Preface

Read this preface to familiarize yourself with the rest of the manual. This preface covers the following topics:

- who should use this manual
- the purpose of this manual
- conventions used in this manual

Who Should Use this Manual

The APS Quick Start for the New User is intended as an introduction of APS software to first-time users. The simple tasks and practice exercises in this manual do not include important user information for actual control applications.

Purpose of this Manual

This manual is an introductory document, designed to allow you to install APS and begin programming in the shortest time possible. It does this by focusing on a simple controller and a simple program. Basic concepts are presented, but only with enough detail to get you started and let you know that there is more to be learned. Read chapter 1 first. It will acquaint you with the rest of the manual.

Contents of this Manual

Chapter	Title	Purpose
	Preface	Describes the purpose, background, and scope of this manual. Also specifies the audience for whom this manual is intended.
1	Setting up Your Equipment	Lists hardware requirements and shows you how to set up a controller, connect your PC to the controller, and install APS software on your PC.
2	Control Basics	Presents basic information you will need to know before you can begin programming with APS.
3	Creating a Processor File	Shows you how to create a processor file, enter a ladder program and add a rung comment.
4	Online Operations, Quick Edit	Shows you how to restore (download) your processor file to the controller, monitor and test the program, and use quick edit.
5	Creating and Printing Reports	Guides you through creating and printing reports. These include program listing, cross reference, processor configuration, and data tables.
Appendix A	Additional Ladder Program Exercises	Introduces you to branching of instructions and the timer instruction.
Appendix B	Troubleshooting Errors	Provides a listing of error messages that you may encounter while working through the manual. Also, offers possible solutions for these errors.
	Glossary	Provides a listing of terms used throughout this manual.

Related Documentation

The table below is a partial list of publications that contain information about installation, programming, and operation of SLC 500 controllers. To obtain a copy, contact your local Rockwell Software office or distributor.

For	Read this Document
A description on how to install and use your *Modular* SLC 500 programmable controller	Installation & Operation Manual for Modular Hardware Style Programmable Controllers, Publication Number 1747-6.2
A description on how to install and use your *Fixed* SLC 500 programmable controller	Installation & Operation Manual for Fixed Hardware Style Programmable Controllers, Publication Number 1747-NI001
A procedural manual for technical personnel who use APS to develop control applications	Advanced Programming Software (APS) User Manual
A reference manual that contains status file data, instruction set, and troubleshooting information about APS	Advanced Programming Software (APS) Reference Manual
A training and quick reference guide to APS	SLC 500 Software Programmer's Quick Reference Guide, Publication Number ABT-1747-TSG001—available on PASSPORT at a list price of $50.00
A guide of common procedures used in APS	SLC 500 Software Common Procedures Guide, Publication Number ABT-1747-TSJ50—available on PASSPORT at a list price of $50.00
A complete listing of current Allen-Bradley documentation, including ordering instructions. Also indicates whether the documents are available on CD-ROM or in multi-languages.	Allen-Bradley Publication Index, Publication Number SD499
A glossary of industrial automation terms and abbreviations	Allen-Bradley Industrial Automation Glossary, Publication Number AG-7.1

Common Techniques Used in this Manual

The following conventions are used throughout this manual:

- Bulleted lists such as this one provide information, not procedural steps.
- Numbered lists provide sequential steps or hierarchical information.
- *Italic* type is used for emphasis.
- Text in `this font` indicates words or phrases you should type.
- Key names match the names shown and appear in bold, capital letters within brackets (for example, **[ENTER]**). A function key icon matches the name of the function key you should press, such as 

.

1 Setting Up Your Equipment

This chapter briefly describes hardware requirements and SLC 500™ controller styles, then shows you how to set up your equipment in preparation for the exercises in later chapters. Topics include:

- hardware requirements
- controller styles
- setting up a demo unit
- setting up a field-wired controller
- connecting the controller to a personal computer
- installing the software
- running APS
- APS display format

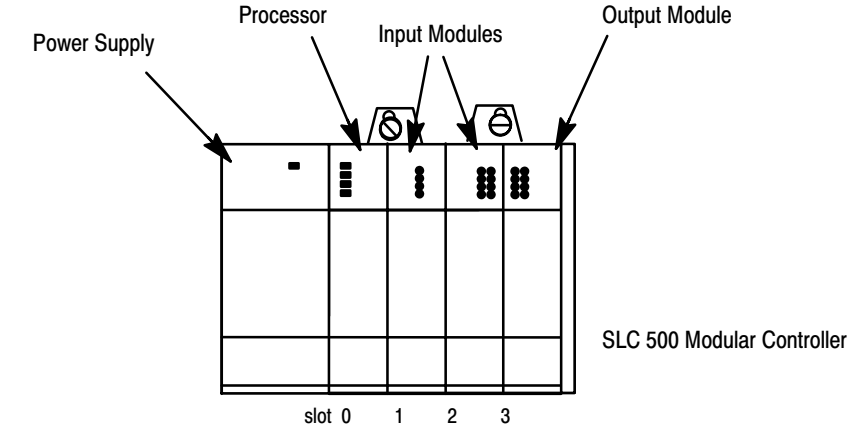
Hardware Requirements

To perform the tasks provided in this manual we recommend the following hardware:

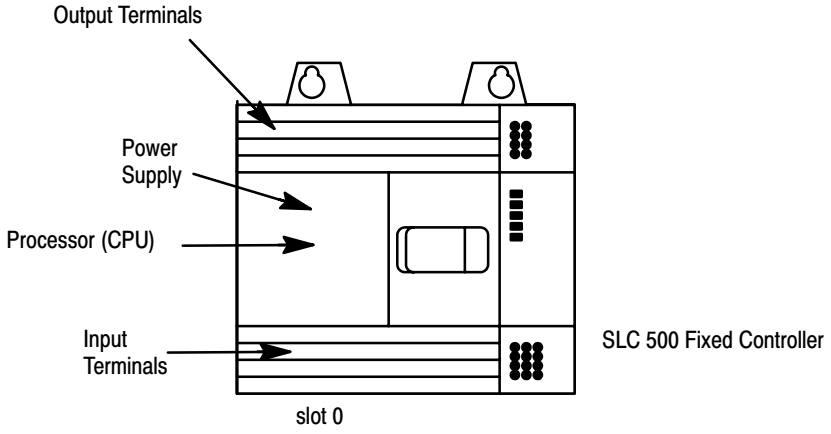
- An SLC 500 modular or fixed controller with external inputs and outputs. An SLC 500 demo unit would be ideal. The programs and examples used in this manual are based on using a modular controller demo unit (catalog number 1747-DEMO 3 or 1747-DEMO 4).
- A compatible personal computer (PC). Refer to the *Advanced Programming Software User Manual* for more information.
- An RS-232/DH-485 Interface Converter (catalog no. 1747-PIC).
- A communications cable for connecting the Interface Converter to the controller (catalog number. 1747-C10). This cable is supplied with the Interface Converter.
- A compatible printer, if you choose to use the “Print Reports” capability described in chapter 5.

Controller Styles

The SLC 500 comes in two different styles: modular and fixed. These styles are illustrated below. The modular controller consists of a chassis, power supply, processor (CPU), and Input/Output (I/O) modules. The fixed controller consists of a power supply, processor (CPU), and a fixed number of I/O contained in a single unit. An expansion chassis can be added to the fixed controller.



SLC 500 Modular Controller



SLC 500 Fixed Controller

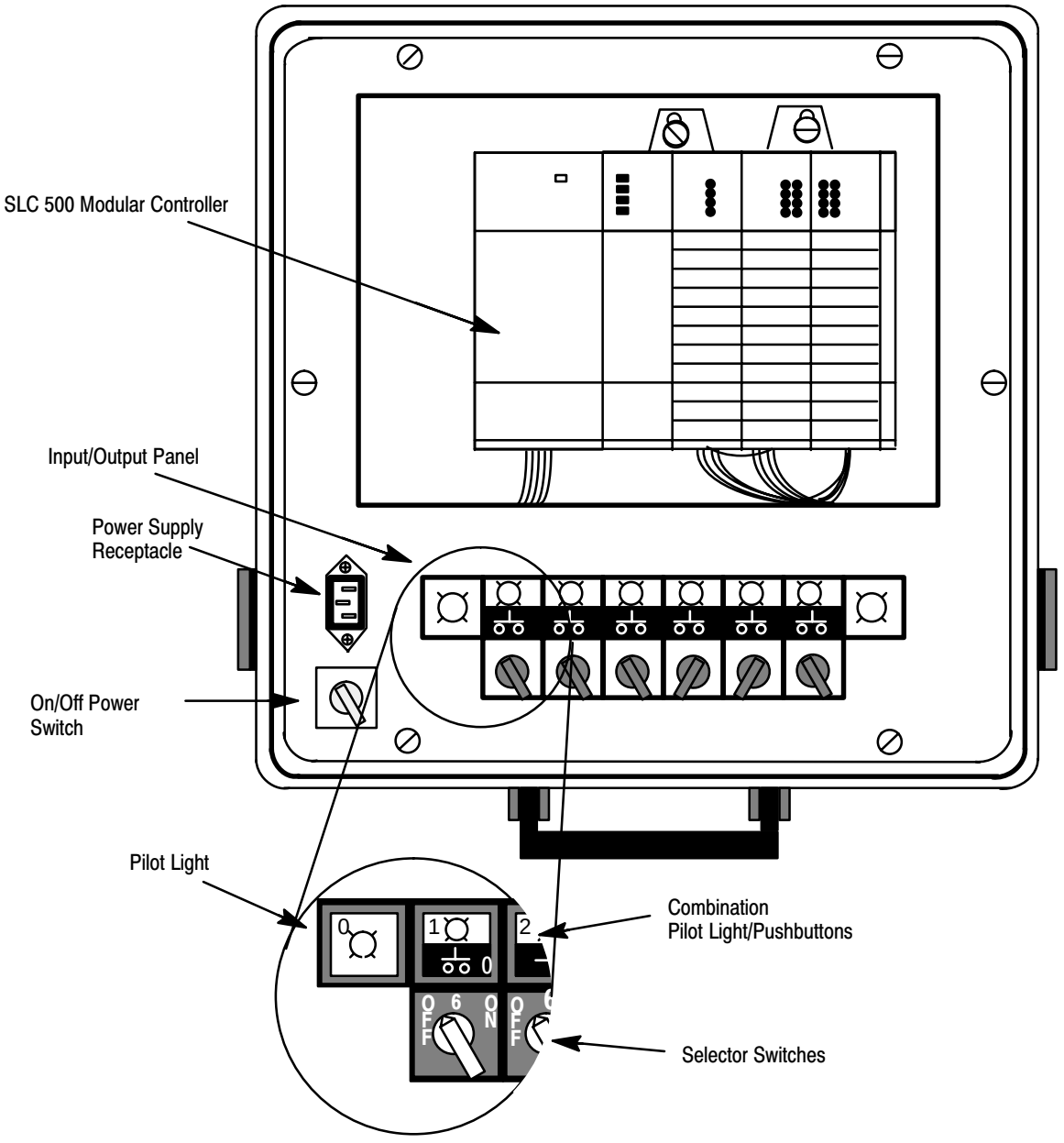
Further information on hardware is found in the Installation and Operation Manuals, catalog number 1747-NI001 (fixed controllers) and publication number 1747-6.2 (modular controllers).

Setting Up a Demo Unit

SLC 500 demo units are available with either a fixed controller or modular controller. This manual assumes you are using a modular controller demo unit for all the programming exercises. If you use a fixed controller demo unit, you will need to use different configuration information and I/O addresses in the exercises. This is explained later.

The figure that follows shows an SLC 500 modular controller demo unit. It is completely wired, with 12 external inputs (6 push buttons and 6 selector switches) and 8 external outputs (pilot lights).

In setting up your system, place the demo unit near your personal computer. Note the On/Off Power Switch and the Power Supply Receptacle on the demo. Make certain that the power switch is Off, then insert one end of the power cord into the power supply receptacle and the other end into an electrical socket.


SLC 500 Modular Controller
Input/Output Panel
Power Supply
Receptacle
On/Off Power
Switch
Pilot Light
Combination
Pilot Light/Pushbuttons
Selector Switches

Setting Up a Field-Wired Controller

The details of installing and wiring the controller and external input/output devices are beyond the scope of this manual.

If you are using a field-wired fixed or modular controller, refer to the Installation and Operation Manuals, catalog number 1747-NI001 (fixed controllers) and publication number 1747-6.2 (modular controllers), for information on installation and wiring of the controller and external input/output devices.

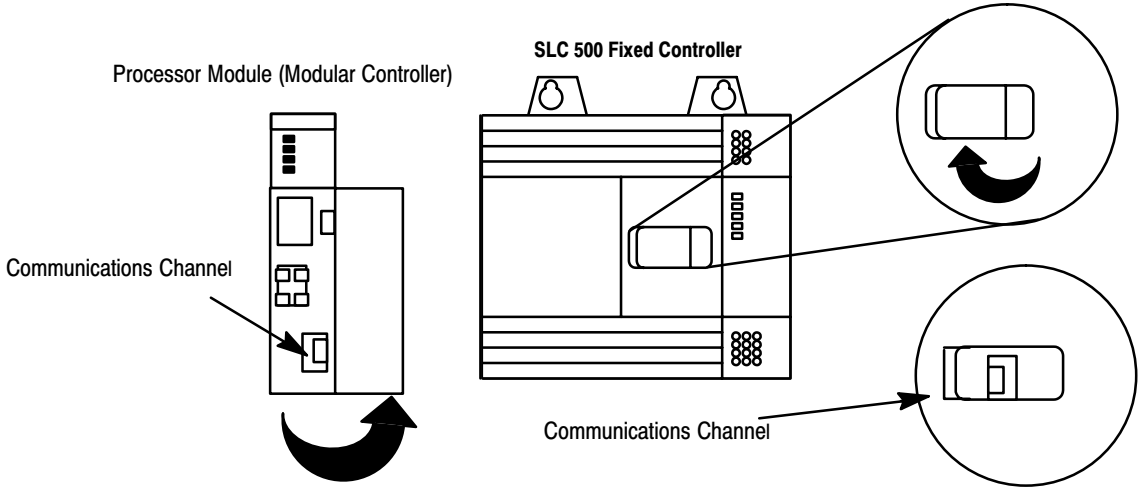
We recommend that your controller have at least two external input devices and two external output devices connected to complete the exercises in this manual.

Connecting the Controller to a Personal Computer

To connect the controller to a personal computer, you need a communications cable, catalog number 1747-C10, and an RS-232/DH-485 interface converter, catalog number 1747-PIC.

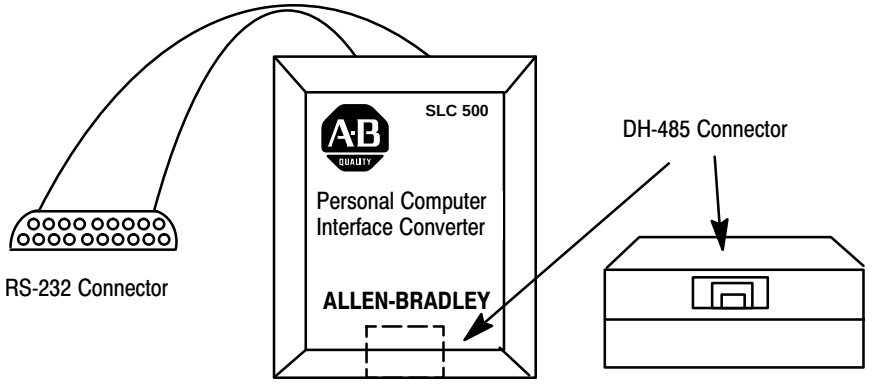
Note *This procedure does not apply to SLC 5/04 processors. For these processors, you need to connect a 1747-CP3 cable from channel 0 of the processor to the personal computer serial port, or you need to use a KT, KTX, KT2, or PCMK card.*

1. Locate the communications channel of the controller. The figure below shows where it is located on modular and fixed controllers.



2. Insert one end of the 1747-C10 cable into the communications port of your controller.

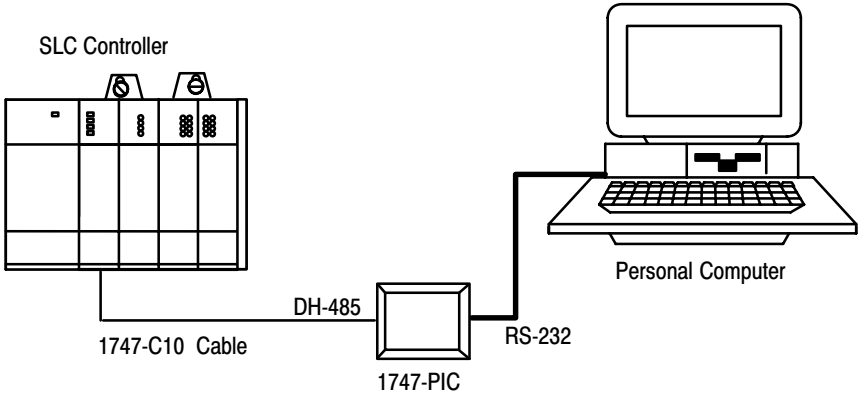
3. Insert the other end of the 1747-C10 cable into the DH-485 connector of the 1747-PIC interface converter. The DH-485 connector is shown in the following illustration.



4. Insert the RS-232 connector (see figure above) of the interface converter into the serial communication port of your computer.

If your computer has a 9-pin serial port, use the 9–25 pin adapter provided with the interface converter.

The figure below shows a modular controller connected to a personal computer.



Installing the Software

Before installing the software, make sure your personal computer meets the memory and RAM requirements needed to run APS. Refer to the A*dvanced Programming Software User Manual* for more detailed information on personal computer requirements and APS software installation.

Locating the Software's Serial Number

During the installation process you are asked for the serial number of your software. The serial number you enter is used to personalize the software.

The serial number is not found on the disks. It can be found in several places though. These are:

- the software registration card
- the registration change card
- the outside of the shipping carton

Note *If you enter the serial number incorrectly or enter the wrong serial number and accept the entry, you will be unable to correct this situation later. Therefore, verify your serial number carefully, before committing your work.*

Installing the Software

To install the software, do the following:

1. Insert the diskette labeled Disk 1 into the appropriate disk drive (either drive A or drive B). For this example, we are using drive A.
2. Type: **A:INSTALL**, then press **[ENTER]**.
3. During the installation process, instructions appear on the screen to prompt you through the procedure. Follow the instructions and type in the information requested.

Note *You can install APS to run in a windows environment, however, that procedure is beyond the scope of this manual. See the Advanced Programming Software User Manual for this information.*

Running APS

To run APS, follow these steps;

1. If necessary, change the drive specified to the drive where the software is installed (typically C). To do this, type:

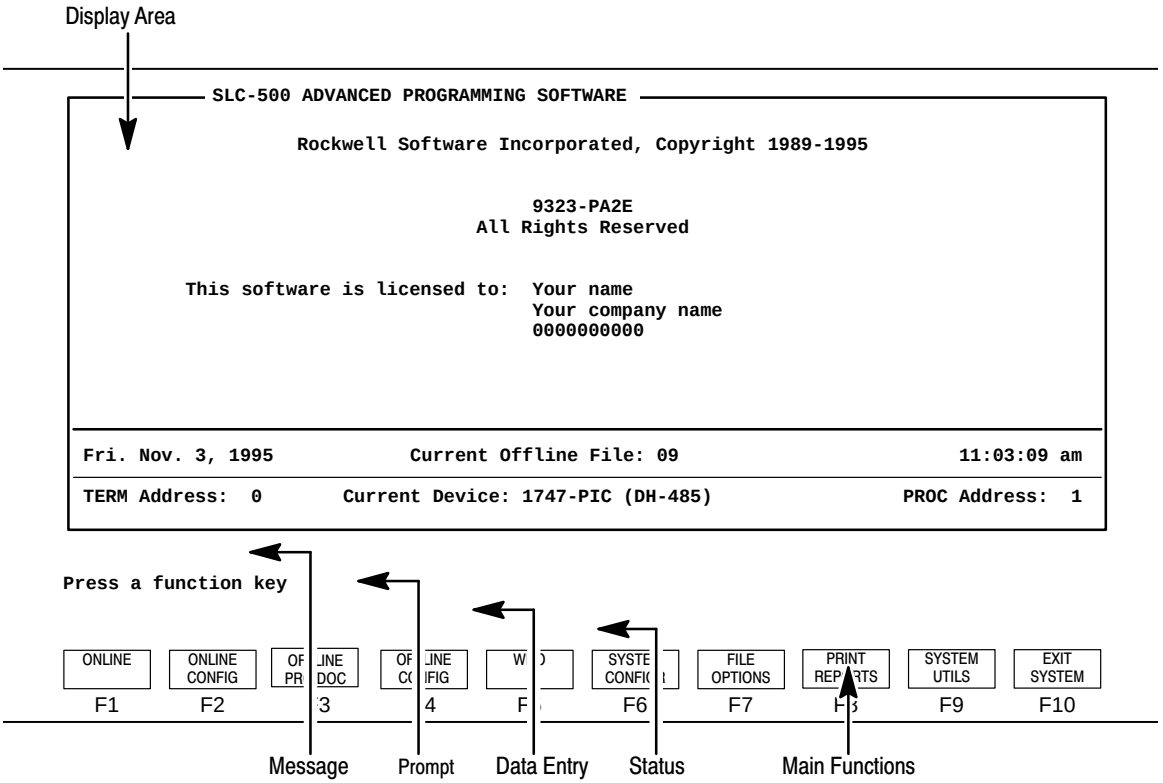
   C: and press **[ENTER]**.

2. If you are using the default directory, at the DOS prompt, type:

   CD \IPDS\ATTACH\SLC500 and press **[ENTER]**.

   If you specified a different directory path, enter that directory instead.

3. Type: **AP** and press **[ENTER]**. The main APS menu appears.



Exiting the System: You can exit APS software and return to DOS by accessing the APS menu, shown above, and pressing EXIT SYSTEM F10 .

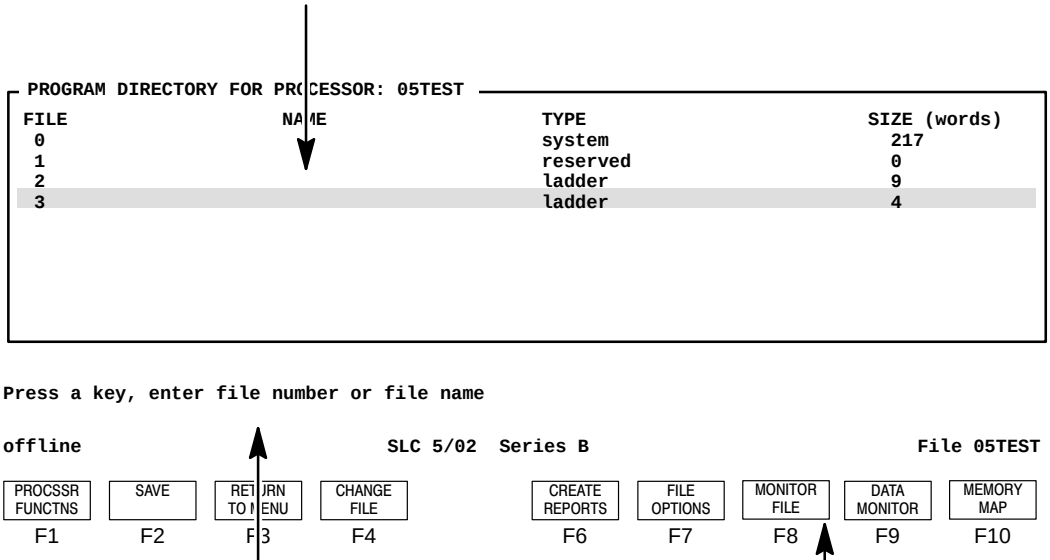
APS Display Format

The APS screen is divided into three areas:

- display area
- message, prompt, data entry, and status lines
- main functions

The following figure indicates what appears in these areas.

- Display Area: The five APS displays appear here - the APS menu, offline program directory (shown here), offline monitor file, online program directory, and online monitor file. Various option windows overlay these displays, depending on the function you are accessing. Also, the status data file and other data files, force tables, and the memory map appear in this area.



- Message, Prompt, Data Entry, and Status Lines

  Message: Error codes/descriptions and information concerning terminal or processor operation appears here.

  Prompt: Indicates action you should take.

  Data Entry: Information entered from the terminal keyboard appears on this line.

  Status: Status information concerning the processor and program file appears on this line.

- Main Functions: APS functions appear here. They are accessed by terminal keys F1 to F10. Pressing a main function key calls up sub-functions and sometimes option windows.

2 Control Basics

This chapter introduces you to basic concepts essential for understanding how the SLC 500 controller operates. It covers:

- SLC 500 file concepts
- how external I/O devices communicate with the processor
- addressing external I/O
- ladder logic concepts

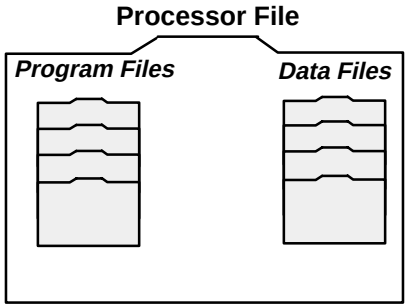
SLC 500 File Concepts

The CPU, or processor, provides control through the use of a program you create. The program you create is called a processor file. This file contains other files that break your program down into more manageable sections. These sections are:

- Program Files - provide storage and control of the main program and subroutines.
- Data Files - contains the status of inputs, outputs, the processor, timers, counters, and so on.

Processor Files

Each CPU can hold 1 processor file at a time. The processor file is made up of program files (up to 256 per controller) and data files (up to 256 per controller).



Processor files are created in the offline mode using APS. These files are then restored, also referred to as downloaded, to the processor for online operation.

Program Files

Program files contain controller information, the main control program, and any subroutine programs. The first three program files are required for each processor file. These are:

- **File 0 - System Program**

  This file stores the controller configuration and other system information.

- **File 1**

  This file is reserved for internal controller use.

- **File 2 - Main Ladder Program**

  This file stores the main control program.

- **Files 3 - 255 - Subroutine Ladder Program**

  These files are optional and used for subroutine programs.

Most of your work with program files will be in file 2, the main program file. This file contains your ladder logic program which you create to control your application.

Data Files

Data files contain the data associated with the program files. Each processor file can contain up to 256 data files. These files are organized by the type of data they contain. Each piece of data in each of these files has an address associated with it that identifies it for use in the program file. For example, an input point has an address that represents its location in the input data file. Likewise, a timer in the timer data file has an address associated with it that allows you to represent it in the program file.

The first 9 data files (0 - 8) have default types. You designate the remainder of the files (9 - 255). The default types are:

- **File 0 - Output Data**

  This file stores the status of the output terminals for the controller.

- **File 1 - Input Data**

  This file stores the status of the input terminals for the controller.

- **File 2 - Status Data**

  This file stores controller operation information.

- **Files 3 - 7**

  These files are pre-defined as Bit, Timers, Counters, Control, and Integer data storage, respectively.

- **File 8 - Float Data**

  This file is used by SLC 5/03™ OS301, OS302 processors and SLC 5/04™ OS400, OS401 processors for Float data storage.

- **Files 9 - 255**

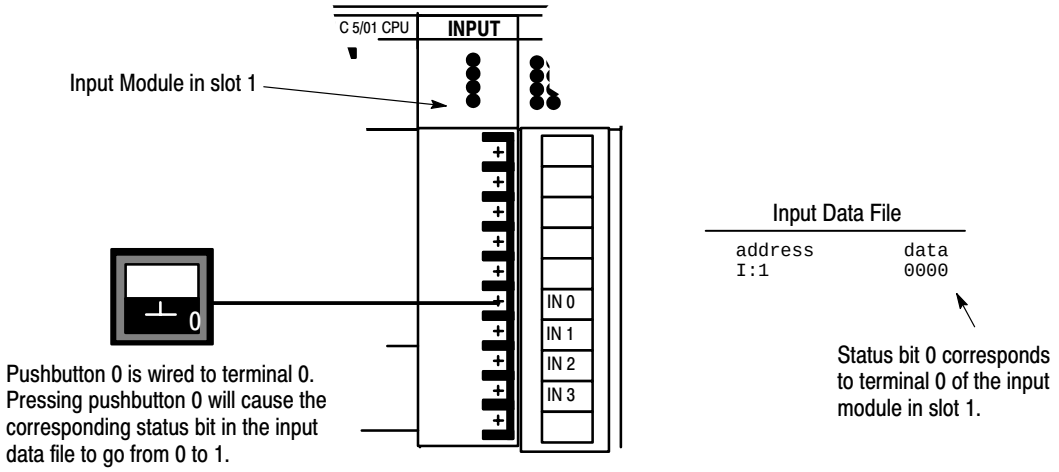
  These files are user-defined as Bit, Timer, Counters, Control, Integer, Float, String, and ASCII data storage.

Most of your work with data files will be in files 0 and 1, the output and input files. Refer to appendix A for an example of the Timer data file.

How External I/O Devices Communicate with the Processor

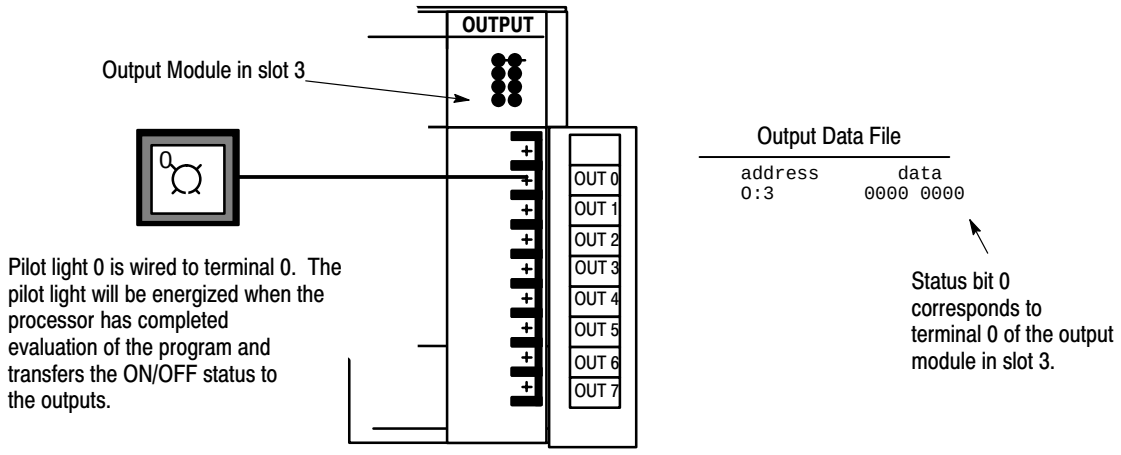
The figures that follow apply to a modular controller demo unit having an input module in slot 1 and an output module in slot 3. See page 1–3 for a diagram of the slot location. To simplify the illustration, only pushbutton 0 and pilot light 0 of the external I/O are shown.

Each of the external input circuits is represented by a status bit in the input data file of the processor file. Each of the external output circuits is represented by a status bit in the output data file of the processor file. During controller operation, the processor applies the input data to the program, solves the program based on the instruction you enter, and energizes and de-energizes external outputs.



Closing an external input circuit changes the corresponding status bit from 0 to 1.

Opening an external input circuit changes the corresponding status bit from 1 to 0.



When an output data file status bit is a 1, the corresponding external output circuit will be energized (ON).

When an output data file status bit is a 0, the corresponding external output circuit is de-energized (OFF).

Addressing External I/O

As pointed out in the last section, external inputs and outputs are linked to the input data file and output data file of the processor file. Each status bit in these files has an address. You specify the appropriate address when you enter an instruction in your ladder program.

For our purposes, input addresses have the form **`I:e/b`**

where

`I`= Input data file

`:`= Element or slot delimiter

`e`= Slot number of the input module

`/`= Bit or terminal delimiter

`b`= Terminal number used with input device

Similarly, output addresses have the form **O:e/b**

where

O= Output data file

: = Element or slot delimiter

e= Slot number of the output module

/= Bit or terminal delimiter

b= Terminal number used with output device

Examples:

I:1/0= Input, slot 1, terminal 0

I:2/0= Input, slot 2, terminal 0

O:3/0= Output, slot 3, terminal 0

O:3/7= Output, slot 3, terminal 7

O:0/7= Output, slot 0, terminal 7 (fixed controllers only because of slot 0)

I:0/4= Input, slot 0, terminal 4 (fixed controllers only because of slot 0)

Eventually, you will be addressing other data files, such as Status, Bit, Timer, Counter, Control, Integer, String, ASCII, and Float. Addressing of these files is discussed in the APS programming manual.

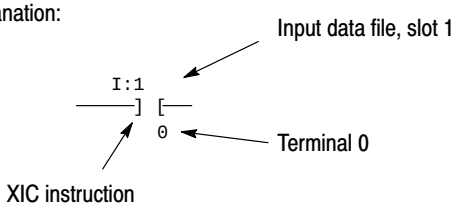
APS Display of Instructions/Addresses

APS displays I/O addresses as shown below.

When you enter an XIC instruction (defined later) and the address I:1/0, APS will display the address with the instruction as follows:

```
   I:1
———] [——
     0
```

Explanation:

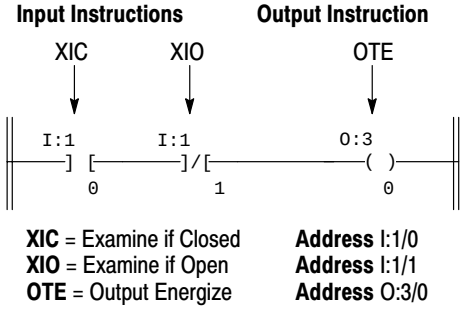


Ladder Logic Concepts

As we mentioned earlier, the program files you create contain the program used for your controlling application. The programs are written in a programming language called Ladder Logic. This name is derived from its ladder-like appearance.

A ladder logic program consists of a number of rungs, on which you place instructions. Instructions each have a data address associated with them and based on the status of these instructions the rung is solved.

The figure below shows a simple 1-rung ladder program. The rung includes two input instructions and an output instruction. Note, in the example below each instruction has a name (Examine if Closed), a mnemonic (XIC), and an address (I:1/0).



A simple rung, using bit instructions.

True/False Status

The data file bits that these instructions are addressed to will be either a logic 0 (OFF) or a logic 1 (ON). This determines whether the instruction is regarded as "true" or "false":

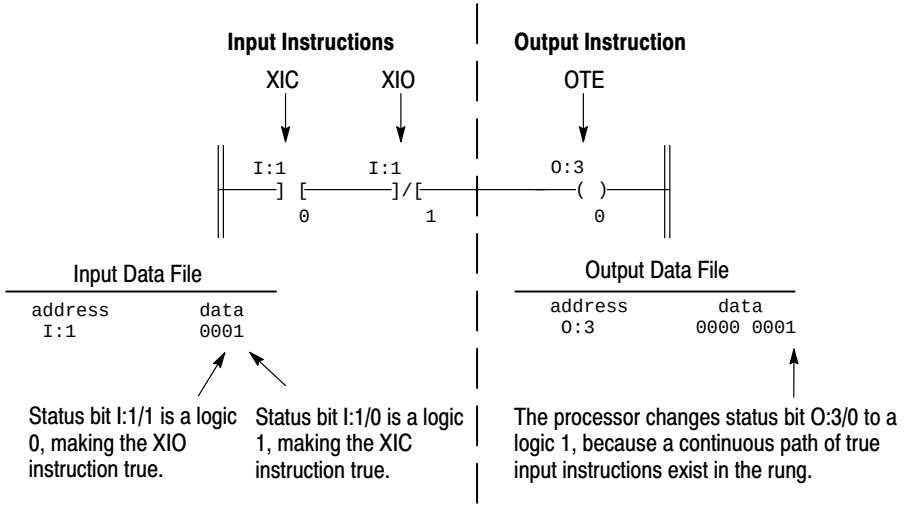
If the data file bit is	The status of the instruction is		
	XIC Examine if Closed —] [—	XIO Examine if Open —]/[—	OTE Output Energize —( )—
Logic 0	False	True	False
Logic 1	True	False	True

Logical Continuity

During controller operation, the processor evaluates each rung, changing the status of instructions according to the logical continuity of rungs. More specifically, input instructions set up the conditions under which the processor will make an output instruction true or false. These conditions are:

- When the processor finds a continuous path of true input instructions in a rung, the OTE output instruction will become (or remain) true. We then say that “rung conditions are true”.
- When the processor does *not* find a continuous path of true input instructions in a rung, the OTE output instruction will become (or remain) false. We then say that “rung conditions are false”.

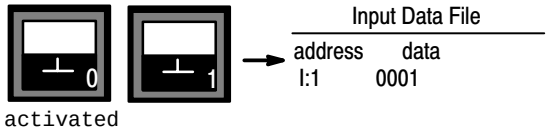
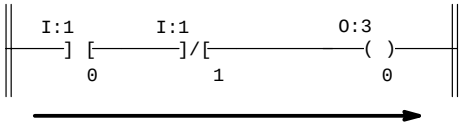
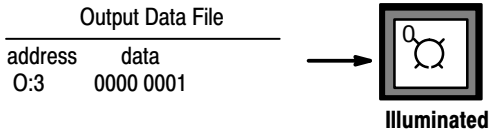
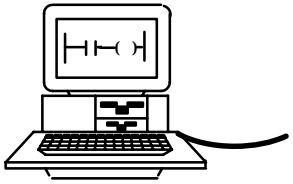
The figure below indicates the data file conditions under which the rung is true:



In the above example, if the input data file was 0000, then the rung would be false and the output data file would read as 0000 0000.

The Processor Operating Cycle

The diagram below indicates the events that occur during the processor operating cycle. This sequence is repeated many times each second.

Event	Description	
Input Scan		The status of external input circuits is read. The input data file is updated with this information.
Program Scan		The ladder program is executed. The input data file is evaluated, the ladder rung is solved, and the output data file is updated.
Output Scan		The output data file information is transferred to the external output circuit, thus energizing or de-energizing it.
Communications		Communications with the programming terminal and other network devices takes place.
Housekeeping	Processor internal housekeeping takes place.	

3 Creating a Processor File

In this chapter you create a processor file. The tasks you will perform:

- For modular controllers: Make a record of the processor module catalog number, the chassis catalog number(s), the I/O module catalog numbers, and the slot locations of I/O modules.
- For fixed controllers: Make a record of the controller catalog number (and I/O module catalog numbers and slot locations if you are using the 1746-A2 expansion chassis).
- Run APS software and initiate the creation of a processor file.
- Name the processor file GETSTART.
- Enter the controller configuration.
- Enter a 1-rung ladder program.
- Add a rung comment.
- Save the processor file to disk.

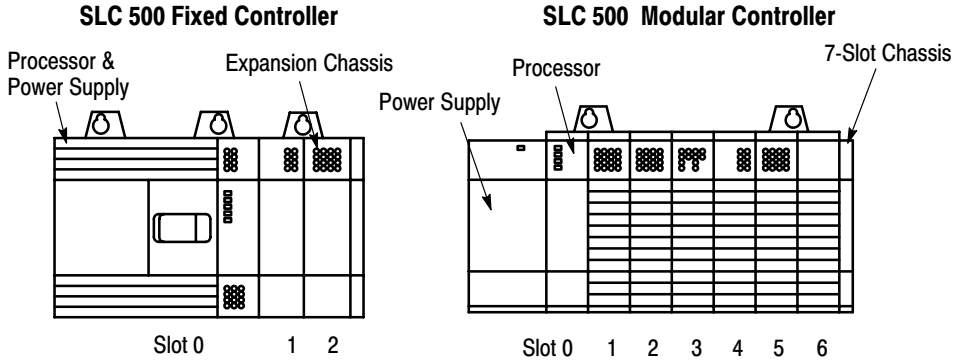
Configuration of SLC 500 Controllers

The following paragraphs briefly describe SLC 500 controllers and indicate the location of catalog numbers on the devices. This information will help you when you create a processor file and enter the specific controller configuration that will run the file.

To make the best use of this manual, you should have access to an SLC 500 Demonstration Unit, which includes completely wired external inputs and outputs. For the exercises in this manual, we arbitrarily assumed that you are using a Demo unit using a modular controller with the components listed on page 3–5.

Controller Styles

As previously mentioned, SLC 500 controllers are available in two styles-the fixed controller and the modular controller. Examples are shown in the figure below.



The fixed controller combines a power supply, processor (CPU), and a fixed number of I/O points in a single unit. You have the option of adding a 2-slot expansion chassis if you want to add I/O points.

The modular controller consists of a power supply, 1–3 I/O chassis, a processor module which you insert in slot 0 of the first chassis, and various I/O modules which you insert in the remaining slots of the chassis.

Slot Numbers

Note that slot numbers are indicated in the figure above. In fixed controllers, slot 0 applies to the processor and fixed I/O points; slots 1 and 2 apply to I/O modules located in the expansion chassis. In modular controllers, slot 0 is always reserved for your processor module; the remaining slots apply to the various I/O modules you have inserted.

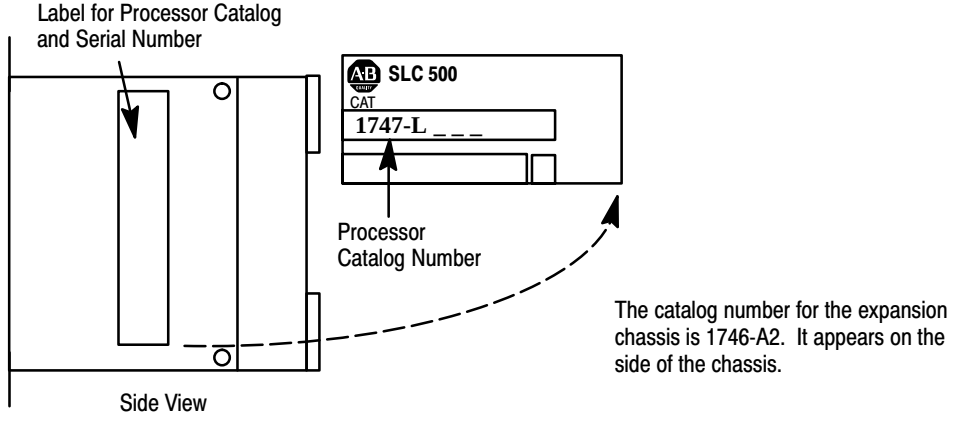
Catalog Numbers

When you configure your controller, you must specify the processor catalog number, chassis catalog numbers, and I/O module catalog numbers as required. The location of the catalog number on the various components is shown in the following figures.

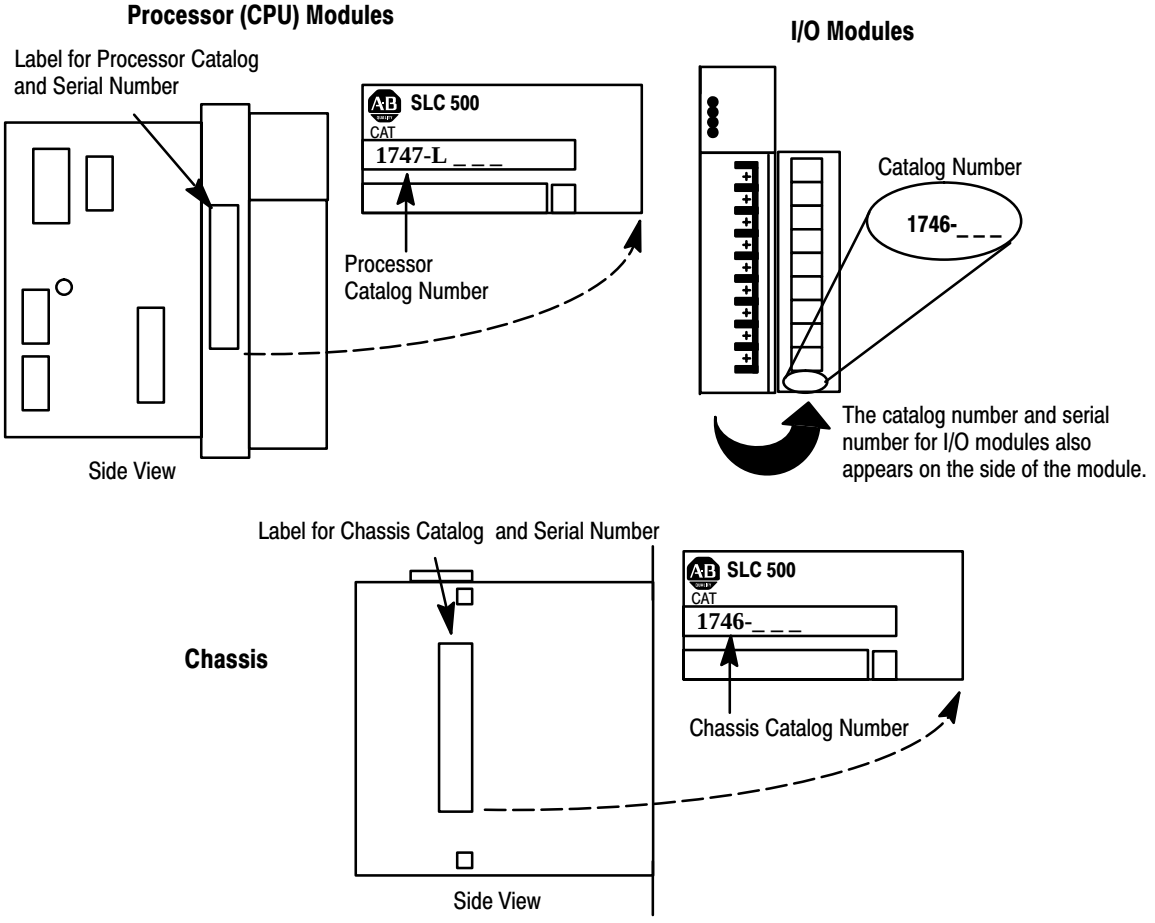
Make a Record of Controller Components

We recommend that you make a list of the processor, chassis, and I/O catalog numbers, and also the chassis numbers assigned to the chassis and the slot locations of all I/O modules. You can then refer to this list as you configure your controller.

Catalog Number Location - SLC 500 Fixed Controllers



Catalog Number Location – SLC 500 Modular Controllers



Arbitrary Controller Used in this Manual

In the following procedures, we have arbitrarily assumed that the controller you are configuring in your processor file is a modular demo unit including the following components:

- Chassis 1746-A4, 4-slot chassis
- Processor 1747-L524 in slot 0
- Input module 1746-IA4 in slot 1
- Input module 1746-IA8 in slot 2
- Output module 1746-OA8 in slot 3

The ladder program shown on page 3–10 contains I/O addresses that are consistent with the configuration indicated above. If you are using some other controller configuration, keep in mind that these addresses may not be valid for your controller.

Creating a Processor File

A processor file is always created offline, in the terminal workspace. In creating the processor file, you will:

- Name the file and configure the controller.
- Enter a ladder program.
- Add a rung comment.
- Save the processor file to disk.

If you are not already running APS, refer to "Running APS," page 1–10. The following procedure begins at the APS menu display.

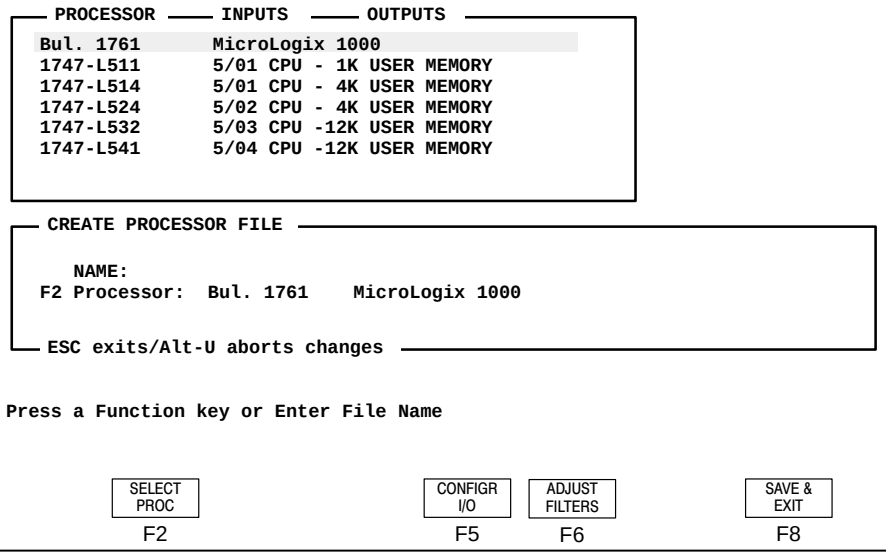
Name the Processor File and Configure the Controller

Complete the following steps:

1. Access the create processor file window.

Press **OFFLINE CONFIG** (F4), then **CREATE FILE** (F6).

The display shown below appears.



2. Enter the name GETSTART.

The prompt line asks you to enter a file name. Type GETSTART, then press **[ENTER]**. GETSTART appears in the Create Processor File window.

3. Enter the appropriate processor catalog number.

The Create Processor File window lists the default processor, Bul. 1761. Our controller uses the 1747-L524 processor. Cursor down to highlight it. (If you are using a different processor, use the cursor keys to locate the appropriate processor in the upper option window.) Press **SELECT PROC** (F2).

4. If you have selected a fixed controller and are not using an expansion chassis, the controller configuration is complete at this point. Press **SAVE & EXIT** (F8) and go to step 8.

5. Configure the chassis of your controller.

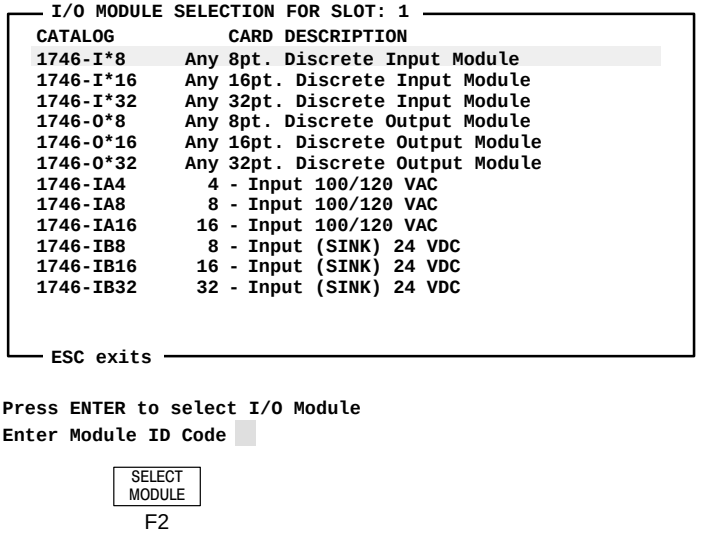
Press **CONFIGR I/O** (F5). The following option window appears. Note that chassis 1 is specified as 1746-A4, the default selection. This is correct for our controller. If you are using a different chassis, press **MODIFY RACKS** (F4), then **RACK 1** (F1). Select the appropriate chassis, using the cursor keys, and press **[ENTER]**. If you are using more than one chassis, follow the same procedure for chassis 2 and 3.

```
I/O CONFIGURATION FOR:GETSTART
RACK 1    =       1746-A4  4-SLOT Backplane
RACK 2    =       NOT INSTALLED
RACK 3    =       NOT INSTALLED
      SLOT      CATALOG #        CARD DESCRIPTION
      *0        1747-L524     5/02 CPU - 4K USER MEMORY
      *1
      *2
      *3
       4
       5
       6
       7
       8

ESC exits

Press a function key

READ CONFIG (F1)  ONLINE CONFIG (F2)  MODIFY RACKS (F4)  MODIFY SLOT (F5)  DELETE SLOT (F6)  UNDEL SLOT (F7)  EXIT (F8)  SPIO CONFIG (F9)
```

Note the asterisks next to slots 0 thru 3. This indicates that we have configured these slots and can now configure I/O modules. Slot 0 is already configured with our processor.

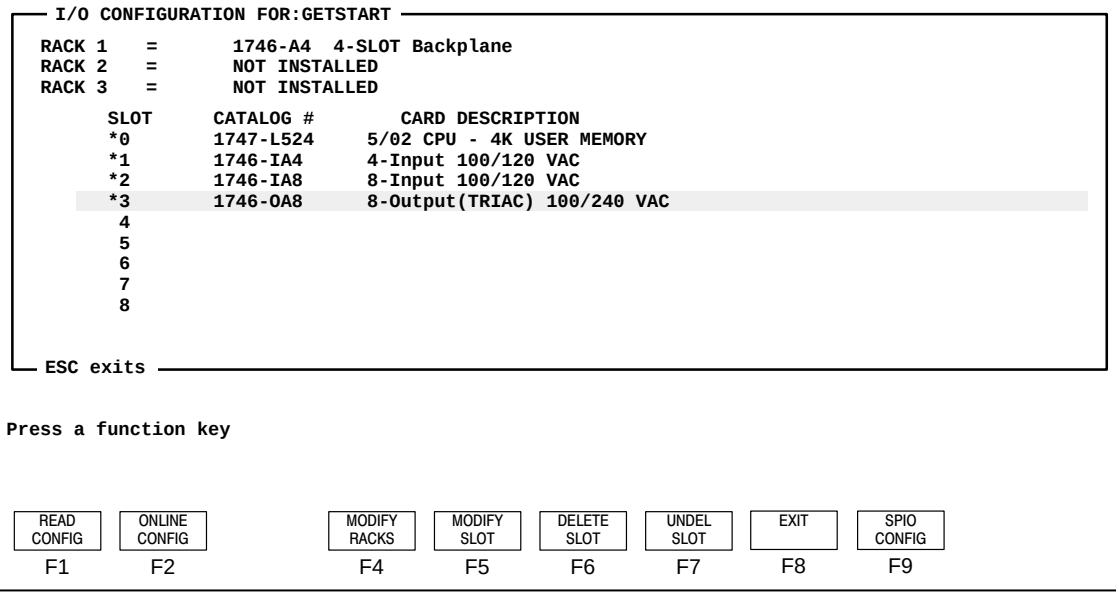
6. Configure the I/O modules.

The cursor is located on slot 1. To configure it, press **MODIFY SLOT** (F5). The following option window appears:



This window allows you to select a module for slot 1. Use the up/down cursor keys to place the cursor on the appropriate module catalog number, then press **SELECT MODULE** (F2). This returns the display to the I/O configuration window with the selected module indicated. Cursor down to the next open slot and repeat the configuration steps.

For our controller, the completed option window appears as follows:



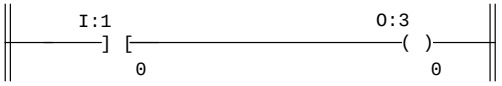
7. Create the archive file GETSTART.

Press **EXIT** (F8), then **SAVE & EXIT** (F8). Archive file GETSTART is created on your computer hard disk and placed in the Offline Processor File window.

8. Press **SAVE TO FILE** (F9) to save GETSTART as the new default file.

Enter the Ladder Program

The following rung consists of an XIC input instruction and an OTE output instruction. The addresses conform to the controller configuration indicated on page 3–5. *If you have entered a different controller configuration, make certain that the addresses are consistent with your configuration.* It is also important that you have an external input, such as a pushbutton, and an external output, such as a pilot light, at the terminal addresses used. You will be using these external devices in later chapters of this manual.



The rung can be entered by completing the following steps:

1. Access the Program Directory of file GETSTART.

Press **OFFLINE PRG/DOC** (F1).

2. Monitor Program File 2.

Press **MONITOR FILE** (F8).

3. Insert a rung.

Press **EDIT** (F10), then **INSERT RUNG** (F4).

4. Enter the Input instruction on the rung.

Press **INSERT INSTR** (F4), then **BIT** (F1), then **XIC -] [-** (F1). Type the address `I:1/0`, then press **[ENTER]**.

5. Enter the Output instruction on the rung.

   Press BIT (F1), then OTE –( )– (F3). Type the address `O:3/0`, then press **[ENTER]**.

6. Accept the rung.

   Press **[ESC]**, then ACCEPT RUNG (F10), then **[ESC]**.

Add a Rung Comment

Complete the following steps to add a rung comment:

1. Configure the display so that rung comments will be visible.

   Press CONFIG DISPLAY (F2). Make sure **[F7]** reads SUPPRSS RNG COM (F7). If it does not, press DISPLAY RNG COM (F7) to toggle **[F7]**, before moving to the next step.

   Press SAVE CONFIG (F10), then **[ESC]**. The display is now configured so that rung comments are visible.

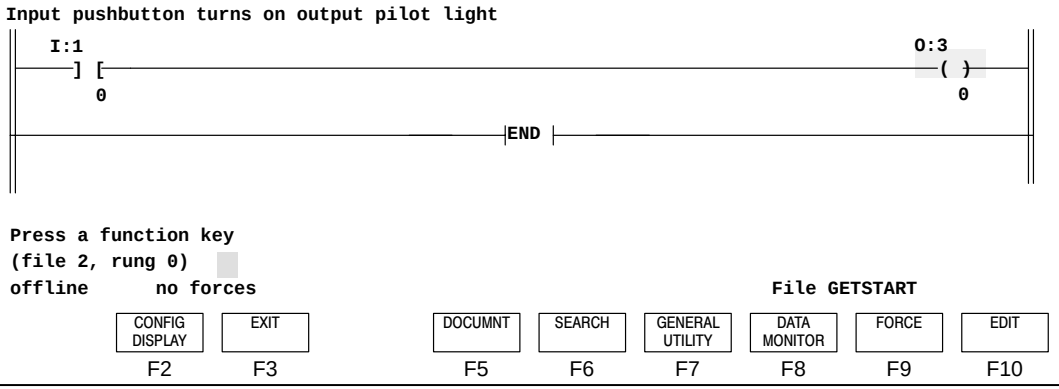
2. Add the rung comment.

   Press DOCUMNT (F5), then RUNG COMMENT (F1). Type the comment `Input pushbutton turns on output pilot light`.

3. Accept and save the comment.

   Press ACCEPT /EXIT (F8), then SAVE DOCUMNT (F10), then **[ESC]**.

Your completed ladder program and rung comment should look like this:



Save the Processor File

Complete the following steps to save the processor file to disk:

1. Return to the Program Directory.

Press **EXIT** (F3).

2. Save the file to disk.

Press **SAVE** (F2). Accept the default Save options by pressing **YES** (F8).

3. Return to the APS menu.

Press **RETURN TO MENU** (F3).

4 Online Operations, Quick Edit

In this chapter you will complete the following tasks:

- download (restore) processor file GETSTART
- monitor the ladder program in the run mode
- test the program
- edit the program using quick edit
- test the edited program
- monitor the input and output data files

Restoring (Downloading) a Processor File

There are two tasks to complete in restoring processor file GETSTART to the processor:

- Check the Online Configuration parameters.
- Go online and download (restore) processor file GETSTART.

The procedures begin at the APS menu display.

Check the Online Configuration Parameters

Complete these steps:

1. Access the Online Configuration window.

Press ONLINE CONFIG (F2). From the menu, choose a 1747-PIC (DH-485) by cursoring to it and pressing DRIVER CONFIG (F2). This display appears:

```
ONLINE CONFIGURATION

F1  Port                          COM1
     Current Device               1747-PIC (DH-485)
F3  Baud Rate                     19200
F4  Terminal Address              0
F5  PROC Address                  1
F6  MAX Node Address              31


F9  Save to File


ESC exits/Alt-U aborts changes

Press a function key

PORT (F1)  SELECT DEVICE (F2)  BAUD RATE (F3)  TERM ADDRESS (F4)  PROC ADDRESS (F5)  MAX ADDRESS (F6)  SAVE TO FILE (F9)
```

Note *This does not apply to SLC 5/04 processors. For these processors, you need to select a driver that corresponds to your specific hardware configuration. Refer to the Advanced Programming Software User Manual for more information.*

2. Verify the parameters.

   The default values are shown for items F1 to F6. If you used the COM1 port of your computer and used a catalog 1747-PIC Interface Converter when connecting your computer to the controller, chances are that you will be able to establish processor-computer communications. If any of these default parameters are incorrect, change them with the function keys, then:

   Press SAVE TO FILE (F9), then **[ESC]**. This returns the display to the APS menu.

Go Online and Restore (Download) Processor File GETSTART

Complete these steps:

1. Access the Restore File window.

   Press ONLINE (F1).

   If the message MESSAGE TIMEOUTS - LOSS OF COMMUNICATIONS appears, one or more of the Online Configuration parameters is incorrect and/or there is an improper connection between the computer and the processor. Refer to appendix B.

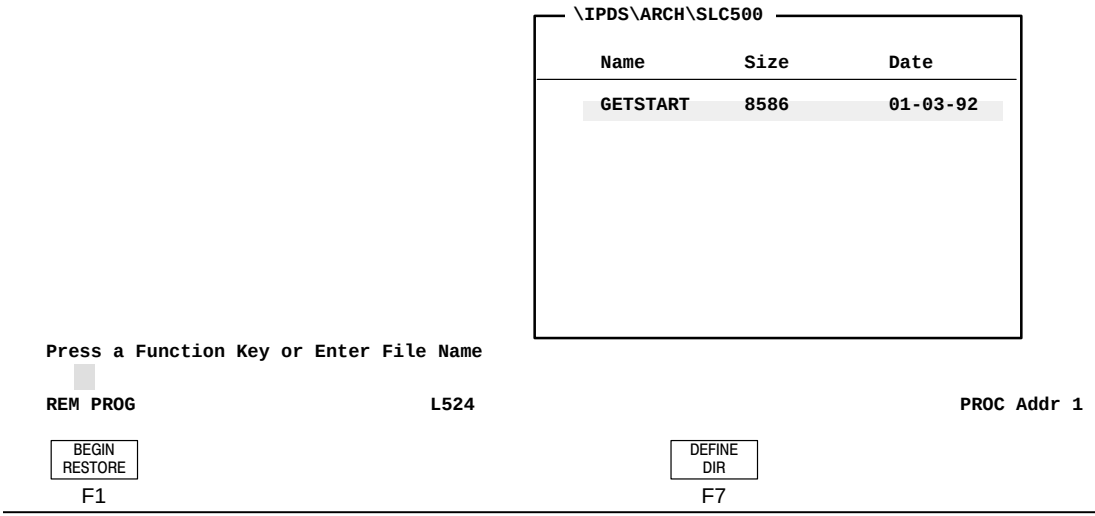
   Once you establish communications with the processor, the program directory display will appear. Do one of these three things:

   a. If the default program directory appears, (the directory is named DEFAULT, and only the system file is listed) press RESTORE (F2).

   b. If a file exists in the processor, and no matching disk file is found on the computer hard disk, you will be asked “Read Processor Program?”. Press NO (F10), then press RESTORE (F2).

   c. If a file exists in the processor, and a matching disk file is found on the computer hard disk, press SAVE RESTORE (F2), then RESTORE PROGRAM (F4).

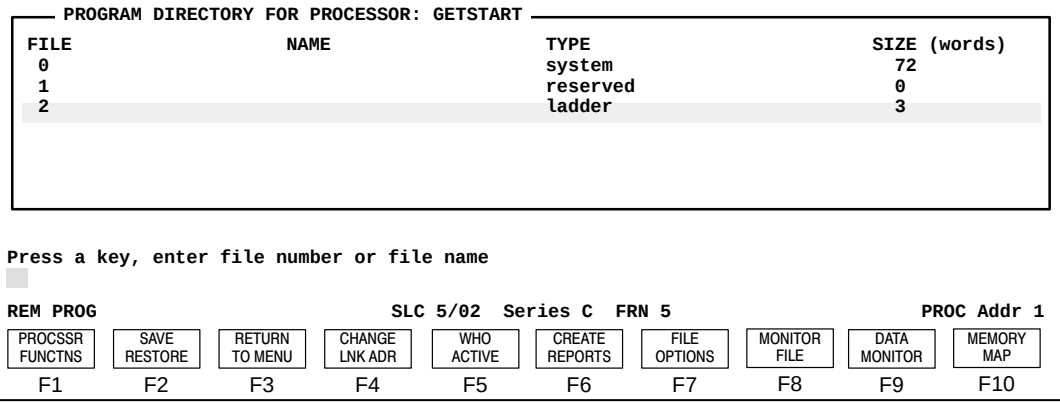
After you have done a), b), or c), the following display appears:



2. Select and accept file GETSTART.

The cursor is located in the right-hand window, which lists all of the processor files saved on disk. Move the cursor to the file GETSTART if it is not already there. Press **BEGIN RESTORE** (F1). If the processor is in the program mode, the file is restored (downloaded). If the processor is in the run mode, you are asked "Change Processor Mode to Program?". Press **YES** (F8). File GETSTART is restored (downloaded).

When the restoring (downloading) process is complete, you are asked to “Press Any Key to Continue”. After you press any key, the program directory for file GETSTART appears:



Testing the Program

To test the ladder program you entered in chapter 3, we will now monitor program file 2, and change the processor mode from program to run. Then activate the external input having address I:1/0 and observe the effect on the external output at address O:3/0.

Begin at the program directory display for processor file GETSTART.

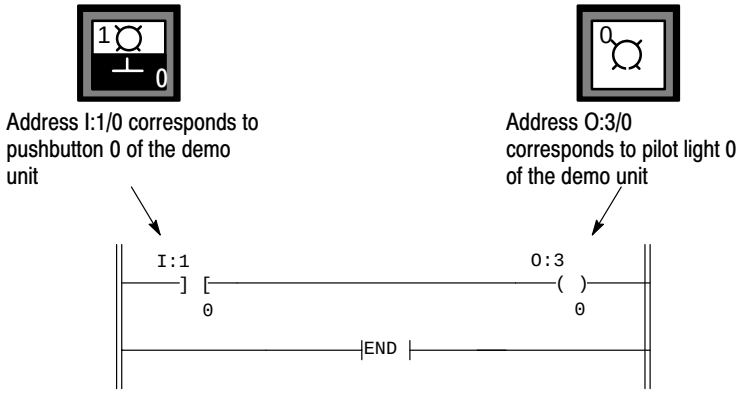
1. Monitor program file 2 and enter the Run mode.

Press **MONITOR FILE** (F8). The ladder program appears.

Press **CHANGE MODE** (F1), then **RUN MODE** (F3), then **YES** (F8). (Note that the status line now indicates REM RUN instead of program.) If you get a fault code on the status line, refer to appendix B to clear the fault.

2. Test the program.

The following diagram shows the rung you entered if you are using the modular controller demo unit discussed on page 3–5. If you are using some other controller configuration, make certain that your external input device and output device are wired to the controller input and output that you addressed in your ladder program.



To test the program, press pushbutton 0. Pilot light 0 should go on. The display should show both the XIC and OTE instructions highlighted to indicate that they are true.

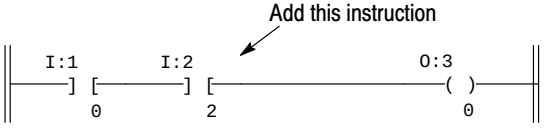
Processor operation: When you pressed pushbutton 0, the input instruction went from false to true. This resulted in a path of true input instructions in the rung, causing the output instruction to go from false to true.

Now release the pushbutton. Pilot light 0 should go off. Neither instruction in the rung should be highlighted. When you released pushbutton 0, the input instruction went from true to false; this broke the path of true input instructions, causing the output instruction to go from true to false.

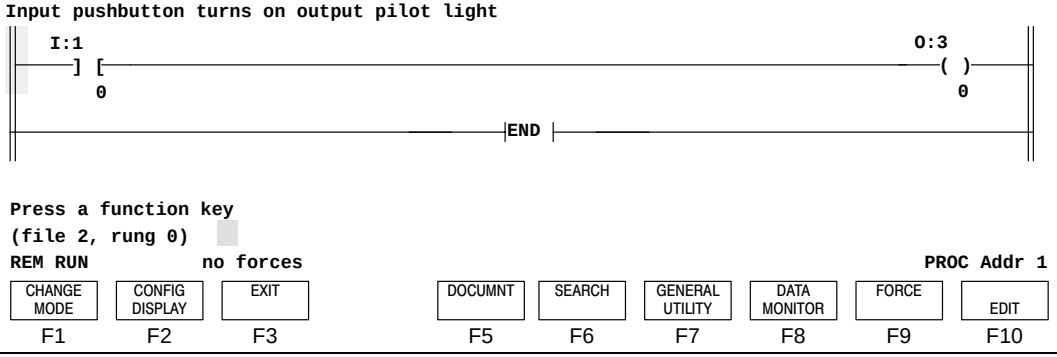
Editing the Program with Quick Edit

The quick edit feature of APS software allows you to move quickly from online monitoring to offline editing, then back to online monitoring. To give you experience at doing this, we will edit the program by adding an input instruction on the rung. The effect of the edit: Selector switch 6 must be on (closed) to allow pushbutton 0 to turn on pilot light 0.

We will place an XIC instruction in series with (to the right of) the XIC instruction already entered. It will have address I:2/2, corresponding to selector switch 6 of the demo unit. See the figure below.



Complete the following six steps to edit and test the edited program. The starting point for this procedure is the online monitor file display, with the processor in the run mode:



1. Go offline and edit the disk version of the file.

Press **EDIT** (F10), then **OFFLINE DISK** (F3). Note that the status line of the display now indicates that you are offline, at file GETSTART.

2. Select Modify Rung and position the cursor.

Press **MODIFY RUNG** (F5).

3. We want to append an instruction to the XIC instruction, so use the cursor keys to position the cursor on the existing XIC instruction.

4. Enter an XIC instruction, address I:2/2.

Press **APPEND INSTR** (F3), then **BIT** (F1), then **XIC -] [-** (F1).

Type the address "I:2/2", then press **[ENTER]**, then **[ESC]**.

5. Accept the rung.

Press **ACCEPT RUNG** (F10).

6. Save the edit and go back online.

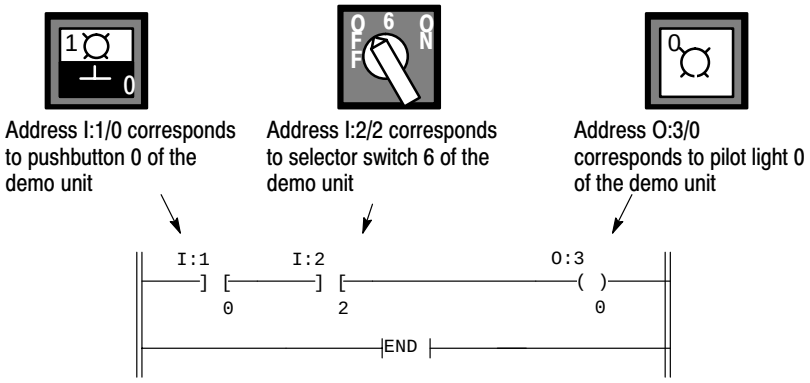
Press **SAVE/GO ONLINE** (F1). Accept the default Save options by pressing **YES** (F8).

Before the software restores the program it asks "Change Processor Mode to Program?". Press **YES** (F8). When the program is successfully restored, the software asks "Change Processor Mode to Run?". Press **YES** (F8).

You are now back online with the edited program, in the run mode.

7. Test the edited program.

The following diagram shows the rung you have modified if you are using the modular controller demo unit discussed on page 3–5. If you are using some other controller configuration, make certain that your external input devices and output device are wired to the controller inputs and output that you addressed in your ladder program.



To test the program, first turn selector switch 6 to the on position. Note that the corresponding input instruction in the rung is highlighted, indicating that it is true. Now press pushbutton 0. Pilot light 0 should go on. The rung should show all instructions highlighted to indicate that they are true.

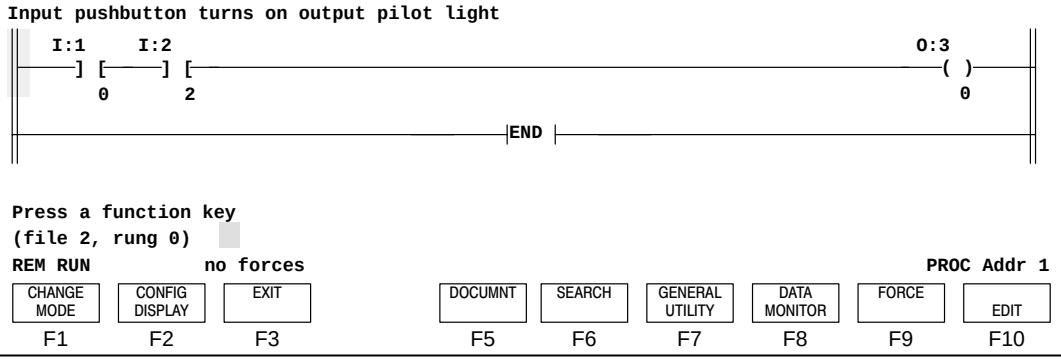
Processor operation: When you pressed pushbutton 0, the corresponding input instruction went from false to true. This resulted in a path of true input instructions in the rung, causing the output instruction to go from false to true.

Now turn selector switch 6 to the off position. Note that the corresponding input instruction in the rung is no longer highlighted, indicating that it is false. Press pushbutton 0. Note that the corresponding input instruction is highlighted, but the output instruction does not go from false to true. This is because a continuous path of true input instructions does not exist in the rung.

Monitoring Data Files

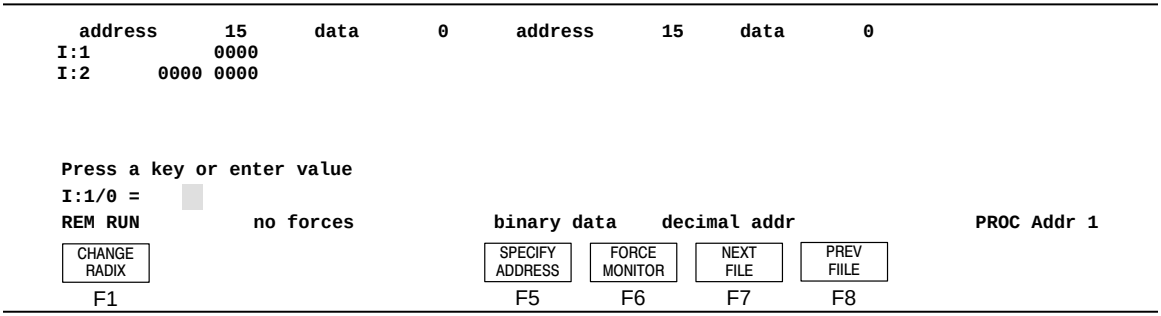
In this procedure, you will monitor the input data file and the output data file. These files include a status bit for each of the configured I/O terminals of the controller. You will monitor data file changes as you operate pushbutton 0 and selector switch 6. To end the exercise, you will go offline to the APS menu.

The starting point is the online monitor file display with the processor in the run mode:



1. Position the ladder cursor and access the input data file.

Use the cursor key to position the cursor on the XIC instruction having address I:1/0, then press **DATA MONITOR** (F8). The input data file appears, with the cursor located on status bit I:1/0. This is shown below.

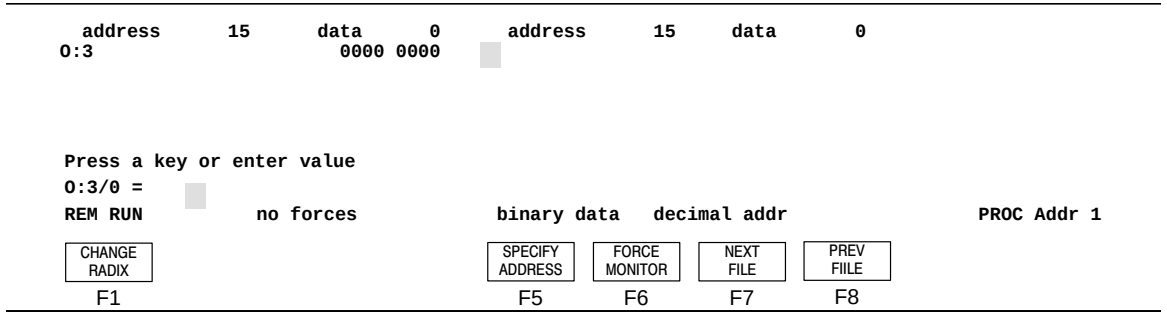


2. Monitor input data changes resulting from input device operation.

Press pushbutton 0. Note that the status bit goes from 0 to 1, as the instruction goes from false to true. Now turn selector switch 6 to the on position. Note that status bit I:2/2 goes from 0 to 1, as the instruction goes from false to true.

3. Access the output data file.

The output data file precedes the input data file in the data table. Press **PREV FILE** (F8). The output data file appears. Since we didn't specify a particular bit address, the cursor is located on the status bit having the lowest address, O:3/0. This is also the status bit for pilot light 0 in our program. This is shown below.



4. Monitor output data changes resulting from input device operation.

Press pushbutton 0 with selector switch 6 in the on position. Note that status bit O:3/0 goes from 0 to 1, as the output instruction of our program goes from false to true.

Continue to press pushbutton 0, as you turn selector switch 6 to the off position. Note that bit O:3/0 goes from 1 to 0. This is because there is no longer a path of true input instructions in the rung, causing the output instruction to go false.

5. Return to the APS menu.

Press **[ESC]**. This returns you to the online monitor file display.

Press **EXIT** (F3). This returns you to the online program directory display.

Press **RETURN TO MENU** (F3). This takes you offline, returning you to the APS menu display.

5 Creating and Printing Reports

This chapter shows you how to create and print reports. The following four hard copy reports can be created and printed:

- Program Listing - Can include a) the main program file and all subroutine files, b) a single file, c) a range of files, or d) a range of rungs.
- Cross Reference - Provides an alphabetical list of addresses and their rungs, in either address or symbol order.
- Processor Configuration - Details the configuration of the processor and associated hardware in the system.
- Data Tables - Details the contents of the offline or online data files.

If you do not have a printer set up in your system, we suggest that you go through these procedures anyway, to familiarize yourself with report capabilities.

Creating Reports

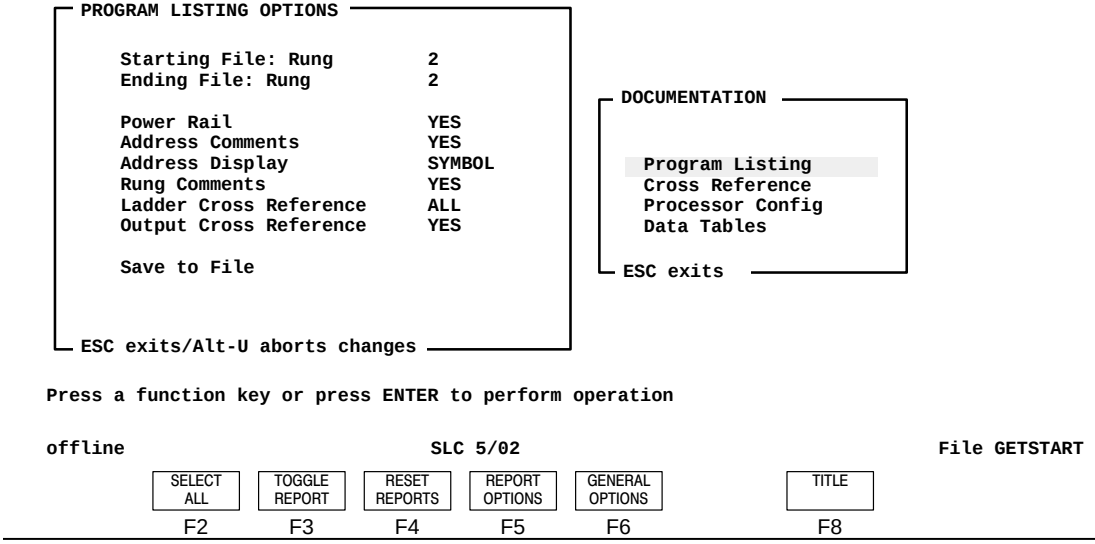
A report can be created at the program directory display, either offline or online. In the following procedure, reports are created offline. The starting point is the APS menu.

Complete the following steps:

1. Access the documentation (reports) and options windows.

Press **OFFLINE PRG/DOC** (F3). This accesses the program directory display.

Press **CREATE REPORTS** (F6). The following windows appear in the display area:



2. Specify documentation.

The “Documentation” window lists the four reports you can create. The cursor is located on the “Program Listing” report. Options for the Program Listing are shown in the window at the left. Function key F5 allows you to change items in the options window.

Move the cursor to “Cross Reference”, then “Processor Config”, then “Data Tables”. Note that as you do this, the option window changes to match the report the cursor is located on.

An explanation of the various options is beyond the scope of this manual. For our purposes, the default options are suitable.

3. Press **SELECT ALL** (F2). In doing this, you have selected all four reports.

This is verified by the appearance of asterisks at the left of each report in the Documentation window.

4. Specify a title and create the reports.

Press **TITLE** (F8). Type `GETPRINT` in the window that appears.

Press **[ENTER]** to accept the title. Then press **[ENTER]** to perform the create reports operation. When the reports have been created, DOCUMENTATION COMPLETE appears in the display area, and PRESS A KEY TO CONTINUE appears on the prompt line.

Press any key. The program directory appears.

5. Return to the APS menu.

Press **RETURN TO MENU** (F3). You will be asked to “Save Cross Reference, Comment, and Symbol Files?” Since we have not changed the ladder program in any way, a Save is not required.

Press **NO** (F10). The APS menu appears.

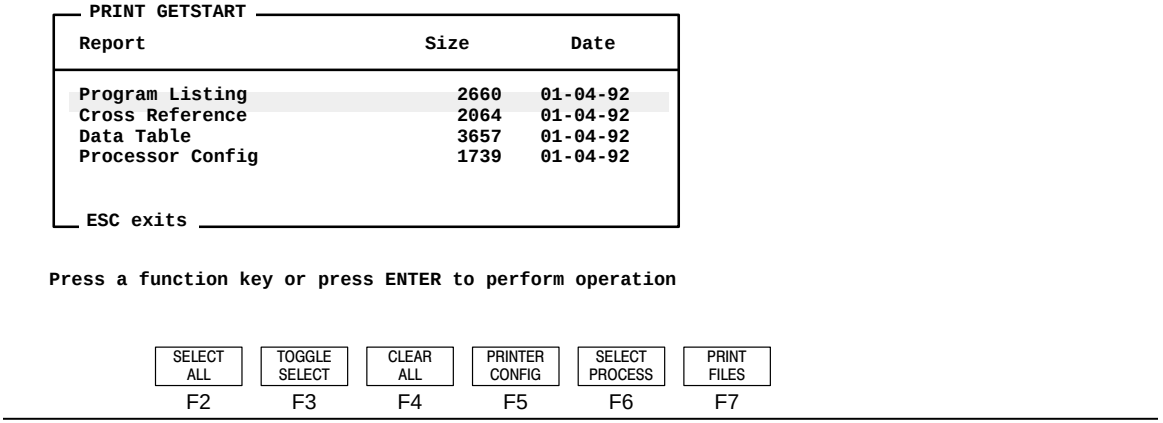
Printing Reports

Printing reports is done from the APS menu. Complete the following steps:

1. Access file GETSTART in the Report Directory.

Press **PRINT REPORTS** (F8). The report directory appears. It lists the processor file names for which reports have been created:

Use the up/down cursor key to move the cursor to GETSTART, then press **[ENTER]**. The display shows the reports you have created for file GETSTART:



2. Configure the printer and prepare it for operation.

Press **PRINTER CONFIG** (F5). Change configuration parameters if necessary, then press **[ENTER]**. Prepare the printer for operation.

3. Select the reports to be printed and initiate printing.

Press **SELECT ALL** (F2). Your choice is verified by the appearance of asterisks at the left of the four reports you have created. Press **PRINT FILES** (F7) to perform the printing operation.

If your printer is not ready for some reason, PRINTER NOT READY appears on the message line. The prompt line asks "Continue Printing?" You can correct the problem and press **YES** (F8), or you can cancel printing by pressing **NO** (F10).

After the printing operation, you can return to the APS menu by pressing **[ESC]**.

A Additional Ladder Program Exercises

This appendix lets you apply what you have learned in the previous chapters. It covers:

- entering a program with I/O branches
- entering a program with a timer instruction

Entering an Input and Output Branch

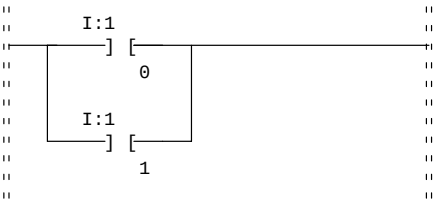
The important feature of this program is the output and input branch. The input branch is based on what is called OR or parallel logic. This means that if either input #0 OR input #1 is true, then output #0 and #1 turn on.

Exercise 1: Entering an Input and Output Branch

We are assuming you have created a new file, configured it and you are now ready to begin entering an instruction. See chapter 3 for help with the above. Begin offline at the edit screen.

1. Enter the rung and XIC instruction.

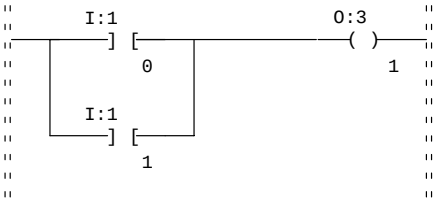
Press **INSERT RUNG** (F4), then **INSERT INSTR** (F4), then **BIT** (F1), then **XIC -] [-** (F1). Type the address `I:1/0` then press **`[ENTER]`**, then **`[ESC]`**.



2. Enter a branch and another XIC instruction.

Cursor left once, so your cursor is on the XIC instruction. Press **BRANCH** (F1), then **INSERT BRANCH** (F4), then **TARGET B** (F2), then **INSERT INSTR** (F4), then **BIT** (F1), then **XIC -] [-** (F1).

Type the address `I:1/1` then press **[ENTER]**.



3. Enter an OTE instruction.

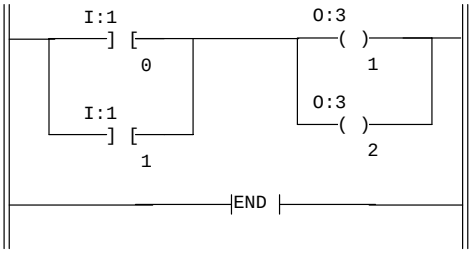
Cursor up, then cursor right so your cursor is at the far right power rail. Press **BIT** (F1), then **OTE -( )-** (F3). Type the address `O:3/1` then press **[ENTER]**, then **[ESC]**.

4. Enter a branch and another OTE instruction.

Cursor left once so you are on the OTE instruction. Press **BRANCH** (F1), then **INSERT BRANCH** (F4), then **TARGET C** (F3), then **INSERT INSTR** (F4), then **BIT** (F1), then **OTE -( )-** (F3). Type the address `O:3/2` then press **[ENTER]**, then **[ESC]**.

5. Accept the rung.

Press **ACCEPT RUNG** (F10), then **[ESC]**.



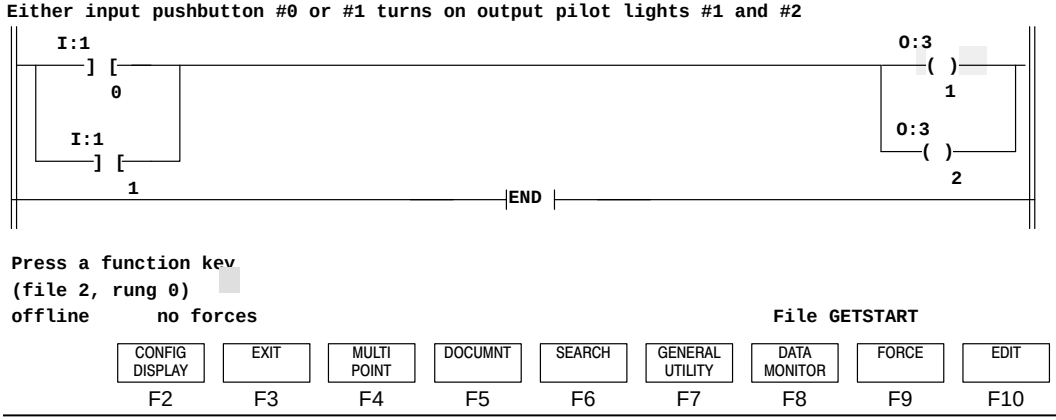
6. Enter the rung comment.

Press **DOCUMNT** (F5), then **RUNG COMMENT** (F1). Type the comment `Either input pushbutton #0 or #1 turns on output pilot lights #1 and #2.`

7. Accept and save the comment.

Press **ACCEPT /EXIT** (F8), then **SAVE DOCUMNT** (F10), then **[ESC]**.

Your completed ladder program and rung comment should look like this:



Save the Processor File

Complete the following steps to save the processor file to disk:

1. Return to the Program Directory.

Press **EXIT** (F3).

2. Save the file to disk.

   Press **SAVE** (F2). Accept the default Save options by pressing **YES** (F8).

3. Return to the APS menu.

   Press **RETURN TO MENU** (F3).

Test the Ladder Program

Complete the following steps to test the processor file:

1. Go online with your processor and restore the new file. Refer to chapter 4 for help.
2. Monitor the file. Refer to chapter 4 for help.
3. Place the processor in the RUN mode. See chapter 4 for help.
4. Press pushbutton #0. Outputs #1 and #2 turn ON.
5. Release pushbutton #0. Outputs #1 and #2 turn OFF.
6. Press pushbutton #1. Outputs #1 and #2 turn ON.
7. Release pushbutton #1. Outputs #1 and #2 turn OFF.

Entering a Timer Instruction

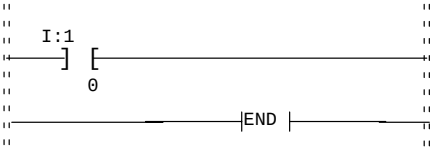
In exercise 2, you enter a timer instruction with a time delay of 10 seconds. Two different types of timer status bits activate output pilot lights #0 and #1. The first type, called a “timer timing” status bit turns on output #0 for 10 seconds. The second type, called a “done” status bit, turns on output #1 *after* 10 seconds.

Exercise 2: Entering a Timer Instruction

We are assuming you have created a new file, configured it and you are now ready to begin entering an instruction. See chapter 3 for help with the above. Begin offline at the edit screen.

1. Enter a rung and an XIC instruction.

Press **APPEND RUNG** (F3), then **INSERT INSTR** (F4), then **BIT** (F1), then **XIC –] [–** (F1). Type the address `I:1/0` then press **`[ENTER]`**.



2. Enter the timer instruction.

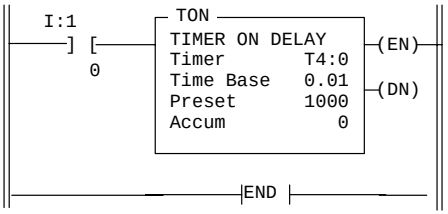
Press **TIMER COUNTER** (F2), then **TON** (F1).

Type the address `T4:0` then press **`[ENTER]`**. This is the Timer Address.

Type the timebase `.01` then press **`[ENTER]`**. This is the timebase in seconds.

Type `1000` then press **`[ENTER]`**. This is the Timer Preset Value in hundredths of a second.

Type `0` then press **[ENTER]**. This is the Timer Accumulated Value.

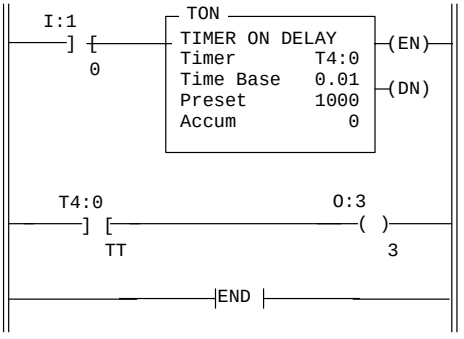


3. Accept the rung.

Press **[ESC]**, then **ACCEPT RUNG** (F10).

4. Enter a second rung and an XIC instruction.

Press **INSERT INSTR** (F4), then **BIT** (F1), then **XIC -] [-** (F1). Type the address `T4:0/TT` then press **[ENTER]**. "TT" represents the timer timing bit.



5. Enter an OTE instruction.

Press **BIT** (F1), then **OTE -( )-** (F3). Type the address `O:3/3`, then press **[ENTER]**.

6. Accept the rung.

Press **[ESC]**, then ACCEPT RUNG (F10).

7. Enter a third rung and an XIC instruction.

Press INSERT INSTR (F4), then BIT (F1), then XIC -] [- (F1). Type the address `T4:0/DN`, then press **[ENTER]**. "DN" represents the timer done bit.

8. Enter an OTE instruction.

Press BIT (F1), then OTE -( )- (F3). Type the address `O:3/4`, then press **[ENTER]**.

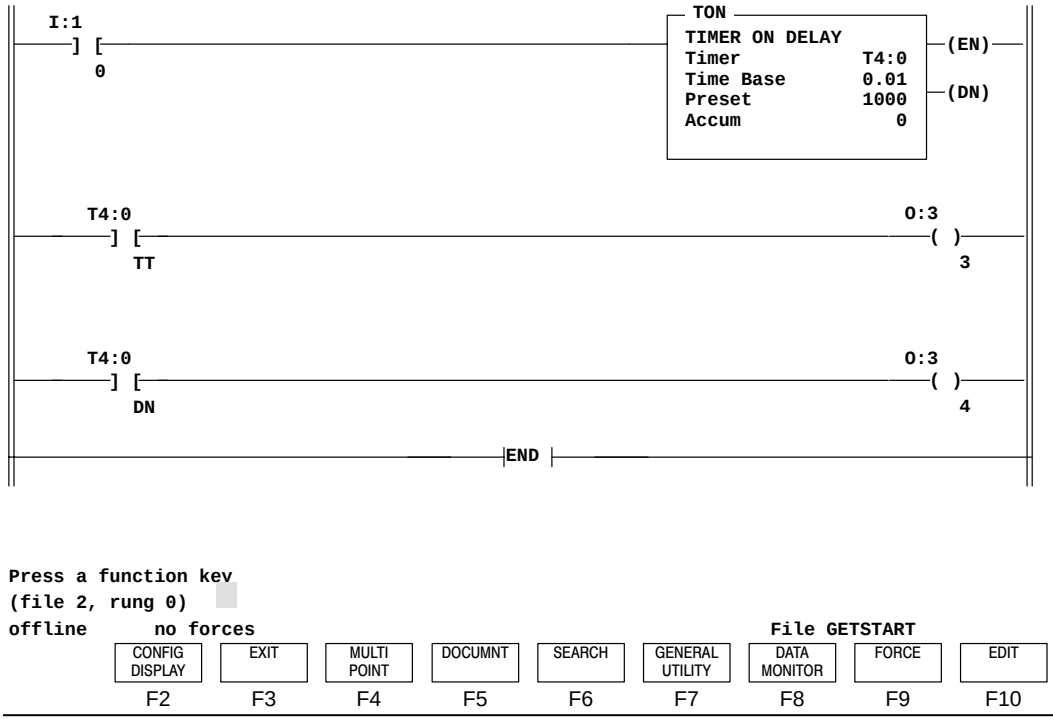
9. Accept the rung.

Press **[ESC]**, then ACCEPT RUNG (F10), then **[ESC]**.

10. Exit the edit mode.

Press **[ESC]**.

Your completed ladder program should look like this:



Save the Processor File

Complete the following steps to save the processor file to disk:

1. Return to the Program Directory.

Press **EXIT** (F3).

2. Save the file to disk.

Press **SAVE** (F2). Accept the default Save options by pressing **YES** (F8).

3. Return to the APS menu.

   Press **RETURN TO MENU** (F3).

Test Your Ladder Program

Complete the following steps to test the timer instruction file:

1. Go online with your processor and restore the new file. Refer to chapter 4 for help.
2. Monitor the file. Refer to chapter 4 for help.
3. Place the processor in the RUN mode. See chapter 4 for help.
4. Press pushbutton #0 for at least 10 seconds. During the first 10 seconds, output #3 turns ON and #4 stays OFF.
5. After 10 seconds, output #3 turns OFF and output #4 turns ON.
6. Release pushbutton #0; the timer resets and both outputs #3 and #4 turn OFF.

B Troubleshooting

This appendix shows you how to identify and correct errors that you may encounter while working through this manual. They include:

- APS error messages
- system LEDs status
- processor error codes

APS Error Messages

Table B.A details APS error messages. Refer to the *Advanced Programming Software User Manual* for a complete list of error messages.

Table B.A
APS Error Messages

Error Message	Possible Causes	Corrective Action
APS Timeout - Loss of Communications	Wrong baud rate	Select different baud rate in F2 "Online Config"; Processor default is 19200.
	Wrong processor node address	Select different processor address in F2 "Online Config"; Processor default is 1.
	Wrong device type	Device type in F2 "Online Config" should be 1747-PIC.
	Incompatible or wrong computer Serial Port	Select different COM port in F2 "Online Config"; verify PC COM port works.
	Bad cable	Check continuity in 1747-C10 cable; contact local Allen-Bradley distributor for replacement.
	Bad 1747-PIC	Contact your local Allen-Bradley distributor for replacement.
	Incompatible 9-25 Pin Adaptor	Consult PC manual for Serial Port type (DCE or DTE); 9-25 Pin Adaptor supplied with 1747-PIC is for a DTE Serial port. If serial port is DCE, you may need a null-modem adaptor.
	Not enough power to 1747-PIC	Check line power to SLC power supply; check position of power supply jumper for modular systems.
Database Read Error	Files and buffers are not set up correctly	Use a word processor or DOS Edline to verify/change your CONFIG.SYS file to contain minimum values of Files = 40 and Buffers = 40. (Minimum values of 46 are required if running APS in a Windows™ environment.) If the file is modified, re-boot PC.
Fatal Communication Hardware Error	Incompatible or non-existent Serial COM port on PC	Select different COM port by pressing F2 "Online Config"; verify COM port works.
Illegal Data or Parameter Value	Maximum node address of the processor exceeds 31	Reduce the maximum node address of the processor to 31 by pressing F5 "WHO"; F5 "Who Active"; F7 "Max Address".

Table B.A
APS Error Messages (continued)

Error Message	Possible Causes	Corrective Action
I/O Address Not Configured	Processor/system configuration does not match entered addresses	Verify correct address format (I:slot/terminal or O:slot/terminal); verify system configuration by pressing F3 “Offline Prg/Doc”; F1 “Procssr Functns”; F1 “Change Procssr”; F5 “Configr I/O”.
No Matching Disk File Found	Processor program does not exist on hard disk	To read the processor program (upload), press F8 “Yes”; otherwise press F10 “No” to continue with other online activities.
No Memory Left or Not Enough Memory to Load Communication Driver	PC does not have enough free RAM memory to continue	Verify your PC has >250K of free RAM to execute APS. (>369K free RAM is required if you are running the INTERCHANGE™ software). Exit APS and type “CHKDSK” at DOS prompt. The last line should read “>250K bytes free” (or >369K). If not, disable TSR, drivers, menus, shells, etc. loaded in AUTOEXEC.BAT or CONFIG.SYS that may be running in background. Re-boot PC.
Incompatible Processor Type	The processor configuration of the program you are restoring does not match your hardware	Verify that processor configuration of your program matches your hardware by pressing F3 “Offline Prg/Doc”; F1 “Procssr Functns; F1 “Change Procssr”.

System LED Status

The system LEDs are located at different places on the modular system and the SLC fixed controller. Refer to the installation and operation manual for more information on system LED status. See Figure B.1.

Figure B.1
System LEDs

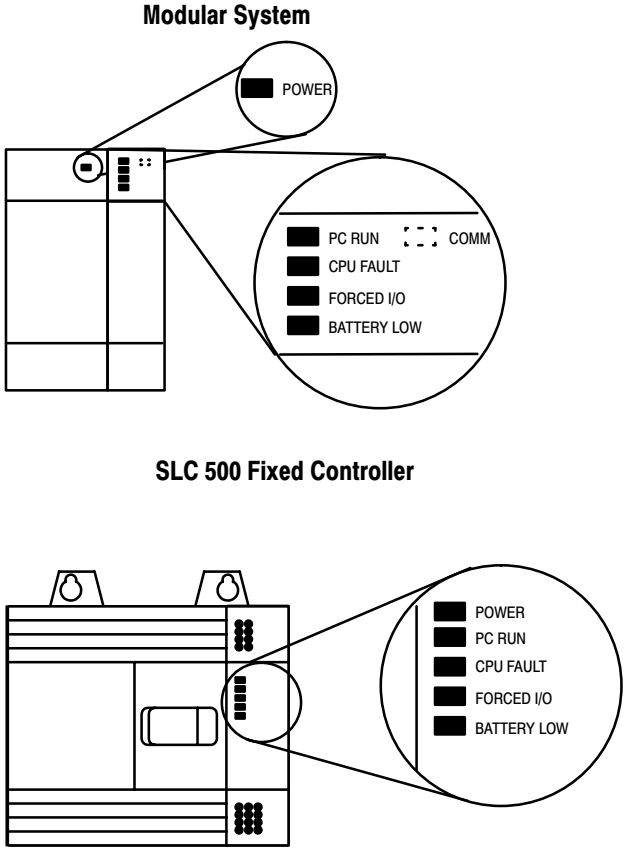


Table B.B
LED Status - Error Conditions

Processor LED	When It Is	Indicates that
RUN (Color: red)	On (steadily)	The processor is in the Run mode.
	Off	The processor is in a mode other than Run.
CPU FAULT (Color: red)	Flashing (at power up)	The processor has not been configured.
	Flashing (during operation)	The processor detects a major error either in the processor, expansion chassis or memory.
	On (steadily)	A fatal error is present (no communication).
	Off	There are no errors.
FORCED I/O (Color: red)	Flashing	One or more input or output addresses have been forced to an On or Off state but the forces have not been enabled.
	On (steadily)	The forces have been enabled.
	Off	No forces are present or enabled.
BATTERY LOW (Color: red)	On (steadily)	The battery voltage has fallen below a threshold level or the battery and the battery jumper are missing.
	Off	The battery is functional, or the battery jumper is present.
COMM (Color: red)	On (steadily)	The SLC 5/02 is receiving data.
	Off	The SLC 5/02 is not receiving data.

Processor Error Codes

Table B.C details some of the processor error codes. Refer to the *Instruction Set Reference Manual* for a complete list of error codes and troubleshooting information.

Table B.C
Processor Error Codes

Error Code	Cause	Corrective Action
0001	RAM program is corrupt due to noise, lightning, improper grounding or loss of capacitor or battery back-up.	Check wiring, layout, grounding. If using a 4K CPU, verify that a battery is installed to retain RAM memory when power is removed. See CPU FAULT-Flashing under system LED status. Restore the program using APS or an HHT.
0012	RAM program is corrupt or RAM itself is bad due to noise, lightning, improper grounding, or loss of capacitor or battery back-up.	Check wiring, layout, grounding. If using a 4K CPU, verify that a battery is installed to retain RAM memory when power is removed. See CPU FAULT-Flashing under system LED status. Restore the program using APS or an HHT.
XX50, XX51, XX52 XX53, XX54, XX55 (xx = slot #)	I/O module configuration/conflict or runtime problem.	Verify that processor configuration matches your hardware by pressing F3 “Offline Prg/Doc”; F1 “Procssr Functns”; F1 “Change Procssr”; F5 “Configr I/O”; See CPU FAULT-Flashing under the system LED status.
XX56 (xx = slot #)	Chassis configuration error.	Verify that chassis configuration in your program matches your hardware by pressing F3 “Offline Prg/Doc”; F1 “Procssr Functns”; F1 “Change Procssr”; F5 “Configr I/O”. If multiple chassis system, verify proper installation of chassis interconnect cable. See CPU FAULT - Flashing under system LED status.

Glossary

The following terms are used throughout this manual. Refer to the Allen-Bradley Industrial Automation Glossary, publication number AG-7.1, for a complete guide to Allen-Bradley technical terms.

address: A character string that uniquely identifies a memory location. For example, I:1/0 is the memory address for the data located in the Input file location 1/0.

APS (Advanced Programming Software): Software used to monitor and develop SLC 500 ladder logic programs.

bit: Binary digit. The smallest unit of information in the binary numbering system. Represented by the digits 0 and 1. The smallest unit of memory.

branch: A parallel logic path within a ladder logic rung.

chassis: A hardware assembly that houses devices such as I/O modules, adapter modules, processor modules, and power supplies.

comment: Text included within a program to explain what the program is doing. Comments do not affect the operation of the program in any way.

communication scan: A part of the SLC CPUs operating cycle in which communication takes place with other devices, such as APS on a personal computer.

controller: A unit, such as a programmable controller or relay panel, that controls machine or process elements.

CPU (central processing unit): The decision-making and data storage section of a programmable controller.

cross reference: A report listing addresses, instructions, and their rung numbers where used.

data file: An area within a processor file that contains the status of inputs, outputs, the processor, timers, counters, and so on.

data table report: A report documenting the contents of the data files.

DOS: The operating system used to operate a personal computer.

edit: To create or modify a ladder program.

expansion chassis: A 2-slot chassis used only with fixed controllers.

false: The status of an instruction that does not provide a continuous logical path on a ladder rung.

file: A collection of information organized into one group.

fixed controller: A controller with a power supply, CPU, and I/O integrated into a single package.

function keys: Keys on a personal computer keyboard labeled F1, F2, F3, and so on. The operation of each of these keys is defined by software and a key may have a different function for each menu display.

hard disk: A disk storage device for storing relatively large amounts of data.

hardware: Mechanical, electrical, and electronic components and assemblies.

I/O (Inputs and Outputs): Consists of input and output devices that provide and/or receive data from the programmable controller.

input device: A digital or analog device, such as a limit switch, push-button switch, pressure sensor, or temperature sensor, that supplies input data through an input circuit to a programmable controller.

input scan: A part of the SLCs operating cycle. Status of the input modules are loaded into the Input data file.

instruction: A mnemonic and data address defining an operation to be performed by the processor. A rung in a program consists of a set of input and output instructions. The input instructions are evaluated by the controller as being true or false. In turn, the controller sets the output instructions to true or false.

interface converter: An Allen-Bradley device, Catalog Number 1747-PIC, used to establish communication between the personal computer and a SLC 500 programmable controller.

ladder logic: A program written in a format resembling a ladder-like diagram. The program is used by a programmable controller to control devices.

mnemonic: A simple and easy to remember term that is used to represent a complex or lengthy set of information.

modular controller: SLC 500 system consisting of a power supply, chassis, CPU, and input and output modules.

module: An interchangeable plug-in device that may inserted into a chassis.

network: A series of stations (nodes) connected by some type of communication medium. A network may be made up of a single link or multiple links.

off line: Describes devices not under direct communication. For example, when programming the software.

online: Describes devices under direct communication. For example, when the software is monitoring the program file in a controller.

operating cycle: The sequential order of operations performed by a programmable controller when in the run mode.

OTE (OuTput Energize): An instruction that energizes when a rung is true and de-energizes when a rung is false.

output device: A device, such as a pilot light or a motor starter coil, that receives data from the programmable controller.

output scan: A part of the SLCs operating cycle. During this scan the output data file information is transferred to the output modules.

processor: A Central Processing Unit. (See CPU.)

processor configuration: A report detailing the configuration of the processor.

processor file: The set of program and data files used by the SLC to control output devices. Only one processor file may be stored in the SLC at a time.

processor overhead: The part of the operating cycle used for housekeeping and setup purposes.

program file: The area within a processor file that contains the ladder logic program.

program listing: A report containing a range of program files or a range of rungs.

program mode: When the SLC is not executing the processor file and all outputs are de-energized.

program scan: A part of the SLCs operating cycle. During the scan the ladder program is executed and the Output data file is updated based on the program and the Input data file.

rack: See chassis.

read: To acquire data from a storage place. For example, the processor READs information from the input data file to solve the ladder program.

report: A printable document containing information about a processor file. For example, a ladder listing, a cross reference, the data tables, and the processor configuration.

restore: To download (transfer) a program from a personal computer to a SLC.

run mode: When the processor file in the SLC is being executed, inputs are read, the program is scanned, and outputs are energized and de-energized.

rung: Ladder logic is comprised of a set of rungs. A rung contains input and output instructions. During Run mode, the inputs on a rung are evaluated to be true or false. If a path of true logic exists, the outputs are made true. If all paths are false, the outputs are made false.

save: To upload (transfer) a program stored in memory from a SLC to a personal computer; OR to save a program to a computer hard disk.

SLC (Small Logic Controller): A controller that comes in 1 of 2 styles: fixed or modular.

slot: The area in a chassis that a module plugs into.

software: Executable programming package used to develop SLC ladder diagrams.

status: The condition of a circuit or system, represented as logic 0 (OFF) or 1 (ON).

terminal: A point on an I/O module that external I/O devices, such as a pushbutton or pilot light, are wired to.

true: The status of an instruction that provides a continuous logical path on a ladder rung.

write: To copy data to a storage device. For example, the processor WRITEs the information from the output data file to the output modules.

eXamine If Closed (XIC): An input instruction that is logically true when the status of the bit located at its address is a 1; false when it's a 0.

eXamine If Open (XIO): An input instruction that is logically true when the status of the bit located at its address is a 0; false when it's a 1.

Index

P

Q

R

S

T