

# **View™ and PC Control PAC Machine Edition Getting Started**

**Version 9.80**

# Contents

<b>Chapter 1: Welcome .....</b>	<b>1</b>
1.1 SYSTEM REQUIREMENTS.....	2
1.2 INSTALLATION.....	4
1.3 PRODUCT AUTHORIZATION .....	5
1.3.1 Automatic License Activation .....	5
<b>Chapter 2: PAC Machine Edition .....</b>	<b>7</b>
2.1 QUICK START.....	8
2.2 MACHINE EDITION ENVIRONMENT .....	8
2.3 GETTING TO KNOW MACHINE EDITION .....	9
2.3.1 Right - click, right - click, right – click .....	9
2.3.2 Getting Help .....	9
2.3.3 Accessing the Right Tool .....	11
2.3.4 Using docking markers.....	11
2.3.5 Projects and the Navigator .....	12
2.3.6 Properties and the Inspector .....	13
2.3.7 Data Watch Lists .....	14
2.3.8 Smart Lists .....	16
2.3.9 The Toolchest .....	16
2.3.10The Feedback Zone .....	17
2.3.11Managing Variables.....	18
2.4 MACHINE EDITION PROJECTS .....	20
2.4.1 Sharing Projects between Machine Edition Workstations.....	20
2.4.2 Running a Sample Project (View /Logic Developer - PC).....	20
2.4.3 Developing a Machine Edition Project.....	22
2.4.4 Validating and Downloading a Project .....	25
2.5 SFC EDITOR .....	29
2.5.1 Sequential Function Chart .....	29
2.5.2 Working with the SFC editor - Offline.....	30
2.5.3 Working with the SFC editor – Online .....	31
2.6 LADDER EDITOR .....	31
2.6.1 Ladder Program .....	32
2.6.2 Working with the ladder editor – Offline.....	33
2.6.3 Working with the ladder editor - Online.....	33

2.7	INSTRUCTION LIST EDITOR .....	34
2.7.1	Instruction List .....	34
2.7.2	Working with the IL editor – Offline .....	35
2.7.3	Working with the IL Editor – Online .....	36
2.8	STRUCTURED TEXT EDITOR .....	36
2.8.1	Structured Text .....	36
2.8.2	Working with the ST Editor - Offline.....	37
2.8.3	Working with the ST editor - Online.....	38
2.9	FUNCTION BLOCK DIAGRAM EDITOR.....	38
2.9.1	Function Block Diagram .....	38
2.9.2	Working with the FBD editor - Offline .....	39
2.9.3	Working with the FBD editor – Online.....	40
2.10	LOGIC DEVELOPER - PC WEB ACCESS .....	41
2.11	CONTROL I/O DRIVERS .....	43
2.11.1	I/O Drivers.....	43
2.11.2	Control I/O Tool .....	44
2.11.3	Working with Control I/O .....	45
2.12	CONTROLLER .....	45
2.12.1	Working with the Controller .....	47
2.12.2	Warm Standby .....	48
2.12.3	Working with Warm Standby .....	48
2.12.4	Hot Standby .....	48
2.12.5	Working with Hot Standby .....	49
2.12.6	OPC Servers, Warm Standby, and Hot Standby .....	50

## **Chapter 3: View™ ..... 51**

3.1	PANEL EDITOR.....	52
3.1.1	Working with the Panel Editor .....	54
3.1.2	Message Displays .....	55
3.1.3	Multi-Touch Gestures Support on QuickPanel+.....	56
3.2	SCRIPT EDITOR .....	57
3.2.1	Scripts.....	57
3.2.2	Scripting Languages .....	58
3.2.3	VBScript Syntax.....	59
3.2.4	Active Scripting Objects .....	59
3.2.5	Working with the Script Editor.....	60
3.3	GRID EDITORS .....	60

3.4	ALARMS.....	61
3.4.1	Alarm Groups .....	61
3.4.2	Alarm Display Objects .....	62
3.5	LOGGING DATA .....	63
3.6	PLC ACCESS I/O .....	64
3.6.1	Drivers .....	64
3.7	OPC.....	66
3.7.1	OPC Client.....	67
3.7.2	Machine Edition OPC Servers .....	67
3.8	VIEW WEB ACCESS .....	68
3.9	LANGUAGES FOLDER .....	70
3.9.1	Working with the Languages editor .....	70
	HTML Dashboard .....	72
	SNMP Variables .....	73
	UDT 74	
3.10	VIEW RUNTIME .....	75
3.10.1	Networking .....	75
3.11	QUICKPANEL APPLICATIONS.....	76
3.11.1	External Keypad Assignment .....	77
3.11.2	Scripts on QuickPanel Targets .....	77
3.11.3	QPScript Language .....	78
	General Contact Information .....	79
	Technical Support.....	79

# Chapter 1: Welcome

Congratulations for purchasing a PAC Machine Edition 9.80 product. This package provides all the tools necessary to create powerful control and HMI (human machine interface) applications for a variety of run-time targets; a truly scalable solution. Machine Edition supports the following products and features:

- A fully integrated environment. Every tool and editor work with the others.
- Logic Developer - PLC™, a software tool for programming the full line of Emerson IP Controllers.
- (See GFK-1918 Getting Started Logic Developer - PLC.)
- Logic Developer - PC™, scalable, computer-based control software with a complete set of IEC standard editors. Available with
  - Windows 10
  - QuickPanel+
  - QuickPanel Control
- View, for creating HMIs on:
  - Windows 10
  - QuickPanel+
  - QuickPanel View
  - QuickPanel
- Web Access features to access real-time HMI data any time, from anywhere.

# 1.1 SYSTEM REQUIREMENTS


To use Machine Edition and its tools, you require the following:

## Development Environment Operating System

Machine Edition Product	Windows 10 †
Logic Developer – PC	Yes
QP+ View	Yes
Logic Developer – PLC	Yes

- a. 32-bit or 64-bit variations of Windows are supported (64-Bit recommended).

---

 **Want to know more?** In the Help Index, look up “Windows 10”.

---

- b. Windows regional settings must be set to English

## Processor speed and memory

- Minimum (for small to medium-sized projects):
  - Intel Core i5 with 4 GB RAM
  - AMD FX or Phenom II X6 with 4 GB RAM
- Strongly recommended for large projects to see multi-threading performance gains:
  - Intel quad-core Core i7 with 8 GB RAM and 64-bit Windows
  - AMD higher FX or Phenom II X6 models with 8 GB RAM and 64-bit Windows 10

## Extra requirements when executing multiple instances of Machine Edition

- Operating system
  - Minimum requirement for very large projects and strongly recommended for any project: 64-bit variations

## Other requirements

- Internet Explorer 11.0 or newer with current updates.

---

**Note:** *Install Internet Explorer before installing Machine Edition.*

---

- TCP/IP Network protocol (if you use an Ethernet connection).
- .NET Framework 4.6 Full. If the Microsoft .NET Framework is not yet installed, it is automatically included during the installation of Machine Edition, and a reboot may be required to complete the installation.
- 2 GB of hard disk space for all the Machine Edition products and sample projects. Additional space is required for your projects and temporary files.
- The Development Environment is supported on VMWare Player or VMWare Workstation.

---

### **Note:**

a. *When using multiple instances, each instance will consume the resources required for a single instance. Running too many instances may cause your system to become unstable. You can run more instances with small projects than with medium or large projects. In general, more than 5 instances of small projects, 3 instances of medium projects or 2 instances of a large project will likely impact performance on most systems.*

b. *A large project is one that exceeds 125,000 variables, 128 blocks, 10 targets, or combinations of each that would exceed these values.*

---

## Windows PC Runtime

- When you install only the Runtime for Logic Developer - PC and View Developer, the run-time executables are automatically updated from the development computer when the project is downloaded.

---

**Note:** *Windows regional settings must be set to English.*

---

- 233 MHz Pentium-based workstation (1.6 GHz recommended)
- 200 MB free hard disk space.
- 128 MB RAM (1 GB recommended).
- Minimum requirements for the range of variables defined in projects:

Requirements	Variables		
	<500	500 - 10000	>10000
CPU	233 MHz	1 GHz	2 GHz
Disk Space	200 MB	200 MB	200 MB
RAM	128 MB	512 MB	1 GB

#### QuickPanel+ or QuickPanel View/Control Runtime

- For HMI and Logic components: QuickPanel+, QuickPanel Control.
- For HMI only: QuickPanel+, QuickPanel View.

## 1.2 INSTALLATION

For last-minute information, release notes, and supported hardware lists for Machine Edition products, see the Important Product Information (IPI) document on the DVD.

There are several ways to view this document:

- When installing Machine Edition, select Important Product Information on the initial Launcher screen.
- On the Machine Edition Help menu, choose Important Product Information.
- When running Machine Edition, click the InfoView tab in the Navigator, then double-click the Important Product Information page under Getting Started in the Table of Contents.
- When running Machine Edition, click the Home button on the InfoViewer toolbar, and click the What's New link under Get Started on the left-hand side.

If you have any problems installing Machine Edition, please contact Technical Support (see page [Error! Bookmark not defined.](#)).

#### To install the full Machine Edition development environment from a DVD

1. Insert the Machine Edition DVD into your computer.  
Windows automatically starts the setup program. If the setup program does not automatically start, run Setup.exe in the root directory of the DVD.

2. Click Install to start the install process.
3. Follow the instructions that appear.

If a previous version of Machine Edition is installed on your workstation, you will be prompted to uninstall Machine Edition during the upgrade process. You should do so only when asked to by the installation process. Do not delete files left behind during uninstallation; these will be used by the new version.

#### To install only the View and Logic Developer - PC Run-times from a DVD

1. Insert the Machine Edition DVD into your computer.  
Windows automatically starts the setup program. If the setup program does not automatically start, run Setup.exe in the root directory of the DVD.
2. Click Install to start the install process.
3. Follow the instructions that appear until you reach the PAC Machine Edition Setup page.



4. On that page, expand the View node. Under that node, select Run-time and deselect everything else. Expand the Logic Developer - PC node. Here too, select Run-time and deselect the rest. At the bottom of the page, click Next.
5. Follow the instructions that appear.

## 1.3 PRODUCT AUTHORIZATION

A new installation of Logic Developer - PLC provides a 4-day trial license with full access to Logic Developer - PLC features. This license overrides all other licensing and cannot be removed. Any licensing added is apparent on the fifth day.

### 1.3.1 Automatic License Activation

Customers can license their copy of Logic Developer – PLC using Emerson’s Entitlement and License Manager software. There are two methods to initiate the activation wizard: from within the application (PME) or from the Entitlement and License Manager itself.

During or following the trial period of PAC Machine Edition, customers can select the *Activate a License* button to initialize the License Activation screen from the Entitlement and License Manager and follow the activation wizard’s prompts. This section will demonstrate how to license PME from within the application.

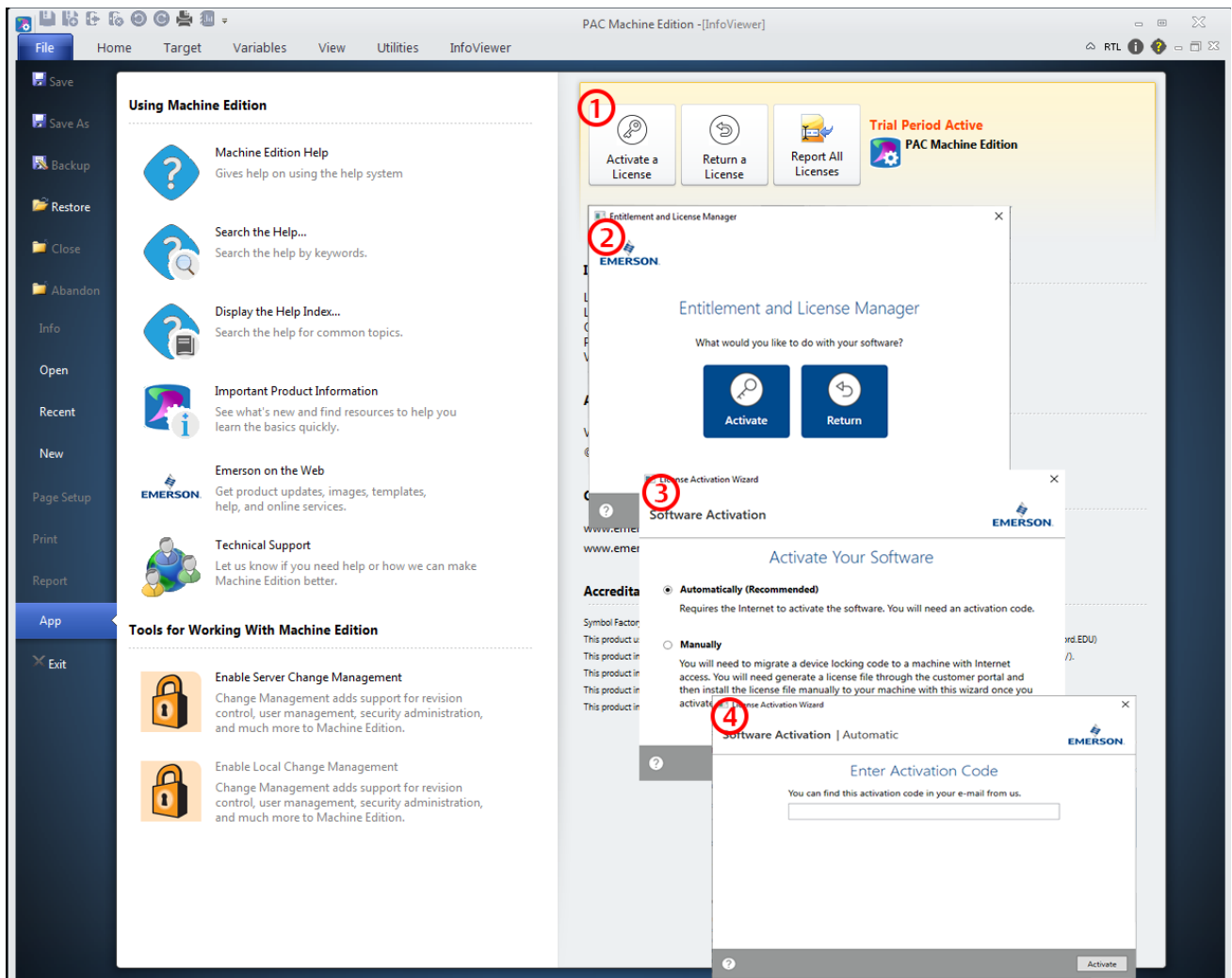
Alternatively, customers can open Entitlement and License Manager directly and activate their copy of PAC Machine Edition. For more information on using Entitlement and License Manager, please refer to GFK-3104.

**Automatic Activation Steps:**

- 1) Navigate to the *File* menu on the ribbon bar.
- 2) Select the *App* tab from the menu.
- 3) Click the *Activate a License* button located on the upper-right side of the menu (Figure 1, Item 1). A new window will open to the Entitlement and License Manager.
- 4) Click the *Activate* button and select the Activation Method (Figure 1, Item 2).
- 5) Select *Automatically* to automatically license your product (Figure 1, Item 3).
- 6) Enter the activation code on the next screen (Figure 1, Item 4)

**Note:** For more information on the manual activation of your software, please refer to GFK-3104.

**Figure 1: Activate a License Button**



## Chapter 2: PAC Machine Edition

PAC Machine Edition offers you a complete solution for the development of automation applications under one roof. With Machine Edition's integrated development environment and tools, you will spend more time building applications and less time learning the software.

Machine Edition products are fully integrated with the environment and with each other:

- They share the same project database. No more wasted time synchronizing data points between applications!
- They share the same set of tools, providing a consistent user interface throughout the development process.
- They feature full drag-and-drop capabilities between tools and editors.
- They feature a true scalable solution. You can choose what type of machine your projects are downloaded to and will run on.


The first part of this chapter provides an overview of the Machine Edition environment. The second part provides some key Machine Edition concepts you need to know before you begin. The third part shows how to open and explore some of the sample projects included with a new Machine Edition installation. When you have finished, you will have a solid foundation for building your own automation projects.

- For more help getting started, in the InfoView tab of the Navigator, expand the Getting Started library.

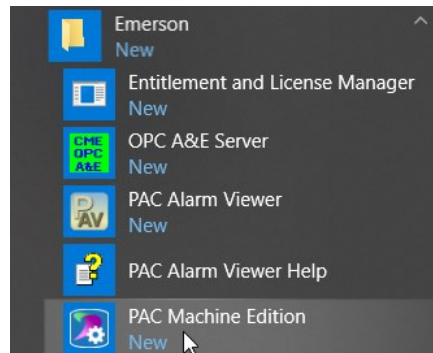
## 2.1 QUICK START

Machine Edition makes it easy to develop a project.

### To start Machine Edition

1. On the Windows toolbar, click , point to Emerson and then click PAC Machine Edition.

**Figure 2: Starting Machine Edition**



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

#### Note:

- If you select either Empty project or Machine Edition template, the New Project dialog box appears, in which you can create a new project.
- If you select Open an existing project, you can choose from Recent Projects (the default) or All Projects. You may need to also specify the location of the project: whether it is located on the local computer (My Computer), in a shared folder (Shared Projects), or on Change Management Server (Server).

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

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

4. If desired, select the do not show this dialog box on startup option.
5. Click OK.

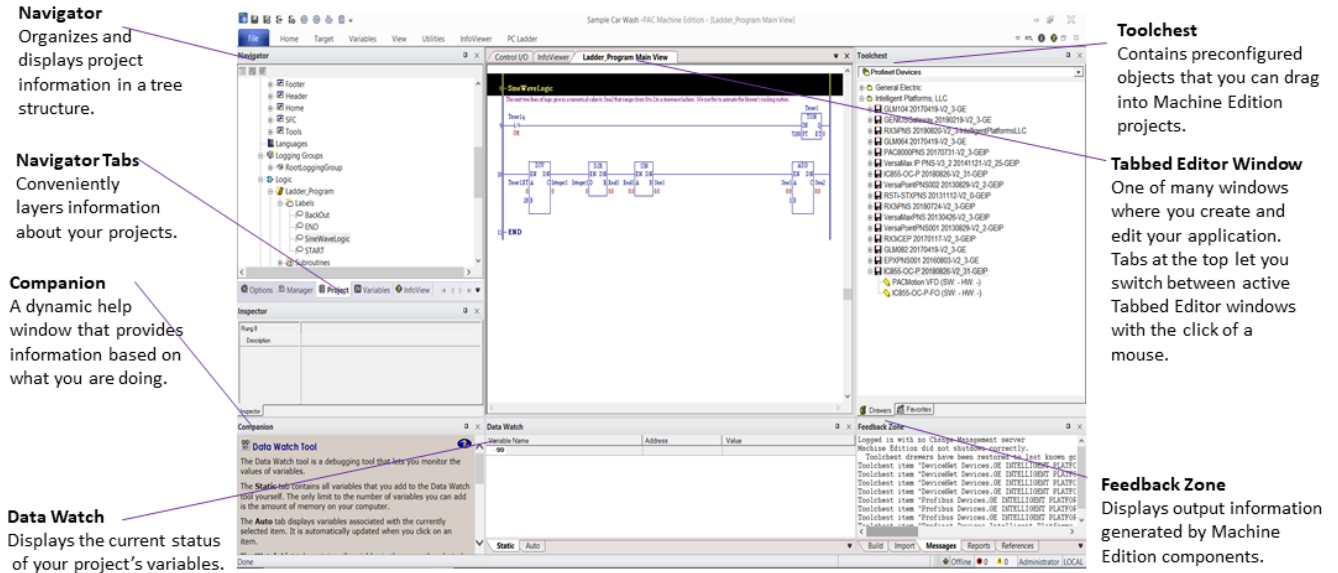
## 2.2 MACHINE EDITION ENVIRONMENT

All Machine Edition tools and editors appear in the Machine Edition window. The run-time programs are separate Windows applications, so you can run a finished application without starting Machine Edition.

The following illustration shows a possible layout of the tools and a couple of the editors available to you. Most of the time, you will be using only a few of these at once—you can open and close tools and editors as you need them. Many tools are also available only

when editing a project. We will look at some of the tools in the following section; for more details on the others, see the online Help.

**Figure 3: Machine Edition Environment**



## 2.3 GETTING TO KNOW MACHINE EDITION

The following are some key features of the Machine Edition environment. Knowing them will make your first few hours with Machine Edition a breeze.

### 2.3.1 Right - click, right - click, right – click

No matter what object appears on your screen while using Machine Edition, you can right-click it to perform operations on it. In fact, this is probably the most common way you will get things done. Machine Edition tailors the menu of commands depending on the current status of your project.

### 2.3.2 Getting Help

There are many ways to access the Machine Edition online help system.

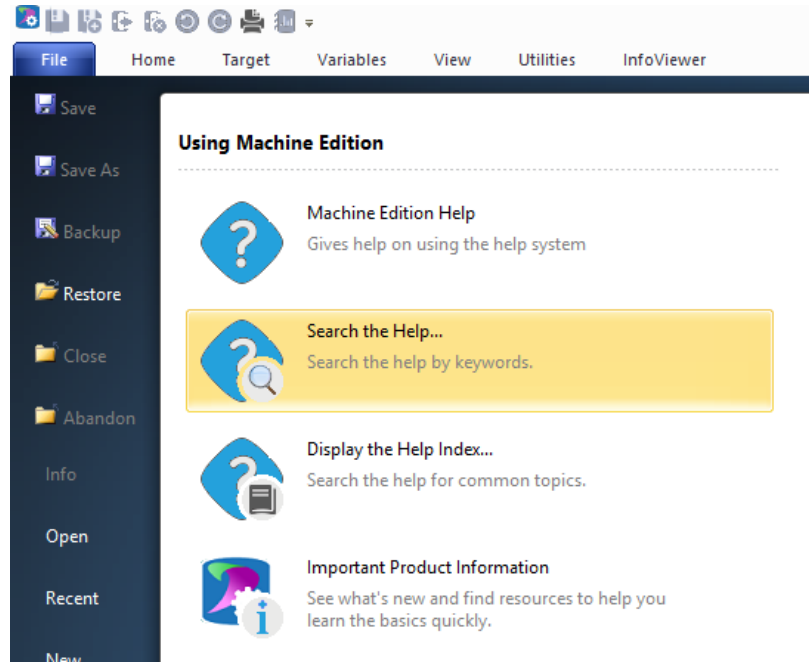
- Press F1 on any selected item for context-sensitive help.
- Browse through the table of contents in the InfoView tab of the Navigator.
- Perform a full-text search of the help in a separate HTML Viewer window: on the Help menu, choose Search.
- Search for keywords using the index: on the Help menu, choose Index.
- Use the Companion to dynamically display a brief description about whatever item you have selected.

You can also access additional help on the web. In Machine Edition on the Help menu, point to Emerson on the Web, and then choose:

- Emerson Support
- Emerson Home Page

Machine Edition provides two kinds of Help windows to display help information: the Companion and the InfoViewer.

Figure 4: Getting Help



**Companion**

The Companion window displays brief information about whatever item you are working with, anywhere in Machine Edition. Whether you need the meaning of a property, the purpose of a node in the Navigator, or an explanation of errors generated during validation, the Companion is always there for you.

By default, the Companion is open while editing projects. If it has been turned off, click on the Tools toolbar to open it.

**InfoViewer**

The InfoViewer window is an embedded Web browser that provides more detailed and procedural help. It is launched whenever you access help topics from the InfoView Tab, the help index, or by pressing F1 on a selected item.

---

**? Want to know more?** In the Help Index, look up “Help” and choose “Finding information in the Help”.

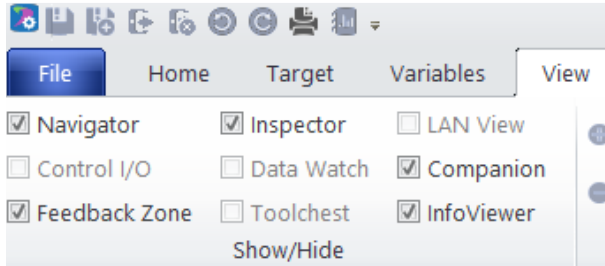
---

### 2.3.3 Accessing the Right Tool


Often, Machine Edition Help will direct you to a specific tool window. If the tool isn't visible, there are two ways to open it:

- On the Tools menu, choose the tool name.
- On the Tools toolbar, click the tool icon. To display the Tools toolbar, on the Tools menu, choose Toolbars and select a menu item.

**Figure 5: Tools Toolbar**



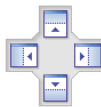
If you are not sure of the name of a tool within a toolbar, hover the mouse pointer over the tool's button to display its name in a tool tip.

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

### 2.3.4 Using docking markers


If the Show Docking Markers option is enabled, you can use docking markers to help dock a tool window to a desired location.

As you drag a dockable window, a series of docking markers appear, indicating valid docking locations. Docking markers appear as a series of large blue arrows. Move the mouse over a docking marker to preview how the window would be docked when using that marker. Release the mouse button while hovering over a marker to use that location.



These markers dock tools to the edge of an existing tool or to the Machine

Edition environment window.

 This marker “stacks” the tool overtop an existing tool window, forming a series of tab-docked windows. You can then switch between tab-docked tools by clicking the tabs that appear at the bottom of the tab-docked windows.

## 2.3.5 Projects and the Navigator

**Navigator** [Pin] [Close] The Navigator window organizes and manages your projects.

- Use the Navigator to create and manage projects, add targets and components to your project, set your environment preferences, create scripts, open editors, create variables, and more.
- The Navigator is organized into several tabs. The available tabs depend on which Machine Edition products are installed and whether a Machine Edition project is open. For example, the Project and Variables tabs appear only when a Machine Edition project is open.
- Within each tab, items are displayed in a tree of nodes or folders. You can expand and collapse the tree, just like folders in Windows Explorer.

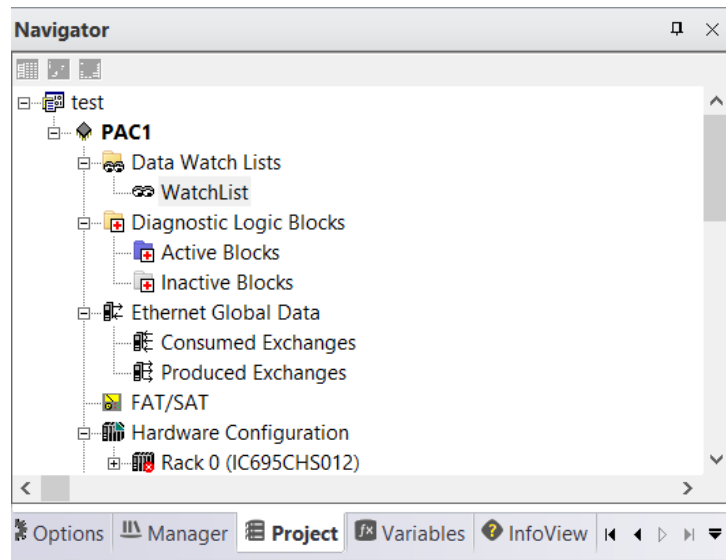
The following picture illustrates the Navigator prior to any projects being opened. All the files listed under My Computer are projects that you can access on your computer. The Samples folder contains sample projects and tutorials.

---

**Want to know more?** In the Help Index, look up “Navigator” and choose “Navigator Tool Window”.

---

**Figure 6: Navigator Window with Local Project List**



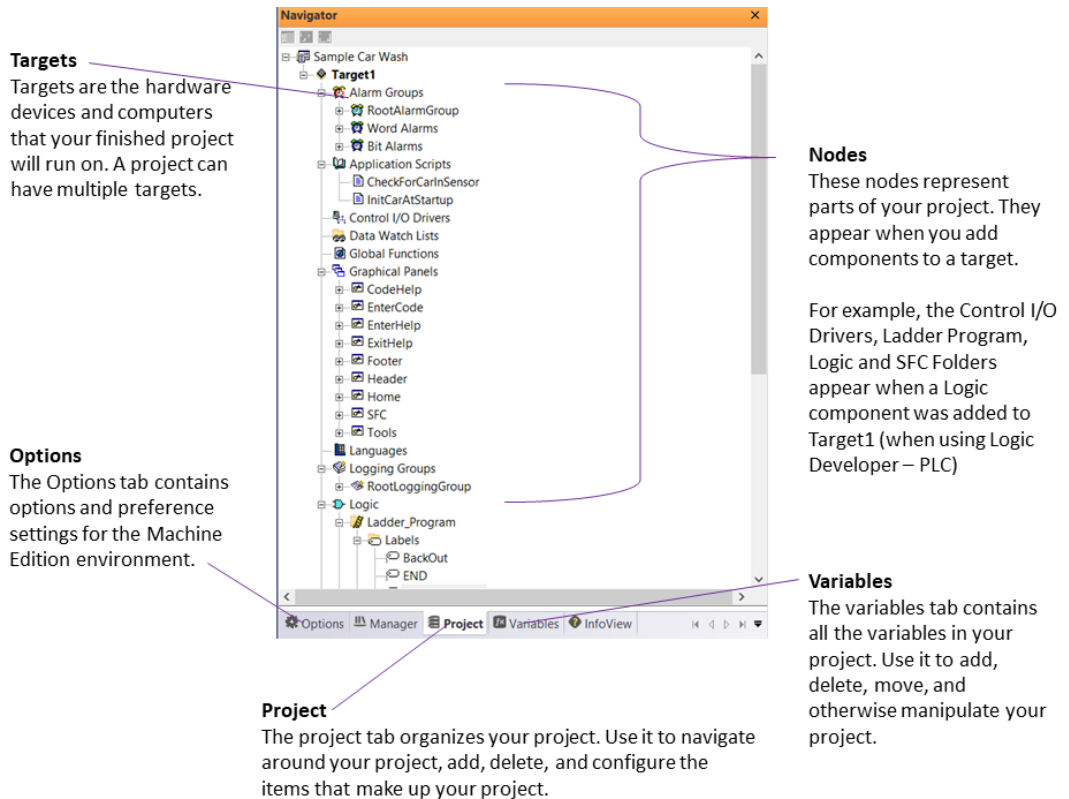


### To open a project in the Navigator

In the Manager tab of the Navigator, right-click an existing project listed under My Computer and choose Open.

The project is opened in Machine Edition. The Variables and Project tabs are added to the Navigator and the nodes change to reflect those components that make up the project you just opened.

**Figure 7: Navigator Window with an Open Project**



## 2.3.6

### Properties and the Inspector

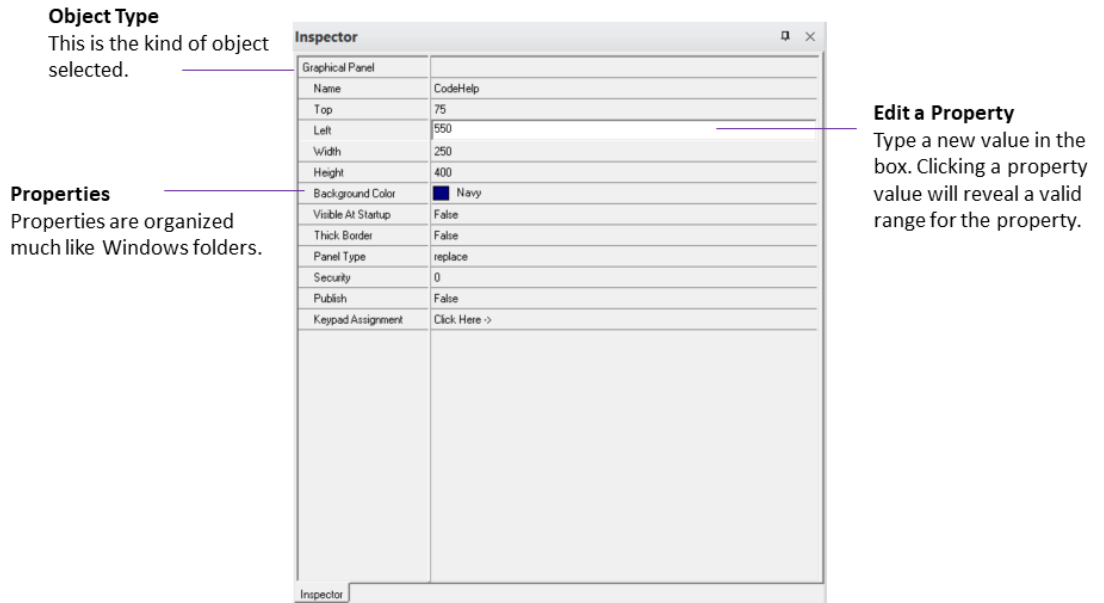
#### Inspector

In Machine Edition, practically every object has properties. Properties are attributes and information about that object. For example, the properties of a box on an HMI graphical panel include Height, Width, Fill Color, and Outline Color.


The properties of an object are edited in the Inspector window:

- To edit an object's properties, select it with the Inspector open. The Inspector displays all those properties associated with the selected object.
- When a property changes an object's appearance, you will see the results of the change immediately in the object's editor.
- To see more properties for an object, select a tab at the bottom of the Inspector.

Figure 8: Inspector Window



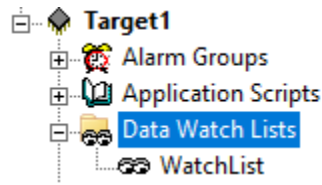
As you select the various properties within the Inspector, the Companion displays a brief description of the selected property.

 **Want to know more?** In the Help Index, look up “Inspector” and choose “Inspector Tool Window”.

## 2.3.7 Data Watch Lists

The Data Watch (available only while editing a project) is a debugging tool that you can use to monitor and edit real-time values of variables defined in your project. This is useful while working online with a run-time target. With the Data Watch tool, you can monitor individual variables or user-defined watch lists of variables. You can change variable values and force the state of discrete (BOOL) variables. Watch lists can be imported, exported, or saved with a project.

Figure 9: Navigator Project tab Data Watch nodes



There are three tabs in the Data Watch tool:

- The Static tab shows variables added to the Data Watch tool.
- The Auto tab contains variables selected in the variable list or associated with the currently-selected instruction in ladder logic.
- The Watch List tab contains all variables in the currently selected watch list. A watch list lets you create and save a separate list of variables to monitor. You can define more than one watch list, but only one watch list can be monitored in the Data Watch tool at a time.

**Want to know more?** In the Help Index, choose “Data Watch List”.

Figure 10: Data Watch Window

The structure variable **Main.Param.ParameterizedBlock** has been expanded to show all of its elements.

The **Static** tab contains variables that you add to the Data Watch tool.

The **Auto** tab displays variables associated with the item currently selected in an editor or tool.

Variable Name	Address	Value
BOOL	%AQ0001.X[00]	Off
BYTE	%R00001	0
MAIN.Param.ParameterizedBlock		(Custom Structure)
BOOL		Off
INT		3
REAL		2.0
DWORD		1
STRING		String

The IEC reference address of variables mapped to I/O terminals or controller memory.

The name and value of each element of the structure variable is shown here.

### To define a Watch List:

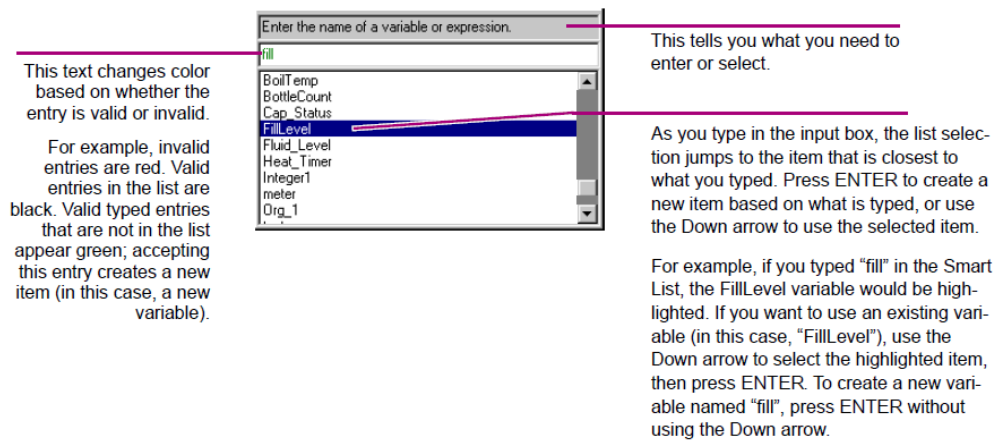
1. Right-click the Data Watch Lists folder and choose New.
2. Double-click the new Watch List to open it in the Data Watch tool.
3. Add variables to the Data Watch as desired.

The changes to the watch list are automatically saved for later use.

## 2.3.8 Smart Lists

While working in Machine Edition editors, you are frequently required to input variable names, instruction mnemonics, and other data items. The Smart List can speed text entry of these items.

**Figure 11: Smart List Display**



## 2.3.9 The Toolchest

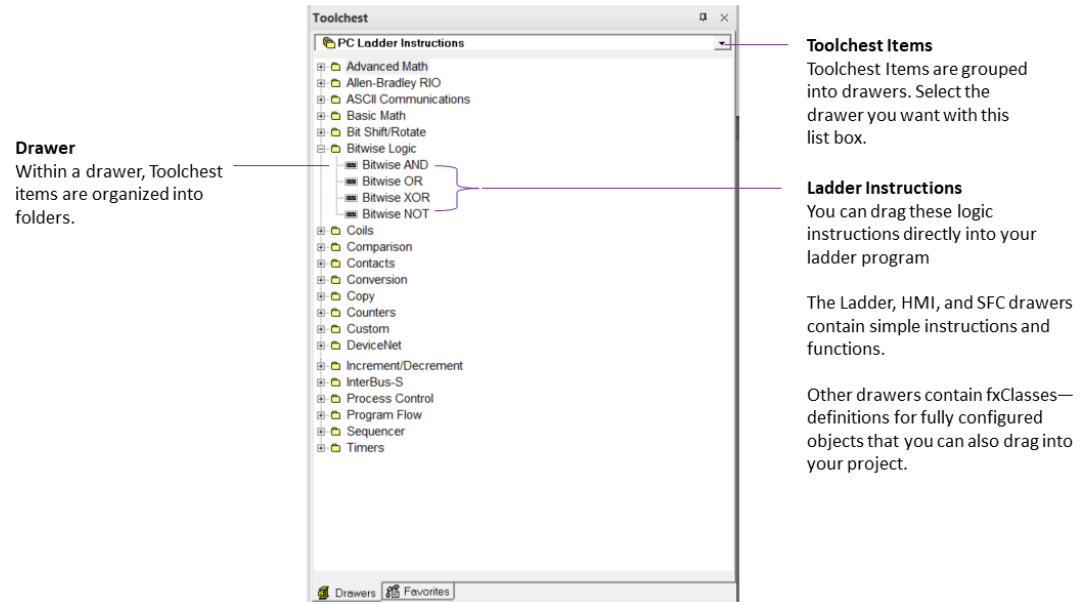
### Toolchest

The Toolchest (available only while editing a project) is a repository of preconfigured object templates you can drag into your project. These objects can be as simple as a single ladder logic instruction, or as complex as a robotic arm with fully-configured ladder logic and HMI animation.

While Machine Edition comes with a set of preconfigured object templates, you can create your own. Because you can add as many copies of them as you want, this can save you hours of development time.

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

Figure 12: Toolchest Window



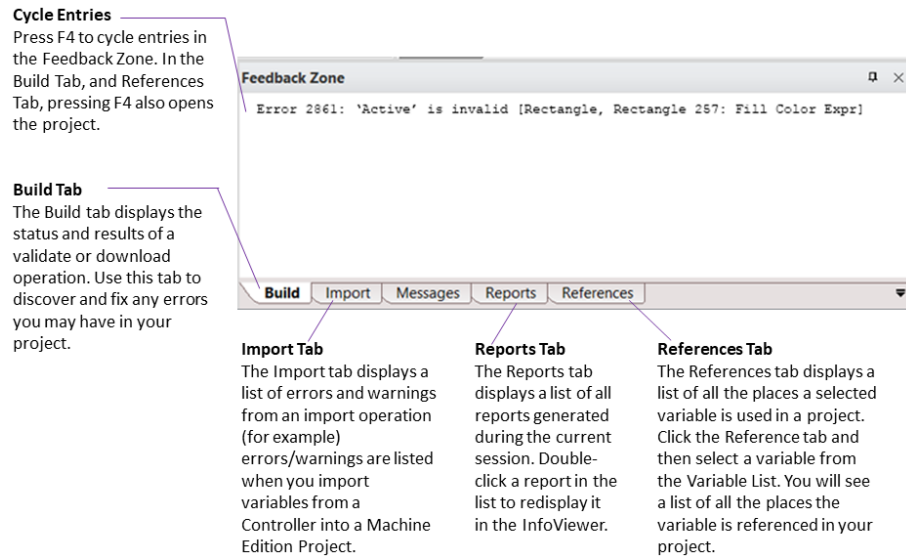
Definitions of most preconfigured objects within the Toolchest are displayed in the Companion.

### 2.3.10 The Feedback Zone

#### Feedback Zone

The Feedback Zone (available only while editing a project) is an interactive window that displays output information generated by Machine Edition components. With the Feedback Zone, you can keep track of project information, locate validity errors within your project, display generated reports, and more. It is organized into several tabs. Click a message to display more information about the error in the Companion Window.

**?** **Want to know more?** In the Help Index, look up “Feedback” and choose “Feedback Zone Tool Window”.

**Figure 13: Feedback Zone**

## 2.3.11 Managing Variables


Variables are named storage spaces for data values defined in Machine Edition projects. A variable could store the current velocity of a motion Controller motor, the height of a robotic arm, or any other value that an application needs to keep track of. Most variables in a project can be shared among various components and targets, such as View panels and Logic Developer - PC's ladder logic.

You manage variables in the Variables tab of the Navigator, also named the Variable List. Like most items in Machine Edition, you configure variables by editing their properties in the Inspector.

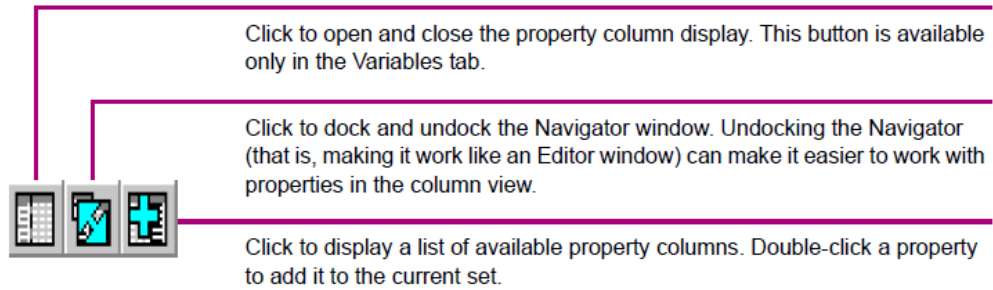
The values a variable can store depends on its data type. For example, a DINT data type indicates that the variable can store “Double Integers”, 32-bit values. The location where a variable’s value is stored is indicated by its data source. Typically, a variable’s value is either stored internally in the target’s memory or is retrieved (and sent) to external Controller hardware, via an I/O terminal or other connection. Available data types and data sources depend on the target type and (if applicable) components added to the target.

You can also use arrays and structure data types in Machine Edition projects. An array is a series of variable elements with identical data types, referenced by a 0-based index (as in “MyArray[3]”). A structure data type is a group of variable elements that may or may not have the same data type, referenced by the name of each element (as in “MyStructure.MyElement”). Custom structure data types are created with fxClasses in the Toolchest.

---

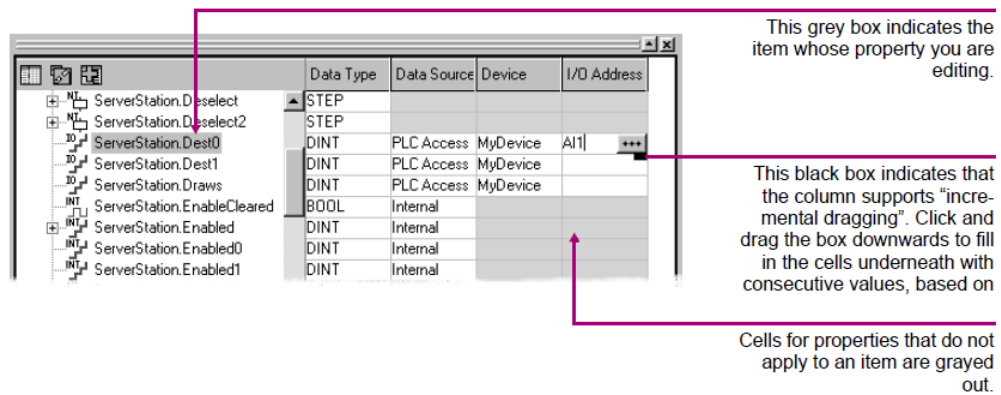
 **Want to know more?** In the Help Index, look up “STRUCTURE data types”.

---



When opened, the Navigator's property columns view consists of a grid of cells:

**Figure 14: The Navigator's Property Columns view**



## Variable Reports

While you can generate reports on many things in Machine Edition, they are especially useful for the Variable List. Among the available reports are:

- All variables by name, filtered or unfiltered.
- Cross-references to variables (that is, the places they are used in the project).
- All forced variables (Logic Developer - PC and Logic Developer - PLC only).
- Unused variables (that is, variables with no references anywhere in the project).  
The unused variables report also has a link that lets you delete all unused variables from the project at once.

Reports are displayed as HTML pages in the InfoViewer. To generate a report, click anywhere in the Variables tab then, on the File menu, select Report.

Want to know more about Variables? Look up the following in the Help Index:

- For information on variables in general, choose "Variables: an Overview".
- For information on variables in View, choose "Variable support in View Developer".
- For information on variables in QuickPanel projects, choose "Variable support in QuickPanel applications".
- For information on variables in Logic Developer - PC, choose "Variable support in Logic Developer - PC".


- For information on variables in Logic Developer - PLC, choose “Variables in Logic Developer - PLC vs. Memory in Emerson IP Controllers: an Overview”.
  - For information on importing variables from other applications or database files, choose one of the “Importing Variables...” topics.
- 

## 2.4 MACHINE EDITION PROJECTS

During development, your automation application is named a project. Each Machine Edition project is made up of targets and (sometimes) components. A target represents the hardware platform where the finished project runs, such as a PACSystems Controller, a Windows PC computer, or a QuickPanel+ unit. Different Machine Edition products support different target types. Targets are often further subdivided into models. Components add specific capabilities to a target. Available components depend on the target type, model, and what Machine Edition products you have installed.

For example, with View, you can add an HMI component to Windows PC, QuickPanel+, and QuickPanel View/Control targets, used for creating human-machine interfaces. With Logic Developer - PC, you can add a Logic component to Windows PC, QuickPanel+ and QuickPanel Control targets; you can then create ladder logic to make a PC act as a Controller. QuickPanel View targets do not support logic; therefore, you cannot download a project that contains logic to one.

---

 **Want to know more?** In the Help Index, look up “projects” and choose “Machine Edition Projects: an Overview”.

---

### 2.4.1 Sharing Projects between Machine Edition Workstations


If your site has multiple Machine Edition workstations connected through a network, you can use the Shared Projects folder to work on the same set of projects. Projects under the Shared Projects folder are stored in a directory you specify, typically a shared directory on the network.

No access or version control is applied to the Shared Projects folder. If multiple users open the same project at the same time, some changes may be lost.

---

**Note:** *Shared Projects and Shared Variables are two completely different and separate features. For information on Shared Variables (which implement proxy variables between targets in different projects), see “PLC Access I/O” on page -64.*

---

 **Want to know more?** In the Help Index, look up “Shared Projects”.

---

### 2.4.2 Running a Sample Project (View /Logic Developer - PC)

If you are using View or Logic Developer - PC, we’ve provided some sample projects that show basic operations of Machine Edition. Take a few minutes to complete the following steps and learn the basics of project development in Machine Edition.

- For a more detailed example of creating projects, in the Help Index, look up “Tutorial” and choose one of the topics listed there.

**To run a sample project**



1. Run Machine Edition from the Start menu | Emerson | PAC Machine Edition | PAC Machine Edition.
2. In the Manager tab of the Navigator, double-click one of the sample applications listed under the My Computer folder.

Sample applications include the following:

- **Alarm Trends:** an HMI application that demonstrates alarms and charts.
- **Animation Features:** an HMI application that depicts the various types of animation available in View.
- **Brewery:** an HMI and logic brewery application that runs on a Windows PC target.
- **Car Wash:** an HMI and logic application that runs on a Windows PC target. This application is built in an SFC document.
- **ControlStation HMI Features:** an HMI and logic application that runs on a Windows PC target. This application also shows some web documents.
- **QP Control 6 inch Brewery:** an HMI and logic brewery application that runs on a QuickPanel Control unit. (Projects containing a logic component cannot be downloaded to a QuickPanel+ or QuickPanel View unit, because such units do not support logic.)
- **QP Control 6 inch Ventilation:** an HMI and logic application that runs on a QuickPanel Control unit. This application runs a four-zone ventilation logic system. It is to be used with the QuickPanel Control unit Tutorial. (Because QuickPanel+ and QuickPanel View units do not support logic, projects with a logic component cannot be downloaded to such units.)
- **ST - Lunar Lander:** an HMI and logic application that runs on a Windows PC target. This project demonstrates the Structured Text (ST) language and its interaction with User Defined Function Blocks (UDFBs).

Tutorials include

- **Animation:** an HMI application that illustrates and provides details on the various animations that you can use in your HMI.
  - **AppExec:** an HMI application that demonstrates the AppExec Script function.
  - **Keystrokes:** an HMI application that displays the analog values associated with the keys on your keyboard.
  - **Logging:** an HMI application that demonstrates how to log production data to an ASCII text file.
  - **Recipes:** an HMI application that shows how one would load and modify recipes.
  - **Scripting:** an HMI application that demonstrates some of View's scripting capabilities.
3. Browse through the project in Machine Edition.

Open the project's graphical panels, ladder logic, and SFCs (if they exist) in their respective editors. To open an editor, right-click the appropriate node in the Project tab of the Navigator and choose Open.

Also, take a look at the properties of the various nodes and objects. To view an object's properties, open the Inspector window, and then select an object.

4. In the newly opened project, press F9 to validate, download, and run the sample application.

The F9 key is a shortcut for the Run command which automatically starts View Runtime and the Controller.

The sample application should now be running in both View Runtime and the Controller. View Runtime appears, displaying the first panel of the project's HMI. You can see the real-time status of the logic by going online to the Controller.

### To go online to the Controller

If you have chosen a project that contains logic and/or an SFC document, you can go online to the application and watch the logic being executed.

1. Minimize the Runtime window (but do not close it) and return to Machine Edition.
2. In the Project tab of the Navigator, right-click the Target node and choose Go Online.

You are now online to the Controller; that is, you are working with the application while it is running. The Controller is the part of the Runtime that solves logic and SFCs.

3. In the Project tab of the Navigator, open the Ladder Editor by double-clicking the Ladder Program node or open the SFC Editor by double-clicking the SFC node.

In the editor, you can watch the logic being solved as the Controller operates. To start and stop Runtime and the Controller, right-click the target, point to Online Commands, and then click Start/Stop Runtimes. To go offline from the Controller, right-click the target, and then click Go Offline.

## 2.4.3 Developing a Machine Edition Project

The first thing to decide when you create a Machine Edition project is where the project will run after it is developed. That is, on what type of target hardware it will run.

For View and Logic Developer - PC, targets can be a Windows PC computer (either the one you are developing the project on, or a remote one that you connect to through a network), QuickPanel+ or a QuickPanel Control unit.

For View alone, your target can also be a PC, QuickPanel+ or QuickPanel View unit.

For Logic Developer - PLC, your target is a PACSystems, Series 90, or VersaMax Controller. You can also configure remote I/O targets that represent a variety of remote I/O adapters and their associated I/O modules. See GFK-1918 Getting Started Logic Developer - PLC.

After you have decided on the target, you need to determine which components your project will include: HMI (with View) and/or Logic (with Logic Developer - PC). Note that View Stations do not support Logic components. A project can have multiple targets of

different types with various components running on each target. In some cases, targets can be converted from one type to another. For example, you can convert a Windows PC target to and from a QuickPanel+ View target.

---

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


---

### To create and develop a project

The following procedure introduces the general steps involved in creating a project using a template and downloading a project to a target computer.

1. Start Machine Edition from the Start menu | Emerson | PAC Machine Edition | PAC Machine Edition.

2. Create a project using a template.

In the Manager tab of the Navigator, right-click  My Computer and choose New. The New Project dialog box appears.

If you are using shared projects, you can also add new projects under the Shared Projects folder. Or, if you are working on a Change Management system, you can add new projects to the Machine Edition folder under the Server.

3. In the New Project dialog box, enter a name for the new project, select a template, and then click OK.

A description of the template appears below your selection. You can click hypertext links in the description for details on the template components.

At this point, you need to know the type of target hardware to which your project will download, as this will determine which template you choose. In some project templates (such as the “View/Control” template), you can select this as one of several parameters within an HTML page on the template dialog box.

The project is opened in Machine Edition and the Navigator changes to reflect those components that make up the project you just created.

---

**?** **Want to know more?** In the Help Index, look up “templates” and choose “Creating a New Machine Edition Project” or “Creating a new project under a Change Management system”.

---

4. In the Variable List, create variables for your application.

In the Variables tab of the Navigator, right-click the Variable List node, point to New Variable, and choose the type of variable you want to create. By default, the Variable List node filters out all system variables. System variables are created automatically when you add components to Windows PC, QuickPanel+, QuickPanel View/Control, QuickPanel, or Emerson IP Controller targets. To display all variables including system variables, right-click the Variable List node, point to Filter By, and choose No Filter.

---

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

---


5. Create your application.

With Logic Developer - PC, add logic (Ladder, FBD, IL, ST, SFC) to your project, and then configure a control I/O driver to model your hardware. If the template you chose did not include a Logic component, add one now—right-click the target,

point to Add Component, and choose Logic. (View-only targets do not support logic.)

- In the Project tab of the Navigator, under the Logic folder, open the Ladder Editor by double-clicking the Ladder Program node.
- Drag ladder instructions from the Toolchest into the editor. You can find ladder instructions in the PC Ladder Instructions drawer of the Toolchest.
- Assign variables to instructions. You can do this with the Smart List, which appears when you insert or double-click an instruction. Or, drag a variable from the Variables tab of the Navigator and drop it on the instruction you want it mapped to.
- You can add additional logic blocks (ladder subroutines, FBD, IL, ST) and organize your logic with SFC.

---


 **Want to know more?** In the Help Index, look up “Ladder Instructions: an Overview (PC)”.

---

With View, you can create the graphical panels and animation for your project. If the template you chose did not include an HMI component, then add one now—right-click the target, point to Add Component, and choose HMI. (If a target has both an HMI and a Logic component, you will typically create the Logic component first.)

- In the Project tab of the Navigator, open the Panel Editor by double-clicking a Panel node.
  - Use the Graphical Panel toolbar to create your HMI and/or drag graphical objects from the Toolchest onto a panel. You can find a set of fully-configured objects (complete with animation) in various Toolchest View Expert Objects drawers.
6. If necessary, configure the I/O hardware connections for your project.
- For Logic Developer - PC, set up Control I/O in the Control I/O Drivers folder (in the Project tab of the Navigator). To add a driver, right-click the Control I/O Drivers folder and choose New Driver. Use the Control I/O tool to configure your drivers. Map variables to I/O terminals.

---

 **Want to know more?** In the Help Index, look up “Control I/O Tool: an Overview”.

---

- For View, set up PLC Access I/O in the PLC Access Drivers folder. To add a driver, expand the PLC Access Drivers folder, right-click View Native Drivers, and click New Driver. Configure the drivers in the Inspector window. On Windows PC targets, you may want to set up OPC I/O in the OPC Client folder.
7. When your project is complete; validate, download, and run your project by pressing the F9 function key.

Machine Edition saves the project, performs a validation, builds the run-time files, and attempts to establish a connection to the target computer.

## 2.4.4 Validating and Downloading a Project

During project development, you will go through the validation and download processes several times. The validation process checks the project for errors. If the project contains errors, they are listed in the Build tab of the Feedback Zone.

- All errors must be fixed before the download process can proceed. Warnings are also listed in the Feedback Zone, but they do not prevent the download from starting.

When fixing project errors, use the F4 function key to scroll through errors in the Feedback Zone, jumping to their locations in the project.

The download process involves two steps. The first step creates (or “builds”) all the run-time files necessary for a target to perform its role in a completed project. The second step downloads those files to the target devices or computers.

- To download a View or Logic Developer - PC project to the Windows PC computer on which you are developing (a local target), ensure that the target property Computer Address is set to “.” (without the quotation marks). Otherwise—for remote targets—enter the IP address or computer name of the remote computer you want to download to, in the Computer Address property.
- To download a QuickPanel project, ensure that the target property Computer Download Port is set to the applicable COM port.

Additional basic information on setting up Windows PC, QuickPanel+ and QuickPanel View/Control targets follows.

### Preparing a Remote Windows PC target for Downloading

The following steps need to be completed on a remote Windows PC target computer before you can download a project to it.

1. If you are downloading to another Windows PC computer, ensure that the runtime files are installed on that device.

QuickPanel+, QuickPanel View/Control targets are shipped with the run-time files preinstalled. If you are using your own Windows PC computer, you will have to install the runtime files yourself from the Machine Edition installation DVD (see page 4).

2. Share the Machine Edition installation directory with the development computer that contains the project files.

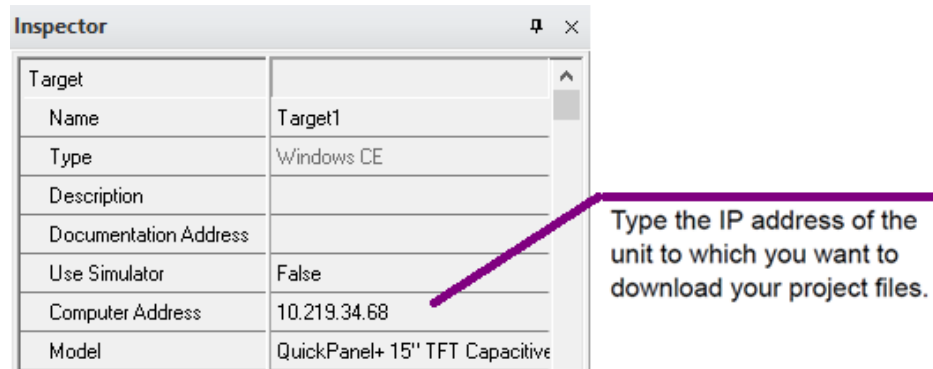
In the Windows PC Explorer, navigate to the PAC Machine Edition folder, right-click it and choose Sharing. Click Shared As and in the Share Name field, enter PAC Machine Edition.

3. Ensure you have the proper permissions to download files to the target computer.

In Windows PC, open the Control Panel, click Administrative Tools, and then click Computer Management. Expand “System Tools” and then expand “Local Users and Groups”. Select the Users folder. In the right pane, double-click the Guest user account. In the Guest Properties dialog box that appears, clear the Account is Disabled check box.

4. Within your Machine Edition project, specify the IP address or computer name of the remote Windows PC target in the Computer Address target property. This enables Machine Edition to locate and connect to the Windows PC target computer.

**Figure 15: Preparing a QuickPanel+ Target for Download**



**?** **Want to know more?** In the Help Index, look up “Preparing Windows PC targets for downloading”.

### Preparing a QuickPanel+ or QuickPanel View/Control target for downloading

You need to complete the following steps on a QuickPanel+ or QuickPanel View/Control target before you can download a project to it. If your local computer is not physically connected to the QuickPanel+ or QuickPanel View/Control unit but you want to test your project, you can download the project to your local computer by setting the Use Simulator target property to True. You cannot download a project that contains a logic component to QuickPanel+ or QuickPanel View targets, because they do not support logic.

1. Physically connect the QuickPanel+ or QuickPanel View unit to the Ethernet network. This step may require a network administrator.
2. Within your Machine Edition project, specify the IP address of the QuickPanel+ or QuickPanel View/Control unit in the Computer Address target property. To find the IP address of a QuickPanel+ or QuickPanel View/Control unit, double-tap the System Information icon located on the unit’s desktop.

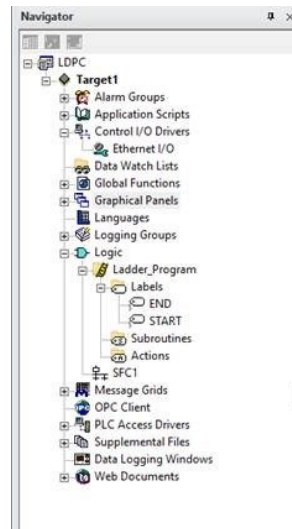
By default, the QuickPanel+ or QuickPanel View/Control unit uses a DHCP (Dynamic Host Configuration Protocol) IP address. This means that the IP address is dynamically assigned by the Ethernet network server using DHCP. If you want to use a fixed IP address, you must manually set the IP address on the unit.

**?** **Want to know more?** In the Help Index, look up “Preparing QuickPanel+ and QuickPanel View/Control targets for downloading”.

## Logic Developer - PC

Logic Developer - PC is Machine Edition's PC Control product, where you edit a project's logic component. With Logic Developer - PC you create control programs using standard IEC 61131-3 editors in a graphical environment. The programs you create are downloaded to and executed on the PC Logic Controller. You interface to a plant or process through the I/O system of your choice; Logic Developer - PC supports a wide range of industry standard hardware.

Figure 16: Navigator Project tab



Logic Developer - PC supports Windows 10 ("Windows PC" for short), QuickPanel+ and QuickPanel Control/View targets. All user-defined logic programs are contained in the Logic folder, including the following:

- Sequential Function Chart (SFC) editor (see page 29)
- Ladder Program (see page 32)
- Instruction List Blocks (see page 34)
- Structured Text Blocks (see page 36)
- Function Block Diagram (see page 38)

The following run-time elements are included with Logic Developer - PC:

- Logic Developer - PC Web Access (see page 41)
- Control I/O Drivers (see page 43)
- PC Logic Controller (on target computer) (see page 45)

---

**Tip**

User-defined folders can be added to the Logic folder or to another user-defined folder to organize your logic. A user-defined folder can contain any type of logic block contained in the logic folder, except ladder (of which there can be only one).

---

**To create a project that includes a logic component**

1. In the Manager tab of the Navigator, right-click My Computer and choose New.

The New Project dialog box appears.

2. In the Project Name box, enter a name for your project.
3. From the Project Template list, choose a template that includes a logic component.

A preview of what each template includes is displayed as you select different templates.

4. Click OK.

Your new project is created, and the Project tab opens in the Navigator.

**To add a logic component to an existing project**

- In the Project tab of the Navigator, right-click a target, point to Add Component, and click Logic.

The Logic folder with an empty ladder program and SFC component is added to your project.

**To add a user-defined folder to your project**

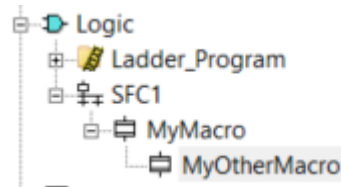
1. Ensure that your project contains a logic component (see above).
2. Right-click the Logic folder or a user-defined folder, point to New, and click Folder. The new folder appears in the Project tab of the Navigator in alphabetical order.
3. (Optional.) Enter a name for the new folder.



## 2.5 SFC EDITOR

SFC logic is a graphical language for organizing the sequential execution of control logic. The SFC editor is an easy-to-use graphical tool for editing sequential function charts in the Machine Edition environment. With the SFC editor, you can work on a disk copy of a sequential function chart (offline) or monitor the execution of an SFC running in the Controller (online). You can also configure the appearance and behavior of the SFC editor.

**Figure 17: Navigator Project tab SFC nodes**



### 2.5.1 Sequential Function Chart

An SFC is much like a flow chart; it is a high-level control program that organizes detailed blocks of logic into a state-like flow. In general, an SFC consists of steps alternating with transitions. A transition can be a BOOL variable or a Structured Text statement, which resolves to a BOOL value. Each step represents a state that the program is in until the transition following it allows execution to proceed to the next step. Each step contains actions that are user-defined sections of ladder, IL, ST, or FBD logic. The actions in a step are executed in the order they are listed and are repeated each Controller scan while the step is active. Each action in a step has a qualifier associated with it that determines when and how the action is executed.

An SFC document is a grid of cells. Each cell can contain an instruction. The instruction set includes steps, transitions, branches, parallels, and jumps.

An SFC can be expanded by adding macros. Each macro is a special SFC document that can be referenced any number of times from its SFC document or other macros.

---

**?** **Want to know more?** In the Help Index, look up “macro” and choose “SFC Macro”. You can also look up “SFC Tutorial” in the online help Index. This tutorial takes you step-by-step through the creation of a Sequential Function Chart that controls traffic signals.

---

### To add a new SFC

1. In the Project tab of the Navigator, right-click the Logic folder or a user-defined folder, point to New, and click SFC.

The new SFC appears under the Logic folder or a user-defined folder in alphabetical order.

2. (Optional.) Enter a unique name for the new SFC.

### To open an SFC or macro for editing

- In the Project tab of the Navigator, under the Logic folder or under a user-defined folder, double-click an SFC or macro node.

The SFC editor opens with the document ready for editing.

### To create a new SFC macro

1. In the Project tab of the Navigator, under the Logic folder or under a user-defined folder, right-click an SFC or macro node and click New.
2. (Optional.) Enter a unique name for the new macro.

Your new macro appears as a child node of the SFC or macro node you started with.

## 2.5.2 Working with the SFC editor - Offline

The SFC editor interacts with the Machine Edition tools to provide maximum flexibility when editing a program. The following picture illustrates some of the operations you can perform.

Figure 18

The screenshot shows the SFC editor interface with several key components:

- Navigator:** A tree view on the left showing the project structure, including folders like 'Logic' and 'User-Defined', and various SFC nodes.
- Inspector:** A pane below the Navigator showing the properties of the selected SFC node.
- SFC Diagram:** The central workspace showing a Stateflow chart with steps, transitions, and actions.
- Toolchest:** A vertical toolbar on the right containing icons for creating and editing SFC elements like steps, transitions, and actions.
- Companion:** A pane at the bottom left providing detailed information about the selected variable, including its data type and current value.
- References:** A pane at the bottom right showing all instances of the selected variable throughout the project.

Annotations in the image point to specific features:

- Variables:** Points to the Navigator, stating that variables in the Navigator display their properties in the inspector pane.
- Toolchest:** Points to the Toolchest, stating that instructions, rows, and columns can be dragged with the SFC toolchest.
- Blank Cell:** Points to a blank cell in the SFC diagram, stating that right-clicking a blank cell starts editing.
- References:** Points to the References pane, stating that it locates all instances of a variable.

### To insert an SFC instruction

1. In the SFC editor, right-click a blank cell and click Place Instruction.
2. Choose an instruction mnemonic from the smart list that appears and press ENTER to finish.

You can set the SFC editor options to automatically assign default variables to instructions or to let you choose and create your own.

### To Quick Edit an SFC

- In the SFC editor, click a blank cell and type instruction mnemonics separated by operators (see the following table). Press ENTER to finish.

Each mnemonic specifies one or more instructions. Each operator moves the location of the currently selected cell. Default variables are assigned when Quick Editing an SFC.

### SFC Operators

To	Use
Move Right One Cell	+
Move Left One Cell	-
Move Down One Cell	;
Stay On Current Cell	/

## 2.5.3 Working with the SFC editor – Online

In the SFC editor, you can view the execution of an SFC as it happens.

### To go online to the Controller's SFC

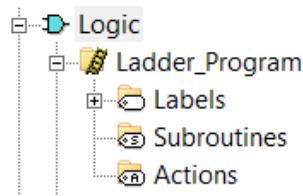
- In the Project tab of the Navigator, right-click the target and click Go Online. (Before you can go online to the Controller, you must first download the project.)

The Controller's SFC appears in the SFC editor. When the Controller is running, active steps and transitions are graphically indicated.

## 2.6 LADDER EDITOR




Ladder logic may be the most popular language in use today for creating control programs. The ladder editor is a graphical tool for editing ladder programs in Machine Edition. With the ladder editor, you can work on a disk copy of a ladder program (offline) or with a live program as it runs in the Controller (online). You can also configure the appearance and behavior of the editor under the Options tab of the Navigator.

**Figure 19: Navigator Project tab ladder logic nodes**



## 2.6.1 Ladder Program

A ladder program is composed of rungs of ladder logic instructions that execute sequentially from left to right, top to bottom. The instruction set includes standard IEC 61131-3 functions supplemented by a large library of advanced and I/O specific functions. Along with rungs and instructions, a ladder program contains:


- Labels: Each default (START and END) or user-defined  label is a rung providing an entry point for program execution.
- Subroutines: Each user-defined  subroutine is a separate, reusable section of logic that can be called from anywhere in a ladder program.
- Actions: Each  action is a separate block of logic that can be referenced in a Sequential Function Chart.

---

### Tip


You can also create User-Defined Function Blocks (UDFBs). Each UDFB is a callable subroutine to and from which you can pass parameters.

---

 **Want to know more?** In the Help Index, look up “ladder” and choose “Ladder Program: an Overview”. Also look up “UDFB” and choose “User-defined Function Blocks: an Overview”.



---

### To open a ladder program for editing

- In the Project tab of the Navigator, under the  Logic folder, double-click the Ladder\_Program node.

Your ladder program opens in the ladder editor.

### To locate a label

- In the Project tab of the Navigator, under the  Logic folder, under the Ladder Program folder, double-click a  label node.

Your ladder program, subroutine or action opens in a ladder editor window with the label selected.

### To create a new subroutine or action

1. In the Project tab of the Navigator, under the Logic folder, under the Ladder Program folder, right-click the Subroutines or Actions node and click New.
2. (Optional.) Enter a unique name for the new subroutine or action. An empty subroutine or action, labeled with the name you entered, is created. Double-click the node to open it for editing. You can add ladder logic to the subroutine or action as required.

## 2.6.2 Working with the ladder editor – Offline

The ladder editor interacts with all of the Machine Edition tools to provide maximum flexibility when editing a program.

### To insert a rung

- In the ladder logic, right-click a rung, shunt, label or description and click Insert Rung.

An empty rung is inserted. You can set the editing options for the ladder editor so that rungs are inserted either before or after the current selection.

### To Quick Edit ladder logic

- In the ladder logic, click a rung and type instruction mnemonics separated by semicolons, and then press ENTER.

The instructions specified by the mnemonic string you entered are inserted, in order, on the rung. You can enter mnemonics followed by variable names if you want to assign instruction parameters while Quick Editing.

**Example:** Add a Normally Open Contact (NO) instruction followed by a Coil (OUT) instruction and assign the variables MySwitch and MyCoil to these instructions respectively.

NO MySwitch; OUT MyCoil.

## 2.6.3 Working with the ladder editor - Online

With the ladder editor you can go online to a ladder program being run by the Controller (page 45). While online, you can view the power flow through the ladder logic and watch data values change in real time. You can also edit the ladder program, just as you would while offline, with the following restrictions:

- Labels, subroutines, and actions cannot be deleted while online.
- Only one rung can be changed before writing the changes to the Controller.

The following picture illustrates the ladder editor while online to the Controller.

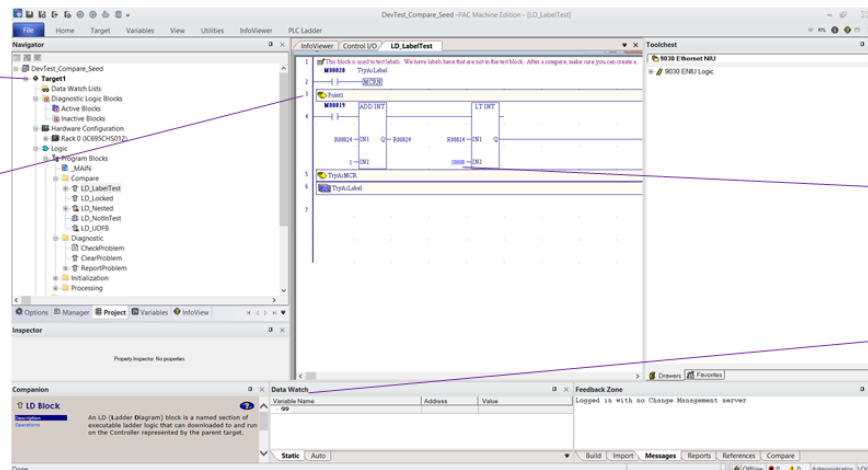
**Figure 20: Ladder Editor Environment – Online**

Online status is indicated by the color of the target icon.

New or edited rungs are indicated with alternately colored and bolded lines until you Write Changes.

Data values are displayed in real time as they change each scan on the Controller.

Use the Data Watch to view or set data values for any variable in your project.



### To go online to the Controller's ladder program

- In the Project tab of the Navigator, right-click the target and click Go Online. (Before you can go online to the Controller, you must first download the project.)  
The Controller's ladder program appears in the ladder editor. When the Controller is running, power or state flow is graphically indicated, and data values updated.

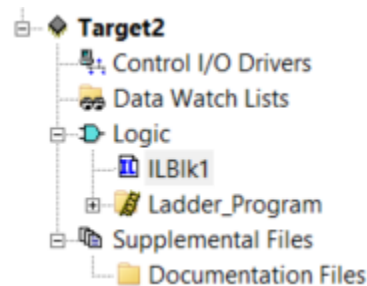
### To write a changed rung to the Controller

- From the Controller menu, choose Write Changes. (The Controller menu is available only when the ladder editor is selected.)  
The Controller's ladder program is updated with the single altered rung.

## 2.7 INSTRUCTION LIST EDITOR

IL logic is one of four languages specified by the IEC 61131-3 standard. The IL editor is an easy-to-use, intelligent, free-form text editor for editing IL blocks in the Machine Edition environment. With the IL editor, you can work on a disk copy of an instruction list (offline) or monitor the execution of an IL block running in the Controller (online). You can also configure the appearance and behavior of the editor under the Options tab of the Navigator.

Figure 21: Navigator Project Tab IL Block Nodes



### 2.7.1 Instruction List

IL is a low-level language composed of basic and advanced Math instructions that execute sequentially from top to bottom. IL blocks can also contain labels providing entry points for program execution. In its simplest form an IL can:

1. Load a data value (operand) into the accumulator.
2. Perform an operation on the accumulator, saving the result in the accumulator.
3. Store the accumulator's value to memory (a variable).
4. Do another accumulator load or operation, and so on.

---

**? Want to know more?** In the Help Index, look up "IL" and choose "IL Editor".

---

### To open an IL block for editing

- In the project tab of the Navigator, under the Logic folder or under a user-defined folder, double-click an IL block node.  
Your IL block opens in the IL editor.

### To create a new IL block

1. In the Project tab of the Navigator, right-click the Logic folder or a user-defined folder, point to New, and click Instruction Language.
2. (Optional.) Enter a name for the new IL block.

Your new IL block appears as a child node of the Logic folder or a user-defined folder.

---

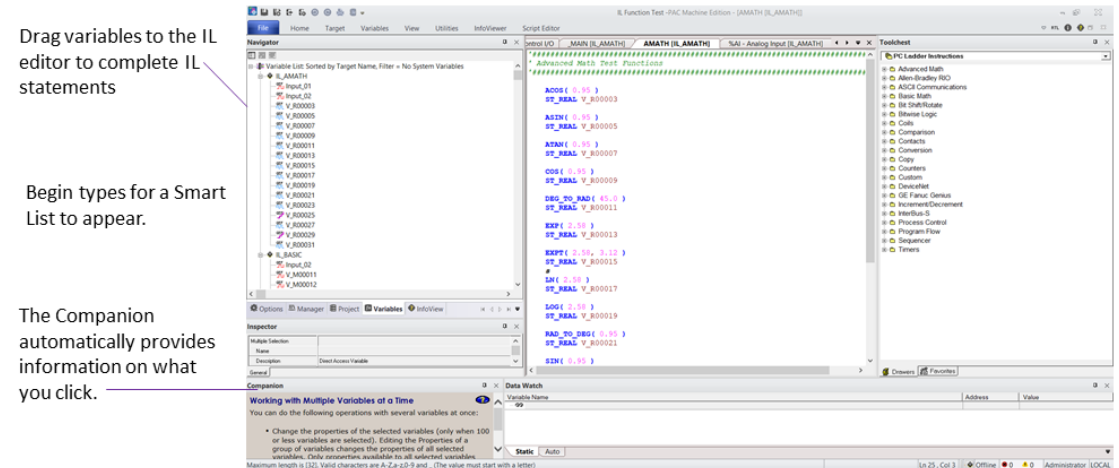
**Note:** *IL blocks can be executed only when called as Actions from an SFC.*

---

## 2.7.2 Working with the IL editor – Offline

The IL editor interacts with the Machine Edition tools to provide maximum flexibility when editing a program. The following picture illustrates some of the operations you can perform.

**Figure 22: IL Editor Environment - Online**



### To insert an IL instruction

1. In the IL editor, begin typing.  
A smart list appears containing any items (labels, mnemonics, or variables) that are applicable at the current cursor position.
2. Enter or choose from the list the mnemonic you want.

## 2.7.3 Working with the IL Editor – Online

In the IL editor, you can view the execution of an IL block as it runs.

### To go online to the Controller's IL

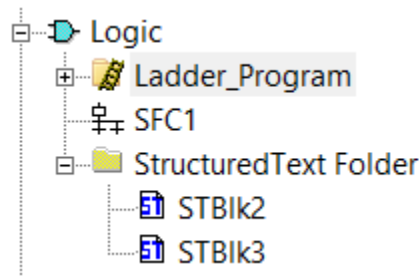
1. In the Project tab of the Navigator, right-click the target and click Validate (F7).
2. Right-click the target and click Download (F8).
3. Right-click the target and click Run (F9).
4. Right-click the target and click Go Online (CTRL+F11).

The Controller's IL block appears in the IL editor as read-only and with a grey background. In the IL editor, you can monitor the values of a variable, by clicking the variable. You can also change or force BOOL variable states.

## 2.8 STRUCTURED TEXT EDITOR

ST logic is one of four languages specified by the IEC 61131-3 standard. The ST editor is an easy-to-use, intelligent, free-form text editor for editing ST blocks in the Machine Edition environment. With the ST editor, you can work on a disk copy of a structured text block (offline) or monitor the execution of an ST block running in the Controller (online). You cannot edit an ST block online.

**Figure 23: Navigator Project tab IL block nodes**



### 2.8.1 Structured Text

ST is a high-level language composed of basic and advanced Math instructions. ST blocks can be executed as Actions called from SFC Steps, or when called as a subroutine from other ST blocks or ladder programs.

If there is an ST block named 'MAIN', it executes after the SFC programs. That is, after downloading a Logic Developer - PC application, the execution of every scan begins with the ladder program, then any SFC programs, and then the ST block named MAIN.

ST supports a timer, arrays, bitwise access, strings, and User Defined Function Blocks (UDFBs—callable subroutines to and from which you can pass parameters). The Return keyword causes an early exit from a subroutine.

---

**Want to know more?** In the Help Index, look up “ST Editor (PC)” or look up “UDFB” and choose “User Defined Function Blocks: an Overview”.

---



### To open an ST block for editing

- In the Project tab of the Navigator, under the Logic folder or under a user-defined folder, double-click an ST block node.

Your ST block opens in the ST editor.

### To create a new ST block

1. In the Project tab of the Navigator, right-click the Logic folder or a user-defined folder, point to New, and click ST Block.
2. (Optional.) Enter a name for the new ST block.

Your new ST block appears under the Logic folder or under a user-defined folder.

## 2.8.2 Working with the ST Editor - Offline

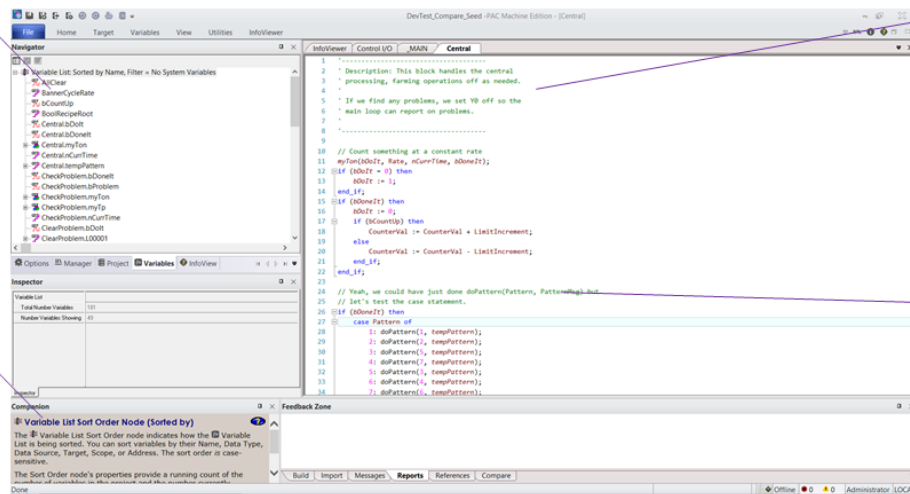
The ST editor interacts with the Machine Edition tools to provide maximum flexibility when editing a program. The following picture illustrates some of the operations you can perform.

Figure 24: ST Editor – Offline

Drag variables to the ST editor to complete ST statements.

Copy and paste ST statements from the ST editor to other sections or ST blocks.

The Companion automatically provides information on what you click.



Multiple-line block comments to document your ST block.

Hover over a variable with your pointer to see its data type.

In-line comments for individual ST statements.

Locate all instances of a variable in the References tab of the Feedback Zone.

### To insert an ST instruction

1. In the ST editor, begin typing. To see a list of valid variables, right-click anywhere and click Insert Variable. Choose a variable from the list that appears, then press ENTER.
2. To obtain a list of valid mnemonics, right-click anywhere and click Insert Keyword. Choose a keyword from the list that appears, then press ENTER.

## 2.8.3 Working with the ST editor - Online

In the ST editor, you can view the execution of an ST block as it runs.

### To go online to the Controller's ST

1. In the Project tab of the Navigator, right-click the target and click Validate (F7).
2. Right-click the target and click Download (F8).
3. Right-click the target and click Run (F9).
4. Right-click the target and click Go Online (CTRL+F11).

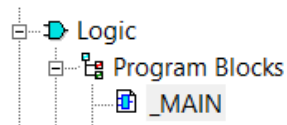
The Controller's ST block appears in the ST editor as read-only and with a grey background. In the ST editor, you can monitor the values of a variable, by clicking the variable. You can also change and/or force BOOL variable states.

## 2.9 FUNCTION BLOCK DIAGRAM EDITOR

Function Block Diagram (FBD) logic is one of four languages specified by the IEC 61131-3 standard. The FBD editor is a graphical, free-form editor for editing FBD logic in the Machine Edition environment. With the FBD editor you can work on a disk copy of a function block diagram (offline) or monitor the execution of an FBD running in the Controller (online). You cannot edit an FBD online.

---

**Figure 25: Navigator Project Tab FBD Nodes**



### 2.9.1 Function Block Diagram

A Logic Developer - PC FBD is a named section of Function Block Diagram logic that is compiled and downloaded to the PC represented by the parent target. The graphical FBD language can be used where there is a flow of signals between control blocks.

FBDs can be executed as Actions called from SFC Steps or called from other FBDs, ladder programs, or ST blocks. An FBD can contain instructions, wires, and variables, as well as ladder user-defined function blocks (UDFBs), and ST UDFBs.

---

**Want to know more?** In the Help Index, look up “FBD: an Overview (PC)”.

---

#### To open an FBD for editing

In the Project tab of the Navigator, under the Logic folder, or under a user-defined folder, double-click an FBD node.

Your FBD opens in the FBD editor.

### To create a new FBD

1. In the Project tab of the Navigator, right-click the Logic folder or a user-defined folder, point to New, and click Function Block Diagram.

A new, empty FBD with the default name “FBDBlkn” appears in the Navigator in alphabetical order under its folder.

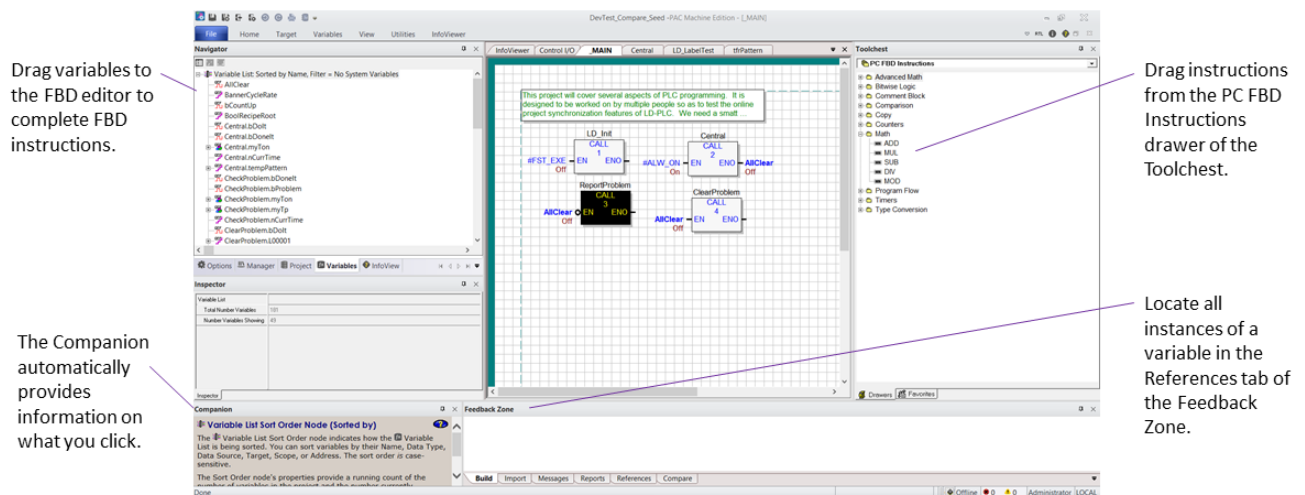
2. (Optional.) Enter a unique name for the FBD.

Your new FBD appears under the Logic folder or under a user-defined folder.

## 2.9.2 Working with the FBD editor - Offline

The FBD editor interacts with the Machine Edition tools to provide maximum flexibility when editing a program. The following picture illustrates some of the operations you can perform.

Figure 26: FBD Editor



### To insert an FBD text box (Toolchest method)

1. If you have not already done so, open the Toolchest.
2. In the Toolchest, choose the PC FBD Instructions drawer.
3. Expand the Comment Block folder.
4. Drag the text box from the Toolchest onto a blank area of the FBD editor.

You can now enter text in the FBD editor.

### To insert an FBD text box (FBD editor method)

1. In the FBD editor, right-click a blank area and click Insert Instruction.
2. In the smart list that appears, do one of the following:  
Enter the word “text” (without quotes), and then press ENTER.  
- or -  
Scroll down the list, choose Text, and then press ENTER.
3. You can now enter text in the FBD text box.

#### To insert an FBD instruction (Toolchest method)

1. If you have not already done so, open the Toolchest.
2. In the Toolchest, choose the PC FBD Instructions drawer
3. Expand an FBD instruction group.
4. Drag an FBD instruction from the Toolchest onto a blank area of the FBD editor.
5. Assign parameters above (for Call, Counter, and Timer instructions) and beside the FBD instruction, or draw wires to or from the FBD instruction.

#### To insert an FBD instruction (FBD editor method)

1. In the FBD editor, right-click a blank area and click Insert Instruction.
2. In the smart list that appears, enter or choose an FBD instruction.
3. Assign parameters above (for Call, Counter, and Timer instructions) and beside the FBD instruction, or draw wires to or from the FBD instruction.

## 2.9.3 Working with the FBD editor – Online

In the FBD editor, you can view the execution of an FBD as it runs.

Online FBD editing is not supported. However, you can change or force BOOL variable states. While online, you can also watch data variables in the Data Watch tool.

#### To watch a variable's value

In the FBD editor, right-click a variable whose value you want to monitor and click Watch.

The variable is added to the Data Watch tool for monitoring.

## 2.10 LOGIC DEVELOPER - PC WEB ACCESS


Machine Edition enables you to view a target's PC logic data with a web browser, such as Microsoft Internet Explorer (version 4.0 or later is required). When a Machine Edition project is downloaded, it automatically launches an integrated web server. Users who connect to the target computer over the web are presented with the Site Index, from which they can select the Logic Navigator. (If the target has an HMI, Web Access options for the HMI appear as well.)

The Logic Navigator presents a tree diagram of PC logic stored in the Controller, where you can browse to specific programs and blocks:

- Published SFCs and SFC macros.
- Ladder program.
- Ladder subroutines.
- Actions.
- ST blocks.
- IL blocks

Click any logic block in the tree to view it.

---

 **Want to know more?** In the InfoView tab of the Navigator, expand the Web Access library, and then select a topic under the “PC Web Access features” book.

---

### To allow remote access to your project

1. For each target that you want to allow access to, set the target's Publish Logic property to True.

This enables all PC logic except SFCs and SFC macros to be remotely viewable.

2. For each SFC and SFC macro in those targets that you want users to access, set its Publish property to True.

For example, to allow users to view an SFC remotely, set the SFC's Publish property to True.

---

**Note:** *The target containing the PC logic to be published must be set as the active target and a successful project download must occur.*

---

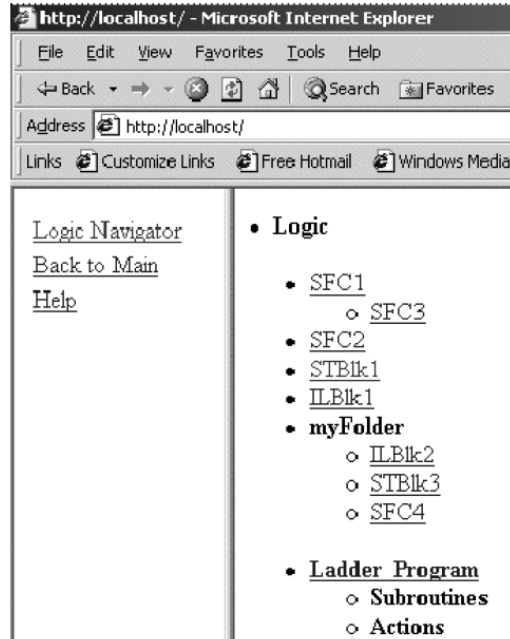
### To access the PC logic Site Index

1. Start Internet Explorer.
2. In the Address box in Internet Explorer, enter the computer name or IP address of the run-time computer.
3. If you are not sure what the run-time computer's name or IP address is, check the properties of the TCP/IP driver in the Control Panel Network setup, or contact your network administrator.
4. If you are testing the project on the local computer, enter “localhost” (without the quotation marks) as the Address. This connects Internet Explorer with the local computer.

When you connect to the run-time computer, the Site Index appears with a list of links to possible ways to view your process data.

### To view PC logic remotely

Figure 27: Viewing PC Logic Remotely



1. Start Internet Explorer.
2. In the Address box in Internet Explorer, enter the computer name or IP address of the run-time computer.  
The Site Index appears.
3. Click Logic, and then click Logic Navigator.  
An index (tree diagram) appears, which contains all published PC logic.
4. Click the SFC, SFC macro, ladder program, ST block, or IL block you want to view.  
The selected PC logic block appears. See To allow remote access to your project above for information on publishing PC logic.

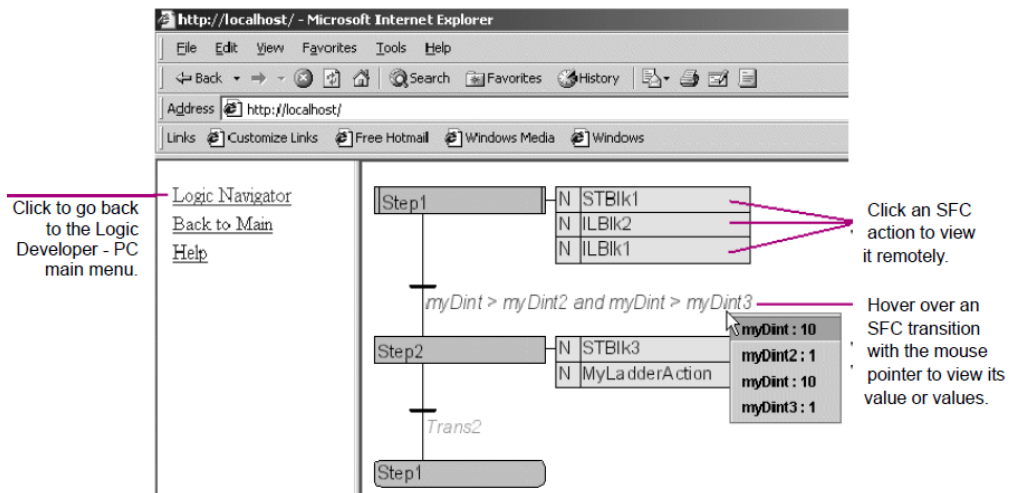
While viewing ladder logic remotely, you can view the values of variables by hovering the mouse pointer over the variable. The value of a BOOL variable on a Coil or Contact instruction is identified by its color:

- If it is green, the variable is set to True (1).
- If it is black, the variable is set to False (0).

You can view the values of variables in SFC Transitions (including Structured Text Transition statements) by hovering the mouse pointer over the Transition. The color of an SFC Transition indicates its result:

- If it is green, its result is True (1).
- If it is black, its result is False (0).

Figure 28

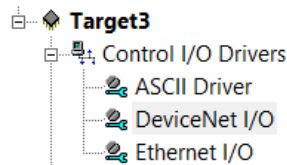


## 2.11 CONTROL I/O DRIVERS

A control program (SFC, ladder, IL, ST, or FBD) generally interfaces with the process it is controlling through some type of physical I/O equipment. For PC based control systems this usually involves installing an adapter card in the target computer that connects to an industry standard I/O network. In this way, data generated by a process can be acquired and acted upon by the Controller as it executes its program(s). Control I/O drivers provide the software required to connect the Controller with an adapter card and the I/O network of your choice.

**Want to know more?** In the Help Index, look up “Control I/O Drivers”.

Figure 29: Navigator Project tab Control I/O nodes



### 2.11.1 I/O Drivers

Logic Developer - PC comes complete with a growing library of drivers. Drivers that require no additional equipment on the target computer are the following:

- **ASCII:** Enables you to use an existing communications port on the target computer to send and receive ASCII formatted data to and from other equipment.
- **Control Peer:** Enables Controllers running on two or more computers to share process data. Messages are passed via ethernet when variable values change.

Other drivers require third-party adapters on the target computer. Logic Developer - PC supports most of today’s most popular I/O networks including the following:

- Allen-Bradley RIO
- ASCII Driver
- Control Peer Driver
- Honeywell SDS I/O
- Interbus-S I/O
- Modbus Slave

- DeviceNet I/O
- Ethernet I/O
- Fanuc CNC I/O
- Emerson Series 90-30 I/O
- Emerson Genius I/O
- Modicon MB + Distributed I/O
- Modicon Quantum-800 Series
- PROFIBUS I/O
- VersaMax Expansion I/O

#### To add a Control I/O driver to a target

- Right-click the Control I/O Drivers node, point to New Driver, and choose a driver from the list.

The new driver appears as a child node under the Control I/O node. A default configuration of the driver is also added to the Control I/O tool. You can add up to nine drivers to a target.

QuickPanel+ and QuickPanel Control supports only few of the above I/Os.

## 2.11.2 Control I/O Tool

### Control I/O

The Control I/O tool is a special editor used to graphically model the I/O equipment driven by Control I/O drivers. Each I/O system is displayed in a tree structure starting with a driver node branching down to terminals that represent physical field connections. Between the driver and terminals are nodes representing any cards, racks, or modules included in the system. Field data is shared with the components of your project by mapping variables to each I/O terminal.

---

**?** **Want to know more?** In the Help Index, look up “Control I/O Tool”.

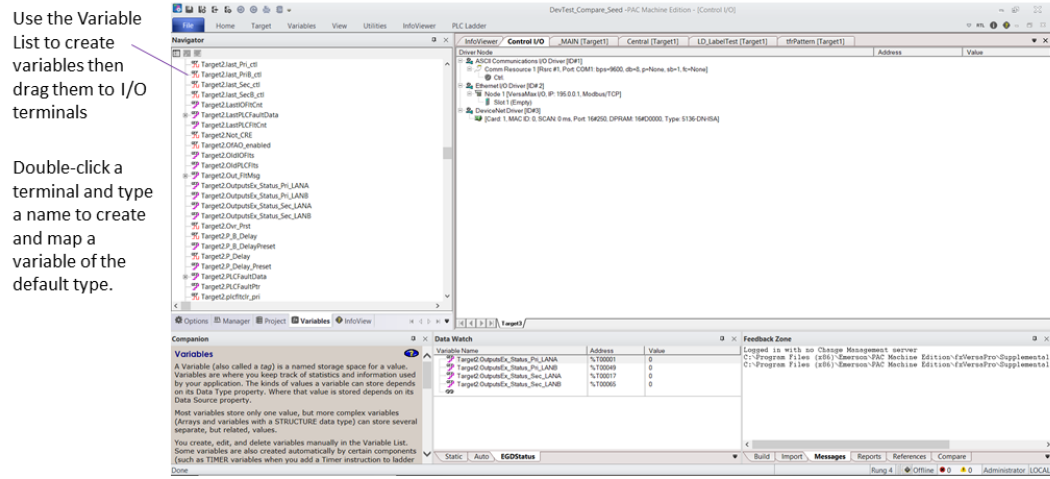
---



## 2.11.3 Working with Control I/O

The Control I/O tool interacts with other Machine Edition tools to speed the creation of your I/O system. The following picture illustrates some of the operations you can perform.

**Figure 30: Control I/O**



### To configure an I/O system with the Control I/O tool

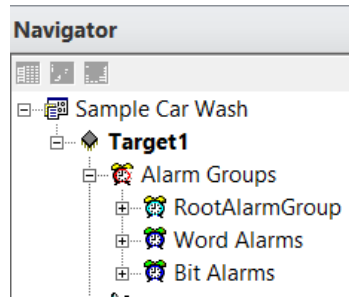
1. Right-click a driver, card, rack, or slot and click Add xyz to build an I/O tree.
2. Double-click a driver, card, rack, or slot node to configure it.
3. Map variables to I/O terminals.

## 2.12 CONTROLLER

The Controller is a run-time engine, hosted on a target, that performs the following operations in a repeated scan cycle:

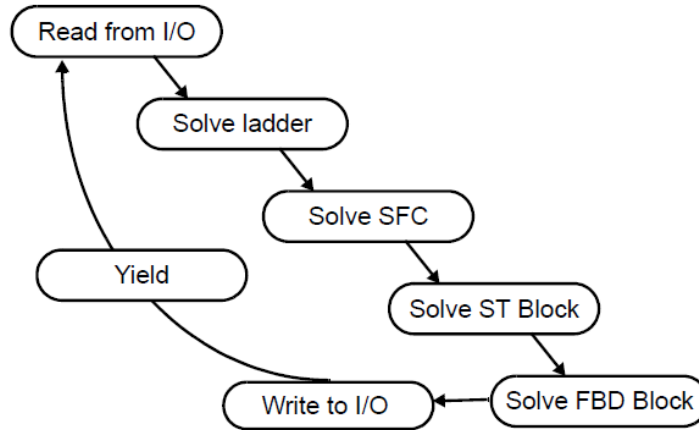
- Solves logic in a ladder program, including any called subroutines.
- Solves logic in an SFC, including any called actions.
- Solves ST block named 'MAIN' (if it exists), including any called subroutines.
- Solves FBD block named 'MAIN' (if it exists), including any called subroutines.
- Interfaces with Control I/O drivers.
- Updates the target's run-time database, which is shared with the View Runtime (see page 75).

Figure 31: Navigator Project tab Target node



The following diagram illustrates the Controller's scan cycle.

Figure 32: Logic Developer - PC Run-time Controller Scan Cycle



The Controller can be hosted on a:

- Windows PC target. The Controller appears as a Logic Developer – PC system service.
- QuickPanel+ and QuickPanel View/Control target. The QuickPanel+ and QuickPanel Control appears as a Logic Developer - PC Controller program.

You specify a target when creating a project. The Controller automatically starts when the target operating system is started.

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

## 2.12.1 Working with the Controller

The Controller's location (IP address or computer name), performance (scan rate or tuning), and behavior (response to faults, watchdog timer, and shutdown procedure) are configured by setting the properties of the target it resides on.

### To configure the Controller

1. In the Project tab of the Navigator, right-click a target and click **Properties**.
2. In the Inspector, adjust the properties.

If your project includes an HMI component, properties of the View Runtime can also be set at this time. You can also configure the Controller's properties by selecting Properties from the Controller menu. In this case, the Controller Properties dialog box appears. The information in the dialog box is the same as in the Inspector—just represented differently.

The logic component of a project must be downloaded to the Controller before the project can be run.

### To download a logic component to the Controller

In the Project tab of the Navigator, right-click a target and click Download. All components on the selected target are validated and, if no errors exist, they are downloaded to the specified computer. Errors listed in the Build tab of the Feedback Zone must be corrected before a download can be completed. Logic components are downloaded to the Controller, while HMI components are downloaded to the View Runtime.

Although the Controller service is always running, you still have control over the execution of your control programs. Offline, you can start, stop, and reset execution. When online to the Controller, additional commands are available to pause, single-scan, or enable forces.

### Run Mode Store


With the Run Mode Store (RMS) functionality, you can update a target Controller's program without stopping the Controller. It is supported on all Logic Developer - PC targets.

When you attempt to download your project to the Controller, the Download to Controller dialog box appears with the following options.

- **Attempt a Run Mode Store:** Machine Edition attempts a Run Mode Store operation. That is, it tries to download your logic to the Controller while the Controller is running.
- **Stop the Controller, then download:** Machine Edition does not attempt to use the Run Mode Store feature. Instead, it stops the Controller before initiating the download operation.
- **Download Retentive Initial Values:** Choose whether to download initial values of retentive variables, which may have changed since the previous download.

Several system tests and checks are performed when attempting a Run Mode Store operation. If these tests fail, the logic program may not be downloaded to the Controller.

---

 **Want to know more?** In the Help Index, look up "Run Mode Store (RMS) (PC)".

---

### To run the Controller

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

The ladder program, SFC program(s), the ST block named 'MAIN', and the FBD block named 'MAIN' begin executing on the target computer or unit. If the target also has an HMI component (see page 93), the View Runtime is started at the same time as the Controller.

To validate, download and run a project, right-click a target and click Download and Start "targetname".

## 2.12.2 Warm Standby


The Warm Standby feature provides an extra level of reliability for critical Windows PC, QuickPanel+ and QuickPanel Control applications. It consists of two Controllers running the Logic Developer - PC Controller (a primary and secondary or warm standby Controller), and an I/O switch that determines which Controller is currently connected to the physical I/O.

When a system using warm standby is started, the primary Controller is connected to the I/O, solves logic, and updates the secondary Controller with critical data. The secondary Controller receives the data and monitors the status of the primary Controller. When a fault is detected in the primary Controller, or when the secondary Controller cannot communicate with the primary Controller, the I/O connection is automatically switched to the secondary Controller and the secondary Controller takes over operations.

## 2.12.3 Working with Warm Standby

You must set up the warm standby hardware before you can configure the warm standby system.

---

 **Want to know more?** In the Help Index, look up "warm" and choose "Warm Standby".

---

### To configure Warm Standby

1. In the Project tab of the Navigator, right-click a Windows PC, QuickPanel+ or QuickPanel Control target and click Properties.
2. In the Inspector, scroll to the +Redundant System group of properties and double-click it to expand the group.
3. Click the Redundant Mode property and click Warm Standby.
4. Click the Controller Mode property and choose a value from the list.

## 2.12.4 Hot Standby

The Hot Standby feature provides a high level of reliability for critical Windows PC applications. It consists of two Controllers running the Logic Developer - PC Controller (a primary and secondary or hot standby Controller) over an Emerson Genius I/O network. The Emerson Genius driver automatically handles the switching between the Controllers.

To download to a target with hot standby, the primary and secondary Controllers must be synchronized, that is, they must both be downloaded.

By default, the primary Controller is active, and the secondary Controller is the backup. The backup Controller sets itself to active when one of the following takes place:


- A fault of a specified type occurs. (See To configure a Hot Standby, below.)
- A loss of communication occurs with the active Controller via the VMIPCI-5565 reflective memory cards.
- The target's Secondary Switch Time has elapsed.

Both Controllers can read from the network; however, only the active Controller can write to the network.

## 2.12.5 Working with Hot Standby

You must set up the hot standby hardware before you can configure the hot standby system.

---

 **Want to know more?** In the Help Index, look up “hot” and choose “Hot Standby”.

---

### To configure Hot Standby

1. In the Project tab of the Navigator, right-click a Windows PC target and click Properties.
2. In the Inspector, scroll to the +Redundant System group of properties and double-click it to expand the group.
3. Click the Redundant Mode property and click Hot Standby.
4. Click the Secondary Address property and enter the IP address (Controller name) of the secondary Controller.
5. Click the Secondary Switch Fault property and choose the type of fault the secondary Controller will become active on.
6. Click the Secondary Node ID property and assign a unique node ID to the VMIPCI-5565 card of the secondary Controller.
7. Click the Secondary Switch Time (ms) property and enter the number of milliseconds the secondary Controller is to wait for communication from the primary Controller before setting itself up as the active Controller.
8. Click the Primary Node ID property and assign a unique node ID to the VMIPCI-5565 card of the primary Controller.
9. Click the Primary Wait Time (ms) property and enter the number of milliseconds the primary Controller is to wait for communication from the secondary Controller before setting its synchronization bit to False.

## 2.12.6 OPC Servers, Warm Standby, and Hot Standby

Warm standby and hot standby are fully supported by the Logic Developer – PC OPC server for Machine Edition. When a configuration file is detected, the OPC server does the following:

- It generates internal objects so that it recognizes all warm standby or hot standby configurations as logical Controllers to OPC clients.
- It creates predefined OPC items for each logical Controller. These items indicate which physical Controller is active and the statuses of the primary and secondary Controllers.

### To configure an OPC server for Hot Standby and Warm Standby

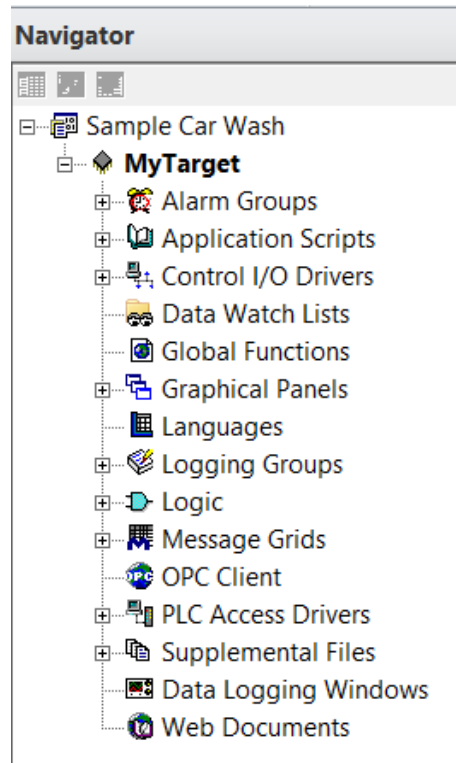
- Edit the “OPCWarmStandby.HTML” file, located in the same folder as LogicPC\_OPC.EXE on the computer hosting the OPC Server.

This XML file is used for both the Warm Standby and Hot Standby features. By default, this file is located in the “C:\Program Files (x86)\Emerson\Pac Machine Edition\OPC” directory.

## Chapter 3: View™

View is the human-machine interface (HMI) component of Machine Edition, providing the means to organize, track, display, and affect data from your plant or process. With View you create graphical panels, write scripts, configure alarm and logging schemes, and launch the project in View Runtime. View supports HMI creation for Windows 10 (“Windows PC” for short), QuickPanel+ (“QP+” for short), QuickPanel View/Control™ targets (“QPV/C” targets for short), and QuickPanel targets.

**Figure 33: Navigator Project tab HMI component nodes**



For Windows PC, QP+, and QPV/C targets, Web Access features publish data from Machine Edition HMIs for remote viewing over the Internet. Most View tools and editors are common to all targets, but a few are QuickPanel-specific; for details, see “QuickPanel Applications” on page 76.

The following are included with a View HMI component:

- Panel Editor
- Script Editor
- Alarms
- PLC Access Drivers
- Language translation support
- Message groups
- Logging (Windows PC, QP+, and QPV/C targets only)
- OPC Client & Servers (HMIs running on Windows PC targets only)

- Web Documents (Windows PC, QP+, and QPV/C targets only)
- Data Logging Windows (Windows PC targets only)
- View Runtime (Windows PC, QP+, and QPV/C targets only)
- Password security (QuickPanel+ targets )

### To create a project that includes an HMI component

1. In the Manager tab of the Navigator, right-click My Computer and click New.
2. In the New Project dialog box that appears: in the Project Name box, enter a name for your project.
3. From the Project Template list, choose “Configuration Wizard (View / Logic Developer PC)”.
4. In the preview pane, select the desired parameters for your new application (such as the target type and model).
5. Click OK.

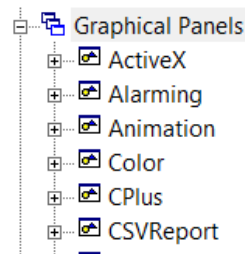
Your new project is created and the Project tab opens in the Navigator.

## 3.1 PANEL EDITOR

The Panel Editor is used to develop the layout of each panel in your HMI application. Each panel is a single screen or window of the HMI. You draw and configure graphical objects on the panel to create each screen.

Graphical objects are the various graphical elements of the HMI screen, such as rectangles, polygons, buttons, alarm displays, real-time graphs, and historical trend graphs. All objects are available from the Drawing Tools toolbar. Many items perform actions when you tap them at run time.

**Figure 34: Navigator Project tab Panel Nodes**




The Panel Editor displays the current panel as it will appear on the target unit’s screen. In addition, the Panel Editor adds a drawing grid of dots to assist you in lining up a series of graphical objects with each other.

- On Windows PC, QuickPanel+, and QuickPanel View/Control targets, many graphical objects can be configured with several types of Animation to move, change color, display data, trigger scripts, and receive input during Runtime.
- On QuickPanel targets, the Panel Editor also has a tiling grid. The tiles let you align graphical objects with the touch-sensitive cells on the surface of a QuickPanel unit’s display.



---

 **Want to know more?** In the Help Index, look up “panels” and choose “Graphical Panels”.

---

#### To add a new panel to an HMI

- In the Project tab of the Navigator, right-click the Graphical Panels folder and click New Panel.

A new panel is added beneath the Graphical Panels folder in the Navigator. If you have created a project by using a template, the Graphical Panels folder likely already contains some default panels.

#### To open a panel for editing

- In the Project tab of the Navigator, double-click a panel node. The panel appears in the Panel Editor.

#### To configure a panel

1. In the Project tab of the Navigator, right-click a panel and click Properties. The Inspector displays the panel’s properties.
2. Set the panel’s properties in the Inspector.

For a description of the panel properties, select a property and look in the Companion. Click in the Tools toolbar to open the Companion window if it is not already open.

## 3.1.1 Working with the Panel Editor

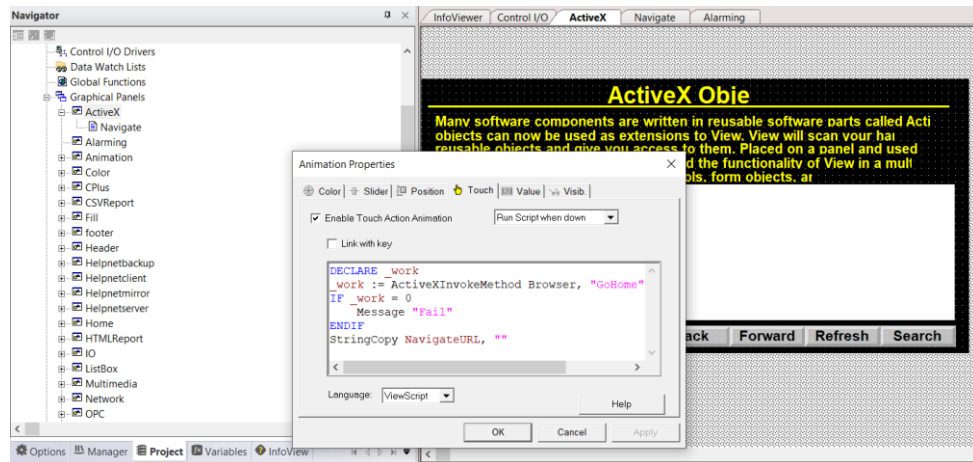
The Panel Editor interacts with all of the Machine Edition tools to assist you in the construction of graphical panels. The following picture illustrates some of the operations you can perform.

---

**? Want to know more?** In the Help Index, look up “panels” and choose “Working with Panels”.

---

**Figure 35**



### To draw a graphical object

1. If the Drawing toolbar is open, select the tool you want to use. Otherwise, right-click in the Panel Editor and choose a drawing tool.

The Companion displays instructions on how to use the selected tool.

2. Follow the instructions in the Companion to draw your object. After drawing the object, set its properties in the Inspector.

### To animate an object (Windows PC, QuickPanel+, and QuickPanel View/Control only)

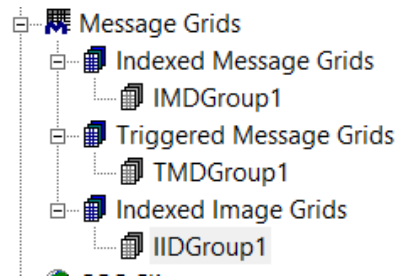
- In the Panel Editor, double-click a graphical object.


The Animation Properties dialog box appears. Settings for each type of animation are grouped on its own tab. Animations become active when the project is launched with View Runtime.

## 3.1.2 Message Displays

The Local Message Display, Triggered Message Display, and Local Image Display graphical objects display different messages or images based on the values or states of associated variables. Each of these three objects is associated with a message group of an appropriate type under the Message Groups folder—an IMD Group, TMD Group, or IID Group.

**Figure 36: Navigator Project tab Message Group Nodes**



Each graphical object can be associated with only a single message group, but that message group may be associated with many graphical objects. (On QuickPanel targets, the size of all graphical objects associated with the same message group must be the same; this size is set in properties of that message group.) You can open the grid for an associated message group by clicking the  button in the object's "IMD/TMD/IID Group" property.

Entries in a message group are edited in an appropriate grid editor.

- An Indexed Message Display graphical object shows text messages that change depending on the value or state of a single associated variable. The colors and appearance of the display can also change. The values and related appearances are set by the display's associated IMD Group.

---

**? Want to know more?** In the Help Index, look up "IMD" and choose "IMD (Local Message Display) group" and "IMD (Local Message Display) grid".

---

- A Triggered Message Display graphical object shows text messages that change depending on the results of boolean expressions. The colors and appearance of the display can also change. The values and related appearances are set by the display's associated TMD Group.

Unlike a Local Message Display, you do not associate a variable with a Triggered Message Display. Instead, you specify a series of boolean expressions in the TMD Group grid editor.

---

**? Want to know more?** In the Help Index, look up "TMD" and choose "TMD (Triggered Message Display) group", "TMD (Triggered Message Display) grid", and "Trigger Expression dialog box (TMD grids)".

---

- An Indexed Image Display shows different images depending on the value or state of a single associated variable. The values and related images are set by the display's associated IID Group. An image in an IID Group is typically a bitmap, imported from a BMP file. Images can also be sets of simple graphical objects (Arcs, Bitmaps, Circles, Lines, Pie Wedges, Polygons, Polylines, Rectangles, Round Rectangles, and Text).

---

**? Want to know more?** In the Help Index, look up IID and choose “IID (Local Image Display) group” and “IID (Local Image Display) grid”.

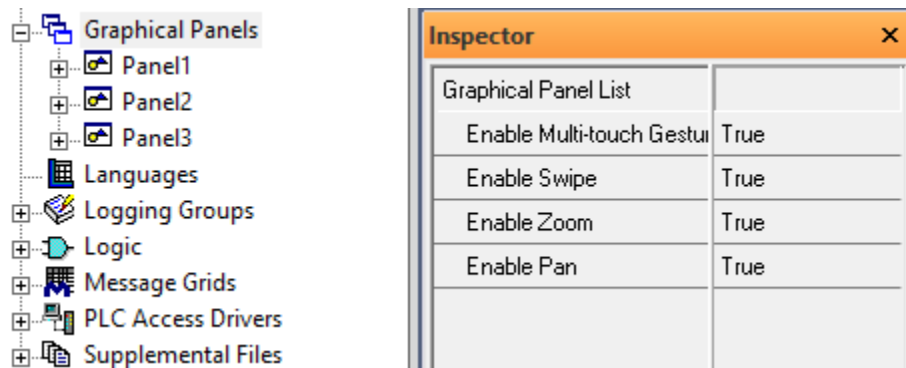
---

### 3.1.3 Multi-Touch Gestures Support on QuickPanel+

QuickPanel+ View Runtime supports the following multi-touch gestures:

- Pinch Zoom
- Panning while Zoomed-in
- Double Tap Zoom
- Swipe between Panels
- Swipe to activate Panel Scripts
- Touch and Hold

**Figure 37: Parameters to Enable Multi-Touch Gestures**



#### 3.1.3.1 To Enable Multi-touch Gestures on a Target

1. In Machine Edition, right-click the Graphical Panels node, and choose **Properties**. The inspector displays the Graphical Panels property.
2. In the Inspector panel, set the Multi-touch Gestures property to **True**.

#### 3.1.3.2 To Disable Multi-touch Gestures on a Target

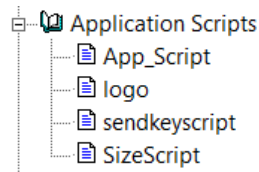
To remove Multi-touch Gestures from a target:

1. In the Machine Edition, right-click the Graphical Panels node, and choose **Properties**. The inspector displays the Graphical Panels property.
2. In the Inspector, set the Multi-touch Gestures property to **False**.

## 3.2 SCRIPT EDITOR

The Script Editor is a text editor for the various HMI scripting languages, such as ViewScript, VBScript, and QPScript. On Windows PC targets, you can also use Structured Query Language (SQL) commands to access a database for which an Open Database Connectivity (ODBC) driver is installed.

**Figure 38: Navigator Project tab Script nodes**



### 3.2.1 Scripts

Scripts are short executable programs composed of a sequence of instructions that tell a project, panel, or touch-animated object how to react to events during run-time. There are four types of scripts:

- **Application Scripts:** An application script is associated with an entire target and can be configured to execute when you start or quit the application (in View Run-time) or based on a frequency or condition while the application runs.
- **Panel Scripts:** A panel script is associated with a specific panel in a target and can be configured to execute when the panel opens, when the panel closes, or based on a frequency or condition while the panel is open.
- **Touch Animation Scripts:** (Windows PC, QuickPanel+, and QuickPanel View/Control targets only.) At run time, Touch Animation scripts are activated with keystrokes and on-screen buttons. They are written directly in configuration dialog boxes of graphical objects.
- **Global Function Library Scripts:** Global Function scripts are associated with an entire target. By default, they are executed before any other scripts, but they can be configured to run periodically or on a condition. They must be written in an Active Scripting language, like VBScript.

#### To create, open, and edit an application script

1. In the Project tab of the Navigator, right-click the Application Scripts node, point to New Script, and click either VBScript or ViewScript. (For more information on scripting, see page 58.)
2. Enter a name for the new application script.  
A new script is added to the Application Scripts node.
3. Double-click the script to open it in the Script Editor and begin editing the script.
4. In the Inspector, edit the application script properties.

#### To create, open, and edit a panel script

1. In the Project tab of the Navigator, right-click a panel node, point to New Script, and choose either VBScript or ViewScript. (For more information on scripting, see page 58.)
2. Type a name for the new panel script and press ENTER.  
A new script is added to the panel node.
3. Double-click the script to open it in the Script Editor and begin editing the script.
4. In the Inspector, edit the panel script properties.

#### To create a touch animation script

1. In the Panel Editor, double-click the object you want to animate. The Configure Animation dialog box appears.
2. Click the Touch tab.
3. Select the Enable Touch Action Animation check box and choose a touch action command that will execute a script.
4. In the large edit box, type the script that will run when the object is touched.

#### To create global function library script

1. In the Project tab of the Navigator, right-click Global Functions, point to New Library, and click VBScript.  
A new script is added to the Global Functions node.
2. Double-click the script to open it in the Script Editor and begin editing the script.
3. In the Inspector, edit the Global Function Library script properties, in other words, the name.

## 3.2.2 Scripting Languages

On Windows PC, QuickPanel+, and some QuickPanel View/Control targets, Machine Edition supports two types of scripting languages: the proprietary ViewScript language and VBScript Active Scripting languages (using Microsoft's Active Scripting engine).

On QuickPanel targets, Machine Edition uses the proprietary QPScript language (see page 78).

#### ViewScript scripting language


The ViewScript scripting language consists of the following:

- **Keywords:** View script keywords are a basic set of commands that are used to control the flow of a script and to create local variables.
- **Operators:** View operators are used to assign variable values and to perform mathematical operations
- **Script Functions:** View script functions are commands that can be written into the body of your script to monitor and react to changes in variable values. Functions are grouped in the following categories: ActiveX, Alarm Management, Animation, Application Access, CSV, File Management, Initialization File Management, List and Combo Box, Logging, Miscellaneous, Network, Panel Management, Screen

Navigation, Security, Serial Communication, Statistical Process Control, SQL, String, System, Trend Management, Video/Sound, and Web.

- Comments: Placing comments in, or between, the lines of your script is useful for debugging your script and for future reference.

---

 **Want to know more?** In the Help Index, look up “Viewscript” and choose “ViewScript HMI scripting language”.

---


### Active Scripting

Active Scripting lets Machine Edition use external scripting languages through a COM interface. Active Scripting languages are supported only on Windows PC, QuickPanel+, and some QuickPanel View/Control targets. Currently, only the VBScript Active Scripting language is supported.

By default, new projects use VBScript. You can override this default by setting a user preference.

A target can contain scripts written in either ViewScript or VBScript. Because troubleshooting such a project can be quite difficult, this practice is not recommended.

---

 **Want to know more?** In the Help Index, look up “scripts” and choose “View Scripts: an Overview”, or look up “scripting” and choose “Active Scripting in View: an Overview”.

---

### Validating a script written in an Active Scripting language


Active scripting languages are validated at run-time. You can also validate a project at any time during development to check:

- Syntax for View function calls.
- Syntax for variable, array, and structure element references.
- Limited checks for VBScript syntax.

## 3.2.3 VBScript Syntax

A detailed explanation of VBScript syntax is beyond the scope of this manual. For more information on the VBScript language usage, syntax and examples, look up VBScript at Microsoft’s website ([www.microsoft.com](http://www.microsoft.com)).

---

 **Want to know more?** In the Help Index, look up “VBScript” and choose “VBScript language: an Overview”.

---

## 3.2.4 Active Scripting Objects

Active Scripting provides a mechanism to extend a scripting language through custom COM objects. Variable syntax is, therefore, limited to using properties and methods on a COM object.

There are four types of objects in VBScript:

- Variable objects.
- Array objects.
- Structure objects.

- Application objects.

**Want to know more?** In the Help Index, look up “VBScript” and choose “VBScript language: an Overview”.

### 3.2.5 Working with the Script Editor

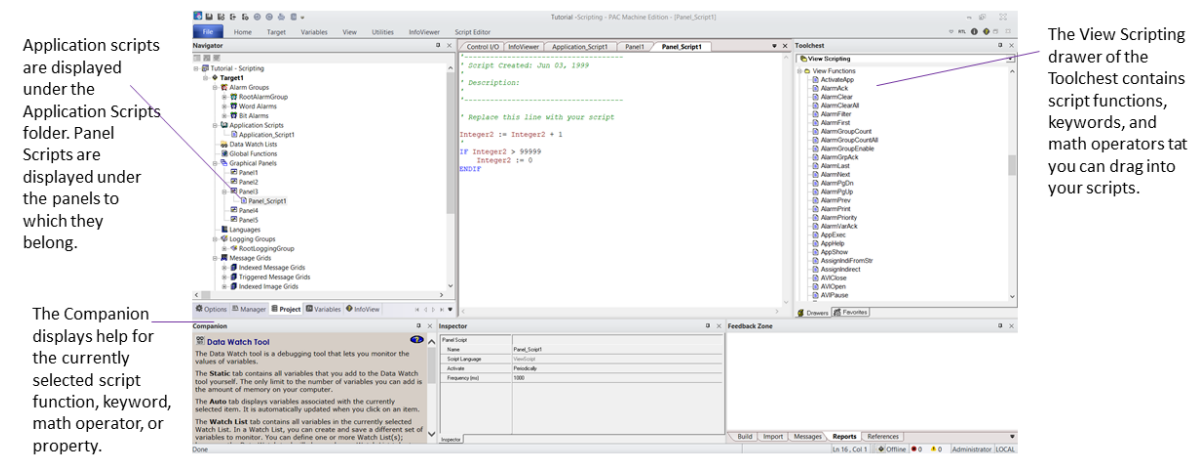
The Script Editor interacts with other Machine Edition tools to provide maximum flexibility when editing a script. The following picture illustrates some of the operations you can perform.

**Want to know more?** In the Help Index, look up “editor” and choose “Working in the View Script Editor”.

There are many ways to insert script words when writing scripts.

- If you are very familiar with the operators, functions, and keywords used in the Script Editor, start typing your script directly in the Script Editor.
- Otherwise, right-click where you want to insert a function, operator, or keyword and click Insert. Then point to the appropriate submenu for the item you want to insert and select it from the list that appears. Submenus are also available for names of the project’s variables, panels, scripts, alarm groups and logging groups. For a description of a script function, keyword, or math operator, while in the Script Editor, move the cursor on the word and press F1.

**Figure 39: Script Editor**



- All the math operators, script functions, and script keywords are also listed in the Toolchest and can be dragged into your script. The Companion provides a description of each item.

## 3.3 GRID EDITORS

With various grid editors, you can create and edit several items in your HMI targets within a spreadsheet-like table. Among the features that use grid editors are:

- Word Alarm groups
- Bit Alarm groups




- IMD (Local Message Display) groups
- TMD (Triggered Message Display) groups
- Selector Switches
- Bar Graphs
- Language translations
- Trend displays
- Passwords list (QuickPanel targets only)
- External Keypad assignment configuration

The exact functionality depends on the specific editor, but in general you can:

- Double-click a cell to edit its contents.
- Right-click a cell to perform various operations on the cell.
- Select an entire row (by clicking the grey button row headers on the left), then right-clicking the selection to perform operations on the row

---

 **Want to know more?** In the Help Index, look up “Working with QuickPanel grid editors”. You can also press F1 while working in the grid for a description of that editor.

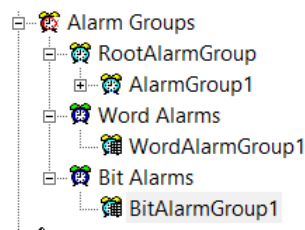
---

## 3.4 ALARMS

Alarms are messages that appear on a display in response to specified conditions in the system. These messages typically inform the operator of a situation that requires immediate attention. When a variable satisfies one of its alarm conditions, it goes into an alarm state.

---

**Figure 40: Navigator Project tab Alarm nodes**



---

The alarm system for a HMI application consists of several different elements.

### 3.4.1 Alarm Groups

There are three general types of alarms, organized into three subfolders under the Alarm Groups folder:

- Word Alarm Groups and Bit Alarm Groups are tables of variables, variable values, and associated alarm messages. When the value of a variable matches an entry in one of these tables, it goes into an alarm state and the appropriate message is added to any Alarm Display objects on the target.
- Variable Alarm Groups (Windows PC, QuickPanel+, and QuickPanel View/Control only) are sets of variables for which alarm limits have been configured. Unlike Bit or Word Alarms, you assign a variable to an alarm group in properties of the variable itself. Variable alarm groups help you organize related alarmed variables. Also, using

more than one variable alarm group makes it easier to manipulate and view alarm data at run time. A history of alarm group data can be logged daily for future analysis. You can also configure e-mail notification if any of a variable alarm group's variable go into an alarm state.

## 3.4.2 Alarm Display Objects

Active alarms can be displayed at run time with Alarm Display objects. At run time, Alarm Displays dynamically update with information about alarm states.

- If you use variable alarm groups (Windows PC, QuickPanel+, and QuickPanel View/Control targets only), you can associate an Alarm Display with a specific variable alarm group. Only variable alarms for alarms for variables in that group and its subgroups will appear on that Alarm Display.
- If you use Bit or Word alarms, alarm states and messages are based on specific values. When a variable and its value match an entry in one of the target's alarm groups, the corresponding message appears in the Alarm Window.

---

**? Want to know more?** In the Help Index, look up either “Alarms (Windows PC, QuickPanel+, and QuickPanel View/Control): an Overview” or “Alarms (QuickPanel): an Overview”.

---

### To set an alarm condition for Variable Alarms (Windows PC, QuickPanel+, and QuickPanel View/Control targets only)

1. In the Variables tab of the Navigator, right-click the variable for which you want to set an alarm condition and click Properties. The Inspector displays the variable's properties.
2. In the Inspector, double-click the Alarming property to set the general alarm parameters (the priority of the alarm, its alarm group, and how long to keep a history of alarm events).
3. Double-click the specific type of alarm (discrete alarms, limit alarms, deviation alarms, and so on) that you want to configure to set its alarm parameters. These parameters establish the ranges that place the variable in an alarm state.

### To add a Variable Alarm Group

- In the Project tab of the Navigator, under the target's Alarm Groups folder, right-click the alarm group to which you want to add the new group and click New. Enter a name for the new alarm group.

Your new alarm group appears beneath the alarm groups node. You can now select this new alarm group when configuring properties of the target's variables.

### To add a Bit or Word Alarm group

1. In the Project tab of the Navigator, under the target's Alarm Groups folder, right-click the Bit Alarms or Word Alarms folder (as appropriate) and click New.
2. Enter a name for the new alarm group.

Names must be unique within a target. You can double-click the Alarm Group to edit it.

### To create an Alarm, Display graphical object

1. In the Panel Editor, right-click and click Alarm.  
The Alarm tool is selected.
2. Draw an Alarm Object, using the information in the Companion as a guide.
3. Right-click the Alarm Object and click Properties.  
The Inspector displays the object's properties.
4. In the Inspector, edit the alarm object's properties as desired.  
The Alarm Display objects supported by Windows PC, QuickPanel+, and QuickPanel View/Control are not the same as those supported by QuickPanel targets. The objects, though similar, have distinct functionalities. For help on a specific property, click the property when the Companion window is open.

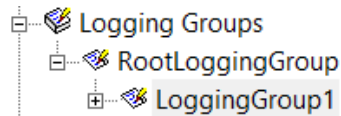
## 3.5 LOGGING DATA

(Windows PC, QuickPanel+, and QuickPanel View/Control targets only.)

Logging keeps a history of data changes in your project for later analysis.

Logging groups help sort and organize your project data. By default, all data is logged using the same logging strategy. If you want variables to have different logging strategies, you can create a hierarchy of logging groups.

**Figure 41: Navigator Project tab Logging Group nodes**



Logging Strategies define when data is logged for a group of variables. You can configure a group to log data periodically, on data changes, or on a condition. For example, one logging group might log data every minute, whereas another might log data every five minutes.

If you are using a Proficy Historian system, you can specify that logging of data from a View HMI be sent to a Historian Collector. A Collector sends data to a central Historian Server for storage and analysis. With a Collector, logging groups have no effect.

**🔗 Want to know more?** In the Help Index, look up “Logging: an Overview”, “Logging Groups: an Overview”, “View Historian Collector”, and “Data Logging Windows”.

### To enable logging of a variable

1. In the Variables tab of the Navigator, right-click a variable and click Properties.  
The Inspector displays the variable's properties.
2. Set the Enable Logging property to True.  
Logging is enabled for the variable, with RootLoggingGroup as the default logging group.

### To add a new logging group

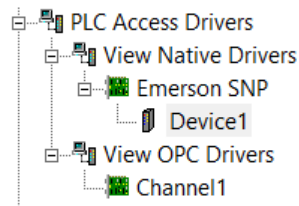
1. In the Project tab of the Navigator, right-click the group node where you want to add the new group, and click New.
2. Enter a name for the new logging group.

Your new logging group appears beneath the logging group node. To view or edit the logging group properties, right-click it and click Properties. During Runtime, the group will log variable data according to its properties. View information about selected properties in the Companion. Click on the Tools toolbar to open the Companion if it is not already open.

## 3.6 PLC ACCESS I/O

PLC Access I/O is used to enable your Machine Edition HMI unit to communicate with I/O devices on the run-time network. Typically, an I/O device is a Controller: a special hardware device that controls input and output for other, simpler devices. Different brands and types of Controllers communicate using different protocols, requiring different drivers.

**Figure 42: Navigator Project tab PLC Access I/O Nodes**



### 3.6.1 Drivers

PLC Access I/O is organized into View Native Drivers and View OPC Drivers. The latter are supported for Windows PC, QuickPanel+, and QuickPanel View/Control targets.

The View Native Drivers folder can contain the following:


- Drivers represent the drivers the target uses to communicate with devices. Windows PC, QuickPanel+, and QuickPanel View/Control targets can have many drivers; QuickPanel targets can have only one.
- Devices represent individual devices that a driver communicates with. All devices that a driver communicates with are located in that driver folder. Most drivers can have several devices. This enables you to talk to multiple Controllers, or (on Windows PC, QuickPanel+ and QuickPanel View/Control targets) to get pieces of data from the same Controller at different rates.

The View OPC Drivers folder can contain multiple channels and devices:

- Channels represent drivers that support serial communications between a target QuickPanel+ and a PLC device by means of an OPC server. Each channel can communicate with multiple devices.

- Devices represent actual hardware that the HMI communicates with by means of an OPC server. A device can be a Controller, I/O module, PC Control, or serial device.

---


 **Want to know more?** The IPI lists all of the supported PLC Access drivers. See “View, Supported H/W”. For more information, in the Help Index, look up “PLC Access Driver I/O”, “View Native Drivers”, and “View OPC Drivers”.

---

To access data values from devices of a PLC Access driver, you must also create variables with a PLC Access data source. The configuration of these variables is unique for each kind of PLC Access driver; for more details, see the online help.


If you want to connect to an Emerson IP Controller (whose application is created with Logic Developer - PLC), you can use proxy variables to simplify configuration of I/O variables in the HMI target. Proxy variables are located in an HMI target and have HMI-only properties, but represent another variable on an Emerson IP Controller target. A proxy variable's I/O-related properties (such as its I/O address) are automatically configured based on its source variable in the Controller target.

---

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

---

#### To add a View native driver to a target

1. In the Project tab of the Navigator, expand the PLC Access Drivers node.
2. Right-click the View Native Drivers node point to New Driver, and select a driver.  
The driver and a single device are added to your project and the Inspector displays the driver's properties.
3. In the Inspector, configure the driver. If applicable, click the  button in the driver's Configuration property and enter values in the dialog box.

#### To add a device to a View Native driver

1. In the Project tab of the Navigator, under PLC Access Drivers and under View Native Drivers, right-click the driver to which you want to add a new device and click New Device.  
The device, with a unique default name, is added to your project.
2. (Optional.) Enter a different name for the device.  
The Inspector displays the device's properties. You can now edit the device's properties in the Inspector.

#### To add a View OPC driver to a target

1. In the Project tab of the Navigator, expand the PLC Access Drivers node.
2. Right-click the View OPC Drivers node, point to New Channel, and select a driver.  
The New Channel wizard appears.
3. Follow the wizard to the end, in order to configure the new channel (new driver).  
After you exit the wizard, the configured driver is added to your project.

#### To add a device to a View OPC driver

1. In the Project tab of the Navigator, under PLC Access Drivers and under View OPC Drivers, right-click the driver to which you want to add a new device and click New Device. The New Device wizard appears.
2. Follow the wizard to the end, in order to configure the new device. After you exit the wizard, the configured device is added to your project.

**Note:** The View OPC drivers are not supported for QuickPanel View/Control targets currently.

## 3.7 OPC

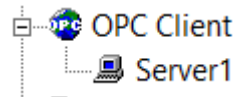
(Available only for HMIs running on Windows PC targets.)

OPC (OLE for Process Control) is an industry-standard protocol that enables software applications to transmit I/O data to each other. The two applications have a client and server relationship. You can think of OPC Servers as being producers of information and OPC Clients as being consumers:

- The OPC Server has data that it makes available to various OPC Clients.
- The OPC Client connects to an OPC Server to gain access to the server's data.

---

**Figure 43: Project tab OPC Client Nodes**



---

Because OPC is an industry standard, any OPC Client can access data from any OPC Server, regardless of the manufacturer. Machine Edition implements both the client and the server side of OPC communications.

---

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

Because an OPC Server and OPC Client are simply programs or applications, they can run on the same computer. For example, in your factory you could have another application that supports an OPC Client and a Machine Edition application running on the same computer. To allow them to share data, you would set up one of them as an OPC Server and the other as an OPC Client, then have the client application establish a connection to the server.

OPC communications rely heavily on Microsoft's OLE and DCOM technology. If the OPC Server and an OPC Client are running on different computers, you must edit the DCOM Security settings on one or both computers.

---

**?** **Want to know more?** In the Help Index, look up “DCOM” and choose “Configuring DCOM Security for OPC Communications”.

---

The OPC specifications are set and maintained by the OPC Foundation, a nonprofit organization. You can find detailed technical information about OPC on the OPC Foundation web site at [www.opcfoundation.org](http://www.opcfoundation.org).

## 3.7.1 OPC Client

The OPC Client driver enables your Machine Edition application to communicate with OPC Servers. This lets Machine Edition act as a client in the OPC Client/Server relationship. The View OPC Client driver supports all interfaces required under the OPC 1.0a, 2.0, and 3.0 specifications.

The OPC Client folder is added to a target when you add an HMI component. Each OPC Server that the application communicates with is located under the OPC Client folder, much like devices of a PLC Access driver.

To access data values from an OPC Server, you must also create and configure variables with an OPC data source.

### To add an OPC Server link to the OPC Client folder

1. In the Project tab of the Navigator, right-click the OPC Client and click New Server Link.
2. Enter a name for the OPC Server Link and click OK.

The OPC Server link is added to your project. Edit its properties in the Inspector.

## 3.7.2 Machine Edition OPC Servers

With Machine Edition OPC Servers, you can share your application's data with external OPC Client applications. This lets your Machine Edition application act as a server in an OPC Client/Server relationship. The Machine Edition OPC Servers support all required OPC interfaces in versions 1.0a and 2.0 of the OPC specifications.

Machine Edition OPC Servers are automatically installed as part of View Runtime and the Controller (sometimes referred to as the Logic Developer – PC Controller). Downloading and running a project to a target automatically enables the appropriate Machine Edition OPC Servers, based on the target's components. For example, if your project has a Logic component, Machine Edition automatically enables the Logic Developer - PC OPC Server.

While configuring an OPC Client depends on the OPC Server, typically each piece of data retrieved from a server is named an "OPC Item". All OPC Clients need to know how to address the data used by the server. The syntax for retrieving OPC items from a Machine Edition OPC Server is as follows:

Location	Syntax
local	<variable name>
remote	<remote machine name>\ <variable name> or <remote machine IP address>\ <variable name>

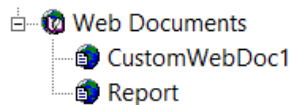
- Local indicates that the appropriate Runtime is located on the same computer as the Machine Edition OPC Server. Remote indicates that it is located on a different computer.
- The Machine Edition OPC Servers do not use access paths. Instead, as described above, the path for remote access is included as part of each OPC Item address.

## 3.8 VIEW WEB ACCESS

(Windows PC, QuickPanel+, and QuickPanel View/Control targets only.)

Web Documents provide remote access to Machine Edition project data over an Intranet or the Internet. When a Machine Edition project goes to run time, it automatically launches an integrated web server. (You can configure this web server with the ME Web Server applet in the Windows Control Panel.) Users who connect to this web site are presented with the Site Index, which contains different ways to view process information.

**Figure 44: Navigator Project tab Web Document nodes**



- **Remote Views:** Dynamic graphical recreations of actual panels from the View Runtime display.
- **Custom Web Documents:** Web pages generated using templates with special HTML tags. These tags are periodically replaced with current project data at run time. You can view web documents online or print them for distribution.
- **Variable Monitor Browser:** With this application, you can drag and drop variables from Web Documents servers onto four unique viewing tools for dynamic graphical representations of process data.
- **Remote Variable Inspector:** With this tool, you can select a variable from a list of project variables in a project and immediately receive the current value.
- **Web Control:** With Web Control, you can write changes to a target's data from Web Documents. Before using this feature, ensure that your intranet installation is secure and that no safety concerns exist with web control of a target.

**Want to know more?** In the Help Index, look up “web” and choose “Web Access: an Overview”.

### To enable remote access to your HMI project

1. For each target that you want to allow remote access to, set the target's Publish HMI property to True.
2. For each panel and variable in the target that you want users to be able to access, set its Publish property to True.



For example, to allow users to view a graphical panel remotely, set the panel's Publish property to True.

#### To access a published HMI's Site Index

1. Start Internet Explorer with current updates.
2. In the Address box in Internet Explorer, enter the computer name or IP address of the Runtime computer.

If you are not sure what the Runtime computer's computer name or IP address is, check the properties of the TCP/IP driver in the Control Panel Network setup or contact your network administrator.

If you are testing the project on the local computer, enter "localhost" (without the quotation marks) as the Address. This connects the browser to the local computer.

- When you connect to the run time computer, the Site Index appears with a list of links to possible ways to view your process data.
- The Site Index also includes a link to a page where you can download the Web Documents Browser. With it, you can configure graphs, tables, and ticker tapes to report on variables from several Machine Edition applications simultaneously.

#### To view an HMI's graphical panels remotely from a web browser

You can also view graphical panels remotely from the Web Documents Site Index.

1. Start Internet Explorer 4 or later.
2. In the Address box in Internet Explorer, enter the computer name or IP address of the run time computer.

The Site Index appears.

3. Click Remote Views.
4. Click the panel you want to view.

After a few moments, the panel will load and run. Note that the panel must be published for you to be able to view it remotely. To publish a panel, set its Publish property to True.

Because the remote viewer uses a different graphics engine to draw and update a graphical panel, there may be a few differences between the remote view of a panel and its actual appearance on an HMI. For example, historical trend objects on QuickPanel+ are not supported in remote views.

#### To add a new Custom Web Document

1. In the Project tab of the Navigator, right-click the Web Documents node and click New Web Document.
2. Enter a name for your new document.

Your new Custom Web Document appears beneath the Web Documents node. You can edit properties of the Custom Web Document in the Inspector.

### To edit a Custom Web Document

1. Outside Machine Edition, create an HTML, XML or text (txt.) document.
2. In the Project tab of the Navigator, right-click the Custom Web Document you want to edit and click Web Document Files.
3. In the Web Document Files folder that appears, place your HTML document and any logos or pictures your HTML document uses.

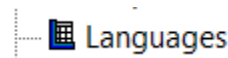
The Web Document is created from these files when you download your project. You can view Custom Web Documents from the Site Index.

## 3.9 LANGUAGES FOLDER

The Languages item opens a grid that enables you to configure multilanguage support for an HMI. At development time, you set up simple translations between an original piece of text (named the “Source”) and one or more languages. At run time, if the target’s Language Translation property is set to Enabled, all text on the HMI’s graphical panels is translated based on one of the language columns in the grid—the “Current Language”. That is, every occurrence in the original text is replaced with its corresponding item in the Current Language.

---

**Figure 45: Navigator Project tab Languages node**



---


During a validation or download, Machine Edition also checks the Source column against all graphical panels to which you have made changes. Text entries (in text objects and button objects) that do not appear in the Source column are automatically added to the grid as rows with empty translations for all languages.

The default language column used for translation is typically indicated with yellow shading. You can also specify the Source language as the default language, in which case the shading does not appear. How language translation works depends on the target type:

**Windows PC, QuickPanel+, and QuickPanel View/Control targets:** Machine Edition includes the entire language table when downloading the project to the target. Translation occurs dynamically at run time. The first language used for translation is the default language. Within HMI scripts, you can change the current language used for translation by calling the SetLanguage script function.

**QuickPanel targets:** Translation occurs only during the download operation, based on the currently-specified default language. You cannot dynamically change the current language at run time.

---

 **Want to know more?** In the Help Index, look up “Languages folder”.

---

### 3.9.1 Working with the Languages editor

When you double-click a target’s Languages node, the Languages grid appears.

Figure 46: The Languages Editor

The "Source" column contains the original words and phrases to be used during the translation process, as they appear in the target's graphical panels.

This column's shaded yellow background indicates that it is the default language. If no column has yellow shading, then the Source language is the default.

Source	Italian	French	Japanese
Tank	Serbatoio	Réservoir	タンク
### # C			### #
0 FT	0 piede	0 pied	ゼロ フット
24 FT	24 piedi	24 pieds	二十四 フット
A			ア
AL:			アル

To enter a value into a cell, double-click it (or select the cell and start typing).

If both the development and run time computers support UniCode, you can use non-English character sets in the grid (such as kanji, pictured). For example, in Windows PC you can use the IME, configured in the Regional Settings setup in the Control Panel.

Insert Row	Ctrl+F3
add Variables	Ctrl+F10
Insert Column	Ctrl+G
Delete Column	Ctrl+J
Set Default Column	Ctrl+L
Set Column Font ...	Ctrl+Alt+F
Rename Column	Ctrl+N
Find ...	Ctrl+F
Copy Cell	Ctrl+C
Paste	Ctrl+V

To edit or enter the translation of a particular word for a language, double-click its cell (or select the cell and either start typing or press F2). For a menu of commands, right-click anything in the Languages grid.

#### To add a new language column

1. Right-click in the Language Grid and click Insert Column.  
A new, empty column is added to the grid.
2. Enter the name of the new language in the column's header text box. To cancel the addition of the column, leave the text box blank and click outside the column header (or press ESC).

You can now edit the translation for the new language as normal.

#### To rename a language column

- Right-click in the Languages Grid and click Rename Column (or double-click the column header).

Enter the new name for the language in the column's header text box.

#### To move a language column

1. In the Languages Grid click and hold the header of the column you want to move.
2. Drag the mouse to the place where you want to move the column, then release the mouse button.

### To update entries in the Source column with new text entries in graphical panels

- In the Navigator, under the target whose Languages grid you want to update, right-click the Languages node and click Update Source Column. View searches every graphical panel in the HMI (regardless of whether or not the panel has changed since the last validation) and ensures that all text entries can be found in the Languages grid.

### To change the display font for a language

1. Right-click in the Languages grid, and click Set Column Font.
2. In the Font dialog box that appears, select the font you want to use to display the entries in the Languages grid, along with the font style and other attributes. The grid's cells automatically resize to fit the height of the largest font used in the grid.
3. Click OK when you are done.

The selected font set is used to display the translated text both in the Languages grid and (in Windows PC, QuickPanel+, and QuickPanel View/Control targets) the downloaded HMI. Other font attributes—such as font sizes and styles—affect only how text appears in the grid at development time.

### To set the default language column

- In the Languages grid, right-click the column you want to set as the default language and click Set Default Column.

The yellow highlight moves to the selected column. The next time you download the project to the target with translations enabled, the default language will be used by default. Note that you can also set the Source column as the default.

### To delete a language column from the grid:

- In the Languages grid, right-click the column you want to remove and click Delete Column.

The column and all its translations are deleted.

This operation cannot be undone. Before deleting a language, ensure that you no longer need information on translated words.

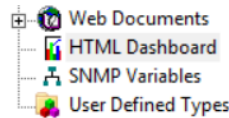
## HTML Dashboard

QuickPanel+ can host a dashboard page that can be viewed remotely on a PC /laptop/mobile device, using any browser that support HTML5. The Dashboard page shall be designed in the PME and downloaded to the QuickPanel+.

To design the Dashboard page, go to the QP+ Dashboard Configuration window by double clicking the 'HTML Dashboard' tree node.

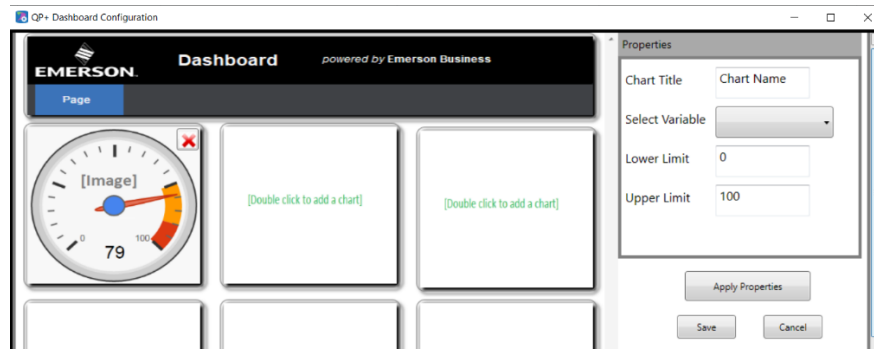
Note: Dashboard page feature uses the open source chart controls provided by 'Google'. So on double clicking the 'HTML Dashboard' node, the user acceptance dialog is displayed, which shall let the user to view the terms and conditions laid by 'Google' for the usage of the feature and then accept or not accept the same. only on accepting the terms and conditions the QP+ Dashboard Configuration window shall be displayed where the user can design the HTML page

Figure 47: HTML Dashboard



The Dashboard page editor has nine layout slots to add chart controls to the page. The three types of chart controls that are supported are Gauge chart, Bar Chart, Pie Chart.

Figure 48: Dashboard Page Editor



To make the QuickPanel+ host the HTML Dashboard page, the 'HTTP Server' service needs to be enabled by executing the following steps

- On the QuickPanel+ navigate to start-> Programs->System->QuickPanel+ Setup Tool
- Go to the 'Services' tab and enable the 'HTTP Server' by checking the check box – 'Enable HTTP Server'
- Restart the QP+

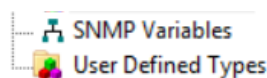
Now Quickpanel+ shall be ready to host the Dashboard page. The page can be viewed from a remote computer or mobile device using any browser that supports HTML5 (Google Chrome, Firefox, Safari, IE 8+ etc). The URL format to be used to view the HTML page is given below.

"http://QP+ IPAddress:9090/Dashboard.html ". For e.g. if the IP Address of the QuickPanel+ is 192.168.101.1, the URL shall be http://192.168.101.1:9090/Dashboard.html.

(Please refer PME help for more details).

## SNMP Variables

Figure 49: SNMP Variables



Configure SNMP variables for MIB file: (Only for Quickpanel+ targets.)

To Select variables for SNMP Communication:

1. In the Navigator, double-click on *SNMP Variables* node.

- In the SNMP Variables Selection dialog box that appears, Select the variable/variable's to be part of MIB file, and click on *Add/All buttons*.

**Note:** If Add button is clicked only last selected variable is selected, when All button is clicked all variables are selected for SNMP communication.

- Click Apply, and validate the target to check for any errors.

SNMP support is available only for below data types:

PME VIEW Data Types	SNMP Data types
BOOL	INTEGER
DINT	INTEGER
LREAL	DISPLAYSTRING
STRING	DISPLAYSTRING

SNMP Variable dialog filters out the following variables System variables, array variables, Proxy variables, Custom Variables as the functionality is not supported for mentioned data types.

After Target is downloaded, Enable SNMP functionality in Quickpanel+, for more information refer Quickpanel+ User Manual.

Limitations:

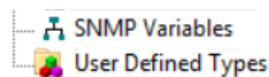
Variables appear for SNMP selection only when the target is set as active.

User has to reconfigure SNMP variables for any modification on variables like variable rename, delete operations

(Please refer PME help for more details).

## UDT

**Figure 50: UDT**



If a project is created with a template, a QuickPanel+ target contains an empty User Defined Types folder by default.

When adding a QuickPanel+ target to a project, the target also contains an empty User Defined Types folder.

The User Defined Types folder has no properties.

UDT feature is supported if and only if HMI component is added to the target.

(Please refer PME help for more details).

## 3.10 VIEW RUNTIME

(Windows PC, QuickPanel+, and QuickPanel View/Control targets only.)

View Runtime is the program that runs your project's completed HMI (human machine interface) on a target computer. It uses the files downloaded to the target during the download process. It is a separate application from Machine Edition; while you can start View Runtime from within Machine Edition, you do not need to be running Machine Edition to use it.

---

**?** **Want to know more?** In the Help Index, look up “runtime” and choose “View Runtime: an Overview”.

---

### 3.10.1 Networking

View Networking provides you with the flexibility to exchange data between run time stations. Based on client-server relationships, and supporting a distributed database system, View Networking allows access to any variable or alarm group running on any other target on the computer network.

---

**?** **Want to know more?** In the Help Index, look up “networking” and choose “View Networking: an Overview”.


---

#### To run a project from Machine Edition

- In the Project tab of the Navigator, right-click a target and click Download and Start.

All components on the selected target are validated, downloaded to the specified computer, and View Runtime is started. Errors listed in the Build tab of the Feedback Zone must be corrected before a download can be completed. HMI components are downloaded to View Runtime while logic components are downloaded to the Controller.

#### To run an HMI project outside Machine Edition

- On the Start menu, point to Emerson, and PAC Machine Edition, and click  View Runtime.

Runtime starts with the last project that was downloaded to that computer. View Runtime opens with all graphical panels configured to appear at startup (that is, all graphical panels whose Visible at Startup property is set to True).

### To interact with View Runtime

You can configure graphical objects so that you can change variable values, trigger graphical animations, and execute custom scripts with a simple touch or click.

You can monitor variable values, scripts, drivers, and the integrated web server in Runtime through interfaces that can be opened using the following shortcut keys:

To	Press
log in	CTRL+L
log out	CTRL+U
close Runtime	CTRL+Z
inspect and change variable values	CTRL+I
view and debug scripts	CTRL+G
view driver diagnostic information	CTRL+D
open a panel	CTRL+O
monitor the Web Documents server	CTRL+O
display version information	CTRL+B

## 3.11 QUICKPANEL APPLICATIONS

QuickPanel applications are HMI (human-machine interface) applications for QuickPanel units. View supports development of HMI applications using standard View and Machine Edition tools.

Development of an HMI for QuickPanel units is nearly identical to that of Windows PC, QuickPanel+, and QuickPanel View/Control units. Some of the differences are as follows:

- **Graphical Objects:** QuickPanel units support a slightly different set of graphical objects than Windows PC, QuickPanel+, and QuickPanel View/Control targets. Also, QuickPanel units do not support Animation beyond that inherent in each type of graphical object.
- **QuickPanel Panel Editor:** The Panel Editor for QuickPanel applications includes a tiling grid. This lets you easily align graphical objects with the touch-sensitive cells on the QuickPanel unit's screen.
- **Alarms:** QuickPanel units support only Bit and Word Alarms.
- **Scripts:** Scripts in QuickPanel applications use the proprietary QPScript language. QuickPanel applications do not support ViewScript or VBScript. For more information, see page 78.
- **Integration with Logic Developer - PLC:** If you have Logic Developer – PLC installed, you can use variables defined on a Emerson IP Controller target directly in your QuickPanel application. To use this feature, both the Emerson IP Controller target and the QuickPanel target must be part of the same project. You must also configure an appropriate PLC Access driver and device on the QuickPanel target, setting its PLC Target property to point to the Emerson IP Controller target. (For more information about using Emerson IP Controller variables, see GFK-1918: Getting Started Logic Developer - PLC.)



---


**?** **Want to know more?** In the Help Index, look up “QuickPanel” and choose “QuickPanel Applications: an Overview”, “QuickPanel Target”, and “QuickPanel vs. Windows PC, QuickPanel+, and QuickPanel View/Control application development”.

---

### 3.11.1 External Keypad Assignment

Some QuickPanel units support an external keypad. You enable the keypad by setting the target’s External Keypad property to True; this enables keypad configuration elsewhere in the application. There are two ways to assign an action to a keystroke:

- In the graphical panel, graphical objects that can have an assigned keystroke display a Key Assignment property. In this property, you can select a key that will activate that graphical object just as if an operator tapped it on the display.
- Actions not associated with a graphical object that appears in the panel are configured in the Keypad Assignment grid. Each panel has a separate Keypad Assignment grid.

To open the Keypad Assignment grid for a panel, select the panel’s Keypad Assignment property and click its  button. Each row of the grid specifies the action to perform when the user presses the indicated key while that panel (and only that panel) is open.

---

**?** **Want to know more?** In the Help Index look up “keypad” and choose “Keypad Assignment Grid”. Also look up QuickPanel and choose “Working with QuickPanel grid editors”.

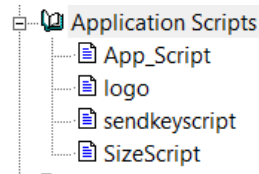
---

### 3.11.2 Scripts on QuickPanel Targets

Scripts are sets of instructions that tell a project or panel how to react to events during Runtime. Scripts on QuickPanel targets are written in a proprietary language named “QPScript”. You create QuickPanel scripts with the same View Script editor as for Windows PC, QuickPanel+, and QuickPanel View/Control targets (see page 56).

---

**Figure 51: Navigator Project tab QuickPanel Script Nodes**



---

QuickPanel applications use two types of scripts:

**Application Scripts:** Application scripts are associated with a QuickPanel target. These scripts can be executed any time your project is running on the target platform. Application scripts are located under the Application Scripts folder in the Navigator.

**Panel Scripts:** Panel scripts are associated with a graphical panel on a QuickPanel target. All of a panel's scripts are located under its Panel node in the Navigator.

---

**? Want to know more?** In the Help Index look up “QuickPanel” and choose “QuickPanel Scripts: an Overview”.

---

### 3.11.3 QPScript Language

The QPScript language is composed of the following elements.

**Statements:** There are three general kinds of statements in a QPScript script:

- Assignment statements set a variable's value to the result of an expression.
- Function calls execute one of the QPScript functions to perform a specific operation.
- Branching statements change the flow of script execution.

**Keywords:** The elements that make up these statements can be broken down into the following categories:

- Operators are used in mathematical expressions and conditional expressions.
- Functions are predefined routines that perform various operations in the QuickPanel application.
- Keywords are other reserved words used in the QPScript language.

**Expressions:** There are two general kinds of expressions in QPScript:

- Mathematical expressions produce numerical results.
- Conditional expressions produce boolean results and are used in if-then-else clauses.

---

**? Want to know more?** In the Help Index, look up “QPScript” and choose “QPScript Language Reference”.

---

# General Contact Information

Home Link: <http://www.emerson.com/industrial-automation-controls>

Knowledge Base: <https://www.emerson.com/industrial-automation-controls/support>

## Technical Support

### Americas

Phone: 1-888-565-4155  
1-434-214-8532 (If toll free option is unavailable)

Customer Care (Quotes/Orders>Returns): [customercare.mas@emerson.com](mailto:customercare.mas@emerson.com)

Technical Support: [support.mas@emerson.com](mailto:support.mas@emerson.com)

### Europe

Phone: +800-4444-8001  
+420-225-379-328 (If toll free option is unavailable)

Customer Care (Quotes/Orders>Returns): [customercare.emea.mas@emerson.com](mailto:customercare.emea.mas@emerson.com)

Technical Support: [support.mas.emea@emerson.com](mailto:support.mas.emea@emerson.com)

### Asia

Phone: +86-400-842-8599  
+65-6955-9413 (All other Countries)

Customer Care (Quotes/Orders>Returns): [customercare.cn.mas@emerson.com](mailto:customercare.cn.mas@emerson.com)

Technical Support: [support.mas.apac@emerson.com](mailto:support.mas.apac@emerson.com)

Any escalation request should be sent to: [mas.sfdcescalation@emerson.com](mailto:mas.sfdcescalation@emerson.com)

**Note:** If the product is purchased through an Authorized Channel Partner, please contact the seller directly for any support.

Emerson reserves the right to modify or improve the designs or specifications of the products mentioned in this manual at any time without notice. Emerson does not assume responsibility for the selection, use or maintenance of any product. Responsibility for proper selection, use and maintenance of any Emerson product remains solely with the purchaser.

© 2019 Emerson. All rights reserved.

Emerson Terms and Conditions of Sale are available upon request. The Emerson logo is a trademark and service mark of Emerson Electric Co. All other marks are the property of their respective owners.

