

InteractX

Getting Started Guide

CTC Parker Automation
50 W. TechneCenter Dr.
Milford, OH 45150
A3-06562-100

Phone: (513) 831-2340
Fax: (513) 831-5042
<http://www.ctcusa.com>

Copyright Notice

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

While every precaution has been taken in the preparation of this manual, CTC Parker Automation and the author assume no responsibility for errors or omissions. Neither is any liability assumed for damages resulting from the use of the information contained herein. The information contained in this manual is subject to change without notice.

Trademarks

InteractX and the InteractX logo are trademarks of Parker Hannifin Corporation. Microsoft, Windows, ActiveX, and Visual Basic are registered trademarks of Microsoft Corporation in the U.S. and/or other countries. All other trademarks are the property of their respective owners.

Table of Contents

Preface	1
Online Help System	1
Technical Support	1
World Wide Web	1
Technical Training	2
System Requirements and Installation	3
Features of InteractX	3
Development System Requirements	4
Runtime System Requirements	4
Development System Installation	5
Runtime Only Installation	6
Project Deployment Installation	7
Licensing Overview	7
Types of Licenses and Codes	8
Using the InteractX License Manager	11
How to Transfer your InteractX License	12
Getting Started with the Designer	16
MachineShop Suite	16
InteractX	18
Exploring the Example Application	20
Creating an InteractX Application	27
Getting Started with Runtime	57
Configuring Runtime Settings	57
Backing Up Projects	57
Restoring Projects	58
Transferring Projects	58
InteractX Time Savers	59
Using the Toolbin	59
Using Templates	60
Using Styles	62
Finding Sound Files	62
Referencing VBA Scripts	63

Table of Contents



Preface

The InteractX™ Getting Started Guide is designed to get you up and running with a simple InteractX application in a short period of time. This guide discusses the basic features, system requirements, and time-saving techniques of InteractX.

It is likely you will need more detailed information about InteractX to develop sophisticated, customized applications. CTC Parker Automation offers a variety of Customer Support Services to meet your needs.




Online Help System

We have created an extensive online help system to answer questions that arise as you develop or run your applications. You can access the help system from within the InteractX Designer by choosing **Contents and Index** from the **Help Menu** located on the menu bar.

Technical Support

Our Technical Support department provides knowledgeable help for any aspect of the InteractX Designer or Runtime.

You can contact Technical Support in any of the following ways:

 <p>USA Telephone 8:00 am - 5:00 pm EST <i>after-hours emergency callback service available</i></p>	513-248-1714
 <p>Fax</p>	513-831-5042
 <p>E-mail</p>	support@ctcusa.com

World Wide Web



CTC Parker Automation's website offers a Technical Support page where you'll find answers to Frequently Asked Questions, application notes, product user guides, and software downloads that may be helpful.

You can visit our website at www.ctcusa.com.

Technical Training

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

You can contact the Training Coordinator by telephone or e-mail:

 Telephone	800-233-3329
 E-mail	training@ctcusa.com

You can view a current technical training schedule on the web at www.ctcusa.com.



System Requirements and Installation

This chapter introduces InteractX and lists its requirements for both development and runtime workstations. This chapter also provides instructions on installing InteractX.

Features of InteractX

InteractX is a Human-Machine Interface (HMI) software package. InteractX is also a Windows based software solution that provides you with a reliable and efficient method for monitoring or controlling machines or processes. InteractX consists of two pieces of software: the Designer and Runtime.

In the Designer, you build applications for use in Runtime. Building applications consists of placing objects on operator interface panels to represent the machine or process. You then set up a communication driver to work with the application and the controller, then control the machine in Runtime.

The Designer and Runtime include the following features:

- Panel Tools
- Graphics
- Alarms
- ActiveX
- Visual Basic for Applications (VBA)
- User Security Levels
- OPC Tag Server
- OPC Client Interface
- Bundled Communications Drivers
- Advanced studio-style development environment
- Multi-Language Support
- Integrated Tag Editor

See the online help for more detailed information about the InteractX Designer or Runtime.

Development System Requirements

The following table lists the system requirements for the development workstation.

Component	Minimum Development System Requirements
Processor	Pentium III 800 MHz
Operating System	Microsoft® Windows® 98, ME, 2000, NT, or XP Professional
Memory	256 MB RAM
Disk Space	250 MB free hard drive space
Additional Drives	CD-ROM drive
Display	1024 x 768 video resolution, 16-bit color <i>(recommended: 1280 x 1024 video resolution or higher)</i>
Mouse	Microsoft compatible mouse or other pointer input device
Networking (Project Deployment)	10 MB Ethernet network interface card or serial port
Additional Software	Microsoft Internet Explorer® 5.5 or higher

Runtime System Requirements

The following table lists the system requirements for the runtime workstation.

Component	Minimum Runtime System Requirements <i>(suitable for low-end panel replacement)</i>	Recommended Runtime System Requirements <i>(suitable for larger applications)</i>
Processor	Pentium II 266 MHz	Pentium III 800 MHz
Operating System	Microsoft Windows 2000, NT, Embedded NT, or XP Professional	Microsoft Windows 2000, NT, Embedded NT, or XP Professional

Component	Minimum Runtime System Requirements <i>(suitable for low-end panel replacement)</i>	Recommended Runtime System Requirements <i>(suitable for larger applications)</i>
Memory	128 MB RAM	256 MB RAM
Disk Space	92 MB free Compact-Flash or hard drive space	256 MB free Compact-Flash or hard drive space
Additional Drives	CD-ROM drive	CD-ROM drive
Display	640 x 480 video resolution, 16-bit color	800 x 600 video resolution or higher, 16-bit color
Mouse	Microsoft compatible mouse or other pointer input device	Microsoft compatible mouse or other pointer input device
Networking (Project Deployment)	10 MB Ethernet network interface card or serial port	10 MB Ethernet network interface card or serial port
Additional Software	Microsoft Internet Explorer 5.5 or higher	Microsoft Internet Explorer 5.5 or higher

Development System Installation

Installing the Development System will install both the Designer and Runtime on your development workstation.

In order to deploy your InteractX applications to a target workstation, that target system must have Runtime software installed. Follow the steps in the next section to install Runtime on your target workstation.

Complete the following instructions to install the Development System on your workstation.

1. Insert the InteractX CD into the CD-ROM drive on your development workstation.
If the CD does not automatically run, click the **Start** button on your Windows task bar and select **Run**. Browse to the location of your CD-ROM drive and open the file named **Setup.exe**.

System Requirements and Installation

- Click **OK**.
- The installation wizard launches. Follow the directions provided by the installation wizard.
- Click **Install InteractX Development** when you reach the wizard dialog shown below.



- Follow the directions provided by the installation wizard. The installation program will detect the version of MachineShop Suite you have installed on your system. If your version is out of date you will be given the opportunity to upgrade to the latest version. InteractX will then be installed.

Runtime Only Installation

Runtime software is required in order for your target workstation to operate your InteractX applications. You must install Runtime software onto your target workstation. InteractX Runtime also installs the Deployment Server by default.

Complete the following instructions to install Runtime software onto your target workstation.

Note You do not need to install Runtime on your development workstation. The InteractX Development Installation includes both the Designer and Runtime software.

1. Insert the InteractX CD into the CD-ROM drive on your runtime workstation.
If the CD does not automatically run, click the **Start** button on your Windows task bar and select **Run**. Browse to the location of your CD-ROM drive and open the file named **Setup.exe**.
Click **OK**.
2. The installation wizard launches. Follow the directions provided by the installation wizard.
3. Click **Install InteractX Runtime Only** when you reach the wizard dialog shown on page 6.
4. Follow the directions provided by the installation wizard.
InteractX will be installed.

Project Deployment Installation

Complete the following instructions to install components to enable you to perform backup, restore, and transfer operations to a Runtime workstation. This installation will install only MachineShop Suite and InteractX deployment programs on your development workstation.

1. Insert the InteractX CD into the CD-ROM drive on your development workstation.
If the CD does not automatically run, click the **Start** button on your Windows task bar and select **Run**. Browse to the location of your CD-ROM drive and open the file named **Setup.exe**.
Click **OK**.
2. The installation wizard launches. Follow the directions provided by the installation wizard.
3. Click **Install Project Development Only** when you reach the wizard dialog shown on page 6.
4. Follow the directions provided by the installation wizard.
MachineShop Suite and InteractX deployment programs will be installed.
You can now backup, restore, or transfer your InteractX applications to a runtime workstation that has the Deployment Server installed.

Licensing Overview

InteractX incorporates a licensing scheme that provides you with a variety of licensing options. This licensing scheme protects your purchase from unauthorized installations and provides CTC Parker with a method to ensure you are notified of any product enhancements or releases.

InteractX Runtime comes with the ability to evaluate Runtime for one hour right out of the box. This enables you to use Runtime for evaluation and demonstration purposes. After one hour, Runtime will automatically shut-down.

System Requirements and Installation

You can easily obtain an Evaluation license to experiment with the Designer if you wish. Refer to the section titled *Using the InteractX License Manager* for information on obtaining an Evaluation license.

The types of InteractX licenses available are described in the next section.

To register your InteractX software you will be using the License Manager. This utility is launched the first time you enter InteractX Designer. From the License Manager, you are able to receive your license and also transfer your license if the need arises. The License Manager is shown below.



Types of Licenses and Codes

InteractX has several types of licenses available for the Designer and Runtime. The type of license available to you is based on what you purchased.

Evaluation License

This license is provided to those users that wish to evaluate InteractX for a period of up to 45 days. You must register the software with CTC Parker to

gain access to this license. Once registered you will have a period of 45 days to access the Designer and an hour of Runtime.

Everytime you enter the Designer you will be informed as to the number of days remaining on your Evaluation license. At the end of your evaluation period this license terminates.

You will convert your Evaluation license into a Single User license when you purchase InteractX. To convert an Evaluation license to a Single User license, please contact your InteractX distributor or CTC Parker at **(513) 831-2340**.

Single User License

You can purchase either a Single User license for Development or a Single User license for Runtime only. This license provides unlimited access to InteractX. You must register the software with CTC Parker to gain access to this license.

Purchasing a Single User license for Development enables both the Designer and Runtime allowing you to create and test your applications.

Each target runtime system must have an InteractX Runtime license for that particular installation.

Emergency Runtime License

This license allows you to run InteractX for 7 days. At the end of your 7 day activation period, InteractX will shut down and the license is terminated. You do not have access to the Designer with this license type.

You must register the software with CTC Parker to gain access to this license.

To convert an Emergency license to a Single User license, please contact your InteractX distributor or CTC Parker at **(513) 831-2340**.

Licensing Codes

To activate certain licenses you will need an alpha-numeric code. These codes are defined below.

Product Code - A unique alpha-numeric code placed on your license agreement, which is located within the InteractX product box for a Single User license type. This code is issued by CTC Parker based on the options you purchased.

System Requirements and Installation

Hardware Code - An alpha-numeric code generated by the License Manager when you press the Display Hardware Code button. This code uniquely identifies your workstation. You will need this code when registering your software for all license types.

Enable Code - An alpha-numeric code provided to you by CTC Parker once you provide us with your Hardware code. If registering for a Single User license you must also provide CTC Parker with your Product Code. The Enable code is used to enable this installation of InteractX on a specific workstation.

Confirmation Code - This code is used to void your InteractX license on a particular workstation. This is an alpha-numeric code generated by the License Manager when you select the **Void License** radio button and press the **Transfer** button on the Transfer License tab. Once you receive this code, send it to CTC Parker as confirmation that you have voided the license on a particular workstation.

The table below summarizes the various types of licenses along with the type of license code required for each one.

License Type	Development	Runtime	License Code Needed
Evaluation	45 days unlimited use. Will revert to 1 hour of Runtime only after 45 days.	Provides 1 hour of Runtime only.	Hardware Code
Single User	Normal purchased license.	Normal purchased license.	Product Code and Hardware Code
Emergency Runtime	No access to the Designer.	7 days unlimited use. Will revert to 1 hour of Runtime only after 7 days.	Hardware Code

Using the InteractX License Manager

The InteractX License Manager provides a simple interface to enable one of the license types listed in the previous section.

If InteractX is not licensed as a Single User license type, a splash graphic appears each time you launch the InteractX Designer from within Machine-Shop, and each time you launch InteractX Runtime on a runtime system. For the Evaluation license and Emergency Runtime license types, the days remaining will be displayed on the graphic.

To access the License Manager, simply press the ENTER key or click on the graphic while it is displayed. The License Manager dialog will appear as shown on page 8, which contains features allowing you to obtain a new license type or transfer an existing license.

While in the InteractX Designer on a development system, the License Manager can be accessed anytime by choosing the **Licensing** option from the Designer's Help Menu.

On a Runtime system, the License Manager can be accessed by pressing the InteractX Licensing **Configure** button on the General tab of the Deployment Server dialog.

Once the License Manager dialog is displayed, follow the directions shown in the bottom half for steps on how to obtain an Enable Code to activate the desired license type. Generally, the steps are as follows:

1. If you wish to activate a Single User license for either a Development or a Runtime system you must first locate and enter the Product Code for the product you purchased. Your Product Code is located within the InteractX product box.
For the Evaluation license and Emergency Runtime license types, you do not require a Product Code.
2. Press the **Display Hardware Code** button on the dialog and accept the Licensing Agreement.
The Hardware Code for your computer will be displayed.
3. For a Single User license, provide us with your Product Code and Hardware Code.
For an Evaluation license or Emergency Runtime license, you only need to provide us with the Hardware Code.
You can do this by calling CTC Parker, emailing the codes to us, or by accessing our Web site and entering the codes into the appropriate fields, as directed by the instructions provided on the site.
4. Upon verification of the appropriate code(s), you will immediately be sent an Enable Code via email.

System Requirements and Installation

5. Enter the Enable Code into the Enable Code control on the Licensing Manager dialog.
6. Press **Validate** to activate your license.

All of the information needed to guide you through these steps is displayed in the bottom half of the License Manager dialog.

How to Transfer your InteractX License

Once you have installed InteractX you might have reason to transfer your license from one workstation to another. InteractX provides a very easy interface to enable you to do that. The process for transferring your Development or Runtime license is identical.

The Transfer License tab located on the License Manager provides several options for you when transferring licenses.

Simply launch the License Manager as described in the previous section, then select the Transfer License tab to display your transfer options as shown below.



You are able to:

- Transfer a license to a networked computer
- Transfer a license to removable media
- Transfer a license from removable media
- Void a license

Transferring to a Networked Computer

Select this option if you will be transferring your license to a workstation connected to your network. You will be transferring to a shared folder on the target workstation. This is useful if you have a single InteractX license and wish to transfer your development system to a laptop computer from your desktop workstation.

Once your license is transferred you will not be able to run InteractX from the original workstation.

To transfer your license to a networked computer, perform the following steps:

Note Performing this procedure will cause InteractX to shut down on the workstation you are transferring the license from. Ensure you save all of your InteractX applications prior to performing this procedure.

1. First you must prepare the target workstation to accept your license by installing InteractX onto the target system.
2. You must then share your InteractX folder so that it can be seen by the networked computer.
In a typical installation of InteractX, the folder is located at:
C:\Program Files\InteractX
Refer to your Windows documentation for information on how to share a folder.
3. Launch the License Manager.
4. Select the Transfer License tab, then select the **Direct License Transfer to Directory** option.
5. Browse to the location on the network where the shared folder resides.
6. Select the folder.
7. Click **OK**.
8. Click **Transfer** from the Transfer License tab.
9. Click **Finish**.

License information will be transferred to the target workstation. Once you exit the License Manager, InteractX will also close. Your target workstation now has a licensed version of InteractX. The license will be of the same type as what was on the original workstation.

Transferring To and From Removable Media

Select these options to transfer your license to removable media and then from the removable media to a target workstation.

You can transfer your license using Compact Flash, ZIP disk, or a floppy disk. This process involves performing actions on both the target workstation and the workstation that contains the licensed version of InteractX.

You will begin at the target workstation by preparing the media you will be transferring the license to. You will then go to the original workstation that contains the licensed version of InteractX and move the license from this workstation onto the media. You will then go back to the target workstation, insert the media, and complete the transfer process.

Once your license is transferred you will not be able to run InteractX on the original workstation.

To transfer your license to and from removable media, perform the following steps:

Note Performing this procedure will cause InteractX to shut down on the workstation you are transferring the license from. Ensure you save all of your InteractX applications prior to performing this procedure.

1. First you must prepare the target workstation to accept your license by installing InteractX onto the target system.
2. On the target workstation that you wish to install the license, launch InteractX.
The License Manager utility displays since you do not have a license for this installation of InteractX.
3. Select the Transfer License tab, then select the **Media Transfer In From Another Computer** option.
4. Click **Browse** to select the media you're using to perform the transfer with.
5. On the original workstation that you wish to transfer the license from, launch the License Manager.
6. On the Transfer License tab, select the **Media Transfer Out To Another Computer** option.
7. Click **Browse** to select the media you're using to perform the transfer with.
8. Click **Transfer** from the Transfer License tab.
License information will be transferred to the media device you have chosen: either Compact Flash, ZIP disk, or a floppy disk. You will imme-

diately be locked out of InteractX that resides on the original workstation.

9. On your target workstation, insert the media and return to the point in the wizard where you left off in step 4.
10. Click **Next**.
11. Click **Finish**.

License information is transferred from the media to your target workstation. You now have a licensed version of InteractX residing on this machine. The license will be of the same type as what was on the original workstation.

Permanently Voiding a License

There might be instances where you will want to permanently disable your InteractX license. This process is referred to as voiding your InteractX license. By permanently voiding your license you will be unable to use your license for any future installation of InteractX.

Contact CTC Parker at **(513) 831-2340** to void your InteractX license.



Getting Started with the Designer

In this chapter, you'll be introduced to MachineShop Suite and InteractX. Then, you'll explore an InteractX application example. Lastly, you'll have a chance to create your own InteractX application following step-by-step instructions.

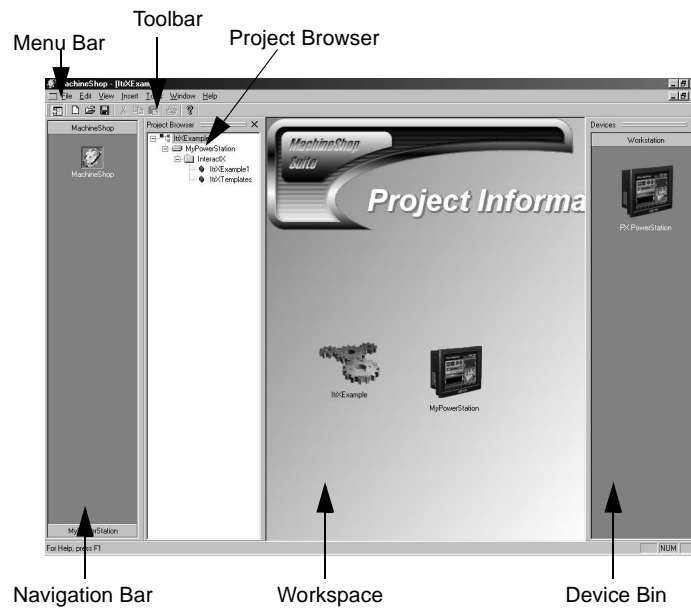
MachineShop Suite

MachineShop Suite installs automatically with InteractX. You use MachineShop Suite to create projects that will manage your InteractX applications. Using MachineShop, you can perform the following tasks:

- Create a new InteractX application
- Open or back up existing InteractX applications
- Transfer InteractX applications to and from runtime workstations
- Backup and Restore projects

You can also use other software besides InteractX within MachineShop. See the MachineShop online help for more information about its capabilities.

MachineShop Suite is shown below. Each element is described in the table that follows.



The table below briefly describes each element identified on the MachineShop screen. For more detailed information about each of the MachineShop elements, consult the MachineShop online help.

Element	Description
Device Bin	The Device Bin contains images of the available hardware devices you may add to your project. The devices are organized by product type into drawers. Only the Workstation drawer contains devices you will use with InteractX. These devices represent runtime workstations.
Menu Bar	The Menu Bar displays conventional Windows menus that are used to perform functions in MachineShop.
Navigation Bar	The Navigation Bar lets you move easily among different devices and applications within a single project. A drawer is designated for each device. When selected, a device's drawer will display all of its applications.
Project Browser	The Project Browser displays your entire project in a tree structure, with branches you can expand and collapse. You can manage your project hierarchy and insert devices, links, and applications by right-clicking within the Project Browser.
Toolbar	The Toolbar houses MachineShop commands you use most frequently, such as open, save, and print.
Workspace	The Workspace displays a graphical representation of your project and of each device within your project. When an application is launched, the application is loaded and displayed within the Workspace.

Creating a New Project

To develop an InteractX application, you must first create a MachineShop project. A project combines all application and device files so that they can be manipulated as a group.

When you launch MachineShop, you are prompted to either create a new project or open an existing project. If you select **Create a New Project**, MachineShop uses a simple dialog box to guide you through the process.

Later in this chapter, you will have the chance to build your own InteractX application. At that time, you will first create a new MachineShop project following step-by-step instructions.

Navigating a Project

MachineShop provides two navigation tools that let you move easily among devices and InteractX applications: the Navigation Bar and the Project Browser.

The Navigation Bar contains a drawer for each device you have added to your project. Clicking a device drawer will display all of that device's associated applications. Double-clicking an application from within a device drawer will launch that application.

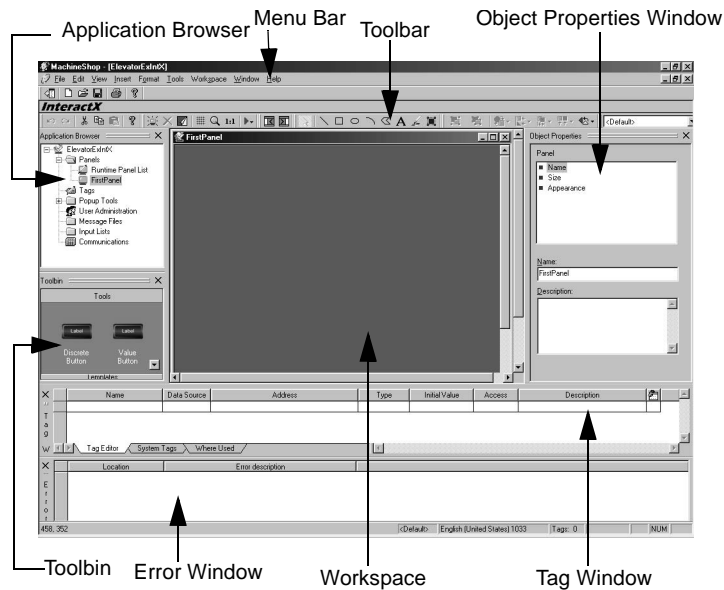
The Project Browser displays the project in a tree structure, similar to Windows Explorer. Expanding a device's branch will display all of its associated applications. Double-clicking an application from within the Project Browser will launch that application.

You can open and edit multiple applications within the MachineShop workspace. Navigation is seamless between applications. Click once within the application if you want to change the focus to that application.

InteractX

You will create InteractX applications within your MachineShop project. An application is designed in the InteractX Designer. The Designer builds customized interfaces that an operator will use in Runtime to control and monitor machines and machine processes.

InteractX Designer is shown below. Each element is described in the table that follows.



The table below briefly describes each element identified on the InteractX screen. For more detailed information about each of the InteractX elements, consult the InteractX online help.

Element	Description
Application Browser	The Application Browser allows you to quickly and easily access various items within your application.
Error Window	The Error Window displays errors that occur in the Tag Window and the Alarm Definitions tab. To display the Error Window, choose View/ Error Window from the Menu Bar.
Menu Bar	The Menu Bar provides conventional Windows menus that are used to perform functions in InteractX.

Getting Started with the Designer

Element	Description
Object Properties Window	The Object Properties window lists all of the property pages for a selected object. You can configure the properties on each property page to customize various components of the object, such as its appearance and its function during Runtime.
Tag Window	The Tag Window displays data associated with each tag, such as the tag's address, data source, and data type. A tag is a variable that is used in place of a controller-specific address or relational expression. Tags reference values from a control device or an expression of multiple values. You can define your own tags and/or use predefined system tags.
Toolbar	The Toolbar consists of several smaller, more specialized, rebars that are commonly referred to as toolbars. These toolbars house frequently-used InteractX commands.
Toolbin	The Toolbin contains frequently used objects that you can drag and drop onto a panel. You can create your own objects and store them in the Toolbin so that they will be available in all of your applications. You can also add your own unique drawer to the Toolbin.
Workspace	The Workspace is the area in the center of the application frame where you build your panels and add objects to your application. The Workspace is where you configure alarms, configure communications, create Message Files, and create Input Lists as well.

Exploring the Example Application

An example InteractX application is included on your installation CD. This application was created to introduce you to the main components of InteractX, and to give you the chance to explore the setup of an existing application before you create your own.

To open the example application, complete the following steps:

1. Launch **MachineShop Suite** from your desktop icon.
MachineShop Suite displays with the Welcome dialog box.
2. Select **Open an Existing MachineShop Project**.
The Open Project dialog box displays.
3. Select the file **ltrXExample.gms**, then click **OK**.
4. After the example MachineShop project has opened, double-click **ltrXExample1** from the Project Browser.
The example InteractX application opens within the InteractX Designer.

Exploring the Example Application in the Designer

You can explore the application in the Designer to investigate the setup and organization of the panels. You may also note the properties of each object, listed in the Object Properties Window, and the tags associated with the application, listed in the Tag Window.

Exploring the Example Application in Runtime

The example application has five main parts: Panel Tools, Graphics, Alarming, ActiveX®, and Visual Basic. You can explore each of these components and some of the capabilities of InteractX by viewing the example application in Runtime. The example application uses the Simulator driver to simulate driver communication connectivity.

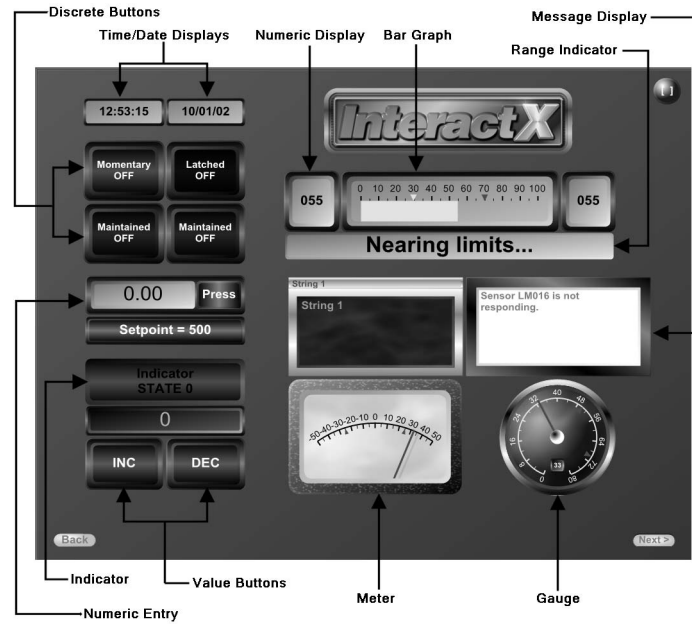
To run the example application, perform the following steps:

1. **Select Tools ►Run Application ►Connected.**
A splash screen displays.
2. Click the **Preview** button.
The Examples panel displays with selections on the left side of the panel.
3. Select the functionality you wish to view in Runtime.
4. To exit Runtime, click the **End** button at the bottom of the Examples panel.

In this section, each of the five main parts of the example application will be discussed. Unless specifically noted, the following information applies to Runtime.

Part 1 - Panel Tools

Select **Panel Tools** from the Examples panel. The panel shown below will display.



Refer to the InteractX online help for detailed information about the functions and properties of the tools identified in the figure above.

There are three panels that illustrate Panel Tools. To view the additional panels, click the **Next** button at the bottom of the panel.

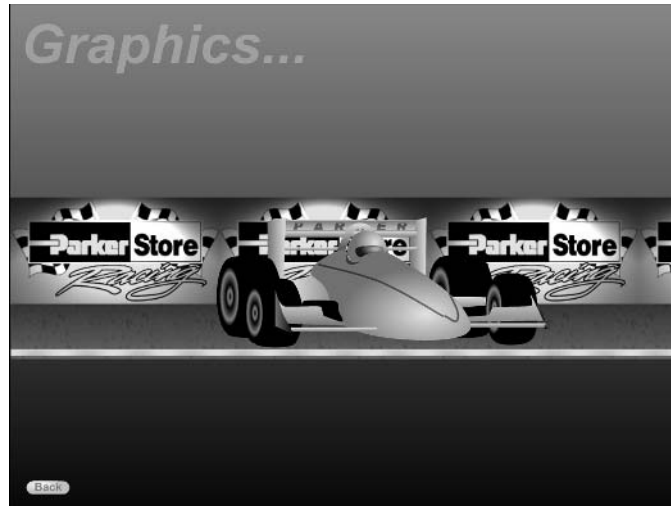
You might find it helpful to explore the different properties of the Panel Tools in the Designer.

To exit Runtime, perform the following steps:

1. Click the **Back** button at the bottom of the panel.
The Examples panel will display.
2. Click the **End** button at the bottom of the panel.

Part 2 - Graphics

When you select **Graphics** from the **Examples** panel, the panel shown below will display.



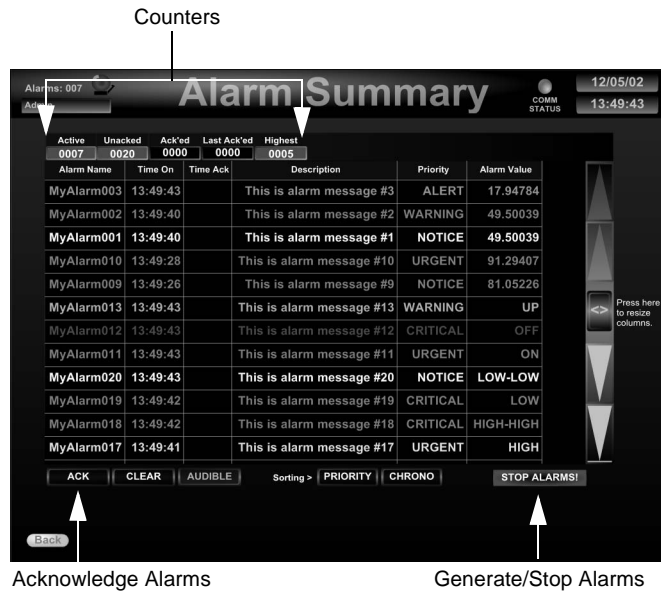
This panel demonstrates the ability of InteractX to apply animation connections such as appearance, motion, and position to graphics. This flexibility provides the maximum freedom in customization of your panels. InteractX can also combine graphics into groups then animate those groups as if they were individual objects.

Observe this panel as the demonstration loops through showing the various graphics available within InteractX.

Explore this panel in the Designer to see how animation connections are applied to the various graphic elements. Also observe how various objects are grouped within the panel.

Part 3 - Alarming

When you select **Alarming** from the **Examples** panel, the panel shown below will display.



Alarms are triggered when a condition that you establish is met. For example, you may want to set an alarm to trigger each time the temperature of a liquid rises above a certain level. The temperature level is the condition that would trigger the alarm. To see what alarms look like when they are triggered, click the **Generate/Stop Alarms** button in the example application.

The alarms appear in the Alarm Viewer. You can configure the Alarm Viewer to display Alarm information that is meaningful to the operator. For more information about configuring the Alarm Viewer, consult the InteractX online help.

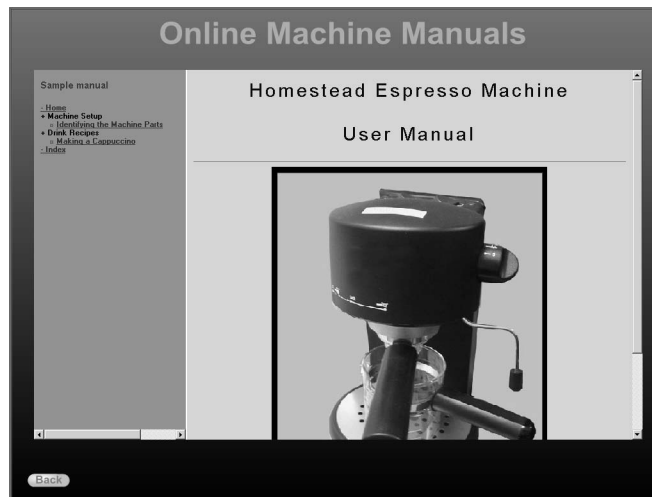
You can set up your Alarms so that the operator can acknowledge them when they trigger. The Acknowledge button is labeled in the image above. Acknowledging an alarm simply means the alarm is recognized by the operator. After an alarm has been acknowledged, it can be cleared.

Alarms can be active or inactive. In the example application, a counter is included to indicate how many alarms are currently active. An active alarm is an alarm that was triggered and has a condition that is still being met. An inactive alarm is an alarm that was triggered, and the alarm condition is no longer being met.

This is a brief overview of alarms. A complete discussion of alarms is included in the InteractX online help. In the section *Creating an InteractX Application* on page 27, you will get the chance to create and configure your own alarms with step-by-step instructions.

Part 4 - ActiveX

When you select **ActiveX** from the **Examples** panel, the panel shown below will display.



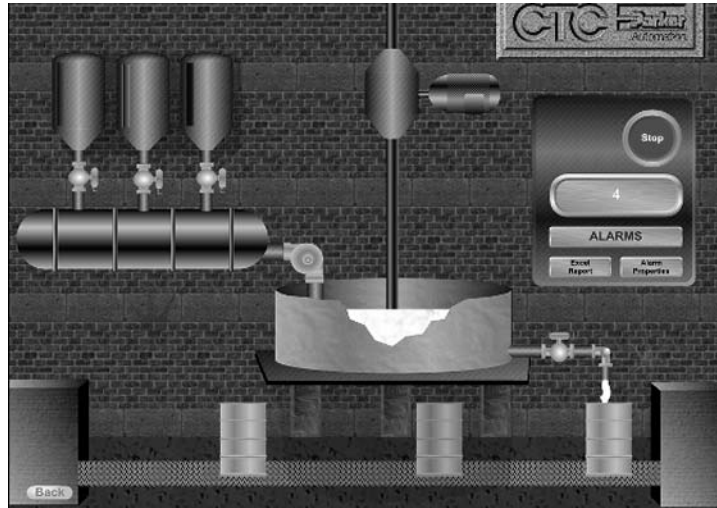
Originally called OLE (OCX) controls, ActiveX controls are small packages of functionality that use COM technologies (Component Object Model) to operate with other types of COM components and services to perform specific tasks. These tasks can range from a calendar that displays the current date to a full featured spreadsheet or animation sequencer.

The ActiveX tool will detect all ActiveX controls present on your computer system and list them in a window where you can select the appropriate control. However, not all ActiveX controls on your system can be used with InteractX.

This panel is using the InteractX Browser Display tool. This tool allows you to display HTML pages within an InteractX panel. A possible use of the Browser Display tool is illustrated in this ActiveX part of the example application.

Part 5 - Visual Basic®

When you select Visual Basic from the Examples panel, the panel shown below will display.



InteractX ships with Microsoft Visual Basic for Applications. Visual Basic for Applications (VBA) is an object-oriented programming language that allows users to perform high-level tasks within InteractX such as the following:

- Control the execution of code with conditional logic.
- Reference third-party objects.
- Create User Forms, Modules and Class Modules and call them from VBA subroutines that you have written for your VBA project.
- Display dialogs that contain native VB and ActiveX controls.

In the example application, VBA is used to simulate a paint factory. You can select either a manual or auto selection mode. If you select manual you can then select between standard and custom color selections. The color selected will fill the mixing vat which in turn will fill the number of barrels you selected. This panel is controlled through VBA subroutines which are modifying the object properties only through VBA.

An extensive VBA online help file is included with InteractX. In the online help, you can find descriptions and examples of objects, properties, events, and methods that are part of the InteractX object model. To access the VBA online help, choose **Help ► Contents and Index** from the Designer menu

bar. This launches the InteractX Help system. From the Table of Contents pane select the **Visual Basic for Applications** book.

This completes an introduction to the example application. You are encouraged to explore the different properties of each panel in the Designer.

To exit Runtime, perform the following steps:

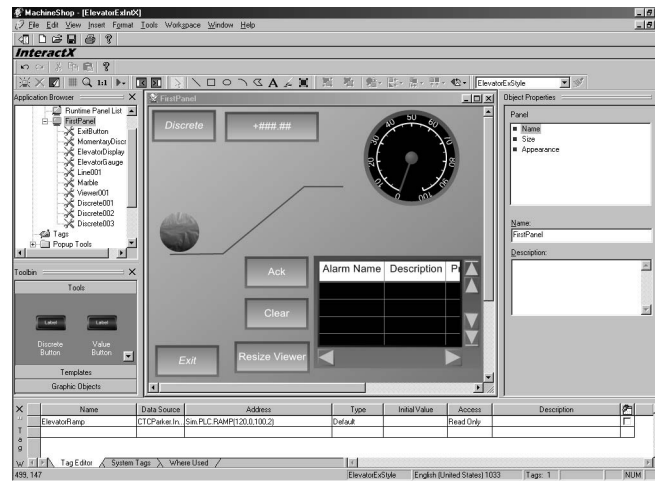
1. Click the **Back** button at the bottom of the panel.
The Examples panel will display.
2. Click the **End** button at the bottom of the panel.

Creating an InteractX Application

In this section, you will have the chance to create your own sample InteractX application following step-by-step instructions. You will perform the following:

- Create a MachineShop project
- Add an InteractX application to your project
- Add a panel
- Add several tools and graphics to your panel
- Animate the graphics to illustrate a process
- Connect the movement of the graphics to gauges and alarms
- Incorporate simple VBA scripts

When you are finished, your development screen will look like the image shown below.



Getting Started with the Designer

The screen displays a marble climbing an elevator. The marble's progress is determined by a single tag, *ElevatorRamp*, and is monitored by a Numeric Display and a Gauge. Alarms are set up to trigger when the marble reaches defined limits. Incorporating VBA will allow you to make the marble disappear and reappear during Runtime.

Task 1 - Create a New MachineShop Project

The following instructions will help you create and configure a new MachineShop project.

1. Select the **MachineShop Suite** icon on your desktop to launch MachineShop Suite.
2. From the Welcome Screen, select **Create a New MachineShop Project**.
3. Type *ElevatorExample* in the **Project Name** box.
4. Accept the default for Project Location.
5. Type *An example application from the Getting Started Guide* in the **Project Description** box.
6. Click **OK**.
This completes the creation of a MachineShop project.

Task 2 - Add a Device to the MachineShop Project

After you have created a new project, the next task would be to add a device, which represents the runtime workstation. To practice this setup, the following instructions will lead you through the process of adding a device and setting up its connections.

However, you will not actually deploy the *ElevatorExample* MachineShop project you have created to a runtime device. The deployment process is covered in *Getting Started with Runtime* on page 57.

To add a device and configure its connections, complete the following steps:

1. From the Workstation drawer of the Device Bin, click and drag the **PX PowerStation** icon onto the workspace.
An Add Device wizard will display.
2. Type *PowerStation* in the **Device Name** box.
3. Check the box next to **InteractX (HMI)** to include it as part of your project.
4. Click **Next**.
5. Click the **Add...** button to add a connection.
A connection is used to transfer your project to the target workstation.
6. Type *ConnectionExample* in the **Connection Name** box.

The Connection Name box is included to allow you to identify your connections with meaningful names.

7. Select **Disk** as the Connection Type.
There are three types of connections from which you can choose when adding a device: Serial, Network, and Disk. For this tutorial, select **Disk** even though it may be different from the connection type you usually use.
8. Click **Configure**.
9. Type *C:\IntXTutorial* as the destination path.
Most users who select Disk as the Connection Type use a CompactFlash card to deploy their projects to a PowerStation. In this case, the destination path would be the drive assigned to the CompactFlash card. However, because your projects will not be deployed in this tutorial, your hard drive is used as the sample destination path.
10. Click **Finish**.
11. Click **Finish** again.
This completes the configuration of the device. Notice that InteractX is added to the Project Browser.

Task 3 - Insert a New InteractX Application

In order to create an InteractX application, you must add a new InteractX application to the MachineShop project. To do this, complete the following steps:

1. From the Project Browser, right-click **InteractX**.
2. Click **Insert New**.
The New Application dialog appears.
3. Type *ElevatorExIntX* in the **Name for the new application** box.
4. Click **OK**.
This completes the creation of a new InteractX application and InteractX will launch.

Task 4 - Configure a Panel

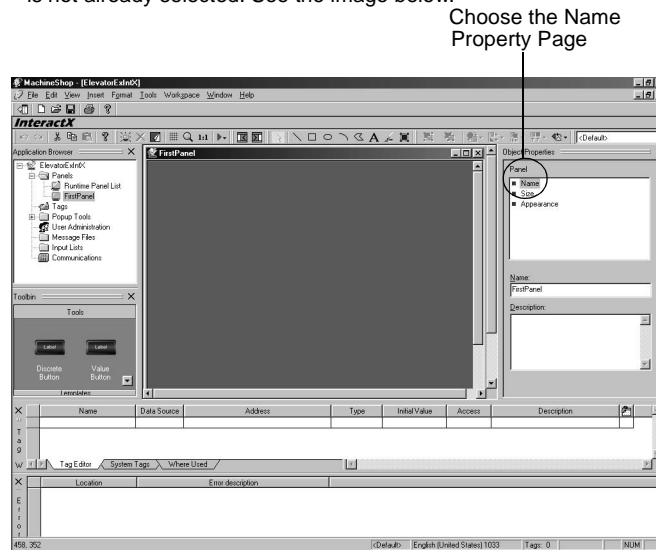
This tutorial will use only one panel. However, most of your InteractX applications will use several panels. You can create a new panel at any time by right-clicking **Panels** in the Application Browser and selecting **Add Panel**.


In this section, you will configure the one panel that is automatically created upon launching InteractX by giving the panel a name, checking its resolution, and changing its appearance.

To configure the panel, complete the following steps:

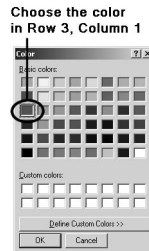
Getting Started with the Designer

1. From the Application Browser, double-click **PanelI001** to select it as the panel in focus.
2. Position the panel so that it covers all of the Workspace.
3. From the Object Properties window, select the **Name** property page if it is not already selected. See the image below.



4. Type *FirstPanel* in the Name field.
5. Select the **Size** property page. Be sure that **800 x 600 Resolution** is selected and that **Scale to Runtime Resolution** is checked.
6. Select the **Appearance** property page to set the panel background color.
7. In the **From** color box, choose **Silver**.
8. In the **To** color box, click the  (browse button) next to the **To** color box to display the Color Palette.

9. Choose the color in Row 3, Column 1, as shown below.



10. Click **OK**.
11. In the Gradient box, choose the fourth gradient in the list.
12. Accept the defaults for all other options on the Appearance property page.
This completes the configuration of the panel.

Task 5 - Setting Up Communications

Before you begin adding tool objects to your panel, you should configure the communications driver you will use with your application.

This application uses the Simulator driver since we will not be communicating with an actual programmable controller. The Simulator driver provides you with the same behavior you would have with a standard communications driver.

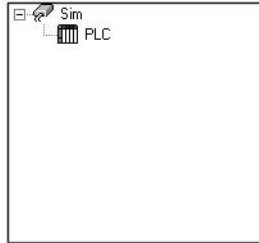
To setup your communications driver, complete the following steps:

1. Double-click the **Communications** icon in the Application Browser.
The Communications window displays.
2. Click the **connector** icon within the Communications window.
The New Channel wizard launches.
3. Enter *Sim* within the **Channel name** box.
4. Click **Next**.
5. Select **Simulator** from the **Device driver** drop-down list box.
6. Click **Next**.
7. Click **Next** again to accept the default **Write Optimizations** settings.
8. Click **Finish**.
A channel has been added to the Communications window with the name of *Sim*.
9. Click the **PLC** icon within the Communications window.
The **New Device** wizard launches.
10. Enter *PLC* within the **Device name** box.

Getting Started with the Designer

11. Click **Next**.
12. Click **Next** again to accept the default **Device model** of **16-Bit Device**.
13. Click **Next** again to accept the default **Device ID** information.
14. Click **Finish**.

The Communications window appears as shown below.



15. Close the Communications window.
You have now added the Simulator driver to this application.

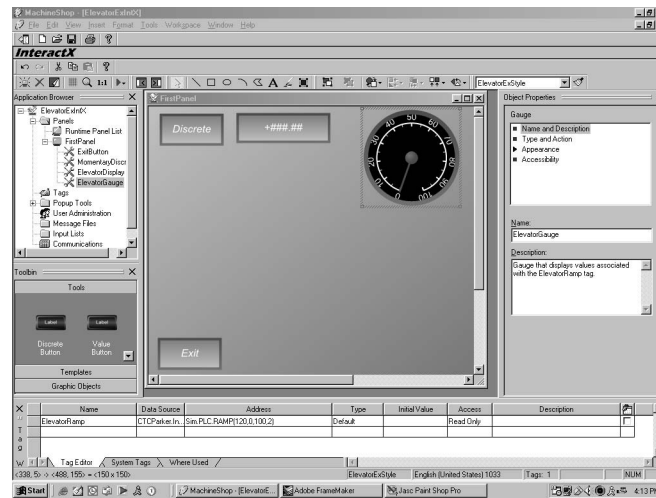
Task 6 - Add Tool Objects

Now that you have configured a panel and your communications driver, you can begin to add objects to the panel. In this section, you will add tools, including an Action Button, a Discrete Button, a Numeric Display, and a Gauge.

Then, you will configure the different property pages of the tools you create, and configure them to communicate with the Simulator in order to demonstrate monitoring a specified value.

Creating an InteractX Application

When you are finished, your development screen will look like the one pictured below.



When applying colors to the properties throughout this tutorial, you may choose whatever colors you desire.

Subtask 6.1 - Add and Configure an Action Button

Action Buttons are buttons that have predefined 'actions' from which you can choose. You will add an Action Button that will allow you to exit the application during Runtime.

After you are finished configuring the properties of the Action Button, you will save the button and the appearance style you applied to the button for future use.

To add an Action Button to your application, complete the following steps:

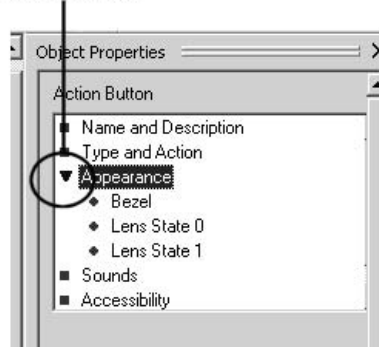
1. From the **Tools** drawer of the **Toolbin**, click and drag the button labeled **Action** onto the bottom left corner of the panel.
2. From the Object Properties window, select the Action Button's **Name and Description** property page.

Note Because you want to configure the property pages of the Action Button, be sure the Action Button is selected within the Workspace. An object is selected when it displays onscreen surrounded by a gray box with handles.

Getting Started with the Designer

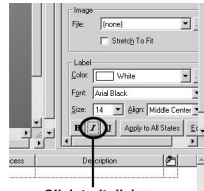
3. In the **Name** field of the **Name and Description** property page, type *ExitButton*.
4. In the **Description** field, type *Exit button for reuse*.
5. Select the Action Button's **Type and Action** property page.
6. Choose **Exit** from the Action drop-down menu.
7. Select the Action Button's **Appearance** property page.
8. Set the **Roundness** to 0% by dragging the Roundness slider.
9. Set the **Button Depth Up and Down** to 0% by dragging its slider. For finer control of the slider, press the arrow keys on your keyboard while the slider is selected.
10. Expand all of the **Appearance** related property pages by clicking the arrow next to Appearance. See the image below.

Expand Appearance-related Property Pages

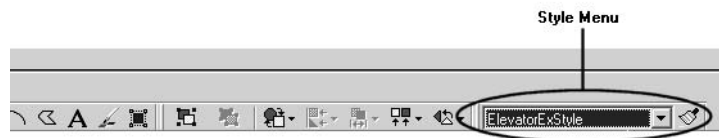


11. Select the Action button's **Bezel** property page and make the following changes:
 - a. In the **From** color box, choose **Green**.
 - b. In the **To** color box, choose **Olive**.
 - c. In the **Gradient** box, choose the third gradient option from the top in the drop-down list.
 - d. Set the **Depth** to approximately 0% and the **Width** to approximately 20%.
12. Select the Action Button's **Lens State 0** property page and make the following changes:
 - a. In the **From** color box, choose **Gray**.
 - b. In the **To** color box, choose **Silver**.
 - c. In the **Gradient** box, choose the third gradient option from the top in the drop-down list.
 - d. In the Label section, click **Edit** to change the label of the Action Button.
 - e. Type the word *Exit*.

- f. Click the **I** button, shown below, to italicize the text.



- g. Click **Apply to All States** so that the same text and text style are applied to the Action Button in both its on and off states.
13. Select the Action Button's **Lens State 1** property page and make the following changes:
 - a. In the **From** color box, choose **Silver**.
 - b. In the **To** color box, choose **Black**.
 - c. In the **Gradient** box, choose the third gradient option from the top in the drop-down list.
 14. Click the **Save** icon on the toolbar to save all changes. This completes the creation and configuration of the Exit Action Button. However, because every application you create requires an Exit Action Button, you will save the Exit Action Button you just created in the Toolbin for future use.
 15. Click and drag the Exit Action Button into the Tools drawer of the Toolbin for use in all InteractX applications you create. In addition, you will save the appearance settings ('style') you applied to the Exit Action Button for use with all other tool objects you create. The easiest way to do this is to create your own style.
 16. Select the Exit Action Button you just created so that it appears in the Workspace surrounded by handles.
 17. From the Menu Bar, choose **Format ► Styles**.
 18. Click **New**.
 19. In the **Style Name** box, type *ElevatorExStyle*.
 20. Click **Merge**, and then click **Save**.
 21. Click **Close**.
 22. Be sure *ElevatorExStyle* appears in the Style Menu on the Toolbar, as shown below.



23. Click the **Save** icon on the Toolbar to save all changes.

Subtask 6.2 - Add and Configure a Discrete Button

Discrete Buttons are buttons that are binary in nature. They have two positions, on and off. Its value is either one or zero. You might use a Discrete Button to stop or start a process, or to control a bit in a controller.

In this section, you will create a Discrete Button and apply to it the *ElevatorExStyle* you created in the last section.

Although you will create the Discrete Button now, you will not use it until you reach the last section of this tutorial entitled *Task 9 - Add VBA Functionality* on page 54.

To add and configure a Discrete Button, complete the following steps:

1. From the **Tools** drawer of the **Toolbin**, click and drag the button labeled **Discrete** onto the panel.
2. Be sure *ElevatorExStyle* is the active style in the Style Menu, and that the Discrete Button is selected in your Workspace.
3. Click the paintbrush next to the Style Menu to apply *ElevatorExStyle* to the Discrete Button, as shown below.

Click Paintbrush to apply active style in Style Menu



4. Position the Discrete Button in the upper-left corner of your Workspace.
5. With the Discrete Button still selected, choose its **Name and Description** property page from the Object Properties window.
6. On the Discrete Button's **Name and Description** property page, make the following changes:
 - a. In the **Name** field, type *MomentaryDiscrete*.
 - b. In the **Description** field, type *Discrete Button for use in VBA section of tutorial*.
7. Expand all of the **Appearance** related property pages by clicking the arrow next to Appearance. Because you saved and applied the same style you used for the Exit Action Button, you will only need to change the text of the Discrete Button.
8. Select the **Lens State 0** property page and make the following changes:
 - a. In the Label section, click **Edit**.
 - b. Type *Discrete*.

- c. Click the **I** button to italicize the text.
 - d. Click **Apply to All States** to display the same text and text style in both the on and off positions.
9. Click the **Save** icon on the Toolbar.
This completes the creation and configuration of the Discrete Button.

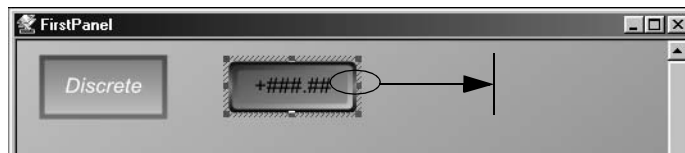
Subtask 6.3 - Add and Configure a Numeric Display

A Numeric Display is used to monitor values you specify through tags. Numeric Displays can be set up to recognize three different limits: Minimum to Limit 1, Limit 1 to Limit 2, and Limit 2 to Maximum. When Numeric Displays are set up in this way, they can be configured to change colors when each new limit is reached. This helps the operator to quickly recognize a change in a monitored condition.

In this section, you will create a Numeric Display and configure its property pages. You will also create a new tag that will be configured later to a specific function, allowing you to monitor changing values with your Numeric Display. You will configure the tag in the section *Subtask 6.5 - Configure a Tag* on page 39.

To add and configure a Numeric Display, complete the following steps:

1. From the **Tools** drawer of the **Toolbin**, click and drag the tool object labeled Numeric Display onto the panel.
2. Drag the right, middle handle to the right to resize the Numeric Display so that it is twice its original width. See the image below.



3. Position the Numeric Display next to your Discrete Button.
4. Be sure *ElevatorExStyle* is the active style in the Style Menu.
5. With the Numeric Display selected, click the paintbrush next to the Style Menu to apply *ElevatorExStyle* to the Numeric Display.
6. Choose the Numeric Display's **Name and Description** property page from the Object Properties window and make the following changes:
 - a. Change the Numeric Display's **Name** to *ElevatorDisplay*.
 - b. Change the Numeric Display's **Description** to *Numeric Display that displays values associated with the ElevatorRamp tag*.
7. On the Numeric Display's **Type and Action** property page, within the '**Tool Value**' **Output** box, type *ElevatorRamp*.

Getting Started with the Designer

As soon as you click out of the text box, notice that *ElevatorRamp* has been added to the Tag Window automatically.

8. Select the Numeric Display's **Appearance** property page and change **Display Depth** to approximately 0%.
9. Expand the Numeric display's **Appearance** related property pages.
10. Select the **Minimum to Limit 1** property page, make the following changes:
 - a. In the **From** color box, select the color **Gray**.
 - b. In the **To** color box, select the color **Silver**.
 - c. In the **Gradient** box, choose the third gradient option from the top in the drop-down list.
 - d. In the Label section, choose **White** as the Label Color.
11. Select the **Limit 1 to Limit 2** and **Limit 2 to Maximum** property pages, leave all settings at their defaults.

This will ensure that when the value you monitor with the *ElevatorRamp* tag is within Limit 1, the Numeric Display will look the same as the other objects on your screen. When the value exceeds Limit 1, the Numeric display will be yellow. When the value exceeds Limit 2, the Numeric display will be red.

12. Click the **Save** icon on the Toolbar.
This completes the creation and configuration of the Numeric Display.

Subtask 6.4 - Add and Configure a Gauge

A Gauge displays monitored values on a graduated scale. Chevron marks define Limit 1 and Limit 2. A Gauge provides the operator a visual representation of relational data.

In this section, you will create and configure a Gauge to monitor the same values as the Numeric Display. These values are defined by a function in the tag *ElevatorRamp*, which you will define in the next section, *Subtask 6.5 - Configure a Tag* on page 39.

To add and configure a Gauge, complete the following steps:

1. From the **Tools** drawer of the **Toolbin**, click and drag the object labeled **Gauge** onto the panel.
2. Position the Gauge in the upper-right corner of your Workspace.
3. With the Gauge selected, apply the *ElevatorExStyle* from the Style Menu.
4. Select the Gauge's **Name and Description** property page and make the following changes:
 - a. Change the **Name** of the Gauge to *ElevatorGauge*.
 - b. Change the **Description** of the Gauge to *Gauge that displays values associated with the ElevatorRamp tag*.

5. On the Gauge's **Type and Action** property page choose the *ElevatorRamp* tag as the '**Tool Value**' **Output** by selecting the tag from the drop down list.
6. Select the Gauge's **Appearance** property page and set the Display Depth to approximately 0%.
7. Expand the **Appearance** related property pages.
8. Select the Gauge's **Bezel** property page and set the **Width** to approximately 10%.
9. Leave the properties on all other property pages at their defaults. This ensures that, like the Numeric Display, the yellow chevron will mark the point where the monitored values have exceeded Limit 1 and the red chevron will mark the point where the values have exceeded Limit 2.
10. Click the **Save** icon on the Toolbar.
This completes the creation and configuration of the Gauge.

Subtask 6.5 - Configure a Tag

In this section, you will configure the tag you created earlier entitled *ElevatorRamp* using the Tag Editor.

You will configure the tag to address the Simulator driver and run a RAMP function. The Simulator driver is automatically installed with InteractX and requires no additional hardware. Using the RAMP function will allow you to monitor constantly changing values in order to gain a better understanding of the capabilities of InteractX.

When the RAMP function is addressed using the *ElevatorRamp* tag, the Numeric Display and the Gauge will both monitor values from 0 - 100. When the value of 30 is reached, the Numeric Display will indicate that Limit 1 has been exceeded by displaying as yellow. The needle on the Gauge will point past the yellow chevron mark, also indicating that Limit 1 has been exceeded. The default value of 30 to mark the end of Limit 1 was accepted on the Type and Action property pages of both the Numeric Display and the Gauge.

Likewise, when the value of 70 is reached, the color of the Numeric Display will change to red, indicating Limit 2 has been exceeded. The needle on the Gauge will point past the red chevron mark.

When the RAMP function completes one cycle, it will repeat, starting again at 0 and continuing to 100. The cycle will be repeated until you exit Runtime.

You can find all of the other Simulator functions in the Simulator help file, as well as instructions on addressing a specific bit within the Simulator driver. To view the Simulator help file, choose **Help ► Contents and Index** from the Menu Bar. This will launch the InteractX Online Help system. From the left

Getting Started with the Designer

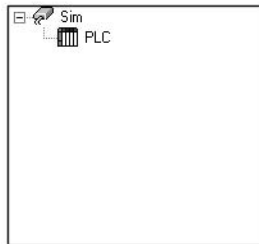
Contents pane, select the **Communications** book then select the Simulator driver help.

To configure the *ElevatorRamp* tag to address a RAMP function within the Simulator driver, complete the following steps:

1. In the Tag Window, select the **Tag Editor** tab.
Note If the Tag Window is not visible, select **View ► Tag Editor** from the Menu Bar.
2. Double-click the **Data Source** field associated with the *ElevatorRamp* tag.
The field changes to a drop-down list box.
3. Select **CTCParker.InteractXCOMMServer** from the drop-down list.
4. Double-click the **Address** field.
5. Type *Sim.PLC.RAMP (120,0,100,2)*.
The following paragraphs explain the syntax you have typed into the Address field:

Sim.PLC

Sim.PLC addresses the Simulator driver. This structure is determined by the devices defined in the Communications window. From the Application Browser, double-click **Communications** to display the window shown below.



RAMP (120,0,100,2)

RAMP (Rate, Low Limit, High Limit, Increment) is the syntax of the Simulator's RAMP function. The RAMP function allows you to create a value that will either increment or decrement through a numeric range. Set the desired range using the Low and High Limits. In this example the range is 0 to 100. The increment value in this example is 2. The Rate is measured in milliseconds. Thus, in this example, the Ramp function ramps up from 0 to 100, incremented by 2 every 120 milliseconds.

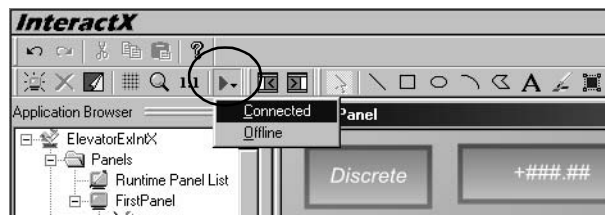
6. Leave all other fields in the Tag Editor at their default settings.
7. Close the Communications window if opened.

- Click the **Save** icon on the Toolbar.
This completes the configuration of the *ElevatorRamp* tag.

Subtask 6.6 - Run the Application

In this section, you will run the application to view the performance of the tool objects you have added.

- From the Toolbar, click the **Run** icon and choose **Connected**, as shown below.



Note You must choose Connected in order to communicate with the Simulator.

- Notice that the Numeric Display and the Gauge are tracking the RAMP function of the Simulator.
As each new limit is reached, the Numeric Display changes color.
- When you are finished, click the **Exit** Button to return to the Designer.

This completes the addition and configuration of tool objects in your InteractX tutorial application. In the next section, you will add graphics that connect to the *ElevatorRamp* tag you created.

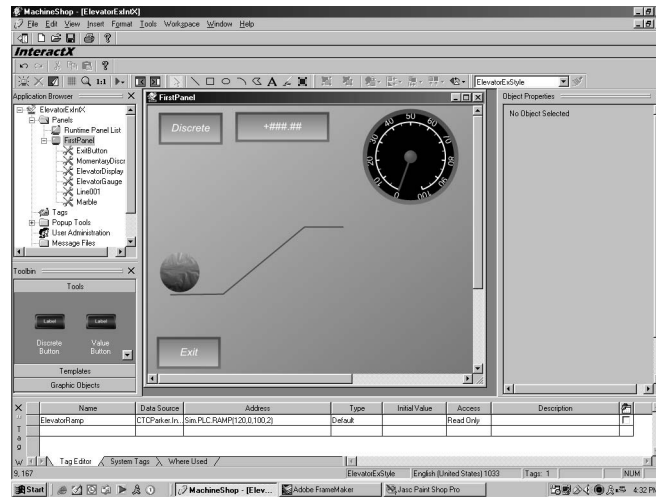
Task 7 - Add Graphic Objects

In this section, you will add simple graphics. These graphics will include an elevator, drawn with the line object, and a marble, drawn with the ellipse object.

You will then configure the different property pages of the graphics you create, configure the graphics to communicate with the Simulator, and add animation.

Getting Started with the Designer

When you are finished, your development screen will look like the one pictured below.



Subtask 7.1 - Use the Line Object

The Line object can be used to draw both single lines and series of connected line segments.

In this section, you will draw a series of three connected line segments to represent an elevator.

To use the Line object to draw an elevator, complete the following steps:

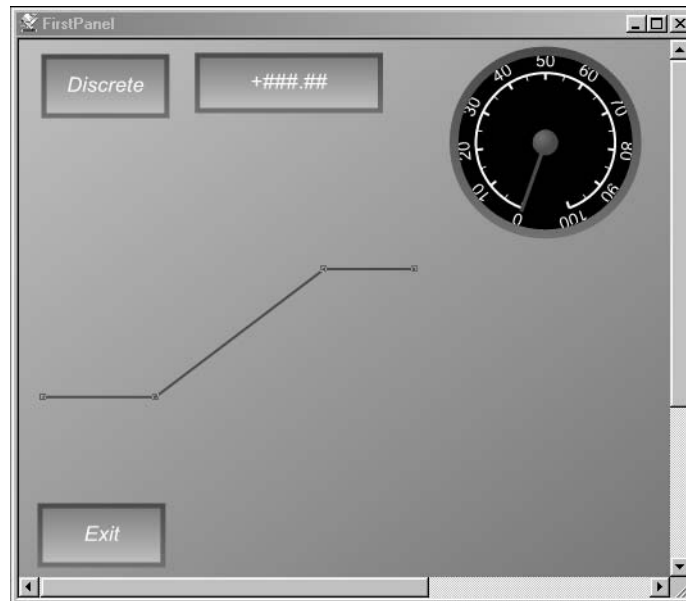
1. From the **Toolbar**, click the **Line** object icon, shown below.

Line command icon



2. Place the cursor on the left side of the panel.
This is where the line will begin. Leave enough room above and to the right of the cursor to create an elevator shape.
3. Press and release the left mouse button to begin the line.
4. Press and release the left mouse button each time you want to create a new point on the line.

The points you create on the line should be as close as possible to those shown in the image below.



5. When the elevator shape shown above is complete, double-click the left mouse button to end the line. You can also terminate the line by pressing **ESC**.
6. From the Object Properties window, make the following change to the line's **Name and Description** property page:
 - a. Change the **Name** of the line to *Elevator*.
 - b. Change the **Description** of the line to read: *A line graphic that represents an elevator.*
7. Select the line's **Appearance** property page and make the following changes:
 - a. Change **Color 1** to the color *Green*.
 - b. Change **Width** to the third line width in the drop-down list.
8. Click the **Save** icon.
This completes the addition and configuration of a line graphic that represents an elevator.

Subtask 7.2 - Use the Ellipse Object

The Ellipse object can be used to draw both ellipses and circles. To create a circle, press and hold the **Ctrl** key on your keyboard as you draw the shape.

Getting Started with the Designer

Likewise, when using the Rectangle object, you can press and hold the **Ctrl** key on your keyboard to draw a square.

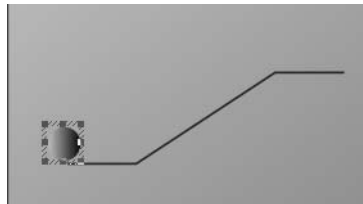
In this section, you will draw a circle that represents a marble. You will position the marble on the elevator you drew in the last section, and animate the marble so that it travels up the elevator. You will use the RAMP function associated with the *ElevatorRamp* tag to control the animation of the marble.

To use the Ellipse object to draw a circle, complete the following steps:

1. From the **Toolbar**, click the **Ellipse** object icon, shown below.



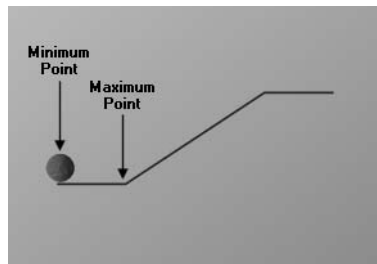
2. Place the cursor close to the elevator you drew earlier.
3. Press and hold the **Ctrl** key as you click and drag the mouse to draw a small circle, such as the one shown below.



4. Position the circle at the point where the elevator begins, as shown in the previous image.
5. On the circle's **Name and Description** property page, make the following changes:
 - a. Change the **Name** of the circle to *Marble*.
 - b. Change the **Description** of the circle to read: *An elliptical graphic that represents a marble*.
6. Select the circle's **Appearance** property page and make the following changes:
 - a. Change **Color 1** to the color **Silver**.
 - b. Change **Material** to the second material option from the top in the drop-down list.
7. Expand the circle's **Connections** property pages by clicking the arrow next to **Connections** in the Object Properties window.
8. Select the circle's **Motion** property page and make the following changes to establish the first connection in animating the circle:

Note Animating a graphic involves making connections based on tags you create. In this case, you will make three new connections, all based on the *ElevatorRamp* tag, and its RAMP function. Each connection you make will animate the circle on a different line segment. Because the elevator is composed of three different line segments, there will be three resulting connections to add. When you are finished, the circle will follow the direction of each line segment as it climbs the elevator.

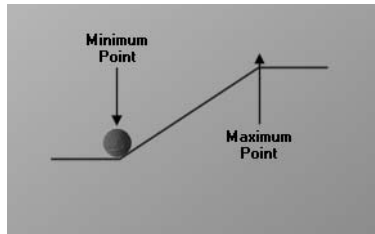
- a. Click **Add New** to add the first connection.
- b. In the **'Conn' Output** field, select the *ElevatorRamp* tag from the drop-down list.
- c. In the **Maximum Value** field, type *30*.
- d. Under Apply Connection, select **Only In Range**.
- e. Under Location Control, select **X**.
- f. Under Connection, click **View**.
Object at Minimum will be selected. This refers to the position on the line segment where the circle will start.
- g. With **Object at Minimum** selected, be sure the circle is at the point where the first line segment begins. See the labeled image below for the correct position of the circle at both its Minimum and Maximum points.



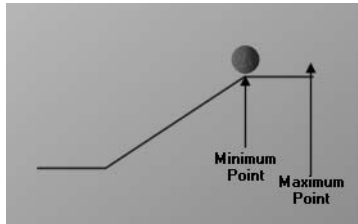
- h. Select **Object at Maximum**.
- i. Drag the circle to the point where the first line segment ends. See the previous image.
- j. Click **Finished** to return to the **Motion** property page.
9. On the circle's **Motion** property page, make the following changes to establish the second connection in animating the circle:
 - a. Click **Add New** to add the second connection.
 - b. In the **'Conn' Output** box, select the *ElevatorRamp* tag from the drop-down list.
 - c. In the **Minimum Value** box, type *30*.
 - d. In the **Maximum Value** box, type *70*.
 - e. Under Apply Connection, select **Only in Range**.
 - f. Under Location Control, select **XY**.
 - g. Under Connection, click **View**.

Getting Started with the Designer

- h. With **Object at Minimum** selected, position the circle at the point where the second line segment begins, as shown below.



- i. Select **Object at Maximum** and drag the circle to the point where the second line segment ends, as shown above.
 - j. Click **Finished** to return to the **Motion** property page.
10. On the circle's **Motion** property page, make the following changes to establish the third connection in animating the circle:
- a. Click **Add New** to add the third connection.
 - b. In the '**Conn**' **Output** box, select the *ElevatorRamp* tag from the drop-down list.
 - c. In the **Minimum Value** box, type *70*.
 - d. In the **Maximum Value** box, type *100*.
 - e. Under Apply Connection, select **Only In Range**.
 - f. Under Location Control, select **XY**.
 - g. Under Connection, click **View**.
 - h. With **Object at Minimum** selected, position the circle at the point where the third line segment begins, as shown below.



- i. Select **Object at Maximum** and drag the circle to the point where the third line segment ends, as shown above.
 - j. Click **Finished** to return to the **Motion** property page.
11. Click the **Save** icon.
This completes the creation, configuration, and animation of a circle that represents a marble.

Subtask 7.3 - Run the Application

In this section, you will run the application again so that you can view your recent graphical additions.

To run the application, complete the following steps:

1. Select the **Run** icon from the **Toolbar** and choose **Connected**.
2. Notice that the circle climbs the elevator at the same rate that the Numeric Display and Gauge change.
All three of these objects are following along the same cycle because they are all three connected to the same *ElevatorRamp* tag. This simple example illustrates how you can animate an object on your screen to provide yet another visual cue for the operator.
3. Click the **Exit** Button to return to the Designer.

This completes the creation, configuration, and animation of graphic objects in your tutorial application.

In the next section, you will add alarms to alert the operator when the limits you establish are reached.

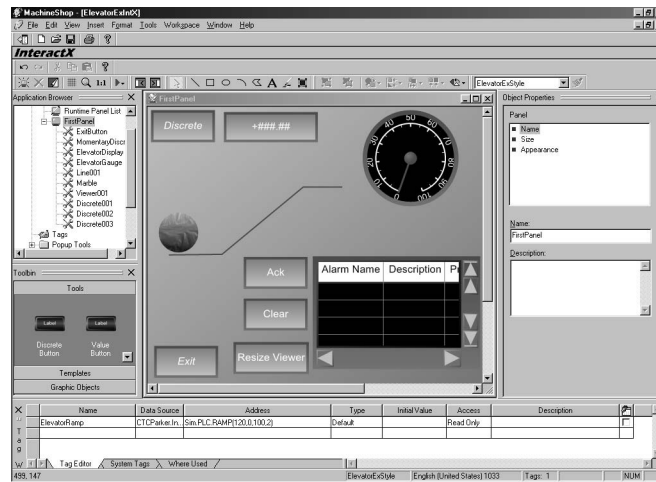
Task 8 - Add Alarms

Alarms are used to immediately alert the operator of an event occurring within the process InteractX is monitoring.

In this section, you will add alarming to your application. First, you will draw and configure an Alarm Viewer, which displays triggered alarms. Then, you will define the conditions that will cause alarms to trigger. Finally, you will add two buttons that allow the operator to acknowledge and clear alarms.

Getting Started with the Designer

When you are finished, your development screen will look like the one pictured below.



Subtask 8.1 - Insert an Alarm Viewer

In this section, you will insert an Alarm Viewer that will display a summary of any alarms that are activated. You can also set up an Alarm Viewer to show alarm history, alarm summary, or alarm controls. Alarm History displays detailed information about alarms that were activated. Alarm Summary provides information about all configured alarms in your application. Alarm Controls allow you to enable or inhibit alarms.

To insert an Alarm Viewer that displays alarm summary, complete the following steps:

1. From the Toolbar, ensure that *ElevatorExStyle* is the active style in the Style Menu.
2. Choose **Insert ► Alarm Viewer** from the Menu Bar.
Note The cursor will change to the draw mode cursor.
3. Click and drag the mouse to draw an Alarm Viewer in the lower right corner of the panel.

4. Click the white arrow on the Toolbar, as shown below. This will change InteractX from draw mode into selection mode.



You can now change the size of the Alarm Viewer by dragging one of its handles. You can also move the Alarm Viewer by clicking in the middle of the Viewer and dragging it to the desired position.

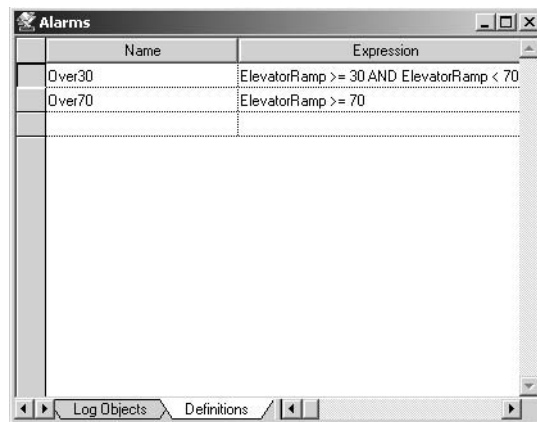
5. Select the Alarm Viewer's **Name and Description** property page and make the following changes:
 - a. Change its **Name** to *AlarmSummary*.
 - b. Change its **Description** to read *Alarm Viewer that displays alarms associated with the ElevatorRamp tag*.
6. Select the Alarm Viewer's **Alarm Fields** property page, make the following changes:
 - a. Uncheck **Time On** and **Time Ack** so that these fields will not display in the Alarm Viewer.
 - b. Be sure the following fields are checked so that these field will display in the Alarm Viewer: **Alarm Name**, **Description**, **Priority**, **Alarm Value**, **Current Value**, and **Group**.
7. On the Alarm Viewer's **Appearance** property page, make the following changes:
 - a. Set the **Button Width** to approximately 10%.
 - b. Set the **Display Depth** to approximately 5%.
 - c. Set the **Button Depth** to approximately 20%.
 - d. Choose **Silver** as the **Button Color**.
 - e. Check **Horizontal Scroll Buttons** to enable this option.
8. Expand the **Appearance** related property pages.
9. Select the Alarm Viewer's **Bezel** property page and set the **Width** to approximately 7%.
10. On the Alarm Viewer's **Header** property page, make the following changes:
 - a. In the Header Text Section, choose black as the **Color**.
 - b. In the Header Background area, choose white as the **From Color**.
 - c. Be sure **Gradient** is set to none (the first option in the list).
 - d. Be sure **Material** is set to none (the first option in the list).
11. Save all changes.
This completes the creation and configuration of the Alarm Viewer.

Subtask 8.2 - Define Alarms

In this section, you will define the alarms that will appear in the Alarm Viewer using mathematical expressions. These expressions will apply to the *ElevatorRamp* tag.

To define the alarms, complete the following steps:

1. Right-click the topmost root folder, *ElevatorExIntX*, and choose **Add-Ins ► Alarms**.
A folder entitled **Alarms** will appear in the Application Browser.
2. Expand the Alarms folder and double-click **Alarm Definitions**.
The window shown below will appear except with no alarms defined.
The image shows what this will look like when you are finished with this tutorial.



3. Make the following entries into the Alarm Definitions window to create the first alarm definition:
 - a. Double-click the first open **Name** field, and type *Over30*.
Because you checked Alarm Name on the Alarm Viewer's **Alarm Fields** property page, the names of the alarms will display in the Alarm Viewer. Therefore, giving the alarms meaningful names will help the operator quickly identify problems.
 - b. Double-click the **Expression** field and type *ElevatorRamp >=30 AND ElevatorRamp < 70*.
This expression will cause an alarm to trigger if the value held in the *ElevatorRamp* tag is greater than or equal to 30 and the value is less than 70.
 - c. Double-click the **Priority** field and choose **Urgent** from the drop-down list.

Note To view the properties of the different Alarm Priorities, choose **Setup** from the expanded Alarms folder in the Application Browser.
4. Make the following entries into the Alarm Definitions window to create the second alarm definition:
 - a. Double-click the next open **Name** field, and type *Over70*.

- b. Double-click the **Expression** field and type *ElevatorRamp >= 70*. This expression will cause an alarm to trigger if the value held in the *ElevatorRamp* tag is greater than or equal to 70.
- c. Double-click the **Priority** field and choose **Critical** from the drop-down list.
5. Save all changes. This completes the alarm definitions.
6. Close the Alarm Definitions window.

Subtask 8.3 - Add an Acknowledge Button

During Runtime, if an alarm triggers, the operator should acknowledge the alarm. Acknowledging an alarm simply means that the operator recognizes that the alarm has triggered. It can be useful for administrators to know how long an alarm sounded before it was acknowledged by the operator. Administrators can set up their Alarm Logs to record the time an alarm triggered and the time an alarm was acknowledged. For complete instructions about setting up Alarm Logs, consult the online help.

In order for the operator to acknowledge an alarm, you must create an acknowledge button. In this section, you will add an acknowledge button to your application by inserting a Discrete Button and assigning a predefined acknowledge system tag to the button.

To add an acknowledge button, complete the following steps:

1. From the **Tools** drawer of the **Toolbin**, click and drag the tool object labeled **Discrete** onto the panel.
2. Position the Discrete Button on the left side of the Alarm Viewer.
3. With the Discrete Button selected, apply the *ElevatorExStyle* from the Toolbar.
4. Select the Discrete Button's **Name and Description** property page and make the following changes:
 - a. Change its **Name** to *AcknowledgeButton*.
 - b. Change its **Description** to read *A Discrete Button set up to acknowledge alarms that appear in the Alarm Viewer*.
5. Select the Discrete Button's **Type and Action** property page, ensure that the **Type** is set to **Momentary** and then make the following changes:
 - a. In the '**Switch**' **Input** box, choose *_Alarm_AckAlarms* predefined system tag from the drop-down list.
 - b. Leave the '**Lens**' **Output** box empty.
6. Select the Discrete Button's **Appearance** property page to expand the pages.
7. Select the Discrete Button's **Lens State 0** property page, and leave all but the following at their defaults:
 - a. Click the **Edit** Button and type *Ack*.

- b. Click **Apply to All States**.
8. Save all changes.
This completes the creation and configuration of an acknowledge button.

Subtask 8.4 - Add a Clear Button

After an alarm's condition is no longer being met, and it has been acknowledged by the operator, it can be cleared (removed from the display).

To allow the operator to clear alarms from the Alarm Viewer, you must create a clear button. In this section, you will add a clear button to your application by copying the acknowledge button you just created and assigning a pre-defined clear system tag to the button.

To add a clear button, complete the following steps:

1. Click the acknowledge button you created in subtask 8.3 to select it.
2. Once selected, right-click the acknowledge button.
3. Select **Copy**.
4. Right-click a blank area of your panel and select **Paste**.
A copy of the acknowledge button is created and overlaps the original button.
5. Position the Discrete Button on the left side of the Alarm Viewer beneath the acknowledge button.
6. Select the Discrete Button's **Name and Description** property page and make the following changes:
 - a. Change its **Name** to *ClearButton*.
 - b. Change its **Description** to read *A Discrete Button set up to clear alarms from the Alarm Viewer*.
7. Select the Discrete Button's **Type and Action** property page, ensure that the **Type** is set to **Momentary** and then make the following changes:
 - a. In the '**Switch**' **Input** box, choose **_Alarm_ClearAlarms** pre-defined system tag from the drop-down list.
 - b. Leave the '**Lens**' **Output** box empty.
8. Press and hold the **Alt** key and then double-click the Discrete Button to edit the lens state without selecting the property page in the Object Property window.
9. Type *Clear*.
10. Select the Discrete Button's **Lens State 1** property page and repeat steps 7 and 8.
11. Align the Acknowledge Button and the Clear Button by completing the following steps:
 - a. Select the Acknowledge Button.
 - b. While pressing and holding the **Ctrl** key on your keyboard, select the Clear Button.

- Note** Pressing and holding the **Ctrl** key allows you to select multiple objects. You may also select multiple object by moving the cursor over the top left edge of an object, holding down the left mouse button, and dragging over all selected objects. This creates a “rubber-band” and selects all objects within the rectangle.
- c. Choose **Format ► Alignment ► Align Left** from the Menu Bar.
12. Save all changes.
This completes the creation and configuration of a clear button.

Subtask 8.5 - Add a Resize Viewer Button

Alarm descriptions may be larger than the original size of the Alarm Viewer as you placed it on the panel. InteractX provides a very convenient system tag that can be used to resize the Alarm Viewer for easier display of the alarm condition.

To allow the operator to resize the Alarm Viewer, you must create a resize button and assign the associated system tag. In this section, you will add a resize button to your application by copying the clear button you just created and assigning a predefined resize viewer system tag to the button.

To add a resize button, complete the following steps:

1. Right-click the clear button you created in subtask 8.4.
2. Select **Copy**.
3. Right-click in a blank area of your panel and select **Paste**.
A copy of the clear button is created and overlaps the original button.
4. Position the Discrete Button on the left side of the Alarm Viewer beneath the clear button.
5. Drag the left sizing handle to increase the width of this tool. Increase the width by approximately 30%.
The text displayed on the lens will be longer than the text displayed on the acknowledge and clear buttons thus you must increase it's width.
6. Select the Discrete Button's **Name and Description** property page and make the following changes:
 - a. Change its **Name** to *ResizeButton*.
 - b. Change its **Description** to read: *A Discrete Button set up to resize the Alarm Viewer.*
7. Select the Discrete Button's **Type and Action** property page, ensure that the **Type** is set to **Momentary** then make the following changes:
 - a. In the '**Switch**' **Input** box, choose **_Alarm_ResizeViewer** predefined system tag from the drop-down list.
 - b. Leave the '**Lens**' **Output** box empty.
8. Select the Discrete Buttons **Appearance** property page to expand the pages.

9. Select the Discrete Button's **Lens State 0** property page, leave all but the following at their defaults:
 - a. Click the **Edit** Button and type *Resize Viewer*.
 - b. Click **Apply to All States**.
10. Align the Acknowledge Button, Clear Button, and the Resize Viewer Button by completing the following steps:
 - a. Holding down the left mouse and dragging over all three buttons.
 - b. Release the mouse button and all three objects will be enclosed within a rectangle.
 - c. Choose **Format ►Alignment ►Align Left** from the Menu Bar.
11. Save all changes.

This completes the creation and configuration of a resize viewer button.

Subtask 8.6 - Run the Application

In this section, you will run the application again so that you can view the alarms you set up as they trigger in the Alarm Viewer.

To run the application, complete the following steps:

1. Select the **Run** icon from the **Toolbar** and choose **Connected**.
2. Notice that as the value of the *ElevatorRamp* tag reaches the limits you defined in the Alarm Definitions, the two different alarms trigger. As the *ElevatorRamp* tag repeats its cycle, and the alarm condition is no longer being met, the alarms become grayed out in the Alarm Viewer. You can try pressing the acknowledge and clear buttons you created to understand how they work. You can also try using the horizontal scroll buttons you added to the Alarm Viewer to view all the data you chose to display. You can press the resize viewer button and notice that the Alarm Viewer increases in size to accommodate the alarm text.
3. Click the **Exit** Button to return to the Designer.

This completes the creation, configuration, and definition of alarms and the Alarm Viewer.

In the next section, you will add basic VBA functionality to your application.

Task 9 - Add VBA Functionality

In this section, you will write basic VBA (Visual Basic for Applications) code to enhance your application.

First, you will enable your panel to process VBA. Then, by adding code to events associated with the MomentaryDiscrete Button you created earlier, you will cause the marble to disappear and reappear as it travels up the elevator.

To add VBA functionality to your application, complete the following steps:

1. From the Application Browser, right-click **FirstPanel** and select **VBA Enabled**.
2. Right-click the **Discrete Button** named **MomentaryDiscrete** (located in the upper left corner of your panel) and choose **View Code** to open the VBA window.

The VBA Editor will launch with the subroutine for the **ButtonPressed()** event of the **MomentaryDiscrete** Button displayed.

3. Change the subroutine so that it reads as follows:

```
Private Sub MomentaryDiscrete_ButtonPressed ( )

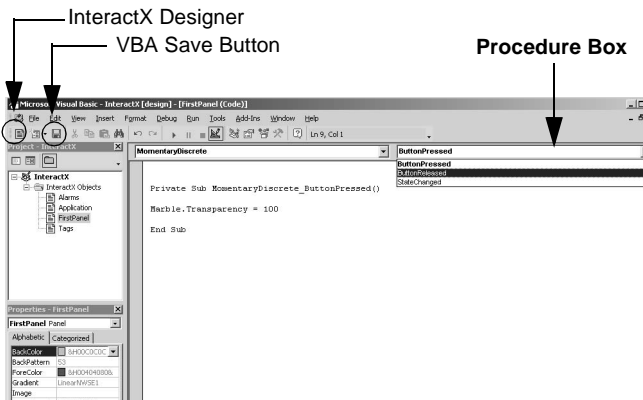
    Marble.Transparency = 100

End Sub
```

Changing the subroutine as shown will assign a value of 100% to the transparency property of the marble when the **MomentaryDiscrete** Button is pressed. In other words, when the **MomentaryDiscrete** Button is pressed, the marble will become invisible.

Note All of the objects and their respective properties are listed in the Object Browser. To explore the Object Browser, right-click anywhere within the VBA window and choose **Object Browser**. When finished using the Object Browser, close it.

4. Position the cursor on the next line after **End Sub**.
5. From the procedure text box shown below, click the arrow and choose **ButtonReleased**.



A subroutine based on the **ButtonReleased** event is displayed.

6. Change the subroutine so that it reads as follows:

```
Private Sub MomentaryDiscrete_ButtonReleased()  
  
    Marble.Transparency = 0  
  
End Sub
```

Changing the subroutine as shown will assign a value of 0% to the transparency property of the marble when the MomentaryDiscrete Button is released. In other words, when the MomentaryDiscrete Button is released, the marble will become completely visible.

7. Click the **Save** icon on the Toolbar and close the VBA window.
8. From the InteractX Designer, click the **Run** icon and choose **Connected**.
If you are prompted to save the application before running, click **Yes**.
9. Press and hold the MomentaryDiscrete Button as the application is running.
The marble disappears.
10. Release the MomentaryDiscrete Button.
The marble reappears.

This completes the tutorial application. You can now exit MachineShop. There are endless possibilities in designing your application using InteractX. The tutorial application was only meant to serve as an introduction to some of the key concepts of the program.

Consult the extensive InteractX online help system for assistance in creating your own, specific InteractX applications.



Getting Started with Runtime

InteractX has two operating environments, the development environment called the “Designer” and the runtime environment called “Runtime”. The Designer lets you use objects to create panels that depict a machine or a process you want to monitor or control during Runtime. For example, this could include connecting graphics up to tags so that they animate during Runtime, reflecting the motion of a process on your plant floor.

InteractX Runtime works in conjunction with the controller to monitor and control machines and processes. Runtime uses COMM Server as the communications interface to connect with controllers.

Configuring Runtime Settings

You can configure different Runtime settings from the Designer. These settings include features such as setting the cursor type, starting InteractX in a window, enabling the secure desktop feature, caching all panels into the workstation’s memory, and displaying data errors during runtime.

There are several additional Runtime settings you can configure. For a detailed description of each of the settings, consult the online help.

To access the Runtime Settings, select **Tools ► Application Properties** from the Menu Bar and then select the **Runtime Settings** tab.

Backing Up Projects

Backing up your project allows you to create a single compressed file that contains all files within your project. MachineShop applies the .ZIP file extension to the backup file.

To back up your project, follow the steps below:

1. From the MachineShop Suite menu bar, select **File ► Backup Project**. The Backup Wizard launches.
2. Select portions of the project to include in the backup. There are several pieces of your project you can choose to include within the backup. The **Project File** is always selected and represents the structure of your MachineShop Suite project. Selecting **Global Objects** will backup any links to other files you have included in your project. Your individual application(s) are also included by default. All of the applications that you have developed for your MachineShop Suite

Getting Started with Runtime

Project will be selected. Deselect those you wish not to include in your backup. This allows you to make backups for specific applications if desired.

3. Click **Next**.
A compressed version of the project is created and stored to the location you specified in the wizard.

Restoring Projects

Restoring your projects is a simple process. Your backup may be located on a variety of devices including removable disk, Compact Flash, hard drive, or a network drive.

To restore your project, follow the steps below:

1. From the MachineShop Suite menu bar, select **File ► Restore Project**. The Restore Wizard launches.
2. Select portions of the project to include in the restore.
There are several pieces of your project you can choose to include within the restore.
The **Project File** is always selected and represents the structure of your MachineShop Suite project.
Selecting **Global Objects** will restore any links to other files included in the project.
Your individual application(s) are also included by default. All of the applications that you have developed for your MachineShop Suite Project will be selected. Deselect those you wish not to include in your restore. This allows you to restore only specific applications if desired.
3. Click **Next**.
The project is created and stored at the location you specified in the wizard.

Transferring Projects

Transfer your projects to the target Runtime system through the MachineShop Suite. You must first transfer your projects prior to running them on the Runtime workstation.

You must configure the two workstations for transfer using the same communications protocol. For example, if you want to transfer files over your Ethernet network, you must configure a network connection on the runtime workstation and a network connection on the development workstation.

To transfer your projects simply double-click the **PowerStation** icon in the MachineShop Suite workspace and follow the instructions on the wizard.



InteractX Time Savers

This chapter describes some of the features of InteractX that help you save time in application development.

There are five features highlighted in this chapter:

- The Toolbin
- Templates
- Styles
- Sound files
- Sample VBA scripts

Using the Toolbin

The Toolbin, shown below, has four different drawers: Tools, Templates, Graphic Objects, and Symbols.



Each of these four drawers holds objects that you can use in all of your InteractX applications. When you first install and open InteractX, the drawers hold predefined objects that you can drag onto the panel for use in your application. When a predefined object from the Toolbin has tags associated with it, those tags are automatically added to your application when you drag the object onto the panel.

You can also drag any object you create from the panel into one of the Toolbin drawers. When you drag an object into the Toolbin, all of the object's associated properties and tags are saved with it. You can then use the object in any InteractX application without having to reconfigure the object each time.

InteractX Time Savers

You can right-click a Toolbin drawer to display a context sensitive menu. Options on the context sensitive menu let you dock or undock the Toolbin; add, delete, or rename drawers; and choose the size of the icons displayed.

Using Templates

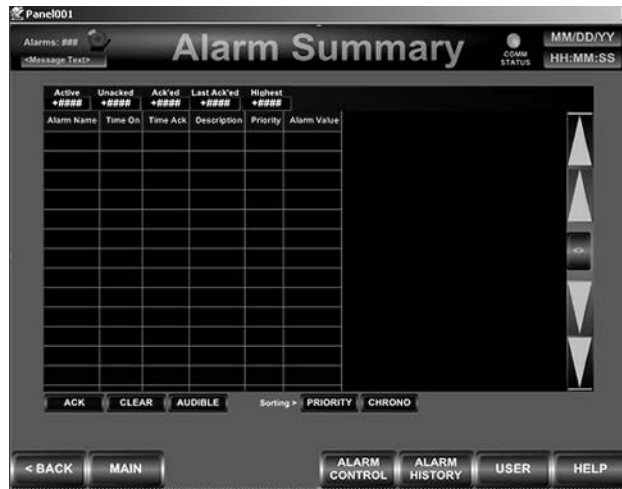
One of the drawers in the Toolbin holds templates. These differ slightly from the objects in the other drawers, as templates are groups of interrelated objects.

There are several predefined templates that install with InteractX. These include:

- Alarm Summary Panel
- Alarm Control Panel
- Alarm History Panel
- Exit Panel
- Navigation Panel with a Previous, Exit, and Next Button

To illustrate how templates save development time, consider the Alarm Summary Panel. If you worked through the tutorial in the last chapter, you created a very basic Alarm Summary by creating and configuring the Alarm Viewer, an acknowledge button, a clear button, a resize viewer button, and a tag association. By dragging the Alarm Summary panel from the Toolbin, you would find your Viewer, Buttons, and tags already defined and configured for you. The Alarm Summary that installs with InteractX is shown below.

You must first create a blank panel prior to dragging your template from the Templates drawer.



When you drag a template onto the panel, all of the objects that comprise the template are grouped. If you want to modify any individual object, you can first ungroup all of the objects or select the object within the Application Browser. To ungroup the object, select the object you wish to ungroup, then click the Ungroup icon on the Toolbar, shown below. All objects will be ungrouped and listed individually in the Application Browser.



You can also save any group of objects that you have created as a template within the Toolbin. All of the properties and tags associated with each object will be saved. To save a group of objects as a template, you must first select all of the objects you wish to make into a template from the panel. Then, click the **Group** icon on the Toolbar, shown below, to save the objects as a group.

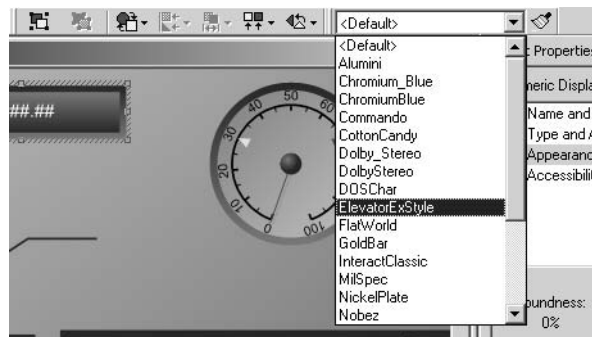


After your objects have been grouped, you can drag them into the Template drawer of the Toolbin. Note that pressing **Ctrl+A** will select all objects on your panel.

Using Styles

Using styles allows you to apply the same appearance properties to all of the objects on your panels.

Several styles are installed with InteractX. These are listed in the Style Menu on the Toolbar, as shown below.



You can also define your own styles. To define your own styles, configure all of the appearance-related properties of a particular object to your specifications. Then, complete the following steps:

1. Select the object so that it appears onscreen surrounded by handles.
2. From the Menu Bar, choose **Format ► Styles**.
3. Click **New**.
4. In the **Style Name** box, type the name you want to assign to the style.
5. Click **Merge**, and then click **Save**.

After you create your own style, it will appear within the Style Menu, and you can apply it to any other object.

To apply a specific style to an object, first select the object. Then, choose the style you want to apply from the Style Menu. This is the active style. Click the paintbrush next to the Style Menu to apply the active style, as shown below.



Finding Sound Files

InteractX has the flexibility that allows you to incorporate sound files within your application. For example, you are able to assign a sound file that will play when a button is pressed. Adding sound to your application can signifi-

cantly improve the usability of your applications by providing additional feedback to an operator.

Sound files are located on your InteractX CD under the **Sounds** folder. You can use Windows Explorer to cut or copy sound files onto your workstation and paste them in your projects application folder.

Referencing VBA Scripts

InteractX comes bundled with Visual Basic for Applications (VBA). The example application shipped with InteractX has an application that demonstrates VBA functionality incorporated within a panel.

Launch MachineShop Suite and open the project **VBA Examples** to view all the VBA example applications.

In a typical InteractX installation you will find the VBA Examples project at the following location:

C:\Program Files\MachineShop Suite\Projects\VBA Examples

This project contains nine VBA examples that you may open and then run to observe how they operate. You may open the Visual Basic Editor code window to review the code within each project.

Example Name	Description
ActiveX	This application demonstrates the use of ActiveX Controls with VBA. All the tools control the light intensity of the panel.
AlarmForm	This application demonstrates how to use a VBA Form to display alarm information.
ButtonTiming	This application will use the Button Pressed and Released events to show the length of time the button was held in the pressed position.
ChangePanelColor	The purpose of this application is to demonstrate a way to change color properties of the panel. The user will decide which color to assign to the BackColor property of the panel.
ChangingButton	The purpose of this example is to demonstrate the ability to change the color and text properties of a button during the ButtonPressed event.

InteractX Time Savers

Example Name	Description
ClickNHit	This application demonstrates how to use the Button Move method, pressed events, and an Operating System timer.
Operator_Activity	This application demonstrates how the operator's activity can be logged to disk.
PanelTimes	This application demonstrates loading and unloading of panels plus the use of the VB Timer function.
Recipe	This application uses a VBA Form to allow the operator to download new values in a recipe to the OPC COMM server.