, nor will it prevent you from creating backup copies of the software.

Connecting the FasTrak-Key

To install the FasTrak-Key, follow the steps below:

1. to which parallel port you want to connect the FasTrak-Key.
2. other security devices or cables attached to that port.
3. the FasTrak-Key to the port.
4. other cables to the FasTrak-Key, if necessary. If the device you attach to the FasTrak-Key is a parallel printer, make sure the printer is turned on before starting 505 WorkShop.

NOTE *The FasTrak-Key must be the first device attached to the parallel port. Other devices or cables can then attach to the FasTrak-Key.*

Troubleshooting the FasTrak-Key

The following is a list of error messages associated with the FasTrak-Key, possible causes, and ways to remedy the situation.

Message

"FasTrak-Key was not found." Check to see if the FasTrak-Key is attached to your parallel port and retry. The demo mode may be entered without a FasTrak-Key.

Possible Cause

1. WorkShop was started without the FasTrak-Key attached to the parallel port.
2. A parallel port driver may be missing.
3. If you are running Windows NT, the NT driver may not be loaded.

Remedy:

Check to see that:

- ❑ The FasTrak-Key is connected to a parallel port.
- ❑ The FasTrak-Key is the first device attached to the computer.
- ❑ That the parallel printer attached to one of the parallel printer ports is turned on.
- ❑ Add the port driver :

1. Go to the DOS prompt of 505 WorkShop directory.
2. Type RNBODRVS and press enter. This will create five subdirectories and extract three files (Readme.txt, Sentinel.ddp, and Setup.exe).
3. Type SETUP and press enter. The setup program automatically installs the proper drivers for your particular operating system.
   If there are no errors, the program immediately returns to the command line prompt. If there are errors, a report prints each error.
4. Restart the computer to activate the driver.

Message

While running 505 WorkShop, the FasTrak-Key was removed from the parallel port.

Remedy

Check to see that:

- ❑ The FasTrak-Key is connected to a parallel port.
- ❑ The FasTrak-Key is the first device attached to the computer.

Message

505 WorkShop for Simatic 505 is not supported by attached FasTrak-Key. Demo mode may be entered.

Possible Cause

The attached key is not authorized for use with Simatic 505.

Remedy

- Check to see that the correct FasTrak-Key has been attached to the parallel port.
- Call FasTrak Technical Support.

Message

Product version is not supported by FasTrak-Key. Call FasTrak to update FasTrak-Key.

Possible Cause

The key date does not support the current software version.

Remedy

- Call FasTrak Technical Support.

Message

FasTrak-KEY is corrupt. Demo mode may be entered.

Possible Cause

You purchased a partial software package, and the partial package information cannot be read from the FasTrak-Key.

Remedy

- Call FasTrak Technical Support.

Connect the Communications Cable

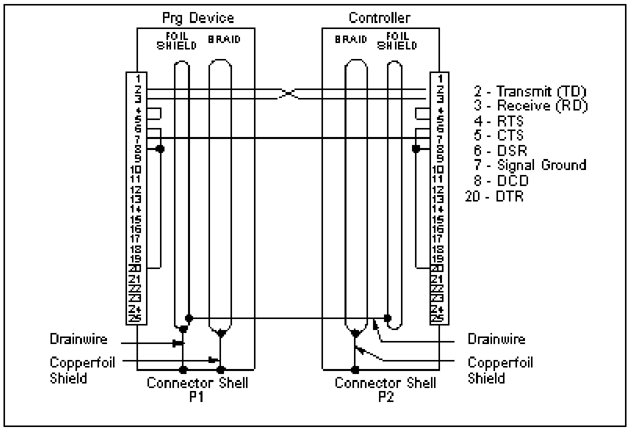
The PLC communications cable connects the serial COM port of the personal computer to the PLC. This enables you to transfer programs and data from the computer's memory (Offline) to the PLC, and vice versa. This cable has a 25-pin or 9-pin connector (computer end) and a 9-pin connector (PLC end).

If your computer has a 9-pin COM port, and you have a 25-pin cable, you can use a 9 to 25-pin converter to connect the communications cable to the processor. If your computer has a 25-pin COM port, this converter is not necessary.

The pin-outs for the communications cable are shown in the following figures.

Cable Pin-Outs for RS-232-C Ports

Connect your programming device to the controller with a double-shielded null modem RS-232-C cable. Figures 2.1, 2.2, and 2.3 show pin-outs for the RS-232-C cables.



2.1 RS-232-C 25-Pin to 25-Pin Figure Cable

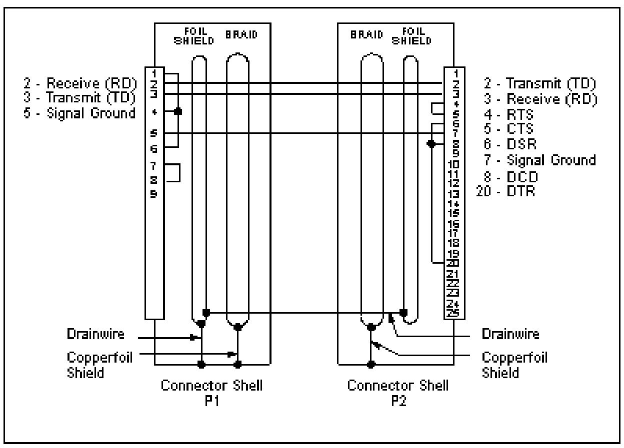


Figure 2.2 RS-232-C 9-Pin to 25-Pin Cable

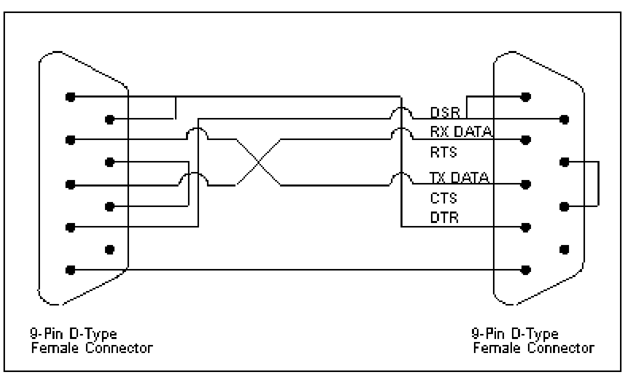


Figure 2.3 RS-232-C 9-Pin to 9-Pin Cable

Cable Pin-Outs for RS-422 Ports

If your controller has an RS-422 port, you can connect through the RS-422 port on your programming device. Use the pin-out values shown in Figure 2.4 for the RS-422 cable.

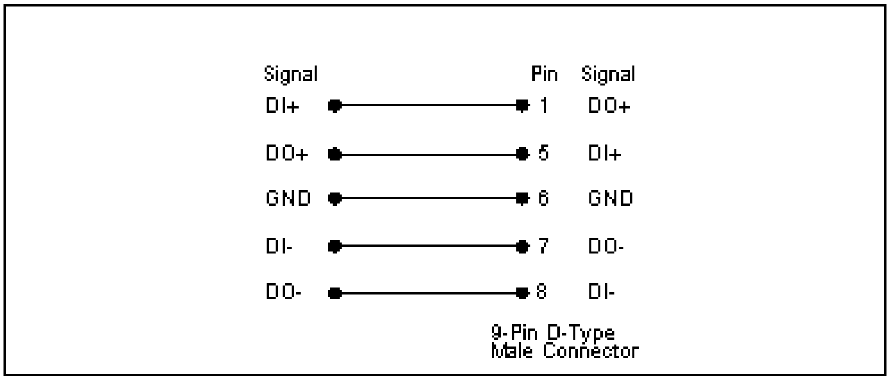


Figure 2.4 RS-422 Pin Values

Ethernet Connection

For Ethernet connection, you need a compatible Ethernet card in your computer; a Siemens PPX:505-cp2572 TCP/IP Ethernet card installed in your 505 system; and the appropriate cabling.

For more information on setting up your Ethernet connection, refer to the *Simatic Ethernet TCP/IP Communication Processor (505-CP2572) User Manual* order number PPX:505-8132-1.

3 - 505 WorkShop Basics

Welcome to 505 WorkShop

Software Features

If you are new to the Windows environment, we suggest reading the preceding chapter before continuing with this chapter.

This chapter will get you started using 505 WorkShop. Although you should progress at a rate comfortable for you, we recommend following the manual outline in your approach to programming. Advancing through the chapters in this order will help ensure a more efficient use of the software. The remaining chapters, in order, include:

- 505 WorkShop Basics
- 505 WorkShop Setup
- PLC Configuration
- Programming
- Documentation

Approaching the software in this order will also help you discover 505 WorkShop's powerful features that include:

- Multiple windows view and edit
- Flexible program setup
- Access to ladder logic through cross-reference
- Write, read, and force addresses from the Data Window
- Multiple documentation options
- Keyboard support for every function and command

Starting 505 WorkShop

After installing 505 WorkShop, start 505 WorkShop by doing one of the following actions:

- Double-click the 505 WorkShop icon with the left mouse button or
- Press Ctrl+F6 until the 505 WorkShop group box is active. If necessary, press the Tab key to highlight the 505 WorkShop icon and press Enter.

While 505 WorkShop loads, the 505 WorkShop copyright screen appears. When 505 WorkShop does not detect a FasTrak-Key, an error message appears.

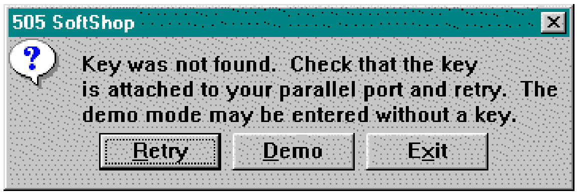


Figure 3.0

If you see this message, check to see that the FasTrak-Key is:

- Connected to a parallel printer port or
- Connected to a parallel port before printers or other devices
- See *Installing the FasTrak-Key* section in Chapter 2 - Installation for more information Installing 505 WorkShop

After you have checked the key's installation, load 505 WorkShop from the icon. If 505 WorkShop continues to not detect the *FasTrak*-Key, please call FasTrak Customer Support at 414-358-8088 immediately.

When 505 WorkShop has loaded completely, the 505 WorkShop window appears as described and defined in the next chapter.

The Landscape: 505 WorkShop Window

The 505 WorkShop window is the starting point for all your work. The key features of the window are designated with arrows on the sample illustrated below. Definitions of each feature are next, followed by more detailed information.

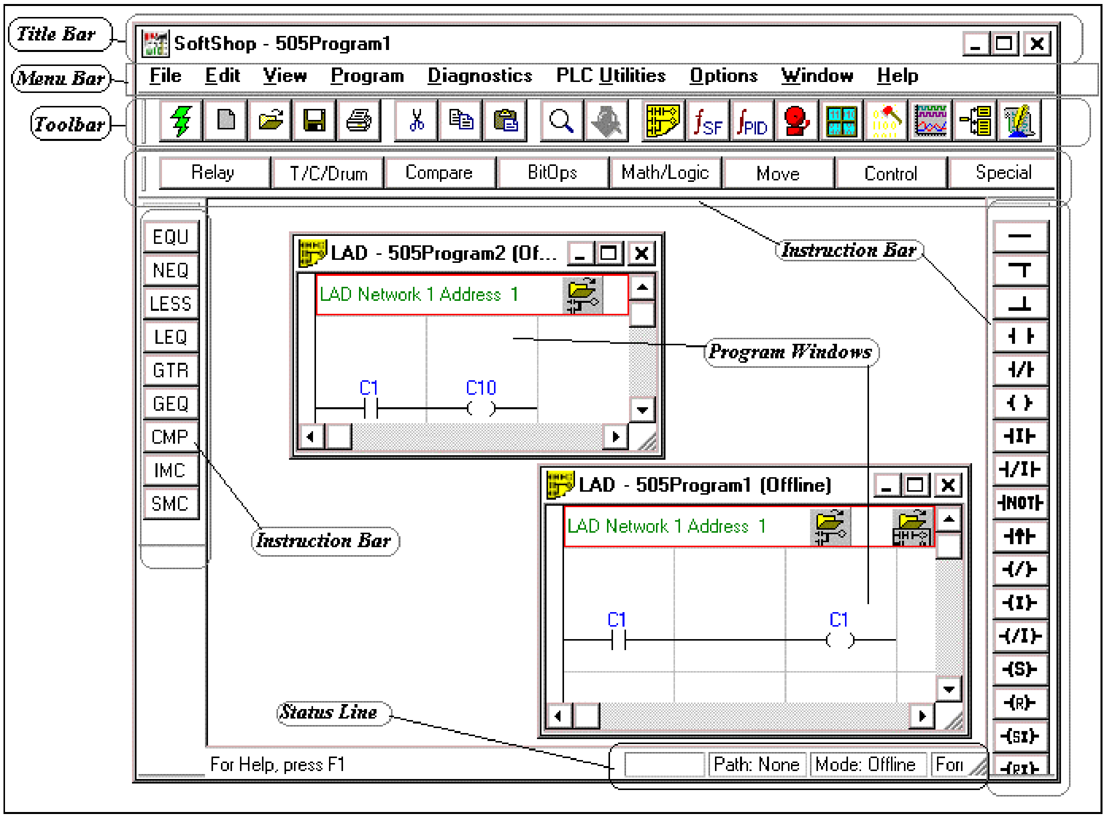


Figure 3.1

Window Description

The key features of the 505 WorkShop window shown in Figure 3.1 are defined below. More detailed descriptions follow.

Window Feature	**Function**
Instruction Bars	Use to add instructions, new rows, and new networks/addresses to a logic program. See *Instruction Bars.*
Menu Bar	Use to select 505 WorkShop functions. See *Menu Bar.*
Multiple Program Windows	Displays view and edit windows of multiple logic programs at the same time, limited only by the size of your computer's memory. See *Program Windows.*
Status Line	Displays information about the operation in progress. See *Status Line.*
Title Bar	Displays the name of the application. Buttons in the upper right corners change the window's size and position. See *Title Bar.*
Toolbar	Use to quickly access frequently used menu options. See *Toolbar.*

Instruction Bars

The instruction bars appear along the sides and the top of the 505 WorkShop window when you are programming ladder logic (Figure 3.1).

Instructions are divided into groups. To display the instructions for a group, click the group button. For example, to display the math instructions click the Math/Logic button. The instructions for that button appear and you can move the window anywhere on the screen. The Math group button remains selected until another group button is pressed.

Inserting an Instruction Bar Item in a Program

1. Click the desired instruction group button on the upper half of the Instruction Bar.
   Result: Instructions for that group appear on the lower half of the bar.
2. Click the button showing the item you want to insert in the program.
   Result: The item attaches to the pointer when you move to the ladder editing area.
3. Move the pointer to the item insertion point on the ladder editing area and click the left mouse button.
   Result: The instruction is dropped into place.
4. Repeat Step 3 each time you want to add the same item.
5. Click on the arrow button in the middle of the Instruction Bar to return the pointer to an arrow.

Menu Bar

The menu bar, located just below the title bar, identifies the names of the available 505 WorkShop functions. To display the menu options for each function, click on the function name. The menu options displayed may change depending upon the operation in progress.

Select File, New Program or Open Program and the following Menu Bar appears (Figure 3.2).

Figure 3.2

The Restore button appears on the Menu Bar when a program window is maximized. Use the Restore button to control the size of the program window.

Multiple Program Windows

505 WorkShop displays more than one logic program window at a time. You can open as many logic program windows as your computer's memory permits.

For example, you may wish to copy part of a logic program to another program. This saves you programming time by not having to retype similar logic statements for each program. To copy a part of one logic program to another, use the following procedure:

1. Open both programs offline.
2. Arrange the logic program windows by selecting Tile or Cascade from the Window menu.
3. Highlight the data you want to copy to the other program by clicking the mouse and dragging across the data.
4. Click Copy from the toolbar or from the Edit menu.
5. Move the pointer to the place you want to insert the data in the other program. Click Paste from the toolbar or from the Edit menu.

Status Line

The Status Line spans the bottom of the 505 WorkShop window. It displays information or describes the current operation on the left side. To display status line information for a menu item or button, click on the item or button and hold the left mouse button.

The right side of the Status Line displays the information listed below.

Status line	Function
PLC Path	The first shadowed box on the status line displays the route defined for the active program. If the PLC route is a direct serial connection to a PLC, then Direct is displayed. The online PLC route is taken directly from the controller from where you are connected. The offline PLC route informs you as to the parameter saved in the PLC setup window for this particular program.
Logic Mode	Indicates whether you are programming online or offline.
Num Lock	Indicates if Num Lock is active for the keyboard When using the keypad, which is usually on the right side of the keyboard, numbers are active. Num Lock is not active when indicator area is blank.
Character Type	Indicates if Caps Lock is active for the keyboard. When typing, capital letters appear if Caps is indicated. Caps Lock is not active when indicator area is blank.

Figure 3.3

Title Bar

The Title Bar spans the top of the 505 WorkShop window (Figure 3.4).

Use the Title Bar to:

- Identify the application you are using. In Figure 3.1, the application is WorkShop.
- Move the window. Click the title bar with the mouse pointer, hold down the left mouse button and drag to the desired location to move the window.
- Change the size or position of the window. The following buttons appear in the corner of the title bar:

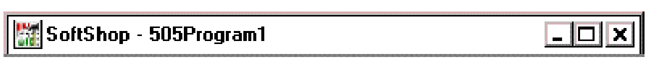


Figure 3.4

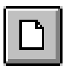
Button Title	Location	Function
Minimize	Left box	Click the dash button to reduce window to an icon.
Maximize	Middle box	Click the window button to enlarge the entire screen.
Close	Right box	Click the **X** button to exit WorkShop.

Toolbar

The Toolbar displays a row of buttons. Each button represents a frequently used option. Select the option by clicking on its button, saving you the steps of selecting several options from a series of menus. Notice that when you click on the button, its purpose appears on the Status Line at the bottom of the window (Figure 3.5).

Figure 3.5

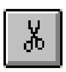
 Fast PLC Connection.

 Create a new program. See *Creating New Logic Programs.*

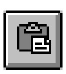
 Open an existing program. See *Opening Logic Programs.*

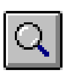
 Save the active program. See *Saving Logic Programs.*

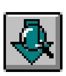
 Print. See *Printing Logic Programs.*

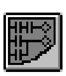
 Cut the highlighted section to the clipboard. See *Editing Logic Programs.*

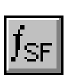
 Copy the highlighted section to the clipboard. See *Editing Logic Programs.*

 Paste a section from the clipboard. See *Editing Logic Programs.*

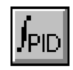
 Find a network, address, or tag.

 Find the next occurrence of the found address or tag.

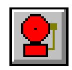
 Display Ladder Editor.

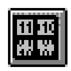


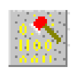
Display Special Functions Editor.

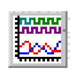


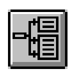
Display PID Loop Directory.

 Display Analog Alarm Directory.

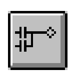
 Display the Data Window.

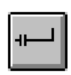
 Display FTLogger (log data.)

 Display FTTrender

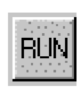
 Display the Cross Reference.

 Display the Documentation Window.

 Add a new network to a program.

 Add a new row to a program.

Puts the PLC in program mode



Puts the PLC in run mode

 Enter and validate the active program.

Working With Logic Programs

505 WorkShop provides you with a number of different ways of editing programs. These include:

- ❑ Connecting to a PLC to view logic stored in the PLC
- ❑ Creating a new program offline
- ❑ Loading a file online
- ❑ Loading a file offline
- ❑ Attaching documentation to an online program

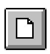
Before you can perform any editing, you must first open a logic program.

Creating a New Logic Program

With 505 WorkShop, it's easy to create a new program.

***NOTE**: Remember, you can create new programs in offline mode only. To connect to a processor online, use Open Program or Fast PLC Connection.*

To create a new logic program and begin programming, do the following:

1. Click  on the Toolbar or select New Program from the File menu; or
1. Press Ctrl+N on the keyboard.
2. Select the PLC Type.
3. Press OK.

Fast PLC Connection - Connecting to a PLC

With 505 WorkShop, you can connect to a PLC with a click of the mouse. You can then view and edit existing logic in the PLC.

Prior to connecting the first time, you need to set up the PLC connection. Refer to Chapter 4, *Fast PLC Setup*, for more details.

***NOTE** Remember, you cannot load a file with Fast PLC Connection. To load a file Online, use Open Program.*

To connect to a PLC online click  on the Toolbar or select Fast PLC Connection from the File menu. You are set to begin programming.

***NOTE** If your controller is a 575, refer to* Connecting Online to a 575 *in this chapter for information on selecting Application ID and configuring ports.*

Open an Existing Logic Program Offline or Online

You can open an existing logic file to edit or update program information in either online or offline mode. Logic programs may contain one or more of the following: logic and data, tags, headers, and descriptions and comments. Several programs may be open at one time without losing memory contents.

Warning	**Editing or modifying a program online may produce unexpected or hazardous results.**

To open an existing program:

4. Click  on the Toolbar or select Open Program from the File menu (Ctrl+O).
 Result: The Open Program dialog box appears (Figure 3.6).

NOTE *The last four files that were opened are saved and listed at the bottom of the File menu. When you select one of these files, the Open Program dialog box automatically opens with the file you selected.*

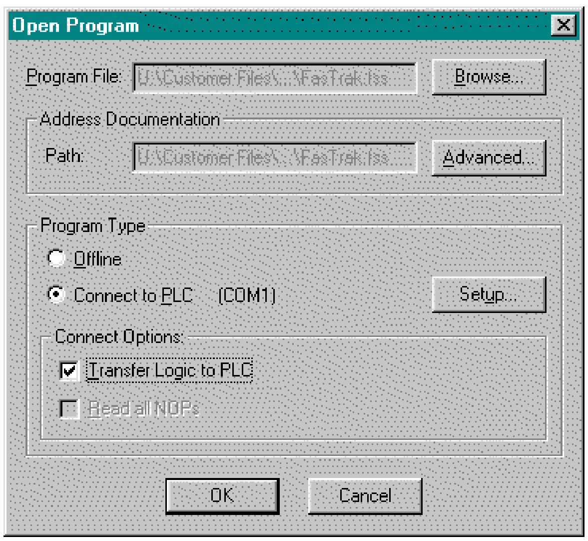


Figure 3.6

5. Click Browse.
 Result: The Open dialog box appears.

6. Type the program name in the Program Name line. Change drives and/or directories, if necessary, to find the program you wish to open. You can open *.FSS (505 WorkShop 32 bit files), *.FTK (505 WorkShop 16 bit files) or *.VP5 (TISOFT V5.0 files).

NOTE: *For TISOFT V5.0 files (*.vp5) loads, a new offline program is created with the logic and documentation always imported. Synonyms for JMP, GTS, and SBY are not imported. However, their associated comments are imported as Headers.*

7. Click on the program name in the File Name scroll box and click OK.
 Result: The name of the program appears in the Program Name line of the Open Program dialog box.

The selected program's file name is put into the Address Documentation Path. If a different documentation program is desired, it must be linked via Share Address Documentation. When a file is configured you can specify a database file that holds and sorts all documentation. Multiple users can simultaneously modify documentation for the same file thus regular updates can be scheduled to get the latest documentation within the database. Documentation can be imported from *.FSS (505 WorkShop 32 bit files), comma or tab separated. Conversely you can export from the database file into a text file or a *.VP5 file (TISOFT V5.0 files with Headers and Tags).

8. To set up shared documentation, click the Advanced button on the Open Program dialog box.
   Result: The Share Address Documentation dialog box appears.
9. Enter the share address documentation program name in the Shared File and Path line or select Browse to locate an existing file.
10. Click Next and enter the refresh rate (time lag between updates from other users of the database). Valid times are from 1 to 1440 minutes.
11. Click Next and Finish.
12. In the Program Type area, select Offline or Connect to PLC. If you select Connect to PLC, you can use the previously saved communication port or select a new communication port by clicking the Setup button. If you select Connect to PLC, you can check other options such as:
    - Logic to PLC. Transfers all blocks and data areas to the PLC, and loads documentation.

      NOTE: *To avoid overwriting the online ladder with the version stored on file and associating documentation with the online ladder; do not select Transfer Logic to PLC.*
    - Read All NOPs. Reads 30 consecutive NOP instructions and assumes the end of program logic. This can substantially increase the time to connect online to a 520, 525, 530, 535, 560, or 565 controller when selected.

      NOTE: *If you select Read All NOPS and have more than 30 NOP instructions separating user logic, all logic past the 30 NOPS will not be read.*
13. Click OK or press Enter to open your program.

NOTE*:* *If your controller is a 575, refer to* Connecting Online to a 575 *in this chapter for information on selecting Application ID and configuring ports.*

Address/Network Mode

1. If there is no file transferred online when the program first opens then the ladder rungs are referenced in the old TISOFT Address Mode. When the complete program has been load the program can be converted over to network mode. To accomplished this select Switch to Network Display from the Options Menu. The rungs are now displayed in Network and Address Mode. Once rungs have been converted Network/Address mode they cannot be converted back to Address Mode only.

Loading Parts of an Existing Logic Program Online

505 WorkShop allows you to load parts of an existing logic file to an online 505 controller. One or more of the following parts can be loaded:

- Ladder
- Special Function Programs
- Special Function Subroutines
- PID Loops
- Analog Alarms
- Force States
- Documentation (Tags, Descriptions, Comments and Headers
- I/O Configuration
- Profibus
- V-Variables
- K-Constants
- Word I/O
- U-Memory

Warning	**Editing or modifying a program online may produce unexpected or hazardous results.**

To Load parts of an existing program you must first established an online connection with the PLC (*see Fast PLC Connection - Connecting to a PLC)*. Then:

1. Select Load By Parts from the File menu (Ctrl+B).
   Result: The Load By Parts dialog box appears (Figure 3.6a).

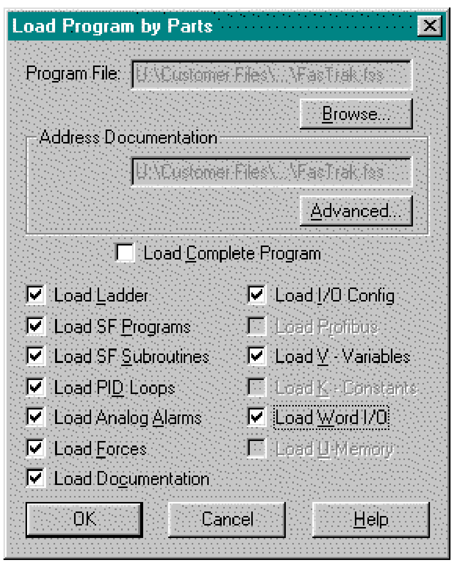


Figure 3.6a

2. In the Load Program by Parts dialog box select the parts of the program you wish to load or select Load Complete Program to load all parts.
3. 13. To enter a different file name Click Browse.
   : The Open dialog box appears.
4. Type the program name in the Program Name line. Change drives and/or directories, if necessary, to find the program you wish to open. You can only open *.FSS (505 WorkShop 32 bit files) programs.
5. Click on the program name in the File Name scroll box and click OK.
   Result: The name of the program appears in the Program Name line of the Open Program dialog box.

The selected program's file name is put into the Address Documentation Path. If a different documentation program is desired, it must be linked via Share Address Documentation. When a file is configured you can specify a database file that holds and sorts all documentation. Multiple users can simultaneously modify documentation for the same file thus regular updates can be scheduled to get the latest documentation within the database. Documentation can be imported from *.FSS (505 WorkShop 32 bit files), comma or tab separated. Conversely you can export from the database file into a text file or a *.VP5 file (TISOFT V5.0 files with Headers and Tags).

6. To set up shared documentation, click the Advanced button on the Open Program dialog box.
   Result: The Share Address Documentation dialog box appears.
7. Enter the share address documentation program name in the Shared File and Path line or select Browse to locate an existing file.
8. Click Next and enter the refresh rate (time lag between updates from other users of the database). Valid times are from 1 to 1440 minutes.
9. Click Next and Finish.
10. Click OK or press Enter to load.

NOTE *If your controller is a 575, refer to* Connecting Online to a 575 *in this chapter for information on selecting Application ID and configuring ports.*

Transfer Offline Program to Online

505 WorkShop allows you to transfer an existing offline logic file to an online 505 controller.

Warning	**Editing or modifying a program online may produce unexpected or hazardous results.**

To transfer an existing offline program to online:

1. Select **Transfer → Online** from the File menu (Ctrl+B).
   Result: The Open Program dialog box appears (Figure 3.6).
2. Select **OK** to Transfer.

Connecting Online to a 575

When connecting to a 575 controller either from Fast PLC Connect or the Open Program window, configure the Application ID and Port Settings before the online connection is made (Figure 3.7).

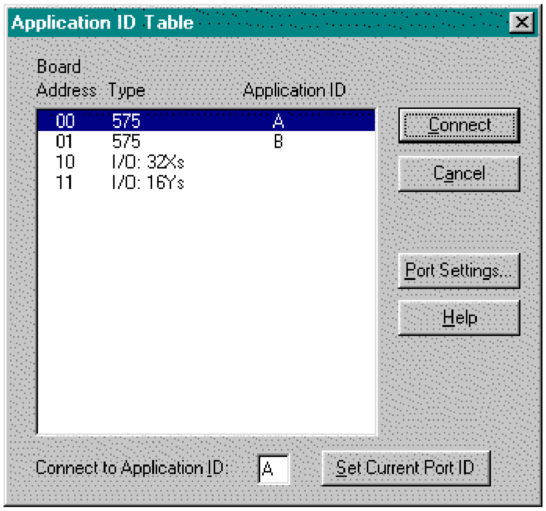


Figure 3.7

To connect to a 575 online:

1. Type the letter (A to Z) of the application you want to connect to in the Connect ID field.
2. Click on the Set Current Port ID and the Connect button to go online, or double-click on the highlighted Application ID in the list box to go online.

If a file is being loaded from disk to the controller and the disk file Application ID is different than the connecting PLC's ID; a warning appears describing that the loading file ID is different. Select OK to continue the load or Cancel to abort.

Reading or Writing 575 Port Configuration

To read or write the 575 port configuration, click on the Port Settings button in the Application ID Table dialog box. The Port Settings Dialog box appears (Figure 3.8).

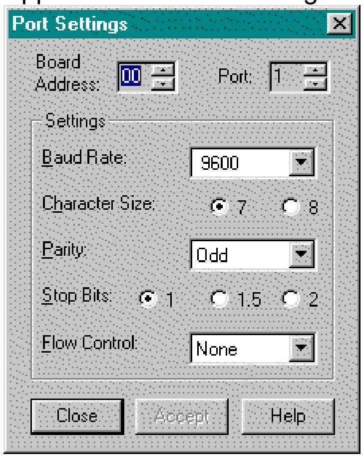


Figure 3.8

To make changes to the port settings:

1. Click on the item to be changed and either type in the change or click on the spin, radio button, or combo box.
2. Click on Accept to send changes to the controller, or Cancel to make no changes and return to the Applications ID Table dialog box.

NOTE *If the port you are changing is the port 505 WorkShop is connected to, but the settings are not what 505 WorkShop uses, when you press Accept the following prompt appears to confirm your action:*

505 WORKSHOP IS ATTACHED TO THE PROCESSOR THROUGH THIS PORT. THERE MAY BE A SHORT DELAY AS THE PORT IS RECONNECTED. DO YOU WANT TO CHANGE THESE PORT SETTINGS?

*Select **YES** to change the port configuration and 505 WorkShop may or may not continue to communicate with the controller.*
*Select **NO** to abort the operation.*

Importing Documentation

Use the following procedure to import documentation.

1. Type the program name in the File name field, or click Browse to search from valid program names (Figure 3.9).
2. Click OK to import the program.

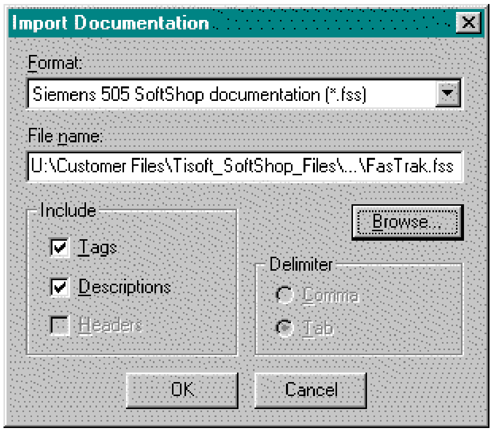


Figure 3.9

Use the Tag/Description Text File to merge Tags or Descriptions from the ASCII text files exported by WorkShop for Windows into the currently opened program. The default file extension is *.TXT but alternate extensions are allowed. You must also select comma or tab delimited records with the following format:

- Characters 1 - 6 (could be greater with Loops or Alarms) contain the 6 digit Siemens address.
- Character 7 is the comma or tab character.
- Character 8 and beyond is the variable length tag followed by the comma or tab character, followed by the variable length description followed by a carriage return.

Exporting Documentation

Use the following procedure to export documentation.

1. Select Export from the file menu.
2. Select a file format to export the documentation to (Figure 3.9a).
3. Type the program name in the File name field, or click Browse to search from valid program names.
4. Select the parts of documentation to export (Tags, Descriptions or Headers). Then determine which delimiter format to use. The delimiter is used to separate the types of documentation (Tags, Descriptions or Headers). These selections are not used for TISOFT.
5. Click OK to export the program.

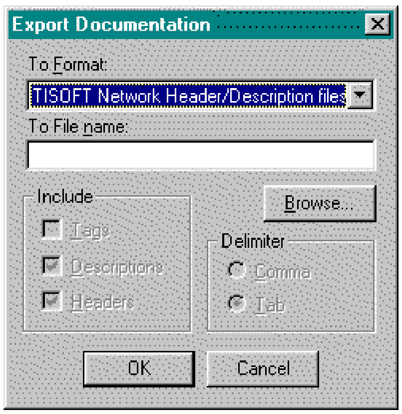


Figure 3.9

Use the Tag/Description Text File to merge Tags or Descriptions from the ASCII text files exported by WorkShop for Windows into the currently opened program. The default file extension is *.TXT but alternate extensions are allowed. You must also select comma or tab delimited records with the following format:

- Characters 1 - 6 (could be greater with Loops or Alarms) contain the 6 digit Siemens address.
- Character 7 is the comma or tab character.
- Character 8 and beyond is the variable length tag followed by the comma or tab character, followed by the variable length description followed by a carriage return.

Saving Logic Programs

Saving Offline

Use the Save Program command to save the active program contents with its existing name. To save the active logic program:

1. Click on the Toolbar or select Save Program from the File menu (Ctrl+S), and a previously saved logic program is saved.
2. If the program has not been previously saved, the Save As dialog box appears (Figure 3.10). Select the desired location to save the program, and type in a name for the program in the File name field.

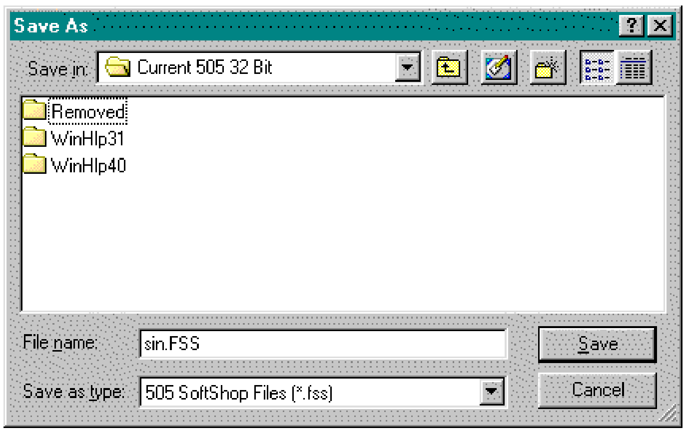


Figure 3.10

3. 3. Click Save or press Enter to save the program.
4. If you select a file name that already exists in that directory, a message appears with options. Select from the following options:

- ❑ YES saves the updated program with the current name, overwriting the previous version.
- ❑ NO cancels the save procedure.

NOTE *You must validate logic before saving. If you have not done so, a message appears stating changes to logic have not been validated or entered. Changes cannot be saved until logic has been validated and entered. Click Validate Logic on the toolbar, or select Validate and Enter Logic from the Program menu. Complete necessary changes to logic and try to save the program again.*

Save/Save As Online

Use the online Save program command to save all the active program contents with its existing name or parts of an existing logic program. One or more of the following parts can be saved:

- Ladder
- Special Function Programs
- Special Function Subroutines
- PID Loops
- Analog Alarms
- Force States
- Documentation (Tags, Descriptions, Comments and Headers)
- I/O Configuration
- Profibus
- V-Variables
- K-Constants
- Word I/O
- U-Memory

To save the active logic program:

1. Click on the Toolbar or select Save Program from the File menu (Ctrl+S).
   Result: The Save dialog box appears (Figure 3.11).

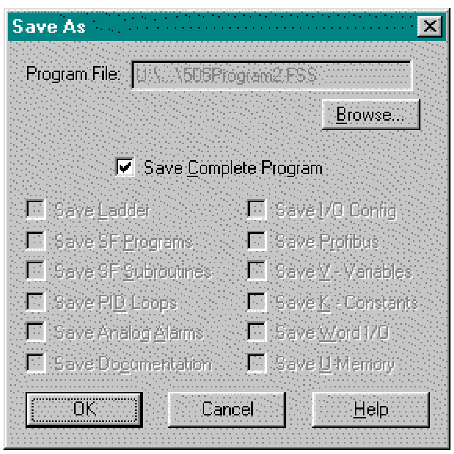


Figure 3.11

2. In the Save select the parts of the program you wish to load or select Load Complete Program to load all parts.
3. If the program has not been previously saved or you wish to change the file name, select the Browse button, and type in a name for the program in the File name field.
4. Click Save or press Enter to save the program.
5. If you select a file name that already exists in that directory, a message appears with options. Select from the following options:
   - **YES** saves the updated program with the current name, overwriting the previous version.
   - **NO** cancels the save procedure.

NOTE *You must validate logic before saving. If you have not done so, a message appears stating changes to logic have not been validated or entered. Changes cannot be saved until logic has been validated and entered. Click Validate Logic on the toolbar, or select Validate and Enter Logic from the Program menu. Complete necessary changes to logic and try to save the program again.*

If you are working online and saving your changes, a warning message appears stating that logic in the active PLC program has been modified, and asks if you wish to save the logic in addition to tags and documentation.

Click Yes to save logic, tags, and documentation. Click No to save documentation and tags only. Cancel aborts the save procedure.

If you are saving documentation to an online logic program and the last network header exceeds the number of networks/addresses, an error message appears.

Click Yes to discard the extra network/address headers, or No to cancel the save procedure.

Similarly, if you attempt to save a logic program and the number of networks/addresses exceeds the saved headers, an error message appears.

Click Yes to continue saving the program. Click No to cancel the save procedure.

Warning	**Editing or modifying a program online may produce unexpected or hazardous results.**

Save Program As Offline

Use Save Program As to save the active logic program with a different program name. This is useful when maintaining the original without changes. For example, open file ABC.FTK, make changes, select Save Program As, and save the program as DEF.FTK. Now you have two files, ABC.FTK retained its same condition before you opened it, and DEF.FTK that contains changes made to ABC.FTK.

NOTE *You must validate logic before saving. If you have not done so, a message appears stating changes to logic have not been validated or entered. Changes cannot be saved until logic has been validated and entered.*
Click Validate Logic on the toolbar, or select Validate and Enter Logic from the Program menu. Complete necessary changes to logic and try to save the program again.

To save a logic program with a new file name:

1. Select Save As from the File menu.
   Result: The Save As dialog box appears (Figure 3.10).
2. In the Save As dialog box, click options in the necessary group boxes to designate a location to save the logic program.
3. Click OK or press Enter.
   Result: Your file is saved with its new name.
4. If you select a file name that already exists in that directory, a message appears with options. Select from the following options:
   - YES saves the updated program with the current name, overwriting the previous version.
   - NO cancels the save procedure.

NOTE *If you are saving documentation to an online program, please see* Saving Online *under Saving Logic Programs on the preceding pages.*

Printing Logic Programs

505 WorkShop provides you with a number of print features that allow customization of your printouts. These include:

- Tags and Documentation
- Cross Reference
- Network/Address Range
- Margins
- Starting Page Number

Before you can print, open a logic program. Make certain that you have loaded the correct print drivers for your printer through the Windows Control Panel. If you have questions regarding loading print drivers, consult your printer's user's manual and the *Windows User's Guide*.

To print logic programs and/or documentation:

1. Click  on the toolbar or select Print from the File menu (Ctrl+P).
   Result: The Print dialog box appears (Figure 3.12).

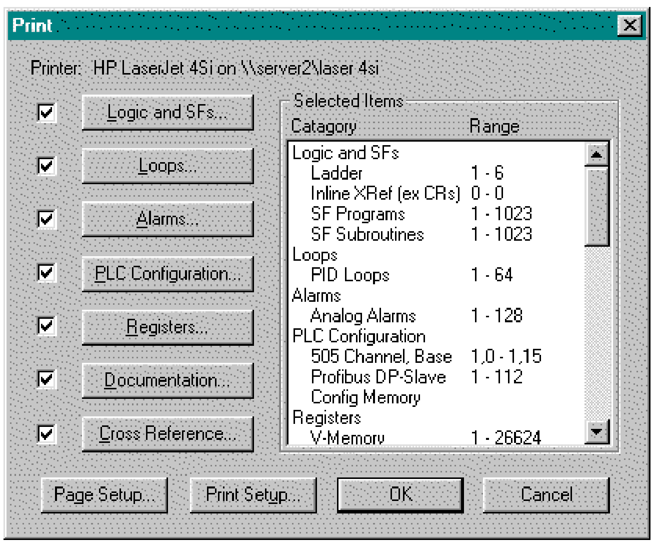


Figure 3.12

2. Click on the check boxes that correspond to the items you want to print. For each item selected, you can choose sort options and the information you want to include for reports. The Selected Items window displays the print range of each print item to be printed.

Print Logic and SFs

Click the Logic and SFs button to print the following:

- All logic, ladder, SFS and SFP
- Selected logic ladder, SFS or SFP
- Selected ladder in address or network mode
- Selected Special Function lines
- Inline Xref
- Ladder with Addresses, Tags, Descriptions, or Headers
- Multiple or single networks/addresses per page

The Print Logic and SFs dialog box determines which items are printed. (Figure 3.13).

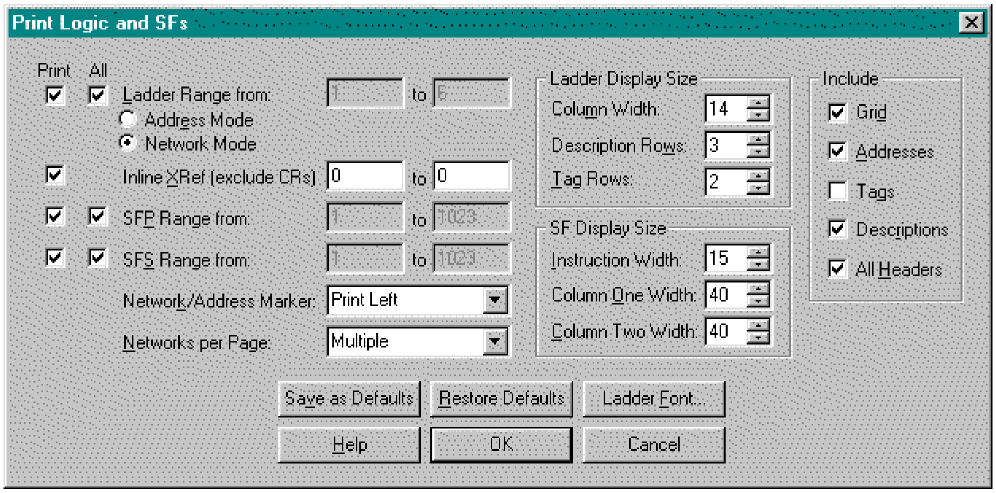


Figure 3.13

Click the All check box to print the entire range of the item selected. To print a selection, deselect the All check box and enter a range in the associated edit box.

If you selected ladder to print, the ladder reference numbering can be either PLC memory address mode or network address mode. The selected radio button below ladder range determines which mode is printed. The Network address or PLC Memory Address is printed on the top left or right of ladder and can be disabled to not print at all. The selection is made in the Network/Address Marker field.

When selecting Networks/Addresses per Page as Single or Multiple, use the following information. When printing Single Networks/addresses per Page, each network begins on a new page. When printing Multiple Networks/Addresses per Page, as many networks/addresses that can fit on a single page will be printed. However, if the network is not the first network on the page and the network is broken across more than one page but can fit on a page and if it would start a new page, then the network begins on a new page. The intent is to keep the entire network on one page whenever possible.

The Ladder Display Size determines the Column width, Description rows, and Tag row size. Column Width adjusts the size of the ladder grid. Click the up or down arrow or enter a value between 7 and 24. Description Rows determines the number of character rows displayed for each description. Click the up or down arrow or type in a value between 1 and 12. Tag Rows determines the number of character rows displayed for each tag. Click the up or down arrow or type in a value between 1 and 4.

The SF Display Size determines the Instruction width, Column One width and Column Two width. Instruction Width determines the width allocated for display of Special Function instructions on each line. Click the up or down arrow or type in a value between 1 and 80. Column one determines the width allocated for display of SF first column variables after the instruction. Click the up or down arrow or type in a value between 1 and 300. Column Two determines the width allocated for display of SF second column variables after the instruction. Click the up or down arrow or type in a value between 1 and 300.

The ladder grid, addresses, tags, descriptions and headers can all individually be turned on or off to be included with ladder printout. To include with the ladder print out, select the appropriate check box under Include.

Ladder Font changes the font displayed in the active program. Any active Window's font can be selected. To change the font:

1. Click on the Ladder Font button.
   Result: The Font dialog box is displayed.
2. Choose a font, font style, and font size. Notice that you can see a sample of the font in the Sample box.
3. Click OK to save your changes or Cancel to cancel your changes.
   Result: The Print Logic and SFs dialog box appears.

Print Loops

Click the Loops check box from the Print dialog box to print the following:

- All PID Loops
- PID Loops and Headers

To change the PID Loops properties click the Loops button on the Print dialog box. The Print Loops dialog box appears (Figure 3.14).

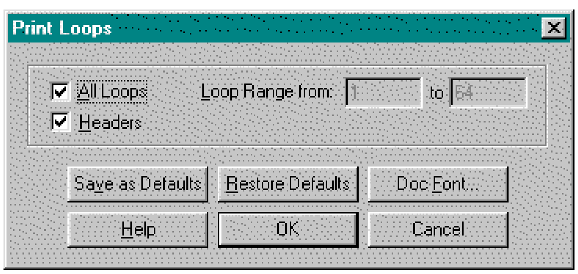


Figure 3.14

Click the All Loops check box to print the entire range loops. To print a selection, deselect the All check box and enter a range in the associated edit box. To include loop headers with the printout, click the Headers check box.

1. Click the Doc Font button to change the Header font in the active program.
   Result: The Font dialog box appears.
2. Choose a new font, font style, or font size. Notice that you can see a sample of the font in the Sample box.
3. Click OK to save your changes or Cancel to cancel changes.
   Result: The Print Loops dialog box appears.

Print Alarms

Click the Alarms check box from the Print dialog box to print the following:

- All Alarms
- Alarms and Headers

To change the Alarms properties, click the Alarms button on the Print dialog box. The Print Alarms dialog box appears (Figure 3.15).

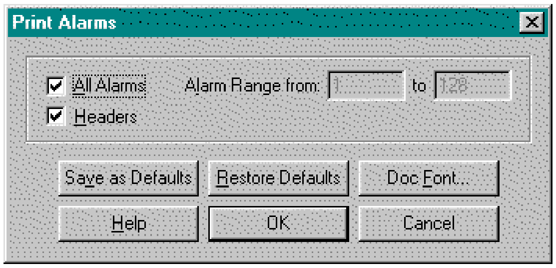


Figure 3.15

Click the All Alarms check box to print the entire range of alarms. To print a selection, deselect the All check box and enter a range in the associated edit box. To include loop headers with the printout, click the Headers check box.

1. Click the Doc Font button to change the Header font in the active program.
   Result: The Font dialog box appears.
2. Choose a font size. Notice that you can see a sample of the font in the Sample box.
3. Click OK to save your changes or Cancel to cancel changes.
   Result: The All Alarms dialog box appears.

Print PLC Configuration

Click the PLC Configuration check box from the Print dialog to print the following:

- 505 Channel Base
- Profibus DP – Slave I/O
- PLC Memory Configuration
- I/O Tags and Descriptions

To change the PLC Configuration properties click the PLC Configuration button on the Print dialog box. The Print PLC Configuration dialog box appears (Figure 3.16).

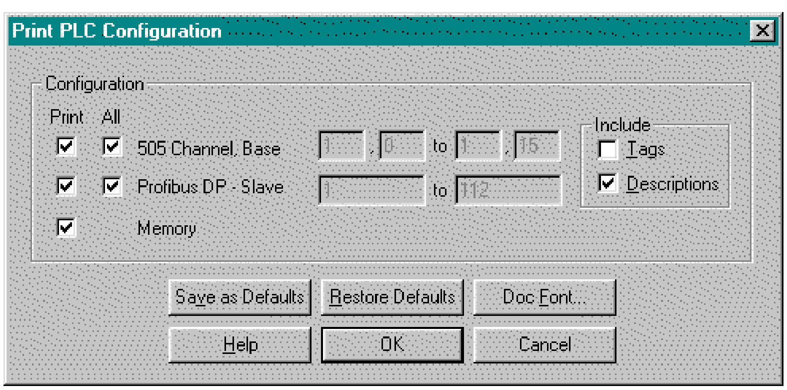


Figure 3.16

Click the All check box to print the entire range of items. To print a selection, deselect the All check box and enter a range in the associated edit box. To include loop headers with the printout, click the Headers check box.

When entering a range for channel and base numbers, the first entry before the comma is the channel number. The second entry after the channel number and comma is the base number. For Example: In Figure 3.16 the printout would start at channel 1 Base 0 and end at channel 1 Base 15.

The tags and descriptions can be individually turned on or off. To include with the PLC Configuration printout, select the appropriate check box under Include.

1. Click the Doc Font button to change the Tag and Description printout font in the active program.
   Result: The Font dialog box appears.
2. Choose a new font, font style, or font size. Notice that you can see a sample of the font in the Sample box.
3. Click OK to save your changes.
   Result: The Print Loops dialog box appears.
4. Click Cancel to make no font changes.
   Result: The Print PLC Configuration dialog box appears.

Print Registers

Click the Register check box from the Print dialog box to print the following:

- V-Memory
- K-Memory
- Register Tags and Descriptions

To change the Register properties click the Register button on the Print dialog box. The Print Register dialog box appears (Figure 3.17).

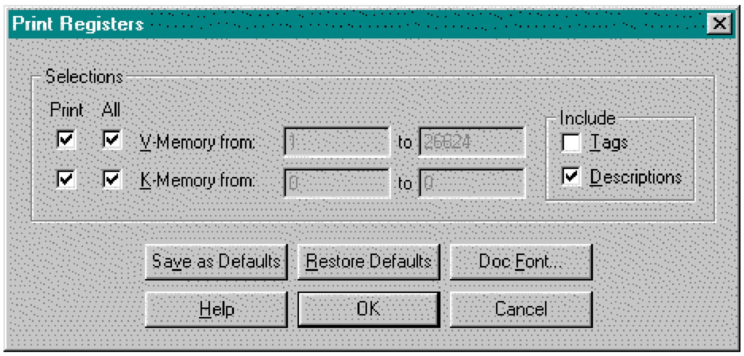


Figure 3.17

Click the All check box to print the entire range of items. To print a selection, deselect the All check box and enter a range in the associated edit box.

The tags and descriptions can be individually turned on or. To include with the Register print out select the appropriate check box under Include.

1. Click on the Doc Font button to change the Tag and Description printout font of the active program.
   Result: The Font dialog box appears.
2. Choose a new font, font style, or font size. Notice that you can see a sample of the font in the Sample box.
3. Click OK in the Font dialog box to save your changes.
   Result: The Print Loops dialog box appears.
4. Click Cancel to make no font changes.
   Result: The Print Register dialog box appears.

Print Documentation

Click the Documentation check box from the Print dialog box to print the following:

- I/O Elements (X, Y, WX and WY)
- Control Relays (C)
- SKP, LBL, GTS, SBR, JMP, MCR and END
- Register Tags, Descriptions and Comments

To change the Documentation properties click the Documentation button on the Print dialog box. The Print Documentation dialog box appears (Figure 3.18).

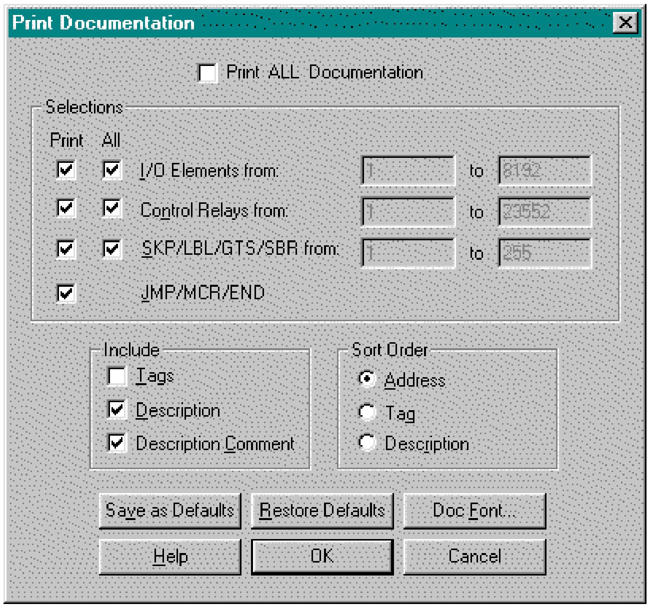


Figure 3.18

Click the All check box to print the entire range of items. To print a selection, deselect the All check box and enter a range in the associated edit box.

The Register Tags, Descriptions and Comments can be individually turned on and off. To include with the Documentation print out select the appropriate check box under Include.

The sort order for documentation print can be based on address, Tag, or Description. To change the Documentation print sort orders select the appropriate check box under Sort Order.

1. Click on the Doc Font button to change the Tag, Description, and Comments printout font of the active program.
   Result: The Font dialog box appears.
2. Choose a new font, font style, or font size. Notice that you can see a sample of the font in the Sample box.
3. Click OK in the Font dialog box to save your changes or Cancel to cancel your changes.
   Result: The Print Documentation dialog box appears.

Print Cross Reference

Click the Cross-reference check box from the Print dialog box to print the following.

- All logic, ladder, SFS and SFP
- Selected logic ladder, SFS or
- Selected ladder in address or mode
- Selected Special Function lines
- Alarms
- Loops
- I/O Elements (X, Y, WX and)
- Control Relays (C)
- V-Memory
- K-Memory
- I/O Elements (X, Y, WX and)
- Control Relays (C)
- Tags Descriptions and Comments

To change the Cross-reference properties click the Cross-reference button on the Print dialog box. The Print Cross-reference dialog box appears (Figure 3.19).

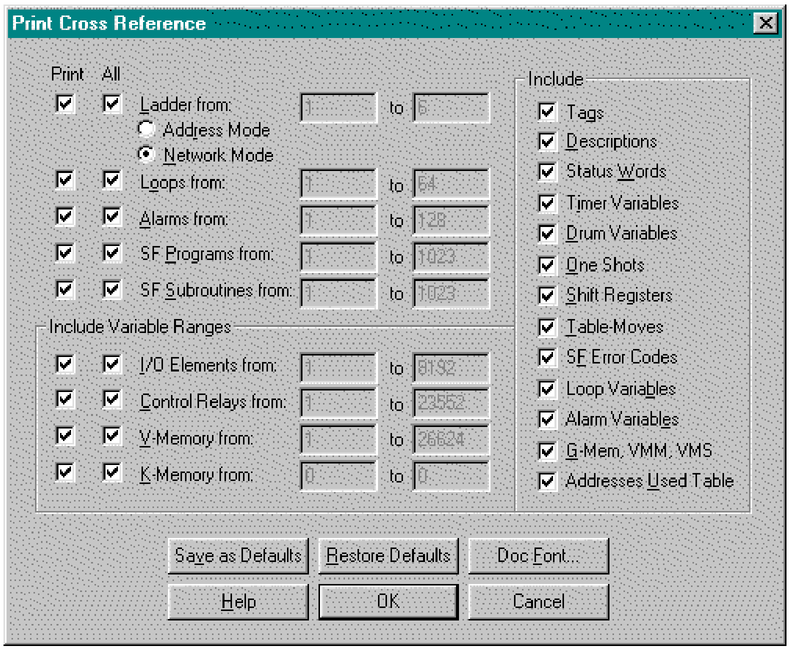


Figure 3.19

Click the All check box to print the entire range of items. To print a selection, deselect the All check box and enter a range in the associated edit box.

The following can be individually turned on and off to be included with Cross reference printout:

- Tags
- Descriptions Status Words
- Timer Variable
- Drum Variables
- One Shots Shift Registers
- Tables-Moves
- SF Error Codes
- Loop Variables
- Alarm Variables
- G-Mem/VMM/VMS

To include Addresses Used Table with the Cross-reference print out select the appropriate check box under Include.

1. Click on the Doc Font button to change the Tag, Description, and Comments printout font of the active program.
   Result: The Font dialog box appears.
2. Choose a new font, font style, or font size. Notice that you can see a sample of the font in the Sample box.
3. Click OK in the Font dialog box to save your changes or click Cancel to cancel changes.
   Result: The Print Cross-reference dialog box appears.

NOTE *The print selection options are stored when saving a program.*

Print to a Text File

1. To print to a text file select Output To Text File from the File menu. The Result: The Print Output to Text File dialog box appears (Figure 3.19b). *See Printing Logic Programs* for dialog box selections.
2. Once selections are made select OK.
3. Enter a file name to save to and select OK

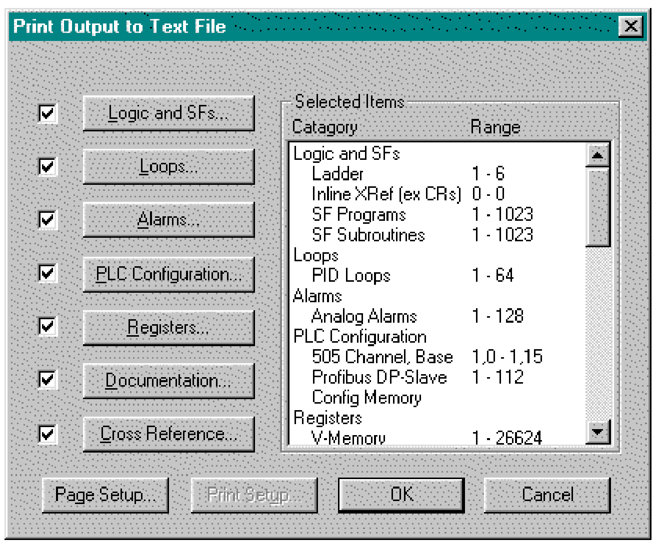


Figure 3.19b

Editing Features

505 WorkShop uses a number of timesaving editing features to help you complete your programming tasks. These include:

- Cut
- Copy
- Paste
- Paste With Rewire
- Undo
- Clear
- Delete
- Insert

The most frequently used editing features are Cut, Copy, and Paste. Use these commands to quickly copy logic and documentation to either another location in the same program or another program. The list below describes Cut, Copy, and Paste differences.

Window Feature	Function
Cut	Removes the selection from the program and places it on the clipboard.
Copy	Copies the selection and places it on the clipboard.
Paste	Inserts clipboard contents into the program at the cursor location.
Paste With Rewire	Inserts clipboard contents into the program at the cursor location and allows the user to re-address any addressable items contained in the clipboard.
Undo	Resets the networks/addresses in a segment to their original data.

The clipboard referred to is the standard Windows clipboard. Refer to your *Windows User's Guide* for more information.

To select the information you want to cut or copy, click, hold and drag the pointer over the desired area. Selected items are highlighted with a different color than your normal workspace color.

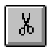
If you want to select all logic in the current window, select Select All from the Edit menu.

Each of three editing commands, described in detail in the following paragraphs, can be accessed several ways.

Cut

To use the cut feature:

1. Select the information you want to cut by highlighting the range instructions or networks/addresses of (SF) lines to be cut.
2. Click and hold down the left mouse button on the item, or network or (SF) lines to be cut, then drag the mouse to the end of the item range or network or (SF) lines to cut.
3. Cut your selection to the clipboard with one of the following:

- ❑  on the toolbar or
- ❑ Cut from the Edit menu

NOTE *If the start and ending networks/addresses or (SF) lines are known, then the cut from and cut to range can be entered directly into the Cut Range dialog box. If a partial network is selected, the Cut Range dialog box is not displayed. The items selected are cut without warning.*

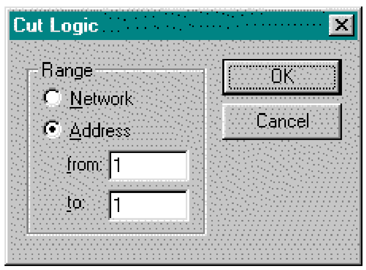


Figure 3.20

4. Select OK.
Result: The range of networks/addresses or (SF) lines displayed are cut out of the program and placed into the clipboard.

Copy

To use the copy feature:

1. Select the information you want to copy by highlighting the range instructions or networks/addresses or (SF) lines to be copied.
2. Click and hold down the left mouse button on the item, or network or (SF) lines to be copied, then drag the mouse to the end of the item range or network or (SF) lines to copy.
3. Copy your selection to the clipboard with one of the following:
   - ❑ on the toolbar or
   - ❑ Copy from the Edit menu

NOTE *If the start and ending networks/addresses or (SF) lines are known, then the copy from and copy to range can be entered directly into the Copy Range dialog box. If a partial network is selected, the Copy Range dialog box is not displayed. The items selected are copied without warning.*

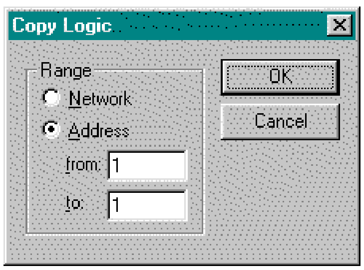


Figure 3.21

4. Click OK.

Result: The range of networks/addresses or (SF) lines displayed are copied and placed into the clipboard.

Paste

To access the paste feature:

1. Move the cursor to the desired location.
2. Paste clipboard contents into the new location with one of the following:
   - ❑ Click on the toolbar
   - ❑ Select Paste from the Edit menu
   - ❑ Press Ctrl+V

Paste With Rewire

Paste With Rewire provides you with a number of timesaving editing features. These include:

- Paste multiple copies
- Paste with address offset
- Include tags and descriptions in the paste
- The ability to Rewire (change address) on an individual basis

To access the rewire feature:

1. Move the cursor to the desired location.
2. Paste clipboard contents into the new location by:
   - Selecting Paste with Rewire from the Edit menu. The Paste With Rewire dialog box (Figure 3.22) appears.
   - Choose the appropriate options.
3. Click OK.

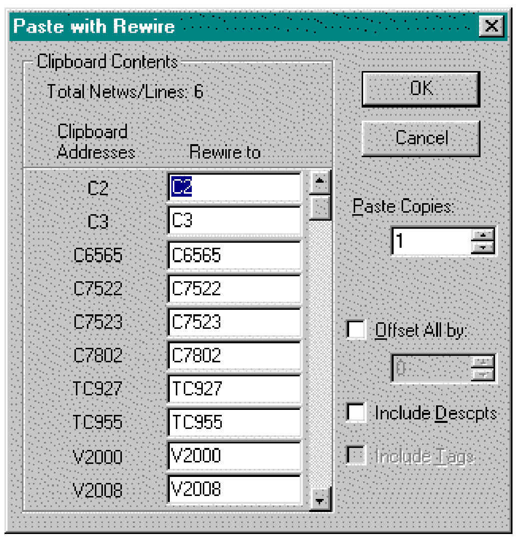


Figure 3.22

NOTE *When pasting, clipboard contents are inserted before existing items. For example, if you are pasting a network and the cursor is positioned at Network 002, click paste and the clipboard contents become Network 002. The previous Network 002 becomes Network 003.*

Undo

Use Undo to reset networks/addresses in a segment to their original data. Any modified or inserted rung can be reset. Deleted rungs cannot be reset.

To access the Undo feature:

1. Select Undo from the Edit menu or press Ctrl+Z.
   Result: The Undo Logic window appears (Figure 3.23).
2. Select the segments to reset and press OK, or press Undo All to reset all networks/addresses displayed.
3. Click Cancel to close the window.

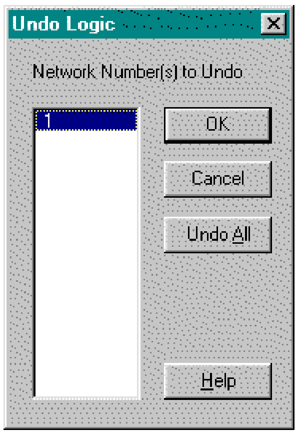


Figure 3.23

Clear

Use Clear to clear an item without removing the space it occupies. Access Clear from the Edit menu using the Logic Editor in either offline or online mode. Clear can also be accessed from the Data Window.

To clear items:

1. Select the item or items you want to clear by clicking, holding, and dragging the pointer over the desired logic.
2. Select Clear from the Edit menu or press the Delete key.
   Result: The Clear dialog box appears (Figure 3.24).
3. Click the items you want to clear.
4. Click OK or press Enter.
   Result: The selected items are cleared.

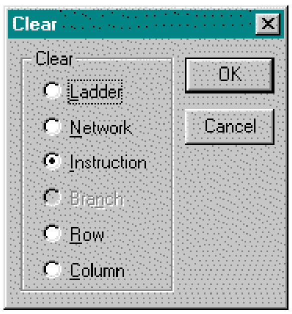


Figure 3.24 – Ladder Clear

The following table describes the clearing items.

Item	**Function**
Ladder	Removes all networks/addresses from the block displayed.
Network	Removes all logic from the network at the cursor position.
Instruction	This option is available if the cursor is positioned at an instruction. When cleared, the instruction is removed; however, attached branches are not affected.
Branch	This option is available if there is a branch to the right of the instruction with the cursor. Clearing removes the branch to the right of the cursor.
Row	This option is available when an instruction is selected. Clearing removes instructions from the row where the cursor is positioned. Branches in this row are cleared only if the resulting logic contains branches unconnected to logic at one or both ends.
Column	This option is available when an instruction is selected. Clearing removes instructions and branches from the current column.

Logic Editor - Online

Using Clear in the Logic Editor while online works the same as in offline mode. However, row and column cannot be cleared while online.

Data Window

While working in the Data Window, you can use Clear to clear all rows or one row at a time. Clear is accessed through the Edit menu or by pressing the Delete key.

Delete

Use Delete to delete an item and remove the space it occupies. Access Delete from the Edit menu using the Logic Editor in either offline or online mode. Delete can also be accessed from the Data Window.

To delete:

1. Select the item or items you wish to delete.
2. Select Delete from the Edit menu.
   Result: The Delete box appears (Figure 3.25).
3. Click the items you want to delete.
4. Click OK or press Enter.

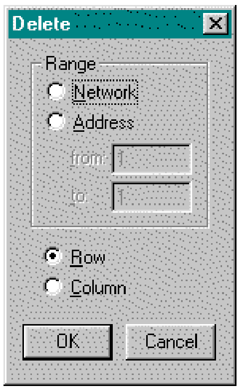


Figure 3.25 – Ladder Delete

The following table describes the deletion items.

Item	**Function**
Network	Deletes a network or a range of networks/addresses. To delete a range of networks/addresses, enter the number of the first network to delete in the From box. Then enter the number of last network to delete in the To box.
Row/Line	In Ladder, selecting a row deletes all instructions and branches from the row where the cursor is positioned. Logic below the deleted row(s) moves up. When box instructions prevent a deletion, an error message appears.
Column	Deletes instructions and branches from the column where the cursor is positioned. Logic to the right of the deleted column(s) moves left. When box instructions prevent a deletion, an error message appears.

Logic Editor - Online

Using Delete in the Logic Editor while online works the same as in offline mode. However, a row or column cannot be deleted unless it is empty.

Data Window

While working in the Data Window, you can use Delete to clear all rows or one row at a time. Delete is accessed through the Edit menu.

Insert

Use Insert to insert a selected object (network, instruction, row, or column) at the point of the current cursor position. Access Insert from the Edit menu using the Logic Editor in either offline or online mode.

To insert an object:

1. Select Insert from the Edit menu.
   Result: The Insert dialog box appears (Figure 3.26).
2. Click on the object you want to insert.
3. Click OK or press Enter.

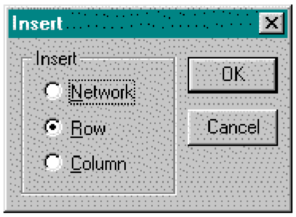


Figure 3.26 – Ladder Insert

Logic Editor - Offline

Item	**Function**
Network/Address	Adds a new network before the current network or address where the cursor is positioned.
Row/Line	Adds a new row is before the row where the cursor is positioned. If box instructions prevent insertion, an error message appears.
Column	Adds a new column before the column where the cursor is positioned. If box instructions prevent insertion, an error message appears.

Logic Editor - Online

Insert is accessed in the same manner online as offline. However, only a network or column can be inserted while working online.

Item	**Function**
Network	With the cursor positioned at a network, Insert places a new network before the current network.
Column	A new column is inserted before the column where the cursor is positioned. If box instructions prevent insertion, an error message appears.

Importing Text or Documentation Files

Import allows you to import tags/descriptors text files or 505 WorkShop *.fss files. For example, you can import tags/descriptors text created in a word processor. You can merge tags/descriptions from another 505 WorkShop (*.fss) program into the currently opened program.

To import tags/descriptors from a text file:

1. Select Import from the File menu.
   Result: The Import dialog box appears (Figure 3.27).
2. Select the type of program you want to import
3. Type the program name in the Filename line, or click Browse to search from valid program names.
4. Select either Comma Delimited or Tab Delimited.
5. Click OK to import the program.

To import tags/descriptors from another 505 WorkShop (*.fss) program into the currently opened program.

1. Select Import from the File menu.
   Result: The Import dialog box appears (Figure 3.27).
2. Select FasTrak 505 WorkShop Documentation (*.fss) from the Import Documentation dialog Format drop down window.
3. Type the program name in the Filename line, or click Browse to search from valid program names.
4. Select Tags, Descriptions, or Headers from Include to merge in.
5. Click OK to import the documentation.

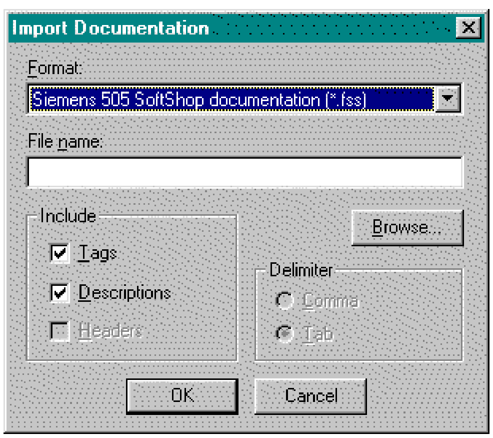


Figure 3.27

Merge Memory

Merge allows you to merge externally developed subroutine programs into User Memory. For example, you can merge compiled C, Pascal, assembly language, and other programs into U-Memory.

Follow these steps to prepare the external subroutine for use in the controller.

1. Compile/assemble the subroutines and header to create object modules.
2. Link the object modules for the header and subroutines to create the load module. The file name must have the extension ".rec". The output must have the header at zero followed by the code and data constants, then the variables, and finally the stack.
3. Select Merge from the File menu and U-Memory from the list Window.
   Result: The Merge dialog box appears (Figure 3.28).
4. Type the program name in the Filename line, or click Browse to search from valid program names.
5. Click OK to merge the program.

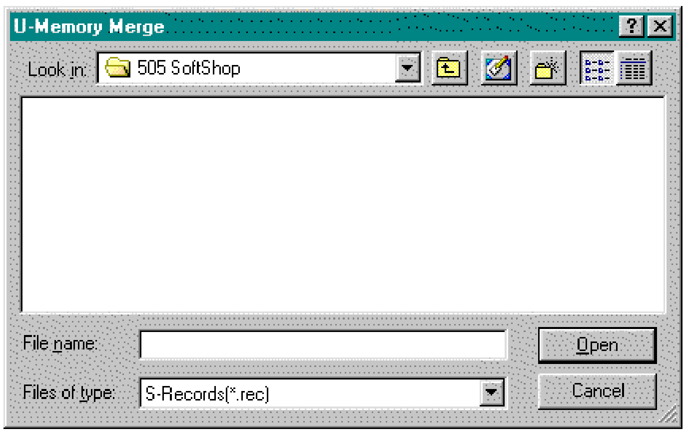


Figure 3.28

4 - 505 WorkShop Setup

Overview

Before you begin programming, you will probably want to spend some time configuring and customizing the software and some of the hardware attached to your computer. This chapter will help you with:

Program Setup	Shows you how to customize 505 WorkShop to fit your preferences.
Communications Setup	Walks you through how to tell your computer which port is attached to your PLC.
Printer Setup	Assists you in determining the correct settings for your printer.
Page Setup	Allows you to configure your pages for printouts.
Fast PLC Setup	Sets up a PLC for a fast connection.

Program Setup

Program Setup reveals different sets of tabs governing your projects layout and appearance. These settings are saved with the program; thus, each time you open this program, you do not have to reset your preferences.

To access the Program Setup options:

1. Select Program Setup from the Options menu.
   Result: The Program Setup dialog box appears (Figure 4.1).

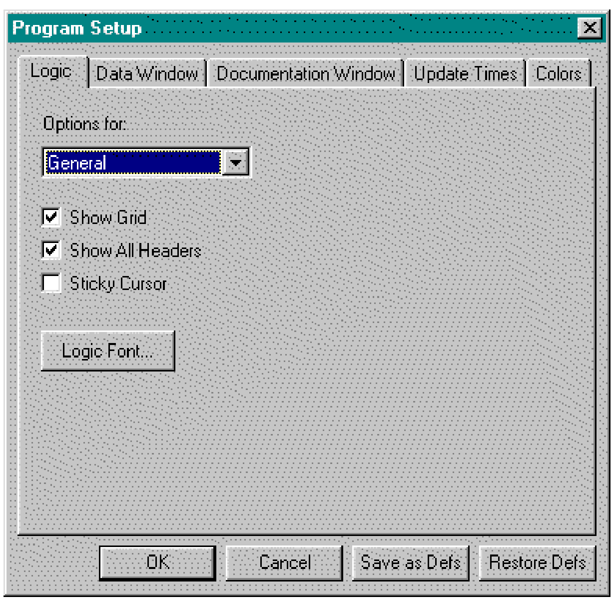


Figure 4.1

2. Click on the desired tab to display setup selections. A check in a box or radio button (circle) means that this feature or object is selected or is "turned on." A drop down window means that there are multiple options for the particular tab item selected. Table 4.1 describes each feature in the Program Setup dialog box.
3. Click **Restore Defs** to return each feature to the last Save as Defaults setting or the original "factory" setting if Save as Defs has not been used.

4. Click **OK** when you are finished making your selections or click **Save as Defs** to save the new settings as defaults. These new settings will be used for every new program created. Or Click **Cancel** to disregard changes to the settings and return to the active window.
 Result: The screen changes based on your new settings for the current program.

Logic Tab/General	**Description**
Ladder Grid	Displays the ladder grid when selected.
Sticky Cursor	When selected, the current ladder instruction is saved as the cursor. You have to manually select the pointer cursor when you are done editing the current instruction. If not selected, the cursor changes back to the pointer after inserting an instruction.
All Headers	Displays ladder, network, and SF headers when selected.
Font	Changes the font displayed in the active program. Any active Windows font can be selected. To change the font: 1. Click Font and the Font dialog box is displayed. 2. Choose a font, font style, and font size. Notice that you can see a sample of the font in the Sample box. 3. Click OK in the Font dialog box to save your changes and return to Program Setup. Click Cancel to make no font changes and return to Program Setup.

Logic Tab/Ladder	**Description**
Addresses	Displays addresses when selected.
Tags	Displays tags when selected.
Descriptions	Displays tag descriptions when selected.
Assign Tags	When selected, a window automatically appears if an address (that does not have a tag attached to it) is entered in ladder. It allows you to assign a tag, description, and comment to the address.
Assign Addresses	When selected, a window automatically appears if a tag (that does not match any current tags) is entered in ladder. It allows you to assign an address, description and comment to the tag.
Column Width	Adjusts the size of the ladder grid. Click the up or down arrow or type in a value between 7 and 24.
Description Rows	Determines the number of character rows displayed for each description. Click the up or down arrow or type in a value between 1 and 12.
Tag Rows	Determines the number of character rows displayed for each tag. Click the up or down arrow or type in a value between 1 and 4.
Use TISOFT Keys	Allows the use of certain TISOFT function keys to be used in windows. Such as; coils (Y, C, WY, V, G, W), contacts (X, Y, C, WX, WY, V, K, G, W), /, N, M, =, >, H, I, <, O, U, Ctrl U, J, U.
Status Thickness	Determines the line thickness of the ladder status line. Settings are between 1 and 6.

Logic Tab/Special Functions	**Description**
Instruction Width	Determines the width allocated for display of Special Function instructions on each line. Click the up or down arrow or type in a value between 1 and

	80.
Column One	Determines the width allocated for display of SF first column variables after the instruction. Click the up or down arrow or type in a value between 1 and 300.
Column Two	Determines the width allocated for display of SF second column variables after the instruction. Click the up or down arrow or type in a value between 1 and 300.
Data Window Tab	**Description**
Column Display Include Address	Turns on and off the Address columns in the data window
Column Display Include Tags	Turns on and off the Tags columns in the data window
Column Display Include Descriptions	Turns on and off the Descriptions columns in the data window
Column Display Include Time Stamp	Turns on and off the Time Stamp columns in the data window
Column Display Include Status	Turns on and off the Status columns in the data window
Tag/ Descriptions Tip Display Include Address	As you scroll through the tag/ descriptions combo box in a data window the corresponding address and description (if it exists) of the highlighted tag will be displayed to the left and right respectively of the combo box. These are referred to as documentation tips and can be turned on and off by checking and un-checking these boxes.
Maximum Rows	The maximum number of data window rows to display. Default is 100. minimum is 10, maximum is 1000
Include	Select Tags, descriptions, both, or neither when the Data window is displayed.
Documentation Window Tab	**Description**
Column Display	Select either Tags or Descriptions (or both) for display in the Documentation Window.
Sort Order	Select Address, Tag, or Description order for display in the Documentation Window.
Colors	**Description**
Item	The colors of Ladder, Ladder Grid, Ladder Background, Edited Ladder Background, Address Foreground, Tags Foreground, Description Foreground, Rung Header Foreground, File Header Foreground, Status Foreground, Status Optimize Foreground and Parameter Cursor Foreground can be changed when selected from the drop down box.

Table 4.1

Serial or Modem Communication

To configure your serial port or modem connection with a PLC:

1. From the Communications Setup dialog box, click the Serial Ports button.
Result: The Settings for Serial Ports and Modem dialog box appears (Figure 4.3).

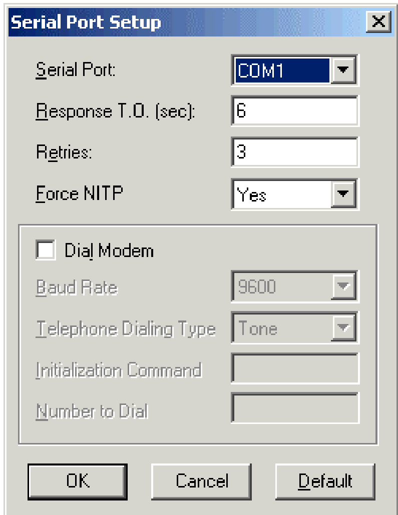


Figure 4.3

2. Select the appropriate setting for each option in the dialog box.
   - **Serial Port:** Location where the serial communication port (COM1, COM2, COM3, or COM4) is configured for communications. You do not select which port to communicate out of at this stage.
   - **Response T.O. (sec):** Specifies the amount of time, in seconds, that the software waits for a response from the PLC before returning a time-out error. Any whole number between 5 and 25 can be used.
   - **Retries specifies** Specifies the number of times the software will try to re-establish communications with the PLC after a time-out error. Any whole number between 0 and 10 can be used. Use 0 for no retries.
   - **Dial Modem:** Selected when the selected form of serial communications is through a modem. The modem parameters must be set to exactly the same communication parameters that you will use. Use the following modem parameters: eight bits, no parity, one-stop bit, and the highest baud rate that your equipment will support.
   - **Baud Rate:** The rate of communications between the computer and modem.
   - **Telephone Dialing:** Specifies which type of dialing to use. Specify pulse dialing only if this is the only type your phone line supports.
   - **Force NITP:** Specifies ASCII communications for the 545, 555, and 575.
   - **Initialization Command:** The initialization commands sent to the modem. Consult your modem manual for a list of appropriate commands.
   - **Number to Dial:** Specifies the phone number to be dialed. The number format can be **dash** (**414-358-8088**), **space** (**414 358 8088**), **period** (**414.358.8088**), or **none** (**4143588088**). Commas (,) may be used if a pause is needed to gain access to an outside line before the number is dialed. For example, if 9 is used to gain access to an outside line and there is a pause between the time 9 is pressed and a dial tone, then the number entered should be 9,414-358-8088.
3. Click OK or press Enter to accept the settings. Click Cancel to disregard changes and return to the Communications Setup dialog box.

505 WorkShop with H1 Communications

Before H1 Communications can be established with 505 WorkShop the following procedures must be completed:

- Install CP 1413 or 1613 card.
- Select I/O range.
- Select unique interrupt address.
- Select dual-port ram address.
- Install TF-1413 or TF-1613 software drivers using Simatic Net Software CD.
- Configure the CP 1413 or 1613 hardware and software using "COML TF" and "Setting the PG-PC Interface" from Siemen's Simatic Net Software CD.

Access H1 Devices Using 505 WorkShop

To access the H1 network with 505 WorkShop:

From the File menu, chose Fast PLC Setup.

1. Select H1 from the Fast PLC Connection Setup dialog box.
2. Click OK on the Setup dialog box.
3. From the File menu, chose Fast PLC Connection (H1 communications can also be established via open program).
   Result : The H1 Network Names screen appears (Figure 4.14).
4. Select a node name. These are the node names you configured using the COML TF-software prior to rebooting. Using the arrow keys or mouse, select an H1 network node name and Click OK .
   Result : 505 WorkShop is online.

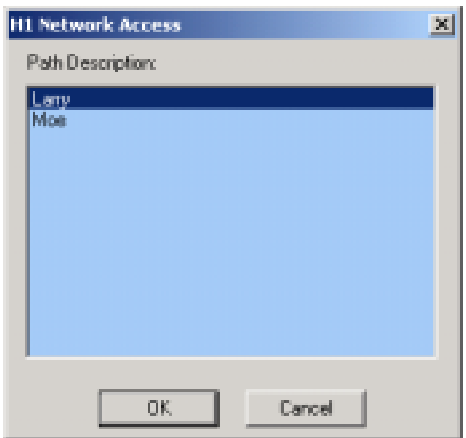


Figure 4.14

NOTE: *The CP1413 and CP1613 drivers do not run in protected mode. When using either of these drivers only one application may be used at a time.*

Setting Up and Using TIWAY in 505 WorkShop

There are two versions of the host adapter hardware for TIWAY: TIWAY I Host Adapter and UNILINK™.

NOTE: *Since the Network Interface Module (NIM) does not support some task codes when the controller is in run mode, you cannot perform the following functions communicating through a NIM: Force/Unforce and Find (except Find Address).*

Setting Up the UNILINK HOST Adapter

To communicate with 505 WorkShop, the UNILINK Host Adapter must conform to the following parameters:

- NITP protocol
- Full duplex
- Asynchronous
- Maximum Host Baud Rate: 19,200

See your *UNILINK Host Adapter User Manual* for details on setting dip switches for the UNILINK Host Adapter.

TIWAY Setup Configuration

1. From the Communications Setup dialog box, click the **TIWAY** button.
   Result: The TIWAY Setup dialog box appears (Figure 4.4).

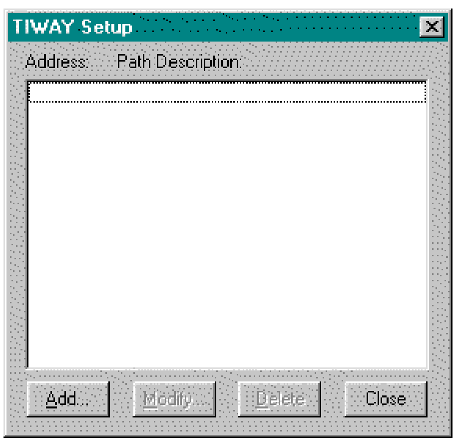


Figure 4.4

Secondary Addresses

1. From the TIWAY Setup screen dialog box, click the **ADD** button to enter a new controller secondary address, or the Modify button to change the controller secondary address information.
   Result: The TIWAY Path dialog box appears (Figure 4.5).

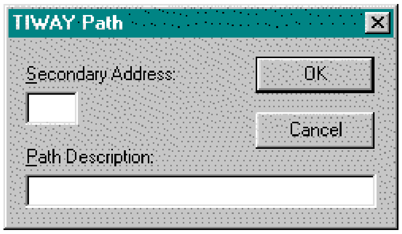


Figure 4.5

2. enter the controller secondary address for the associated controller. A controller secondary address is a unique number from 1 to 254 that is used to identify a controller on the TIWAY link. Each PLC on TIWAY has a controller secondary address.
3. Enter the controller secondary address Path Description. A Path Description is a 32-character alphanumeric description for the controller secondary address.
4. Click OK or press Enter to accept the settings. Click Cancel to disregard changes and return to the TIWAY Setup dialog box.

Using TCP/IP in 505 WorkShop and Setting Up TCP/IP Communications

Connect Your PC to Ethernet

Your system administrator must determine what type of cable is best suited for your installation because it affects your choice of Ethernet card for your PC.

NOTE The PPX:505-CP2572 module directly supports 10BaseT (UTP) cabling. If your existing network does not use UTP, but the cabling medium is IEEE 802.3 compliant, you can purchase a transceiver that connects that media to the AUI port on the module. If you are installing a new network, discuss your cabling requirements with your network administrator or your local Siemens distributor. Your Ethernet card purchase should be guided by the type of cabling medium that is best suited for your network. There are really only three PC card options: 10BaseT(UTP), 10Base2(Thin Ethernet), and AUI. Some cards are combinations. The most commonly used are either 10bT or AUI with Fiber Optics cable. 10Base2 is not as widely used, and is not recommended as highly because of the potential network problems. 10Base5 (Thick Ethernet) cable, which also requires the use of transceivers, is sometimes used and can be obtained through your Siemens distributor.

In order to communicate to a 505 PLC, which is connected to a TCP/IP network, a TCP/IP stack needs to be installed on your Windows machine. Windows 95 and Windows NT ship with a TCP/IP stack - WINSOCK.DLL.

If TCP/IP is not listed under the protocol section of your network settings, it needs to be added. You may be asked to insert a Windows disk or CD-ROM. After it has been added, click on properties and enter an IP address, subnet mask, and possibly a default gateway. See your network administrator for more information on these fields if you are not sure what to enter. Every machine on your network must have a unique IP address.

At this point, the 505 TCP/IP Ethernet board needs to be configured (see the following section). Make sure the IP address does not conflict with other devices on the network. A sample configuration appears below:

Devices	IP Address	SubNet Mask
PC #1:	201.98.1.1	255.255.255.0
PC #2:	201.98.1.2	255.255.255.0
505 TCP/IP #1:	201.98.1.8	255.255.255.0
505 TCP/IP #2:	201.98.1.9	255.255.255.0

To use 505 WorkShop over an Ethernet network with the TCP/IP protocol, you need to purchase and install the PPX:505-CP2572 module in a Series 505 base. Refer to the SIMATIC 505 Ethernet TCP/IP Communication Processor (505-CP2572) User Manual, PPX:505-8132-x, for instructions.

Assign an IP Address to the Module

There are two ways to configure the IP address for your PPX:505-CP2572 module. Each procedure has different advantages, as outlined in Table 4.2.

Procedure	Pros	Cons
Autostart	Communications are functional any time CPU GOOD LED is on. Can re – use same ladder logic program for multiple controllers.	Must reprogram EEPROM whenever CP2572 module is replaced. A CP2572 module programmed in one application and installed in another would respond to the wrong IP address, unless reprogrammed. Cannot swap CP2572 modules without reprogramming EPROMs through serial port by a PC.
PLC Start	Easy to troubleshoot problems by swapping CP2572 modules; proper IP address is loaded from ladder logic program as soon as CPU enters RUN mode.	IP Address is not loaded unless CPU enters RUN mode; if power is lost when CPU is not in RUN mode, you must manually bring CPU back to RUN mode (via programming device in CPU's RS-232 port) to restore Ethernet communications. If multiple controllers perform the same function, you must modify ladder logic program for each one so that it contains a unique IP address.

Table 4.2 Alternatives for Loading IP Address

The *SIMATIC 505 Ethernet TCP/IP Communication Processor (505-CP2572) User Manual* describes the Autostart and PLC Start procedures in detail; follow the instructions in the manual to configure the IP address for your module.

NOTE: *New or modified IP address configurations do not take effect until you power cycle the base containing the PPX:505-CP2572 module.*

Tips for Using the PLC Start Option

The *SIMATIC 505 Ethernet TCP/IP Communication Processor (505-CP2572) User Manual* describes how to build a Startup Network Command Block table, used with the PLC Start option, in the chapter on *Installation* (Chapter 2). An easy way to construct this table is to open a Data window (see *Using The Data Window*) and enter the desired V-memory address. (For instance, the Ladder Logic Example in the TCP/IP manual assumes that the command block is located in V-memory, starting at location V500.) From the desired location, you can simply key in the values from the Startup Network Command Block example table in the manual, supplying the correct IP address, IP route address, and subnet mask for your network.

For even more permanency, you can put the table into K-memory instead of V-memory. If you use the example ladder program from the TCP/IP manual, you can just add a MOVW box to move the values in K to V, as shown in Figure 4.6. You can also trigger your logic from the first-scan flag bit in status word 201, instead of creating a lock with coil C1.

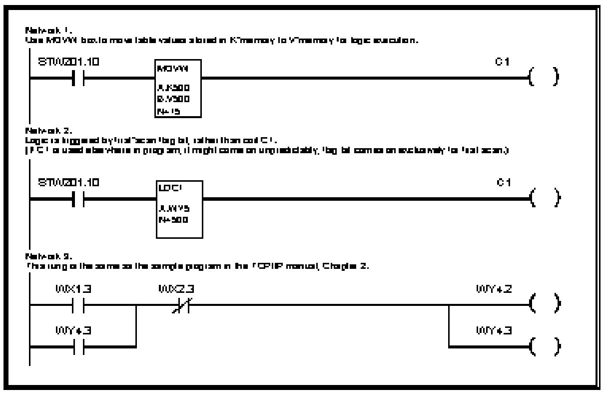


Figure 4.6

Connect the Module to the Ethernet

The PPX:505-CP2572 module directly supports 10BaseT (UTP) cabling. If your existing network does not use UTP, but the cabling medium is IEEE 802.3 compliant, you can purchase a transceiver that connects the media to the AUI port on the module. Consult the SIMATIC 505 Ethernet TCP/IP Communication Processor (505-CP2572) User Manual for information about how to connect cables to the 10bT or AUI port of your module. Consult the Siemens IK 10 catalog for information about how to purchase a Siemens transceiver if you need to use it with the AUI port.

Testing the Connection with PING

Once the board has been configured, you can test the settings by using a DOS command line utility called PING (comes with the TCP/IP stack). The program sends several test messages to the IP address that you specify on the command line. For example, PING 201.98.1.8; will test the connection to 505 TCP/IP #1 shown above. There should be four successful replies to the PING command. If PING is not successful, there is a problem with the network settings or the 505 TCP/IP board configuration. If PING does not work, 505 WorkShop will not be able to connect to the PLC.

After PING has successful replies, you can configure a TCP/IP connection in 505 WorkShop and open an online window.

TCP/IP Communication Settings

To configure your interface board port for communication with a PLC:

1. From the Communications Setup dialog box, click the **TCP/IP** button.
   Result: The Settings for the TCP/IP Setup dialog box appear (Figure 4.7).

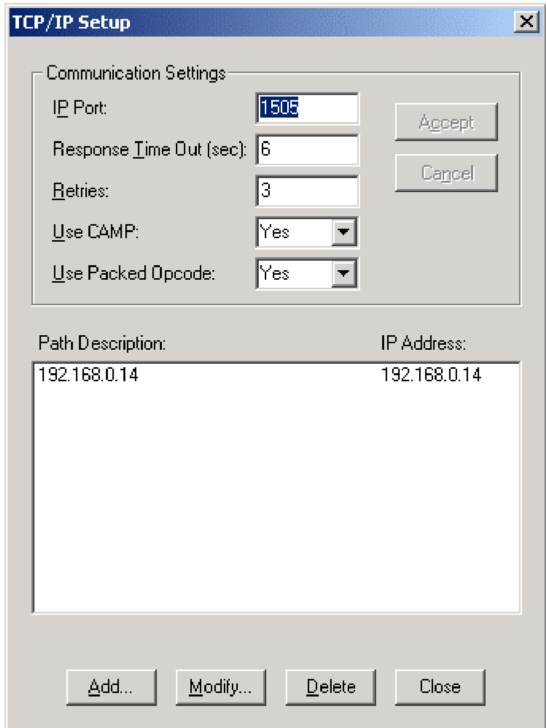


Figure 4.7

2. Enter the communication settings. Then Click the **Accept** button.

Communication Settings

- **Camp Protocol:**The "Use Camp" option is for optimizing upload and download of PLC programs specifically to CTI 2772 cards. This is defaulted to Yes and should only be turned off if the CTI card does not support the CAMP protocol.
- **Packed Opcode:**The "Use Packed" option is for optimizing the Ladder Status and Data Window updates. This is defaulted to Yes and should only be turned off if the Packed opcode is not supported.
- These settings are held in the computer Registery.
- **Path Descriptions:** When attaching to a PLC using TCP/IP, the user is given a path description. This description represents a specific TCP/IP address if a PLC. The description and associated path is entered at the bottom of this dialog.
- **IP Port**: Any number is acceptable as long as it does not interfere with other protocol numbers. This number must match the IP port configured in the 2572 Ethernet TCP/IP module
- **Response T.O. (sec):** Specifies the amount of time, in seconds, that the software waits for a response from the PLC before returning a time-out error. Any whole number between 5 and 25 can be used.
- **Retries:** Specifies the number of times the software will try to re-establish communications with the PLC after a time-out error. Any whole number between 0 and 10 can be used. Use 0 for no retries.

IP Addresses

1. From the IP Addresses section of the TCP/IP Setup dialog box, click the **ADD** button to enter a new IP address, or select an existing IP address and click the Modify button to change the IP information.
   Result: The IP Addresses dialog box appears (Figure 4.8).

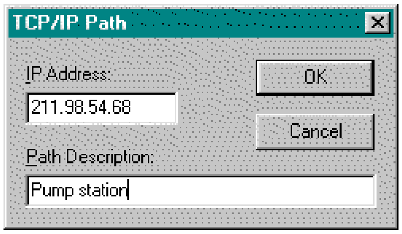


Figure 4.8

NOTE *The 505 WorkShop allows 1,000 different IP Addresses and Path Descriptions.*

---

2. Enter the IP Address for the associated Interface Board. An IP Address is a 32-bit value that is divided into four 8-bit fields, each separated by a period. For example, 192.3.2.1 is an IP Address. Each computer on a network has a unique IP Address. You should consult your network administrator for the correct IP Addresses for your computer and board.
3. Enter the IP Address Path Description. A Path Description is a 32-character alphanumeric description for the IP Address.
4. Click OK or press Enter to accept the settings. Click Cancel to disregard changes and return to the TCP/IP Setup dialog box.

NOTE *Refer to the* 2572 Ethernet TCP/IP Module *manual for proper settings.*

Setting Up and Using PROFIBUS–FMS in 505 WorkShop

Installing the PROFIBUS–FMS Communications Processor

The SIMATIC 505-CP5434-FMS Communications Processor module (referred to hereafter as the FMS CP module) provides the interface required for a SIMATIC 505 programmable logic controller system to communicate with other devices over a common PROFIBUS network.

To install the FMS CP module in a SIMATIC 505 base, follow the installation instructions in the *SIMATIC 505 PROFIBUS–FMS Communication Processor (505-CP5434-FMS) User Manual*.

The FMS CP module must be configured with the COM5434 Configuration software included with the module. The COM5434 Configuration software operates in Windows 95 or Windows NT only.

To install the software, follow the installation instructions in the *SIMATIC 505 PROFIBUS–FMS Communication Processor (505-CP5434-FMS) User Manual*.

Each FMS CP module in a 505 base must be configured to communicate over the PROFIBUS network. The "module local configuration" for each FMS CP module identifies its station address and the network communication parameters it uses to operate on the network.

***NOTE**: You must define the module local configuration for each FMS CP module using the RS-232 port with the standard 505 WorkShop programming cable before you can communicate with it using the PROFIBUS-FMS port.*

Refer to the *SIMATIC 505 PROFIBUS-FMS Communication Processor (505-CP5434-FMS) User Manual* for complete information on configuring the FMS CP module with the COM5434 Configuration software.

***NOTE:** Make sure to select the bus parameters, the baud rate, and the highest station address (HSA) that match those of all the other modules on the network. Also be sure to select a unique station address for each FMS CP module on the network.*

Once each FMS CP module has been configured to operate on the network, connect each FMS CP station and the CP 5412 card in your PC to the PROFIBUS network, using the PROFIBUS cables and connectors described in the *SIMATIC 505 PROFIBUS-FMS Communication Processor (505-CP5434-FMS) User Manual*.

Fast PLC Setup

The Fast PLC Setup allows you to configure a single PLC Connection when using Fast PLC Connection.

To access the Fast PLC Setup:

1. Select Fast PLC Setup from the File menu.
   Result: The Fast PLC Setup dialog box appears (Figure 4.17).

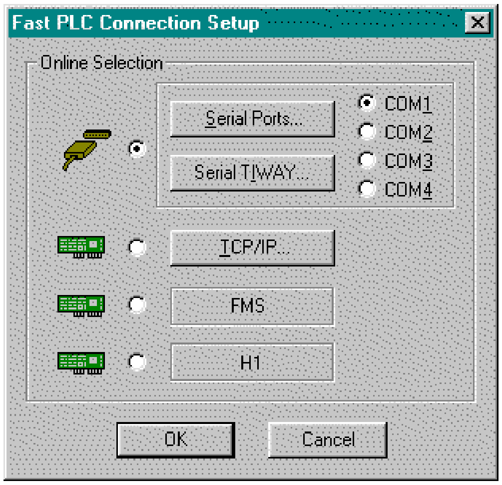


Figure 4.17

2. To configure and select your serial port or board for communication with a PLC:

- Select the appropriate Communication Port from the list in the Fast PLC Connection Setup dialog box. If you need to configure a port, click Serial Ports (Refer to *Communications Setup* for more information.)
- Click OK or press Enter to accept the settings. Click Cancel to disregard changes.

NOTE: *The connection options for your Fast PLC Connection is stored in the 505-registry. If you attempt to use Fast PLC Connection and the Fast PLC Setup has not been configured, the Fast PLC Connection Setup window automatically displays before continuing. Connecting to a PLC using Fast PLC Connection does not load any documentation or tag information. If you need to load documentation or tags, use Open Program.*

Printer Setup

Use Print Setup to select a printer and determine where and how your printouts appear.

To access the Print Setup:

1. Start or open a logic program. Select Print Setup from the File menu.
   Result: The Print Setup dialog box appears.
2. Depending on which printer you want to use, choose either Default printer or Specific printer. If you select Specific Printer, choose a printer or device from the drop-down box.
3. Additional setup options may be available depending on the printer you selected. If an Options button is available, click it and another dialog box appears. Make your selections and select the OK button to return to the Print Setup dialog box.
4. Other options in the Print Setup dialog box include Orientation and Paper Source. Click the desired settings.
5. Click OK in the Print Setup dialog box to save your settings and return to the active logic window.

NOTE: *The print setup options can also be accessed from the Print box that appears after selecting Print from the File menu.*

Page Setup

Use Page Setup to select page margins, starting page number, and whether to include a Title page in your printout.

To access the Page Setup:

1. Start or open a logic program.
2. Select Page Setup from the File menu or click on Page Setup in the Print Window.
   Result: The Page Setup dialog box appears.
3. Depending on which features you need to customize, choose left, right, top and bottom margin. To change the size of the margins, type the measurement (in inches) for the margin you want to adjust in the Top, Bottom, Left or Right boxes.
4. If you would like to start your printout with a page number other than 1, change the Starting Page number.
5. Normally a title page does not print. The title page contains the information you entered in the Title Page Print Editor. If you would like to include this page as the first page in your printouts, click on Include Title Page.
6. Click OK in the Page Setup dialog box to save your page settings and return to the active logic window.

NOTE*: The page setup options can also be accessed from the Print box that appears after selecting Print from the File menu. The page setup options are stored when saving a logic program file.*

5 - PLC Memory & I/O Configuration

Overview

This chapter shows you how to set up and configure your Siemens family PLC. You must configure your PLC before you can create a ladder logic program. Configuration is part of the program; it performs the important function of relating the hardware components to the logic components.

The setup and configuration process is completed in three steps in recommended order:

1. 1. PLC Type Setup
   - ❑ PLC Type
   - ❑ Memory Size
1. 2. PLC Memory Configuration
   - ❑ Ladder
   - ❑ Variable
   - ❑ Constant
   - ❑ Special
   - ❑ User Sub
   - ❑ Timer/Counter
   - ❑ Drums
   - ❑ Shift Register
   - ❑ Table Move
- ❑ One Shots

2. I/O Configuration

   PLC Type Setup, available in offline mode only, is how you tell 505 WorkShop what Siemens processor you are creating a logic program for. PLC Memory Configuration and I/O Configuration allow you to configure your PLC. These are available in online and offline mode.

PLC Type Setup (Offline)

The setup and configuration process begins with PLC Type Setup. You must tell 505 WorkShop what processor you are using before you can configure the processor.

Valid PLC types are:

- Simatic 520 revision 1101.
- Simatic 520c revision 1101 and 1102.
- Simatic 525 revision 1102 and 1104.
- Simatic 530 revision 1102, 1104, and 1108.
- Simatic 530c revision 1104, 1108, and 1112.
- Simatic 535 revision 1104, 1108, and 1112.
- Simatic 560 revision 2120.
- Simatic 560/565 revision 2120.
- Simatic 560T revision 2820.
- Simatic 560/565T revision 2820.
- Simatic 545 revision 1101, 1102, 1103, 1104, 1105 and 1106.
- Simatic 555 revision 1101, 1102, 1103, 1104, 1105, and 1106.
- Simatic 575 revision 2102, 2103, 2104, 2105, and 2106.

To set up the PLC:

1. Select PLC Type Setup from the File menu.
   Result: The PLC Type Setup dialog box appears (Figure 5.1).

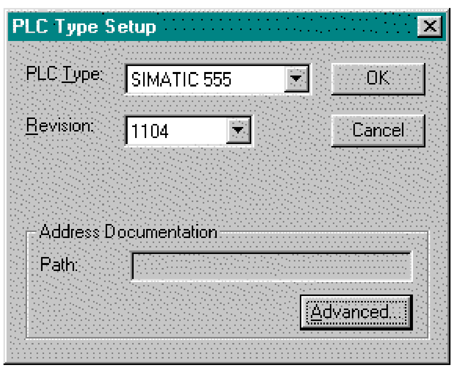


Figure 5.1

NOTE: *to access a 545-1101 extended memory offline the extended memory check box must be selected at this point, as shown in figure 5.1a.*

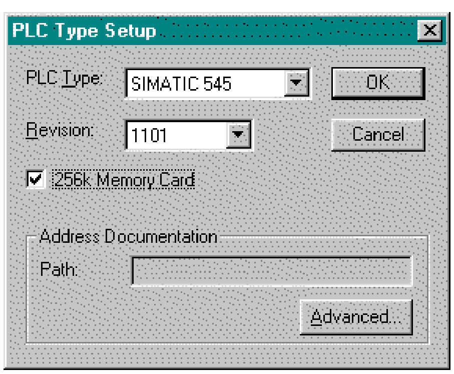


Figure 5.1a

2. Select the appropriate PLC Type and PLC Memory Size. A detailed explanation of each setup option is described on the following table (Table 5.1). If shared documentation is to be associated with the new program it must be setup at this stage *see Shared Documentation*.

PLC Setup Option	**Description**
PLC Type	Specifies the type of processor. Selection of PLC type determines choices available for remaining setup options in the PLC Type Setup dialog box.
PLC Memory Size	Specifies the PLC's memory size. Only the valid memory sizes for the PLC selected in PLC Type are displayed.

Table 5.1

3. Click OK or press Enter to save your settings and return to the active logic program.

Changing PLC Types

The PLC type can be changed even after programming has been initiated. Various error or warning messages can occur when changing from one PLC type to another. These error messages indicate the block and segment of the error. The errors must be fixed before the new PLC's logic can be validated.

To change the PLC type:

1. Select PLC Type Setup from the PLC Utilities menu.
   Result: The PLC Type Setup dialog box appears (Figure 5.1).
2. Select the appropriate PLC Type and PLC Memory Size.

PLC Memory Configuration

Controller memory is composed of several functional types (refer to Table 5.2). You can configure the amount of memory that is allotted to some these areas. This is dependent upon your application and PLC type. The configurable memory sizes are given in the SIMATIC 545/555/575 System Reference Manual.

Ladder	User Program Memory • Ladder Memory stores RLL program. • Special Memory stores loops, analog alarms, and SF Programs. • User Memory stores user-defined subroutines.
User Data	Data Area Memory • Variable Memory stores variable data. • Constant Memory stores constant data. • Global and VME Memory are used for VME data transfer (applies to 575 only).*
System Operation	System Memory • RLL instruction tables: drum, timer/counter, shift register, etc. • Image registers and control and relays.* • Subroutine parameter area.* • SF program temporary memory.*

- Status Word memory.*

* Not Configurable

Table 5.2

The configuration process begins with PLC Configuration.

Accessing PLC Configuration

1. Select PLC Configuration from the PLC Utilities menu.
   Result: The PLC Configuration dialog box appears (Figure 5.2).

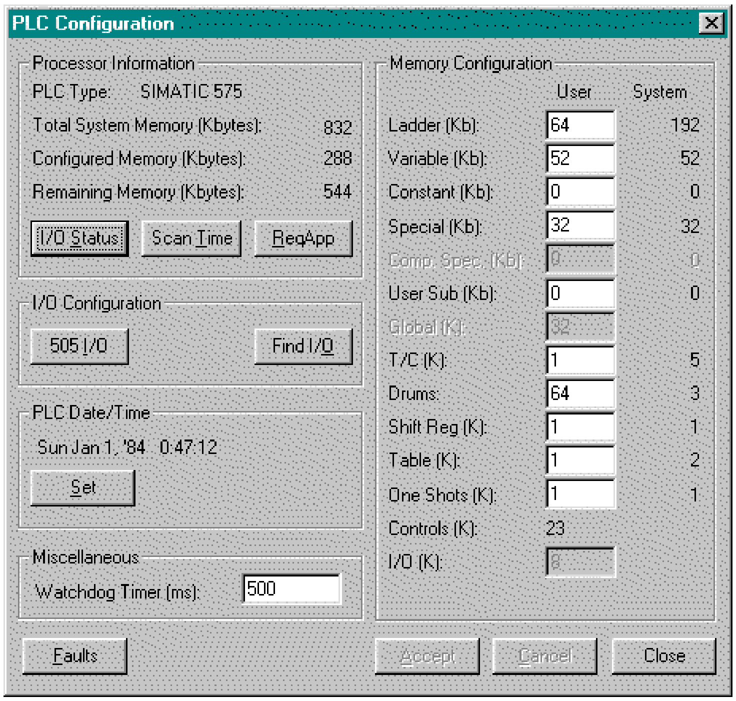


Figure 5.2

2. Enter the appropriate ranges for the selected PLC memory type.

NOTE: *The processor type must be selected before configuration.*

The various memory types are described in the pages that follow. Memory types are classified for RLL programming purposes in the following ways:

- Writeable - This memory type is read/write. It can be used for both input and output fields of RLL instructions.
- Readable - This memory type is read only. It can be used only for the input fields of RLL instructions.
- No access - RLL instructions has no access to this memory.

3. Click Accept to save your settings and return to the active logic program.

Controller Memory Types

Ladder Memory

A block of memory within the controller is reserved for the RLL program. This memory type is called Ladder Memory (L-Memory). Each RLL instruction used in the program requires one or more 16-bit words of L-Memory.

Image Register Memory

A block of memory within the controller is reserved for maintaining the status of discrete inputs/outputs. This memory type is called the discrete image register. A word image register holds the values of word inputs/outputs.

Control Relay Memory

A block of memory within the controller is reserved for control relays. Control relays are single-bit internal memory locations and do not represent actual hardwired devices.

Special Memory

A block of memory within the controller may be allocated for loops, analog alarms, and Special Function programs. This memory type is called Special Memory (S-Memory). All loop and analog alarm parameters are stored in S-Memory when you program the loop or analog alarm. Likewise, when you create a Special Function program or subroutine, the program is stored in S-Memory.

Temporary Memory

A block of memory within the controller is temporarily reserved during run time whenever a Special Function program is run. One block is allocated for each SF program that is being run. This memory type is 16 words in length and is called Temporary Memory (T-Memory) since it is not saved when the program has completed running. The controller writes data related to the Special Function program to the first 7 words. You can read this data and/or write over it if you choose. You can use all 16 words just as you would use Variable Memory, except no data is saved when the program has completed.

Variable Memory

A block of memory within the controller may be allocated for user operations. This memory type is called Variable Memory (V-Memory). For example, you can do a math operation and store the result in V-Memory. You can enter values directly into V-Memory with a programming unit.

Constant Memory

A block of memory within the controller may be allocated for constants (unchanging data). This memory type is called Constant Memory (K-Memory). You can use a programming unit to load a table of data into K-Memory and read the table during run time whenever you need the data for an operation.

Status Word Memory

A block of memory within the controller is allocated for storing status information relating to controller operations. This information is stored in one or more status words: STW1, STW2, etc. These status words can be used in the RLL program to signal and/or correct alarm conditions.

Timer/Counter Memory

A block of memory within the controller is reserved for the operation of the timer/counter group of RLL instructions, including the following:

Timer (TMR, TMRF)	**Counter (CTR)**
Discrete Control Alarm Timer(DCAT)	Motor Control Alarm Timer(MCAT)
Up/Down Counter (UDC)	

Warning	**When you assign a number to a timer, counter, up/down counter, or discrete/motor control alarm timer, be sure that you do not use that number for any other timer, counter, up/down counter, or discrete/motor control alarm timer. For example, if you configure a Timer 6 (TMR6), do not configure any other operation, e.g., a counter (CTR) or a discrete control alarm timer (DCAT) with the number 6.** **Assigning the same number more than once could cause unpredictable operation by the controller, which could result in death or serious injury to personnel and/or damage to equipment.** **Do not use the same reference number more than once for timer, counter, up/down counter, and discrete/motor control alarm timer instructions.**

NOTE: *If you use an operator interface to change the time/counter values, the new values are not changed in the original RLL program. If the RLL presets are ever downloaded, e.g., as the result of a complete restart or an edit of the network containing the Timer/Counter instruction, the changes made with the operator interface are replaced by the values in the RLL program.*

This memory type is divided into areas for storing two types of information. This information consists of Timer/Counter Preset (TCP) data and Timer/Counter Current (TCC) data. When you designate a preset value for one of the instructions in this group, this value is stored as a 16-bit word in TCP-Memory. When the instruction is actually operating, the current time or count is stored as a 16-bit word in TCC-Memory.

Table Move Memory

A block of memory within the controller is reserved for the operation of the table move instructions, including the following:

- Move Word To Table (MWTT).
- Move Word From Table (MWFT).

Warning	**When you assign a number to a table move instruction, be sure that you do not use that number for any other table move instruction. For example, if you configure a Move Word To Table #1 (MWTT1), do not configure a Move Word From Table #1 (MWFT1).** **Assigning the same reference number to more than one table move instruction could cause unpredictable operation by the controller, which could result in death or serious injury to personnel and/or damage to equipment.** **Do not use the same reference number more than once for a table move instruction.**

This memory type consists of one word per table move instruction configured. This word is used to maintain the current count of moves done since the MWTT or MWFT instruction was last reset.

One Shot Memory

A block of memory within the controller is reserved for the operation of the various instructions of the One Shot group, including the following:

- One Shot
- Time Set
- Date Set

Warning	**When you assign a number to a One Shot instruction, be sure that you do not use that number for any other One Shot instruction type. For example, do not configure more than one OS11.** **Assigning the same number for more than one One Shot instruction type can cause unpredictable operation by the controller, which could result in death or serious injury to personnel and/or damage to equipment.** **Do not use the same number more than once for the same instruction type (e.g., use it only once in One Shot, in Timer Set, etc.).**

This memory type consists of one byte per configured One Shot instruction. This byte is used to save the previous state of the instruction input.

Because the instructions in the One Shot group use different bits of one byte, these instructions can be assigned identical reference numbers**.** That is, if you configure a One Shot #11 (OS11), you can configure a Date Set #11.

Shift Register Memory

A block of memory within the controller is reserved for the operation of the shift registers, which include the following:

- Bit Shift Register (SHRB).
- Word Shift Register (SHRW).

This memory type consists of one byte per shift register. This byte is used to save the previous state of the instruction input.

Warning	**When you assign a number to a shift register, be sure that you do not use that number for any other shift register type. For example, do not configure SHRB11 and SHRW11.** **Assigning the same number for more than one shift register could cause unpredictable operation by the controller, which could result in death or serious injury to personnel and/or damage to equipment.** **Do not assign the same reference number to more than one shift register instruction.**

Drum Memory

A block of memory within the controller is reserved for the operation of the various drum types, including the following:

Drum (DRUM)	Event Drum (EDRUM)
Maskable Event Drum Discrete(MDRMD)	Maskable Event Drum Word(MDRMW)

Warning	**When you assign a number to a drum type instruction, be sure that you do not use that number for any other drum type instruction. For example, if you configure a Maskable Event Drum Word #1 (MDRMW1), do not configure an Event Drum #1 (EDRUM1).** **Assigning the same reference number to more than one drum type instruction could cause unpredictable operation by the controller, which could result in death or serious injury to personnel and/or damage to equipment.** **Do not assign the same reference number to more than one drum type instruction.**

Drum memory is divided into areas for storing the following types of information:

Drum Step Preset (DSP)	Drum Step Current (DSC)
Drum Count Preset (DCP)	Drum Count Current (DCC)

When you specify step and counts-per-step (count preset) values for a drum type, the step preset is stored as a 16-bit word in DSP-Memory, and the counts-per-step values are stored as 16 consecutive 16-bit words in DCP-Memory (except for the DRUM). For the DRUM instruction, counts-per-step values are stored in L-Memory; DCP is not used.

NOTE *If you use an operator interface to change the drum preset values (DSP or DCP), the new values are not changed in the original RLL program. If the RLL presets are ever downloaded, e.g., as the result of a complete restart or an edit of the network containing the drum instruction, the changes made with the operator interface are replaced by the values in the RLL program.*

When the instruction is actually operating, the current step is stored as a 16-bit word in DSC-Memory. The current count for this step is stored as a 16-bit word in DCC-Memory.

PGTS Discrete Parameter Area

The Parameter Go To Subroutine (PGTS) discrete parameter area (Figure 5.3) is an area of memory within the controller that is reserved for holding the status of discrete bits referenced as parameters in a PGTS RLL instruction. Because up to 32 PGTS subroutines can be programmed, the controller has 32 discrete parameter areas, each capable of storing the status for 20 discrete parameters. When you use a parameter in the subroutine, refer to discrete points as Bn where n = the parameter number.

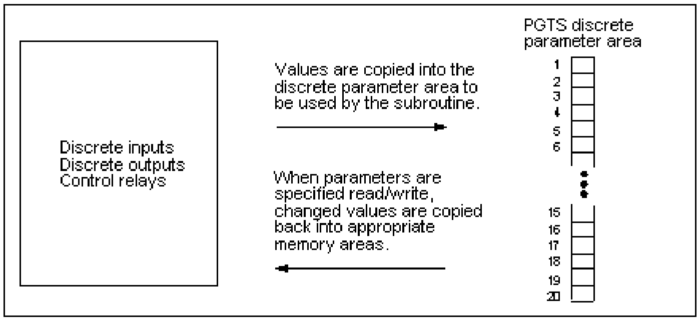


Figure 5.3

PGTS Word Parameter Area

The PGTS word parameter area (Figure 5.4) is an area of memory within the controller that is reserved for holding the contents of 16-bit words referenced as parameters in a PGTS RLL instruction. Because up to 32 PGTS subroutines can be programmed, the controller has 32 word parameter areas, each capable of storing the status for 20 word parameters. When you use a parameter in the subroutine, refer to words as Wn, where n = the parameter number.

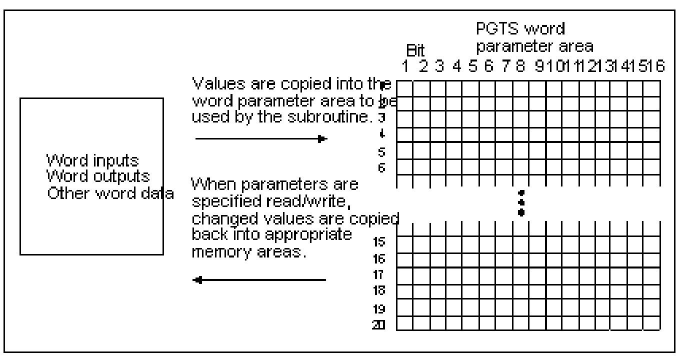


Figure 5.4

User External Subroutine Memory

A block of memory within the controller may be allocated for storing externally developed programs written in C, Pascal, Assembly language, etc. This memory type is called User Memory (U-Memory). The size of U-Memory is user configurable.

Global Memory: 575 Only

The 575 CPU allocates a 32K-word block of memory that allows you to transfer data over the VME back-plane. This memory type is called Global Memory (G-Memory). Refer to Appendix I of the processor manual for more information about G-Memory.

VME Memory: 575 Only

The 575 controller also allows access to physical VME addresses using the VMM-Memory or VMS-Memory.

- VMM corresponds to VME address modifier 39 (standard non-privileged data access).
- VMS corresponds to VME address modifier 29 (short non-privileged access).

Warning	**The 575 controller allows you to use a VME address (VMM or VMS) as a parameter to most word-oriented RLL instructions, e.g., ADD, SUB, or MOVW, etc. When a VME address is used and is not recognized by any installed board, a VMEbus error occurs. If the instruction that used the address was other than MOVE or XSUB (with the U-Memory header's E bit set to 1--see Appendix H in the Programming Reference Manual), the controller enters the Fatal Error mode, freezes analog outputs and clears discrete outputs. Use the XSUB or MOVE instruction to access the VME address.**

I/O Configuration

Controller Functionality in Configuration

Keep in mind that, while you can configure I/O either online (with the controller) or offline (programming device only), functional differences exist between the two modes. When online, you can perform those functions that require interfacing with the controller. For instance, you must be online to write your I/O configuration to the controller or to read the configuration of a base from the base itself. Configurations saved offline go to the selected program on disk.

I/O Configuration Guidelines

Before entering your I/O configuration, be sure that the I/O points you select conform to the following guidelines:

- The number for the I/O address must begin on an 8-point boundary. An 8-point boundary is *(n*8) + 1*, e.g., 1, 9, 17, etc. Addresses not starting on an 8-point boundary are changed to do so when you write the values.
- Refer to the I/O module manual for the number of bit and/or word I/O points required for each module. Valid entries for modules with more than 8 points are even numbers from 2 through 28, 32 ,and 64.
- Locations assigned to an I/O module cannot cross I/O channel boundaries. See your controller manual for details.
- 505 WorkShop does not flag duplicate I/O points.

Accessing I/O Configuration

1. Select PLC Configuration from the Options menu.
   Result: The PLC Configuration dialog box appears (Figure 5.2).

NOTE *You can Enable or Disable a base by highlighting the desired base number in the display screen, then clicking on the Enable or Disable button*

---

1. Select the 505 I/O button under I/O configuration on the PLC Configuration dialog box.
   Result: The I/O Configuration dialog box appears (Figure 5.5).

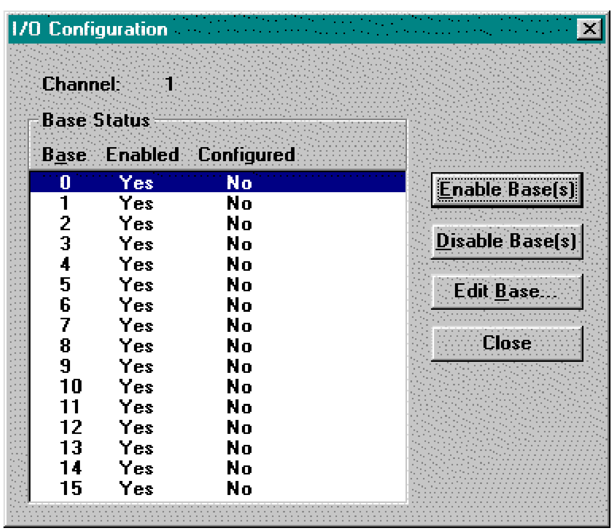


Figure 5.5

I/O Configuring Procedure

2. Use the mouse or arrow keys to highlight the desired base number on the I/O Configuration dialog box to edit or configure.

3. Then select the Edit Base button on the I/O Configuration dialog box.
   Result: The Edit I/O Base dialog box appears (Figure 5.6).

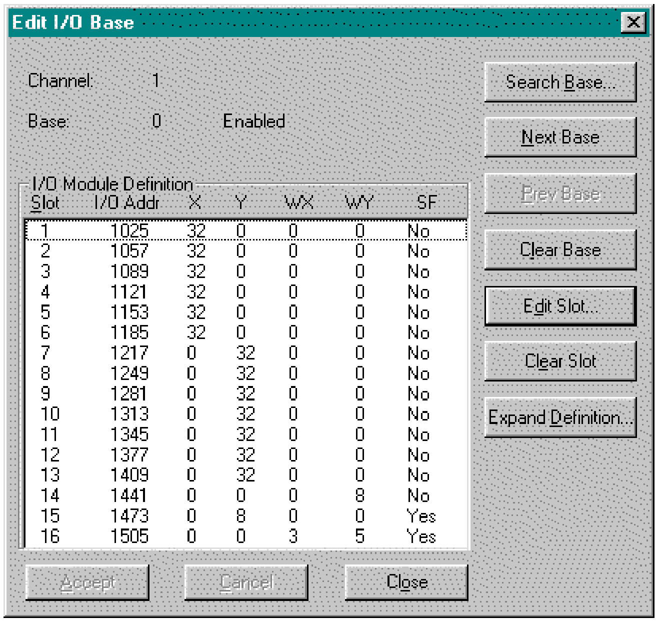


Figure 5.6

***NOTE**: If you need to select a different Base number, you can do so by clicking on the **Next Base** or **Prev Base** button on the Edit I/O Base dialog box. You can also click on the **Search Base** button on the Edit I/O Base dialog box and enter a different base number.*

***NOTE:** The individual I/O module can be displayed with its associated Slot number, Tag and Description by choosing the Expanded Definition button on the Edit I/O base dialog box.*

---

4. Use the mouse or arrow keys to highlight the desired slot number to be configured.

5. Click on the Edit Slot button on the dialog box or press Enter.
   Result: The Edit I/O Slot dialog box appears (Figure 5.6a).

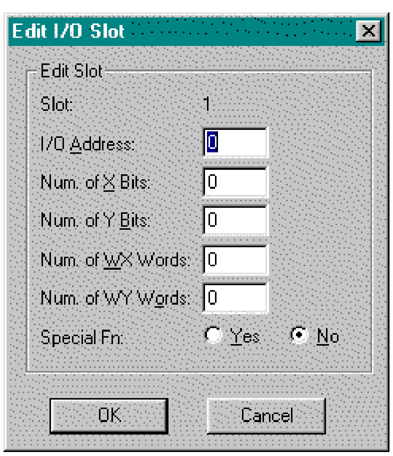


Figure 5.6a

1. Enter the beginning I/O address.
2. Enter the number of I/O points (X, Y, WX, or WY) required for the type of module being configured.
3. If a special function module is to be configured, click the Yes Special Function radio button.
4. Click on the OK button of the Edit I/O Slot dialog box.
5. The Edit I/O Base dialog box becomes the active dialog box (Figure 5.6).
6. To accept the changes, click on the Accept button on Edit I/O Base dialog box.

NOTE: *If you are online and have not yet accepted the changes to the base, you can select the Read I/O Base button on the Edit I/O Base dialog box to reset the base display to the readings in the controller. All prior changes will be lost.*

Clearing a Base I/O Configuration

1. Start from Edit I/O Base dialog box (Figure 5.6).
2. Click on the Clear Base button on the Edit I/O Base dialog box.

You have finished setting up the software and the hardware. The next step is programming. The Programming chapter discusses 505 WorkShop's many different features for developing ladder logic. You can program in online or offline mode.

Profibus-DP I/O Configuration

To configure I/O used on the PROFIBUS-DP channel for your Series 505 system, you must use two configuration utilities.

The first utility you use is COM PROFIBUS that allows you to perform the following tasks:

- Select a master and host, thereby determining communication parameters.
- Configure and parameterize the I/O slaves (including any 505 PROFIBUS-DP RBCs) that you wish to use on the PROFIBUS-DP I/O channel.
- Save your configuration. (This file should be archived in case you want to make configuration changes at a later time.)
- Export your configuration to a binary file. 505 WorkShop uses information from this binary file to perform further I/O configuration.
- If you need to add or delete slaves or modules or modify bus parameters later, you must use COM PROFIBUS to modify the configuration file and adjust the necessary information before exporting and merging it into 505 WorkShop again.

The second program you use is 505 WorkShop. 505 WorkShop allows you to perform the following tasks:

- Use the Merge function to import information from the binary file that you created with COM PROFIBUS.
- Assign or modify starting I/O addresses for the modules of all PROFIBUS-DP slaves used by your Series 505 CPU.
- Load bus parameters and slaves.
- Toggle between operate and stop mode, and between synchronous and asynchronous communications.
- Enable or disable slaves.

Configuring I/O Modules in COM PROFIBUS

This section is not intended as a tutorial in COM PROFIBUS. The SIMATIC S5 ET 200 Distributed I/O System Manual describes how to use COM PROFIBUS; consult that manual for detailed information. Also, COM PROFIBUS has extensive, context-sensitive help.

Your objective in creating a configuration file with COM PROFIBUS is to define module types for each slave. Once you accomplish that, you can export the configuration file in binary form to 505 WorkShop, where you assign starting I/O addresses to each module.

Table 5.3 provides the information you need in order to configure your slave devices in COM PROFIBUS.

Task	**Comment**
Open a file in COM PROFIBUS.	You can modify an existing configuration file, if appropriate, or create a new file.
Select a master.	Required. Choose the 505-CP5434-DP.
Select a host.	Required. Choose the appropriate Series 505 CPU.
Modify bus parameters.	Optional. You can accept the default settings, unless you wish to modify one, such as baud rate.
Assign slave(s).	Required. Choose slave(s), such as the 505 PROFIBUS-DP RBC, up to 112.
Parameterize slave(s).	Optional. (Note: If you are configuring a 505 PROFIBUS-DP RBC, PPX:505-6870, you may want to change the settings for I/O mismatch handling and RS-232 port communications; consult the SIMATIC 505 PROFIBUS-DP RBC User Manual for a description of the software parameters of the RBC.)
Configure slave(s).	Required. You can use either the "ID" or the "Order Number" list to select I/O module types and assign them to the appropriate slots in your slave device. The "Comment" field is optional. Do not fill in the "I. Addr." or "O. Addr." fields; you will assign starting I/O addresses to each module in 505 WorkShop, and any data you assign to these fields in COM PROFIBUS is overwritten.
Save the file.	Optional. Allows you to add or delete slaves from your configuration later, without re-entering all data into COM PROFIBUS.
Export binary file to 505 WorkShop.	Required. For convenience, you can select the disk drive and directory where your 505 WorkShop program is stored, and assign the binary file the same name as the program where you intend to use the I/O. An extension (.2BF) is automatically assigned to the binary file; do not change this extension.
Exit COM PROFIBUS.	You are ready to perform configuration tasks in 505 WorkShop.

Table 5.3

Importing Configuration Data From COM PROFIBUS Into 505 WorkShop

The 505 WorkShop Profibus-DP I/O dialog box allows you to import configuration data from the binary file that you created with COM PROFIBUS (described above). The steps you take are the same regardless of whether you are importing data from an entirely new configuration, or merely importing selected items that have been modified from a previous configuration session in COM PROFIBUS. The Merge function performs a comparison between the current configuration information and the contents of the selected binary file exported from COM PROFIBUS. Only new information is written to 505 WorkShop; pre-existing, unchanged configuration information (e.g., I/O addresses) is not lost.

To import and/or configure data from the COM PROFIBUS software, you must be running 505 WorkShop, and the binary file (with its .2BF extension) that you created with COM PROFIBUS must be in a known location. Follow the steps below to execute:

1. Select PLC Configuration from the PLC Utilities menu.
   Result: The PLC Configuration dialog box appears (Figure 5.2).
2. Select the **Profibus I/O** button under I/O configuration on the PLC Configuration dialog box.
   Result: The Profibus-DP I/O dialog box appears.

The Profibus-DP I/O dialog box dialog box Slave Status window shows the following information:

1. Slave numbers programmed from 1 to 112 or 1 to 32 for the 545L.
2. The Enabled state.
3. The Assigned state:
   - Yes, if all slave modules have I/O addresses assigned to them.
   - No, if some or all slave modules do not have I/O addresses assigned to them.
1. Online:
   - Yes, if in operate mode and PROFIBUS slave is connected.
   - No, if in stop mode or PROFIBUS slave is not connected.
2. Status:
   - Match, if the configuration and the I/O assignments match the online controller or the offline program.
   - Match, if the configuration matches the online controller or the offline program and the I/O assignments do not match online controller or the offline program.
   - New, if the slave is programmed in the .2bf file but not in the online controller or the offline program.
   - Mismatch, if the slave number exists in the .2bf file and in the online controller or the offline program, but has been configured differently.
   - Delete, if the slave is in the online controller or the offline program, but not in the .2bf file. When the program is written to the online controller or the offline program, the slave will be deleted.

The Profibus-DP I/O dialog box allows you to execute the following functions:

- Merge.
- Write All.
- Write Slave(s).
- Enable Slaves.
- operations (online only).
- Disable Slaves.
- Delete all slaves.
- Edit slave.
- Profibus

Merge COM PROFIBUS

Follow the steps below to merge in the COM PROFIBUS:

1. Click on the Merge button to execute the merge function. You are prompted to supply the name of the binary file from which you wish to import configuration data.
2. At the prompt, either type in the name of the binary file, or click the browse button to search other drives or directories.
   Result: The PROFIBUS-DP Slave Status screen displays configuration information obtained from the COM PROFIBUS binary file that you have just merged into 505 WorkShop.

NOTE: *The merged information has not been loaded into the controller or the offline program at this stage. You must first enter the I/O assignments, then use the* ***Write All*** *button on the Profibus -DP I/O dialog box to load the program.*

Updating Online/Offline Profibus Configuration

To update bus parameters and any previously-configured data slaves to the online controller or the offline program, you must use the Write All or Write Slave(s) button on the Profibus-DP I/O dialog box.

To write all configured slaves to the online controller or the offline program at one time, click on the Write All button.

To write selected slaves to the online controller or the offline program, click on the Write Slave(s). You can only Write Slave(s) with a Match or *Match under status in the slave status window.

Enable/Disable Profibus Slaves

To Enable or Disable a PROFIBUS slave status, you must use the Enable Slave(s) or Disable Slave(s) button on the Profibus-DP I/O dialog box.

Deleting or Adding Slaves

To add or delete slaves or modules, you must return to COM PROFIBUS, modify your configuration file as appropriate, and export it again. Follow the procedure outlined in Merge COM PROFIBUS to import the new information by running Merge. You can, however, delete all slaves by clicking on the Delete All button on the Profibus-DP I/O dialog box.

Assigning or Modifying Profibus I/O Addresses

To assign or modify the starting module addresses of your PROFIBUS-DP slaves, you must import either configuration data from the binary file that you created with COM PROFIBUS (described above); or connect to a 505 controller that contains PROFIBUS-DP slaves; or open an existing 505 WorkShop program from disk that contains PROFIBUS-DP slaves.

Follow the steps below to execute edit slaves:

1. Select PLC Configuration from the PLC Utilities menu.
   Result : The PLC Configuration dialog box appears (Figure 5.2).
2. Select the Profibus I/O button under I/O configuration on the PLC Configuration dialog box.
   Result : The Profibus-DP I/O dialog box appears (Figure 5.7).
3. Select the Edit S lave button on the Profibus-DP I/O dialog box.
   Result : The Edit Slave dialog box appears (Figure 5.8).

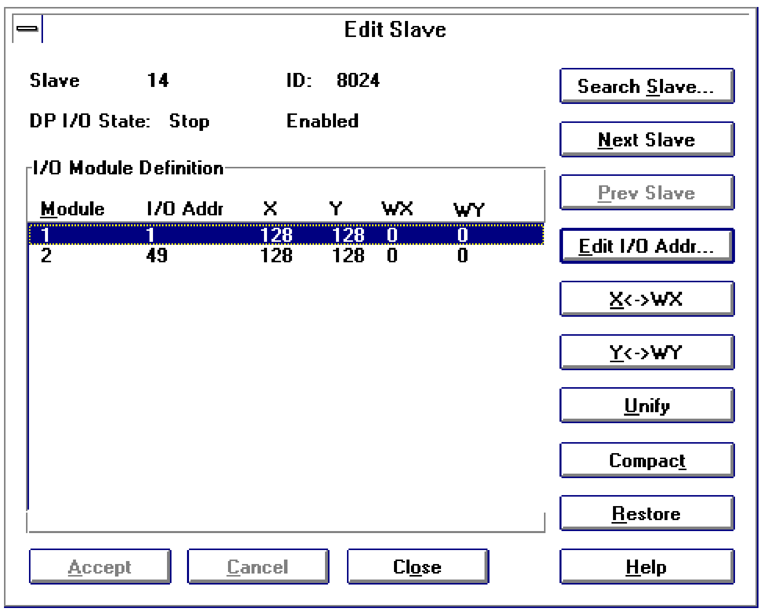


Figure 5.8

The Edit Slave dialog box allows you to perform the following tasks:

- The Search Slave button allows you to go to and edit other slave addresses without going back to the Profibus-DP I/O dialog box.
- The Next Slave button allows you to go to and edit the next slave address in numerical increasing order without going back to the Profibus-DP I/O dialog box.
- The P rev Slave button allows you to go to and edit the previous slave address in numerical decreasing order without going back to the Profibus-DP I/O dialog box.
- The Edit I/O Addr button allows you to change or program a slave's module's I/O address.
- The X<->WX button toggles the image register type for the selected module between discrete and word inputs. To ensure that discrete are toggled to words, the discrete must be on a 16-bit boundary.
- The Y<->WY button toggles the image register type for the selected module between discrete and word outputs. To ensure that discrete are toggled to words, the discrete must be on a 16-bit boundary.

- The Unify button moves all modules for the selected slave into the first module. Unification is only allowed if all modules are either discrete or words. Only the first address is retained.
- The Compact button moves all discrete image register types to word image register types for every module of the selected slave. It then unifies all the modules into the first module. To ensure that discrete are toggled to words, the discrete must be on a 16-bit boundary. Only the first address is retained.
- The Restore button restores modules for the selected slave according to the COM PROFIBUS configuration, including the image register types. Zero addresses are assumed for all but the first address, which is kept.

Before entering your I/O configuration, be aware of the following restrictions:

- For bit or bit-and-word modules (but not for word-only modules), the number for the I/O address must begin on an 8-point boundary. An 8-point boundary is (n * 8) + 1 , e.g., 1, 9, 17, etc. Addresses not starting on an 8-point boundary are changed to do so when you write the values.
- WorkShop does not flag duplicate I/O points. However, the Find I/O functions under the PLC Configuration dialog box can be used to search for duplicate I/O points.

To Edit the I/O Address:

1. Select the slave module number address to be edited by double- clicking on the module number, or by highlighting the module number and then clicking on the Edit I/O Addr. button.
2. Enter an I/O address that is a multiple of (*n**8) + 1.
3. After entering the I/O address click on the Accept button, then the Close.
4. Click on the Write All button to send the configuration for all the slaves to the online controller or the offline program. Or click the Write Slave(s) button to send the selected slave number configuration to the online controller or the offline program.

***NOTE**: The standard for PROFIBUS-DP I/O (DIN 19245, Part 3) stipulates certain bus parameters that cannot be changed while communications are in OPERATE mode. If you modified those bus parameters when you set up your configuration file in COM PROFIBUS, set communications to STOP mode before you execute Write All or Write Slave(s), in order to prevent bus errors.*

Profibus Operations

Profibus Operations allows you to change the operations mode from Operate to Stop, or Stop to Operate. You can also select from Synchronous to Asynchronous, or Asynchronous to Synchronous.

The Profibus Operations mode can be selected from the Profibus-DP I/O dialog box (Figure 5.7) by:

1. Clicking on the Profibus Ops. Button or selecting Profibus Operations from the PLC Utilities menu.
   Result: The Profibus Operations dialog box appears (Figure 5.9).

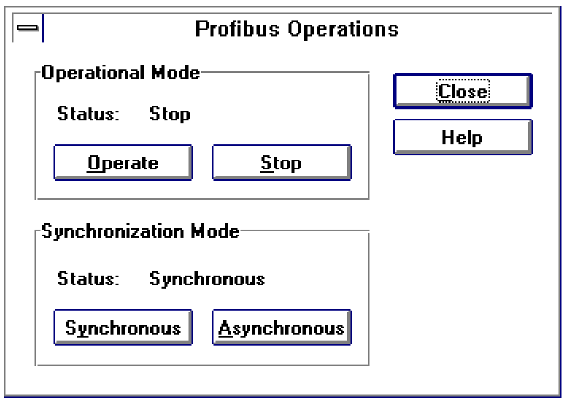


Figure 5.9

Find Configured I/O

To find configured I/O:

1. Select PLC Configuration from the PLC Utilities menu.
   Result: The PLC Configuration dialog box appears (Figure 5.2).
2. Select the **Find I/O** button under I/O configuration on the PLC Configuration dialog box.
   Result: The Find I/O Address dialog box appears (Figure 5.10).

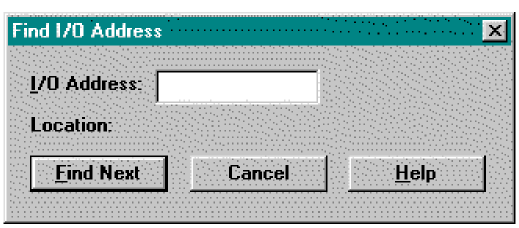


Figure 5.10

3. Enter an X, Y, WX, or WY address and click on the Find Next button.
4. Find will first search 505 I/O, then search PROFIBUS I/O. When the address is found, you can click on the Find Next button to find the next location of the address.

6 - Programming

Overview

In this chapter, you will discover how to enter and modify logic using 505 WorkShop's features. Remember, in the Windows environment there is usually more than one way to complete a task. 505 WorkShop provides keyboard support to access all commands. You may find that the most efficient method of programming will be through a combination of keyboard and mouse techniques.

Refer to Appendix A for a concise list of keyboard supported commands.

Online Versus Offline Programming

Before you begin programming, it is important that you understand the differences between programming online and programming offline.

While programming online, you are connected to a PLC, which may be running. Changing logic in one network may affect logic in another network. These changes may create unexpected or hazardous results.

	**Editing or modifying a program online may produce unexpected or hazardous results.**

In the online mode, PLC Status in Ladder can be displayed. However, some editing features are not available, including Cut and Paste.

Using the Logic Editor

The Logic Editor is where to begin programming since it allows you to display, access, and/or modify logic in the active logic program. You can view existing programs, or you can create a new one.

To view an existing program:

1. Open or Import a logic program (see open program).

2. If there is logic in the program but does not appear, click  on the toolbar or select Logic Editor from the View menu.

Programming Ladder

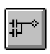
Several logic program windows can be displayed simultaneously. However, only one window is active at a time. In the active logic window, the ladder editor allows you to enter and modify ladder logic.

To make a logic window active, simply place the pointer on the desired window and click the left mouse button. From the keyboard, press Alt+W and press the number of the corresponding logic window. You will notice that the active window comes to the front of all other windows.

Insert a New Network

In the active logic program window, you can insert a new network using the mouse or the keyboard.

To insert a new network using the mouse:

1. Click  (new network button) on the Toolbar. Notice that the new network attaches to the pointer.
2. Bring the pointer into the active logic window. Position the pointer where you want the new network.
3. Click the left mouse button. The new network is inserted. For example, if the cursor is positioned in Network 002, the new network becomes Network 002 and existing Network (002) becomes 003.
4. Repeat Step 3 to insert additional networks.
5. Remove the new network from the pointer by clicking the arrow on the Instruction Bar.

To insert a new network using the keyboard:

1. Press Alt+P and the Program menu drops down.
2. Press S or use the down arrow keys to highlight New Network and press Enter. The New Network dialog box appears.
3. Enter the number of the network you wish to enter. If a network 002 exists and you enter 2 in the Network Number box, the existing Network 002 becomes Network 003 and the new network becomes 002.
4. Press Enter and the new network is inserted.
5. Repeat Step 3 to enter additional networks.

Insert a New Row

You can insert a new row to a network similar to inserting a new network. In the active logic program window, use the mouse or keyboard to place the cursor on an existing row in the network.
To enter a new row using the mouse:

1. Click  (new row) on the Toolbar. Notice that the new row attaches to the pointer.
2. Place the pointer in the position where you want the new row to appear.
3. Click the left mouse button and the new row is inserted.
4. Repeat Step 3 to insert additional rows.
5. Remove the new row from the pointer by clicking the arrow on the Instruction Bar.

To insert a new row using the keyboard:

1. Press the Insert key to turn the Ins mode on. See the Status Line near the bottom of the screen.
2. Use the arrow keys to position the parameter cursor (red or highlighted box) in the last row of the network.
3. Press Enter and the new row appears as the last row.
4. Repeat Step 3 to enter additional networks.

Ladder Instructions

You can enter instructions into your ladder logic program efficiently using either the Instruction Bar (mouse) or the Menu Bar (mouse and/or keyboard). See Chapter 6 of the *545/555/575 Programming Reference Manual* (Order Number: PPXSS505-8101-1) for all RLL instructions.

To enter instructions using the mouse and Instruction Bar:

1. Move the pointer to the Instruction bar. If the insruction or Item bar is
2. Click with the left mouse button on the desired instruction on the lower half of the Instruction Bar. If the desired instruction is not displayed, click the appropriate instruction group on the top half of the instruction bar.
3. Move the pointer over the logic program. Notice that the instruction is attached to the pointer.
4. Position the pointer where you wish to place the instruction.
5. Click the left mouse button, and the instruction is placed in that location. If an instruction cannot be placed in that location, an error message is displayed.
6. If the sticky cursor has been turned on in the Program Setup under General then the instruction will remain attached to the pointer. Click the left mouse button once for each additional instruction you want to insert. Click the arrow in the middle of the Instruction Bar or another instruction to remove the instruction from the pointer.
7. If the sticky cursor has not been turned on in the Program Setup then after the instruction has been dropped into place the pointer returns to an arrow.

To Enter Mnemonic Instructions with the Keyboard:

1. In the active logic program, position the cursor (by using the arrow keys) where the instruction is to be located.
2. Type in the instruction mnemonic at the cursor location and press ENTER. If you forget an instruction mnemonic just enter a ? and press enter. The following mnemonic pick list will be displayed. Pick the mnemonic you desire by double clicking on the mnemonic or arrow to the mnemonic and press ENTER. The mnemonic pick list will also display if an illegal mnemonic is entered.

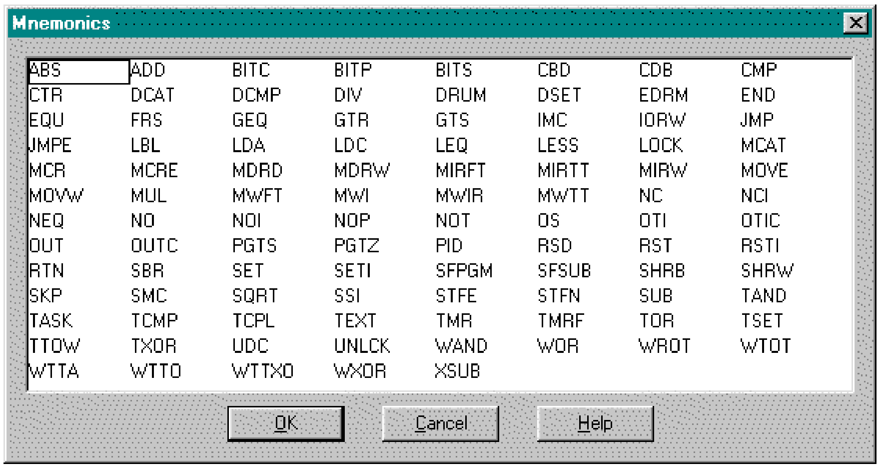


Figure 6.1

1. The following is a list of all 505 ladder instruction mnemonics:

Mnemonic	**Instruction**
ABS	Take absolute value of a word.
ADD	Addition.
BITC	Clears a specified bit.
BITP	Examines status of a specified bit.
BITS	Sets a specified bit.
J	Creates Down line
U	Creates Up line
H	Draws horizontal line
CBD	Converts binary to BCD value.
CDB	Converts BCD inputs to binary.
CMP	Compare.
CTR	Counts recurring events.
DCAT	Discrete control alarm timer.
DCMP	Compares current date with a specified date.
DIV	Division
DRUM	Simulates electro-mechanical stepper switch.
DSET	Sets date in real-time clock.
EDRUM	Simulates electro-mechanical stepper switch. Can be indexed by timer, event, or timer and event.
END	Unconditionally terminates a scan.
EQU	Equal to.
FRS	Force role swap
GEQ	Greater than or equal to.
GTR	Greater than.
GTS	Calls a subroutine.
IMC	Compares status of discrete points with a specified bit pattern in a set of patterns.
IORW	Does immediate read or write to discrete or word I/O.
JMP	Freezes outputs in zone of control.
JMPE	Freezes outputs in zone of control.
LBL	Selectively enable/disable program segments during scan.
LDA	Copies the logical address of a memory location into a memory location.
LDC	Copies the logical address of a memory location into a memory location.
LEQ	Less than or equal to.
LESS	Less than
LOCK	Used together with UNLOCK and provide a mechanism whereby multiple applications in the 575 system can coordinate access to

	shared resources.
MCAT	Motor control alarm timer.
MCR	Master control relay.
MCRE	Master control relay.
MDRMD	Drum; uses configurable mask to control coils.
MDRMW	Drum; uses configurable mask to write to words.
MIRF	Copies a table into the control relay memory or discrete image register.
MIRT	Copies status of control relays or discrete image register bits to table.
MIRW	Copies bit status from control relays or discrete image register to a word.
MOVE	Copies bytes, words, or long words from a source location to a destination location.
MOVW	Copies words from one location to another.
MUL	Multiplication
MWFT	Move word from table.
MWI	Copies words from one location to another using indexed addresses.
MWIR	Copies bits of a word to the discrete image register, or the control relay memory.
MWTT	Copies a word to a table.
NC	Normally closed contact.
NEQ	Not equal to.
NO	Normally open contact.
NOP	No operation.
NOT	Inverts power flow.
OS	Turns on output for a single scan.
OTI	Immediate coil
OTIC	Immediate closed coil
OUT	Coil
OUTNC	Coil Normally Closed .
PGTS	Calls an RLL subroutine and passes parameters to it.
PGTSZ	Calls an RLL subroutine and passes parameters to it. Discrete parameters indicated as outputs are cleared when the subroutine is not executed.
RET	
RSD	Transfers a PROFIBUS-DP slave's current diagnostic to user memory.
RST	Reset coil/bit.
RSTI	Immediate reset of a coil/bit.
SBR	Designates the beginning of an RLL subroutine.
RTN	Returns control from an RLL subroutine to the main RLL program.

SET	Set coil/bit.
SETI	Immediate set of a coil/bit.
SFPGM	Calls a special function program from RLL.
SFSUB	Calls a special function subroutine from RLL.
SHRB	Bit shift register.
SHRW	Word shift register.
SHT	
SKP	Selectively enable/disable program segments during scan.
SMC	Compares status of discrete points with a set of specified bit patterns.
SQRT	Square Root.
SSI	Scan synchronization inhibit
STFE	Searches for a word in a table equal to a specified word.
STFN	Searches for a word in a table not equal to a specified word.
SUB	Subtraction
TAND	ANDs the corresponding bits in two tables.
TASK	Start a new RLL program segment.
TCMP	Compares current time with a specified time.
TCPL	Inverts status of each bit in a table.
TEXT	Places textual information into L-Memory.
TMR	Times events.
TMRF	Times events.
TOR	ORs the corresponding bits in two tables.
TSET	Sets time in real-time clock.
TTOW	Copies a word from a table.
TXOR	Does an EXCLUSIVE OR on the corresponding bits in two tables.
UDC	Counts events up or down.
UNLCK	Used together with LOCK and provide a mechanism whereby multiple applications in the 575 system can coordinate access to shared resources.
WAND	Does logical bit-by-bit AND on two words.
WOR	Does logical bit-by-bit OR on two words.
WROT	Rotates the 4-segment bits of a word.
WTOT	Copies a word into a table.
WTTA	ANDs bits of a word with the bits of a word in a table.
WTTO	ORs bits of a word with the bits of a word in a table.
WTTXO	Does an EXCLUSIVE OR on the bits of a word with the bits of a word in a table.
WXOR	Does logical bit-by-bit EXCLUSIVE OR on two words.
XSUB	Calls an externally developed subroutine and passes parameters to it.

If the TISOFT mode is turned on in the program setup (see Ladder options under LOGIC tab in Program Setup) the following list of the function key alternatives and the CTRL/ALT functions are available.

Hot Keys

Key	**Comment**
Y	COILS Valid if cursor is in an output column
C	COILS Valid if cursor is in an output column
WY	COILS Valid if cursor is in an output column
V	COILS Valid if cursor is in an output column
G	COILS Valid if cursor is in an output column
W	COILS Valid if cursor is in an output column
X	CONTACTS Valid if cursor is in an input column
Y	CONTACTS Valid if cursor is in an input column
C	CONTACTS Valid if cursor is in an input column
WX/WY	CONTACTS Valid if cursor is in an input column
V	CONTACTS Valid if cursor is in an input column
K	CONTACTS Valid if cursor is in an input column
G	CONTACTS Valid if cursor is in an input column
W	CONTACTS Valid if cursor is in an input column
/	Open/closed contact/coil toggle
DEL	Delete current element
END	Moves to last column/row 1 of current rung
ENTER	Edit: moves to next row, current rung, or to next rung
	Display: moves to next row or rung
=	Equal-to Relational Contact
ESC	Deletes current address and puts you into edit mode for address.
TAB	Deletes current address and puts you into edit mode for address.
>	Greater-than-or-Equal-to Relational Contact
? or \	List mnemonics for current field
HOME	Moves to column 1, row 1 of current rung
H	Draws horizontal line
Key	Comment
N	Deletes horizontal line
I	Immediate contact and coil
INS	Insert Mode
<	Less-than Relational Contact
!	Not-Equal Relational Contact
O	Create/edit output coil
PgUp	Page Up
PgDn	Page Down

Ctrl L	Edits synonym/descriptor
U	UNFORCE
Ctrl U	Usage Table
J	Creates Down line
U	Creates Up line
M	Deletes vertical line

Using the SF Program and Subroutine Editor

The SF Program and Subroutine Editor gives you the ability to display, access, and/or modify program using special functions. See Chapter 7 of the *545/555/575 Programming Reference Manual* (Order Number: PPXSS505-8101-1) for all Special Function instructions.

To access the Special Function Program Editor:

3. Click Special Function Program Editor from the View menu (Alt+V, F). The Special Function Program dialog box appears (see Figure 6.1a).

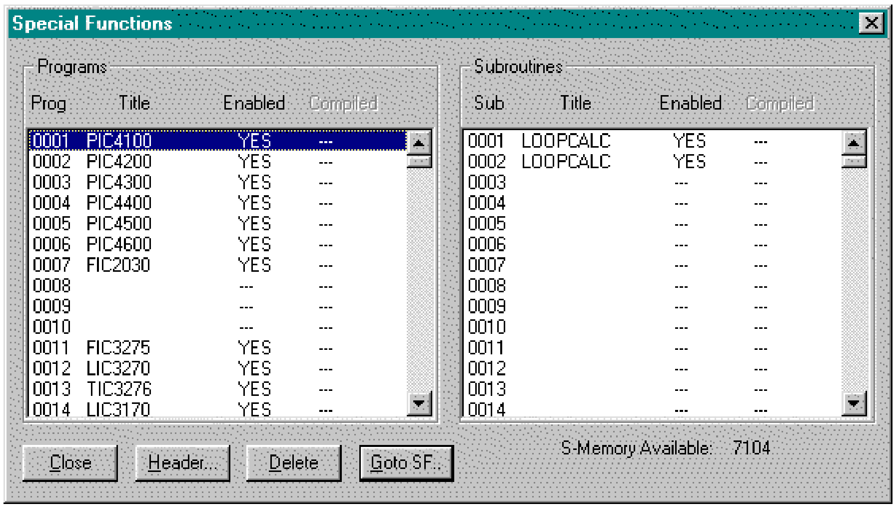


Figure 6.1a

4. Select the Special Function Program that you want to enter (Program 1, Program 2, etc.) by using the left mouse button or the Tab and Arrow keys on the keyboard. (Note: If a Header has already been programmed, skip ahead to item number 10.)

5. Now Click on the Header button and the following dialog box appears (see Figure 6.2).

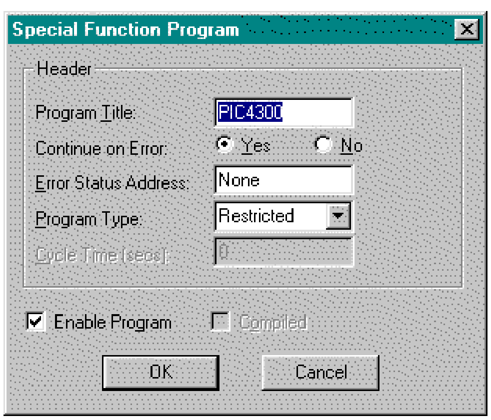


Figure 6.2

6. Enter a title for the program in the Program Title window. The title is optional and can be left blank.

7. Click Yes or No in the Continue On Error selection to have the program continue or stop when an error occurs.

8. Select the Program Type from the Combo Box.

9. Enter the cycle time in seconds (0.5-6553.5).
10. Check the Enable Program to allow exaction of the SF program.You can also enable and disable the program from within the SF editor by clicking Shift-F2.
11. For the 555-1105/-1106 and 575-2105/-2106 CPUs, the compile mode is selectable. When compiled mode is selected, the SF program or subroutine is translated to the native instruction set of the CPU's microprocessor. The compiled code is then executed whenever the program or subroutine is scheduled for execution. The advantages of compiled execution are:
    - Significant execution speed improvement. For example, a MATH statement that adds two floating-point values will execute in less than 10 µs when compiled, versus more than 100 µs when executed by the SF interpreter. Depending on the program's size and the placement of the target LABEL within the program, a GOTO statement may take 1 ms or more when executed by the interpreter. Compiled execution of a GOTO statement takes less than 1 microseconds no matter where in the program the LABEL is located. This represents a 1,000x improvement.
    - A compiled SF program or subroutine can be executed in-line to the user RLL program. This means that when the enable input to the SFPGM or SFSUB box instruction is on, the program or subroutine is executed immediately and its result is available for use in the next rung of the current RLL scan.
12. There are several reasons to choose interpreted mode for a SF program. The primary reasons are as follows:
    - If the program has one or more statements that are not allowed in a compiled program, or if it calls a subroutine that is not compiled, then it may not be compiled.
    - A compiled program requires both S-Memory and Compiled Special (CS)-Memory, while an interpreted program requires only S-Memory. As a rule of thumb, the compiled code for a SF program requires twice as much CS-Memory as S-Memory. For example, a SF program that uses 1 Kbyte of S-Memory also uses 2 Kbytes of CS-Memory.
    - A second SF program or subroutine on the same execution queue cannot pre-empt a compiled SF program or subroutine. This may present a scheduling problem for a cyclic, loop, or analog alarm queue. For example, if a compiled program is executing on a loop setpoint, a higher priority loop will not execute until the compiled program completes. This is not a problem if the program's execution time is small. However, if the program requires significant execution time, this could cause unnecessary loop overruns.

NOTE: *Most SF programs and subroutines can be compiled. However, an SF program or subroutine which contains any of the following instructions cannot be compiled: The data compacting instructions: PACK, PACKLOOP, PACKRS, and PACKAA The shift register instructions: SSR, FTSR–IN and FTSR–OUT The PRINT instruction The BCD instructions: BCDBIN and BINBCD Additionally, the CDT and SDT instructions, when used in a compiled SF program or subroutine, must specify a static table; that is, the table's base address must be a V, K, G, VMS, or VMM address and the table's size must be specified as a constant.*

13. Save Header changes and return to the SF Programs Subroutines dialog box by clicking on OK. To disregard changes and return to the SF Programs Subroutines dialog box, click Cancel.

Insert a New Row

You can insert a new row to a Special Function Program similar to inserting a new network. In the active logic program window, use the mouse, or keyboard to place the cursor on an existing row in the Special Function Program.

To enter a new row using the mouse:

1. Click (new row) on the Toolbar. Notice that the new row attaches to the pointer.
2. Place the pointer in the position where you want the new row to appear.
3. Click the left mouse button, and the new row is inserted.
4. Repeat Step 3 to insert additional rows.
5. Remove the new row from the pointer by clicking the arrow on the Instruction Bar.

To insert a new row using the keyboard:

1. Press the Insert key to turn the Ins mode on. See the Status Line near the bottom of the screen.
2. Use the arrow keys to position the parameter cursor (red or highlighted box) in the last row of the Special Function Program.
3. Press Enter and the new row appears as the last row.
4. Repeat Step 3 to enter additional rows.

Entering Special Function Program Instructions

You can enter instructions into your Special Function Program efficiently using either the Instruction Bar or Menu Bar.

To enter instructions using the Instruction Bar:

1. Move the pointer to the Instruction bar.
2. Click with the left mouse button on the desired instruction on the lower half of the Instruction Bar. If the desired instruction is not displayed, click the appropriate instruction group on the top half of the instruction bar.
3. Move the pointer over the Special Function Program. Notice that the instruction is attached to the pointer.
4. Position the pointer where you wish to place the instruction.
5. Click the left mouse button, and the instruction is placed in that location. If an instruction cannot be placed in that location, an error message is displayed.
6. The instruction remains attached to the pointer. Click the left mouse button once for each additional instruction you want to insert. Click the arrow in the middle of the Instruction Bar or another instruction to remove the instruction from the pointer.

Using the menu to enter instructions with the keyboard:

1. In the active Special Function Program, use the arrow keys to move the cursor to the spot you wish to place the instruction.
2. Type in the character mnemonic for the instruction or Press Alt+P, I to open the Selection Instruction dialog box from the Program menu.
3. Use the arrow keys to move up and down the Group and Instruction scroll boxes, and the Tab key to move between the boxes.
4. Highlight desired instruction, and press Enter to insert the instruction.
5. Repeat Steps 1 through 5 for each instruction you want to enter.

Editing Logic

505 WorkShop uses a number of time-saving editing features to help you complete your programming tasks. These include:

- Cut
- Copy
- Paste
- Paste with Rewire
- Insert
- Clear
- Delete

These editing features are accessed through the Edit menu and keyboard commands. Please see Chapter 3, 505 WorkShop Basics, for a more complete description of the editing features.

Cut, Copy, and Paste are probably the most frequently used editing features. The list below describes Cut, Copy, and Paste differences.

Cut	Removes the selection from the program and places it in the clipboard.
Copy	Copies the selection and places it in the clipboard.
Paste	Inserts clipboard contents into the program at the cursor location.
Paste With Rewire	Inserts clipboard contents into the program at the cursor location and allows the user to re-address any addressable items contained in the clipboard. You can also copy tags and descriptions.

Cut

To use the cut feature:

1. Select the information you want to cut. You can cut parts of a network, all of a network or multiple networks.
2. Cut your selection to the clipboard with one of the following:

- Click on the toolbar.
- Select Cut from the Edit menu.
- Press Ctrl+X.

Copy

To use the copy feature:

1. Select the information you want to cut. You can cut parts of a network, all of a network or multiple networks.
2. Copy your selection onto the clipboard with one of the following:

- Click on the toolbar.
- Select Copy from the Edit menu.
- Press Ctrl+C.

Paste

To access the paste feature:

1. Move the cursor to the desired location.
2. Paste clipboard contents into the new location with one of the following:

- Click  on the toolbar.
- Select Paste from the Edit menu.
- Press Ctrl+V.

NOTE: *When pasting, clipboard contents are inserted before existing items. For example, if you are pasting a network and the cursor is positioned at Network 2, click paste and the clipboard contents become Network 2. The previous Network 2 becomes Network 3.*

Paste With Rewire

Paste With Rewire allows you to past the contents of the clipboard and change the elements address at the same time.

To access the rewire feature:

1. Move the cursor to the desired location.
2. Paste clipboard contents into the new location.
3. Select Paste with Rewire from the Edit menu.
4. Choose the appropriate options. You can select the number of copies you wish to copy and/or you can select to offset each address by a certain value. If the addresses you select have descriptions or tags associated with them, you can choose to paste those also.

NOTE: *When pasting with rewire, valid addresses for instructions are not checked until you "Validate and Enter Logic. Validate and Enter "*

5. Choose OK.

NOTE: *When pasting with rewire, clipboard contents are inserted before existing items. For example, if you are pasting a network and the cursor is positioned at Network 2, click paste and the clipboard contents become Network 2. The previous Network 2 becomes Network 3.*

Clear

Use Clear to clear an item without removing the space it occupies.

To clear an item or items:

1. Select the item or items you want to clear by clicking, holding and dragging the pointer over the desired logic.
2. Select Clear from the Edit menu or press the Delete key.
3. Click the items you want to clear.
4. Click OK or press Enter and the selected items are cleared.

	**Editing or modifying a program online may produce unexpected or hazardous results.**

Delete

Use Delete to clear an item and remove the space it occupies.

To delete:

1. Select the item or items you wish to delete.
2. Select Delete from the Edit menu. The Delete box appears.
3. Click the items you want to delete.
4. Click OK or press Enter.

Insert

Use Insert to insert a selected object (network, row or column) at the point of the current cursor position.

To insert an object:

1. 1. Select Insert from the Edit menu. The Insert dialog box will appear.
2. 2. Click on the object you want to insert.
3. 3. Click OK or press Enter.

Right Mouse Button Pop-Up Menus

The menu items available with the right mouse button are as follows:

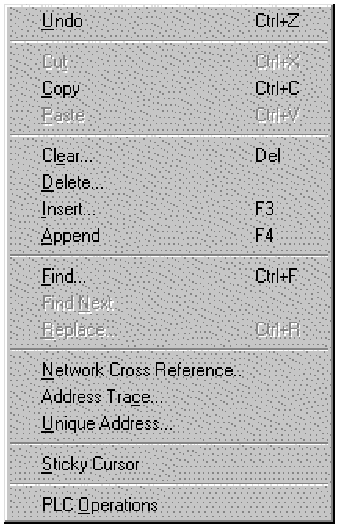


Figure 6.5

Validate and Enter

While programming in online or offline mode, logic must be validated and entered before it can be saved to disk or transferred online. To validate and check your logic:

1. Press function key F8, click  on the toolbar or select Validate and Enter from the Program menu.
2. The message "Validating and Enter" appears on the screen.
3. After logic is validated and entered and, if necessary, problems fixed, the logic program can be saved or transferred online.

Finding Logic

You can use the Find command on the Edit menu to search for the following functions within the active logic window.

- Find elements in ladder, SF programs, and SF subroutines with designated elements and identifiers.
- Find a Specific network number, network address, address/tag, instruction, or instruction number.

Use the following steps to conduct a search for the desired item.

Find a Network, Network Address or Line Number

4. Click on the toolbar or select Find from the Edit menu (Ctrl+F), and the Find dialog box appears (Figure 6.6).

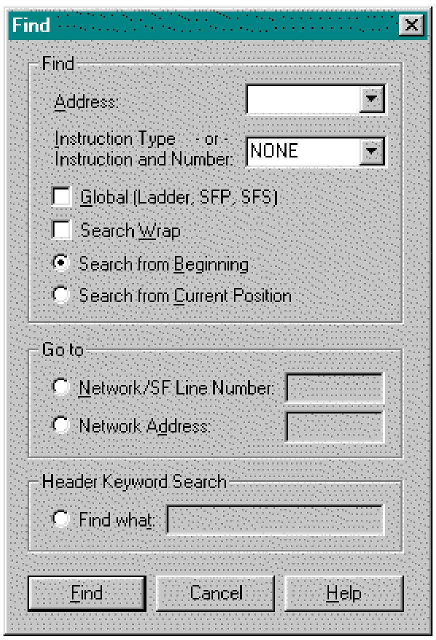


Figure 6.6

5. In the Go to group, select the **Network/SF Line Number** radial button to find a specific network number or special function line number. Select the **Network Address** radial button to find a specific network address.
6. In the **Network/SF Line Number** or **Network Address** enter box; type the number of the network, network address or line number you wish to find.
7. Click **Find** to begin the search. If the item is found, the Find box disappears, and the cursor moves to the network, network address or line number you wanted to find.
8. Click Cancel to disregard the search and return to the active logic program.

Find an Instruction, Address or Tag

1. Click on the toolbar or select Find from the Edit menu (Ctrl+F), and the Find dialog box appears (Figure 6.6).
2. By default, 505 WorkShop only searches the current program environment for the Instruction, Address or Tag entered. To search the entire program the Global (Ladder, SFP, SFS) check box must be selected.
3. In the Search box, identify where you want to start the search from, either the Beginning of the program or Current position of the cursor. If the Search Wrap check box is not checked. The search will start from the defined starting position to the end of logic. If it is checked the search will wrap around to the starting position.
4. In the Address box, type the address or tag you wish to find. If no address or tag is entered, the Find will locate all instruction identified in the Instruction selection box.

NOTE: *The last five address/tags searched for are saved. Click on the down arrow within the address field to display and/or select them.*

5. In the Instruction field, select the instruction group to search on:
   - Contact, coil, timer, etc. for ladder.

   If no instruction is selected, the Find will locate all instruction items with the selected address or tag.
6. Click OK to begin the search. If the item is found, the Find box disappears, and the cursor is positioned over the address.
7. Click Cancel to disregard the search and return to the active logic program.

Find Documentation String

The Keyword Search, checks each Tag, Description, and Documentation Comment for the text string entered. The text string can be up to 100 characters in length.

1. Click on the toolbar or select Find from the Edit menu (Ctrl+F), and the Find dialog box appears (Figure 6.6).
2. In the Header Keyword Search group, select the **Find What** radio button.
3. In the Find What box, type the text you want to find. Then chose the Find button to begin the search.

Find Next

Click on the toolbar or select Find Next on the Edit menu to find the next occurrence of an address or tag.

Search and Replace Address

You can use the Replace command on the Edit menu to search for a specified address and replace them with another address. The following logic windows support Search and Replace by address:

- SF programs
- SF subroutines
- Ladder Editor
- Alarms
- PID Loops

Use the following steps to conduct a Search and Replace Address:

1. From the Edit menu (Ctrl+R), select **Replace**, and the Replace dialog box appears (Figure 6.6a).

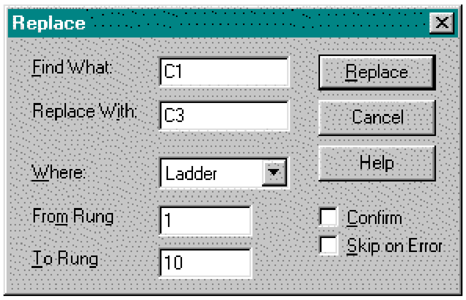


Figure 6.6a

2. Specify the address number to be replaced in the Find What field and the address replacing it in the Replace With field. For example, to replace C1 with C2, simply type these two address numbers in the Find What and Replace With field respectively. With this setting, all occurrences of address C1 are Changed to address C2 for the selected logic type range.

3. Specify the logic type by clicking on the drop-down box in the **Where** field and selecting one of the 5 logic types.

4. In the From Rung To Rung fields, specify the search range for the replacement address. If no range is selected, the software defaults to the total number of networks/lines in the target logic type.

5. **Confirm**. When the Confirm check box is not chosen on the Replace dialog box, the program automatically changes all Find What addresses to Replace With addresses. When the Confirm check box is chosen on the Replace dialog box, the program displays each address where the Replace is to occur. The address to be replaced is surrounded by a red outlined rectangle and the following options are displayed:
   - Click the **Replace** button to perform the specified replacement.
   - Click the **Skip** button to skip the address and go on to the next item to be replaced.
   - Click the **Stop** button to Cancel the Replace operation and return to the specified logic window.

6. **Skip on Error**. When the Skip on Error's check box is chosen on the Replace dialog box. The program will skip illegal replacements and go on to the next item to be replaced automatically. If skip is not chosen the program will flag illegal replacements and the following options are displayed:
   - Click the **Skip** button to skip the address and go on to the next item to be replaced.
   - Click the **Stop** button to Cancel the Replace operation and return to the specified logic window.

Search and Replace Table

The "Search and Replace Table" feature expands the capabilities of the original "Replace" feature, illustrated below.

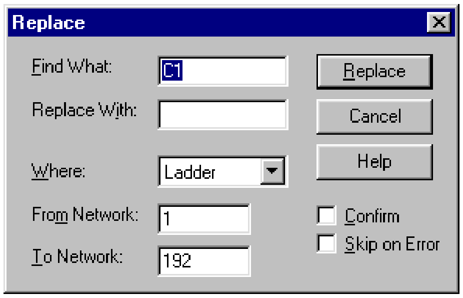


Figure 6.6a

The original "Replace" feature can search and replace only one address at a time. The "Search and Replace Table", seen below, allows up to ***32 address ranges*** to be searched and replaced in one continuous process within an offline PLC program.

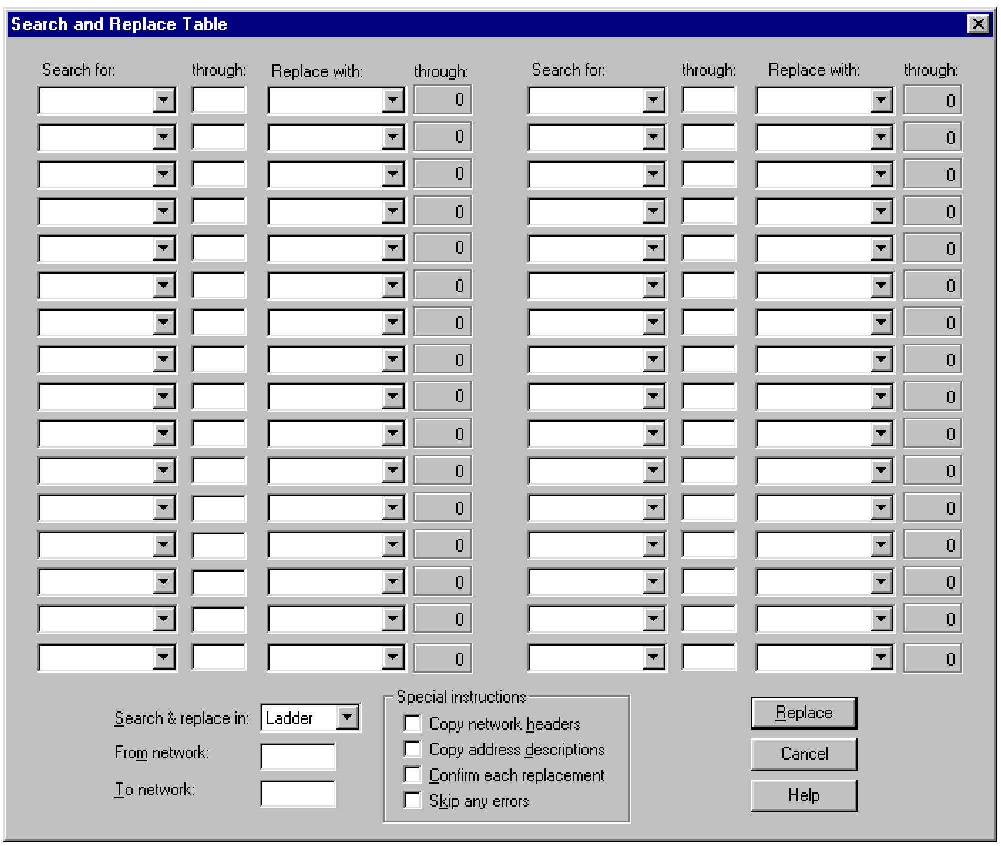


Figure 6.6b

The "Search and Replace Table" Dialog

Select "Search & Replace Table…" from the "Edit" menu to display the Search and Replace Table dialog below.

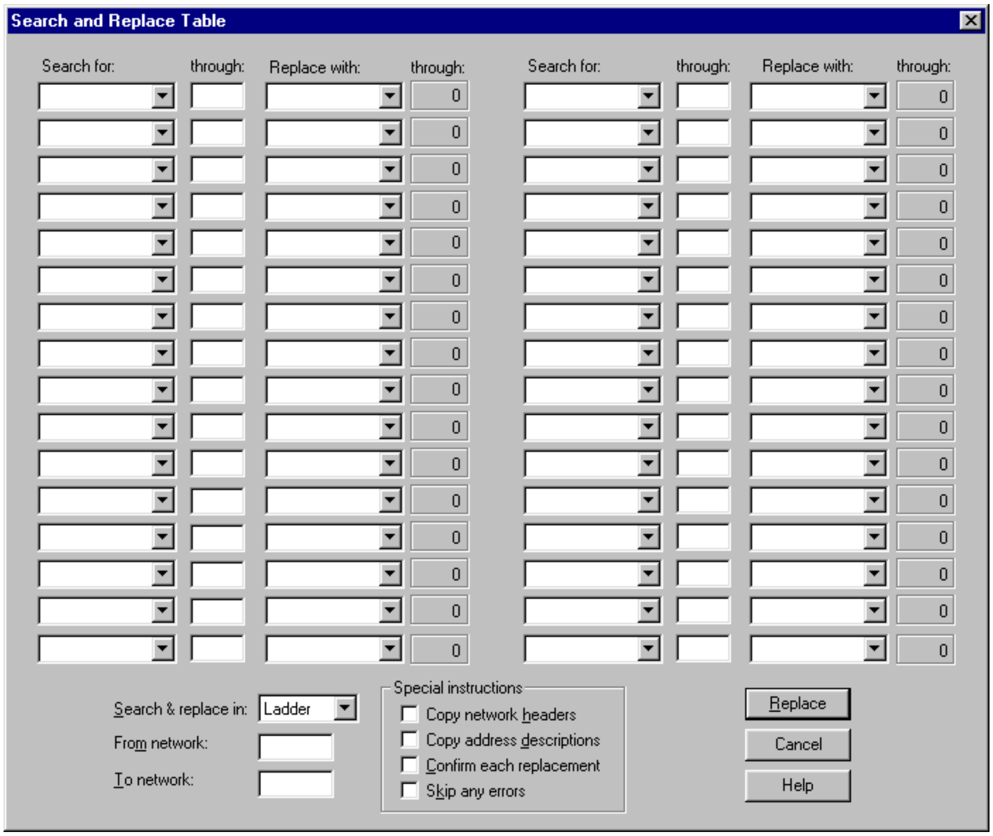


Figure 6.6c

The "Search for" Combo Boxes

Each "Search for" combo box specifies the address (type and number) from which that search and replace operation will be performed. The drop lists in these combo boxes contain prefixes of all valid address and instruction types. Enter a number after the address prefix to designate the starting point of the address range to be searched and replaced.

For example, the partial drop list may contain the following address prefixes:

X

Y
C
V

K
WX
WY

.

.

.

Select address prefix **C** and enter **12** after it. This indicates the starting address of this search and replace range is **C12**.

The Search for "through" Edit Boxes

The numeric values entered in the Search for "through" edit boxes specify the address at which the search and replace range ends. This value must be greater than or equal to the address number entered in the "Search for" combo box.

Continuing the example above, entering **47** in the Search for "through" edit box indicates this search range starts with address **C12** and ends with address **C47**.

The "Replace with" Combo Boxes

The "Replace with" combo box specifies the address (type and number) which will replace the address range entered in the "Search for" combo box.

The items in the drop list are based upon the address type chosen in the "Search for" combo box. If a bit address type was selected in the "Search for" combo box, the "Replace with" drop list contains only bit address types. Similarly, if a WORD address type was selected in the "Search for" combo box, the "Replace with" drop list contains only WORD address types.

But when box instructions are selected in the "Search for" combo box, an instruction of the same type only is allowed in the "Replace with" combo box. So if an **EDRUM** instruction is entered in the "Search for" combo box, only an **EDRUM** instruction can be entered in the "Replace with" combo box.

Continuing the example above, entering **X4** in the "Replace with" combo box indicates the "Search for" addresses starting with **C1** will be replaced with a range of addresses starting with **X4**.

The Replace with "through" Static Boxes

The numbers displayed in the Replace with "through" static boxes are automatically calculated based upon the ranges entered in the "Search for" and its "through" boxes.

Continuing the example above, the "Search for" range starts at **C12** and ends at **C47**. This is a range of **47 – 12 = 35**, or **35** addresses of the **C** type. Since the starting address entered in the "Replace with" combo box is **X4**, the last address in the replace range is **4 + 35 = 39**, or **X39**. The number **39** is displayed in this Replace with "through" static box.

When the search and replace process begins, any occurrence of **C12** is replaced with **X4**. Any **C13** is replaced with **X5,** and so on as illustrated below:

C12 is replaced with **X4**

C13 is replaced with **X5**

C14 is replaced with **X6**

C15 is replaced with **X7**

C16 is replaced with **X8**

.

.

.

C45 is replaced with **X37**

C46 is replaced with **X38**

C47 is replaced with **X39**

Performing Multiple Search and Replace Operations

Any or all of the 32 "Search for" and "Replace with" ranges can be used to perform multiple search and replace operations in one continuous process. However, once an address is changed during a given process, it will not be changed again even when another "Search for" and "Replace with" range is set to change the new address.

For example, one search and replace range is set to change **C1** to **X4**. A following search and replace range is set to change **X4** to **B19**. The search and replace process will not perform both changes, **C1** to **X4** then change **X4** to **B19**. Instead, the first change, **C1** to **X4**, occurs. Then the process moves to the next address in the PLC program and determines if it will be replaced.

The "Search & replace in" Combo Box

The search and replace operation can be performed on 5 isolated sections of the PLC program; Ladder, SFS (Special Function Subroutines), SFP (Special Function Programs), Alarms and PID Loops.

Use the "Search & replace in" combo box to select the desired section of the PLC program in which to perform the search and replace operation.

The "From" and "To" Edit Boxes

A search range within the program area selected in the "Search & replace in" combo box can be entered in the "From" and "To" edit boxes.

As the "Search & replace in" selection changes, the default values displayed in the "From" and "To" edit boxes include the entire range available in that PLC program section.

For example, when "Ladder" is selected, the default range displayed starts at network **1** and ends at the last network in the ladder.

The "Special Instructions" Group Box

Additional actions can be performed as the search and replace operation executes. These actions can be selected within the "Special Instructions" group box. These actions are available when "Ladder", "SFS" or "SFP" only are selected in the "Search & replace in" combo box.

The "Copy network headers" Check Box

A network can be documented with a network header. This network header is actually associated with the address of the network’s output coil. When the search and replace process changes the address of a network’s coil, the original network header is no longer tied to that network.

There are two options in this situation. Either the network header (if any) associated with the coil’s new address replaces the network’s previous header. Or the network header associated with the coil’s original address is copied to the coil’s new address.

As an example of the first option, the address of a network’s output coil is **C1**. The network header associated with address **C1** is " **PROCESS 1 INDICATOR LIGHT** ". The search and replace operation changes **C1** to **X44**. The output coil of this network is now address **X44**. Address **X44** has no network header associated with it. As a result, this particular network loses its network header.

In the second option, when the search and replace operation changes the network output coil address from **C1** to **X44**, the network header " **PROCESS 1 INDICATOR LIGHT** " is copied from address **C1** to address **X44**. This allows the network to retain its original network header.

To select the first option, do not check the "Copy network header" check box.

To select the second option, check the "Copy network header" check box.

The "Copy address descriptions" Check Box

The documentation of each address includes the Tag, Description and Description Comment. The Description portion of the original address can be copied to the address which replaces it.

Check the "Copy address descriptions" check box to copy the original address's Description to the address which replaces it.

The "Confirm each replacement" Check Box

Check the "Confirm each replacement" check box to force the search and replace process to stop and request user confirmation before performing each replacement.

Or uncheck this box to allow the search and replace process to proceed without pausing for user confirmation.

The "Skip any errors" Check Box

Some replacements may be invalid, such as replacing addresses which are beyond configured limits or selecting instructions which are not available for the PLC type. When these errors occur, search and replace may be set to stop, display the error and request user confirmation before proceeding.

Alternately, search and replace may ignore any errors and continue its process without pausing for user confirmation.

Check the "Skip any errors" check box to allow search and replace to continue past any errors without requesting user confirmation.

The "Replace" Button

When the desired "Search for" and "Replace with" ranges have been entered, click the "Replace" button to start the search and replace process.

The "Cancel" Button

Click the "Cancel" button to exit the Search and Replace Table dialog without performing the search and replace process.

The "Help" Button

Click the "Help" button to display general help text about the Search and Replace Table dialog.

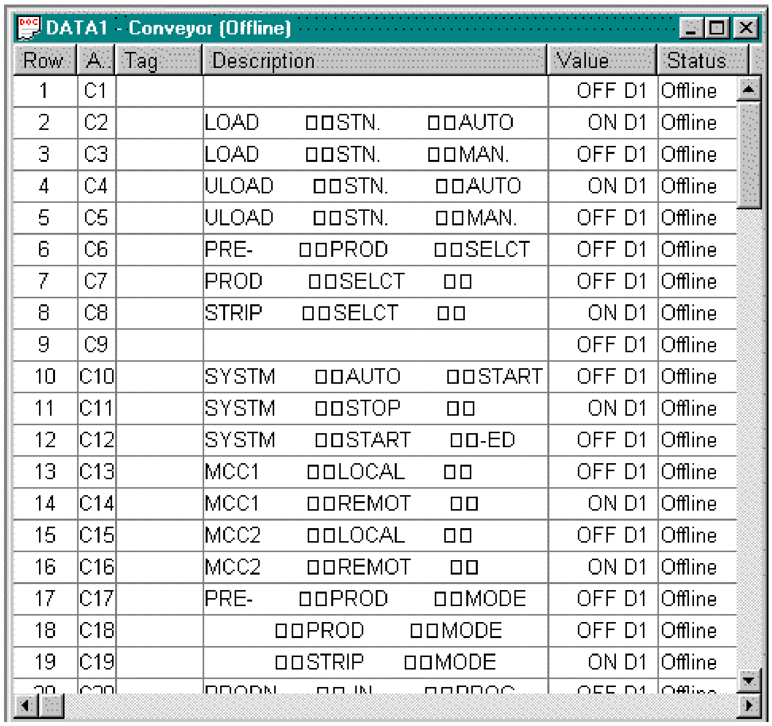
Using the Data Window

The Data Window menu item opens a new window and allows you to view and change the data values of your program.

Once you display addresses in the Data Window, you can enter specific values. If monitoring data online, you can enter a value for a particular address, send it directly to the processor, and you can force I/O address on and off.

You can also create tables of addresses, store them on a disk, and transfer them to the PLC at a later time.

1. With a logic program open, click on the toolbar or select Data Window from the View menu. The Data Window dialog box appears (Figure 6.7).
2. Use the Program Setup from the Options menu or the right mouse menu to change which columns are displayed in the Data Window. See *Program Setup* in Chapter 4.



DATA1 - Conveyor (Offline)

Row	A..	Tag	Description	Value	Status
1	C1			OFF D1	Offline
2	C2		LOAD □□STN. □□AUTO	ON D1	Offline
3	C3		LOAD □□STN. □□MAN.	OFF D1	Offline
4	C4		ULOAD □□STN. □□AUTO	ON D1	Offline
5	C5		ULOAD □□STN. □□MAN.	OFF D1	Offline
6	C6		PRE- □□PROD □□SELCT	OFF D1	Offline
7	C7		PROD □□SELCT □□	OFF D1	Offline
8	C8		STRIP □□SELCT □□	ON D1	Offline
9	C9			OFF D1	Offline
10	C10		SYSTM □□AUTO □□START	OFF D1	Offline
11	C11		SYSTM □□STOP □□	ON D1	Offline
12	C12		SYSTM □□START □□-ED	OFF D1	Offline
13	C13		MCC1 □□LOCAL □□	OFF D1	Offline
14	C14		MCC1 □□REMOT □□	ON D1	Offline
15	C15		MCC2 □□LOCAL □□	OFF D1	Offline
16	C16		MCC2 □□REMOT □□	ON D1	Offline
17	C17		PRE- □□PROD □□MODE	OFF D1	Offline
18	C18		□□PROD □□MODE	OFF D1	Offline
19	C19		□□STRIP □□MODE	ON D1	Offline

Figure 6.7

3. Press Enter to accept the address/tag.
4. Press Ctrl L on the Description or Tag column to bring up the documentation editor.
5. Type an address or tag in the address field.
6. Repeat Steps 2 and 3 for each value you want to view. After entering a value, use Next Address (F5) to move down one cell and fill in the next address or tag, or Previous Address (F6) to move down one cell and fill in the previous address or tag. Both Next Address and Previous Address are located in the Data menu or by clicking the right mouse button.
7. Select Clear from the Edit menu if you want to clear a row or rows. Select Delete from the Edit menu to delete a row or rows; subsequent rows will move up.
8. Windows selection techniques can be employed here. See Chapter 3,"*Some Things You Should Know About Windows.*"

9. You can create tables of addresses, store them to a file, and load them again offline or online. To save a list of addresses, select Save Template from the Data menu. To load a list of addresses, select Load Template from the Data menu.
10. Select On/Off from the right mouse menu to set or reset bit addresses.
11. Select Force On/Off from the right mouse menu to force on or off bit addresses.
12. Select Force Word from the right mouse menu to force a word address.
13. Select Clear Force from the right mouse menu to clear the forced address.
14. Use Clear All Forces from the right mouse menu to clear all forced addresses.
15. Press Show forces from the right mouse menu to display all forced addresses. You will be prompted to enter a starting address, and any addresses that are forced will be displayed.
16. Press Status On/Off from the right mouse menu to turn off the status updates.
17. Select Fill from the right mouse menu to enter addresses from the current location with the next addresses, until the end of the window is reached. For instance, if your current address location is 3000 in the fifth row, pressing Fill will enter the addresses starting in row 6 with address 30002, etc.
18. Double-click the Control Menu box in the upper left corner of the active window to exit the Data Window.

Customizing the Display

The Data Window can be customized to display Tags, Descriptions, both Tags and Descriptions, or neither. These columns can be displayed if selected in the Program Setup.

To modify the display, select Options, Program Setup (Alt+O, P). In the Data Window Tab dialog section, click on Tag to include tags and/or click Descriptions to include Descriptions. De-select both if you do not want to include Tags and Descriptions. For more Data Window display options See Program Setup in Chapter 4.

Data Format

1. Data Format is available for byte, word and double word addresses. Select Format from the Data menu while in the Data Window. The Data menu is available only while in the Data Window. The Format dialog box appears (Figure 6.8).

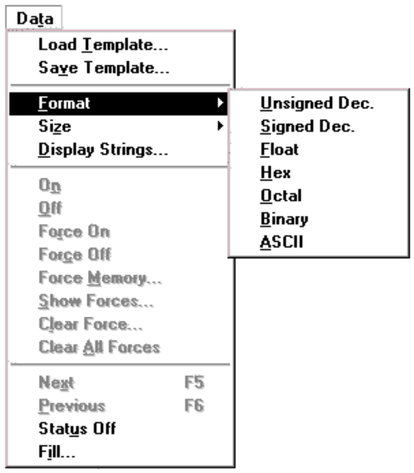


Figure 6.8

2. Click the desired data format. For the String field, you can enter the number of characters to be displayed (from 1-64).
3. Click OK to select the format and return to the Data Window.

Forcing an Element

As a troubleshooting tool, control relays can be forced. The force attribute bit, also shown in Figure 6.8a, provides a single-bit memory location for storing the forced status of control relays. If a control relay has been forced, the control relay retains that forced status during a power cycle as long as the battery is good.

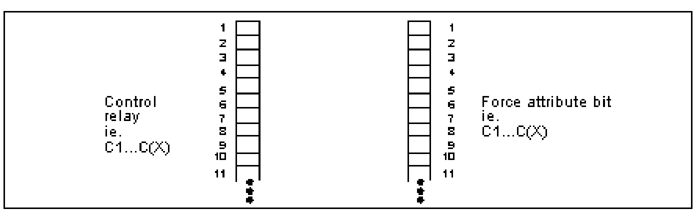


Figure 6.8a

You can force an element from either Data Window or Ladder logic editor complete the following steps.

From the Data Window:

1. Click on the right mouse button while in the Data field of an Element or the Data menu while in the Data field of an Element.
2. From the menu select the type of forcing action you would like to do.

From the Ladder Logic:

1. 1. From the Ladder Editor select the element to force.
2. 2. While on the element select the Diagnostics menu. From the menu select the type of forcing action you would like to do.

Show Forces

To show all forced elements complete the following steps.

1. Select the Data Window from the Veiw menu.
2. Once in the Data Window click on the right mouse button or left click on the Data menu.
3. Select Show Forces from the menu.

Clear a Forced Element

To Clear a Forced Element from either Data Window or Ladder logic editor complete the following steps.

From the Data Window:

1. Click on the right mouse button while in the Data field of an Element or the Data menu while in the Data field of an Element.
2. From the menu select the type of forcing action you would like to do.

From the Ladder Logic:

1. While on the element select the Diagnostics menu. From the menu select the type of forcing action you would like to do.
2. From the Ladder Editor select the element to force.

Complete Cross Reference

In both online (network mode only) and offline programming, you can track addresses in a program by viewing the Cross Reference table.

1. Click on the toolbar or selectCross Reference from theView menu. The Cross Reference window appears, Figure 6.8b. If you have more than one program loaded, the information displayed is for the program in the active window. The cross reference can be based on address or networks by checking the desired radio button on the cross-reference dialog box.

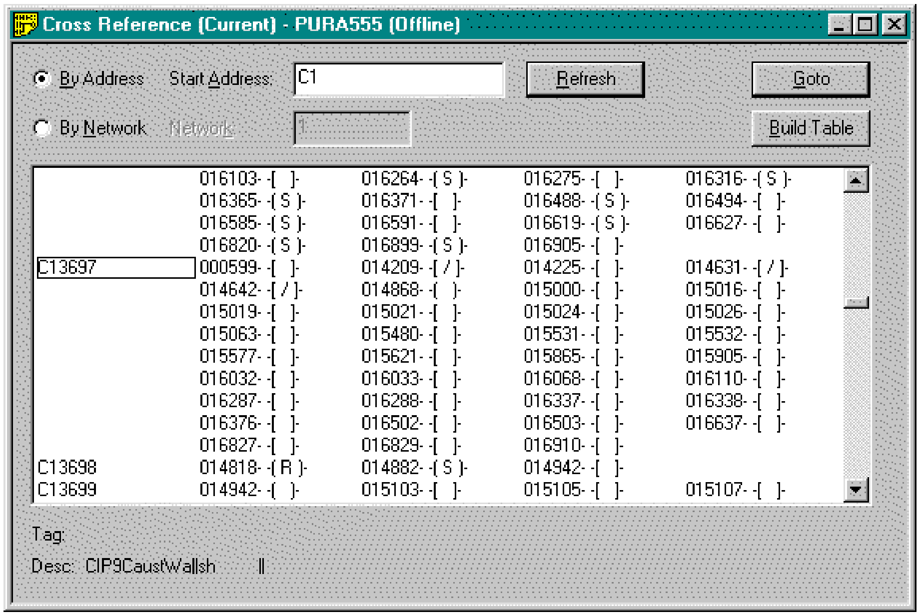


Figure 6.8b

NOTE *Only validated and entered logic is considered part of the current program. Thus, logic not entered and validated will not show in the Cross Reference Table. See Validate and Enter Logic in this chapter.*

---

2. Type in the starting address in the corresponding address box or network number in the corresponding network box, if selected. Click on the Refresh button. The cross-reference will display the selected information in the view window. The address is displayed on the left side with the network number next to the element. The Tag and Description of the selected address are displayed on the bottom left of the screen.

3. Select the logic item you want to view from the view window. Click the Goto button to jump to the first occurrence of the address you have highlighted or you can double click on the item to do the same.

NOTE *If the Cross-Reference Table Status is not current, the title line gives the current status of the cross-reference, click the Build Table button to update the table.*

---

4. Close the Cross Reference window by double clicking on the Control-menu box in the upper right corner of the Cross Reference window.

NOTE *If you have Update Cross Ref Table checked in the Program Setup, all changes made to segments that are validated and entered are automatically updated in the Cross Reference window.*

Ladder Network Cross Reference

In both online and offline programming, you can track addresses in a program by viewing the Cross Reference table.

1. Position the cursor over the instruction address to cross-reference. Select Network Cross Reference from the View menu. The Network Cross Reference window appears (see Figure 6.9). If you have more than one program loaded, the information displayed is for the program in the active window.

NOTE *Only validated and entered logic is considered part of the current program. Thus, logic not entered and validated will not show in the Cross Reference Table. See Validate and Enter Logic in this chapter.*

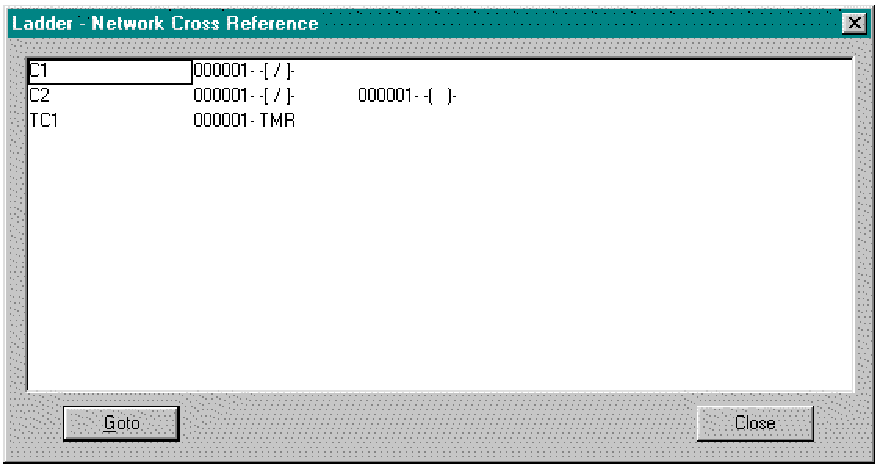


Figure 6.9

2. The complete address information appears in the Ladder -Network Cross Reference Item box.

NOTE *To get a complete Cross-Reference list (including tags and descriptions), use the Cross-Reference feature in Print. Refer to* Print *in Chapter 3 for more information.*

3. Scroll through the list of addresses in the Cross Reference dialog box and pick the instruction address of the logic you want to view more closely.
4. Click the Goto button to jump to the logic of the first occurrence of the address you have highlighted.
5. Close the Cross Reference window by double clicking on the Control-menu box in the upper right corner of the Cross Reference window.

Address Used

To show if specific elements are used in your program, you can build an element usage listing. The following element types can be shown in the Ladder Element Used Table.

X	WX	C	K	TC	SHR	OS
Y	WY	V	STW	DRM	MWT	

Creating an Address Used Listing

To create an element usage listing both online and offline programming, you can track addresses in a program by viewing the Address Used Listing.

1. Select Address Used Listing from the View menu. The Address Used Listing window appears (Figure 6.09b). If you have more than one program loaded, the information displayed is for the program in the active window. The Address Used Listing uses information from the Cross Reference Table. If the Cross Reference Table is not current, a warning will appear to build the table.

NOTE *Only validated and entered logic is considered part of the current program. Thus, logic not entered and validated will not show in the Address Usage Listing. See Validate and Enter Logic in this chapter.*

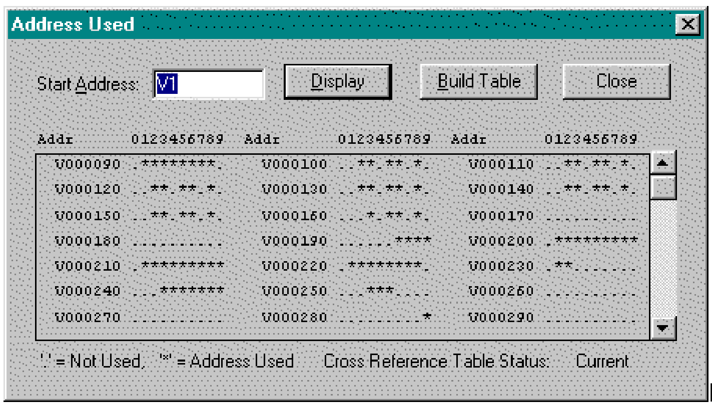


Figure 6.09

2. Type in the address or tag in the corresponding Address box. The starting address will default to C1 or the last address selected in the active program window. Whether Address or Tag is displayed is determined in the Program Setup. The box below the Address/Tag input box contains all addresses used in the active program. Elements are shown on the display in increments of 10 (C0, C10, C20, etc.). Usage of the points between is shown in the adjacent column from 0 through 9 (C1 under 1, C2 under 2, etc.). You receive a display showing use of the selected elements according to the following table:

Blank	Point does not exist.
Dot (.)	Point is not used in the program.
Asterisk (*)	Point is used in the program

3. Scroll through the list of addresses on the right side of the Address Used Listing dialog box to select the next grouping of addresses.

4. Click on the Address input box to start usage list from another address or tag. Repeat Steps 2 through 4 for each item you want to find.

5. Close the Address Used Listing window by double clicking on the Control-menu box in the upper right corner of the Address Used Listing.

NOTE *If you have Table Update checked in the Program Setup, all changes made to logic that are validated and entered are automatically updated in the Address Used Listing.*

Address Trace

The Address Trace is an online or off-line ladder scan for a specific address. The search address is based on the cursor position.

The Trace scans for address instances based on the type of instruction the address is found. If the cursor is located on an input instruction, then the corresponding address for the output instruction is searched for. If the cursor is located on an output instruction then the corresponding address for the input instruction is searched for. Each time the Trace function is selected a new addresses search is invoked.

The address supported by trace are: X, Y, C, WX, WY, K, V, TCP, TCC, STW, DSP, DCP, DSC, DCC, all Gs, VMM, VMS, DRUM, MOVE, OS, TIMER and COUNTER. Trace does not work however, on expression parameters in SFSUB.

Invoking Address Trace

To create an Address Trace both online and offline:

1. Select Address Trace from the View menu. The Address Trace window appears (Figure 6.09c). If you have more than one program loaded, the information displayed is for the program in the active window.

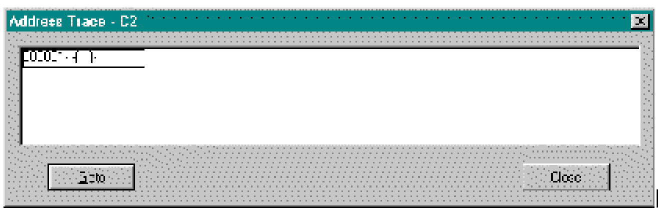
   

   Figure 6.09c

   The list box has five columns each one holds a location. The location consists of rung number and item found. The list box displays the locations in order of rung number.

   The GOTO button closes the Address Trace window and places the cursor in ladder logic window at the network number selected in the Address Trace list box.

2. Close or ESC or clicking on the X in the upper right corner of the dialog closes the dialog.

Unique Address

Unique determines if an element already exists in your ladder program and, if so, gives you the location. For example, if a TIMER is needed for a new feature, you can use the Unique function to see if TMR*nnnn* is already used. Since a TMR is a global memory box, 505 WorkShop also checks to see if any TMR, TMRF, CTR, UDC, MCAT, or DCAT has the same nnnn designator.

NOTE: *Unique checks only for occurrences of the designated element in L-memory. It does not search for an element in Loops, Analog Alarms, SFPGMs, SFSUBs, Intelligent I/O, or Operator Interface devices.*

To access the Unique function: Select an item in the logic window to search on. Then select Unique Address from the View menu. The Unique Address window appears.

The two results of the search are:

- Address exists only at current network.
- Address exist at network number XXXX.

If you have more than one program loaded, the information displayed is for the program in the active window only.

Compare

File Program Compare compares:

- Ladder
- Forced Word I/O
- Forced Discrete I/O
- Forced Control Relays
- Sequencer Scan Time
- Loops
- Alarms
- Special Function Programs
- Special Function Subroutines
- U-Memory
- V and K Memory

data of the selected program on disk to the data in the controller (online) or offline.

To perform a File Program Compare you must first have an opened online or offline program see Chapter 3 Open an Existing Logic Program Offline or Online.

Select File Program Compare from the File menu.

Result: The File Program Compare window appears (Figure 6.09d).

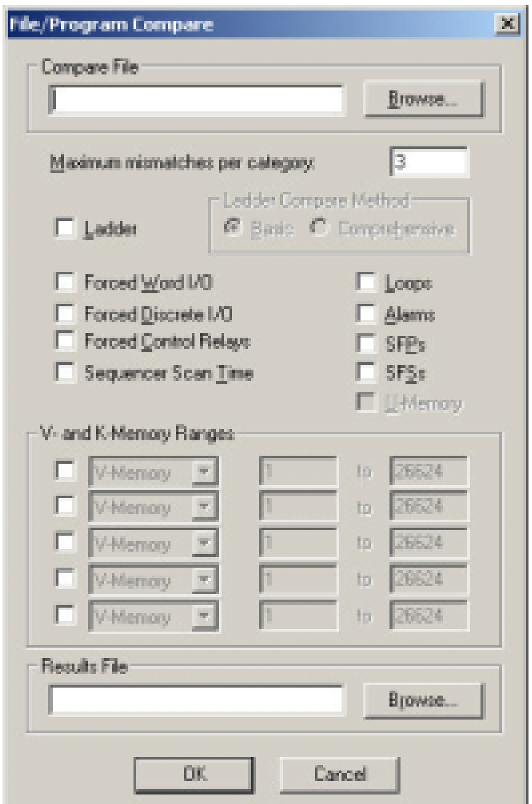


Figure 6.09d

1. Click the Browse button to locate the disk file to compare or type in the file extension and file name in the Compare File box.
2. Next, enter the Maximum mismatches per category. Maximum mismatches per category allows you to abort the compare process if the number of mismatches exceeds the number entered in the Maximum mismatches per category selection box.
3. Now click on the check boxes that correspond to the items you want to compare. For each of the V or K memory address, you can also choose the range for the comparison.

4. If you are comparing ladder logic, select the method. The **Comprehensive** compare provides a better, more detailed comparison. The **Basic** compare is much faster.
5. Then enter a result file path and name in the Result File box if the results of the compare are to be saved to a file.
6. Click OK to start the compare process.

NOTE: *If there are more than one program opened, the File Program Compare compares the program in the active window.*

When the compare finishes, you receive a display such as that shown in (Figure 6.09E). The display highlights any mismatches between the File and controller or offline program.

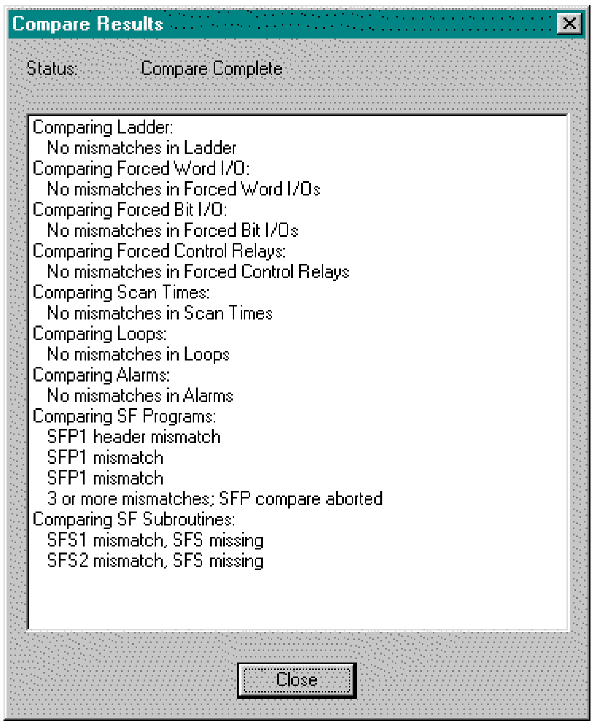


Figure 6.09e

Displaying Processor Faults

You can display the Faults of your processor while online.

To display the processor faults:

1. Click Faults from the Diagnostics menu (Alt+D, A).
2. The online Faults are displayed (see Figures 6.10f).

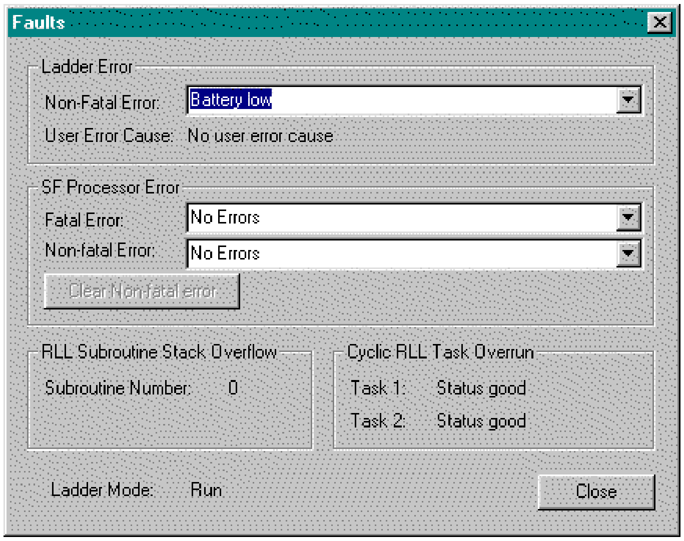


Figure 6.09f

Auxiliary PLC Functions and Displays

Displaying Processor Status

You can display the status of your processor while online or offline.

To display the processor status:

1. Click PLC Status from the PLC Utilities menu (Alt+U, P).
2. Either the PLC Status for online or offline is displayed (see Figures 6.10 and 6.11).
3. Click Close to close the PLC Status box, and return to the active logic program.

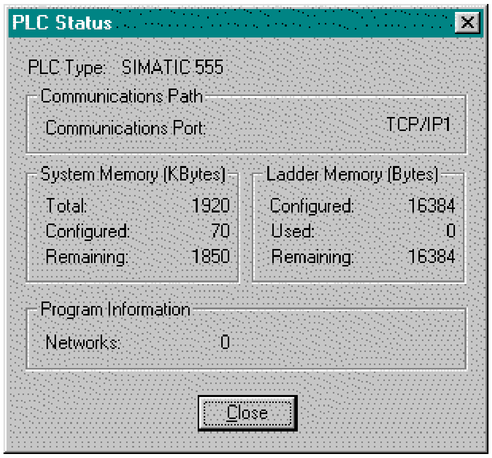


Figure 6.10 Offline PLC Status

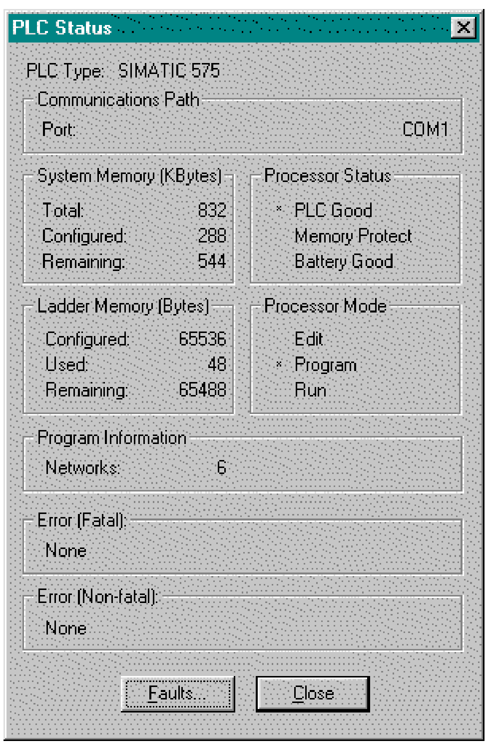


Figure 6.11 Online PLC Status

PLC Operations

This menu item allows you to view and modify PLC operations while online.

To access PLC Operations:

1. Click PLC Operations from the PLC Utilities menu (Alt+U, O). The PLC Operations dialog box appears (Figure 6.12).
2. When you have S-memory configured, you receive a display with three fields, as shown in Figure 6.12, to change operating mode. Use the mouse or arrow keys to select the desired field.

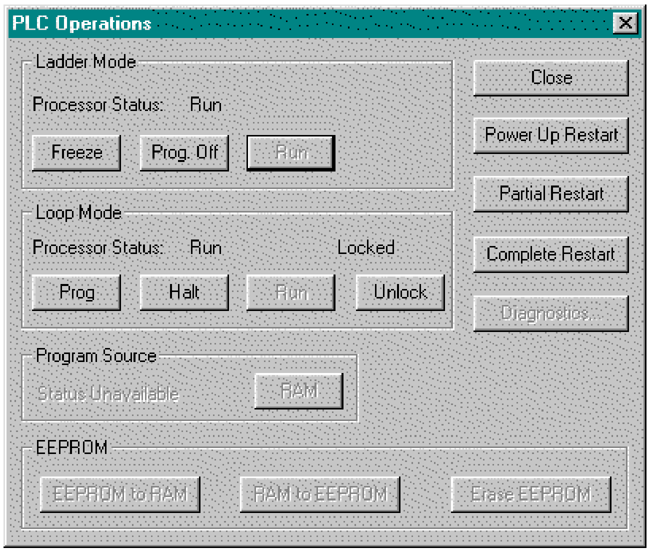


Figure 6.12

Changing Ladder Mode

Without S-memory configured, the LADDERS field is the only option for changing program mode.

Freeze places the controller in PROGRAM FREEZE mode. All outputs are frozen in their current states. However, intelligent I/O modules still can update outputs.

1. To select PROGRAM FREEZE mode, Click on the Freeze button on the PLC Operations dialog box. The following message appears, Stop the PLC with outputs frozen? Select either YES or NO.

   Prog_Off places the controller in PROGRAM OFF mode. Discrete outputs are set to zero, and all word outputs are frozen. However, intelligent I/O modules still can update outputs.

2. To select PROGRAM OFF mode, Click on the Prog_Off button on the PLC Operations dialog box. The following message appears, Stop the PLC with outputs cleared? Select either YES or NO.

Warning	**When you elect to go to PROGRAM OFF mode, you receive the message, Stop the PLC with outputs cleared? Select either YES or NO. Intelligent I/O modules (e.g., the 386/ATM module, the Programmable BASIC module, the Servo Axis module, or the High Speed Pulse Input module) can update outputs even when the controller is in PROGRAM mode, if your code permits. If an intelligent I/O module writes directly to an image register point, its write takes precedence even when PGMFRZ/PGMOFF is in effect. This could cause unexpected control action, resulting in death or serious injury to personnel, and/or damage to equipment. Do not write directly to output image register points from an intelligent I/O module. Instead, write to a control relay or V-memory location and have your RLL program copy this location to the output point.**

Run places the controller in RUN mode, beginning execution of the ladder program.

3. To select PROGRAM RUN mode, Click on the RUN button on the PLC Operations dialog box. The following message appears: Run PLC? Select either YES or NO.

Changing Loop Mode

When you select Loop Mode, you have the following options available.

NOTE *Your loop card cannot be in PROGRAM mode while the discrete CPU is in RUN mode.*

Prog places the loop card in PROGRAM mode, and control blocks are neither running nor being queued for running.

1. To select LOOP PROGRAM mode, Click on the Prog button on the PLC Operations dialog box. The following message appears: Set the loop processor to program mode? Select either YES or NO.

   Halt places the loop card in HOLD mode; enabled control blocks are being queued to run, but are not running. Upon returning to RUN mode, execution resumes where it was halted.

2. To select LOOP HALT mode, Click on the Halt button on the PLC Operations dialog box. The following message appears: Set the loop processor to halt mode? Select either YES or NO.

   Run places the loop card in the RUN mode; enabled control blocks are being queued and are running.

3. To select LOOP RUN mode, Click on the Run button on the PLC Operations dialog box. The following message appears: Set the loop processor to run mode? Select either YES or NO.

   The loop mode can either follow that of the discrete CPU or be in RUN mode independent of the ladder program. To invoke this option, select the LOCKED or UNLOCKED field.

   - Unlock frees the loop card for selection of RUN operating mode independent of the ladder CPU.
   - Lock locks the loop card to follow the operating mode selected for ladders.

To select LOOP LOCK/UNLOCK mode:

1. Click on the Lock/Unlock button on the PLC Operations dialog box. The following message appears, Lock or Unlock loop processor from ladder operation mode?
2. Select either YES or NO.

Run Controller Diagnostics

Use Diagnostics to run diagnostics on your controller. To execute Diagnostics:

1. Click **Diagnostics** from the PLC Operations dialog box.
2. When you receive the message, **Execute Diagnostics?** Select either YES or NO.

Programming EEPROMS

To perform the following **EEPROM** tasks, your controller must be in PROGRAM mode.

1. Copy the contents of RAM to EEPROM.
2. Copy the contents of EEPROM to RAM.
3. Erase the contents of EEPROM.

To execute the above EEPROMS tasks:

1. Click on one of the above EEPROM tasks from the PLC Operations dialog box.
2. When you receive the message, **Do you want to copy EEPROM to RAM?** Select either YES or NO.

Programming Port Lockout

Use Port Lockout to prevent program changes from being made at different ports simultaneously.

If the controller is unlocked, the Port Lockout button will display Locked. If the controller is locked, the Port Lockout button will display Unlock.

To execute Port Lockout:

1. Click Port Lockout from the PLC Utilities menu (Alt+U, O). The Port Lockout dialog box appears (Figure 6.12a). Click **Lock or Unlock** from the Port Lockout on the Port Lockout dialog box.

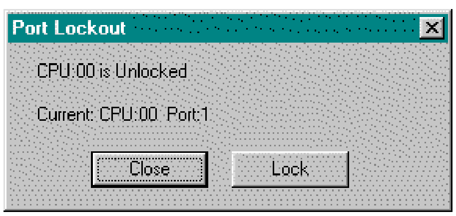


Figure 6.12a

NOTE Programming devices connected through the same Dual Communication Port (DCP) cannot lock each other out.

---

2. When you receive the message**, Lock or Unlock communication ports?** Select either YES or NO.

System Part Number

Use System Part Number to read the software part number and the release number of the cards installed in your controller.

To execute System Part Number:

1. Click System Part Number from the PLC Utilities menu (Alt+U, N). The System Part Number dialog box appears (Figure 6.12b). The part number and release for each card, along with the slot number and name of the card, displays as shown.

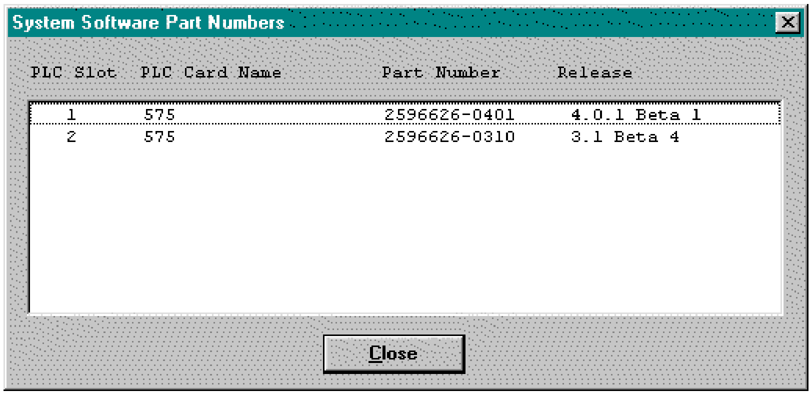


Figure 6.12b

Performing Syntax Check

Use the Syntax Check to check your RLL program for errors that will prevent the controller from entering RUN mode (such as **UNKNOWN INSTRUCTION**, **LADDER ELEMENT OUT OF RANGE**, **NO CORRESPONDING PAIR**, or **OUT OF MEMORY)**. Syntax Check is currently available only on 545, 555, and 575 controllers.

To execute the Syntax Checker:

1. Click Syntax Checker on the Diagnostics menu (Alt+D, Y). The Syntax Checker menu dialog box is displayed (Figure 6.12a).

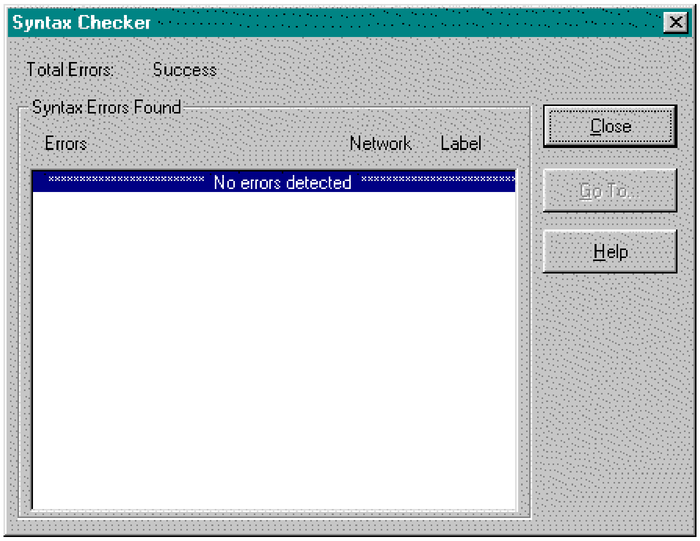


Figure 6.12a

Up to 16 errors can be displayed on the Syntax Check screen; if more than 16 errors are detected, the prompt line announces **ADDITIONAL ERRORS DETECTED**. To display the additional errors, use the scroll bar on the dialog box.

If no errors are detected, Syntax Check reports **"No errors detected"** on the bottom line.

If errors are detected, they will be listed within the dialog box. The logic errors can be viewed by selecting the error and pressing the go to error button on the dialog box.

Ladder Status (Online)

You can display the values of your addressees in the PLC by using Status. Status can be displayed for Ladder networks. Status is an online feature only.

Status will continue to update when you scroll and cursor within the program.

To display Ladder status:

1. Click Status on the Diagnostics menu (Alt+D, S). The Status menu item displays a check if Status has been turned on.
2. Ladder status is indicated by the following features:
   - Contacts and Coils, when on, are displayed in Red.
   - Addresses in box instructions indicate the current value for each address.

   NOTE Status mode is automatically exited when an attempt is made to edit a network. When the network is validated and entered, Status is automatically displayed again.

3. Click Status on the Diagnostics menu to stop displaying status.

Initiating a Single Scan

Using a single scan allows you to view a single execution of your program. You must have the controller in program mode in order to execute a single scan.

To perform a single scan of your program, complete the following steps.

1. Place the controller in program mode.
2. Click Single Scan Setup on the Diagnostics menu (Alt+D, I). The Single Scan Setup menu dialog box is displayed (Figure 6.12dd). If your programmable controller supports the Single Scan pop-up task box for more than one task, you can select which tasks to execute during the single scan.

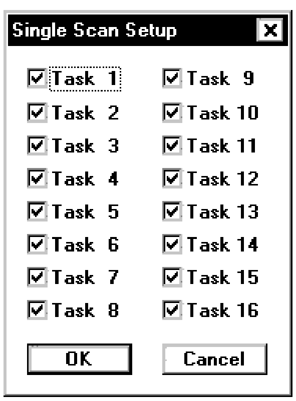


Figure 6.12dd

3. Click in the task number radial box to include a task in the scan., followed by clicking **OK** to saves the task box selection.
4. Click Single Scan on the Diagnostics menu (Alt+Q) to execute a single scan according to the single scan task configuration.

RBC Part Number

You can display the RBC Software part number and release number while online by:

1. Click PLC RBC Part Number from the PLC Utilities menu (Alt+U, R).
2. The RBC software number(s) and release numbers are displayed in the format shown in Figure 6.12b.
3. Click Close to exit the RBC Part Number box and return to the active logic program.

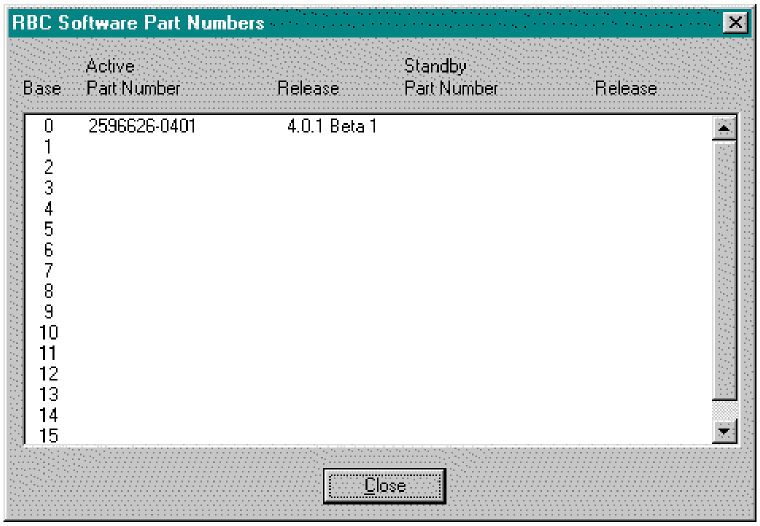


Figure 6.12b

Task Codes Per Scan

Task Codes Per Scan is used to set the number of task codes processed per scan for SF modules on each channel. To invoke the function:

1. Click Task Codes Per Scan from the PLC Utilities menu (Alt+U, A).
2. The Task Codes Per Scan dialog box is displayed in the format shown in Figure 6.12c. Only numbers for the channels you have configured are displayed; e.g., if you have four channels, the number goes to 4.
3. Type in the data field the number of task codes for SF modules per scan.
4. Click on the Write button to enter the numbers.
5. Click Close to exit the Task Codes Per Scan and return to the active logic program.

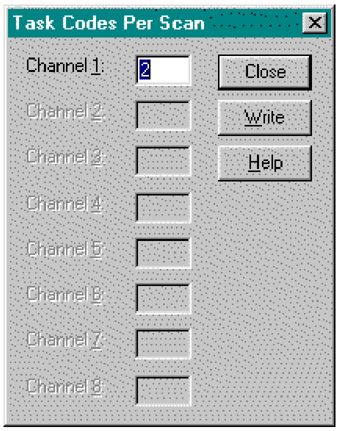


Figure 6.12c

Diagnostics on Base

Diagnostics on Base is used to run diagnostics on one or all the I/O bases. To invoke the operation:

1. Click Diagnostics on Base from the PLC Utilities menu (Alt+U, D).
2. The Diagnostics on Base dialog box is displayed in the format shown in Figure 6.12d.
3. To run base diagnostics on one base:
   - ❑ Type in the channel and base number in their associated data field.
   - ❑ Click on the Run Current button, and the data will be displayed in the dialog box.
4. To locate the Dual Media base:
   - ❑ Type in the channel and base number in their associated data field.
   - ❑ Click on the Next DM button, and the data will be displayed in the dialog box.
5. To change the active and standby roles of RBCs on a particular base:
   - ❑ Type in the channel and base number in their associated data field.
   - ❑ Click on the Swap RBCs.
6. To run base diagnostics on all bases:
   - ❑ Click on the Run All button and the data will be displayed in the dialog box.
7. Click Close to exit the Task Codes Per Scan and return to the active logic program.

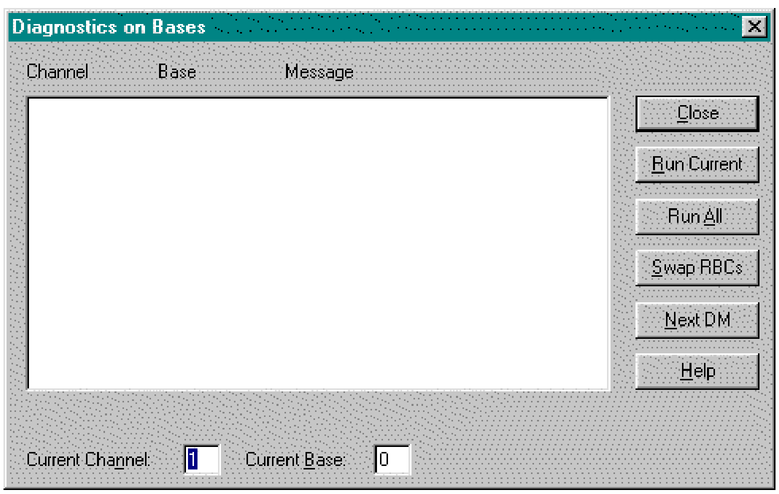


Figure 6.12d

Hot Backup

Hot Backup is used to select the status of a hot backup unit. Hot Backup is only valid for the 565 controller. To execute the function:

1. Click Hot Backup from the PLC Utilities menu (Alt+U, H).
2. The Hot Backup dialog box is displayed, with the status of the Hot Backup.
3. There are three status modes to select from while in the Hot Backup.
   - ❑ Standby to Offline: Puts the standby unit in PROGRAM mode.
   - ❑ Standby to Online: Puts the standby unit in RUN mode.
   - ❑ Switch: Switches the roles of the active unit and standby unit.
4. Click Close to exit the Hot Backup and return to the active logic program.

Password

The password feature provides protection for areas of memory that are part of the program. There are two passwords, one for the selected program on disk and one for the controller. They do not have to be the same.

NOTE *You do not need to enter a password to go online. However, if the selected program on disk is protected, you must enter a password to go offline.*

A password value consists of 1 to 8 alphanumeric digits (for example, 0-9 or uppercase A-Z). Online, you can be prompted for the password for the controller program and/or for the selected program on disk. Offline, you are prompted for a password only for the selected program on disk.

The selected program on disk may be in one of two states of password protection:

- No Password: The selected program on disk is not protected. Any authorized user may enter an initial password.
- Disabled Password: The selected program on disk is not protected. Any authorized user may change or delete the password.

The programmable controller may be in one of three states of password protection:

- No Password: The controller program is not protected. Any authorized user may enter an initial password.
- Disabled Password: The controller program is not protected. The user may change or delete the password. Any authorized user may enable the password.
- Enabled Password: The controller program is protected according to the protection level assigned to the password (see below). If a protected operation is attempted from any communications port, the operation is denied and an error response is given. Only an authorized user may change, delete, or disable the password.

Three levels of access are available when a password has been entered and enabled in the controller:

- No Access: The controller program cannot be read or modified.
- Read-only Access: The controller program can be read but it cannot be modified.
- Full Access: The controller program is not protected.

Password Operational Modes

There are nine online operational modes for password. The following paragraphs describe online password operational modes and online disk password operations.

The following numbered paragraphs detail the corresponding mode-number information found in Online Password Operations Table.

NOTE *If converting files from TISOFT to 505 WorkShop, check to make sure that the selected program on disk has TISOFT 5.0 or greater format.*

Mode 1: If the controller and the selected program on disk do not have a password, you will not be prompted for a password. (Save and Open functions will be allowed.)

No password will be written to the selected program on disk or to the controller.

Mode 2: If the selected program on disk has a password and you enter a password for the selected program on disk at the online prompt, you will be able to use (Save and Open) functions.

Mode 3: If the selected program on disk has a password and you do not enter a password for the selected program on disk at the online prompt, you will not be allowed to use the (Save) function. You will, however, be able to use the (Open) function, but only if the controller supports password. You will also have full access to the controller and the selected program on disk.

Mode 4: If the controller has a password and you enter a password for the controller at the online prompt, you will be able to use the (Save) and the (Load) functions. Save writes the controller password to the selected program on disk. Open retains the controller password. You will also have full access to the controller and the selected program on disk.

Mode 5: If the controller has a password and you do not enter a password for the controller at the online prompt, you will not be able to use the (Save) and the (Open) functions. Depending on the controller password access level, you will be allowed partial to full access to the controller. You will also have full access to the selected program on disk.

Mode 6: If both the controller and the selected program on disk have passwords and you enter a password for both at the online prompt, you will be able to use the (Save) and the (Open) functions. The passwords for both the selected program on disk and for the controller are retained. You will also have full access to the selected program on disk and to the controller.

Mode 7: If both the controller and the selected program on disk have passwords and you enter a password only for the selected program on disk at the online prompt, you will not be able to use the (Save) and (Open) functions. You will have partial to full access to the controller depending on the controller access level. You will also have full access to the selected program on disk.

Mode 8: If both the controller and the selected program on disk have passwords and you enter a password for the controller at the online prompt, you will not be allowed to use the (Save) function. You will, however, be able to use the (Open) function. Open will write the selected program on disk password to the controller and enable the password in the controller. You will also have full access to the controller. Some reads and no writes will be allowed to the selected program on disk.

Mode 9: If both the controller and the selected program on disk have passwords, and you do not enter a password for either the controller or the selected program on disk at the online prompt, you will not be able to use either the (Save) or the (Open) functions. You will, however, have partial to full access to the controller depending on the controller access level. Some reads and no writes will be allowed to the selected program on disk.

All online password operational modes are summarized in the following table.

Online Password Operations Table

Mode No.	Controller Password	Offline Password	User Password	Save Allowed to Disk?	Open Allowed to PLC?	Comments
1	No	No	Not asked for.	Yes. No password.	Yes. No password.	Selected program on disk must have 5.0 format or greater. There is no password for selected program on disk or in the controller.
2	No	Yes	User gives password for selected program on disk.	Yes. Selected program on disk password is retained.	Yes. Disk password is written to controller, if controller supports password.	Full access to controller and selected program on disk is allowed.
3	No	Yes	User does not give password for program on disk.	No. Selected program on disk is password protected.	Only if controller supports password.	Full access to controller is allowed. Some reads and no writes are allowed to selected program on disk.
4	Yes	No	User gives controller password.	Yes. Controller password is written to disk.	Yes. Controller password is retained.	Full access to controller and to selected program on disk is allowed.
5	Yes	No	User does not give a password.	No. Controller is password protected.	No. Controller is password protected.	Partial to full access to controller is allowed depending on access level. Full access is allowed to selected program on disk.
6	Yes	Yes	User gives disk and controller password.	Yes. Selected program on disk password is retained.	Yes. Controller password is retained.	Full access to selected program on disk and to controller is allowed.
7	Yes	Yes	User gives disk password.	No. Controller is password protected.	No. Controller is password protected.	Partial to full access to controller is allowed depending on access level. Full access is allowed to selected program on disk.
8	Yes	Yes	User gives controller password.	No. Selected program on disk is password protected.	Yes. Selected program on disk password is written to controller and enabled.	Full access to controller is allowed. Some reads and no writes are allowed to selected program on disk.
9	Yes	Yes	User gives no password.	No.	No.	Partial to full access to controller is allowed depending on access level. Some reads and no writes are allowed to selected program on disk.

To invoke Password from Online or Offline:

1. Click on Password from the PLC Utilities menu (Alt+U, W).
2. The Password dialog box is displayed in the format shown in Figure 6.12e.

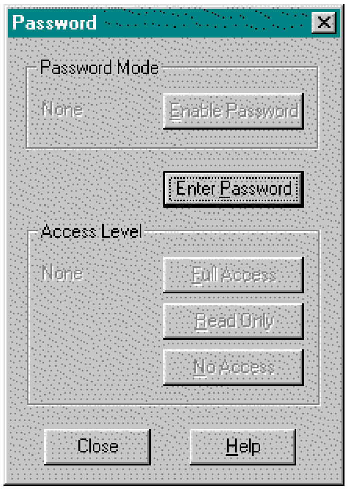


Figure 6.12e

Online Password Selection and Access Level

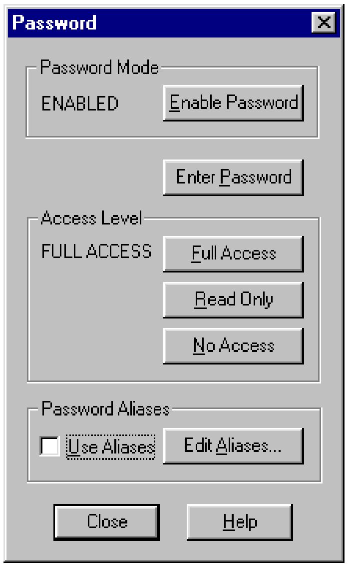


Figure 6.13

The buttons in Figure 6.13 are defined below:

- **Close-**Allows you to return to the ladder display screen.
- **Enter Password**-Allows you to enter or change the password for the selected program in the controller. WorkShop prompts you for the new password twice. If the new passwords are not the same, the password is not changed. To delete a password (or to make the selected program on disk unprotected), enter a null password or press clear and enter a new password.
- **No Access-**Allows you to set the protection level of the controller for no access. This option is written to the controller. No Access does not work without a password.
- **Read Only-**Allows you to set the protection level of the controller for read-only access. This option is written to the controller. Read Only does not work without a password.
- **Full Access-**Allows you to set the protection level of the controller for full read/write access. This option is written to the controller. Full Access does not work without a password.
- **Enable Password-**Allows you to enable password protection for the controller.
- **Disable Password-**Allows you to disable password protection for the controller. If you have not already entered a password for the controller, 505 WorkShop prompts you for the current password. If the password you provide is not correct, 505 WorkShop does not disable the password protection. This function does not work unless you provide a password.

Offline Password Selection and Access Level

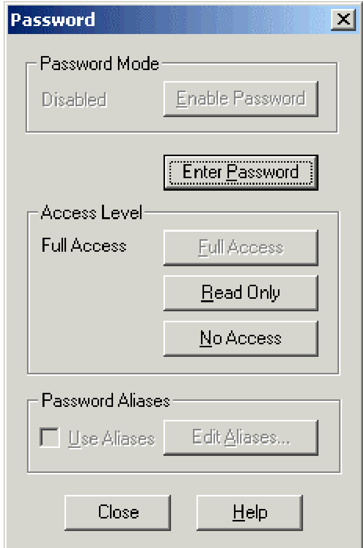


Figure 6.12e

The buttons in Figure 6.12e are defined below:

- **Close-**Allows you to return to the ladder display screen.
- **Enter Password**-Allows you to enter or change the password for the selected program on disk. WorkShop prompts you for the new password twice. If the new passwords are not the same, the password is not changed. To delete a password (or to make the selected program on disk unprotected), enter a null password or press clear and enter a new password.
- **No Access**-Allows you to set the protection level of the controller for no access. This option is written to the controller when you perform an (Open) function. No Access does not work without a password.
- **Read Only-**Allows you to set the protection level of the controller for read-only access. This option is written to the controller when you perform an (Open) function. Read Only does not work without a password.
- **Full Access-**Allows you to set the protection level of the controller for full read/write access. This option is written to the controller when you perform an (Open) function. Full Access does not work without a password.

PLC Password Alias

Online access to many PLC’s can be restricted with its onboard password feature. This PLC password is a case insensitive alphanumeric string of up to 8 characters. When attaching online to a PLC containing a password, the "Enter Password" dialog below requests the correct PLC password before allowing access to the PLC.

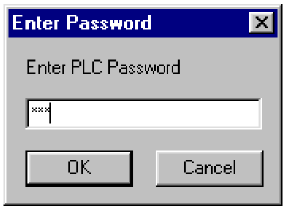


Figure 6.14a

his PLC password is actually stored in the PLC. Therefore programming packages other than 505 WorkShop also request the same PLC password when attaching to the PLC.

Some administrators prefer using 505 WorkShop but are aware their PLCs can be accessed through other programming packages when users know the actual passwords stored in the PLCs.

These administrators may place an additional layer of security over the onboard PLC password feature by using *PLC Password Aliases*. Administrators maintain a list of PLC password aliases in 505 WorkShop. These aliases are case insensitive alphanumeric strings up to 8 characters long and are stored <u>on the PC on which 505 WorkShop runs</u>. Each alias is associated with a PLC password stored in a PLC.

When PLC password aliases are used, a PLC password is requested as seen in the dialog above. But instead of entering the password actually stored in the PLC, the password alias is entered.

505 WorkShop looks up this alias in its list of password aliases to find the associated PLC password. This associated PLC password is then used to grant online access to the PLC.

Note: *PLC password aliases can be used only when either FasTrak Security or NT Authentication Security are in use.*

Accessing the PLC Password Alias Feature

PLC password aliases are maintained through the PLC "Password" dialog. Select "Password…" from the "PLC Utilities" menu below to display this dialog.

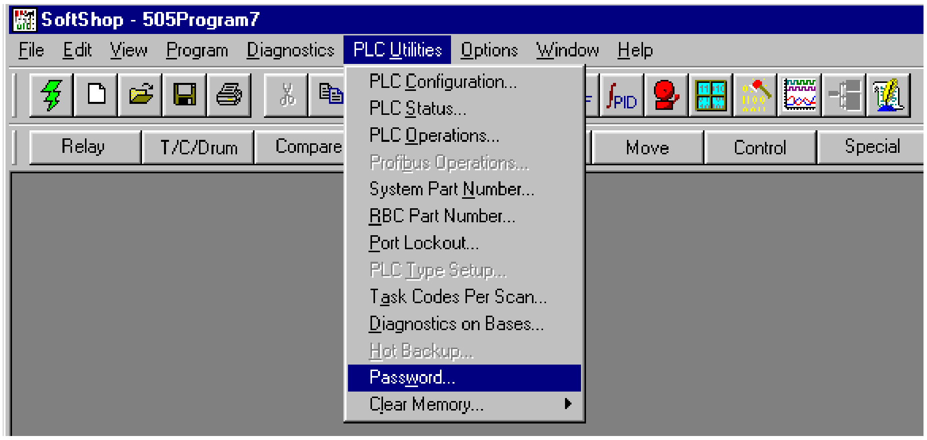


Figure 6.14b

The PLC "Password" Dialog

The PLC "Password" dialog below retains its original functionality which allows entry and modification of the actual PLC password, setting the access level (full, read only or no access) within the PLC, etc.

However, the addition of the controls within the "Password Aliases" group box offers access the PLC password alias feature.

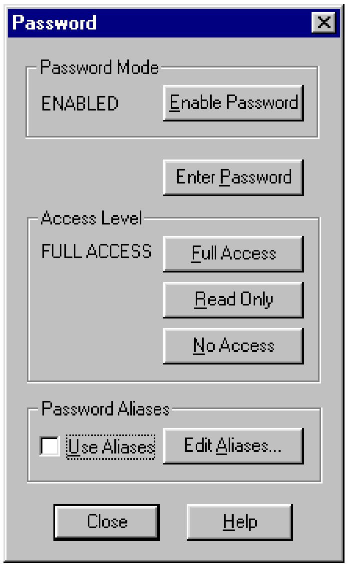


Figure 6.14c

The controls within the "Password Aliases" group box are enabled only when either FasTrak Security or NT Authentication Security are in use. When these controls are clicked, a valid security administrator password must be entered.

- **Use Aliases -** When attaching to a PLC which contains a password, the "Enter Password" dialog requires the correct PLC password before granting access to the PLC.

  If the "Use Aliases" check box is unchecked the actual password stored in the PLC must be entered. If his check box is checked, the PLC password alias associated with the PLC's actual password must be entered instead.
- **Edit Aliases -** The list of PLC passwords and the aliases with which they are associated can be access by clicking the "Edit Aliases" button. Clicking this button displays the "PLC Password Aliases" dialog below.
- **PLC Password Aliases -** Click the "Edit Aliases…" button above to display the "PLC Password Aliases" dialog below. The list box contains the PLC password and PLC password alias pairs already entered.

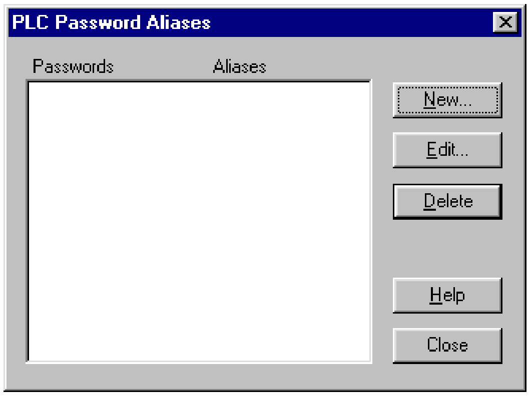


Figure 6.14d

- **New… -** Click the "New…" button to enter a new PLC Password/PLC Password Alias pair. The "Enter New PLC Password Alias" dialog below appears.

Figure 6.14e

In the "PLC Password" group box, enter the actual PLC password in the "Password" edit box. Verify the password by entering it again in the "Verify password" edit box.

In the "PLC Password Alias" group box, enter the password alias in the "Alias" edit box. Verify the alias by entering it again in the "Verify alias" edit box.

When all four edit boxes have been entered, click "OK". The "Password" and "Verify password" entries are compared to ensure they match. The "Alias" and "Verify alias" entries are also compared to ensure they match.

Finally, the "Alias" is compared to the password aliases already on file. Since the each password alias may be associated with only one PLC password, aliases cannot be duplicated. If the entered alias is found in the list, an error message appears and the entered PLC Password/PLC Password Alias pair is not saved.

But when the entered password and alias are approved, this new PLC Password/PLC Password Alias pair is added to the list and return to the "PLC Password Alias" dialog.

❑ **Edit…-** To edit a PLC Password/PLC Password Alias pair, highlight the desired pair in the list box and click the "Edit…" button. Alternately, double click a PLC Password/PLC Password Alias pair in the list box.

The PLC password and PLC password alias selected from the list box appear in the "Edit PLC Password Alias" dialog below.

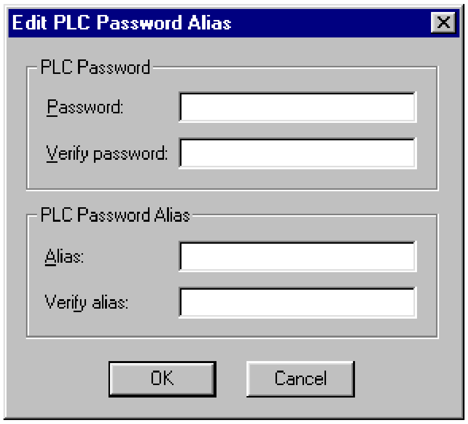


Figure 6.14f

This dialog operates much like the "Enter New PLC Password Alias" dialog. The "Password" and "Verify password" entries must match just as the "Alias" and "Verify alias" entries must match.

Note: *If the PLC password remains the same but the alias changes, the original alias is replaced by the new alias. The PLC password is now associated with a new alias.*

However, if the alias remains unchanged and the password changes, the alias becomes associated with the new PLC password and is no longer associated with the original PLC password.

❑ **Delete -** Any or all PLC Password/PLC Password Alias pairs may be permanently removed from the list stored on the PC.

Highlight one or more items (single click, control-click, shift-click, etc.) from the list box in the "PLC Password Aliases" dialog and click the "Delete" button. A confirmation message states the selected item(s) will be permanently deleted. When this message is acknowledged, the item(s) are permanently removed from the list.

Powering Up/Restart the Controller

Power Up Restart

Use **Power Up Restart** to clear all unforced X, Y and non-retentive C elements on power up or restart of the controller. Retentive control relays are not cleared. The WX and WY elements are not affected.

Caution	**If you execute Power Up Restart with the controller battery switch set to Off, all programs residing in the controller will be cleared. Be certain to check battery switch position before using Power Up Restart.**

To execute **Power Up Restart:**

1. Click **Power Up Restart** from the PLC Operations dialog box.
2. When you receive the message, **Execute a PLC power up restart?** Select either YES or NO.
3. For 575 controllers, you are also prompted with **Coordinate Reset With Other Applications:** and **Coordinate Reset With Entire System:** For each of these prompts, press NO or YES as required for your process.

Partial Restart

Use a Partial Restart to clear all discrete elements except retentive C and forced elements. The word elements and presets are not reset.

To execute a **Partial Power Up Restart:**

1. Click **Partial Restart** from the PLC Operations dialog box.
2. When you receive the message, **Execute a Partial restart?** Select either YES or NO.
3. For 575 controllers, you are also prompted with **Coordinate Reset With Other Applications:** and **Coordinate Reset With Entire SYSTEM:** For each of these prompts, press NO or YES as required for your process.

Complete Restart

Use Complete Restart to clear all discrete elements and word elements, including retentive C elements. Complete Restart also clears controller fatal errors. Forced discrete elements and forced word elements are not reset.

To execute Complete Restart:

1. Click Complete Restart from the PLC Operations dialog box.
2. When you receive the message, Execute a Complete restart? Select either YES or NO.

For 575 controllers, you are also prompted with Coordinate Reset With Other Applications: and Coordinate Reset With Entire System: For each of these prompts, press NO or YES as required for your process.

Clearing Memory

This option allows you to clear all or parts of logic memory, data, tags, and documentation in the current program.

You can use clear memory in either online or offline mode. When programming offline, you clear the entire active program or parts. When programming online, you clear the PLC memory. However, you cannot clear memory online while the processor is in Run mode. You must first stop the processor before clearing the memory online.

The following is a list of parts of memory that can be cleared:

- Ladder
- V Data
- K Data
- Word I/O Data
- TCC/TCP Data
- DSP/DSC Data
- PID Loops
- Analog Alarms
- Special Functions Programs
- Special Functions Subroutines
- U-Memory

To access the Clear Memory option:

1. If you want to save the changes you've made to your logic and documentation, save them before going to Step 2. Use the Save Program or Save Program As options in the File menu.
2. Click Clear Memory from the PLC Utilities menu (Alt+U, L, A). A warning message appears stating all program logic, data values, tags, and documentation will be deleted.
3. Click Yes to clear all memory. If changes to the program were not saved, another warning message appears stating that changes to your program were not saved; do you wish to continue with the clear memory procedure. Click Yes to clear the memory.

Selecting (575) Application

When you configure a 575 controller offline, you must first select the application:

1. Select PLC Configuration from the PLC Utilities menu. The PLC Configuration dialog box appears (Figure 5.2).
2. Select the **ReqApp** button under the Processor Information window on the PLC Configuration dialog box. The Required Application dialog box appears (Figure 6.15).

You can change the Application ID only during offline configuration. (Configuring the controller online displays the current application.)

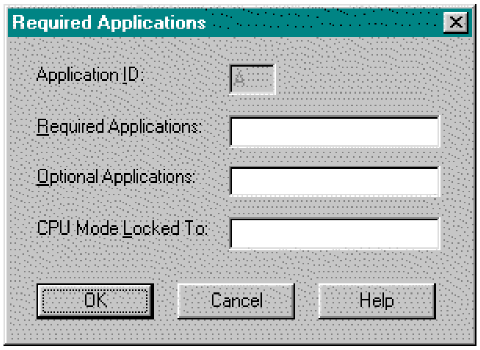


Figure 6.15

The following fields display information about applications. Only the IDs listed in the REQUIRED or OPTIONAL fields are valid for use as G-memory parameters in your RLL program.

- APPLICATION ID displays the ID of the current application.
- REQUIRED APPLICATIONS displays a listing of the application IDs needed for the current application to complete a process.
- OPTIONAL APPLICATIONS displays a listing of the application IDs that are not required, but may be present.
- CPU MODE LOCKED TO displays a listing of the application IDs that must transition to RUN mode at the same time.

Connecting online displays the current application ID configuration.

After modifying the configuration, click OK to enter the new configuration relationships.

7 - Documentation

Ladder Header

505 WorkShop provides a simple way to document ladder while you are creating or editing your program. The maximum number of characters you can enter for each header is 16K.

Double-click the Ladder Header icon in the active logic program window (see Figure 7.1). The Ladder Header dialog box is displayed (see Figure 7.2). Type in your header. Select OK when you're finished.

Figure 7.1

To see ladder headers in the active logic program, select Program Setup from the Options menu. Click Ladder Headers in the Logic Editor box and click OK.

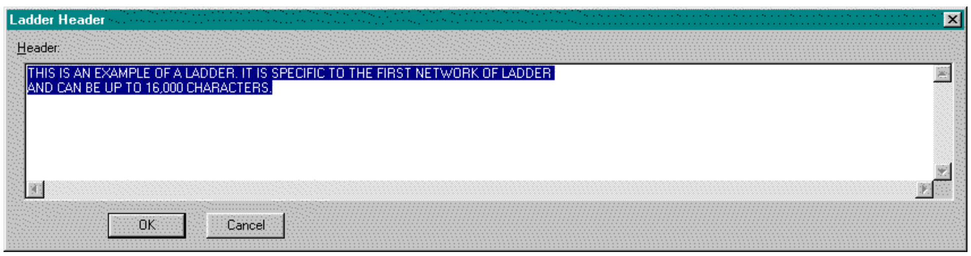


Figure 7.2

Network Header

505 WorkShop provides a simple way to document networks while you are creating or editing your logic program. The maximum number of characters you can enter for each header is 16K. The network header is tied to the corresponding network output. If a XCALL, PGSTS OR PGTSZ are the outputs, the header is tied to the first occurrence of an X, Y or C address.

Double-click the Network Header Input icon in the active logic program window (see Figure 7.1). The Network Header Input dialog box is displayed (see Figure 7.3). The drop down box paging preference in the top left-hand corner of the header input box. Controls the page print out of the particular network header and ladder that is selected. If Page None is selected there will be no page break between the current network and last network printed. Page Odd prints the current selected network on the next odd page. Page Before prints the current selected network on the next blank page. Type in your header. Select OK when you're finished.

To see network headers in the active logic program, select Program Setup from the Options menu. Select the Logic tab. Then General from the drop down box and check Show All Headers.

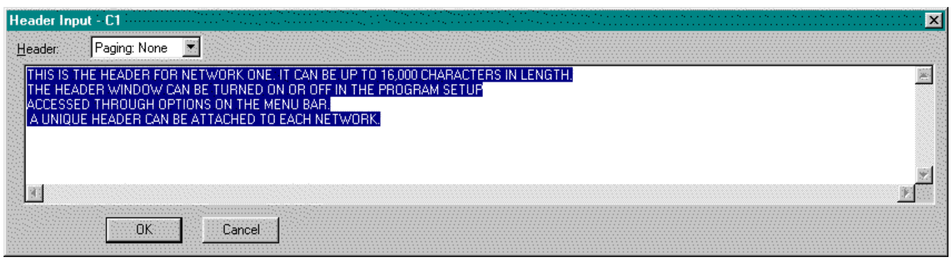


Figure 7.3

SF Header

505 WorkShop provides a simple way to document SF while you are creating or editing your SF program. The maximum number of characters you can enter for each header is 16K.

Double-click the SF Header Input icon in the active logic program window. The Network Header Input dialog box is displayed (see Figure 7.5). Type in your header. Select OK when you're finished.

Figure 7.4

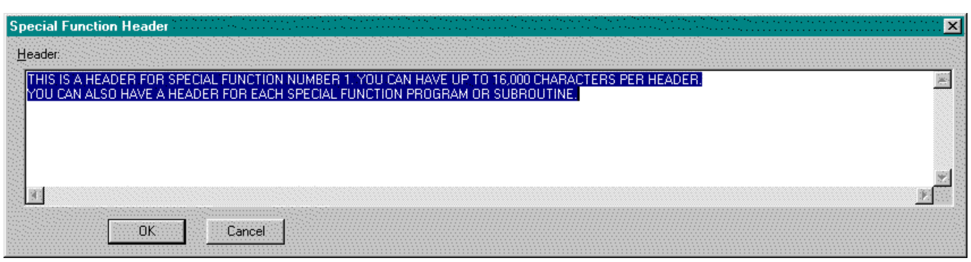


Figure 7.5

To see SF headers in the active logic program, select Program Setup from the Options menu. Click All Headers in the Logic Editor box and click OK.

Using the Documentation Window

In both online and offline programming, you can view and edit tags, descriptions, and comments in your program using the Documentation Window.

The Documentation Window allows you to view, create, edit, and delete tags, descriptions, and comments for the active logic program. The maximum number of characters for each item is defined below:

Descriptions	96
Tags	32
Comments	2048

To open the Documentation Window, click on the toolbar or select Documentation Window from the View menu (Figure 7.6).

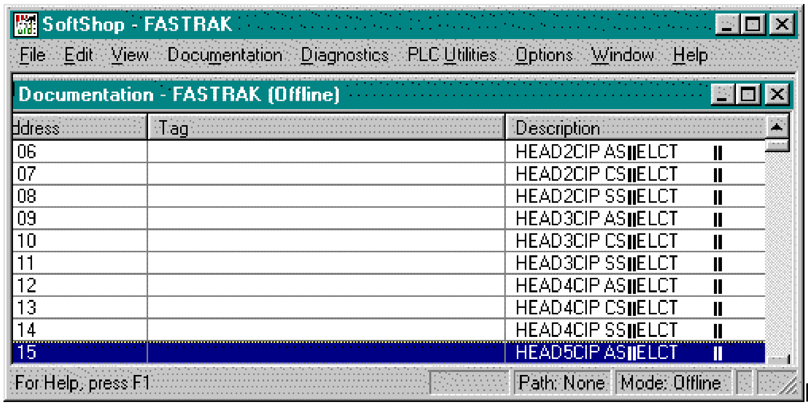


Figure 7.6

The window is displayed with all of the addresses, tags, and descriptions in the current program. The first column indicates the method of sorting.

NOTE *Only one (1) documentation window can be displayed per program.*

The window also can be sized and moved to another location within the viewing area using the standard window features.

Customizing the Display

The Documentation Window can be customized to display Tags, Descriptions, or both Tags and Descriptions. Setting options in the Program Setup can customize these Program Setup. The Program Setup also allows you to select the sorting method: by address, tags, or descriptions.

To change the display, select Options, Program Setup (Alt-O, P). In the Documentation Window section, click Tag to include tags and/or click Descriptions to include address descriptions. You must include either Tags or Descriptions in your window. In the Sort Order group, select Address, Tag, or Description to change the sort method. The sort order also modifies the Documentation Window by displaying the sort order method as the first column.

For example, if the Sort Order is defined as Tag, then the first column in the Documentation Window is the Tag column. The address column is always included in the display. See Program Setup in Chapter 4 for more information.

Editing and Creating New Documentation

You can create a new address tag, description, and comment by using one of the following methods:

1. From the Documentation menu, select New Doc (Alt-D, N).
2. From the Pop-up menu in the Documentation Window, select New Doc.
3. Select an address in the Documentation Window press the Enter key, Ctrl-L or double click on the address.

NOTE *You can also edit documentation in the Data Window by pressing Ctrl L on the description or tag columns.*

The Edit Documentation window is displayed. You can enter a new tag, description, and comment (see Figure 7.7) Select OK when you're finished.

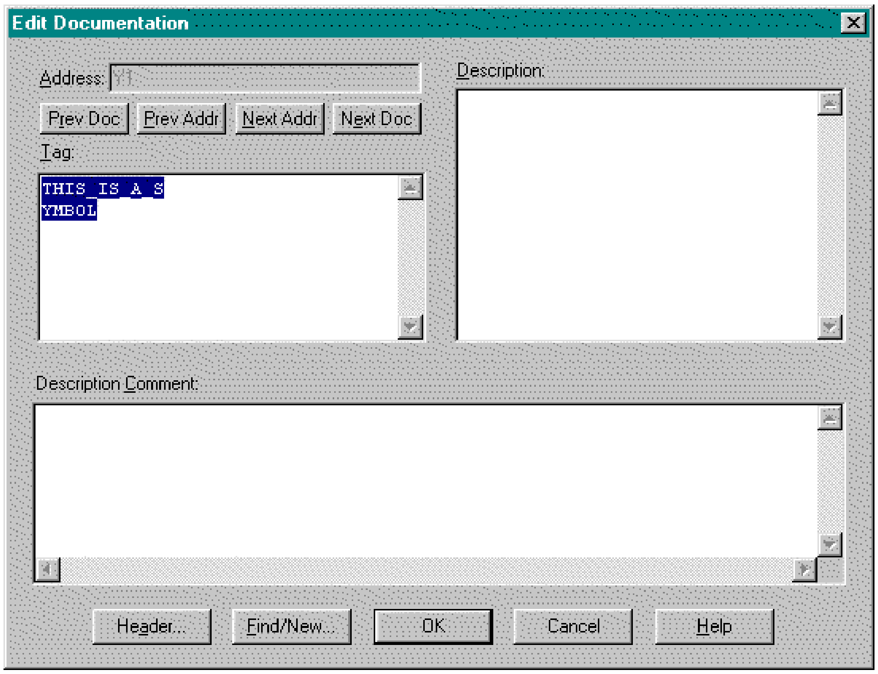


Figure 7.7

The Tag and Description fields are sized according to the Documentation Window Column Width variable in the Program Setup. The font selected in your program setup will also be used for the tag and description fields. This will show the documentation, as it actually will be displayed in your ladder program.

NOTE *If the font and size selected in the program setup is too large to be represented in the window, a standard font will be used. When this situation occurs, the warning message is displayed. The tag and description will NOT be shown in its actual size in this case.*

To edit an existing tag or description, you can use any of the following methods:

1. From the Documentation menu, select Modify Doc (Alt-D, M).
2. From the Pop-up menu in the Documentation Window, select Modify Doc.
3. Press the Enter key, Ctrl-L or double click on the actual address, tag, or description in the Documentation Window (Figure 7.6).

NOTE *If you double-click in the description column to edit existing documentation, the cursor will be placed in the description column in the Edit Documentation window (Figure 7.7). Similarly, if you double-click in the tag column, the cursor will be placed in the tag column in the Edit Documentation Window.*

The Edit Documentation window is displayed. You can change the tag, description, and comment (Figure 7.7). Select OK when you're finished.

Cut, Copy, Paste and Deleting Documentation

Probably the most frequently used editing features are the three interrelated commands: Cut, Copy, and Paste. Use these commands to quickly copy documentation to either another location in the same program or another program. The list below describes Cut, Copy, Paste and Deleting Documentation differences.

Cut	Removes the selection from the program and places it on the clipboard.
Copy	Copies the selection and places it on the clipboard.
Paste	Inserts clipboard contents into the documentation window at the start and end address.
Delete	Removes selected contents from the documentation window.

The clipboard referred to in the list above is the standard Windows clipboard. Refer to your Windows User's Guide for more information.

To select the information you want to cut or copy, click, hold and drag the pointer over the desired area. Selected items will be highlighted with a different color than your normal workspace color.

Each of the three editing commands, described in detail in the following paragraphs, can be accessed several ways.

Cut

To use the cut feature, select the information you want to cut by:

1. Clicking on the toolbar or select Cut from the Edit menu and the Cut Range dialog box appears. Enter the start and ending address to cut, in the dialog box from and to location. If tags are to be cut with the address, the Include Tags check box must be selected.

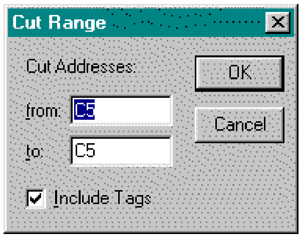


Figure 7.7a

2. Select OK and the selected range of addresses are cut out of the Documentation Window and placed into the clipboard.

Copy

To use the copy features:

1. Click on the toolbar or select Copy from the Edit menu and the Copy Range dialog box appears. Enter the start and ending address to copy into the Copy Range dialog box, from and to locations. If tags are to be copied with the address the Include Tags check box must be selected.

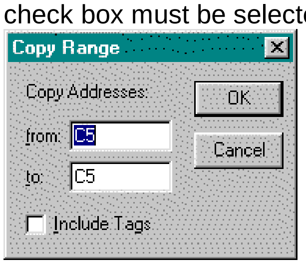


Figure 7.7b

2. Select OK and the selected range of address are copied from the Documentation Window and placed into the clipboard.

Paste

To access the paste feature:

1. Click  on the toolbar or select Paste from the Edit menu and the Paste Range dialog box appears. Enter the start address to paste to, in the Paste Range dialog box at location.

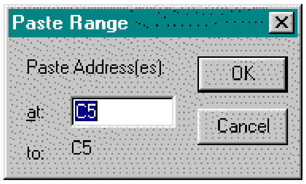


Figure 7.7c

2. Select OK and the addresses in the clipboard are pasted into the Documentation Window starting with the at address.

Delete

To delete an existing address tag, description, and comment:

1. From the Edit menu, select Delete (Alt-E, D) and the delete dialog box appears.

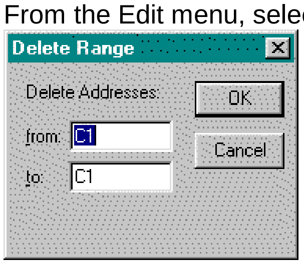


Figure 7.7d

2. Enter the start and ending addresses to be deleted into the Delete Range dialog box from and to locations.
3. Select OK and the selected range of addresses are deleted from the Documentation Window.

Searching for an Address, Tag, or Description

You can find the documentation associated with a specific address, tag, or description by using the Find option. To find an item, select one of the following:

1. From the Edit menu, select Find (Alt-E, F).
2. From the Pop-up menu, select Find.
3. Click  on the toolbar.

After you select Find, the window in Figure 7.8 is displayed.

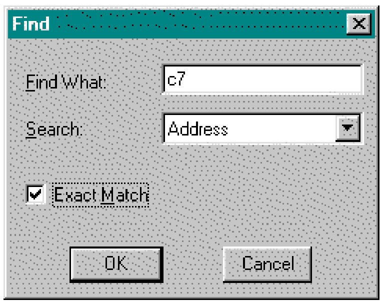


Figure 7.8

Enter the address, tag, or description you wish to find in the *Find What* field. In the *Search* field, select whether you are searching for an address, tags, or description. Find will try to locate the closest match to the entered search information. If you want to find the exact match, click Exact Match. Press OK, and the search item will be located and displayed in the Documentation Window.

Pop-Up Menus

The menu items available with the right mouse button are displayed in Figure 7.9

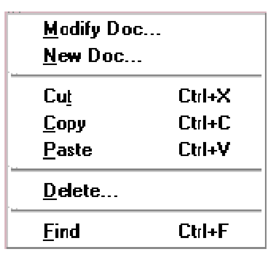


Figure 7.9

Documenting in Ladder

In addition to editing and creating new documentation in the Documentation Window, you can also edit and create new documentation in the logic program as you enter and edit your logic. The Documentation Window can also be used to help you program your logic.

In the logic editor, these features are available:

- Assign Tags.
- Assign Addresses.
- Edit and Create Documentation in Ladder.
- Look up tags and use them in ladder.

These items are discussed in the following sections.

Assign Tags

The Assign Tags option allows you to assign tags, descriptions, and comments to an undocumented address that you are currently using in your ladder program. For example, if you enter an address in an ADD instruction and that address does not have a tag or description, the Edit Documentation window in Figure 7.7 will automatically appear when you move off the address. Enter a tag, description, and comment, and press OK to save the documentation. This allows you to document undocumented addresses as you program without leaving the ladder editor.

To use this feature, select Options, Program Setup (Alt-O, P). Then select the Logic tab dialog. Select Ladder from the Options for drop down window. Then check the Assign Tag in the selection check box.

Assign Addresses

The Assign Addresses option allows you to assign addresses, descriptions, and comments to tags as you use them in your ladder program. For example, if you enter the tag NEW_TAG (and NEW_TAG doesn't exist), the Edit Documentation window in Figure 7.7 will automatically appear with the tag filled in. You can enter the address, description, comment, and press OK to save the documentation. This allows you to assign addresses to tags as you program without leaving the ladder editor.

To access the Assign Address feature, select Options, Program Setup (Alt-O, P). Then select the Logic tab dialog. Select Ladder from the Options for drop down window. Then check the Assign Address in the selection check box.

Editing Documentation in Ladder Editor

You can automatically assign documentation and edit existing documentation by pressing Ctrl L on an address in your ladder program. For example, if you would like to change the documentation for the address 00001 that is used on a contact, move the cursor to that location and press Ctrl L. The Edit Documentation window in Figure 7.7 automatically appears. Enter the tag, description, and comment, and press OK to save the documentation.

Shared Documentation

Address Documentation can be used in two ways:

The first is the traditional style. Where a temporary DATABASE file is not shared and all edits are buffered until the file is saved. If the file is not saved then any edits are lost. Upon saving the temporary DATABASE file is copied to the same destination as the logic file.

A shared file works differently. This file can be shared (Opened more then once.) by different programs. Edits are not buffered but are immediately saved. Upon saving a program the Address Documentation DATABASE file does not get saved, thus Save As does not affect the file at all. A refresh time can be entered that causes the database to reread the file to acquire any new or changed documentation. Also, a shared database can reference a database created for another purpose as long as the fields are map-able to our defaults. A shared file can be created at user direction, but once created is the responsibility of the user to delete.

To modify a shared documentation file select Shared Documentation setup from the Options menu.

Toggling the Display of Documentation

Network headers and the "Tag" and "Description" portions of address documentation can be displayed in the Logic Window. Any or all three of these pieces of documentation can be shown or hidden in the ladder logic. Two methods may be used: through the "Program Setup" within PLC WorkShop or via keyboard "hotkeys." Both methods are described in detail below.

Toggling display through Program Setup

To specify what portions of documentation appear in the Logic Window, select "Program Setup" from the "Options" menu below.

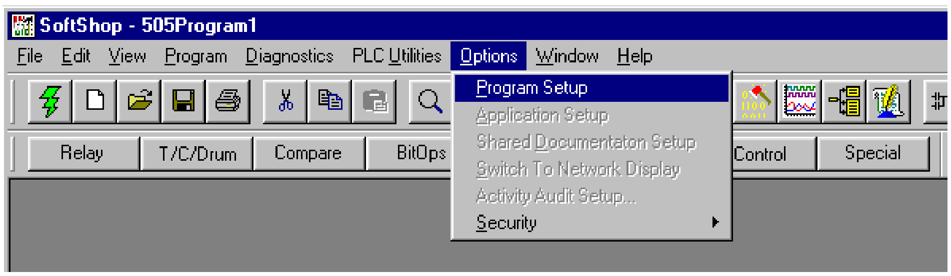


Figure 7.1

When the "Program Setup" dialog box below appears, click the "Logic" tab to display the Logic Window settings.

Select "General" in the "Options for" combo box.

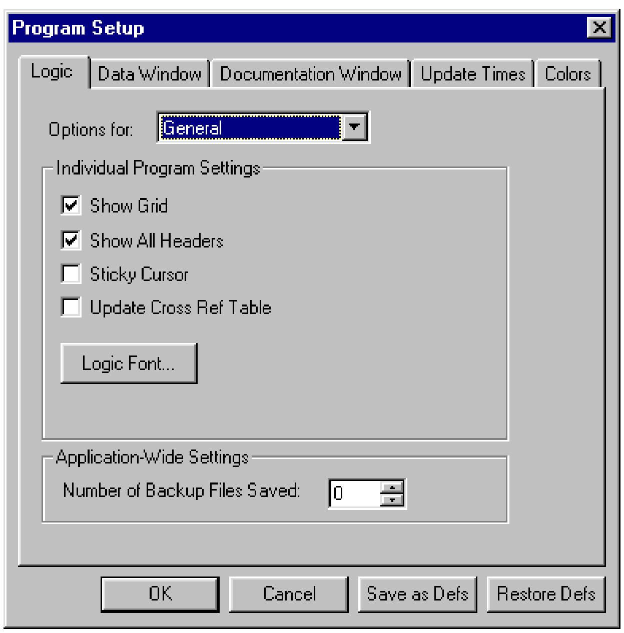


Figure 7.2

Check or uncheck the "Show All Headers" box to display or hide the program and network headers.

Select "Ladder" in the "Options for" combo box

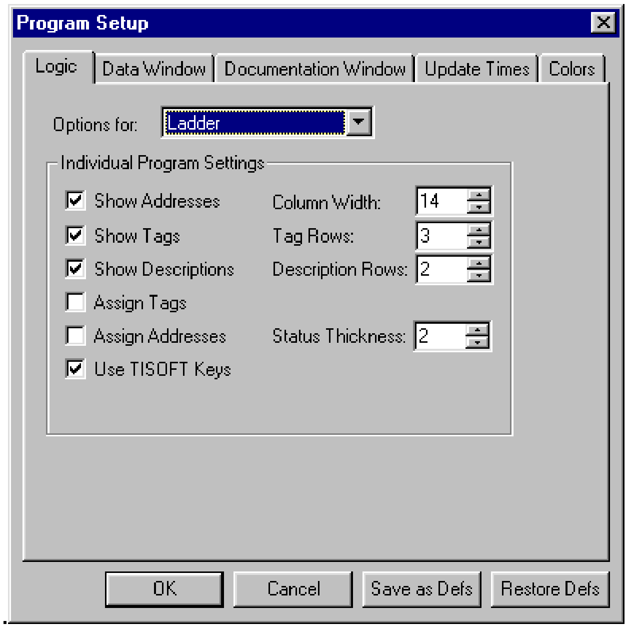


Figure 7.3

Check or uncheck the "Show Tags" and "Show Descriptions" boxes to display or hide the Tag and Description portions of address documentation.

Click "OK" to activate the documentation display selections.

Toggling display with keyboard hotkeys

Network headers and the "Tag" and "Description" portions of address documentation can be displayed or hidden by pressing a set of hot keys while viewing the Logic Window. All hot key combinations which display and hide documentation use the F5 key as described below.

Address Descriptions:

Press F5 to display address Descriptions in the Logic Window. Press F5 again to hide the Descriptions.

- **Address Tags -** Press Ctrl-F5 to display address Tags in the Logic Window. Press Ctrl-F5 again to hide the Tags.
- **Program and Network Headers -** Press Shift-F5 to display program and network headers in the Logic Window. Press Shift-F5 again to hide the headers.

Edit Title Page (Print Only)

This option allows you to display descriptive information at the beginning of your printouts.

To access the Edit Title Page option:

1. Select Title Page Print Editor from the View menu.
2. The following dialog window is displayed (Figure 7.10):

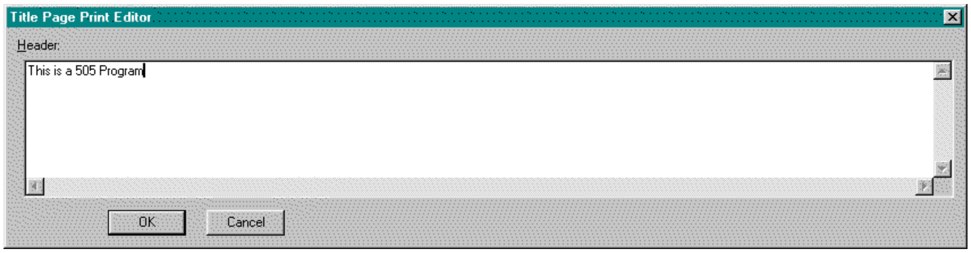


Figure 7.10

3. Enter the text and press OK when you are done.

8 - Analog Alarms

Overview

The analog alarm functions allow you to monitor an analog input signal by setting standard alarm on a process variable (PV) and a target setpoint (SP). Analog alarms are referenced by a user-assigned number from (1-512) depending on the processor type. All analog alarm parameters are stored in Special Memory (S-Memory) when you program the analog alarm. The size of S-Memory is user configurable.

Analog Alarm Editor

The Analog Alarm Editor gives you the ability to display, access, and/or modify analog alarms.

To access the Analog Alarm Editor:

1. Click Alarm from the View menu (Alt+V, A).
2. The Analog Alarm Directory dialog box appears (see Figure 8.1).
3. The dialog box shows the Loop Mode, Alarm number (1-512) depending on the processor type, Alarm Title, and Enable/Disable state.

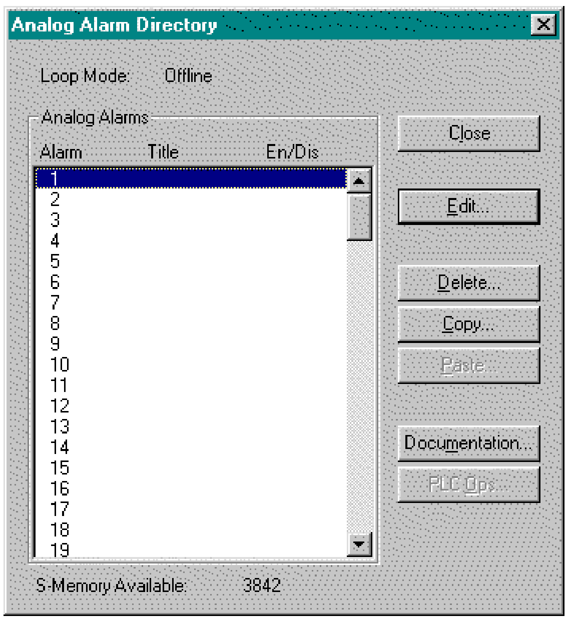


Figure 8.1

4. Select the Analog Alarm number that you want to enter or edit by using the left mouse button or the Arrow keys on the keyboard.
5. To edit the selected alarm number, double-click on the alarm number or click on the dialog box Edit button, or use the keyboard to Tab to the Edit button and press Enter. The following dialog box appears (Figure 8.2).

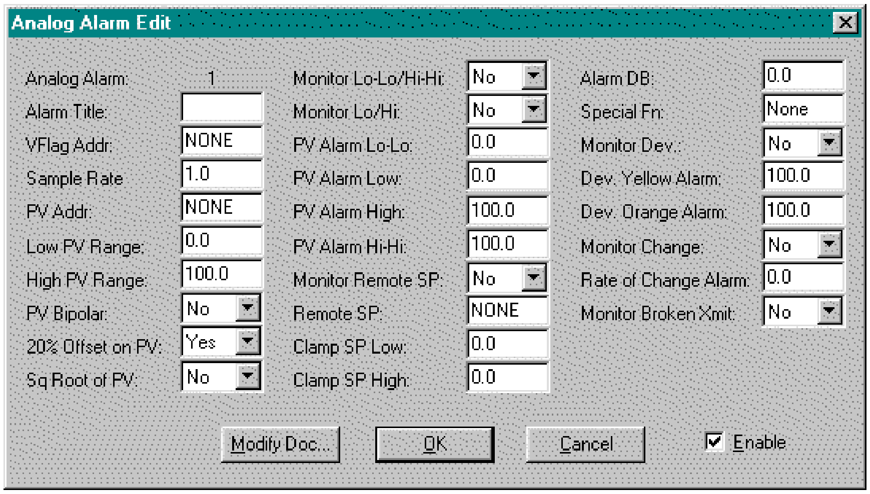


Figure 8.2

To delete Analog Alarm

1. To delete an Analog Alarm, Click Alarm from the View menu (Alt+V,A). The Analog Alarm Directory dialog box appears (see Figure 6.10). The dialog box shows the Loop Mode, Alarm number (1-512) depending on the processor type, Alarm Title, and Enable/Disable state.

NOTE *To view or edit documentation of a selected address. Click on the Modify Doc. button.*

---

2. Select the Analog Alarm number that you want to delete by using the left mouse button or the Arrow keys on the keyboard. Click on the dialog box Delete button or use the keyboard to Tab to the Delete button and press Enter. The following dialog box appears (Figure 8.3).

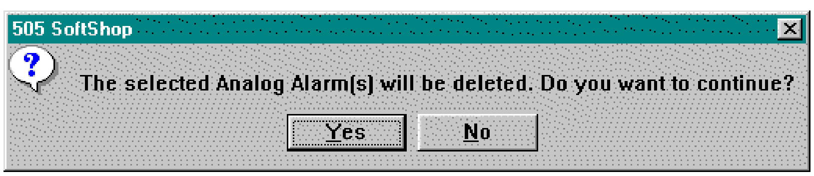


Figure 8.3

3. Select Yes to delete the analog alarm, or No to return to the previous dialog box.

Copy and Paste Alarms

To Copy and Paste an Alarm:

1. Click Alarm from the View menu (Alt+V, A).
2. The Analog Alarm Directory dialog box appears (see Figure 8.1). The dialog box shows the Loop Mode, Alarm number (1-512) depending on the processor type, Alarm Title, and Enable/Disable state.
3. Select the Alarm number that you want to Copy by using the left mouse button or the Arrow keys on the keyboard. The drag and left mouse button or Shift + arrow key can be used to select multiple Alarm numbers. Click on the dialog box Copy button to place the selected Alarm number in the clipboard.
4. Move the cursor to the desired location and click on the dialog box Paste button to paste the clipboard contents into the new location.

Analog Parameters

Figure 8.4 shows a template of the Analog Alarm Edit window.

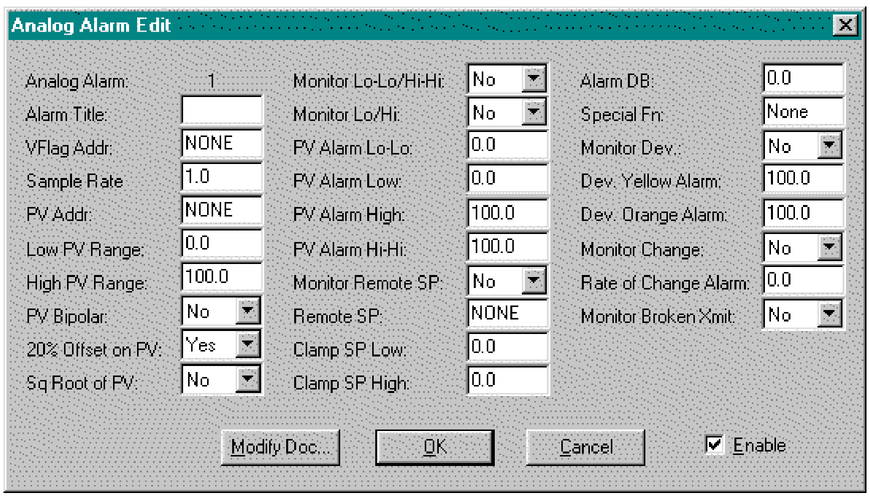


Figure 8.4

Alarm Title

An eight-character title can be entered for each specific Analog Alarm number. The title is optional and can be left blank.

V-Flag Address

Enter an address: C, Y, V, or WY in the ALARM VFLAG ADDRESS field. If you select **NONE,** no data is written from the V-Flags in the analog alarm.

The V-Flags contains the operational data for an analog alarm. The V-Flags comprises the individual bits making up the 16-bit word. The bits are defined in Table 8.1.

Table 8.1 Analog Alarm V-Flags 1

Bit	Analog Alarm Function
1	1=Enable alarm.
2	1=Disable alarm.
3	1=Process Variable is in high-high alarm.
4	1=Process Variable is in high alarm.
5	1=Process Variable is in low alarm.
6	1=Process Variable is in low low alarm.
7	1=Process Variable is in yellow deviation alarm.
8	1=Process Variable is in orange deviation alarm.
9	1=Process Variable is in rate of change alarm.
10	1=Broken transmitter alarm.
11	1=Analog alarm is overrunning.
12	1=Alarm is enabled.*
13-16	1=Unused.

* If a word is selected for the analog alarm V-Flags, bit 12 is written. If a C or Y is selected, bit 12 is not used.

NOTE: *If you program an analog alarm and do not disable it, the controller begins to monitor the programmed variable as soon as you place the controller in Run mode.*

Sample Rate

Enter a time in seconds in the SAMPLE RATE field.

The sample rate determines how often deviation alarm bits and associated math are evaluated. Sample rates are programmable in 0.1-second increments, with alarms checked at least once every two seconds. The sample rate can be any floating point number between 0.1 and 1.6772 x 106 seconds.

Process Variable Address

Enter an address: V, WX, or WY in the PROCESS VARIABLE ADDRESS field.

A process variable must be specified for each analog alarm. The process variable can be taken from the following:

- A word input or output module - The programming table uses a WX or WY address.
- A location in V-Memory - The programming table uses an address in V-Memory.

If you select NONE, the analog alarm does not read an address to obtain the Process Variable.

Low Process Variable Range

Enter the low value of the process variable in the Process Variable Range Field.

You must specify the engineering values that correspond to the lower range of the input span.

High Process Variable Range

Enter the high value of the process variable in the Process Variable Range Field.

You must specify the engineering values that correspond to the upper range of the input span.

Process Variable Bipolar

Select YES or NO to specify analog inputs as bipolar or not. Bipolar inputs span have spans of -5 to 5 volts, or -10 to 10 volts.

Square Root of Process Variable

Select Yes if the input for the process variable is from a device (such as an orifice meter) that requires a square root calculation to determine the correct value to use.

Monitor Low-Low/High-High

Select Yes to have the controller monitor the Low-Low/High-High Alarm; otherwise, select No. The Low-Low/High-High can be entered as values requiring critical action.

Monitor Low/High

Select Yes to have the controller monitor the Low/High Alarm; otherwise, select No. The Low/High Alarm can be entered as values requiring remedial action.

Process Variable Alarm Low-Low

Enter a real number in engineering units; must be less than or equal to low alarm value, and greater than or equal to low range of Process Variable.

Process Variable Alarm Low

Enter a real number in engineering units; must be less than or equal to high alarm value of Process Variable.

Process Variable Alarm High

Enter a real number in engineering units; must be less than or equal to high-high alarm value of Process Variable.

Process Variable Alarm High-High

Enter real number in engineering units; must be greater than or equal to high alarm value, and less than or equal to high range of Process Variable.

Monitor Remote Setpoint

To have the controller monitor the remote setpoint, select Yes in the Monitor Remote Setpoint field. If you select No, the analog alarm uses the current value in the analog alarm variable.

Remote Setpoint

Select NONE if there is no remote setpoint. Otherwise, enter an address: V, K, WX, or WY, or a value in the remote setpoint field.

Clamp Setpoint Low/High

Enter values for the setpoint limits in the CLAMP SETPOINT LIMITS field. If there are no limits, enter zeroes in the High and Low fields.

Alarm Deadband

Enter a value in engineering units for the alarm deadband in the ALARM DEADBAND field. When you specify an alarm deadband, the controller can provide hysteriesis on all alarms except the rate of change alarm to prevent them from chattering when the process variable is near one of the alarm limits.

Special Function

Enter a SF program number in the SF field. Select NONE if no SF program is to be called for execution.

Monitor Deviation

To have the controller monitor the deviation alarm limits select Yes in the Monitor Deviation field; otherwise, select No.

Deviation Yellow Alarm

Enter values in engineering units for the setpoint deviation limits. The deviation alarm bands are always centered around the target or setpoint; i.e., the deviation alarm test is actually on the control error. This value indicates the maximum allowable error (SP-PV) that sets the yellow alarm deviation alarm. The yellow deviation limit must be within the span of the process variable, and it must be less than or equal to the orange deviation alarm.

Deviation Orange Alarm

Enter values in engineering units for the setpoint deviation limits. The deviation alarm bands are always centered around the target or setpoint; i.e., the deviation alarm test is actually on the control error. This value indicates the maximum allowable error (SP-PV) that sets the orange alarm deviation alarm. The orange deviation limit must be within the span of the process variable, and it must be greater than or equal to the yellow deviation alarm.

Monitor Rate of Change

To have the controller monitor the rate of change, select Yes in the Monitor Deviation field; otherwise, select No.

Rate of Change Alarm

Enter a value in engineering units for the rate of change alarm.

Monitor Broken Transmitter Alarm

To have the controller monitor the Broken Transmitter Alarm, select Yes in the Monitor Broken Transmitter Alarm field; otherwise, select No.

If you program the controller to monitor for the broken transmitter condition, an alarm occurs if the raw process variable is outside the valid range designated for the Process Variable. Valid ranges are:

- Bipolar:-32000 to 32000
- 0% offset: 0 to 32000
- 20% offset: 6400 to 32000

9 - PID Loops

Overview

Process and batch control capability is provided using the controller's proportional-integral-derivative (PID) loop functions, illustrated in Figure 9.1. When you program a loop, you can set the same eight alarm types used by analog alarms and described in Chapter 8.

- ❑ • High-high alarm point on the process variable (PV).
- ❑ • High alarm point on the PV.
- ❑ • Low alarm point on the PV.
- ❑ • Low-low alarm point on the PV.
- ❑ • Yellow deviation alarm point referenced to the setpoint (SP).
- ❑ • Orange deviation alarm point referenced to the SP.
- ❑ • Rate of change alarm, for a PV changing too rapidly.
- ❑ • Broken transmitter, for a PV outside the designated valid range.

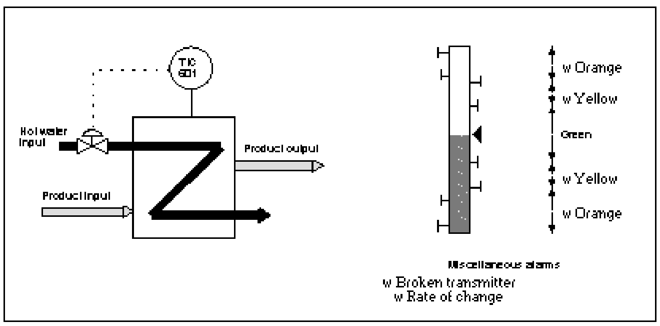


Figure 9.1 Example of Loop Control

The high-high, high, low, and low-low alarms are fixed absolute alarms and may correspond to warnings and shutdown limits for the process equipment itself. The yellow and orange deviation alarms move up and down with the setpoint, and may refer to specification tolerances around the setpoint.

A PV alarm deadband is provided to minimize cycles in and out of alarm (chattering) that generate large numbers of messages when the PV hovers near one of the alarm limits.

An option is also available to call a Special Function Program to initiate a special function calculation. The SF program call can be scheduled on the PV, the SP, or the output.

PID Documentation

505 WorkShop provides a simple way to document PID Loops while you are creating or editing loops. The maximum number of characters you can enter for each header is 16K.

1. To Document a PID Loop, Click on PID Loop... from the View menu or (Alt+V, P). The PID Loop Directory dialog box appears (Figure 9.1a). The dialog box shows the Loop Mode, Loop number (1-64), Loop Title, and Enable/Disable state.
2. Select the PID Loop number that you want to document by using the left mouse button or the Arrow keys on the keyboard. Click on the dialog box Document button or use the keyboard to Tab to the document button and press Enter or (Alt+O). The following dialog box appears (Figure 9.11).

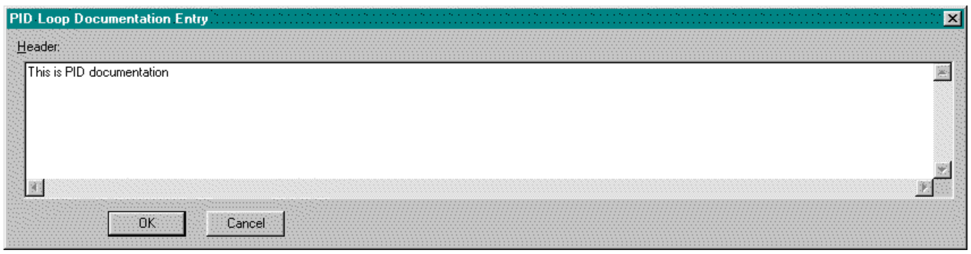


Figure 9.11

Programming PID Loop

The PID Loop Editor gives you the ability to display, access, and/or modify PID Loops.

To access the PID Loop Editor:

1. Click PID Loop from the View menu (Alt+V,P).
2. The PID Loop Directory dialog box appears (Figure 9.1a).
3. The dialog box shows the Loop Mode, Loop number (1-512) depending on the processor type, Loop Title, and Enable/Disable state.

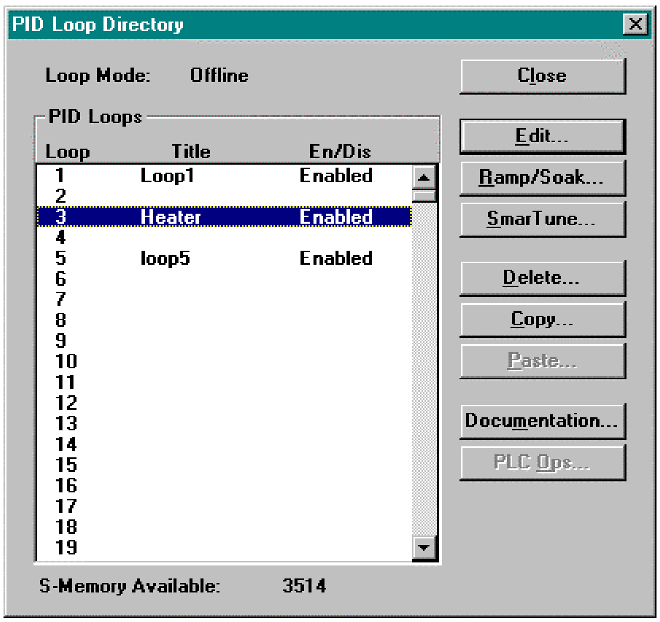


Figure 9.1a

4. Select the PID Loop number that you want to enter or edit by using the left mouse button or the Arrow keys on the keyboard.
5. To edit the selected PID Loop number, double-click on the Loop number or click on the dialog box Edit button, or use the keyboard to Tab to the Edit button and press Enter or (Alt + E). The following dialog box appears (see Figure 9.2).

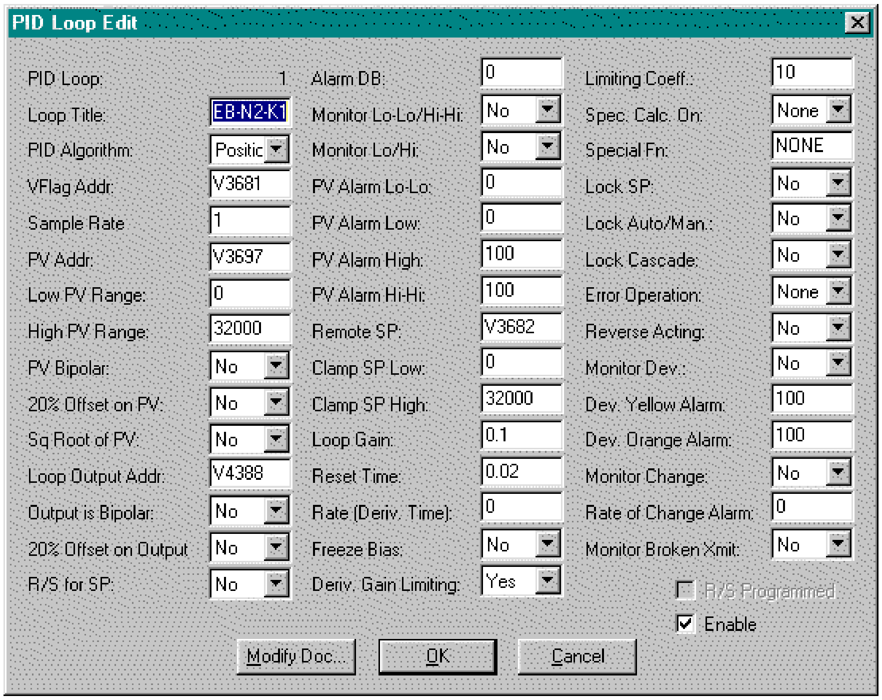


Figure 9.2

NOTE: *To view or edit documentation of a selected address. Click on the Modify Doc. button.*

PID Title

An eight character title can be entered for each specific PID Loop number. The title is optional and can be left blank.

PID Algorithm

Select **POS** for the position algorithm or **VEL** for the velocity algorithm in the PID Algorithm field. For the position algorithm, the position of the device being controlled is computed based on the error. The velocity form of the PID algorithm computes the change in the device position based on the error.

V-Flag Address

Enter an address: C,Y,V, or WY in the LOOP VFLAG ADDRESS field. If you select **NONE,** no data is written from the V-Flags in the PID Loop. You can still control the loop mode by using a SF program to change the control flag bits in the Loop V Flag.

The V-Flags contains the operational data for a PID Loop. The V-Flags corresponds to individual bits making up the 16-bit word. The bits are defined in Table 9.1.

Table 9.1 PID Loop V-Flags 1

Bit	PID Loop Function
1	1=Go to manual mode
2	1=Go to auto mode
3	1=Go to cascade mode
4 & 5	4 5
	0 0 Loop is in manual mode
	1 0 Loop is in auto mode
	0 1 Loop is in cascade mode
6	1=Error is positive
	1=Error is negative
7	1=Process Variable is in high high Alarm
8	1=Process Variable is in high Alarm
9	1=Process Variable is in low Alarm
10	1=Process Variable is in low low Alarm
11	1=Process Variable is in yellow deviation alarm
12	1=Process Variable is in orange deviation alarm
13	1=Process Variable is in rate of change Alarm
14	1=Broken transmitter alarm
15	1=PID Loop is overrunning
16	1=Unused

Sample Rate

Enter a time in seconds in the SAMPLE RATE field.

The sample rate determines how often deviation Alarm bits and associated math are evaluated. Sample rates are programmable in 0.1 second increments, with Alarms checked at least once every two seconds. The sample rate can be any floating point number between 0.1 and 1.6772 x 106 seconds.

Process Variable Address

Enter an address: V, WX, WY, or select **NONE** in the PROCESS VARIABLE ADDRESS field.

A process variable must be specified for each PID Loop. The process variable can be taken from the following:

- A word input or output module - The programming table uses a WX or WY address.
- A location in V-Memory - Uses an address in V-Memory in the programming table. When a special calculation is performed on a process variable, the result is stored in V-Memory where the Loop accesses it.

If you select **NONE**, the PID Loop does not read an address to obtain the Process Variable.

Low Process Variable Range

Enter the low value of the process variable in the Process Variable Range Field.

You must specify the engineering values that correspond to the lower range of the input span.

High Process Variable Range

Enter the high value of the process variable in the Process Variable Range Field.

You must specify the engineering values that correspond to the upper range of the input span.

Process Variable Bipolar

Select YES or NO to specify analog inputs as bipolar or not. Bipolar inputs span have spans of -5 to 5 volts, or -10 to 10 volts.

Square Root of Process Variable

Select Yes if the input for the process variable is from a device (such as an orifice meter) that requires a square root calculation to determine the correct value to use.

Loop Output Address

Enter an address: V, or WY, in the Loop Output Address field. Select **NONE** when you do not want the loop to write the output to an address.

Use the LOOP OUTPUT ADDRESS field to specify the address into which the loop writes the value of the output. You can select **NONE** in situations such as cascaded loops in which the outer loop does not require an output address.

Output is Bipolar

Select YES or NO in the OUTPUT IS BIPOLAR field. If you Select Yes, the output range is -32000 to +32000.

Ramp/Soak for SP

Select **YES** or **NO** in the RAMP/SOAK FOR SP field to indicate whether a ramp/soak program for the loop is to be executed.

Alarm Deadband

Enter a value in engineering units for the Alarm deadband in the ALARM DEADBAND field. When you specify an Alarm deadband, the controller can provide hysteriesis on all Alarms except the rate of change Alarm to prevent them from chattering when the process variable is near one of the Alarm limits. The loop does not exit the alarm condition until the process variable has come inside the alarm limit minus the deadband.

Monitor Low-Low/High-High

Select Yes to have the controller monitor the Low-Low/High-High Loop; otherwise, select No. The Low-Low/High-High can be entered as values requiring critical action.

Monitor Low/High

Select Yes to have the controller monitor the Low/High Alarm; otherwise, select No. The Low/High Loop can be entered as values requiring remedial action.

Process Variable Alarm Low-Low

Enter a real number in engineering units; must be less than or equal to low Alarm value, and greater than or equal to low range of Process Variable.

Process Variable Alarm Low

Enter a real number in engineering units; must be less than or equal to high Alarm value of Process Variable.

Process Variable Alarm High

Enter a real number in engineering units; must be less than or equal to high-high Alarm value of Process Variable.

Process Variable Alarm High-High

Enter real number in engineering units; must be greater than or equal to high Alarm value, and less than or equal to high range of Process Variable.

Remote Setpoint

Select **NONE** if there is no remote setpoint. Otherwise, enter an address: V, K, WX, WY, or LMN in the REMOTE SETPOINT field.

Clamp Setpoint Low/High

Enter values for the setpoint limits in the CLAMP SETPOINT LIMITS field. If there are no limits, enter zeroes in the High and Low fields.

Loop Gain

Enter the value for the tuning constant LOOP GAIN in this field. See Note 1 below.

Reset Time

Enter the value for the tuning constant Reset Time (INTEGRAL TIME) in this field. See Note 1 below.

Rate Derivative Time

Enter the value for the tuning constant RATE DERIVATIVE TIME in this field. See Note 1 below.

NOTE *It is not always necessary to have full three-mode PID control of a loop. Parts of the PID equation can be eliminated by choosing appropriate values for the gain (Kc), reset (Ti), and rate (Td), thus yielding a P, PI, PD, I, and even an ID or a D loop. To eliminate **integra**l action, set (Ti) to infinity. To eliminate **derivative** action, set (Td) to zero. To eliminate **proportional** action, set (Kc) to zero.*

Freeze Bias

Select YES or NO in the Freeze Bias field.

Select No to have the bias adjusted when the output goes out of range. Select Yes to have the bias frozen when the output goes out of range.

Derivative Gain Limiting

Select Yes or No in the Derivative Gain Limiting field to have derivative gain limiting done. If you specify No, then derivative gain limiting is not done, even if a value is entered in the Limiting Coefficient field.

Limiting Coefficient

Enter a value of the Derivative Gain Limiting Coefficient in this field. Typically, Derivative Gain Limiting Coefficient should be in the range of 10 to 20.

In the standard PID algorithm, the algorithm responds excessively to the process noise if the coefficient of the derivative term (rate time/sample time) is significantly above the 10 to 20 range. This causes disturbances that lead to erratic behavior of the process.

To solve this problem, the controller allows you the option of selecting a derivative gain limiting coefficient. Using this coefficient enables the process variable to be filtered with a time constant that is proportional to the derivative time. The PID equations with the derivative gain limiting coefficient follow.

Special Calculation On

Enter a **PROCESS VARIABLE**, **SETPOINT**, or **OUTPUT** in the Special Calculation On field. Select **NONE** if no SF program is to be called for execution.

Special Function

Enter a SF program number in the SF field. Select **NONE** if no SF program is to be called for execution.

Lock Setpoint

Select YES or NO in this field to lock or not lock the SETPOINT.

Lock Auto/Manual

Select **YES** or **NO** in this field to lock or not lock the AUTO/MANUAL.

Lock Cascade

Select **YES** or **NO** in this field to lock or not lock the CASCADE.

Error Operation

Select SQUARED or DEADBAND in the Error Operation field. The Error Squared and the Deadband options are mutually exclusive. Select NONE if there is be no calculation on the error value.

In calculating the control equation, the controller uses an error value equal to or less than 1.0 (% of PROCESS VARIABLE span over 100). Therefore, selecting error squared gives a lower gain for a higher error. The control, equation with error squared is based on signed error squared, instead of the error alone.

To implement a high gain for large errors, and no gain for small errors, incorporate an error deadband. When error deadband is selected, the controller does not take any action on the output if the process variable is within the yellow deviation limits.

Squared error calculation:

en = (SP - PVn) x abs (SP - PVn)

Deadband error calculation:

en = 0 if abs (SP - PVn) < YDEV

en = (SP - PVn) - YDEV if (SP - PVn) > YDEV

en = (SP - PVn) + YDEV if (SP - PVn) < - YDEV

No error calculation:

en = (SP - PVn)

Reverse Acting

Select **YES** for reverse-acting loop in the REVERSE ACTING field. Select **NO** for a direct-acting loop.

A reverse acting loop is defined to have a negative gain; i.e., a positive change in error (SP - PROCESS VARIABLE) results in a negative change in the output from the controller. The value of the output signal decreases as the value of the error increases.

A direct acting loop is defined to have a positive gain; i.e., a positive change in error (SP - PROCESS VARIABLE) results in a positive change in the output from the controller. The value of the output signal increases as the value of the error increases.

Monitor Deviation

To have the controller monitor the deviation Loop limits, select Yes in the Monitor Deviation field; otherwise, select No.

Deviation Yellow Alarm

Enter values in engineering units for the setpoint deviation limits. The deviation Alarm bands are always centered around the target or setpoint; i.e., the deviation Alarm test is actually on the control error. This value indicates the maximum allowable error (SP-PROCESS VARIABLE) that sets the yellow Alarm deviation Alarm. The yellow deviation limit must be within the span of the process variable, and it must be less than or equal to the orange deviation Alarm.

Deviation Orange Alarm

Enter values in engineering units for the setpoint deviation limits. The deviation Alarm bands are always centered around the target or setpoint; i.e., the deviation Alarm test is actually on the control error. This value indicates the maximum allowable error (SP-PROCESS VARIABLE) that sets the orange Alarm deviation Alarm. The orange deviation limit must be within the span of the process variable, and it must be greater than or equal to the yellow deviation Loop.

Monitor Rate of Change

To have the controller monitor the rate of change, select Yes in the Monitor Deviation field; otherwise, select No.

Rate of Change Loop

Enter a value in engineering units for the rate of change Alarm in the Rate of Change Alarm field.

Monitor Broken Transmitter Alarm

To have the controller monitor the Broken Transmitter Alarm, select Yes in the Monitor Broken Transmitter Alarm field; otherwise, select No.

If you program the controller to monitor for the broken transmitter condition, an Alarm occurs if the raw process variable is outside the valid range designated for the Process Variable. Valid ranges are:

- Bipolar: -32000 to 32000
- 0% offset: 0 to 32000
- 20% offset: 6400 to 32000

R/S Programmed

The RAMP/SOAK PROGRAMMED field is a read-only field and contains an **X** (YES) or blank (NO) to indicate the creation of a ramp/soak program for the loop.

To create a ramp/soak profile for a loop, exit the PID Loop Edit window and select the Ramp/Soak button on the PID Loop Directory dialog box.

Delete PID Loop

1. To Delete a PID Loop, Click on PID Loop... from the View menu or (Alt+V, P). The PID Loop Directory dialog box appears (Figure 9.1a). The dialog box shows the Loop Mode, Loop number (1-64), Loop Title, and Enable/Disable state.
2. Select the PID Loop number that you want to delete by using the left mouse button or the Arrow keys on the keyboard. Click on the dialog box Delete button or use the keyboard to Tab to the Delete button and press Enter or (Alt+D). The following dialog box appears (Figure 9.3).

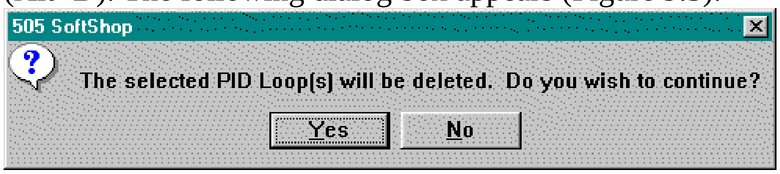


Figure 9.3

3. Select Yes to delete the PID Loop, and No to return to the previous dialog box.

Ramp/Soak

The ramp/soak feature allows you to define a variation for the process variable by specifying the time characteristics of the loop setpoint (Figure 9.4). The capability of varying the loop setpoint can be useful in a number of processes such as heat treating and batch cooking.

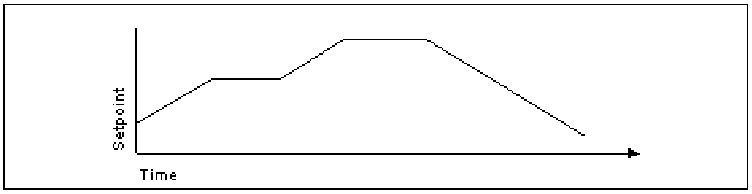


Figure 9.4 Example Ramp/Soak Cycle

You can use simple ramp operations to improve some process startup procedures. For example, the controllers do a bumpless transfer from manual to automatic mode. This transfer holds the process at the initial state when the mode change occurs. A two-step ramp/soak profile can then move the setpoint to a predefined value following the mode change, with minimal disturbance to the process.

Defining Ramp/Soak Steps:

Ramp/Soak is programmed as a set of time periods, or steps. A step can be one of three types: a ramp, a soak, or an end*.*

- The ramp step changes the loop setpoint linearly from its current value to a new value, at a specified rate of change.
- The soak step holds the setpoint constant for a specified period of time. You can guarantee a soak period by entering a deadband value. This form of soaking ensures that the process variable is within a specified deadband around the setpoint for a specified period of time.
- The end step terminates a ramp/soak profile. When the program reaches an end step, the loop remains in automatic mode and holds the setpoint constant.

You can program a status bit for each step of the ramp/soak. This bit is set to 1 when the loop is executing this step. It is reset when the loop leaves the step. This allows for easy tracking in the RLL program.

Controlling the Ramp/Soak Operation

Ramp/Soak operation can be controlled by two methods: allowing the profile to be executed automatically, or by writing values to the variables that control ramp/soak.

Automatic - Whenever the loop changes from manual to automatic mode, the loop begins to execute the ramp/soak profile at the initial step (Step 1). The loop continues to execute the profile until an end step is encountered in the profile. At this point, the loop remains in automatic mode, and the setpoint is held at the last value in the profile.

Using Ramp/Soak Number - Each loop ramp/soak profile has a corresponding 16-bit variable, LRSN that contains the current step. You can monitor LRSN with a SF program and also write a step number to it with a SF program. The ramp/soak profile changes to the step that is currently contained in LRSN. Note that the step number is zero-based. LRSN contains 0 when the profile is on step #1, 1 when the profile is on step #2, etc.

Using the Ramp/Soak Flags - Each loop ramp/soak profile has a corresponding 16-bit variable, LRSF, that contains operational and status information for the profile.

When you program a ramp/soak profile, you may optionally specify a RAMP/SOAK FLAG ADDRESS. When you enter an address into this field, the controller writes the ramp/soak data from LRSF to this address. You can use TISOFT or APT, or design your RLL program to write to the first three bits at the specified address. The controller reads these bits and then writes their status over the corresponding bits in LRSF. This enables you to change the ramp/soak operation by setting/clearing the three bits as needed. The controller ignores changes that you make in bits 4-16.

You can also monitor LRSF with a SF program and write changes to bits 1-3 with a SF program.

NOTE: *The step number is zero-based. LRSN contains 0 when the profile is on step #1, I when the profile is on step #2, etc.*

Editor

The Ramp/Soak Editor allows you to Edit or Delete an existing programmed step or Insert a new one.

To access the Ramp/Soak Editor:

1. Click PID Loop from the View menu (Alt+V,P). The PID Loop Directory dialog box appears (Figure 9.1a). The dialog box shows the Loop Mode, Loop number (1-64), Loop Title, and Enable/Disable state.
2. Select the PID Loop number that you want to enter or edit Ramp/Soak by using the left mouse button or the Arrow keys on the keyboard.
3. To edit, delete, or insert the selected PID Loop number Ramp/Soak, click on the dialog box Ramp/Soak button or use the keyboard to Tab to the Ramp/Soak button and press Enter or (Alt + R). The following dialog box appears (Figure 9.4a).

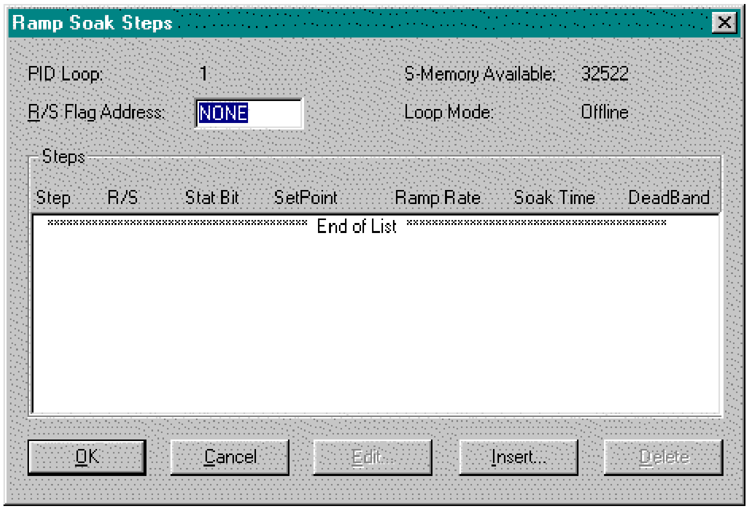


Figure 9.4a

The first field in the table contains the ramp/soak flag address. An entry in this field causes ramp/soak data to be written from the ramp/soak variable (LRSF) to another address, as described above. The address can be either a bit (Y or C) that allocates 5 contiguous bits, or a word (WY or V) that allocates one word for ramp/soak data. The format of the bits in a ramp/soak flag address correspond to the individual bits making up the 16-bit word LRSF. Bits are defined in Table 9.2.

Table 9.2

BIT	**LOOP/FUNCTION**
1	1= Restart at the first step. To restart, toggle bit off, on, then off again. The restart occurs on the trailing edge of a square wave.
2	1= Hold at current step. To hold, set bit on.
3	1= Jog to next step. To jog, set bit on. Jog occurs on the rising edge of a square wave.
4	1= Finish. Indicates ramp/soak is completed.
5	1= Wait. This bit is set during a soak period when the process variable is not within a specified deviation from the SP. The loop holds the soak timer when bit 5 is set.

6 1= Hold in progress at current step.

7-8 Unused (always returns 0).

9-16 1= Contains step number loop is currently executing. Step number is zero-based. Step number contains 0 when ramp/soak is on step #1, 1 when the ramp/soak is on step #2 etc.

4. Enter an address: C,Y,V, or WY in the RAMP/SOAK FLAG ADDRESS field. If you select NONE, no data is written from the Loop Ramp/Soak Flags. See the following table for Loop Functions Bits.

5. To edit the selected Ramp/Soak step number, click on the dialog box EDIT button or use the keyboard to Tab to the EDIT button and press Enter or (Alt + E). The following dialog box appears (Figure 9.5).

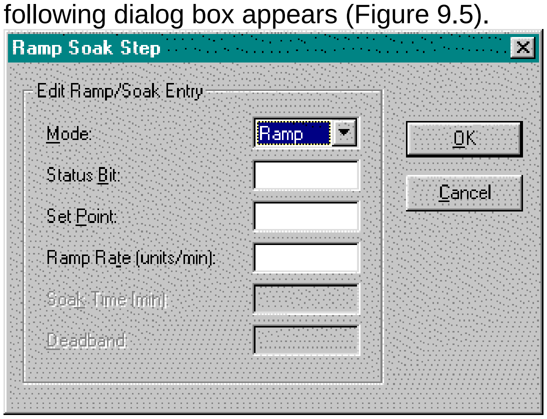


Figure 9.5

6. Select Ramp, Soak, or End from the Mode field:

- The RAMP step changes the loop setpoint (you can enter the setpoint in the Setpoint field) linearly from its current value to a new value at a specified rate of change (you can enter the rate of change in the Ramp Rate field). You can program a status bit (C or Y) for each step of the ramp/soak. This bit is set to 1 when the loop is executing this step. It is reset when the loop leaves the step.
- The SOAK step holds the setpoint constant for a specified period of time (you can enter a soak time in the Soak Time field). You can guarantee a soak period by entering a deadband value (you can enter a deadband in the Deadband field). This form of soaking ensures that the PV is within a specified deadband around the setpoint for a specified period of time.
- The END step terminates a ramp/soak profile. When the program reaches an end step, the loop remains in automatic mode and holds the setpoint constant.

SmarTune Automatic Loop Tuning

The Proportional Component

Temperature control with PID has two regions of operation, the proportional band, and the saturated region. The proportional band is the region above and below the setpoint where the controller output is less than 100%. The heat or cooling output is time proportioned as determined by the PID output. The proportional gain value determines the proportional band.

A typical proportional band might be around ±30°F for a given machinery temperature control, as shown in Figure 9-7. For example, with a setpoint of 300°F, a proportional band of ±30°F would equate to the region between 270°F and 330°F, where the controller would be in the proportional band. Outside of this region, on either side, is the saturated region where the controller output would be 100%, which equates to 100% heating or cooling.

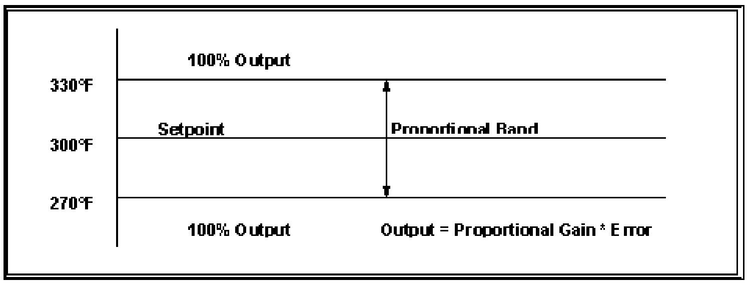


Figure 9-7 Proportional Band

A temperature controller using only the proportional component of the PID expression may experience a steady state error, as shown in Figure 9-8. This error is induced by thermal loading on a temperature zone. As the thermal loading on a temperature zone increases, the magnitude of the steady state error is increased.

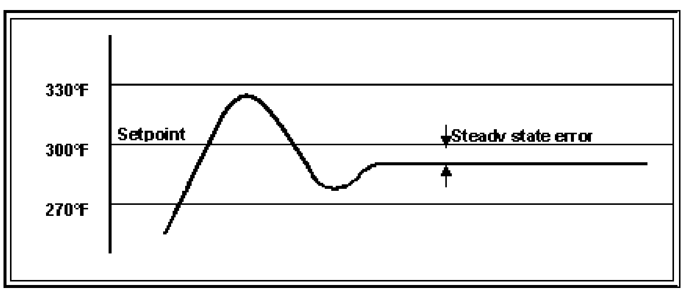


Figure 9-8 Steady State Error

Thermal loading is induced by energy losses to the surroundings, conduction through the machine, as well as the process. A proportional-only controller can resolve this error only to a limited degree.

The Integral Component

The integral term of the PID expression provides a means to eliminate the error in the proportional band. This term is defined as the *Error* integrated over time. Thus, in the case of the steady state error, the output would be increased (or decreased depending on the sign of the *Error*) over time. The amount of the integral adjustment is determined by the magnitude of the *Error*, and the Integral gain. Excessive Integral gain would cause an oscillation about the setpoint. Likewise, minimal Integral gain would not reduce the *Error* in a timely manner and be ineffective.

The Derivative Component

The Derivative term of the PID expression provides a mathematical means for limiting the rate of change of the process variable. As the rate of change becomes larger, the derivative term reduces the output, resulting in the reduction of the rate of change of the process variable. The Derivative gain defines the magnitude of the output reduction as a function of the rate of change of the process variable. Excessive Derivative gain would result in an undesirable output oscillation as the controller continues to eliminate the error.

When the PID gains are set appropriately, the resulting process variable curve would take on the "ideal curve" appearance, as shown in Figure 9-9.

Figure 9-9 Ideal Process Variable Curve

Many factors affect the process variable curve. These factors may take the process beyond where the controller can create the ideal curve. It is the function of the PID SmarTune utility to determine the optimum PID gain values to achieve a response as close to the ideal curve as possible.

Essentially, the SmarTune utility creates a disturbance by initiating a step increase of the PID output. Process variable samples are collected as this increase in output precipitates a change in the process variable. When the sample period is complete, the data collection is analyzed for time lag, gradient, overshoot, steady state error, and oscillation. Using a frequency analysis method, the optimum PID gain values are determined. You can choose to accept the newly calculated gain values, or keep the present PID gain settings.

The SmarTune variable parameters are listed and described in this section. Start Variable is the only variable that must be specified. It names a discrete variable used to activate a SmarTune session. The others may be null.

Variable parameters provide the coupling between a PLC program and SmarTune. If only Start Variable is specified, no program coupling is needed; a session begins when Start Variable becomes true and ends with a loop changing back to its previous mode and SP. Since coupling is done with variables, any program type may be used to monitor and control SmarTune (relay ladder logic, SFPGM, or SFSUB).

Table 9-3 lists the variable parameters used by SmarTune. The following paragraphs describe the parameters.

Table 9-3 Variable Parameters

Name	Type	Allowable Variable Types
Start Variable	discrete	X Y C WX WY V
Abort Variable	discrete	X Y C WX WY V
Ack Variable	discrete	Y C WY V
SmarTune Restart	discrete	X Y C WX WY V

Status Variable	word	WY V
PIN Variable	Word	WX WY V
Previous Mode	Word	V
Previous SP	Real	V
Previous Output	Word	V
Previous Gain	Real	V
Previous Reset	Real	V
Previous Rate	Real	V
Calculated Gain	Real	V
Calculated Reset	Real	V
Calculated Rate	Real	V

Start Variable, Abort Variable, Ack Variable

These three discrete variables allow easy activation/deactivation via an RLL program, as shown in Figure 9-10.

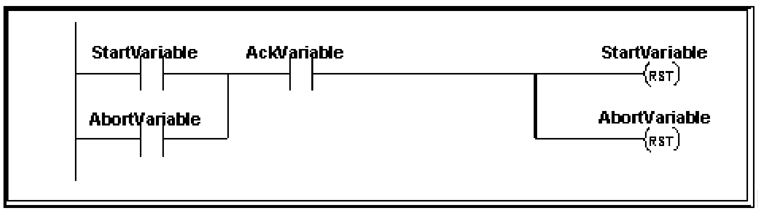


Figure 9-10Example of Activation/Deactivation of Auto Tuning Process

These variables could just as easily be manipulated with IF, IMATH or MATH statements in an SFPGM or SFSUB. Allowed discrete variables include bits in a V-memory word.

- When Start Variable transitions from a false to a true, a SmarTune session is activated.
- When Abort Variable is true, a SmarTune session is deactivated.
- If both are true, a session is deactivated, and Start Variable must transition before a session will be activated.
- If a SmarTune session is already queued or in progress, Start Variable transitions are ignored.

Ack Variable acknowledges that SmarTune has detected that Start Variable or Abort Variable is true. It is used to synchronize Start Variable and Abort Variable program logic with SmarTune. If not used, Start Variable and Abort Variable should remain true for a relatively large amount of time. What constitutes a large amount of time depends on program size and time slice assignments. See the discussion for Activation Time Slice for further guidance.

SmarTune Restart

If this discrete variable is true, then SmarTune is restarted completely. SmarTune will act as if a run-program-run transition occurred. If SmarTune Restart is specified in more than one configuration, all are tested for true and acted upon.

Status Variable

This word variable reports on the current state of a session. Three bits are used in the word to allow easy use by an RLL program. Bit 2 is set when a SmarTune session is completed, with or without errors or warnings. If bit 3 is also set then an error was detected. Similarly, if bit 4 is set, then a warning condition occurred. If only bit 2 is set, then a SmarTune session completed with no errors or warnings. See Table 9-4 for a complete listing. Note that entries with X's represent ranges of values.

PIN Variable

PIN Variable and PIN are provided to force a two-step procedure to be followed before a loop is tuned. To use this feature, PIN Variable and PIN must both be set. If PIN Variable is a null or PIN is zero, then SmarTune activation is a one-step procedure dependent only on Start Variable. If both are specified, then PIN Variable must equal PIN or a SmarTune session will not be started or queued.

Previous Mode

If Previous Mode is used, SmarTune sets it to a value, which will switch a loop to its pre-session mode when written to a loop's LVF. This was conceived for use when Automatic Download has been configured as false, but may be used for other purposes. If Automatic Download is false, a loop is left in manual mode with its output set to Safe Output when a tuning session has completed. When Automatic Download is true, a loop is switched back to its previous mode and is assigned its previous SP on completion.

Previous SP, Previous Output, Previous Gain, Previous Reset, Previous Rate

You can use these five parameters to record the prior SP, Mn, Kc, Ti, and Td of a loop before a SmarTune session starts. See Previous Mode for a short discussion on why they would be configured.

Calculated Gain, Calculated Reset, Calculated Rate

You can use these three variables to record the tuning values calculated by SmarTune. See Previous Mode for a short discussion on why they would be configured.

Table 9-4 lists the loop tuning errors written to the Status Variable word.

Table 9-4 Status Code Bit Values

r	c	e	w	f	g	h	i	j	k	l	m	n	o	p	q

r – reserved (always 0)
c – complete (operation is complete)
e – error (error condition detected)
w – warning (warning condition detected)
f->q meanings dependent on c-e-w bits
x – irrelevant for this condition

rcew fghi jklm nopq	Description
0000 xxxx xxxx xxxx	SmarTune in progress or not active
0000 0000 0000 0000	Not active
0000 0000 0000 1000	Waiting in SmarTune queue
0000 0000 0001 0000	Waiting for Loop to enter manual mode
0000 0000 0001 1000	Wait 1 (PV value stabilize)
0000 0000 0010 0000	Wait 2 (PV value stabilize)
0000 0000 0010 1000	Wait 3 (PV value stabilize)
0000 0000 0011 0000	Calculating Tuning Parameters
0100 0000 0000 0000	SmarTune complete with no errors or warnings
0101 xxxx xxxx xxxx	SmarTune complete with warning(s)
0101 xxxx xxxx xx01	Data questionable, tuning may not be reliable
0101 xxxx xxxx xx10	Data questionable, tuning is not reliable
0101 xxxx xxx0 01xx	Sample interval too large for optimal tuning
0101 xxxx xxx0 10xx	Small PV changes; Step too small?
0101 xxxx xxx0 11xx	PV near span low; Range marginal?
0101 xxxx xxx1 00xx	PV near span high; Range marginal?
0101 xxxx xxx1 01xx	Small output change; Step too small?
0101 xxxx xxx1 10xx	Output near span low; Range marginal?
0101 xxxx xxx1 11xx	Output near span high; Range marginal?
0101 xxxx xx1x xxxx	PV changes before output; Noisy signal?
0101 xxxx x1xx xxxx	PV changes inconsistent with output; Noisy signal?
0101 xxxx 1xxx xxxx	Gain clamped to high/low limit
0101 xxx1 xxxx xxxx	Reset clamped to high/low limit
0101 xx1x xxxx xxxx	Rate clamped to high/low limit
0110 xxxx xxxx xxxx	SmarTune complete with error(s)
0110 0000 0000 0000	Unanticipated error
0110 xxxx xxxx 0001	PIN mismatch
0110 xxxx xxxx 0010	Loop would not go to Manual Mode
0110 xxxx xxxx 0011	Loop not completely under SmarTune control

0110 xxxx xxxx 0100	SmarTune timeout (Maximum time exceeded)
0110 xxxx xxxx 0101	Not enough free memory
0110 xxxx xxxx 0110	Out of required system resources
0110 xxxx xxxx 0111	PV greater than high stop
0110 xxxx xxxx 1000	PV lower than low stop
0110 xxxx xxxx 1001	PV change too small
0110 xxxx xxxx 1010	Operation aborted
0110 xxxx xx01 xxxx	Sample interval (LTS) range error (allowed range: 0.1 ms to 2 hours)
0110 xxxx xx10 xxxx	PV (LPV) or output (LMN) range error (range < 0.00001)
0110 xxxx xx11 xxxx	Sample size too small (probably would never happen) size < 33 (increase STEP or decrease NOISE)
0110 xxx0 01xx xxxx	PV/output inconsistent 1; Noisy PV/output signal?
0110 xxx0 10xx xxxx	PV/output inconsistent 2; Noisy PV/output signal?
0110 xxx0 11xx xxxx	PV/output inconsistent 3; Noisy PV/output signal?
0110 xxx1 00xx xxxx	PV/output inconsistent 4; Noisy PV/output signal?
0110 xxx1 01xx xxxx	PV/output inconsistent 5; Noisy PV/output signal?

Table 9-5 lists the value parameters used by SmarTune, with the default values and the ranges possible for each.

Table 9-5 Value Parameters

Name	**Default Value**	**Range**
Max Time	30.0 minutes	0 to 71582 minutes (maximum is about 49 days)
Noise Band	0.005 of PV range	PV range (engineering units)
Step Change	0.07 of PV range	PV range (engineering units)
Wait Time	0.5 minutes	same as Max Time
PIN	0 (PIN not required)	0 to 32767
Automatic Download	TRUE	TRUE/FALSE
Calculate Derivative	FALSE	TRUE/FALSE
Safe Output	use Previous Output	Previous Output, 0 to 32000
High Stop	0.8 of PV range	PV range (engineering units)
Low Stop	0.2 of PV range	PV range (engineering units)
Largest Gain	8000000.0 %/%	real
Smallest Gain	0.0000008 %/%	real
Largest Reset	8000000.0 minutes	real
Smallest Reset	0.0000008 minutes	real
Largest Rate	8000000.0 minutes	real
Smallest Rate	0.0000008 minutes	real
Activation Time Slice	0	0:not configured here, 1 to 255 ms
Calculation Time Slice	0	0:not configured here, 1 to 255 ms

Table 9-5 Value Parameters

Max Time

Max Time is a time in minutes. When a SmarTune session is started, a timer is set to this value. If that timer expires before the session has completed, the session is aborted with an error (see Table 9-4).

Noise Band

When electrical signals are converted to values, they vary randomly by insignificant amounts. An insignificant amount is application dependent. Noise Band gives a value in engineering units denoting the boundary between a significant and an insignificant change. If a PV value differs from a prior value by a Noise Band or greater amount, then a PV change has occurred. Otherwise the PV is considered unchanged. An incorrect Noise Band setting could cause some errors and warnings in Table 9-2. A correct setting may be calculated from hardware specifications, or determined by experiment and observation, or both.

Step Change

SmarTune works best with a PV change of about 7%. Changing Mn proportional to the ratio between Step Change and PV span accomplish this change. Step Change is specified in engineering units of the PV. If a PV span is 0 to 60 degrees and Step Change is 5 degrees, then Mn would be changed by about 2667 (5/60 * 32000). Due to round-off error, the actual value might be slightly different. This example is based on a Mn span of 0 to 32000. If a 20% offset on output is selected for a loop, a Mn change of about 2133 (Mn span of 25600) would be accomplished. See Table 9-2 for possible warnings and errors associated with Step Change.

Wait Time

The SmarTune sample algorithm looks for a PV to change by Step Change or to quit changing. Wait Time is required to determine when a PV has quit changing. If a PV value does not change by a Noise Band amount within a Wait Time period, then it has stopped changing.

PIN

PIN and PIN Variable are provided to force a two-step procedure to be followed before a loop is tuned. To use this feature, PIN and PIN Variable must both be set. If PIN is a zero or PIN Variable is a null, then SmarTune activation is a one-step procedure dependent only on Start Variable. If both are specified, PIN Variable must equal PIN or a SmarTune session will not be started or queued.+

Automatic Download

If Automatic Download is true, a loop tuning session is accomplished with minimum additional support. After tuning values are calculated, three actions are taken:

- Calculated Kc, Ti, and Td are written to a loop.
- The loop is changed to its prior mode.
- The loop's SP is assigned its prior value.

Calculate Derivative

If Calculate Derivative is false, only Kc and Ti are calculated, and Td is set to zero. If Calculate Derivative is true, Kc, Ti, and Td are calculated.

Safe Output

Safe Output is a Mn value that will not cause any harm to a process. The default is to use the loop Mn value just prior to a tuning session start.

High Stop

If a PV goes above High Stop, Mn is set to Safe Output and an error is declared (see Table 9-4).

Low Stop

If a PV goes below Low Stop, Mn is set to Safe Output and an error is declared (see Table 9-4).

Largest Gain, Largest Reset, Largest Rate

If a calculated value is larger than a configured value, then it is reduced to a configured value and a warning is declared (see Table 9-4).

Smallest Gain, Smallest Reset, Smallest Rate

If a calculated value is smaller than a configured value, then it is increased to a configured value and a warning is declared (see Table 9-4).

Activation Time Slice, Calculation Time Slice

These two values set how much impact SmarTune will have on PLC scan time. If zero in all configurations, a default will be used (2 milliseconds). Otherwise, in each category, the largest value specified will be used.

Activation Time Slice controls how responsive SmarTune is to tuning session requests. Increase this value if SmarTune is taking an excessive amount of time to start a tuning session. Remember that as this value is increased, PLC scan time will increase.

Calculation Time Slice determines how much real time it will take to calculate tuning parameters. It is possible a calculation might take 20 seconds or more of PLC time. If a PLC has a scan time of 10 milliseconds and Calculation Time Slice is 2 milliseconds, then a 20-second calculation would take about 120 seconds in real time: (10ms + 2ms) / 2ms * 20s = 120s. The above formula is an algebraic simplification of: Xs / (2ms / 12ms) = 20s where X is real time in seconds. This value should be increased if a SmarTune session takes an excessive amount of time with a status of calculating (see Table 9-4 and Status Variable). Remember that as this value is increased, PLC scan time will increase while a SmarTune session is calculating.

10 - FTLogger/FTTrend

FTLogger

FTLogger collects data from multiple PLC devices to selected database files. Addresses are entered in a spreadsheet format with up to 1,000 addresses per log window. Data is logged based on a specific time frame or used defined trigger.

Configuring FTLogger Devices

To configure a device, you need to set up addresses using the Device Source Configuration screen. Only devices from running servers can be configured. Opening a log window starts the server and it will continue to run throughout the duration of the application. The Device Source Configuration screen is disabled if no servers are running (Figure 0-8).

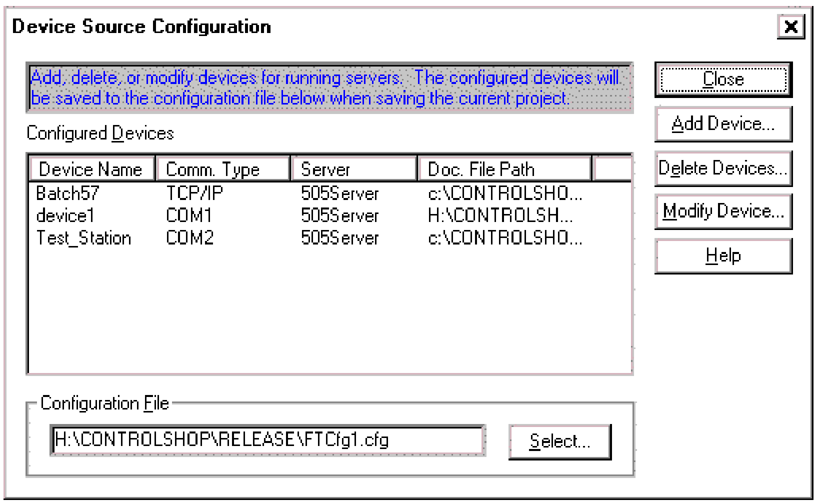


Figure 0-8

- Two different messages are available in the static message box.
  - **Add, delete, or modify for running servers. The configured devices will be saved to the configuration file below when saving the current project.** This message appears when devices are attached to the server and a log window is opened.
  - **One or more devices must be configured for <server name> before any addresses/tags can be entered into the active trend window.** This message appears when no devices are attached to the server and a new log window is opened.
- **Configured Devices -** Lists the configured devices for all the servers. If a new configuration file is selected, no devices are available.
- **Configuration File -** Change the configuration file at any time by clicking on the Select button to bring up the Select Configuration File (Figure 0-9). All the configured devices are saved here when selecting Save or Save As from the File menu.

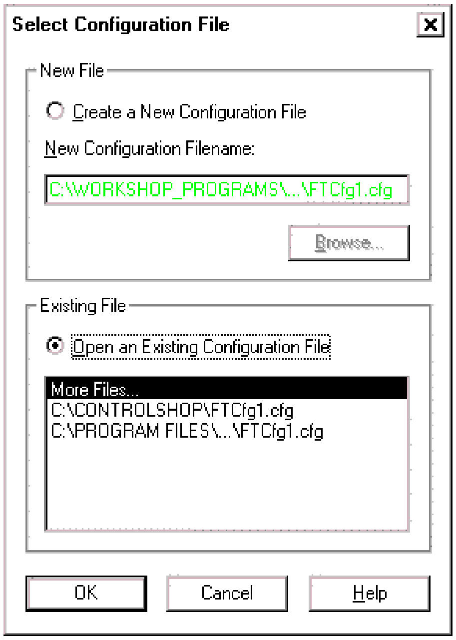


Figure 0-9

The Select Configuration File screen allows you to create a new configuration file or open an existing file

- **New File -** Create a New Configuration File - Opens a new file.

  New Configuration Filename - The path for the configuration file defaults to the same path as the executable. Modify the new file or path by clicking the Browse button. The Select a New Configuration File screen appears (Figure 0-10).

  Enter a file name or select a name using the drop down arrow in the Look in field.

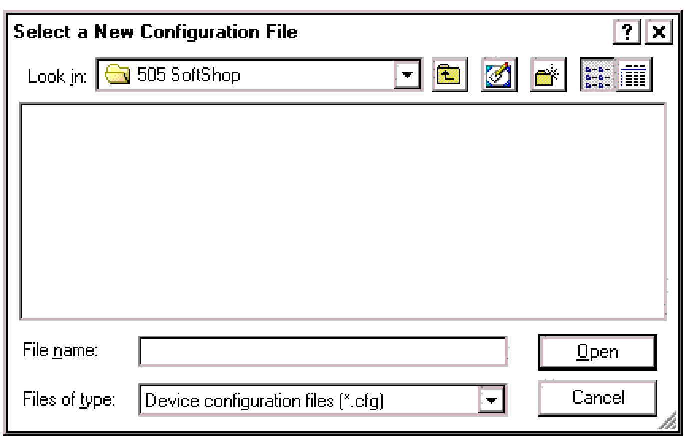


Figure 0-10

- **Existing File -** Open an Existing Configuration File - Opens an existing file.

The list contains the Most Recently Used (MRU) configuration files read from the registry similar to the MRU list of .LGR files from the File menu. Select **More Files** to browse for other existing configuration files.

The database file must include at least four fields (address, tag, descrip, and comment) in order. If these fields are not found, a documentation-mapping dialog appears to map user fields to FasTrak fields described above. A tag is synonymous with a Workshop symbol and a description is synonymous with a Workshop label. Every device configured must have a corresponding documentation file.

Adding Devices

1. To add a device, click Add Device from the Device Source Configuration screen.
   Result: The Add Device screen appears (Figure 0-11).

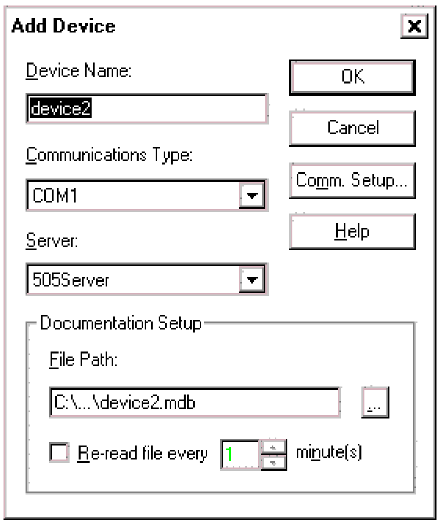
   

   Figure 0-11

2. Enter information in the Device Name, Communications Type, and Server fields.
   Notes: The first running server (alphabetically) appears in the Server field.
   Modification of the device name automatically updates the doc filename so that they mirror one another.
   The Re-read file every <> minutes field is not available at this time.

3. Click the Comm. Setup… button to initialize vendor specific device settings as shown for Siemens 505 (Figure 0-12).

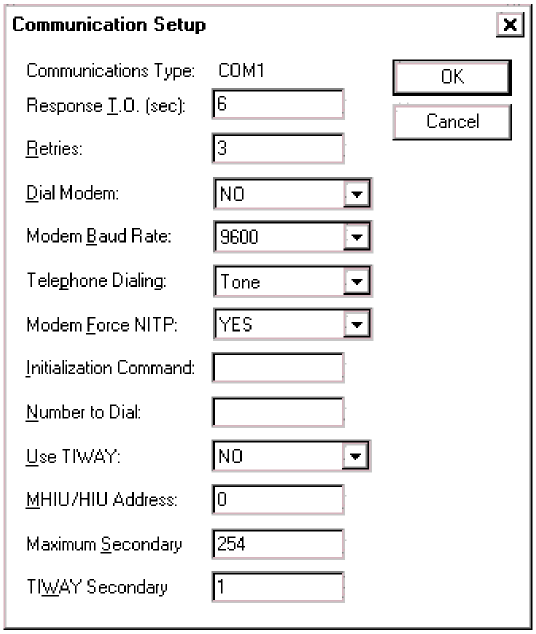


Figure 0-12

4. Click OK to send the configuration to the server.
   Result: A connection is made to the doc database file for reading and writing.

If the database you are connecting to does not match specifications such as a mismatch between table name or field names, the Documentation Database Mapping screen appears to map specific fields so that the software knows where various types of documentation reside (Figure 0-13).

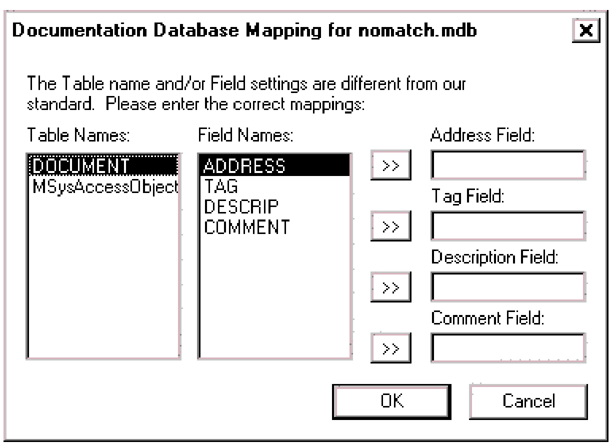


Figure 0-13

For the mapping to be successful, the fields names must map to the table names. The fields that are mapped must be of type "Text" for address, tag, and description and "Memo" for the comment field. The lengths of the fields must be less than or equal to FasTrak's lengths for all fields except the comment field. The maximum character length is:

- Address = 64
- Tag = 32
- Description = 96

Failure to provide a valid mapping prevents the device from being added.

Modifying Devices

1. To modify a device, click Modify Device from the Device Source Configuration screen.
   Result: The Modify Device screen appears (Figure 0-14).

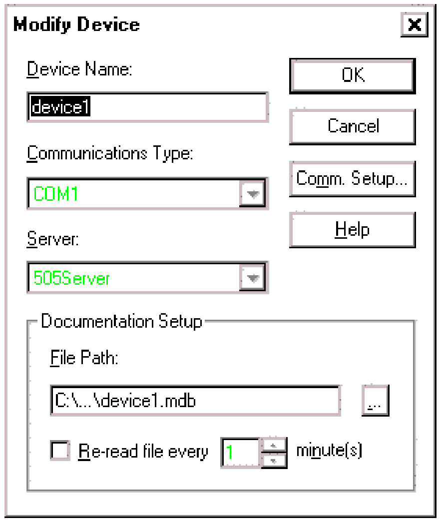


Figure 0-24

2. Enter information in the Device Name and Documentation Setup fields. If the device is offline, you can change the Communications Type field.
3. Click OK to send the configuration to the server.

Deleting Devices

1. To delete a device, click Delete Device from the Device Source Configuration screen.
   Result: The Delete Devices screen appears (Figure 0-15).

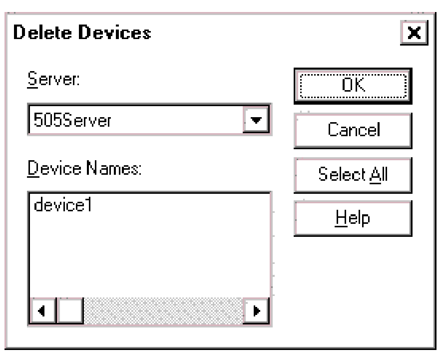


Figure 0-35

2. Select a server from which you wish to delete a device.
3. Select a name in the Device Names field to delete devices.
4. Click Select All and OK to delete all the devices for a selected server.

Creating a Log Window

To create a new Log Window, perform the following steps:

1. Launch FTLogger
2. Click  on the Toolbar or select New Program from the File menu.

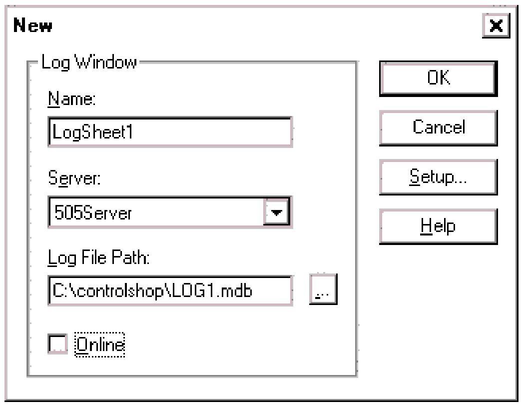


Figure 0-2

3. Enter a unique name, up to 128 characters, for the new Logger Window spreadsheet in the Name field.
4. Select the appropriate communications server (see appendix ? for listing of sever device associations).
5. The default path appears in the Log File Path field. Enter a path if different from the default.
6. Click the radio button Online to select the device port.
 Result: Communication is established as soon as the addresses are inserted in the log window.

   Until addresses are entered into the log window, the server is not actually online. Once the log window is created, pause logs online by selecting Mode ® Pause Logging. By remaining online there is a significant speed increase because the port remains open and the addresses logged remain in the server.
7. 7. Click the Setup button.
 Result: The Log Window Setup screen appears (Figure 0-3).

Note *Selecting OK (on the New dialog box) without configuring the Data Window Setup will set values to the factory defaults.*

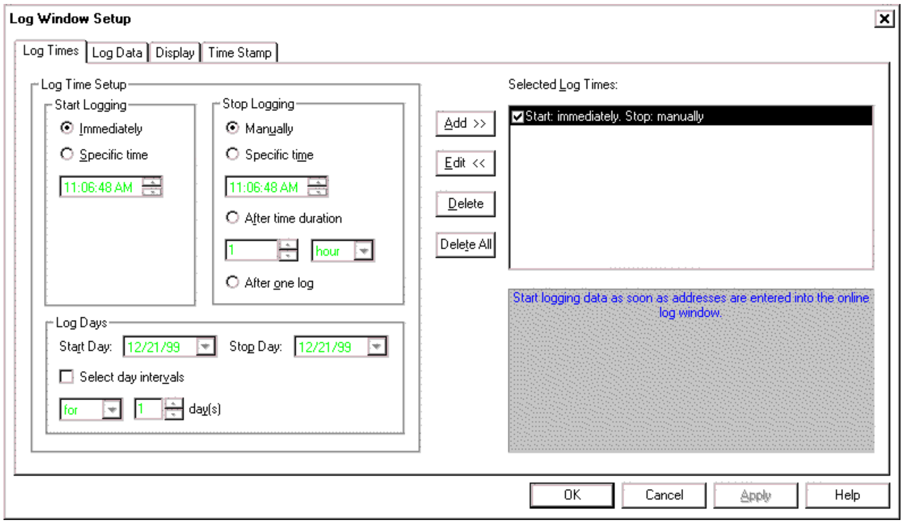


Figure 0-3

Log Times

Log Times allows you to set when the log window starts and stops logging data within the window.

- **Start Logging**

  Immediately - Data is logged to the Md. logfile as soon as data is entered into the log window provided that the log window is online and active.

  Specific time - Data is read from the server and logged after the time indicated just below the radio button.

- **Stop Logging**

  Manually - Data is logged to the .Md logfile as long as the log window is open. Deselecting the Online menu or selecting **Mode → Pause Logging** overrides this action and suspends logging.

  Specific time - Logging terminates after the time indicated just below the radio button.

  After time duration - Logging occurs for the duration of the time selected starting from the time indicated under the Start Logging group box.

  After one log - Logging is terminated after the first and only log. Closing the log window prior to the selected stop logging time overrides the time and terminates all logging from the closed window. A message appears warning you of the consequences of shutting down this window.

- **Log Days**

  Start Day - Select which days to apply the start times.

  Stop Day - Select which days to apply the stop times.

  Select day intervals - Select start and stop log times that can be applied a number of times to a number of days. Options are For (for consecutive days) or Every (for incremental days).

- **Selected Log Times**

  Enable and disable each log time by clicking the check box next to the log description. FTLogger defaults to logging data immediately after addresses are entered into the log window and continues until the log window is closed.

Add>> - When a log time is configured, click Add **>>** to save the log time and display it in text form in the Selected Log Times list box. 256 log times may be added. Two examples of log times in text format are listed below:

Start logging at 1:30 PM on 10/25/1999. Stop logging after 5 minutes.

Every 2 days - start logging at 1:30PM. Stop logging at 2:30PM.

Edit>> - Highlight text and click Edit>> to edit the log time. The text is removed and updated in the static text box with settings that correspond to the text.

Log Data

The Log Data screen allows you to configure how the server reads the data and updates the cache that determines how data is logged (Figure 0-4).

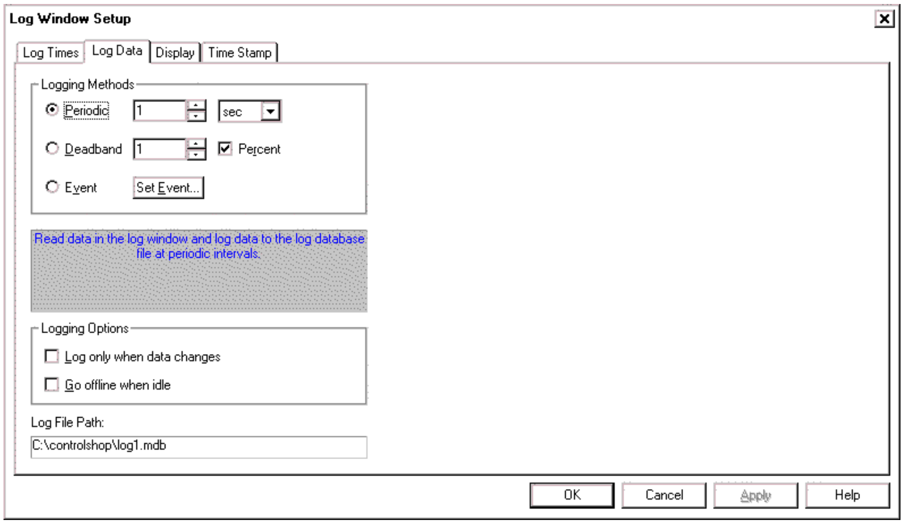


Figure 0-4

- **Logging Methods**

Periodic - Updates the cache in time intervals specified in edit box. The format is msec, sec, min, and hour. Minimum value is 0 (updates as quickly as possible) and maximum value is 24 hours.

Deadband - Data variation value or a percentage of a data variation. Compares each data value from the addresses in the log window with the deadband value. If a particular value fluctuates by the deadband amount (+ or -), the server updates the cache for that data item. Deadband then compares to the updated value for that particular item.

Event - Updates only if the event created in Set Event is true and selected. The Set Event screen appears (Figure 0-5).

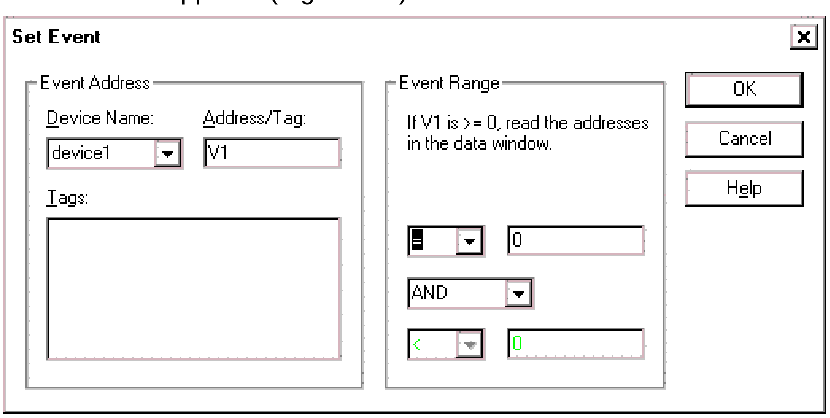


Figure 0-5

Event Address

Device Name - Displays the device name.

Address/Tag - Displays the event address.

Tags - Displays addresses and tags. Click an item and the corresponding address/tag appears in the Address/Tag edit box.

Event Range

An expression involving the event address. The addresses in a data window are updated as long as the expression is true.

The text at the top of the edit box describes the complete expression with the event address.

The first combo box and adjacent edit box contain the low trigger range. The combo box contains the following options: <, >, <=, >=, =, and !=.

The last combo box and adjacent edit contain high trigger range. The combo box contains the following options: <, >, <=, >=, =, and !=.

The middle combo box represents the trigger relationship. This combo box contains the following strings:

END
AND
AND NOT
OR
OR NOT

When END is selected, only the low trigger range is used in the expression.

Logging method description box

Describes the log method that is currently checked. This text changes whenever you select a different log method.

Logging Options

Update only when data changes - Log addresses on an individual basis only if the address value (including format) or status has changed.

Note: *When this box is not checked the addresses are logged every time they are read from the server.*

Go offline when idle - Allows the devices in the corresponding log window to be offline. When offline with a device, the port is only closed when no other log windows or clients are using the ports for their own data.

Display

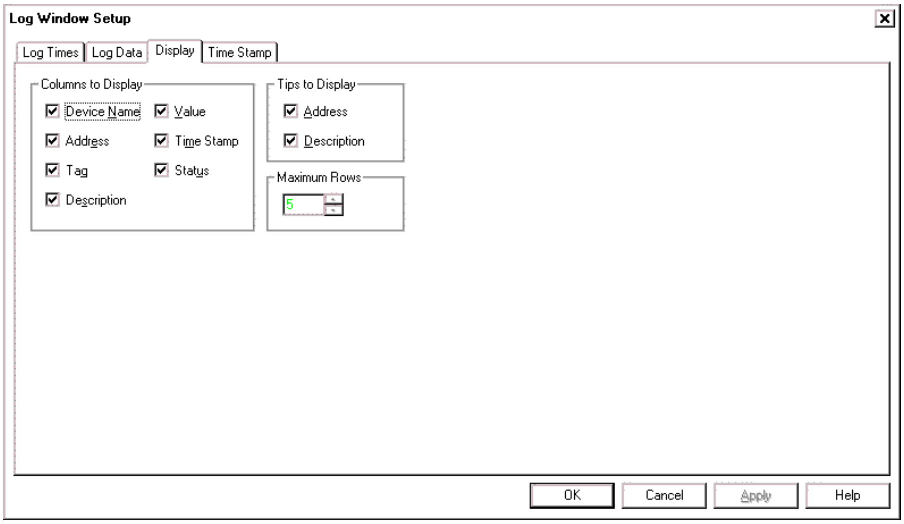


Figure 0-6

- **Columns to Display -** Turns on and off columns in the log window (see Log Window Appearance). All the columns are checked by default except for Value, Time Stamp, and Status.
- **Tips to Display** - Scroll through the Tag combo box in a log window to display the corresponding address and description (if it exists) of the highlighted tag. The address and description are referred to as doc tips and can be turned on and off with these boxes. All tips are checked by default.

Time Stamp

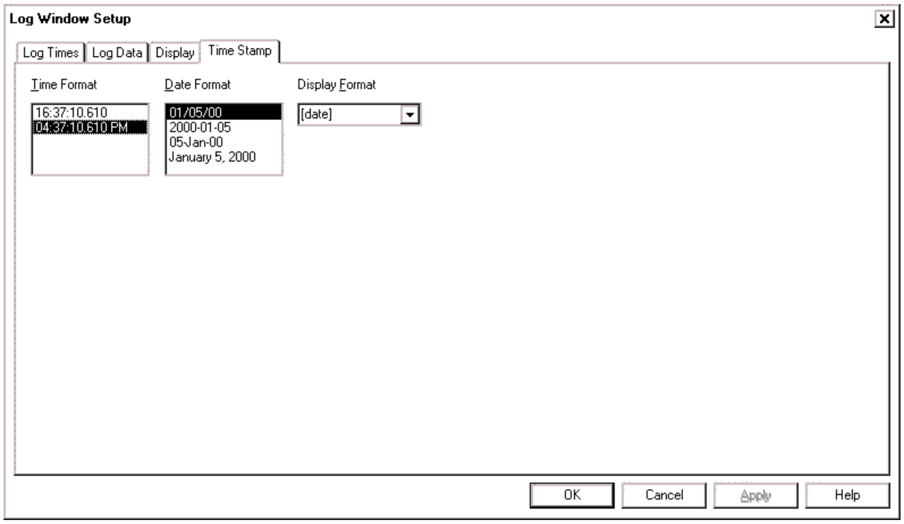


Figure 0-7

- **Time Format -** Displays the current time (time at which the page was activated) in a standard format or military format. The default is standard format.
- **Date Format -** Displays the current date (date at which the page was activated) in four different formats.

  The default is format 1 (**[month]/[day]/[year]** - each represented by 2 digits).

  [month]/[day]/[year] - each represented by 2 digits.

  [year]-[month]-[day] - The year is displayed in 4 digits while the month and day are displayed in 2 digits.

  [day]-[month]-[year] - The day and year are displayed in 2 digits while the month is displayed as a 3-character abbreviation.

  [month] [day], [year] - The month is completely spelled out, the day is displayed in minimal digits and the year is displayed as 4 digits.

Using the Log Sheet

The Log Sheet displays information about the devices such as address, tag, description, value, timestamp, and status. You can change the way the data is presented and determine your status - online or offline.

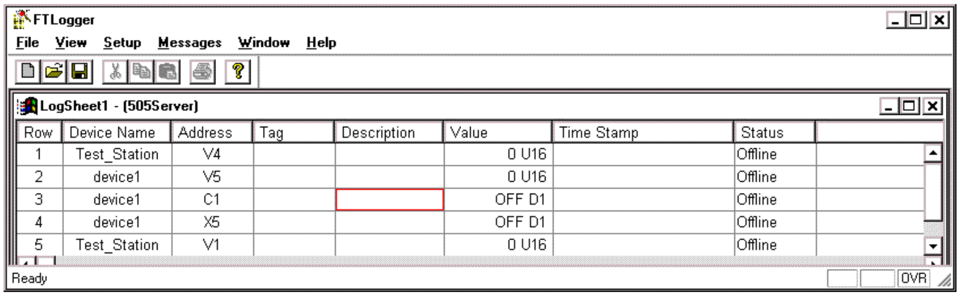


Figure 0-56

Column Layout

Each log window is a spreadsheet with columns that can be sized in different methods.

- Click and hold the left mouse button when between the columns. Move the cursor to the right or left to increase or decrease the column.
- Double-click between the columns. This action adjusts the left column to display the largest string within this column. When no strings are displayed in the column, no adjustment is made.

Row

Each row represents a single address from a single device that is configured in the server from which the log window is connected. Currently, rows are not sizeable.

Device Name

The device name changes the current device on the status bar. Click in this column to start an auto-search for all the device names for the log window server. Select the device name when it appears.

Address

The address corresponds to the device in the Device Name column of the same row. If a device name is not listed, the address applies to the current device listed on the status bar. Press any key or click in this column to enter text in the address field. Entering an address creates an update of all other columns. You can enter a range of addresses by using the fill format equations below.

Address\+-[count][\increment][\format]

Address1-Address2[\increment][\format]

Tag

A tag is a very short description of an address and can be used in place of an address. Devices must exist if you want to edit this field. Tags are stored in database files containing addresses and their corresponding descriptions.

Press any key or click in this column to start an auto-search for all the tags for the device indicated on the status bar (if an empty row) or entered in the Device Name column. A tag can be assigned to an address by first entering the address in the Address column. Then click in the column and type a unique tag. Press Enter or the Tab key or click outside the column to enter the tag into the database and associate the new tag to the address listed in the same row. Delete a tag by clicking in the column, removing all the text, and entering information by one of the methods described. Editing a tag creates an update of all other columns.

Description

An address and/or tag must be present to edit this field. Descriptions are stored in database files containing addresses and their corresponding tags. Press a key or click in this column to display a vertically scrolling multi-line edit box containing the current description if one exists. Approximately six rows can be viewed at one time. The edit box accepts carriage returns. Press the Tab key or click outside the column to enter the description into the database. Delete a description by clicking in the column, removing all the text, and entering the information by one of the methods described. Editing a description creates an update of all other columns.

Value

This column allows you to view data values. When the log window is offline or online-pause, the last value read from the server is displayed. Addresses or tags entered while the log window is offline or online-pause display a zero value.

Time Stamp

This column allows you to view the time that the data value was read and put into the server cache.

Status

Status of the data read from the device. Any errors reading addresses will be displayed here. Since many addresses are read in one packet, a status error is usually displayed in multiple rows.

For example, assume we configured 200 "V" addresses in a 505 PLC and are displaying them all in FTLogger. If a packet can consist of 256 addresses and we enter V201, a status error is displayed on all 201 rows because the entire packet has failed by including 1 erroneous address. Unlike SoftShop, ControlShop does not implement the type of error checking that would prevent a user from entering an invalid address so it is up to the PLC/device to report these types of errors. When offline, all the status rows display the string Offline. This field is not displayed by default.

Data Menu Functionality

The following options are available from the Data menu.

Format

Displays and logs the value where the cursor is located as well as any highlighted values.

Size

Two sizes - 16-bit or 32-bit

Next/Previous

Based on the vender address, displays the next device register address.

Based on the vender address, displays the previous device register address.

Fill

Fills the log window with addresses that are set up in the Fill screen (Figure 0-6).

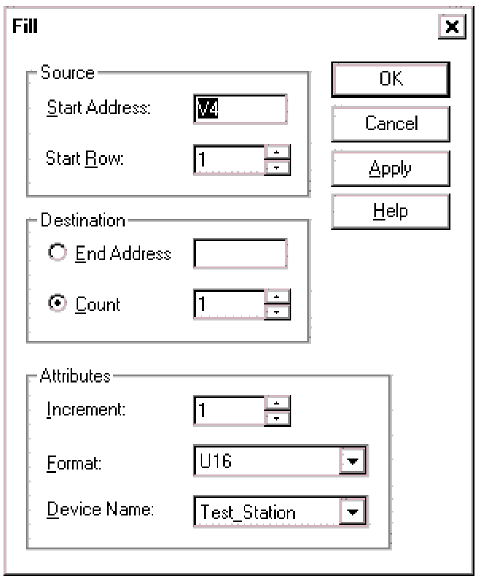


Figure 0-6

"Start Address" holds the first address in a range of addresses. "Start Row" will hold the starting address location in the current log window. Destination can be in the form of an end address or a count. For example, a start address of 400001 and an end address of 400003 will display 400001, 400002, and 400003 in the log window. Selecting a count of 3 instead of an end address will give the same results. However, if increment is set to 2 and end address is selected addresses 400001 and 400003 will be displayed while addresses 400001, 400003, and 400005 will be displayed if count is selected. The format will determine how to display the data values for the addresses selected.

Clicking on the "Apply" button will fill the log window with the addresses specified in the Fill dialog without closing the dialog. Pre-existing addresses in the log window that fall within the fill range will be overwritten if "OVR" is enabled in the status bar. After clicking on Apply, the following fields will be modified:

Start Address – This will now indicate the address at the new start row.

Start Row – This will now indicate the first empty row after the last address in the fill.

Fill Example

Below is an example of how you might set up the fill dialog.

Start Address = 400001

Start Row = 5

End Address = 400003

Increment = 1

After "Apply" is selected and the data window is modified, the Fill dialog will have the following values:

Start Address= 400004

Start Row = 8

End Address= 400006

Increment = 1

Editing Features

Cut, Copy, and Paste can be performed on log window rows a variety of different ways. The menu items can be found under the main "Edit" menu along with hot keys. Alternative hotkeys shift-delete, ctrl-insert, and shift-insert are supported for Cut, Copy, and Paste respectively. Toolbar items are also provided for all three features. Log window rows can be cut from one log window and pasted into another across applications provided that the log window's servers match.

Cut, Copy, and Paste will observe highlighted rows similar to excel. Highlighting log window rows is also similar to excel with the exception of column highlighting.

Select All

The "Select All" menu item under the main "Edit" menu can be used to unconditionally highlight all the rows in the active log window. Features such as delete, format, size, on, off, cut, copy, and paste can be performed on the entire log window.

Log Status Window

The log status window shown below lists successful and failed logs as well as any miscellaneous errors that may occur while attempting to log data (dependant upon setup criteria) for all log windows that are currently logging data. If no online log windows are up, the message "No logs are currently active" will be displayed. A successful log entry in this window will read "At [date and time of log] data was successfully logged from [log window name] to [Access log filename]". A failed log entry in this window will read "At [date and time of log] data failed to log. [error message describing reason of failure]". Each message type can be displayed in a different color (See section 5.2). Entries into this window may be stopped at any time by checking the "Pause Logging" menu under the main menu "Mode". Un-checking the "Pause Logging" menu will re-enable the window so that it may receive status messages once more. Status messages that occurred when the window was paused are lost.

A floating menu brought on by a right mouse button click while the cursor is over this window can be utilized to allow a user to clear or hide the window.

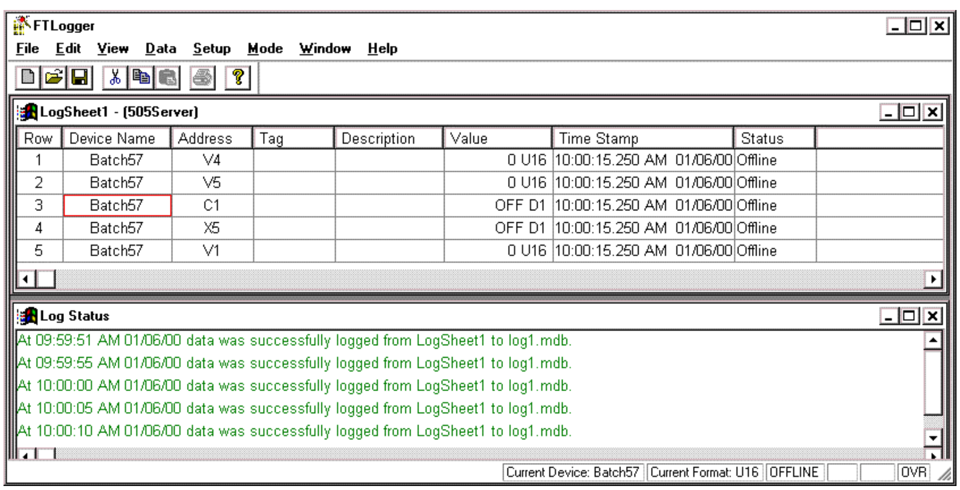


Figure 0-7

Log Status Window Creation

This window will be created and hidden upon startup. To access this window you must select the "Log Status" menu under the main menu "View". A check mark next to "Log Status" will indicate that we are currently viewing this window. Clicking again on this menu will remove the check mark and hide the window. Hiding the window will remove all the entries. Entries will not be added to this window until is viewable or iconized. If this window is full, entries will be removed from the top down as new ones are appended to the bottom.

Log Status Setup Dialog

The log status setup dialog can be accessed by clicking on the "Log Status" menu under the main menu "Setup". This dialog is shown below.

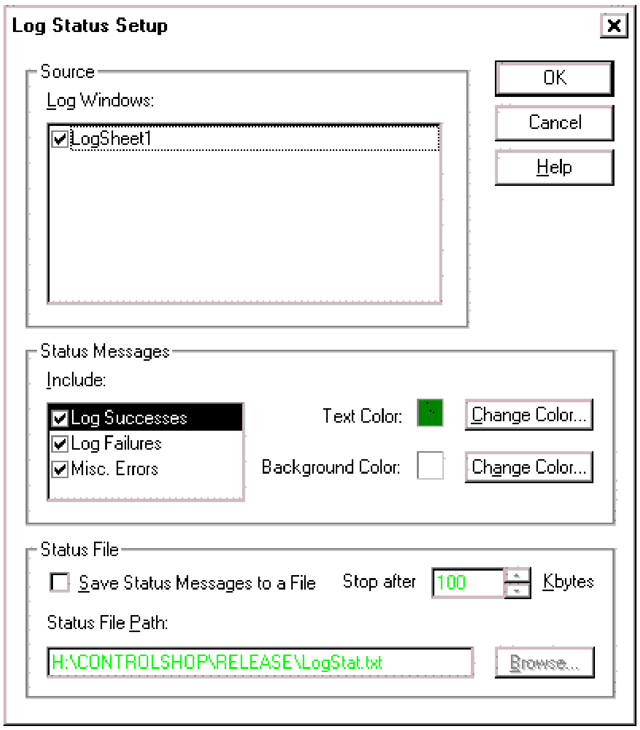


Figure 0-8

This dialog allows you to select which log windows we will be receiving status messages from (Source group box) as well as which status messages we wish to receive (Include group box). These messages can also be written to a file by checking the "Save Status Messages to a File" box in the "Status File" group box and including a path and filename. You can click on the "Browse" button to bring up a common file search dialog as shown above. If you've selected an existing file and click on "OK" the software will ask you if you would like to append the new messages to this file or clear the file and begin from scratch. Writing status messages to a file vs. a window allows more flexibility in two ways. First, a file can hold more entries than a window. Secondly, status messages from one log session can be appended to messages from a previous session.

Error Response Dialog

The Error Response dialog is accessed by clicking on the ""Error Response…" menu under the main menu "Setup". This dialog allows you to configure how the logger should react to a variety or error situations. This dialog is displayed below.

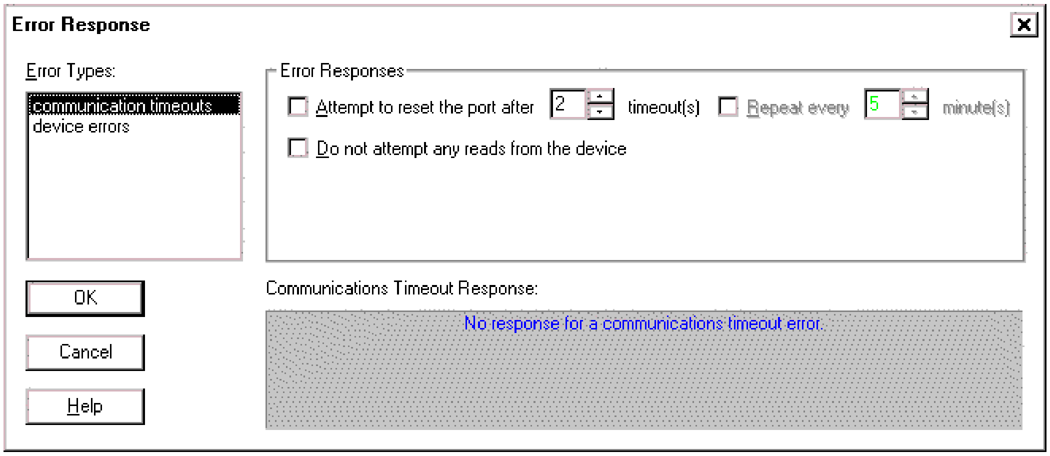


Figure 0-9

The "Error Types" list box hold all the error situations in which a response can be preprogrammed. Error situations include communication timeouts, device errors, log database file errors, and status file errors. The "Error Responses" group box lists all the possible responses for the item that is currently highlighted in the "Error Types" list box. If no items are checked in the group box, no action will be taken when the corresponding error situation arises. The text in bottom right corner gives a more detailed description of the error response for the error type that is highlighted.

FTTrend

FTTrend displays real-time data collected from multiple PLC devices or historical data collected from the FTLogger. This information is displayed in a graph with up to 256 points per graph. You can view different devices on the same graph and control how often the information is gathered and displayed.

Creating a Trend Window

To create a new Trend Window:

- First launch FTTrender
- Click  on the Toolbar or select New Program from the File menu.

Result: A default trend widow appears (figure 0-10)

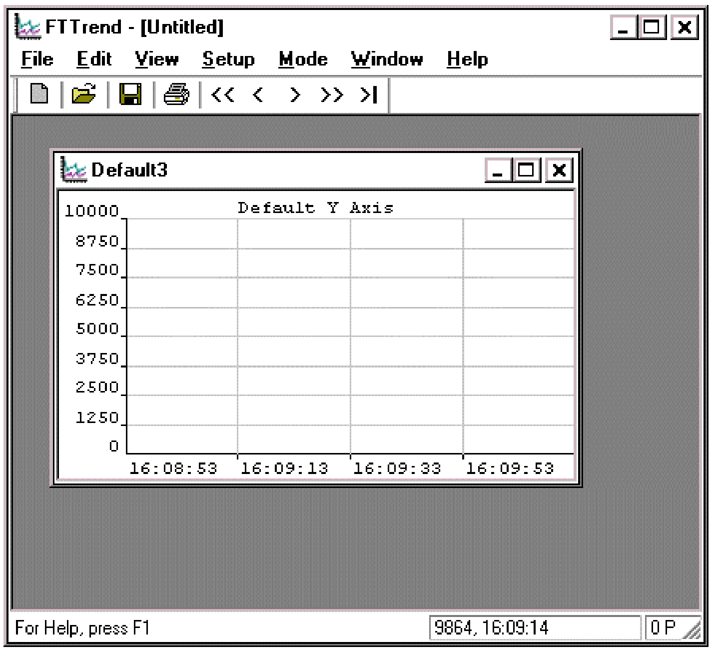


Figure 0-10

Configuring FTTrend Devices

Before any addresses can be inserted into a trend window, devices must be configured so that the servers know where to read the addresses. This is done by selecting "Device Source…" from the Setup menu list. Only devices from "running" servers can be configured. Opening up a trend window starts a server for the selected server. Once a window is opened and the server starts running, it will continue to run throughout the duration of the application. The menu "Device Source…" will be disabled if no servers are running. When this menu item is selected or when a trend window is created with no devices attached to the server, the dialog below will be displayed. If the dialog was prompted by a new trend window, the text in the box at the top will read "One or more devices must be configured for [server name] before any addresses/tags can be entered into the active trend window".

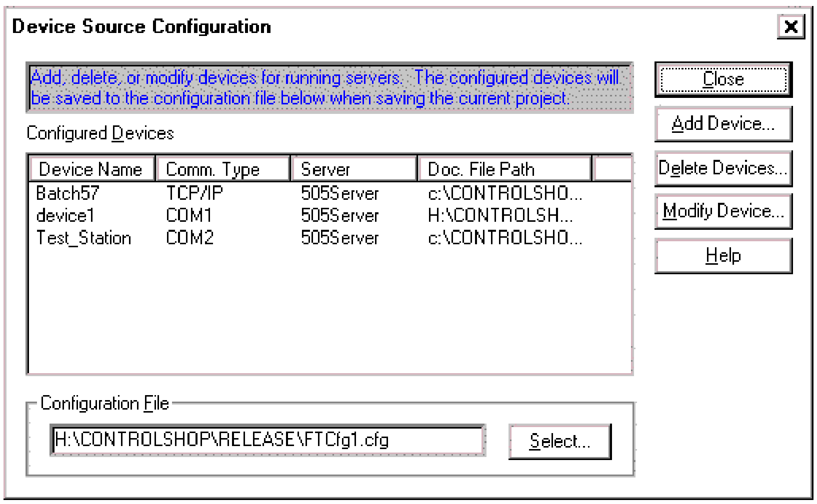


Figure 0-11

All the configured devices for all the servers will be listed here sorted by device name. The name of the recently selected configuration file will be displayed in the static box on the bottom. If a new configuration file is selected this window will show nothing under "Configured Devices". You can change the configuration file at any time by clicking on the "Select" button to bring up the "Select Configuration File" dialog shown below. All the configured devices will get saved here when you attempt a Save or Save As from the File menu.

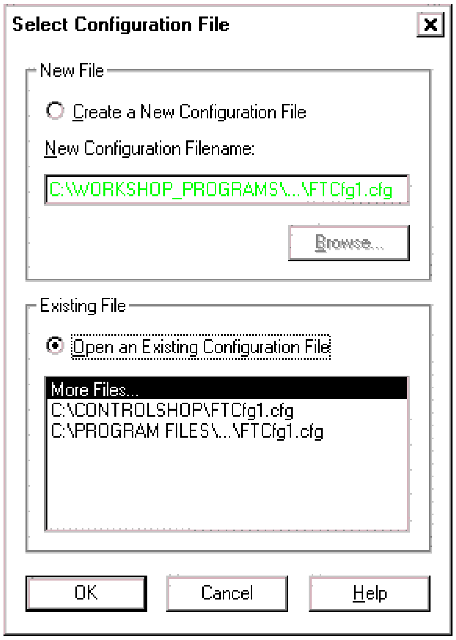


Figure 0-12

Within the "Select Configuration File" dialog, you can create a new configuration file or open up an existing one. If no configuration files exist, the "New File" radio button will be checked and the edit box within this group will hold a default configuration path and file. The path for the configuration file will default to the same path as the executable. You can modify the new file and/or path by editing this manually or select a new path and/or file by clicking on the "Browse" button to bring up common file dialog displayed below.

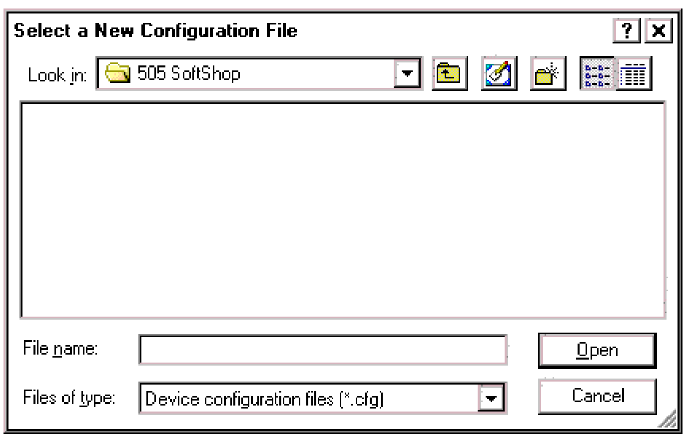


Figure 0-13

The list box within the "Existing File" group box will hold the most recently used (MRU) configuration files read from the registry similar to the MRU list of .LGR files under the "File" menu

The database file must include at least 4 fields (ADDRESS, TAG, DESCRIP, and COMMENT) in order. If these fields are not found, a documentation-mapping dialog will be displayed to map user fields to FasTrak fields described above. A tag is synonymous with a Workshop symbol and a description is synonymous with a Workshop label. Every device configured must have a corresponding doc file.

Adding Devices

To add a device select the "Add Device" button and the dialog below will be displayed.

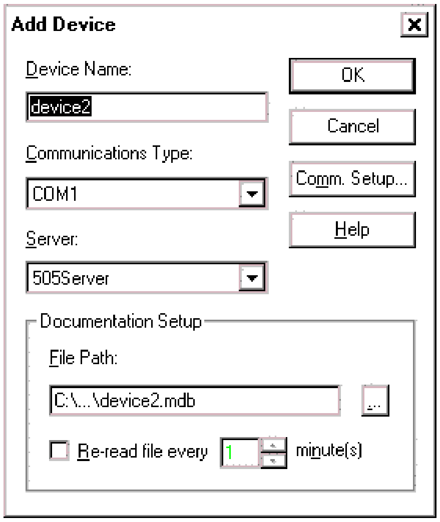


Figure 0-14

The first running server (alphabetically) will be displayed in the "Server" combo box along with any other running servers. Modification of the device name will automatically update the doc filename so that they mirror one another. You can select a different doc path and/or file by manually editing this string or by selecting a new or existing doc file by clicking on the "…" button to bring up a common file select dialog. Since database files can be shared with other applications a "Re-read" check box is included along with a time increment. When this is checked, the database will be re-read and the tags and descriptions will be modified in the trend windows to reflect any changes in the database. This can create a short wait time depending on how many tags are in the database.

Select the "Comm. Setup…" button to initialize vendor specific device settings as shown below for Siemens 505.

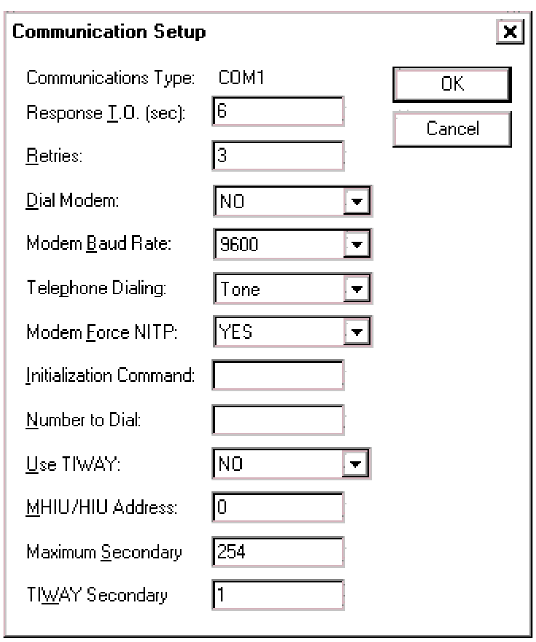


Figure 0-15

When "OK" is selected from the "Add Device" dialog, the configuration is sent to the server and a connection is made to the doc database file for reading and writing.

If the database you are connecting to does not match specifications (i.e. there is a mismatch between table name or field names) then a dialog will be displayed to map specific fields so that the software knows where various types of documentation reside. This dialog is shown below.

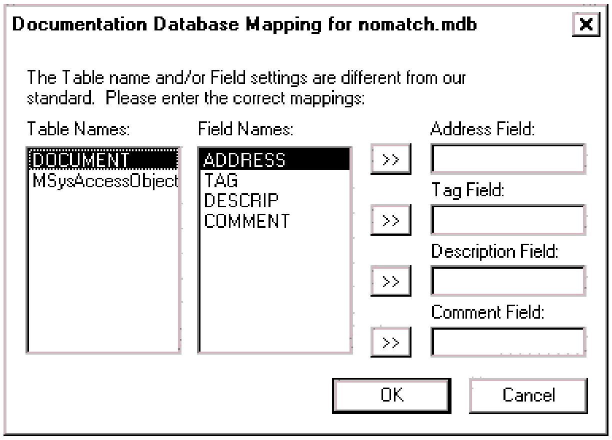


Figure 0-16

For the mapping to be successful, you must map to all four fields from one of the tables listed in the "Table Names" list box. The fields that are mapped must be of type "Text" for address, tag, and description and "Memo" for the comment field. The lengths of the fields must be less than or equal to FasTrak's lengths for all fields except the comment field. The max character length for "Address" is 64. The max character length for "Tag" is 32. The max character length for "Description" is 96. Failure to provide a valid mapping will prevent the device from being added.

Modifying Devices

To modify a device that is already configured select the "Modify Device" button from the "Device Source Configuration" dialog and the dialog below will be displayed.

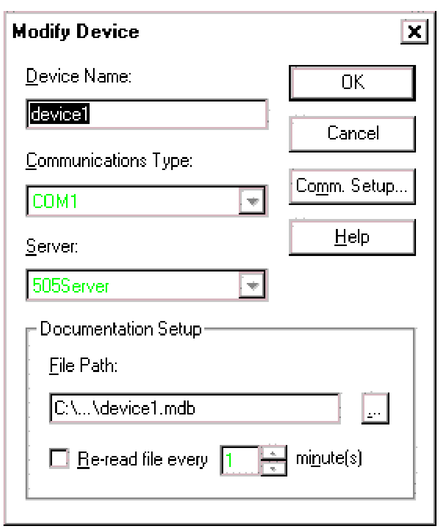


Figure 0-17

In this dialog only the device name and documentation setup info can be modified. If no trend windows are online with this device then the communication settings can also be modified. If the device is online then the communication settings can be viewed but cannot be modified by selecting the "Comm. Setup" button. Unlike the "Add Device" dialog, when we modify the device name here, the doc filename does not change.

Deleting devices

To delete a device select the "Delete Devices" button in the "Device Source" setup dialog to bring up the dialog below.

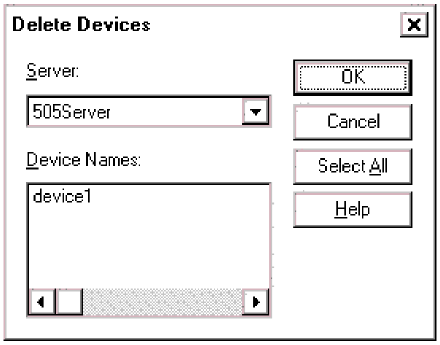


Figure 0-18

Clicking on "Select All" followed by "OK" will delete all the devices for a selected server. You can also click in the "Device Names" list box to manually select which devices to delete.

Trend Window Graph Setup

To setup the graph settings:

1. Select Graph from the Setup menu.
   Result: The Graph Settings tab dialog box appears (Figure 0-19).

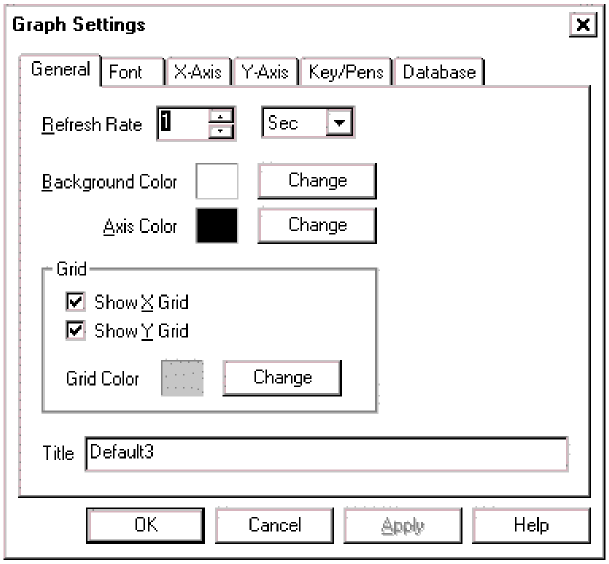


Figure 0-19

There are six pages on this property sheet.

General

Trend Times allow you to set when the trend window starts and stops trending data within the window. If "Immediately" is checked under the "Start Trending" group box, data will be read from the server and logged to the. Md. trend file as soon as data is entered into the trend window provided that the trend window is online and active. If "Specific time" is checked then data will be read from the server and logged after the time indicated just below the radio button.

If the first radio button under "Stop Trending" is checked, then any data in the trender window will be logged to the .mdb file as long as the trend window is open. Un-checking the "Online" menu or checking the "Pause Trending" menu under the main menu item "Mode" will override this and suspend trending. If "Specific time" is checked, then trending will terminate after the time indicated just below the radio button. If "After time duration" is checked, then trending will occur for the duration of the time selected starting from the time indicated under the "Start Trending" group box. If "After one trend " is checked, then trending is terminated after the first and only trend . Closing the trend window prior to the selected stop trending time will override the time and terminate all trending from the closed window. A message prior to termination will warn you of the consequences of shutting down this window.

The "Trend Days" group box allows a user to select which days to apply the start and stop times from above. You can apply the start time to the start day and the stop time to the stop day or by clicking on the "Select day intervals" check box, the start and stop trend times can be applied a number of times to a number of days. The combo box in the bottom left corner can be set to "for" (for consecutive days) or "every" (for incremental days).

When a trend time is configured, you can click on the "Add >>" button to save the trend time and display it in text form in the "Selected Trend Times" list box. Two examples of trend times in text format are listed below:

Start trending at 1:30 PM on 10/25/1999. Stop trending after 5 minutes.

Every 2 days - start trending at 1:30PM. Stop trending at 2:30PM.

You can enable and disable each trend time by checking and un-checking the box next to the trend description. To edit a trend time in the "Selected Trend Times" list box, simply highlight the text to be edited and click on the "Edit <<" button. The text in the list box will be removed and the "Trend Time Setup" group box will be updated with the settings corresponding to the text. You can modify the settings and re-enter them by clicking on the "Add >>" button at any time. 256 trend times may be saved as long as they don't overlap.

The static text box in the lower right corner of the property page describes the currently active control. This text will change whenever a different control is selected.

The FTTrender defaults to **trending** data immediately after addresses are entered into the trend window and we will continue until the trend window is closed.

Trend Window Setup (Trend Data)

("Trend Data") of the property Trend Window sheet is listed below.

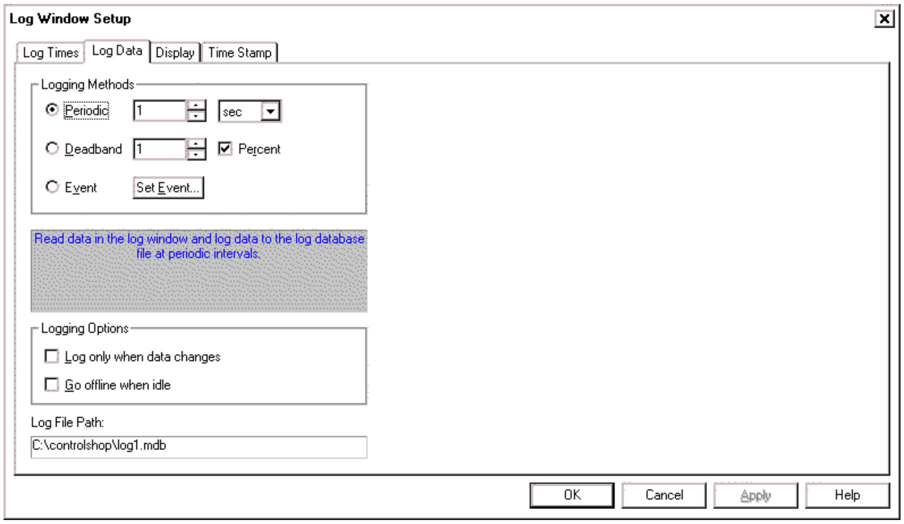


Figure 0-20

Within this dialog you can configure how the server reads the data and updates the cache which will dictate how data is logged.

Trending methods

The Periodic method will update the cache in time intervals specified in edit box. The combo box list four time units (msec, sec, min, and hour). Minimum value is 0 (updates as quickly as possible) and maximum value is 24 hours. Deadband is a data variation value or a percentage of a data variation if checked. Each data value from the addresses in the trend window will be compared with the deadband value. If a particular value fluctuates by the deadband amount (+ or -), the server will update the cache for that data item. Deadband will then be compared with the updated value for that particular item. Event will only cause an update if the event created in "Set Event" is true. The "Set Event" dialog is below.

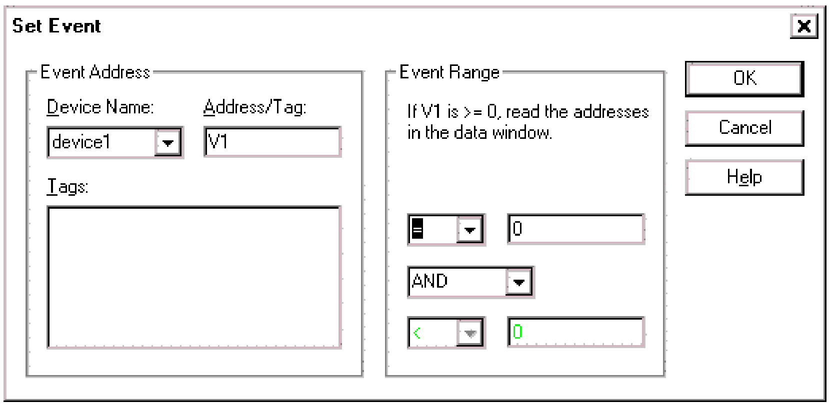


Figure 0-21

An event is set up by choosing an address or tag (referred to as the event address) from a device. Clicking on an item in the "Tags" list box will display the corresponding address in the "Address/Tag" edit box.

The "Event Range" represents an expression involving the event address. The addresses in a data window will be updated as long as the expression is true. The first combo box and adjacent edit box is referred to as the low trigger range. The combo box contains the following operators: <, >, <=, >=, =, and !=. The last combo box and adjacent edit box is referred to as the high trigger range. The combo box contains the following operators: <, >, <=, >=, =, and !=. The middle combo box represents the trigger relationship. This combo box contains the strings "END", "AND", "AND NOT", "OR", and "OR NOT". If "END" is selected, only the low trigger range is used in the expression. The text at the top of the edit box describes the complete expression with the event address included.

Trending method description box

The trending method static box will give a description of the trend method that is currently checked. This text changes whenever you select a different trend method.

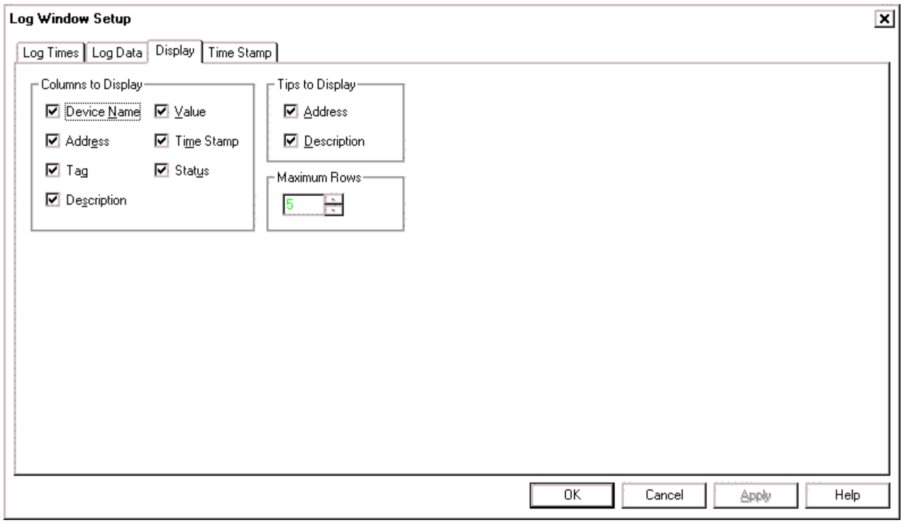
Trending Options

If the box labeled "Update only when data changes" is checked, then addresses in the trend window will be logged on an individual basis only if the address' value (including format) has changed or its status has changed. If this box is not checked then addresses will be logged every time they are read from the server.

If the box labeled "Go offline when idle for more than 5 minutes" is checked, then we will go offline with all the devices we are using in the corresponding trend window. When going offline with a device, the port will only be closed if no other trend windows or clients are using the ports for their own data.

Trend Window Setup (Display)

Trend Window Setup (Display) of the property sheet is listed below.



Columns to Display

"Columns to Display" simply turns on and off columns in the trend window (see Trend Window Appearance). All the columns are checked by default except for Value, Time Stamp and Status.

Tips to Display

As we scroll through the tag combo box in a trend window, the corresponding address and description (if it exists) of the highlighted tag will be displayed to the left and right respectively of the combo box. These are referred to as doc tips and can be turned on and off by checking and un-checking these boxes. All tips are checked by default.

Trend Window Setup (Time Stamp)

("Time Stamp") of the property sheet is listed below.

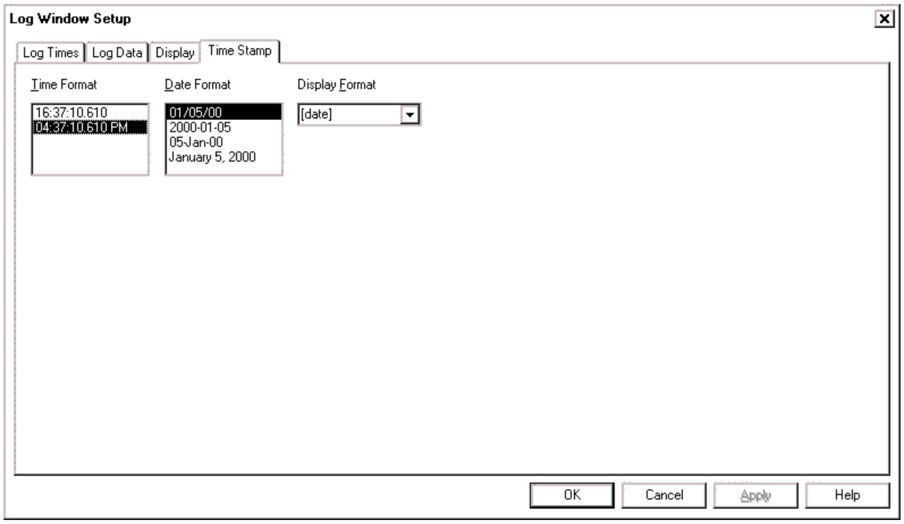


Figure 0-22

Time Format

The "Time Format" list box will display the current time (time at which the page was activated) in a standard format or military format. The default is standard format.

Date Format

The "Date Format" list box will display the current date (date at which you've activated the page) in four different formats.

[month]/[day]/[year] - each represented by 2 digits.

[year]-[month]-[day] - The year is displayed in 4 digits while the month and day are displayed in 2 digits.

[day]-[month]-[year] – The day and year are displayed in 2 digits while the month is displayed as a 3-character abbreviation.

[month] [day], [year] – The month is completely spelled out, the day is displayed in minimal digits and the year is displayed as 4 digits.

The default is format 1 (**[month]/[day]/[year] -** each represented by 2 digits).

Trend Window

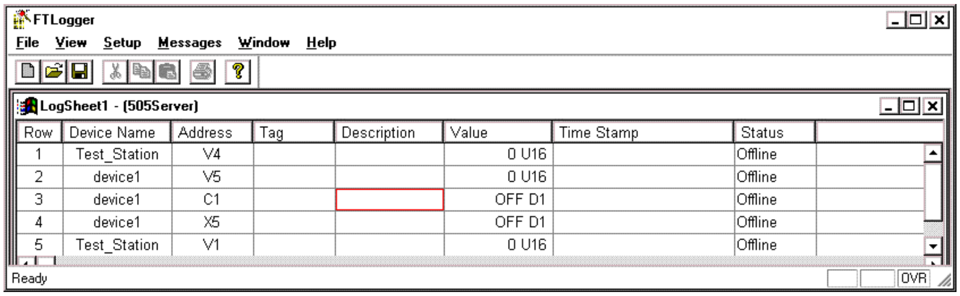


Figure 0-23

Column Layout

Each trend window will is in a spreadsheet form with customizable columns. Each column can be sized in one of two ways. You can click and hold the left mouse button when in between 2 columns. By moving the cursor to the right or left a user can increase or decrease the column on the left. You can also double click when in between 2 columns. This will adjust the left column so that it's large enough to display the largest string within this column. If no strings are displayed in the column, no adjustment will be made.

Row

Row indicator. The number of rows is configurable by you when creating a new trend window. Each row represents a single address from a single device that's configured in the server that the trend window is connected to. Currently, rows are not sizeable.

Device Name

Since addresses in the trend window can refer to any device connected to the trend window's server, a device name is necessary. When a key is hit in this column or clicked, an auto searching floating combo box control will be displayed containing all the device names for the trend window server. The control will highlight the current device when clicking in this column. Changing the device name in this column changes the current device on the status bar.

Address

When a key is hit in this column, an in-place edit box will be enabled. The address that is entered here will apply to the device in the "Device Name" column of the same row if one exists. If a device is not listed in the "Device Name" column, the address entered will apply to the device that is listed as the current device on the status bar. Entering an address will cause an update of all other columns. It is possible to enter a range of addresses by using the fill format equations below.

Address\+-[count][\increment][\format]

Address1-Address2[\increment][\format]

Tag

Devices must be added to edit this field. Tags will be stored in database files containing addresses and their corresponding descriptions. When a key is hit in this column or the user clicks in this column an auto searching floating combo box control will be displayed containing all the tags for the device indicated on the status bar (if an empty row) or entered in the "Device Name" column. A tag can be assigned to an address by first entering the address in the "Address" column. Next, activate the tag combo box and type in a unique tag. Selecting the "enter" or "tab" key or by clicking outside the combo box will enter the tag into the database and associate the new tag to the address listed in the same row. A tag can be successfully deleted by activating the tag combo box, removing all the text, then entering the info by one of the methods described above. Editing a tag will cause an update of all other columns.

If the "Re-read" box was checked in the "Documentation Setup" group box of the "Add Device" or "Modify Device" dialogs then this combo box would be periodically (depending on the update rate) flushed and reassigned.

Description

An address and/or tag must be present to edit this field. Descriptions will be stored in database files containing addresses and their corresponding tags. When a key is hit in this column or the user clicks in this column a vertically scrolling multi-line edit box will appear containing the current description if one exists. Approximately 6 rows can be viewed at one time. Unlike other "in-place" edit boxes within the trend window, this edit box will accept carriage returns. Selecting the "tab" key or clicking outside this edit box will enter the description into the database. A description can be successfully deleted by activating the edit box, removing all the text, then entering the info by one of the methods described above. Editing a description will cause an update of all other columns.

Value

Data Value with format indicator. This column enables you to view data values. This field is not editable. When the trend window is switched to "offline" mode or "online-pause" mode this field will display the last value read from the server. Addresses or tags that are entered while the trend window is offline or online-paused will all display a 0 value.

Time Stamp

Time that data value was read and put into the server cache. This column cannot be directly edited.

Status

Status of the data read from the device. Any errors reading addresses will be displayed here. Since many addresses are read in one packet, a status error is usually displayed in multiple rows. For example, assume we configured 200 "V" addresses in a 505 PLC and we were displaying them all in Fttrender. If a packet can consist of 256 addresses and we enter V201, a status error will be displayed on all 201 rows because the entire packet would have failed by including 1 erroneous address. Unlike SoftShop, ControlShop does not implement the type of error checking that would prevent a user from entering an invalid address so it is up to the PLC/device to report these types of errors. When offline, all the status rows will display the string "Offline". This field is not displayed by default.

(DATA) Menu Functionality

Format

Selecting a format will display and trend the value that the cursor is on (as well as any highlighted values) in that format.

Size

Allows you to select a 16-bit or 32-bit format size.

Next/Previous

Based on the vender address, get the next device register address.

Based on the vender address, get the Previous device register address.

Fill

This will fill the trend window with addresses that are set up in the dialog below.

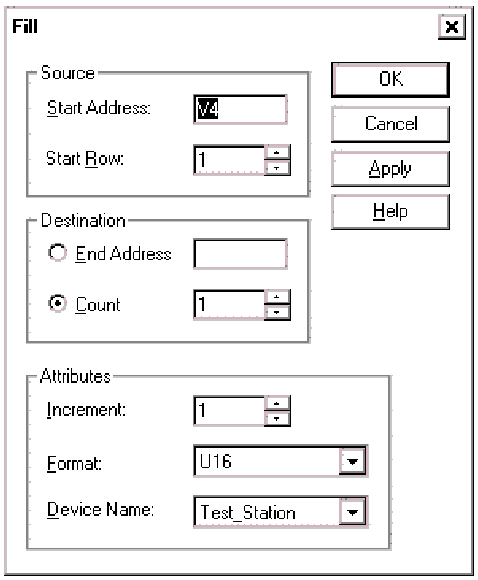


Figure 0-24

"Start Address" will hold the first address in a range of addresses. "Start Row" will hold the starting address location in the current trend window. Destination can be in the form of an end address or a count. For example, a start address of 400001 and an end address of 400003 will display 400001, 400002, and 400003 in the trend window. Selecting a count of 3 instead of an end address will give the same results. However, if increment is set to 2 and end address is selected addresses 400001 and 400003 will be displayed while addresses 400001, 400003, and 400005 will be displayed if count is selected. The format will determine how to display the data values for the addresses selected.

Clicking on the "Apply" button will fill the trend window with the addresses specified in the Fill dialog without closing the dialog. Pre-existing addresses in the trend window that fall within the fill range will be overwritten if "OVR" is enabled in the status bar. After clicking on Apply, the following fields will be modified:

Start Address – This will now indicate the address at the new start row.

Start Row – This will now indicate the first empty row after the last address in the fill.

Fill Example

Below is an example of how you might set up the fill dialog.

Start Address = 400001

Start Row = 5

End Address = 400003

Increment = 1

After "Apply" is selected and the data window is modified, the Fill dialog will have the following values:

Start Address= 400004

Start Row = 8

End Address= 400006

Increment = 1

Editing Features

Cut, Copy, and Paste

Cut, Copy, and Pastes can be performed on trend window rows a variety of different ways. The menu items can be found under the main "Edit" menu along with hot keys. Alternative hotkeys shift-delete, ctrl-insert, and shift-insert are supported for Cut, Copy, and Paste respectively. Toolbar items are also provided for all three features. Trend window rows can be cut from one trend window and pasted into another across applications provided that the trend window's servers match.

Cut, Copy, and Paste will observe highlighted rows similar to excel. Highlighting trend window rows is also similar to excel with the exception of column highlighting.

Select All

The "Select All" menu item under the main "Edit" menu can be used to unconditionally highlight all the rows in the active trend window. Features such as delete, format, size, on, off, cut, copy, and paste can be performed on the entire trend window.

Trend Status Window

The trend status window shown below lists successful and failed logs as well as any miscellaneous errors that may occur while attempting to trend data (dependant upon setup criteria) for all trend windows that are currently trending data. If no online trend windows are up, the message "No logs are currently active" will be displayed. A successful trend entry in this window will read "At [date and time of trend] data was successfully logged from [trend window name] to [Access trend filename]". A failed trend entry in this window will read "At [date and time of trend] data failed to trend . [error message describing reason of failure]". Each message type can be displayed in a different color (See section 5.2). Entries into this window may be stopped at any time by checking the "Pause Trending" menu under the main menu "Mode". Un-checking the "Pause Trending" menu will re-enable the window so that it may receive status messages once more. Status messages that occurred when the window was paused are lost.

A floating menu brought on by a right mouse button click while the cursor is over this window can be utilized to allow a user to clear or hide the window.

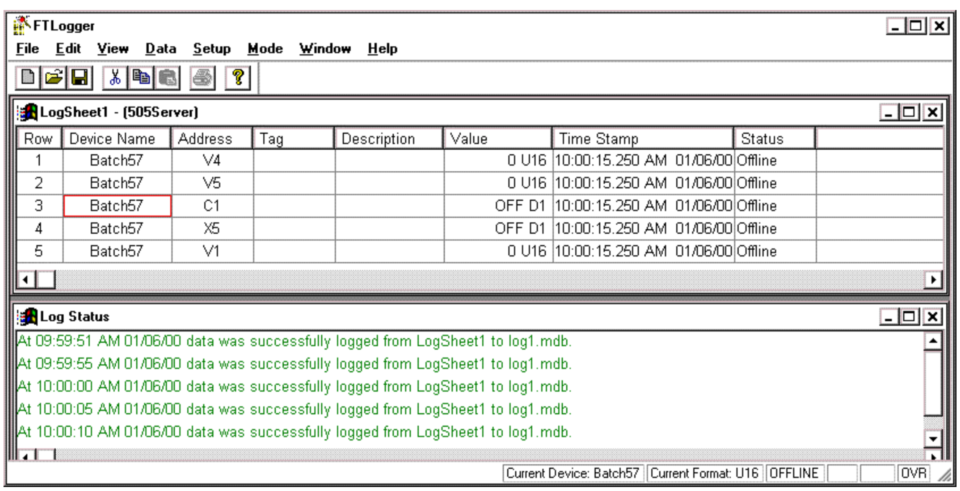


Figure 0-25

Trend Status Window Creation

This window will be created and hidden upon startup. To access this window you must select the "Trend Status" menu under the main menu "View". A check mark next to "Trend Status" will indicate that we are currently viewing this window. Clicking again on this menu will remove the check mark and hide the window. Hiding the window will remove all the entries. Entries will not be added to this window until is viewable or iconized. If this window is full, entries will be removed from the top down as new ones are appended to the bottom.

Trend Status Setup Dialog

The trend status setup dialog can be accessed by clicking on the "Trend Status" menu under the main menu "Setup". This dialog is shown below.

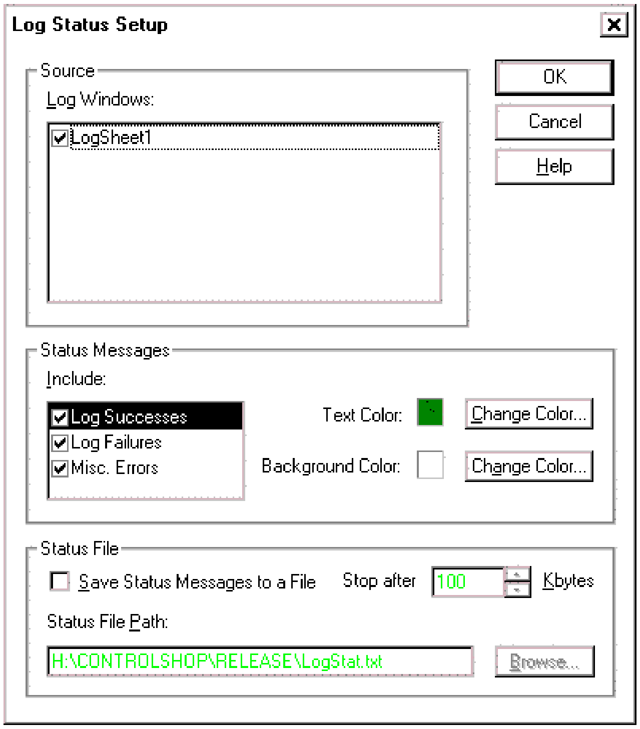


Figure 0-26

This dialog allows you to select which trend windows we will be receiving status messages from (Source group box) as well as which status messages we wish to receive (Include group box). These messages can also be written to a file by checking the "Save Status Messages to a File" box in the "Status File" group box and including a path and filename. You can click on the "Browse" button to bring up a common file search dialog as shown above. If you've selected an existing file and click on "OK" the software will ask you if you would like to append the new messages to this file or clear the file and begin from scratch. Writing status messages to a file vs. a window allows more flexibility in two ways. First, a file can hold more entries than a window. Secondly, status messages from one trend session can be appended to messages from a previous session.

Error Response Dialog

The Error Response dialog is accessed by clicking on the ""Error Response…" menu under the main menu "Setup". This dialog allows you to configure how the trender should react to a variety or error situations. This dialog is displayed below.

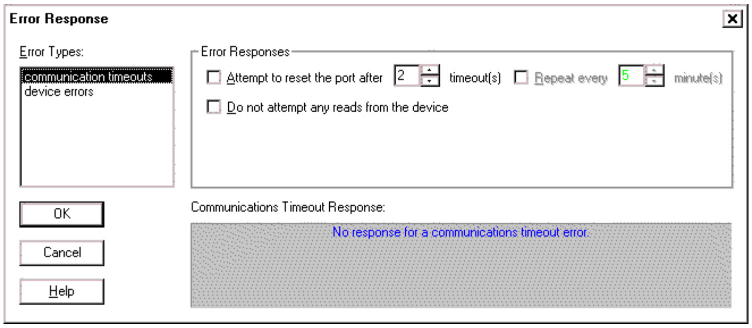


Figure 0-27

The "Error Types" list box will hold all the error situations in which a response can be pre-programmed. Error situations include communication timeouts, device errors, trend database file errors, and status file errors. The "Error Responses" group box will list all the possible responses for the item that's currently highlighted in the "Error Types" list box. If no items are checked in the group box, no action will be taken when the corresponding error situation arises. The text in bottom right corner will give a more detailed description of the error response for the error type that's highlighted.

Documentation

Once a device is configured you can specify a database file to hold and sort all documentation. Multiple users can simultaneously modify documentation for the same device so regular updates can be scheduled to get the latest documentation within the database. One documentation file is available for each device.

Documentation can be imported from comma or tab separated text files into an existing documentation database file by selecting File ® Import Documentation from the menu. The Import Documentation window appears (Figure 0-1).

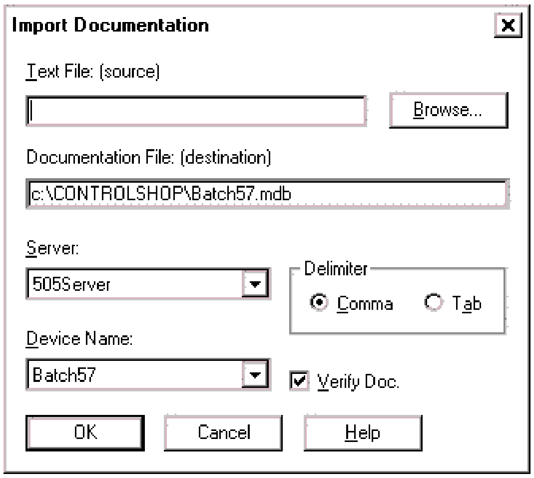


Figure 0-4

Documentation is merged into an existing database file attached to a single device. Note, when the radio button Verify Doc. is checked, tags, addresses, and the general file format are verified. This action causes the import process to be slightly slower. To start the import process, click the OK button.

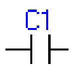
Documentation can also be exported by selecting File ® Import Documentation from the menu.

11 - RLL Instructions

Relay Instructions

Normal Contact (STR)

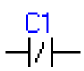
When the referenced address of a Normal Contact is ON, the contact is closed and passes power. When the referenced address is OFF, the Normal contact is open and does not pass power.



	Parameter Type	Valid Parameter Types
STR	Bit Address	X,Y,C,V,K,WX,WY,STW,B,W, TCP,TCC,DCC,DSP,DSC,DCP

Normally Closed Contact (STRN)

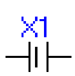
When the referenced address of a Normally Closed contact is ON, the Normally Closed contact is open and does not pass power. When the referenced address is OFF, the Normally Closed contact is closed and passes power.



	Parameter Type	Valid Parameter Types
STRN	Bit Address	X,Y,C,V,K,WX,WY,STW,B,W, TCP,TCC,DCC,DSP,DSC,DCP

Normally Open Immediate Contact

When the discrete point of an I/O module of a Normal Open Immediate Contact is ON, the contact is closed and passes power. When the referenced address is OFF, the Normally Open Immediate contact is open and does not pass power.

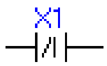


NOTE *Only the power flow for an immediate X contact is updated. The value in the image register table is not updated.*

	Parameter Type	Valid Parameter Types
LDI	Bit Address	X

Normally Closed Immediate Contact

When the discrete point of an I/O module of a Normally Closed Immediate contact is ON, the contact is open and does not pass power. When the referenced address is OFF, the Normally Closed Immediate contact is closed and passes power.



NOTE: *Only the power flow for an immediate X contact is updated. The value in the image register table is not updated.*

	Parameter Type	Valid Parameter Types
LDNI	Bit Address	X

Logical Not Contact (NOT)

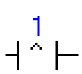
The NOT instruction inverts the power flow to the state opposite its current state. The NOT instruction does not have any parameters

Warning	**Do not program a NOT in parallel with any network that does not connect to the power rail.**

One Shot Contact (OS)

The One Shot instruction turns on an output for a single scan. The single parameter in the instruction contains the instruction reference number.

When the input transitions from OFF to ON, the output is turned on for exactly one scan. After the One Shot is executed, its input must be off for at least one scan before the instruction can be executed again. If the input is OFF, the instruction is not executed and there is no power flow at the output.



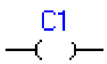
NOTE: *Do not assign the same reference number more than once for the One Shot instruction. You can use the same reference number in a One Shot in another instruction in the One Shot group because they use different bits of one byte to store the previous state.*

	Parameter Type	Valid Parameter Types
OS	Constant	A valid reference number, 1-1025

Coil

The Coil is an output instruction used to represent a field device or internal memory location that needs to be controlled. Use a normal coil when your application requires the referenced address to equal ON (1) when the coil has power flow.

When the network logic passed power to the coil, the coil turns on and the address equals 1. When the network does not pass power to the coil, the coil remains OFF (0) and the address equals 0.

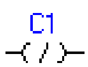


	Parameter Type	Valid Parameter Types
Coil	Bit Address	Form C

Coil Not

The NOT coil is used similarly like the normal coil, but if the referenced address equals OFF (0), the coil has power flow.

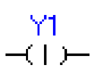
When the network logic does not pass power to the NOT coil, the coil remains energized and the reference address equals ON (1). When the network logic passes power flow to the NOT coil, the coil is de-energized and the referenced address equals OFF (0).



	Parameter Type	Valid Parameter Types
Coil Not	Bit Address	Form C

Immediate Coil

The Immediate coil is used similarly like the normal coil, but the immediate I/O module update is done when the coil is executed. The immediate coil is updated any time during the controller scan and is not limited to the normal I/O update portion of the timeline.

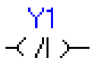


NOTE: *Both the image register and the I/O module are updated when the immediate coil is executed.*

	Parameter Type	Valid Parameter Types
Immediate Coil	Bit address	Y

Immediate Closed Coil

The Immediate coil is used similarly like the NOT coil, but the immediate I/O module update is done when the coil is executed. The immediate coil is updated any time during the controller scan and is not limited to the normal I/O update portion of the timeline.



NOTE: *Both the image register and the I/O module are updated when the immediate coil is executed.*

	Parameter Type	Valid Parameter Types
Immediate Closed Coil	Bit address	Y

Set Coil (SET)

The SET Coil is used to set a specified bit to ON (1) when the network passes power flow. If the network does not have power, the bit remains unchanged.

	Parameter Type	Valid Parameter Types
SET	Bit Address	Form C

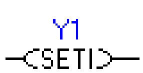
Reset Coil (RST)

The RST Coil is used to set a specified bit to OFF (0) when the network passes power flow. If the network does not have power, the bit remains unchanged.

	Parameter Type	Valid Parameter Types
RST	Bit Address	Form C

Set Immediate Coil (SETI)

The SETI Immediate Coil is used to set a specified bit to ON (1) when the network passes power flow, and the bit is updated immediately. If the network does not have power, the bit remains unchanged.



	Parameter Type	Valid Parameter Types
SETI	Bit Address	Y

Reset Immediate Coil (RSTI)

The RSTI Immediate Coil is used to set a specified bit to OFF (0) when the network passes power flow, and the bit is updated immediately. If the network does not have power, the bit remains unchanged.

```
  Y1
-<RSTI>-
```

	Parameter Type	Valid Parameter Types
RSTI	Bit Address	Y

Timers, Counters and Drums

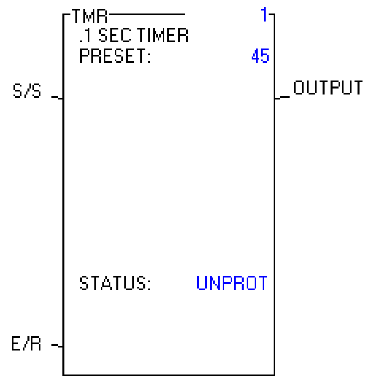
.1 s Timer (TMR)

The Timer instruction (TMR) is used to time events. The timer output is turned on after the timer has timed down, making this an 'on delay' timer. A slow timer is denoted by TMR, a fast timer by TMRF.

The timer times down from the preset value specified by Preset. The Preset value is stored in the TCP memory. The timers current time is stored in the TCC memory area. Timers have the following properties:

- The Enable/Reset must be on for the timer to operate.
- When the Start/Stop input is on and the Enable/Reset is on, the timer begins to time down.
- Timing begins at the preset value Preset and continues down to zero.
- If the Start/Stop input turns off and the Enable/Reset input remains on, the timer stops but it saves the current value, TCC. If the Start Stop input turns on again, the timer resumes timing. TCC is also saved if the Enable/Reset input is on and a loss of power occurs, provided the controller battery backup is enabled.
- If the Enable/Reset input is turned off, the timer is reset to the preset time specified in Preset.
- The output is turned on when the timer reached zero, and it stays on until the timer is reset (the Enable/Reset input is turned off).

Timer On-Delay



TMR	Parameter Type	Valid Parameter Types
Timer number:Preset:	Constant OnlyConstant Only	Varies with configured T/C ranges0000.0 to 3276.7

Warning	**Do not use the same reference number more than once for timer, counter, up/down counter, and discrete/motor control alarm timer instructions. Using the same reference number can cause unpredictable operations.**

.001 s Timer (TMRF)

The Timer instruction (TMR) is used to time events. The timer output is turned on after the timer has timed down, making this an 'on delay' timer. A slow timer is denoted by TMR, a fast timer by TMRF.

The timer times down from the preset value specified by Preset. The Preset value is stored in the TCP memory. The timers current time is stored in the TCC memory area. Timers have the following properties:

- The Enable/Reset must be on for the timer to operate.
- When the Start/Stop input is on and the Enable/Reset is on, the timer begins to time down.
- Timing begins at the preset value Preset and continues down to zero.
- If the Start/Stop input turns off and the Enable/Reset input remains on, the timer stops but it saves the current value, TCC. If the Start Stop input turns on again, the timer resumes timing. TCC is also saved if the Enable/Reset input is on and a loss of power occurs, provided the controller battery backup is enabled.
- If the Enable/Reset input is turned off, the timer is reset to the preset time specified in Preset.
- The output is turned on when the timer reached zero, and it stays on until the timer is reset (the Enable/Reset input is turned off).

Retentive Timer On-Delay

/7 TMR	**Parameter Type**	**Valid Parameter Types**
Timer number:	Constant Only	Varies with configured T/C ranges
Preset:	Constant Only	0000.0 to 32.767

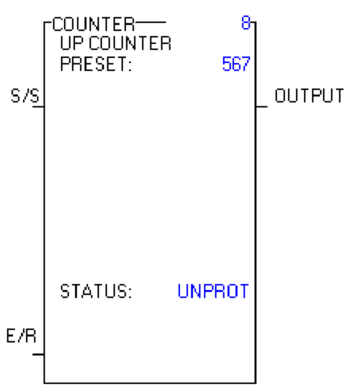
Warning	**Do not use the same reference number more than once for timer, counter, up/down counter, and discrete/motor control alarm timer instructions. Using the same reference number can cause unpredictable operations.**

Up Counter (CTR)

The Counter instruction (CTR), an up counter, counts recurring events. The counter output is turned on after the counter has counted up to a preset value.

The counter counts up to the preset value specified in Preset, which is stored in the TCP-memory. The current count is stored in TCC memory. The counter has these properties:

- The Enable/Reset must be on for the counter to operate.
- When the Enable/Reset is ON, the counter is incremented by one each time the Counter input transitions from off to on.
- Counting begins at zero (0) and continues to the preset value specified by Preset.
- If the Enable/Reset is turned off, the count is reset to zero.
- TCC is saved if the Enable/Reset input is on and loss of power occurs, provided the controller battery backup is enabled.
- The output is turned on when the current count equals the preset count specified by Preset.
- If the Enable/Reset does not receive poser flow, the instruction is not executed and the output does not turn on.



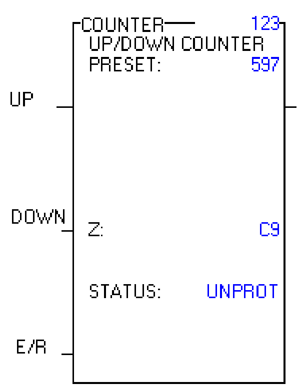
CTR	**Parameter Type**	**Valid Parameter Types**
Counter Number:	Constant	Depends on configured quantities.(Max. 128)
Preset:	Constant	Counts to this value, 0-32767

Up/Down Counter (UDC)

The Up/Down Counter (UDC) instruction counts the number of events (up or down) from 0 32767.

When the counter counts up, it counts to the preset value specified in Preset, which is stored in TCP memory. The current count is stored in TCC memory. The UDC instruction has the following properties:

- The Enable/Reset must be on for the counter to operate.
- When the Enable/Reset is on, the counter is incremented by one when the Up input transitions from off to on.
- When the Enable/Reset is on, the counter is decremented by one when the Down input transitions from off to on. The UDC cannot be decremented to a number less than zero.
- TCC does not change if the Up and Down inputs both change from off to on during the same scan.
- If the Enable/Reset turns off, TCC is reset to zero.
- The output specified in Z is turned on whenever TCC equals zero. This output is turned off when TCC does not equal zero.
- The box output is turned on whenever TCC equals zero or TCP.
- After having counted to the preset value (TCP), the box does not require resetting in order to resume counting in the opposite direction. TCC does not ever exceed TCP.
- You can protect the preset values from being changed in the Data Window by selection ????? NOTE: If you use an operator interface to change the TCP values, the new TCP values are not changed in the original RLL program until the entire program is downloaded to the PLC.



UDC	Parameter Type	Valid Parameter Types
Counter Number:	Constant Only	Varies with T/C configured range
Preset:	Constant Only	0-32767
Z	Bit Addr	Y,C,B

Warning	**Do not use the same reference number more than once for timer, counter, up/down counter, and discrete/motor control alarm timer instructions. Using the same reference number can cause unpredictable operations.**

Discrete Control Alarm Timer (DCAT)

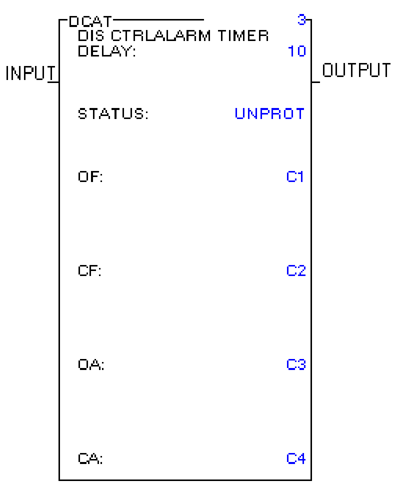
The Discrete Control Alarm Timer (DCAT) instruction is used with a single input, double feedback device. The input to the DCAT instruction should be derived from the preceding logic that determines the state of the device. The output of the instruction should control the device.

The DCAT timer times down from the preset value (Delay) which is stored in TCP memory. The timer s current time is stored in TCC memory.

When the Open (OA)/Close (CA) input to the DCAT goes from OFF to ON, the followings operations occur:

- The time delay is set to the preset value defined by Delay, both outputs OA and CA are turned off and the DCAT output turns on.
- While the Open/Close input to the DCAT remains ON, the timer begins timing until the OF input turns ON or the timer times out.
- If the OF input turns on before the timer times out, the time delay is set to zero and the OA remains off.
- If OF does not turn on before the timer times down, OA is turned on.
- IF OF turns on before the timer times down, but then goes off again while the Open/Close input is ON, OA is turned on. The OA is turned off if OF then turns on again.
- When the Open (OA)/Close (CA) input to the DCAT goes from ON to OFF, the followings operations occur:
- The DCAT output turns off, the time delay is set to the preset value defined by Delay, and both alarm outputs OA and CA are turned off.
- While the Open/Close input to the DCAT remains off, the timer begins timing until the CF input turns on or the timer times out.
- IF the CF input turns on before the timer times down, the time delay is set to zero and the CA remains off.
- If CF does not turn on before the timer times down, CA is turned on.
- IF CF turns on before the timer times down, but then goes off again while the DCAT input is off, CA is turned on. The CA is turned off if CF turns ON again.

NOTE: *If both OF and CF are simultaneously ON, the OA and CA turn on.*



DCAT	Parameter Type	Valid Parameter Types
Alarm Number:	Constant Only	Varies with T/C configured range
Delay:	Constant Only	0000.1 - 3276.7
OF:	Bit Addr	X,Y,C,B
CF:	Bit Addr	X,Y,C,B
OA:	Bit Address	Y,C,B
CA:	Bit Address	Y,C,B

Use the following table for state changes.

Input Condition	**IF**			**THEN**		
0 = Open 1 = close X = don't care	Feedback **AND** Timer Action			Alarm Status		Output
	OF	CF		OA	CA	
1	0	1	timing	0	0	1
1	0	0	timing	0	0	1
1	1	0	Reset[2]	0	0	1
1	0	0	timed out[1]	1	0	1
0	1	0	timing	0	0	0
0	0	0	timing	0	0	0
0	0	1	reset	0	0	0
0	0	0	timed out	0	1	0
X	1	1	X	1	1	Follows input

1 Timed Out: Timer has timed a full preset value of time without a sensor closing.

2 Reset: Timer is at preset value and not timing.

Warning	**Unexpected alarm conditions may occur when the DCAT exists within the zone of control of a JMP or MCR. The DCAT output and alarms are under the control of the JMP or MCR.**

Motor Control Alarm Timer (MCAT)

The Motor Control Alarm Timer (MCAT) instruction is designed for use with a double input, double feedback device. The MCAT operates similarly to the DCAT instruction, but the MCAT provides the ability to operate motor-driven devices that drive in opposite direction. You can use the MCAT to replace several networks of logic that are required to time the field device's operation and generate alarms in case of failure.

The MCAT timer times down from the preset value specified in Timer, which is stored in TCP memory. The time current time is stored in TCC memory.

When the Open input transitions from off to on, and the close and Stop inputs are both off, the OO turns on and the timer starts. Once triggered, OO remains on independent of the open input until one of the following event occur:

- The timer times to 0. The OA is turned on and the OO is turned off.
- The OF turns on while the CF remains off. The OO is turned off and the timer resets to 0. If OF turns on and then turns off, the OA comes on immediately (no time delay) the next timer the box is executed.
- The Stop input turns on. The OO, CO, OA, and CA are turned off, and the timer stays where is was when STOP was received. If the Stop inputs turns off while the Open was when Stop was received. If the Stop input turns off while the Open input is on, then the timer starts at the preset value again - not at the value when the Stop input turned on.
- The Close input turns on after the Open turns off. The CO is turned on and the timer starts counting at the preset. The OO is turned off.

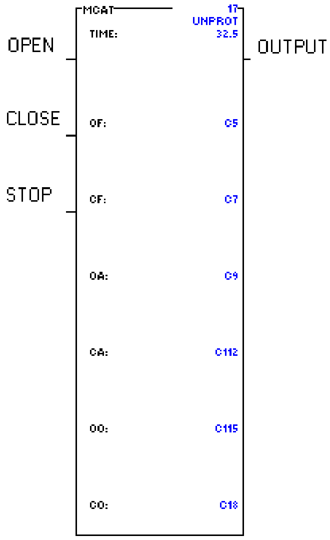
  When the Close input transitions from off to on, while the Open command and Stop command inputs are both off, the CO turns on and the timer starts. CO should turn on the motor that closes the valve. Once triggered, the CO remains on independent of the Close input until one of the following events occurs:

- The timer times to 0. The CA is turned on and the CO is turned off.
- The CF turns on while the OF remains off. The CO is turned off and the timer is reset. If \CF turns on and then turns off, the CA comes on immediately the next timer the box is executed.
- The Stop input turns on. The OO, CO, OA, and CA are turned off.
- The Open input turns on after the Close input turns off. The OO is turned on. The CO is turned off.

The condition in which both the Close and Open inputs are on simultaneously is treated as a Stop. The input remaining on when the other turns off is seen as a transition from off to on, and the MCAT enters the appropriate state.

When the Stop input overlaps an Open or Close input, the Stop overrides as long as it is on. When the Stop turns off, the remaining input is seen as a transition from off to on and drives the MCAT to the corresponding state.

The condition in which both Feedback signals are on simultaneously is an error condition. Both open and Close are turned on and both Open and Closed Outputs are turned off. Removing the conflicting feedback signals does not clear the Open and Close Alarms. One of the MCAT inputs (Open, Close or Step) must change state in order to clear the error state.



MCAT	Parameter Type	Valid Parameter Types
Alarm Number:	Constant Only	Varies with T/C configured ranges
Time:	Constant Only	time, 0000.1 – 3276.7
OF:	Bit Address	open feedback, X,Y,C,B
CF:	Bit Address	closed feedback, X,Y,C,B
AO:	Bit Address	open alarm, Y,C,B
CA:	Bit Address	closed alarm, Y,C,B
OO:	Bit Address	open output, Y,C,B
CO:	Bit Address	close output, Y,C,B

Warning	**Do not use the same reference number more than once for timer, counter, up/down counter, and discrete/motor control alarm timer instructions. Using the same reference number can cause unpredictable operations.**

Time Driven Drum (DRUM)

The Drum Instruction (DRUM) simulates an electromechanical stepper switch or drum. It provides 15 output coils and 26 steps which are operated on multiples of the time base setup for the drum. Each step controls all 15 output coils.

When the drum begins to run, it starts at the step specified by the Drum Step Preset, which is stored in DSP memory. The drum current step is stored DSC memory. The counts per step, set in the Count/Step field, is stored in L-memory and cannot be changed without reprogramming the DRUM. The current count (counts remaining for a step) is stored in DCC memory. The drum has these features:

- The drum is enabled when the Enable/Reset input is ON.
- When the Enable/Reset is on and the Start input turns on, the drum begins to run. The drum begins at the step specified by DSP and remains at this step until DCC counts down to zero.
- When DCC for a step reaches zero, the drum advances to the next step, and the coils are turned on/off according to the drum mask for the new step. Each 1 in the mask designates that a coil is to be turned on, while each 0 designates that a coil is to be turned off.
- When the Enable/Reset turns off, the drum output is turned off and the drum returns to the step specified in the DSP.
- IF the Start input is turned off but the Enable/Reset remains on, the rum remains at the current step and DCC holds its current count. All coils maintain the condition specified by the drum mask for this step.
- When the drum is at the Preset step, the output coils follow the states specified by the drum mask for that step, even if the Enable/Reset input is off. Take care to program the mask with a bit pattern that is a safe state for the Preset step.

DRUM	**Parameter Type**	**Valid Parameter Types**
Drum Number:	Constant	Varies with drum memory
Preset:	Constant	1-16
Sec:	Constant	0.0 - 32.767

Warning	**Do not use the same reference number more than once for timer, counter, up/down counter, and discrete/motor control alarm timer instructions. Using the same reference number can cause unpredictable operations.**

Time/Event Driven Drum (EDRUM)

The Time/Event Driven Drum (EDRUM) instruction simulates an electromechanical stepper switch or drum. The EDRUM can be indexed by a timer only, an event contact only or a time and event. A job input enables you to allow either timer or an event to advance the drum a step. The EDRUM provides 15 coils and 16 steps which are operated on multiples of the drum time base. Each step controls all 15 output coils.

When the drum begins to run, it starts at the step specified by the Drum Step Preset, which is stored in DSP memory. The drum current step is stored in DSC memory. The counts per step, set in the Count/Step field, is store in DCP memory. The drum current count is stored in DCC memory.

- The drum in enabled when the Enable/Reset input is on.
- When the Enable/Reset is on and the Start input turns on, the drum begins to run. The drum begins at the step specified by DSP and advances to the next step depending upon operation of the timer and/or event.
- When the drum advances a step, coils are turned on or off according to the mask for the new step. Each 1 in the mask designates that a coils I to be turned on, while each 0 designates that a coils is to be turned off.
- The drum output turns on, and remains on, after the last programmed step has been executed. The last programmed step is the last step having an event programmed or having a non-zero Count/Step preset value. The event must be on and the DCC must be zero. If the event turns off after DCC reaches zero, the drum output remains on and the EDRUM remains at the last programmed step until the drum is reset.
- When the Enable/Reset turns off, the drum output is turned off and the drum returns to the step specified in DSP.
- If the Start input turns off and Enable/Reset remains on, the drum remains at the current step (DSC) and DCC holds its current count. All coils maintain the condition specified by the drum mask.
- When the drum is at the preset step, the output coils follow the states specified by the drum mask for that step, even if the Enable/Reset input is off. ake care to program the mask with a bit pattern that is a safe step for the Preset step.
- The drum advances to the next step immediately if the Jog input transitions from off to on and the Enable/Reset input is also on.

EDRUM Parameter	**Type**	**Valid Parameter Types**
#	Varies with configured memory	Instruction reference number. Refer to controller user manual for number supported. The assigned instruction number must conform to the requirements of drum memory discussed on page 4–7 in Section 4.2.
PRESET	1-16	Step to which the drum returns when reset.
SEC/CNT	0-32.767	Time base. Amount of time in seconds for one count.
EVENT	X, Y, C, B	Discrete point that starts countdown of a step and that advances the drum to the next step when count equals zero.
Coils	Y, C, B, or blank	Coils controlled by drum. C0 represents no coil.
STP	1-16	Step number.
CNT	0-32767	Specifies time that drum remains at step. Actual time/step equals CNT × SEC/CNT in seconds.
Mask	0-1	Mask controls coils turned on (1) or off (0).

Warning	**Do not use the same reference number more than once for timer, counter, up/down counter, and discrete/motor control alarm timer instructions. Using the same reference number can cause unpredictable operations.**

Miscible Event Drum, Discrete (MADRID)

The Discrete Miscible Event Drum (MADRID) instruction operates similarly to the event drum but is capable of specifying a configurable mask for each step, which allows selection of the coils to be under the control of the fixed mask in each MADRID step.

When the Drum begins to run, it starts at the step specified by the Drum Step Preset, which is stored in DST memory. The current step is stored in D.Sc. memory. The counts per step, set in the CNN field, is stored in DIP memory. The current count is stored in BCC memory. The MADRID operates as follows:

- The drum is enabled when the Enable/Reset input is on.
- When the Enable/Reset is on and the Start input turns on, the drum beings to run. The drum begins at the step specified by DSP and advances to the next step based on operations of the timer and/or event.
- When the drum advances a step, coils are turned on/off according to the fixed mask and the current bit pattern in the configurable mask.
- The drum output comes on, and remains on, after the last programmed step has been executed. The last programmed step is the last step having an event programmed or having a non-zero CNT preset value. The event must be on and DCC must be zero. If the event goes off after DCC reaches \zero the drum output remains on and the MDRMD remains at the last programmed step until the drum is reset.
- When the Enable/Reset turns off, the drum output is turned off and the drum returns to the step specified in DSP.
- If the start input turns off and Enable/Reset remains on, the drum remains at the current step(DSC) and DCC holds its current count. All coils specified in the configurable mask maintain the condition specified by the fixed mask.
- When the drum is at the preset step, the coils specified in the configurable mask follow the states specified by the fixed mask for that step, even if the Enable/Reset input is off. Be sure to program the mask with a bit pattern that is a safe state for the preset step.
- The drum advances to the next step immediately if the Jog input transitions from off to on and the Enable/Reset input is also on.

You can use the MDRMD in applications that require a configurable on/off pattern for the drum coils. To do this, specify all 1s for the fixed mask for every programmed step of the MDRMD and precede the MDRMD and MDRMDs coils. The configurable mask table in memory must then contain the on/off patterns that are to be written to the coils for each step.

MDRMD	Parameter Type	Valid Parameter Types
Drum Number:	Constant Only	Varies with Drum configured ranges
Preset:	Constant Only	1-16
Sec:	Constant Only	00.001-32.676
Mask:	Word Address	V,W,G

Warning	**Do not use the same reference number more than once for timer, counter, up/down counter, and discrete/motor control alarm timer instructions. Using the same reference number can cause unpredictable operations.**

Maskable Event Drum Drum, Word (MDRMW)

The Maskable Event Drum, Word (MDRMW) instruction operates similarly to the event drum, but the MDRMW writes the date to a word instead of to individual coils. The MDRMW also is capable of specifying a configurable mask for each step. The allows the selection of the bits in the work to be changed by the fixed mask in each MDRMW step.

When the drum begins to run, it starts at the step specified by the Drum Step Preset, which is stored in DSP memory. The current step is stored in DSC memory. The counts per step, set in the CNT field, is stored in DCP memory. The current count is stored in DCC memory. The MDRMW has the following operations:

- The drum is enabled when the Enable/Reset input is on.
- When the Enable/Reset is on and the Start input turns of, the drum begins to run. The drum begins at the step specified by DSO and advances to the next step based on operation of the timer and/or event.
- When the drum advances a step, individual bits of the output word are turned on/off based on the fixed mask and the current bit pattern in the configurable mask.
- The drum output comes on, and remains on, after the last programmed step has been executed. The last programmed step is the last step having an event programmed or having a non-zero CNT preset value. The event must be on and DCC must be zero. If the event goes off after DCC reaches zero, the drum output remains on and the MDRMW remains at the last programmed step until the drum is reset.
- When the Enable/Reset turns off, the drum output is turned off and the drum returns to the step specified In DSP.
- If the Start input turns off and the Enable/Reset remains on, the drum remains at the current step and DCC holds its current count. All bits specified in the configurable mask maintain the condition specified by the fixed mask.
- When the drum is at the Preset step, the bits specified in the configurable mask follows the states specified by the fixed mask for that step, even if the Enable/Reset input is off. You should program the mask with a bit pattern that is a safe state for the Preset step.
- The drum advances to the next step immediately if the Jog input transitions from off to on and the Enable/Reset input is also on.

The configurable mask is specified for each step by a memory location in the mask field of the instruction. The configurable mask is located in 16 consecutive memory locations. The first location corresponds to step 1 of the drum, the second to step 2, etc. The mask is defined as being configurable because you can change the mask by writing data to the memory locations.

MDRMW	Parameter Type	Valid Parameter Types
Drum Number:	Constant Only	Varies with Drum configured ranges
Preset:	Constant Only	1-16
Sec:	Constant Only	00.001-32.676
Mask:	Word Address	V,W,G

Warning	**Do not use the same reference number more than once for timer, counter, up/down counter, and discrete/motor control alarm timer instructions. Using the same reference number can cause unpredictable operations.**

Compare Instructions

Equal To (EQU)

EQU	Parameter Type	Valid Parameter Values
Type		INT or UINT
A B	WX, WY, V, K, TCP, TCC, W, STW, DSP, DCP, DSC, DCC.	-32768 to 32767

Not Equal To (NEQ)

EQU	Parameter Type	Valid Parameter Values
Type		INT or UINT
A B	WX, WY, V, K, TCP, TCC, W, STW, DSP, DCP, DSC, DCC.	-32768 to 32767

Less Than (LESS)

EQU	Parameter Type	Valid Parameter Values
Type		INT or UINT
A B	WX, WY, V, K, TCP, TCC, W, STW, DSP, DCP, DSC, DCC.	-32768 to 32767

Less Than or Equal To (LEQ)

EQU	Parameter Type	Valid Parameter Values
Type		INT or UINT
A B	WX, WY, V, K, TCP, TCC, W, STW, DSP, DCP, DSC, DCC.	-32768 to 32767

Greater Than (GTR)

EQU	Parameter Type	Valid Parameter Values
Type		INT or UINT
A B	WX, WY, V, K, TCP, TCC, W, STW, DSP, DCP, DSC, DCC.	-32768 to 32767

Greater Than or Equal To (GEQ)

EQU	Parameter Type	Valid Parameter Values
Type		INT or UINT
A B	WX, WY, V, K, TCP, TCC, W, STW, DSP, DCP, DSC, DCC.	-32768 to 32767

Compare (CMP)

The Compare instruction (CMP) compares a signed integer value in A with a signed integer value in B. Based upon the comparison, the coil or relay addresses may be turned on or off.

If the input is ON, then the compare instruction will be executed on every scan. The CMP instruction works as follows:

- The value in A is compared against the value in B. A and B are not affected.
- If A < B. LT is turned on (1), GT is turned off (0), there is no power flow.
- If A > B, GT is turned on, LT is turned off and there is no power flow.
- If A = B, GT and LT are turned off and the output is turned on.
- If the input is off, the GT and LT coils are turned off.

CMP	Parameter Type	Valid Parameter Types
Reference Number	Constant	0-32767
Parameter A:	Word Address	Form A
Parameter B:	Word Address	Form A
Parameter LT:	Bit Address	Y, C, B
Parameter GT:	Bit Address	Y, C, B

Indexed Matrix Compare (IMC)

The Indexed Matrix Compare (IMC) instruction compares a predefined 15-bit mask pattern to the status of up to 15 discrete points. The mask to be compared is selected from a field of up to 16 masks by the step number currently located in Cur Ptr. If a match is found, the output is turned on.

The IMC instruction is described as follows:

- The Enable input must be on for the instruction to be executed.
- When Enable is ON and the Start input is turned on, the instruction is executed.
- The current status of up to 15 X, Y and C points is checked against the predefined bit patter identified by the step number loaded into Cur Ptr.
- If a match is found, the box output is turned on.
- If no match is found and the Start input remains on, the IMC checks the step selected by the Cur Ptr on every scan.
- If the Cur Ptr value is out of range, the controller automatically writes 16 to the Cur Ptr address. This means that mask 16 is used anytime the Cur Ptr is out of range.

IMC	Parameter Type	Valid Parameter Types
Reference Number:	Constant	0-32767
CUR PTR:	Word Address	V, W, G

Scan Matrix Compare (SMC)

The Scan Matrix Compare Instruction (SMC) compares up to 16 predefined bit patterns to the current states of up to 15 discrete points. If a match is found, the step number that contains the matching bit pattern is entered into the memory location specified by the pointer, and the output is turned on.

The SMC has the following properties:

- The instruction is executed when the Start input is on.
- If the Start input remains on, the SMC instruction checks all programmed steps on every scan.
- The status of up to 15 discrete points is checked against the predefined bit patterns.
- If a match is found, the step number of the matching mask is entered into the memory location specified by Cur Ptr, and the output is turned on.
- If no match is found, Cur Ptr is cleared to 0 and the output is turned off..

SMC	Parameter Type	Valid Parameter Types
Last	Constant	1-16
Cur:	Word Addr	V,W,G

Bit Operations

Bit Clear (BITC)

The Bit Clear instruction clears a specified bit to zero. When the input is on, the BITC instruction is executed. The output is turned on during each scan in which the instruction is executed. The operation is as follows: Bit N of element A is cleared to 0 is power is passed

BITC	Parameter Type	Valid Parameter Types
Reference Number	Constant	0-32767
Parameter A:	Word Address	Form B
Parameter N:	Constant	Bit Number, 1-16
		(most significant bit is 1,
		least significant bit is 16)

Bit Pick (BITP)

The Bit Pick Instruction (BITP) examines the status of a specified bit. When the input is turned on, the BITP instruction is executed. The status of bit N of input element A is checked as follows:

- The output is turned on if the selected bit is 1.
- The output is turned off if the selected bit is 0.

BITP	Parameter Type	Valid Parameter Types
Reference Number	Constant	0-32767
Parameter A:	Word Address	Form B
Parameter N:	Constant	Bit Number, 1-16
		(most significant bit is 1,
		least significant bit is 16)

Bit Set (BITS)

The Bit Set Instruction (BITS) sets a specified bit to one. When the input is on, the BITS instruction is executed. The operations executed is as follows: Bit N of word A is set to 1 if power is passed.

BITS	Parameter Type	Valid Parameter Types
Reference Number	Constant	0-32767
Parameter A:	Word Address	Form B
Parameter N:	Constant	Bit Number, 1-16
		(most significant bit is 1,
		least significant bit is 16)

Bit Shift Register (SHRB)

The Bit Shift Register (SHRB) instruction creates a bit shift register using a specified number of control relays or points in the discrete image register. The shift register may be up to 1023 bits long.

The SHRB instruction has the following features:

- When the Enable/Reset is turned on, the SHRB instruction is enabled.
- When the clock transitions from zero to one, the following actions occur. The last (highest numbered) bit of the shift register is moved to the output. The data in the shift register is shifted one address. The status of the Data input (0 or 1) is moved into the lower numbered point as specified in the IR field.
- When the clock does not transition from zero to one, the last bit of the shift register is moved to the output. The data is not shifted.
- The Enable/Reset must be kept on as long as the data is be shifted into and kept in the SHRB. When the Enable/Reset loses power flow, the SHRB is cleared, and all control relays or image register points comprising the SHRB are cleared to zero.
- If the Enable/Reset does not receive power flow the instruction is not executed and the output does not turn on.

SHRB	Parameter Type	Valid Parameter Types
Register Number:	Constant	Varies with configured Shift Register ranges
IR:	Bit Address	Y,C,B
N	Constant	1-1023

Warning	**Do not use the same reference number more than once for the SHRB and SHRW instructions. Assigning the same number can cause unpredictable machine operation.**

Word Shift Register (SHRW)

The Word Shift Register Instruction (SHRW) copies words from a memory location into a shift register. The shift register is located in V memory and can be up to 1023 words long.

The SHRW instruction has the following features:

- The Enable/Reset inputs must both be on for the SHRW instruction to be executed.
- When the Clock transitions from off to on, the word currently in memory location A is shifted into the shift register at the memory location specified by B. The shift occurs as follows:
- Word B + (N-1) is discarded.
- Word B + (N-2) is then copied to word B+(N-1); word B+(N-3_ is copied to word B+(N-2), etc.
- Word B is copied to word B+1; word A is copied to word B.
- After each shift is completed, the output is turned on for one scan.
- If the Enable tuns off, but the Reset remains on, all words currently in the SHRW are retained, but no words are shifted.
- If the Reset turns off, all words in the shift register are cleared to zero. The instruction is not executed, and there is no power flow at the instruction.

SHRW	**Parameter Type**	**Valid Parameter Types**
Register Number:	Constant Only	Varies with configured Shift Register ranges
A:	Word Addr	Form A
B:	Word Addr	V,W,G
N:	Constant	1-1023

Warning	**Do not use the same reference number more than once for the SHRB and SHRW instructions. Assigning the same number can cause unpredictable machine operation.**

Word Rotate (WROT)

The Word Rotate (WROT) instruction operates on the 4-bit segments of a word, rotating them to the right.

When the input is turned on, the WROT instruction is executed. IF the input remains on, the instruction is executed on every scan. The WROT instruction has the following operation:

- Each 4-bit segment of the word specified in memory location A is shifted to the right.
- A segment may be shifted up to 3 positions as specified by N.
- If A is not zero, the output is turned on when the instruction is executed. If A is zero, the output is turned off.

WROT	**Parameter Type**	**Valid Parameter Types**
Reference Number	Constant	0-32767
A	Word Address	Form B
B	Constant	1-3

Math and Logic Instructions

Add (ADD)

The Add instruction adds a signed integer (A) to a signed integer (B), and stores the result in R. When the input is ON (1), the ADD box is executed. If the input remains ON (1), the instruction is executed on every scan.

The ADD operation is executed as follows: R = A + B. If the sum is between -32768 and 32767, the output is turned ON (1). Otherwise, the output is turned OFF (0), indicating an addition overflow, and the R (result) contains the truncated sum (16 bits).

ADD	**Parameter Type**	**Valid Parameter Types**
Reference Number	Constant	0-32767
Parameter A:	Word Address	Form A
Parameter B:	Word Address/Constant	Form A or Const (-32768-32767)
Parameter R:	Word Address	Form B

Subtract (SUB)

The Subtract instruction (SUB) subtracts a signed integer in memory location B from a signed integer in memory location A and stores the result in memory location C.

When the input has power, the instruction is executed. The operation executed is C = A - B. If the result is less than (or equal) -32767 or greater than (or equal to) 32767, the output is turned on. Otherwise, the output is turned off and the truncated 16 bit result is stored in R.

SUB	**Parameter Type**	**Valid Parameter Types**
Reference Number	Constant	0-32767
A:	Word/Const	Form A or Const (-32768-32767)
B:	Word/Const	Form A or Const (-32768-32767)
R:	Word Address	Form B

Multiply (MUL)

The Multiply (MUL) instruction multiplies a signed integer in location A by a signed integer in location B. The product is stored in on long word, R and R+!.

When power is passed to the input of the MUL, the instruction is executed. The MUL operates as follows:

- The values in A and B are not affected by the operation
- When the multiplication is executed, the output is turned on.

MULT	**Parameter Type**	**Valid Parameter Types**
Reference Number	Constant	0-32767
A:	Word Address	Form A
B:	Word/Constant	Form A or Constant (-32768-32767)
R:	Word Add	Form B

Divide (DIV)

The Divide Instruction (DIV) divides a 32-bit signed word integer stored in A and A+1 by a 16-bit signed integer in B. The quotient is stored in R and the remainder is stored in R+1.

When the input is on, the DIV instruction is executed. IF B is non-zero, the division is done and the output is turned on. Otherwise, the output is turned off and the contents of R and R+1 do not change.

DIV	Parameter Type	Valid Parameter Types
Reference Number	Constant	0-32767
A:	Word Address or Constant	Form A or Constant (-32768-32767)
B:	Word Address or Constant	Form A or Constant (-32768-32767)
R:	Word Address	Form B

Absolute Value (ABS)

The absolute value instruction calculates the absolute value of a signed integer. When the input is turned ON (1), the absolute value instruction is executed. If the input is OFF (0), the instruction is not executed and there is not power at the output. The absolute value function is executed with the parameter A, as follows:

- If A>=0, A is not changed.
- If -32768 < A < 0, A is replaced with the value (0 - A).
- If A = -32768, A does not change.

ABS	Parameter Type	Valid Parameter Types
Parameter A:	Word Address	Form B

Square Root (SQRT)

The Square Root instruction (SQRT) finds the integer square root of a 32-bit (lng word) positive integer stored in memory locations AA and AA + 1. The result is stored in memory location B.

The SQRT instruction has the following properties:

- If the result of the square root is not an integer, the SQRT reports only the integer portion of the root.
- The operations is valid if 0 <= AA <=32767 * 32767.
- If the result is valid, the outputs turned on when the operation is executed. Otherwise, the instructions turned off and the contents of B does not change.

SQRT	Parameter Type	Valid Parameter Types
Reference Number	Constant	1-32767
A	Word Address	Form A
B	Word Address	Form B

Binary to BCD (CBD)

The Convert Binary to BCD instruction converts a binary representation of an integer (BIN) to an equivalent Binary Coded Decimal (BCD) value. It converts a 16-bit integer into a 32-bit BCD word. A BCD word is made up of 4 digits (0-9), with each digit represented by groups of 4 bits. Values up to 327667 can be converted into BCD.

CBD is described as follows:

- IF BIN contains an integer 0-32767. the value is converted to BCD, stored in BCD and (BCD+1) as shown below and the instruction output is turned on.

MSB			LSB	MSB			LSB
0	0	0	Ten Thousands	Thousands	Hundreds	Tens	Ones
	BCD				(BCD + 1)		

- If A is not within the range 0-32767, BCD and (BCD+1) are not changed and no power is passed.

CBD	Parameter Type	Valid Parameter Types
Instruction reference	Constant	0-32767
Parameter BIN:	Word Address	Form A
Parameter BCD:	Word Address	Form B

BCD to Binary (CDB)

The Convert BCD to Binary Instruction (CDB) converts a BCD element into its integer equivalent. When the input is O, the CDB instruction is executed. The CDB operation is described below:

- The number of digits (N) of the BCD value located in BCD is converted to its equivalent binary integer value stored in BIN.
- N may range from 1-4, and the BCD digit count is from right to left. For example, if N = 2 and the BCD number is A=1234, then 34 is converted and the value stored in B is 00100010.
- The output is turned on after the instruction is executed if the digits of the input words are valid. Each digit of the BCD value in BCD must be less than or equal to 9. The binary values 1010, 1011, 1100, 1101, 1110, and 1111 are invalid.

CDB	Parameter Type	Valid Parameter Types
Instruction reference	Constant	0-32767
Parameter BCD:	Word Address	Form A
Parameter BIN	Word Address	Form B
N	Constant	Number of digits converted, 1-4/

Word And (WAND)

The Word AND (WAND) instruction logically ANDs a word in memory location A with a word in memory location B, bit for bit. The result is stored in memory location C.

When the input is on, the instruction is executed. If the input remains on, the instruction is executed on every scan. The WAND has the following properties:

- The word stored in the memory location specified by A is AND with the word stored in the memory location B. The operation is done bit by bit. The words in A and B are not affect by the WAND instruction and will retains their original values.

A	B	C
0	0	0
0	1	0
1	0	0
1	1	1

- The result is stored in the memory location C.
- If C is not zero, the output is turned on when the instruction is executed. If C is zero, the output is turned off.

WAND	**Parameter Type**	**Valid Parameter Types**
Reference Number	Constant	0-32767
A	Word Address	Form A
B	Word/Const	Form A, Const –32767 - 32767
C	Word	Form B

Word Or (WOR)

The Word OR (WOR) instruction logically ORs a word in memory location A with a word in memory location B, bit for bit. The result is stored in memory location C.

When the input is on, the instruction is executed. If the input remains on, the instruction is executed on every scan. The WOR has the following properties:

- The word stored in the memory location specified by A is OR with the word stored in the memory location B. The operation is done bit by bit. The words in A and B are not affect by the WOR instruction and will retains their original values.

A	B	C
0	0	0
0	1	1
1	0	1
1	1	1

- The result is stored in the memory location C.
- If C is not zero, the output is turned on when the instruction is executed. If C is zero, the output is turned off.

WOR	Parameter Type	Valid Parameter Types
Reference Number	Constant	0-32767
A	Word Address	Form A
B	Word/Const	Form A, Const -32767 - 32767
C	Word	Form B

Word Exclusive Or (WXOR)

The Word Exclusive OR (WXOR) instruction Exclusive ORs a word in memory location A with a word in memory location B, bit for bit. The result is stored in memory location C.

When the input is on, the instruction is executed. If the input remains on, the instruction is executed on every scan. The WXOR has the following properties:

- The word stored in the memory location specified by A is XORed with the word stored in the memory location B. The operation is done bit by bit. The words in A and B are not affected by the WXOR instruction and will retain their original values.

A	B	C
0	0	0
0	1	1
1	0	1
1	1	0

- The result is stored in the memory location C.
- If C is not zero, the output is turned on when the instruction is executed. f C is zero, the output is turned off.

WXOR	**Parameter Type**	**Valid Parameter Types**
Reference Number	Constant	0-32767
A	Word Address	Form A
B	Word/Const	Form A, Const -32767 - 32767
C	Word	Form B

Move Instructions

Move Image Register From Table (MIRFT)

The Move Image Register From Table (MIRFT) instruction allows you to copy information into the control relays or the discrete image register from a table of consecutive word locations.

When the input is on, the MIRFT instruction is executed. The operation of the MIRFT instruction is defined as follows:

- The values of up to 256 (N) words (16-4096 bits) are copied, starting at the memory location specified by TS.
- The copy is placed in the control relays or the discrete image register. The LSB of the first word is copied into the point specified by IR.
- The beginning point in the control relays or the discrete image register must be on a eight point boundary (1, 9, 17, 25, etc.).
- All words are copied into the control relays or the image register during each scan.
- The output is turned on when the instruction is executed.

MIRFT	**Parameter Type**	**Valid Parameter Types**
Reference Number	Constant	0-32767
TS:	Word Address	Form A
IR:	Bit Address	X,Y,C,B *
N:	Constant Only	1-256

Move Image Register to Table (MIRTT)

The Move Image Register to Table (MIRTT) instruction allows you to copy information from the control relays or the discrete image register to a table of consecutive word locations.

When the input has power, the MIRTT instruction is executed. The operation of the MIRTT is defined as follows:

- The On/Off state of up to 4096 bits (256 words X 16 bits) is copied from the control relays or the discrete image register, starting at the bit address specified by IR.
- The starting point must be on a eight point boundary (1,9,17, etc.)/ bits are copied in groups of 16 bits.
- The copy begins with the lowest numbered bit address and is placed into work locations, beginning with the LSB of the word specified by TD.
- All bits are copied into the work locations during each scan. There must be a sufficient number of discrete points to copy all bits into the table of N words.
- The output is turned on when the instruction is executed.

MIRTT	**Parameter Type**	**Valid Parameter Types**
Reference Number	Constant	0-32767
IR:	Bit Address	X,Y,C,B *
TD:	Word Address	Form A
N:	Constant Only	1-256

Move Image Register to Word (MIRW)

The Move Image Register to Word (MIRW) instruction copies a specified number of bits from the discrete image register or the control relay memory locations to a designated word memory location. Up to 16 bits can be copied in a single scan.

When power is passed to the input of a MIRW, the instruction is executed. The operation of the MIRW is defined as follows:

- Up to 16 bits (N) are copied, beginning with the lowers numbered address, which is specified by IR.
- The bits are moved into the work memory location specified by A, beginning with the LSB of the word. If fewer than 16 bits are moved, the remaining bits are set to 0. All bits are copied during a single scan.
- The output is turned on when the instruction is executed.

MIRW	**Parameter Type**	**Valid Parameter Types**
Reference Number	Constant	0-32767
IR:	Bit Address	X,Y,C,B
A:	Word Address	Form B
N	Constant Only	1-16

Move Byte, Word, Element (MOVE)

The Move Element (MOVE) instruction copies data bytes, word, or long words from a source location to a destination location.

When the input is turned on, the MOVE instruction is executed. The operation of the MOVE is described as follows:

- The elements are specified in TS are copied to the destination specified by TD.
- The output is turned on after the instruction is executed, unless an error occurs.
- When the count is invalid or any reference data element are undefined, bits 6 and 11 in STW01 are set and STW200 contains a value of 5. The contents of the destination are not changed.
- You can specify the type of the data element to be moved. It can be a byte (8 bits long), word (16 bits), or long words (32 bits).
- The source can be specified as a constant value (signed integer), a direct address or an indirect address.

You can specify an index for the source address using SI as an index into a table when you want to copy elements of a table to a destination. SI designates the relative element, in the table reference by TS, which is to be copied. The element TS[0] is the first element in the table. SI can be a Constant, 0-65535. You can leave the field blank or enter a 0 for no indexing. You can enter any readable word (defined in the table below) that gives the element number of the first element to copy. If an indirect source address is used, the controller first resolves the address and then it indexes it.

You can specify a destination address by using the TD field. You can specify a direct address, any writeable word. MOVE copies the source elements into the memory location starting at this address. You can also specify an indirect address. The long word at this indirect address must contain another address, and MOVE copies the source elements into the memory locations start at this second address.

You can specify an index into the destination address using the optional field DI. This is an index into a table when you want to copy an element into a table. DI designates the relative element in a table, referenced by TD, into which the source is copied. The element at TD[0] is the first element in the table. DI can be either a constant (0-65535), blank or 0 for no indexing or you can enter a variable index. A variable index specifies any readable word. The content of this address is an unsigned integer, 0 -65535, that gives the element number of the first element in the table to which the source elements are copied. If an indirect destination address is indexed, the controller first resolves the address and then it indexes it.

You can specify the number of elements to be copied in the count field N. Count is defined as follows:

- A Constant count used to specify an unsigned integer in the range 1-32767.
- A variable count, with any readable word. The value of the count is determined by the contents of this work when the MOVE executes. The count range is 0 -32767, where 0 means that no elements are moved.

MOVE	Parameter Type	Valid Parameter Types
Reference	Constant	0-32767
TS:	Word/Constant	Form A (direct or indirect) , Constant -2048 - 2047
IS:	Blank/Constant/Word	Form A, Blank, Constant 0-65535
TD:	Word Address	Form A (Indirect addr), Form B (Direct Addr)
DI:	Blank/Constant/Word	Form A, Blank, Const 0-65535
N:	Word/Constant	Form A, Const 0-32767

Move Word (MOVW)

The Move Word (MOVW) instruction copies up to 256 contiguous words from one location to another. The starting address for the words to be moved is specified by A and the starting memory location for their destination is specified by B. All words are copied in a single scan.

When power is passed to the input of the MOVW, the instruction is executed. The operation of the MOVW is described as follows:

- A table of up to 256 (N) words having a starting memory location specified by A are copied.
- If a constant value is specified in A, then the constant is copied to all destination locations.
- The words are copied to a destination beginning at the memory location designated by B.
- The output is turned on when the instruction is executed.

MOVW	Parameter Type	Valid Parameter Types
Reference Number	Constant	0-32767
A:	Word/Const	Form A, Const -32768 - 32767
B:	Word	Form B
N	Constant Only	1-256

Move Word From Table (MWFT)

The Move Word from Table instruction (MWFT) copies a word from a table to a V-memory location. A table pointer designates the address of the next word in the table to be copied. One word is copied during each scan.

The MWFT operates as follows:

- When the Enable/Reset is off, the table starting address S is loaded into pointer A.
- When the Enable/Reset turns on, the box is enabled. When the input is turned on, the following actions occur.
- A word is copied from the table address specified by the value contained in pointer A to the memory location specified by B.
- After the word is copied, table A which holds the address of the next word in the table to be copied, is incremented by 1.
- If the Input and the Enable/ Reset remain on, one word is copied every scan. As each word is copied, the table pointer is incremented until N words have been copied.
- The output is trend on when the last word has been copied.
- When the instruction is reset, all table values remain unchanged and the destination address B contains the last word copied from the table.

MWFT	Parameter Type	Valid Parameter Types
Table Number:	Constant	Varies with Table Move configured range
A:	Word Addr	V,W,G
B:	Word Addr	V,W,G
S:	Word Addr	V
N	Constant	1-256

Warning	**Do not use the same reference number more than once for the MWTT and MWFT instruction. Assigning the same reference number can cause unpredictable machine operation.**

Move Word to Table (MWTT)

The Move Word to Table instruction (MWTT) copies a word from a source in memory to a destination within a table. A pointer designates the memory location in the table into which the next word is copied. One word is copied per scan.

The MWTT instruction works as follows:

- ❑ When the Enable/Reset is off, the table starting address S is loaded into pointer B.
- ❑ When the Enable/Reset runs on, the box instruction is enabled. When the input also turns on, the following can occur.

A word is copied from the memory location specified by A to the table memory location specified by the value contained in pointer B.

Pointer B, which holds the destination memory location in the table fro the next word, is incremented by 1.

If the input remains on, one word is copied every scan. As each word is copied, the table pointer is incremented until N words have been copied.

- ❑ The output is turned on when the last word has been copied.
- ❑ When the instruction is reset, all value in the table remain unchanged.

MWTT	**Parameter Type**	**Valid Parameter Types**
Table Number	Constant	Varies with Table Move configured range
A:	Word Addr	V,W,G
B:	Word Addr	V,W,G
S:	Word Addr	V
N	Constant	1-256

Warning	**Do not use the same reference number more than once for the MWTT and MWFT instruction. Assigning the same reference number can cause unpredictable machine operation.**

Move Word with Index (MWI)

The Move Word with Index (MWI) instruction allows you to copy up to 256 words from one area of V-memory to another area of V-memory during a single scan.

When power is passed to the input, the instruction is executed. The MWI has the following features:

- A table of up to 256 memory locations having a starting index specified in A is copied.
- The copied words are placed in a destination table in memory, beginning at the start index specified in B.
- All words are copied into the destination table each scan.
- The output is turned on when the instruction is executed.
- If either the source or the destination pointer plus table size exceeds memory size, instruction is not executed. The output is turned off and bit 11 in Status Word 1 is set.

MWI	Parameter Type	Valid Parameter Types
Reference Number	Constant	0-32767
A	Word Addr	V,W,G
B	Word Addr	V,W,G
N	Word Addr	V,W,G

Move Word to Image Register (MWIR)

The Move Word to Image Register instruction (MWIR) copies a specified number of bits from a word memory location to the discrete image register or into the control relay memory locations. All bits are copied in a single scan.

When power is passed to the input, the instruction is executed. The instruction has the following features:

- Up to 16 bits (N) in word memory location A are copied, beginning with the least significant bit of the word.
- Bits are copied into the discrete image register or into the control relay memory locations, starting at the address designated by IR. The bits are copied during a single scan.
- The output is turned on when the instruction is executed.

MWIR	Parameter Type	Valid Parameter Types
Reference Number	Constant	0-32767
A	Word Addr	Form A
IR	Bit Addr	Y,C,B
N	Constant Only	1-16

Search Table for Equal (STFE)

The Search Table for Equal (STFE) instruction locates the next occurrence of a word in a table that is equal to a source word. The position of the matching word is shown by an index.

The STFE has the following properties:

- The Reset must be turned off to initialize the index, setting it to -1.
- The Reset must then be turned on before the STFE can operate.
- When the Enable turns on, the index is incremented by one and specifies the next word in the table to be compared with the source word. The value contained by the index ranges from 0 to N-1 while the STFE is being executed. N is the length of the table.
- The source word WS and the word in the table TS specified by the index are compared.
- If the two words are equal, the STFE output turns on for one scan and then turns off. The index contains the position of them attaching word in the table for the duration of the scan. The contents of the index must be used or saved during this scan since the STFE looks for the next match on the next scan as long as the Enable and Reset remains on.
- If the two words are not equal, the index is incremented by one and the next word in the table is compared to the source word.
- If no matches are found in the table, the output remains off. The index contains the position of the last word in the table.
- The entire table is searched ruing one scan until one match or no match is found.
- If the Enable turns off while the Reset is on, the index holds its current value. If the Reset turns off, the index resets to -1.
- After the entire table has been searched, the STFE must be reset in order to be executed again.

STFE	**Parameter Type**	**Valid Parameter Types**
Reference Number	Constant	0-32767
WS:	Word Address	Form A
TS:	Word Address	Form A
IN:	Word Address	V,W,G
N	Constant Only	1-256

Search Table for Not Equal (STFN)

The Search Table for Not Equal Instruction (STFN) locates the next occurrence of a word in a table that is not equal to a source word. The position of the non-matching word is shown by an index and the value of the non-matching word is copied into a specified memory location.

The STFN has the following properties:

- The Reset must be turned off to initialize the index, setting it to -1.
- The Reset must then be turned on before the STFN can operate.
- When the Enable turns on, the index is incremented by one and specifies the next word in the table to be compared with the source word. The value contained by the index ranges from 0 to N-1 while the STFN is being executed. N is the length of the table.
- The source words WS and the word in the table TS specified by the index are compared.
- If the two words are not equal, the STFN output turns on for one scan then turns off. The value of the non-matching word is copied into another memory location specified by WO. The index contains the position of the non-matching word in the table for the duration of the scan. The contents of the index must be used or saved during the scan since the SRFN looks for the next match on the next scan as long as the Enable and Reset remains on.
- If the two words are equal, the index is incremented by one and the next word in the table is compared to the source word.
- If no mismatches are found in the table, the output remains off. The index contains the position of the last word in the table.
- The entire table is searched during one scan until mismatch or no mismatch is found.
- If the Enable turns off while the Reset is on, the index holds its current value. If the Reset does turn off, the index resets to -1.
- After the entire table has been searched, the STFN must be reset in order to executed again.

STFN	**Parameter Type**	**Valid Parameter Types**
Reference Number	Constant`	0-32767
WS:	Word Address	Form A
TS:	Word Address	Form A
IN:	Word Address	V,W,G
WO:	Word Address	Form B
N	Constant	1-256

Table to Table And (TAND)

The Table to Table AND (TAND) instruction ANDs the corresponding bits in two tables and places the results in a specified third table. If both bits are 1, then the resultant bit is set to 1, otherwise, the result bit is 0.

The TAND has the following properties:

- When the input turns on, a comparison is made between each bit of each word in the first (T1) and second (T2) tables.
- Each pair of bits is ANDed and the resultant bit is placed in the third table (TD). If both bits are 1, then the resultant bit is set to 1. Otherwise, the resultant bit is set to 0.

TAND	**Parameter Type**	**Valid Parameter Types**
Reference Number	Constant	0-32767
T1:	Word Address	Form A
T2:	Word Address	Form A
TD:	Word Address	Form B
N	Constant Only	1-256

Table to Table Or (TOR)

The Table to Table OR (TOR) instruction ORs the corresponding bits in two tables and places the results in a specified third table. If either bit is 1, then the resultant bit is set to 1. Otherwise, the resultant bit is set to 0.

The TOR operation is described as follows:

- When the input is on, a comparison is made between each bit of the each word in the first, T1 and the second. T2 tables.
- Each pair of bits is Ored, and the resultant bit is placed in the third table (TD). If either bit is 1, then the resultant bit is set to 1. Otherwise, the resultant bit is set to 0.
- The bits in all the word of the two tables are Ored each scan.
- The output is turned on when the instruction is executed.

TOR	**Parameter Type**	**Valid Parameter Types**
Reference Number	Constant	0-32767
T1:	Word Address	Form A
T2:	Word Address	Form A
TD:	Word Address	Form B
N	Constant Only	1-256

Table to Table Exclusive Or (TXOR)

The Table to Table Exclusive OR (TXOR) instruction executes an Exclusive OR on the corresponding bits in two tables and places the results in a specified third table. If the bits compared are the same, the resultant bit is set to a 0. If the bits compared are different, the resultant bit is set to 1.

The TXOR instruction is described below:

- When the input turns on, a comparison is made between each bit of each word in the first T1 and the second T2 tables.
- An Exclusive OR is executed on each pair of bits, and the resultant bit is placed in the third table (TD). If the bits compared are either both 1 or both 0, the resultant bit is set to a 0. If the bits compared are unlike (1 and 0), the resultant bit is set to 1.
- An Exclusive OR is executed on the bits in all of the words of the two tables each scan.
- The output is turned on when the instruction is executed.

TXOR	**Parameter Type**	**Valid Parameter Types**
Reference Number	Constant	0-32767
T1:	Word Address	Form A
T2:	Word Address	Form A
TD:	Word Address	Form B
N	Constant Only	1-256

Table Complement (TCPL)

The Table Complement (TCPL) instruction inverts the status of each bit in a table and places the results in another specified table.

The TCPL instruction has the following operations:

- When the input turns on, each bit in the source table specified by TS is inverted and stored in the destination table specified by TD. A 0 is inverted to 9. A 1 is inverted to 0.
- The bits in ALL words of the table are inverted each scan.
- The output is turned on when the instruction is executed.

TCPL	**Parameter Type**	**Valid Parameter Types**
Reference Number	Constant	0-32767
TS:	Word Address	Form A
TD:	Word Address	Form B
N	Constant Only	1-256

Text (Text Box Description)

The Text box allows you to place textual information, such as copyright, software version, or other text into your RLL program. The instruction forms a single network and takes no action. The Text Box's sole purpose is for documentation.

The text box can hold up to five lines of 40 characters each. Characters allowed in the text box are: A through Z, 0 through 9, space, and printable special characters. Text must be entered within quotation marks.

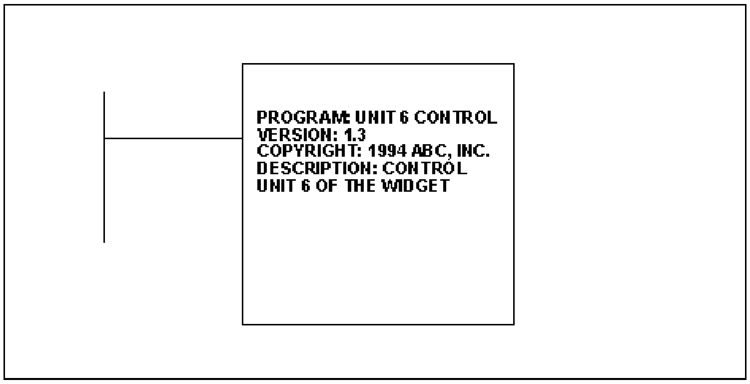


Table to Word (TTOW)

The Table to Word (TTOW) instruction copies a word in a table and places it in another location.

The operation of the TTOW instruction is described below:

- The Reset must be on for the instruction to be executed.
- When the Enable turns on, a copy is made of the specified word in the table TS. The index (IN) indicates which word in the table is copied. The value contained by the index ranges from 0 to N-1, when N is the length of the table. If 0 <= IN <= N, the word is copied. If N <= IN or N < 0, the word is not copied.
- The word is placed in the memory location specified by WD. After the word is placed there, the value contained by the index is incremented by one.
- If both Enable and Reset remain on, one word is duplicated each scan.
- If the Enable turns off while the Reset is on, the index holds its current value and the word is not moved. If the Reset turns off, the index resets to 0.
- The TTOW output remains on until the last word in the table is copied, then turns off.
- The TTOW must be reset after the output turns off in order to be executed again.

TTOW	**Parameter Type**	**Valid Parameter Types**
Reference Number	Constant	0-32767
WD:	Word Address	Form B
TS:	Word Address	Form A
IN:	Word Address	V,W,G
N	Constant only	1-256

Word to Table (WTOT)

The Word to Table (WTOT) instruction places a copy of a word at a specified address within a table.

The operation of the WTOT is described as follows:

- The Reset must be on for the instruction to be executed.
- When the Enable turns on, a copy of the source word WS is placed in the destination table TD.
- The index (IN) indicates where the word is placed in the table. The value contained by the index ranges from 0 to N-1, where N is the length of the table. If 0 <= IN <= N, the word is moved. If N < IN or N < 0, the word is not moved.
- After the word is placed into the table, the value contained by the index is incremented by one.
- If both Enable and Reset remain on, one word is moved each scan.
- If the Enable turns off while the Reset is on, the index holds its current value and the word is not moved. If the Reset turns off, the index resets to 0.
- The WTOT output remains on until a word is placed in the last position in the table. It then turns off.
- The WTOT must be reset after the output turns off in order to be executed again.

WTOT	**Parameter Type**	**Valid Parameter Types**
Reference Number	Constant	0-32767
WS:	Word Address	Form A
TD:	Word Address	Form B
IN:	Word Address	V,W,G
N:	Constant Only	1-256

Word To Table And (WTTA)

The Word to Table AND (WTTA) instruction ANDs each bit in a source word with the corresponding bit of a designated word in a table. The results are placed in a destination table. If both bits are 1, a 1 is stored in the destination table. Otherwise, the resultant bit is set to 0.

The WTTA instruction is described below:

- The Reset must be on for the instruction to be executed.
- When the Enable turns on, each bit of the source word WS and of the specified word in the table TS is compared. The index (IN) indicates which word in the table is ANDed. The value contained by the index ranges from 0 to N-1, where \N is the length of the table. If 0 <=IN<N, the word in ANDed. If N <= IN or N < 0, the word is not ANDed.
- Each pair of bits is ANDed, and the resultant bit is placed in the destination table TD. If both bits are 1, the resultant bit is set to 1. Otherwise, the resultant bit is set to 0. After a word in the table is compared, the value contained by in the index is incremented by one.
- If both Enable and Reset remain on, the source word and a word in the table are ANDed each scan.
- If the Enable turns off while the Reset is on, the index holds its current value and the AND does not occur. If the Reset turns off, the index resets to 0.
- The WTTA output remains on until the last word in the table has been ANDed with the source word. It then turns off.
- The WTTA must be reset after the output turns off in order to be executed again.

WTTA	**Parameter Type**	**Valid Parameter Types**
Reference Number	Constant	0-32767
WS:	Word Address	Form A
TS:	Word Address	Form A
TD:	Word Address	Form B
IN:	Word Address	V,W,G
N:	Constant Only	1-256

Word to Table Or (WTTO)

The Word To Table OR (WTTO) instruction ORs each bit in a source word with the corresponding bit of a designated word in a table. The results are placed in a destination table. If either bit is 1, a 1 is stored in the destination table. Otherwise, the resultant bit is set to 0..

The WTTO instruction is described below:

- The Reset must be on for the instruction to be executed.
- When the Enable turns on, each bit of the source word WS and of the specified word in the table TS is compared. The index (IN) indicates which word in the table is Ored. The value contained by the index ranges from 0 to N-1, where N is the length of the table. If 0 <= IN < N, the word is Ored. IF N <= IN or N < 0, the word in not Ored.
- Each pair of bits is Ored, and the resultant bit is placed in the destination table TD. If either bit is 1, then the resultant bit is set to 1. Otherwise, the resultant bit is set to 0. After a word in the table is compared, the value contained by the index is incremented by one.
- If both Enable and Reset remain on, the source word and a word in the table are Ored each scan.
- If the Enable turns off while the Reset is on, the index holds its current value and the OR does not occur.
- If the Reset turns off, the index reset to 0.
- The WTTO output remains on until the last word in the table has been Ored with the source word. It then turns off.
- The WTTO must be reset after the output turns off in order to be executed again.

WTTO	**Parameter Type**	**Valid Parameter Types**
Reference Number	Constant	0-32767
WS:	Word Address	Form A
TS:	Word Address	Form A
TD:	Word Address	Form B
IN:	Word Address	V,W,G
N:	Constant Only	1-256

Word to Table Exclusive Or (WTTXO)

The Word to Table Exclusive OR (WTTXO) instruction executes an Exclusive OR on each bit in a source word with the corresponding bit of a designated word in a table. The results are placed in a destination table. If the bits compared are the same, the resultant bit is set to a 0. Otherwise, the resultant bit is set to a 1.

The WTTXO instruction is described below:

- The Reset must be on for the instruction to be executed.
- When the Enable turns on, each bit of the source word WS and of the specified word in the table TS is compared. The index (IN) indicates which word in the table is Exclusive Ored. The value contained by the index ranges from 0 to N-1, where N is the length of the table. If 0 <= IN < N, the word is Ored. If N <= IN or N < 0, the word in not Ored.
- Each pair of bits is Exclusive Ored, and the resultant bit is placed in the destination table TD. If the bits compared are the same, the resultant bit is set to a 0. If the bits compared are different, then the resultant bit is set to 1. After a word in the table is compared, the value contained by the index is incremented by one.
- If both Enable and Reset remain on, the source word and a word in the table are Exclusive Ored each scan.
- If the Enable turns off while the Reset is on, the index holds its current value and the Exclusive OR does not occur.
- If the Reset turns off, the index reset to 0.
- The WTTXO output remains on until the last word in the table has been Ored with the source word. It then turns off.
- The WTTXO must be reset after the output turns off in order to be executed again.

WTTXO	**Parameter Type**	**Valid Parameter Types**
Reference Number	Constant	0-32767
WS:	Word Address	Form A
TS:	Word Address	Form A
TD:	Word Address	Form B
IN:	Word Address	V,W,G
N:	Constant Only	1-256

Control Instructions

Special Function Program Call (SFPGM)

The Special Function Program Call (SFPGM) instruction is used to call a SF program for execution.

The SFPGM instruction can be used anywhere within the RLL program that a box instruction can be used. When a priority/non-priority or cycle SF program is called by the SFPGRM instruction, the SF program is placed in a queue for execution. Up to 32 SF programs of each type (total 96 in 3 queues) can be queued at a given time. If a queue is full, the request for placement in the queue is made again on the next scan. This continues as long as the input to the RLL SFPGM instruction remains on.

Priority/Non-Priority SF Programs. When power flow to the RLL SFPGM instruction transitions from off to on, the output from the instruction is examined. If the output is off, and the SF program is not currently being executed, the SF program is placed in the queue for execution. The SFPGM instruction has the following features:

- After the SF program is executed, the output is turned on.
- The SF Program does not executed again until the input to the SFPGM instruction transitions from off to on.
- If the controller changes from PROGRAM to RUN mode while the input to the RLL SFPGM instruction is on, the SF program is queued for execution.
- **Cyclic Programs.** When power flow to the RLL SFPGM instruction transitions from off to on, the cyclic SF program is placed in the queue for execution. The SF Program has the following features:
- After the cyclic SF program is executed one time, the output is turned on. The SF program is automatically re-queued for execution, based on the programmed cycle time. This process continues as long as the input to the RLL SFPGM instruction is on.
- The output remains on until the input to the RLL SFPGM instruction is turned off.
- A cyclic SF program is removed from the queue when it completes a scheduled cycle, and the SFPGM instruction's input is turned off.

SFPGM	Parameter Type	Valid Parameter Types
SF Program Number	Constant	1-1023

Special Function Subroutine (SFSUB)

The Special Function Subroutine (SFSUB) instruction is used to call a SF subroutine for execution.

If the subroutine number is 0, on the instruction parameters are evaluated (this is a special SFSUB 0). You can use a SFSUB 0 to executed up to 5 expressions without calling an actual SF subroutine or program. The programming device may limit the length of the expression that can be placed into the P[n] fields. Multiple SFSUB instructions with the same value of Program Number can be used in your program since your application may require multiple accesses to the same SF subroutine but with different parameters for each access.

The field P[n] can be one of the following types:

- Constant - any integer or real number.
- Discrete or word element.
- Expression - an expression is a logical group of tokens evaluating to an address or a value, where a token is the smallest indivisible unit (element address, operator, constant, and parenthesis).

The SFSUB instruction can be used anywhere within the RLL program that a box instruction can be used. When power flow to the SFSUB instruction transitions from off to on, the output from the FLL SFSUB instruction is examined to determine subsequent actions

If the instruction is not currently executing, then the instruction is placed in one of the SFSUB queues for execution. There are two SFSUB execution queues, one to handle SFSUB 0 instructions and the other to handle all other SFSUB instructions.

When an SFSUB 0 instruction is pulled off from its execution queue, the instruction parameters are evaluated and the instruction output is turned on. When SFSUB instructions are pulled off from the other execution queue, the instruction parameters are evaluated, statements in the corresponding SF subroutine are executed, and the instruction output is turned on.

SFSUB	**Parameter Type**	**Valid Parameter Types**
Program Number:	Constant Only	1-1023
ER:	Bit Address/Blank	C,Y,WY,V or Blank
P1:	Const/Bit/Word/Expression	Form A, complex expressions (refer to sec 7.8)***, Constants
P2: etc.		

End (END)

The END instruction unconditionally terminates the scan. The instruction does not have any parameters.

The END instruction should always be used to terminate your program. When a controlled executes the END instruction, the program scan is terminated. Any instruction after the END are ignored. The END instruction must be the only instruction in the network.

Do not use the END instruction to separate RLL tasks. If you use and RLL subroutine, place an END instruction between the last network of the main RLL program and the first network of the subroutine.

End Conditional (ENDC)

The ENDC instruction can terminate the program scan under specific conditions. Any instruction after the ENDC instruction are not executed.

When the ENDC instruction is executed, the current program scan is terminated. ENDC operates in conjunction with an input and is executed only when there is power at the input. When the input is off, the ENDC instruction is not executed and the program scan is not terminated.

When the ENDC instruction is active, ladder logic following the ENDC is not executed and outputs following the ENDC are frozen. An active ENDC functions as an end statement for MCRs and JMPS that preceded it, if it is in their zones of control. Outputs between the MCR and JMP and the END remain under the control of the MCR or JMP.

For an ENDC contained within a SKP zone of control, the ENDC is overridden if the SKP receives power flow,

Jump (JMP)

The Jump instruction (JMP) is used to freeze the values of the discrete image register points of the controlled outputs in the JMPs zone of control. The instruction can be used when you need to duplicate the outputs when the outputs are controlled by different logic.

The JMP operates as an output update enable instruction. The JMP must have power flow and cannot be nested within the zone of control of a JMP not having power flow (logic in the JMP zone of control changes the status of the outputs). The JMP supports these features:

- Dscrete outputs between a JMP and its corresponding JMPE do not change when the JMP loses power flow.
- JMPE marks the end of the zone of control for the JMP having the same reference number. If you do not use the JMPE, the remainder of the program is placed under the control of the JMP. You can make the JMPE conditional by placing a contact on the same network as the JMPE.
- When a MCR loses power flow, JMP instructions within the MCR's zone of control are overridden. This means that all outputs in the MCR's zone of control are turned off when the MCR loses power flow, even when the outputs are frozen in an ON state by a JMP. This includes network outputs with the network.

Jump End (JMPE)

Master Control Relay (MCR)

The Master Control Relay is used to turn off blocks of outputs controlled by segments of RLL programs. This is done by clearing the discrete image register points of the controlled outputs to zero.

Although the MCR controls the coils and discrete outputs of box instruction within its zone of control, it does not control the power rail. This means that box instructions will continue to operate normally. In order to disable a box, use an MCR controlled coil output as a normal contact on the same network that contains the box.

The MCR operates as an output/enable instruction with the following features:

- The MCR must have power flow, and not be nested within the zone of control of an MCR not having power flow, for discrete outputs in the MCR zone of control to turn on or stay on.
- The MCR controls the coils and discrete outputs of box instruction in its zone of control.
- MCRE marks the end of the zone of control for the MCR having the same reference number. If you do not use the MCRE, the remainder of the program is placed under the control of the MCR.
- You can make the MCRE conditional by placing a contact on the same network as the MCRE. Make user that the contact that controls the MCRE is not controlled by the MCR.

MCR	Parameter Type	Valid Parameter Types
Reference Number	Constant	1-8

Return From Subroutine (RET)

The Return from Subroutine (RTN) instruction ends the execution of a RLL subroutine, and returns program execution to the network following the GTS instruction.

A RLL subroutine is executed until a RTN instruction is encountered. When an active RTN is reached in the subroutine, execution is returned to the first instruction following the GTS instruction in the RLL program. The RTN instruction can be either unconditional or conditional. The conditional RTN can be used within a subroutine to satisfy a condition that requires termination of the subroutine. The unconditional RTN must be use d as the last instruction in a subroutine.

Go to Subroutine (GTS)

The Go to Subroutine (GTS) allows you to write RLL programs preceded by a subroutine number and call them to be used where they are needed. A reference number is entered to designate the subroutine number.

When power is passed to the input of the GTS instruction, the RLL program calls the subroutine indicated by the reference number. If there is no power flow to the GTS instruction, the subroutine is not called.

Warning	**When online, it is important to create the subroutine and all of the instructions required to define a subroutine (END, RTN, SBR, GTS/PGTS/PGTSZ) BEFORE placing the processor in RUN mode. If you enter a call to a subroutine that does not exist, unpredictable PLC results may occur.**

GTS	**Parameter Type**	**Valid Parameter Types**
Reference number	Constant	Subroutine number, 1-255

Call Subroutine (SBR)

The Subroutine (SBR) instruction is used before a set of RLL instructions (the RLL subroutine) to be executed only when they are called by the GTS, PGTS, or PGTSZ instruction.

When a subroutine is called, it executes until either a conditional RTN with power flow or an unconditional RTN is encountered. When this occurs, RLL execution returns to the instruction following the calling GTS, PGTS or PGTSZ instruction. Subroutines have the following features:

- Place all subroutines at the end of the main RLL program.
- Separate the main RLL program from the subroutines with an unconditional END instruction.
- A subroutine must be terminated by an unconditional RTN instruction, or a compile error will be generated. An END within a subroutine also generates an errors.
- The unconditional RTN instruction separates a subroutine from a subsequent subroutine.
- You can nest subroutines to 32 levels. A run-time nonfatal errors occurs when you exceed 32 levels (bit 7 in STW1 is set indicating a stack overflow).
- When you pass parameters to the subroutine by calling the subroutine from a PGTS instruction, refer to discrete parameters as Bn and word parameters as Wn, where n is the number of the parameter in the PGTS.
- MCRs, JMPs and SKPs are effected with subroutines. All MCRs and JMPs in a subroutine remain active after a RTN if the instruction within the SBR do not turn them off before the RTN. MCRs and JMPs that are active at the time that the subroutine is called, remain active while the SBR is executing. A SKP/LBL pair must be defined within the same SBR or a compile error occurs.

SBR	Parameter Type	Valid Parameter Types
Subroutine Number	Constant	1-255 (GTS)
		1-32 (PGTS or PGTSZ)

Warning	**Do not use the same subroutine number more than once. Assign the same subroutine number more than once can cause unpredictable machine operation.**

Skip (SKP)

The SKP and LBL instructions provide a means of enabling or disabling program segments during a scan. These instructions are often used when duplication of outputs is required and those outputs are controlled by different logic. These instructions can be used to decrease scan time since the interactions between any active SKP and LBL instructions are not executed.

The SKP and LBL support the following features:

- The SKP and LBL instruction must be used together. The LBL must appear before the instruction that terminates the current program segment (TASK, END, RTN). A SKP without a LBL generates a compile error.
- If you use RLL subroutines, you can use up to 255 SKP/LBL instructions within each subroutine and up to 255 SKP/LBL instructions for each TASK segment in the program.
- The reference numbers for the SKP/LBL instruction range from 1 to 255, and numbers cannot be duplicated within a given subroutine or TASK segment.
- The subroutine set distance from the main RLL program and reference numbers used in the subroutine can also be used in the main program. That is SKP23 is the main program does not interfere with a SKP23 in the subroutine.
- When the SKP receiver power flow, all ladder logic between the SKP and its associated LBL is ignored by the controller. Output between the SKP and the LBL are frozen in their current states.
- All ladder logic within the SJP zone of control is executed normally when the SKP does not have power flow.
- For a SKP to LBL function located within the zone of control of an MCR or JMP, the SKP or LBL function overrides the MCR or JMP when the SKP has power flow.
- The zone of control for a SKP is limited to the task segment or subroutine in which the SJP is used; the matching LBL must be defined after the SJP and be located in the same task segment or subroutine as the SKP.
- For a JMPE or MCRE contained within a SKP's zone of control, the program functions as if the JMPE or MCRE is located at the end of the program whenever the SKP is active.

NOTE: *hen a SKP is active, timers between the SKP and its LBL do not run. Be careful when using timer and drum instructions if you want them to operate while a SKP is active.*

SKP	Parameter Type	Valid Parameter Types
Reference Number	Constant	1-255

Label (LBL)

The SKP and LBL instructions provide a means of enabling or disabling program segments during a scan. These instructions are often used when duplication of outputs is required and those outputs are controlled by different logic. These instructions can be used to decrease scan time since the interactions between any active SKP and LBL instructions are not executed.

The SKP and LBL support the following features:

- The SKP and LBL instruction must be used together. The LBL must appear before the instruction that terminates the current program segment (TASK, END, RTN). A SKP without a LBL generates a compile error.
- If you use RLL subroutines, you can use up to 255 SKP/LBL instructions within each subroutine and up to 255 SKP/LBL instructions for each TASK segment in the program.
- The reference numbers for the SKP/LBL instruction range from 1 to 255, and numbers cannot be duplicated within a given subroutine or TASK segment.
- The subroutine set distance from the main RLL program and reference numbers used in the subroutine can also be used in the main program. That is, SKP23 is the main program and does not interfere with a SKP23 in the subroutine.
- When the SKP receiver power flow, all ladder logic between the SKP and its associated LBL is ignored by the controller. Output between the SKP and the LBL are frozen in their current states.
- All ladder logic within the SJP zone of control is executed normally when the SKP does not have power flow.
- For a SKP to LBL function located within the zone of control of an MCR or JMP, the SKP or LBL function overrides the MCR or JMP when the SKP has power flow.
- The zone of control for a SKP is limited to the task segment or subroutine in which the SKP is used; the matching LBL must be defined after the SKP and be located in the same task segment or subroutine as the SKP.
- For a JMPE or MCRE contained within a SKP's zone of control, the program functions as if the JMPE or MCRE is located at the end of the program whenever the SKP is active.

LBL	Parameter Type	Valid Parameter Types
Reference Number	Constant	1-255

Scan Synchronization Inhibit (SSI)

560/565 only

Parameterized Go to Subroutine (PGTS)

The PGTS instruction operates similarly to the GTS instruction. The PGTS is used to call a section of the RLL program that is preceded by a subroutine number and executes it. Unlike the GTS, the PGTS allows you to pass parameters to a subroutine. These parameters allow you to write a generic subroutine using parameter identified (IN1-IN20) instead of specific memory locations. Several PGTS instruction using different memory locations as parameters can then call the same general subroutine number.

To Program the Instruction:

To program the instruction, first insert the instruction into the ladder network. Using the Program menu, new parameters may be added or deleted by clicking the appropriate menu option. Once a parameter has been added, it can be selected and edited by clicking once on said parameter from within the box instruction. To delete a parameter, use the menu selection found in the Program menu.

The PGTS instruction works as follows:

- When the input is turned on, the contents of each parameter are set equal to the contents of the memory location specified in the parameter field. Then the subroutine indicated by the PGTS number is called.
- When the subroutine returns control to the main RLL program, the contents of the memory location specified in each read write (IO) parameter field is set equal to the contents of the parameter. The contents of memory locations designated IN are not changed.
- Contents of parameters are stored in PGTS discrete and word parameter areas. When you use a parameter in the subroutine, refer to discrete points as B[n] and words as W[n] where n = number of the parameter.
- You must enter the parameters consecutively. An error will be displayed if you do not.
- If you do not need to specify parameters, it is recommended that you use the GTS instruction instead. The GTS instruction uses less L memory.
- While you can still access any memory location from a subroutine, the PGTS allows you create a generic subroutine that is called by multiple PGTS instruction, varying the parameters.
- If you use an instruction that copies long words into or from the subroutine, you need to allocate a parameter for each word of each long word that is copied.

NOTE: *void a direct reference in a subroutine to a memory location that is also identified as a parameter in the PGTS instruction. If you don't, you may create a condition where the value of the parameter and the value in the memory location do not agree.*

PGTS	Parameter Type	Valid Parameter Types
Program Number:	Constant	1-32
IN: (input)	Word/Bit	Form A (Word), Form C (Bit)
I0: (output)		

Parameterized Go to Subroutine 0 (PGTSZ)

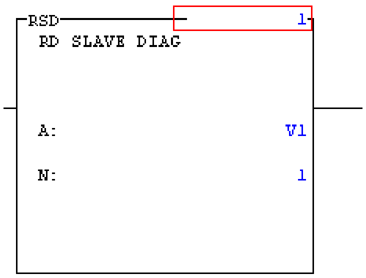
The PGTSZ instruction operates similarly to the PGTS instruction. The PGTSZ calls an RLL subroutine for execution and passes parameters to it. Unlike PGTS, the PGTSZ clears all discrete I/O parameters when the input to the PGTSZ is off.

When the input is turned on, operation is identical to that of the PGTS. If the input is turned off, all discrete I/O parameters are turned off, and the subroutine is not called.

PGTSZ	Parameter Type	Valid Parameter Types
Program Number:	Constant	1-32
IN: (input)	Word/Bit	Form A (Word), Form C (Bit)
I0: (output)		

Read Slave Diagnostic (RSD)

The Read Slave Diagnostic instruction transfers a PROFIBUS-DP slave's diagnostic buffer to user memory.



Field	Valid Values	Function
#	1 - 112	Instruction reference number. The number entered indicates the address of the PROFIBUS-DP slave whose diagnostic is to be read. Numbers can be repeated.
A	Any writeable word	Starting memory location for the destination.
N	1 - 256	Maximum number of words to be read. See Table 6–3.

The diagnostic buffer, whose address in user memory is specified by A, is formatted as shown below.

Word	Byte	Content
A	0	Status as follows: 0 Transfer successful. 1 Transfer successful. A previous diagnostic was signaled and not read. 2 Transfer failed. The specified slave has not signaled a diagnostic.
	1	Length, in bytes, of actual diagnostic.
A+1...A+N-1	all	Diagnostic area

NOTE: *The length (byte 1 of word A) indicates the actual diagnostic length, as signaled by the PROFIBUS-DP slave. If the size [(N-1)*2] of the destination buffer's diagnostic area is less than the actual diagnostic length, the diagnostic is truncated by the transfer.*

RSD Operation

- When the input is on, the RSD box executes. If the input remains on, the operation executes on every scan. The operation of RSD is as follows:
- If the PROFIBUS-DP I/O subsystem is stopped or if the indicated slave has not signaled a diagnostic since the last execution of an RSD instruction for the slave, the destination buffer's status byte is set equal to 2 and the length is set equal to 0.
- If the slave has not signaled more than one diagnostic since the last execution of an RSD instruction for the slave, the destination buffer's status byte is set equal to 0, the length byte is set equal to the length of the last diagnostic signaled, and the value (possibly truncated) of the latest signaled diagnostic is copied to the diagnostic area.
- If the slave has signaled more than one diagnostic since the last execution of an RSD instruction for the slave, the destination buffer's status byte is set equal to 1, the length byte is set equal to the length of the last diagnostic signaled, and the value (possibly truncated) of the latest signaled diagnostic is copied to the diagnostic area.

If the input is off, the instruction does not execute and the output is off.

NOTE: *Status words STW232 through STW238 indicate the PROFIBUS-DP slaves that have signaled a diagnostic that has not been read by an RSD instruction. Use a bit-of-word contact specifying the slave's status word bit as the input to the RSD instruction. Do this in order to execute the instruction whenever there is a diagnostic for the slave corresponding to the bit.*

NOTE: *The format of a slave's diagnostic buffer is dependent upon the PROFIBUS-DP slave type. See the user documentation for your slave(s).*

External Subroutine Call (XSUB)

The External Subroutine Call (XSUB) allows you to pass parameters to a subroutine that is developed offline in a non-RLL programming language such as C or Pascal, and then call the subroutine for execution.

The operation of the XSUB is described below:

- Parameters must be numbered consecutively.
- When the input is turned on, the parameters are pushed on the user stack in order from the last parameter to the first parameter and then the subroutine is called. When a discrete data element (X,Y,C,B) is specified as an IN parameter, the discrete value is passed in the least significant bit of a long word. All other bits of the long word are unspecified. When a discrete data element is specified as an I/O parameter, the address of the data element is passed. The actual value of the data element is contained in the least significant bit of the byte at this address. Other bits of this byte are unspecified. When a word data element (V,K) is specified as an IN parameter, the value of the long word at this specified data element and the specified data element + 1 is passed. The addressed word is in the most significant half, and the next consecutive word is in the least significant half. Any readable data element is allowed. When a word data element is specified as an IO parameter, the address of the data element is passed. The value of the parameter is contained at this address.
- An XSUB in RLL with no external subroutine causes the user program error bit 6 to be set in STW1. The reason is defined as 6 in STW200. The controller remains in RUN mode.

XSUB	Parameter Type	Valid Parameter Types
Program Number:	Constant Only	1-32767 or 1-65535 (software)
IN:	Bit/Word	Form A (Word), Form C (Bit) (Bits don't work?)
I0:		
IN		

New Task (TASK)

The Start New RLL Task (TASK) instruction is used to delimit the main RLL task and the cyclic RLL task.

The TASK instruction is described as follows:

- The Task instruction indicates the RLL instruction which follow it comprise an RLL task segment, where Task Number = 1 for segments of the main RLL task, = 2 designates segments of the cyclic RLL task, and = 8 designates segments of the interrupt task.
- Task 1 is assumed when the first rung does not contain a TASK instruction. A task can consist of multiple segments, each preceded by a TASK instruction. The segments do not have to be contiguous. Terminate an RLL task with another TASK instruction or with the END instruction.
- Task 2 is executed with a higher priority than Task 1. Therefore, normal RLL execution is interrupted by a cyclic RLL task.
- Task 8 is executed with a higher priority than Task 1 or Task 2. Therefore, both the normal RLL and the cyclic RLL are interrupted by a configured I/O interrupt.
- If you specify the cycle time T for a Task 2 task as a readable word, you can change the cycle time on a cycle-by-cycle basis. When T = 0, the default time of 10 ms is used.
- When the normal RLL task fails to complete executed within the specified cycle time, Bit 1 is set in STW219 and Bit 14 is set in STW1 on the next Task 1 scan. When the cyclic RLL task fails to completed execution within the specified cycle time, Bit 2 is set in STW219 on the next Task 2 scan. When a cyclic task overruns, the cycle on which the overrun is detected, is skipped.
- You can display the peak execution times for a task using the Data Window and specifying TPET1 or Task 1 or TPET2 for Task 2.
- You can call any subroutine from a task and the normal subroutine nesting rules apply. A given subroutine should be called from only one task. Subroutines are not re-entrant and subroutine execution initiated by a second task.

TASK	Parameter Type	Valid Parameter Types
Task Number	Constant Only	1,2,8
T	Word/Const	Form A. Const 0-32767

Special Instructions

Immediate I/O Read/Write (IORW)

The Immediate I/O Read/Write (IORW) instruction allows you to perform an immediate read or write to a discrete or word I/O module on the local base. For inputs, the data transfer is directly from the I/O modules into the image register. For outputs, the data transfer is directly from the image register to the I/O modules.

When the instruction has power, the IORW instruction is executed. The IORW supports these features:

- The data transfer takes place when the instruction is executed in RLL.
- For inputs (X and WX), the status of the specified number of points is copied from the I/O module to the image register.
- For outputs (Y and WY), the status of the specified number of points is copied from the image register to the I/O module.
- Output status follows input status, unless an error occurs.
- For inputs, when the module is not preset or does not match I/O configuration, the specified input points in the image register are cleared to zero and the output turns off.
- For outputs, when the module is not present or does not match I/O configuration, the status of the specified output points in the image register is not copied to the I/O module and the output turn off.

NOTE: *When the IORW copies Y values from the image register to a module, the current state of the Y points in the image register are written to the module. IF you want these Ys to be controlled by an MCR or a JMP, the MCR or JMP must be used to control the coils which write to the Ys. The IORW operation is not directly affected by MCRs and JMPs.*

IORW	Parameter Type	Valid Parameter Types
Reference Number	Constant	0-32767
ST	Bit Address	Starting address, X, Y, WX, WY
Points	Constant	Number of points to move, 1-64

NOTE: *The number of points to move must be a multiple of 8. All points must reside in the same I/O module.*

No Operation (NOP)

Date Compare (DCMP)

The Data Compare instruction (DCMP) compares the current date of the real-time clock with the values contains in the designated memory locations.

When power is passed to the DCMP instruction, the current date in the real-time clock is compared to the date stored in the memory location in the instruction. If the dates match, the output of the instruction is turned on.

DCMP	**Parameter Type**	**Valid Parameter Types**
Reference Number	Constant	0-32767
Date:	Word Address	V, W, G

The Data parameter uses 4 words to represent the actual date. These are defined as follows:

- Date = Year - CDB 0000-0099
- Date + 1 = Month - BCD 0001- 0012
- Date + 2 = Day of Month - BCD 0001-0031
- Date + 3 = Day of Week - BCD 0001-0007

Date Set (DSET)

The Date Set (DSET) instruction sets the date portion of the real-time clock to the values contained in the designated address.

When the input to the DSET instruction goes from off to on, the date portion of the real-time clock is set to the values in the memory locations designated by Date and the output is turned on.

	Parameter Type	**Valid Parameter Types**
Reference Number	Constant	0-32767
Date:	Word Address	V, W, G

The Data parameter uses 4 words to represent the actual date. These are defined as follows:

- Date = Year - CDB 0000-0099
- Date + 1 = Month - BCD 0001- 0012
- Date + 2 = Day of Month - BCD 0001-0031
- Date + 3 = Day of Week - BCD 0001-0007

Time Set (TSET)

The Time Set (TSET) instruction sets the time portion of the real time clock to the values contained in designated memory locations.

When the input to the TSET instruction transitions from off to on, the time portion of the real time clock is set to the values contained within the three consecutive V-memory locations designated by TM, and the output is turned on for one scan.

TSET	Parameter Type	Valid Parameter Types
Reference Number	Constant Only	Varies with configured One shot range
TM	Word Address	V,W,G

The TM parameter uses 3 words to represent the actual time. These are defined as follows:

- TM = Hour - BCD 0000-0023
- TM + 1 = Minute - BCD 0000- 0059
- TM + 2 = Second - BCD 0000-0059

Time Compare (TCMP)

The Time Compare instruction (TCMP) compares current time in the real time clock with values in the designated V- memory locations.

When power is passed to the input, the TCMP instruction is executed. It compares the current hours, minutes and seconds in the real time clock to the values in the designated memory location, TM. If a match occurs, the output of the instruction is turned on. If the time represented by the memory location is less than the real time value in the clock, the bit designated by LT is turned on. If the time represented by the memory locations is greater than the real time value in the clock, the bit designated by GT is turned on.

TCMP	Parameter Type	Valid Parameter Types
Reference Number	Constant	0-32767
TM:	Word Address	V,W,G
LT:	Bit Addr/Blank	Y,C,B, Blank
GT:	Bit Addr/Blank	Y,C,B, Blank

The TM parameter uses 3 words to represent the actual time. These are defined as follows:

- TM = Hour - BCD 0000-0023
- TM + 1 = Minute - BCD 0000- 0059
- TM + 2 = Second - BCD 0000-0059

Load Address (LDA)

The Load Address instruction (LDA) copies the logical address of a memory location into a specified memory location (a long or double word). The LDA instruction can be used to before a MOVE instruction, when the indirect addressing method is needed.

When power is passed to the LDA instruction, the LDA instruction is executed. The LDA instruction works as follows:

- The address of the memory location specified in A is copied to the destination specified by BB.
- The output is turned on after the instruction is executed (unless an error occurs).
- If the destination location is invalid, bits 6 and 11 in STW01 are set and the STW200 address contains a value of 5. The destination address is not changed.
- If the input is off, the instruction is not executed and power does not pass to the output.
- The source address should be a direct or indirect address - specify any readable word. For a direct address, LDA copies the logical address for this word into the destination. For an indirect address, precede the address with the @ character. The long word at this indirect address must contain another address, and the LDA copies this second logical address into the destination.

Warning	**The address that is copied to the destination is the logical address and NOT the physical address. To avoid unpredictable machine operation, do not use this address as a pointer within an external subroutine.**

LDA	**Parameter Type**	**Valid Parameter Types**
Reference Number	Constant only	0-32767
A:	Word Address	Source address, Form A
AI:	Blank/Word	Source Index, Form A, Blank, -32768 - 32767
BB:	Word Address	Destination address: Form B (direct addr), Form A (indirect addr)
BI:	Blank/Word	Destination Index, Form A, Blank, -32768 - 32767

You can specify the field AI as an index into the source address when you want to copy an address that is in a table. AI should designate the relative word in the table referenced by A. The element A[0] is the first element in the table. The following is a list of valid types for AI:

- Enter a constant 0-65535. If you leave the field blank or enter a 0, then no indexing will be done.
- Enter any readable word (see table above). The contents of this word should be an unsigned integer (0-65535) that gives the value of the index.

If an indirect source address is indexed, the controller first resolves the address, then it indexes it.

You can specify the field B1 as an index into the destination address when you want to copy an address into a word in a table. BI should designate the relative work in a table referenced by BB. Element BB[0] is the first element in the table. The following is a list of valid types for BI:

- Enter a constant 0-65535. If you leave the field blank or enter a 0, then no indexing will be done.
- Enter any readable word (see table above). The contents of this word should be an unsigned integer (0-65535) that gives the value of the index.

If an indirect destination address is indexed, the controller first resolves the address, then it indexes it.

Load Constant (LDC)

The Load Data Constant instruction (LDC) loads a positive integer constant into the designated memory location.

When the input receives power, the LDC instruction is executed. The data constant designated by N is loaded into the memory location specified by A. When the function is executed, the output is turned on.

LDC	Parameter Type	Valid Parameter Types
Reference Number:	Constant	0-32767
A:	Word Address	Form B
N:	Constant Only	0-32767

Lock Memory (LOCK)

The LOCK instruction works with the UNLCK instruction to provide a means in which multiple application in the TI575 coordinate access to shared resources, generally G-memory data blocks.

The LOCK can be either Exclusive or Shared. An Exclusive lock signals other application programs that the resource is unavailable for reading or writing. A shared lock signals other application programs that the resource locations are available for reading only.

The LOCK instruction does not specify the G-memory locations that are protected, nor does the LOCK actually prevent an application from reading or writing to these memory locations. You should write your programs so that the G-memory locations are protected when you gain control of a LOCK. When you program an exclusive lock, no other application program can acquire control of the lock. When you program a shared lock, more than one application program can acquire control of the lock. Use these capabilities in programs that update the shared resource protected by the lock.

In order to use a lock properly, follow these steps:

- AA and (AA+1) must be initialized to 0 prior to the first time.
- When the input is on, the application attempts to acquire the lock. If the lock is not available, the application continues to attempt acquisition of the lock until the lock is acquired or the specified time-out has expired. A value of 0 for T results in a single attempt to obtain the lock. A value of 3276.7 indicates that the application should try until it obtains the lock or the scan watchdog fatal error occurs.
- If an application obtains the lock before the time-out expires, the output is turned on and the scan continues.
- If the time-out expires before the application obtains the lock, the output is turned off and the scan continues.
- When an application program attempts to acquire control of the lock, the value in AA and AA+1 is examined. If this value indicates that the lock is free, control of the lock passes to the inquiring application program, the output is turned on, and RLL execution continues at the next network.
- When an application program obtains control of the lock, the LOCK instruction increments the value of a lock/unlock counter. The UNLCK instruction decrements the lock/unlock count when an application program relinquished control of a lock. If the counter is not equal to zero at the end of the RLL scan, Bit 6 in STW01 is set to 1 and a value of 3 is written to STW200.

LOCK	**Parameter Type**	**Valid Parameter Types**
Reference Number:	Constant	0-32767
T	Constant Only	time, in ms, 0-3276.7
AA	Word Address	V,G (W, doesn't work) Must be 2 words

Unlock Memory (UNLCK)

The UNLCK instruction works with the LOCK instruction to provide a means in which multiple application in the TI575 coordinate access to shared resources, generally G-memory data blocks.

UNLCK	**Parameter Type**	**Valid Parameter Types**
Reference Number:	Constant	0-32767
AA	Word Address	V,G (W, doesn't work) Must be 2 words

Force Role Swap (FRS)

The Force Role Swap (FRS) instruction allows you to design a program to switch the active controller with the standby controller in hot backup configurations.

The role swap may be the result of programmed diagnostic processors that detect a switch over convert. You can also use the instruction to allow routine maintenance processors. The role swap may be initiated by having a switch close in the I/O or by using a timer to trigger the swap.

The FRS instruction depends upon the current power flow, the power flow on the previous scan and the current state of the controller. This instruction will only work in online only. If no standby controller is preset, the active controller interprets this instruction as a NOP.

If the active controller with a standby detects an off to on transition on the input, it queues a role swap to occur at the beginning of the next scan. Upon completing the swap, both controllers write the instruction reference number in the assigned memory address. This value can be used to indicate why the role swap occurred.

On each scan, the FRS address is compared to the specified memory location. The output turns on independent of its input whenever the memory location contents match the instruction reference number.

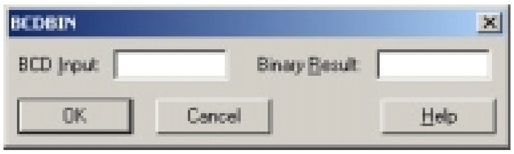
FRS	**Parameter Type**	**Valid Parameter Types**
Instruction Reference	Constant	Depends on configured One-shot memory configuration
ST	Word Address	V

12 - Special Function Instructions

Defining Special Function Programs

Convert BCD To Binary

The Convert BCD to Binary statement converts binary coded decimal to (BCD) inputs to a binary representation of the equivalent integer.



- BCD Input is the memory location of the BCD word to be converted. Valid Descriptor is Integer Address (address containing an integer value (e.g., V100 or WX77 or V100(2)))
- Binary Result is the memory location of the integer value after conversion. Valid Descriptors are Integer Address (address containing an integer value (e.g., V100 or WX77 or V100(2))), and Writeable Address (e.g., WY1000 or V23. C55)

BCDBIN Operation

The operation of BCDBIN is described below and illustrated in Figure 9.1

Each time the BCDBIN statement executes, the four digits of the BCD value located in the address specified by A are converted to the binary representation of the equivalent integer value.

The result is stored in the address specified by B.

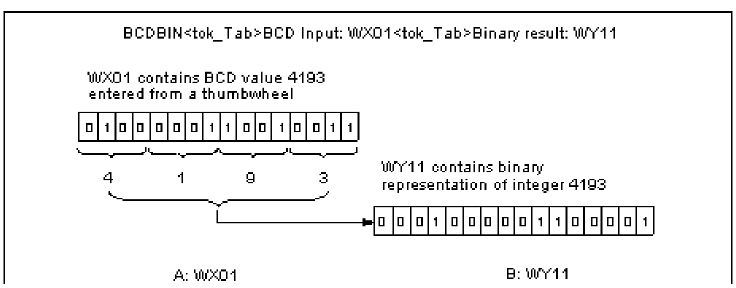
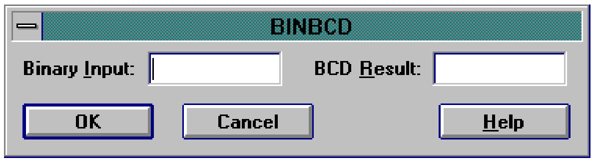


Figure 9.1 Example of BCDBIN Operation

Convert Binary Inputs To BCD

The Convert Binary Inputs to BCD statement converts the binary representation of an integer to the equivalent Binary Coded Decimal (BCD) value. Values up to 9999 are converted to equivalent BCD values.



- Binary Input is the memory location of the BCD word to be converted. Valid Descriptor is Integer Address (address containing an integer value (e.g., V100 or WX77 or V100(2)))
- BCD Result is the memory location of the integer value after conversion. Valid Descriptors are Integer Address (address containing an integer value (e.g., V100 or WX77 or V100(2))), and Writeable Address (e.g., WY1000 or V23. C55)

BINBCD Operation

The operation of BINBCD is described below and illustrated in Figure 9.2

- Each time the BINBCD statement executes, the integer located in the address specified by A is converted to BCD.
- An error occurs if the input value contained in A is less than zero or greater than 9999.
- The BCD value is stored in the address specified by B.

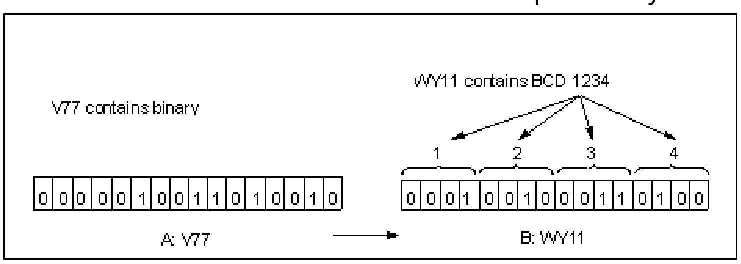
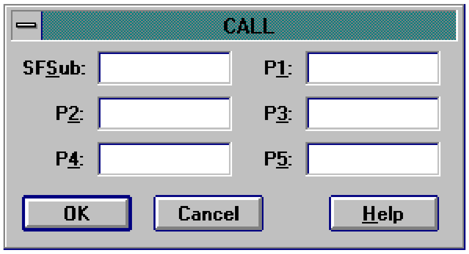


Figure 9.2 Example of BINBCD Operation

Call Subroutine

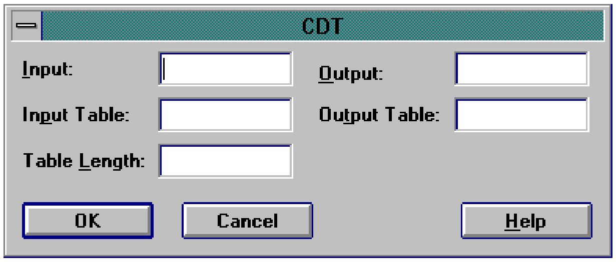
The call statement calls an Special Function subroutine for execution. Up to five parameters may be passed to the subroutine by the Call statement.



- SFSub is the number of the SF subroutine to be called and ranges from 1 to 1023. Valid Descriptor is Literal Integer constant.
- P1 is a value or constant to be passed between the SF subroutine that is called, and SF program or the SF subroutine that contains the Call statement. Valid Descriptors are Integer Address or Value (address containing an integer value (e.g., 3642,V100 or WX77 or V100(2))); Real Address or Value (address containing a real value (e.g., 33.642, or V120. or WY55 or V100.(2))); Optional the field can be left blank.
- P2 is a value or constant to be passed between the SF subroutine that is called, and SF program or the SF subroutine that contains the Call statement. Valid Descriptors are Integer Address or Value (address containing an integer value (e.g., 3642,V100 or WX77 or V100(2))); Real Address or Value (address containing a real value (e.g., 33.642, or V120. or WY55 or V100.(2))); Optional the field can be left blank.
- P3 is a value or constant to be passed between the SF subroutine that is called, and SF program or the SF subroutine that contains the Call statement. Valid Descriptors are Integer Address or Value (address containing an integer value (e.g., 3642,V100 or WX77 or V100(2))); Real Address or Value (address containing a real value (e.g., 33.642, or V120. or WY55 or V100.(2))); Optional the field can be left blank.
- P4 is a value or constant to be passed between the SF subroutine that is called, and SF program or the SF subroutine that contains the Call statement. Valid Descriptors are Integer Address or Value (address containing an integer value (e.g., 3642,V100 or WX77 or V100(2))); Real Address or Value (address containing a real value (e.g., 33.642, or V120. or WY55 or V100.(2))); Optional the field can be left blank.
- P5 is a value or constant to be passed between the SF subroutine that is called, and SF program or the SF subroutine that contains the Call statement. Valid Descriptors are Integer Address or Value (address containing an integer value (e.g., 3642,V100 or WX77 or V100(2))); Real Address or Value (address containing a real value (e.g., 33.642, or V120. or WY55 or V100.(2))); Optional the field can be left blank.

Correlated Data Table

The Correlated Data Table statement compares an input value(the input) to a table of values (the input table), and locates a value in the input table that is greater than or equal to the input. The CDT then writes the value located in a second table (the output table), that is correlated with the value located in the input table, to an output address (the output). The CDT format is shown in Figure 11.10.



- Input - is the input address.
- Output - is the address to which the output value is written.
- Input Table - is the starting address for the input table.
- Output Table -is the starting address for the output table.
- Table Length - is the length of each table and must be a value greater than zero.

When CDT is executed, the CDT compares the value of an input

- Element specified in INPUT to a pre-exist in table of values having a starting address specified in Input Table . The first value in the input table that is greater than or equal to the input is located.
- A value in a second pre-existing table (starting address specified in Output Table) that correlates with the selected value in the input table is written to an output address specified in Output.
- The input table must be in ascending order. That is, the lowest value is located in the lowest memory location and the highest value is located in the highest memory location.
- Table length Table Length depends upon the memory location that you choose, and how much memory you allocated if the memory is user configurable.
- Both tables must have the same number of entries.

The input address V1 contains the value 40. The value in the input table that is greater than or equal to 40 is 43, contained in k68. The correlated value in the output table is in k88. The value written to the output address v2 is 72.

Exit on Error

The EXIT statement allows you to terminate a SF program or SF subroutine and have an error code logged. The EXIT format is shown in Figure 11.12.

- **Errcode** - contains the value of the error code and can range from 0 to 255. Errcode = Integer, Optonal Literal Constant

Operation of the EXIT

The operation of the EXIT statement is described below.

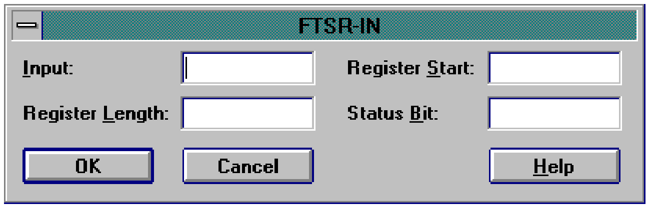
- When the SF program encounters the EXIT statement, program execution terminates. If an SF subroutine encounters the EXIT statement, control returns to the statement in the SF program following the SF subroutine call.
- If you use the EXIT statement in conjunction with an IF statement, you can terminate the program under specific conditions.

You can leave A blank and the current error code is written to the ERROR STATUS ADDRESS that you specify in the SF program header. IF this address is a discrete point, it turns on.

You can define an error condition and assign it an error code 200-255 (codes 0-199 are reserved). When the EXIT statement executes, the program terminates and this error code is written to the ERROR STATUS ADDRESS. If this address is a discrete point, it turns on.

Fall Through Shift Register - Input

The Fall Through Shift Register Input statement operates an asynchronous shift register. The shift register is essentially a table of 16-bit words. The FTSR-IN moves a word into the shift register each time the statement executes. The FTSR-IN is used in conjunction with the Fall Through Shift Register Output statement (FTSR-OUT) tat moves words out of the shift register. The FTSR-IN format is shown in Figure 11.13.



- **Input** - is the input address from which the words are moved. **Input = Integer Address**
- **Register Start** - is the starting address for the shift register. Four words (**Register Start** through **Register Start** + 3) are automatically reserved for the operation of the statement and make up the header of the shift register. The first word of your data is shifts into address **Register Start** + 4. **Register Start = Integer, Writeable Address**

NOTE: *Do not write data to the header fields. The shift register does not operate correctly if any of these fields is modified by an external action. These fields may be redefined in future software releases.*

- Register Length - is the length of the table. If a constant is used, in must be greater than zero. The total length of the shift register is Register Length + header. Register Length = Integer Address *or value*
- Status Bit - is the status bit and can be C or Y. The bit specified by Status Bit turns on when the register is full. The bit (Status Bit + 1) is automatically reserved as a second status bit. The bit specified by (Status Bit +1) turns on when the register is empty. Status Bit = Bit, Writeable Element

Operation of the FTSR-IN

The operation of the FTSR-IN statement is described below:

- FTSR-IN is used in conjunction with an FTSR-OUT; you must use the same corresponding values for register start, register length, and status bit in the two FTSR statements.
- **Input** is the input address from which the words are moved into the shift register.
- The starting address **Register Start** determines the memory area in which the shift register is located. The first word of your data shifts into address **Register Start** + 4.
- The four words (**Register Start** through **Register Start** + 3) are automatically reserved for the operation of the shift register.

  (**Register Start**) contains the Count, which equals the current number of entries in the shift register.

  (**Register Start** + 1) contains the Index, which acts like a pointer to indicate the next available location of the shift register into which a word can be shifted. When Index equals one, the next available location is (**Register Start** +5), and so on.

  (**Register Start** + 2) contains the Length, which equals the maximum size of the shift register in words.

  (**Register Start** +3) contains the Checkword. The checkword is used internally to indicate whether the FTSR is initialized.

- The register length **Register Length** determines the size of the shift register. The register length depends upon the memory location that you choose and how much memory you have allocated (if the memory is user-configurable).
- The status bit specified by **Status Bit** is turned on to indicate that the register is full. The bit (**Status Bit** + 1) is automatically reserved as a second status bit and turns on whenever the shift register is empty.

  Use the same status bits for FTSR-IN that you use for the FTSR-OUT. FTSR-IN sets (**Status Bit** +1) when the register is empty. FTSR-IN clears this bit.

- If the shift register is empty, status bit **Status Bit** is off and (**Status Bit** + 1) is on.
- When the FTSR-IN executes, the following actions occur.

  The word currently in memory location A is shifted into the location specified by the Index.

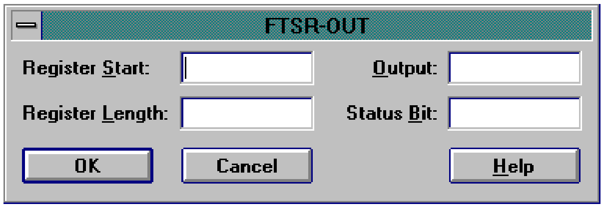
  The Count and the index are each incremented by one.

  Status bit (**Status Bit** + 1) turns off.

- Each time the FTSR-IN executes, another words moves into the next available location; the Index and the Count increment by one. When the Index equals the length, it resets to zero after the next execution by the FTSR-IN.
- When the shift register is full, another word cannot be shifted in until one is shifted out by the FTSR-OUT statement.
- When the shift register is full, status bit **Status Bit** turns on. If you attempt to shift in another word, an error generates. (error 87).
- You can use FTSR-OUT to remove words from the shift register before all locations are full. You can use FTSR-IN to shift more words into the shift register before all words are removed.

Fall Through Shift Register - Output

The Fall Through Shift Register Output statement operates an asynchronous shift register. The shift register is essentially a table of 16-bit words. The FTSR-OUT moves data out of the shift register each time the statement executes. The FTSR-OUT is used in conjunction with the Fall Through Shift Register Input statement (FTSR-IN) that moves words into the shift register. Figure 11.15 shows the FTSR-OUT format.



- **Register Start** - is the starting address for the shift register. The four words (**Register Start** through **Register Start** + 3) are automatically reserved for the operation of the statement and make up the header of the shift register. **Register Start = Integer, Writeable Address**

NOTE: *Do not write data to the header fields. The shift register does not operate correctly if any of these fields is modified by an external action. These fields may be redefined in future software releases.*

- **Output** - is the output address to which the words are moved. **Output = Integer, Writeable Address**
- **Register Length** - is the length of the table. If a constant is used, it must be greater than 0. **Register Length = Integer Address *or value***
- **Status Bit** is the status bit and can be C or Y. The bit specified by **Status Bit** is turned on when the register is full. The bit (**Status Bit**+ 1) is automatically reserved as a second status bit. The Bit specified by (**Status Bit** + 1) is automatically reserved as a second status bit. The bit specified by (**Status Bit** + 1) is turned on when the register is empty. **Status Bit = Bit, Writeable Element**

Operation of the FTSR-OUT

- FTSR-OUT is used in conjunction with a FTSR-IN; you must use the same corresponding values for register start, register length, and status bit in the two FTSR statements.
- Starting address **Register Start** determines the memory area in which the shift register is located. The first word of user data is located in address **Register Start** + 4.
- The four words (**Register Start** through **Register Start** + 3) are automatically reserved for the operation of the shift register.

  (**Register Start**) contains the Count, which equals the current number of entries in the shift register.

  (**Register Start** + 1) contains the Index, which acts like a pointer to indicate the next available location of the shift register into which a word can be shifted. When the Index equals zero, the next available location is (**Register Start** + 4); when the Index equals one, the next available location is (**Register Start** +5), and so on.

  (**Register Start** + 2) contains the length, which equals the maximum size of the shift register in words.

  (**Register Start** +3) contains the Checkword. The checkword is used internally to indicate whether the FTSR has been initialized.
- **Output** is the output address into which the words are moved.

- The register length **Register Length** determines the size of the shift register. The register length depends upon the memory location that you choose and how much memory you allocated (if the memory is user configurable).
- **Status Bit** is the status bit and can be C or Y. The bit specified by **Status Bit** turns on to indicate that the register is full. The bit (**Status Bit** + 1) is automatically reserved as a second status bit and turns on whenever the shift register is empty.

  Use the same status bits for the FTSR-OUT that you use for the FTSR-IN. FTSR-IN sets **Status Bit** when the register is full. FTSR-OUT clears this bit as the function executes. FTSR-OUT sets (**Status Bit** + 1) when the register is empty. FTSR-IN clears this bit.
- If the shift register contains one or more words, the Count equals the number of current entries. The Index points to the next available location of the shift register into which a word can be moved. Status bit (**Status Bit** + 1) is off. Status bit **Status Bit** is on if the shift register is full.
- When the FTSR-OUT executes, the following actions occur.

  The oldest word in the shift register shifts into memory location **Output**.

  The Count decrements by one.

  The Index is unchanged and continues to point to the next available location into which a word can be moved.
- Each time the FTSR-OUT executes, another word moves out of the shift register and the Count is decremented by one. The Index remains unchanged.
- After the shift register is empty, the Index and Count contain zero.
- Status bit D turns off and status bit (**Status Bit** +1) turns on. If you attempt to shift a word out of an empty shift register, an error is generated (error 86).
- You can use FTSR-OUT to remove words from the shift register before all locations are full. You can use FTSR-IN to shift more words into the shift register before all words are removed.

Goto Function

The GOTO statement continues program execution at a specified LABEL statement. The GOTO and the LABEL statements are always used together. The format of the two statements is shown in Figure 11.17.

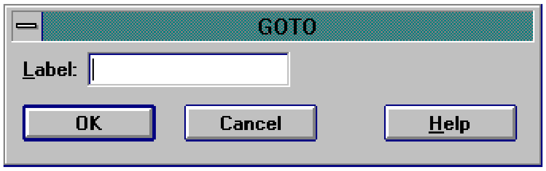
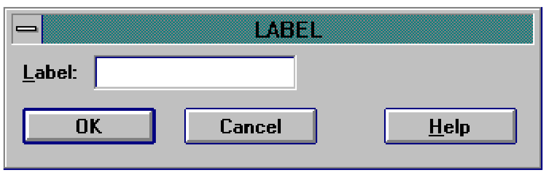


Figure 11.17

The <SF statement> may be any of the SF program statements.

Label is the label and can range from 0 to 65535. **Label = Integer Literal constant**

When the SF program encounters the GOTO, program execution continues at the LABEL specified by **Label**.



IF Function

The IF or IIF (Integer IF) statement is used for the conditional execution of statements and operates in conjunction with the ELSE and the ENDIF statements. When an IF statement is used, a THEN result is understood. The IF format is shown in Figure 11.19.

NOTE: *Integer IF operations are available only in CPUs that support PowerMath.*

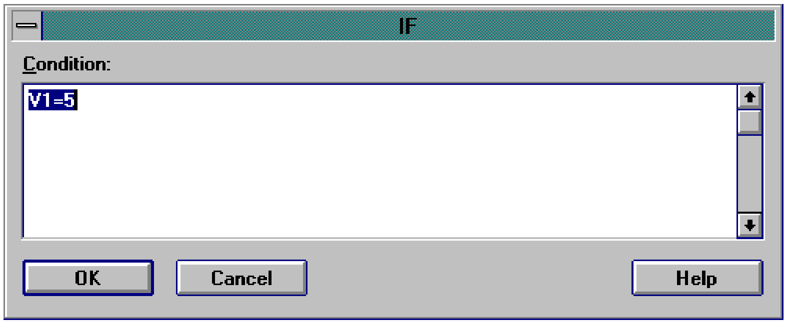


Figure 11.19 IF Format

The <Sf statement> may be any of the SF program statements.

Each time the IF executes, the condition defined within the statement is tested.

If the <expression> is true (non-zero), statements in the THEN section execute; any statements in the ELSE section are skipped.

If the <expression> is false (zero), statements in the THEN section are skipped; any statements in the ELSE section execute.

The <expression> can be any MATH expression. See Table 11.5 for a list of the MATH functions. The use of the assignment operator (:=) in an expression is optional.

The If statement operates in conjunction with the ENDIF statement and an optional ELSE statement.

The ENDIF indicates the end of an IF-THEN ELSE structure.

If there is no ELSE statement, the statements between the IF and the ENDIF are treated as THEN statements.

If an ELSE statement is used, then any statements between IF and ELSE constitute by default a THEN section. An ELSE statement indicates the end of the THEN section and the beginning of the THEN section and the beginning of the ELSE section in an IF-THEN-ELSE structure.

Statements between ESLE and ENDIF constitute the ELSE section in the IF statement.

IF, ELSE and ENDIF statements may be nested to any level.

ELSE Function

There is no data needed in this field. When the ELSE is used between an IF and ENDIF statement any statement between IF and Else constitute by default a THEN section. An ELSE statement indicates the end of the THEN section and the beginning of the ELSE section in an IF-THEN-ELSE structure.-

ENDIF Function

There is no data needed in this field. The ENDIF indicates the end of an IF-THEN-ELSE structure.

Integer Math Operation

The Integer Math statement executes integer arithmetic computations. The IMATH format, based on the functions in Table 11.5, is shown in Figure 11.22.

NOTE: *Non-PowerMath CPUs do not support the following operators:= <> < <= > >=, AND, OR, and the ABS intrinsic function.*

NOT	Unary Not-The expression "NOT X" returns the one's complement of X.
>>	Shift right (arithmetic)
<<	Shift left (arithmetic)
*	Multiplication
/	Integer division-Any remainder left over after the division is truncated.
MOD	Modulo arithmetic-The expression "X mod Y" returns the remainder of X after division by Y.
+	Addition
-	Subtraction/unary minus (negation)
=	Equal. The expression X = Y returns 1 if X equals Y, and zero if not.[2]
<>	Not equal. The expression X <> Y returns 1 if X is not equal to Y, and zero if so.[2]
<	Less than. The expression X < Y returns 1 if X is less than Y, and zero otherwise.[2]
<=	Less Than or Equal. The expression X <= Y returns 1 if X is less than or equal to Y, and zero otherwise.[2]
>	Greater Than. The expression X > Y returns 1 if X is greater than Y, and zero otherwise.[2]
>=	Greater Than or Equal. The expression X >= Y returns 1 if X is greater than or equal to Y, and zero otherwise.[2]
AND	Logical AND. The expression X AND Y returns 1 if both X and Y are non-zero, and zero otherwise.[2]
OR	Logical OR. The expression X OR Y returns 1 if either X or Y is non-zero, and zero otherwise.[2]
&	Bit-by-bit AND of two words
|	Bit-by-bit OR of two words
^	Bit-by-bit exclusive OR of two words[2]
:=	Assignment[2]
ABS	Math intrinsic function Absolute Value[2]

2 Supported by PowerMath CPUs only.

Table 11.5 IMATH Operators

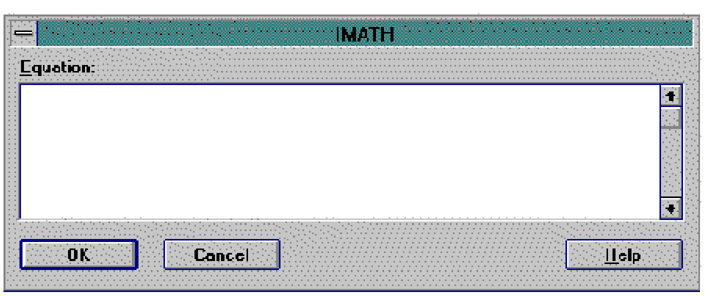


Figure 11.22

Figure 11.22 illustrates the operation of the IMATH statement described below.

1. Each time the IMATH statement executes, the calculations within the statement are made.
2. The IMATH computations are executed using the rules of precedence for arithmetic operations listed in Table 11.6.

Functions within a group are equivalent in precedence. Execution takes

place from left to right. For example, in the operation (X * Y / Z), X is multiplied by Y, and the result is divided by Z.

A subexpression enclosed in parentheses is evaluated before surrounding operators are applied, e.g., in (X+Y) * Z, the sum of X + Y is multiplied by Z.

- Parentheses, constants, and subscript variables are allowed in the expressions.
- You can use only integers in an IMATH statement. Mixed mode operation (integer and real numbers) is not supported.
- Denote a binary number by the prefix OB (e.g.0B10111), a hexadecimal number by the prefix 0H (e.g. 0H7FFF) .
- The programming device checks to see if a statement is valid as you enter the statement and reports an error by placing the cursor in the field where the error occurs.

Table 11-6 Order of Precedence for IMATH Operators

Highest Precedence	Intrinsic function ABS [1] , NOT, Negation NOT –
	Multiplication, Division, MOD * / MOD,
	Addition, Subtraction + –
	Shift left, Shift right << >>
	Relational Operators (= < < = >= <>)[1]
	&, Logical AND [1]
	|, ^, Logical OR [1]
Lowest Precedence	Assignment (:=) :=

[1] Supported by PowerMath CPUs only.

Lead/Lag Operation

The LEAD/LAG statement (Figure 11.23) allows filtering to be done on an analog variable. This procedure calculates an output based on an input and the specified gain, lead, and lag values. The LEAD/LAG statement can only be used with cyclic processes, such as loops analog alarms, and cyclic SF programs.

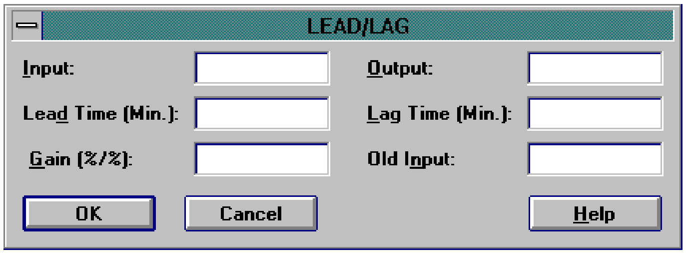


Figure 11.23

Input specifies the location of the input value of the current sample period that is to be processed.

Output specifies the location of the output variable, the result of the LEAD/LAG operation.

Lead Time specifies the lead time in minutes.

Lag Time specifies the lag time in minutes.

Gain specifies the ratio of the change in output to the change in input at a steady state, as shown in the following equation. The constant must be greater than zero.

Old Input specifies the memory location of the input value from the previous sample period.

For sample time, LEAD/LAG algorithm uses the sample time of the loop, analog alarm, or cyclic SF program from which it is called.

The first time it executes, LEAD/LAG is initialized equals input.

The LEAD/LAG algorithm uses the following equation.

where Yn = present output, Yn - 1= previous output,

X n = present input, X n - 1= previous input

Ts = sample time in minutes.

The output depends on the ratio of lead to lag as explained below. Assume the following values in each example:

input and gain = 1.0

Real/Integer Math Operation

The MATH statement executes arithmetic computations involving both integers and real numbers. The MATH format, based on the operators in Table 11.5, is shown in Figure 11.25.

Parentheses, constants, subscript variables, and a set of intrinsic functions (listed in Table 11.8) are allowed in the expressions. Assignment operator (: =) is required.

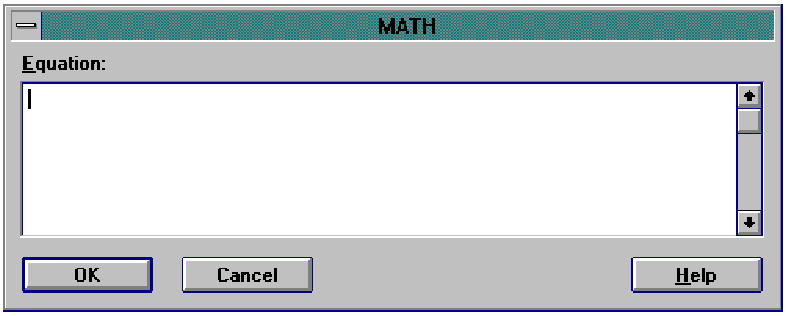


Figure 11.25

Operation of MATH

The operation of MATH is described below and illustrated in Figure 11.25.

Each time the MATH statement is executed, the calculations within the statement are made.

The MATH computations are executed using the rules of precedence for arithmetic operations listed in Table 11.9. Functions within a group are equivalent in precedence. Execution takes place from left to right for all operators except exponentiation.

Pack Data

The Pack Data statement moves discrete and/or word data to or from a table. You can access the image register directly by using the PACK statement. PACK is primarily intended for use in consolidating data so that it can be efficiently transmitted to a host computer. The PACK format is shown in Figure 7–26.

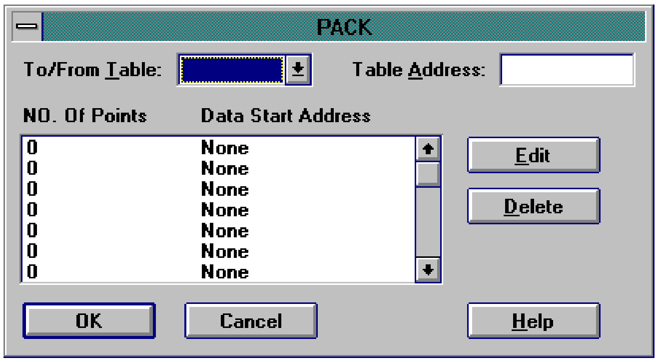


Figure 11.26 PACK Format

- Q (To/From Table) specifies whether you are writing data to or from the table.
- B (Table Address) specifies the address of the table, to or from which data are written or read.
- C (Number Of Points) is an integer number that specifies how many points or words are to be moved. To enter Number Of Points you must highlight one of the points in the dialog box and click on Edit.
- For a TO table, D (Data Start Addr) specifies the starting address of the points or words that are to be written to the table.
  For a FROM table, D specifies the starting address in memory into which data is to be read from the table.
  D + (C-1) must be within configured memory range. To enter Data Start Addr. you must highlight one of the points in the dialog box and click on Edit.
- Fields C and D can be repeated for up to 20 writes/reads to and from the table (Figure 7–27).

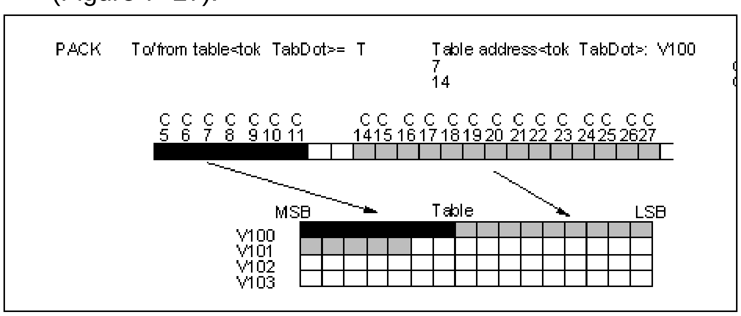


Figure 11.27 Example of PACKing Multiple Blocks of Bits Into Table

PACK TO Operation

The operation of the PACK TO statement is described below.

- For a TO Table, data are written into a table. This write operation begins with the data starting at the first Data Start Address and writes the specified number of points or words into the table, beginning with the first word of the table.

Bits are written sequentially as illustrated in Figure 7–28 below.

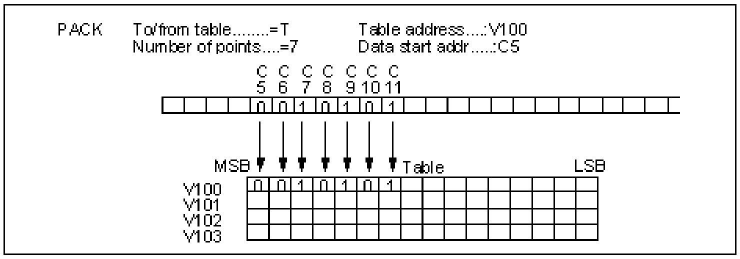


Figure 11.28 Example of PACKing Bits Into Table

You can specify multiple blocks of data to be written into the table. When the first word of the table is full, PACK begins to fill the second word.

Words are written sequentially into the table, as illustrated in Figure 7–29. You can also PACK multiple blocks of words.

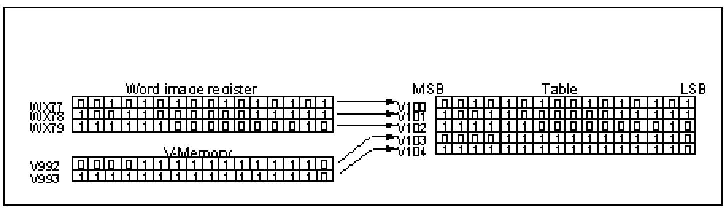


Figure 11.29 Example of PACKing Words Into Table

- You can PACK blocks of words and blocks of bits into a table with one PACK statement. See Figure 7–30. The data are PACKed according to these rules.

Discrete points are PACKed into the next available bit in the table.

Words are PACKed into the next available word in the table. Unused bits in the previous word fill with zeros when a word is written to the table.

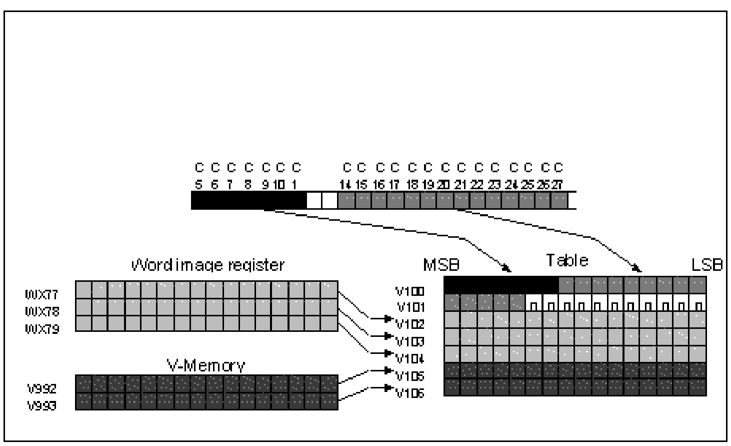


Figure 11.30 Example of PACKing Bits and Words Into Table

PACK FROM Operation

The operation of the PACK FROM statement is described below.

For a FROM Table, data are read from a table. This read operation begins with the table starting address and reads the specified number of points or words from the table. PACK then writes this data, starting with the address designated in the Data Start Address.

Bits are written sequentially as illustrated in Figure 7–31.

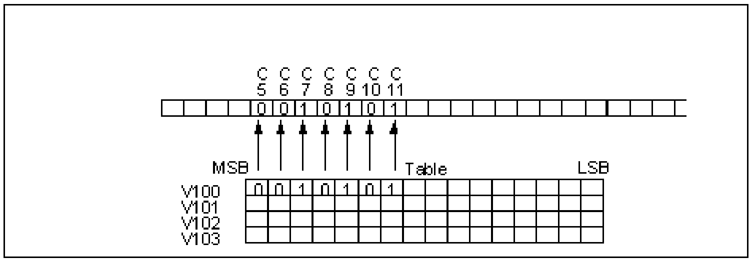


Figure 11.31 Example of PACKing Bits from a Table

You can specify multiple blocks of data to be PACKed from the table, as illustrated in Figure 7–32. You cannot skip sections of the table to PACK data located within the table. For example, refer to Figure 7–32. If the data that you want to read are located in the least significant nine bits of V100 and the most significant five bits of V101, you must still PACK out the first seven bits of V100 and discard them.

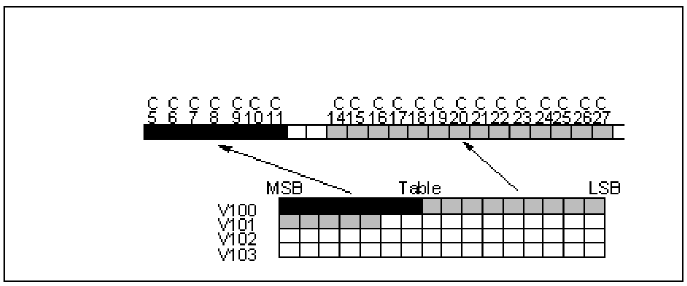


Figure 11.32 Ex.of PACKing Multiple Blocks of Bits from a Table

Words are read sequentially from the table, as illustrated in Figure 11.33. You can also PACK multiple blocks of words.

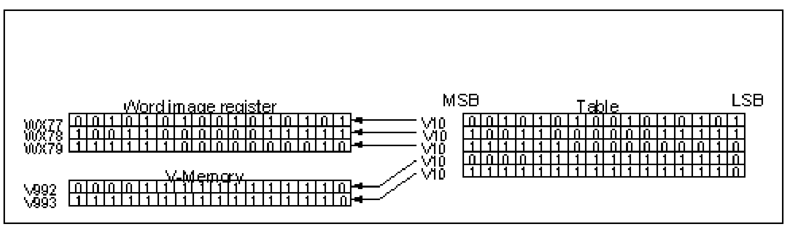


Figure 11.33 Example of PACKing Words from a Table

You can PACK blocks of words and blocks of bits from a table with one PACK statement. See Figure 7–34. The data are packed according to these rules.

All discrete points designated in the Number of Points field are packed from the table.

Words are packed from the first available word in the table. That is, unused bits in the previous word of the table are not included as part of a word that is PACKed from the table.

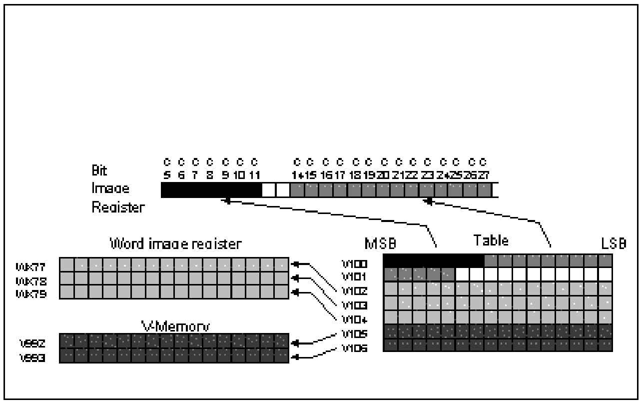


Figure 11.34 Example of PACKing Bits and Words from a Table

Pack Analog Alarm Data

The Pack Analog Alarm Data statement moves analog alarm data to or from a table. PACKAA is primarily intended for use in consolidating analog alarm data to be accessed from an operator interface. The PACKAA format is shown in Figure 7.35.

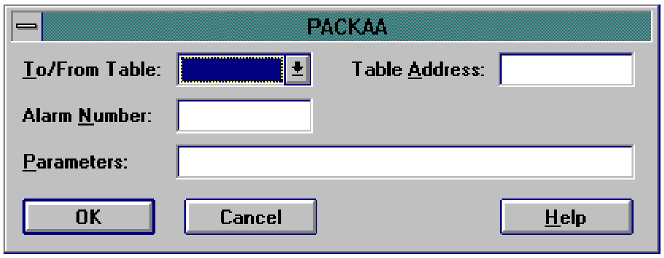


Figure 11.35 PACKAA Format

- A specifies whether you are writing data to or from the table.
- B specifies the address of the table, to or from which data are moved.
- C specifies the number of the analog alarm to be accessed. C may range from 1 to the maximum number of alarms.
- D specifies the analog alarm variables. Up to eight variables can be designated. See Table 7–10 for a list of the analog alarm variables.

Mnemonic	**Variable Name**	**Mnemonic**	**Variable Name**
AACK	Acknowledge	APV*	Process Variable
AADB*	Deadband	APVH.	Process Variable High Limit
ACF	C-Flags (32 bits)	APVL.	Process Variable Low Limit
ACFH	Most Significant Word of C-Flags	ARCA.	Rate of Change Alarm Limit
ACFL	Least Significant Word of C-Flags	ASP*	Set Point
AERR*	Error	ASPH*	Set Point High Limit
AHA*	High Alarm Limit	ASPL*	Set Point Low Limit
AHHA*	High-High Alarm Limit	ATS.	Sample Rate
ALA*	Low Alarm Limit	AVF	V Flags
ALLA*	Low-Low Alarm Limit	AYDA*	Yellow Deviation Alarm Limit
AODA*	Orange Deviation Alarm Limit		

*Variables with an asterisk can be either a real number or an integer. Variables followed by a period are real numbers. Variables not followed by a period are integers. When you execute PACKAA using real numbers, two memory locations are allocated for each real number.

PACKAA Operation

The operation of the PACKAA statement is described below and illustrated in Figure 7–36 and Figure 7–37. When the PACKAA statement executes, the following actions occur.

- For a TO Table, the value of the analog alarm variable specified in D is written into the table at the address designated by B.

  If additional variables are specified, the second variable is written to (B + 1), the third to (B + 2), and so on up to eight variables.

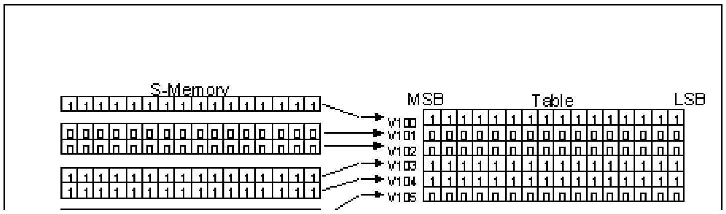


Figure 11.36 Example of PACKAA TO Table Operation

- For a FROM Table, PACKAA writes the word in the table starting address B into the specified analog alarm variable.

  If additional variables are specified, the second word in the table is written to the second variable, and so on up to eight variables.

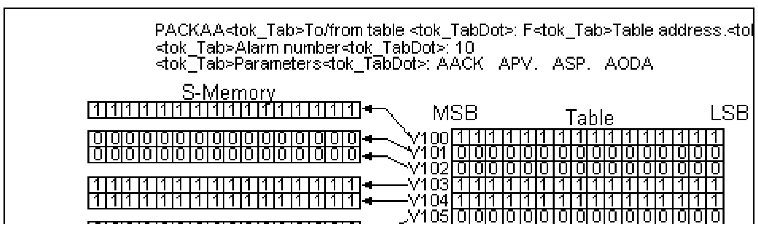


Figure 11.37 Example of PACKAA FROM Table Operation

Pack Loop Data

The PACKLOOP statement (Figure 11.38) moves loop data to or from a table. PACKLOOP is primarily intended for use in consolidating loop data to be accessed from an operator interface.

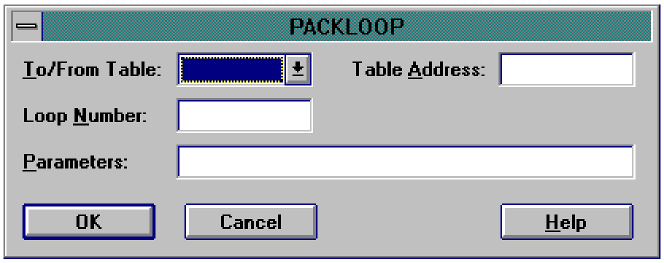


Figure 11.38 PACKLOOP Format

- A specifies whether you are writing data to or from the table.
- B specifies the address of the table, to or from which data are moved.
- C specifies the number of the loop to be accessed. The range for C is from 1 to the maximum number of loops.
- D specifies the loop variables. Up to eight variables can be designated. See Table 7–11 for a list of the loop variables.

PACKLOOP Operation

The operation of the PACKLOOP statement is described below. PACKLOOP operates similarly to the PACKAA statement. See Figure 7–36 and Figure 7–37 for an example of how the PACKLOOP statement executes.

When the PACKLOOP statement executes the following actions occur.

- For a TO Table, the value of the loop variable specified in D is written into the table at the address designated by B.

  If additional variables are specified, the second variable is written to (B + 1), the third to (B + 2), and so on up to eight variables.
- For a FROM Table, PACKLOOP writes the word in the table starting address B into the specified loop variable.

If additional variMables are specified, the second word in the table is written to the second variable, and so on up to eight variables.

Mnemonic	**Variable Name**
LACK	Alarm Acknowledge
LADB*	Alarm Deadband
LCF	C-Flags (32 bits)
LCFH	Most Significant Word of C-Flags
LCFL	Least Significant Word of C-Flags
LERR*	Error
LHA*	High Alarm Limit
LHHA*	High-high Alarm Limit
LKC.	Gain
LKD.	Derivative Gain Limiting Coefficient
LLA*	Low Alarm Limit

LLLA*	Low-low Alarm Limit
LMN*	Output
LMX*	Bias
LODA*	Orange Deviation Alarm Limit
LPV*	Process Variable
LPVH.	Process Variable High Limit
LPVL.	Process Variable Low Limit
LRCA.	Rate of Change Alarm Limit
LRSF	Ramp/Soak Flags
LRSN	Ramp/Soak Step Number
LSP*	Set Point
LSPH*	Set Point High Limit
LSPL*	Set Point Low Limit
LTD.	Rate
LTI.	Reset
LTS.	Sample Rate
LVF	V-Flags
LYDA*	Yellow Deviation Limit

*Variables with an asterisk can be either a real number or an integer. Variables followed by a period are real numbers. Variables not followed by a period are integers. When you execute PACKLOOP using real numbers, two memory locations are allocated for each real number.

Pack Ramp/Soak Data

The Pack Ramp/Soak Data statement moves one or more elements (steps) of the ramp/soak profile for a given loop to or from a table. PACKRS is primarily intended to make the ramp/soak profiles accessible to an operator interface and to provide a method for dynamic ramp/soak profiling. The PACKRS format is shown in Figure 7–39.

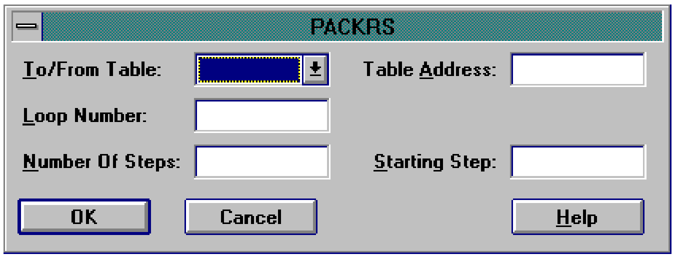


Figure 11.39 PACKRS Format

- A specifies whether you are writing data to or from the table.
- B specifies the address of the table, to or from which data are moved.
- C specifies the loop number whose ramp/soak profile is involved in the pack operation.
- D specifies the number of ramp/soak steps to pack.
- E specifies the starting step in the ramp/soak profile at which the pack operation will begin.

PACKRS Operation

The number of steps in a ramp/soak profile is established when it is programmed using TISOFT or your programming package. The PACKRS instruction cannot expand or shorten the ramp/soak profile for a given loop. This instruction can only read or modify existing steps in a preexisting profile.

PACKRS instructions that specify operations on non-existent profile steps are invalid, and the execution of this instruction terminates.

If TO Table is specified, this instruction copies the specified number of steps from the ramp/soak profile of a given loop, starting at the specified step number, to a table in memory whose starting address is indicated in the instruction.

If FROM Table is specified, this instruction copies the specified number of profile steps from a memory table into the ramp/soak profile for the indicated loop starting at the specified step number. The new step values overwrite the affected step values in the profile.

NOTE: *Care should be taken when using the PACKRS instruction with a FROM Table specified. If the ramp/soak profile being modified is in progress when the PACKRS instruction executes, then your process could react erratically due to the sudden replacement of values in the profile steps. You can use one of the following methods to ensure that the profile update is done when the current profile is not in progress.*

• In your program, check the state of the profile finished bit (bit 4) in LRSF for the corresponding loop. Do not execute the PACKRS statement unless the finished bit is set.

• In your program, place the loop in the manual mode, execute the PACKRS to update the ramp/soak profile, then return the loop to automatic mode. (Remember, this causes the ramp/soak profile to be restarted at the initial step.)

When stored in a memory table, ramp/soak profile steps are six words long and have the following format:

- Word 1 (bit 1): Step type — 0 = ramp step, 1 = soak step (bit)
- Word 1 (bits 2-16) + Word 2: Address of status bit (special address format)
- Words 3/4: Setpoint, if ramp step, or Soak time, if soak step (REAL number)

Words 5/6: Ramp rate, if ramp step, or Deadband, if soak step (REAL number)

The status bit address points to either an output point (Y) or a control relay (C). This address takes a short form for point numbers C1 - C512 and Y1 - Y1024. Higher point numbers use a long form of address. If all bits of the status bit address field are 0, then no status bit is selected for the step.

The short address form is shown in Figure 7–40.

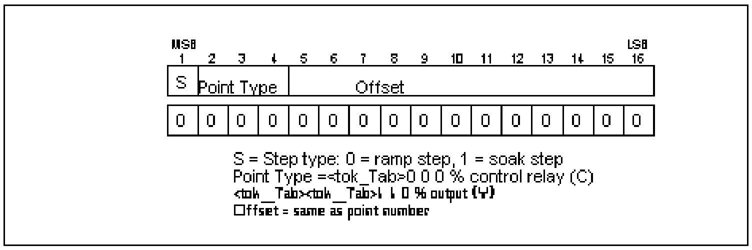


Figure 11.40 Address Format — Short Form

For example, the encoded address for Y23 using the short form is shown in Figure 11.41.

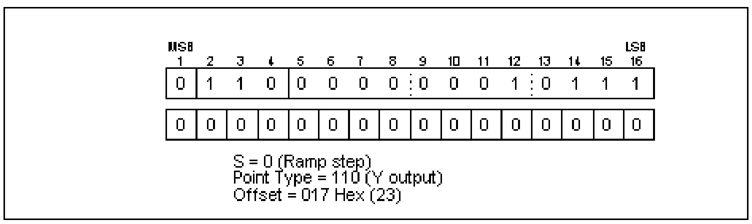


Figure 11.41 Short Form Address Example

The long address form is shown in Figure 11.42.

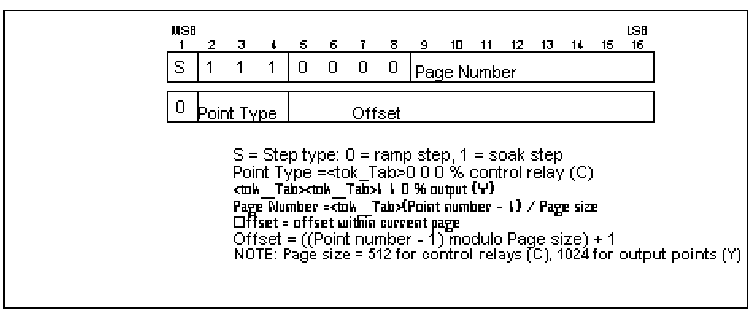


Figure 11.42 Address Format — Long Form

For example, the encoded address for C514 using the long form is shown in Figure 11.43.

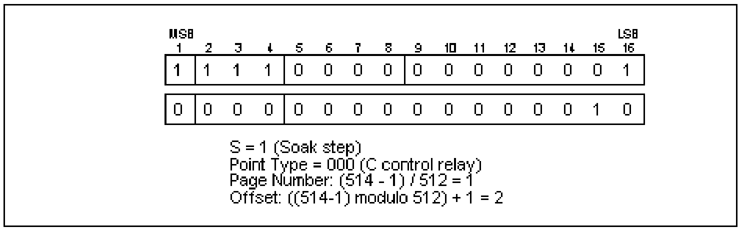


Figure 11.43 Long Form Address Example

Figure 11.44 shows an example of the PACKRS instruction moving values from a ramp/soak profile to a V-memory table.

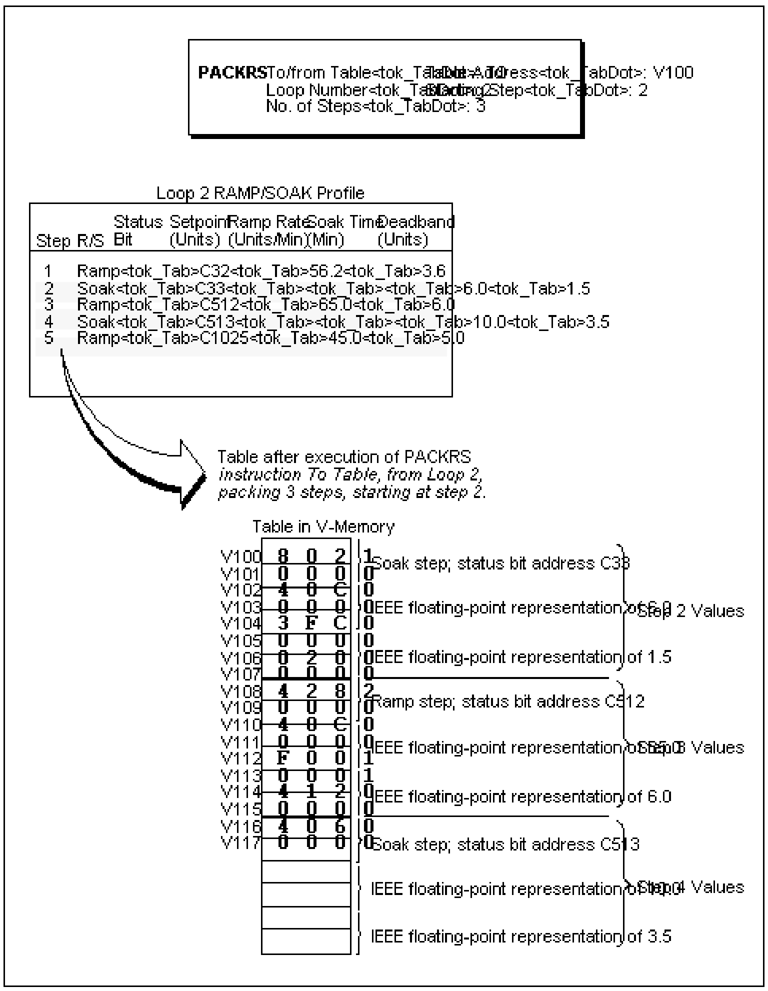


Figure 11.44 Example of PACKRS to a Table in V-Memory

Figure11.45 shows an example of the PACKRS instruction moving values from a V-memory table to a Loop Ramp/Soak profile, changing two of the values in the profile, and leaving the remaining values unchanged.

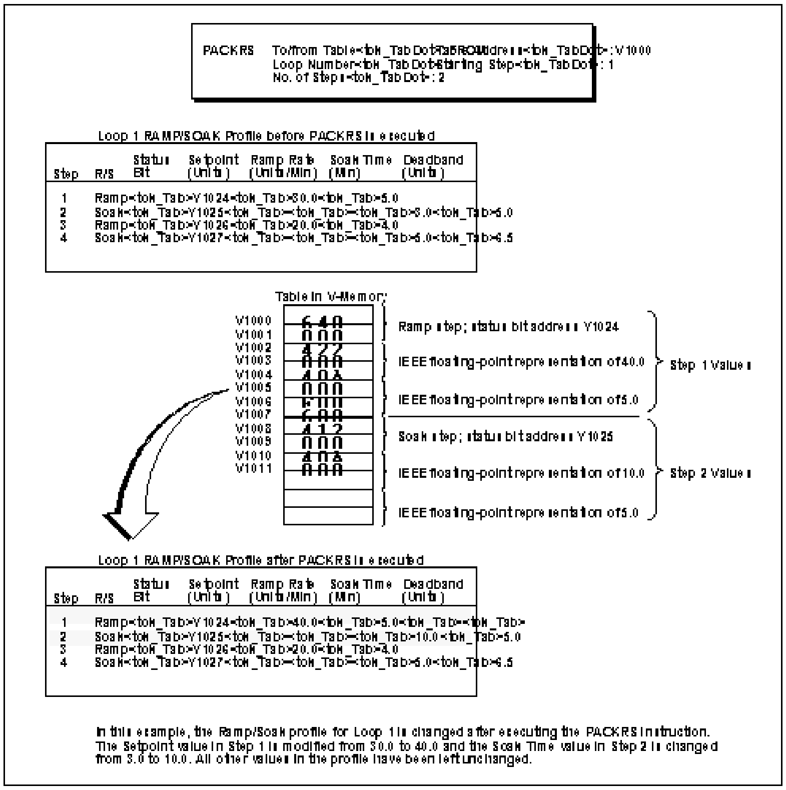


Figure 11.45 Example of PACKRS from a Table in V-Memory

PETWD

The PETWD (Pet Scan Watchdog) allows you to extend the scan watchdog limit while performing an in-line SF program or SF subroutine from an RLL program. When the PETWD instruction executes, the scan watchdog timer is reset at that instance of time during the scan, therefore extending the scan watchdog limit beyond the configured scan watchdog limit.

A possible use would be in a large table lookup operation performed while the controlled process is at a steady-state condition.

The RETURN format has no subfields.

The PETWD instruction is intended to be used in the normal RLL task

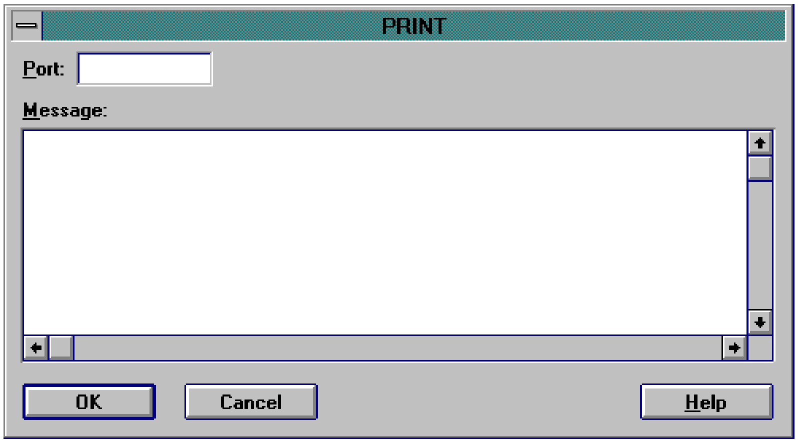
(TASK 1).

PETWD is available only for in-line compiled SF programs or subroutines in CPUs that support PowerMath.

Warning	**The PETWD instruction allows you to place the PETWD instruction in an infinite loop, therefore preventing the scan watchdog limit from ever being reached. If the PETWD instruction is in an infinite loop, the PLC would not issue a scan watchdog FATAL ERROR to shut the process down, therefore leaving your process uncontrolled. An uncontrolled process could result in death or serious injury to personnel, and/or damage to equipment. Ensure that the PETWD instruction is not located in an infinite loop. To ensurethat the PETWD instruction is not located in an infinite loop within an SF program or subroutine, place the PETWD instruction without a label at the beginning of the SF program or subroutine.**

Printing

The Print statement sends a message to the ASCII communication ports. This statement can be used to print both text and the contents of integer and real variables. The PRINT format is shown in Figure 11.46.



- A is the port number. You must enter a 1 in this field.
- B contains a free format message. The message begins on the line following the port and message designator fields. Element addresses and Expressions are separated by a space. No embedded space or the assignment operator (:=) in an expression is accepted.

PRINT Operation

- The operation of the PRINT statement is described below.
- When the PRINT statement executes, the message is sent to the port(s) specified.
- The maximum message length is 1019 characters, with characters counted in entries as follows:
  Each text character = 1 character
  Each variable entry = 6 characters
  Each variable text entry = 6 characters
  Carriage Return & Linefeed = 2 characters
- Text Entries contain ASCII text to be printed. Text entries are enclosed in quotation marks.
- Example: PRINT PORT=1 MESSAGE:
  "END OF SHIFT REPORT"
- Variable Entries print the contents of variables in either integer or real format. Variables must be separated by spaces. Real numbers are indicated by following the address with a period (.). Integers are printed right-justified in a six character field with a floating minus sign. Real numbers are printed right-justified in a twelve character field using a FORTRAN G12.5 format.
- Example: PRINT PORT=1 MESSAGE:
  "THE VALUES ARE" WX5 V104.
- Time Entries are used to print out a variable in time format. The variable is printed out as hh:mm:ss. Time entries are indicated by following the address of the variable (EL or EXP) with :TIME.
- Example: PRINT PORT=1 MESSAGE:
  "THE TIME IS NOW" STW141:TIME
- Date Entries are used to print out a variable in date format. The variable is printed out as yy/mm/dd. Date entries are indicated by following the address of the variable (EL or EXP) with :DATE.
- Example: PRINT PORT=1 MESSAGE:
  "THE DATE IS NOW" STW141:DATE

- ❑ Variable Text Entries are used to print out text stored in either V or K memory. Variable Text Entries are indicated by following the address of the text (EL or EXP) to be printed with a percent sign (%) and the number of characters to be printed. If the number is coded as zero, PRINT assumes that the first word of the indicated variable contains the number of characters to print.

  Example: PRINT PORT=1 MESSAGE:
  "BOILER" V250%16
  "DESCRIPTION" V102%0

  "Boiler" V250%16 causes the 16 characters in V-Memory locations V250-V257 to be printed. Each word contains two 8-bit characters.

  "Description" V102%0 causes the number of characters specified in V102 to be printed. If V102 contains 5, then the characters in V103-V105 are printed.

Variable Text Entries are a valuable tool for embedding control characters to be used by the device receiving the ASCII characters. The next page gives instructions about how to embed a control character in variable text.

The form-feed indicator <FF> is entered as: "<FF>".

Follow these steps.

1. Enter the double quote character **"**
2. Enter the less than character **<**
3. Enter the F character **F**
4. Enter the F character **F**
5. Enter the greater than character **>**
6. Enter the double quote character **"**

Example: PRINT PORT=1 MESSAGE:
"THERE IS A FORMFEED
AFTER THIS <FF>"

To enter a <CR><LF> (Carriage return/Linefeed), follow these steps.

1. Enter the double quote character **"**
2. Press the carriage return key or
3. Enter the double quote character **"**

Example: PRINT PORT=1 MESSAGE:
"THERE IS A CARRIAGE RETURN LINEFEED AFTER THIS
"

To print the double quotes "", precede it with another double quote " as shown in the example below.

Example: PRINT PORT=1 MESSAGE:
" ""THIS QUOTED TEXT IS PRINTED INSIDE DOUBLE QUOTE

Return from SF Program/Subroutine

The RETURN statement is used to terminate a SF program or a SF subroutine. If invoked from a SF program, the program terminates. If invoked from a SF subroutine , control returns to the statement in the SF program following the SF subroutine call. The return format has no subfields. If there is no RETURN statement, the program terminates after the last statement.

Scaling Values

The Scale statement uses as input an integer input and converts it to engineering units scaled between high and low limits. The SCALE format is shown in Figure 11.48.

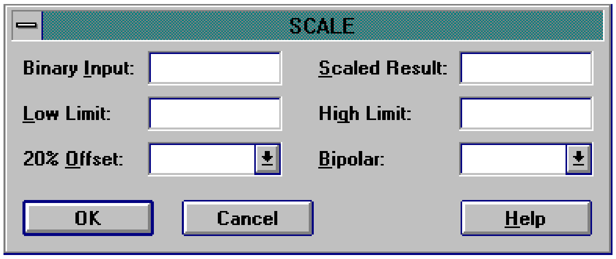


Figure 11.48 SCALE Format

- A is the memory location of the input.
- B is the memory location of the result after conversion.
- C is the lower limit to which the input can be scaled.
- D is the upper limit to which the input can be scaled.
- E indicates if the input is 20% offset (Yes) or 0% offset (No).
- F indicates if the input is bipolar (Yes) or not (No).

NOTE *You cannot choose both bipolar and 20% offset for an input (Fields E-F).*

SCALE Operation

The operation of the Scale statement is described below and illustrated in Figure11.49.

- Each time the SCALE statement executes, an integer located in A converts to an integer or real number in engineering units, scaled between high and low limits.

  If the input is a variable that could range from -32000 to +32000, the variable is bipolar. Set option F to Y(es). If the input is a variable that could range from 0 to 32000, the variable is unipolar. Set option F to N(o).

  If the input is a variable that has a 20% offset (ranges from 6400 to 32000), set option E to Y(es). If the input is a variable that has a 0% offset, set option E to N(o).
- The result is stored in the address specified by B.

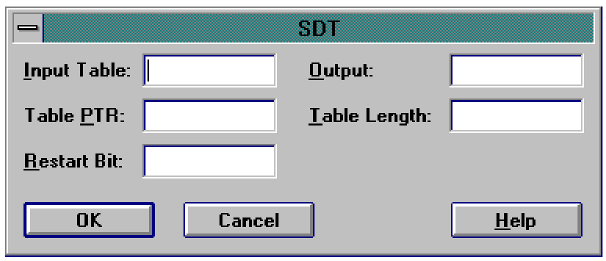
  The low and high limits specified in C and D determine the range of the converted number. Values of C and D may fall within the following limits.

  Range = to
  to
- An error is logged if the input value is outside the low-limit to high-limit range; and the output is clamped to the nearer of either the low limit or the high limit.

  You can use the SCALE statement to convert an input signal from an analog input module to a value in engineering units. For example, consider these conditions.
- The input is a 4-20 mA signal that is converted by the analog input module to a value between 6400 and 32000 (unipolar, 20% offset) and sent to WX33.
- You want the result of the SCALE statement to be a real number ranging between 0 and 100 and be placed in V100.

Sequential Data Table

The Sequential Data Table statement moves words one at a time from an existing table to a destination address. A pointer designates the address of the next word in the table to be moved. Each time the statement is executed, one word moves and replaces the word at the destination address. The SDT format is shown in Figure 11.50



- A is the starting address for the table.
- B is the output address to which the words are moved.
- C is the address of the pointer.
- D is the length of the table and must be a value greater than zero.
- E is the address of the restart (status) bit and can be a C or Y.

SDT Operation

The operation of the SDT statement is described below and illustrated in Figure11.51.

- o The SDT moves words from a pre-existing table.

  The size of the table depends upon the memory location that you choose and, if the memory is user-configurable, how much memory you allocated.
- o Before the SDT is executed, pointer C contains zero. You must design your program to set the pointer to zero.
- o Each time the SDT is executed, the following actions occur.

  The table pointer is incremented by 1. Then the word in the table location specified by the pointer is moved to the destination address specified by B.

  The process is repeated until the number of words specified in D has been moved.
- o When the last word has been moved, the pointer is reset to zero.
- o The restart bit E is on, except for the following conditions.

When the SDT resets the pointer, the restart bit turns off.

Prior to the first execution of the SDT, the bit could be either off or on depending upon prior usage.

The value of the pointer does not change when the SDT is not executing. All values in the table remain the same, and destination address B contains the value of the last word moved from the table.

You can use other logic to reset the pointer to zero, but the restart bit does not turn off.

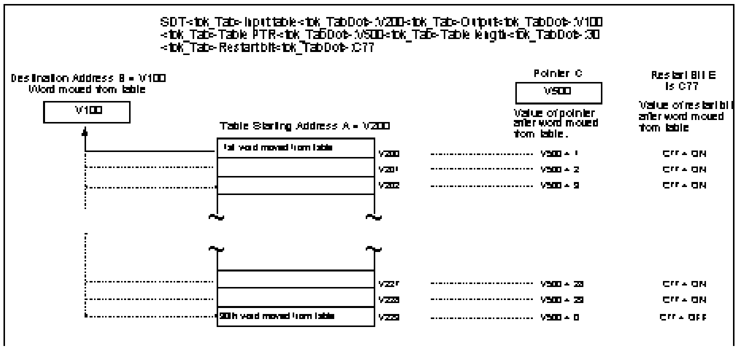
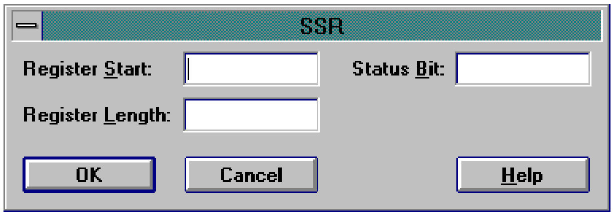


Figure 11.51 SDT Statement Example

Before the SDT executes, the pointer V500 contains 0 (zero). When the statement executes, the pointer increments by 1, and the value in V200 is moved to V100. This process repeats each time the statement executes. After the last word is moved, the pointer resets to 0.

Synchronous Shift Register

The Synchronous Shift Register statement builds a table that functions as synchronous shift register. The SSR format is shown in Figure 11.52



- A is the starting address for the shift register.
- B is the status bit (C or Y) and is turned on when the register is empty.
- C is the length of the shift register. The maximum number of elements stored in the register is C. If a constant value is entered, it must be greater than zero.

SSR Operation

The operation of the SSR statement is described below and illustrated in Figure 11.53.

- The starting address A designates the memory area in which the shift register is located.
- The register length C determines the size of the shift register. Size depends upon the memory location that you choose and how much memory you allocated (if the memory is user-configurable). The maximum number of elements stored in the register is C.
- The first position of the register, Register Start A, is empty until an element moves into A from another source.
- Each time the SSR executes, the element currently in memory location A shifts to A + 1. The element in A + 1 shifts to A + 2. Elements move down the shift register to A + 3, A + 4, etc., and A resets to zero.
- After the register is full, shifting in a new word causes the loss of the last word in the register at location [A + (C - 1)].
- The register is considered empty when it contains all zeros. The status bit B turns on when the register is empty.

NOTE: *If the register contains the value -0.0, the register is not recognized as empty, and the status bit does not turn off.*

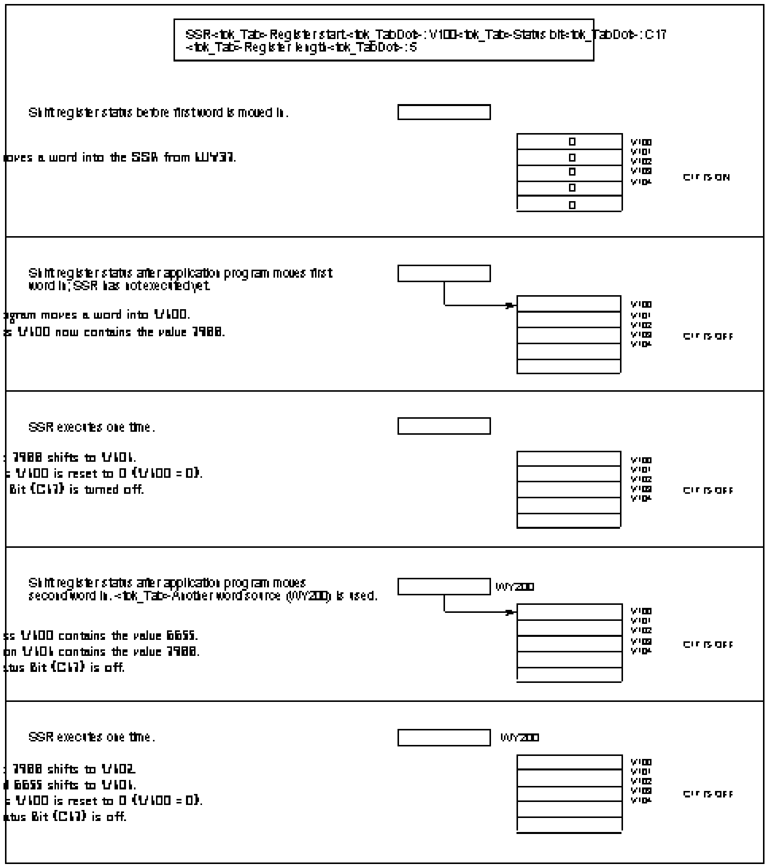
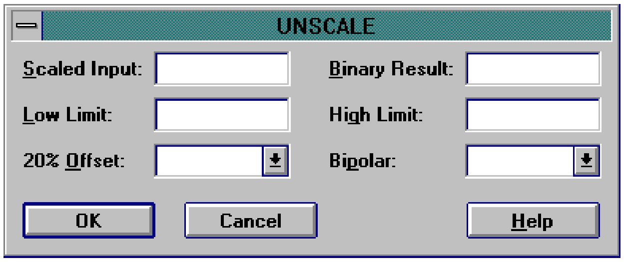


Figure 11.53 Example of SSR Operation

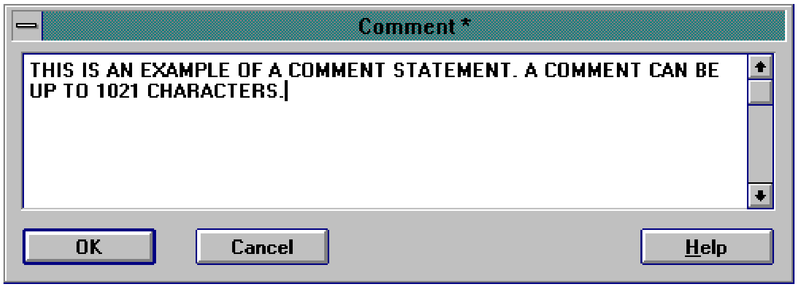
Unscaling Values



The UNSCALE statement takes an input value in engineering units, scaled between high and low limits, and converts it to an integer.

- **Scaled Input** is the memory location of the input Descriptors are Integer Address (address containing an integer value (e.g., V100 or WX77 or V100(2))); Real Address (address containing a real value (e.g., V120. or WY55 or V100.(2))).
- **Binary Result** is the memory location of the result after conversion. Valid Descriptors are Integer Address (address only accepts an integer value and Writeable Address (e.g., WY1000 or V23(2), C55).
- **Low Limit-**is the lower limit of the **Scaled Input.** Valid Descriptors are Literal constants Less Then or Equal to the **High Limit** Literal constants.
- **High Limit-**is the upper limit of the **Scaled Input.** Valid Descriptors are Literal constants Greater Then or Equal to the Lower **Limit** Literal constants.
- **20% Offset-**indicates if the output is 20% offset (yes) or 0% offset (no). If the output is a variable that has a 20% offset (range from 6400 to 3200), set option E to Y(es). If the output is a variable that has a 0% offset, set option to N(o).
- **Bipolar-**indicates if the output is bipolar (yes) or not (no). If the output is a variable that could range from -32000 to +32000, the variable is bipolar. Set option to Y(es). If the output is a variable that could range from 0 to 32000, the variable is unipolar. Set option to N(o).

Comment



The comment statement inserts a comment in a program for documentation purposes. The comment statement is ignored during program execution. A comment statement can contain a maximum of 1021 characters.

13 - Auditing

Activity Audit

The Activity Audit contains records which list programming and setting changes made by users while running WorkShop. These records contain:

- the date and time the activity occurred
- a general description of the activity
- the machine name of the computer on which the activity occurred
- and, when available, the name of the user who performed the activity

Examples of particular Audit activities that may be recorded in the Activity Audit file include:

- each time PLC WorkShop attaches online to a PLC
- whenever logic is changed in the ladder program
- and PLC memory configuration is modified.

Activity Audits can be written to Microsoft Access files, Structured Query Language (SQL) files or the Windows Activity Log (when the FT Security Server is running on an accessible computer). For more on the Ft Security Server, please see Chapter 5 of this manual.

Activity Audits written to Microsoft Access and SQL files may be read with many popular viewers. Audits written to the Windows Activity Logs may be read with the Windows Event Viewer which is found in the Windows NT/2000/XP "Administrative Tools" menu.

The Activity Audit feature may be used in conjunction with Password Security but can used independently from security.

Setup

Select the "Activity Audit Setup…" item of the "Options" menu as illustrated below to setup the activity audit.

Figure 22.1a

If security is not enabled, the Activity Audit Setup option is not protected and any user can access this feature.

When security is enabled, WorkShop determines under which operating system it is running. If WorkShop is running under Windows 95/98/ME, the "Password Required" dialog below appears.

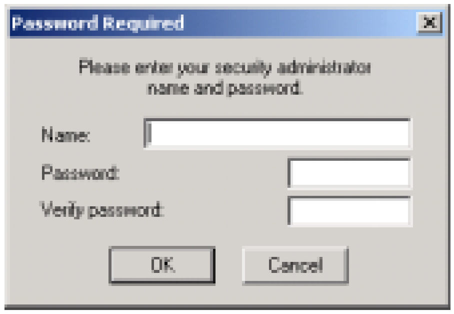


Figure 22.1b

This dialog verifies the entered "Name" and "Password" are those of an administrator. Only an Administrator is allowed to Edit/Enable Audits when security is enabled. If the entries are those of an administrator, the Activity Audit Setup dialog below appears.

Activity Audit Setup Dialog

If the current computer is running under Windows NT/2000/XP, PLC WorkShop determines if the user logged into the operating system is an administrator. If the user is an administrator, the Activity Audit Setup dialog below appears. If security is enabled but the user logged into the operating system is not an administrator, access is not granted to the Activity Audit Setup dialog.

There are two types of Auditing available to the user.

1. FasTrak Auditing allows for audits to be written to a database file.
2. NT Activity Auditing requires the FTSecSer.exe application to be installed. (See Chapter 5 of this manual)

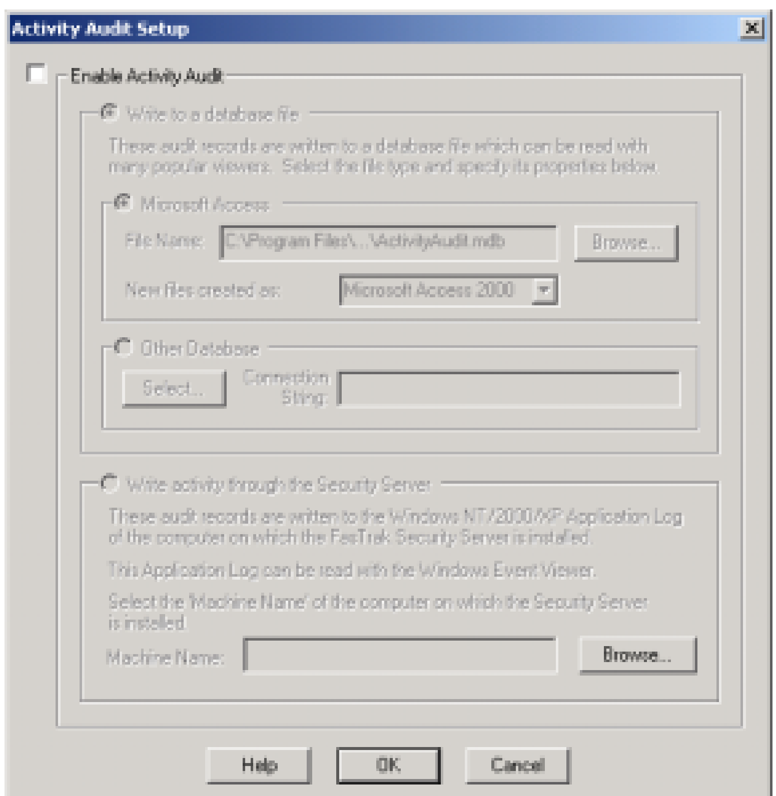


Figure 22.2a

Check the "Enable Activity Audit" check box to select Auditing. Then choose the method by which audit records are written.

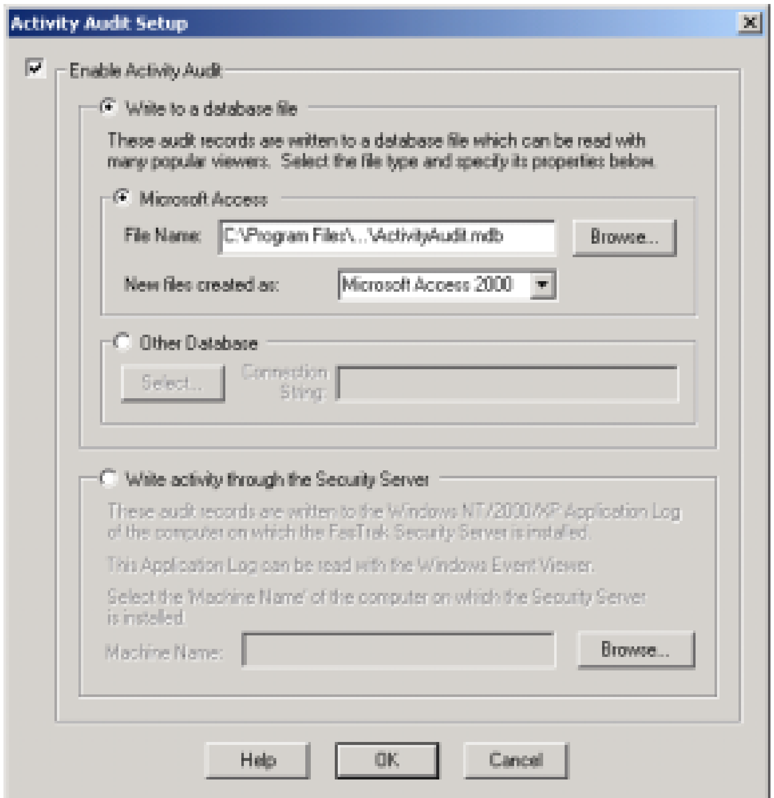


Figure 22.2.b

Click the "Write to a database file" radio button to record the activity audits to database files. These can either Microsoft Access or Structured Query Language (SQL) files. These log files are simply spreadsheet-like files and may also be viewed with applications such as Microsoft's Excel, Lotus, or even text editors.

Click the "Microsoft Access" radio button to write Activity Audits to a Microsoft Access file. Either enter or click the "Browse" button to specify the Access file name to which activity records will be written. The pull-arrow of the "New files created as:" combo box can be used to select the format of Microsoft Access files desired

Click the "Other Database" radio button to select the SQL file type as illustrated below.

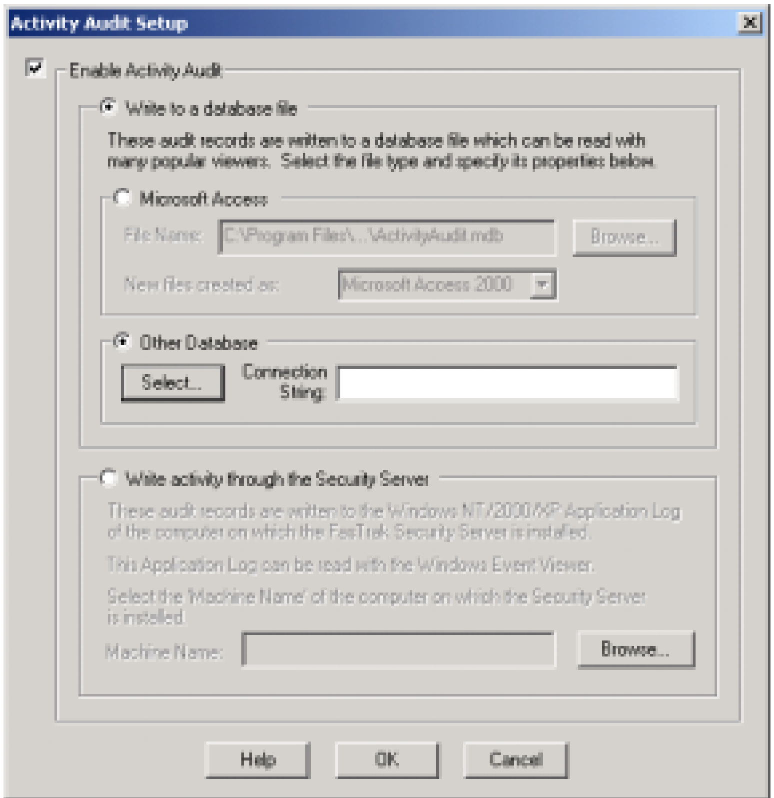


Figure 22.2c

Either enter the Connection String in the edit box or click the "Select…" button in the "Other Databases" group box to display the Data Link Properties wizard dialogs below to assemble the SQL file connection string.

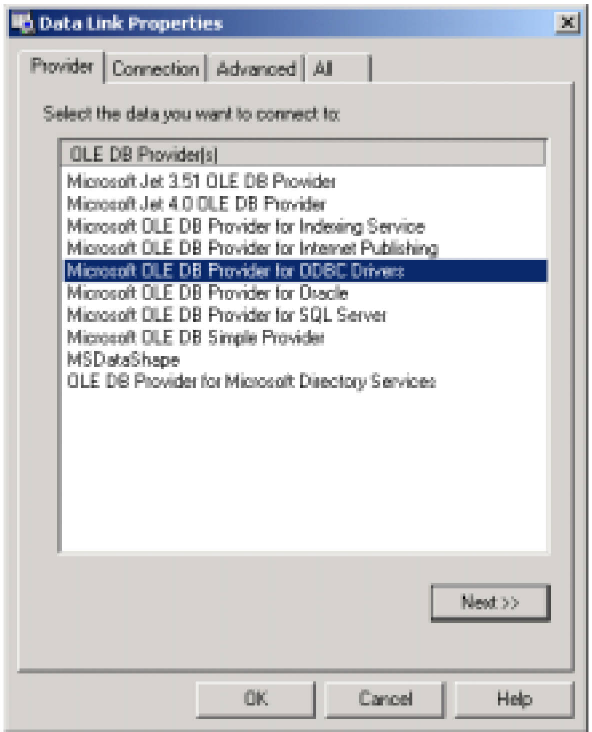


Figure 22.2d

The activity audit can also be written to the Application Log of the computer running the FasTrak Security Server.

Click the "Write activity through the Security Server" radio button the select the "Machine Name" of the computer on which the FasTrak Security Server is installed. The Application Log may also be viewed used Windows' "Event Viewer." More information for this utility is found in Chapter 5 of this manual or by clicking here.

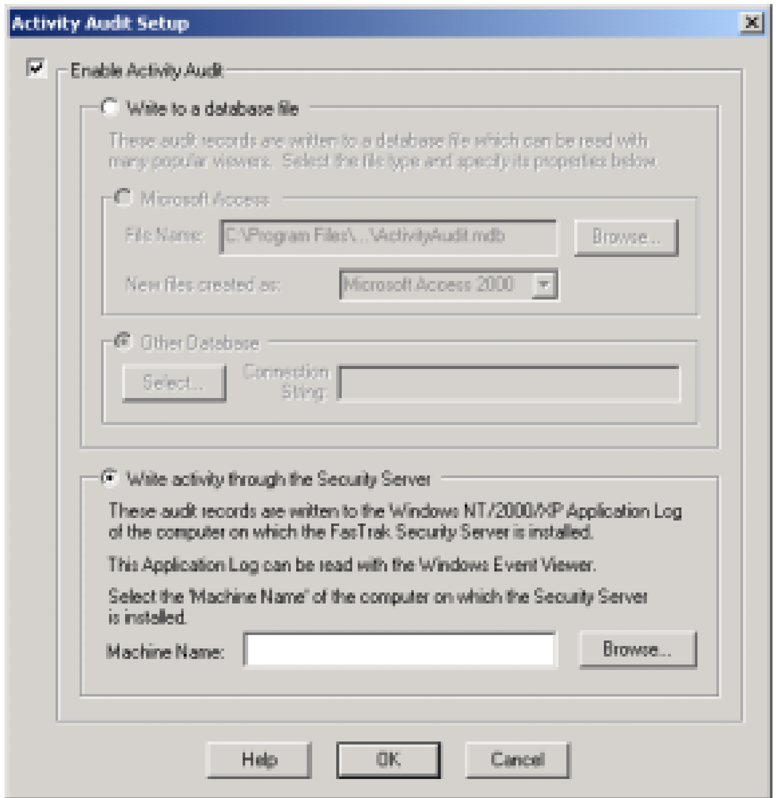


Figure 22.2e

Audits written to Windows Activity Logs may be read with the Windows Event Viewer which is found in the Windows NT/2000/XP "Administrative Tools" menu.

14 - FasTrak Authentication and NT Security

Overview

The WorkShop "Password Security" feature allows one or more security administrators to maintain a list of users and their access privileges. Access privileges restrict which functions of the application (such as online and offline editing, I/O forcing, loading, saving changes to disk etc.) individual users can perform.

Additional on-line security can be achieved using PLC Password Aliases. For more on this subject, see PLC Password Aliases.

Security Setup

Password security in WorkShop is disabled by default. Without password security enabled, any user may access any portion of the programming package.

Before the password security feature can be used, a security administrator must be established by clicking the Options menu followed by Security Setup.

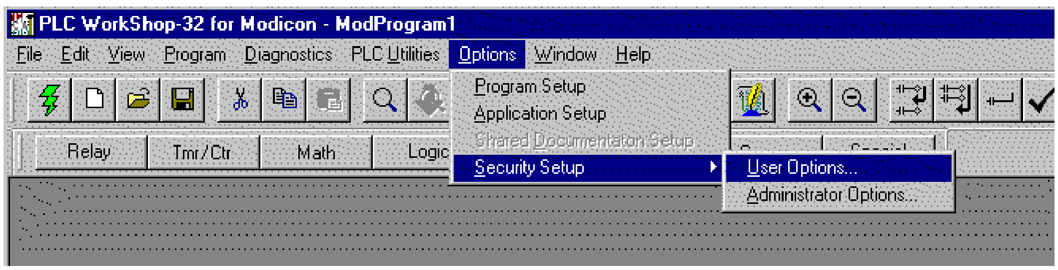


Figure 14.4

The "Security Setup" menu contains two items, "User Options…" and "Administrator Options…". The user security options allow users to change their own passwords or allow other users to enter their password without having to exit and restart WorkShop.

The administrator security options allow the security administrator to add, edit and delete users and their access privileges to specific features in WorkShop.

Administrator Options

By default, there is no security administrator and the password security feature is not enabled. Therefore, the "User Options…" menu item is disabled until a security administrator is established. Select "Administrator Options…" to establish a security administrator. The "Password Required" dialog below appears.

The security administrator is required to enter a name and password. The administrator name can be up to twenty characters long. The password can be up to fourteen characters long. Valid characters for both fields are alpha-numericsand the other keyboard-entered characters (! @ # $ %, etc.).

The first time a security administrator is established, the password must be entered twice. This original security administrator can add other security administrators through the "Security Administration" dialog discussed below.

Figures 14.5a and 14.5b

If a security administrator has already been established, the password does not need to be entered a second time for verification. When a security administrator exists, "Verify password" is disabled as illustrated below.

Enter the security administrator name and password, then click "OK". If the two passwords match, the "Security Administration" dialog below appears.

Choosing a Security Type

PLC WorkShop for Modicon 32-bit offers two unique security systems for use. If the computer on which PLC WorkShop is installed is running under MS-Windows NT, 2000, or XP, PLC WorkShop determines if the user currently logged in to the operating system is an administrator (or belongs to the Administrator group).

To activate the security feature of PLC WorkShop,click on Options, followed by Security, and then Administrator Options. (see below)

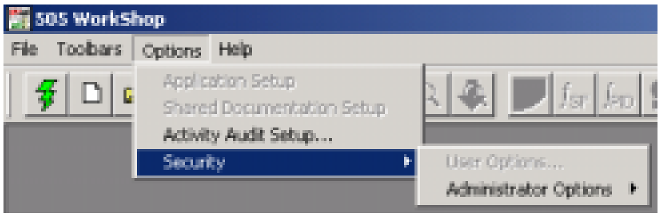


Figure 14.1a

If PLC WorkShop is running under MS-Windows 95, 98, or ME, the "Password Required" dialog appears (see below). This dialog allows the user to become the first security administrator.

Figure 14.1b

If the users already on file belong to the Administrator's group, then the fields above are filled in and the dialog is used to verify the user's password. Clicking OK launches the "Security Type Selection" dialog, seen below.

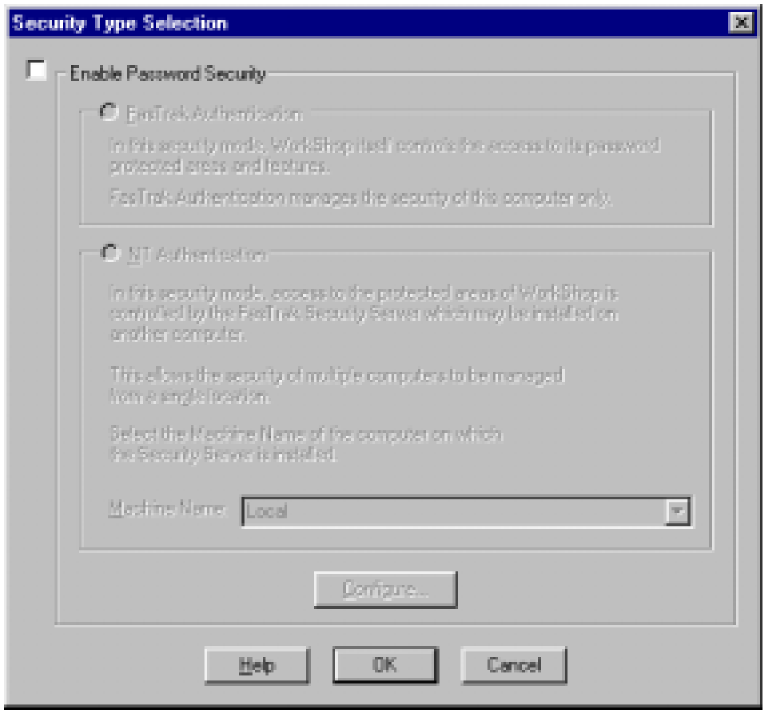


Figure 14.1c

Clicking on "Enable Password Security" activates security and the use is free to choose from either FasTrak Authentication or NT Authentication.

- FasTrak Authentication- Whereby the application itself regulates user access to password-protected areas. Management takes place on the local computer only. For more on this area of Security, click here or go to Chapter 5 of the manual for PLC WorkShop.
- NT Authentication- When enabled, PLC WorkShop requests permission to access password-protected areas from the FasTrak Security Server. This server is a separate application which can be installed on the local computer or on any other computer running under MS-Windows, NT, 2000, or XP. This server is capable of managing multiple computers from a single, central location. Click here to go to the chapter on this feature or please see other sections in this chapter of this PLC WorkShop manual.

FasTrak Authentication Security

The Authentication Security Dialog is accessible from the Security Type Selection screen (please see that section of the manual). The dialog is seen below.

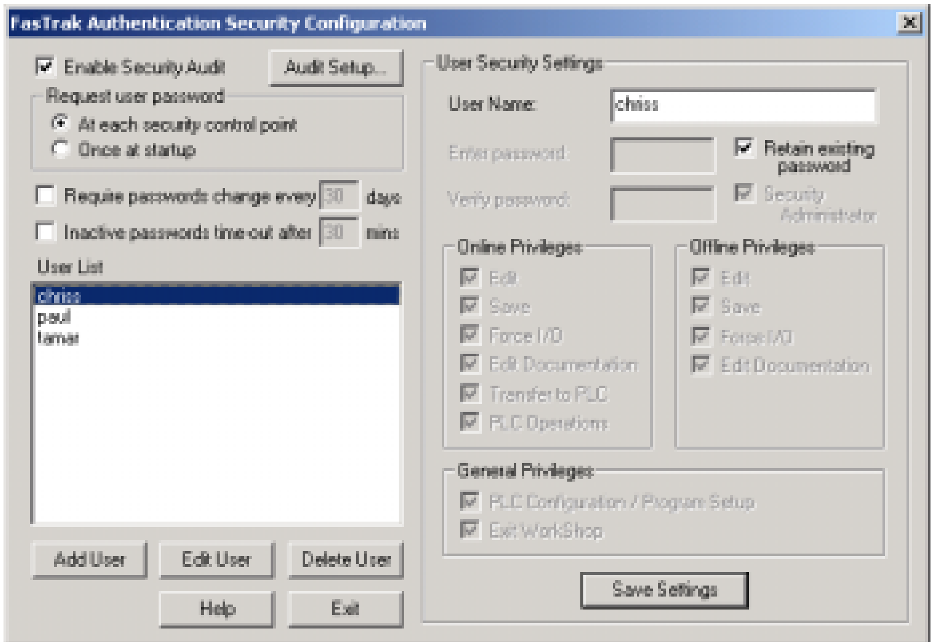


Figure 14.2

In addition to controlling user access, this security mode also can record each attempt made by a user to the parts displayed above. To enable this, click on the "Enable Security Log" check box.

Note: *If, when exiting this dialog with the check box checked but having not setup the security log, PLC WorkShop will warn the user that no log file had been configured and the check box selection will not be retained.*

To learn more about setting up the FasTrak Authentication Log, please refer to this chapter or click here.

FasTrak Authentication Log Setup

The following dialog is launched from the FasTrak Authentication Configuration dialog by clicking on "Log Setup."

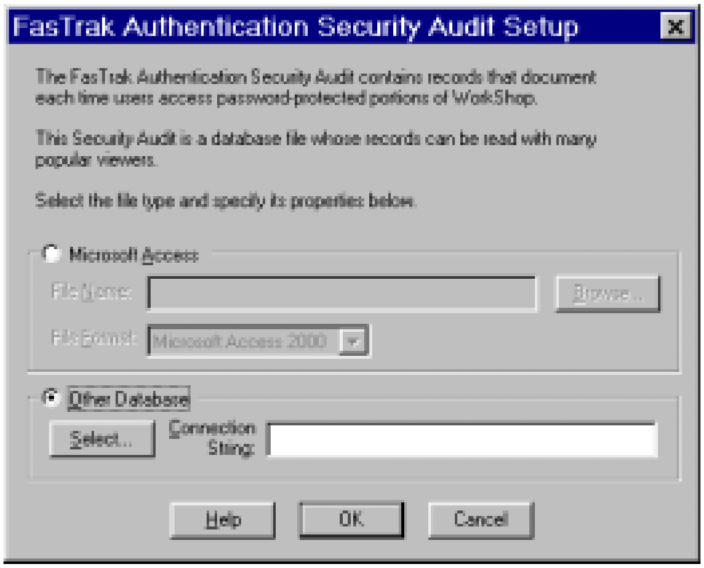


Figure 14.3a

To write the local security log to a Microsoft Access database file, click the "Microsoft Access" radio button. Enter or "Browse…" the name of the Access file and select a "File Format" from the group box.

To write the local security log to an SQL database file, click the "Other Database" radio button. Enter the file "Connection String" in the edit box.

Data Link Properties Dialog

If choosing a database other than Microsoft Access, and the user does not provide a Connection String, click on "Select" and the Data Link Properties dialog opens. This dialog is the first in a series of tabbed dialogs which help the users create the necessary string of the local security log file.

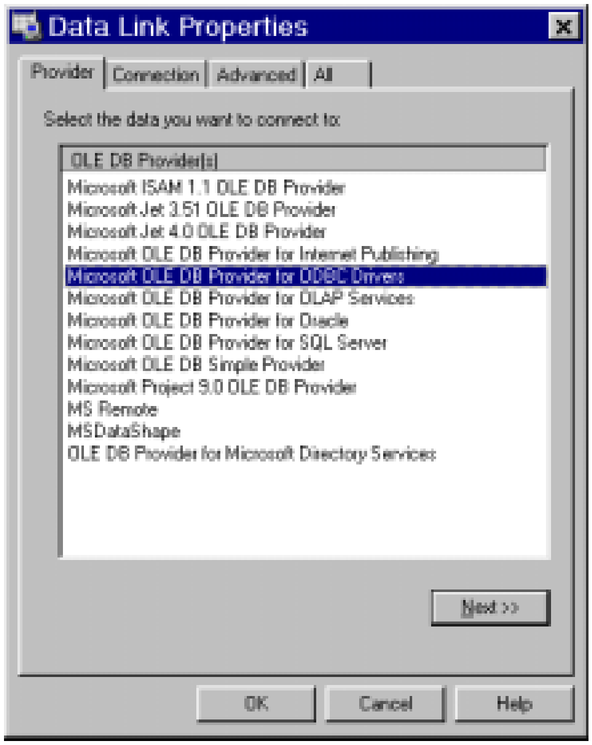


Figure 14.3b

Security Administration Dialog

Security administrators control the password security feature of WorkShop through the "Security Administration" dialog below.

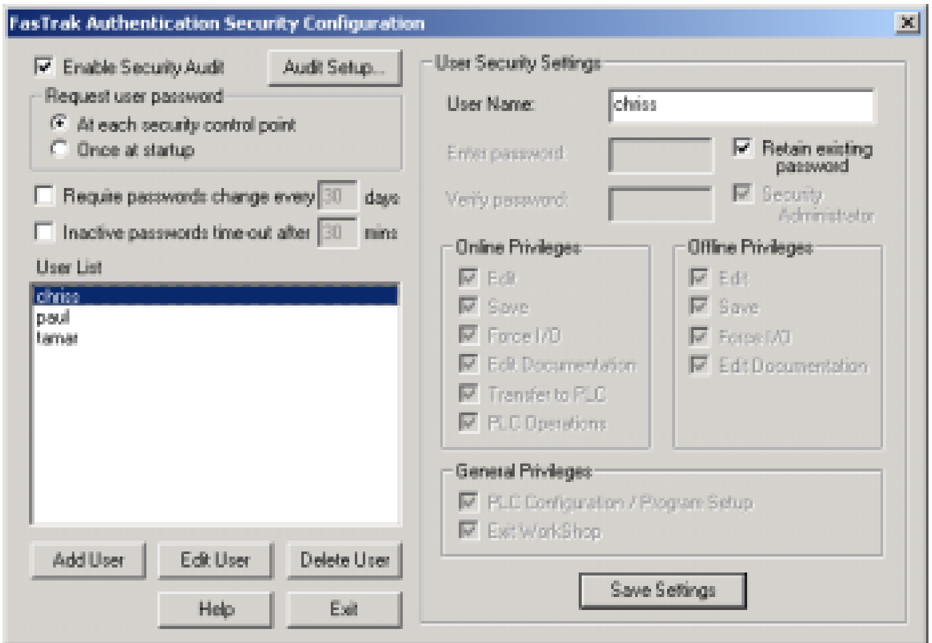


Figure 14.6

The security feature can be enabled and disabled and the places at which user passwords are requested can be specified through this dialog. The list of users and their rights to access specific features within WorkShop can be managed within this dialog.

Enable Password Security

The password security feature is disabled in WorkShop by default. The security administrator can turn this feature on by checking "Enable password security". When this box is checked, users are requested to enter their passwords either once each time WorkShop is started or each time they attempt to enter a password-protected section of the application.

The current security mode is indicated in the status bar at the bottom of the WorkShop window as seen in the example below. "Security:Enabled" appears when security is enabled. "Security:Disabled" appears when security is disabled.

Inactive Passwords Timeout in *N* Minutes.

Once users enter their passwords, they can operate PLC WorkShop indefinitely (within their access privileges). However, if PLC WorkShop runs unattended for a designated number of minutes without user interaction, the password under which the application is running can be timed-out.

Click "Inactive passwords time-out after *N* mins" to activate this feature. Enter the number of minutes PLC WorkShop may remain inactive before the current password times-out. The default number of minutes is 30 but the valid range is 1 to 999.

Request User Password

Once password security is enabled, PLC WorkShop requests users to enter their passwords at various times to operate the software.

Passwords can be requested each time users attempt to enter a password-protected portion of the application. Alternately, passwords may be requested only once when WorkShop is first started.

At Each Control Point

Password control points are designated as "privileges" within the "User Security Settings" group box illustrated in the following pages. Specifying that passwords are requested at these control points requires users to enter their name and password every time they attempt to use these features.

The advantage of this selection is that multiple users can operate WorkShop without having to exit and restart the application under another user name and password. Once a user has completed a password protected operation and exits that function, other users can access password protected operations by entering their own user names and passwords.

The disadvantage of this selection is that it requires users - even the same user - to re-enter user names and passwords each time they attempt to access one of the password controlled features.

For example, saving a program to disk is one of these control points. Assume a user edits a ladder program. The user then selects "Save" from the "File" menu to write the changes to disk. The "Password Required," pictured here, appears requesting the user name and password to access the save feature.

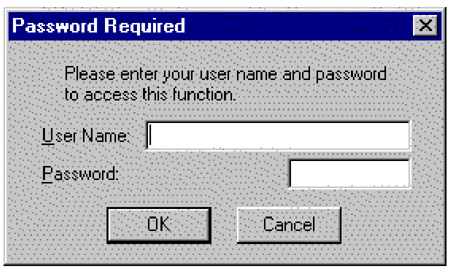


Figure 14.7

Upon entering a valid user name and password (and the security administrator has given this user privileges to this feature), the user is granted access to the save feature. After saving, the user immediately returns to the program, makes another change, then selects "Save" from the "File" menu again. Even though the user was just granted access to the Save feature, the "Password Required" dialog reappears requesting the user name and password.

Once At Startup

Alternately, user name and password can be requested once when WorkShop is started – or after selecting "Switch user" in the "User Security Setup" dialog, illustrated later in this document. Upon entering a valid user name and password, the new user can access each password-controlled feature (to which the security administrator has granted privileges) without having to re-enter the user name and password.

The advantage of this selection is that users enter their names and passwords once at startup and are not required to re-enter them each time they choose a password-controlled feature.

The disadvantage of this selection is that one user can start WorkShop with a correct name and password but another operator can continue to use the application with all the original user's access privileges.

Adding Users

Click the "Add User" button to add a new user to the list. The dialog controls for a new user are set as illustrated below.

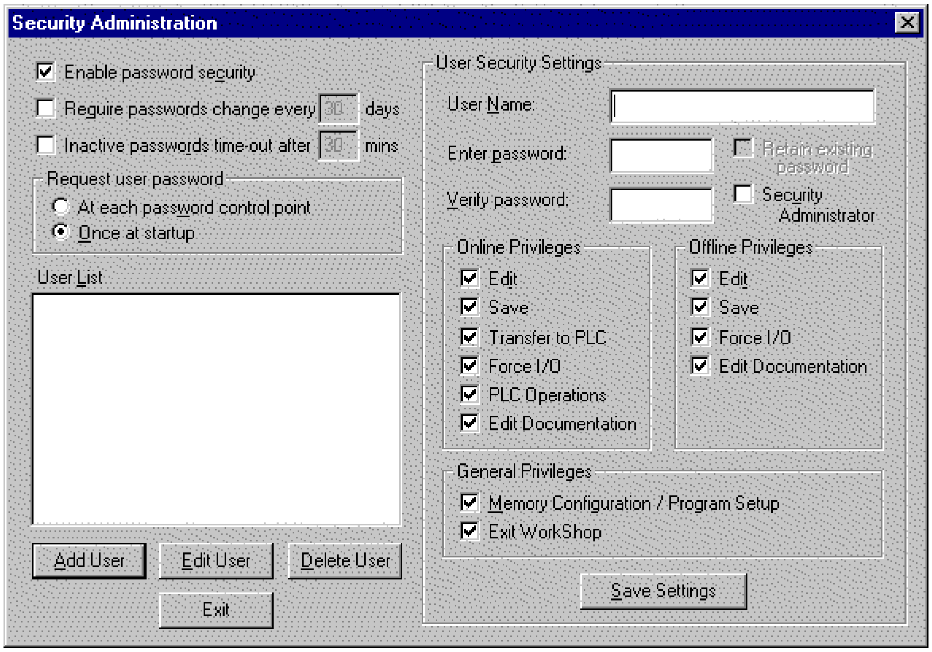


Figure 14.8

User names can be up to twenty characters long. Passwords can be up to fourteen characters long. Valid characters for both fields are alpha-numerics and the other keyboard-entered characters (! @ # $ %, etc.).

Enter the user name then enter and re-enter the password.

All privileges are initially checked for new users. Check/uncheck the privileges appropriate for the user.

Security administrators can create other security administrators. Check the "Security Administrator" box to grant the new user all administrator rights and privileges.

Click "Save Settings". The list of existing users is checked to assure the new user name is not a duplicate. If the new user name is unique, the two password entries are compared. Finally, the new user name is added to the "User List" box.

Editing Users

There are two ways to select an existing user to edit. Either double click a user name from the "User List" box or highlight a user name in the "User List" box and click the "Edit User" button.

By default, "Retain existing password" is checked and the password edit boxes are disabled. This allows the security administrator to modify privileges without needing to re-enter the user identity information (the user name and password).

To change the user name, type an alternate entry in the "User Name" edit box.

To change an existing user password, uncheck "Retain existing password" to enable the password edit boxes. Enter and verify a new password. Both password entries must match in order to save the new password.

Security administrator rights may be granted to or revoked from the user by checking or un-checking "Security Administrator".

Deleting Users

To remove a user, highlight a name in the "User List" box and click the "Delete User" button. A verification message appears which asks to confirm the deletion. Click "OK" and the user is removed.

User Security Setup Dialog

Once the security administrator adds users, their initial passwords and access privileges, the "User Options…' item of the "Security Setup" menu (as illustrated earlier) is enabled. Users can select this menu item to change their passwords through the "User Security Setup" dialog below.

If the security administrator elects to request user passwords once at startup, the "Switch User" radio button is enabled. Otherwise, if passwords are requested at password control points, this radio button is disabled.

Changing the User's Password

Users may change their own passwords. Security administrators may also change their passwords. Click the "Change user password" radio button and the dialog controls are set as illustrated to the left.

Figure 14.9

Enter the user name and current password. Then enter and verify the user’s new password and click "OK".

If the entered user name is in the list of users previously added by the security administrator, the "Enter current password" is compared to the password already associated with that user.

If the "Enter current password" matches the existing password for that user, the "Enter new password" and "Verify new password" are compared with each other. If the two new passwords match, then the new password replaces the current.

Switching the User

If the security administrator specifies user passwords are requested only once when WorkShop is started, the password entered at startup (and the access privileges associated with it) is in effect until WorkShop is exited. Another password is not requested until WorkShop is restarted.

However, the "Switch user" name and option allows another user to enter another password without requiring the application be exited and restarted.

Enter the user name and password of the user who will assume operational control of WorkShop. If a valid user name and its matching password are entered, the security access privileges are reset to those of the new user.

Figure 14.10

NT Authentication Security Server

Overview

The FasTrak NT Authentication Security uses Windows NT security. NT Security is a feature that is part of the Windows NT Operating System and is also found in Windows 2000 and XP. The server must therefore be installed on a machine running Microsoft Windows NT (3.1 or later), Windows 2000, or Windows XP. The following hardware requirements are recommended.

A personal computer with an Intel Pentium 100 processor or higher.

32 MB or more of RAM.

An 800 X 600 VGA monitor with at least 256 colors.

100 MB free disk space on your hard drive.

Both the FasTrak Security Server (FTSecSvr.exe) and the FasTrak Security Configurator (FTSecCfg.exe) must be installed and configured prior to activating and utilizing these features.

The Security Server Application (FTSecSvr.EXE) handles all client requests to access secured FasTrak features. The server grants or denies access to a feature request depending on the configuration provided by FTSecCfg. When security is enabled for FasTrak applications, all secure features are unaccessible unless security is configured via FTSecCfg and the security server is running. All NT auditing including security and application audits are handled by the server. Security audits are configured in FTSecCfg and application audits are configured in the FasTrak applications that support security.

The following breakdown of steps will aid the user as they go through the installation, configuration, and use of these parts as found in this help manual. They are:

1. Installation- The procedure for installing both the Security Server and Configurator on local and remote machines (the Sever must be installed regardless of operating system and whether it is ran locally or remotely)
2. WorkGroups and Domains- Details to how both relate to NT Security
3. Configuring Users and Groups- Procedures for adding and setting up groups for the variety of supported Operating Systems
4. Configuring User's Rights and Audit Policy- Instructions for configuring specific user rights and audit policy on the machine to which the server will be running from
5. DCOM (Distributed Component Object Module) Configuration- Instructions for configuring DCOM on the available variety of Operating Systems and how to setup the Security Administrator
6. Security Configuration- Specific instructions for how to setup, configure and launch the Security Configurator. Also, instructions for configuring the users, groups and auditing features as well as the Event Viewer, used to view generated logs.

Installing the Security Server

If the FasTrak client applications are installed on a machine other than the one FTSecSvr is running on (remote security), users may want to install FTSecCfg on the client machine(s) as well as the server machine so that the server can be configured locally or from the client machine.

FasTrak security must be installed on a machine running Windows NT (3.1 or later), 2000 or XP. Two applications get installed to a user-selected directory on the local machine. These files are FTSecCfg.exe and FTSecSvr.exe. If running Windows XP, FTSecSvr.exe can only be installed on the Professional version, not on the Home version. Regardless of what Operating System the user has, they still must install the server (FTSecSvr.exe) locally on the client machine as well as on the remote machine. This is true even if the client machine is a non-NT type machine. By default, all "checkable" items get installed including FTSecSvr.exe (under NT Security\Server) and FTSecCfg.exe (under NT Security\Configurator). If installing to Win9x or Me, NT Security\Configurator is not listed and therefore FTSecCfg.exe does not get installed.

Installing to a Local Machine

If the user wishes to install the Security Server locally, this may be achieved by installing from off of CD or from a self-extracting executable file available from FasTrak's Website (http://www.fast-soft.com).

A CD installation requires only that all items for security be checked during the InstallShield process. If a client software application has been previously installed, and it supports NT Security, then re-inserting the installation CD or running the self-extracting executable and un-checking the client during the InstallShield process is all that is required.

Installing to a Remote Machine

Installation begins with the installation of the client (ie. PLC WorkShop) to a local, or client, machine. In this case, installing the Security Configurator is an option.

After this installation is complete, the user must install to the machine which will act as the remote server. Installation is identical to the above with the exception that the client need not be installed and therefore may be unchecked from InstallShield. It is recommended that both the Security Server (FTSecSrv) and Configurator (FTSecCfg) be installed.

If a client software application is already installed to the machine which the user wishes to run as the remote security server, installation of security is identical to installing to a local machine, in the above sectopn.

Installation Dialogs

The following two dialogs are examples of what a user should expect to see during installation. Below is the dialog whereby the user selects to install the client application, security, or both.

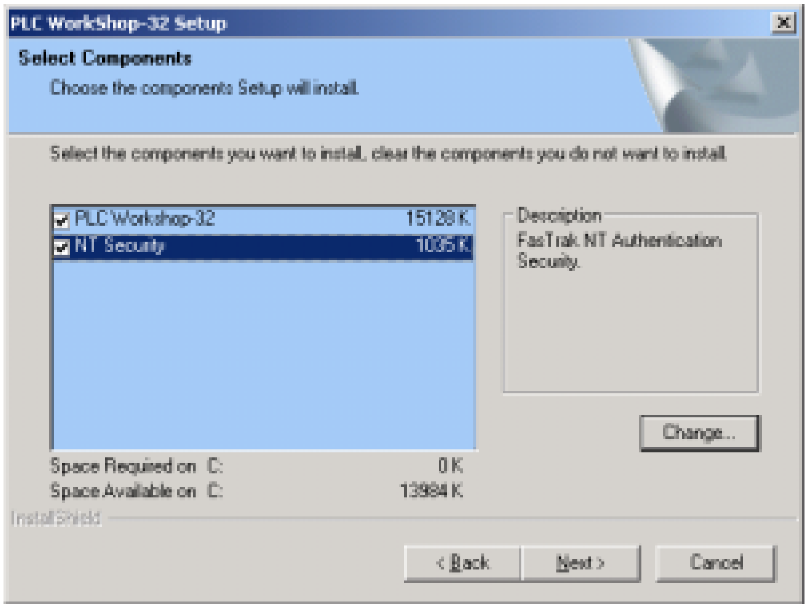


Figure 14.12a

To add or remove which component of security the user wishes, click on the "Change..." button from the dialog above while selecting "NT Security". The next dialog will then open.

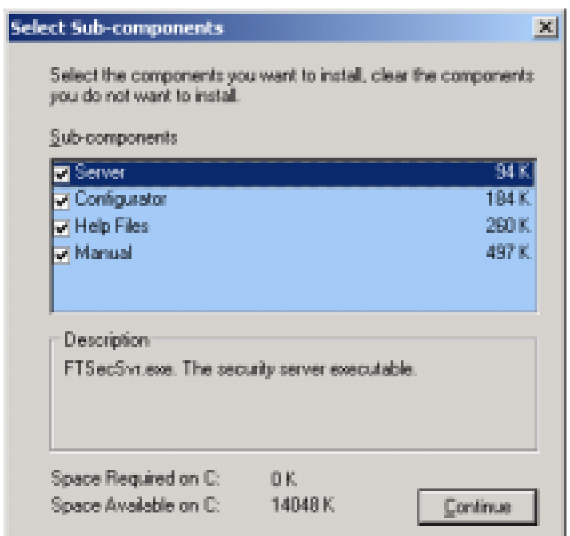


Figure 14.12b

These are the only two dialogs that will be needed during the installation process.

Workgroups and Domains

Workgroups

NT security works slightly different in workgroups than in domains. This difference is only a factor when using the security server remotely. When a computer is part of a Workgroup and a user enters their user name and password to log onto the operating system, the user name they logged in as, the rights they have, and groups they belong to are known only to the local computer.

When a local computer (running a client application such as PLCWorkMod) connects to a remote computer running the security server, it tries to log on to the remote computer with the same user name and password. If the exact user name and password cannot be found on the remote computer, the client (local computer) gets logged onto the remote computer as "Guest". For this reason, a guest user normally has minimal rights. This account is disabled by default. If a guest account is enabled, anyone can log on to the computer because a password is not required for this account. In general, when using workgroups, all users that need to get security clearance from FasTrak's security server must exist on the server machine. The rights a user has on the local machine may differ than the rights they have on a remote machine. Similarly, the groups a user belongs to on the local machine may differ than the groups they belong to on a remote machine.

Domains

When a computer is part of a Domain and a user logs onto any computer that's a member of the domain, the user name and password they supply is stored on the domain controller vs. the local machine. There is no "re-logging" when connecting to a remote machine on a domain. The rights a user has and the groups they belong to are the same amongst all the machines on the domain. This eliminates the need to declare users twice. Once on the local machine and once on the remote machine. When setting up a server on a domain, its important to choose the domain controller machine because this machine holds all the domain users and groups.

Adding Users and Groups to the Operating System

Configuring users and groups differ depending on the operating system your on. Because NT Security is based on Users and Groups, the following must be done.

Adding Users for Windows 95, 98, and ME

In Windows 9x and Me, users may logon to the system to get their unique user profile, but they do not have any rights since the OS is unprotected and they do not belong to any groups. A user can be created simply by entering a unique user name and password when logging on to the OS. A user may skip the login procedure altogether, in which case they become the "Guest" user.

Adding Users for Windows NT

In Windows NT users and groups can be created by entering Administrative Tools from Control Panel and selecting User Manager. This is shown in the figure below.

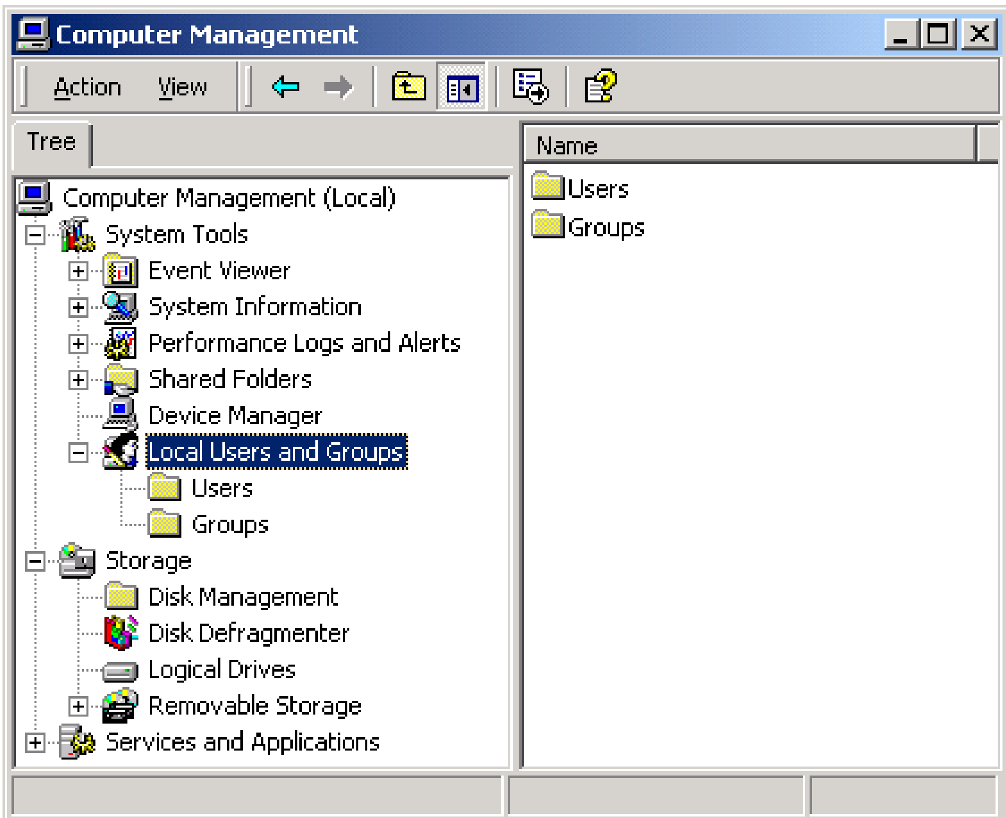


Figure 14.13a

Adding Users for Windows 2000 and XP

In Windows 2000 and XP users and groups can be created by entering Administrative Tools from Control Panel and selecting Computer Management. Within the Computer Management application, users and groups can be created by right clicking on the corresponding folders under the "Local Users and Groups" tree item. This is shown in the figure below.

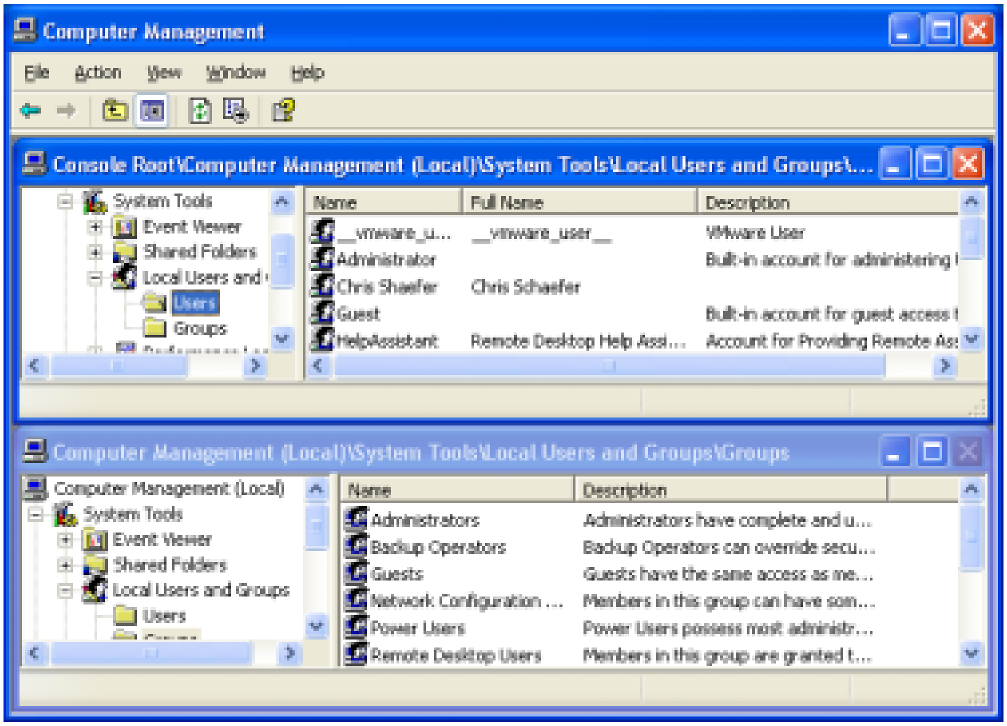


Figure 14.13b

Adding Users and Groups on a Domain Controller

If on a domain controller, select "Active Directory Users and Computers" from the Administrative Tools menu. From the tree control, right click on the "Users" folder to add users and/or groups.

Configuring User Rights and Audit Policy

Correctly configuring specific user rights and audit policy on the server machine is essential for proper security and for the server to operate correctly. The following user rights need to be configured:

Access this computer from the network - Enter all users/groups that will need to access this computer from the network. These users will represent the server's clients that need to request security clearance.

- Deny access to this computer from the network - Make sure that any user/groups that are listed in the above right are not listed here.
- Generate security audits - A user with administrative rights should be added here. This should be the same user that the server is launched under. See DCOM in the Centralized Security Server- Server Configuration of this manual. This right should be configured even if security audits are not used. This Administrator will be referenced later in this manual, in the section dealing with DCOM.
- Manage auditing and security log - A user with administrative rights should be added here. This should be the same user that the server is launched under. See DCOM in the Centralized Security Server- Server Configuration portion of this manual. This right should be configured even if the logs are not used.

Note *this is the same Administrator as mentioned in the above point.*

Configuration location differs depending on the operating system your on. In Windows NT enter the User Manager and choose User Rights from the Policies menu. If interested in generating security audits, select the Audit option from the Policies drop-down menu in the User Manager. From this dialog, check the Success and Failure buttons next to "File and Object Access".

In Windows 2000 and XP, users rights can be modified under Administrative Tools under Local Security Policy. While on a domain controller, users rights may be modified under Administrative Tools which is under Domain Controller Security Policy. From either application, right click on the User Rights Assignment to list all user rights.

To enable security audits, click on Audit Policy from within Local Security Policy under Local Policies. Double click on the Audit Object Access policy and check both Success and Failure boxes. The computer's operating system must have the latest service pack installed as this fixes auditing problems with earlier releases.

The following table may assist a user in determining what rights must be given to a user on the machines involved in FasTrak's NT Authentication Security.

Remote Server Machine	Local Server Machine	Client Machine
Access this computer from the network	Generate Security Audits	N/A
Generate Security Audits	Manage auditing and security log	
Manage auditing and security log		

Note: *It is mandatory that you reboot the machine for the new user rights to take effect.*

Configuring DCOM for NT Authentication Security

The machine that the user wishes to run FTSecSvr (FasTrak Security Server) on must be conigured for proper server/client communication. Below are the needed steps for configuring the server.

Configuring DCOM on Windows 9x, 2000, and NT

DCOM enables FasTrak clients, such as PLC WorkShop for Modicon to communicate with remote security servers. The dialog window, shown below, is launched by running dcomcnfg.exe from a DOS prompt or by clicking on the Windows Start button followed by clicking Run. Changes made in DCOM are applied immediately and there is no need to reboot the PC.

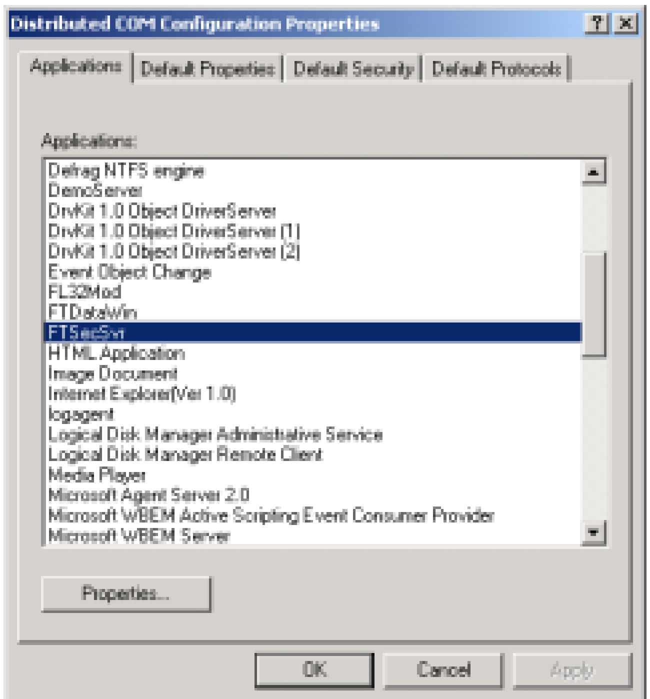


Figure 14.14a

Instructions for Configuration

1. In the Applications listbox, scroll down and select FTSecSvr. If this item is missing, it was not properly installed; refer to earlier sections in this manual.
2. Select the Properties pushbutton. (See Figure 5.14a)
3. The Properties dialog window opens. Click the Security tab. (Figure 5.14e)
4. Under Security, choose the Administrator (from section"Configuring User Rights and Audit Policy) who will have access permission. The following affects both the areas of Access and Launch. For Custom security for Launch and Access Permissions, follow these steps.
   - While in the Properties dialog, with the Security tab being active, click on Edit.
   - Click Add and choose the Administrator.
5. Choose the Identity tab and click the "this user" radio button. (See Figure 5.14f below)
6. Browse and select the Administrator. Click OK to return to the main dialog.
7. Finally, click the Default Properties tab and choose "connect" for Default Authentication and "impersonate" for Default Impersonation. (See Figure 5.14g below)
8. If the server is on a remote machine, then for the client, step 7 from above needs only be repeated.

Configuring DCOM for Windows XP

Configuring DCOM under Windows XP begins with launching the application, dcomcnfg.exe, from a DOS prompt or the Run dialog, just as in the previous instructions. The following dialog launches.

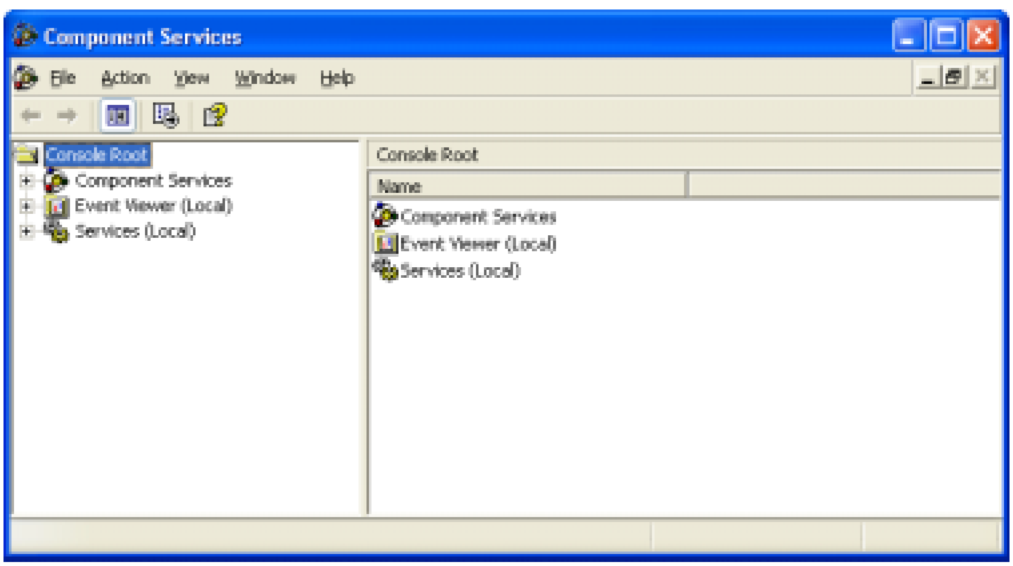


Figure 5.14b

Using the tree structure in the left-most pane, expand the selection of Component services to Computer and then to My Computer, as shown in the dialog window below.

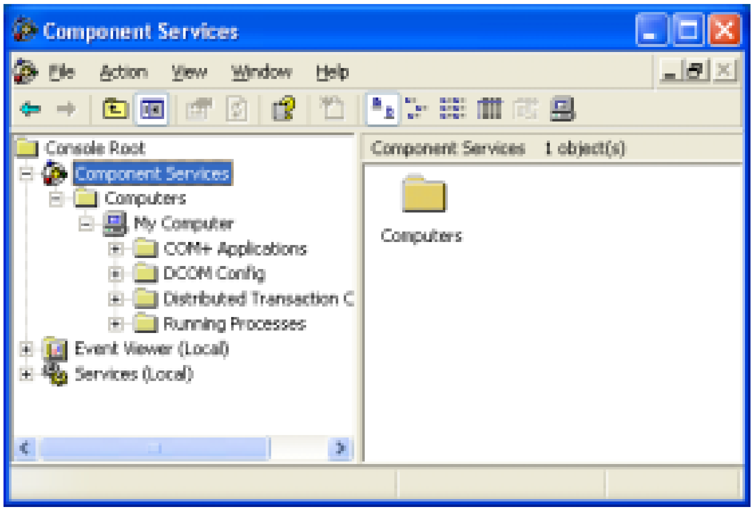


Figure 5.14b

Right-click on My Computer and Select "Properties." In the Default Properties tab, select the default authentication and impersonation levels. This dialog resembles the dcomcnfg interface of Windows NT and 2000, from earlier above.

To set the security and identity properties for FTSecSvr, select DCOM Config under My Computer in the above dialog. The available registered COM applications will be displayed. Scroll and select FTSecSvr.

Open its properties by right-clicking on it. The dialog is shown below.

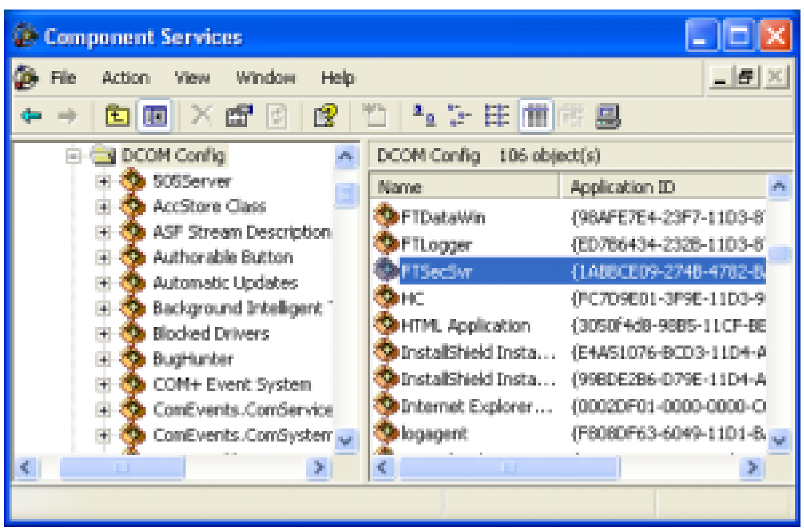


Figure 5.14c

The dialog that opens is identical to the one earlier in this chapter.

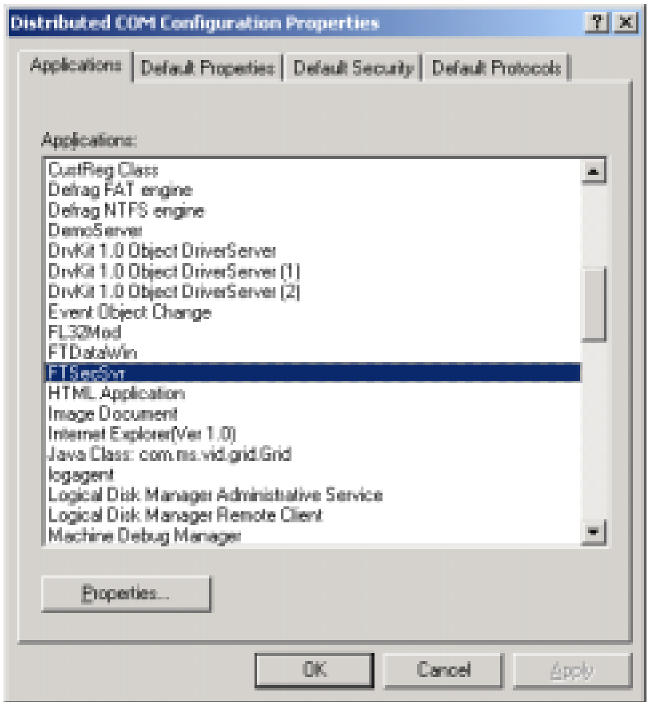


Figure 5.14d

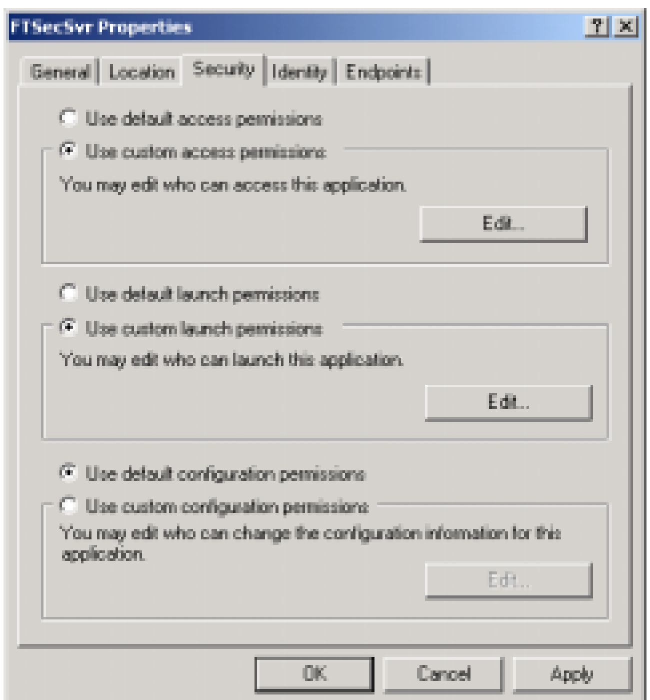


Figure 5.14e

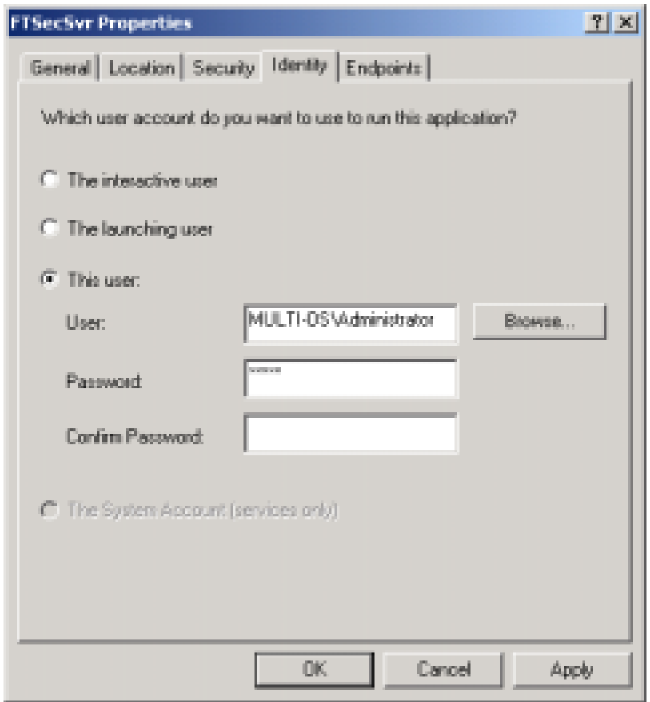


Figure 5.14f

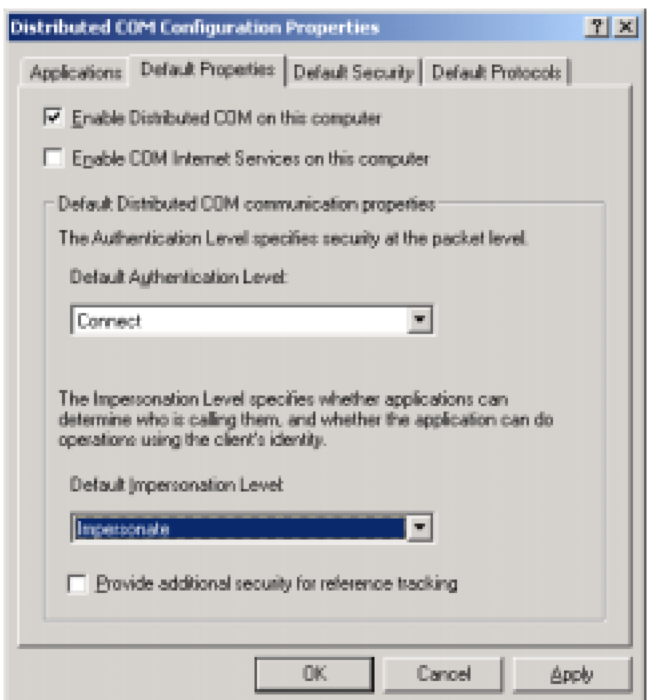


Figure 5.14g

Security Configurator

The Security Configuration Application (FTSecCfg.EXE) allows users with administrator rights to configure their centralized server based security. This .EXE will reside on the same machine and directory as the server. Users will also be able to install FTSecCfg on client machines running Windows NT (3.1 or later)/2000/XP so that they can configure their security server remotely. The following features are supported.

1. Ability to configure which users and or groups have access to various FasTrak features. This includes users and groups from the machine running the server. Separate secure features will exist for individual FasTrak applications including programming packages as well as Controlshop applications.
2. Security Auditing can be configured for each specific feature on an individual or group basis. Audit information will be logged to the "Event Viewer" in the "Security Log" section.
3. A link to the Event Viewer will be provided allowing quick access to the Security Log. The event viewer can also be accessed under "Administrative Tools" in Control Panel.

Launching the Security Configurator

The Configuration Application may be launched from either the Start menu in Windows or from within the FasTrak client should it be installed.

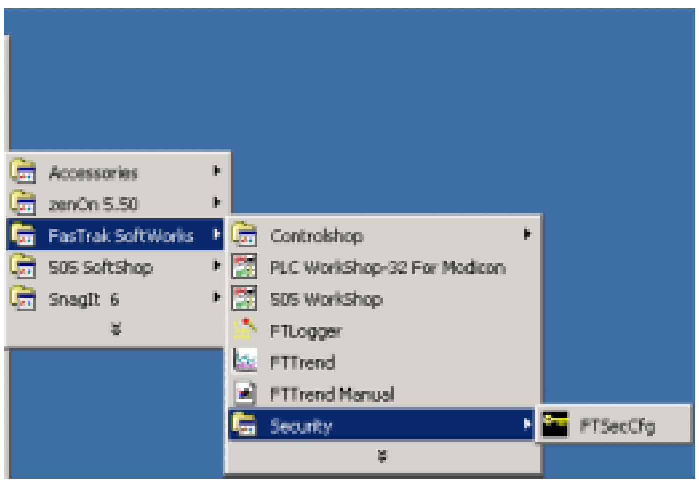


Figure 14.15a

The Security Configurator application will launch and (if the user is not in the client), at the same time, the Security Server Location dialog may open. Both are shown below.

When launched from the Start menu, the Security Server Location dialog will prompt the user to select the machine that the server will run on.

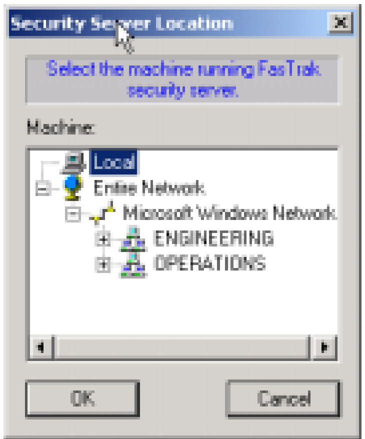


Figure 14.15b

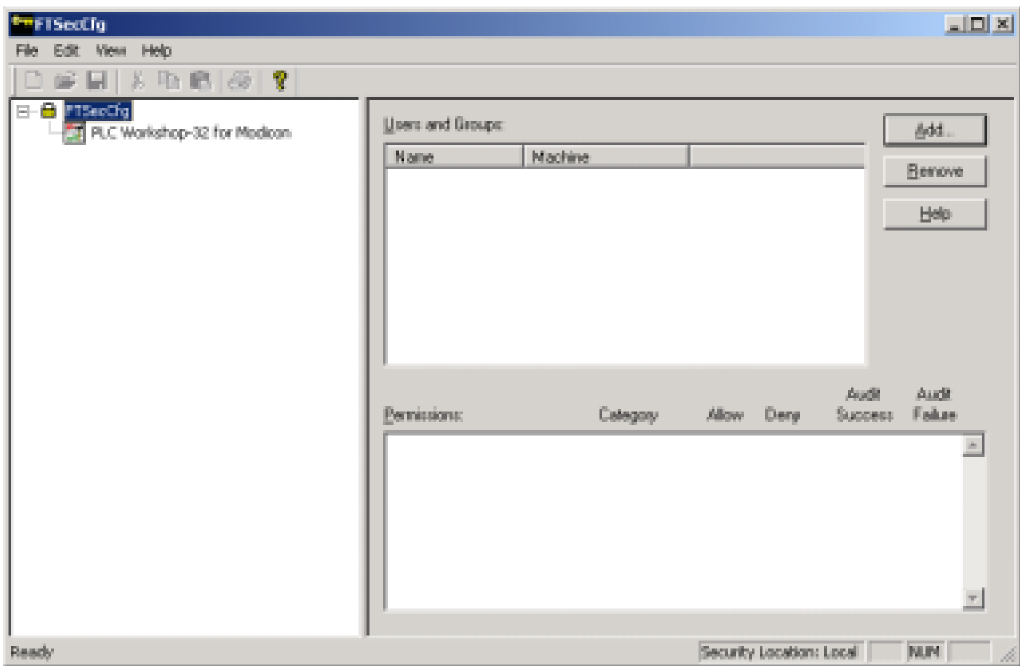


Figure 14.15c

Configuring Users, Groups, and Security Audit

After having configured security and chosen a server location, individuals Users, Groups of Users, and permissions may be set from within the FTSecCfg utility. The dialog below will be referenced in this manual with regards to these features.

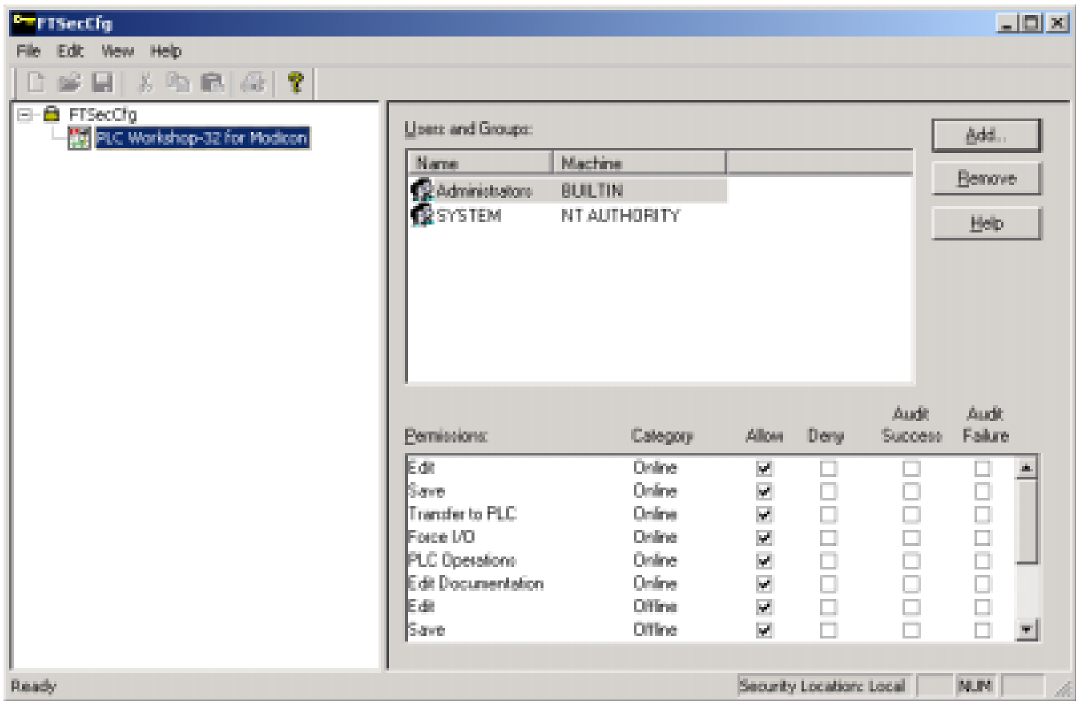


Figure 14.16

The selected server machine appears in the status bar in the lower right portion of the application window.

In the tree control on the left-most pane, pick "PLC Workshop-32 for Modicon". In the "Users and Groups" list box, two names (System and either the administrators group or the current user) will exist by default with access to all features.

Adding and Removing Users and Groups

To Add or Remove individual Users or Groups, use the pushbuttons on the far right of this dialog.

Add additional users and/or groups by selecting the "Add…" button. Remove users or groups by highlighting the name and selecting the "Remove" button.

Users and groups that are not displayed in the list box are automatically denied access to all features.

The list box is sorted with groups listed first followed by users. A group is identified with the two person icon to the left while the user is identified with the one person icon. A combination of around 16 users and groups can be configured for an application.

Setting Permissions

The lower list box in the right pane reflects the access rights and audits for individual permissions based on the highlighted name in the "Users and Groups" list box. When the "Allow" box is checked the user or group will be given access by the server when requested from the client (PLCWorkMod for example). When the "Deny" box is checked the user or group will be denied access by the server when requested from the client.

In cases where a user has conflicting rights on a permission, the denied permission will always take precedence and the user will not be able to access the feature. one such example would be a user is allowed access to a specific permission but a group they belong to is denied access to the same permission. If neither box is checked, the user will not be granted access to the feature unless one of the groups they belong to has access to the feature

Security Audits

Security audits for individual features can be performed on user and groups by selecting the "Audit Success" and/or "Audit Failure" boxes. These audits will appear in the security log on the server machine and can be accessed using Microsoft's Event Viewer. See the topic entitled "Event Viewer" or click here for more details on the Event Viewer. A "Success" audit will appear in the security log when a user is given access to a secure feature and the "Audit Success" box is checked for that feature. A "Failure" audit will appear in the security log when a user is denied access to a secure feature and the "Audit Failure" box is checked for that feature.

All security changes in FTSecCfg are accumulated and do not get committed to the server until confirmation upon exit of FTSecCfg.

Event Viewer

The Event viewer is a Windows application for displaying application, security, and system logs. FasTrak uses and writes to this log to hold audits. It is launched from within the Security Configurator. To launch the Event Viewer, click on View, Event Viewer as shown in the figure below. The Event Viewer is also accessible from the Administrative Tools of Window's Control Panel.

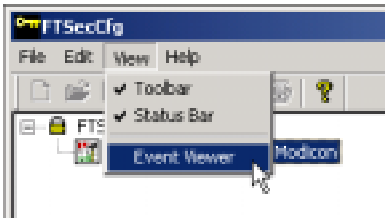


Figure 14.17a

The following window opens. Double-clicking an entry in the right-most pane above will display more information for the item.

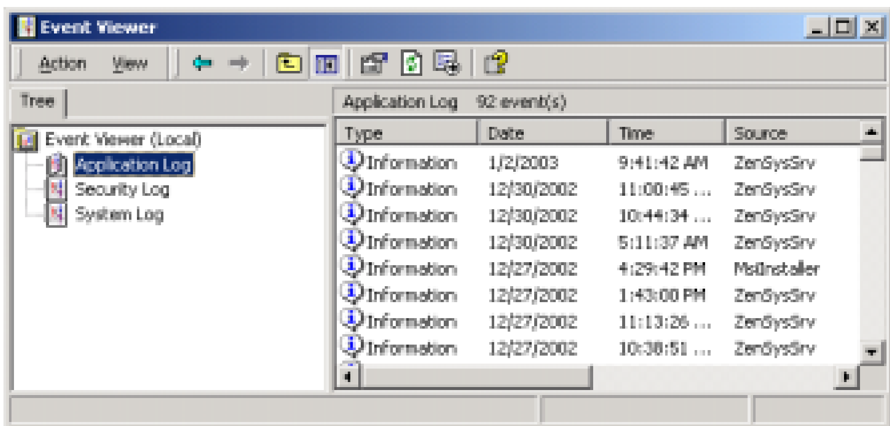


Figure 14.17b

From this application, the user may view the Security log. If running the Security Configurator from a different machine than the security server (FTSecSvr), then the security server's machine name must be specified in the event viewer to view the security log entries referring to FasTrak security. This can be accomplished by highlighting the "Event Viewer" in the tree control and then picking the "Action" menu. This is in the figure below.

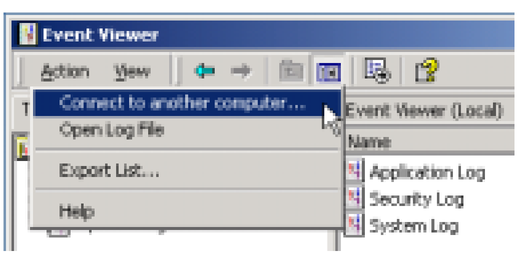


Figure 14.17c

Selecting "Connect to another computer..." launches the following window.

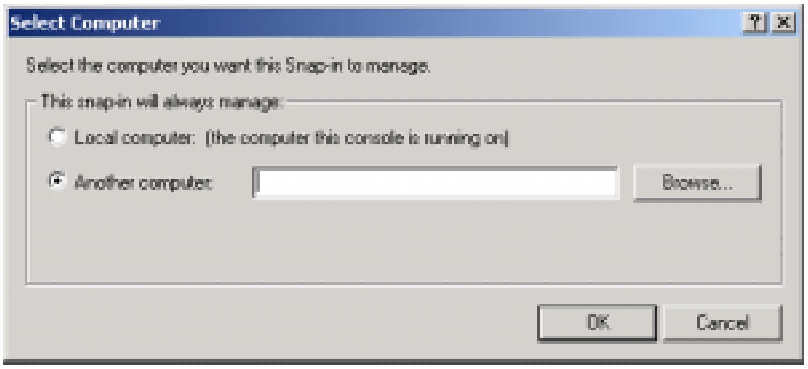


Figure 14.17d

Use the Browse button for searching for and selecting another computer. Once chosen, click OK to accept the selection.

Index

(

.

5

A

B

C

F

M

O

P

R

T

U

V

W