# GFK-1918



# GE Fanuc Manual Series 90-30

CIMPLICITY Machine Edition Logic Developer-PLC
Getting Started

# Getting Started

# Logic Developer - PLC
## PLC Programming Software

**Version 3.00**
June 2002
GFK-1918C

CIMplicity Machine Edition

FANUC

## Disclaimer of Warranties and Liability

The information contained in this manual is believed to be accurate and reliable. However, GE Fanuc Automation North America, Inc. assumes no responsibilities for any errors, omissions or inaccuracies whatsoever. Without limiting the foregoing, GE Fanuc Automation North America, Inc. disclaims any and all warranties, expressed or implied, including the warranty of merchantability and fitness for a particular purpose, with respect to the information contained in this manual and the equipment or software described herein. The entire risk as to the quality and performance of such information, equipment and software, is upon the buyer or user. GE Fanuc Automation North America, Inc. shall not be liable for any damages, including special or consequential damages, arising out of the use of such information, equipment and software, even if GE Fanuc Automation North America, Inc. has been advised in advance of the possibility of such damages. The use of the information contained in the manual and the software described herein is subject to GE Fanuc Automation North America, Inc. standard license agreement, which must be executed by the buyer or user before the use of such information, equipment or software.

## Notice

GE Fanuc Automation North America, Inc. reserves the right to make improvements to the products described in this publication at any time and without notice.


CIMPLICITY is a registered trademark of GE Fanuc Automation. Any other trademarks referenced herein are used solely for purposes of identifying compatibility with the products of GE Fanuc Automation North America, Inc..

We want to hear from you. If you have any comments, questions, or suggestions about our documentation, send them to the following email address:

doc@gefanuc.com

# Contents

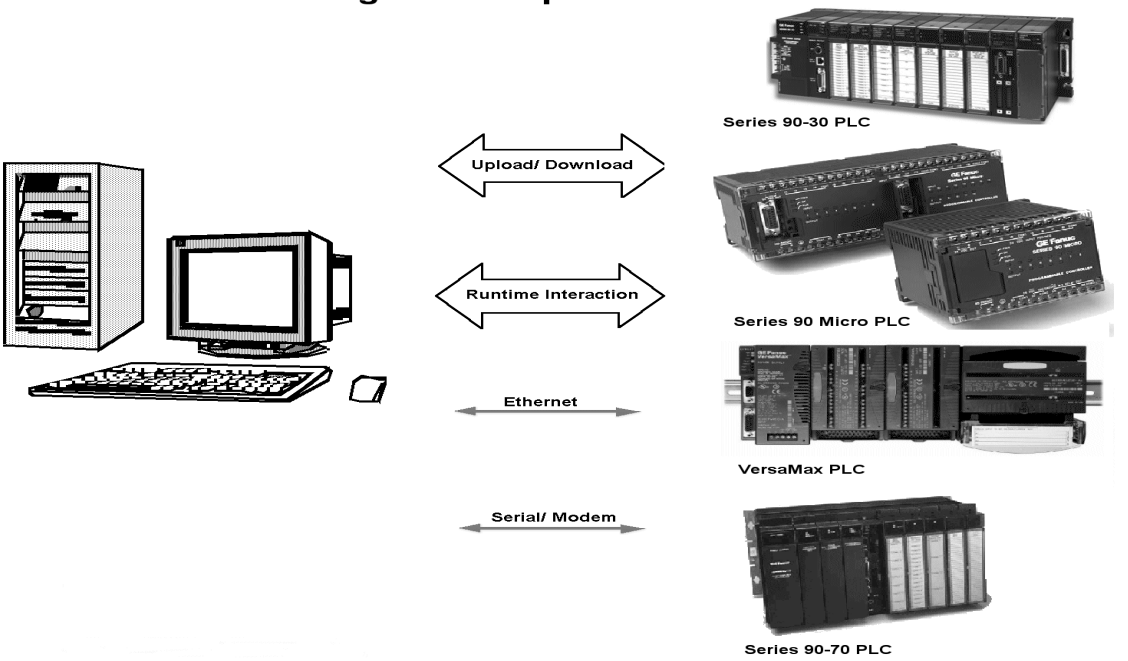
# 1 Welcome

Congratulations on your purchase of Logic Developer - PLC, the GE Fanuc PLC programming component of Machine Edition automation software. This software package provides all the tools necessary to create powerful control applications.

Logic Developer - PLC provides a way to configure your PLC hardware or remote I/O, create and edit PLC logic, upload and download projects, and monitor and debug the execution of control programs. Projects can be imported from Logicmaster, VersaPro, and CimplicityControl folders.

Logic Developer - PLC makes it possible to develop control applications on a personal computer and download them to a PLC via Ethernet or serial/modem connection.

**CIMPLICITY Logic Developer - PLC Environment**



Hosted in the Machine Edition environment, Logic Developer - PLC takes advantage of a powerful set of common programming tools (see page 14). The same tools can be applied to Logic Developer - PC (PC Control), Logic Developer - State, View and Motion components, providing a single programming environment. The Machine Edition environment unites and organizes components, providing data sharing and networked operation.

The following features are included in this version of Logic Developer - PLC:

- Hardware Configuration (HWC): a comprehensive tool for configuring and customizing GE Fanuc PLCs or racks of Remote I/O for your specific operating needs. For more information on Hardware Configuration, see page 37.
- LD Editor: an intelligent, cell-based, graphical editor for developing LD (Ladder Diagram) logic. You can customize the look and feel of the LD Editor. For more information on the LD Editor, see page 51.
- IL Editor: an easy-to-use, free-form text editor for creating Instruction List Logic. Configurable formatting rules and color coding make your scripts easy to read. For more information on the IL editor, see page 64.
- PLC Motion Editor: a free-form text editor that enables you to enter the program in your preferred style. The PLC Motion editor is specifically designed for support of the DSM314 motion module. For more information on the Motion editor, see page 77.
- Local Logic Editor: a text-based editor used for developing logic that executes locally on a DSM314 motion module. Local Logic runs synchronously with the PLC Motion program, but is independent of the PLC's CPU. For more information on the Local Logic editor, see page 83.
- CAM Editor: an accessory for the Logic Developer - PLC motion editor specifically designed for support of the DSM314 motion module. It provides a graphical means to create, edit, and manage electronic CAM profiles. For more information on the CAM editor, see page 86.
- C Blocks and C Programs: independent sections of executable code, written in C language, that are developed outside of Logic Developer - PLC, using GE Fanuc's C *Programmer's Toolkit for Series 90 PLCs User's Manual* (GFK-0646). Precompiled C blocks and C programs (.exe files) are imported into your project. For more information on C blocks, see page 71. For more information on C programs, see page 73.

# SYSTEM REQUIREMENTS

- Windows® NT version 4.0 with service pack 4.0 or later

  **OR**

  Windows 2000 Professional

  **OR**

  Windows XP

  **OR**

  Windows 98 SE.
- Internet Explorer version 5.5 with Service Pack 2 or later. (You must install IE5.5 SP2 before installing Machine Edition.)
- 200 MHz Pentium-based computer. (300 MHz if the operating system is Windows XP or Windows 2000.)
- 128 MB RAM.
- TCP/IP Network protocol-based computer.
- 110 MB free hard disk space (and up to 310 MB if other CIMPLICITY Machine Edition products are concurrently installed).
- 200 MB hard disk space for sample projects (optional).
- Additional hard disk space for projects and temporary files.

# INSTALLATION

### To install Logic Developer - PLC

1. Insert the CIMPLICITY Machine Edition CD into your CD-ROM drive.
   By default, the setup program will automatically start. If the setup program does not automatically start, run *Setup.exe* in the root directory of the CD.
2. Click Install to start the install process.
3. Follow the instructions as they appear on the screen.

# PRODUCT AUTHORIZATION

Before you can start developing projects with Logic Developer - PLC, you must authorize the software with a program called Product Authorization. If you don't authorize the software, you can use it for only a four-day trial period. This procedure takes only a few minutes and enables you to take advantage of any product support for which you qualify. You will need to contact us by telephone, fax, or e-mail as part of the authorization process.

### To authorize a copy of Machine Edition

1. Have your serial numbers ready.

   The serial numbers can be found on the License Key sheet that came with your product.

2. Run the Product Authorization program from the **Start** menu > **Programs** > **Product Authorization**.

   The Product Authorization dialog box appears.

3. Click **Add**.

4. Select the medium with which you are authorizing: Internet, Phone/Fax/E-mail, or Floppy Disk Transfer. Click Next.

   If you choose the Internet option, proceed to step 5.

   If you choose the Phone/ Fax/E-mail option, proceed to step 5.

   If you choose the Floppy Disk Transfer option, ensure you have an authorization disk before proceeding.

5. Fill in the fields in the dialog box.

   Fields that are identified with an asterisk (*) must be filled in.

6. If authorizing by:

   - **Internet**, click **Submit Authorization**. We will reply by e-mail with your new key code(s).
   - **Phone**, click **Phone/Fax** and call the number on the screen to receive a new key code(s).
   - **Fax**, click **Phone/Fax.** In the dialog box that appears, click **Print FAX**. Fax the Product Authorization Request to us, using our fax number on the printout. We will reply by fax with your new key code(s).
   - **E-mail**, click **Send E-mail**. In the dialog box that appears, click **Authorize** to e-mail us. We will reply by e-mail with your new key code(s).

   Product Authorization is complete after you enter the new key code and it has been accepted. Depending on the product you have purchased, you may need to run the Product Authorization program a number of times.

### To move the authorization to another computer

You can run the software on only the computer that the Product Authorization was run on. If you want to develop your projects on a different computer, you need to complete the following steps to move the authorization from one computer to another.

1. Install Logic Developer - PLC on the computer that the authorization will be moved to. Run the Product Authorization program from the **Start menu** > **Programs** > **CIMPLICITY Machine Edition** > **Product Authorization**.
   The Product Authorization dialog box appears.
2. Click **Software**.
   There is a Target Site Code on the top right-hand side of the screen. Write down this site code carefully. It must be accurate for the move to work. You will need the Target Site Code when you move the authorized software from the source computer.
3. Click **Add**.
   The Product Authorization wizard appears.
4. Click **Authorize by disk**.
   At this point, you need to go to the source computer that has the authorized software, and move the authorization to a disk.
5. From the source computer, run the Product Authorization program and click **Software**.
6. Click **Move**, and then click **OK**. Enter the Target Site Code that you wrote down from Step 3 and click **Next**. Verify that the site code is correct and click **OK**.
7. Insert a blank formatted floppy disk into the floppy drive and click **Next**.
   The authorization code will be moved to the disk and a dialog box should appear telling you it was successful.
8. Click **OK**.
9. Go back to the computer to which you are moving the authorization and insert the floppy disk.
   The screen that is asking for an authorization disk should be displayed.
10. Click **Next**.
11. Click **Finish**.
    A screen should appear telling you the move was successful.
12. Click **OK**.

    The authorization has now been moved to the new computer.

# TECHNICAL SUPPORT

Support is available to registered users at no charge for 90 days after purchase. A *Support and Free Enhancements (SaFE)* subscription can be purchased from your local GE Fanuc distributor if extended support is required.

If problems arise that can't be solved using the information in your product manual, online Help system, or the GE Fanuc Technical Advisor knowledge base, contact us by telephone, fax, or mail.

When contacting us, call from a telephone near your computer and have your CIMPLICITY Machine Edition software running. Have the following information handy to help us assist you as quickly as possible:

- The serial number from your installation CD case, and the Product name and version number from the **Help**>**About** dialog box.
- The brand and model of any hardware in your system.
- Operating system and version number.
- The steps you performed prior to the problem occurring.

## North America

**Support Hotline:** 1-800-GEFANUC (1-800-433-2682) **Fax:** (780) 420-2049
**Internet:** http://www.gefanuc.com **Email:** support@gefanuc.com
**Comments about our manuals and help:** doc@gefanuc.com
**Mailing Address:** GE Fanuc
2700 Oxford Tower, 10235 - 101 St.
Edmonton, AB, Canada, T5J 3G1

## South America

**Telephone:** +58 (261) 760 2862 **Fax:** +58 (261) 765 0909
**Internet:** http://www.gefanuc.com (visit our Portuguese web site at www.gefanuc.com.br)
**E-Mail:** luis.wilhelm@gefanuc.com
**Mailing Address:** GE Fanuc Automation Latin America
Calle 120 con Av. 17, Los Haticos -GE Turbimeca
Maracaibo, Venezuela

## Europe

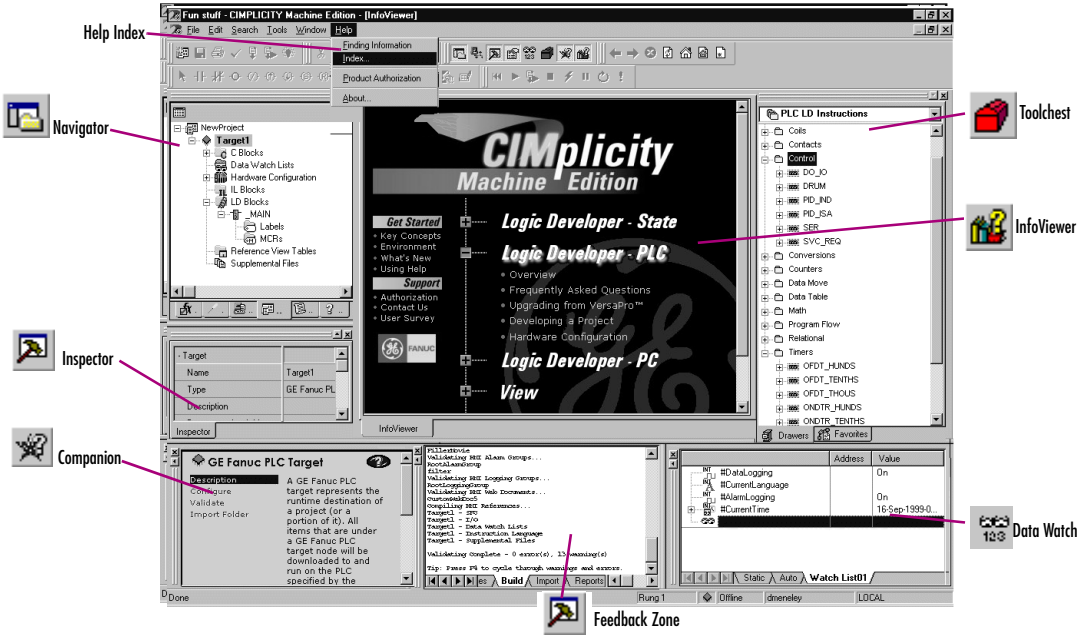
**Internet:** For up-to-date contact information, visit www.gefanuc-europe.com and click on "Offices and Services".
For up-to-date technical information visit www.gefanuc.com/support.
**E-mail:** plc_europe@gefanuceur.ge.com

# 2 CIMPLICITY Machine Edition

Machine Edition offers you a complete solution for the development of automation applications in one package. Machine Edition features an integrated development environment and tools which allow you to spend more time building applications and less time learning the software. All Machine Edition products are fully integrated into the environment and interact with each other.

- They share a common project database. No more wasted time synchronizing data points between applications.
- They share the same set of tools providing a consistent interface throughout the development process.
- They feature full drag-and-drop capabilities between tools and editors.
- They feature a true scalable solution. You have the choice of what type of machine your projects run on.
- They automatically display Help in the Companion for what you click on.

The following illustrates a Machine Edition project:



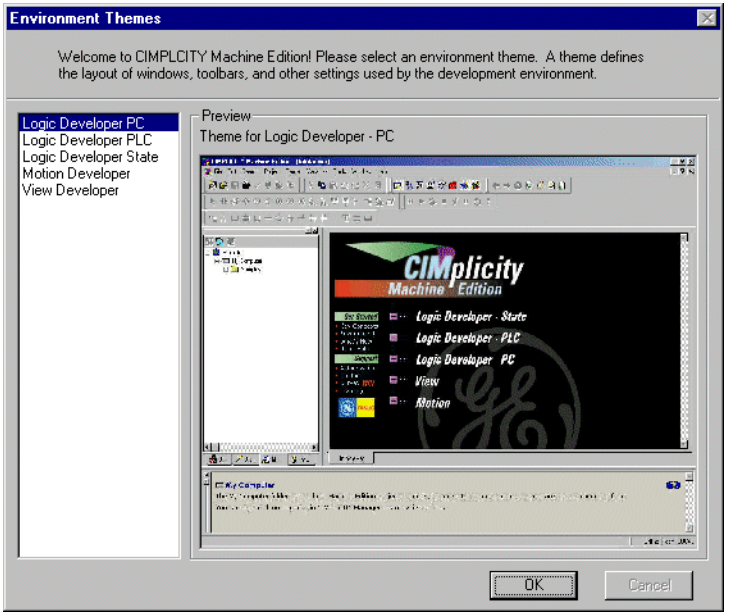
# QUICK START

Machine Edition makes it easy to get started developing a project.

## To start Machine Edition

1. Click Start **Start**, point to **Programs**, point to **CIMPLICITY Machine Edition** and then choose **CIMPLICITY Machine Edition**.

   After Machine Edition initializes, the Environment Themes dialog box appears.

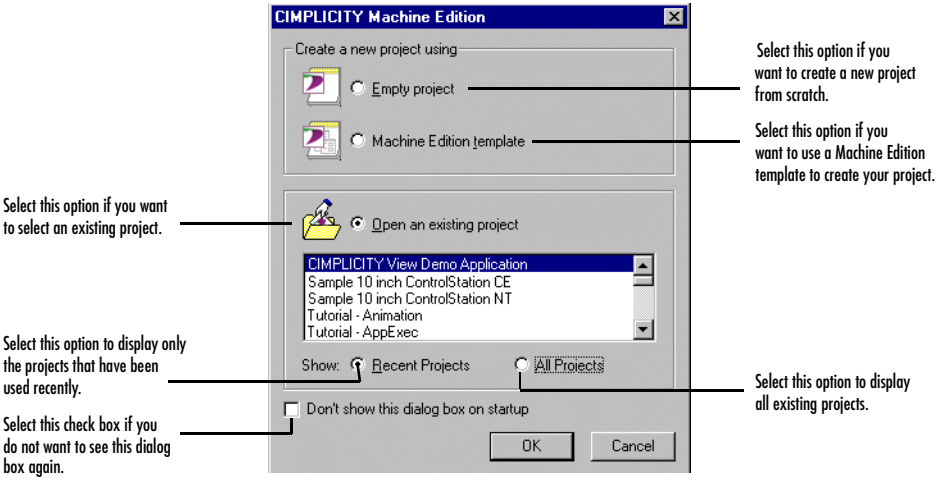



   **Note:** The Environment Themes dialog box appears automatically the first time you start Machine Edition. To change the theme later, choose **Apply Theme** from the Windows menu.

2. Choose the environment theme you want to work in.
3. Click **OK**.

   When you open a project, the appearance of your Machine Edition screen will match the preview in the Environment Themes dialog box.

The CIMPLICITY Machine Edition dialog box appears.




4. Select the appropriate option to open a project. The **Open an existing project** option is selected by default.

   **Notes**:

   - If you select either the **Empty project** option or the **Machine Edition template** option, the New Project dialog box appears and you can continue creating a new project (see page 12).
   - If you select the Open an existing project option, you can also select either the Recent Projects option or the All Projects option. The Recent Project option is selected by default.

5. If you selected the **Open an existing project** option, select from the list the project that you want to open.

   The existing projects include samples and tutorials that you can open and use to familiarize yourself with Machine Edition.

6. If you want, select the Don't show this dialog box on startup option.
7. Click OK.

   Your project opens in the Machine Edition environment that you specified in the Environment Themes dialog box.

# PROJECTS

You can create and edit Machine Edition projects using products such as View, Logic Developer - PC, Motion Developer, Logic Developer - State, and Logic Developer - PLC. These products share Machine Edition tools to provide a high level of integration between the various parts of your project.

Folders created with Logicmaster, CimplicityControl, or VersaPro can be imported.

Using Logic Developer - PLC, you can build a number of different projects to suit your specific requirements.

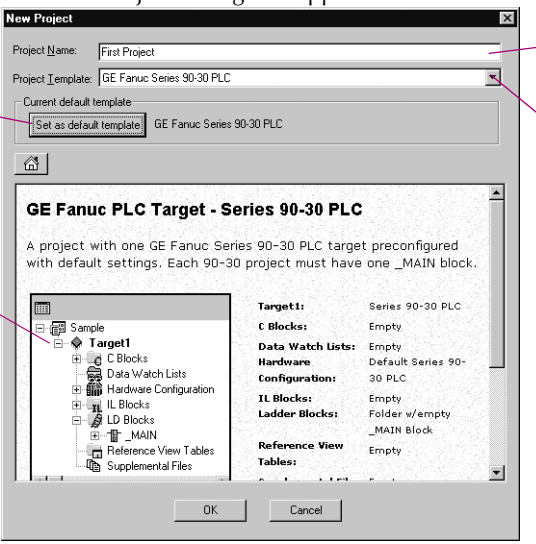
## To create a new project using a template

Before creating a project, there are some things you should know:

- The primary components your project will contain.
- The PLC you project will run on.

1. From the File menu, choose New Project, or click on the File toolbar.

The New Project dialog box appears.




You can set the selected project template as the default template by clicking the default template button.

A sample description of the project appears on the screen.

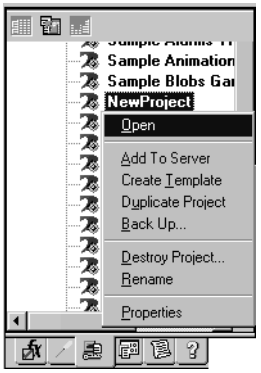
Enter the name of your project in the Project Name box.

Choose your project template from the Project Template list.

2. Choose a Project Template that suits your needs.
3. Enter a descriptive Project Name.
4. Click **OK**.

Your project opens in the Machine Edition environment.

### To open an existing project for editing




**Navigator: Manager tab**

1. Open the Navigator and choose the Manager tab.

   A list of projects is displayed.

2. Right-click a project and choose **Open**.

   Your project loads and is ready for editing.

> **Want to know more?** In the Help Index, look up "Projects".

### To import a folder

1. Open the Navigator and choose the Project tab.
2. Select the target that you want to import the folder into.
3. Right-click the target, point to **Import**, and choose the folder type.
4. In the dialog box that appears, navigate to and select the folder to import, and click **OK**.

> **Want to know more?** In the Help Index, look up "import a CimplicityControl folder", "import a Logicmaster folder", or "import a VersaPro folder".

## GLOBAL SEARCH

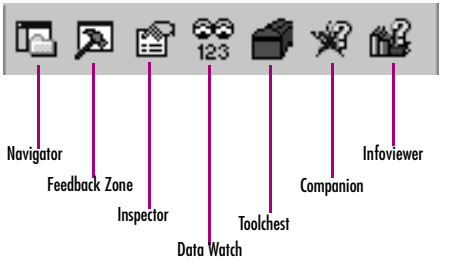
Logic Developer - PLC provides the capability to search for some text in an entire target or portions thereof, which you can specify. Various options are available to further narrow a search.

> **Want to know more?** In the Help Index, look up "Global" and select the "Conducting a Global Search" help topic.

# TOOLS

Project development is supported by the Machine Edition tools. Each tool is opened and closed by means of a button on the Tools toolbar.

A description of each tool's function is outlined below.




### Navigator

The Navigator is a docking tool window containing a set of tabs. Each tab displays information about your development system in a hierarchical tree structure similar to Windows Explorer. The available tabs depend on which Machine Edition products you have installed and what kind of work you are developing or managing. The Project tab shows the overall organization of your application.

### Feedback Zone

The Feedback Zone window is a docking window used to display several types of output information generated by Machine Edition –enabled components. This interactive window uses category tabs to organize the output generated from the Machine Edition products you have installed.

### Inspector

The Inspector lists the properties and current settings for a selected object or element. You can edit these properties directly in the Inspector. When you select several objects, the Inspector window lists the properties commom to all of them. The Inspector window provides a simple method of viewing and setting properties for all objects.

### Data Watch

The Data Watch tool is a runtime debugging tool that enables you to monitor and edit the values of variables. This tool is useful while working online to a target. With the Data Watch tool, you can monitor individual variables or user-defined lists of variables. Data Watch lists can be imported, exported, or saved with a project.

### Toolchest

The Toolchest is a powerful storehouse of objects you can add to your project. You can drag most items directly from the toolchest into Machine Edition editors. You can choose from predefined objects or create your own reusable fxClasses. The Toolchest adds true object oriented capability to Machine Edition.

### Companion

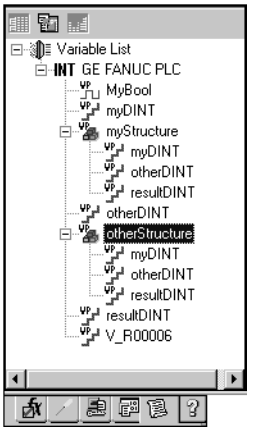
The Companion provides useful tips and information while you work. While the Companion is open, it tracks your moves and displays help on whatever item is currently selected in the Machine Edition environment. It is context-sensitive and displays a description of whatever you click on the Machine Edition screen.

### InfoViewer

Machine Edition online help system is an integrated display engine and Web browser. If you are familiar with Internet Explorer or Netscape Navigator, then you are already familiar with the basic InfoViewer interface. Like the Companion, the InfoViewer is context-sensitive. Simply press F1 to get help on any item you select in the Machine Edition environment. A table of contents is found in the InfoView tab of the Navigator.

**Want to know more?** In the Help Index, look up “Tools: an Overview”.

# VARIABLES




**Navigator: Variables tab**

A variable (sometimes called a tag) is a named storage space for data values.

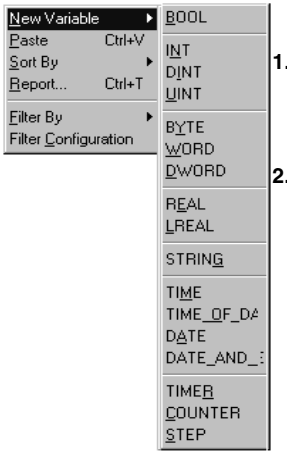
All variables in a project are presented in the Variables tab of the Navigator. A variable represents a memory location in the target PLC. Each variable is mapped to a reference address (for example, %R00001). The reference address and other properties of a variable, such as data type, are configured in the Inspector.

Arrays and compound structure variables are supported by Machine Edition. Variable definitions can be imported from and exported to a variety of file types. You can edit your variables in a spreadsheet and then import them. The following picture shows the Inspector displaying a typical set of variable properties.

| Variable [MyTargetName] | |
|---|---|
| Name | myBOOL |
| Description | |
| Publish | True |
| Array Dimension 1 | 0 |
| Data Source | GE FANUC PLC |
| Ref Address | %I00001 |
| Data Type | BOOL |
| Current Value | Off |
| Initial Value | Off |
| Default Display Format | On / Off |
| Retentive | True |
| Force State | Not Forced |

General

## To create a variable




**New Variable menu**

1. In the Variables tab of the Navigator, right-click **Variable List**, point to **New Variable** and then choose the data type of the variable.

   A new variable with a default name appears in the list.

2. Type a name for the variable and press **ENTER**.

   Variable names can range from 1 to 32 characters, begin with a letter, contain upper or lower case letters, use numbers between zero and nine, and use the underscore character ("_").

## To map a variable to PLC memory

In order to successfully download a project to the PLC, variables must be properly mapped to the PLC's memory. There are two ways to map a variable.

**First method:**

1. In the Variables tab of the Navigator, right-click a variable and choose Properties.

   The Inspector opens showing the properties of the variable.

2. In the Ref Address field, enter an address. You can:
   - Spell out the exact address, for example, %R00123 or 123R. In either case this maps the variable to %R00123.
   - Enter only the memory area, for example %R. This maps the variable to the next highest available address. For example, if %R00122 was the last address used, entering %R maps the variable to %R00123.

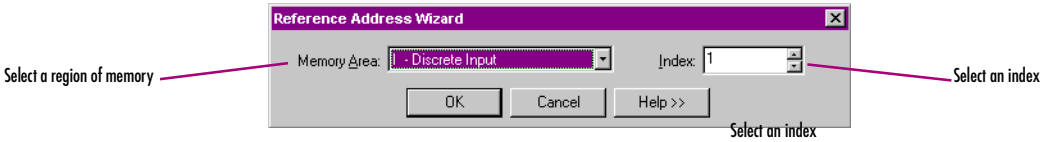
**Second method:**

1. In the Variables tab of the Navigator, right-click a variable and choose Properties.

   The Inspector opens showing the properties of the variable.

2. Click the ••• button in the Ref Address field.


The Reference Address Wizard appears.




3. From the Memory Area list, choose a region of PLC memory.
4. In the Index box, enter an index from the beginning of the region.
5. Click **OK**.

   Your variable is mapped to the Reference Address.

**Want to know more?** In the Help Index, look up "Variables" and "Importing variables from a file".

# MACHINE EDITION HELP

CIMPLICITY Machine Edition includes a comprehensive online help system that enables you to access specific help topics while working with Machine Edition.

Use the InfoViewer or Companion to access help.

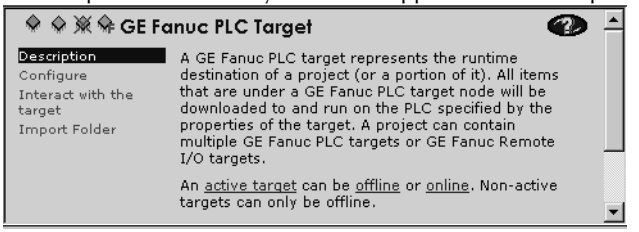
## Companion Help

The Companion is a Machine Edition help system that provides useful tips and information while you work. While the Companion is open, it displays help on whatever item is currently selected in the Machine Edition environment, tracking your movements while you work.

### To use Companion help

1. Ensure the Companion window is open. To open it, press Shift+F11.
2. Click on any item on the screen.

   A description of the item you clicked appears in the Companion.



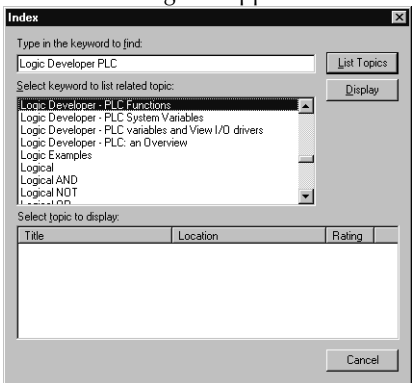

## InfoViewer Help

The InfoViewer, another Machine Edition help system, provides detailed information. The InfoViewer has its own toolbar for navigating the help system, a Table of Contents (in the InfoView tab of the Navigator), and a searchable index. As with the Companion, InfoViewer help is context-sensitive. Click an item on the screen and press F1 to display the appropriate topic in the InfoViewer.

### To use the Help index

1. From the Help menu, choose Index.

   The Index dialog box appears.



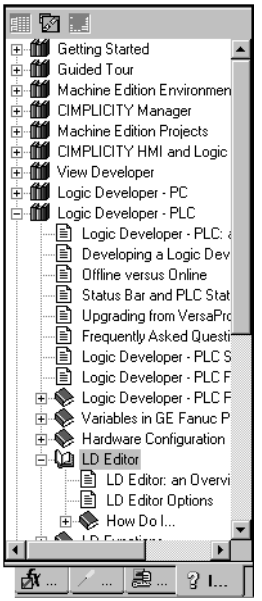

2. Enter or choose a keyword from the list.
3. Click **List Topics**.

   A list of topics appears. The topics are sorted by their rating or likelihood of containing the correct information.

4. Select a topic to display.
5. Click **Display**.

   The selected Help topic is displayed in the InfoViewer.

## To look up topics in the Help Table of Contents



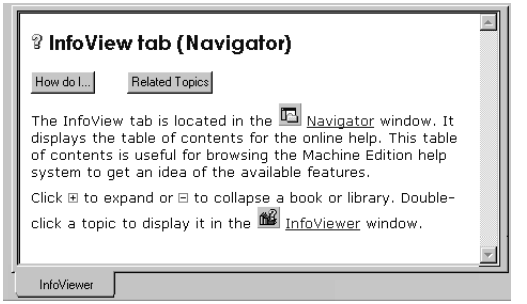

**Navigator: InfoView Tab**

1. Open the InfoView tab of the Navigator.

   A table of contents for the entire help system appears.

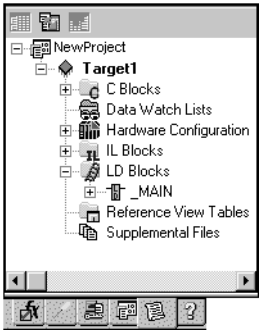
2. Expand Libraries and Books to locate a topic of interest
3. Double-click a topic.

   The topic is displayed in the InfoViewer.



# 3

# GE Fanuc PLC Targets




**Navigator: New 90-30 Project**

A target represents a runtime destination of the programs you develop with Logic Developer - PLC. Each target contains all of the components that will download to it. Logic Developer - PLC supports the following GE Fanuc PLC targets:

- Series 90™-30 PLC
- Series 90™ Micro PLC
- VersaMax® PLC
- VersaMax® Nano/Micro PLC
- Series 90™-70 PLC

and the following remote I/O interface targets:

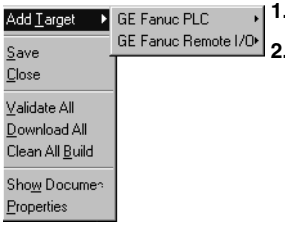
- GE Fanuc VersaMax Ethernet
- GE Fanuc VersaMax Profibus
- GE Fanuc VersaMax Genius
- GE Fanuc Series 90™-70 Genius Remote I/O Scanner

# CREATING AND CONFIGURING TARGETS

## Creating Targets

Normally a target is present in a project when you use a template to create the project (see page 12). A project can contain any number of targets. One target is required for each PLC or remote I/O adapter your project accesses.

### To add a target to a project




**Project right-click menu**

1. In the Project tab of the Navigator, right-click the Project node and point to **Add Target**.
2. Point to **GE Fanuc PLC** or **GE Fanuc Remote I/O** and then choose the PLC family.

## Configuring Targets

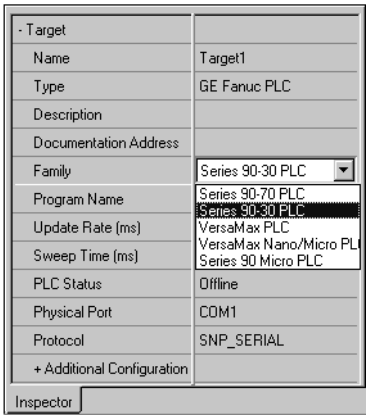
The properties of a target specify the PLC family, the communication connections between your computer and the PLC, and various other settings. All properties are edited in the Inspector. The following table describes these target properties:

| | |
|---|---|
| **Name** | Edit the name for your target in this field. |
| **Type** | The type of target is set by default to GE Fanuc PLC. Note: Logic Developer - PLC is only operational with GE Fanuc PLCs. |
| **Description** | Enter a brief description of your project in this field. |
| **Documentation Address** | Enter the URL where your project documentation is stored. |
| **Family** | Change the PLC type in this field. **Caution: Changes are irreversible.** |
| **Program Name** | The name of the target known to the PLC. |
| **Update Rate (ms)** | Set the rate at which the screen is updated while online. |
| **Sweep Time (ms)** | The sweep time of the PLC when online is also displayed on the status bar. |
| **PLC Status** | The online/offline, run/stop status of the PLC. |
| **Dual HWC** | (Series 90™-70 only) Indicates whether HWC redundancy is set up. Change the field to False to delete the secondary HWC. |
| **Selected HWC** | (Displayed only if Dual HWC is set to True) Change the selected HWC in this field. |
| **Physical Port** | Choose the type of connection to the PLC (Ethernet COM or modem). |
| **IP Address** | Set the IP address of the PLC (Ethernet protocol only). |
| **Additional Configuration** | Group of properties used for the detailed configuration of your communication connection. |

### To change the PLC family

1. In the Project tab of the Navigator, right-click an existing target and choose Properties.

   The properties of the target appear in the Inspector.
2. In the Inspector, click Family.
3. From the list, choose the PLC family type.




**Warning:** Changing the PLC Family can result in logic in your project becoming invalid.

**Note:** PLC hardware must be configured before it will be operational. For detailed information on hardware configuration, see page 37.

## Configuring Communication

In order for Logic Developer - PLC to communicate with a target PLC, a connection must be properly configured. The properties of a target are adjustable to accommodate your connection(s).

### To configure an Ethernet or serial connection

1. In the Project tab of the Navigator, right-click a target and choose **Properties**.

   The Inspector opens showing the target's properties.
2. In the Inspector, set the Physical Port property to Ethernet or a COM port (or a modem if one is installed).
3. If the Physical Port is Ethernet, enter the IP Address of the target PLC.
4. Double-click Additional Configuration to access the detailed settings for your connection.

   **Note:** Serial communication with a PLC is always available if it is the only device connected and an SNP_ID is not specified. An IP address must be set in the PLC before an Ethernet link can be established.

**Want to know more?** In the Help Index, look up "Connections".

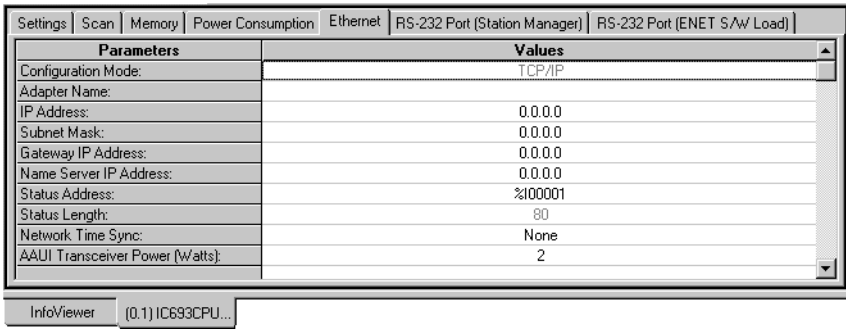
### To set an IP address for a Series 90™-30 CPU364 or CPU374

**Note:** For CPU364 and higher-end Series 90™-30 CPUs, the Ethernet connection is built into the CPU. For lower-end Series 90™-30 CPUs, the Ethernet connects through an optional expansion slot on the rack.

1. Obtain an IP address, perhaps from your network administrator.
2. In the Project tab of the Navigator, expand the Hardware Configuration node to reveal the rack and slot containing a CPU364 or CPU374.
3. Right-click the slot containing the CPU364 or CPU374 and choose **Configure**.

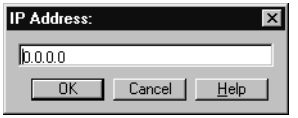
   The Parameter editor appears.
4. In the Parameter editor, click the Ethernet tab.

The Parameter editor displays the contents of the Ethernet tab.




5. Double- click the IP address field.

   The IP Address dialog box appears.




6. Enter the IP address in the box and click **Ok**.

### To set an IP address for lower-end Series 90™-30 CPUs

1. In the Project tab of the Navigator, expand the Hardware Configuration node.
2. Right-click the slot containing an Ethernet Communication Module and choose Configure.

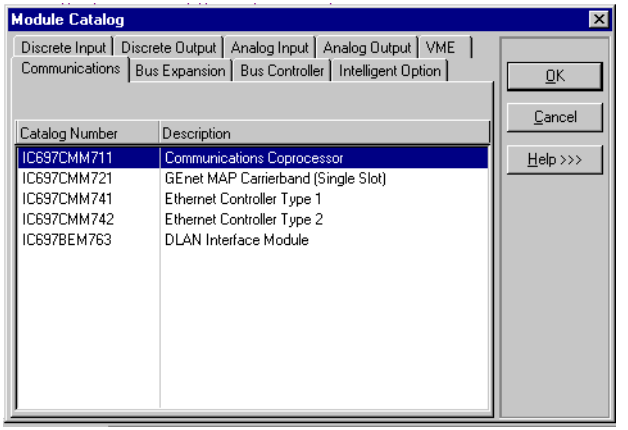
   The Parameter editor opens showing the configuration settings for the module.
3. In the Settings tab, double-click the IP Address field.

   The IP Address dialog box appears.
4. Enter the IP address and click **OK**.

### To set an IP address for Series 90™-70 PLCs

1. In the Project tab of the Navigator, expand the Hardware Configuration node.
2. Expand the Main Rack and right-click the slot where you want to add a communications module.

The Module Catalog appears.




3. Choose the Communications tab and select the communications module you want to add.

   Select either IC697CMM741 or IC697CMM742.

4. Click **OK**.

   The Parameter editor opens showing the configuration settings for the module.

5. In the Settings tab, double-click the **IP Address** field.

   The IP Address dialog box appears.

6. Enter the IP Address and click **OK**.

### To download an IP address to a PLC

1. Configure a CPU or Ethernet COM Module with an I/P address using Hardware Configuration.
2. Connect to the PLC via serial connection.

   **Note**: Do not specify an SNP_ID.

3. Download the Hardware Configuration to the PLC.

   The IP address is assigned to the PLC.

4. Connect to the PLC via Ethernet connection.

# INTERACTING WITH THE PLC

Communicating with the PLC is necessary to perform such operations as storing and loading programs or monitoring data values and PLC status information. You can connect to a PLC from Logic Developer - PLC over a serial, Ethernet or modem connection, depending on the capabilities of the PLC.

All interactions with a target are available from the target's right-click menu.

## Validating a Target

Validating your project detects any syntax errors that may be present. Error messages are generated for each error and displayed in the Feedback Zone. A project containing errors cannot be downloaded. For that reason, whenever you initiate a download, the logic is automatically validated.

### To validate a target

- In the Project tab of the Navigator, right-click a target and choose **Validate**.

  Logic Developer - PLC checks all items under the target for syntax errors. Any errors detected are noted with messages in the Build tab of the Feedback Zone.

  **Tip:** Double-click an error message to locate the noted error in your project. The appropriate editor or tool opens automatically with the item in question selected. Tips showing you how to proceed are displayed in the Companion.

## Upload/Download

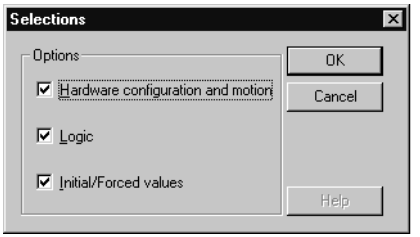
The download process creates (or builds) and validates all runtime files necessary for a target to perform its role in a completed project. The compiled project is then transferred to the target hardware via the communication connection previously configured.

The upload process acquires a project from an active PLC target and transfers it to Logic Developer - PLC for editing.

## To download to a target PLC

1. In the Project tab of the Navigator, right-click the target to which you want to download files and choose **Download to PLC**.

   The Selections dialog box appears.




   **Note:** If the PLC is running, you can only download logic that is not equal to the PLC's current logic and the Selections dialog box does not appear.

2. Choose the items you want to download and click **OK**.

   **Tip:** To download files for all targets in a project, right-click the project node and choose **Download All.** If you want to download files to the active target, choose **Download Active Target.**

   **Note:** Only one project can be downloaded to a target at a time. If you download to a target machine that already has a project on it, the existing project is overwritten.

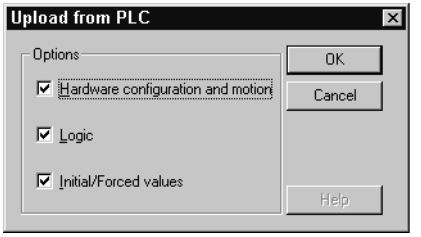
   For each target that you download to, Machine Edition saves the project, performs a validation, builds the runtime files and attempts to establish a connection to the target. Any errors that occur are displayed in the Build tab of the Feedback Zone. If connection to a target is successful, Machine Edition sends all the necessary runtime files to the PLC.

**Want to know more?** In the Help Index, look up "Downloading".

### To upload files from a PLC

1. In the Project tab of the Navigator, right-click the target PLC from which you want to upload information and choose **Upload from PLC**.

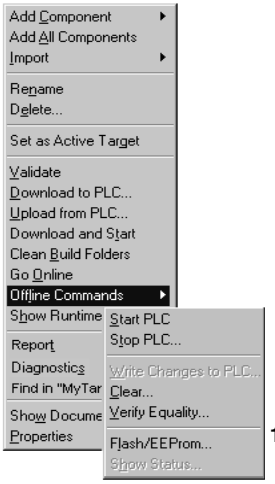
   The Upload from PLC dialog box appears.




   **Note:** If you are uploading from Logic Developer - State, only the Hardware Configuration can be uploaded.

2. Choose the item(s) you want to upload and click **OK**.

   Logic Developer - PLC connects to the PLC and uploads the selected item(s) to Logic Developer - PLC. If you already had a version of the PLC's project open, the uploaded project merges with the existing project. This ensures that all variable names are retained throughout the uploading process. If you upload to any empty target, all variables are assigned default names. For example, %R00001 is named R00001.




**Target: Right-click menu**

**Offline commands**

## Run/Stop

You can set a target PLC to Run or Stop mode. In Run mode, you can choose to have outputs enabled or disabled. In Stop mode, you can choose to enable or disable the outputs.

- In the Project tab of the Navigator, right-click a target, point to **Online Commands** or to **Offline Commands**, and then choose **Start PLC**.

  The target PLC begins executing its program.

1. In the Project tab of the Navigator, right-click a target, point to **Online Commands** or to **Offline Commands**, and choose **Stop PLC**.

   The Stop PLC dialog box appears, prompting you to enable or disable the PLC's outputs.

2. Select an option.

   The target PLC stops executing its program.
3. Click OK.

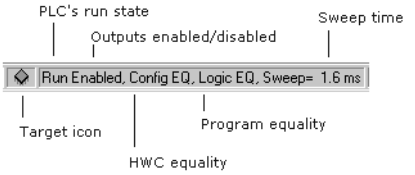
## Online/Offline

When offline from a target PLC, the number and type of interactions available are limited. You can upload, download, or start, stop, or clear the PLC. You can also verify the equality of the PLC's program to your open project. However, while online to a target PLC, you can interact with the target in real-time and monitor the operation of the PLC using the DataWatch, Reference View and Fault tables. Also, while online, the LD editor displays a graphical representation of LD logic as it executes, if the PLC's project is equal to the current project. You can edit LD logic while online; you are prompted to download your word-for-word changes.

### To go online to a PLC

- In the Project tab of the Navigator, right-click a target PLC and choose **Go Online**.

  Logic Developer - PLC connects your project to the PLC. The online status is indicated by the target icon in the Project tab and on the status bar.




**Note:** When online to a PLC, the target icon in the Project tab of the Navigator appears as (equal), (not equal), or (faulted).

### To go offline from a PLC

- In the Project tab of the Navigator, right-click a target PLC and choose **Go Offline**.

  Logic Developer - PLC goes offline.

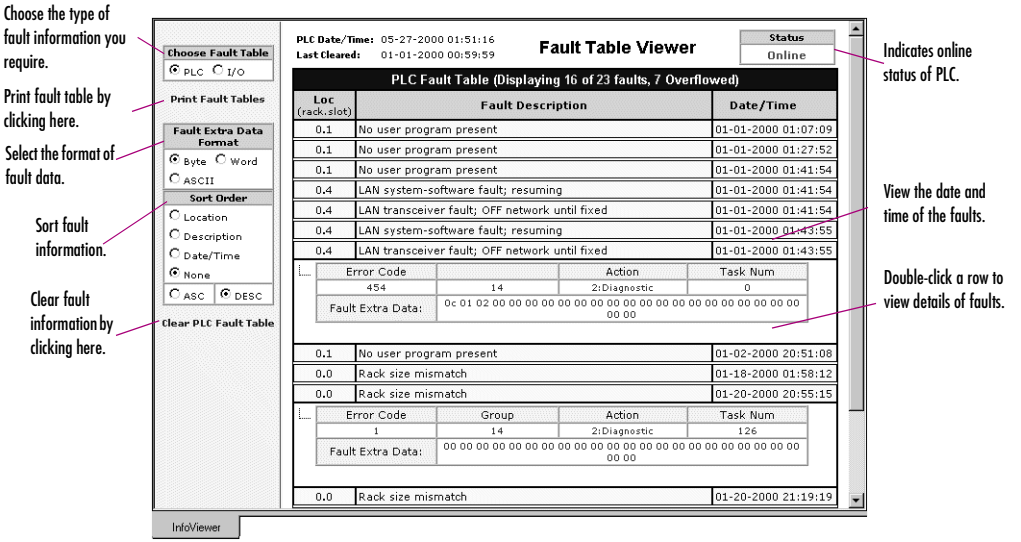
## Fault Tables

The PLC and I/O Fault Tables display fault information logged by the CPU or modules in the PLC. This information is used to determine if there are problems with the PLC hardware or software running in the PLC's CPU. To view the PLC and I/O fault tables, your computer must be online to the PLC.

### To view the fault table reports

- In the Project tab of the Navigator, right-click the target you want to report on and choose **Diagnostics**.

The fault table appears in the InfoViewer window.




## Reference View Tables

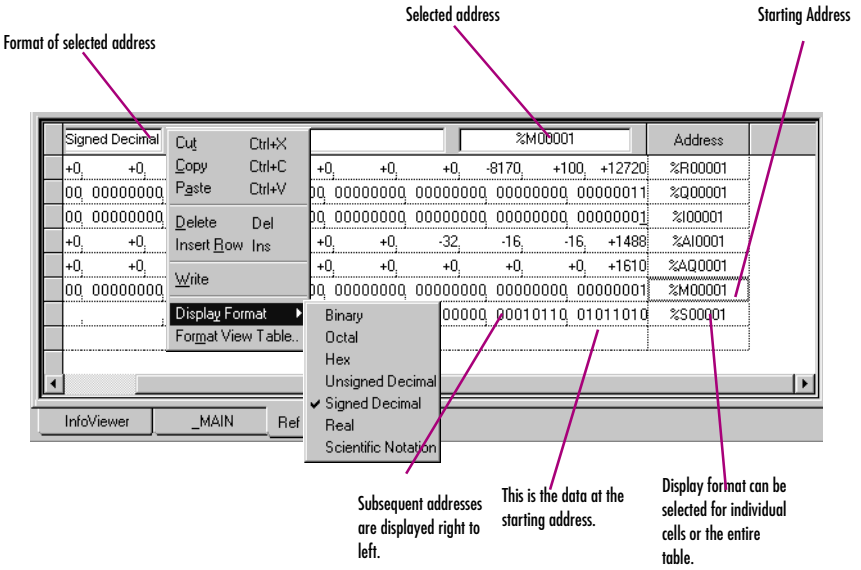
Reference View Tables (RVTs) contain a user-defined list of reference addresses that can be monitored and changed in real-time. A target can have zero or more user-defined RVTs, included in the Reference View Tables folder in the Project tab of the Navigator.

The number of entries contained in an RVT does not affect performance. Performance is affected only by the number of entries that are displayed and have to be updated in the view.

An RVT displays data when the parent GE Fanuc PLC target is active and online.

You can configure the appearance of your RVTs in the Options tab of the Navigator.

Data values at sequential addresses are displayed, by default, from right to left, starting at the reference address specified. The amount of data displayed in the eight columns depends on the data display format.

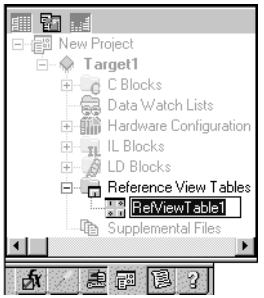



## To create a Reference View Table

- In the Project tab of the Navigator, right-click on the Reference View Tables folder and choose **New**.

  A new Reference View Table with a default name is added to the folder.

### Working with a Reference View Table




**Navigator: Project Tab**

**Reference View Table node**

1. In the Project tab of the Navigator, expand the Reference View Tables folder and double-click the table you want to view.

   The Reference View Table appears.
2. Add reference addresses to the table as required.
3. Format the table entries as desired.

**Want to know more?** In the Help Index, look up "RVT".

## Reports

Reports provide summaries and tables of information about your project. Most reports are generated and displayed in the InfoViewer window. The Reports tab of the Feedback Zone contains a list of all reports generated since the last Machine Edition project was opened. The following list shows the types of reports and logic printouts available in Logic Developer - PLC:

| | |
|---|---|
| Address Use report | C block report* |
| Hardware Configuration report | IL block report* |
| Variables report(s) | Local Logic block report * |
| CAM profile report | Motion block report* |

**Notes**

- An asterisk (*) indicates a logic printout.
- Reports are not generated for LD blocks. Rather, these blocks are printed directly from the LD editor.

### To create reports

- In the Project tab of the Navigator, right-click a node and choose Report to generate a report on that node.

  A report is automatically generated and displayed in the InfoViewer.

### To redisplay a previously generated report

1. In the Feedback Zone, double-click the Reports tab.

   A list of previously generated reports appears in the Feedback Zone.

2. Choose, from the list, the report you want to view.

   The report displays in the Infoviewer.

   Many items in a report contain hyperlinks. Click a hyperlinked item to jump to that item in the project. For example, if a variable's name appears hyperlinked in a report, clicking it selects that variable in the Variables tab of the Navigator.

   Large reports are often separated into several pages. To view a different page of the report, scroll to the bottom of the report in the InfoViewer and click the number of the page you want to view.

### To print a report appearing in the InfoViewer

1. Do one of the following:

   Generate the report you want to print.

   or

   Redisplay a previously generated report.

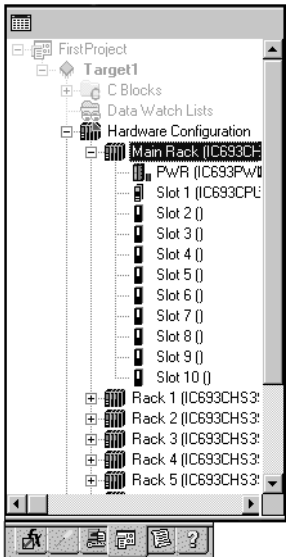
2. When the report is displayed, right-click the InfoViewer window and select **Print**.

---

**Want to know more?** In the Help Index, look up "Reports".

# 4

# HWC Hardware Configuration




**Navigator: Project Tab**

**HWC node**

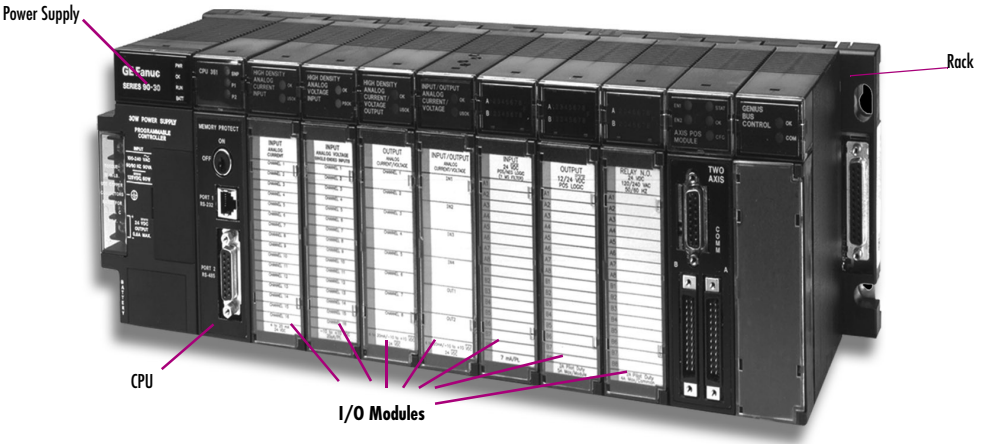
Logic Developer - PLC supports five GE Fanuc PLC families and various remote I/O interfaces (see page 46) with a variety of CPUs, racks and modules for each. In order to operate, PLC hardware must be configured with Logic Developer - PLC or some other GE Fanuc tool. The HWC component of Logic Developer - PLC provides a way to completely configure your target equipment. This chapter details specifics on configuring PLC hardware for your operational needs.

The first step in configuring PLC hardware is to select the PLC you want to configure (see page 22). When creating a new project, either you use a project template containing default hardware configuration, or you create an empty project and configure it manually.

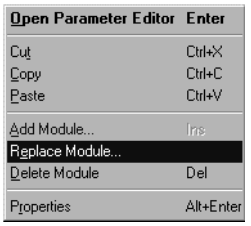
## Configuring Series 90™-30 PLC Hardware

One of the most popular GE Fanuc PLCs in use today is the Series 90™-30, the configuration of which is detailed in the following procedures. Procedures for the other PLC types supported by Logic Developer - PLC are almost identical.

By default, each Series 90™-30 target is configured with eight racks: a main rack and seven secondary racks. Each rack has either five or ten slots; the first slot in the main rack is always reserved for the CPU. The following diagram shows a typical 10-slot Series 90™-30 PLC.



## To choose a CPU




**HWC right-click menu**

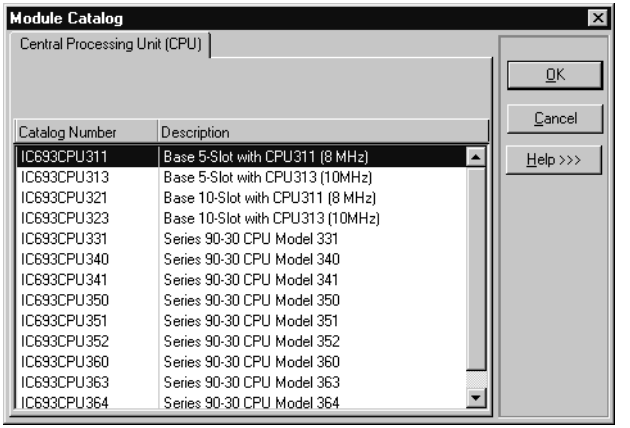
The default CPU specified in all project templates for a Series 90-30 is the CPU364. To change the CPU:

1. In the Project tab of the Navigator, expand the Hardware Configuration folder.

   All racks are revealed.

2. Expand the Main rack.
3. Right-click Slot 1 and choose **Replace Module**.

   The Module Catalog dialog box appears.




4. From the list, select a CPU.
5. Click **OK**.

   You are prompted to confirm the replacement.

6. Click **Yes**.

   A dialog box appears asking you if you want to retain the settings from the existing CPU.

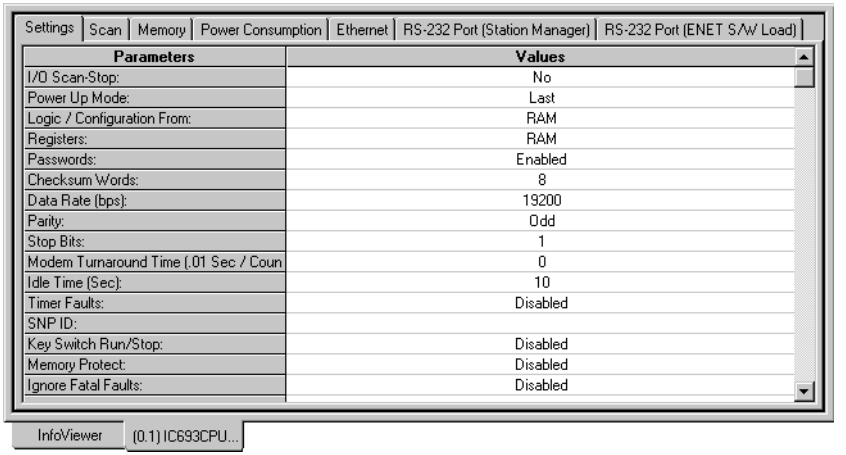
7. Click **Yes** or **No**.

   The target is configured with the selected CPU.

## To configure a CPU

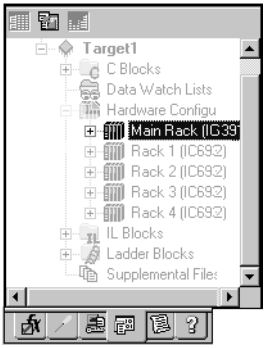
1. In the Project tab of the Navigator, right-click a slot containing a CPU and choose **Configure**.

   The Parameter editor appears showing all configurable settings for the CPU.




2. Modify settings as required. For further information, see your PLC manual.

## To select a rack type

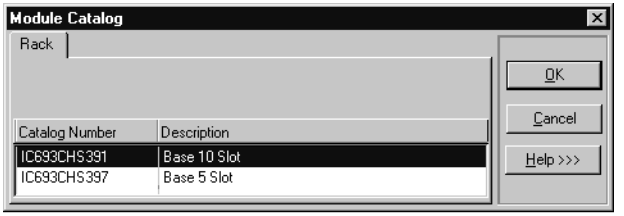



**Navigator: Project tab**

**HWC configuration**

1. In the Project tab of the Navigator, right-click a rack node and choose Replace Rack.

   The Module Catalog dialog box appears listing available rack types.



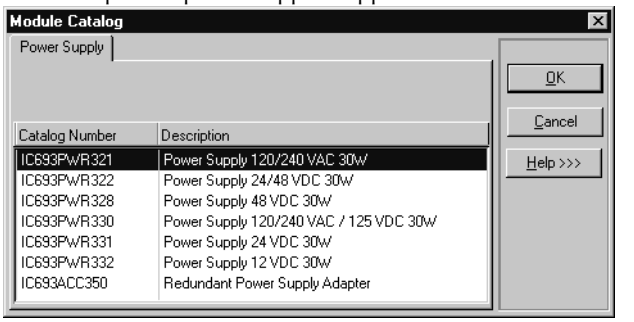

2. Select a rack and click **OK**.

### To choose a power supply

The default power supply for the Series 90™-30 is the PWR321. To change the power supply:

1. In the Project tab of the Navigator, right-click the PWR slot and choose Replace Module.

   A list of optional power supplies appears.




2. Select the power supply you have installed in your rack and click OK.

## DSM314 Motion Module

The Motion Mate DSM314 (see page 77) is a high performance, easy-to-use multi-axis motion control module. Compatible with 90-30 PLC logic solving and communications functions, the DSM314 supports the following servo types:

- Digital - GE Fanuc digital servo amplifiers and motors.
- Analog - GE Fanuc SL Series analog servos and third-party servos are supported.

The DSM314 Motion module has four axes that can be individually configured in Standard or Follower mode. It can control up to four analog mode servo axes or up to two digital mode servo axes. When axis one and two are controlling digital mode servos, axis three can control an analog servo.

In order to perform motion programming in Logic Developer - PLC, you must have a DSM314 motion module properly configured (see page 77).

**Note:** The Motion Mate DSM314 is compatible with only Series 90™-30 PLCs.

### To add a DSM-314 motion module

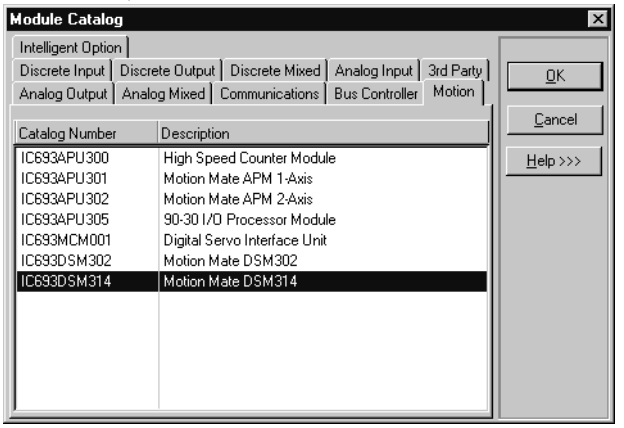
1. With a rack of the Hardware Configuration node expanded, right-click the empty slot you want to add a DSM314 motion module to and choose **Add Module**.

   The Module Catalog dialog box appears.

2. On the Module Catalog dialog box, click the Motion tab.

   A selection of motion modules appears in the Module Catalog dialog box.

3. From the list, select Motion Mate DSM314.



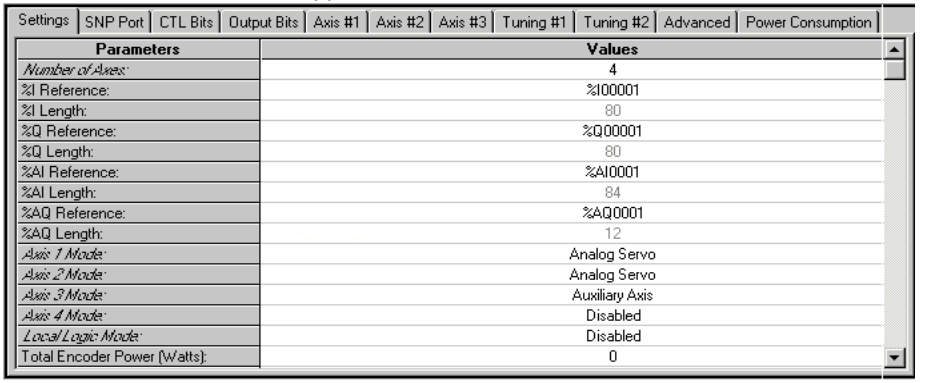

4. Click **OK**.

   A DSM314 is added to the hardware configuration of your project.

## To configure a DSM314

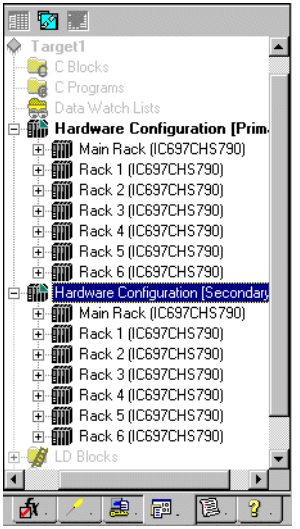
1. In the Project tab of the Navigator, double-click the slot containing a Motion Mate DSM314.

   The Parameter editor appears.




2. Configure the DSM314 using the Parameter editor.

   **Note:** For details on configuring the DSM314, see *GFK-1742, Motion Mate DSM314 for Series 90-30 PLCs Users Manual* .




**Navigation: Project Tab**
**HWC Node**
**Redundant HWC**

## Series 90™-70 PLC Redundancy

In redundant systems, two PLCs are set up and configured to share the responsibility of a single PLC. If one unit fails or is taken offline the other unit assumes responsibility without interrupting operation of the entire system.

Redundancy is used with Series 90™-70 PLCs. A target is associated with two physical PLCs, a Primary and a Secondary. Both PLCs share the same logic, but each has its own Hardware Configuration (HWC): Primary or Secondary. The selected HWC is the HWC that you can go online with, download to, upload from, and so on.

There are three types of redundant systems:

- Basic CPU Redundancy
- Genius Redundancy
- CPU Redundancy Over Genius

### Basic CPU Redundancy

Three types of basic CPU redundancy are supported for Series 90™-70 rack systems. These redundant systems can be used in combination with Genius redundancy schemes.

- Single Bus with Preferred Master - uses a single Genius bus with one or more bus controllers in each PLC. The primary unit is always chosen as the active unit when the units initially synchronize.
- Single Bus with Floating Master - uses a single Genius bus with one or more bus controllers in each PLC. No switchover occurs on initial synchronization to make the primary unit the active unit.
- Dual Bus with Floating Master - uses dual busses with one or more bus controllers in each PLC. No switchover occurs on initial synchronization. Bus Switching Modules (BSMs) are required in accordance with configuration of a dual bus network. This option provides redundancy of both the PLC and the I/O bus.

## Genius Redundancy

A redundant Genius system contains duplicate components that are controlled in a way that keeps the Genius system operating properly even if one of the duplicate components fails or is taken out of service. Redundant Genius systems can be used in combination with Series 90™-70 redundant CPU systems.

Five types of redundant Genius systems can be configured.

- Genius Dual Bus Redundancy (Paired GBC Internal)
- Genius Dual Bus Redundancy (Paired GBC External)
- Genius Dual GBC Redundancy (Paired GBC Internal)
- Genius Dual GBC Redundancy (Paired GBC External)
- Genius Dual Bus & Dual GBC Redundancy

## CPU Redundancy Over Genius

A CPU Redundancy Over Genius System contains duplicate components that are controlled in a way that keeps the system operating properly if one of the duplicate components fails or is taken out of service.

Five types of CPU Redundancy Over Genius system can be built upon the various types of redundant Genius systems combined with the basic CPU Redundancy schemes.

- CPU Redundancy (GHS) Using Genius Dual GBC Redundancy (Paired GBC External) - Single Bus with Preferred Master
- CPU Redundancy (GDB) using Genius Dual GBC Redundancy (Paired GBC External) - Single Bus with Floating Master
- CPU Redundancy (GDB) using Genius Dual Bus Redundancy (Paired GBC External) - Single Bus with Floating Master
- CPU Redundancy (GDB) using Genius Dual Bus and Dual GBC Redundancy - Dual Bus with Floating Master
- CPU Redundancy (GDB) using a Mixed Genius Redundancy Scheme

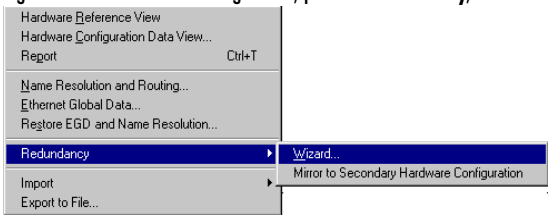
**Want to know more?** In the Help Index, look up "Redundant HWC: an Overview".

## Configuring Redundant Systems

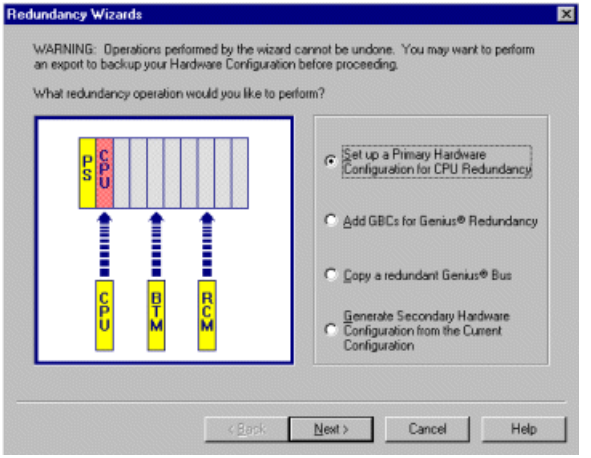
The procedures below are the general steps to follow for configuring redundant systems. See the online help for the procedures on configuring specific redundancy sytems.

### To set up the primary hardware configuration for CPU redundancy

1. In the Project tab of the Navigator, expand the target node for which you want to set up CPU redundancy.
2. Right-click Hardware Configuration, point to **Redundancy**, and choose **Wizard**.

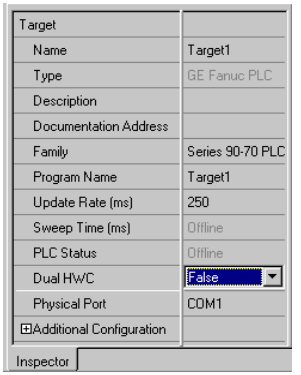



The Redundancy Wizards dialog box opens, with the *Set up a Primary Hardware Configuration for CPU Redundancy* option selected by default.



3. Click **Next** and follow the wizard to the end. The wizard adds a redundant CPU, the Bus Transmitter Module (BTM) and the Redundancy Communications Module (RCM) to the configuration.

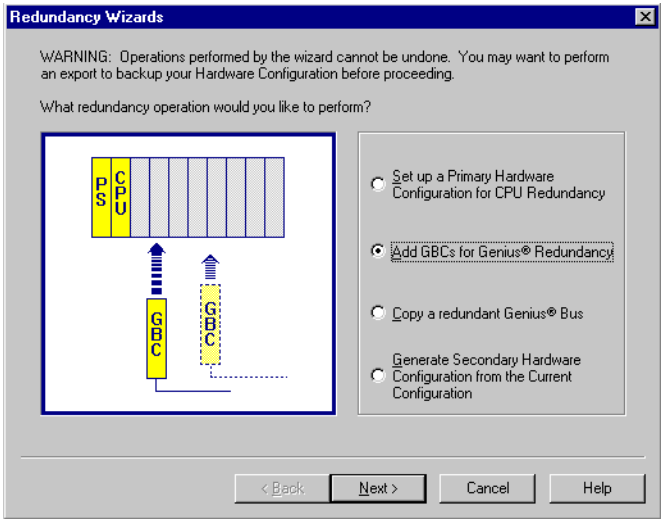
   **Note:** The target property, *Dual HWC*, is now available in the Inspector and is set to False.




## Adding Genius Bus Controllers (GBC) to your system

1. In the Project tab of the Navigator, expand the target for which you want to set up CPU redundancy.
2. Right-click Hardware Configuration, point to **Redundancy**, and choose **Wizard**.

   The Redundancy Wizards dialog box appears.

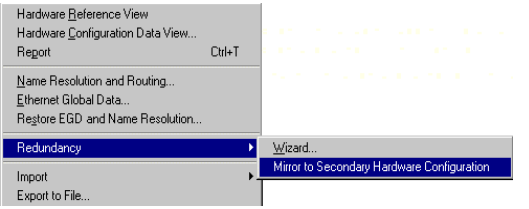


3. Select **Add GBCs for Genius Redundancy**.
4. Click **Next** and follow the wizard to the end. The wizard allows you to select a Genius redundancy scheme and the location of the GBC modules. You can run this wizard multiple times to configure additional pairs of redundant busses in the same system.

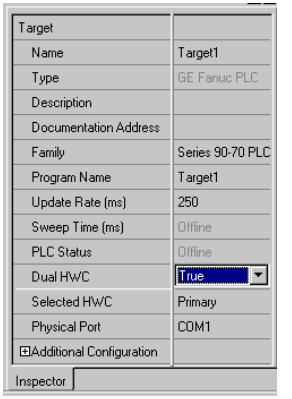
   **Note:** When you add a Genius Bus Controller (GBC) to a Series 90™-70 rack, a new Genius bus network is automatically created and associated with that slot and GBC module. Up to 31 Genius I/O devices (blocks) can be connected to a GBC through its Genius bus.

## Configuring the secondary hardware configuration

1. In the Project tab of the Navigator, select the Hardware Configuration folder.
2. Right-click Hardware Configuration, point to **Redundancy**, and choose **Mirror to Secondary Hardware Configuration**.



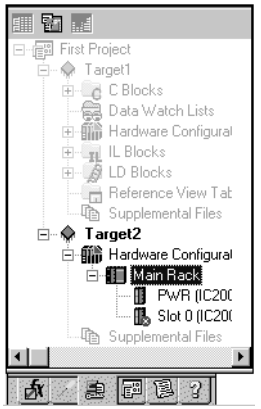

A secondary rack system that is a copy of the primary rack system is generated. The target now displays two Hardware Configurations (HWCs), respectively labeled [Primary] and [Secondary]. The Primary HWC is bolded because it is currently selected. The target property, *Dual HWC*, is now set to True.



**Note:** You can mirror as many times as necessary to synchronize the two HWCs after modifying the primary HWC. Each time you mirror the primary HWC, the secondary HWC is updated to reflect those changes.

**Want to know more?** In the Help Index, look up "Redundancy".

## VersaMax Remote I/O




**Navigator: VersaMax Remote I/O**

VersaMax remote I/O consists of a Network Interface Unit (NIU) (Ethernet, Genius or Profibus), and one or more I/O modules. If your target type is remote I/O, you can use Logic Developer - PLC to configure this hardware. Once configured, the remote I/O can be controlled from a PLC or PC controller. The differentiating factor between a VersaMax remote I/O and a PLC is that a remote I/O is simply an input/output device with a communication interface. Unlike a PLC, a remote I/O has no CPU.

When working with the VersaMax Remote I/O hardware configuration, Logic Developer - PLC enables you to add racks, configure the power supply and configure modules.

Remote I/O targets are generally added to a project when you create a project with a template (see page 12). You can add additional Remote I/O targets later.

### To create a project containing a GE Fanuc Remote I/O target from a template

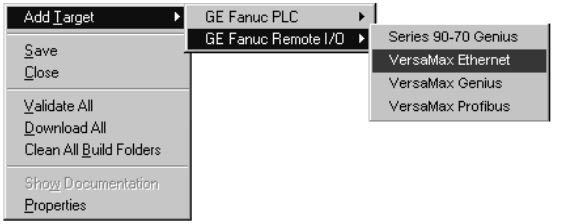
1. From the **File** menu, choose **New Project**.

   The New Project dialog box appears.
2. From the Project Template list, choose the GE Fanuc Remote I/O you want to add to your project.
3. Enter a descriptive Project Name.
4. Click **OK**.

   A new Remote I/O project is started.

### To add a GE Fanuc Remote I/O target to an existing project

1. In the Project tab of the Navigator, right-click the Project node.
2. Point to **Add Target**, then to **GE Fanuc Remote I/O,** and choose the remote I/O you want to add.

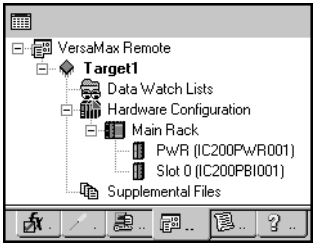



**Note:** The **Series 90-70 Genius** option is a Series 90™-70 Remote I/O Scanner.

### To add a power supply to your remote I/O configuration

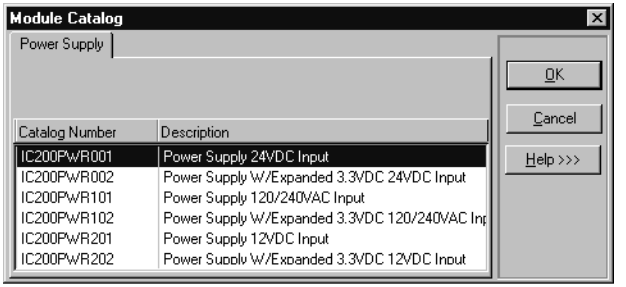
1. In the Project tab of the Navigator, expand the Hardware Configuration node of the remote I/O.

   The Navigator displays the following:




2. Right-click the PWR node and choose **Replace Module**.

   The Module catalogue dialog box appears:

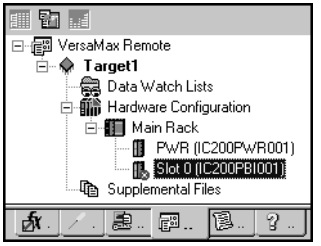



3. From the list, select the power supply that you want to configure for your system.

### To add a new Carrier/Base to your VersaMax Remote I/O

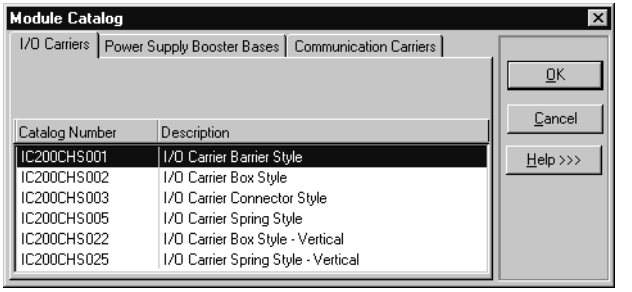
1. In the Project tab of the Navigator, select a remote I/O target.
2. Expand the Hardware Configuration node and the Main Rack node.

   The Navigator displays the following:




3. Right-click the Slot 0 node and choose **Add Carrier/Base**.

The Module Catalog dialog box appears.




4. **Select the carrier/base that you want to add to the remote I/O target.**

   **Note:** A maximum of eight carrier modules can be added to each VersaMax® rack.

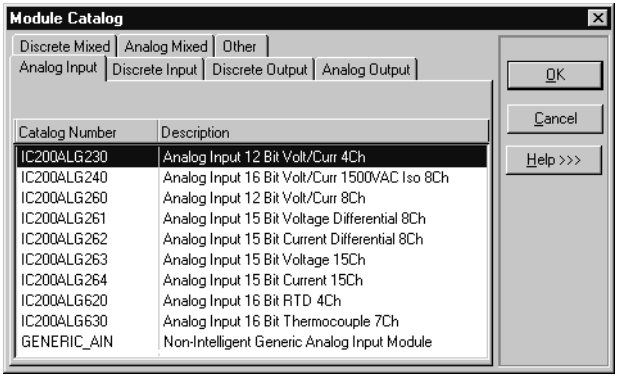
5. **Click OK.**

### To add a module to a carrier/base

1. **Double-click an empty carrier/base.**

   The Module Catalog appears.

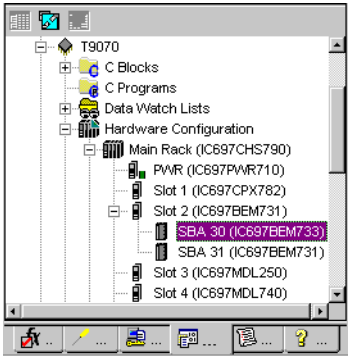
2. **From the Module Catalog list, select the module that you want to install.**



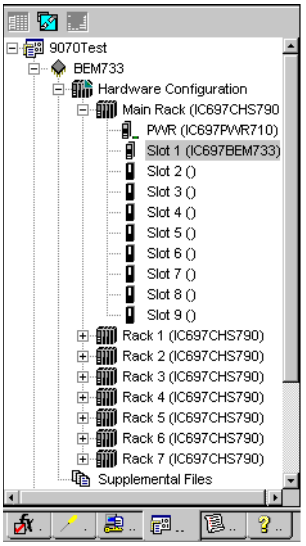

**Want to know more?** In the Help Index, look up "Hardware Configuration".

## Series 90™-70 Genius Remote I/O Scanner

A Remote I/O Scanner (IC697BEM733) is a Genius device that is part of a Genius Bus system. As such, it is added as a Genius device to the Genius Bus Controller (GBC) and it is represented in Logic Developer - PLC by an SBA node under the GBC node in a Series 90™-70 target.




**Navigator: Series 90™-70 Remote I/O Scanner as a Genius device under a GBC**




**Navigator: Series 90™-70 Remote I/O Scanner as a target and as a module in Slot 1 of the main rack**

A Remote I/O Scanner resides in Slot 1 of the main rack of a Series 90™-70 PLC. I/O modules can be inserted into the other slots of the main rack and on all the slots of the other racks. As such, the Remote I/O Scanner is represented by a target in the Project tab of the Navigator, and is also represented by a module in slot 1 of the main rack of that target.

### To set up a Remote I/O Scanner

1. Add the Remote I/O Scanner as a Genius device to the GBC.
2. When prompted to add a new target for the Remote I/O Scanner, click **Yes**.

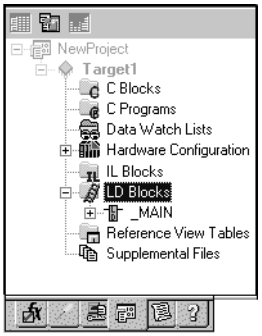
   Logic Developer - PLC automatically sets up some of the properties that link the Remote I/O Scanner to its GBC.
3. Set up the SBA to the same value to the Remote I/O Scanner as a target and to the Remote I/O Scanner as a Genius Device.

   Now whenever you update the I/O map on the Remote I/O Scanner, the I/O map is automatically updated to the GBC.
4. Configure the Remote I/O Scanner's target properties.
5. Add and configure the I/O on the Remote I/O Scanner's rack system.
6. Configure the Remote I/O Scanner's module properties.
7. Configure the properties of the Remote I/O Scanner as a Genius bus device.

**Want to know more?** In the Help Index, look up "IC697BEM733, Remote I/O Scanner".

# 5

# Logic Editors




Logic is organized into blocks and user programs. Each block and user program is a named section of executable code. Blocks and programs are grouped into folders. The following logic types are supported by Logic Developer - PLC (not including motion programming) and are compiled and downloaded to the PLC represented by the associated GE Fanuc PLC target:

- Ladder Diagram (LD)
- Instruction List (IL)
- C Blocks
- C programs

Each target must contain one block named "_MAIN". On any target except a release 6 or later Series 90™-70 PLC, the _MAIN block is first to execute when the project is downloaded to a PLC. Logic editors work in conjunction with global variables to operate and provide addressable units of logic that function as subroutines, called from another block.

Depending on the type of target PLC, blocks other than"_ MAIN" can be scheduled for execution based upon timer or I/O interrupts.

**Note:** On release 6 or later of Series 90™-70 PLCs, the LD program can be scheduled. The Series 90™-70 PLC does not necessarily execute the _MAIN LD block first.

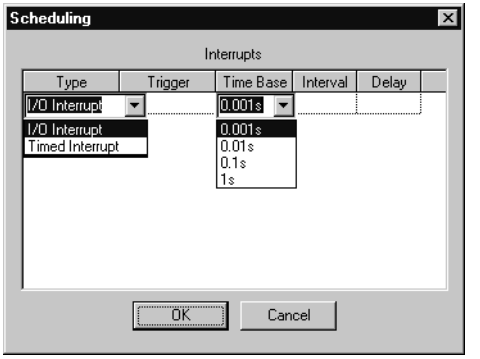
**Want to know more?** In the Help Index, look up "LD Program".

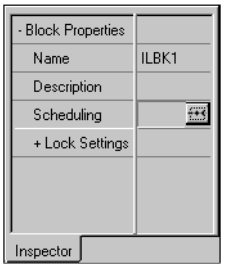
### To schedule execution of a logic block

1. In the Projects tab of the Navigator, right-click an existing LD, C or IL block, and choose **Properties**.

   The block's properties appear in the Inspector.
2. In the Inspector click the button in the Scheduling field.

The Scheduling dialog box appears.




**Inspector: IL, LD, C properties**

3. Configure scheduling by entering values in each of the fields.

**Want to know more?** In the Help Index, look up "Scheduling."

## To set access control

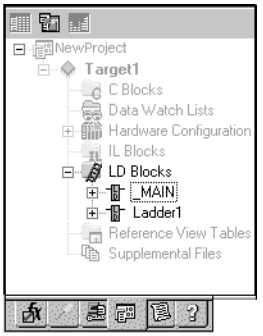
1. Right-click on an LD, IL or C Block and choose **Properties**.
2. In the Inspector, expand the Lock Settings property.
3. In the Lock Type property, choose a setting from the list.
4. In the Password property, enter a password.

   After a lock type has been set, you must enter the password to change the setting. When a block is unlocked, the password is destroyed.

## To search/replace in one block

1. Double-click an LD, IL, Local Logic, or Motion block to open it.
2. From the **Search** menu, choose **Find** or **Replace**.

   A dialog box appears.
3. In the **Find what** or in the **Text to find** field, enter the text to find.
4. (Only if you want to replace text) In the **Replace with** field or in the **New text** field, enter the text to replace the found text with.
5. Select or clear each search/replace option, as required.
6. Click **Find**, **Find next**, **Replace**, **Replace All**, **Close**, or **Cancel**, as required.




**Navigator: Project tab**

**Ladder Editor**

# LD EDITOR

The Ladder Diagram (LD) editor is used to create programs with the Ladder Diagram programming language. LD logic graphically represents the programmed actions performed by a PLC as it executes.

The LD editor is cell-based with rungs constructed of horizontal sequences of instructions that are wired together. A given instruction and its operands can occupy one or more cells.

You can work with the LD editor while offline to edit a disk copy of a project, or while online to monitor the execution of the logic while you fine tune the project by making word-for-word changes (see page 58).

You can customize the appearance and behavior of the LD editor.

An LD block is a named section of LD Logic that is compiled and downloaded to the PLC represented by the associated target. VersaMax and the Series 90-30 CPUs support a maximum of 64 Subroutine blocks plus one _MAIN block for a maximum of 65 blocks for a given target. The Series 90-70 CPUs support a maximum of 255 Subroutine blocks plus one _MAIN block for a maximum of 256 blocks for a given tartet.

### To customize the LD editor

1. In the Options tab of the Navigator, expand the Editors folder and then the Ladder folder.
2. Right-click a page (Confirmations, Editing, Font and Colors, or View), and choose **Properties**.

   The configurable settings appear as properties in the Inspector.
3. In the Inspector, adjust settings as required.

### To create an LD block

1. In the Project tab of the Navigator, right-click the LD blocks folder and click **New**.

   A new LD block with a default name is created.

   **Note:** If you use a template or add a target, the first block added to a target is named "_MAIN" and subsequent blocks added are named LDBK1, LDBK2, and so on by default.
2. Rename the block as desired.

### To open an LD block for editing

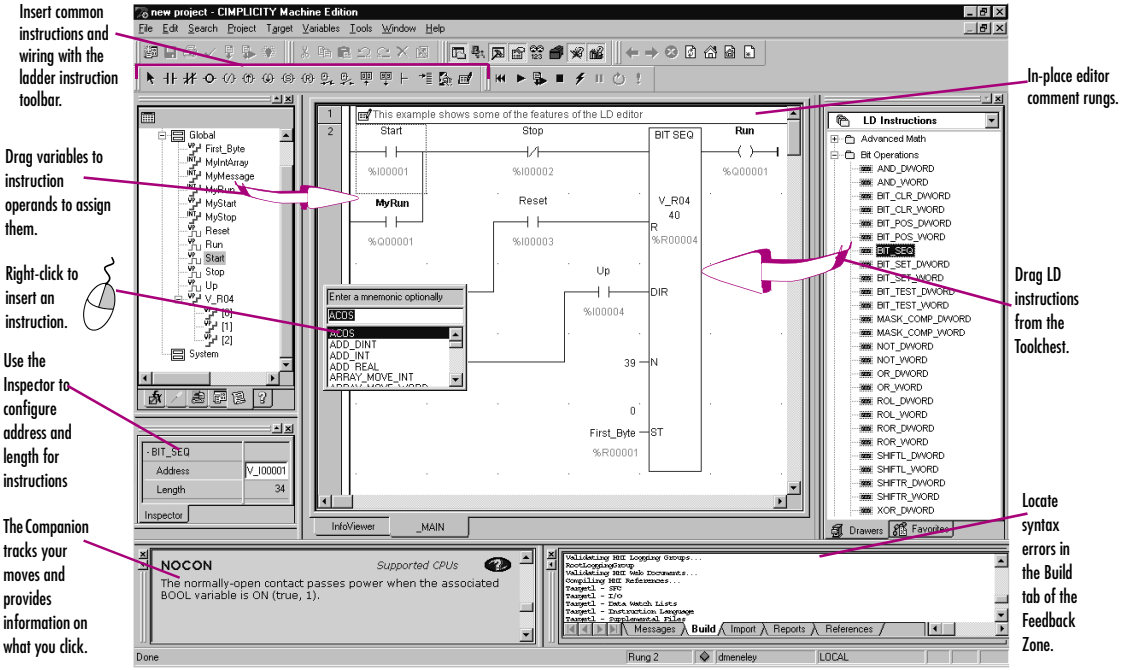
- In the Project tab of the Navigator, double-click an LD block.

The block opens in the LD editor.

**Note:** You can have multiple blocks open for editing. To navigate to another open LD block, click the tab that displays its name at the bottom of the editor window.

## Working with the LD Editor Offline

While in offline mode, there is no live communication between the LD editor and the target. Logic development is mostly done while offline. The following diagram illustrates some of the more common operations you can perform using the ladder editor offline.




### To insert an instruction

1. In the LD editor, right-click an empty cell and chose **Place Instruction**.

   A smart list appears listing all available instruction mnemonics.

2. Choose an instruction mnemonic from the list and press ENTER.

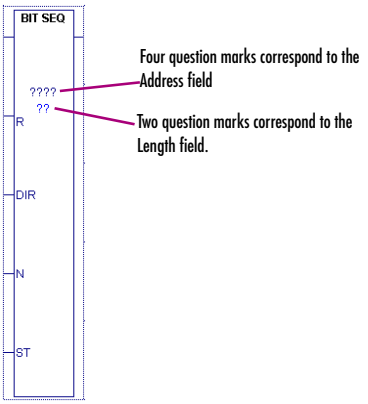
   **Note:** Only a contact or a horizontal wire can be placed in the first column.

## To configure an instruction's address and length properties

There are two methods to configure an instruction's address and length properties.

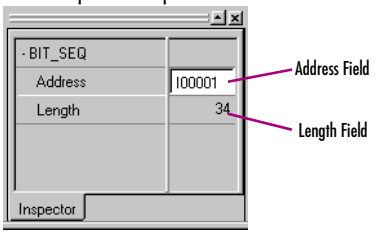
**First method**

1. In the LD editor, insert an instruction that requires Address and Length configuration.




2. Right-click the instruction and choose Properties.

   The Inspector opens with the instruction's properties shown.



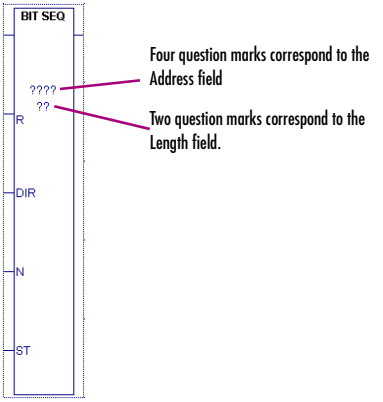

3. In the Address field, enter a variable name or a reference address to specify the start of a memory block.

   **Note**: If you type a reference address, it is converted to a variable name automatically.

4. In the Length Field, enter the number of PLC registers that the instruction requires in the memory block.

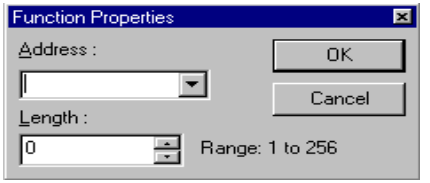
### Second method

1. In the LD editor, insert an instruction that requires Address and Length configuration.




2. Double-click the instruction or press ENTER.

   The Function Properties dialog box appears.




3. In the Address field, enter a variable name or a reference address to specify the start of a memory block.

   **Note:** If you type a reference address, it is converted to a variable name automatically.

4. In the Length field, enter the number of PLC registers that the instruction requires in the memory block.

## To assign variables to instruction operands

1. In the LD editor, double-click beside an operand of an instruction, or click there and press ENTER.

   A smart list appears prompting you to enter a variable name or reference address.

2. Type or choose from the list, a variable name or reference address.

   If you enter a reference address, a variable name will be automatically substituted.

## Working with the LD Editor Online

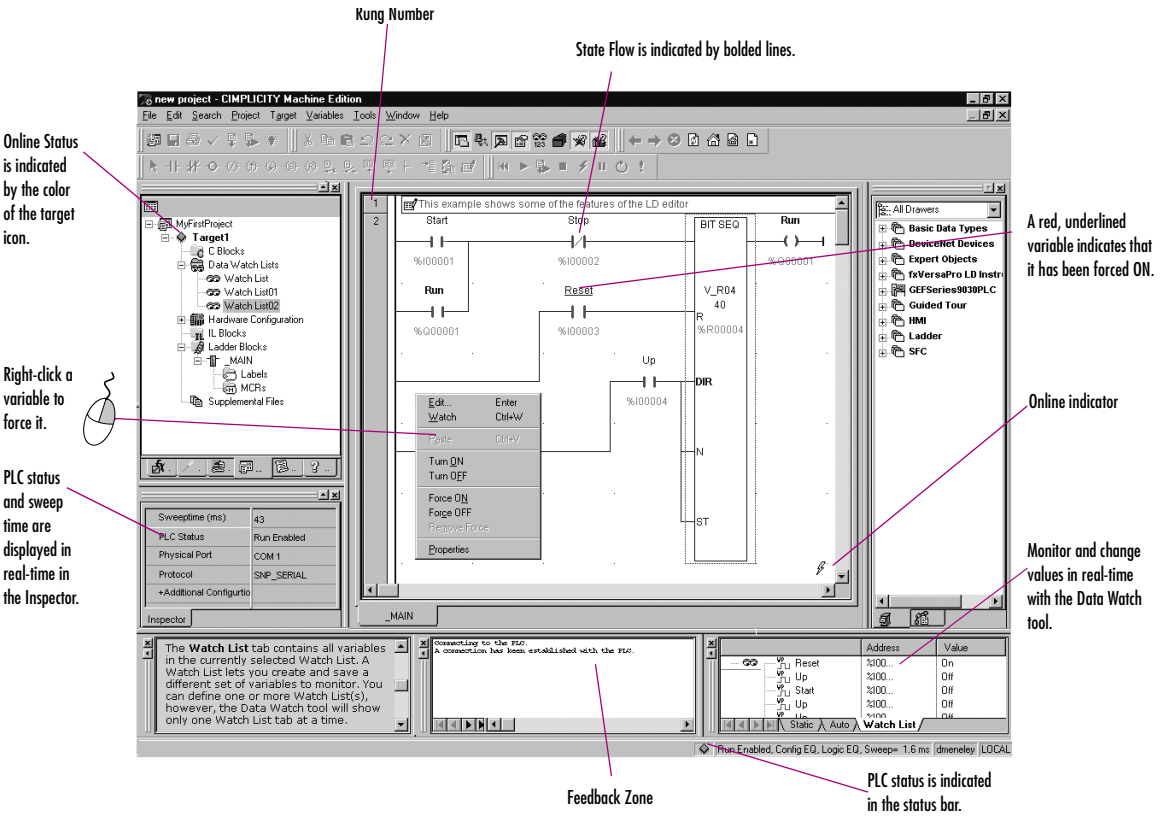
In online mode, there is a live connection to the target PLC enabling increased interaction with Logic Developer - PLC. The PLC's project must be equal to the current project for full functionality.

While online, the LD editor animates the LD logic to reflect program execution in the target PLC. Data values change in real time while coils and contacts indicate state flow. You can edit the LD program just as you would while offline. You can write the changed logic to the PLC while online.

### To go online to a target PLC

- In the Project tab of the Navigator, right-click the Target and choose **Go Online**.

  The target icon changes and the status bar updates to reflect the online status. The LD editor, if equal to the PLC, displays various online features, as shown in the following illustration:




## Word-for-Word Changes

Word-for-word changes may be completed online to Series 90™-30, Series 90™-70, Series 90™ Micro, and VersaMax® modular PLCs. (VersaMax Nano/Micro PLCs do not support word-for-word changes).

Changes to the program that do not modify its size are considered word-for-word changes. Examples include changing the type of contact or coil, or changing a reference address used for an existing function block.

### To make word-for-word changes

1. While online to a target PLC, make a change to LD logic that does not change the logic size.

   You are prompted to download the change.

2. Click **Yes**.

The changes are written to the PLC.

## Writing Changes to a Target PLC

If you change the size of the program, you must write the changes to the PLC to maintain equality. Writing changes to a target PLC requires the PLC to be online and running. All PLC families support this run-mode store capability.

**Note:** If you go offline, you can do a regular download.

### To write changes to the target PLC

1. While online to a target PLC that is running, make a change to LD logic.
2. In the Project tab of the Navigator, right-click the target, point to **Online Commands** and choose **Write Changes**.

   New logic is immediately written to the PLC without stopping the PLC. If the LD logic was altered, other than a word-for-word change, you are prompted to download to the running PLC. The entire target is validated, but only the changed blocks are downloaded.

### To turn on/off or force a variable

- In the LD editor, right-click a BOOL variable anywhere in LD logic and choose **Force ON**, **Force OFF**, **Turn ON**, or **Turn OFF**.

  **Note:** Forcing a variable ON or OFF overrides any actions the application may take during runtime. That is, if a variable is forced OFF (0), but LD logic is trying to turn it ON (1), it stays OFF.

**Want to know more?** In the Help Index, look up "LD".

## LD Functions

The following is a list of all LD functions available in Logic Developer - PLC. An asterisk (*) indicates the functions exclusive to Series 90™-70 PLCs. Companion help indicates which PLCs support the other functions.

**Want to know more?** In the Help Index, look up "LD Functions".

### Advanced Math

| | | | | |
|---|---|---|---|---|
| ACOS | COS | LN | SQRT_DINT | TAN |
| ASIN | EXP | LOG | SQRT_INT | |
| ATAN | EXPT | SIN | SQRT_REAL | |

### Bit Operations

| | | | |
|---|---|---|---|
| AND_DWORD* | BIT_SET_DWORD* | NOT_WORD | SHIFTL_DWORD* |
| AND_WORD | BIT_SET_WORD | OR_DWORD* | SHIFTL_WORD |
| BIT_CLR_DWORD* | BIT_TEST_DWORD* | OR_WORD | SHIFTR_DWORD* |
| BIT_CLR_WORD | BIT_TEST_WORD | ROL_DWORD* | SHIFTR_WORD |
| BIT_POS_DWORD* | MASK_COMP_DWORD | ROL_WORD | XOR_DWORD* |
| BIT_POS_WORD | MASK_COMP_WORD | ROR_DWORD* | XOR_WORD |
| BIT_SEQ | NOT_DWORD* | ROR_WORD | |

### Coils

| | | |
|---|---|---|
| COIL | NEGCOIL | SETCOIL |
| CONTCOIL | POSCOIL | |
| NCCOIL | RESETCOIL | |

### Contacts

| | | |
|---|---|---|
| CONTCON | LOALR* | NOCON |
| FAULT* | NCCON | NOFLT* |
| HIALR* | NEGCON* | POSCON* |

### Control

| | | | |
|---|---|---|---|
| DO_IO | EXIT_FOR* | PID_ISA | SVC_REQ |
| DRUM | FOR_LOOP* | SER | |
| END_FOR* | PID_IND | SUS_IO* | |

### Conversions

| | | | | |
|---|---|---|---|---|
| BCD4_TO_INT | DINT_TO_BCD8* | INT_TO_REAL | REAL_TO_WORD | UINT_TO_REAL* |
| BCD4_TO_REAL | DINT_TO_INT* | INT_TO_UINT* | TRUNC_DINT | WORD_TO_REAL |
| BCD4_TO_UINT* | DINT_TO_REAL | RAD_TO_DEG | TRUNC_INT | |
| BCD8_TO_DINT* | DINT_TO_UINT* | REAL_TO_DINT | UINT_TO_BCD4* | |
| BCD8_TO_REAL* | INT_TO_BCD4 | REAL_TO_INT | UINT_TO_DINT* | |
| DEG_TO_RAD | INT_TO_DINT* | REAL_TO_UINT* | UINT_TO_INT* | |

### Counters

| | |
|---|---|
| DNCTR | UPCTR |

### Data Move

| | | | |
|---|---|---|---|
| BLK_CLR_WORD | DATA_INIT_DLAN* | MOVE_UINT* | VME_RMW_BYTE* |
| BLKMOV_DINT* | DATA_INIT_DWORD* | MOVE_WORD | VME_RMW_WORD* |
| BLKMOV_DWORD* | DATA_INIT_INT* | SHFR_BIT | VME_TS_BYTE* |
| BLKMOV_INT | DATA_INIT_REAL* | SHFR_DWORD* | VME_TS_WORD* |
| BLKMOV_REAL | DATA_INIT_UINT* | SHFR_WORD | VME_WRT_BYTE* |
| BLKMOV_UINT* | DATA_INIT_WORD* | SWAP_DWORD* | VME_WRT_WORD* |
| BLKMOV_WORD | MOVE_BOOL | SWAP_WORD* | |
| COMM_REQ | MOVE_DINT* | VME_CFG_READ* | |
| DATA_INIT_ASCII* | MOVE_DWORD* | VME_CFG_WRITE* | |
| DATA_INIT_COMM* | MOVE_INT | VME_RD_BYTE* | |
| DATA_INIT_DINT* | MOVE_REAL | VME_RD_WORD* | |

### Data Table

| | | | |
|---|---|---|---|
| ARRAY_MOVE_BOOL | FIFO_WRT_WORD* | SEARCH_GE_UINT* | SEARCH_NE_DINT |
| ARRAY_MOVE_BYTE | LIFO_RD_DINT* | SEARCH_GE_WORD | SEARCH_NE_DWORD* |
| ARRAY_MOVE_DINT | LIFO_RD_DWORD* | SEARCH_GT_BYTE | SEARCH_NE_INT |
| ARRAY_MOVE_DWORD* | LIFO_RD_INT* | SEARCH_GT_DINT | SEARCH_NE_UINT* |
| ARRAY_MOVE_INT | LIFO_RD_UINT* | SEARCH_GT_DWORD* | SEARCH_NE_WORD |
| ARRAY_MOVE_UINT* | LIFO_RD_WORD* | SEARCH_GT_INT | SORT_INT* |
| ARRAY_MOVE_WORD | LIFO_WRT_DINT* | SEARCH_GT_UINT* | SORT_DINT* |
| ARRAY_RANGE_DINT* | LIFO_WRT_DWORD* | SEARCH_GT_WORD | SORT_WORD* |
| ARRAY_RANGE_DWORD* | LIFO_WRT_INT* | SEARCH_LE_BYTE | TBL_RD_DINT* |
| ARRAY_RANGE_INT* | LIFO_WRT_UINT* | SEARCH_LE_DINT | TBL_RD_DWORD* |
| ARRAY_RANGE_UINT* | LIFO_WRT_WORD* | SEARCH_LE_DWORD* | TBL_RD_INT* |
| ARRAY_RANGE_WORD* | SEARCH_EQ_BYTE | SEARCH_LE_INT | TBL_RD_UINT* |
| FIFO_RD_DINT* | SEARCH_EQ_DINT | SEARCH_LE_UINT* | TBL_RD_WORD* |
| FIFO_RD_DWORD* | SEARCH_EQ_DWORD* | SEARCH_LE_WORD | TBL_WRT_DINT* |
| FIFO_RD_INT* | SEARCH_EQ_INT | SEARCH_LT_BYTE | TBL_WRT_DWORD* |
| FIFO_RD_UINT* | SEARCH_EQ_UINT* | SEARCH_LT_DINT | TBL_WRT_INT* |
| FIFO_RD_WORD* | SEARCH_EQ_WORD | SEARCH_LT_DWORD* | TBL_WRT_UINT* |
| FIFO_WRT_DINT* | SEARCH_GE_BYTE | SEARCH_LT_INT | TBL_WRT_WORD |
| FIFO_WRT_DWORD* | SEARCH_GE_DINT | SEARCH_LT_UINT* | |
| FIFO_WRT_INT* | SEARCH_GE_DWORD* | SEARCH_LT_WORD | |
| FIFO_WRT_UINT* | SEARCH_GE_INT | SEARCH_NE_BYTE | |

### Math

| | | | | |
|---|---|---|---|---|
| ABS_DINT* | ADD_UINT* | MOD_DINT | MUL_REAL | SUB_REAL |
| ABS_INT* | DIV_DINT | MOD_INT | MUL_UINT* | SUB_UINT* |
| ABS_REAL* | DIV_INT | MOD_UINT* | SCALE_INT | |
| ADD_DINT | DIV_MIXED* | MUL_DINT | SCALE_WORD | |
| ADD_INT | DIV_REAL | MUL_INT | SUB_DINT | |
| ADD_REAL | DIV_UINT* | MUL_MIXED* | SUB_INT | |

### Program Flow

| | | | | |
|---|---|---|---|---|
| CALL | ENDMCR | JUMP | LABELN | V_WIRE |
| COMMENT | ENDMCRN | JUMPN | MCR | |
| END | H_WIRE | LABEL | MCRN | |

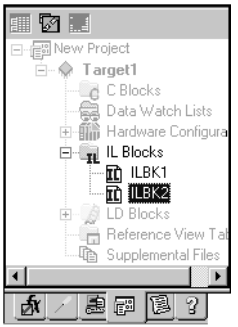
### Relational

| | | | | |
|---|---|---|---|---|
| CMP_DINT* | EQ_UINT* | GT_REAL | LT_INT | RANGE_DINT |
| CMP_INT* | GE_DINT | GT_UINT* | LT_REAL | RANGE_DWORD* |
| CMP_REAL* | GE_INT | LE_DINT | LT_UINT* | RANGE_INT |
| CMP_UINT* | GE_REAL | LE_INT | NE_DINT | RANGE_UINT* |
| EQ_DINT | GE_UINT* | LE_REAL | NE_INT | RANGE_WORD |
| EQ_INT | GT_DINT | LE_UINT* | NE_REAL | |
| EQ_REAL | GT_INT | LT_DINT | NE_UINT* | |

### Timers

| | | |
|---|---|---|
| OFDT_HUNDS | ONDTR_HUNDS | TMR_HUNDS |
| OFDT_SEC* | ONDTR_SEC* | TMR_SEC* |
| OFDT_TENTHS | ONDTR_TENTHS | TMR_TENTHS |
| OFDT_THOUS | ONDTR_THOUS | TMR_THOUS |

**Note:** An asterisk (*) indicates the functions exclusive to Series 90™-70 PLCs.




**Navigator: Project tab**

**IL Blocks nodes**

# IL EDITOR

Instruction List (IL) is a programming language specified by the IEC 61131-3 standard. This text language is accumulator-based and much like the assembly languages used for programming microprocessors. The instructions executed by the IL program modify or use an accumulator that is located in PLC memory. Two types of accumulators are defined: one analog accumulator for numeric and bitwise operations and eight boolean accumulators for discrete logic to support eight levels of nested boolean expressions. The IL editor is free-form with an option to apply a standard formatting rule. The appearance and behavior of the IL editor is user-configurable.

**Note:** IL logic is not supported on Series 90™-70 PLCs.

## To configure accumulators

1. In the Project tab of the Navigator, right-click the IL Blocks folder and choose **Properties**.

   The Inspector opens, displaying the Accumulator Address properties.
2. In the Boolean Start field, enter the reference address of the first of eight PLC memory locations to use for boolean accumulators.

   The ending address is calculated automatically. The memory area must be %T, %M, or %Q.
3. In the Analog Start field, enter the PLC memory locations to use for the analog accumulator.

   The ending address is calculated automatically. The memory area must be %R, %AI or %AQ.

## To create an IL block

1. In the Projects tab of the Navigator, right-click the IL blocks folder and choose **New**.

   An empty IL block with the default name "ILBk*n*" is added to the folder, where *n* represents a unique number.
2. Rename the block as desired.

## To open an IL block for editing

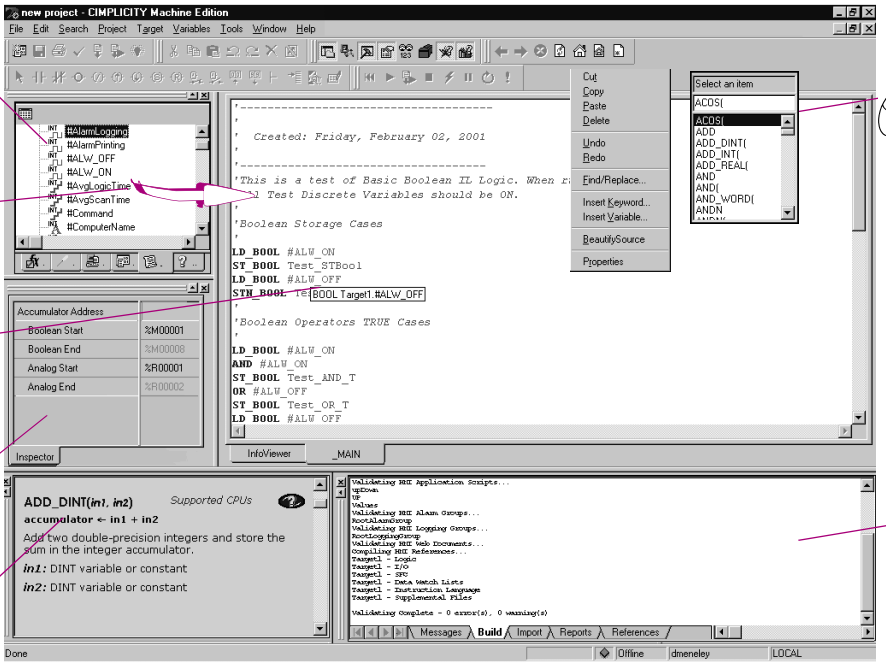
- In the Project tab of the Navigator, right-click an IL Block and choose **Open**.

  The block opens in the IL editor.

  **Note:** You can have multiple blocks open for editing. To navigate to another open IL block, click the tab displaying its name at the bottom of the editor window.

## Working with the IL Editor Offline

Most project development is carried out while offline from the target PLC. Editing while offline provides maximum flexibility in editing and allows you to interact with the Machine Edition tools as shown in the following diagram.




### To insert an instruction or function

1. In the IL editor, right-click and choose Insert Keyword.

   A smart list appears listing all available instruction mnemonics.

2. From the list select an instruction and press ENTER.

   The instruction is inserted in your logic.

### To assign operands to an instruction

1. In the IL editor, right-click and choose Insert Variable.

   A smart list appears showing all your defined variables.

2. Type or choose from the list, a variable name or reference address and then press ENTER.

   The name appears in your logic.

**Note:** If you entered a reference address or a new variable name, you must create a variable from it.

### To create a variable from a referenc address

1. In the IL editor, right-click a reference address, point to **Create "name" as**, and then choose a data type.

   A variable is created and a default name is applied. For example, if the reference address is %R0032, the auto-created variable is named R00032.

### To create a variable from a name

1. In the IL editor, right-click a name, point to **Create "name" as**, and then choose a data type.

   A variable is created with the name you right-clicked.
2. Map the variable to PLC memory. (See page 17.)

### To move or duplicate IL logic

1. In the IL editor, select a range of logic.
2. To move it, click the selection and drag it to a new location. To duplicate it, press CTRL while clicking the selection, and drag the selection to where you want to place the duplicate.

   When you release the mouse button, the selection is respectively moved to the new location or a duplicate copy of the selected logic is placed in the new area.

### To insert an inline comment

1. In the IL editor, click where you want to insert an inline comment.
2. Type an apostrophe (') followed by comment text.

   Press ENTER to complete the comment.

### To insert a block comment

1. In the IL editor, click where you want to insert a block comment.
2. Type (* followed by comment text.

   A block comment can contain any number of characters and can span multiple lines.
3. Type *) to complete the block comment.

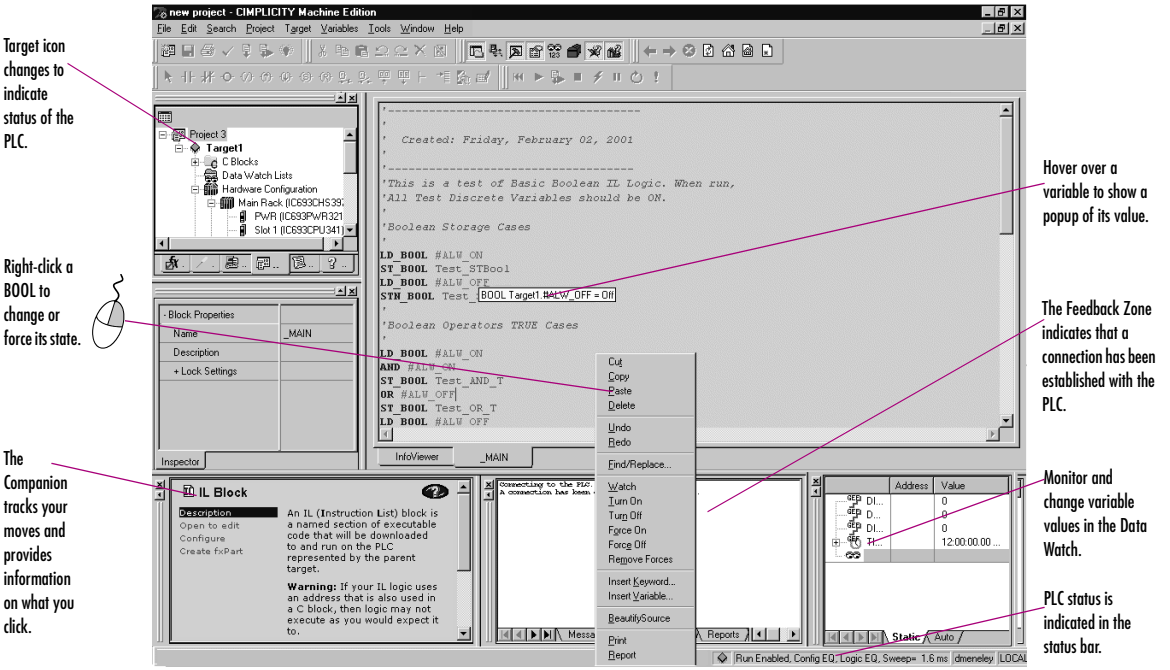
### To reformat IL code

- In the IL editor, right-click and choose Beautify Source.

  The entire content of the IL editor is reformatted according to the default formatting (indentation) rules.

## Working with the IL Editor Online

While online to a target PLC, the IL editor enables you to edit logic, monitor data values and change or force the state of BOOL variables in real-time. This allows you to test the execution of logic, to trigger an event within an executing project, or to remove an element from logic execution.




### To monitor a data value

- Click anywhere in the IL editor and hover the mouse pointer over a variable.

  A tooltip appears, showing the variable's current value. This value, however, does not update. You must move the mouse pointer away and back over the variable.

### To change a BOOL variable's state

- In the IL editor, right-click the BOOL variable whose value you want to change and choose **Turn On** or **Turn Off**.

  The state of the variable in the target PLC changes when the command is received. It remains in that state until acted on by the PLC's logic.

### To force a BOOL variable's state

- In the IL editor, right-click a BOOL variable and choose **Force On** or **Force Off**.

  The state of the forced variable will remain unchanged, regardless of any actions by the PLC's logic.

### To remove the force from a BOOL variable

- In the IL editor, right-click a BOOL variable and choose **Remove Forces**.

  The state of the forced variable will be controlled by the PLC's logic.

## Writing Changes to a Target PLC

You can change your IL logic while online, but you must write the changes to the PLC to maintain equality. Writing changes to a target PLC requires the PLC to be online and running.

**Note:** If you go offline, you can do a regular download.

### To write changes to the target PLC (if the target PLC supports it)

1. While online to a target PLC, make a change to IL logic.

   In the Project tab of the Navigator, right-click the target, point to **Online Commands**, and choose **Write Changes**.

   You are prompted to confirm a run mode store. That is, the altered IL block will be downloaded to the PLC without stopping the PLC.

**Want to know more?** In the Help Index, look up "IL"

## IL Instructions and Functions

The following is a list of all IL instructions and functions available in Logic Developer - PLC. An asterisk (*) indicates the functions exclusive to VersaMax® PLCs. Companion help indicates which PLCs support the other functions.

### Basic Instructions

| | | | | | | |
|---|---|---|---|---|---|---|
| ADD | CAL | GT | MOD | NT | R | SUB |
| AND | DIV | LE | MUL | OR | RET | XORN |
| AND_WORD | EQ | LT | NE | ORN | RETC | |
| ANDN | GE | MCRN | NOT | PT | S | |

### Advanced Math

| | | | | |
|---|---|---|---|---|
| ACOS | COS | LN | SQRT_INT | TAN |
| ASIN | EXP | LOG | SQRT_DINT | |
| ATAN | EXPT | SIN | SQRT_REAL | |

### Bit Operations

| | | | | |
|---|---|---|---|---|
| AND_WORD | BIT_SEQ | MASK_COMP_DWORD | OR_WORD | SHIFTL_WORD |
| BIT_CLR_WORD | BIT_SET_WORD | MASK_COMP_WORD | ROL_WORD | |
| BIT_POS_WORD | BIT_TEST_WORD | NOT_WORD | ROR_WORD | |

### Control

| | |
|---|---|
| DO_IO | PID_ISA |
| DRUM | SER |
| PID_IND | SVC_REQ |

### Conversions

| | | | | |
|---|---|---|---|---|
| BCD4_TO_INT | DINT_TO_REAL | RAD_TO_DEG | REAL_TO_WORD | WORD_TO_REAL |
| BCD4_TO_REAL | INT_TO_BCD4 | REAL_TO_DINT | TRUNC_DINT | |
| DEG_TO_RAD | INT_TO_REAL | REAL_TO_INT | TRUNC_INT | |

### Counters

| | |
|---|---|
| DNCTR | UPCTR |

### Data Move

| | | | |
|---|---|---|---|
| BLK_CLR_WORD | BLKMOV_DINT | MOVE_BOOL | MOVE_WORD |
| BLKMOV_INT | COMM_REQ | MOVE_REAL | SHFR_WORD |
| BLKMOV_REAL | MOVE_INT | SHFR_BIT | |

### Data Table

| | | | | |
|---|---|---|---|---|
| ARRAY_MOVE_BOOL | SEARCH_EQ_INT | SEARCH_GT_DINT | SEARCH_LT_BYTE | SEARCH_NE_WORD |
| ARRAY_MOVE_BYTE | SEARCH_EQ_WORD | SEARCH_GT_INT | SEARCH_LT_DINT | |
| ARRAY_MOVE_DINT | SEARCH_GE_BYTE | SEARCH_GT_WORD | SEARCH_LT_INT | |
| ARRAY_MOVE_INT | SEARCH_GE_DINT | SEARCH_LE_BYTE | SEARCH_LT_WORD | |
| ARRAY_MOVE_WORD | SEARCH_GE_INT | SEARCH_LE_DINT | SEARCH_NE_BYTE | |
| SEARCH_EQ_BYTE | SEARCH_GE_WORD | SEARCH_LE_INT | SEARCH_NE_DINT | |
| SEARCH_EQ_DINT | SEARCH_GT_BYTE | SEARCH_LE_WORD | SEARCH_NE_INT | |

### Math

| | | | | | |
|---|---|---|---|---|---|
| ADD_DINT | DIV_DINT | MOD_DINT | MUL_DINT | SCALE_WORD* | SUB_REAL |
| ADD_INT | DIV_INT | MOD_INT | MUL_REAL | SUB_DINT | |
| ADD_REAL | DIV_REAL | MUL_INT | SCALE_INT* | SUB_INT | |

### Program Flow

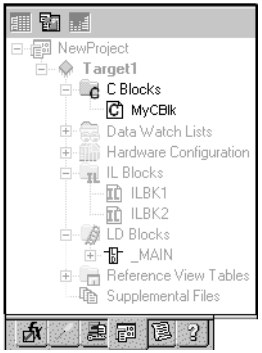
| | | | | |
|---|---|---|---|---|
| CALL | ENDMCR | JUMP | LABELN | V_WIRE |
| COMMENT | ENDMCRN | JUMPN | MCR | |
| END | H_WIRE | LABEL | MCRN | |

### Relational

| | | | | | | |
|---|---|---|---|---|---|---|
| EQ_DINT | GE_DINT | GT_DINT | LE_DINT | LT_DINT | NE_DINT | RANGE_DINT |
| EQ_INT | GE_INT | GT_INT | LE_INT | LT_INT | NE_INT | RANGE_INT |
| EQ_REAL | GE_REAL | GT_REAL | LE_REAL | LT_REAL | NE_REAL | RANGE_WORD |

### Timers

| | | |
|---|---|---|
| OFDT_HUNDS | ONDTR_HUNDS | TMR_HUNDS |
| OFDT_TENTHS | ONDTR_TENTHS | TMR_TENTHS |
| OFDT_THOUS | ONDTR_THOUS | TMR_THOUS |




**Navigator: Project tab**

**C Blocks node**

# C BLOCKS

A C block is an independent section of executable code written in the C programming language, that is downloaded to and executed on the target PLC. C blocks are created externally using GE Fanuc's C Programming Toolkit and then imported into a project. A C block is an .exe file and can be called as a subroutine from another block (LD or IL) but cannot call another block.

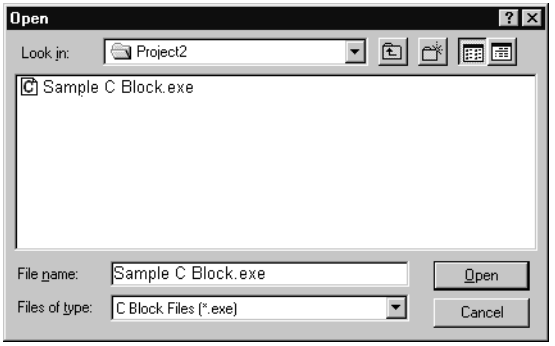
For more information on developing C Blocks, refer to *GFK-0646 E, C Programmer's Toolkit for Series 90 PLC's*.

## Working with C blocks

### To import a C block

1. In the Projects tab of the Navigator, right-click the C Blocks folder and choose Add.

   The Open dialog box appears.




2. Browse to the .exe file you want to import and click **Open**.

   The .exe file is added to the C Blocks folder with the same name as the .exe file.

**Want to know more?** In the Help Index, choose "C".

### To set a C block's parameters

**Note:** This procedure applies only to Series 90™-70 PLCs and to C blocks that were written to require parameters.

1. In the Projects tab of the Navigator, right-click the C block and choose **Properties**.

   The Inspector opens, showing the block's properties.

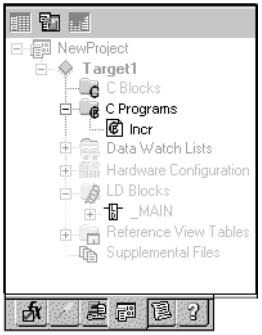
2. In the Inspector, select the Parameters property and click the ... .

   The Parameters dialog box opens.

3. Type a name and description for each required input and output parameter.

   For information on the required parameters, consult the written documentation for the C block. The names you enter will display in the CALL instruction. The names and descriptions will display as tooltip help when you hover over the CALL instruction.

**Want to know more?** In the Help Index, enter "Parameterized" and select the "C Block" help topic.




**Navigator: Project Tab**

**C Programs node**

# C PROGRAMS

**Note:** C programs are supported only on Series 90™-70 CPUs, firmware release 6.00 and later.

A C program is an independent section of executable code, written in C language, that is downloaded to and executed on the associated target PLC.

A C program has access to all the % reference tables of the PLC except for the _MAIN LD block's %P memory and the %L memory of any other LD block. A C program can also call any of the numerous PLC-embedded function blocks, which are included in the C Programmer's Toolkit.

A C program cannot be called as a subroutine. Execution is controlled only through scheduling. A C program cannot call another block as a subroutine.

By setting the parameters of a C program, you enable it to access memory directly. When a C program begins to execute, it reads the data for all the parameters and makes a copy of the data. If the C program's execution is interrupted or time-sliced over multiple scans, the C program, when it resumes execution, uses the copy of the data that it made upon beginning to execute.

The maximum number of C programs that can be used on a target depends on the nature of the _MAIN block:

- If the _MAIN block is an LD block, the LD program is treated as a user program and can be scheduled, and the maximum number of C programs per target is 15.
- If the _MAIN block is not an LD block, the maximum number of C programs per target is 16, and the LD program cannot be scheduled.

C programs can coexist with State Logic on a Series 90-70 PLC. For such a target, the State Logic is compiled into a C program. This C program is named _MAIN if there is no LD logic; however, if State Logic coexists with LD logic, the State Logic is compiled into a C program named _STATE and one LD block must be named _MAIN.

## User Programs

User programs consists of:

- C programs
- The LD program, if the _MAIN block is an LD block.

## Working with User Programs

### To configure execution scheduling

1. In the Projects tab of the Navigator, right-click a C program or the _MAIN LD block, and choose **Properties**.

   The Inspector opens, showing the user program's properties.
2. In the Inspector, expand the **Scheduling** property.
3. Select a Schedule Mode.
4. Select the other properties used in the schedule mode you selected.
5. Click **OK**.

   **Note:** If you made any entries, they are checked when you click **OK**. You must correct any errors before you can close the dialog box with the OK button.

### Setting a C program's parameters

**Note:** You must provide a list of all the input and output parameters that the C program requires. The LD program does not use parameters.

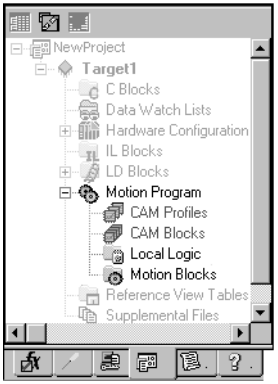
1. In the Projects tab of the Navigator, right-click a C program and choose **Properties**.

   The Inspector opens, showing the C program's properties.
2. In the Inspector, select the Parameters property and click ... .

   The Parameters dialog box opens.
3. On the Input and Output tabs, enter up to 8 input and up to 8 output parameters. Each parameter has its own row on the tab. For each parameter, double-click the following cells and enter the required data:
   - **Name:** The parameter's name.
   - **Type:** The parameter's data type.
   - **Length:** The length of the input or output reference.
   - **Variable:** The first data item associated with the parameter.
   - **Description:** (Optional) The parameter's description.

**Want to know more?** In the Help Index, choose "C Program".

# 6 Motion Programming




**Navigator: Project tab**

**Motion Program nodes**

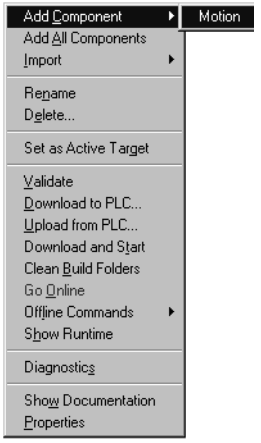
Logic Developer - PLC supports motion programming for the Motion Mate DSM314 motion control module. A high performance, easy-to-use, multi-axis motion control module, the Motion Mate DSM314 is highly integrated with the Series 90-30 PLC logic solving and communication functions.

The DSM314 (see page 38) supports 10 motion programs, 40 subroutines and a maximum total of 1000 motion program statements. Logic Developer - PLC, making motion programming possible, supports the following motion editors:

- PLC Motion Editor
- Local Logic Editor
- CAM Editor

This chapter outlines basic procedures that will get you started with Logic Developer - PLC to create motion programs, using these three motion editors.

## To add a Motion component to a target




**Target right-click menu**

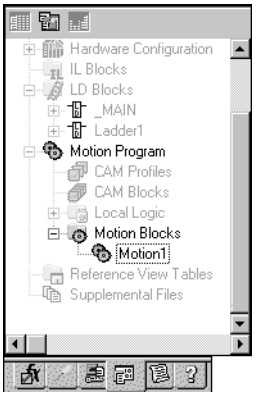
- In the Project tab of the Navigator, right-click the target, point to **Add Component** and then choose **Motion**.

  A Motion Program node is added to your project. Included are empty Motion Blocks, Local Logic, CAM Profiles and CAM Blocks folders.

# PLC MOTION EDITOR

Logic Developer - PLC includes a PLC Motion Editor, which enables you to create PLC Motion blocks for the DSM314. This text-based editor is configurable as to its appearance and behavior. Comments and white space are not considered as PLC Motion program statements. Motion programming has a different programming syntax from Local Logic.

### To add a Motion Block




**Navigator: Project tab**

**Motion Blocks**

1. In the Project tab of the Navigator, expand the Motion Program folder.
2. Right-click the Motion Blocks folder and choose **New**.

   A new empty PLC Motion block with a default name is added to your project.
3. Rename the block as desired.

### To open a motion block for editing

- In the Project tab of the Navigator, right-click the **Motion Block** node and choose Open.

  The block opens for editing in the PLC Motion editor

## Working with the Motion Editor

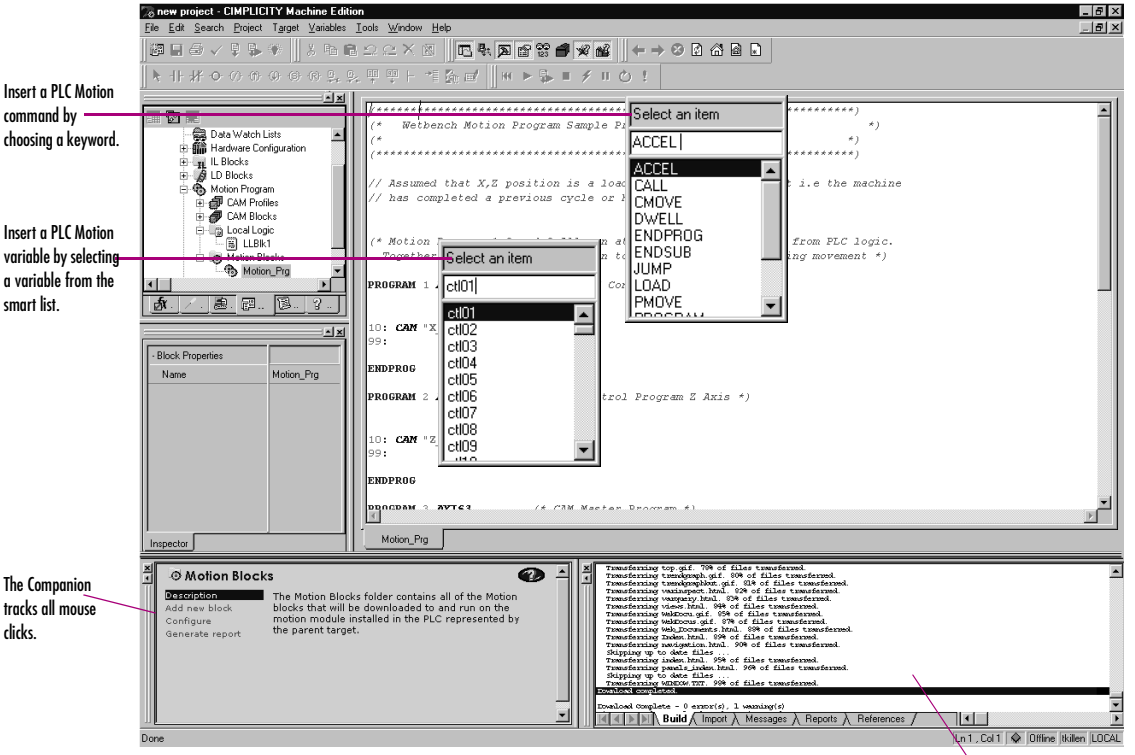
### To insert a command

1. In the Motion editor, right-click and choose **Insert Keyword**.

   A smart list appears showing all available motion commands.

2. Select the appropriate command from the smart list and press ENTER.

   The command is placed in the motion editor.

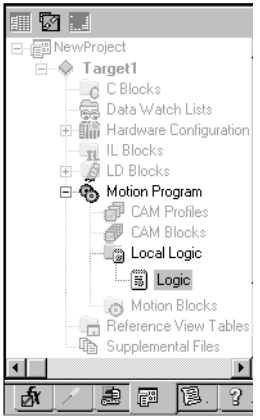



**Want to know more?** In the Help Index, choose "PLC Motion Editor".

## Motion Commands

| | |
|---|---|
| ACCEL | The ACCEL statement sets the axis acceleration for subsequent moves and remains in effect in a given program unless changed. Note: If a move instruction is executed before ACCEL, the tag Acceleration is used. |
| BLOCK NUMBER | Block numbers may be used as the destination of JUMP commands. Block numbers must be unique and can be between 1 and 65535. |
| CAM | The CAM statement runs a CAM profile. |
| CAM PHASE | CAM PHASE set the phase of a CAM profile. |
| CAM MARKER | CAM LOAD loads a CAM profile. |
| CALL | The CALL command executes another block as a subroutine |
| CMOVE | The CMOVE command programs a continuous move using the specified position and acceleration mode. |
| DWELL | DWELL causes motion to cease for a specified time period before processing the next command. |
| ENDP | The ENDPROG statement terminates a PLC Motion program definition. |
| ENDS | The ENDSUB statement terminates a PLC Motion subroutine definition. |
| JUMP | Jump to a block number or a sync block within the current program or subroutine. The jump may be conditional or unconditional based on the status of a CTL bit. |
| LOAD | Initializes or changes a parameter data register with a 32-bit twos-complement integer value. |
| PMOVE | The PMOVE command programs a positioning move using the specified position and accelerator mode. |
| PROGRAM | The PROGRAM statement is the first statement in a motion program. The program statement identifies the program number (1-10) and the axis configuration. Program definitions cannot nest. |
| SUBROUTINE | The SUBROUTINE statement is the first statement in a motion subroutine. The subroutine statement identifies the subroutine number (1-40) and the axis configuration. |
| SYNC BLOCK | A sync block is a special case of a block number. A sync block may only be used in multi-axis programs. |
| VELOC | Sets the process VELOCITY used by subsequent motion program move commands and remains in effect until changed by another VELOC statement. |

# LOCAL LOGIC




**Navigator: Project tab**

**Local Logic**

A Local Logic program runs synchronously with the Motion program, but is independent of the PLC's CPU scan. This allows the DSM314 to interact much more quickly with motion I/O signals on its faceplate connectors than would be possible if the logic for the signals was handled in the main ladder program running on the PLC.

Local Logic language uses free-form, text-based circuits and contains basic mathematical and logical constructs. Local Logic programming syntax allows you to assign a variety of logic tasks to your motion programs while working in conjunction with PLC Logic programs and motion programs to yield a flexible programming environment. Because it uses straightforward, understandable syntax, it is easy to gain proficiency with this editor.

The Local Logic programming language supports assignments, conditional statements, arithmetic, logical and relational statements. Local Logic provides the user access to motion controller data, parameters using a fixed set of variables, control and status bits.

- Parameter data - accessible from Local Logic host PLC and motion programs. The parameter data are similar to variables in a program.
- CTL bits - allow the Local Logic program or host PLC to signal the motion program to start an event.
- Motion program block numbers - the current block number can be used within the Local Logic program or host PLC to make an action occur only during a specific motion programming section.

### To create a Local Logic block

1. In the Projects tab of the Navigator, right-click the Local Logic folder and choose **New**.

   A new Local Logic block with a default name is created.

   **Note:** The Local Logic folder can contain only one Local Logic block.
2. Rename the block as required.

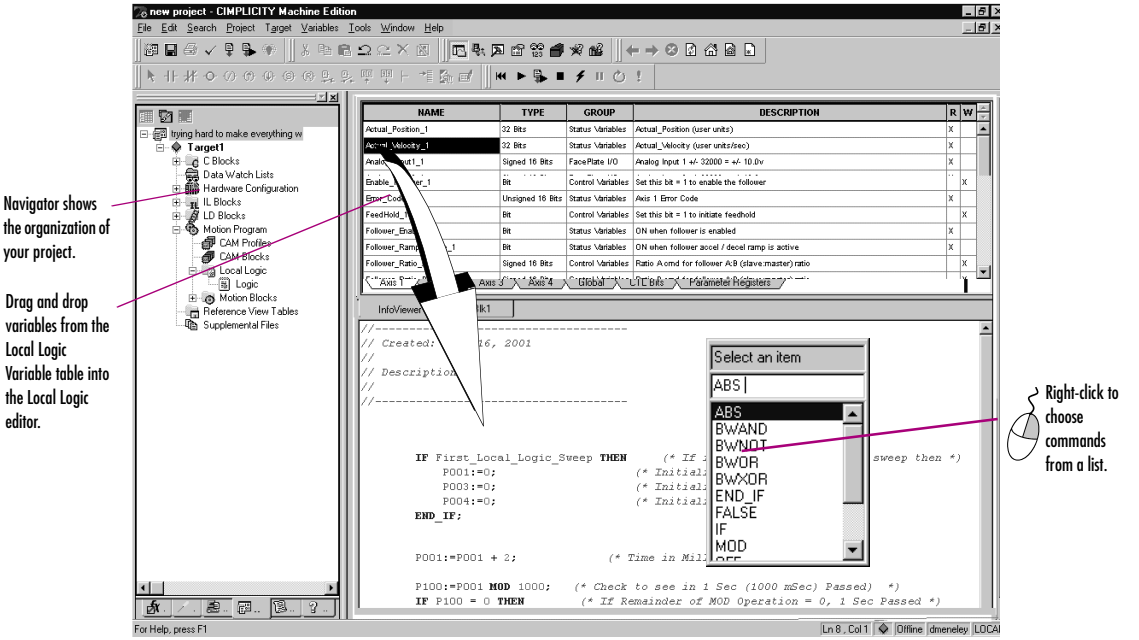
### To open a Local Logic block for editing

- In the Projects tab of the Navigator, right-click the Local Logic block and choose **Open**.

  The Local Logic block opens in the Local Logic editor and is ready for editing.

## Working with Local Logic Editor

The Local Logic editor has its own distinctive syntax for constructing Local Logic programs.




### To insert a Local Logic command

1. In the Local Logic editor, right-click and choose **Insert Keyword** from the menu.

   A smart list appears showing all available Local Logic commands.

2. Select the desired command in the smart list and press ENTER

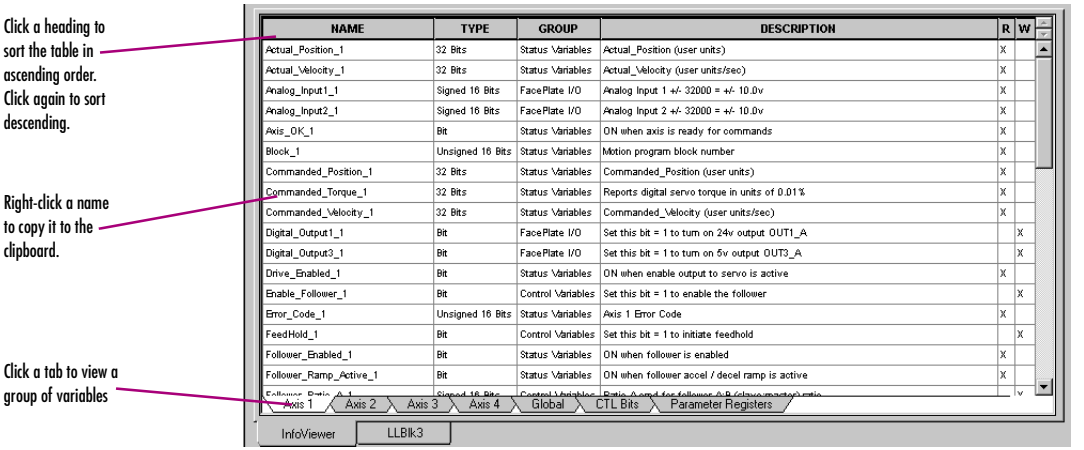
   The command is inserted.

   **Note:** You can drag variables from the Local Logic Variable Table to the Local Logic editor.

## Local Logic Variables

Local Logic is designed to complement a PLC's logic and mathematical capabilities. Solving small Local Logic and mathematical sets require a tight synchronization with the controlled motion.

Logic Developer - PLC includes a table containing Local Logic variables, the Local Logic Variable Table (LLVT), that you can drag into your logic programs. As illustrated in the diagram below, the LLVT has several tabs that organize the variables by category:




### To view the LLVT

1. In the InfoViewer tab of the Navigator, expand the Logic Developer - PLC library then expand the Local Logic Editor book and double-click Local Logic Variable Table.

The LLVT appears in the Infoviewer.

The variables or data in each tab are:

| | |
|---|---|
| Axis 1 | Variables specific to axis 1 |
| Axis 2 | Variables specific to axis 2 |
| Axis 3 | Variables specific to axis 3 |
| Axis 4 | Variables specific to axis 4 |
| Global | Global data such as module status code |
| CTL bits | DSM general Control/Status bits |
| Parameter Registers | DSM parameter data |

The table has six columns:

| | |
|---|---|
| Name | Contains the variable name that is to be used within a Local Logic program |
| Type | The data type for this variable. For example, 32-bit means that this variable is a 32-bit variable. |
| Group | The group this variable is placed in. For example, Faceplate I/O means that this variable refers to a point on the module faceplate. |
| Description | This column contains a textual description of the variable. When the user hovers the mouse pointer over the description, a tool tip displays the description. |
| R | This column indicates if the variable can be read by a Local Logic program. |
| W | This column indicates if the variable can be written by a Local Logic program. |

### To insert a Local Logic variable

1. In the Local Logic editor, right-click and choose **Insert Variable**.

   A smart list appears prompting you to choose a Local Logic variable name.

2. Select a variable in the list and press ENTER.

   The variable is inserted in your Local Logic.

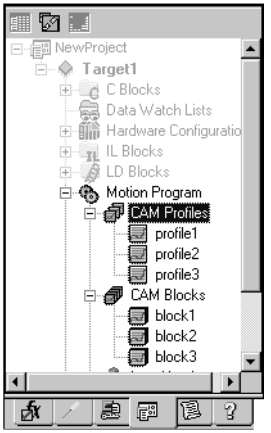
## Local Logic Commands and Operators

Local Logic provides the capability of executing basic logic and mathematical functions on the DSM module. Commands use upper case characters only and are case sensitive.

| | | | |
|---|---|---|---|
| ABS | TRUE | -(minus) | <= (less than or equal to.) |
| BWAND | FALSE | / (divide) | <> (not equal) |
| BWOR | IF | * (multiply) | |
| BWXOR | THEN | :=(assign) | |
| BWNOT | END_IF | >(greater than) | |
| ON | MOD | < (less than) | |
| OFF | + plus | >= (greater than or equal to.) | |

**Want to know more?** In the Help Index, choose "Local Logic: an Overview".

# CAM EDITOR




**Project Tab: Navigator**

**CAM Blocks**

The CAM editor is an accessory for Logic Developer - PLC motion programming that provides a means to create, edit, and manage electronic CAM profiles. Each CAM profile is a curve that specifies the response of a slave servo to a master position index. CAM profiles are referenced by name in the associated motion program and grouped into CAM blocks. Each block is intended for download to a specific motion module by means of a PLC controller. The hardware components are specified in the HWC for the associated project.

CAM profiles are user-defined in a library and then grouped into blocks by aliasing them. This allows you to reuse CAM profiles by including them in multiple CAM blocks.

### To create a CAM block

1. In the Project tab of the Navigator, expand the Motion Program folder.
2. Right-click the CAM blocks folder and choose **New**.

   A new CAM block with a default name is created.
3. Rename the block as desired.

### To import CAM blocks

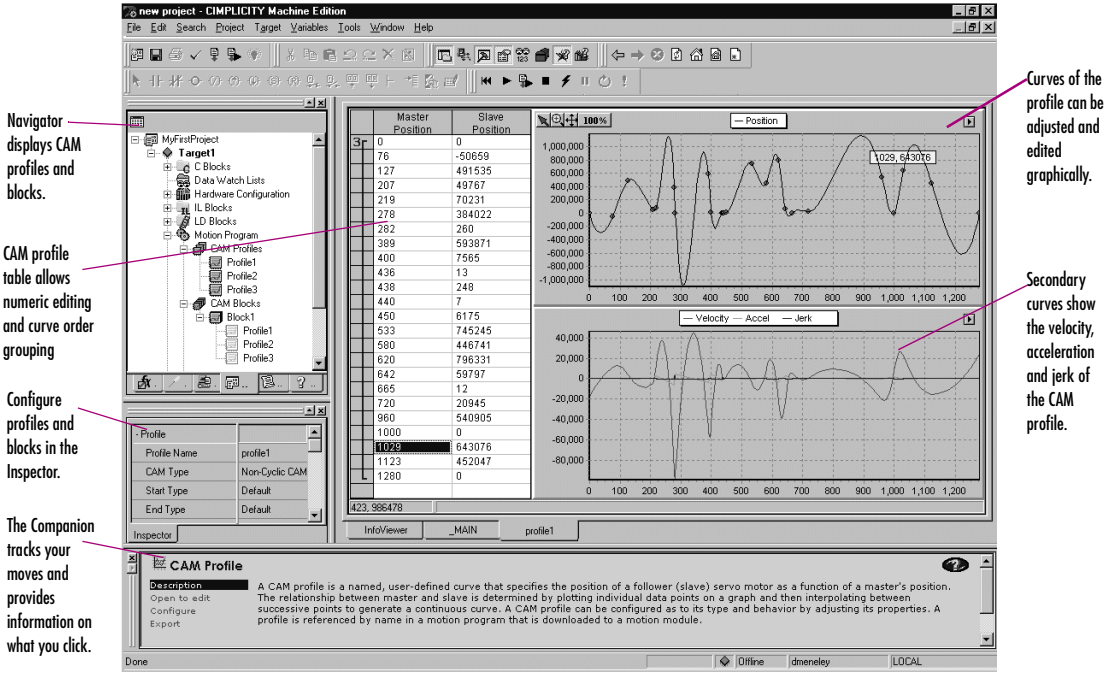
1. In the Project tab of the Navigator, expand the Motion Program folder.
2. Right-click the CAM blocks folder and choose Import from File.

   The Open dialog box appears.
3. Browse to the CAM block you want to import (.csv or .txt file).
4. Click Open.

   The imported block appears in your project.

## Working with the CAM Editor

You can adjust the curves of your CAM profile to suit the specific needs of your project. With the CAM editor you create profiles by defining points on a master/slave position curve. Groups of adjoining points are allocated to sectors. Each sector is assigned a polynomial curve fit order (1,2,3) that specifies how the curve will be interpolated between points.




### To create a CAM profile

- In the Project tab of the Navigator, right-click the CAM Profiles folder and choose **New**.

  A new CAM profile with a default name is added to your project.

### To configure a CAM profile

1. In the Project tab of the Navigator, right-click a CAM profile and choose Properties.

   The properties of the CAM profile appear in the Inspector

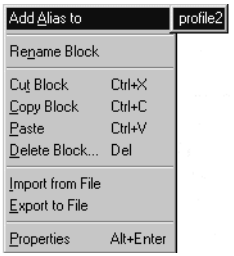
2. Adjust the properties of the CAM profile in the Inspector to specify its type and boundary conditions.

### To edit a CAM profile

1. In the Project tab of the Navigator, expand the Motion Program folder and double-click a CAM profile.

   A graphical representation of your profile appears in the profile editor and a numeric representation appears in the profile table.
2. Insert and move points in the profile editor or table.
3. Group points into sectors in the profile table and assign curve fit order to each sector.

### To add an alias to a CAM block




- In the Project tab of the Navigator, right-click a CAM block, point to **Add Alias to**, and then choose a profile.

  **Note:** Aliases correspond to CAM profiles within CAM blocks. In order to create aliases for CAM blocks, you must have previously created a CAM profile.

  **Want to know more?** In the Help Index and choose "CAM".

# Index