

**AUTOMATION**



Quick Start Guide

**QS SWD EN Think & Do**

Order No: 0091345

# AUTOMATION

## Think & Do Quick Start Guide

This Quick Start Guide provides the initial five sections of the full “Think & Do User Manual”. It introduces topics on developing and running control programs and Human-Machine Interface (HMI) operator interface screens using Think & Do.

04/2008

---

Designation: QS SWD EN Think & Do

Revision: A

Order No: 0091345

This user manual is valid for:

Designation	Version	Order No.
Think & Do	8.0	

## Please Observe the Following Notes

In order to ensure the safe use of the product described, read this manual carefully. The following notes provide information on how to use this manual.

### User Group of This Manual

The use of products described in this manual is oriented exclusively to qualified application programmers and software engineers, who are familiar with the safety concepts of automation technology and applicable standards.

Phoenix Contact accepts no liability for erroneous handling or damage to products from Phoenix Contact or third-party products resulting from disregard of information contained in this manual.

### Explanation of Symbols Used

**DANGER:**

This indicates a hazardous situation which, if not avoided, will result in death or serious injury.

**WARNING:**

This indicates a hazardous situation which, if not avoided, could result in death or serious injury.

**CAUTION:**

This indicates a hazardous situation which, if not avoided, may result in minor or moderate injury.

The following types of messages provide information about possible property damage and general information concerning proper operation and ease-of-use.

**NOTE**

This symbol and the accompanying text alerts the reader to a situation which may cause damage or malfunction to the device, either hardware or software, or surrounding property.



This symbol and the accompanying text provides additional information to the reader. It is also used as a reference to other sources of information (manuals, data sheets, literature) on the subject matter, product, etc.

## Please Observe the Following Notes

---

### **General Terms and Conditions of Use for Technical Documentation**

Phoenix Contact reserves the right to alter, correct, and/or improve the technical documentation and the products described in the technical documentation at its own discretion and without giving prior notice, insofar as this is reasonable for the user. The same applies to any technical changes that serve the purpose of technical progress.

The receipt of technical documentation (in particular data sheets, installation instructions, manuals, etc.) does not constitute any further duty on the part of Phoenix Contact to furnish information on alterations to products and/or technical documentation. Any other agreement shall only apply if expressly confirmed in writing by Phoenix Contact. Please note that the supplied documentation is product-specific documentation only and that you are responsible for checking the suitability and intended use of the products in your specific application, in particular with regard to observing the applicable standards and regulations.

Although Phoenix Contact makes every effort to ensure that the information content is accurate, up-to-date, and state-of-the-art, technical inaccuracies and/or printing errors in the information cannot be ruled out. Phoenix Contact does not offer any guarantees as to the reliability, accuracy or completeness of the information. All information made available in the technical data is supplied without any accompanying guarantee, whether expressly mentioned, implied or tacitly assumed. This information does not include any guarantees regarding quality, does not describe any fair marketable quality, and does not make any claims as to quality guarantees or guarantees regarding the suitability for a special purpose.

Phoenix Contact accepts no liability or responsibility for errors or omissions in the content of the technical documentation (in particular data sheets, installation instructions, manuals, etc.).

The aforementioned limitations of liability and exemptions from liability do not apply, in so far as liability must be assumed, e.g., according to product liability law, in cases of premeditation, gross negligence, on account of loss of life, physical injury or damage to health or on account of the violation of important contractual obligations. Claims for damages for the violation of important contractual obligations are, however, limited to contract-typical, predictable damages, provided there is no premeditation or gross negligence, or that liability is assumed on account of loss of life, physical injury or damage to health. This ruling does not imply a change in the burden of proof to the detriment of the user.

### Statement of Legal Authority

This manual, including all illustrations contained herein, is copyright protected. Use of this manual by any third party is forbidden. Reproduction, translation, and public disclosure, as well as electronic and photographic archiving or alteration requires the express written consent of Phoenix Contact. Violators are liable for damages.

Phoenix Contact reserves all rights in the case of patent award or listing of a registered design. Third-party products are always named without reference to patent rights. The existence of such rights shall not be excluded.

Windows 2000, Windows XP, Windows Vista, Visual Basic, VBScript, and Visual C++ are trademarks of the Microsoft Corporation. OPC is a registered trademark of OPC Foundation. Modbus is a registered trademark of Schneider Automation, Inc. INTERBUS is a registered trademark of Phoenix Contact GmbH and Co.

### How to contact us

#### Internet

Up-to-date information on Phoenix Contact products and our Terms and Conditions can be found on the Internet at:

[www.phoenixcontact.com](http://www.phoenixcontact.com).

Make sure you always use the latest documentation.

It can be downloaded at:

[www.download.phoenixcontact.com](http://www.download.phoenixcontact.com).

A conversion table is available on the Internet at:

[www.download.phoenixcontact.com/general/7000\\_en\\_00.pdf](http://www.download.phoenixcontact.com/general/7000_en_00.pdf).

#### Subsidiaries

If there are any problems that can't be solved using the documentation, please contact your Phoenix Contact subsidiary.

Subsidiary contact information is available at [www.phoenixcontact.com](http://www.phoenixcontact.com).

#### Published by

PHOENIX CONTACT GmbH & Co. KG  
Flachsmarkstraße 8  
32825 Blomberg  
Germany  
Phone +49 - (0) 52 35 - 3-00  
Fax +49 - (0) 52 35 - 3-4 12 00

PHOENIX CONTACT  
P.O. Box 4100  
Harrisburg, PA 17111-0100  
USA  
Phone +1-717-944-1300

Should you have any suggestions or recommendations for improvement of the contents and layout of our manuals, please send your comments to

[tecdoc@phoenixcontact.com](mailto:tecdoc@phoenixcontact.com).

# Table of Contents

1	Before You Begin.....	1-3
1.1	About This Manual .....	1-3
1.2	Manual Conventions .....	1-4
1.3	Starting Think & Do .....	1-5
1.4	Using Help .....	1-6
1.5	System Concepts.....	1-7
1.6	Phoenix Contact Technical Service.....	1-14
2	Creating a Project.....	2-3
2.1	Starting a New Project.....	2-3
2.2	Selecting a Runtime Target Type .....	2-9
2.3	Specifying Inputs, Outputs, and Data Items .....	2-12
2.4	Creating Control Programs.....	2-12
2.5	Creating an Operator Screen .....	2-14
3	Drivers, Devices, and Tags.....	3-3
3.1	Specifying Inputs, Outputs, and Data Items .....	3-3
3.2	Configuring I/O .....	3-6
3.3	Adding an I/O Driver .....	3-8
3.4	Connecting to I/O Devices .....	3-10
3.5	Mapping I/O Points to Data Items.....	3-13
3.6	Scanning and Monitoring I/O.....	3-15
3.7	Saving the I/O Configuration .....	3-16
3.8	Tips on Toolbar Functions.....	3-17
3.9	Watchdog Timeout Default I/O States .....	3-19
4	General Programming Techniques .....	4-3
4.1	Flow Charts and Subcharts.....	4-4

## Think & Do

---

4.2	Exploring Flow Charts and Subcharts .....	4-5
4.3	Creating a Flow Chart .....	4-6
4.4	Changing Flow Chart Settings.....	4-7
4.5	FlowView Orientation .....	4-9
4.6	Displaying Multiple Flow Charts .....	4-13
4.7	Editing Flow Charts .....	4-14
4.8	Flow Chart Block Introduction .....	4-15
4.9	Connecting Flow Chart Blocks .....	4-18
4.10	Editing Flow Chart Block Expressions.....	4-20
4.11	Entering Block Description .....	4-20
4.12	Operator Screens.....	4-21
4.13	Scripting Languages in ScreenView.....	4-34
5	Flow Charts.....	5-3
5.1	Flow Chart Types and Categories .....	5-3
5.2	Basic Flow Chart Blocks .....	5-3
5.3	Using Subcharts.....	5-31
A	Appendix.....	A-1
A 1	System Data Items .....	A-1
A 2	Notes on Reading Error Data .....	A-7
A 3	List of Figures.....	A-9
A 4	List of Tables.....	A-13
B	Index.....	B-1

# Section 1

This section informs you about:

- Features and Benefits of the Think & Do software
- System Concepts

Before You Begin .....	1-3
1.1 About This Manual .....	1-3
1.2 Manual Conventions .....	1-4
1.2.1 Text Conventions .....	1-4
1.2.2 Keyboard Commands .....	1-5
1.3 Starting Think & Do .....	1-5
1.4 Using Help .....	1-6
1.5 System Concepts.....	1-7
1.5.1 I/O Systems.....	1-7
1.5.2 Think & Do Methods and Terminology .....	1-9
1.5.3 Target Systems .....	1-10
1.5.4 Where's the Runtime Software?.....	1-11
1.6 Phoenix Contact Technical Service .....	1-14



## Think & Do

---

# 1 Before You Begin

Welcome to Think & Do™ Version 8 – an integrated control environment that lets you do more with a given PC platform. It supports development, deployment and operation of high-value automated control systems for material handling and manufacturing. Like all Phoenix Contact products, Think & Do provides an intuitive, open-architecture environment that readily integrates with hardware and software components from virtually all major suppliers.

Projects created with Think & Do integrate seamlessly with enterprise information systems to provide valuable data about system operation. Major components of Think & Do are:

- ProjectCenter: provides ready access to all project elements and the fully integrated tagname database using the Data Item Explorer.
- FlowView: where you create control logic.
- ScreenView: where you create HMI screens.
- I/O View: used to configure project I/O.
- AppTracker: a fast, graphical debugger.
- Runtime Engine: provides a robust, deterministic project execution environment.

Think & Do makes it easy to target your project to the Microsoft Windows operating system that best suits your needs. Whether you create a project for a Windows-based PC or Windows CE-based system, scaling for a different platform requires only minor adjustment.

## 1.1 About This Manual

This manual provides the initial five sections of the full “Think & Do User Manual”. It introduces topics on developing and running control programs and Human-Machine Interface (HMI) operator interface screens using Think & Do. For the complete manual, check the Think & Do CD or Phoenix Contact website ([www.phoenixcontact.com](http://www.phoenixcontact.com)).

### What’s Inside

In this manual, you’ll find the following chapters:

Section 1, “Before You Begin”, this chapter, provides an overview of Think & Do and this manual.

Section 2, “Creating a Project”, describes the steps required to create a project and start using Think & Do. It includes a brief introduction to the concept of Think & Do Runtime targets, and how to create input, output and data items using IOView. It also introduces flow charts and operator screens and how to create them using FlowView and ScreenView.

Section 3, “Drivers, Devices, and Tags”, provides details on how to configure I/O using IOView.

Section 4, “General Programming Techniques”, describes concepts for both flow charts and operator screens.

Section 5, “Flow Charts”, provides details of flow chart programming.

The “Index” provides a handy cross reference to common terms.

## 1.2 Manual Conventions

This manual uses a few special symbols and conventions.

### 1.2.1 Text Conventions

This section discusses text conventions used throughout the manual. Folder and Filenames File names and folder paths appear in quotes. For example, “C:\Sample” is a folder path; “Project.pdb” is a filename.

#### **Titles, Buttons, Keywords, Mnemonics**

Command buttons, keywords and phrases found in dialog boxes appear in quotes. For example, the text might say, ‘click the “Apply” button’ to indicate there is a button in the dialog box labeled “Apply.” Menu selections also appear in quotes (for example, “File” menu) and submenu selections follow an ellipsis after the main menu (for example, “File... Print” menu).

There are a few hypertext links to Web pages. They appear as underlined text, such as [www.phoenixcontact.com](http://www.phoenixcontact.com). When viewing this manual in Adobe Acrobat, hypertext links also appear as blue text. Clicking one of these links launches the web browser and attempts to connect to the link.

## 1.2.2 Keyboard Commands

Key names are shown as <Alt>, <Ctrl>, <Enter>, or <Home>. You'll find keyboard commands, key combinations, and key sequences as shown in Table 1-1:

Table 1-1 Manual Conventions for Keyboard Command

Keystroke(s)	Description
KEY1+KEY2	A plus sign (+) between key names means to press and hold the first key (KEY1) and type the second key (KEY2). For example, <Alt>+<F4> means to hold down the <Alt> key, press the <F4> key, and then release both the <Alt> and <F4> keys.
KEY1, KEY2	A comma (,) between key names means to type the keys individually in the sequence shown. For example, if instructed to type <R>, <Enter>, type the letter <R>, release it, then press the <Enter> key.

## 1.3 Starting Think & Do

After installation, an entry appears under "Start" for Think & Do. To start, follow these steps:

1. Select the "Start... Programs... Phoenix Contact... Think & Do... Project-Center" menu. This initiates execution of Think & Do.
2. Open or create a project by:
  - Selecting the "File...New" menu to create a project with the "untitled.pdb" name in the title bar.
  - Selecting the "File...Open" menu or the "Open" button to display a standard "Open" dialog box.

## 1.4 Using Help

There are several techniques for obtaining help from Think & Do – select “Help” from the menu bar, click a “Help” button, or press <F1>. Figure 1-1 illustrates the Help menu as it appears when selected from the menu bar.

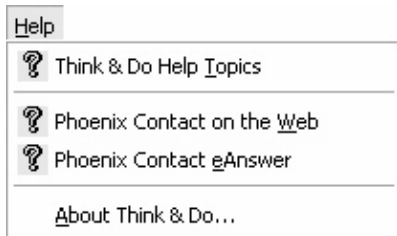


Figure 1-1 The Think & Do “Help” menu

The “Help... Think & Do Help Topics” menu item displays a “Think & Do Help” window showing the Contents tab with a list of all Help topics for Think & Do. You can also display Help Contents by pressing <F1>.

The other Help menu items are:

Table 1-2 Help Sources

Source	Result
Phoenix Contact on the Web	Connects to the Phoenix Contact website so that you can get technical support directly from Phoenix Contact.
Phoenix Contact eAnswer	A web-based support system that includes previously answered questions and a place where you can ask your questions and receive answers. You can also submit product suggestions.
About Think & Do...	Displays important information about the Think & Do and ProjectCenter, such as the copyright and version.

To locate specific information about Think & Do, click “Help... Think & Do Help Topics.” You’ll also find that some Help topics refer to other topics. You can view these related items by clicking on underlined words or phrases.

For additional information on using Help, see the operating system documentation.

## 1.5 System Concepts

It is important to learn key system concepts and terms before creating flow charts for your application. Please read this section to understand Think & Do structure.

### 1.5.1 I/O Systems

The traditional PC (personal computer) has I/O (input/output devices) oriented to office computing: printer ports, keyboard and mouse connections, etc. When using a PC for industrial control, the PC must be equipped to control industrial I/O devices. Factory I/O can include one or more racks (bases) of I/O modules (Figure 1-2), or a full factory network with many kinds of devices. We give these the collective name I/O subsystem. In order to connect with and control a machine or process, most PC-based control solutions, such as Think & Do, use either external rack(s) of Input/Output (I/O) modules or a factory network of nodes and devices.

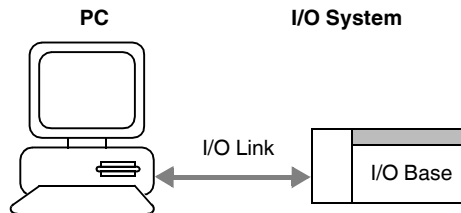


Figure 1-2 PCs connect to I/O bases in racks

## Think & Do

---

Some factory networks such as DeviceNet and INTERBUS use a trunk line with drops as shown in Figure 1-3. The device type can vary from a limit switch to a motor controller, but each one is identified by a node address. By installing the appropriate I/O scanner (adapter) card in the PC, Think & Do easily controls a factory network.

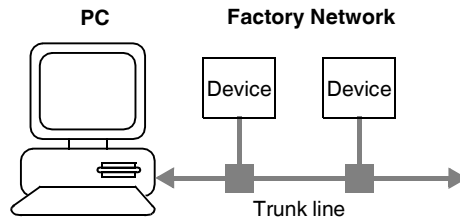


Figure 1-3 A PC can connect to a network with drops to devices

The Ethernet topology (Figure 1-4) uses an Ethernet adapter card in the PC and a central hub. The hub distributes individual connections to I/O controllers such as the popular Ethernet Base Controllers H2-EBC and H4-EBC from Automationdirect.com. The base controllers are similar to I/O concentrators in other networks, and occupy one network address on the Ethernet network.

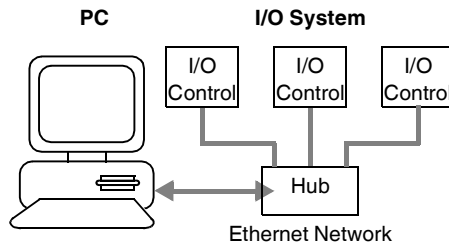


Figure 1-4 PC connected to multiple I/O controllers

The arrival of the Windows CE-based controllers makes it possible for an embedded system running Think & Do software to fit in the base with the I/O modules (Figure 1-5). The Think & Do WinPLC includes the I/O system in the base.

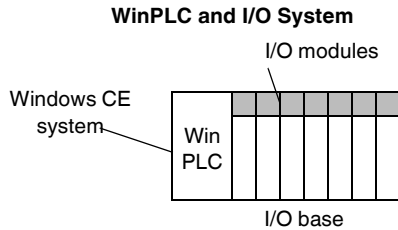


Figure 1-5 WinPLC and its one base of I/O modules

## 1.5.2 Think & Do Methods and Terminology

The software you create to control applications is called a project. A project can have one or more types of components (flow charts, HMI screens, I/O configuration, etc.).

Think & Do consists of interrelated development tools. ProjectCenter is the top-level project management tool, and includes a spreadsheet-like tool for creating a tagname database. FlowView is a development environment for designing flow charts; ScreenView provides the environment to create HMI screens; and IOView is a tool for creating the I/O configuration for the project. You can switch seamlessly from one tool to another during project development. During development, you open a project and have access to all its components through each tool.

Programmers may be used to thinking of software development in these steps:

1. Write program code.
2. Compile the code into an executable object.
3. Run the executable.
4. Test the program and debug until complete.
5. Commission the machine, wait for a machine problem or obsolescence, start at Step 1.

Think & Do equivalent activities have these steps:

1. Use FlowView to create or edit project flow charts.
2. Use ScreenView to create or edit HMI screens.
3. Build the project into appropriate Runtime components.
4. Run the project.



5. Test the project, using AppTracker and EasyTrac to debug until complete.
6. Commission the machine.

The word *Runtime* refers to the system's execution of the project created; it runs the project at Runtime. When project development is completed, use Runtime software to run the compiled project.

A *Runtime target* is the intended hardware platform that will control the final application. The target may be an industrial PC or an embedded controller. When the target is a different platform from the PC used for development, the terms *remote* and *local* are used (with respect to the programmer) for differentiation. The next section describes typical configurations in these terms.

### 1.5.3 Target Systems

Many applications require the PC used for project development to also run the project. However, you can create a project on a PC to run on a different platform at a later time. In either case, the target system is the PC, which has the Think & Do Runtime software and runs the project. If you are developing and running the project on the same PC the target is a "Certified PC".

The concept of having a target machine run a project that is different from the development platform provides flexibility for various scenarios. In all cases, the development system must run on a Windows-based PC. The Runtime may run on a Windows-based PC or Windows CE system. In fact, there can be several Runtime platforms attached to a single development system, as shown below. This allows the targeting of a particular machine from the group to run the project just developed.

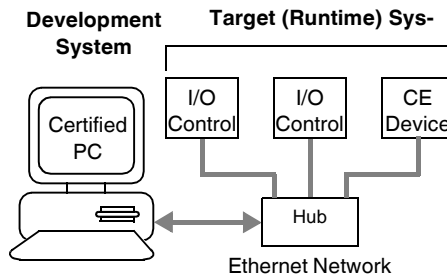


Figure 1-6 Group development system



In Think & Do, a PC running Windows 2000, XP, or Vista is called a “Certified PC”. This manual and Help use the term Certified PC or Windows-based PC when referring to a PC that is using one of these operating systems .

#### 1.5.4 Where’s the Runtime Software?

Each Think & Do package includes a CD-ROM that equips one Windows-based PC to develop and run Think & Do projects. But what about Windows CE hardware platforms? Windows CE-based systems, such as the WinPLC, come with a Think & Do Runtime license and have the Runtime software (Windows CE version) already loaded in its firmware. When you purchase a Think & Do version of a Windows CE platform, it already knows how to run Think & Do projects. Just select the station name of the Windows CE device and run. The PC used to develop the project downloads that project to the Windows CE engine to run. In this way, you can use one Think & Do software package on a Windows-based PC to develop projects for many Windows CE machines.

When you have completed the project for the target system, you can permanently disconnect the development system PC. The Windows CE system contains the Think & Do Runtime and the project – everything it needs to be a stand-alone controller.



When developing projects targeted to run on an off-the-shelf Windows-based PC, a Think & Do package must be purchased for each PC, just as each one must have a copy of the Windows 2000/XP/Vista operating system.

### 1.5.4.1 Local and Remote Systems

When developing an application and running it on the same PC, the target system is said to be the local PC. This configuration is the simplest to do and is shown in Figure 1-7. An I/O scanner card connects the PC to the I/O subsystem.

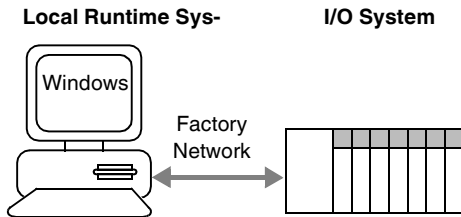


Figure 1-7 Local PC used for development and Runtime

In some factory situations, it may be necessary to access the PC-based control system from another location. The project engineer may create the application on a development system in an office environment, while the Runtime system exists some distance away in a manufacturing environment. The company network can connect the development system to the Runtime system as shown in Figure 1-8. In another common scenario, a project engineer will use a laptop as a local system and attach it to the factory-floor controller as a remote PC.

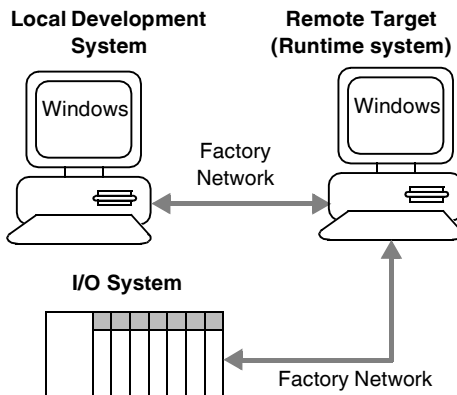


Figure 1-8 Development and Runtime control on different systems

When using remote and local systems, each PC platform must have Windows 2000/XP/Vista and be equally capable of running Think & Do. Each PC must also have its own installation (and license) of Think & Do.

Local and remote systems communicate using Windows built-in messaging called DCOM (Distributed Component Object Model). This is included as part of Microsoft's Windows 2000/XP/Vista operating systems. To implement a local/remote connection, both PCs must be configured for this connection. Just follow the instructions in page 8-23 in the "Think & Do User Manual".

## 1.6 Phoenix Contact Technical Service

If you have a question about the Think & Do and can't find the answer in this manual or the online help system, contact Phoenix Contact Technical Service by phone, fax, e-mail, or the Web. Our staff will give you the advice you need to get the most from the Think & Do.

We suggest that you try to duplicate the problem before calling Technical Service. During this process, write down each step that you perform and any error messages that appear. To obtain the best possible support, please be at your computer when you call, and have the following available:

- This manual.
- The version number on your Think & Do CD.
- The serial number on your license. You can view this by selecting “Start... Programs... Phoenix Contact... Think & Do... System... Hardware Key Information” to display the “Read Hardware Key” window.
- Windows version number and Service Pack level. To find this information, select the Control Panel, and double-click the System icon.
- Information about your computer hardware: type and model of computer, monitor, video card, I/O boards installed in the system, and the amount and type of installed memory.

PHOENIX CONTACT

Internet: [www.phoenixcontact.com](http://www.phoenixcontact.com)



Users outside the United States should contact their local sales office.

# Section 2

This section informs you about:

- Starting a new project
- Selecting a Runtime target
- Specifying inputs, outputs, and data items
- Basics on creating control programs and operator screens

Creating a Project.....	2-3
2.1 Starting a New Project.....	2-3
2.1.1 Creating a New Project .....	2-4
2.1.2 Naming a New Project .....	2-5
2.1.3 Opening a Project.....	2-6
2.1.4 Project Explorer Bar .....	2-7
2.1.5 Flow Chart and Screens Explorer Bars.....	2-8
2.1.6 Data Item Explorer Bar .....	2-9
2.1.7 Other Explorer Bars.....	2-9
2.2 Selecting a Runtime Target Type .....	2-9
2.2.1 Local/Remote .....	2-10
2.2.2 Selecting a Windows CE Target.....	2-10
2.3 Specifying Inputs, Outputs, and Data Items .....	2-12
2.4 Creating Control Programs.....	2-12
2.4.1 Action Block .....	2-13
2.4.2 Branching Block .....	2-13
2.4.3 Building Flow Charts .....	2-13
2.5 Creating an Operator Screen .....	2-14
2.5.1 HMI Overview.....	2-14
2.5.2 Screens as Part of a Project .....	2-14

## Think & Do

---

## 2 Creating a Project

In everyday life, a project usually refers to a series of tasks completed in order to accomplish a specific goal. Similarly, the goal of using Think & Do is to create a Runtime program that does whatever you design it to do. This is done by working on small, manageable pieces of the project with special tools within Think & Do. So, when using the development tools, you will be creating a project that consists of the following parts:

- A flow chart, or collection of flow charts
- An I/O configuration for the devices being controlled
- HMI screen(s) which provide an operator interface (optional for some applications or not available on some Windows CE-based devices)

The flow charts and screens must communicate with each other and the I/O to control and monitor the application. To make this possible, create a common tagname database of meaningful names. The database contains lists of system inputs, outputs, and internal variables.

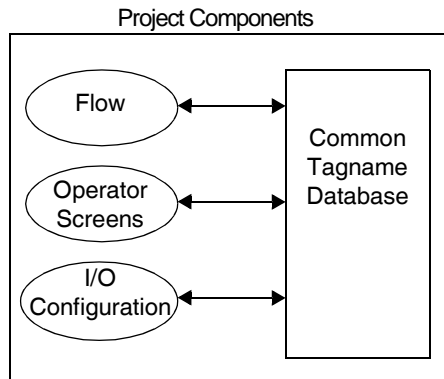


Figure 2-1 Project components

### 2.1 Starting a New Project

This section provides basic procedures for creating a project and a very simple flow chart program using Think & Do FlowView. It then shows how to create an operator screen, with ScreenView. Later chapters go into more detail on flow charts and screens.



### 2.1.1 Creating a New Project



ProjectCenter Icon

To create a new project, follow these steps:

1. From the Windows desktop, click “Start... All Programs... Phoenix Contact... Think & Do... ProjectCenter” to launch ProjectCenter or double-click the “ProjectCenter” icon on the desktop to start.
2. Select the “File... New” menu, and then take a moment to orient yourself to ProjectCenter (see Figure 2-2).

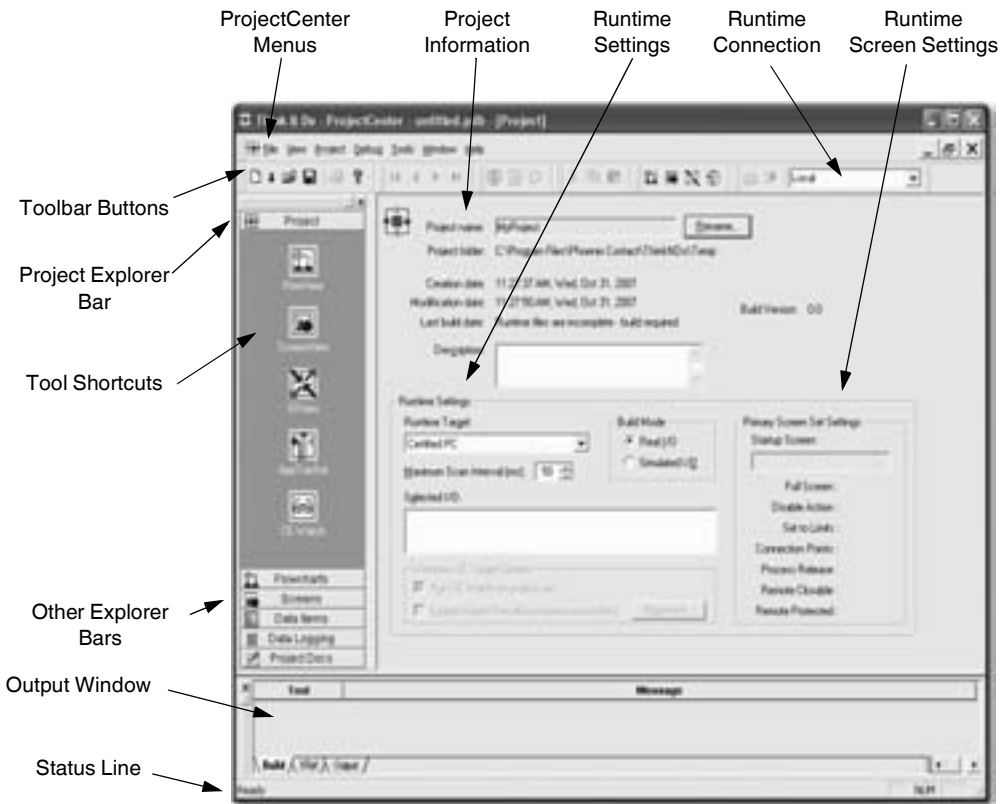


Figure 2-2 ProjectCenter provides access to Think & Do development tools

## 2.1.2 Naming a New Project

To name the project, follow these steps:

1. Click the “Rename” button next to the “Project Name” field. This displays the “Rename Project” dialog box shown in Figure 2-3.

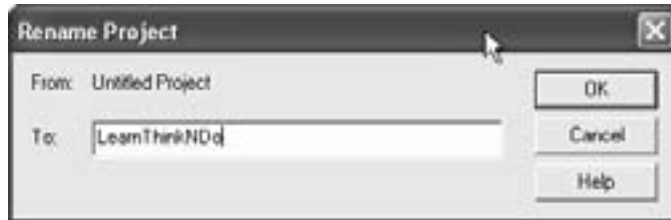


Figure 2-3 The “Rename Project” dialog box with a name filled in

2. Enter the name of the project in the “To” field.
3. Click the “OK” button.

To save the project, follow these steps:

1. In ProjectCenter, select the “File... Save As...” or “File... Save” menu, or click the “Save” button to display the “Save As” dialog box shown in Figure 2-4.

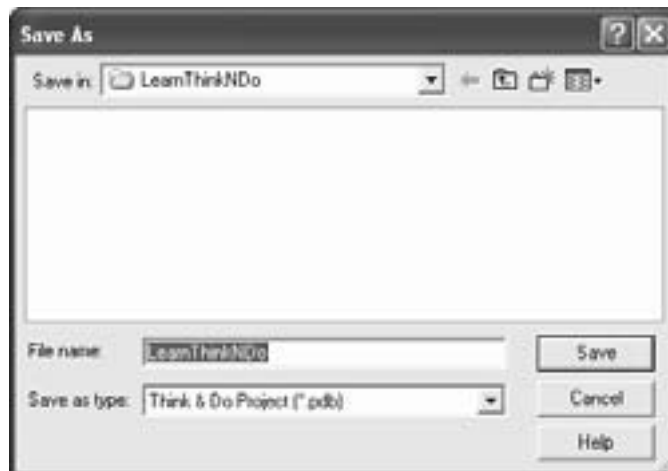


Figure 2-4 Use the “Save As” dialog box to define the project name

## Think & Do

---

2. To use the default path (Program Files... Phoenix Contact... Think & Do... Projects) and project name, simply click the “Save” button.



Renaming a new project before saving it automatically creates a new directory with the name of the project.



Only one project can be stored in a directory.



Create New Folder Button

3. To use a different directory, use the “Save In” drop-down to navigate to the desired drive and directory. Create a new directory by clicking the “Create a New Folder” button. When the new folder appears in the “Save As” list, type the desired name and press the <Enter> key. Then double-click the folder to open it.
4. To change the name of the project, enter a new name in the “File name” field. Note that only alphanumeric characters are valid.
5. Click the “Save” button to save the project.

### 2.1.3 Opening a Project



Open Button

To open a project, follow these steps:

1. In ProjectCenter, select “File... Open...” or click the “Open” toolbar button to display a file selection browser.
2. Navigate to the folder that has the “.pdb” file.
3. Select the file and click “Open” or double-click the file in the browser.

## 2.1.4 Project Explorer Bar

Click the “Project” Explorer bar (at top of pane) to view the Think & Do tool shortcuts. Click any of these shortcut icons to directly access the Think & Do tools (see Figure 2-5).

- FlowView
- ScreenView
- IOView
- AppTracker
- CE Watch



Figure 2-5 The “Project” Explorer bar provides access to Think & Do tools

### 2.1.5 Flow Chart and Screens Explorer Bars

Click the second and third Explorer bars to view the project's flow charts and screens. The initial folders name the categories that can be expanded after flow charts and screens exist, listing them in a tree structure for project management.

Flow Charts

Screens



Figure 2-6 The “Flow Charts” and “Screens” Explorer bars

### 2.1.6 Data Item Explorer Bar

The “Data Item” Explorer bar, fourth on the list, shows folders categorizing project data items (or variables). For a complete discussion of data item types, see page 7-6 in the “Think & Do User Manual”.



Figure 2-7 The “Data Items” Explorer bar

### 2.1.7 Other Explorer Bars

The “Data Logging” and “Project Docs” Explorer bars provide additional capability discussed in later chapters.

## 2.2 Selecting a Runtime Target Type

When creating a project, it is essential to select the type of target for the project at Runtime. For many users it will be the same PC on which they are developing the project. Others may have remote or embedded controllers.

To select the Runtime target:

1. Click the “Project” Explorer bar (see Figure 2-2 on page 2-4). The project information displays in the main ProjectCenter window.
2. In the “Runtime Settings” group (see Figure 2-2 on page 2-4), choose the proper Runtime target from the drop-down list.
3. If you select one of the “Windows CE” options, there are additional available options. You can choose to have CE Watch run automatically whenever the project runs (see page 9-9 in the “Think & Do User Manual”). There’s also an option to “Enable Instant Recall” (see page 9-12 in the “Think & Do User Manual”).

### 2.2.1 Local/Remote

If you select “Certified PC” as the Runtime target, ProjectCenter lists “Local” (the default name for the current PC) in the “Runtime Connection” drop-down menu (see Figure 2-2 on page 2-4). To use a remote PC for the Runtime, type the name of that PC in the Runtime target field. (For more information on connecting to a remote PC, see page 8-23 in the “Think & Do User Manual”).

### 2.2.2 Selecting a Windows CE Target

One or several Windows CE systems may be on the network, connected to the local PC. When several systems are attached, ProjectCenter can only target one at a time. Think & Do interacts with the proper device for project uploads/download, project run/halt actions, etc.



The PC must have a network configuration with TCP/IP protocol, a network adapter card, and special network settings. Otherwise, the procedure that follows may not work. Please refer to the Getting Started booklet that accompanied the Windows CE hardware for instructions. It may also be necessary install the NWLink IPX/SPX protocol if the target device cannot be located and must be reset to establish a compatible IP address.

To select a Windows CE target:

1. Choose one of the “Windows CE” options in the “Runtime Target” field. The “Runtime Connection” field in the toolbar will display “Find CE Runtime...”
2. Click the drop-down arrow and then click “Find CE Runtime...” from the drop-down list.



If the network is disconnected or power is removed from target devices, the “Find CE Runtime” may display the “Target Picker error” dialog box. If that occurs, check the network connections and power supply to target device(s).

3. The “Find/Configure CE Runtime” dialog box, shown in Figure 2-8, appears. Review the list of all Windows CE devices found on the network.

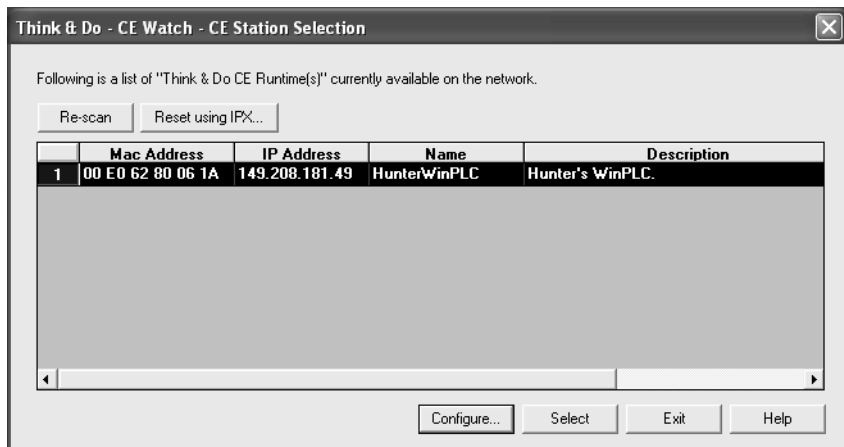


Figure 2-8 The “Find...Configure CE Runtime” dialog box

For example, Figure 2-8 lists a WinPLC at a particular network IP address, which Think & Do uses to identify the device.

4. Highlight the desired Windows CE target device in the list.
5. Click the “Select” button to choose the selected target device and exit the dialog box.



The “Find/Configure CE Runtime” dialog box in Figure 2-8 provides utilities for resetting and configuring Windows CE Runtime targets. The IP address is user-configurable, but the MAC address (media access control) is a unique hardware address for each device that cannot be changed. Please refer to the Windows CE device documentation for details on how to reset and configure the IP address.



The name of the selected target will appear in the list box for the “Runtime Target” field (see Figure 2-2 on page 2-4). Later, when the project runs, Project-Center automatically sends compiled programs to run on the selected target.

### 2.3 Specifying Inputs, Outputs, and Data Items



Think & Do uses a separate application, IOView, to define a project’s inputs, outputs, and data items. To launch IOView, select “Tools... IOView” or click the “IOView” icon in the “Project” Explorer bar.

The IOView defines the drivers in the project and shows them graphically. For many drivers, IOView can automatically detect and display connected devices. For more information on IOView, see Section 3, “Drivers, Devices, and Tags”.

### 2.4 Creating Control Programs

Think & Do uses flow chart control programs developed in FlowView. A flow chart is a type of diagram that uses symbols to illustrate a control process. The symbols in the flow chart depict two classes of items, action blocks and branching blocks (see Figure 2-9). These are defined in the following sections.

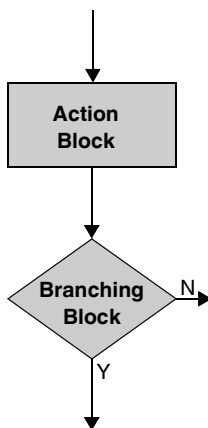


Figure 2-9 Flow chart showing an action and branching block

### 2.4.1 Action Block

A box representing an operation on the data in the system, which may include input or output. In the flow chart, an action block has one entry point (top) and one exit point (bottom).

### 2.4.2 Branching Block

A diamond-shaped box represents a branch in the control path based on available data. The decision may compare one variable to another, a variable to a constant, or a true or false condition. Branching blocks have one entry point (top) and two exit points (either side and bottom, or both sides).

With just branching and action blocks, a program can have complete command of the I/O sub-system, or control a machine or process. A project may consist of one or several flow charts.

### 2.4.3 Building Flow Charts

Use FlowView to create flow charts. To create a new flow chart follow these steps:

1. From the ProjectCenter, launch FlowView using one of the following techniques:
  - Select “Tools... FlowView.”
  - Click the “FlowView” icon in the Project Explorer bar.
  - Select “File... New... Flow Chart” to launch FlowView, the flow chart drawing tool. If using this technique, continue with Step 3.
2. In FlowView, select “File... New... Flow Chart”.
3. In FlowView, the “Name Flow Chart” dialog box appears (see Figure 2-10). Type a name for the flow chart.



FlowView Icon



Figure 2-10 “Name Flow Chart” dialog box

4. Click the “OK” button.

For more information on how to use FlowView, see Section 4, “General Programming Techniques”. For a complete description of flow chart programming, see “Flow Charts” on page 5-3.

## 2.5 Creating an Operator Screen

Think & Do provides more than just PC-based control with flow charts – it includes a full-featured operator interface or HMI (Human-Machine Interface). Use ScreenView to create operator screens.



Windows CE target Runtime systems – some devices, such as the WinPLC, do not have direct screen output available to its operating system.

### 2.5.1 HMI Overview

The final output of the development work is a screen or series of screens. These appear in a window on the PC at Runtime. The purpose of a screen is to communicate important information to a machine operator or process engineer. It also provides an opportunity for operator-controlled input. Screens typically include:

- The current state of the machine or process (such as Idle, Run, or Stop)
- Alarm conditions, if they exist
- Production data (such as number of parts, percentage defects, downtime reasons, and trends)
- Diagram of process (on-screen diagram to help orientate the operator)

ScreenView is a full-featured HMI or operator interface design package, which helps create screens that really communicate. In addition to standard drawing constructs, the screen object shape toolbars provide an ability to create dynamic objects (change color, size, position, or rotate) and active objects (alarms, trend controls, ActiveX controls, and .NET controls). Operator input support includes function keys and data entry with text fields, check boxes, radio buttons, push buttons, and drop-down selection lists.

### 2.5.2 Screens as Part of a Project

The screens developed using ScreenView are part of a project. At Runtime, the screens appear in an independent window on the PC desktop. Only one screen is visible in the window at a time. You can resize or minimize HMI screens, just as any window. However, you cannot exit or quit screen windows (the Runtime

must be stopped instead, which shuts down the operator screens in the project). Data entry or function key entry requires that the screen be in the active window.



When a project runs, its flow charts are the highest priority task – not screens. During the “Pause” phase of each scan period, the system executes screen tasks and other applications. Adding screens affects system loading, so be sure the scan time setting is large enough to allow the Runtime to service operator screen(s).

To create a new screen in the project:

1. In ProjectCenter, click “File... New... Screen” to open ScreenView and display the “Screen Attributes” dialog box (Figure 2-11).

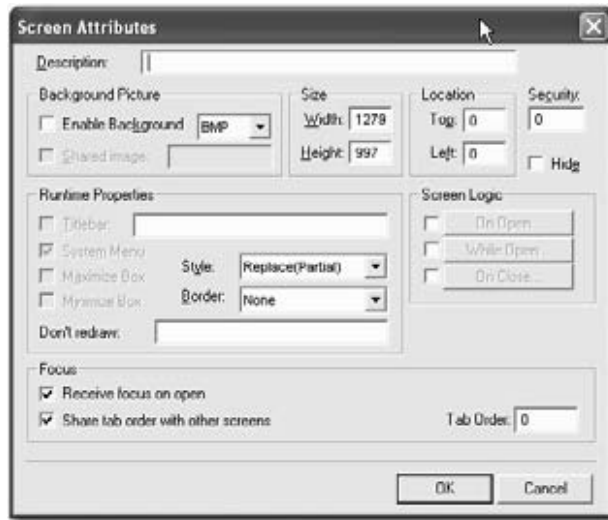


Figure 2-11 “Screen Attributes” dialog box

2. Enter a description for the screen, and make any other required changes.
3. Click “OK” to create the screen.

For more information on how to use ScreenView, see “General Programming Techniques” on page 4-3. For more information on HMI screen development, see page 6-3 in the “Think & Do User Manual”.

## Think & Do

---

# Section 3

This section informs you about

- Specifying inputs, outputs, and data items
- Configuring I/O
- Adding I/O drivers and devices
- Mapping I/O
- Scanning and monitoring I/O, saving the I/O configuration
- Using a watchdog timeout

Drivers, Devices, and Tags .....	3-3
3.1 Specifying Inputs, Outputs, and Data Items .....	3-3
3.1.1 Configuring External I/O .....	3-3
3.2 Configuring I/O .....	3-6
3.3 Adding an I/O Driver .....	3-8
3.4 Connecting to I/O Devices .....	3-10
3.4.1 Manually Adding an I/O Module .....	3-12
3.4.2 Editing an I/O Module .....	3-12
3.4.3 Deleting an I/O Module .....	3-12
3.5 Mapping I/O Points to Data Items .....	3-13
3.6 Scanning and Monitoring I/O .....	3-15
3.7 Saving the I/O Configuration .....	3-16
3.8 Tips on Toolbar Functions .....	3-17
3.9 Watchdog Timeout Default I/O States .....	3-19
3.9.1 Startup/Shutdown Output State .....	3-20

## Think & Do

---

## 3 Drivers, Devices, and Tags

This section discusses the techniques required to define input, output and internal data items used in control programs and HMI screens. Think & Do uses a common tagname database for all I/O and variables throughout the project.

### 3.1 Specifying Inputs, Outputs, and Data Items

All variables in the control system may be classified into two simple categories:

- External: Inputs and Outputs or any tagnames mapped to external I/O. These may include I/O bases, motion axes, or serial port devices.
- Internal: internal variables for storage, calculations, and manipulation

All data items can have meaningful tagnames of up to 30 characters. All data types are for internal use in a Think & Do project. However, you can map Inputs, Outputs, Counters, Numbers, Floats, Strings, and Comm data items to external I/O devices.



The space character is not valid for tagnames. This ensures that tagnames comply with common standards to transfer data items across networks and between different applications.

#### 3.1.1 Configuring External I/O

Think & Do has many available software drivers that provide the connection between the PC and I/O network or scanner cards. Each card has a corresponding driver. An I/O network or scanner card may have one or more nodes. For example, the Momentum Ethernet and DeviceNet cards support multiple nodes, each with a single *module*. A module is either an I/O base (rack) or a field device. Some network cards may have nodes that support multiple modules. For example, Seriplex and INTERBUS cards have a single node that supports multiple modules. Finally, there are scanner cards that support multiple nodes with multiple modules on each node, such as Automationdirect.com Ethernet and Opto 22 Optomux I/O.



## Think & Do

---

### Adding Data Items

To add new data items:

1. In ProjectCenter, click the “Data Items” Explorer bar. The “Data Types” folder tree appears in the “Data Items” Explorer bar (Figure 3-1).



Figure 3-1 ProjectCenter Data Items

- 2. Click the data type folder for the data type desired to display the Data Item grid. The Data Item grid shown in Figure 3-2 is for Counter data types. The Data Item grid has a fixed “C” (for Counter) in the left-most column of each row.

Index	Tagname	Initial	Retentive	Description	Mapped to...
C 1	WhiteAccumulationCounter	0	<input type="checkbox"/>		
C 2	GreenAccumulationCounter	0	<input type="checkbox"/>		
C 3	RedAccumulationCounter	0	<input type="checkbox"/>		
C 4	YellowAccumulationCounter	0	<input type="checkbox"/>		
C 6	WhiteDiverterPosition	7	<input type="checkbox"/>		
C 7	GreenDiverterPosition	12	<input type="checkbox"/>		
C 8	RedDiverterPosition	10	<input type="checkbox"/>		
C 10	WhitePackageID	1	<input type="checkbox"/>		
C 11	GreenPackageID	2	<input type="checkbox"/>		
C 12	RedPackageID	3	<input type="checkbox"/>		
C 23	YellowPackageID	4	<input type="checkbox"/>		
C			<input type="checkbox"/>		

Figure 3-2 The Data Items grid for Counter data types

- 3. In the main Data Item grid, type the tagname desired. Press the <Enter> key or click outside the tagname field to enter.



By default, Think & Do increments the Index as tagnames are entered in the Data Item grid. You can, however, change these values to non-sequential numbers. The data type and index uniquely identify each tagname throughout Think & Do.



Save Button

- 4. Click the “Save” button in the toolbar or select “File...Save” to save the data items added.

**Deleting Data Items**

To delete a row in the Data Item grid:

- 1. Select a row – Move the mouse cursor to the left-most column in the row to delete. The cursor changes to a right-pointing arrow.
- 2. Click to select the entire row, which will then be highlighted.
- 3. Press the <Del> key to delete the row.



You cannot cut-and-paste an entire row because tagnames must be unique within a project.

### Editing Text in the Data Item Grid

To copy and paste text in Data Item grid cells:

1. Select the text within a cell.
2. Click the “Copy” toolbar button.
3. Click a different cell and then click the “Paste” toolbar button.

## 3.2 Configuring I/O

Use the IOView to map data items to I/O points. To launch IOView, follow these steps:

1. In ProjectCenter, click the “Project” Explorer bar (left-most pane, top bar) to view the tool shortcuts.
2. Click the IOView icon to launch IOView.



IOView Icon

IOView is a separate tool (not a window in ProjectCenter). Using it produces a single I/O configuration for the project that Think & Do stores with all other project components.

The IOView window (see Figure 3-3) draws a picture of the I/O network as you configure it. After configuration, the I/O network or scanner card(s) in the PC are in the upper-left pane.

I/O network and scanner cards connect to *modules*. Modules can be either I/O bases or field devices. Modules appear in the upper-right pane. Configuration information appears in the bottom pane of the window as modules are select modules and different tabs at the bottom of the pane.

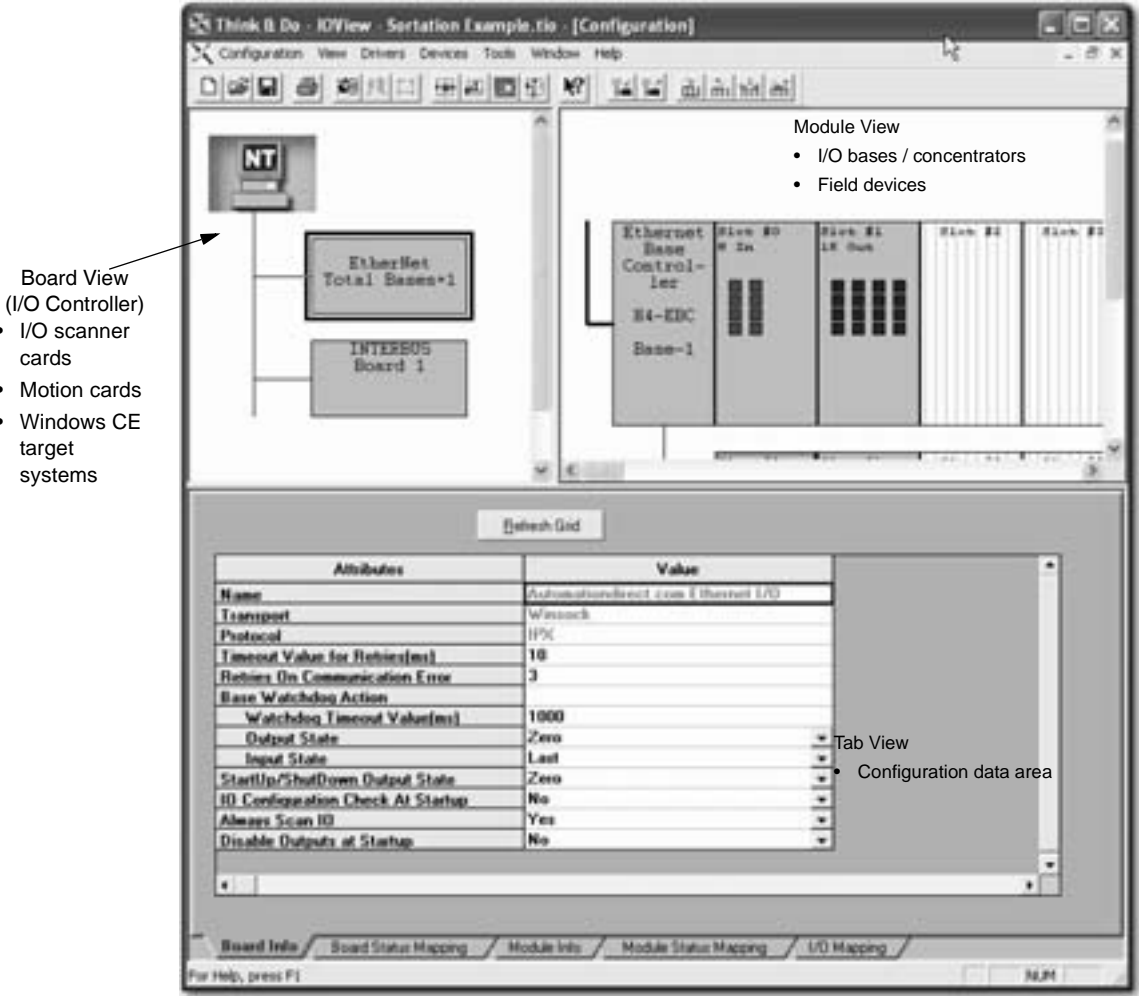


Figure 3-3 IOView window includes a board view, module view, and tab view

The general steps required to configure and map an I/O network are:

1. If a scanner card is available, install it in the PC along with the required driver software that came with the card.
2. Add the appropriate driver for the I/O network or scanner card associated with the Board View I/O controller (the I/O controller may be an I/O scanner or network card, or a Windows CE Runtime target).
3. Do one of the following:
  - If I/O devices are available:
    - a. Connect the scanner card to the I/O devices on the network and build a physical I/O network.
    - b. Use Auto-Discovery to find all devices on the network.
    - c. Monitor inputs with IOView in real time and turn on the outputs by clicking on the graphic.
  - If I/O devices are not available or the Think & Do driver doesn't support Auto-Discovery, generate the configuration manually using these steps:
    - a. Add the device or module manually, as described in "Manually Adding an I/O Module" on page 3-12.
    - b. Map the input points to tagnames.
    - c. Map the output points to tagnames.
4. Save the I/O configuration.

### 3.3 Adding an I/O Driver

You must add a driver for the I/O network, scanner card, or backplane (also called an I/O adapter) connected to the Runtime target. The driver is a special piece of software that Think & Do uses to communicate over the I/O link to physical devices.



Add Driver  
Button

To add an I/O driver:

1. Either select “Drivers...Add” or click the “Add Driver” toolbar button.
2. From the list of I/O drivers (Figure 3-4), select the appropriate one for the I/O connected to the target hardware.

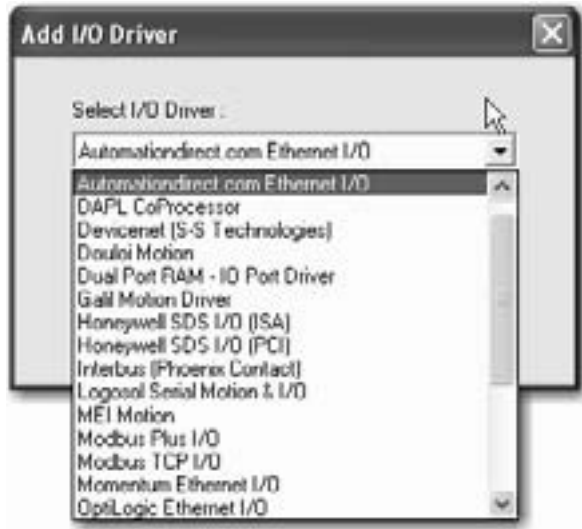


Figure 3-4 The “Add I/O Driver” dialog box and its driver listing



Phoenix Contact is constantly adding new drivers and device types to stay current with the industry's latest products. Visit [www.phoenixcontact.com](http://www.phoenixcontact.com) to check the latest news on supported I/O.

3. The next dialog box may prompt for board number, serial port number, etc., depending on the I/O type.
4. Click the “OK” button.

IOView displays an image of the I/O adapter as shown in Figure 3-5 (multiple cards appear in a tree structure). IOView also displays board information related to the I/O card in the lower pane of the window, where you may edit some

## Think & Do

of the parameters. Generally, it's a good idea to leave all of the parameters at their default settings, unless you know for certain that they require changing (see online help for details).

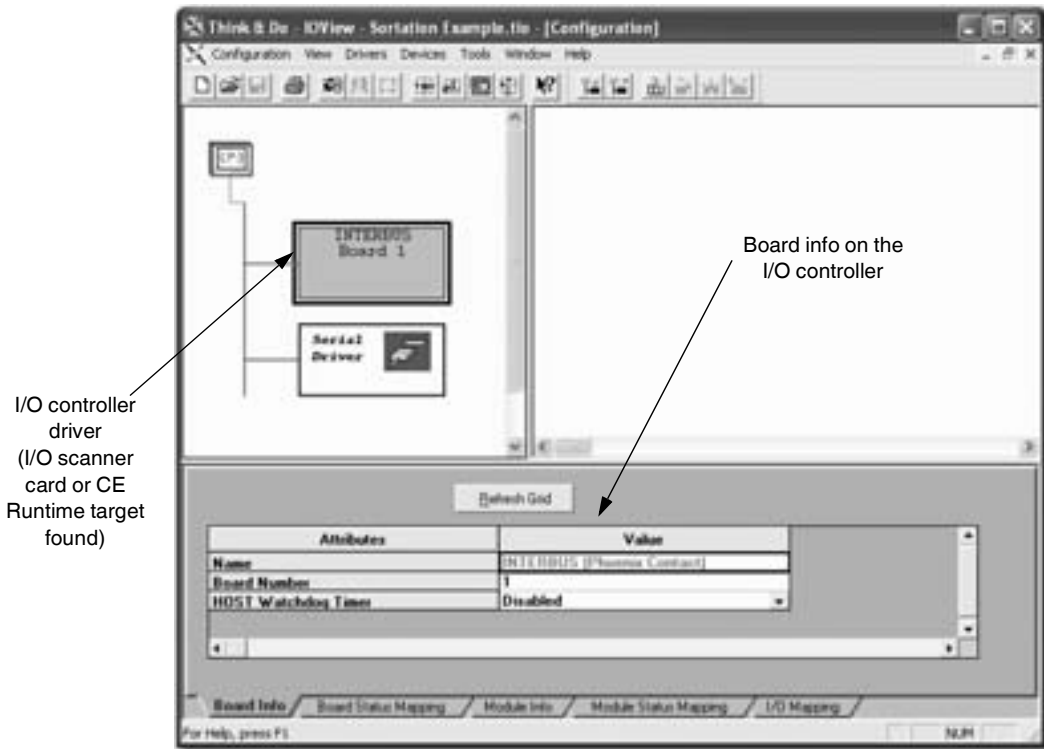


Figure 3-5 IOView displays an image of the I/O scanner card or Runtime target

### 3.4 Connecting to I/O Devices



Connect Button

After adding an I/O driver to the configuration, you can connect it to actual I/O, if it exists and is connected to the I/O network or scanner card in the PC.

To connect to an I/O system:

- From the “Configuration...Connect” menu (or use the “Connect” toolbar button).

IOView instructs the I/O adapter card to connect to the bases or devices on the network immediately. It reads the status (I/O count and type) from all devices in order to create an image of the physical I/O. Note that not all I/O networks support this capability.

When IOView connects to the I/O, it creates an image similar to the one shown in Figure 3-6. This example matches a DL205 base, which uses an Ethernet Base Controller in the left-most slot. The modules adjacent to it include an 8-point input simulator (F2-08SIM) and an 8-point relay output module (D2-08TR).

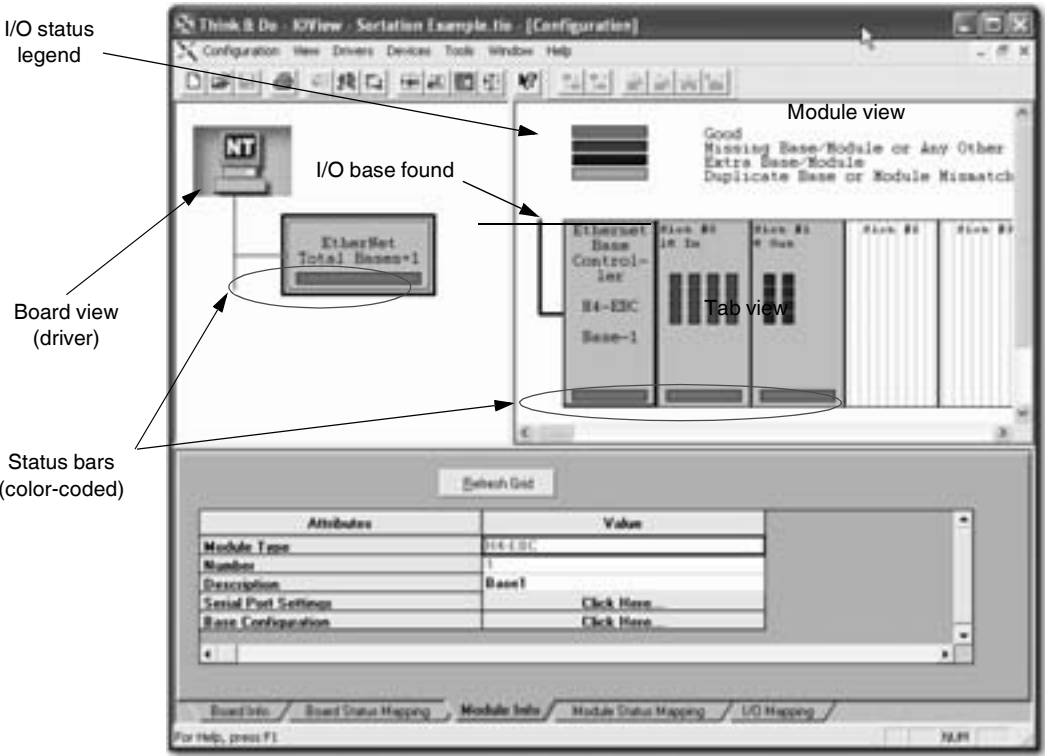


Figure 3-6 IOView showing the "Module Info" tab



### 3.4.1 Manually Adding an I/O Module



Add Device  
Button

To manually add an I/O module (or device), follow these steps:

1. Select the driver in the Board view of the IOView.
2. Click the “Devices...Add” menu (or click the “Add Devices” toolbar button) to display the “Add/Insert Device Type” dialog box. This dialog box has one or more drop-down lists depending on the type of driver the device connects to.
3. Choose the device from the drop-down list and make any other selections required.
4. Click the “OK” button to display a picture of the device/module in IOView.

### 3.4.2 Editing an I/O Module

To select a module and view its I/O configuration, follow these steps:

1. Click the module image in the module view pane of IOView. When selected, the module outline has a red border.
2. Click the tabs in the tab view area to see information about the selected module.

The four colored bars above the I/O base form a legend to help interpret the indicator status bars located on each module. Each module’s status bar is green if the selected module is okay.



The image of the base in the figure has more slots than the base which is used in the I/O system. IOView uses the maximum slot count when the base status information does not include slot count.

### 3.4.3 Deleting an I/O Module



Remove  
Device Button

To delete an I/O module (or device), follow these steps:

1. Select the device or module in IOView.
2. Select the “Devices...Remove” menu (or click the “Remove Devices” toolbar button) to delete the device or module without any confirmation.

### 3.5 Mapping I/O Points to Data Items

After selecting the input module, click the “I/O Mapping” (right-most) tab at the bottom of the View window. The tab view (Figure 3-7) displays the I/O mapping information for the selected module above it.

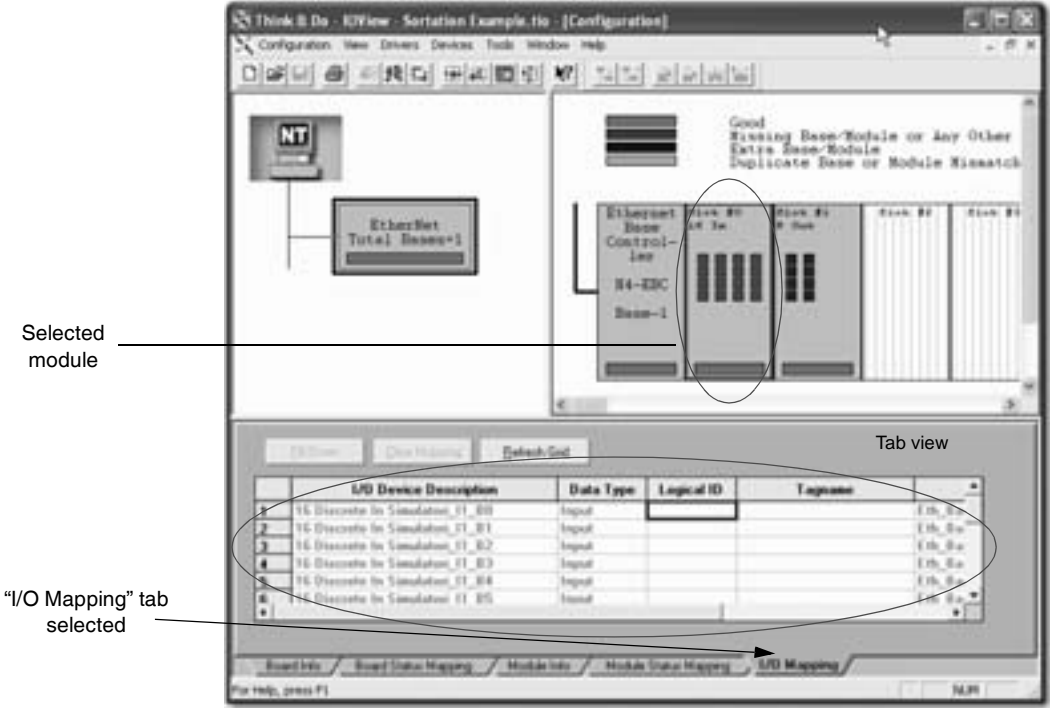


Figure 3-7 IOView showing the "I/O Mapping" tab for selected module

The “I/O Mapping” tab of the IOView window presents information about the selected module in a spreadsheet format.

The column titles and their meanings are:

- “I/O Device Description”: Each bit in an I/O module has a table entry and a descriptive prefix that can be user-defined in the “I/O Device Description” field in the “Module Info” tab.
- “Data Type”: The data type corresponds to the classification of the tag-name in the Data Item grid in ProjectCenter.
- “Logical ID”: This is the number the common database uses to access each I/O point. The next section describes how to map Logical IDs.

## Think & Do

- “Tagname”: The name of the input or output point. This is assigned in ProjectCenter’s Data Item grid, but IOView displays it here when I/O points are mapped to data items.
- “Physical I/O”: Lists the type of I/O network, base number, slot number, and bit number.

The I/O configuration is determined by mapping physical I/O points to logical ID numbers or tagnames (Figure 3-8). In this example, the first four connections go straight across, but the last four connections cross each other. You can also choose a straight-through mapping or re-map I/O points by changing the relative order of logical ID (such as I-0, I-1, I-2, I-7, I-4, etc.) to the physical I/O. You can assign unique tagnames to each logical ID or group of IDs (depending on data type).

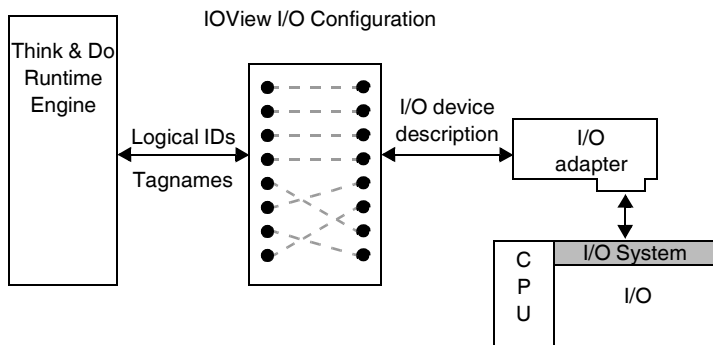


Figure 3-8 Physical I/O points map to logical I/O points in any sequence

To map the input or output module points:

1. Select a module in IOView.
2. In the “Tab” panel, select the “I/O Mapping” tab.
3. Click the cursor in the “Logical ID” column of the first row.

4. Type the logical ID number desired to define (for example, if an input module is selected, you might enter **0** as the first logical ID), and then press the <Enter> key. The Logical ID (following the example, **I-0**) appears, along with the corresponding tagname defined in ProjectCenter (see “Adding Data Items” on page 3-4).



A convenient way to map I/O points is to double-click any cell in the “Logical ID” or “Tagname” column, and IOView displays the Data Item grid in ProjectCenter to selection of the desired tagname from the tagname database. To automatically copy a logical ID and tagname to the I/O mapping table, double-click the first column of the “Data Item” table, or select the row and click the “OK” button in the dialog box.

Since many projects use a simple, straight-through mapping, IOView provides an auto-complete function, called Fill Down. The Fill Down feature automatically enters sequential, logical IDs and tagnames.

To use Fill Down to complete I/O mappings:



Fill Down Button

1. Enter the first Logical ID number in the top cell of the Logical ID column.
2. Select the entry in the top cell in the Logical ID column.
3. Click and drag the cursor downward, releasing it over the bottom row. This selects the remaining logical ID cells and highlights them in black. This enables the “Fill Down” button that appears above the grid.
4. Click the “Fill Down” button. IOView automatically fills in the selected cells, incrementing the logical ID number for each cell. The associated tag-names appear automatically.

Think & Do ensures tagname changes are only made once. For example, changing a tagname in ProjectCenter’s Data Item grid is reflected automatically in IOView or in any other part of Think & Do.



Edit the default “I/O Device Description” field to create a meaningful input/output point name for the application (located in the “Module Info” tab).

### 3.6 Scanning and Monitoring I/O



Connect Button

IOView has the ability to monitor actual inputs and turn on outputs to test wiring independent of control logic.

To monitor I/O from IOView:

1. Click the “Configuration... Connect” menu, or use the “Connect” toolbar button.

## Think & Do



Scan Button

2. Click “Configuration... Scan” menu, or use the “Scan” toolbar button.

To check an input point:

1. Turn an input ON or OFF.
2. Watch the on-screen image of the input module as the physical input changes to monitor its status.

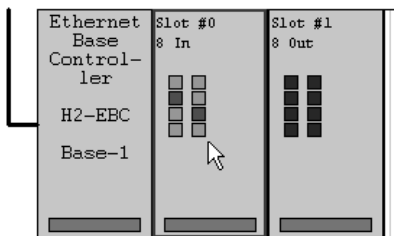


Figure 3-9 The on-screen images in IOView reflect input and output states

Turn on outputs by clicking the LED on the graphic (available in several I/O drivers). A warning message appears before the actual I/O update.

Set analog values by clicking on the graphic of the module and channel. IO-View displays a dialog box to input the desired value for the output channel.

### 3.7 Saving the I/O Configuration



Disconnect Button

After configuring and checking the status of I/O, it's a good idea to save the I/O configuration for the project.

To save a newly scanned configuration:

1. Disconnect the I/O by selecting the “Configuration...Disconnect” menu or clicking the “Disconnect” button. (This automatically makes I/O scan inactive).
2. The “Save” dialog box appears. Click the “Yes” button to save the configuration with the project.

Each project can have only one I/O configuration associated with it. Edit an I/O configuration by using the “Devices” menu selections. Click the “Save” menu button after any I/O configuration edits.

### 3.8 Tips on Toolbar Functions

The toolbar in IOView includes special-purpose buttons for convenience.

- The buttons “Connect,” “Disconnect” or “Scan” access the I/O directly from the toolbar.

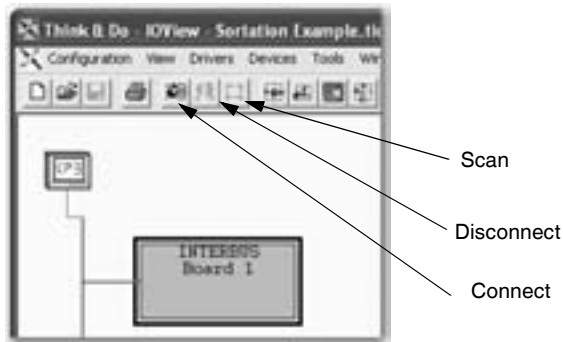


Figure 3-10 Connection-related buttons available in the toolbar

- Connect and Disconnect are opposite (and alternately available) functions.
- Scan works only when the Connect function is active.
- A Disconnect while Scan is active automatically makes Scan inactive.

## Think & Do

---

The second set of toolbar buttons make off-line editing easier, when either the devices are not available to scan or the driver does not support scanning.

- The “Add Driver” and “Remove Driver” buttons configure I/O adapters, motion control boards, or serial ports in the target Runtime.

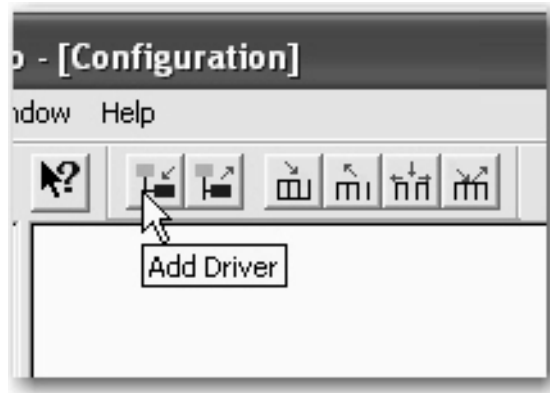


Figure 3-11 “Add” and “Remove” driver buttons

- Each driver's I/O controller is shown in the Board view pane of IOView in a tree diagram.

- The remaining buttons on the toolbar perform the Add Device, Remove Device, Insert Device, and Replace Device functions. For most I/O these buttons can configure a desired I/O network for the project before the actual devices are connected to the PC.

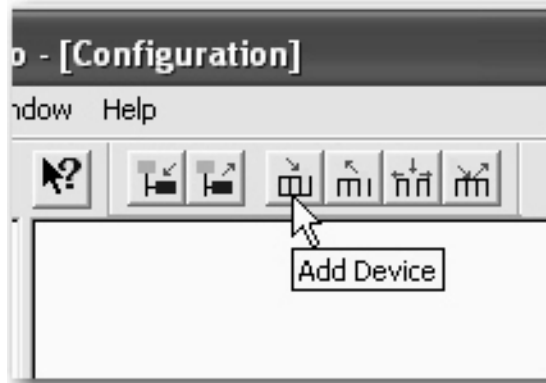


Figure 3-12 “Add,” “Remove,” “Insert” and “Replace” device buttons

- Undocking a toolbar creates a floating window of buttons by clicking and dragging on an empty part of the toolbar. You can dock a toolbar window on any window border by dragging the window to the border and releasing the mouse button after the window snaps into place.

### 3.9 Watchdog Timeout Default I/O States

For most projects, the actual state of I/O points during a communications failure or during startup and shutdown is a major concern. Many I/O devices support special features to specify the state of the I/O at these critical times with a specified communications watchdog timer action. IOView provides complete access for configuring preferences for these situations.

To configure watchdog timeout states:

1. Click the module graphic for the desired module to configure.
2. Click the “Board Info” tab and note the “Output State” or “Input State” attributes in the “Watchdog Action” group. The default settings are “Zero” (off) for outputs and “Last State” (like a latch) for inputs. The accompanying list boxes select “Zero”, “Last State”, or “Pattern” for inputs and outputs.



### 3.9.1 Startup/Shutdown Output State

This attribute defines what the outputs will do during startup or shutdown periods. For example, the Ethernet base controller supports two modes:

#### FailSafe

If the network communication times out between the Runtime and the base controller, you can configure output and input states.

#### Startup/Shutdown

This feature defines the on-off pattern used for outputs when the Runtime starts and stops.

Choosing a pattern as an I/O state permits a combination of on/off states for input or output modules. This requires further definition of the pattern for each module in the base. Select the “Module Info” tab, and note the Fail-safe I/O Pattern attribute, with the “Click Here...” prompt.



The pattern specified applies to either or both “Watchdog time-out” and “Startup/Shutdown” categories, if pattern is the specified I/O state.

To configure an I/O Pattern, follow these steps:

1. Use the “Click Here...” button to access the Pattern dialog box. The entry field is in hex format and Think & Do decodes and displays the value in binary (MSB to LSB, left to right) to confirm individual I/O point states.
2. Click the “OK” button when finished.

# Section 4

This section informs you about

- General techniques for creating and editing flow charts and subcharts
- General techniques for creating and editing operator screens

General Programming Techniques .....	4-3
4.1    Flow Charts and Subcharts .....	4-4
4.2    Exploring Flow Charts and Subcharts .....	4-5
4.3    Creating a Flow Chart .....	4-6
4.4    Changing Flow Chart Settings.....	4-7
4.4.1    Changing Execution Order .....	4-7
4.4.2    Changing a Flow Chart's Category .....	4-8
4.4.3    Renaming, Copying, Deleting, or Printing Flow Charts .....	4-8
4.5    FlowView Orientation .....	4-9
4.5.1    Adding Flow Chart Blocks to a Flow Chart .....	4-10
4.5.2    Turning the Grid On and Off .....	4-11
4.5.3    Increasing the Number of Columns .....	4-11
4.5.4    Changing Zoom Level .....	4-12
4.6    Displaying Multiple Flow Charts .....	4-13
4.7    Editing Flow Charts .....	4-14
4.7.1    Cut-Copy-Paste.....	4-14
4.8    Flow Chart Block Introduction .....	4-15
4.8.1    Enable Block .....	4-15
4.8.2    Freeze Block .....	4-15
4.8.3    Types of Action Blocks .....	4-15
4.8.4    Freeze Block .....	4-16
4.8.5    Notes.....	4-18
4.9    Connecting Flow Chart Blocks .....	4-18
4.10   Editing Flow Chart Block Expressions.....	4-20
4.11   Entering Block Description .....	4-20
4.12   Operator Screens.....	4-21
4.12.1   Screens as Part of a Project .....	4-21
4.12.2   Creating Operator Screens .....	4-22
4.12.3   ScreenView Development Environment .....	4-23
4.12.4   Object Properties Dialog Box .....	4-26

## Think & Do

---

4.12.5	Creating and Using Tags in ScreenView .....	4-32
4.13	Scripting Languages in ScreenView.....	4-34

## 4 General Programming Techniques

Think & Do uses flow chart control programs. The symbols in the flow chart depict two classes of items: action blocks and branching blocks.

### Action Block

An action block is a box (see Figure 4-1) representing an operation on the data in the system, which may include input or output data. In the flow chart, an action block has one entry point (top) (top, left, or right) and one exit point (left, right, or bottom). Think & Do has four types of action blocks that are described later (see “Types of Action Blocks” on page 4-15).

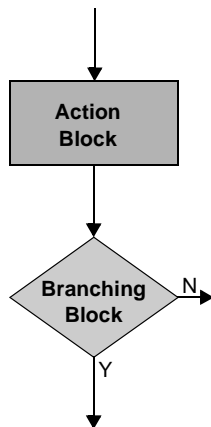


Figure 4-1 Flow chart segment showing an action and branching block

### Branching Block

A branching block is a diamond-shaped box (see Figure 4-1) that represents a branch in the control path based on available data. A branching block may compare one variable to another, a variable to a constant, or a true or false condition. Branching blocks have one entry point (top) and two exit points (either side and bottom). Think & Do has two types of branching blocks that are described later (see “Compare Block” on page 5-11 and “Decision Block” on page 5-17).

Using just action and branching blocks, flow charts can have complete control of I/O systems, or control a machine or process. A project may consist of one or several flow charts.

## 4.1 Flow Charts and Subcharts

Flow chart programs in Think & Do provide two chart types for hierarchical structure: *Standard* flow charts and *Subcharts*. Their relationship is similar to main program and subroutine in BASIC language programs. A flow chart Call block calls a subchart, which transfers execution to the subchart. The subchart does a return to the original flow chart, transferring execution back again (Figure 4-2).

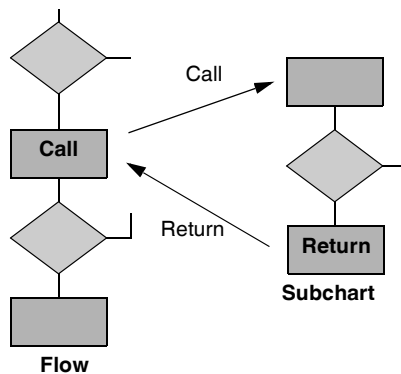


Figure 4-2 A Call block calls a subchart, which returns to the original flow chart

Using subcharts with flow charts provides great flexibility in project design. A flow chart can pass variables to a subchart through parameters you define. Then the subchart can pass results back to the calling flow chart. Well-designed subcharts are generic, making them easy to re-use in future projects. This saves time in future project development. This aspect of project design is covered in detail in “Flow Charts” on page 5-3.

## 4.2 Exploring Flow Charts and Subcharts

In the ProjectCenter “Project” Explorer bar, click the “Flow Charts” Explorer bar to see a list of the project’s flow charts (see Figure 4-3).



Figure 4-3 The “Project” Explorer bar showing the Flow Chart explorer

Folders group flow charts by categories. The categories are not necessarily exclusive, meaning that a flow chart can appear in multiple folders. The categories are:

- “Standard 123...”: lists flow charts in the order that they are executed during each scan.
- “All abc...”: lists both flow charts and subcharts together in alphabetical order.
- “Simulation”: lists simulation flow charts (can write to input points).
- “Subcharts”: lists the project’s subcharts (if any) in alphabetical order. Note that the actual order of execution is determined by when they are called.
- “Unused”: lists any unused flow charts (a category to temporarily disable execution).



You can also place unfinished flow charts in the “Unused” category.

## Think & Do

When the project has both standard flow charts and subcharts, the hierarchical relationship is shown in ProjectCenter's "Flow Chart" Explorer bar (see Figure 4-4).

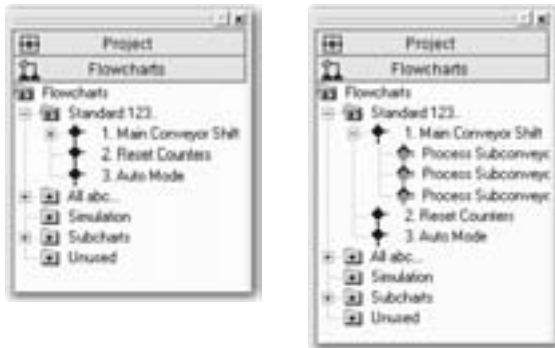


Figure 4-4 Heirarchical sublist list collapsed (left), expanded (right)

In the example in Figure 4-4, the "Main Conveyor Shift" flow chart has one or more associated subcharts as indicated by the + to its left.

To expand the Flow Chart explorer tree:

- Click the + beside the folder to expand ("Main Conveyor Shift" in the example).
- Press the keypad <+> key to expand the "Standard 123..." the selected folder. Use the keypad <-> key to collapse a selected folder

In Figure 4-4, three subcharts are called from the flow chart "Main Conveyor Shift". Note that more than one standard flow chart can call a particular subchart, so an instance of that subchart appears in the directory tree under each standard flow chart that calls it. In addition, an instance appears for each call from a flow chart. The subchart appears once in the "All 123..." folder and the "Subchart" folder.



FlowView Icon

## 4.3 Creating a Flow Chart

To create a new flow chart, follow these steps:

1. In ProjectCenter, launch the FlowView using one of the following techniques:
  - a. Click the "Tools...FlowView" menu.
  - b. Click the "FlowView" icon in the "Project" Explorer bar.

- c. Click the “File... New... Standard Chart” menu. This will launch FlowView, the Think & Do flow charting tool. If using this technique, continue with Step 3.
2. In FlowView, select “File... New... Standard Chart” menu.
3. In FlowView, a “Name Flow Chart” dialog box appears (see Figure 4-5). Type a name for the flow chart.



Figure 4-5 “Name Flow Chart” dialog box

4. Click the “OK” button.

## 4.4 Changing Flow Chart Settings

All standard flow charts in a project execute once each logic scan. The “Standard 123...” folder under the “Flow Chart” Explorer bar lists the execution order (top one first, bottom last).

### 4.4.1 Changing Execution Order

To change the order of flow chart execution:

1. Click the + beside the “Standard 123...” folder to expand it and show all standard flow charts.
2. Place the cursor over the diamond symbol by the flow chart name to re-order.
3. Click and drag the flow chart to a new location on the list. The tree listing then shows the new order of execution.



Only the “Standard 123...” listing of flow charts may be re-ordered. Other folders simply list the category of flow charts they contain in alphabetical order.



### 4.4.2 Changing a Flow Chart's Category

To change a flow chart to/from Simulation or Unused category:

1. Select the flow chart name in the "Flow Chart" explorer window.
2. Do one of the following:
  - In the main ProjectCenter window, click the check box for the desired setting as shown in Figure 4-6.

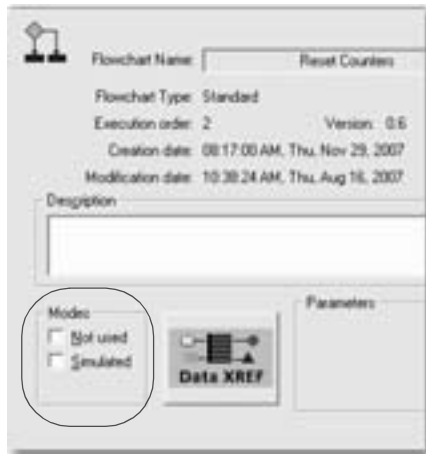


Figure 4-6 Select the "Unused" or "Simulation" check box to move a flow chart

- Drag the flow chart to the appropriate folder.

### 4.4.3 Renaming, Copying, Deleting, or Printing Flow Charts

To rename, duplicate, delete, or print an existing flow chart:

1. Move the cursor over the name of the flow chart in the Flow Chart Explorer pane.
2. Right-click to get the pop-up menu, and select the desired action.



Before creating a new flow chart in a new project, it's best to name the project first, by saving it to a new directory (each project must be in a unique directory).

## 4.5 FlowView Orientation

When you create a new flow chart, FlowView shows it maximized in the workspace (see Figure 4-7). Use the “Window” menu to show multiple open flow charts by selecting “Cascade” or one of the tile options. Switch to other open flow charts either using the list in the “Window” menu, or by pressing the <F6> or <Ctrl>+<Tab> (<Shift>+<F6> or <Shift>+<Tab> to reverse the order) key.

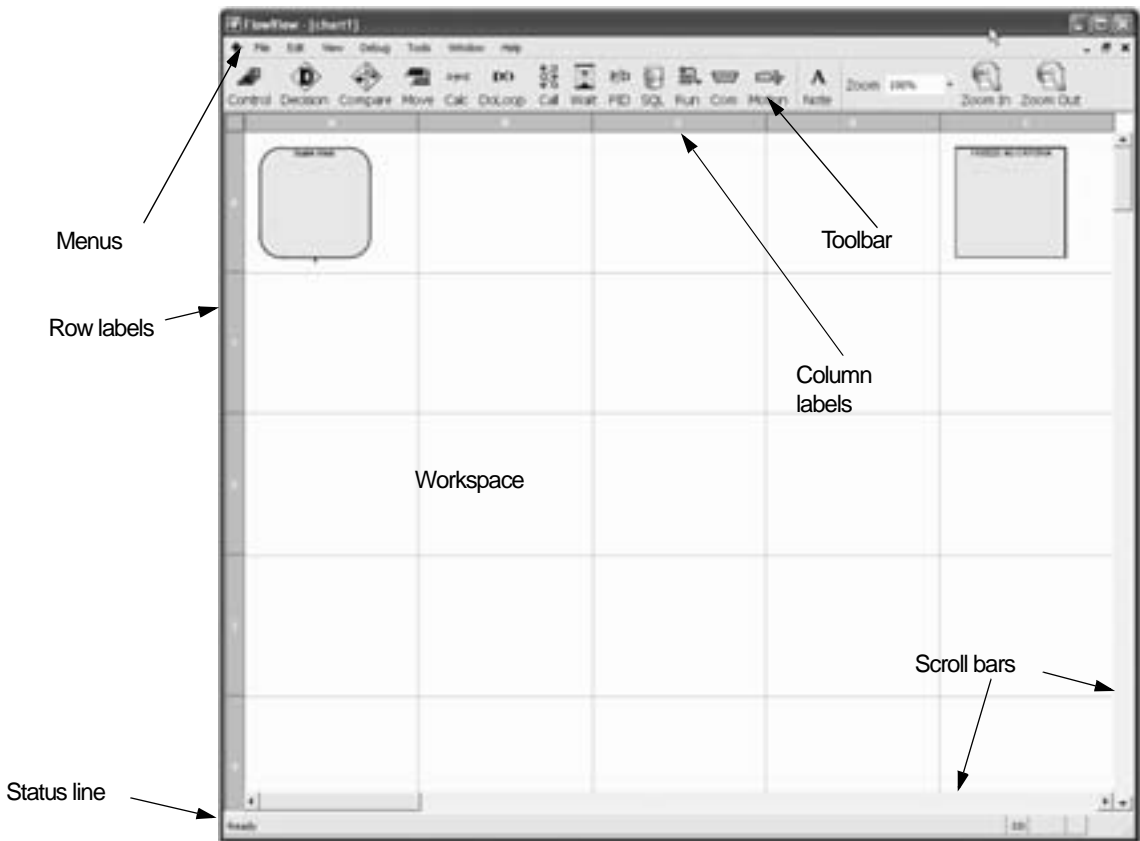


Figure 4-7 FlowView showing a new Flow Chart

The key elements of the FlowView window, shown in Figure 4-7, are:

- **Menu:** Provide access to all the commands in FlowView, as well as online help.

## Think & Do

---

- **Toolbar:** Contains toolbar buttons that trigger actions. These actions duplicate many commands that are available in the menus.
- **Column labels:** Provide a guide for locating shapes in the drawing area. FlowView provides five columns for flow chart objects.
- **Row labels:** Provide a guide for locating shapes in the drawing area. FlowView provides an unlimited number of rows as they are needed... limited only by available memory.
- **Drawing area:** Build flow charts in the drawing area. Horizontal and vertical grid lines appear by default. Blocks must appear within a grid cell with only one block per cell.
- **scroll bars:** Use the scroll bars to bring portions of the drawing area that are off-screen into view. To scroll in small increments, click the arrows at the end of each scroll bar.
- **Status line:** Provides information about FlowView or selected shapes, etc.

Use multiple flow charts for all but the simplest projects with each flow chart controlling a single process, machine, or major function. A modular approach to developing flow chart programs makes it easier to create, debug, and maintain control projects.

All enabled flow charts execute in the order they appear in the “Standard 123...” list. The project can programmatically control which flow charts execute during any given scan cycle. For more information on enabling flow charts, see “Enable Block” on page 5-4.

To create a flow chart, select the “File... New... Standard Chart” menu. To open a flow chart, do one of the following:

- Select the “File... Open Flow Chart...” menu, and then select a flow chart from the cascaded menu list.
- To display a list of current flow charts in the project, use the Flow Chart Explorer in ProjectCenter. Open a flow chart from the list by double-clicking it.

### 4.5.1 Adding Flow Chart Blocks to a Flow Chart

To add a flow chart block to the current flow chart, follow these steps:

1. Click the icon in the toolbar for the flow chart block that you want to add.
2. Click the cell where you want to place the flow chart block.



Each cell of the flow chart can have only one flow chart block.

### 4.5.2 Turning the Grid On and Off

To turn the page grid on or off, follow these steps:

1. Select the “File...Preferences” menu to view the “Preferences” dialog box.
2. In the “Preferences... General” tab, clear the “Show Grid Lines” check box to turn off the grid lines. Select the check box to turn the grid lines back on.

The full design of a flow chart may be large, and any page may be larger than the viewable window. You can see more of the page by using the zoom factor setting.

### 4.5.3 Increasing the Number of Columns

To increase the number of columns, do one of the following sequence of steps:

#### For the Current Flow Chart

1. Select the “File... Chart Properties” menu to display the “Chart Properties” dialog box.
2. Type a number or use the up/down buttons in the “Columns” field to change the number of columns.
3. Click the “OK” button to close the dialog box.

#### For All Flow Charts

1. Select the “File... Preferences” menu to display the “Preferences” dialog box.
2. In the “Preferences... General” tab, type a number or use the up/down buttons in the “Columns” field to change the number of columns.
3. In the “Save Preferences” dialog box that appears, choose the option desired. You can choose to:
  - Apply the Changes to this project only
  - Apply the Changes to this project only, and all future projects created from this point forward
  - Leave the preferences set the way they are
  - Reset the project to the default settings for new projects
  - Abandon changes. Reset this project and all future projects to the factory default settings.
4. Click the “OK” button to close the dialog box.

#### 4.5.4 Changing Zoom Level

To change the page magnification:

- Click the “Zoom” drop-down list on the toolbar (Figure 4-8) or click a magnification tool in the menu bar (+) or (–) to increment the zoom to the next respective setting.

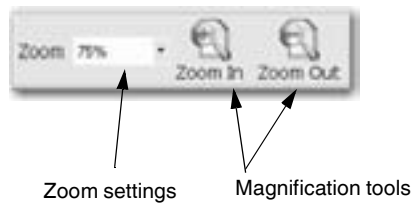


Figure 4-8 Zoom tools in FlowView

- Another method is to select the “View...Zoom Factor...” and select the desired setting.

## 4.6 Displaying Multiple Flow Charts

You can open multiple flow charts in FlowView. This is very useful for copying flow chart logic from one to another. As shown in Figure 4-9, each flow chart occupies its own window within FlowView.

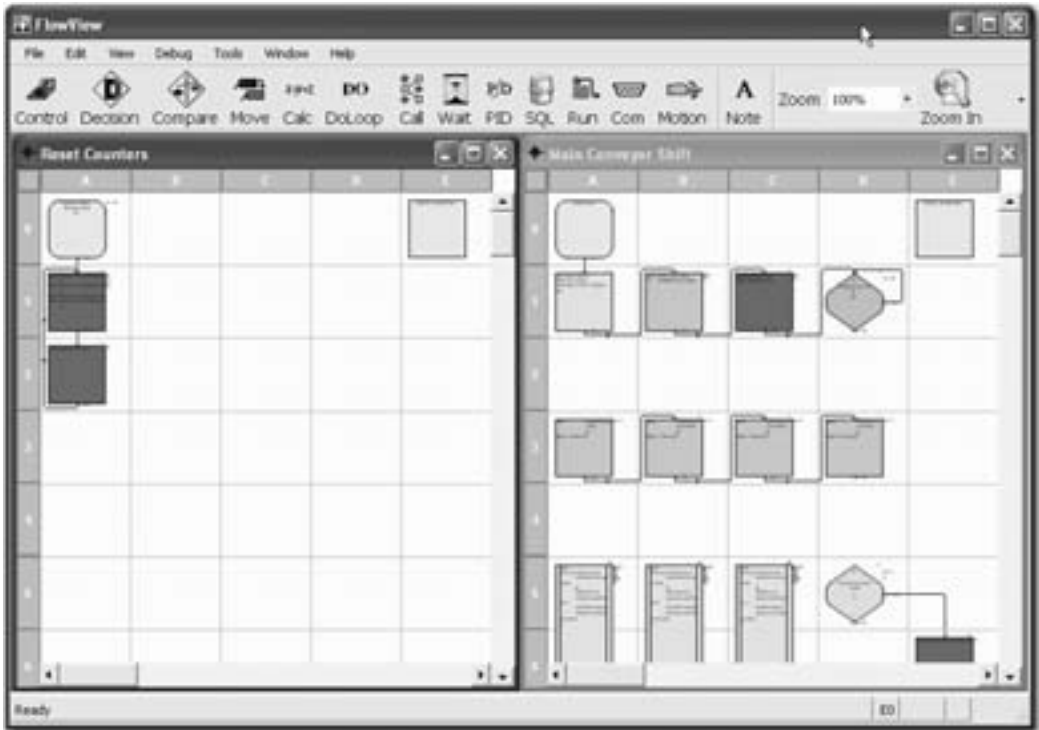


Figure 4-9 Open multiple flow charts in FlowView, each with its own window

The window arrangement in Figure 4-9 is called *tiled*. A *cascaded* arrangement places one behind the other, slightly offset. You can also maximize, minimize, or close any flow chart window by using the “Minimize”, “Maximize”, and “Close” buttons in the upper right corner.

To choose tiled or cascaded windows, click the “Window...Cascade” or “Window...Tile Vertically” or “Window... Tile Horizontally” menu from the FlowView main menu.

## 4.7 Editing Flow Charts

FlowView provides intuitive editing capability to move or cut-and-paste flow chart blocks from one grid location to another. Many functions are available with a mouse right-click, or use toolbar buttons as described:

To select flow chart elements for editing:

1. Click a block or connection to select it.



By default, FlowView is always in selection mode, except after clicking a block's exit connector anchor that doesn't have a flow line connecting to another block (see "Connecting Flow Chart Blocks" on page 4-18).

2. To select multiple blocks or connections do one of the following:
  - Select a rectangular area by pressing (hold down) the left mouse button, dragging the cursor diagonally to size a rectangular area, and then releasing the mouse button.
  - Hold the <Ctrl> key while clicking blocks or connections. Clicking a selected block or connection with the <Ctrl> key pressed deselects that object without deselecting any others.
3. Perform the action (such as cut, copy, move, or delete).

### 4.7.1 Cut-Copy-Paste

After selecting the object(s), use the cut (<Ctrl>+<x>), copy (<Ctrl>+<c>), and paste (<Ctrl>+<v>) shortcuts to speed up flow chart development, or use the menu selections from the "Edit" menu. With multiple flow charts open, it is possible to cut/copy and paste from one flow chart to another.

## 4.8 Flow Chart Block Introduction

This section provides a brief introduction to flow charts. For a more complete discussion, see Section 5, “Flow Charts”.

To place a new flow chart block on the page, follow these steps:

1. In the toolbar, click the icon of the block type to place.
2. Click in the desired location on the drawing page.

### 4.8.1 Enable Block

The Enable block always appears at the top of a flow chart. It allows the flow chart to begin executing. It can contain an expression that conditionally activates the flow chart while the project is running. To develop the control program, add flow chart blocks below the Enable block.

### 4.8.2 Freeze Block

The default Freeze Block (upper right corner of flow chart page) does not connect to other flow chart blocks. When configured, its expression can conditionally freeze or resume a flow chart execution.

### 4.8.3 Types of Action Blocks

FlowView has the following types of action blocks (in the order they appear in the toolbar):

- Control: Control blocks initiate an action such as turning on an output or resetting a timer. One Control block can have up to ten control expressions. Control expressions set values for flags, turn outputs on and off, and control timers and counters.
- Move: Move blocks simply move data from one data item to another. The source data item remains unchanged
- Calc: Calculation blocks provide comprehensive math capabilities. They operate on a variety of data types with either simple arithmetic or transcendental functions such as sine, log, etc.
- Do Loop: Do Loops are a construct that allow blocks to execute more than once during a scan.
- Call: Call blocks call a subchart. Call Blocks transfer execution to the called subchart with specified parameters.



## Think & Do

---

- Wait: Wait blocks insert a wait period in the execution of the flow chart. A Wait block can have a wait period set in milliseconds or hours, minutes, and seconds. If the wait duration is zero, the block defaults to one scan cycle.
- PID: PID blocks perform a mathematical algorithm for process control, which is the Proportional Integral Derivative (PID) algorithm. Think & Do permits up to 64 PID loops in a project.
- SQL: SQL blocks provide a tabbed dialog box that helps you construct SQL queries for database accesses.
- Run: Run Program block lets you start other Windows programs or batch files under the flow chart control.
- Com: Communication blocks permit a flow chart to communicate directly with an external device through a serial port on the PC.
- Motion: Motion blocks provide integrated motion control. Integrated motion control provides convenience and time savings during system design, since you avoid having to link external motion programs with the project.

### 4.8.4 Freeze Block

Every main flow chart can contain a Freeze block. The Freeze block does not connect to other flow chart blocks. When configured, its expression can conditionally freeze or resume a flow chart execution.

#### Types of Branching Blocks

Branching blocks (both Decision and Compare blocks) perform a test and determine which of two paths to take after the test. Each branching block has one entry point and two exit points indicating Yes and No paths from the block. By default, the bottom connector is the Yes path, and the right side is the No path.

A Do Loop is a branching construct that includes a Compare block at the beginning and end of the loop. You place any number of action blocks between the beginning and ending Do Loop blocks. A Do Loop only has one exit. Branches to blocks outside the Do Loop aren't permitted.



The standard orientation of flow chart is top-to-bottom for ease in reading.

Both Decision and Compare blocks appear as diamonds in the flow chart, while the Do Loop appears as a pair of diamonds (see Figure 4-10).

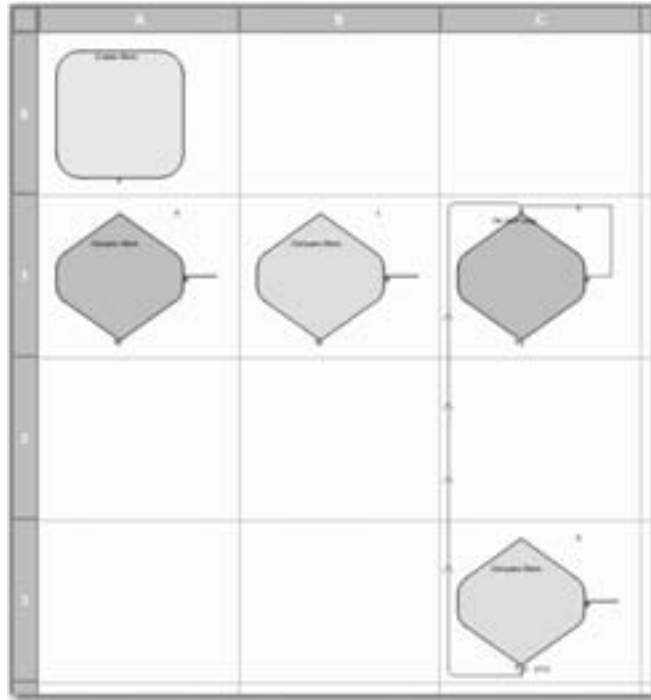


Figure 4-10 Decision, Compare, and Do Loop blocks appear as diamonds

Decision blocks examine the state of a data item. For example, a Decision block can test inputs or outputs to determine if they are ON or OFF or transitioning from one state to the other. A Decision block can have up to seven decision expressions. If there are multiple expressions, an OR or AND condition links them. For example, a Decision block could have the following expressions:

- Input1 ON AND Input2 OFF

Compare blocks compare two data items. For example, a Compare block can compare an integer register value to a constant or another register. A Compare block can compare values using one of the following comparison operators:

- = (Equal To)
- > (Greater Than)
- < (Less Than)

<> (Not Equal To)

>= (Greater Than or Equal To)

<= (Less Than or Equal To)

The Do Loop blocks are similar to Compare blocks. The first one, at the top of the Do Loop provides a test for a maximum loop count. This is primarily to prevent infinite loops and should not be used as a normal exit. The bottom Compare block is a standard Compare block. It is the normal (and only valid) exit point from the loop.



To change the bottom Compare block to a Decision block, right-click on the block, and then select "Change this Compare block to a Decision block" from the pop-up menu.

### 4.8.5 Notes

Notes don't actually appear in a block. They can appear anywhere on the flow chart page, and don't connect to or affect program flow.

## 4.9 Connecting Flow Chart Blocks

To draw flow chart block connecting lines, follow these steps:

1. Move the cursor to the bottom (or side for Decision and Compare blocks) connecting point of a block that doesn't have an out-going flow line. The connecting point turns orange (or the specified color) to indicate a potential out-going connector pointer, and the cursor changes to the connector pointer.



If the flow chart editor is inadvertently in connect mode, press the <Esc> key to switch back to select mode.

2. Click the out-going connector point.
3. Move the cursor to the next block's top or left (in-coming) connector point, which will turn green (or the specified color).
4. Click the in-coming connector to complete the line. The cursor changes back to the selector pointer.



FlowView supports automatic routing from one block to another, so it is only necessary to click the out-going connector of one block and the incoming connector of the next to generate the flow line. It is always possible to manually draw the flow line by clicking from point-to-point to route the line.

For example, Figure 4-11 shows a simple project flow chart with all required connecting lines.

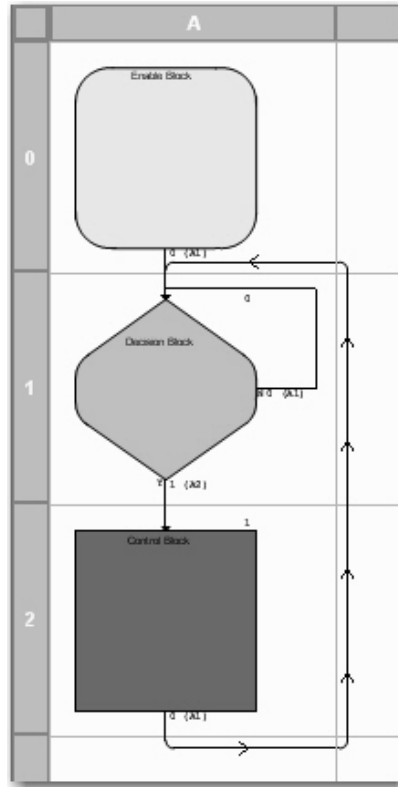


Figure 4-11 Sample flow chart with flow lines connecting all blocks



Unlike CAD programs that permit ending a line where it intersects another line, FlowView requires guiding the pending connection to the common destination.



If the connector appears as a red line, it is not properly connected. This can occur if the connection doesn't meet the input connection point of the block. Try moving the end point of the line until the region that represents the block's input connection becomes highlighted.

## 4.10 Editing Flow Chart Block Expressions

The content of a flow chart block is an expression. To enter and edit block expressions, double-click the block to display the “Configuration” tab of its properties dialog box. For a discussion of each block’s “Configuration” tab, see Section 5, “Flow Charts”.

## 4.11 Entering Block Description

Enter a description of a block to summarize actions taken in the block. This is useful to provide an overview version of the flow chart. Specify whether flow charts should display block descriptions or block expressions by toggling the “View... Block Descriptions” menu.

To enter or edit block descriptions:

1. Right-click on the block and select “Block Properties” from the pop-up menu. The “Properties” tab of the dialog box (see Figure 4-12) appears.



Figure 4-12 “Compare Block... Properties” tab

2. Enter the block “Description”, and then click the “OK” button.



The “Address” fields for “Yes” and “No” in the “Compare” and “Decision” block “Properties” tabs permit entry of a block number without drawing a connection.

3. To view all block descriptions on the flow chart instead of expressions, select the “View... Block Descriptions” menu.

## 4.12 Operator Screens

Think & Do provides more than just PC-based control with flow charts – it includes a full-featured operator interface or HMI (Human-Machine Interface). Use ScreenView to create operator screens. The remainder of this chapter focuses on the unique features that you need to know to develop HMI screens.



Windows CE target Runtime systems – some devices, such as the WinPLC, do not have direct screen output available to its operating system.

The final output of the development work is a screen or series of screens. These appear in a window on the PC at Runtime. The purpose of a screen is to communicate important information to a machine operator or process engineer. It also provides an opportunity for operator-controlled input. Screens typically include:

- The current state of the machine or process, such as Idle, Run, or Stop.
- Alarm conditions, if they exist
- Production data (such as number of parts, percentage defects, downtime reasons, and trends)
- Diagram of process (on-screen diagram to help orient the operator)

ScreenView provides a full-featured HMI or operator interface design package, which helps to create screens that really communicate. Operator input support includes function keys and data entry.

### 4.12.1 Screens as Part of a Project

The screen(s) developed using ScreenView become part of a project. Then at Runtime, the screens appear in an independent window on the desktop. Only one screen is visible in the window at a time. You can re-size or minimize

## Think & Do

screens just as any window. You can also run in full-screen mode, where screen size adjustment isn't available to the operator. Data entry or function key entry requires that the screen be in the active window.



When a project runs, its flow charts—not operator screens—are the highest priority task. During the “Pause” phase of each scan period (see page 9-3 in the “Think & Do User Manual”), the system executes screen tasks and other applications. Adding screens affects system loading, so be sure that the scan time setting is large enough to allow the system to service your operator screen(s).

### 4.12.2 Creating Operator Screens

To create a new screen in the project, follow these steps:

1. In ProjectCenter, select the “File...New...Screen” menu to display Screen-View and the “Screen Attributes” dialog box (see Figure 4-13).

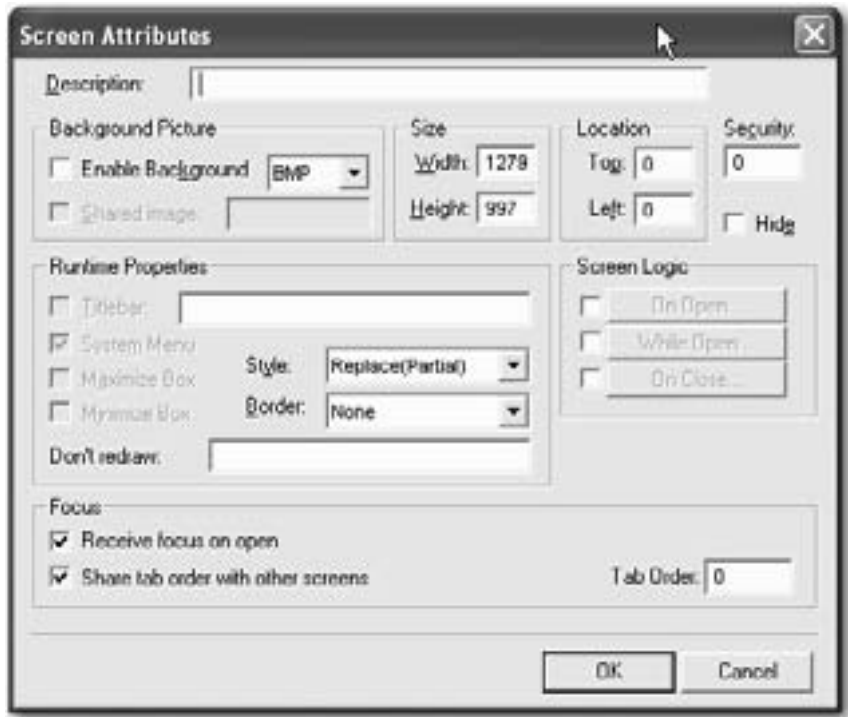


Figure 4-13 Use the “Screen Attributes” dialog box to set up a new screen

2. In the “Screen Attributes” dialog box, type a description of the screen in the “Description” field.
3. Set the “Size” and “Location” for the Runtime screen.
4. Click the “OK” button.

For a discussion of additional options in this dialog box, see page 6-112 in the “Think & Do User Manual”.

### 4.12.3 ScreenView Development Environment

ScreenView uses standard tools and interfaces to make the product user-friendly. ScreenView provides an integrated and unique development environment that gives you fast and easy access to tools and information.

The ScreenView development environment (see Figure 4-14) consists of the following basic areas:

- Title Bar: Indicates the active display or worksheet
- Menu Bar: Contains the main product options and controls, which you can easily access using the cursor or your keyboard
- Toolbars: Provide access to features and tools used in the development environment
- Workspace: Provides tree-view control from which you can access project worksheets and screens
- Screens / Worksheets: Provides an area where you can edit displays and worksheets



## Think & Do

---

The following figure shows the development environment areas and windows in their default position. You can customize this environment by changing the position of the different areas.



Figure 4-14 ScreenView development environment

### ScreenView Explorer

The ScreenView Explorer is a user-friendly interface that lets you quickly find application components (screens or worksheets). The ScreenView Explorer uses a tree-view with two main folders—one for screens and the other for worksheets.



Figure 4-15 ScreenView Explorer

Whenever you create a screen or worksheet, ScreenView adds the new object to the tree. Double-clicking an entry in the ScreenView Explorer displays that object in the ScreenView workspace. Right-clicking an object in the tree (not a folder) displays a pop-up menu with options to “Rename”, “Duplicate”, or “Delete” the selected object.

The “Screens” folder contains screens with finished graphic compilations and working drafts. To view a screen, expand the “Screens” folder and double-click the desired screen to display it in the Workspace.

The “Worksheets” folder contains other worksheet objects that you can create in ScreenView. Among them are alarm (see page 6-115 in the “Think & Do User Manual”), math (see page 6-121 in the “Think & Do User Manual”), and trend (see page 6-127 in the “Think & Do User Manual”) worksheets.

### 4.12.4 Object Properties Dialog Box

The “Object Properties” dialog box is the interface that allows you to configure the settings for the objects and dynamics designed with ScreenView. It can be launched by double-clicking on the object or by selecting the object (clicking once on it) and pressing <Shift>+<F2>.

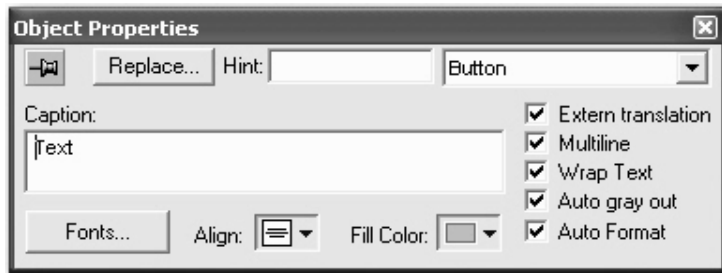



Figure 4-16 The “Object Properties” dialog box

The content of this dialog depends on the object (or group of objects) selected by the user. However, the following interfaces are common for any object (or group of objects):

Table 4-1 “Object Properties” dialog box properties

Field	Remarks	Syntax
(Pin button)	When the pin button is released, focus is passed to the object on the screen as soon as that object is selected. When the pin button is pressed focus remains on the Object Properties window, even when you click the objects on the screen.	Button
“Replace”	Launches the Replace Dialog, where you can replace strings, tags or properties for the selected object (or group of objects).	Button

Table 4-1 “Object Properties” dialog box properties

Field	Remarks	Syntax
“Hint”	Tooltip displayed during runtime, when the mouse cursor remains oved the object. This option is useful to provide hints to the operator during runtime. The text on the “Hint” field is also written to the Internal Tag Hint during runtime. Therefore, you can trigger actions based on the value of this tag.	Text and/or {Tag} (up to 256 chars)
	The combo-box at the right side of the dialog box lets the user select a specific object of objects that must be edited.	Combo-box



The “Enable Tooltip” option must be enabled (checked) on the “Project Settings” dialog box. Otherwise, tooltips are NOT displayed during runtime when the user points the mouse cursor on the objects. You can enable/disable this feature for the local server (select the “Project... Settings” menu).

Clicking the “Replace” button from the “Object Properties” dialog box displays the “Replace” dialog box (see Figure 4-17).

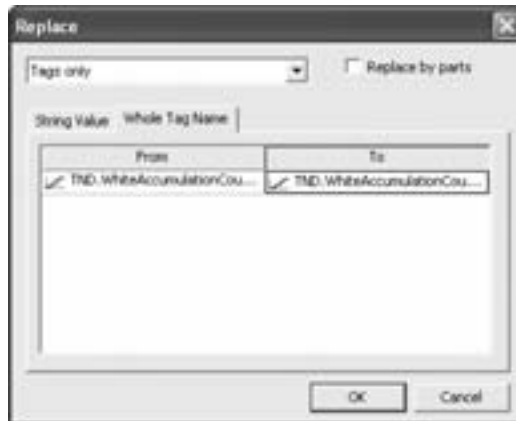



Figure 4-17 The “Replace” dialog box

## Think & Do

This dialog provides a fast and simple tool to replace strings (“String Value” tab), tags and/or properties for an object (or group of objects). This dialog display grids where you can edit the “To” column with the string, tag and/or property that must be replaced. The main interfaces in this dialog as described in the following table:

Table 4-2 Object Properties “Replace” dialog box parameters

Field	Remarks	Syntax
	Allows you to filter the type of information that can be replaced: <ul style="list-style-type: none"> <li>– “Tags only”: Displays only the tags configured directly (e.g.: Second, Time, Level, and so forth).</li> <li>– “Tags + Properties”: Displays both the tags configured directly and the properties configured as mnemonics (e.g.: #PumpStatus).</li> <li>– “Custom Properties”: Displays only the mnemonics configured as custom properties (e.g.: #PumpStatus:).</li> </ul>	Combo-box
“Replace by Parts”	When checked allows the user to replace each part of a tag separately: Main Tag-name, Array Index, Class Member and Tag Field. It is very useful when you want to replace a specific settings (e.g.: Array Index) for different tags configured in the same Group of Objects.	Button



When pressing the “Replace” button after selecting the “Group of Objects” option in the “Object Properties” dialog box, the replace feature affects all objects and dynamics which are part of the selected Group of Objects. However, when pressing Replace button after selecting a specific object or dynamic (e.g.: Button, Color, etc), the replace feature affects ONLY the selected object or dynamic.

### Color Interface



Color  
Icon

You can edit the color of a component with the Color interface.

1. Click the “Color” icon in the toolbar.
2. Do one of the following:

- Click the desired color from the twenty that display when the pop-up box opens:



The selected color will be applied to the component that you are editing.

- Click the “More Colors...” link if you want to apply a different color. This displays the “Colors” dialog box, which shows the 143 standard colors from your operating system. If the color you want is available here, you can select a color, and then click the “OK” button.

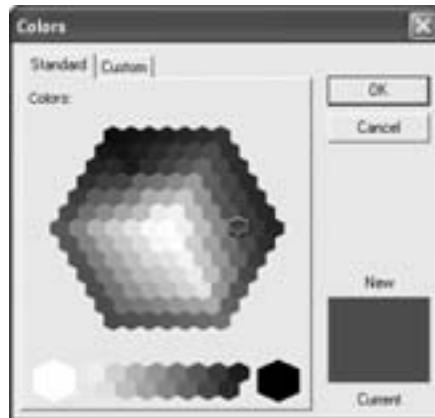


Figure 4-18 The “Colors” dialog box

## Think & Do

---

- To use a color that isn't available on the “Colors... Standard” tab, click the Custom tab. You can then edit the “Hue”, “Sat”, “Lum” or “Red”, “Green”, “Blue” codes of any of the 143 standard colors, creating a custom color.

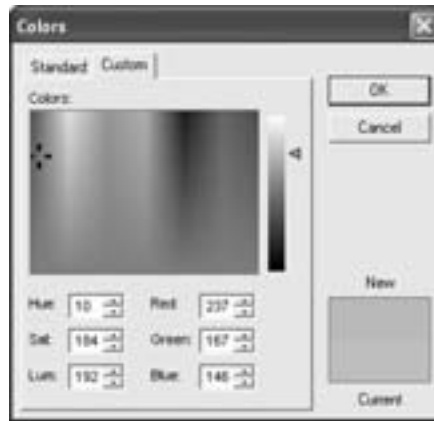


Figure 4-19 The “Colors... Custom” tab

3. Click the “OK” button to apply the selected color to the component that is being edited.
4. Depending on the component that you are editing, the “Fill Effects” option may be available from the pop-up interface (see step 2 above). Clicking this link displays the “Fill Effects” dialog box, which lets you apply gradient colors with different styles and variants.



Figure 4-20 The “Fill Effects” dialog box

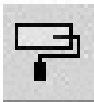
5. Select two colors in the “Start” and “End” fields, select an option from the “Styles” group, and then click the “OK” button to apply the fill effect to the component.



For applications that run on the Windows CE operating system, the Fill Effects interface is available only for the Rectangle object.



Although “Fill Effects” is a useful tool for enhancing the look and feel of your screens, the operating system takes a longer time to fill an object with fill effects than with plain colors. You should develop criteria for using the feature without decreasing the performance of the system, especially under the Windows CE operating system.



Color  
Icon

Using the “Colors” dynamic tool, you can modify the color of a static object during the runtime. When configuring this dynamic with “Type” is “By Color”, you can set the color that will be applied in the object during the runtime, by the color code. See the Think & Do online help for a table of code values as well as the RGB values for the most commonly used colors.



The number of colors available when developing the application depends on the color settings configured on the operating system of the development station. The number of colors available when running the application (Runtime) depends on the color settings configured on the operating system of the Runtime station.





Pin Button  
(Released  
position)

### Configuring the Focus of the Object Properties Dialog Box

Double-clicking any object (or group of objects) in ScreenView displays the “Object Properties” dialog box, which lets you configure the selected object’s settings. The content of this dialog box varies according to the specific object/dynamic selected. However, there is always a pin button in the left upper corner of this dialog box.

When the pin button is released, the focus is passed to the object on the screen as soon as that object is selected. Therefore, we recommend you keep this button released when you want to manipulate (copy, paste, cut or delete) the objects. Although the “Object Properties” dialog box is on the top, the keyboard commands (<Ctrl>+<C>, <Ctrl>+<V>, <Ctrl>+<X> or <Del>) are sent directly to the objects.



Pin Button  
(Pressed  
position)

When the pin button is pressed focus remains on the “Object Properties” dialog box, even when you click objects on the screen. We recommend you keep this button pressed when you want to modify the settings of objects. You can click an object and type the new property value directly in the “Object Properties” dialog box (it is not necessary to click on the window to bring focus to it). Also, when the pin button is pressed, the “Object Properties” dialog box does not automatically close when you click on the screen.

## 4.12.5 Creating and Using Tags in ScreenView

In ScreenView, you use tags to display state, results of calculations, alarm points, and so forth. In ScreenView, all tags are organized according to their origin— application, internal, or shared. Application and internal tags originate in ScreenView, while shared tags originate in the Think & Do Data Items editor.

The following is a description of the different ScreenView tag types:

- Application tags are user-defined, screen tags created for displays, to read from and write to field equipment, for control, auxiliary tags to perform mathematical calculations, and so forth.
- Internal tags are system tags predefined by ScreenView. Internal tags have predetermined functions (time, date, acknowledge alarms, storage of the logged-on user name and so forth). You cannot delete or modify these tags, but you can access their values from any ScreenView task.
- Shared tags are created in the Data Items editor and imported into the ScreenView environment. You cannot edit shared tags in the ScreenView environment, but you can modify these tags in ProjectCenter. Consequently, you can configure shared tags for any ScreenView task just as any other tag.

### Understanding the Screen Tag Syntax

Observe the following guidelines when naming a screen tag:

- You can use letters, numbers, and the character "\_" (underscore)
- Do not use the following characters:

```
`~!@#$%^&*()-=\+\[ ] { } < > ?
```

These characters are parsed by the ScreenView scripting language as logic and arithmetic operators.

- The tag must begin with a letter
- Maximum tag length is 255 characters
- Maximum class member length is 255 characters
- Tag names must be unique. You cannot specify the same name for two tags.
- Tags are not case sensitive (though we recommended using uppercase and lowercase characters to make names more readable. For example, use "TankLevel" instead of "tanklevel".
- Tag names must be different from internal tag names and math functions.



Use the @ character at the beginning of a tag name to indicate that the tag will be used as an indirect tag in the application.

For additional information, see the ScreenView scripting language in the online help.

Some valid tag examples include:

- Temperature
- pressure1
- count
- x

### Using Screen Tag Values

A screen tag value can be one of the following types:

- Boolean is a Boolean or digital variable (0 or 1).
- Integer is a number (positive, negative, or zero). Equivalent to C-type long integer. Examples: 0, 5, -200.
- Real is a number internally stored as a double word. Equivalent to C-type double.
- String (ASCII text) is a character string up to 1024 characters that holds letters, numbers, or special characters. Examples: Recipe product X123, 01/01/90, \*\*\* On \*\*\*.

## 4.13 Scripting Languages in ScreenView

ScreenView supports a built-in scripting language and Microsoft Visual Basic Script language (VBScript<sup>®</sup>). These languages have familiar syntax, operators, and functions. They provide:

- The ability to create new variables and procedures (functions and/or sub-routines)
- Access to properties, methods and/or events from COM objects, including ActiveX controls
- The ability to execute the logic on any platform

For details of both the built-in scripting language and VBScript, see the Think & Do online help.

# Section 5

This section informs you about

- Flow charts blocks and how to use them
- Creating and using subcharts

Flow Charts .....	5-3
5.1 Flow Chart Types and Categories .....	5-3
5.2 Basic Flow Chart Blocks .....	5-3
5.2.1 Enable Block .....	5-4
5.2.2 Calculation Block.....	5-6
5.2.3 Call Block .....	5-8
5.2.4 Note .....	5-9
5.2.5 Communication Block .....	5-10
5.2.6 Compare Block.....	5-11
5.2.7 Control Block.....	5-15
5.2.8 Decision Block.....	5-17
5.2.9 Do Loop Block.....	5-20
5.2.10 Motion Block .....	5-22
5.2.11 Move Block.....	5-24
5.2.12 PID Block .....	5-26
5.2.13 Run Program Block .....	5-28
5.2.14 SQL Block .....	5-29
5.2.15 Wait Block .....	5-30
5.3 Using Subcharts.....	5-31
5.3.1 Blocks for Subcharts .....	5-31
5.3.2 Begin Expression .....	5-33
5.3.3 Call Expression .....	5-35
5.3.4 Interaction of Call and Begin Blocks .....	5-37
5.3.5 Subchart Design Summary.....	5-37
5.3.6 Subchart Runtime Characteristics .....	5-38
5.3.7 When to Use Subcharts.....	5-39

## Think & Do

---

## 5 Flow Charts

This section provides an in-depth discussion of the flow charting features of Think & Do.

### 5.1 Flow Chart Types and Categories

Think & Do has a two-level flow chart hierarchy implemented with two types of flow charts:

- Standard flow charts
- Subcharts

You use standard flow charts to define a project. If your project has only one flow chart, it must be a standard flow chart. Standard flow charts can call subcharts (much like a main program calls subroutines).

Standard and subchart type flow charts can pass variables back and forth. Note that in this manual, we use the term “flow charts” to mean either the standard or subchart type.

### 5.2 Basic Flow Chart Blocks

The first block in a flow chart is the starting point of execution at runtime. By default, the proper flow chart starting block is at the top of the drawing page when you create a new flow chart:

- Standard flow chart: Enable Block is first
- Subchart: Begin block is first



You can determine the flow chart type at a glance by looking at the first block. Each flow chart can have just one of these two blocks as a starting point. FlowView automatically creates these starting blocks when you create a new flow chart or new subchart.

### 5.2.1 Enable Block

Each flow chart must have a beginning point, and the Enable block serves that purpose for standard flow charts. The Enable block is in the upper left grid cell in a new flow chart (see Figure 5-1). It has the following characteristics:

- The Enable block is the first block of every standard flow chart. It has a single exit from which flow passes on to the rest of the control program. The Enable block cannot be deleted, copied, or moved on the drawing area.

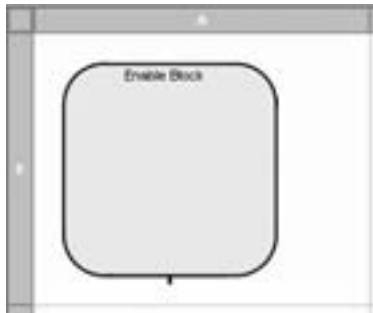


Figure 5-1 Every new flow chart opens with a predefined Enable block

- An empty Enable block (no expression created) causes a flow chart to always run.
- Double-click the block to edit its expression.
- Add statements to the expression using logical operators (see Figure 5-2). The result of the expression is either TRUE or FALSE, enabling or disabling the flow chart respectively.

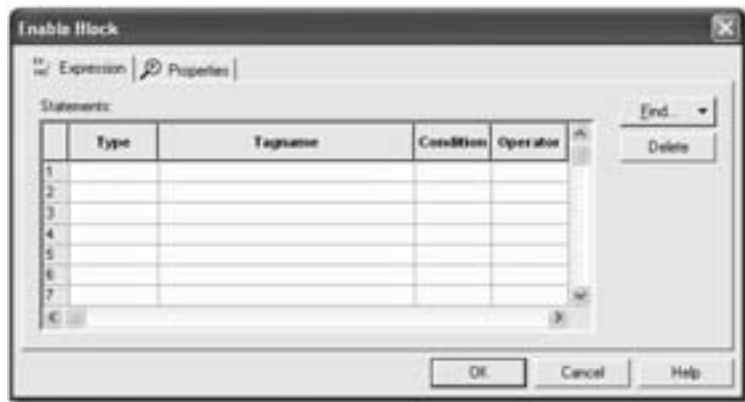


Figure 5-2 The “Enable Block... Expression” tab

- The AND and OR “Operator” in the dialog box provides for additional Boolean conditions in the expression. AND has greater precedence than OR (in other words AND expressions are evaluated before OR expressions, and then they are evaluated from top-down). Think & Do Runtime evaluates the Enable block every scan before executing the flow chart (based on the position of the flow chart in the flow chart list—see “Changing Execution Order” on page 4-7). The flow chart only runs if its expression is TRUE at the start of the scan.
- On the first scan or whenever the Enable block expression transitions from FALSE to TRUE, flow chart execution starts at the Enable block exit. Otherwise, execution starts where it left off the previous scan.

Table 5-1 defines all possible Enable block combinations. This table shows valid data item types and the conditions that appear in the “Condition” drop-down list. For example, it is possible to test to see if an Input is “ON” or “OFF”.

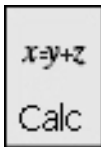
Table 5-1 Valid Enable Block Expressions<sup>a</sup>

Data Item Type	ON	OFF
Array	X	X
Comm	X	X
Flag	X	X
Input	X	X
Output	X	X
Timer	X	X

<sup>a</sup> X=valid



### 5.2.2 Calculation Block



Calculation block Button

The Calculation block provides comprehensive math capabilities. It operates on a variety of data types with either simple arithmetic or transcendental functions, such as sine, log, etc. By editing the “Calculation Block” dialog box (see Figure 2-2), you select the math operations the block performs, as well as the data types of the two operands and the result.



Figure 5-3 The “Calculation Block... Expression” tab

The “Calculation Block... Expression” tab features a “Calculation” field. The “Calculation” field lets you enter an equation producing one result, which can be a function of several variables. The result and variables, which comprise the expression, must be defined in the tagname database. You may define the tagnames in advance, or just click “Find” drop-down button to go to the Data Item Grid to create and/or select the variable. You can also use the “Data Items” group to enter or select data items. Follow these steps:

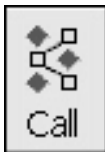
1. Select the data item “Type” from the drop-down list.
2. Do one of the following:
  - Enter the data item name in the “Data Item” field. While typing, Think & Do attempts to auto-complete the name.

- Enter the data item “Index”, for example “01”. When tabbing to the next field, Think & Do displays the data item name associated with the index.
  - Use the “Find...” menu button or double-click the field to display the Data Item Grid.
3. Click the “Add to Calculation” button. If the tagname does not exist, Think & Do prompts you to create the data item.

A single Calculation block produces one result. Connect several Calculation blocks together if you need to produce intermediate results.

The Calculation block includes a Validate function, which checks the syntax of the expression when you click the “OK” button. If the expression is valid, the “Calculation Block” dialog box closes. Otherwise, an error message pops-up. For more information on number ranges and types for math operations, see page 7-3 in the “Think & Do User Manual”.

### 5.2.3 Call Block



Call block Button

You use the Call block to call a subchart. Just as a program can call a subroutine, the Call block transfers execution to the called subchart. Double-click the Call block to access the “Call Block... Expression” tab shown in Figure 2-3.

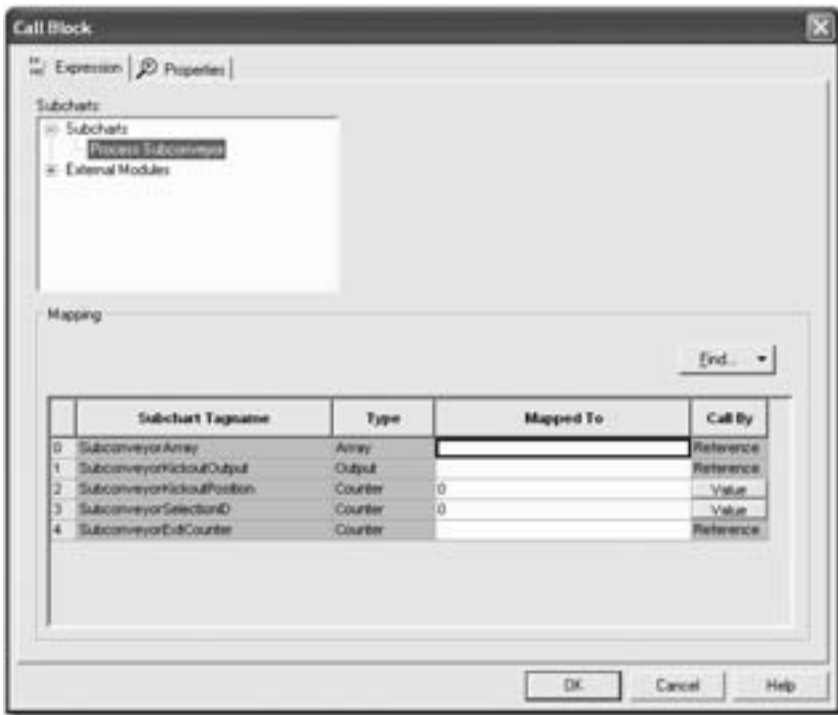


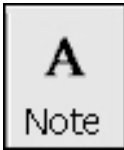
Figure 5-4 The “Call Block... Expression” tab

To use the Call block, select the name of the subchart from the “Subcharts” tree. After selecting a subchart, parameters defined in the subchart appear in the “Mapping” group. This lets you match subchart parameters with those in the calling flow chart. See “Using Subcharts” on page 5-31.



A subchart has to exist before a flow chart can call it. This may sound obvious, but you cannot create the name of a subchart you want to call in the future in the calling flow chart. You must first create a new subchart and save it using the desired name before it appears in the list box in the “Call Block... Expression” tab.

## 5.2.4 Note



**Note Button**

Notes are re-sizable text that can appear anywhere on the drawing page. Typically, notes should appear near relevant flow chart blocks. Once created, drag-and-drop or cut-and-paste operations are possible on notes.

To enter a note:

1. Click the “Note” button in the toolbar, and then click the position in the flow chart where the note should appear. This displays the “Note Properties” dialog box (see Figure 5-5).



Figure 5-5 “Note Properties” dialog box

2. To change the font, font size, or style, click the “Font...” button to display a standard font selection dialog box.
3. Set the text alignment by selecting the appropriate radio button.
4. Optionally, select the “Show Border” check box to turn on a border for the note.
5. Enter the note text in the text entry field.
6. Click the “OK” button.

To edit a note:

1. Double-click the note to display the “Note Properties” dialog box.
2. Modify the note and/or note properties as desired.
3. Click the “OK” button.

To delete a note, select it and press the <Delete> key.



Communication block Button

### 5.2.5 Communication Block

The Communication block allows a flow chart to communicate directly through a serial port on the Runtime with an external device. Each instance of a block handles either a read, write, or control operation. The example shown in Figure 2-5 reads from the serial port into a string named “Drill\_Start”. The tag-name “COM\_1\_CNC\_Link” is a Comm data item that is mapped to a particular serial port using IOView. The device sends a single byte of data.



Figure 5-6 The “Communication Block... Expression” tab

Double-click the Communication block to edit the Communication expression. This displays the “Communication Block... Expression” tab. “Read”, “Write”, and “COM control” operations are independently selectable. In the Communications Read block, you must select at least one of the check box options in the “Termination” group for a valid operation. You can also access an ASCII character chart of hex codes, using the “ASCII Chart” button, to select a special character as a terminator.

When you select a “Write” operation, you can optionally choose terminators that you want appended to the data output. Selecting “COM control” provides a choice of “Control Flags” that let you cancel read or write operations, clear buffers, or set modem control signals.

### Structured data types

Comm data items are “structured” in that they have associated flags. The purpose of these flags is to indicate transmission/reception completion or an error condition. Each flag is identified in the Data item grid in the “Structured Fields” column. The three available flags are “ReadDone”, “WriteDone”, and “Error”. The data item name includes the flag name as a suffix, separated by a dot (for example: “GetMessage.ReadDone”). The flags are not independent flag data items. You can use them in Decision blocks by referencing their complete name as a structured data item.

## 5.2.6 Compare Block



**Compare  
block Button**

The Compare block can make numerical or string comparisons. Flow follows one of the two exit paths of the Compare block based on the result of the comparison. The type of comparisons available are *less than*, *equal to*, *greater than*, *not equal to*, *greater than or equal to*, and *less than or equal to*. For string comparisons, you can further specify if the first string of the compare is considered a substring of the second with the “Instr” (is contained in) function. Optionally, you can use the “Instr” function to perform case-sensitive comparisons.

## Think & Do

---

Double-click the Compare block to edit its expression. This displays the “Compare Block... Expression” tab (see Figure 5-7). This example compares the register with the name “NoOfUnits” to a “Fixed Integer” constant of “4”.



Figure 5-7 The “Compare Block... Expression” tab



When using a constant in the “Compare... Expression” tab, the variable operand must be “Item1” and the constant must be “Item2”.

When Think & Do Runtime executes a Compare block, control will pass through the “Yes” path to the next block in that path if the comparison is TRUE. If FALSE, control passes through the “No” path to the next block in that path.

### Editing a Compare Block

To edit a Compare block expression, follow these steps:

1. While in Selection Mode, double-click the Compare block to open the “Compare Block... Configuration” tab (see Figure 5-7).
2. Click the “Data Type” drop-down to select the data item type.
3. Enter the “Data Item” using one of the following techniques:

- Enter the data item name in the “Data Item” field. While typing, Think & Do attempts to auto-complete the name.
  - Enter the data item “Index”, for example “01”. When tabbing to the next field, Think & Do displays the data item name associated with the index.
  - Use the “Find...” menu button or double-click the field to display the Data Item Grid.
4. Use the “Comparison Type” drop-down list to select the desired comparison.
  5. Use the “Item2” group to select a “Data Type” and “Data Item” (or fixed “Value”).
  6. Click the “OK” button.

### Yes/No path Selection

By default, the connection that goes out the bottom of the Compare block is the “Yes” path; the exit on the right is the “No” path. The “Compare Block... Properties” tab permits adding a block “Description” and swapping the positions the Yes and No exit paths (another way is to right-click a block, and then select “Swap Yes/No Connectors” from the pop-up menu). Entering the block “Number” in the “Yes” or “No” group explicitly sets or modifies the target block of the “Yes” or “No” connector.

### Wait One Scan

Selecting the “Wait One Scan” check box for either the “Yes” or “No” group indicates that the Think & Do Runtime should evaluate the comparison and then wait for the next scan cycle before taking the resulting path.



## Think & Do

### Compare Block Summary

Table 5-2 defines all possible Compare block combinations. Moves that are valid for individual data items are also valid in arrays. You may also move data to Inputs when the flow chart is a Simulation type.

Table 5-2 Valid Compare Block Comparisons

From... To...	1 Output	N Outputs	1 Flag	N Flags	1 Timer	N Timers	1 Register	N Registers	1 Counter	N Counters	1 Number	N Numbers	1 Byte	N Bytes	System	1 Float	N Floats	1 String	N Strings
1 Float	✓	✓	✓	✓	✓	✗	✓	✗	✓	✗	✓	✗	✓	✗	✓	✓	✗	✓	✗
N Floats	✗	✗	✗	✗	✗	=	✗	=	✗	=	✗	=	✗	=	✗	✗	=	✗	=
Float const.	✓	✓	✓	✓	✓	✗	✓	✗	✓	✗	✓	✗	✓	✗	✓	✓	✗	✓	✗
1 Input	✓	✗	✓	✗	✓	✗	✓	✗	✓	✗	✓	✗	✓	✗	✓	✓	✗	✗	✗
N Inputs	✗	=	✗	=	32	✗	16	✗	16	✗	32	✗	8	✗	32	64	✗	✗	✗
1 Output	✓	✗	✓	✗	✓	✗	✓	✗	✓	✗	✓	✗	✓	✗	✓	✓	✗	✗	✗
N Outputs	✗	=	✗	=	32	✗	16	✗	16	✗	32	✗	8	✗	32	64	✗	✗	✗
1 Flag	✓	✗	✓	✗	✓	✗	✓	✗	✓	✗	✓	✗	✓	✗	✓	✓	✗	✗	✗
N Flags	✗	=	✗	=	32	✗	16	✗	16	✗	32	✗	8	✗	32	64	✗	✗	✗
1 Timer	✓	32	✓	32	✓	✗	✓	✗	✓	✗	✓	✗	✓	✗	✓	✓	✗	✓	✗
N Timers	✗	✗	✗	✗	✗	=	✗	=	✗	=	✗	=	✗	=	✗	✗	=	✗	=
1 Register	✓	16	✓	16	✓	✗	✓	✗	✓	✗	✓	✗	✓	✗	✓	✓	✗	✓	✗
N Registers	✗	✗	✗	✗	✗	=	✗	=	✗	=	✗	=	✗	=	✗	✗	=	✗	=
1 Counter	✓	16	✓	16	✓	✗	✓	✗	✓	✗	✓	✗	✓	✗	✓	✓	✗	✓	✗
N Counters	✗	✗	✗	✗	✗	=	✗	=	✗	=	✗	=	✗	=	✗	✗	=	✗	=
1 Number	✓	32	✓	32	✓	✗	✓	✗	✓	✗	✓	✗	✓	✗	✓	✓	✗	✓	✗
N Numbers	✗	✗	✗	✗	✗	=	✗	=	✗	=	✗	=	✗	=	✗	✗	=	✗	=
1 Byte	✓	8	✓	8	✓	✗	✓	✗	✓	✗	✓	✗	✓	✗	✓	✓	✗	✓	✗
N Bytes	✗	✗	✗	✗	✗	=	✗	=	✗	=	✗	=	✗	=	✗	✗	=	✓	=
Fixed Integers	✓	32	✓	32	✓	✗	✓	✗	✓	✗	✓	✗	✓	✗	✓	✓	✗	✓	✗
Current Date	✗	✗	✗	✗	✗	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✓	✗
Current Time	✗	✗	✗	✗	✗	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✓	✗
System	✓	32	✓	32	✓	✗	✓	✗	✓	✗	✓	✗	✓	✗	✓	✓	✗	✓	✗
1 String	✗	✗	✗	✗	✓	✗	✓	✗	✓	✗	✓	✗	✓	✓	✓	✓	✗	✓	✗
N Strings	✗	✗	✗	✗	✗	=	✗	=	✗	=	✗	=	✗	=	✗	✗	=	✗	=
String Const.	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗	✗	✗	✓	✗
Legend	<input checked="" type="checkbox"/> Valid move <input type="checkbox"/> Valid move, if the number of <i>From</i> and <i>To</i> data items are equal <input checked="" type="checkbox"/> Invalid move <input type="checkbox"/> 2 <input type="checkbox"/> 4 <input type="checkbox"/> 8 <input type="checkbox"/> 16 <input type="checkbox"/> 32 <input type="checkbox"/> 64      Can move up to 2, 4, 8, 13, 32, or 64 data items																		



### Control block Button

## 5.2.7 Control Block

The Control block is the most basic building block of a flow chart. It is one of the action blocks of the Think & Do (see “Types of Action Blocks” on page 4-15). Its job is to do the discrete actions listed in its expression list. A Control block is capable of setting bits ON or OFF (flags or outputs), and doing internal timer or counter operations.

Use the “Control Block” dialog box to select the action(s) for the control block:

- Turn a Flag or Output ON or OFF.
- Pulse an Output (where a pulse turns off after a specified duration).



A Pulse output with a duration of 0 (zero) causes the output to be on for a single scan only.

- Start, stop, reset, or restart a Timer.
- Increment, decrement, or reset a counter.



Unlike ladder logic, these actions are *latched*. In other words, when turning an output on, it stays on until the flow chart (or another flow chart) turns it off. The only exception is the pulse output, which automatically turns off after the specified duration (entered in the Data Item grid).

The Control Block can perform up to ten actions. Double-click the Control block to edit its expression. This displays the “Control Block... Expression” tab (see Figure 5-8).

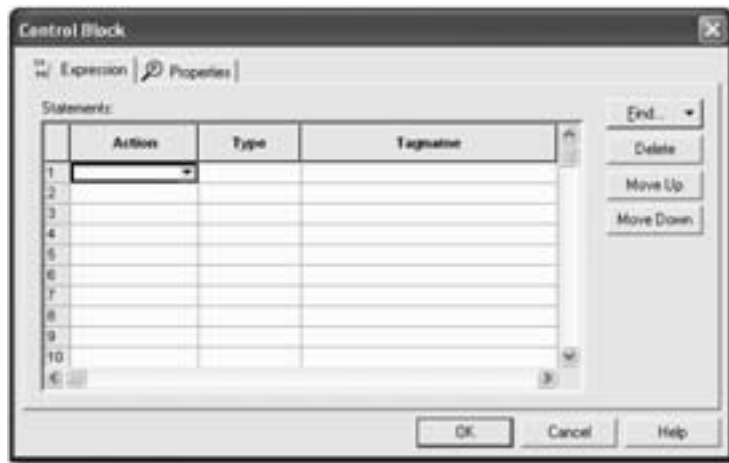


Figure 5-8 The “Control Block... Expression” tab

Since actions in the Control block execute in the order in which they appear, the Control block permits reordering actions. To reorder an action, select it (it appears highlighted), and then use the “Move Up” or “Move Down” buttons to move the selected action. For these buttons to be available, there must be more than one action defined in the block.



The order rarely makes a difference. This feature is primarily to improve readability. It does, however, make a difference when starting a timer, and then resetting that timer, as opposed to the inverse. This is applicable if the timer was previously stopped but did not complete (elapsed time operations).

### Editing a Control block

To edit a Control block’s expression, follow these steps:

1. Double-click the Control block to open the “Control Block... Expression” tab.
2. Use the “Action” drop-down list to select the action desired.
3. Use the “Type” drop-down list to select a data item type that can perform the selected “Action”.
4. Enter the “Tagname” using one of the following techniques:
  - Enter the tagname in the “Tagname” field. While typing, Think & Do attempts to auto-complete the name.
  - Enter the data item “Index”, for example “01”. When tabbing to the next field, Think & Do displays the data item name associated with the index.
  - Use the “Find...” menu button or double-click the field to display the Data Item Grid.
5. Repeat steps 2 through 4 as required for additional expressions. A block must have at least one expression. After entering more than one expression, the “Move Up” and “Move Down” buttons become available. These buttons permit re-ordering the expression, since the Think & Do Runtime executes them in the order they appear in this dialog box. Select one of the expressions, and then click the desired button to move it in the list.
6. Click the “OK” button.

## 5.2.8 Decision Block



**Decision block  
Button**

The Decision block evaluates a Boolean expression as TRUE or FALSE. The expression may consist of one or more bits. The Decision block has one entry point and two exit points (Yes/No), based on the Boolean decision. Use the “Decision Block” dialog box to select the bit(s) to be evaluated, which are of these types:

- Array — when testing a bit in a flag or timer array
- Comm
- Flag
- Input
- Output
- Timer

The AND and OR “Operator” in the dialog box permits additional Boolean conditions in the expression. AND has greater precedence than OR (in other words AND expressions are evaluated before OR expressions, and then they are evaluated from top-down). Double-click the Decision block to edit its expression. This displays the “Decision Block... Expression” tab (see Figure 5-9).

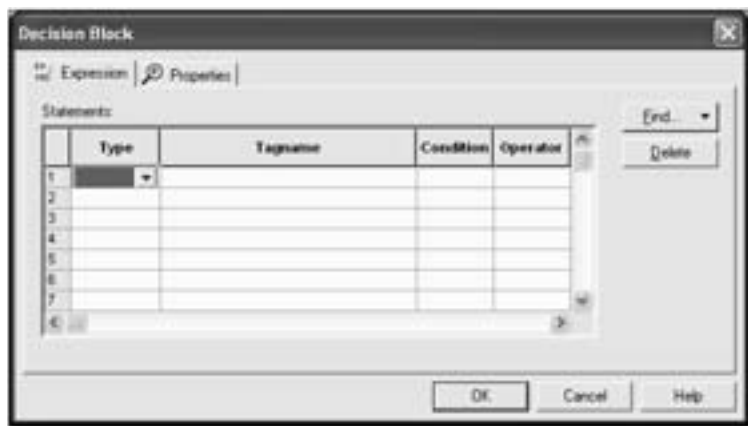


Figure 5-9 The “Decision Block... Expression” tab

### Editing a Decision Block

To edit a Decision block expression, follow these steps:

1. While in Selection Mode, double-click the Decision block to open the “Decision Block... Expression” tab.
2. Use the “Type” drop-down list to select the data item type. Only valid data types appear in the list. Available types are Array, Comm, Flag, Input, Output, and Timer.
3. Enter the “Data Item” using one of the following techniques:
  - Enter the tagname in the “Tagname” field. While typing, Think & Do attempts to auto-complete the name.
  - Enter the data item “Index”, for example “01”. When tabbing to the next field, Think & Do displays the data item name associated with the index.
  - Use the “Find...” menu button or double-click the field to display the Data Item Grid.
4. Use the “Condition” drop-down list to set the condition desired. Conditions that appear are: ON or OFF.
5. Click the “OK” button.

### Yes/No Path Selection

By default, the connection that goes out the bottom of the Decision block is the “Yes” path; the exit on the right is the “No” path. The “Decision Block... Expression” tab permits adding a block “Description” and swapping the positions the Yes and No exit paths (another way is to right-click a block, and then select “Swap Yes/No Connectors” from the pop-up menu). Entering the block “Number” in the “Yes” or “No” group explicitly sets or modifies the target block of the “Yes” or “No” connector.

### Decision Block Summary

Table 5-3 defines all possible Decision block combinations. Moves that are valid for individual data items are also valid in arrays. You may also move data to Inputs when the flow chart is a Simulation type.

Table 5-3 Valid Decision Block Comparisons

From... To...	1 Output	N Outputs	1 Flag	N Flags	1 Timer	N Timers	1 Register	N Registers	1 Counter	N Counters	1 Number	N Numbers	1 Byte	N Bytes	System	1 Float	N Floats	1 String	N Strings
1 Float	✓	✓	✓	✓	✓	✗	✓	✗	✓	✗	✓	✗	✓	✗	✓	✓	✗	✓	✗
N Floats	✗	✗	✗	✗	✗	=	✗	=	✗	=	✗	=	✗	=	✗	✗	=	✗	=
Float const.	✓	✓	✓	✓	✓	✗	✓	✗	✓	✗	✓	✗	✓	✗	✓	✓	✗	✓	✗
1 Input	✓	✗	✓	✗	✓	✗	✓	✗	✓	✗	✓	✗	✓	✗	✓	✓	✗	✗	✗
N Inputs	✗	=	✗	=	32	✗	16	✗	16	✗	32	✗	8	✗	32	64	✗	✗	✗
1 Output	✓	✗	✓	✗	✓	✗	✓	✗	✓	✗	✓	✗	✓	✗	✓	✓	✗	✗	✗
N Outputs	✗	=	✗	=	32	✗	16	✗	16	✗	32	✗	8	✗	32	64	✗	✗	✗
1 Flag	✓	✗	✓	✗	✓	✗	✓	✗	✓	✗	✓	✗	✓	✗	✓	✓	✗	✗	✗
N Flags	✗	=	✗	=	32	✗	16	✗	16	✗	32	✗	8	✗	32	64	✗	✗	✗
1 Timer	✓	32	✓	32	✓	✗	✓	✗	✓	✗	✓	✗	✓	✗	✓	✓	✗	✓	✗
N Timers	✗	✗	✗	✗	✗	=	✗	=	✗	=	✗	=	✗	=	✗	✗	=	✗	=
1 Register	✓	16	✓	16	✓	✗	✓	✗	✓	✗	✓	✗	✓	✗	✓	✓	✗	✓	✗
N Registers	✗	✗	✗	✗	✗	=	✗	=	✗	=	✗	=	✗	=	✗	✗	=	✗	=
1 Counter	✓	16	✓	16	✓	✗	✓	✗	✓	✗	✓	✗	✓	✗	✓	✓	✗	✓	✗
N Counters	✗	✗	✗	✗	✗	=	✗	=	✗	=	✗	=	✗	=	✗	✗	=	✗	=
1 Number	✓	32	✓	32	✓	✗	✓	✗	✓	✗	✓	✗	✓	✗	✓	✓	✗	✓	✗
N Numbers	✗	✗	✗	✗	✗	=	✗	=	✗	=	✗	=	✗	=	✗	✗	=	✗	=
1 Byte	✓	8	✓	8	✓	✗	✓	✗	✓	✗	✓	✗	✓	✗	✓	✓	✗	✓	✗
N Bytes	✗	✗	✗	✗	✗	=	✗	=	✗	=	✗	=	✗	=	✗	✗	=	✓	=
Fixed Integers	✓	32	✓	32	✓	✗	✓	✗	✓	✗	✓	✗	✓	✗	✓	✓	✗	✓	✗
Current Date	✗	✗	✗	✗	✗	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✓
Current Time	✗	✗	✗	✗	✗	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✓
System	✓	32	✓	32	✓	✗	✓	✗	✓	✗	✓	✗	✓	✗	✓	✓	✗	✓	✗
1 String	✗	✗	✗	✗	✓	✗	✓	✗	✓	✗	✓	✗	✓	✗	✓	✓	✗	✓	✗
N Strings	✗	✗	✗	✗	✗	=	✗	=	✗	=	✗	=	✗	=	✗	✗	=	✗	=
String Const.	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗	✗	✗	✓	✗
Legend	<div style="display: flex; justify-content: space-between;"> <div style="width: 45%;"> <p><input checked="" type="checkbox"/> Valid move</p> <p><input checked="" type="checkbox"/> Invalid move</p> </div> <div style="width: 45%;"> <p><input type="checkbox"/> Valid move, if the number of <i>From</i> and <i>To</i> data items are equal</p> <p><input type="checkbox"/> 2   <input type="checkbox"/> 4   <input type="checkbox"/> 8   <input type="checkbox"/> 16   <input type="checkbox"/> 32   <input type="checkbox"/> 64</p> <p>Can move up to 2, 4, 8, 13, 32, or 64 data items</p> </div> </div>																		

### 5.2.9 Do Loop Block



**Do Loop  
Block Button**

Think & Do provides a Do Loop construct (see Figure 5-10), which you can use within a standard flow chart or subchart. Of course, most flow charts loop back to execute part or all of the chart, but that repetition must wait until the next scan. The Do Loop is the only construct that allows blocks to execute more than once during a scan.

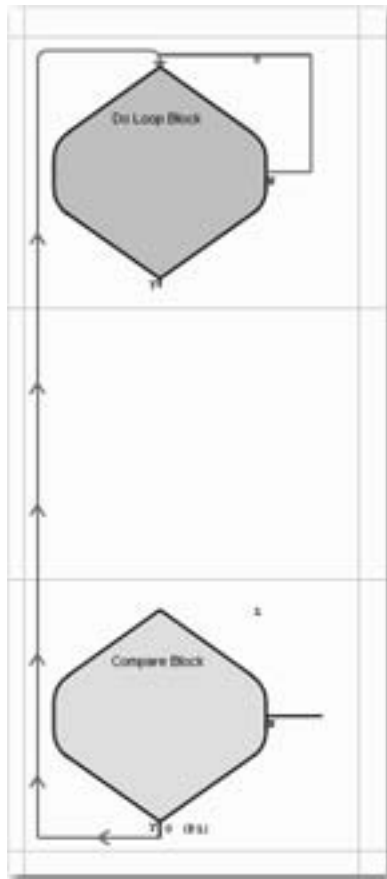


Figure 5-10 A Do Loop consists of Do Loop block and a Compare block



The best uses for Do Loops are for data operations such as manipulating arrays and sorting. Remember that I/O read/write operations occur only once per scan, so trying to pulse outputs rapidly from a Do Loop doesn't work.

The Do Loop and Compare blocks shown in Figure 5-10 have a loop-back line from the exit of the Compare to the entry of the Do Loop block. This connecting line will vary in length, allowing you to place several blocks between the Do Loop and Compare blocks.

The Do Loop has the following rules and characteristics:

- The Do loop executes only if logic flow reaches the Do Loop block.
- You specify a maximum loop count, as a safeguard, in the Do Loop block expression.
- The loop executes the blocks you insert within it each scan until the Compare block expression is satisfied (TRUE if the loop exit is “Yes”; FALSE if the loop exit is “No”).
- The block(s) within the Do Loop cannot have an exit point that connects to another point outside the Do Loop. In other words, the Compare block at the bottom of the loop is the only valid loop exit point.
- To change the Compare block to a Decision block, right-click on the Compare block, and then select “Change this Compare block to a Decision block” from the pop-up menu.



Be aware that looping an extremely large number of times can cause logic solve time to exceed the specified scan interval for the project.

The Do Loop has a user-defined maximum loop count (default=1) as a safety exit to prevent an infinite loop at runtime. You should not use it as a normal loop exit method.

To set a maximum loop count:

1. While in Selection Mode, double-click the Do Loop block to open the “Do Loop Block... Expression” tab.
2. Enter the maximum value, in the field provided, and click the “OK” button.



## Think & Do

After defining the “Maximum Loop Count”, the Do Loop block displays that value as shown in Figure 5-11.

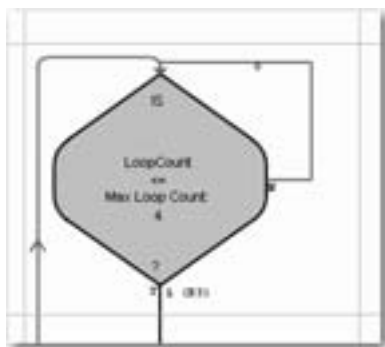


Figure 5-11 Do Loop block showing “Maximum Loop Count”



### Motion Block Button

## 5.2.10 Motion Block

Think & Do’s integrated motion control provides a programming interface for motion. Integrated motion control provides convenience and time savings during system design, since you avoid having to link external motion programs with the project. The system block diagram in Figure 5-12 shows a typical I/O subsystem, along with a motion control system design.

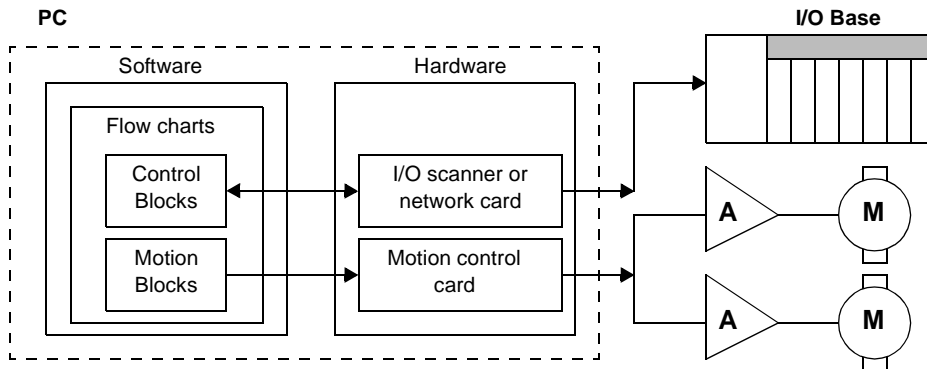


Figure 5-12 Typical I/O subsystem and a motion control systems design

Just as a Control block can set I/O points ON or OFF, a Motion block can issue a command to or request data from a motion control card. In the example “Motion... Expression” tab in Figure 5-13, the command “BeginMoveTo” for “Axis1” has a destination of “Part\_Length”, where “Axis1” and “Part\_Length” are tag-names.



Figure 5-13 The “Motion Block... Expression” tab

There are generic commands and commands that are specific to particular motion systems. By default, all generic and specific commands appear in the list.

A motion system runs asynchronously to the logic solve scans. This means flow charts need to verify that the requested move or action has completed before requesting the next move or action from the motion card.



Think & Do supports several motion cards that appear in IOView. These including Douloi, Galil, and MEI (see the Drivers menu). Check the Phoenix Contact website at [www.phoenixcontact.com](http://www.phoenixcontact.com) for the latest I/O and motion card driver information.

### 5.2.11 Move Block



Move block Button

The Move block simply moves data from one location to another. All data items, including I/O, exist as data items. Therefore, doing a move from one data item to another simply moves (copies) the data of the first data item into the second data item. The value of the first data item is preserved.

The “Move Block... Expression” tab (see Figure 5-14) permits definition of the “From” and “To” data fields. After selecting the “Data Type” from the drop-down list, the “Starting data item” becomes available. The “From” and “To” variables may be of types listed in the table below. Moves that are valid for individual data items are also valid in arrays. You may also move data to Inputs when the flow chart is a Simulation type.



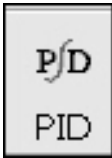
Figure 5-14 The “Move Block... Expression” tab

Table 5-4 Valid Moves

From... To...	1 Output	N Outputs	1 Flag	N Flags	1 Timer	N Timers	1 Register	N Registers	1 Counter	N Counters	1 Number	N Numbers	1 Byte	N Bytes	System	1 Float	N Floats	1 String	N Strings
1 Float	✓	✓	✓	✓	✓	✗	✓	✗	✓	✗	✓	✗	✓	✗	✓	✓	✗	✓	✗
N Floats	✗	✗	✗	✗	✗	=	✗	=	✗	=	✗	=	✗	=	✗	✗	=	✗	=
Float const.	✓	✓	✓	✓	✓	✗	✓	✗	✓	✗	✓	✗	✓	✗	✓	✓	✗	✓	✗
1 Input	✓	✗	✓	✗	✓	✗	✓	✗	✓	✗	✓	✗	✓	✗	✓	✓	✗	✓	✗
N Inputs	✗	=	✗	=	32	✗	16	✗	16	✗	32	✗	8	✗	32	64	✗	✗	✗
1 Output	✓	✗	✓	✗	✓	✗	✓	✗	✓	✗	✓	✗	✓	✗	✓	✓	✗	✓	✗
N Outputs	✗	=	✗	=	32	✗	16	✗	16	✗	32	✗	8	✗	32	64	✗	✗	✗
1 Flag	✓	✗	✓	✗	✓	✗	✓	✗	✓	✗	✓	✗	✓	✗	✓	✓	✗	✓	✗
N Flags	✗	=	✗	=	32	✗	16	✗	16	✗	32	✗	8	✗	32	64	✗	✗	✗
1 Timer	✓	32	✓	32	✓	✗	✓	✗	✓	✗	✓	✗	✓	✗	✓	✓	✗	✓	✗
N Timers	✗	✗	✗	✗	✗	=	✗	=	✗	=	✗	=	✗	=	✗	✗	=	✗	=
1 Register	✓	16	✓	16	✓	✗	✓	✗	✓	✗	✓	✗	✓	✗	✓	✓	✗	✓	✗
N Registers	✗	✗	✗	✗	✗	=	✗	=	✗	=	✗	=	✗	=	✗	✗	=	✗	=
1 Counter	✓	16	✓	16	✓	✗	✓	✗	✓	✗	✓	✗	✓	✗	✓	✓	✗	✓	✗
N Counters	✗	✗	✗	✗	✗	=	✗	=	✗	=	✗	=	✗	=	✗	✗	=	✗	=
1 Number	✓	32	✓	32	✓	✗	✓	✗	✓	✗	✓	✗	✓	✗	✓	✓	✗	✓	✗
N Numbers	✗	✗	✗	✗	✗	=	✗	=	✗	=	✗	=	✗	=	✗	✗	=	✗	=
1 Byte	✓	8	✓	8	✓	✗	✓	✗	✓	✗	✓	✗	✓	✗	✓	✓	✗	✓	✗
N Bytes	✗	✗	✗	✗	✗	=	✗	=	✗	=	✗	=	✗	=	✗	✗	=	✓	=
Fixed Integers	✓	32	✓	32	✓	✗	✓	✗	✓	✗	✓	✗	✓	✗	✓	✓	✗	✓	✗
Current Date	✗	✗	✗	✗	✗	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✓	✗
Current Time	✗	✗	✗	✗	✗	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✓	✗
System	✓	32	✓	32	✓	✗	✓	✗	✓	✗	✓	✗	✓	✗	✓	✓	✗	✓	✗
1 String	✗	✗	✗	✗	✓	✗	✓	✗	✓	✗	✓	✗	✓	✓	✓	✓	✗	✓	✗
N Strings	✗	✗	✗	✗	✗	=	✗	=	✗	=	✗	=	✗	=	✗	✗	=	✗	=
String Const.	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗	✗	✗	✓	✗
Legend	<input checked="" type="checkbox"/> Valid move <input type="checkbox"/> = Valid move, if the number of <i>From</i> and <i>To</i> data items are equal <input checked="" type="checkbox"/> Invalid move <input type="checkbox"/> 2 <input type="checkbox"/> 4 <input type="checkbox"/> 8 <input type="checkbox"/> 16 <input type="checkbox"/> 32 <input type="checkbox"/> 64      Can move up to 2, 4, 8, 13, 32, or 64 data items																		

For more information on how the Move block performs powerful data manipulations, refer to page 7-17 in the “Think & Do User Manual”.

### 5.2.12 PID Block



**PID Block Button**

The PID block performs a mathematical algorithm for process control, which is the Proportional Integral Derivative (PID) algorithm. Think & Do permits up to 64 PID loops in a project.

In the block diagram shown in Figure 5-15, the PID loop has two inputs and one output to the outside world (the process control application). These variables are the Setpoint (SP), the Process Variable (PV), and the Output Variable. These data items also appear in the “PID... Expression” tab (see Figure 5-16).

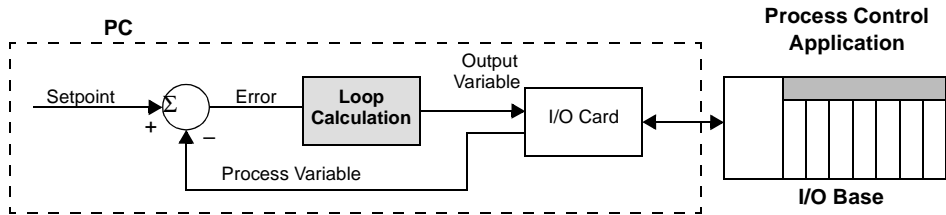


Figure 5-15 PID loop block diagram



Figure 5-16 The “PID Block... Expression” tab

You must design your flow chart to execute PID Blocks at a frequency suitable for the process control application. This frequency, or loop sample rate, can often be much slower than the normal scan interval. The proper value conserves system processing power while still guaranteeing loop stability. Therefore, the PID Block must generally be placed inside a timing loop in your flow chart. It is a good idea to put your PID Block in a subchart and call it when needed.



A PID block has a dialog box with several tabs to configure. These are covered in detail in the online help and “PID for Process Control User Manual.” This document is installed as part of the standard Think & Do setup.

### 5.2.13 Run Program Block



Run Program Block Button

The Run Program block lets you start other Windows programs or batch files under the flow chart control. This is one of the most powerful aspects of having a PC-based control platform. The “Run Program Block... Expression” tab (see Figure 5-17) has parameters that specify the program path, arguments, runtime process parameters, and an error value. You may also specify these parameters indirectly using string data items. The external program runs while the flow chart continues normal execution. The Run Program block can start a custom C++ or Basic program or any “.exe” (executable) file.



Figure 5-17 The “Run Program Block... Expression” tab



The flow chart design must take into account the execution time of the program it calls. Be aware of potential problems with executing Run Program every scan. It's easy to inadvertently start a program every scan.

## 5.2.14 SQL Block



**SQL Block  
Button**

The SQL block provides a dialog box that helps you construct SQL queries for database access.

An enterprise-enabled Think & Do project is one that integrates factory floor control with enterprise-level operations and databases related to custom orders, inventory, parts procurement, warehousing, shipping, and so on. The SQL block gives Think & Do the ability to communicate in realtime over large corporate networks to enterprise systems.

Double-click a SQL block to display its expression dialog box (see Figure 5-18), which has the following tabs:

- General — configures the request type, record type, and query result data items
- Tag Mappings — map record fields to project tagnames
- Filters (Where) — specify operators and conditions for field names
- Sorting (Order By) — specifies field names and the sorting order
- SQL Statement — displays the resulting SQL statement (cannot be directly edited — use the tabs to change the SQL statement)

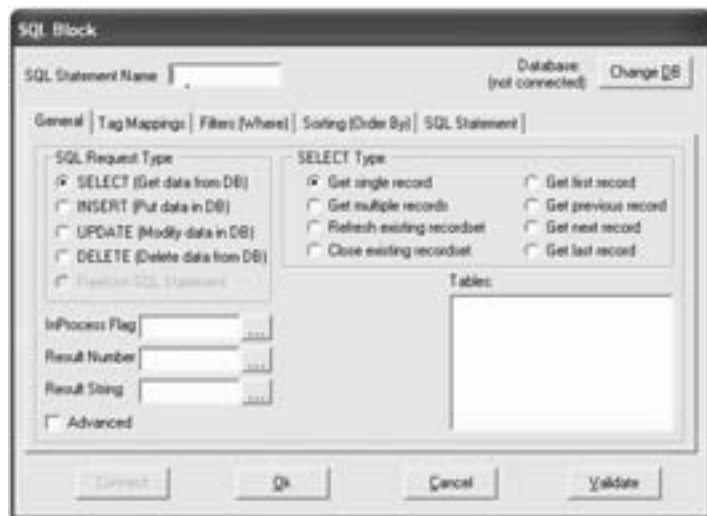


Figure 5-18 The “SQL Block... General” tab

For a full discussion on SQL database operations, see page 7-39 in the “Think & Do User Manual”.



### 5.2.15 Wait Block



Wait block Button

The Wait block inserts a wait period in the execution of the flow chart. Configure the wait period in milliseconds. Select the “One Scan Wait” check box to have the block for exactly one scan cycle. Setting the “Wait Time” to zero automatically sets the “One Scan Cycle” check box.

To configure the wait time:

1. Double-click the Wait block to display the “Wait Block... Expression” tab (see Figure 5-19).

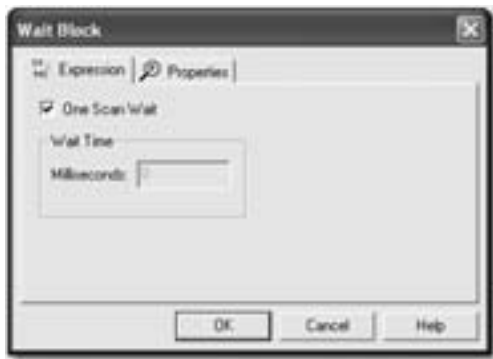


Figure 5-19 The “Wait Block... Expression” tab

2. Do one of the following:
  - To wait for one scan cycle, select the “One Scan Wait” check box.



The duration changes to zero (the next time the “Wait Block” dialog box opens) if it was previously non-zero, and “Wait Time” becomes unavailable.

- To wait for a specified period of time, enter the number of milliseconds to wait in the field.



If the duration is zero, the Wait block defaults to one scan cycle. The Wait block automatically selects the “One Scan Wait” check box upon clicking the “OK” button.

3. Click the “OK” button.

During the wait period, many scan periods may pass. However, the flow chart with the Wait block will not execute further until the wait period ends. On the scan when the wait time expires, flow chart execution resumes at the next block following the Wait block.

---

The Wait block is easier to use than constructing a timing loop for a delay in the flow chart, since there is no need to create a timer data item or create the flow chart loop. When a Wait block is executing, no other blocks in the current flow chart execute until the wait completes. Other flow charts continue to execute normally as long as they are enabled.

## 5.3 Using Subcharts

A flow chart can call a subchart, which transfers execution to the subchart. The subchart returns to the original flow chart, transferring execution back again. A subchart can also call another subchart. However, a subchart cannot call itself (recursion is not allowed). Flow charts can nest subchart calls up to 16 levels for tremendous flexibility and support for large projects.

To start a new subchart:

1. Do one of the following:
  - From the ProjectCenter main menu, select the “File... New... Subchart” menu.
  - From the FlowView main menu, select the “File... New... Subchart” menu.
2. Either action displays a dialog that prompts you to name the subchart. After naming it, the subchart appears in the “Subchart” folder in ProjectCenter’s Flow chart explorer, and an empty subchart appears in FlowView.

### 5.3.1 Blocks for Subcharts

When creating a subchart, the first block must be a Begin block and the last one a Return block. FlowView creates these blocks for you. Naturally, the remainder of the subchart you create must go between and connect to these two blocks. There can be only one Begin and Return block for a subchart (they cannot be copied, pasted, or deleted). The properties of the Begin block determine how the subchart communicates with flow charts that call it.

In discussing subcharts, we use the word parameters in a special sense. Figure 5-20 shows a representation of how data values pass between calling and called programs as parameters. Flow charts call subcharts and pass parameters by reference or value. When passed by reference, the subchart is actually reading and writing the data item used by the calling flow chart. When passed by value, the subchart receives the value of the calling parameter at the time of the call. The subchart uses a local copy that does not directly impact the calling flow chart’s data. When the subchart reaches its first loopback point,

## Think & Do

its execution pauses for the current scan. The calling flow chart resumes execution when the subchart executes its Return block. Upon return, the calling flow chart is free to use the new values set by the subchart.

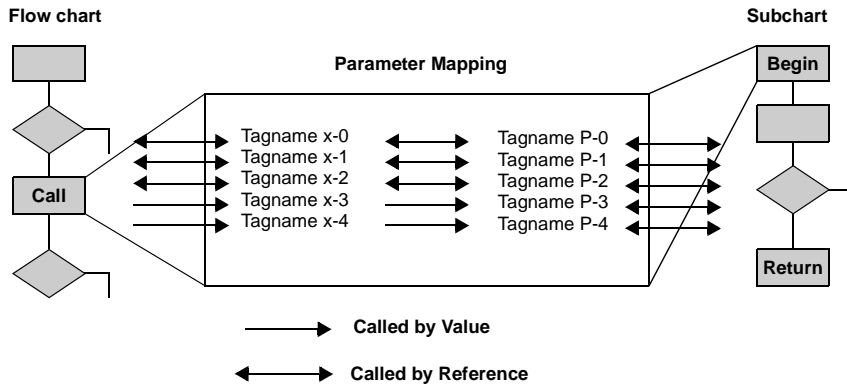


Figure 5-20 Calling flow chart passes values to subcharts via parameters

The benefits of parameter mapping are many. It allows you to create a generic subchart with generic parameter names. When used in a project, many other flow charts can use their own tagnames and call the subchart. As long as the calling flow chart has properly mapped all the parameters defined in the subchart, the subchart has all the data it needs to function.



A subchart has to exist before it can be called. In fact, it's best to completely write the subchart and define its parameters before configuring the call block of the calling flow chart.

### 5.3.2 Begin Expression

Double-click the Begin block to edit the Begin Expression expression dialog shown in Figure 5-21.

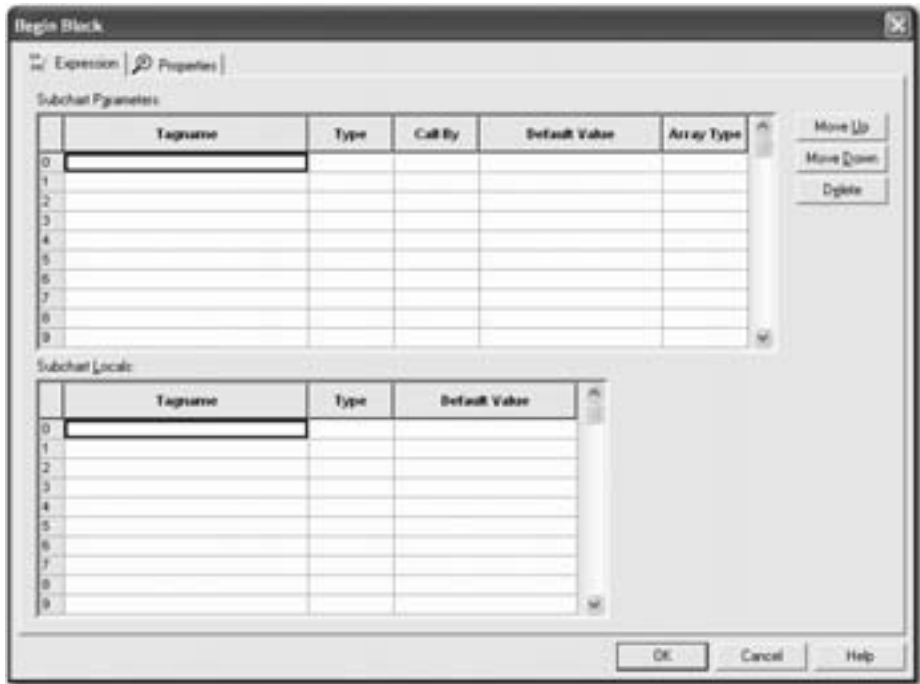


Figure 5-21 The “Begin Block... Expression” tab defines parameters and locals

You’ll notice first that the Begin Block expression dialog has two lists:

- “Subchart Parameters”: These data items map externally to data items of the calling flow chart. Parameter data items are assigned unique names that can be up to 30 characters long.
- “Subchart Locals”: Local data items are only used internally in the sub-chart. Local data items are assigned unique names that can be up to 30 characters long.

### Parameters

All fields for each parameter, except “Default Value”, become drop-down lists when you select in them. In the “Call By” field, you can choose either “Reference” or “Value”. Use call by reference when the calling flow chart needs to see changes made by the subchart. The subchart’s data item references the calling flow chart’s data item, so they change together. Use call by value when you want the subchart to use the data item as a copy. The “Default Value” entry only applies to data items called by value. For ease of use, Think & Do assigns this default value when the calling flow chart does not assign a value for it in its Call block. The default value field does not apply for call by reference data items. You use the “Array Type” field entry to specify the type of array, when you select “Array” from the “Type” entry and use call by reference.

### Locals

Since locals are accessible only within a subchart, there is no call by selection. The default value field assigns a value for the local data item each time the subchart is called. During execution, the subchart can modify this value as necessary.

The “Move Up” and “Move Down” buttons on the right side of the “Begin... Expression” tab provide extra tools to manage subchart parameters and locals. These buttons let you arrange data items in the best order for readability. Parameter order does not affect subchart functionality. You also can select a data item (use the row header) and perform cut-copy-paste operations.

### 5.3.3 Call Expression

A flow chart (standard or subchart) executes a subchart when the control passes to a Call block. Double-click the Call block to display its expression dialog as shown in Figure 5-22.

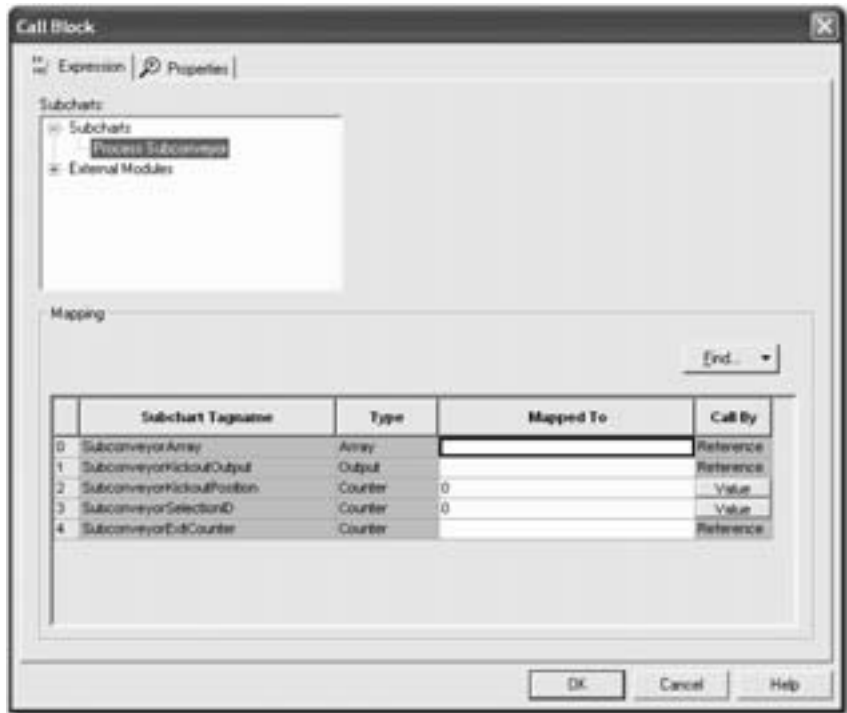


Figure 5-22 The “Call Block... Expression” tab

#### Subchart Name

Use the “Subcharts” tree-structure list to select the subchart you want to call. You must have previously created the subchart, named it, specified any parameters that it requires, and saved it before it appears in this list box. The list includes all existing subcharts in a project and pre-defined “External Modules”. Standard flow charts in the “Unused” category do not appear in the list.

### External Modules

The “External Modules” branch includes the following entries:

- “AutomationDirect.com Ethernet I/O” includes custom commands for the CTRIO module.
- “Cognex” includes the Cognex vision system functions.
- “Interbus Special Functions” includes the PCP functions.
- “Modbus Master” includes custom commands for supporting Modbus Master I/O.
- “Utilities” includes array, string, timer, and inverse trigonometric functions.

For detailed descriptions of these external functions, see the online Help.

### Parameter Mapping

When you select the subchart name, the “Mapping” group lists all subchart parameters. All you need to do is use the “Mapped To” field to map (equate) data items of the calling flow chart to those of the subchart. The tagnames you place in the “Mapped To” field are generally different from the subchart parameter names, but a difference is not required.

### Call By

This field shows whether the parameter is call by reference or call by value. If call by value, there are two options with “Value” as the default:

- Pass a constant value (enter the value). Click the “Tag” button to enter a value. This toggles the button title to “Value”.
- Pass the value of a tagname. Click the “Value” button to enter a tagname. This toggles the button title to “tag”.

### 5.3.4 Interaction of Call and Begin Blocks

The selections you make for each data item in a subchart's Begin block configuration determines the available (and relevant) choices in the calling flow chart's Call block. This means you need to specify the Begin block configuration before configuring related Call block(s).

Table 5-5 Begin and Call Block Parameter Relationships

Expression choices			
Begin (in Subcharts)			Call (in Flow charts)
Parameters	Pass by Reference	N/A	Call by Reference
	Pass by Value	Default value	Call by Value
			Call by Tag
Locals	N/A	Default value	N/A

An explanation of Table 5-5 (line-by-line) follows:

- A subchart's parameters allow exchange of data between a subchart and the calling flow chart. A subchart's locals are not accessible from any other flow chart. The subchart's locals are not retained (remembered from one call to the next call of the subroutine).
- A parameter passed by reference must be called by reference. This method of mapping allows a flow chart's data item to reference (map) to a different subchart tagname. So, when you change the value of the subchart tagname, it changes the value of the calling flow chart's data item.
- A parameter passed by value is a copy of the value of the original data item. The value of the original data item remains unchanged by the subchart. It may be called by value, or called by tag. When called by value, the flow chart simply uses the value at the time of the call. When called by tag, the flow chart uses a tagname to point to a data item for the subchart to accept as a constant.
- Subchart locals have a default value (value at the time the subchart is called). The subchart can subsequently modify that value, but all this is transparent to the calling flow chart. Therefore, a "Call Block... Expression" tab will not show any locals of the subchart being called.

### 5.3.5 Subchart Design Summary

The main points of using subcharts in a project are:

- Only a standard flow chart has an Enable block, permitting it to run automatically at run time (if enable conditions are satisfied).



## Think & Do

---

- A subchart only runs if it is called. Either a standard flow chart or another subchart can call a subchart.
- The first block in a subchart is the Begin block, and the last one is the Return block. The subchart program you write goes between these blocks.
- The “Begin Block... Expression” tab defines parameters and locals for the subchart.
- If a subchart operates only on global tagnames (in the project’s Data Item grid), you do not have to define parameters or locals in the “Begin Block... Expression” tab. However, this is generally not good design practice, since the subchart will be unsuitable for generic use by several calling flow charts.
- The subchart must exist and be saved to disk before you can edit the Call block parameters in the calling flow chart.
- Parameter mapping lets a subchart create a temporary working copy of the calling flow chart’s data items using different local names. The mapping is defined in the “Call Block... Expression” tab of the calling flow chart.

### 5.3.6 Subchart Runtime Characteristics

The following traits show how subcharts work when you run the project:

- A single subchart can be called by one or more flow charts (standard or subchart) in a project, even at the same time. This multiplies a subchart’s capacity to do work in the project.
- Since subcharts can be called from many places, each call is termed a “context.” When debugging such a subchart, FlowView prompts you to choose a debug context — asking you which call you want to debug.
- A flow chart or subchart that is executing has an “Active” status.
- A flow chart or subchart that is currently executing a Wait block has a “Waiting” status until the block times out.
- A single call to a subchart does not constitute parallel processing. Specifically, the calling flow chart has an “In Subchart” status until the subchart completes (returns). If the subchart is down two or more levels, all flow charts above it have an “In Subchart” status until it returns.
- A subchart that is not the destination of any Call block will not appear in the Flow Chart Execution Path.
- A subchart that is the destination of a Call block will not execute until the Call block executes. The subchart has an “Out of Scope” status until it is called. If that subchart is called, its status becomes “Active”. If it calls yet another subchart, its status becomes “In Subchart”.

### 5.3.7 When to Use Subcharts

The real efficiency in using subcharts is their self-duplicating nature at runtime, and your opportunities to re-use them from project to project. Consider writing a generic subchart to do a frequently needed function in your industry. Thus, you save programming time each time you can re-use the subchart in various projects (consider creating a subchart library for use in your company).

One misconception about subcharts suggests a flow chart can call a subchart and then continue doing other things. This is not true. While a subchart executes, the calling flow chart is suspended until the subchart returns. If you need to solve logic in parallel each scan, you must use separate standard flow charts for each logic section.



Think & Do has comprehensive flow chart debugging capabilities. Refer to page 10-3 in the "Think & Do User Manual" for details.

## Think & Do

---

# A Appendix

## A 1 System Data Items

The table in this appendix describes System Data Items that are applicable to Windows-based PCs. While they also apply to the Think & Do WinPLC Embedded CE device, not all of them apply to all Windows CE-based target devices.

\*Indicates an error message. For information on interpreting errors, see “Notes on Reading Error Data” on page A-7

ID	System Data Item	Description
SYS-0	ScanInterval	Read only. The last runtime scan interval. This time is in milliseconds.
SYS-1	PeakScanInterval	The peak, or largest, scan interval encountered by the runtime. Each time the runtime updates ScanInterval, it compares it to PeakScanInterval, if ScanInterval is larger, it is written to PeakScanInterval. Typically you would set this value to zero to clear the current PeakScanInterval and observe the new value. This value is stated in milliseconds.
SYS-2	AvgScanInterval	Read only. The average scan interval of the runtime. This value is stated in milliseconds.
SYS-3	OverScanRatio	Read only. The ratio of scan intervals that are over the desired scan time for the application. This value is stated in tenths of a percent. For example, a value of 12 means that 1.2 percent of the scan intervals were over the specified scan interval.
SYS-4	LogicTime	Read only. The time taken by the runtime system to read the inputs, execute the flow charts, and write the outputs for the last scan. This value is stated in milliseconds.

## Think & Do

ID	System Data Item	Description
SYS-5	PeakLogicTime	The peak, or largest, logic time encountered by the runtime. Each time the runtime updates LogicTime, it compares it to PeakLogicTime, if LogicTime is greater, PeakLogicTime is updated with the new value. Typically you will set this value to zero to clear the PeakLogicTime and observe the new value. This value is stated in milliseconds.
SYS-6	AvgLogicTime	Read only. The average logic time required of the runtime. This value is stated in milliseconds.
SYS-7	Reserved7	Read only. Reserved for internal use.
SYS-8	Reserved8	Read only. Reserved for internal use.
SYS-9	Reserved9	Read only. Reserved for internal use.
SYS-10	Reserved10	Read only. Reserved for internal use.
SYS-11	ArrayIndexError*	Contains error information on flow chart blocks that illegally index into arrays. This value is written by the runtime whenever flow charts try to use an invalid array index. The error code is set to 1 if the array index is too small and set to 2 if the array index is too large. See "Notes on Reading Error Data" on page A-7 for information on decoding this error.
SYS-12	DoLoopExit*	Contains error information on Do Loops. This value gets written to by the runtime when a flow chart exits a Do Loop because it went through the loop the maximum number of times. The error code is set to 1 to indicate the error. See "Notes on Reading Error Data" on page A-7 for information on decoding this error.
SYS-13	ScanCount	The running counter of the number of scans executed by the runtime. This is a 32-bit unsigned number, so it will eventually wrap around to zero. Do not write to this value. It may be used by a runtime component or other application.

ID	System Data Item	Description
SYS-14	Reserved14	Read only. Reserved for internal use.
SYS-15	DataLogging Status	Is used by the Think & Do Data Logging runtime to indicate the status of the data logging task. Do not write to this data item. See Data Logging in help for more details.
SYS-16	RollingMsecCounter	Read only. Counts the number of milliseconds since the project runtime was started. This is a 32-bit unsigned number, so it will eventually wrap around to zero.
SYS-17	LastShutdown	Read only. Contains an encoded time stamp of the last time the runtime shut down. It is valid only when the runtime is configured for retentive data items. Bits 0-7=hour, bits 8-15=minute, bits16-23=second, bits 24-31=day of the week.
SYS-18	Reserved18	Read only. Reserved for internal use.
SYS-19	PowerFailShutdown	When the value is set, the logic stops executing, I/O stops scanning, and data items marked as retentive are written to disk. See online Help for more information.
SYS-20	Reserved20	Read only. Reserved for internal use.
SYS-21	Reserved21	Read only. Reserved for internal use.
SYS-22	Reserved22	Read only. Reserved for internal use.
SYS-23	Reserved23	Read only. Reserved for internal use.
SYS-24	Reserved24	Read only. Reserved for internal use.
SYS-25	Reserved25	Read only. Reserved for internal use.
SYS-26	Reserved26	Read only. Reserved for internal use.
SYS-27	Reserved27	Read only. Reserved for internal use.
SYS-28	Reserved28	Read only. Reserved for internal use.
SYS-29	OverScans	Is a 32-bit counter containing the number of times the measured scan interval exceeded the desired scan interval. This is a 32-bit unsigned number, so it will eventually wrap around to zero.

## Think & Do

---

ID	System Data Item	Description
SYS-30	LastIOTime	Read only. Is the time the runtime took to read the inputs and write the outputs on the last scan.
SYS-31	PeakIOTime	Is the peak LastIOTime encountered by the runtime. Each time the runtime updates LastIOTime, it compares it to PeakIOTime, if LastIOTime is larger, it is written to PeakIOTime. Typically the user would set this value to zero to clear the current PeakIOTime to observe the new value. This value is in milliseconds.
SYS-32	ClockMin	Read only. Toggles between zero and one every minute. It is zero for 30 seconds and one for the next 30 seconds. It repeats this pattern forever.
SYS-33	ClockSec	Read only. Toggles between zero and one every second. It is zero for 0.5 seconds and one for the next 0.5 seconds. It repeats this pattern forever.
SYS-34	Clock100ms	Read only. Toggles between zero and one every 100 milliseconds. It is zero for 50 milliseconds and one for the next 50 milliseconds. It repeats this pattern forever.
SYS-35	Clock50ms	Read only. Toggles between zero and one every 50 milliseconds. It is zero for 25 milliseconds and one for the next 25 milliseconds. It repeats this pattern forever.
SYS-36	FirstScan	Read only. Is one on the very first scan of the project, from then on it is zero.
SYS-37	AltScan	Read only. Toggles between zero and one every scan. It is zero for a scan and one for the next scan. It repeats this pattern forever.
SYS-38	IOStatus	Read only. Contains information on the status of the I/O. A value of zero means the I/O is working normally. A value of one means the runtime is running in simulated I/O mode. A value of two means there is an I/O fault.

ID	System Data Item	Description
SYS-39	MathError*	The error code is set to 1 on an overflow, underflow, divide by zero, and other math errors. See “Notes on Reading Error Data” on page A-7 for information on decoding this error.
SYS-40	Reserved40	Read only. Reserved for internal use.
SYS-41	StringError*	Indicates the flow chart block, the string error code, and the flow chart number where the last error occurred. Error = 1 if string data item could not be read from data table Error = 2 if internal string buffer overflow Error = 3 if string written to data table is truncated because it is longer than the maximum length specified in the data view grid Error = 4 if illegal offset into string. See “Notes on Reading Error Data” on page A-7 for information on decoding this error.
SYS-42	ConversionError*	Indicates the flow chart block, the error code, and the flow chart number where the last error occurred. Conversion errors are generated when converting between different data types and the destination is too small for the value being written to it. See “Notes on Reading Error Data” on page A-7 for information on decoding this error.
SYS-43	Write RetentiveData	Your flow chart or screen can set this bit = 1 to cause the runtime project to save retentive data items to disk. The runtime system automatically resets this bit after saving the retentive data.
SYS-44	Reserved44	Read only. Reserved for internal use.
SYS-45	Reserved45	Read only. Reserved for internal use.



## Think & Do

ID	System Data Item	Description
SYS-46	ExtendedFunctionBlockError*	Block, error code and chart number where last error occurred. Error = 0 if timeout Error = 1 to 15 if error is returned from extended block. See "Notes on Reading Error Data" on page A-7 for information on decoding this error.
SYS-47	ScreenInputIgnored	All HMI screen data entry is ignored when this value = 1.
SYS-48	Reserved48	Read only. Reserved for internal use.
SYS-49	Reserved49	Read only. Reserved for internal use.
SYS-50	LastExtendedFunctionBlockTime	Read only. Execution time (microseconds) of the last Extended Function Block call.
SYS-51	PeakExtendedFunctionBlockTime	Peak execution time (microseconds) of all Extended Function Block calls.
SYS-52	AvgExtendedFunctionBlockTime	Read only. Average execution time (microseconds) of all Extended Function Block calls.
SYS-53	Spare1	Reserved for internal use.
SYS-54	Spare2	Reserved for internal use.
SYS-55	Spare3	Reserved for internal use.
SYS-56	Spare4	Reserved for internal use.
SYS-57	LiveCaptureStatus	Status of the Live! capture set(s). Do not write to this data item. See Live! capture in help for more details.
SYS-58	MajorBuildVersion	Read only. Major build version number — read only
SYS-59	MinorBuildVersion	Read only. Minor build version number — read only
SYS-60	MaximumFlowchartTime	Read only. The maximum flow chart execution time (one component of the time reported in SYS-1). This value is stated in milliseconds.

---

## A 2 Notes on Reading Error Data

Error data contains the error code and the block and flow chart where the last error occurred. The bits in the error data can be read as follows:

- Bits 0–15 contain a reference to the flow chart that has the error. If the error occurred in a standard flow chart, you can find the number in the “Standard 1,2,3...” folder of the flow chart Explorer in ProjectCenter. If the error occurred in a subchart, the number will be in the 10000 range. Subchart numbers are available in the “Chart Properties” dialog box, which you open by selecting the “File... Chart Properties” menu in FlowView. For simulation flow charts, you can calculate the flow chart number by subtracting the number of simulation flow charts found before it in the list. The runtime system writes to the system data item whenever it encounters an error, so if a project contains several errors, only the last one is captured in the data item.
- Bits 20–31 contain the block number. When an error occurs in the Enable or Freeze Blocks, the Runtime identifies the block as block 0.
- Bits 16–19 contain the error code.

## Think & Do

---

## A 3 List of Figures

Figure 1-1:	The Think & Do “Help” menu .....	1-6
Figure 1-2:	PCs connect to I/O bases in racks.....	1-7
Figure 1-3:	A PC can connect to a network with drops to devices .....	1-8
Figure 1-4:	PC connected to multiple I/O controllers .....	1-8
Figure 1-5:	WinPLC and its one base of I/O modules .....	1-9
Figure 1-6:	Group development system .....	1-11
Figure 1-7:	Local PC used for development and Runtime .....	1-12
Figure 1-8:	Development and Runtime control on different systems	1-12
Figure 2-1:	Project components .....	2-3
Figure 2-2:	ProjectCenter provides access to Think & Do development tools .....	2-4
Figure 2-3:	The “Rename Project” dialog box with a name filled in.....	2-5
Figure 2-4:	Use the “Save As” dialog box to define the project name .	2-5
Figure 2-5:	The “Project” Explorer bar provides access to Think & Do tools .....	2-7
Figure 2-6:	The “Flow Charts” and “Screens” Explorer bars .....	2-8
Figure 2-7:	The “Data Items” Explorer bar .....	2-9
Figure 2-8:	The “Find...Configure CE Runtime” dialog box .....	2-11
Figure 2-9:	Flow chart showing an action and branching block .....	2-12
Figure 2-10:	“Name Flow Chart” dialog box .....	2-13
Figure 2-11:	“Screen Attributes” dialog box .....	2-15
Figure 3-1:	ProjectCenter Data Items .....	3-4
Figure 3-2:	The Data Items grid for Counter data types .....	3-5
Figure 3-3:	IOView window includes a board view, module view, and tab view .....	3-7
Figure 3-4:	The “Add I/O Driver” dialog box and its driver listing.....	3-9
Figure 3-5:	IOView displays an image of the I/O scanner card or Runtime target.....	3-10
Figure 3-6:	IOView showing the “Module Info” tab.....	3-11

## Think & Do

---

Figure 3-7:	IOView showing the "I/O Mapping" tab for selected module .....	3-13
Figure 3-8:	Physical I/O points map to logical I/O points in any sequence .....	3-14
Figure 3-9:	The on-screen images in IOView reflect input and output states .....	3-16
Figure 3-10:	Connection-related buttons available in the toolbar.....	3-17
Figure 3-11:	"Add" and "Remove" driver buttons .....	3-18
Figure 3-12:	"Add," "Remove," "Insert" and "Replace" device buttons	3-19
Figure 4-1:	Flow chart segment showing an action and branching block .....	4-3
Figure 4-2:	A Call block calls a subchart, which returns to the original flow chart.....	4-4
Figure 4-3:	The "Project" Explorer bar showing the Flow Chart explorer .....	4-5
Figure 4-4:	Heirarchical sublist list collapsed (left), expanded (right) .	4-6
Figure 4-5:	"Name Flow Chart" dialog box.....	4-7
Figure 4-6:	Select the "Unused" or "Simulation" check box to move a flow chart.....	4-8
Figure 4-7:	FlowView showing a new Flow Chart .....	4-9
Figure 4-8:	Zoom tools in FlowView .....	4-12
Figure 4-9:	Open multiple flow charts in FlowView, each with its own window .....	4-13
Figure 4-10:	Decision, Compare, and Do Loop blocks appear as diamonds .....	4-17
Figure 4-11:	Sample flow chart with flow lines connecting all blocks..	4-19
Figure 4-12:	"Compare Block... Properties" tab .....	4-20
Figure 4-13:	Use the "Screen Attributes" dialog box to set up a new screen .....	4-22
Figure 4-14:	ScreenView development environment.....	4-24
Figure 4-15:	ScreenView Explorer.....	4-25
Figure 4-16:	The "Object Properties" dialog box.....	4-26
Figure 4-17:	The "Replace" dialog box .....	4-27
Figure 4-18:	The "Colors" dialog box .....	4-29

## List of Figures

---

Figure 4-19:	The “Colors... Custom” tab .....	4-30
Figure 4-20:	The “Fill Effects” dialog box .....	4-31
Figure 5-1:	Every new flow chart opens with a predefined Enable block .....	5-4
Figure 5-2:	The “Enable Block... Expression” tab .....	5-4
Figure 5-3:	The “Calculation Block... Expression” tab.....	5-6
Figure 5-4:	The “Call Block... Expression” tab .....	5-8
Figure 5-5:	“Note Properties” dialog box.....	5-9
Figure 5-6:	The “Communication Block... Expression” tab .....	5-10
Figure 5-7:	The “Compare Block... Expression” tab.....	5-12
Figure 5-8:	The “Control Block... Expression” tab.....	5-15
Figure 5-9:	The “Decision Block... Expression” tab .....	5-17
Figure 5-10:	A Do Loop consists of Do Loop block and a Compare block .....	5-20
Figure 5-11:	Do Loop block showing “Maximum Loop Count”.....	5-22
Figure 5-12:	Typical I/O subsystem and a motion control systems design .....	5-22
Figure 5-13:	The “Motion Block... Expression” tab .....	5-23
Figure 5-14:	The “Move Block... Expression” tab .....	5-24
Figure 5-15:	PID loop block diagram .....	5-26
Figure 5-16:	The “PID Block... Expression” tab .....	5-27
Figure 5-17:	The “Run Program Block... Expression” tab .....	5-28
Figure 5-18:	The “SQL Block... General” tab .....	5-29
Figure 5-19:	The “Wait Block... Expression” tab .....	5-30
Figure 5-20:	Calling flow chart passes values to subcharts via parameters .....	5-32
Figure 5-21:	The “Begin Block... Expression” tab defines parameters and locals .....	5-33
Figure 5-22:	The “Call Block... Expression” tab .....	5-35

## Think & Do

---

## A 4 List of Tables

Table 1-1:	Manual Conventions for Keyboard Command.....	1-5
Table 1-2:	Help Sources .....	1-6
Table 4-1:	“Object Properties” dialog box properties .....	4-26
Table 4-2:	Object Properties “Replace” dialog box parameters.....	4-28
Table 5-1:	Valid Enable Block Expressions.....	5-5
Table 5-2:	Valid Compare Block Comparisons.....	5-14
Table 5-3:	Valid Decision Block Comparisons.....	5-19
Table 5-4:	Valid Moves .....	5-25
Table 5-5:	Begin and Call Block Parameter Relationships .....	5-37



## Think & Do

---

# B Index

## A

Action block..... 4-3

## B

Begin block..... 5-31

    Begin expression ..... 5-33

    Interaction with Call block. 5-37

Block

    Comments ..... 4-20

    Multiple selection ..... 4-14

    Selecting..... 4-14

Branching block..... 4-3

## C

Calculation block ..... 5-6

Call block..... 5-8

    Call expression ..... 5-35

    Interaction with Begin block.....  
    5-37

Certified PC..... 1-10, 1-11

Column labels..... 4-10

Command button..... 1-4

Communication block..... 5-10

Compare block ..... 5-11

    Editing..... 5-12

    Summary ..... 5-14

Configuring I/O ..... 3-6

Connecting flow chart blocks.. 4-18

Connections ..... 4-18

    Multiple selection ..... 4-14

    Selecting..... 4-14

ConnectivityCenter ..... 3-6

    Toolbar functions ..... 3-17

Control block ..... 5-15

    Editing..... 5-16

Customer support ..... 1-14

Cut-Copy-Paste ..... 4-14

## D

Database operations..... 5-29

Decision block..... 4-16, 5-17

    Editing ..... 5-18

    Summary ..... 5-18

Development tools ..... 1-9

Devices ..... 3-3

Dialog..... 1-4

Do Loop block ..... 5-20

Drawing area..... 4-9, 4-10

Drivers..... 3-3

## E

Editing flow chart block expressions..... 4-20

Enable block ..... 4-15, 5-4

Error

    Error code interpretation..... A-7

Explorer Bar ..... 2-7

## F

Filldown I/O mapping feature .. 3-15

Flow chart

    Calculation block ..... 5-6

    Call block..... 5-8

    Communication block..... 5-10

    Compare block ..... 5-11

    Connecting ..... 4-18

    Connecting blocks ..... 4-18

    Control block ..... 5-15

    Decision block ..... 5-17

    Do Loop block ..... 5-20

- Editing block expressions . 4-20
- Enable block ..... 4-15, 5-4
- Expressions ..... 4-20
- Freeze block ..... 4-15
- Motion block ..... 5-22
- Move block ..... 5-24
- Order of execution ..... 4-7
- PID bLock ..... 5-26
- Run Program block ..... 5-28
- Simulation ..... 4-8
- SQL Express block ..... 5-29
- Unused..... 4-8
- Wait block ..... 5-30
- FlowView
  - Orientation ..... 4-9
- Freeze block ..... 4-15, 4-16

## H

- Help ..... 1-6
  - Using help ..... 1-6
- HMI screen. See Operator Screen

## I

- I/O configuration ..... 3-6
  - Adding I/O driver ..... 3-8
  - Filldown feature..... 3-15
  - Mapping I/O points ..... 3-13
  - Saving ..... 3-15
  - Scanning and monitoring .. 3-15
- Index..... 3-5

## K

- Keyboard commands..... 1-5
- Keywords ..... 1-4

## L

- Local systems ..... 1-12

## M

- Mapping I/O points ..... 3-13
- Menu selections ..... 1-4
- Menus..... 4-9
- Motion block ..... 5-22
- Move block ..... 4-15, 5-24
- Multiple selection ..... 4-14

## N

- Naming a project ..... 2-5
- Note
  - Editing..... 5-9
  - Entering ..... 5-9

## O

- Opening a project ..... 2-6
- Operator screen
  - Creating ..... 2-14, 4-21
- Order of flow chart execution ... 4-7

## P

- Page grid ..... 4-10
- Parameter mapping..... 5-31
- Pass by reference ..... 5-37
- Pass by value ..... 5-37
- PID block ..... 5-26
- PID loop
  - PID block ..... 5-26
- Placing flow chart blocks ..... 4-15
- Pointer tool ..... 4-18
- Process control..... 5-26
- ProjectCenter ..... 2-3

## R

- Remote systems..... 1-12
- Remote target..... 1-12
- Return block ..... 5-31

- 
- Row labels..... 4-10
- Run Program block..... 5-28
- Runtime software ..... 1-11
- Runtime target..... 1-9
- Windows CE ..... 2-9
- S**
- Scroll bars ..... 4-10
- Selecting blocks and connections ..  
4-14
- SQL Express block..... 5-29
- Starting a new project..... 2-3
- Status line..... 4-10
- Structured data types
- Comm data items ..... 5-11
- Subchart ..... 4-4
- Begin block ..... 5-31
- Call block ..... 5-8
- Design summary..... 5-37
- Parameter mapping ..... 5-31
- Return block ..... 5-31
- Runtime characteristics ..... 5-38
- using in a project ..... 5-39
- Subcharts..... 5-3
- System data items
- Descriptions..... A-7
- T**
- Tags ..... 3-3
- Target systems ..... 1-10
- Technical Service..... 1-14
- Technical support..... 1-14
- Toolbar ..... 4-10
- W**
- Wait block ..... 4-16, 5-30
- Watchdog timeout ..... 3-19
- Windows CE ..... 1-10
- As embedded controller .... 1-11

## Think & Do

---