

# ***MACHINELOGIC***

## **Getting Started Guide**

***Parker Automation - CTC  
50 W. TechneCenter Drive, Milford, Ohio 45150***

***Phone: 513-831-2340  
Technical Support: 513-248-1714***

A3-05508-105

## Copyright and Trademark Notice

Copyright © 2002 by Parker Automation - CTC. All rights reserved. No part of this publication may be reproduced, transmitted, transcribed, stored in a retrieval system, in any form or by any means, mechanical, photocopying, recording or otherwise, without the prior written consent of Parker Automation - CTC.

While every precaution has been taken in preparing this manual, Parker Automation - CTC and the author assume no responsibility for errors or omissions. Neither is any liability assumed for damages resulting from the use of the information contained herein. All product and company names are trademarks of their respective companies and licenses.

## Product Warranty Information

Parker Automation - CTC provides top quality products through rigid testing and the highest quality control standards. However, should a problem occur with your hardware or with the software protection key, our standard product warranty covers these items for 15 months from the date of shipment. Exceptions appear below:

- PowerStation backlight bulbs have a 90-day warranty.
- Third-party products, such as bus cards, carry the manufacturer's specified warranty.
- For all displays, image retention (burn-in) is not covered by warranty.
- Software revisions that occur within 60 days after purchase are available, under warranty, upon request. Please review the MachineShop License Agreement for additional software warranty information.

If you have any questions about your application or need technical assistance, please call Technical Support at 513-248-1714, 8:00 a.m. to 5:00 p.m., Eastern Time. You may call this same number after hours for emergency assistance. See *Customer Support Services* on page 4 of this manual for more information about CTC's support products and services.

# Table of Contents

---

<b>Chapter 1 Overview and Support Services</b> . . . . .	<b>1</b>
What's in this User Guide? . . . . .	2
Documentation Standards . . . . .	3
ISO Symbols . . . . .	3
Customer Support Services . . . . .	4
Technical Support . . . . .	4
Training and New Business Development . . . . .	5
Training . . . . .	5
Application Engineering Services . . . . .	5
Product Support Program . . . . .	6
Documentation . . . . .	6
<b>Chapter 2 Introducing MachineLogic</b> . . . . .	<b>7</b>
Development System Requirements . . . . .	8
Hardware Requirements . . . . .	8
Software Requirements . . . . .	8
Installing the Software . . . . .	8
Runtime System Requirements . . . . .	9
MLCA-specific Requirements . . . . .	9
MLCA Hardware Requirements . . . . .	9
MLCA Software Requirements . . . . .	9
MLSC-specific Requirements . . . . .	9
MLSC Hardware Requirements . . . . .	9
MLSC Software Requirements . . . . .	10
Installing the Hardware . . . . .	10
Installing the Software . . . . .	10
Opening MachineLogic . . . . .	11
MachineLogic User Interface . . . . .	12
Menus . . . . .	12

# Table of Contents

---

File Menu . . . . .	12
Edit Menu . . . . .	13
View Menu . . . . .	13
Project Menu . . . . .	13
Build Menu . . . . .	13
Objects Menu . . . . .	13
Layout Menu . . . . .	14
Online Menu . . . . .	14
Options Menu . . . . .	14
Window Menu . . . . .	14
Help Menu . . . . .	14
Toolbars . . . . .	15
Detaching the Toolbars . . . . .	15
Main Screen and Workspace . . . . .	15
Arranging Worksheet Windows . . . . .	16
Maximizing the Workspace . . . . .	16
Viewing and Hiding the Project Tree Window . . . . .	17
Message Window . . . . .	17
Accessing Compiler Errors . . . . .	17
Detaching the Message Window . . . . .	18
Viewing and Hiding the Message Window . . . . .	18
Cross Reference Window . . . . .	18
Detaching the Cross Reference Window . . . . .	20
Viewing and Hiding the Cross Reference Window . . . . .	20
Status Bar . . . . .	20
Editors . . . . .	21
Project Manager . . . . .	21
Project Tree Editor . . . . .	21
Instance Tree . . . . .	22

# Table of Contents

---

Graphic Editor . . . . .	23
Text Editor . . . . .	24
Pagelayout Editor . . . . .	24
The Edit Wizard . . . . .	25
Overview Window for Graphical Worksheets . . . . .	28
Saving Changes . . . . .	29
Closing Worksheets. . . . .	29
Exiting MachineLogic. . . . .	29
<b>Chapter 3 Using the MachineLogic Templates . . . . .</b>	<b>31</b>
Creating a Project . . . . .	32
Configuring a Project. . . . .	35
Configuring Program Organizational Units (POUs) . . . . .	35
MyProgram POU . . . . .	36
Adding a New POU . . . . .	36
Configuring the Physical Hardware . . . . .	38
Configuring the Resource using serial COM port . . . . .	38
Configuring the Resource using TCP/IP . . . . .	39
Defining Retentive and Non-retentive Regions . . . . .	40
Adding a Task . . . . .	42
Configuring the I/O . . . . .	44
Testing a Project . . . . .	45
Downloading a Project for Testing. . . . .	45
COM port Download . . . . .	45
Ethernet Download . . . . .	47
Running a Project for Testing . . . . .	47
Backing Up a Project. . . . .	49
Downloading Project Files . . . . .	50
Cables . . . . .	50

# Table of Contents

---

MachineLogic Slot Card only . . . . .	50
Download . . . . .	50
Running a Project from the Runtime System . . . . .	51
<b>Appendix A MachineLogic Extended Flags . . . . .</b>	<b>53</b>
<b>Appendix B MachineLogic System Flags. . . . .</b>	<b>57</b>
<b>Index. . . . .</b>	<b>59</b>

# Overview and Support Services

Welcome to MachineLogic. MachineLogic is a complete programming environment designed to create programs for performing operations that are typically performed by PLCs. MachineLogic is based on the IEC 1131-3 specification in order to provide a level of compatibility with other software and hardware that is based on this standard.

This chapter explains what is in this User Guide. It also describes the documentation standards used in the Guide and important CTC customer support services.

### Chapter Contents

---

<i>What's in this User Guide?</i>	2
<i>Documentation Standards</i>	3
<i>Customer Support Services</i>	4

## What's in this User Guide?

This Guide describes the MachineLogic user interface and provides instructions on how to create and download MachineLogic projects.

This guide contains the following chapters:

Chapter	Title	Contents
<b>1</b>	Overview and Support Services	Guide contents, users, and Support Services.
<b>2</b>	Introducing MachineLogic	Hardware and software requirements, user interface.
<b>3</b>	Using the ML Templates	Create, backup, and transfer projects.
<b>Appendix A</b>	MachineLogic Extended Flags	Extended lags and their address, type, name, and description
<b>Appendix B</b>	MachineLogic System Flags	System flags and their address, type, name, and descriptions





## Documentation Standards

This Guide uses the following documentation standards:

Style	Type of Text
<b>Bold</b>	Important terms
<code>Courier font</code>	Text to be entered from the keyboard.
<b>Note</b>	Alternative approaches or issues.
<b>Important</b>	Information to save time and minimize problems.
<b>Warning</b>	Information that will prevent equipment damage or personal injury.

## ISO Symbols

Symbol	Meaning
	International Standards Organization (ISO) symbol for Caution (ISO 3864 No. B.3.1). Denotes activity that could affect operation of the computer if instructions are not followed.
	PowerStation PS Series User Guide ISO symbol for Caution—risk of electrical shock (ISO 3864 No B.3.6). Denotes activity that could cause personal injury from electrical shock or damage to equipment if instructions are not followed.

## Customer Support Services

We welcome your thoughts and suggestions on our products and services. You can contact us by telephone, email, or fax. You can also visit Parker Automation - CTC on the World Wide Web to learn the latest about our hardware, software, and customer support services.

<b>Telephone</b>	513-831-2340
<b>Technical Support</b>	513-248-1714
<b>Fax</b>	513-831-5042
<b>E-mail</b>	sales@ctcusa.com support@ctcusa.com DocComments@ctcusa.com
<b>World Wide Web</b>	www.ctcusa.com

We recognize that every customer and every application have different support needs, so we offer four types of customer support services:

- Technical Support
- Training and New Business Development
- Product Support Program
- Documentation

## Technical Support

The Technical Support department welcomes your questions as you develop or run your applications. We offer complimentary support for any customer, whether you are an end-user, original equipment manufacturer (OEM), system integrator, or distributor.

If you have a question about MachineShop, Interact, or MachineLogic, be sure to complete the following steps:

1. Check the Readme files installed with the software. These files provide general information about the release.
2. Consult the appropriate documentation and other printed materials included with MachineShop, Interact, and MachineLogic.
3. Check the online help. MachineShop, Interact, and MachineLogic each have extensive online help facilities that cover all aspects of the product.

If you cannot find a solution using one of the preceeding sources:

4. Call your CTC products distributor for support.
5. View the Technical Support web page at [www.ctcusa.com](http://www.ctcusa.com). You'll find answers to Frequently Asked Questions, application notes, product user guides, and software downloads that may be helpful.
6. Call Technical Support at 513-248-1714, 8:00 a.m. to 5:00 p.m., Eastern Time.

You can call this same number after hours for emergency assistance.

## Training and New Business Development

### Training

We offer training on all of our products, either at Parker Automation - CTC, in our state-of-the-art training center, or at your site. You can learn how to:

- Write custom interfaces
- Develop specialized applications
- Implement your complete operator interface application

To contact the Training Coordinator:

<b>Telephone</b>	1-800-233-3329
<b>E-mail</b>	<a href="mailto:training@ctcusa.com">training@ctcusa.com</a>

### Application Engineering Services

We offer consulting services through our Application Engineering Services group. Our Application Engineering group can build your application from the ground up:

- Write custom communications drivers
- Design special modules to perform functions unique to your application

Our specialists are flexible, so they can create a project development schedule that meets your needs. You can contact Application Engineering Services in one of the following ways:

<b>Telephone</b>	1-800-233-3329
<b>E-mail</b>	<a href="mailto:appeng@ctcusa.com">appeng@ctcusa.com</a>

## Product Support Program

The Product Support Program (PSP) is designed to keep you up-to-date with the current versions of Parker Automation - CTC software. It's an easy, cost-effective way to receive the most recent CTC software and associated utilities. The PSP consists of a renewable, one-year membership that provides you with the following:

- Free upgrades and utilities
- E-mail notification of software updates
- Valuable tools for MachineShop, Interact, and MachineLogic

Single-user, multi-user, and corporate licenses are available. To learn more about PSP, contact us:

<b>Telephone</b>	1-800-233-3329
<b>E-mail</b>	sales@ctcusa.com

## Documentation

You can download CTC product documentation from our Web site at [www.ctcusa.com](http://www.ctcusa.com). Click **Support** and then **Product Manuals**.

We also welcome your comments on this User Guide or any of our other documentation. Please e-mail your comments to: [DocComments@ctcusa.com](mailto:DocComments@ctcusa.com)

# Introducing MachineLogic

This chapter describes the system requirements for the MachineLogic development and runtime systems. It also describes how to open MachineLogic and introduces you to the MachineLogic user interface.

If you would like information about some IEC basics, refer to the *MachineLogic Runtime Reference Guide* or to the available online help.

### Chapter Contents

---

<i>Development System Requirements</i>	8
<i>Runtime System Requirements</i>	9
<i>Opening MachineLogic</i>	11
<i>MachineLogic User Interface</i>	12
<i>Editors</i>	21
<i>The Edit Wizard</i>	25
<i>Overview Window for Graphical Worksheets</i>	28
<i>Saving Changes</i>	29
<i>Closing Worksheets</i>	29
<i>Exiting MachineLogic</i>	29

## Development System Requirements

The development system is the computer you plan to use to create and manage your MachineLogic projects. This computer will not be used to run the projects that you create. Instead, you will use this computer to download your projects to a CTC PowerStation or a third party PC that will be used to run your projects. The PowerStation is referred to as the *runtime system* throughout this manual.

### Hardware Requirements

Your development system must meet or exceed the following system requirements in order for MachineLogic to perform as intended:

Minimum hardware requirements:

- Processor: Pentium 200MHz
- Memory: 64 MB RAM
- Fixed Storage: 60 MB of free hard disk space
- Removable Storage: CD-ROM drive
- Input Devices: keyboard and Microsoft compatible mouse
- 10MB or 100 MB Ethernet interface or serial port

### Software Requirements

Your MachineLogic development PC requires the following software in order to develop MachineLogic projects:

- Operating System: Windows 95/98, NT, 2000, ME

**Note** DCOM 95 must be installed if using Windows 95/98. A self-extracting executable install is provided on the MachineLogic install disk in the DCOM 95 directory.

- Interact: Version 6.0 or later
- MachineShop Toolbar Version 1.03
- MachineLogic: Version 2.11 or later recommended (minimum Version 2.0)

### Installing the Software

Refer to the *MachineShop Installation Booklet*, located in the CD jewel case, for instructions on installing MachineLogic.

**Important** In order to run MachineLogic in the development environment, the Windows development option on your security key must be enabled. Refer to the *MachineShop Getting Started Guide* for details on the security key.

## Runtime System Requirements

The MachineLogic runtime system is the CTC PowerStation or the third party PC you plan to use to run your MachineLogic projects. This computer will only be used to run the projects you create. You will develop your projects on a development system and download them to this computer.

Requirements depend on which card will be used, either the MachineLogic Control Adapter card (MLCA) or the newer MachineLogic Slot Card (MLSC).

### MLCA-specific Requirements

#### MLCA Hardware Requirements

Minimum hardware requirements for a Runtime System using the MachineLogic Control Adapter card:

- Processor: 80386-SX 25MHz or higher
- Memory: 4MB
- Fixed Storage: 8MB
- Video: 320 x 240 or 640 x 480 resolution, 16 colors
- Input Device: Microsoft compatible mouse or equivalent
- Third Party I/O Card and free card slot
- MachineLogic Control Adapter card and one free EISA slot
- 10MB Ethernet network interface or serial port

#### MLCA Software Requirements

Your runtime PC requires the following MachineLogic software in order to run MachineLogic projects:

- Operating Systems: MS-DOS 6.22 or later
- Interact Runtime: Version 6.0 or later
- MachineLogic Runtime: Version 1.11 or later
- MachineShop Shell: Version 100 or later

### MLSC-specific Requirements

#### MLSC Hardware Requirements

Minimum hardware requirements for a Runtime System using the MachineLogic Slot Card:

- Processor: Pentium-based
- PCI Memory: 16KB of free upper memory (UMB) region which can be mapped to a PCI slot (PCI Powerstations meet

this requirement)

- Memory: as dictated by Interact runtime
- Fixed Storage: as dictated by Interact runtime
- Input Device: Microsoft compatible mouse or equivalent
- Third Party I/O card which will mount on MLSC  
AND/OR 10MB or 100MB Ethernet network interface
- MachineLogic Slot Card and free PCI slot

### MLSC Software Requirements

The MachineLogic software is included on the MachineLogic Slot Card, and does not need to be installed on your runtime machine.

Your runtime PC requires the following software in order to run MachineLogic projects:

- Operating Systems: MS-DOS 6.22 or later
- Interact Runtime: Version 6.0 or later
- MachineShop Shell: Version 107 or later

**Important** One of the MachineLogic cards (MLCA or MLSC) must be installed before you can use MachineLogic Runtime.

#### Installing the Hardware

In order to run MachineLogic projects on your runtime system, you must install either the MLCA or the MLSC and the appropriate I/O card. Refer to the *MLCA or MLSC Installation* documents and any *I/O Interface Card* documentation you might have for step-by-step instructions on how to install the cards.

#### Installing the Software

Refer to the *MachineShop Installation Booklet*, located in the CD jewel case, for step-by-step instructions on how to install and setup the MachineLogic runtime software.



## Opening MachineLogic

MachineLogic is a Windows application designed to run in the Microsoft Windows environment. MachineLogic must be opened from the MachineShop toolbar.

**To open MachineLogic, complete this step:**

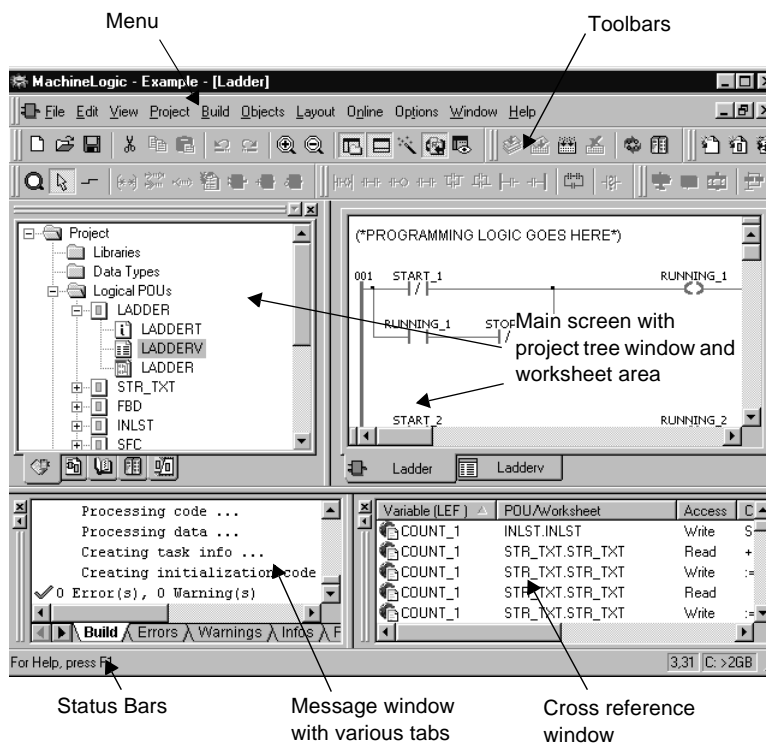
- Click the **MachineLogic** button on the MachineShop Toolbar or select **MachineLogic** from the Tools menu.

If you start the program for the first time, it opens without any project. Otherwise, the last project you were working on displays on the screen.

## MachineLogic User Interface

The MachineLogic user interface consists of the following parts: Menus, Toolbars, Status Bar, Worksheet area, Project Tree window, Message window and Cross Reference window.

A typical MachineLogic screen is displayed as shown below:



The Project Tree window, Message window, Cross Reference window and Status Bar can be optionally displayed in MachineLogic.

### Menus

Below the title bar are conventional Windows menus that are used to gain access to the wide variety of functions available in MachineLogic.

### File Menu

The File menu is used to manage, save and zip/unzip projects. It also contains commands for printing, print setup, print preview, and print project.

## **Edit Menu**

The Edit menu includes the commands for editing such as marking, choosing different working modes, or cutting and pasting. You can search and replace text strings in textual worksheets including description worksheets, variable worksheets, or structured text worksheets. The find and replace also works globally, letting you scan through all of the sections (POUs, worksheets) of your project.

## **View Menu**

The View menu includes the commands to hide or show the different windows of the user interface and the status bar. The different windows include the Project Tree window, the Message window, Cross Reference window, the Watch window, the Edit Wizard, and the Logic Analyzer window.

## **Project Menu**

The Project Menu allows you to insert new objects such as data type worksheets and announced libraries, POUs and configurations.

## **Build Menu**

The Build menu includes the different commands for compiling the project. Compiling means translating and transforming the contents of the worksheet into special code that will be executed by your runtime PC.

## **Objects Menu**

The Objects menu is available when you are using an editor. The Object menu allows you to insert a new variable into the variable list of the current POU. When editing a graphical worksheet, the submenu provides additional menu items to insert and edit graphical objects such as connectors, jumps, contacts, and coils. Depending on the graphical language you are using, some items may be grayed.

## Layout Menu

The Layout menu is available when you are using the graphic editor. It includes several designing utilities. For example, you can display page borders or a grid for better organizing the content of your worksheets. Or, you can zoom into and out of the worksheet, modify the worksheet size, the auto scroll speed and the object size. The Layout menu also disposes of some features for the online layout.

## Online Menu

The Online menu includes the commands for debugging a project, calling the Resource Control, setting up the Logic Analyzer, and activating the powerflow. In addition, the Online Layout command allows you to set the appearance of the graphical worksheets in online mode.

## Options Menu

The Options menu allows you to access the Shortcut Keys and the Preference dialog as well as other optional tools such as the page-layout editor. The Shortcut Keys dialog allows you to define your own keyboard shortcuts or customize the default shortcuts. The Preferences dialog allows you to customize the menus, toolbars, text editors and text colors.

## Window Menu

The Window menu allows you to arrange the windows and the symbols on your screen and to close all open windows in one step.

## Help Menu

The Help menu includes the commands for finding help on MachineLogic. The Contents command opens the MachineLogic Help window. From here you can browse for the desired information.

## Toolbars

The Toolbars provide quick access to the commands you use most often in MachineLogic. The program provides several Toolbars which contain different buttons. The Toolbars can be categorized as follows:

- **General Toolbars** - The General Toolbars include buttons that are available everywhere in the program.
- **Specific Toolbars** - The Specific Toolbars include buttons that can be used only in specific editors. All the buttons are visible but will be grayed when they cannot be used in a specific editor.

The name of each button on the Toolbar can be seen by placing the mouse cursor on the button. Additionally, the Status Bar displays the function of the desired button.

## Detaching the Toolbars

The default position for the Toolbars is below the Menu Bar but you can detach each Toolbar. The Toolbar is then displayed in a window that can be resized and moved to any location on your screen.

### To detach a Toolbar, complete this step:

- Double-click on the gray Toolbar background.

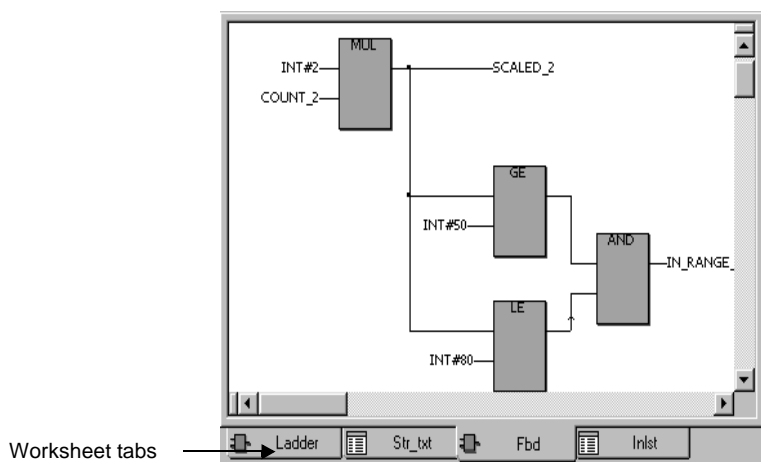
### To reinsert the Toolbar window, complete this step:

- Double-click on the blue Toolbar window title bar.

## Main Screen and Workspace

The main screen is divided into two parts: the Project Tree window and the workspace. The workspace contains the opened worksheets. You can open a worksheet by double-clicking on the corresponding worksheet icon in the project tree.

If several worksheets are opened, only one worksheet is visible in the workspace. A sheet tab is assigned to every opened worksheet as shown below:



You can select or activate a particular worksheet by clicking on the corresponding tab.

## Arranging Worksheet Windows

Using the Window menu, you can arrange your worksheet windows in a desired combination. This allows you to easily prepare the workspace for different working phases.

**To arrange the worksheet windows in the workspace, complete this step:**

- Select the **Cascade** command or the **Tile** command from the Window menu.

## Maximizing the Workspace

You can also maximize the workspace, which means, that the project tree is not displayed and the workspace is enlarged to the whole main screen width. This could be useful for displaying large networks in graphical editors.

## Viewing and Hiding the Project Tree Window

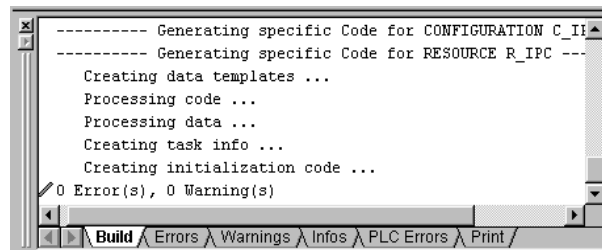
You can toggle between hiding and displaying the Project Tree window.

**To display or hide the Project Tree window, complete this step:**

- Select the **Project Tree Window** command from the View menu or click on the **Project Tree Window** button on the Toolbar.

## Message Window

The Message window is located right below the main screen. It contains several pages that can be accessed by clicking on the corresponding tab. An example of the Message window appears below.



## Accessing Compiler Errors

The Message window displays status information during compiling such as the compiler errors and warnings. One of the main benefits of the Message window is that you can directly access the error or warning in the worksheet.

**To access a compiler error in a worksheet, complete the following steps:**

1. Click on the **Errors** tab in the Message Window.
2. Double-click the corresponding error.

## Detaching the Message Window

The Message window is a dockable window. This means that you can detach the window from the desktop. You can then change the size and move the window to any position on the screen.

### To detach the Message window, complete this step:

- Double-click the gray window border in the Message Window.

### To reattach the Message window, complete this step:

- Double-click in the blue window title bar of the Message window.

## Viewing and Hiding the Message Window

You can toggle between hiding and displaying the Message window.

### To display or hide the Message window, complete this step:

- Select the **Message Window** command from the View menu or click on the **Message Window** button on the Toolbar.

## Cross Reference Window

The Cross Reference window contains all external variables, local variables and function blocks, which are used with the current project. It is a helpful tool for debugging and fault isolation.

**Note** Every POU contains its own local data. This means if you open a particular worksheet, the local variables in the cross reference list are updated.

In order to display the cross references in the Cross Reference window, you must first build the cross reference list.

### To build the cross reference list, complete this step:

- Select the **Build Cross References** command from the Build menu.

The cross reference list is created automatically and is displayed in the Cross Reference window. An example of the



cross reference list appears below:

Variable (LEF)	POU/Worksheet	Access	Command	Type	Comment	Line/Position(X/Y)
COUNT_1	INLST.INLST	Write	ST	INT	Variable comment goe...	6
COUNT_1	STR_TXT.STR_TXT	Read	+	INT		2
COUNT_1	STR_TXT.STR_TXT	Write	:=	INT		2
COUNT_1	STR_TXT.STR_TXT	Read		INT		4
COUNT_1	STR_TXT.STR_TXT	Write	:=	INT		4
COUNT_2	FBD.FBD	Read		INT		7/13
COUNT_2	INLST.INLST	Write	ST	INT		7
COUNT_2	SFC.ACTION1.ACTION1	Read	=	INT		1

For each entry the following information is available:

- **POU/Worksheet** - The POU name in which the variable/FB is declared and the particular worksheet name where it is implemented.
- **Access** - The access of the variable is Read (Load) or Write (Stored).
- **Command** - The Command in which the variable is used. This is only applicable for IL and LD code body worksheets.
- **Type** - The associated data type.
- **Comment** - The user-defined comment.
- **Line/Position(X/Y)** - The line number in the textual worksheets or the element position in the graphical worksheets.

You can open the corresponding worksheet in which a particular variable is used by double-clicking on the particular variable in the Cross Reference window. The worksheet is opened automatically and the variable is marked. Furthermore, a variable is marked in the Cross Reference window if you select the variable in the editor.

You can filter the cross reference list to show only a particular subset of variables or function blocks. You can also sort the list entries alphabetically. Each column can be used as sort criterion. The entries can be sorted in an ascending and descending order. Refer to the available online help for instructions on how to filter and sort the cross reference list.

## Detaching the Cross Reference Window

Like the Message window, the Cross Reference window is dockable. This means that you can detach the window from the desktop. See *Detaching the Message Window* for instructions on how to detach and reattach the window.

## Viewing and Hiding the Cross Reference Window

You can toggle between hiding and displaying the Cross Reference window.

**To display or hide the Cross Reference window, complete this step:**

- Select the **Cross References Window** command from the View menu or click on the **Cross References Window** button on the Toolbar.

## Status Bar

The Status Bar displays different messages while you are working the program. The left part of the Status Bar provides information about the operations you have done or displays system messages. If your mouse is placed on a button or a menu item, the status bar displays a short description of the desired button or menu item.

The fields on the right side of the Status Bar show the cursor position in the editor. For example, in the graphical editor, the x-y coordinates are shown. In the text editors, the fields display the current row and column. Beneath the cursor position, the free hard disk space is displayed. If there is not enough disk space available, this field appears red.

The different states of the programming system are represented in the Status Bar by colors. The following colors are used for the different states:

- gray: offline
- green: online
- red: timeout

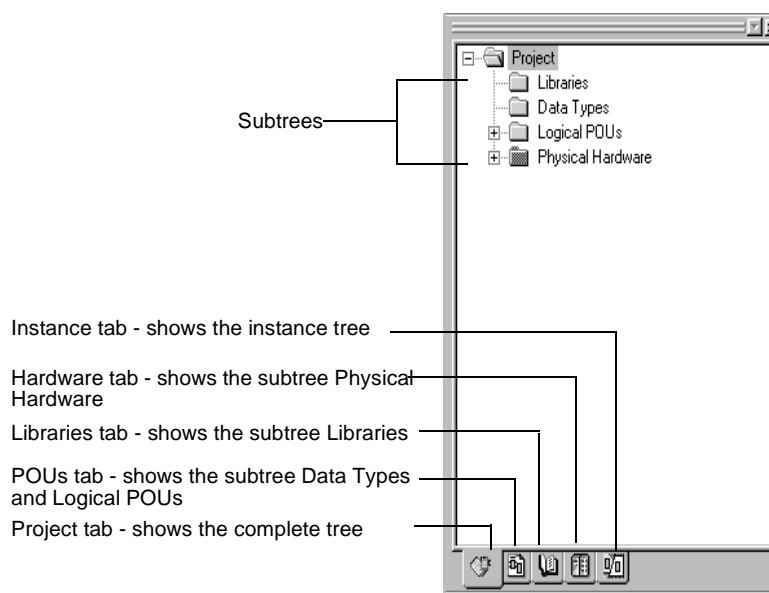
## Editors

### Project Manager

The project manager is an easy and powerful tool for program organization and project management. It includes the Project Tree Editor and the Instance Tree.

### Project Tree Editor

The Project Tree Editor allows you to edit the structure of your project. A sample of the Project Tree Editor appears below.



You can edit your project within four subtrees:

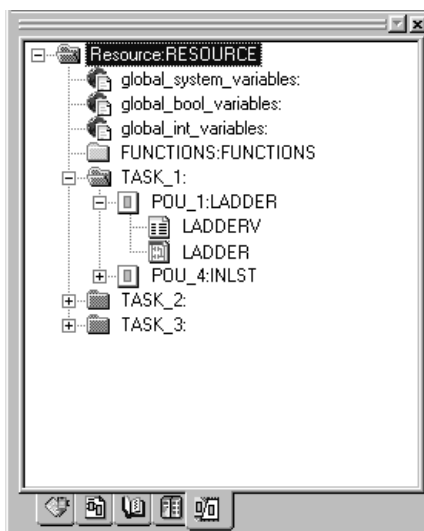
- **Library subtree** - The Library subtree includes all the libraries that are used in your project. Libraries are either user libraries or firmware libraries.
- **Data Types subtree** - The Data Types subtree includes data type declaration worksheets.

- **Logical POU's subtree** - The Logical POU's subtree includes all the POU's that are used in your projects.
- **Physical Hardware subtree** - The Physical Hardware subtree includes the configuration element icons such as global variables and tasks. The configuration elements available in this subtree may vary from one runtime system to another.

You can display the complete project tree or only particular subtrees. For this purpose, the Project Tree window provides several tabs at the bottom of the window. The tabs can be used to switch between the various views.

## Instance Tree

The Instance Tree displays the available resources, tasks and associated programs with all the functions and instantiated function blocks of a resource. The tree structure is created during the compilation of the resource. Consequentially the Instance Tree cannot be edited. The Instance Tree is used to open the worksheets in online mode. A sample of the Instance Tree appears below:



**Graphic Editor**

The Graphic Editor is one of the editors that you use for programming graphical code body worksheets for the POUs. MachineLogic supports three graphical programming languages: Sequential Function Chart (SFC), Function Block Diagram (FBD), and Ladder Diagram (LD).

The following features facilitate editing in the graphical languages:

- Edit Wizard provides full edit functionality for inserting and exchanging functions and function blocks.
- All graphical editors provide simple keyboard operation for inserting and scrolling.
- Duplication of function inputs can be done directly via the keyboard, the Toolbar, and the menu.
- Negation of Inputs, Outputs, Contacts, and Coils can be done directly via the keyboard, the Toolbar, and the Menu Bar.
- Items can be inserted directly on a line or to the inputs or outputs of already existing items (only in the FBD editor).
- Splitter and overview windows are available.
- Freestyle editing allows you to arrange items smoothly wherever you want.
- Double-clicking on user functions and function blocks opens the contents of a user POU.

You can mix the graphical programming languages in one project. MachineLogic checks all user entries to detect and avoid invalid connections (e.g. connection between two outputs). While inserting new graphical elements, the layout of an existing network is modified automatically.

**Text Editor**

The Text Editor is used for programming textual code body worksheets of the POU's. MachineLogic supports two textual programming languages: Instruction List (IL) and Structured Text (ST). The Text Editor is also used to edit data type worksheets, variable worksheets and I/O configuration worksheets.

The handling of the Text Editor is similar to the handling of a typical ASCII-editor.

The following features facilitate editing in IL and ST:

- An Edit Wizard provides full edit functionality for inserting pre-edited Data Types, Operands, Keywords, functions and function blocks.
- Full drag and drop functionality is provided.
- Optional line numbers can be displayed.
- Syntax highlighting is possible.
- Multiple undo/redo.
- Multiple zooms.
- Different views are possible via the splitter window.

**Pagelayout Editor**

The Pagelayout Editor provides an easy way to create pagelayouts for printing the project documentation. A pagelayout represents a template that is used to print the contents of a code body worksheet. It defines an area in which the content of the worksheet is going to be printed.

## The Edit Wizard

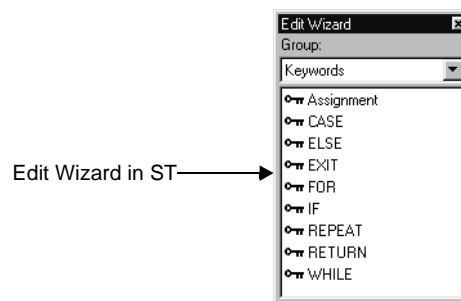
The Edit Wizard is a useful tool that facilitates the insertion and replacement of:

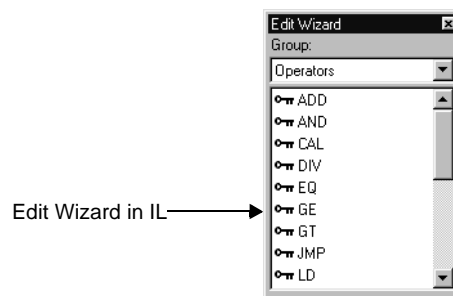
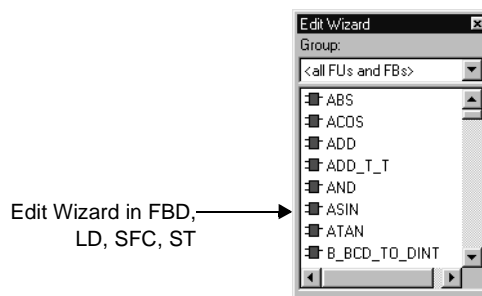
- keywords and statements (in ST),
- operators (in IL),
- function and function blocks (all languages).

Especially in the text editors, the usage of Edit Wizard provides the following advantages:

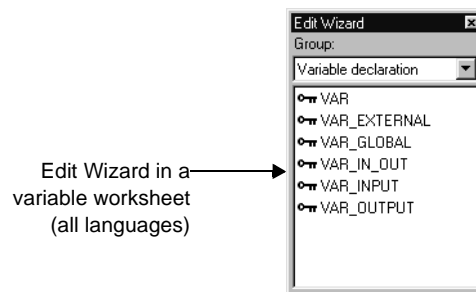
- It prevents you from entering syntactical faults, such as forgotten semicolons, selection or iteration statements without end statements, etc. This is done by inserting pre-edited statements, functions or function blocks. The statement structure is already prepared with place holders and the variables and values are inserted as comments.
- It is not necessary that you know the syntax of all the different statement types such as functions or function blocks.

Samples of the Edit Wizard for the various languages appear below and on the following page:

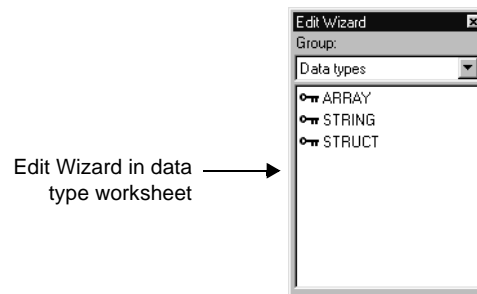




The Edit Wizard can also be used to insert variable definitions in the various variable worksheets as well as data types in a data type worksheet. Samples of the Edit Wizard for various worksheets appear below and on the following page:







**To use the Edit Wizard, complete the following steps:**

1. Click the **Edit Wizard** button on the Toolbar to activate the Edit Wizard.
2. Place the cursor in your worksheet where a new keyword, statement, function or function block is to be inserted.
3. In the Edit Wizard, select the code body **element** for the corresponding language from the Group list box.

A list of available elements for the selected group displays in the selection area.

4. Click the desired **keyword, statement, function, or function block** in the selection area.

The keyword, statement, function, or function block is inserted automatically at the cursor position. The actual variables and values are replaced by comments (green text enclosed by parenthesis and asterisks).

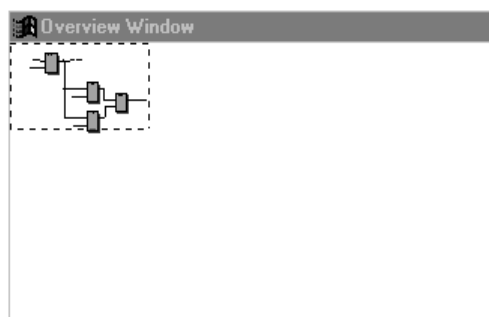
## Overview Window for Graphical Worksheets

When using the Graphic Editor, you can access the Overview Window to display an overview of the complete contents of the worksheet. The Overview Window simplifies the navigation in your worksheet.

**To access the Overview Window, complete the following steps:**

1. Double-click the desired graphical worksheet icon in the Project Tree Editor.
2. Select the **Overview Window** command from the Layout menu.

The Overview Window appears:



The Overview Window displays the entire contents of the current graphical worksheet. You can use the Overview Window to move the worksheet area in the graphic editor.

**To navigate the graphical worksheet using the Overview Window, complete the following steps:**

1. Click on the drawing in the Overview Window.  
The cursor changes to a hand symbol and the visible area is represented as a rectangle.
2. Hold down the left mouse button, and drag the rectangle to the area you want to be visible in the graphic editor.

## Saving Changes

While you are editing, you should regularly save the changes you have made.

**To save your current changes, complete the following step:**

- Click the **Save** button on the Toolbar.

## Closing Worksheets

Once you have finished editing the worksheet, you can close the worksheet.

**To close the worksheet, complete the following step:**

1. Select the **Close** command from the File menu.

If you have not saved your current changes, a MachineLogic dialog appears.

2. Click one of the following buttons:
  - Yes - to exit the worksheet with saving the changes
  - No - to exit the worksheet without saving the changes
  - Cancel - to return to the worksheet without exiting

## Exiting MachineLogic

When you exit MachineLogic, it does not matter whether one or several editors are still open or all windows are already closed.

**To exit MachineLogic, complete the following steps:**

1. Select the **Exit** command from the File menu.

If you have not saved your current changes, a MachineLogic dialog appears.

2. Click one of the following buttons:
  - Yes - to exit the worksheet with saving the changes
  - No - to exit the worksheet without saving the changes
  - Cancel - to return to the worksheet without exiting



# Using the MachineLogic Templates

MachineLogic comes with a number of pre-configured templates that you can use for new projects that you create. These templates include a small program and I/O configurations to help you get your project running quickly.

This chapter provides instruction on using the MachineLogic templates to create a new project. It also describes how to test the project on a runtime system, how to back up the project after changes have been made, and more.

If you would like more instruction on creating or configuring projects, refer to the *MachineLogic Tutorial* or to the available online help.

## Chapter Contents

---

<i>Creating a Project</i>	32
<i>Configuring a Project</i>	35
<i>Testing a Project</i>	45
<i>Backing Up a Project</i>	49
<i>Downloading Project Files</i>	50
<i>Running a Project from the Runtime System</i>	51

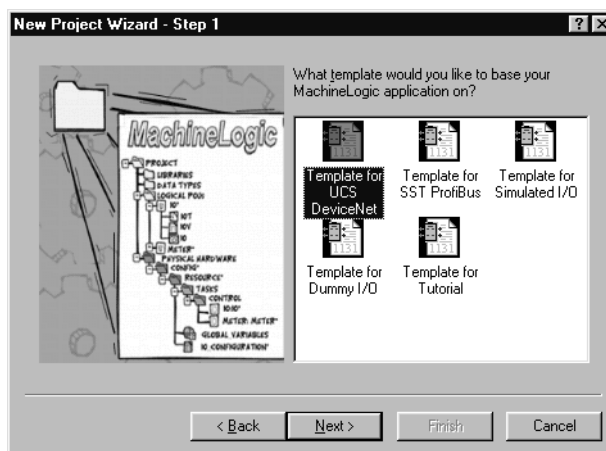
## Creating a Project

The first step is to create a new project using one of the MachineLogic templates. The templates include global variables and I/O configurations for the specific device drivers (e.g. DeviceNet and Profibus).

CTC recommends that you use MachineShop to create and manage your integrated projects. MachineShop collects all of the runtime application files and configuration data for control projects and HMI applications into a central location on your development system to facilitate storage, archival, and transfer-ence activities. For a complete discussion on MachineShop, refer to the *MachineShop Getting Started Guide*.

**To create a project using the Default Template, complete the following steps:**

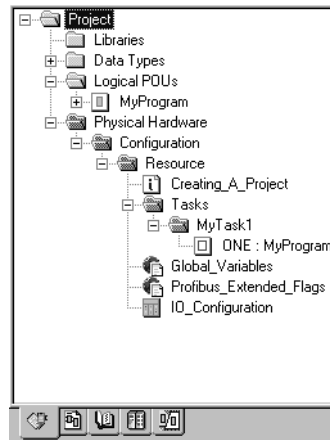
1. Start MachineShop.
2. Select the **New Project** command from the File menu.  
The New Project dialog box appears.
3. Enter a project name, a project description, and then select a Runtime type (MLPC or MLSC).
4. Click **OK** and begin the New Project Wizard.



5. Select one of the following templates from the list of available templates, and then click **Next**.
  - Template for UCS DeviceNet - includes global variables and I/O configuration for a UCS DeviceNet I/O driver.
  - Template for SST ProfiBus - includes global variables and I/O configuration for an SST Profibus I/O driver.
  - Template for Simulated I/O - includes global variables and I/O configuration for simulating a MachineLogic project on your development PC.
  - Template for Dummy I/O - includes global variables and I/O configuration for an application that does not communicate with I/O devices. This allows you to debug your application prior to putting the control online.
  - Template for Tutorial - includes global variables and I/O configuration for the MachineLogic Tutorial.
  - Other Templates may be available: Template for Enomb - includes global variables and I/O configuration for the Enomb driver.
6. Follow the instructions provided by the New Project Wizard to finish creating the project.
7. When the New Project Wizard is finished, click the **MachineLogic** button on the MachineShop Toolbar.

MachineLogic opens and your new project appears. Use the Project Tree Editor to view the default settings and make modifications to the project.

A sample of the Project Tree editor for a project that uses the Profibus template appears below:





## Configuring a Project

Even if you create a project using a template such as the Profibus Template, you will likely need to change some of the default settings to meet your project's needs. This section will describe the steps that are involved in configuring a project and preparing it for download.

### Configuring Program Organizational Units (POUs)

Program Organizational Units (POUs) are the language elements of a PLC program. They are small, independent software units containing the program code. The name of a POU should be unique within the project. POUs must not be recursive, meaning that the invocation of a POU should not cause the invocation of a POU of the same type.

Every POU consists of two different parts: The declaration part and the instruction or “code body” part. In the declaration part, all necessary variables are declared. In the instruction part, instructions are programmed in the desired programming language.

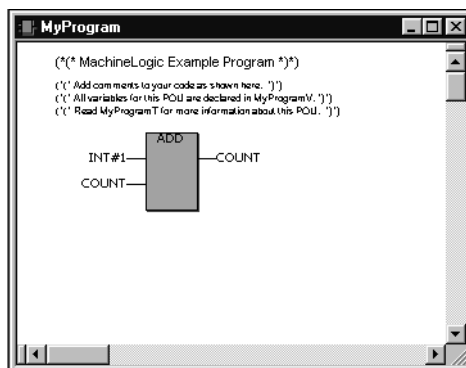
POUs are accessible from the Project Tree Editor. To edit or configure a POU, simply right-click the POU and a menu of available commands will display on the screen. These commands are also accessible from the main menu bar.

## MyProgram POU

Projects you create using the templates include one POU called “MyProgram.” This POU includes a small program that consists of an ADD Function Block which increments the variable COUNT each time the program is executed.

The MyProgram POU is designed specifically for applications that use the Function Block programming language. If you are developing an application that uses this language, you can begin entering the program logic immediately by clicking on the MyProgram icon in the MyProgram folder.

The MyProgram programming window appears.



## Adding a New POU

You can add additional POUs to this project.

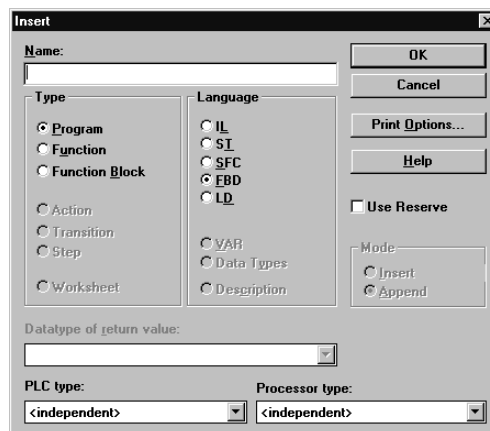
**To add a new POU, complete the following steps:**

1. Open the Project Tree Editor.
2. Right-click the **Logical POUs** folder in the Project Tree Editor.

The context menu appears.

3. Select the **Insert** command from the context menu.

The Insert dialog box appears.



4. Enter a **name** for the POU.
5. Select the program **Type** and the **Language** you plan to use.
6. Verify that the **PLC type** and the **Processor type** are set to <independent>.
7. Click **OK** to create the POU.

A new folder appears within the Logical POU folder of the Project Tree Editor.

You can begin entering the program logic immediately by double-clicking on the last icon in the new folder. The name of this icon will vary depending on the name you assigned to the POU.

## Configuring the Physical Hardware

Each project must have a configuration, a resource, and at least one task. The configuration defines the type of PLC to be used for the project. The resource defines the processor type to be used for the project as well as the communication protocols for the development system and the system resources for the runtime system. The task specifies the POU's to be executed by the PLC.

These components have been pre-defined for you in the MachineLogic templates.

### Configuring the Resource using serial COM port

You must configure communication protocols for the development system and the system resources for the runtime system.

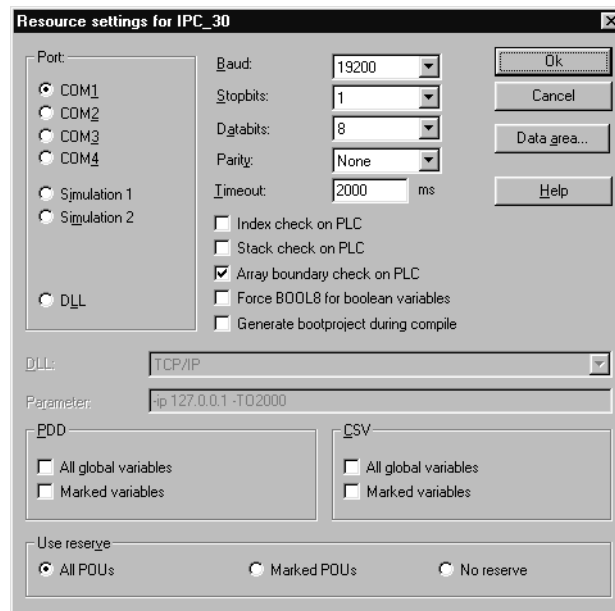
**To configure the resource settings, complete the following steps:**

1. Open the Project Tree Editor.
2. Double-click the **Physical Hardware** folder in the Project Tree Editor.
3. Right-click the **Resource** folder.

The context menu appears.

4. Select the **Settings** command from the context menu.

The Resource Settings dialog box appears.



5. Select the COM port you want to use for downloading the project to the target PC.

**Important** The settings should match those of COM1 on the runtime system.

## Configuring the Resource using TCP/IP

You must configure communication protocols for the development system and the system resources for the runtime system.

**Note** When using the MachineLogic Slot Card in your runtime system, you must configure the card's I/O Ethernet port (address and mask) from the MachineShop shell if you want to use TCP/IP.

**To configure the resource settings, complete the following steps:**

1. Open the Project Tree Editor.
2. Double-click the **Physical Hardware** folder in the Project

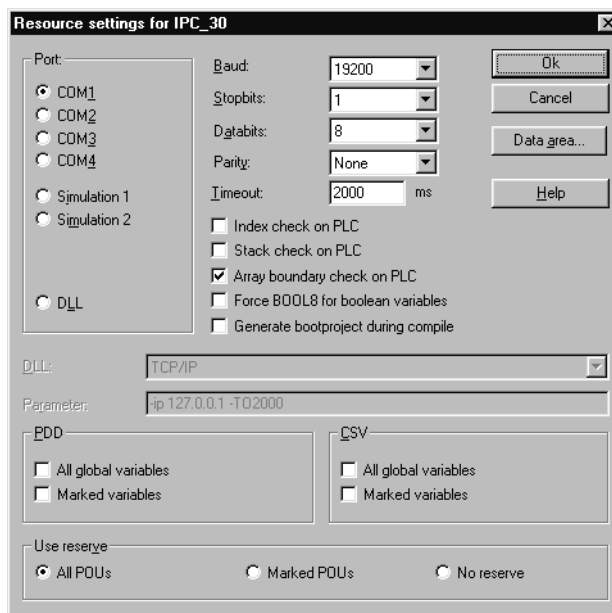
Tree Editor.

3. Right-click the **Resource** folder.

The context menu appears.

4. Select the **Settings** command from the context menu.

The Resource Settings dialog box appears.



5. Select DLL option.
6. Select TCP/IP as the DLL
7. For the Parameter, type the same IP address as was set up on the runtime system.

**Important** For the MLSC card, the settings should match those of the Ethernet port on the runtime system.

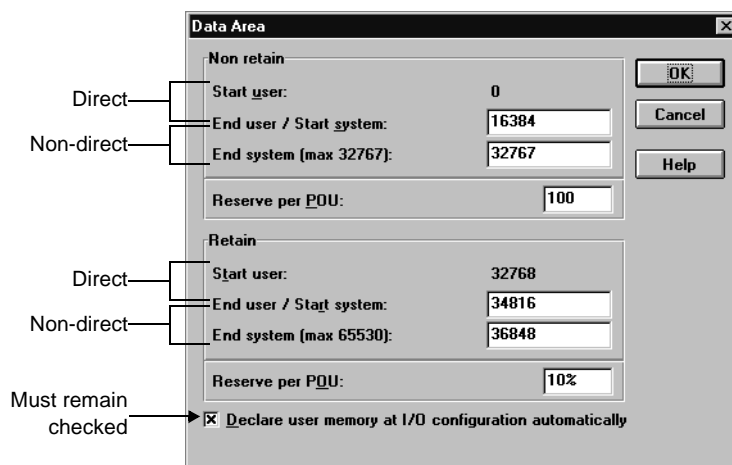
## Defining Retentive and Non-retentive Regions

While you are configuring the resource settings, you must also configure the retentive (retain) and non-retentive (non-retain) memory areas of the runtime system. This is accomplished

using the Data Area dialog box. For more information about MachineLogic memory management, see the Runtime Guides.

8. Click the **Data area** button.

The Data Area dialog box appears.



The above example shows the default settings for the Non-retain and Retain memory areas. If you find it necessary to use different settings, you may do so; however, the default settings were selected to accommodate most applications.

**Important** The settings above assume that you are using the default memory allocations for inputs and outputs (1k each). If you increase the memory allocations for inputs and/or outputs, the amount of space available to the non-retain area of memory may decrease.

Likewise, if you decrease the memory allocations for inputs and/or outputs, the amount of space available to the non-retain area of memory may increase.

By keeping the *Declare user memory at I/O configuration automatically* check box checked, the I/O configuration will be updated automatically whenever you change the Direct Non-retain memory region in the Data Area dialog box.

## Adding a Task

Each resource must include at least one task. The task specifies the POUs to be executed by the PLC. Projects you create using the MachineLogic templates have one predefined task called, “MyTask1.” This task is a cyclic task.

The MyTask1 configuration may be adequate for your project; however, if you need to add additional tasks, you can do so by following the procedure below.

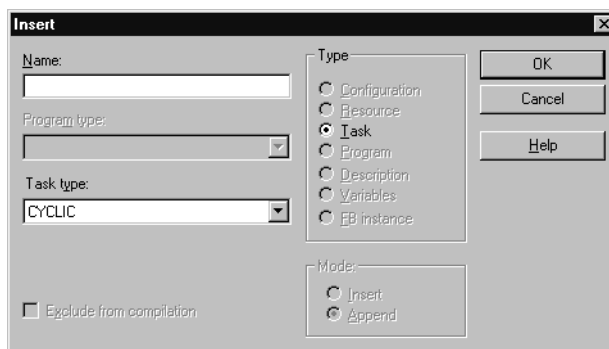
**To add a new task, complete the following steps:**

1. Open the Project Tree Editor.
2. Open the **Physical Hardware** folder in the Project Tree Editor.
3. Right-click the **Tasks** folder.

The context menu appears.

4. Select the **Insert** command from the context menu.

The Insert dialog box appears.



5. Enter a **name** for the task.
6. Verify that the **Type** is set to Task.
7. Verify that the **Task type** is set to CYCLIC.
8. Click **OK** to create the task.

A new folder appears in the Tasks folder in the Project Tree Editor.



### Associate a Program to a Task

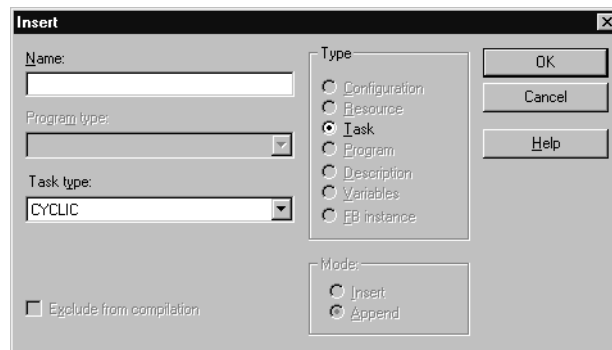
After you add a task, you must associate programs to the task. This means deciding in which task the program is processed.

9. Right-click the new **Task** folder.

The context menu appears.

10. Select the **Insert** command from the context menu.

The Insert dialog box appears.



11. Select **Program** from the Type group box.
12. Enter an **instance name** for the program.
13. Select the **Program Type** you want to insert.
14. Click **OK** to create the program.

The program appears in the Tasks folder in the Project Tree Editor.

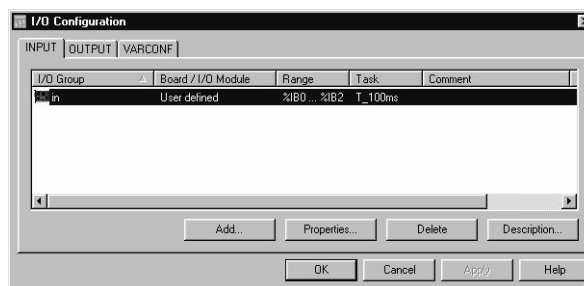
## Configuring the I/O

Each resource has an I/O configuration that corresponds to the type of I/O you will be using to communicate with your machinery. You define the input and output address ranges and their association with the different tasks using the I/O configuration dialog.

**To configure the I/O, complete the following steps:**

1. Open the Project Tree Editor.
2. Open the **Physical Hardware** folder in the Project Tree Editor.
3. Double-click the **Configuration** folder.
4. Double-click the **Resource** folder.
5. Double-click the **IO\_Configuration** icon.

The I/O configuration dialog appears:



Select the tab Input or the tab Output or the tab Varconf. Select the module to be changed, and click **Properties**. The Properties dialog appears.

6. Fill in or change the dialog fields. A description of the fields can be found in the online help.
7. Select **OK** when finished.

**Note** For instructions on how to configure your specific I/O bus type, refer to online help.

**Note** To add a new I/O configuration, click **Add** in the I/O Configuration dialog illustrated above.

## Testing a Project

After you create and configure a project, it is a good idea to test it on your runtime system. To test a project on your runtime system, you need to perform the following steps:

1. Download the project to the runtime system.
2. Start the project from your development system.

## Downloading a Project for Testing

To download a MachineLogic project to the runtime system for testing, complete the following steps

### COM port Download

#### Runtime system

**Note** For the MLSC only, you must connect the com cable between the Development workstation's COM1 port and the MachineLogic Slot Card's serial port, not the serial port of the runtime workstation.

To download a project for testing using the COM ports, follow these steps:

1. Connect a serial download cable between the COM1 port on the runtime system and the desired serial port on the development system.

**Note** The COM1 port on the runtime system is used only for downloading and debugging MachineLogic projects.

2. Make sure MachineLogic is enabled on the runtime system. Refer to the *MachineShop Shell Runtime Guide* for information on how to enable MachineLogic.
3. If you are also using COM1 to perform serial downloads from the MachineShop toolbar, you must set the network configuration in the MachineShop Shell to "download only". Reboot the runtime system if prompted and then exit the Shell once it is loaded again.

The Download Only option allows other applications, such as MachineLogic, to use the COM1 port once you exit the Shell. Refer to the *MachineShop Shell Runtime Guide* for instruc-

tions.

**Important** Interact cannot be in Run mode when you download the project from MachineLogic.

**Development system:**

4. Open the project to be downloaded.
5. Open the **Physical Hardware** folder in the Project Tree Editor.
6. Open the **Configuration** folder or the configuration folder that contains the resource you want to use to run this project.
7. Right-click the **Resource** folder or the resource folder that contains the controller where you want to download the project.
8. Select the **Settings** command from the Edit menu.
9. Select the COM port you want to use for downloading the project to the PowerStation. Select the communication port to which you connected the download cable in step 1.
10. Verify that the development communication protocols for the port you selected match those of COM1 on the runtime system.

The runtime system communication protocols are defined in the MachineLogic configuration file.

11. Click the **CONTROL** button on the MachineLogic Toolbar to download the project.

The resource control window appears.

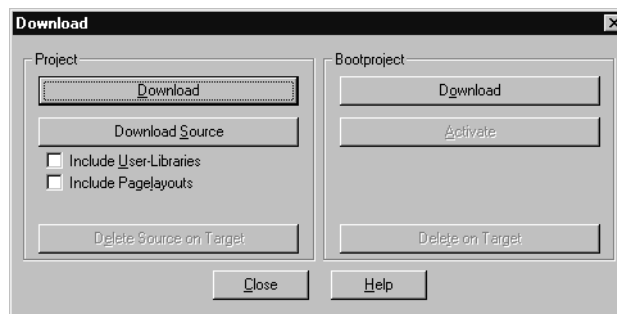


12. Click **Stop** if the current state is Run.



13. Click **Download**.

The Download window appears.



14. Click the **Project: Download** button.

This will load the project into the runtime system's volatile memory.

## Ethernet Download

You can also download a MachineLogic project to the runtime system for testing using Ethernet. The Ethernet cable goes from the Ethernet port on the development machine to either the Ethernet port on the runtime workstation or, if you have the MachineLogic Slot Card, on the Ethernet port on the card.

## Running a Project for Testing

Once the project has been downloaded to the runtime system, you are ready to run the project **from your development system**. You run the project from your development system through a serial

connection between the runtime system and the development system.

**To start a project from the development system, complete the following steps:**

1. Open the project that you created earlier in this chapter.
2. Download the project if you haven't already. Use the steps described in the *Downloading a Project* section in this chapter.
3. Open the **Physical Hardware** folder in the Project Tree Editor.
4. Open the **Configuration** folder or the configuration folder that contains the resource you want to use to run this project.
5. Open the **Resource** folder or the resource folder that contains the controller you want to use to run the project.
6. Click on the **Control** button on the MachineLogic Toolbar.

The resource control window appears.



7. Click one of the following buttons to begin running the project:
  - Cold - initializes all data and activates all user tasks
  - Warm - initializes non-retentive data and activates all user tasks
  - Hot - does not initialize data but does activate all user tasks (not available for starting a program immediately following download)
8. Click the **Stop** button to end program execution, or click the **Reset** button to delete the project and reset the runtime system.

## Backing Up a Project

It is a good idea to back up your project. CTC recommends that you back up your projects using MachineShop. MachineShop provides a Backup Project Wizard to automate the process of backing up your projects. For practice, we will back up the project that you created in the previous section.

**To back up a project, complete the following steps:**

1. Open the project you created in the previous section.
2. Select the **Create Backup** command from the MachineShop Tools menu.

The Create Backup submenu appears.

3. Select the **Project** command from the Create Backup submenu.

The Backup Wizard appears.

4. Follow the instructions provided by the Backup Wizard to finish backing up the project.

For more information about MachineShop, refer to the *MachineShop Getting Started Guide*.

**Important** MachineLogic also includes a utility for backing up the project source files. Refer to the MachineLogic online help for instructions on zipping the project source files into an archive file.

## Downloading Project Files

After you create, configure and test a project, you are ready to download the HMI and Control project files to the runtime system using MachineShop. MachineShop provides a Transfer Wizard to guide you through the process of downloading your projects.

**Notes** Make sure the MachineShop Shell has been started on the runtime system. Project files can only be transferred to the runtime system when the MachineShop Shell Main Menu is displayed on the screen.

Also, if you are using COM1 to perform serial downloads from MachineShop, make sure the network configuration in the MachineShop Shell is set to “download only”. This allows other applications (MachineLogic) to use the COM1 port once you exit the Shell.

### Cables MachineLogic Slot Card only

If using COM1 to perform serial downloads, the serial cable must connect the COM1 port on the development machine to the COM1 on the runtime workstation (not to the COM port on the MachineLogic Slot Card).

If using Ethernet to perform downloads, the Ethernet cable must connect the Ethernet port on the development machine to the Ethernet port on the runtime workstation (not to the Ethernet port on the MachineLogic Slot Card).

### Download To download the project files using MachineShop, complete the following steps:

1. Open the project to be downloaded.
2. Select the **Download** command from the MachineShop Transfer menu.

The Transfer Wizard appears.

3. Follow the instructions provided by the Transfer Wizard to finish downloading the project.

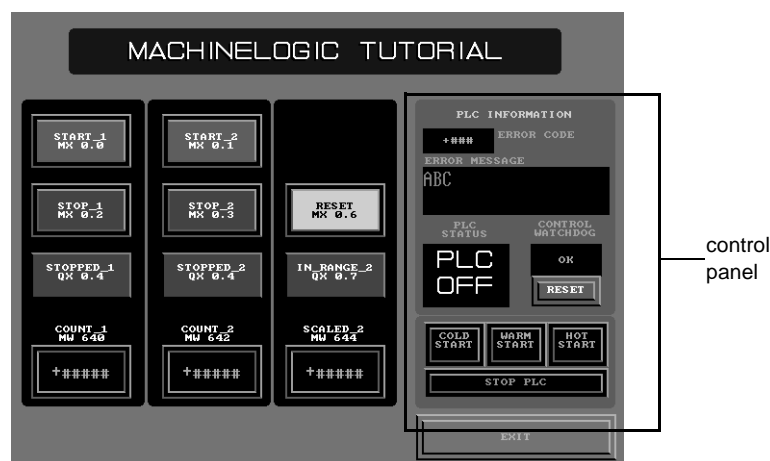
For more information about MachineShop, refer to the *MachineShop Getting Started Guide*.



## Running a Project from the Runtime System

If you need to start or stop your project on the runtime system without being connected to the development system or if you would like to access basic status information, you can create a panel in Interact that acts as a “control panel.” The control panel would have similar functionality to the resource control window found in MachineLogic.

If you want to create your own control panel in Interact, you can use the Interact Panel Toolkit Module (PTM). CTC has provided a sample control panel in the Tutorial application that you can use as a template. This panel is called Overview.ptm.



Finally, you will need to create a push button tool on your main panel that the operator can use to display the control panel during runtime.



## Appendix A MachineLogic Extended Flags

This appendix lists each of the available extended flags that may be specified from MachineLogic. The following table provides the address, data type, name, and a description for each flag.

Address	Data Type	Variable Name	Description
MB3.0	BYTE	CTRL_ACTION	PLC actions (stop, cold / warm / hot start)
MX3.0.0	BOOL	CTRL_ACTION_STOP	True := stop PLC
MX3.0.1	BOOL	CTRL_ACTION_COLD	True := cold start PLC
MX3.0.2	BOOL	CTRL_ACTION_WARM	True := warm start PLC
MX3.0.3	BOOL	CTRL_ACTION_HOT	True := hot start PLC
MX3.0.4	BOOL	CTRL_ACTION_STOPPED	True := PLC stopped (read-only)
MX3.0.5	BOOL	CTRL_ACTION_INITIALIZING	True := PLC initializing (read-only)
MX3.0.6	BOOL	CTRL_ACTION_STARTED	True := PLC started (read-only)
MB3.1	BYTE	CTRL_PROGRAM_STATUS	Control program status
MX3.1.0	BOOL	CTRL_NEW_PROGRAM	New control application downloaded (read-only)
MX3.1.1	BOOL	CTRL_WARM_START_ALLOWED	Warm start allowed (read-only)
MX3.1.2	BOOL	CTRL_HOT_START_ALLOWED	Hot start allowed (read-only)
MB3.2	BYTE	INTERACT_STATUS	Status of the Interact MMI Product
MB3.2.0	BOOL	INTERACT_WATCHDOG	Watchdog flips between zero and one every 55ms to allow control applications to monitor the status of Interact (read-only)
MB3.3	BYTE	CTRL_KERNEL_STATUS	Control kernel status
MX3.3.0	BOOL	CTRL_POWER_STATUS	True := Power Failure Detected (read-only)
MX3.3.1	BOOL	CTRL_WATCHDOG_STATUS	True := Watchdog Tripped (read-only)
MX3.3.2	BOOL	CTRL_RESET_WATCHDOG	True := Reset Tripped Watchdog
MX3.3.3	BOOL	CTRL_NVRAM_DISABLE	True := Disable write access to NVRAM
MX3.3.4	BOOL	CTRL_NVRAM_ENABLE	True := Enable write access to NVRAM
MX3.3.5	BOOL	CTRL_NVRAM_STATUS	True := NVRAM enabled (read-only) False := NVRAM disabled (read-only)
3.4	R e s e r v e d   f o r   F u t u r e   U s e		
3.5	R e s e r v e d   f o r   F u t u r e   U s e		
3.6	R e s e r v e d   f o r   F u t u r e   U s e		
3.7	R e s e r v e d   f o r   F u t u r e   U s e		
3.8	R e s e r v e d   f o r   F u t u r e   U s e		
3.9	R e s e r v e d   f o r   F u t u r e   U s e		
3.10	R e s e r v e d   f o r   F u t u r e   U s e		

Address	Data Type	Variable Name	Description
MB3.11	STRING	CTRL_PRODUCT_VERSION	Product name and version (read-only)
MW3.11	INT		Maximum string length
MW3.13	INT		Current string length
MB3.15	STRING		Product name and version (20 bytes)
MB3.35	BYTE		NULL terminator
MD3.36	DWORD	CTRL_PROGRAM_MEMORY	Control program memory available (read-only)
MD3.40	DWORD	CTRL_DATA_MEMORY	Control data memory available (read-only)
MD3.44	DWORD	CTRL_RETAIN_SIZE	Control retain memory size (read-only)
MD3.48	REAL	CTRL_SYSTEM_TICK	Control system tick (read-only)
MB3.52	STRING	CTRL_PROJECT_NAME	Control project name (read-only)
MW3.52	INT		Maximum string length
MW3.54	INT		Current string length
MW3.56	STRING		Project name (11 bytes)
MW3.67	BYTE		NULL terminator
MB3.68	STRING	CTRL_BOOTPROJECT	Control boot project name (read-only)
MW3.68	INT		Maximum string length
MW3.70	INT		Current string length
MB3.72	STRING		Boot project name (11 bytes)
MB3.83	BYTE		NULL terminator
MB3.84	STRING	CTRL_ZIPPEDPROJECT	Control zipped project name (read-only)
MW3.84	INT		Maximum string length
MW3.86	INT		Current string length
MB3.88	STRING		Zipped project name (11 bytes)
MB3.99	BYTE		NULL terminator
MW3.100	INT	CTRL_EXPT_NUMBER	Control exception number (read-only)
MB3.102	STRING	CTRL_EXPT_MESSAGE	Control exception error message (read-only)
MW3.102	INT		Maximum string length
MW3.104	INT		Current string length
MB3.106	STRING		Exception error message (58 bytes)
MB3.164	BYTE		NULL terminator
MB3.165	STRING	CTRL_EXPT_TASK_NAME	Control exception task name (read-only)
MW3.165	INT		Maximum string length
MW3.167	INT		Current string length
MB3.169	STRING		Task name (9 bytes)
MB3.178	BYTE		NULL terminator
MW3.179	INT	CTRL_EXPT_TASK_NUMBER	Control exception task number (read-only)
MW3.181	INT	CTRL_EXPT_TASK_INTERVAL	Control exception task interval (read-only)

Address	Data Type	Variable Name	Description
MW3.183	INT	CTRL_EXPT_TASK_WATCHDOG	Control exception task watchdog (read-only)
MW3.185	INT	CTRL_IO_DVR_ERROR	Control driver error number (read-only)
MB3.187	STRING	CTRL_IO_DVR_ERROR_STRING	Control driver error string (read-only)
MW3.187	INT		Maximum string length
MW3.189	INT		Current string length
MB3.190	STRING		Error string (64 bytes)
MB3.254	BYTE		NULL terminator
2048—6143	R e s e r v e d f o r I / O D r i v e r s		
6144—9999	R e s e r v e d f o r F u t u r e U s e		
10000—10039	R e s e r v e d S y s t e m A r e a		

For the MachineLogic Slot Card, there can be multiple drivers and therefore multiples of the following section covering Driver error and Driver error strings:

INT	CTRL_IO_DVR_ERROR	Control driver error number (read-only)
STRING	CTRL_IO_DVR_ERROR_STRING	Control driver error string (read-only)
INT		Maximum string length
INT		Current string length
STRING		Error string (64 bytes)
BYTE		NULL terminator

Therefore, the subsequent addresses will be offset.



## Appendix B MachineLogic System Flags

This appendix lists each of the available system flags that may be specified from MachineLogic. The following table provides the address, data type, name, and a description for each flag.

Address	Data Type	Variable Name	Description
MB1.0	BYTE	PLCMODE	Current PLC mode
MX1.0.0	BOOL	PLCMODE_ON	True := current PLC mode is POWER ON
MX1.0.1	BOOL	PLCMODE_RUN	True := current PLC mode is RUN
MX1.0.2	BOOL	PLCMODE_STOP	True := current PLC mode is STOP
MX1.0.3	BOOL	PLCMODE_HALT	True := current PLC mode is HALT
MX1.1.4	BOOL	PLCDEBUG_BPSET	True := one or more breakpoints are set on the PLC for debugging purposes
MX1.2.0	BOOL	PLCDEBUG_FORCE	True := one or more Input / Output variables are forced on the PLC for debugging purposes
MX1.2.3	BOOL	PLCDEBUG_POWERFLOW	True := power flow is active on the PLC for debugging purposes. The MachineLogic debug kernel samples state information during the execution of the CPU code using software interrupts. The software interrupts are set on addresses of the CPU code, generated by the internal MachineLogic compiler, which are related to the addresses of the intermediate PLC instructions downloaded by MachineLogic.
MW1.44	INT	PLC_TICKS_PER_SEC	Number of ticks per second used by MachineLogic as system time base. This value is related to the resolution of MachineLogic <ul style="list-style-type: none"> <li>• Software timer function blocks like TON, TOF or TP</li> <li>• To the shortest cycle time of the default user task := <math>2 * (1 \text{ sec} / \text{PLC\_TICKS\_PER\_SEC})</math></li> <li>• To the resolution of the task interval cycle time of user tasks from type INTERVAL</li> </ul>
MD1.52	DINT	PLC_SYS_TICK_CNT	This variable is incremented each MachineLogic system tick and holds therefore the number of counted system ticks.
MW1.60	INT	PLC_MAX_ERR	Maximum size of the MachineLogic error table.
MW1.62	INT	PLC_ERR_CNT	Current amount of errors beginning at location %MD 1.64
MD1.64	DWORD	ERROR	First error number
MD1.68	DWORD	ERROR_N	Next error number offset calculated by $64 + N * 4$ where N is the error index.

Address	Data Type	Variable Name	Description
MW1.1000	INT	MAXTASK	Maximum amount of tasks available.
MW1.1002	INT	CURTASK	Current amount of user task.
MB1.1004	ARRAY 1...10 OF BYTE	TASKNAME	Name of the task, NULL terminated string.
MW1.1014	INT	TASKPRIO	Priority of the task.
MW1.1018	INT	TASKPERIOD	Period of the task if the task type is CYCLE in milliseconds.
MW1.1020	INT	TASKSTACK	Stack size of the task.
MW1.1024	INT	TASKWATCHDOG	Watchdog time in milliseconds.
MW1.1032	INT	CURDURATION	Current task duration including preemption in ticks.
MW1.1034	INT	MINDURATION	Minimum task duration including preemption in ticks.
MW1.1036	INT	MAXDURATION	Maximum task duration including preemption in ticks.
MW1.1040	INT	CURDELAY	Current task delay including preemption in ticks.
MW1.1042	INT	MINDELAY	Minimum task delay including preemption in ticks.
MW1.1044	INT	MAXDELAY	Maximum task delay including preemption in ticks.
MB1.1068	ARRAY 1...10 OF BYTE	TASKNAME_N	Next task info offset calculated by $1004 + N * 64$ where N is the task number.



## Index

### A

- associating a program to a task 43

### B

- backing up project 49
- building cross reference list 18

### C

- Compiler Errors
  - accessing 17
- configurations, I/O 44
- configuring
  - associate program to a task 43
  - I/O 44
  - new POU 36
  - project 35
  - task 42
- Cross Reference list, building 18
- Cross Reference window 18
- Customer Support Services 4

### D

- Data Area dialog box 41
- development system 8, 46, 47
- documentation
  - standards 3
- Download window 47
- downloading
  - using MachineShop 50

### E

- Edit Wizard 25
- Editors
  - Graphic Editor 23
  - Instance Tree 22

- Pagelayout Editor 24
- Text Editor 24

### F

- firmware flags 53

### G

- graphic editor 23

### I

- I/O
  - add 44
  - configuring 44
- I/O configuration dialog 44
- Instance Tree 22

### M

- MachineLogic
  - Cross Reference window 18
  - development system 8, 46, 47
  - Edit Wizard 25
  - firmware flags 53
  - Graphic Editor 23
  - Instance Tree 22
  - menu descriptions 12
  - message window 17
  - Overview Window 28
  - Pagelayout Editor 24
  - Project Tree Editor 21, 33
  - runtime system 9, 45, 50, 51
  - software requirements 8
  - starting MachineLogic 11
  - system flags 57
  - Text Editor 24
  - user interface 12
  - workspace, main screen 15
- memory
  - non-retentive region 40
  - retentive region 40

# Index

---

Message window 17  
MLCA requirements 9  
MLSC requirements 9

## N

non-retentive memory 40

## O

Overview Window 28

## P

Pagelayout Editor 24  
physical hardware, configuration 38  
Product Support Program 6  
Program Organizational Units (POUs) 35  
programs 43  
project  
    backup 49  
    configuration 38  
    configure 35  
    final download 50  
    physical hardware 38  
    POUs 36  
    resource 38  
    run 47  
    run from runtime system 51

task 38  
testing 45  
Project Tree Editor window 21, 33

## R

resource control window 46, 48  
Resource Settings 39, 40  
resources, tasks 42  
retentive memory 40  
running project 47  
runtime system 9, 45, 50, 51

## S

software requirements  
    MachineLogic Development 8  
    MachineLogic Runtime 10  
starting MachineLogic 11  
system flags 57

## T

tasks 42  
Technical Support 4  
testing a project 45  
Text Editor 24  
Training and New Business Development  
    5