Pro-face
by Schneider Electric

BLUE Open Studio

Training Manual

INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT NOTICE.

Pro-face and the author(s) assume no responsibility for errors or omissions and no liability is assumed for damages resulting from the use of the information contained herein. Use of the Pro-face product described in this document is subject to the terms of the applicable Pro-face license. These terms include provisions that limit your rights such as use restrictions, disclaimers of warranties and limitations of Pro-face liability. By installing or using the Pro-face product, you agree to accept all of the terms of the applicable Pro-face license. A copy of the applicable license will be displayed upon initial installation of the software. If a copy of the license is not displayed or you require an additional copy of the license, please contact your local distributor. http://www.pro-face.com/trans/en/manual/1015.html.

Table of Contents

Module 1 - General Concepts

SECTION 1 - INTRODUCTION

BLUE Open Studio is a powerful tool that allows you to build HMI (Human Machine Interface) applications for the Automation Industry.

Each BLUE Open Studio project consists of:

- A project tags database to manage all runtime data, including both internal variables and scanned I/O
- Configurable drivers to communicate in real-time with programmable logic controllers (PLCs), remote I/O devices, and other data-acquisition equipment
- Animated HMI screens and controls
- Optional modules such as alarms, events, trends, recipes, reports, scriptable logic, schedulers, a security system, and a complete database interface

This BLUE Open Studio Training Manual is intended for anyone who is using the development software for the first time, and is designed to help you quickly familiarize yourself with the basic features of BLUE Open Studio.

Conventions used in this manual

This manual uses standardized formatting and terminology to make it easier to understand for all users.

Text conventions

This manual uses special text formatting to help you quickly identify certain items:

- Titles, labels, new terms, and messages are indicated using *italic* text (for example, *Object Properties*).
- File names, screen text, and text that you must enter are indicated using `monospace` text (for example, `D:\Setup.exe`).
- Buttons, menu options, and keyboard keys are indicated using a **bold** font style (for example, **File** menu).

Mouse and selection conventions

Because most PCs used for project development run a version of Microsoft Windows with a mouse, this manual assumes you are using a mouse. Generally, a PC mouse is configured for right-handed use, so that the left mouse button is the primary button and the right mouse button is the secondary button.

This manual uses the following mouse and selection conventions:

- Click and Select both mean to click once on an item with the left mouse button. In general, you click buttons and you select from menus and lists.
- Double-click means to quickly click twice on an item with the left mouse button.
- Right-click means to click once on an item with the right mouse button.

- Select also means you should use your pointing device to highlight or specify an item on the computer screen. Selecting an item with a touchscreen is usually the same as selecting with a mouse, except that you use your finger to touch (select) a screen object or section. To select items with your keyboard, you typically use the Tab key to move around options, the Enter key to open menus, and the Alt key with a letter key to select an object that has an underlined letter.
- Drag means to press down the appropriate mouse button and move the mouse before releasing the button. Usually an outline of the item will move with the mouse cursor.

Window conventions

This manual uses the following Windows conventions:

- Dialogs are windows that allow you to configure settings and enter information.
- Text boxes are areas in dialogs where you can type text.
- Radio buttons are white circles in which a black dot appears or disappears when you click on the button. Typically, the dot indicates the option is selected or enabled. No dot indicates the option is cleared or disabled.
- Check boxes are white squares in which a check ( ) appears or disappears when you click on it with the cursor. Typically, a check indicates the option is selected or enabled. No check indicates the option is cleared or disabled.
- Buttons are icons in boxes appear "pressed" when you click on them.
- Lists are panes (white boxes) in windows or dialogs containing two or more selectable options.
- Combo boxes have arrows that, when clicked, show part or all of an otherwise concealed list.
- Dockable windows are windows that you can drag to an edge of the interface and merge with that edge.

Keyboard Shortcuts

To delete a selected item from a screen and store it on the Windows clipboard (replacing any previously selected objects stored on the clipboard), either click **Cut** in the Clipboard group on the Home tab of the Ribbon or press **CTRL+X**.

To copy a selected item without deleting it and store it on the Windows clipboard, either click **Copy** in the Clipboard group on the Home tab of the Ribbon or press **CTRL+C**.

To paste the contents of the Windows clipboard (cut or copied objects) onto the active screen, either click **Paste** in the Clipboard group on the Home tab of the Ribbon or press **CTRL+V**. You can paste a cut or copied object multiple times.

To undo the last action performed (and up to 20 actions taken prior to the last action), either click the Undo tool on the Quick Access Toolbar or press **CTRL+Z**.

To disable the dragging of objects in the screen editor to prevent accidental moves after you've layed out the screen exactly as you want it, either click Disable Drag in the Editing group on the Graphics tab of the Ribbon or press **CTRL+D**.

A complete list of keyboard shortcuts can be found in the appendices at the end of this manual.

Module 2 - Installation and Licensing

SECTION 1 - INSTALLATION

The BLUE Open Studio software suite actually comprises several individual components that can be installed on different computers to perform different functions.

Component	Functions	Platforms
BLUE Open Studio	Project development Project runtime server Project runtime client Remote management Agent to allow remote management	Windows Windows Server Windows Embedded Standard

The architecture of your finished BLUE Open Studio project depends on which components you install, where you install them, and how you connect them to each other.

In most cases, you should first install the full BLUE Open Studio software on your primary workstation. Not only does it set up the project development environment on your computer, it also unpacks the rest of the components so that they can be installed on other computers.

There are separate, downloadable installers for the thin clients.

Requirements for installing and running the software

Note: You must have local Administrator privileges on the computer in order to install the software.

BLUE Open Studio

To install and run BLUE Open Studio, you must have:

- A Windows-compatible computer with a standard keyboard, pointer input (e.g., mouse, trackpad, or touchscreen), and SVGA-minimum display.
- One of the following Windows operating systems:
  - Microsoft Windows 7 Service Pack 1

SECTION 2 - LICENSING

Any machine running BLUE Open Studio requires a license. License settings include:

- Protection type
- Product version
- Product type
- Execution mode
- Remote clients

Protection type

Each license is provided as a softkey.

Softkey

When you install BLUE Open Studio, the program generates a unique Site Code. You can send this Site Code to your software vendor, who will then generate a Site Key to match your Site Code. You can then use the Site Key to install the license on your Windows computer or Windows Embedded device, as opposed to having the license stored on a hardkey.

Note: When you use a softkey, the license is recorded in the computer or device's permanent memory. If the computer is damaged or lost, you will lose the license.

Product version

BLUE Open Studio should all have the same version number, which uses the syntax X.Y+SPWW, where:

- X represents the Family version. The Family version changes only when major enhancements are added to the product technologies and concepts.
- Y represents the Sub-version: The Sub-version changes when minor enhancements and/or new features are added to the product.
- WW represents the Service Pack. The Service Pack version changes when you must install add-on packages to accomplish the following:
  - Upgrade files for the version previously installed
  - Fix bugs in the product (showstoppers and no-workarounds)
  - Provide minor enhancements before releasing the next version of the product

  Each Service Pack release supersedes the previous Service Pack release. For example, SP2 includes all the contents of SP1 and all newly upgraded files, bug fixes, and enhancements. SP3 includes all the contents of SP2 and all new upgraded files, bug fixes, enhancements and so on.

BLUE Open Studio can all run projects that were developed in previous versions of the product. However, previous versions cannot run projects that were developed or modified in newer versions. Opening and modifying a project in a newer version will update the project to that version.

Product type

The product type determines the maximum number of tags and communication drivers you can use in the project.

Execution mode

The execution mode determines what features and functions of the software are operable.

Evaluation mode

Enables all of the BLUE Open Studio development software features.

The first time you install BLUE Open Studio on a computer, the product runs for forty (40) hours in Evaluation Mode. This evaluation period includes any time you run a product module (engineering or runtime). You can use this evaluation period continuously or not; for example, 10 hours a day for 4 days, or 5 hours a day for 8 days, or 10 hours a day for 3 days plus 5 hours a day for 2 days, and so on.

After running for 40 hours in the Evaluation Mode, the evaluation period ends and the license automatically converts to Demo mode until you activate a valid license. You cannot reactivate Evaluation Mode, even if you reinstall the software on your computer.

Engineering Only mode

Allows you to use development software features for an unlimited time. This mode also allows you to run your project for test and development for 72 hours. After that period, the project will shut down, but you can restart it and run for another 72 hours, and so on.

Runtime Only mode

Allows you to use the runtime software to run projects for an unlimited time.

Demo mode

Allows you to use the development software to download projects to remote stations and use the remote management tools. You cannot create or modify screens, worksheets, or project settings in Demo mode.

Allows you to use the runtime software to run projects for testing or demonstration purposes, but they shut down automatically after running for two hours continuously. You can restart the project again and run for another two hours, and so on.

Remote clients

The maximum number of concurrent remote client connections depends on settings of the license installed on the project runtime server. Every license includes at least one Mobile Access connection. The license can be upgraded to add more client connections. The user does not need to install any license on the thin client.

LAB 1 - INSTALLATION AND LICENSING

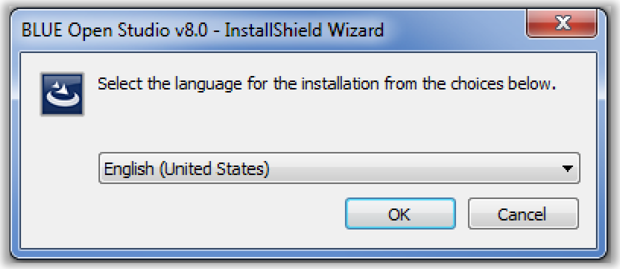
In this lab you will:

- Install BLUE Open Studio
- License the development environment
- Launch the development environment

The instructor will tell you where the BLUE Open Studio installation media is installed. Alternatively, you can download the BLUE Open Studio installation media from the Pro-face website.

Installing BLUE Open Studio

1. Turn on your PC and make sure there are no other programs running.
2. Disable any anti-virus or related programs.
3. Launch Windows Explorer and open the folder containing the BLUE Open Studio installation media. Double-click `setup.exe` to launch the InstallShield Wizard. The installation wizard runs and asks you to select a language for the installation.



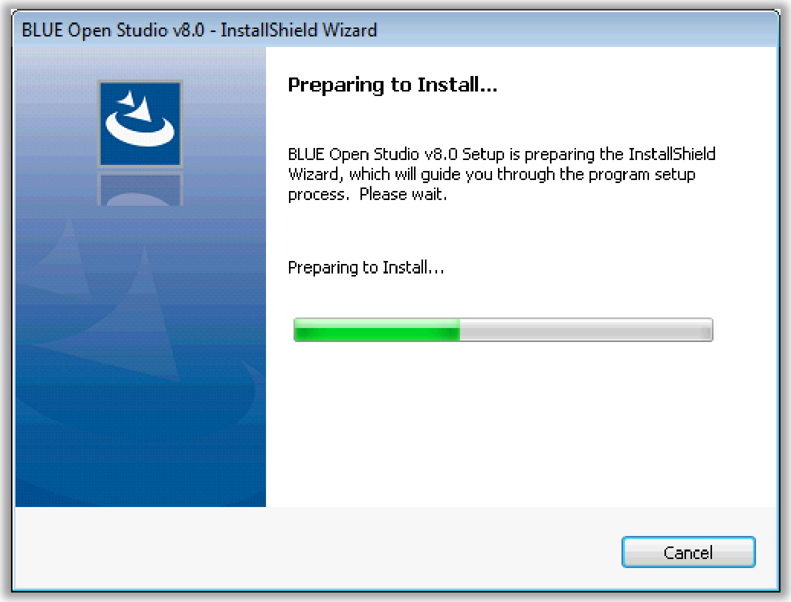
BLUE Open Studio v8.0 - InstallShield Wizard

Select the language for the installation from the choices below.

English (United States)

This selection determines what language the installation wizard will use, as well as what language the project development application's user interface will be in. You can change the language of the user interface later, after you have installed the software.

4. Select a language from the list, and then click **OK**. The Preparing to Install page is opened.

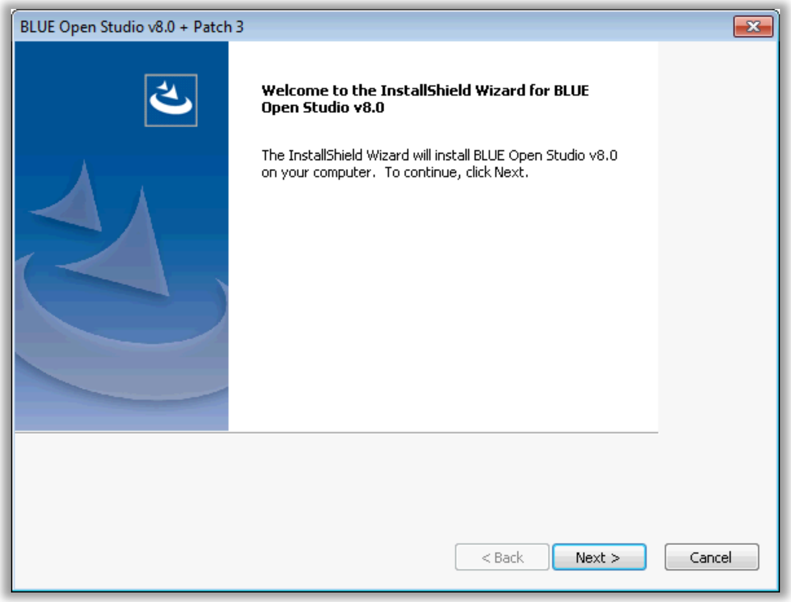

BLUE Open Studio v8.0 - InstallShield Wizard

Preparing to Install...

BLUE Open Studio v8.0 Setup is preparing the InstallShield Wizard, which will guide you through the program setup process. Please wait.

Preparing to Install...

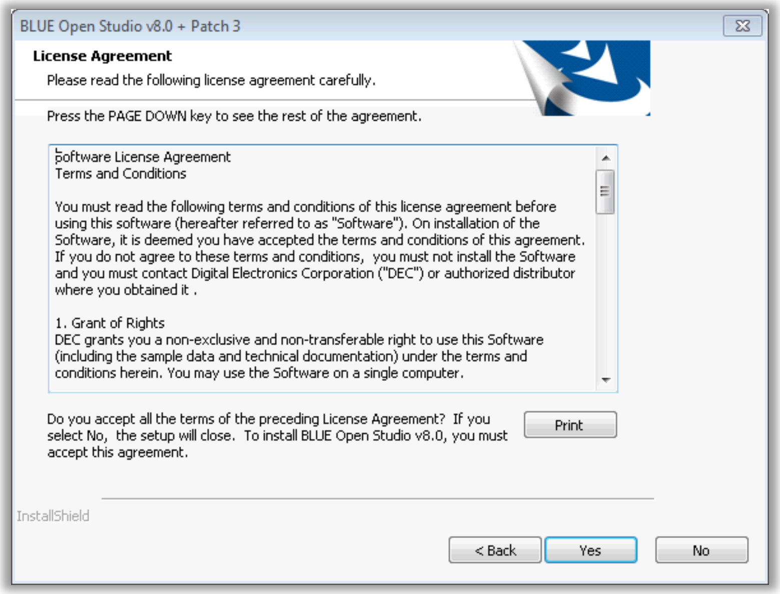
During this step, it automatically installs SafeNet's Sentinel drivers (a part of the software licensing mechanism), Microsoft .NET Framework, and other prerequisites, if required. When the preparation is complete, then the Welcome page is opened.


BLUE Open Studio v8.0 + Patch 3

Welcome to the InstallShield Wizard for BLUE Open Studio v8.0

The InstallShield Wizard will install BLUE Open Studio v8.0 on your computer. To continue, click Next.

5. Click **Next**. The License Agreement page is opened.



License Agreement

Please read the following license agreement carefully.

Press the PAGE DOWN key to see the rest of the agreement.

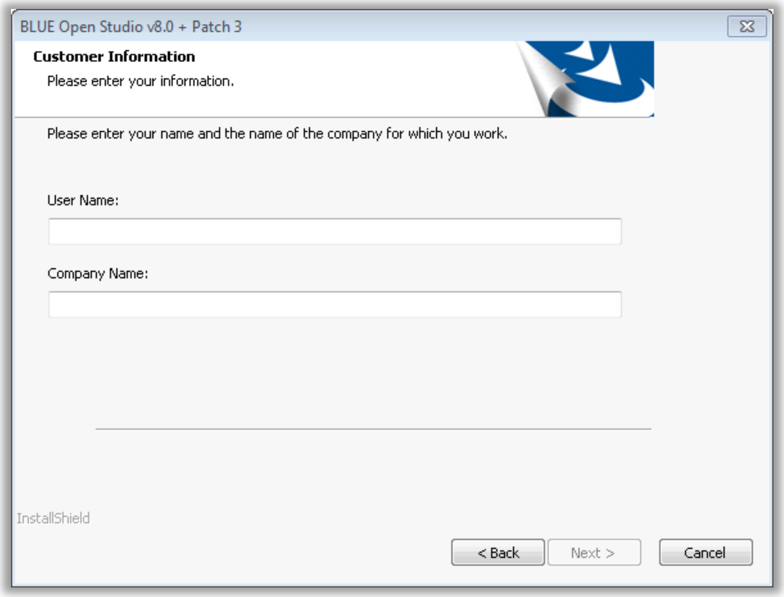
Software License Agreement
Terms and Conditions

You must read the following terms and conditions of this license agreement before using this software (hereafter referred to as "Software"). On installation of the Software, it is deemed you have accepted the terms and conditions of this agreement. If you do not agree to these terms and conditions, you must not install the Software and you must contact Digital Electronics Corporation ("DEC") or authorized distributor where you obtained it .

1. Grant of Rights
DEC grants you a non-exclusive and non-transferable right to use this Software (including the sample data and technical documentation) under the terms and conditions herein. You may use the Software on a single computer.

Do you accept all the terms of the preceding License Agreement? If you select No, the setup will close. To install BLUE Open Studio v8.0, you must accept this agreement.

6. Click **Yes**. The Customer Information page is opened.



Customer Information

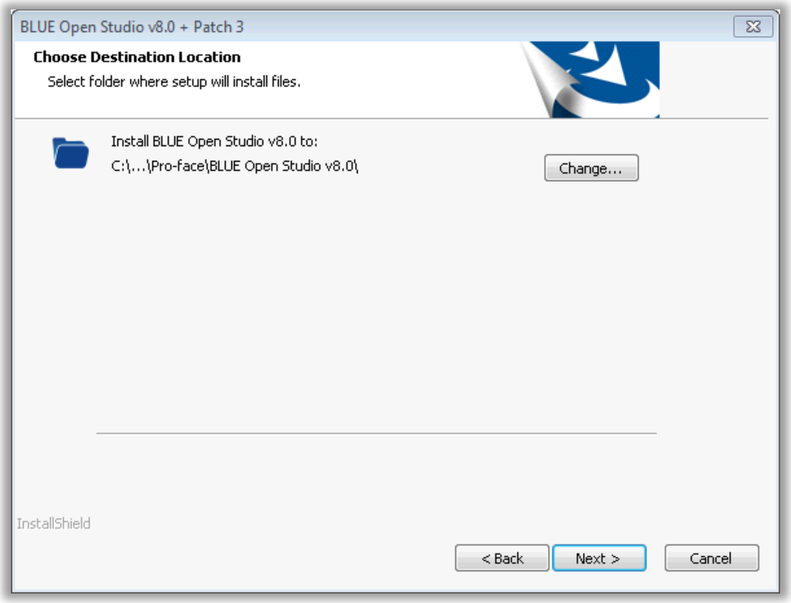
Please enter your information.

Please enter your name and the name of the company for which you work.

User Name:

Company Name:

7. Enter your name in the **User Name** field. Enter your company name in the **Company Name** field. Click **Next**. The Choose Destination Location page is opened.



BLUE Open Studio v8.0 + Patch 3

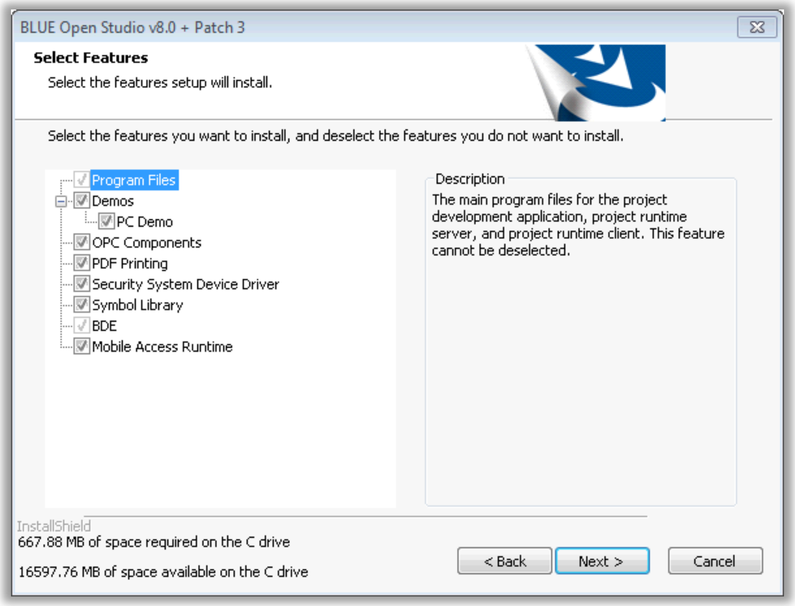
Choose Destination Location

Select folder where setup will install files.

Install BLUE Open Studio v8.0 to:
C:\...\Pro-face\BLUE Open Studio v8.0\

Note: This page is not displayed if you are upgrading an existing BLUE Open Studio installation.

8. Click **Next**. The Select Features page is opened.



BLUE Open Studio v8.0 + Patch 3

Select Features

Select the features setup will install.

Select the features you want to install, and deselect the features you do not want to install.

- ☑ Program Files
- ☑ Demos
  - ☑ PC Demo
- ☑ OPC Components
- ☑ PDF Printing
- ☑ Security System Device Driver
- ☑ Symbol Library
- ☑ BDE
- ☑ Mobile Access Runtime

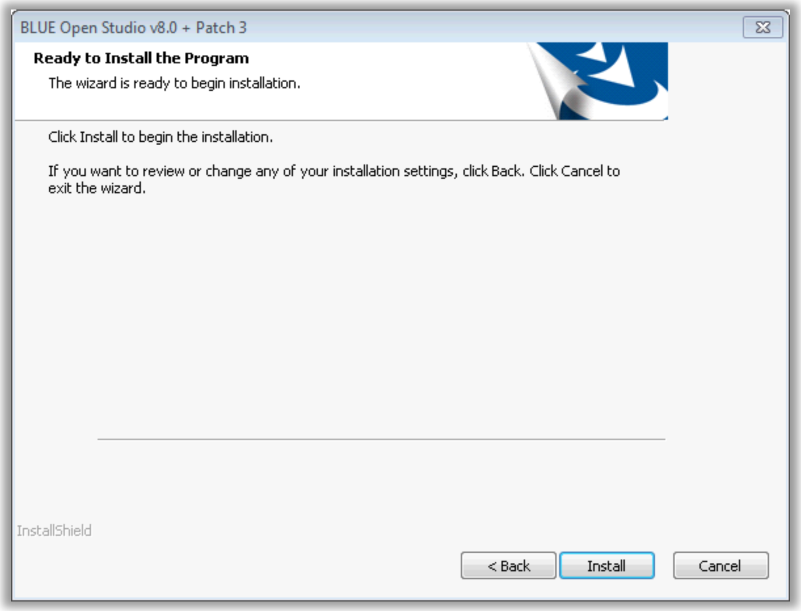
Description

The main program files for the project development application, project runtime server, and project runtime client. This feature cannot be deselected.

667.88 MB of space required on the C drive

16597.76 MB of space available on the C drive

9. Click **Next**. The Ready to Install the Program page is opened.



BLUE Open Studio v8.0 + Patch 3

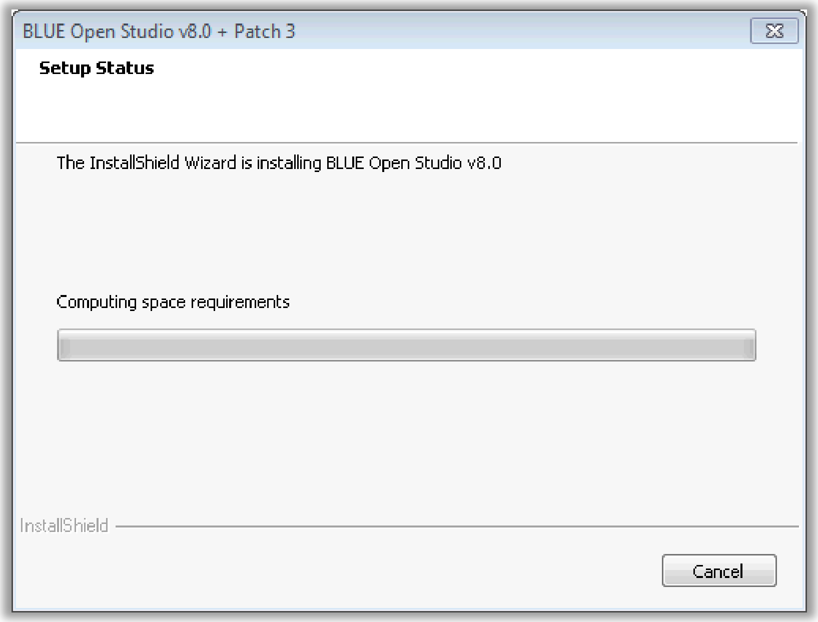
Ready to Install the Program

The wizard is ready to begin installation.

Click Install to begin the installation.

If you want to review or change any of your installation settings, click Back. Click Cancel to exit the wizard.

10. Click **Install**. The Setup Status page is opened.



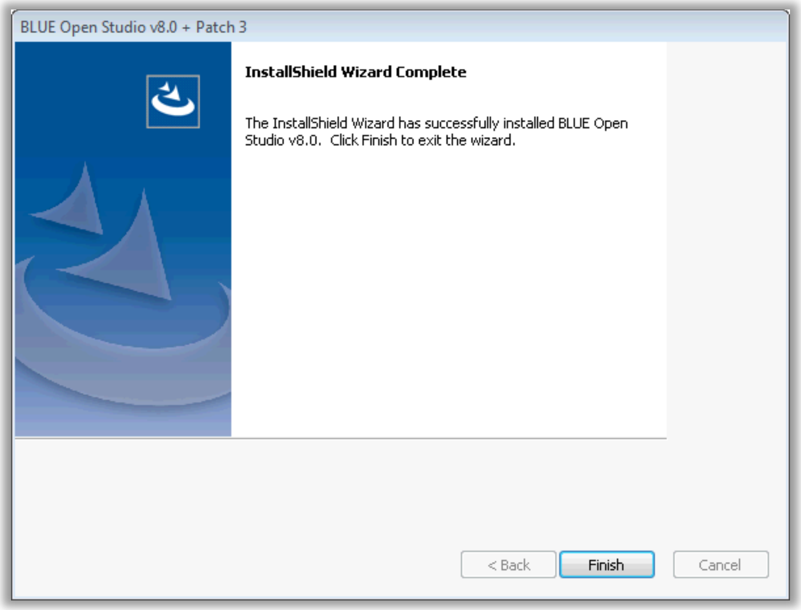
BLUE Open Studio v8.0 + Patch 3

Setup Status

The InstallShield Wizard is installing BLUE Open Studio v8.0

Computing space requirements

When the setup is complete, then the InstallShield Wizard Complete page is opened.



BLUE Open Studio v8.0 + Patch 3

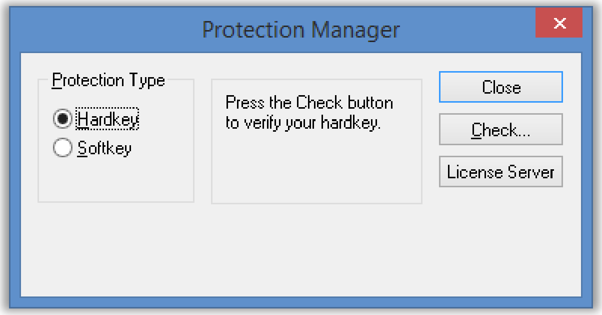
InstallShield Wizard Complete

The InstallShield Wizard has successfully installed BLUE Open Studio v8.0. Click Finish to exit the wizard.

11. Click **Finish**. Restart the PC if prompted.

Licensing the BLUE Open Studio development environment

1. Stop the BLUE Open Studio development environment if it is running.
2. Open the Windows Start menu and launch **All Programs > Pro-face > BLUE Open Studio v8.0 > Register**. The Protection Manager is opened.



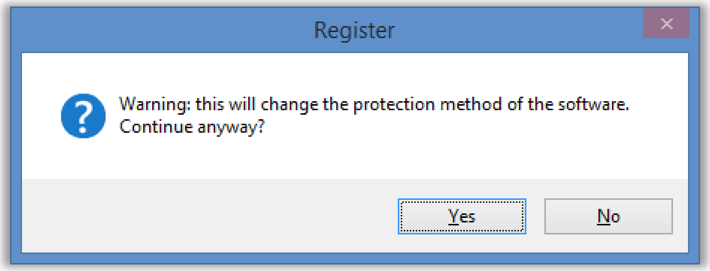
Protection Manager

Protection Type

● Hardkey
○ Softkey

Press the Check button to verify your hardkey.

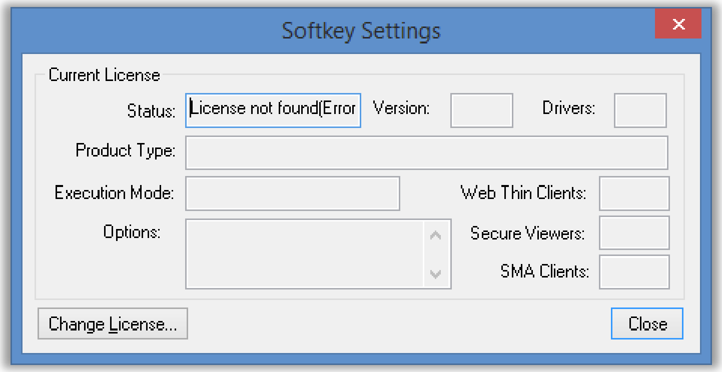
3. In the Protection Type group, select **Softkey** and click **Check**. The Softkey Settings dialog is opened. If you see the following warning, then click **Yes**.



Register

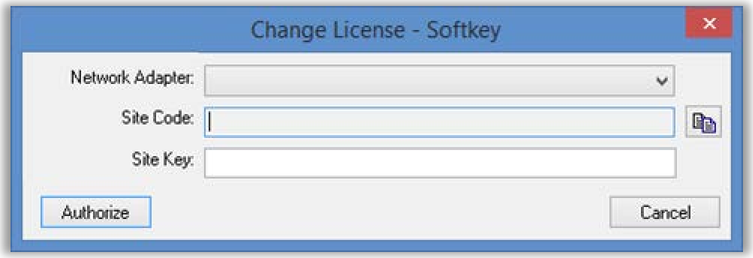
Warning: this will change the protection method of the software. Continue anyway?

Otherwise, the Softkey Settings dialog is opened.



Status: License not found(Error

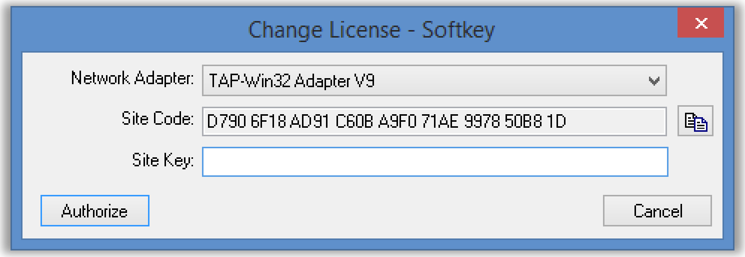
4. Click **Change License**. The Change License - Softkey dialog is opened.



5. Select a network adapter from the list of available adaptors on the **Network Adapter** field.

Note: Do not pick a virtual adapter or an adapter that may be disabled. You must select an adapter that is “always on”. If the selected network adapter becomes disabled or the MAC address of the selected adapter changes, then the license will be invalidated.

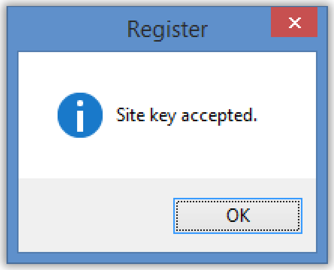
6. After you select a network adapter, then a Site Code is created and displayed in the **Site Code** field.



Network Adapter: TAP-Win32 Adapter V9

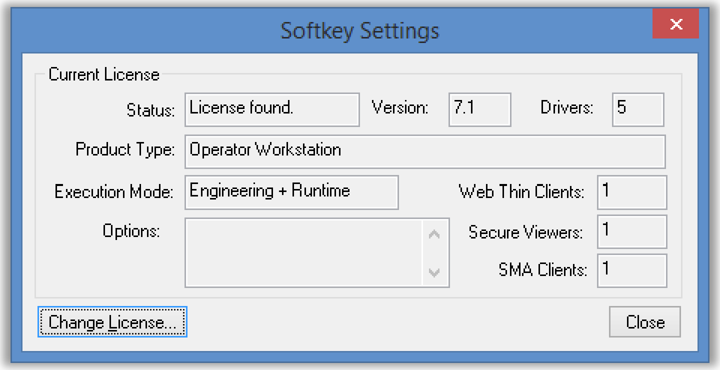
Site Code: D790 6F18 AD91 C60B A9F0 71AE 9978 50B8 1D

7. Submit the Site Code to the BLUE Open Studio licensing department or web licensing service. A Site Key will be provided. Enter this Site Key into the **Site Key** field and click **Authorize**. You will see the following dialog upon success.


Register

Site key accepted.

8. Click **OK**. The Change License - Softkey dialog is closed. The Softkey Settings dialog is updated to reflect your license settings.


Softkey Settings

Current License

Status: License found. Version: 7.1 Drivers: 5

Product Type: Operator Workstation

Execution Mode: Engineering + Runtime Web Thin Clients: 1

Options: Secure Viewers: 1

SMA Clients: 1

Note: Your license settings may be different than those shown above depending on your Site Key.

9. Click **Close**. The Softkey Settings dialog is closed.

10. Click **Close**. The Protection Manager is closed.

Launching the BLUE Open Studio development environment

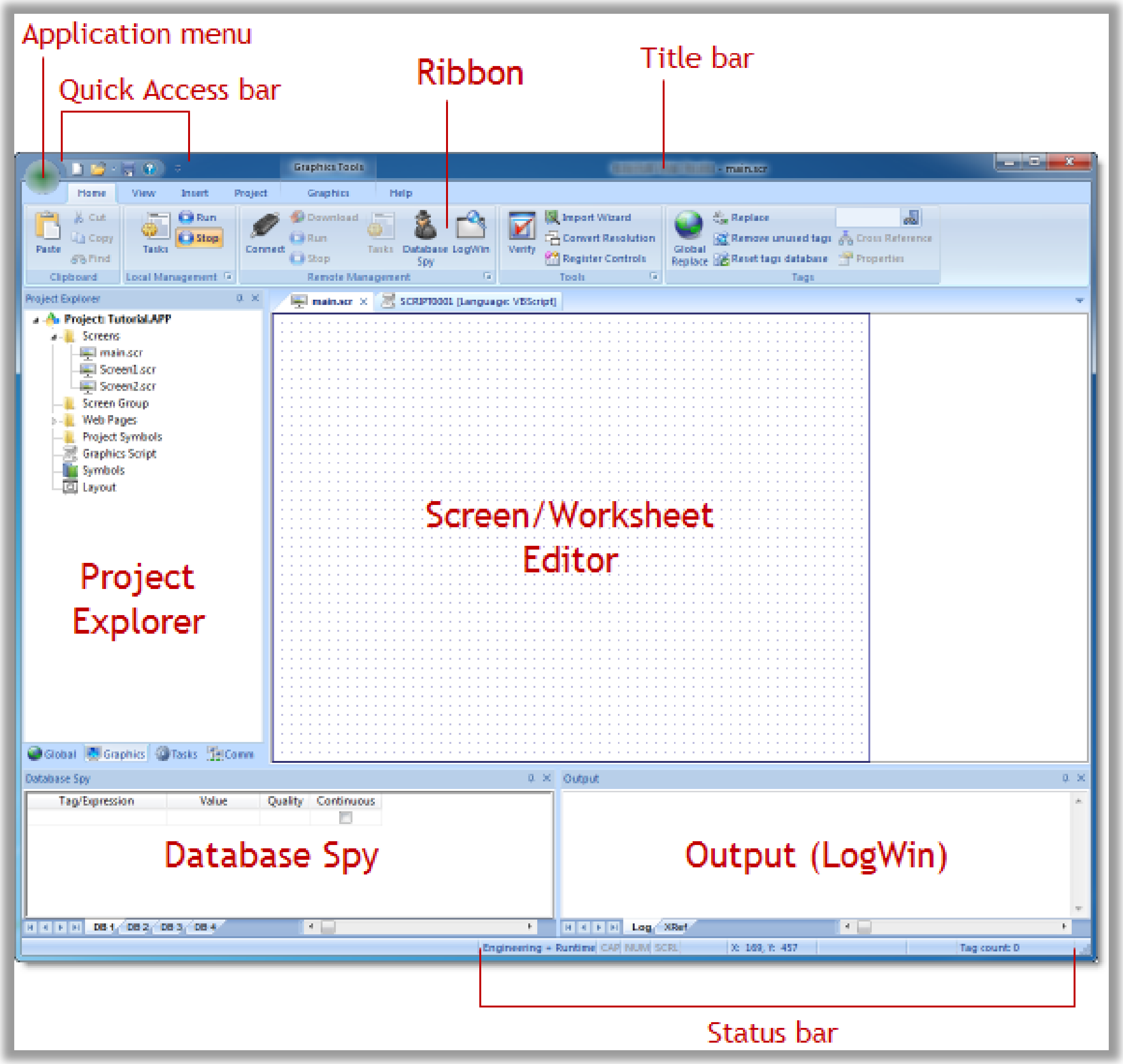
1. Double-click on the **BLUE Open Studio v8.0** icon on your desktop; or

2. Open the Windows Start menu and launch **All Programs > Pro-face > BLUE Open Studio v8.0 > BLUE Open Studio v8.0**.

Module 3 - Getting Started

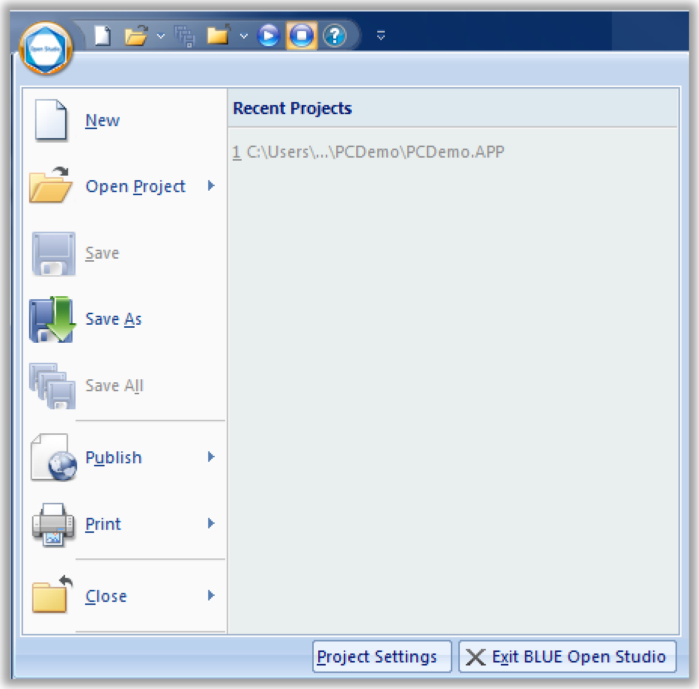
SECTION 1 - THE DEVELOPMENT ENVIRONMENT

BLUE Open Studio incorporates a modern Windows ribbon-based interface to provide an integrated and user friendly development environment.



Application Menu

The Application Menu button opens a menu of standard Windows application commands like New, Open, Save, Print, and Close.



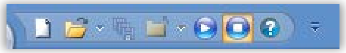
Available commands and features on the Application menu include:

- **Recent Projects**: Lists the most recently opened projects. Click on a project file to open it.
- **New**: Create a new project or project file.
- **Open Project**: Open a saved project or saved project file.
- **Save**: Save the active screen or worksheet file
- **Save As**: Save the active screen or worksheet file at another location.
- **Save All**: Save all open screens or worksheet files.
- **Publish**: Save your project's screens and screen groups in HTML format.
- **Print**: Print the active screen or worksheet or preview what the active worksheet would look like when it is printed.
- **Close**: Close open screen or worksheet files.
- **Project Settings**: Shortcut to Preferences in the Settings group on the Project tab of the Ribbon.
- **Exit BLUE Open Studio**: Save the project database and close the development environment. If you have any open screen or worksheet files that have not been saved, then you are prompted to save your work.

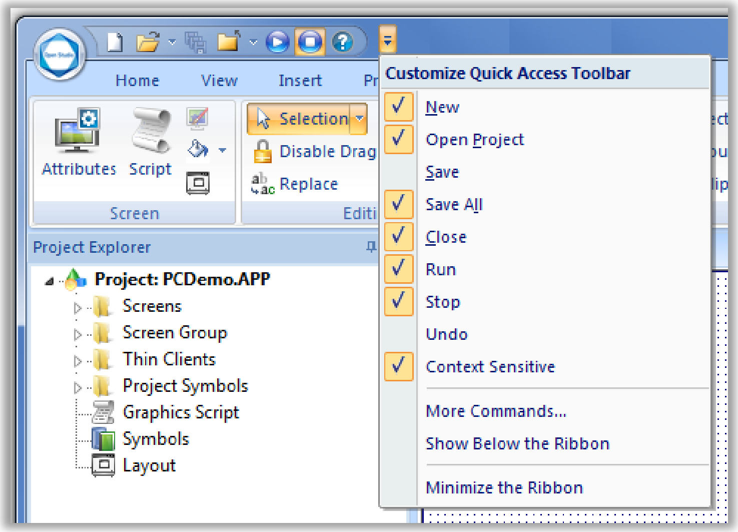
> Note: Closing the development environment does not stop the project runtime server or client, if they are running. Select the Home tab of the Ribbon and click Stop in the Local Management group to stop the project runtime server and client.

Quick Access Toolbar

The Quick Access Toolbar is a customizable toolbar that contains a set of commands that are independent of the Ribbon tab that is currently displayed.



Click the **Customize Quick Access Toolbar** button at the far right of the Quick Access Toolbar to view or modify the commands on the Quick Access Toolbar.

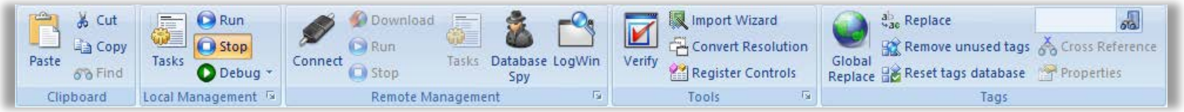


Ribbon Interface

The Ribbon Interface combines the numerous menus and toolbars into a single, user-friendly interface. Almost all application commands are in the ribbon, organized into tabs and groups according to general usage.

Home tag

The **Home** tab of the Ribbon is used to manage your project within the development environment.



The tools are organized into the following groups:

- **Clipboard**: Cut, copy, paste, and find items in project screens and task worksheets.
- **Local Management**: Run and stop the project on the local station (i.e., where the development environment is installed), as well as manage the execution tasks. You can also run a project in Debug mode, for debugging VBScript.

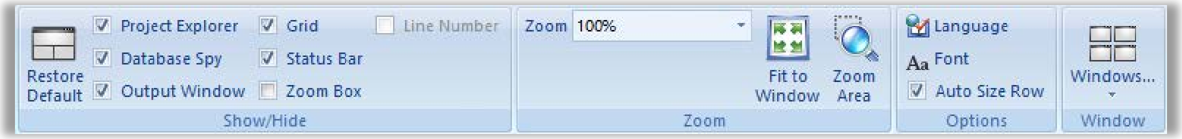
- **Remote Management**: Connect to a remote station (e.g., a Windows Embedded device) so that you can download the project to it, and then run, stop, and troubleshoot the project on that station.

- **Tools**: Miscellaneous tools to verify the project, import tags from other projects, convert screen resolutions, and register ActiveX controls.

- **Tags**: Manipulate tags and tag properties in the project database.

View tab

The **View** tab of the Ribbon is used to customize the look of the development environment itself.



The tools are organized into the following groups:

- **Show/Hide**: Show and hide the different parts of the development environment, as well as restore the default layout.

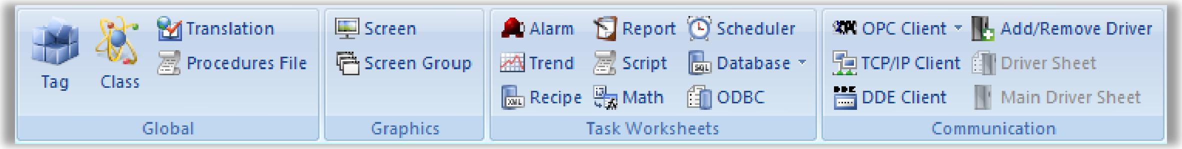
- **Zoom**: Zoom in and out of the screen editor.

- **Options**: Change the language and font used in the development environment.

- **Window**: Arrange the windows in the development environment.

Insert tab

The **Insert** tab of the Ribbon is used to insert new tags, screens, worksheets, and other components into your project.



The tools are organized into the following groups:

- **Global**: Insert tags, classes, translations, and procedures into the Global tab of the Project Explorer.

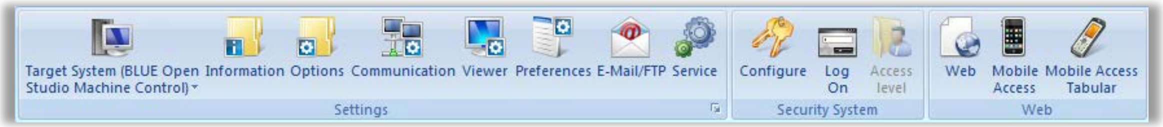
- **Graphics**: Insert screens and screen groups into the Graphics tab of the Project Explorer.

- **Task Worksheets**: Insert task worksheets into the Tasks tab of the Project Explorer.

- **Communication**: Insert server configurations and communication worksheets into the Comm tab of the Project Explorer.

Project tab

The **Project** tab of the Ribbon is used to configure your project settings.

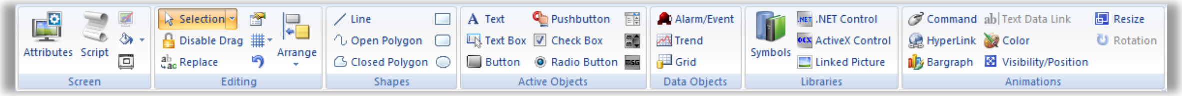


The tools are organized into the following groups:

- **Settings**: Configure the general project settings or enable workgroup collaboration and version control.
- **Security System**: Enable and configure the project security system.
- **Web**: Configure the project to accept connections from a variety of thin clients.

Graphics tab

The **Graphics** tab of the Ribbon is used to draw project screens.



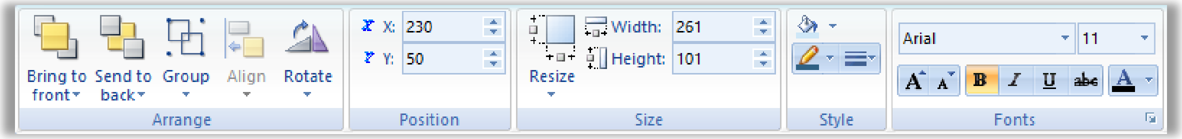
The tools are organized into the following groups:

- **Screen**: Configure settings for the project screen itself, such as its attributes, script, and background color or image.
- **Editing**: Select and edit objects in the project screen.
- **Shapes**: Draw static lines and shapes.
- **Active Objects**: Draw active objects, like buttons and check boxes.
- **Data Objects**: Draw objects that display historical data, like alarms, events, and trends.
- **Libraries**: Select from libraries of premade objects, such as symbols, ActiveX controls, and external image files.
- **Animations**: Apply animations to other screen objects.

Note: The Graphics tab is available only when you have a project screen open for editing.

Format tab

The **Format** tab of the Ribbon is used to format and arrange objects in a project screen.



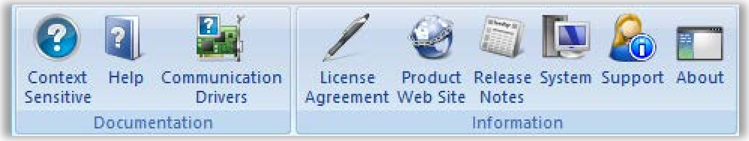
The tools are organized into the following groups:

- **Arrange**: Arrange objects in a project screen, including bring to front and send to back, group, align, and rotate.
- **Position**: Precisely adjust the position of a screen object in a project screen.
- **Size**: Precisely adjust the size of a screen object.
- **Style**: Change the fill and line color of a screen object.
- **Fonts**: Change the caption font of a screen object.

Note: The Format tab is available only when you've selected one or more objects in a project screen.

Help tab

The Help tab of the Ribbon provides additional help with using the software.



The tools are organized into the following groups:

- **Documentation**: Access the documentation for the development environment, including this help file / technical reference and notes for the individual communication drivers.
- **Information**: Access other information about BLUE Open Studio, including the license agreement, product website, and release notes, as well as system and support details that make it easier for Customer Support to assist you.

Project Explorer

The Project Explorer window organizes all of the screens, worksheets, and other items that comprise your project and presents them in an expandable tree-view.

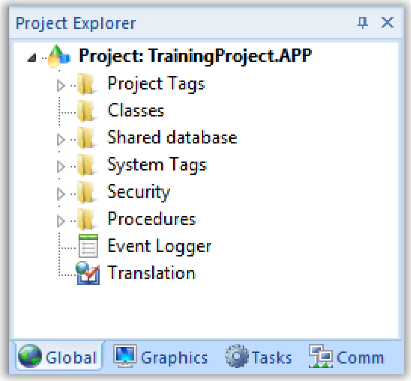
To open or close a folder and view its contents, either click the chevron to the left of the folder or double-click the folder itself.

If you right-click any item in the Project Explorer, then a shortcut menu is opened with contextual commands for that item.

There are four main sections, or tabs, in the Project Explorer: Global, Graphics, Tasks, and Comm.

Global tab

The **Global** tab of the Project Explorer contains the project tags database, as well as other features that apply to the entire project such as the security system, VBScript procedures, and the user interface translation.



Project Explorer

Project: TrainingProject.APP
- Project Tags
- Classes
- Shared database
- System Tags
- Security
- Procedures
- Event Logger
- Translation

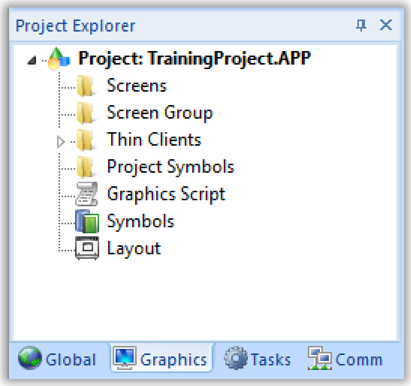
Global | Graphics | Tasks | Comm

The Global tab includes the following folders and worksheets:

- **Project Tags**: The project tags database contains all of the data tags that you create during project development, such as screen tags or tags that read from / write to connected devices.
- **Classes**: Classes are compound tags that you can create to associate a set of values, rather than a single value, with an object. For example, where you may normally create separate tags for a tank's pressure, its temperature, and its fill level, you can instead create a "tank" class that includes all three.
- **Shared Database**: The shared database contains tags that were created in another program and then imported into or integrated with your project.
- **System Tags**: System tags are predefined values such as the date, the time, the name of the current user, and so on. You can use these values to develop supervisory functions and housekeeping routines. All system tags are read-only, which means you cannot add, edit, or remove these tags from the database.
- **Security**: If you choose to enable it, you can use the project security system to control who may log on to your project and what they may do during runtime.
- **Procedures**: Procedures are VBScript functions and sub-routines that can be called by any other script in your project.
- **Event Logger**: The event logger saves important runtime messages and task results to an external database.
- **Translation**: You can use the translation table to develop a multilingual user interface (MUI) for your project.

Graphics tab

The **Graphics** tab of the Project Explorer contains all of the screens, screen groups, and symbols in your project.

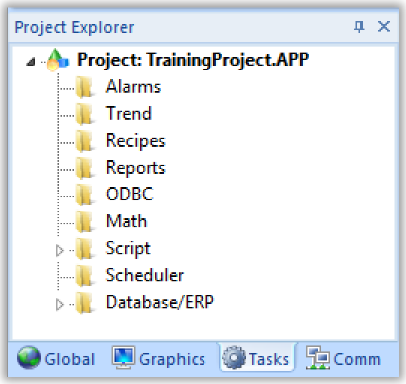


The Graphics tab includes the following folders and worksheets:

- **Screens**: You create screens to provide a graphical interface for your project. Each screen can contain many buttons, sliders, dials, indicators, graphs, and so on.
- **Screen Groups**: You can combine individual screens into screen groups, so that they all open together at the same time.
- **Thin Clients**: You can deploy your project as a web application to be accessed by thin clients such as tablets, and smartphones. You can even deploy different versions of your project with different levels of functionality for each type of client.
- **Project Symbols**: This folder contains all of the custom symbols that you create for your project. A symbol is a group of interconnected screen objects that work together to perform a single function — for example, lines, rectangles, and text fragments that have been arranged to make a slider control.
- **Graphics Script**: You can use this worksheet to define VBScript sub-routines that are called only when the graphics module starts (i.e., when a client station connects to the server and displays the graphical interface), while it is running, and when it ends.
- **Symbols**: The symbols library contains custom symbols that you create and a large selection of premade symbols that are installed with the development environment.
- **Layout**: The layout editor displays all of the screens currently open for editing. You can use it to visualize how the screens are arranged together and reuse screens in multiple layouts — for example, to create a common navigation bar across your entire project.

Tasks tab

The **Tasks** tab of the Project Explorer organizes the worksheets that are processed as background tasks (i.e., server-based maintenance tasks that are not directly related to screen operations or device I/O) during project runtime.

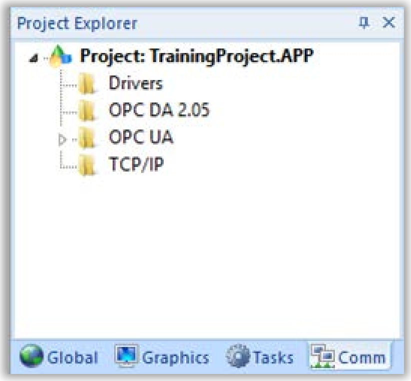


The Tasks tab includes the following folders and worksheets:

- **Alarms**: You can use Alarm worksheets to define when alarms are trigged, how they must be handled, and what messages they generate.
- **Trends**: You can use Trend worksheets to select project tags that should be displayed as data trends and/or saved as historical data.
- **Recipes**: You can use Recipe worksheets to select project tags that will load values from and/or save values to an external file. These worksheets are typically used to execute process recipes, but you can store any type of information such as passwords, operation logs, and so on.
- **Reports**: You can use Report worksheets to design runtime reports that are either sent to a printer or saved to disk.
- **Math**: You can use Math worksheets to develop complex runtime logic using the built-in scripting language.
- **Script**: You can use Script worksheets to develop complex runtime logic using VBScript.
- **Scheduler**: You can use Scheduler worksheets to run commands at specified times, dates, or trigger events.
- **Database**: You can use Database worksheets to set up connections and exchange data with external databases using the standard ADO.NET interface.

Comm tab

The **Comm** tab of the Project Explorer organizes the worksheets that control communication with remote devices, using either direct communication drivers or other common protocols.

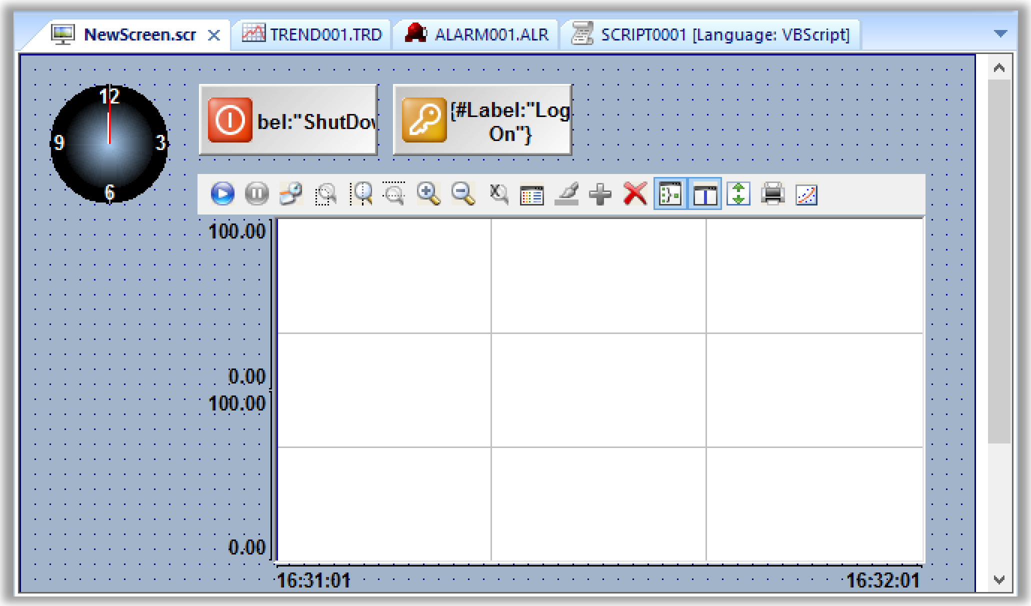


The Comm tab includes the following folders and worksheets:

- **Drivers**: You can use Driver worksheets to communicate with PLCs and other hardware, using any of the hundreds of direct communication drivers that are installed with the development application.
- **OPC DA 2.05**: You can use OPC worksheets to communicate with OPC servers via the OPC Classic protocol.
- **OPC UA**: You can use OPC UA worksheets to communicate with OPC servers via the new OPC Unified Architecture protocol.
- **TCP/IP**: You can use TCP/IP worksheets to configure communication between your own project and other BLUE Open Studio projects. The TCP/IP Client and TCP/IP Server modules enable two or more projects to keep their databases synchronized using the TCP/IP protocol.

Workspace

Use the powerful, object-oriented screen and worksheet editor to create and edit a variety of screens and worksheets for your projects.



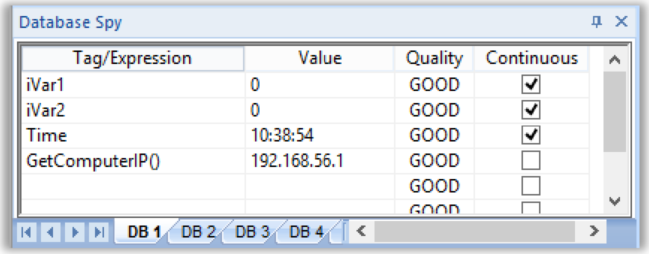
You can input information using your mouse and keyboard, output control data to your processes, and automatically update screens based on data input from your processes.

Other features of the screen editor include:

- Simple point-and-click, drag-and-drop interface
- Grouping objects to preserve the construction steps of individual objects
- Editing objects without having to ungroup internal object components or groups
- Handling bitmap objects and background bitmaps
- Status line support in project windows and dialogs

Database Spy

The Database Spy window is a debugging tool that allows you to monitor and force values to project tags, execute and test functions, and execute and test math expressions in the development.



Tag/Expression	Value	Quality	Continuous
iVar1	0	GOOD	☑
iVar2	0	GOOD	☑
Time	10:38:54	GOOD	☑
GetComputerIP()	192.168.56.1	GOOD	□
		GOOD	□
		GOOD	□

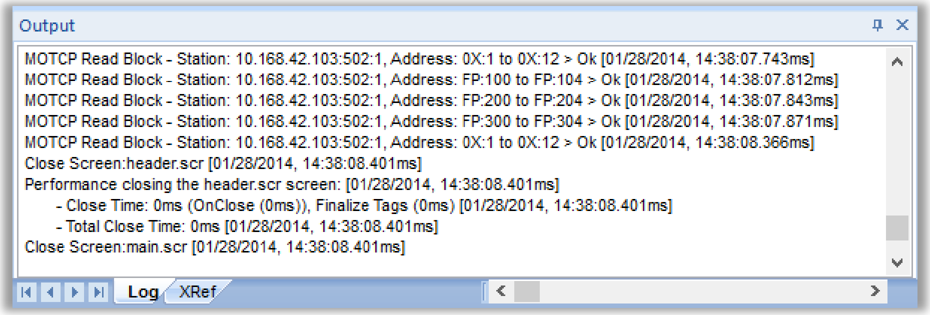
The window contains the following elements:

- For each item that you want to monitor during runtime:
  - **Tag/Expression**: Specify a project tag, system tag, or expression that you want to monitor.
  - **Value**: Displays the value returned by the tag/expression.
  - **Quality**: Displays the quality (GOOD or BAD) of the value returned by the tag/expression.
  - **Continuous**: Select this option to have the development environment continuously evaluate the tag/expression.
- **DB1, DB2, DB3,** and **DB4** tabs: The windows is divided into multiple sheets, so that you can keep your items organized.

> Note: The Remote Management group on the Home tab of the Ribbon includes a Database Spy tool that you can use to connect to a runtime running on a different machine.

Output window

You can use the Output window to view debugging messages.



```
MOTCP Read Block - Station: 10.168.42.103:502:1, Address: 0X:1 to 0X:12 > Ok [01/28/2014, 14:38:07.743ms]
MOTCP Read Block - Station: 10.168.42.103:502:1, Address: FP:100 to FP:104 > Ok [01/28/2014, 14:38:07.812ms]
MOTCP Read Block - Station: 10.168.42.103:502:1, Address: FP:200 to FP:204 > Ok [01/28/2014, 14:38:07.843ms]
MOTCP Read Block - Station: 10.168.42.103:502:1, Address: FP:300 to FP:304 > Ok [01/28/2014, 14:38:07.871ms]
MOTCP Read Block - Station: 10.168.42.103:502:1, Address: 0X:1 to 0X:12 > Ok [01/28/2014, 14:38:08.366ms]
Close Screen:header.scr [01/28/2014, 14:38:08.401ms]
Performance closing the header.scr screen: [01/28/2014, 14:38:08.401ms]
     - Close Time: 0ms (OnClose (0ms)), Finalize Tags (0ms) [01/28/2014, 14:38:08.401ms]
     - Total Close Time: 0ms [01/28/2014, 14:38:08.401ms]
Close Screen:main.scr [01/28/2014, 14:38:08.401ms]
```

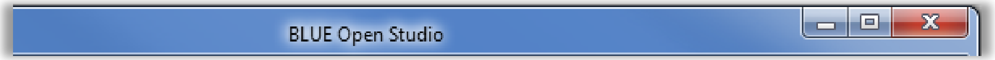
The window contains the following elements:

- **Log** tab: Displays messages generated during project runtime from communication drivers, OPC and TCP/IP transactions, module activation, trace tags, and so on. Also displays error, warning, and informational that may get created while developing your project.
- **XRef** tab: Use the Cross Reference tool in the Tags group on the Home tab of the Ribbon to find every place in the project where a tag is being used. Results appear on this tab, providing path and filename, column, row in the spreadsheet, and so on. So, if something changes in the tag, and produces unexpected or unsuccessful results, you can locate all instances of the tag for debugging purposes.

Note: The Remote Management group on the Home tab of the Ribbon includes a tool called LogWin that you can use to connect to a runtime running on a different machine.

Title Bar

The Title Bar located along the top of the development environment displays the application name (i.e., BLUE Open Studio) followed by the name of the active screen or worksheet (if any).

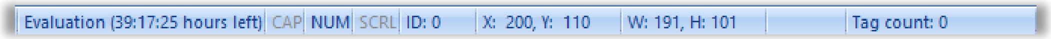


The title bar also offers the following buttons (from left to right):

Button	Action
Minimize —	Minimize the development environment window to the Taskbar.
Restore Down 🗗	Reduce the window to its original (default) size.
Maximize 🗖	Enlarge the window to fill your computer screen.
Close ✕	Save the project database and close the development environment. If you have any open screen or worksheet files that have not been saved, then you are prompted to save your work. This button's function is the same as Exit BLUE Open Studio on the Application menu.

Status Bar

The Status Bar located at the bottom of the development environment provides information about the active screen (if any) and status of the application.



Evaluation (39:17:25 hours left) | CAP | NUM | SCRL | ID: 0 | X: 200, Y: 110 | W: 191, H: 101 | | Tag count: 0

The fields of the Status Bar (from left to right) are described in the following table:

Field	Description
Execution mode	The Execution mode of the application based on the license. (Development Only, Evaluation or Demo)
CAP	Indicates whether the **Caps Lock** key is on (bold) or off (gray).
NUM	Indicates whether the **Num Lock** key is on (bold) or off (gray).
SCRL	Indicates whether the **Scroll Lock** key is on (bold) or off (gray).
ID	The identification number of the object selected on the screen.
Position	When a screen object is selected, this field indicates the position of the object in pixels, relative to the top left corner of the screen, where X is the number of pixels from the left and Y is the number of pixels from the top. When no screen object is selected, the X, Y coordinates are the coordinates of the mouse cursor relative to the active screen.
Object size	The size (in pixels) of a selected screen object, where W is the width and H is the height.
Dragging status	Indicates if drag is disabled (No DRAG) on the active screen.
Tag count	The total number of tags used thus far in the project.

LAB 2 - CREATE THE TRAINING PROJECT

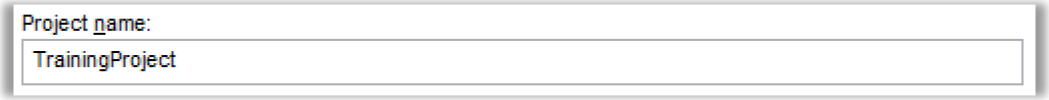
In this lab you will create a new project that you will use for the remaining labs. A folder will be created called `C:\HMI\TrainingProject` for the various worksheets, screens, and settings that will be created as you complete the remaining labs.

1. Launch Windows Explorer and create a folder called `C:\HMI`.

2. Launch the BLUE Open Studio development environment if it is not already open.

3. Click the Application button, and then click **New** on the Application menu. The New dialog is opened.

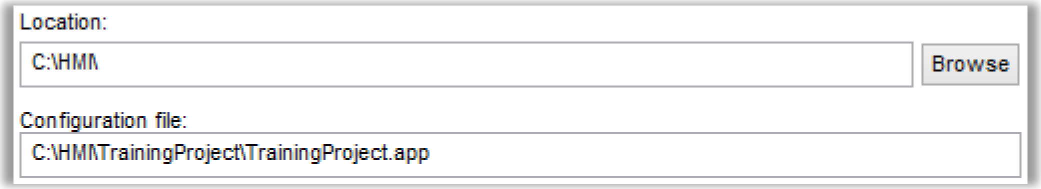
4. In the **Project Name** field, enter `TrainingProject`.



Project name:
TrainingProject

5. Click **Browse** to the right of the **Location** field. The Browse for Folder dialog is opened.

6. Browse to the folder called `C:\HMI` and click **OK**. The Browse for Folder dialog is closed. The read-only **Location** and **Configuration file** fields are filled based on the project name and location.

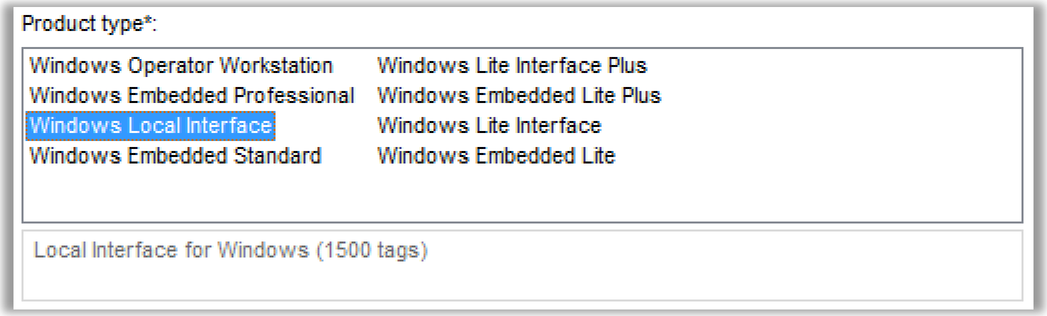


Location:
C:\HMI\

Configuration file:
C:\HMI\TrainingProject\TrainingProject.app

Pro Tip: You can set the default location for new projects by opening Preferences in the Settings group on the Project tab of the Ribbon.

7. In the **Product Type** list box, select **Windows Local Interface**.



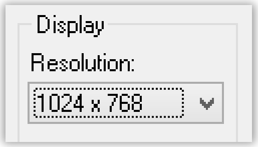
Product type*:

Windows Operator Workstation
Windows Embedded Professional
Windows Local Interface
Windows Embedded Standard
Windows Lite Interface Plus
Windows Embedded Lite Plus
Windows Lite Interface
Windows Embedded Lite

Local Interface for Windows (1500 tags)

Note: The list of available product types is determined by the engineering license installed on your machine. Depending on your license type, you may need to use the scroll bar on the bottom of the Product Type list box to find the product type.

8. Click **OK**. The New dialog is closed and the Project Wizard dialog is opened.
9. In the Display group, click on the **Resolution** combo box and select **1024 x 768**.

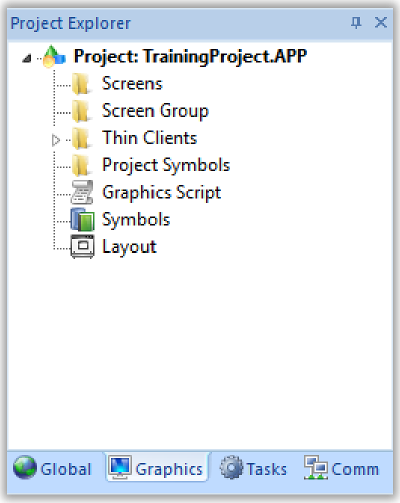


Display

Resolution: 1024 x 768

Note: In most cases, the resolution should match the physical display size of the device that will run your project runtime server and client.

10. Click **OK**. The Project Wizard dialog is closed. The project is created and displayed in the Project Explorer.



11. Launch Windows Explorer and navigate to `C:\HMI\TrainingProject`.

   The `C:\HMI\TrainingProject` folder is called the project folder. Everything relative to your project is stored in this folder. The file called `TrainingProject` with the `APP` extension is your project application file where all your projects settings are stored.

   Other files will be created in the project folder and sub-folders as you complete the remaining labs. You are encouraged to come back and visit this folder as the training course progresses to see what's changed.

SECTION 2 - TAGS AND THE PROJECT DATABASE

Concept

Tags are a core component of any BLUE Open Studio project. Simply put, tags are variables used to receive and store data obtained from communication with communication devices, from the results of calculations and functions, and from user input. In turn, tags can be used to display information, to manipulate screen objects, and to control runtime tasks.

But tags are more than simple variables. BLUE Open Studio includes a real-time database manager that provides a number of sophisticated functions such as time-stamping of any value change, checking tag values against runtime minimum and maximum values, comparing tag values to alarming limits, and so on. A project tag has various properties beyond the value that can be accessed, some even being able to be edited at runtime.

Overview

There are three categories of tags:

- **Project Tags:** Project tags that you create during project development. Places where project tags are used include:
    - Screen tags
    - Tags that read from/write to field equipment
    - Control tags
    - Auxiliary tags used to perform mathematical calculations

  Project tags count against your target system's tag limit whether or not they are used.

- **Shared Tags:** Shared tags are tags from a third-party control application or device added to the project using Tag Integration. They can be used anywhere in your project that you would normally use project Tags.

  You cannot modify shared tags within the project — you must modify the tags in the third-party control application or device.

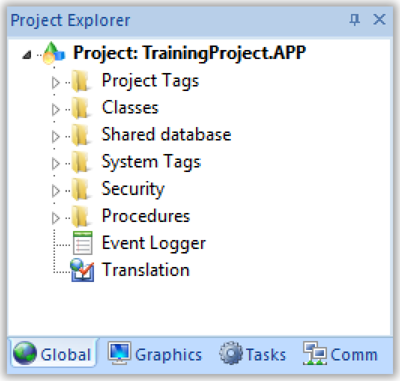
  Shared tags count against your target system's tag limit only when you use them in your project.

- **System Tags:** System tags are predefined tags with predetermined functions that are used for supervisory tasks. For example,
    - Date tag holds the current date in string format
    - Time tag holds the current time in string format

  They can be used anywhere in your project that you would normally use project tags. However, most system tags are read-only, which means you cannot add, edit, or remove these tags.

  System tags do not count against your target system's tag limit.

Project, shared, and system tags are organized into the Project Tags, Shared database, and System Tags folders



Tag Properties

Tag Name

The following guidelines must be observed when naming a tag:

- Tags names must be unique – you cannot specify the same name for two different tags (or functions). If you enter a name of an existing tag, BLUE Open Studio recognizes that the name already exists and will not create this new tag.

- You must begin each tag name with a letter. However, you can use letters, numbers and the underscore character (_) in the tag name.

- You cannot use the following symbols in a tag name:

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

- You can use a maximum of 255 characters for a tag name or Class member name. You can use uppercase and lowercase characters. However, because tags are not case sensitive, you can use character case to make tag names more readable. (For example: **TankLevel** instead of **tanklevel**.)

- Tags names must be unique from system tags and function names.

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

Examples of valid tags include:

Temperature

Pressure_1

x

Data Type

BLUE Open Studio recognizes the following data types:

- **Boolean**: Simple boolean with the possible values of 0 (false) and 1 (true). Typically used for turning objects off and on or for closing and opening objects.
- **Integer**: Integer number internally stored as a signed 32-bit. Typically used for counting whole numbers or setting whole number values.
- **Real**: Real number that is internally stored as a signed 64-bit IEEE-754 flaoting point value. Typically used for measurements or for decimal or fractional values.
- **String**: Character string up to 1024 characters that holds letters, numbers, or special characters. Supports both ASCII and UNICODE characters.

Scope

BLUE Open Studio allows you to decide whether a tag "lives" on the project server or on each remote client station:

- **Server** (default): The tag is maintained on the project server and shared by all connected clients. A change to the tag value affects the entire project.
- **Local**: A virtual copy of the tag is maintained separately on each local station (server + clients). A change to the tag value affects only the station on which the change was made.

Array tags

An Array Tag consists of a set of tags with the same name but with single array index (a matrix of n rows and one column) to distinguish each Tag.

The following guidelines must be observed when creating an array:

- Maximum array size is 16,384.
- You must set the maximum size of each array.

Each array position (including position 0) counts as a tag for licensing restrictions, because each position has an independent value.

To access the array of a tag, follow this syntax:

TagName[ArrayN], or **TagName[ArrayN+c]**

For example: **Tank[0]**, **Tank[1]**, **Tank[2]**, **Tank[500]**

Use the array tag whenever possible, since it optimizes memory usage and simplifies the task of configuration. For example, if you want a display to monitor each tank, you can use array tags to configure a single display, containing tags linked to any tank. For example (using the **nt** tag as an index containing the tank number): **pressure[nt]**, **temperature[nt]**, and **temperature[nt+1]**.

You must specify a maximum index for each array tag in the **size** column of any product data sheet. You can specify n to indicate the array tag has positions 0 through n. For example, if the size of TagA is 3, the tag elements can be **TagA[0]**, **TagA[1]**, **TagA[2]**, and **TagA[3]**.

An array index can be a tag, a numeric value, or an expression with a "**+**" arithmetic operation. When you refer to an array with an index using the + arithmetic operation, you must use the following syntax:

ArrayTagName[Value1+Value2]

Where Value1 and Value2 can be an entire tag or a numeric constant. For example: **temperature[nt+2]** or **temperature[nt+6]**.

Classes

Classes allow for a high degree of encapsulation. A class is a model composed of two or more tag definitions, each with its own data type. You can use Classes in the projects that have items (e.g., Tank liquids) with multiple attributes (e.g., Level, temperature, pressure) to be monitored or controlled.

The following guidelines must be observed when creating a Class:

- Class names **must be unique** – you cannot specify the same name for two different classes (or functions). If you enter a name of an existing class, BLUE Open Studio recognizes that the name already exists and will not create this new class.
- You must define the members and their types.
- The maximum number of members for one Class is 4096.
- The member of one Class cannot belong to another Class. However, you can create members with the same name in different Classes.

When a new Class Tag is created, each member of the class counts as a tag used, because each member can obtain a value. (For example, if you create a class with five members, and then create five tags of this class, this adds up to a total of 25 tags).

To access the member of a Class, use the following syntax:

"**TagName.MemberName**", or " **TagName[ArrayN].MemberName**"

For example: **Motor.RPM**, **Tank[5].Level**.

Use the class tag, whenever possible, since it optimizes memory usage and simplifies the task of configuration. For example, if you want a display to monitor the attributes of a tank, you can use class tags, containing members linked to the tank tag. For example (using the tank tag with members of a tank class): **tank.level**, **tank.temperature**, **tank[4].flow**.

Indirect tags

BLUE Open Studio supports indirect access to tags in the database. For example, consider a tag **X** of type String. This tag can contain the name of any other tag in the database (i.e., it can provide a pointer to any other type of tag, including a class type). The syntax for an indirect tag is simple: **@IndirectTagName**. For example, suppose a tag called X contains a "**TEMP**" string. Reading and/or writing to @**X** gives access to the value of the **TEMP** tag.

Any tag created as a string type tag is potentially an indirect tag (pointer). To refer to a class-type tag, you can declare a string type tag that points to a class tag.

To configure a tag in an Indirect Tag, use the following syntax:

IndirectTag = “ProjectTag”

To access the value of an Indirect Tag, use the following syntax:

@IndirectTag

For example, a project exists which contains the following three tags:

IntTagA, a single tag of type Integer

IntTagB, a single tag of type Integer

StrPointer, a single tag of type String

In this project, the two integer tags have the following values:

IntTagA = 1234

IntTagB = 6789

If the value of StrPointer is “IntTagA”, the value of @StrPointer would be1234

If the value of StrPointer is “IntTagB”, the value of @StrPointer would be 6789

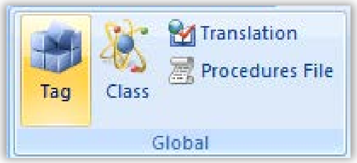
If the value of StrPointer is null (“”) or any value other that the name of an existing tag, the indirect tag @StrPointer would have no value with a BAD quality.

LAB 3 - CREATE PROJECT TAGS

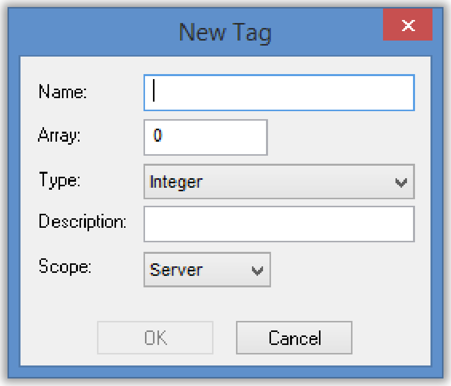
In this lab you will create tags used in subsequent labs.

Create a single tag

1. On the Insert tab of the Ribbon, in the Global group, click **Tag**.



The New Tag dialog is opened.



Name:

Array: 0

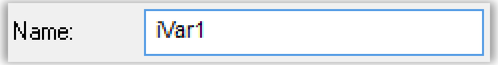
Type: Integer

Description:

Scope: Server

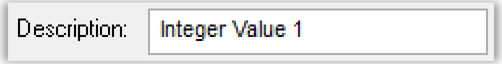
Pro Tip: You can also insert a tag by right-clicking on the Project Tags folder on the Global tab of the Project Explorer and selecting Insert Tag from the shortcut menu.

2. In the **Name** field, enter `iVar1`.



Name: iVar1

3. In the **Description** field, enter `Integer Value 1`.

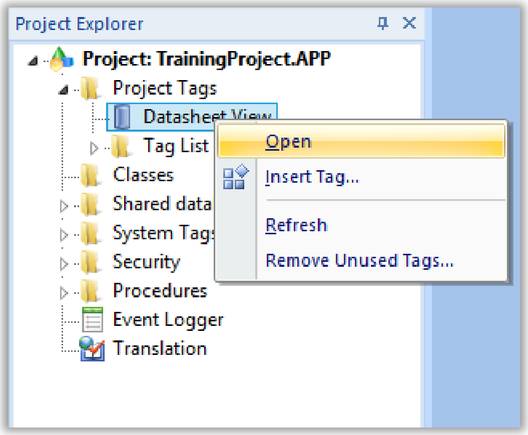


Description: Integer Value 1

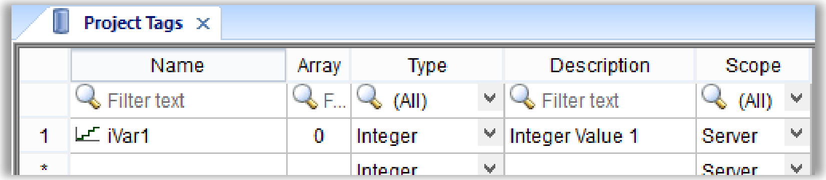
4. Click **OK** to close the New Tag dialog.

Create multiple tags

1. On the Global tab of the Project Explorer, expand the Project Tags folder, right-click Datasheet View, and select **Open** on the shortcut menu.

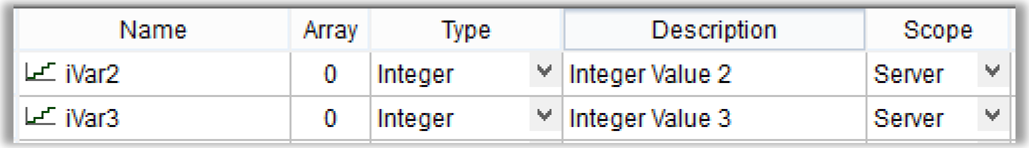


The Project Tags worksheet is opened in the Workspace.


Project Tags

	Name	Array	Type	Description	Scope
	Filter text	F...	(All)	Filter text	(All)
1	iVar1	0	Integer	Integer Value 1	Server
*			Integer		Server

2. Create two more tags called iVar2 and iVar3 by filling out the tags database worksheet as shown below.



Name	Array	Type	Description	Scope
iVar2	0	Integer	Integer Value 2	Server
iVar3	0	Integer	Integer Value 3	Server

3. Close the tags database worksheet.

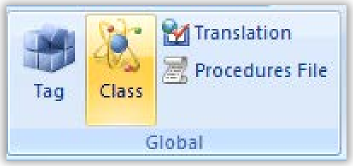
Create classes and class tag arrays

In this training class, you will create a project to monitor and control a tank farm as follows:

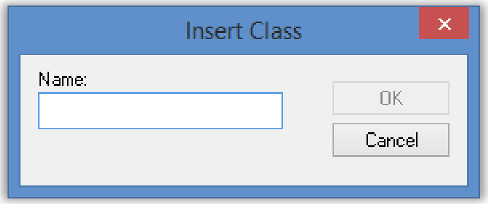
- Each storage tank has values for pressure, temperature, level, and contents
- Each tank has an empty valve and fill valve. Each valve has a command bit and a state bit.

Create class-type tags for your tanks and valves.

1. On the Insert tab of the Ribbon, in the Global group, click **Class**.

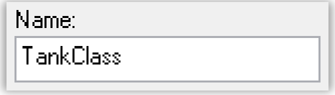


The Insert Class dialog is opened.



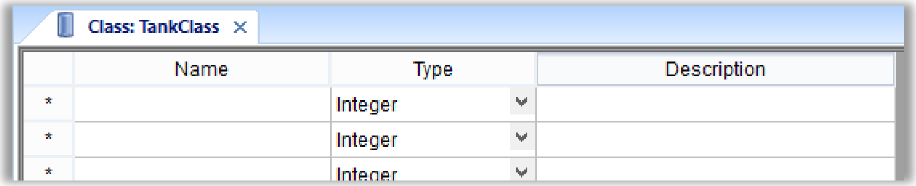
Pro Tip: You can also create a new class by right-clicking on the Classes folder on the Global tab of the Project Explorer and selecting Insert Class from the shortcut menu.

2. In the **Name** field, enter `TankClass`.



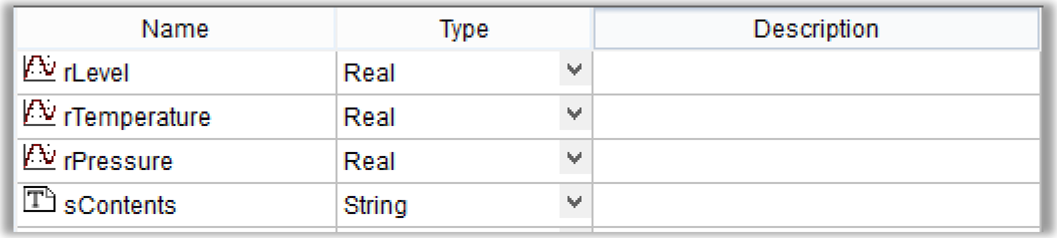
Name: TankClass

3. Click **OK**. The Insert Class dialog is closed and the Class: TankClass worksheet is opened in the Workspace.



	Name	Type	Description
*		Integer	
*		Integer	
*		Integer	

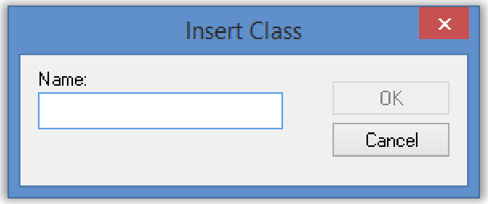
4. Configure the class as shown below.



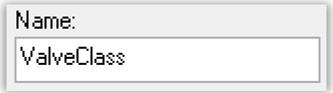
Name	Type	Description
rLevel	Real	
rTemperature	Real	
rPressure	Real	
sContents	String	

5. Close the Class: TankClass worksheet.

6. On the Insert tab of the Ribbon, in the Global group, click **Class**. The Insert Class dialog is opened.

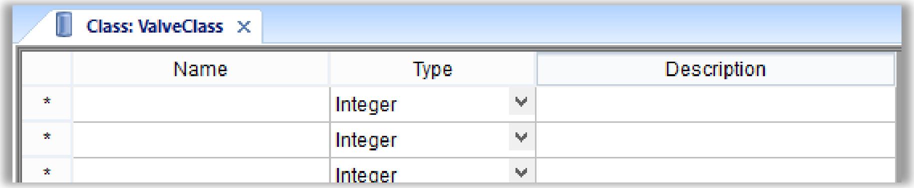


7. In the **Name** field, enter `ValveClass`.



Name:
ValveClass

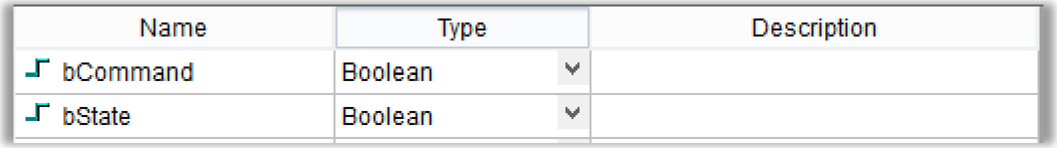
8. Click **OK**. The Insert Class dialog is closed and the Class: ValveClass worksheet is opened in the Workspace.



Class: ValveClass

	Name	Type	Description
*		Integer	
*		Integer	
*		Integer	

9. Configure the class as shown below.

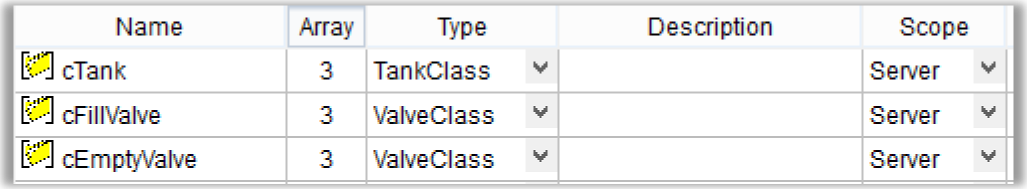


Name	Type	Description
bCommand	Boolean	
bState	Boolean	

10. Close the Class: ValveClass worksheet.

11. On the Global tab of the Project Explorer, expand the Project Tags folder, right-click Datasheet View, and select Open on the shortcut menu. The Project Tags worksheet is opened in the Workspace.

12. Create the tags shown below.



Name	Array	Type	Description	Scope
cTank	3	TankClass		Server
cFillValve	3	ValveClass		Server
cEmptyValve	3	ValveClass		Server

13. Close the Project Tags worksheet.

Note: The far right of the status bar shows the current count of licensed tags used in your project. This may be different than the number of rows in your Project Tags worksheet. The licensed tag count is based on the number of unique stored values derived from the number of tags on the project worksheet and the array and class size of each tag.

SECTION 3 - DOWNLOAD TO AN INDUSTRIAL COMPUTER

This section explains how to download the BLUE Open Studio software and your project to an industrial computer using Remote Management,

- Connect to the runtime device using the Remote Management tool
- Install the runtime on the industrial computer
- License the installed software
- Download the project to the target
- Run the project.

Concept

BLUE Open Studio is designed to run on devices running Windows Embedded. The Remote Management interface allows you to perform installation and maintenance from your development device.

Overview

Installing the BLUE Open Studio on a target system is a two-part procedure. First, you will copy the Remote Agent utility (CEServer.exe) to the target system and then run it. Remote Agent allows you to connect to the runtime device from the Remote Management utility within the BLUE Open Studio development environment to the target system. Once connected, you will then install the rest of the runtime software through the Remote Management.

Before you begin, you must have already installed the full BLUE Open Studio development software on your computer, because the redistributable runtime software is included in the BLUE Open Studio program folder and will be installed from the Remote Management feature.

To install and run BLUE Open Studio, you must have:

- A Windows Embedded-compatible computer or device (a.k.a. the target system) using an operating system that is currently supported by Microsoft.
- 100 MB free hard drive space or non-volatile memory;
- An Ethernet or Wi-Fi network adapter, for TCP/IP networking.

The following items are optional but recommended:

- At least one USB port; and
- Serial COM ports to be used for direct communication with PLCs and other devices.

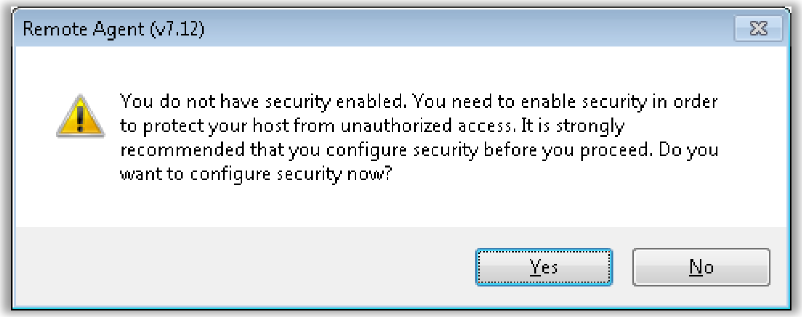
LAB 4 - DOWNLOAD TO A WES7 COMPUTER

In this lab, you will install the BLUE Open Studio runtime and your project on a target system running Windows Embedded Standard 7.

Launch the Remote Agent

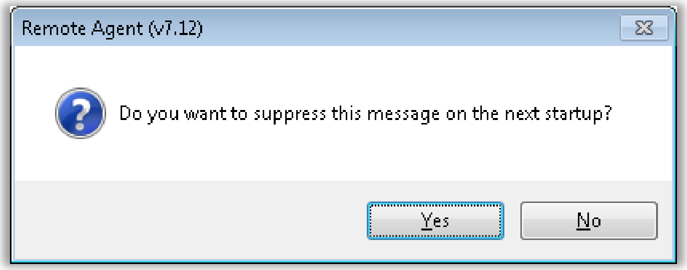
1. Open the Windows File Explorer and navigate to **C:\STUDIO**.

2. Run **CEServer.exe** on the device. The first time you run the utility you will receive a message requesting to enable security.



You do not have security enabled. You need to enable security in order to protect your host from unauthorized access. It is strongly recommended that you configure security before you proceed. Do you want to configure security now?

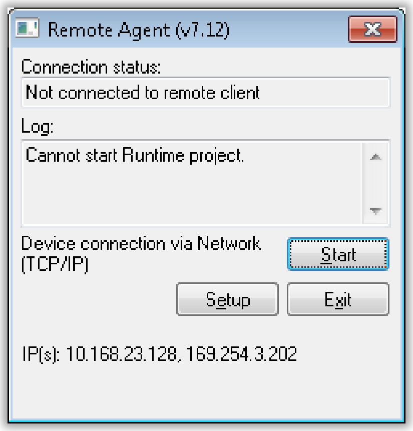
3. Select **No**. Next a dialog will ask if you wish to display this message in the future.



Do you want to suppress this message on the next startup?

4. Select **Yes.**

5. The Remote Agent window is opened.



Connection status:
Not connected to remote client

Log:
Cannot start Runtime project.

Device connection via Network (TCP/IP)

IP(s): 10.168.23.128, 169.254.3.202

Do not be concerned with the message that says `Cannot start Runtime project`. You get this message because you haven't copied your project to the industrial computer, yet.

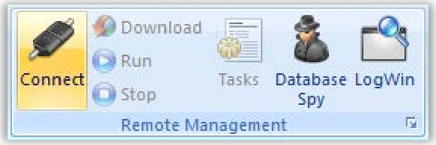
Note: If CEServer.exe is not already installed on the machine then you can copy it from the machine where BLUE Open Studio is installed. It is in the Redist\BLUE Open Studio\Bin folder of the BLUE Open Studio installation folder (typically C:\Program Files (x86)\BLUE Open Studio v8.0\Redist\EmbeddedView\).

Connecting to the Remote Agent

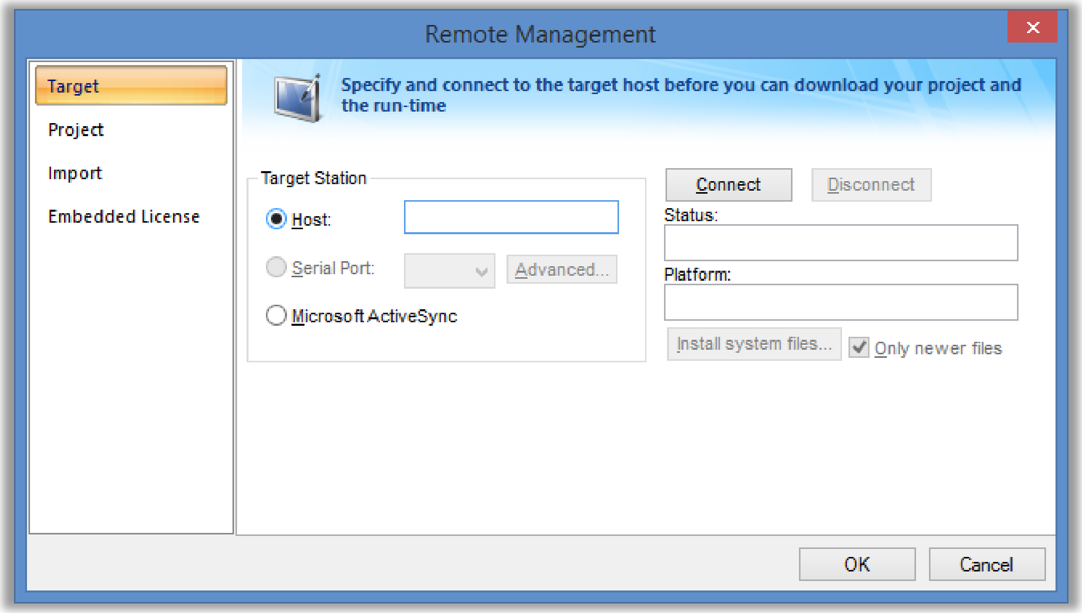
1. Make sure that you leave the Remote Agent running on the target device to allow for the Remote Management tool in the Development device to connect.

2. Run the BLUE Open Studio development environment on your computer.

3. On the Home tab of the Ribbon, click Connect under Remote Management.

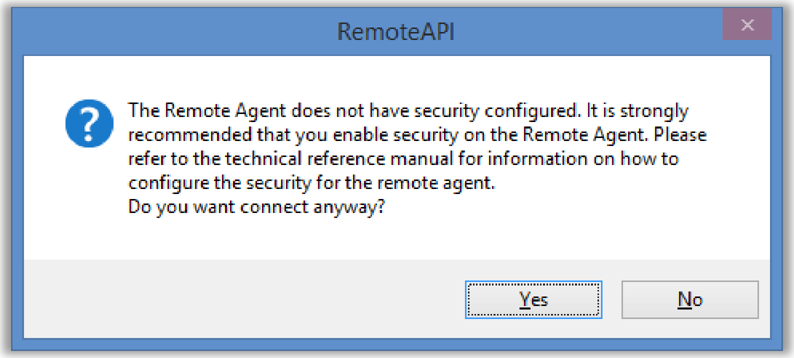


4. The Remote Management tool is opened.



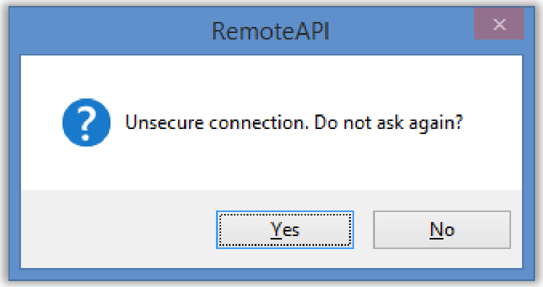
5. Enter the IP address for the target system in the **Host** field.

6. Click **Connect.** If security has not been enabled on the Remote Agent, then you will see this message.



The Remote Agent does not have security configured. It is strongly recommended that you enable security on the Remote Agent. Please refer to the technical reference manual for information on how to configure the security for the remote agent.
Do you want connect anyway?

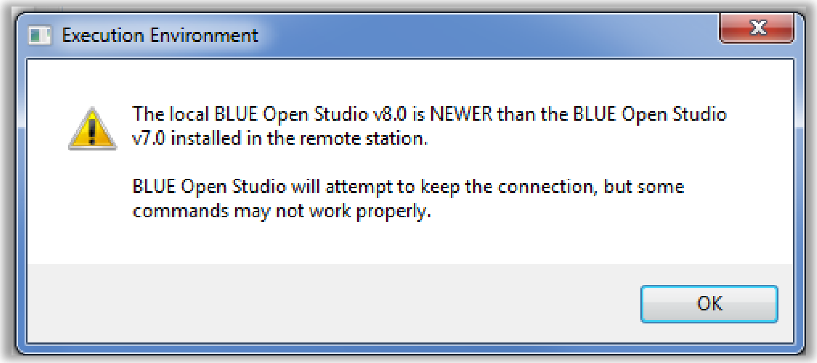
Click Yes. You will see this.



Unsecure connection. Do not ask again?

Click Yes.

7. The first time you connect to a new Remote Agent, you will receive this message.

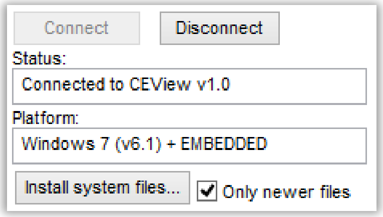


The local BLUE Open Studio v8.0 is NEWER than the BLUE Open Studio v7.0 installed in the remote station.

BLUE Open Studio will attempt to keep the connection, but some commands may not work properly.

This indicates that the Remote Management tool did not locate installed system files on the industrial computer. Select **OK.**

8. If you are successfully connected, then the connection status is shown in the Status box and the target system's specifications are shown in the Platform box.



Status:
Connected to CEView v1.0

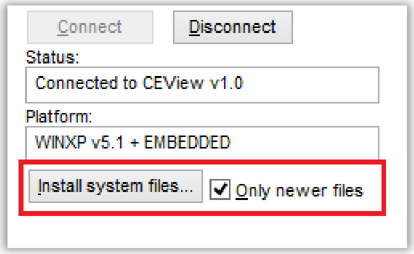
Platform:
Windows 7 (v6.1) + EMBEDDED

☑ Only newer files

Note: If you are not connecting properly, then check both the connection settings and the physical connections. In particular, if you have selected Network IP as the connection type, then check to make sure that the target station firewall rules have been configured to allow remote access to CEServer.exe according to the instructions in the appendix.

Install System Files

1. In Remote Management on the **Target** tab, click **Install system files**.

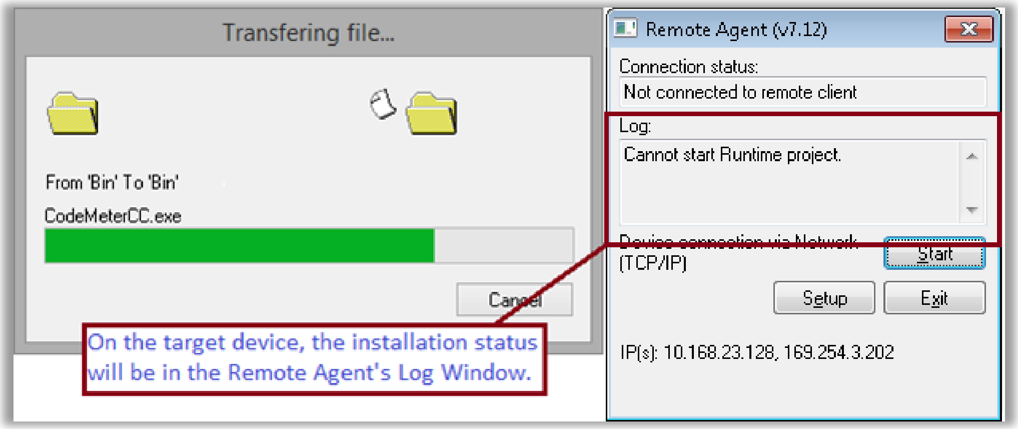


Status:
Connected to CEView v1.0

Platform:
WINXP v5.1 + EMBEDDED

☑ Only newer files

2. The Remote Management Tool will then upload and install the BLUE Open Studio runtime software to the industrial computer. You can monitor the progress on both the Development and the Industrial computer.



From 'Bin' To 'Bin'
CodeMeterCC.exe

Connection status:
Not connected to remote client

Log:
Cannot start Runtime project.

Device connection via Network (TCP/IP)

IP(s): 10.168.23.128, 169.254.3.202

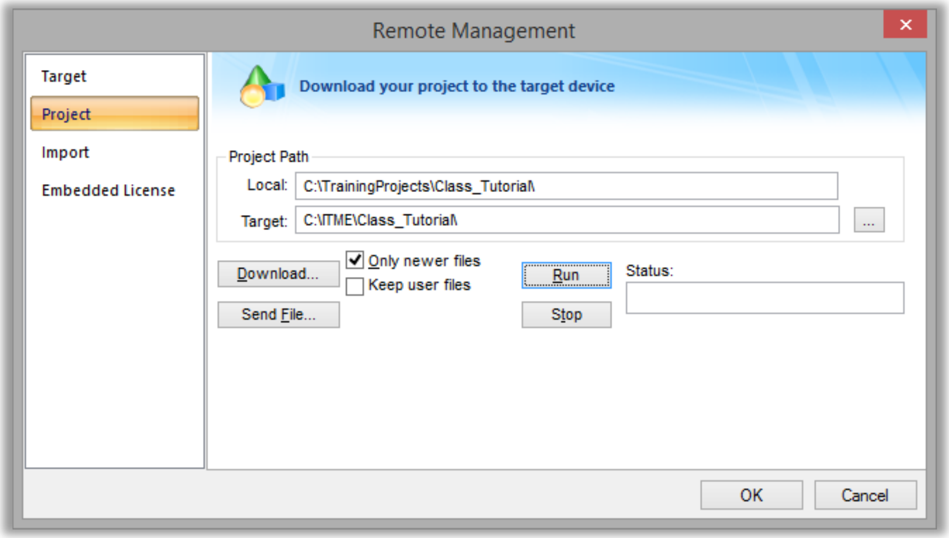
On the target device, the installation status will be in the Remote Agent's Log Window.

When the installation is finished, the target system's updated status is displayed in the Status box.

Note: When a new version of *BLUE Open Studio* is available, the upgrade should be applied to the development server. After upgrading the development server, connect to the runtime using the Remote Management as above and select **Install System Files.** The upgrade will automatically be applied.

Download your project

1. Select the **Project** tab on the Remote Management tool.



Remote Management

Local: C:\TrainingProjects\Class_Tutorial\

Target: C:\ITME\Class_Tutorial\

☑ Only newer files

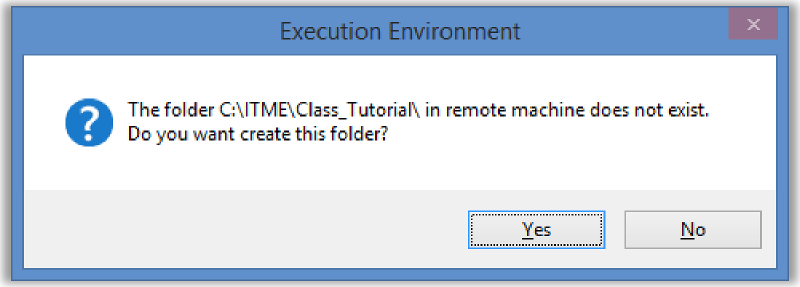
□ Keep user files

Status:

In this window, the fields/options are defined as follows:

- **Local:** The project folder location on the development PC.
- **Target:** The location for the project on the target runtime device. The default location is a folder with the same name as your project in the folder where CEServer.exe was installed.

2. Click **Download** to download the entire project to the industrial computer. You will see a message similar the following the first time you download your project. Click **Yes**.



Execution Environment

The folder C:\ITME\Class_Tutorial\ in remote machine does not exist.
Do you want create this folder?

Note: Best practice is to always stop a running project prior to downloading changes.

Pro Tip: Once you've initially configured the Remote Management settings and connect to the target machine, then you can use Download in the Remote Management group of the Home tab of the Ribbon to send updated project files at any time without opening this dialog.

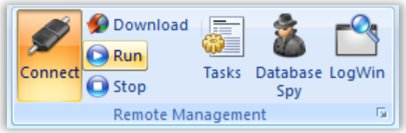
You can compress the project files to make them download more quickly over a slow network connection. To do this, select the Enable File Compression check box in the Communication tab of the Project Settings dialog.

Run or stop your project on the industrial computer

There are three ways to run or stop a project on an industrial computer. The first two require you to be connected to the industrial computer with the *Remote Management Tool.*

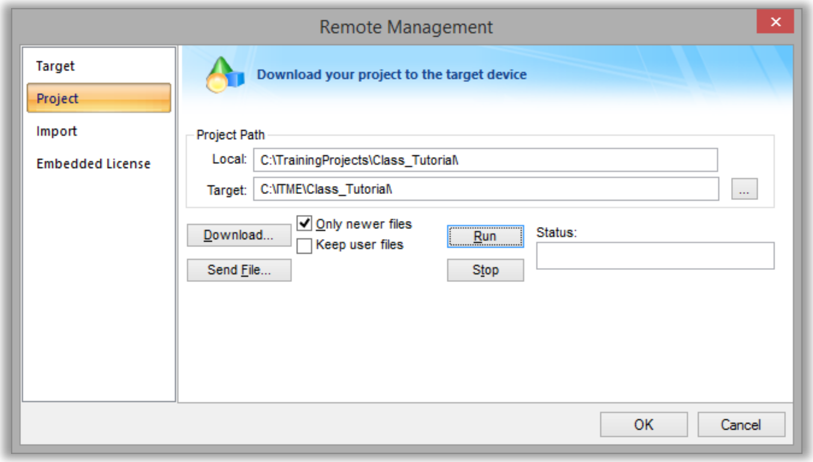
Using the Ribbon

Click **Run** or **Stop** in the Remote Management group on the Home tab of the Ribbon.



Using the Remote Management Tool

Use the **Run** and **Stop** buttons on the **Project** tab on the Remote Management tool.



Local: C:\TrainingProjects\Class_Tutorial\

Target: C:\ITME\Class_Tutorial\

☑ Only newer files

□ Keep user files

Using the Remote Agent directly

Click **Start on the Remote Agent**. Clicking **Exit** will stop both the Remote Agent as well as any running project.

Learning Services

Pro-face
by Schneider Electric

Module 4 - Communicating with External Devices

SECTION 1 - COMMUNICATION DRIVERS

This section explains using a step-by-step tutorial how to configure a Driver task and execute communication between your project and field equipment.

You will configure a Driver task using MOTCP (MODBUS protocol via TCP/IP) to perform communication between some tags and addresses.

Concept

The Driver Runtime task is responsible for exchanging data with other systems. These other systems may be other software (i.e. PC Based Control Software), an external device (such as a PLC, Robot, Remote I/O, Barcode reader), or any other device that supports a protocol that is implemented by the communication driver.

A communication driver is a DLL containing specific information about the remote equipment and implements the communication protocol. Drivers for dozens of common and not-so-common devices are installed with BLUE Open Studio.

Overview

Driver messages are pieces of structured communications sent over some physical medium, such as a serial connection, Ethernet, Wi-Fi, or even using a proprietary protocol and a specialized port. Messages are encoded with an addressing scheme that a receiving or “listening” station recognizes. Messages are constructed in layers called abstraction layers and each layer consists of a set of functionality providing a needed service to the preceding layer.

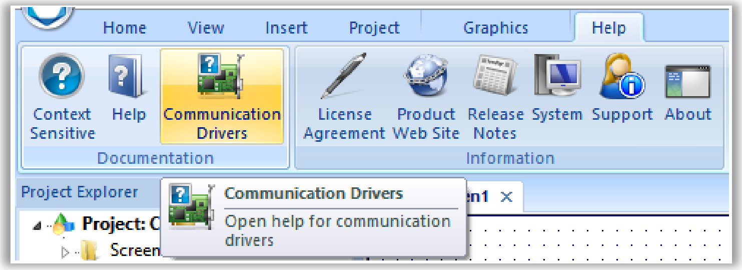
BLUE Open Studio provides two interfaces which you can use to configure the driver (associating project tags to device addresses):

Main Driver Sheet: Provides the easiest method for configuring communication between project tags and device addresses. This interface allows you to automatically group tags to provide the best performance during runtime. You cannot use this interface to control the time needed to scan a group of tags individually.

Standard Driver Sheets: Allows you to control the time needed to scan a group of tags individually.

Driver Help Files

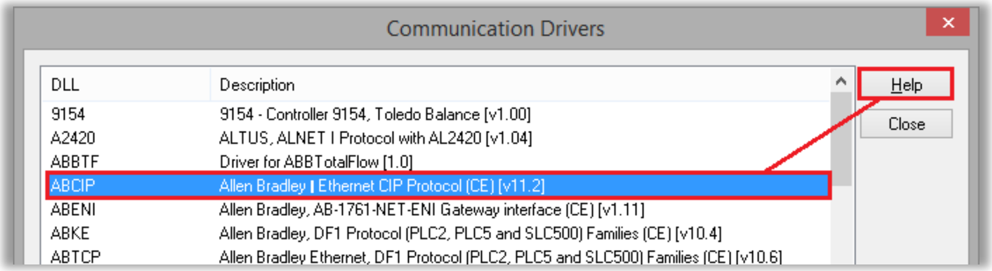
Each driver has a specific help file **(**[Driver Name].**pdf)** that provides a detailed description about the driver settings and also includes a step-by-step process for configuring that driver. This file can be easily accessed by selecting Communication Drivers in the Documentation **group** on the Help tab of the Ribbon.



Communication Drivers

Open help for communication drivers

Selecting this will open the Communication Drivers Help dialog.



Communication Drivers

DLL	Description
9154	9154 - Controller 9154, Toledo Balance [v1.00]
A2420	ALTUS, ALNET I Protocol with AL2420 [v1.04]
ABBTF	Driver for ABBTotalFlow [1.0]
ABCIP	Allen Bradley I Ethernet CIP Protocol (CE) [v11.2]
ABENI	Allen Bradley, AB-1761-NET-ENI Gateway interface (CE) [v1.11]
ABKE	Allen Bradley, DF1 Protocol (PLC2, PLC5 and SLC500) Families (CE) [v10.4]
ABTCP	Allen Bradley Ethernet, DF1 Protocol (PLC2, PLC5 and SLC500) Families (CE) [v10.6]

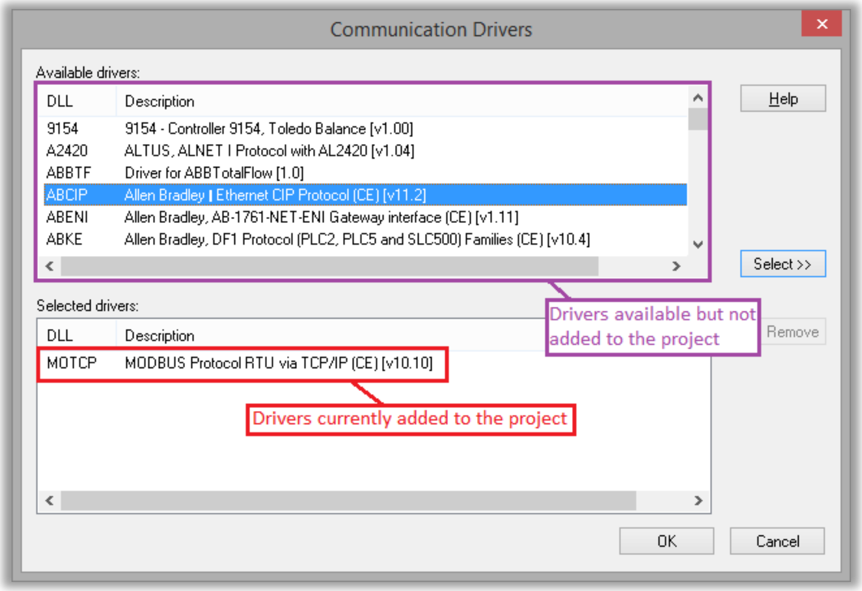
Select the driver help document you wish to open and select **Help**. The file will open in your device's default pdf reader.

Adding a Driver

Use one of the following methods to add or remove a driver:

- On the Insert tab of the Ribbon under Communication, click **Add/Remove Driver**; or
- Right-click the Drivers folder in the Comm tab of the Project Explorer and click **Add/Remove drivers**.

Both methods open a Communication Drivers dialog, which displays a list of available drivers.



Communication Drivers

Available drivers:

DLL	Description
9154	9154 - Controller 9154, Toledo Balance [v1.00]
A2420	ALTUS, ALNET I Protocol with AL2420 [v1.04]
ABBTF	Driver for ABBTotalFlow [1.0]
ABCIP	Allen Bradley I Ethernet CIP Protocol (CE) [v11.2]
ABENI	Allen Bradley, AB-1761-NET-ENI Gateway interface (CE) [v1.11]
ABKE	Allen Bradley, DF1 Protocol (PLC2, PLC5 and SLC500) Families (CE) [v10.4]

Selected drivers:

DLL	Description
MOTCP	MODBUS Protocol RTU via TCP/IP (CE) [v10.10]

Drivers available but not added to the project

Drivers currently added to the project

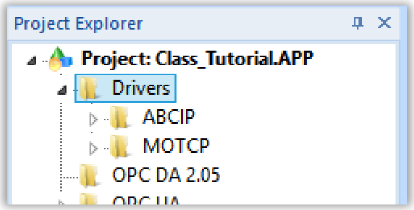
To add a driver to the current project, select the desired driver under Available drivers: and click **Select >>**. To remove a driver from the project, highlight the driver listed under Selected Drivers: and click the **>> Remove** button.

Pro-Tip: You can also access the Driver's help file from this dialog by selecting the desired driver under Available drivers and clicking the **Help** button.

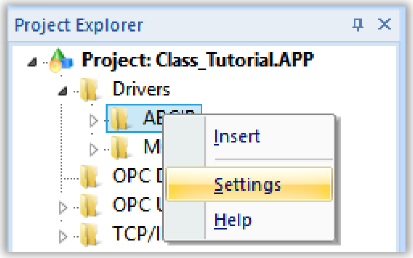
Driver Settings

The Driver Settings dialog allows the user to adjust the settings that will be used by all communication worksheets configured for each communication driver. Note that the settings may likely be different to that driver and protocol. Therefore, it is highly recommended to open and follow the driver's help file mentioned above.

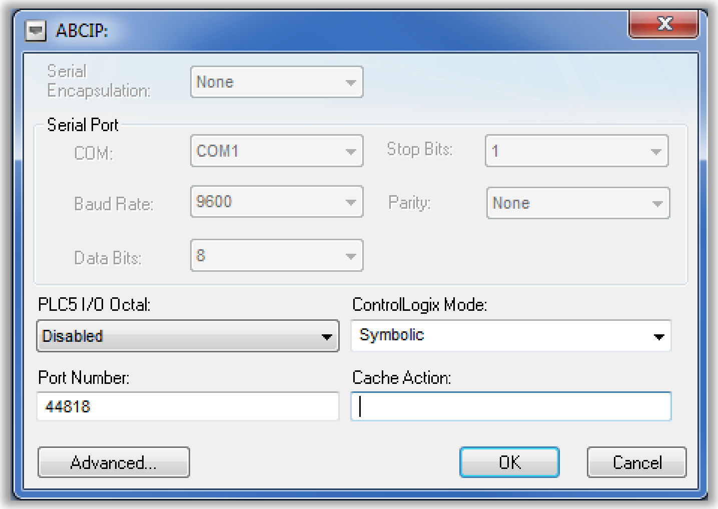
To access the Driver Settings, open the Comm tab of the Project Explorer and expand the **Drivers** folder to display all currently added drivers.



Right-click on the driver you want to configure and select **Settings**.



This will open the **Driver Settings** dialog.



ABCIP:

Serial Encapsulation: None

Serial Port

COM: COM1

Stop Bits: 1

Baud Rate: 9600

Parity: None

Data Bits: 8

PLC5 I/O Octal: Disabled

ControlLogix Mode: Symbolic

Port Number: 44818

Cache Action:

Note: The settings dialog will be different depending on which driver is selected. Refer to the driver-specific help document for information about configuring these settings.

Main Driver Sheet

The Main Driver Sheet or MDS is a single worksheet available for each communication driver. The main advantage of the MDS is to provide a user-friendly interface to configure all communication addresses of the application. The BLUE Open Studio development environment automatically inserts the Main Driver Sheet into the driver folder as soon as you add the driver to your project.

Note: The **Main Driver Sheet** is not available for all drivers.

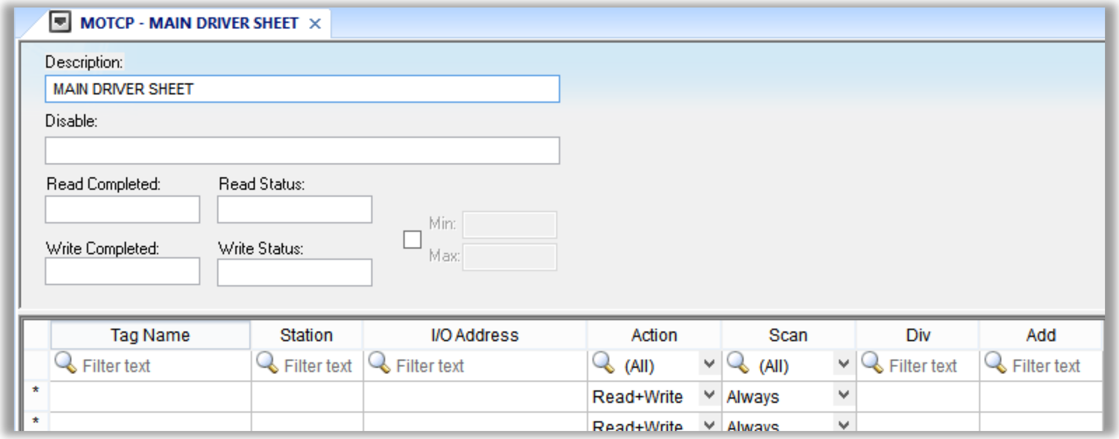
Using the MDS the user simply needs to associate one tag to each station/address and configure one of the following actions for each tag:

Read

Write

Read+Write

When the driver runs, the MDS creates virtual groups for reading commands, the size of which will not exceed the maximum block size supported by the protocol. The **READ** command is executed using the READ BLOCK Method (reads a block of addresses in each message), triggered whenever the System Tag Blink Slow toggles (600 ms by default). The **WRITE** command is executed using the WRITE ITEM Method (writes only one tag value to one address in each message), triggered when the respective tag changes value.



Description: MAIN DRIVER SHEET

Disable:

Read Completed:

Read Status:

Write Completed:

Write Status:

□ Min:

Max:

	Tag Name	Station	I/O Address	Action	Scan	Div	Add
	Filter text	Filter text	Filter text	(All)	(All)	Filter text	Filter text
*				Read+Write	Always		
*				Read+Write	Always		

The Main Driver Sheet worksheet is divided into two areas:

- **Header area (top section)**, contains parameters that affect the all tags configured in the Body area of this worksheet; and
- **Body area (bottom section)**, where you define the relationship between tags in the project and their field equipment address.

Use the **Header area** parameters as follows:

- **Description field**: Type a description of the Main Driver Sheet for documentation purposes.
- **Disable field**: Type a tag or an expression to enable and disable the communication of each Main Driver Sheet on the fly.

    - Type a value (or expression result) that is greater than zero to disable the Main Driver Sheet.
    - Type a zero (or leave this field blank) to enable the Main Driver Sheet.
- **Read Completed field:** Type in a tag and the communication driver toggles the tag when it completes a read command.
- **Read Status field:** An integer tag updated with the returned status of the last read command from the driver (See Troubleshooting in the driver's help file for values)
- **Write Completed field:** Type in a tag and the communication driver toggles the tag when it completes a write command.
- **Write Status field:** An integer tag updated with the returned status of the last write command from the driver (See Troubleshooting in the driver's help file for values)
- **Min and Max checkbox:** Click (check) to specify minimum and maximum values for data from the field equipment.
- **Min and Max fields** (become active only when you enable the Min and Max checkbox): Type a range of values, which can be converted into an engineering format.
    - The project uses these fields to determine a minimum/maximum range of values for data from the field equipment. The scaling is done automatically. You must configure the engineering range using the Min and Max parameters on the Tag Properties dialog. This range affects all tags in the worksheet, except those with customized Min and Max values, as specified in the Body area of the driver sheet (Min and Max columns).

Use the **Body area** parameters as follows:

- **Tag Name field:** Type the name of a project tag to be used by the communication driver.
- **Station field:** Type the number of the equipment station within the network. The syntax in this field varies with each communication driver. Refer to the appropriate driver's documentation for further information.
- **I/O Address field:** Type the address of the field equipment related to the project tag. The syntax in this field varies with each communication driver. Refer to the appropriate driver's documentation for further information.
- **Action field:** Specify the communication direction, using one of the following options:
    - **Read** (the project reads the address from the field device and updates the Tag value on a 600 ms trigger.)
    - **Write** (the project writes the tag value to the field device when the tag value changes.)
    - **Read+Write** (Combines the procedures of both the Read and Write parameters.)
- **Scan field:** Specify the condition under which the tag value is read from the remote device or server and then updated in the project database, using one of the following options:

  - ○ **Always** means the tag is read and updated during every scan of the communication worksheet, regardless of whether the tag is used in any other project screens, scripts, or worksheets.
    - ▪ This option is recommended for tags that must be continuously monitored in the background, such as tags that trigger alarms, tags used in recipes, tags that are recorded in the historical database, and so on.
  - ○ **Screen** means the tag is read and updated only if it is being used in at least one open project screen, either locally or on another client station.
    - ▪ This option is recommended for tags that are used in screen objects, because the project may not need to update tags that are not being visualized anywhere. Selecting this option can improve project performance.
  - ○ **Auto** means the project will automatically choose either Always or Screen, based on where the tag is used in your project. If the tag is only used in a screen object on a project screen, then the scan will default to Screen. But if the tag is configured in any other interface (e.g., Script, Math, Alarm, Trend, Recipe, Report, Scheduler), then the scan will switch to Always and remain there until the project is stopped.
    - ▪ If you are not sure of which option to select, select **Always**. This will guarantee the tag is read and updated.

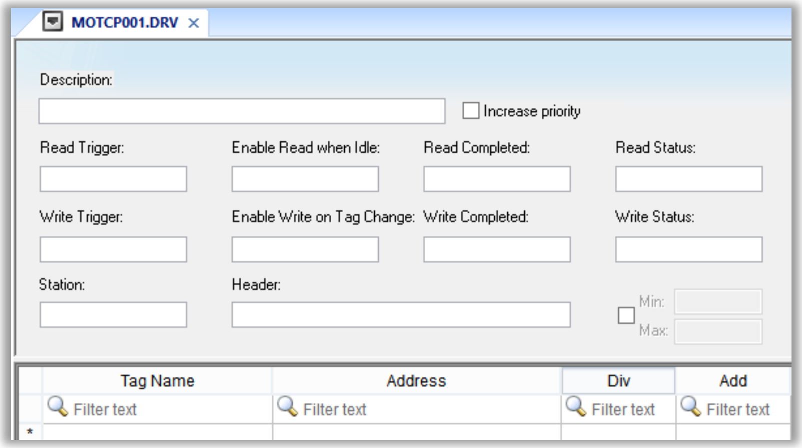
- **Div field:** Specify the division constant when scale adjustment is required. This value is a division factor in a read operation and a multiplication factor in a write operation. Do not use this field if you are already using Min or Max in the configuration body.
- **Add field:** Specify the addition constant when scale adjustment is required. This value is an addition factor in a read operation and a subtraction factor in a write operation. Do not use this field if you are already using Min or Max in the configuration body.

Note: The Main Driver Sheet can have up to 32767 rows. If you need to configure more than 32767 communication addresses, then either configure Standard Driver Sheets or create additional instances of the driver.

Standard Driver Sheets

In addition to the unique **Main Driver Sheet** that is available for each driver, you can create several Standard Driver Sheets for each driver. The Standard Driver Sheets provide additional fields which you can use to control communication.

To open a Standard Driver Sheet, right-click on a driver in the Project Explorer and select **Insert**.



Note: You can have a total of 9,999 Standard Driver Sheets for all drivers in your project.

The Standard Driver Sheet dialog is divided into two areas:

Header area (top section): contains parameters that affect the all tags configured in the Body area of this worksheet

Body area (bottom section): defines the relationship between tags in the project and their field equipment address.

Use the **Header area** parameters as follows:

- **Description:** Type a description of the Standard Driver Sheet for documentation purposes.
- **Increase Priority checkbox**: Click (check) to keep the reading and writing commands for this sheet on the top of the communication queue whenever they are triggered.
- **Read Trigger:** Type a tag that triggers the project to read the worksheet automatically when you change this tag's value.
- **Enable Read when Idle:** Type a tag or constant value. Use a tag (or constant) value greater than zero, to enable reading from the equipment.
- **Read Completed:** Type in a tag and the communication driver toggles the tag when it completes a read command.
- **Read Status**: An integer tag updated with the returned status of the last read command from the driver (See Troubleshooting in the driver’s help file for values).
- **Write Trigger**: Type a tag value to activate a group reading. Whenever you change this tag value, the program writes an equipment worksheet.
- **Enable Write on Tag Change**: Type a tag or constant value (not zero) to enable the communication driver to check the worksheet continuously for changes in the tag value. If a change occurs, the project writes this value to an address in the field equipment.
- **Write Completed:** Type in a tag and the communication driver toggles the tag in this field when a write command completes.

- **Write Status**: An integer tag updated with the returned status of the last write command from the driver (See Troubleshooting in the driver's help file for values)
- **Station:** Type the equipment station number within the network. The syntax in this field varies with each communication driver. Refer to the appropriate driver's documentation for further information.
- **Header**: Specify the data type and/or initial address to be read or written in the equipment. The syntax in this field varies with each communication driver. Refer to the appropriate driver's documentation for further information.

Pro-Tip: You can dynamically change the Station and Header fields in runtime by using a string tag within curly braces (e.g. {TagName}) in these fields

- **Min and Max checkbox**: Check this box to specify the minimum and maximum values for field equipment data.
- **Min and Max fields (**become active only when you enable the Min and Max checkbox): Type a range of values to be converted into an engineering format. These fields determine the minimum and maximum range of values. These values affect all tags in the worksheet.

Use the **Body** area parameters as follows:

- **Tag Name:** Type a tag name for the communication driver to use.
- **Address:** Type a field equipment address (or address offset) related to the project tag. The syntax in this field varies with each communication driver. Refer to the appropriate driver's documentation for further information.
- **Div:** Specify a division constant to use when scale adjustment is required. The project uses this value as a division factor in a read operation and a multiplication factor in a write operation. Do not use this field if you are already using Min or Max in the configuration body.
- **Add:** Specify an addition constant to use when scale adjustment is required. The project uses this value as an addition factor in a read operation and a subtraction factor in a write operation. Do not use this field if you are already using Min or Max in the configuration body.

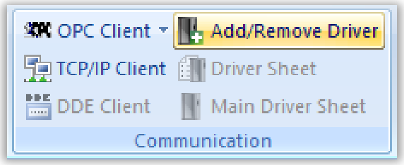
Note: Each Standard Driver Sheet can have up to 4096 rows. However, the Read Trigger, Enable Read When Idle, and Write Trigger commands attempt to communicate the entire block of addresses that is configured in the sheet, so if the block of addresses is larger than the maximum block size that is supported by the driver protocol, then you will receive a communication error (e.g., "invalid block size") during run time. Therefore, the maximum block size imposes a practical limit on the number of rows in the sheet, and that limit varies by driver. For more information, please refer to the driver documentation for your selected driver.

LAB 5 - CONFIGURE A MODBUS TCP DRIVER

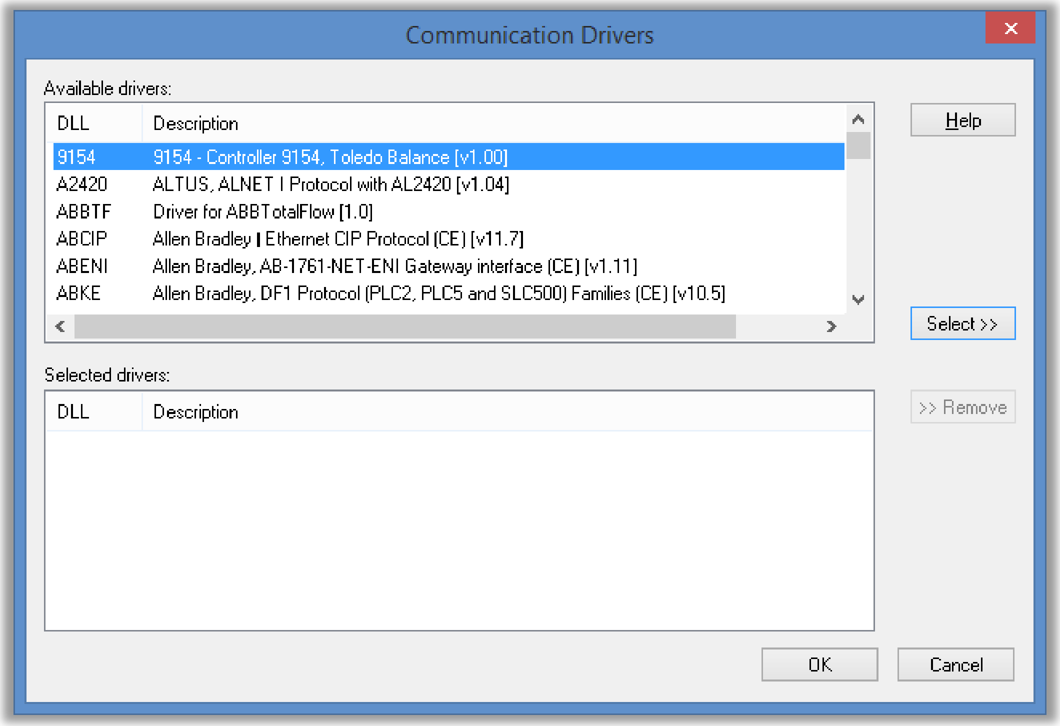
In this lab you will add a MOTCP communication driver to your project and configure it to communicate with a PLC as directed by your instructor.

Add the MOTCP driver

1. On the Insert tab of the Ribbon, in the Communication group, click **Add/Remove Driver**.



The Communication Drivers dialog is opened.



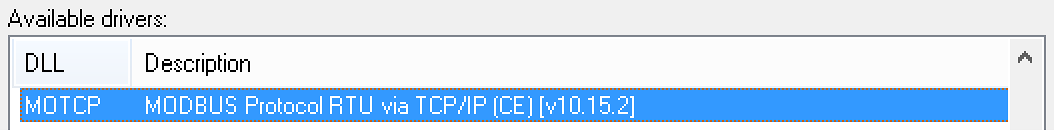
Available drivers:

DLL	Description
9154	9154 - Controller 9154, Toledo Balance [v1.00]
A2420	ALTUS, ALNET I Protocol with AL2420 [v1.04]
ABBTF	Driver for ABBTotalFlow [1.0]
ABCIP	Allen Bradley I Ethernet CIP Protocol (CE) [v11.7]
ABENI	Allen Bradley, AB-1761-NET-ENI Gateway interface (CE) [v1.11]
ABKE	Allen Bradley, DF1 Protocol (PLC2, PLC5 and SLC500) Families (CE) [v10.5]

Selected drivers:

DLL	Description

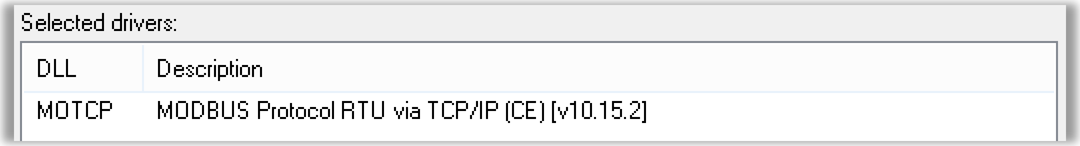
2. Scroll through the **Available drivers** list box and select the **MOTCP** driver.



Available drivers:

DLL	Description
MOTCP	MODBUS Protocol RTU via TCP/IP (CE) [v10.15.2]

3. Click **Select**. The MOTCP driver is added to the **Selected drivers** list box.



Selected drivers:

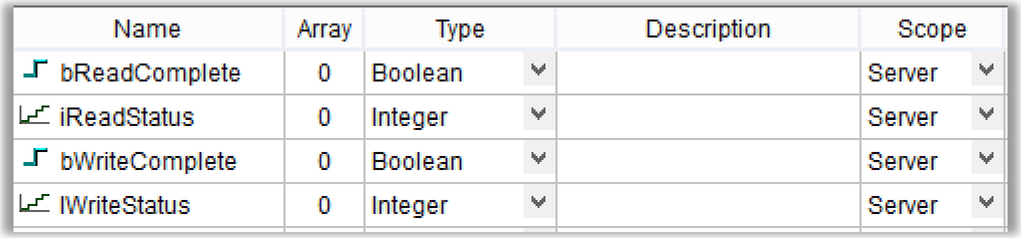
DLL	Description
MOTCP	MODBUS Protocol RTU via TCP/IP (CE) [v10.15.2]

4. Click **OK** to close the Communication Drivers dialog.

Configure the Main Driver Worksheet

1. On the Global tab of the Project Explorer, expand the Project Tags folder, right-click Datasheet View, and select Open on the shortcut menu. The Project Tags worksheet is opened in the Workspace.

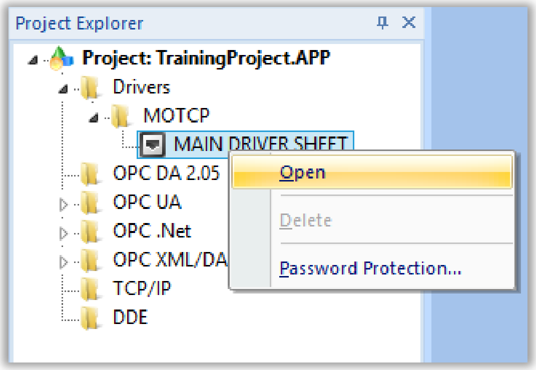
2. Create the tags shown below.



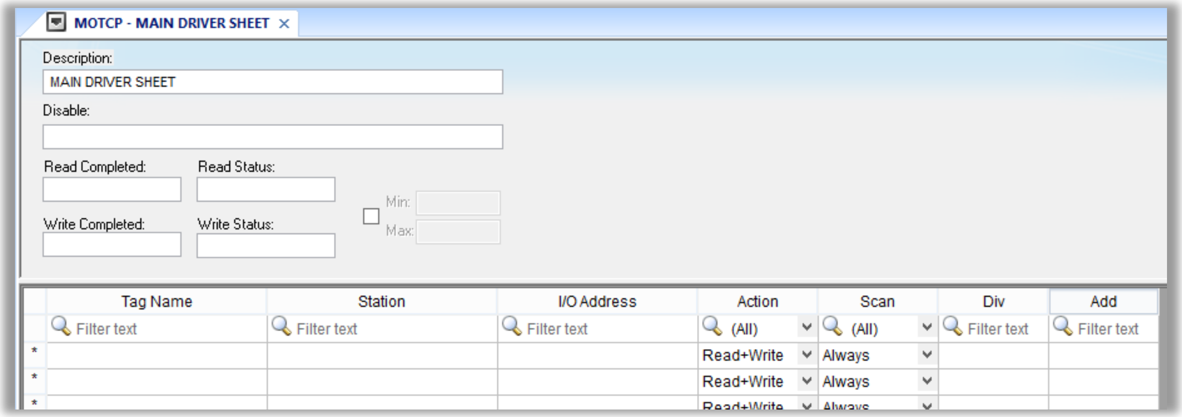
Name	Array	Type	Description	Scope
bReadComplete	0	Boolean		Server
iReadStatus	0	Integer		Server
bWriteComplete	0	Boolean		Server
IWriteStatus	0	Integer		Server

3. Close the Project Tags worksheet.

4. On the Comm tab of the Project Explorer, expand the Drivers folder, expand the MOTCP folder, right click MAIN DRIVER SHEET, and select Open from the shortcut menu.



The MOTCP - MAIN DRIVER SHEET worksheet is opened in the Workspace.



MOTCP - MAIN DRIVER SHEET

Description: MAIN DRIVER SHEET

Disable:

Read Completed:

Read Status:

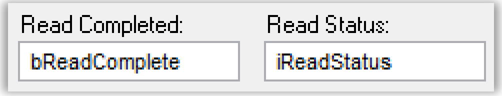
Write Completed:

Write Status:

□ Min: Max:

	Tag Name	Station	I/O Address	Action	Scan	Div	Add
	Filter text	Filter text	Filter text	(All)	(All)	Filter text	Filter text
*				Read+Write	Always		
*				Read+Write	Always		
*				Read+Write	Always		

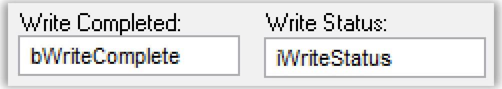
5. In the **Read Completed** and **Read Status** fields, enter `bReadComplete` and `iReadStatus`, respectively.



Read Completed: bReadComplete

Read Status: iReadStatus

6. In the **Write Completed** and **Write Status** fields, enter `bWriteComplete` and `iWriteStatus`, respectively.



Write Completed: bWriteComplete

Write Status: iWriteStatus

7. Configure the remainder of the worksheet as instructed by the instructor.

8. Save and close the MOTCP - MAIN DRIVER SHEET worksheet.

SECTION 2 - OPC

This section explains how to configure a connection with an OPC DA Server and how to execute the communication between your project and the Server.

You will configure a worksheet to perform the communication between some project tags and OPC server variables using the OPC protocol.

Concept

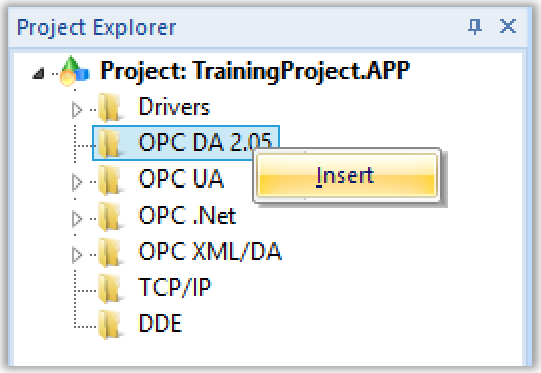
The OPC worksheet is used to communicate with any system that implements the OPC Server standard for data access. Most OPC servers use the Data Access (OPC DA 2.05) or Unified Architecture (OPC UA) standards introduced by OPC Foundation.

LAB 6 - CONFIGURE THE OPC CLIENT

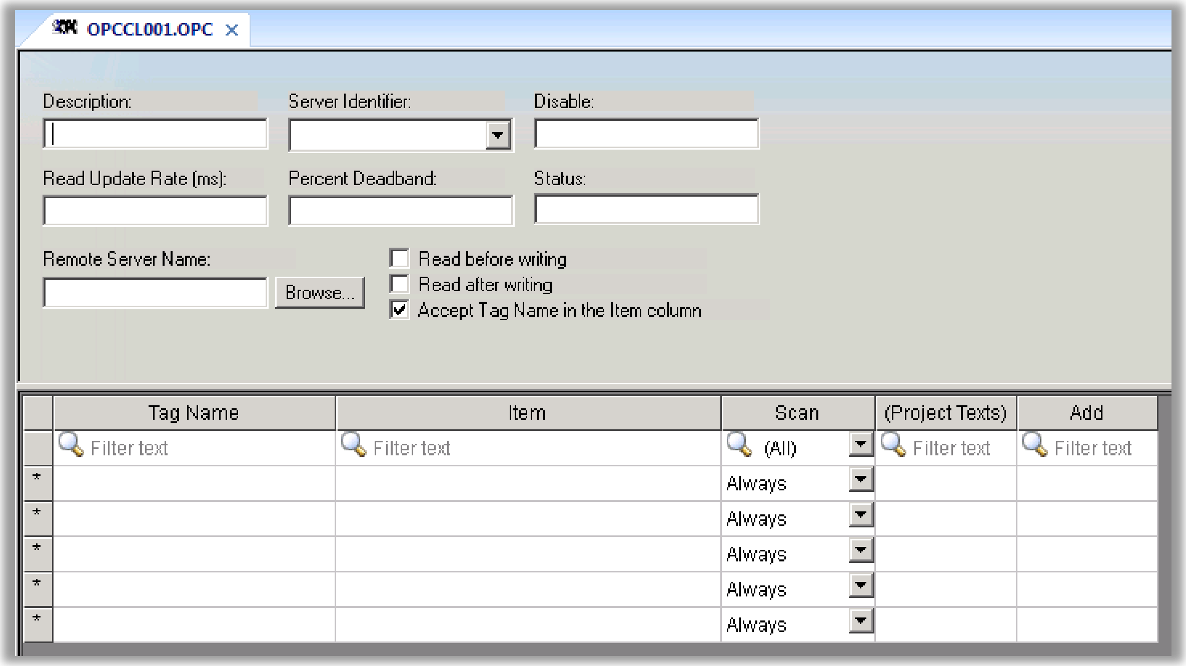
In this lab you will configure the OPC DA Client to communicate with an OPC server as directed by your instructor.

Insert an OPC DA 2.05 client worksheet

1. On the COMM tab of the Project Explorer, right-click OPC DA 2.05, and select **Insert** from the shortcut menu,



A new OPC DA 2.05 client worksheet is opened in the Workspace.



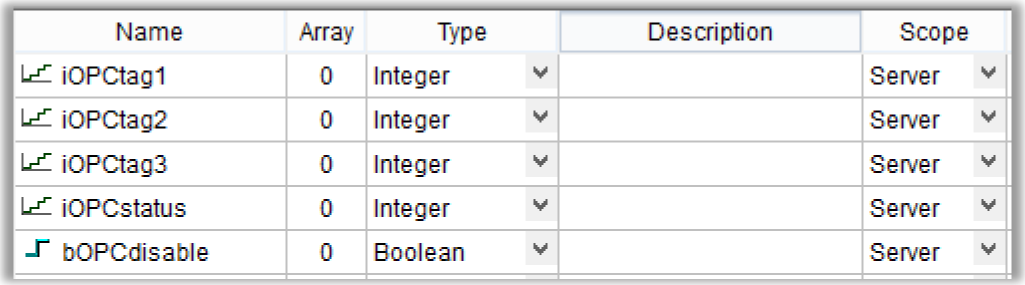
Description:
Server Identifier:
Disable:
Read Update Rate (ms):
Percent Deadband:
Status:
Remote Server Name:
☐ Read before writing
☐ Read after writing
☑ Accept Tag Name in the Item column

	Tag Name	Item	Scan	(Project Texts)	Add
	Filter text	Filter text	(All)	Filter text	Filter text
*			Always		
*			Always		
*			Always		
*			Always		
*			Always		

Configure the OPC DA 2.05 client worksheet

1. On the Global tab of the Project Explorer, expand the Project Tags folder, right-click Datasheet View, and select Open on the shortcut menu. The Project Tags worksheet is opened in the Workspace.

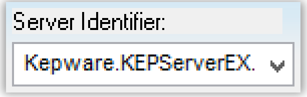
2. Create the tags shown below.



Name	Array	Type	Description	Scope
iOPCtag1	0	Integer		Server
iOPCtag2	0	Integer		Server
iOPCtag3	0	Integer		Server
iOPCstatus	0	Integer		Server
bOPCdisable	0	Boolean		Server

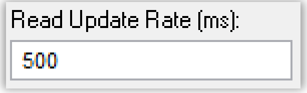
3. Close the Project Tags worksheet.

4. In the Server Identifier field, select **Kepware.KEPServerEX.V5** or other as specified by your instructor.



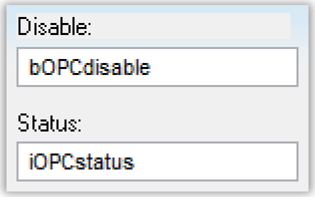
Server Identifier: Kepware.KEPServerEX.

5. In the **Read Update Rate (ms)** field, enter 500.



Read Update Rate (ms): 500

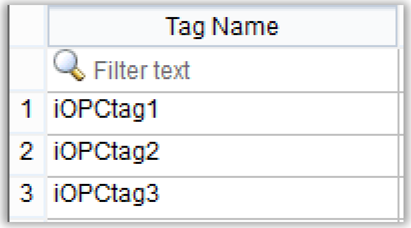
6. In the **Disable** and **Status** fields, enter bOPCdisbale and iOPCstatus, respectively.



Disable: bOPCdisable

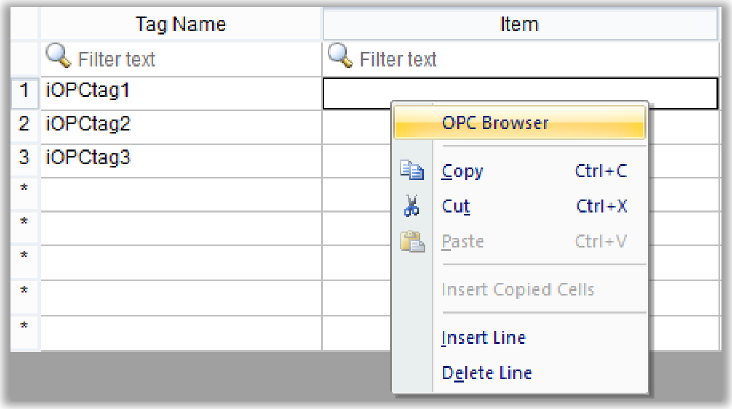
Status: iOPCstatus

7. In the **Tag Name** column, enter the tags as shown below.



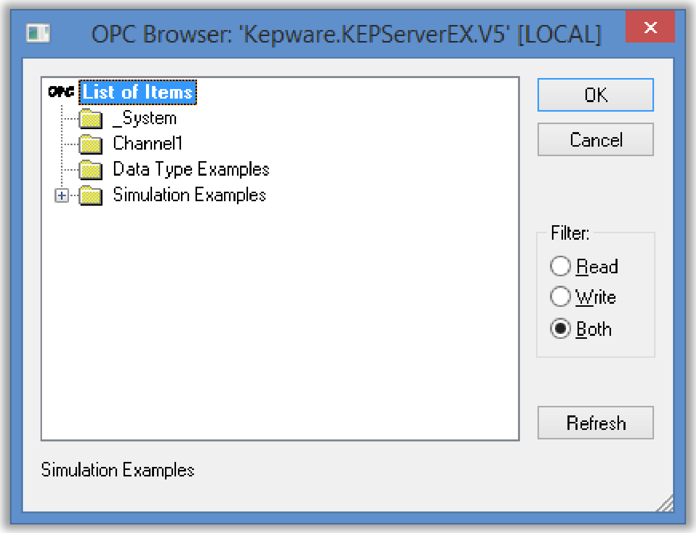
	Tag Name
	Filter text
1	iOPCtag1
2	iOPCtag2
3	iOPCtag3

8. In the **Item** column, on the row corresponding to the tag called iOPCtag1, right-click the mouse and select **OPC Browser**.



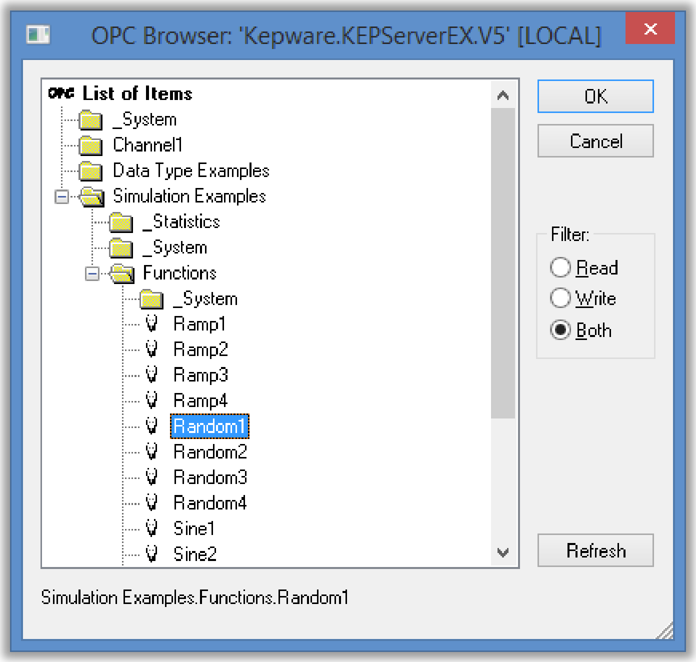
	Tag Name	Item
	Filter text	Filter text
1	iOPCtag1	
2	iOPCtag2	
3	iOPCtag3	
*		
*		
*		
*		
*		

The OPC Browser dialog is opened.



Filter:
○ Read
○ Write
● Both

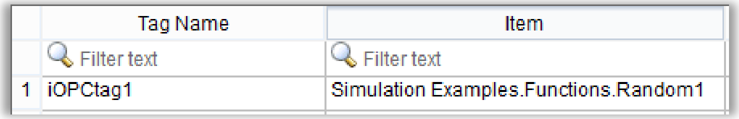
9. Expand the List of Items by double-clicking the folder called **Simulation Examples**, and then double-clicking the folder called **Functions**. Select the item called **Random1** as shown below.



Filter:
○ Read
○ Write
● Both

Simulation Examples.Functions.Random1

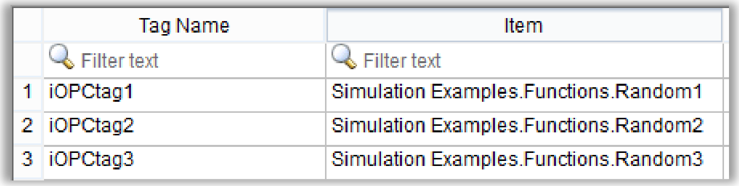
Click **OK**. The selected item is added to the OPC client worksheet.



	Tag Name	Item
	Filter text	Filter text
1	iOPCtag1	Simulation Examples.Functions.Random1

Pro Tip: If you know the full item name of the item you want to add, then you can type the name into the Item column instead of using the OPC Browser.

10. Repeat the steps 8 and 9 above for the tags called iOPCtag2 and iOPCtag3 and the items called Random2 and Random3, respectively.



	Tag Name	Item
	Filter text	Filter text
1	iOPCtag1	Simulation Examples.Functions.Random1
2	iOPCtag2	Simulation Examples.Functions.Random2
3	iOPCtag3	Simulation Examples.Functions.Random3

11. Save the worksheet as sheet number 1 and close the worksheet.

SECTION 3 - TAG INTEGRATION

This section explains how to integrate tags from a third-party control application or device into your BLUE Open Studio project using the Tag Integration tool.

Concept

Tag integration allows you to directly integrate tags from a third-party control application or device into your BLUE Open Studio project.

It does so by sharing the Tags Database of the other application or device with the tags database of your BLUE Open Studio project, allowing the project to utilize the tag within the project. It also creates consistency, since the tag names being used are the same names as in the other device.

Furthermore, Tag Integration will create the connection automatically by use of the driver task.

Currently there are seven sources that may be used by use of the Tag Integration tool. These are:

- TwinCAT (uses the TWCAT driver)
- CodeSys (uses the COSYS driver)
- RSLogix 5000 Family (uses the ABCIP driver, also supports OPC)
- Allen-Bradley PLC5, SLC500 (uses the ABTCP driver)
- AutomationDirect Koyo (uses the KOYO driver)
- AutomationDirect PAC 3000 (uses the PAC3K driver)
- Schneider Unity Modbus (uses either the MOTCP or SCHNE drivers)

NOTE: Using Tag Integration will add the device’s driver to the project, and as such is subject to any tag and driver limitations on the runtime license. Therefore, it is important when using this feature to ensure the current license level supports adding any additional tags and/or drivers.

Overview

Tag integration is an enhanced framework for device communication. It is built on the same communication drivers that are described in the Drivers section, but instead of manually configuring driver worksheets to associate project tags with device registers, you can use the Object Finder to browse the device registers and add them directly to your project.

Registers added in this way appear as tags in the Shared Database folder. They are "live" — that is, they are continuously and bilaterally updated during run time as long as the source is also running and properly connected — and they can be used anywhere in your project that you would normally use project tags.

These tags count against your target system's tag limit only when you use them in your project. This is different from the driver worksheets, where you might create a group of tags in order to associate them with device registers but then never actually use them in your project.

Tag integration is configured in the Communication tab of the project settings.

This feature is supported only for certain third-party control applications and devices, because additional work is required to upgrade an existing communication driver for tag integration.

SECTION 4 - MONITORING AND TROUBLESHOOTING

This section explains how to remotely connect to a runtime running a different machine from the BLUE Open Studio development environment using tools called Database Spy, Output window, and LogWin.

Concept

Database Spy, Output window, and LogWin are designed to troubleshoot and debug the project from the development environment. These tools allow the user to access project tags, receive feedback on internal and communication tasks, and even run certain functions from the development environment. These tools are accessible even without being physically at the runtime device.

The Database Spy is a monitoring tool that allows you to: monitor and force values to project tags; execute and test functions as well as math expressions.

The Output window is a troubleshooting tool that allows you to view log messages generated during project runtime. The window displays Driver and OPC transactions, module activation, trace tags, and so on.

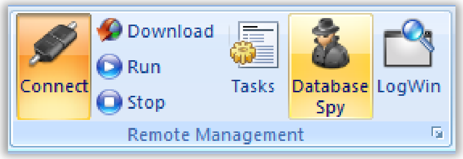
The LogWin is similar to the Output window except that is allows you to view log messages generated by a runtime running on a remote machine. It also provides that ability to same log messages to a disk file.

LAB 7 - USING THE DATABASE SPY AND LOGWIN

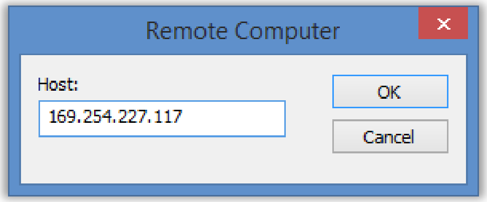
In this lab you will use the Database Spy and LogWin to monitor and troubleshoot your project running on out local machine.

Remote Database Spy

1. Open the **Remote Management** tool and connect to the target device as before.
2. Open the **Project** tab. Download the project changes made in the last few labs to the device. Once complete, select **Run**. Verify the project is running on the target device.
3. Close the **Remote Management** tool. This will not stop the project.
4. On the Home tab of the Ribbon, click **Database Spy** under the Remote Management group.



5. The Remote Computer dialog is opened. Enter the IP address of your target runtime device.

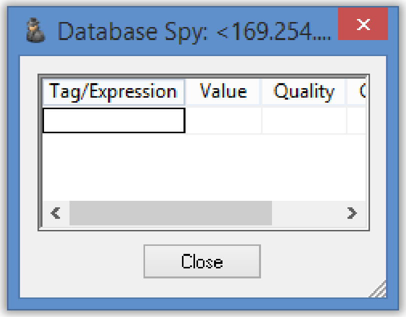

Remote Computer

Host: 169.254.227.117

6. Click **OK**.

> **Note**: If you have trouble connecting, then make sure that the runtime is running on the target station and the firewall rules have been configured to allow remote access to the runtime software according to the instructions in the appendix.

7. If the connection is good, then the Remote Database Spy window is displayed.

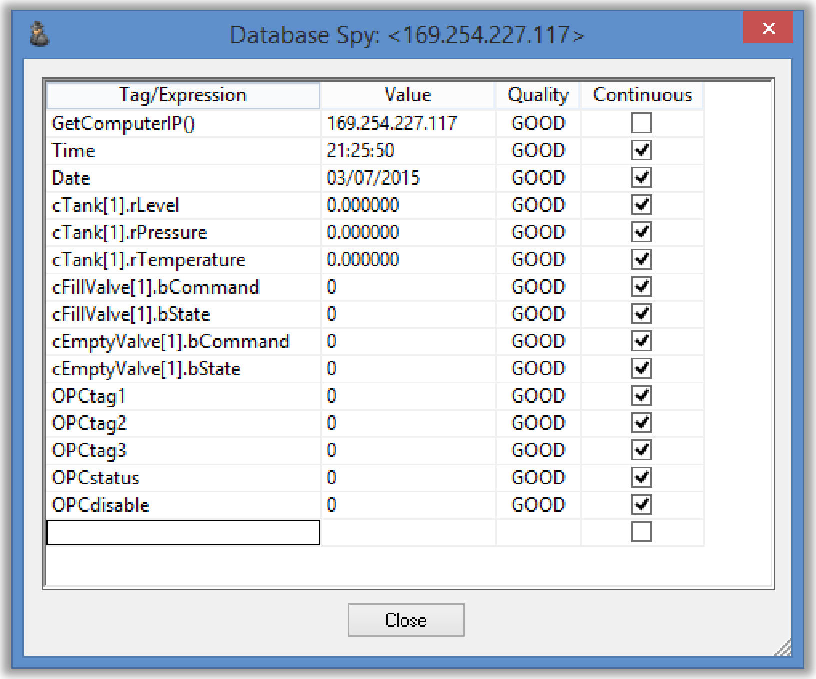

Database Spy: <169.254....

Tag/Expression	Value	Quality

The window contains the following elements:

- **Tag/Expression**: Specify a project tag, system tag, or expression that you want to monitor.
- **Value:** Displays the value returned by the tag/expression.
- **Quality:** Displays the quality (GOOD, BAD or UNCERTAIN) of the value returned by the tag/expression.
- **Continuous:** Select this option to have the project continuously evaluate the tag/expression.

8. Make the Database Spy window bigger, and add the following functions and tags to the Tag/Expression column.



Tag/Expression	Value	Quality	Continuous
GetComputerIP()	169.254.227.117	GOOD	☐
Time	21:25:50	GOOD	☑
Date	03/07/2015	GOOD	☑
cTank[1].rLevel	0.000000	GOOD	☑
cTank[1].rPressure	0.000000	GOOD	☑
cTank[1].rTemperature	0.000000	GOOD	☑
cFillValve[1].bCommand	0	GOOD	☑
cFillValve[1].bState	0	GOOD	☑
cEmptyValve[1].bCommand	0	GOOD	☑
cEmptyValve[1].bState	0	GOOD	☑
OPCtag1	0	GOOD	☑
OPCtag2	0	GOOD	☑
OPCtag3	0	GOOD	☑
OPCstatus	0	GOOD	☑
OPCdisable	0	GOOD	☑
			☐

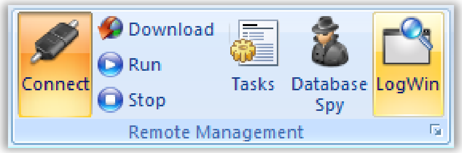
If everything is working, then you should see tag values and GOOD quality. If you are experiencing driver or OPC communication errors, then the tags would show BAD quality.

9. When you are done, click **Close** to disconnect from the remote computer.

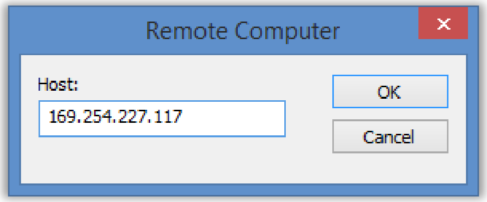
Pro Tip: Keep the Remote Database Spy on top of other windows even when switching programs, by right-clicking the Database spy icon () and selecting **Always on Top.**

Remote LogWin

1. On the Home tab of the Ribbon, click **LogWin** under the Remote Management group.



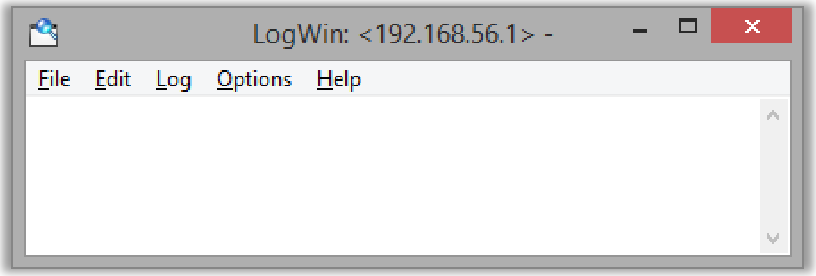
2. The Remote Computer dialog is opened. Enter the IP address of the Target Runtime device.



Host: 169.254.227.117

3. Click **OK**.

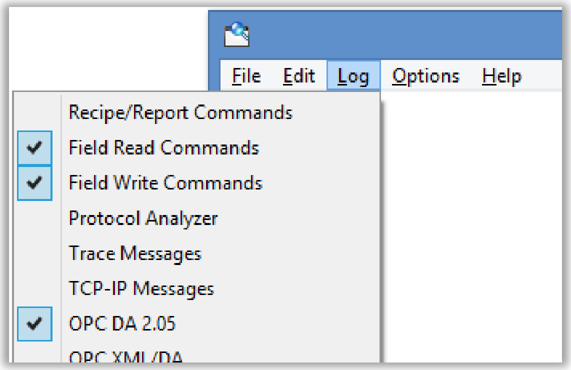
4. If the connection is good, then the Remote LogWin window is displayed.



5. The menu contains the following options:

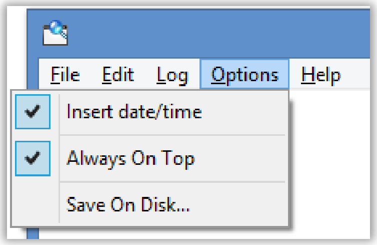
   - **File:** Gives **Save**, **Print**, and **Exit** settings
   - **Edit:** Allows you to **Pause**, **Copy**, and **Clear** the contents
   - **Log:** Configure the LogWin message settings. Refer to the table under the **Output Window** for descriptions of these items.
   - **Options:** Allows you to insert a timestamp on messages, locks the LogWin on top of other windows, and configure autosave options.
   - **Help:** Displays LogWin version

6. Open the Log menu, and select Field Read Commands, Field Write Commands, and OPC DA 2.05.



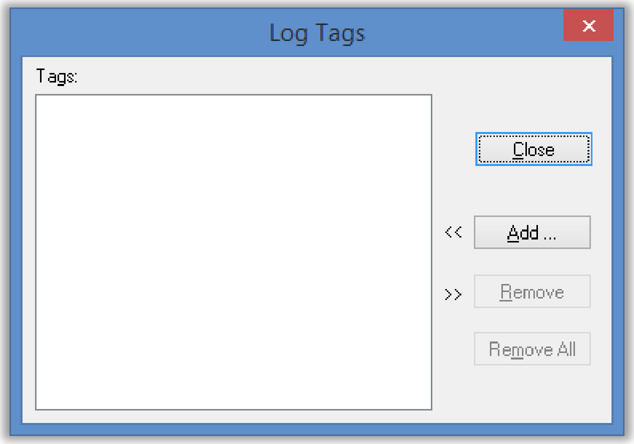
Recipe/Report Commands
☑ Field Read Commands
☑ Field Write Commands
Protocol Analyzer
Trace Messages
TCP-IP Messages
☑ OPC DA 2.05
OPC XML/DA

7. Open the Options menu, and click Insert Date/Time and Always on Top.



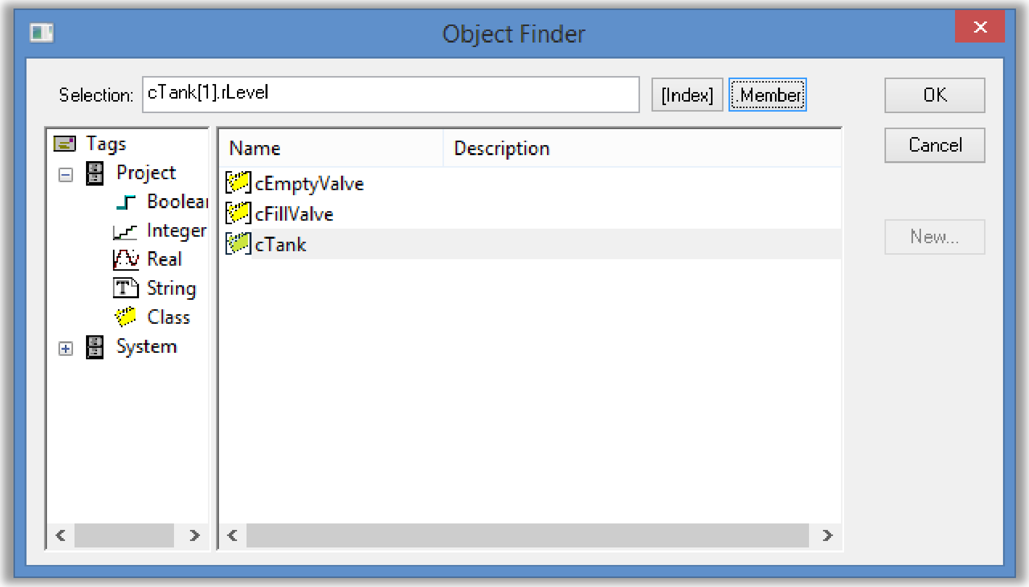
8. The Driver and OPC task messages will now be displayed.

9. Open the Log Menu, and select **Tags…** to open the Log Tags window.



10. Click **Add** to open the Object Finder.

11. Use the Object Finder to add the tag called **cTank[1].rLevel**.


Selection: cTank[1].rLevel

12. Anytime the tag called **cTank[1].rLevel** tag changes value, then the new tag value will be logged to the LogWin window.

Note: If you need to pause the display, you can select **Pause** from the **Edit** menu or by pressing **<F9>.**

13. When you are done, just close the window to disconnect.

Pro Tip: Keep the Remote Logwin on top of other windows even when switching programs. Open the Options menu from the menu bar and select **Always on Top.**

Pro-face
by Schneider Electric

Module 5 - Screens and Graphics

SECTION 1 - SCREENS AND SCREEN GROUPS

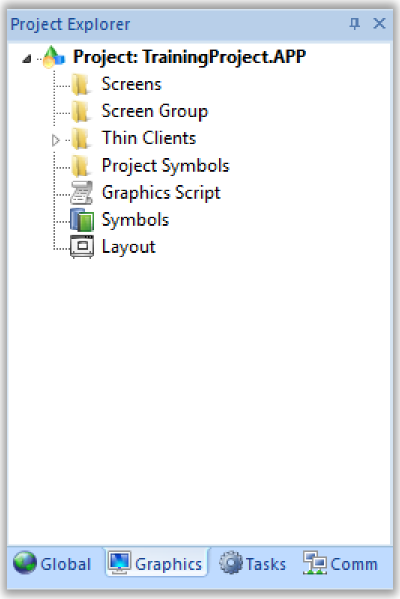
This section explains how to create screens and screen groups.

Concept

The most basic function performed by BLUE Open Studio is to provide a window into the process. The ability to display the status of the process by interacting with instrumentation (or computers) is described as the Human-Machine Interface (HMI).

BLUE Open Studio allows you to create projects that can monitor processes using high-resolution color screens. Each screen is an association of static and animated objects. Individual screens can be combined into more manageable screen groups.

Screens and screen groups are organized into the Screens and Screen Group folders on the Graphics tab of the Project Explorer.



Overview

Screens

The Screens folder contains all of your Screen worksheets, both completed and still in development.

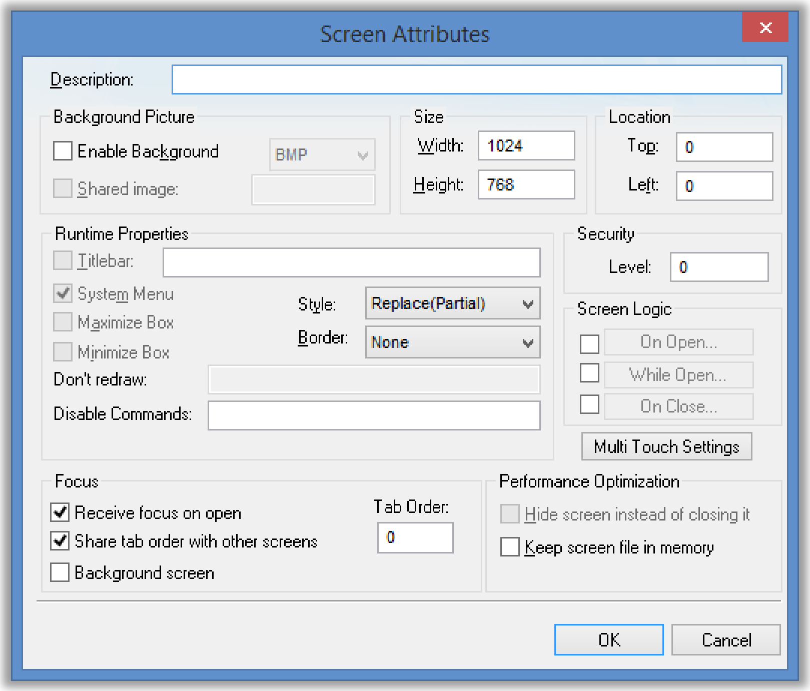
To create a new Screen worksheet, do one of the following:

- On the Insert tab of the ribbon, in the Graphics group, click **Screen.**
- On the Graphics tab of the Project Explorer, right-click the **Screens** folder and click **Insert** on the shortcut menu.
- Click the Application button, click **New** on the Application menu, click the **File** tab in the New dialog, select **Screen** from the list of worksheet types, and click **OK**.

Pro Tip: You cannot rename a screen using the development environment. If you want to rename a screen, then open a screen for editing, click the Application button to open the Application menu, select Save As, save the screen using the new name, and delete the old screen.

Screen attributes

When a Screen worksheet is opened for the first time, the Screen Attributes dialog for that worksheet is automatically displayed.



Screen Attributes

Description:

Background Picture

□ Enable Background BMP

□ Shared image:

Size

Width: 1024

Height: 768

Location

Top: 0

Left: 0

Runtime Properties

□ Titlebar:

☑ System Menu

□ Maximize Box

□ Minimize Box

Style: Replace(Partial)

Border: None

Don't redraw:

Disable Commands:

Security

Level: 0

Screen Logic

□ On Open...

□ While Open...

□ On Close...

Focus

☑ Receive focus on open

☑ Share tab order with other screens

□ Background screen

Tab Order: 0

Performance Optimization

□ Hide screen instead of closing it

□ Keep screen file in memory

You can also access the Screen Attributes dialog for an existing Screen worksheet (assuming the worksheet is open for editing) by doing one of the following:

- On the Graphics tab of the ribbon, in the Screen group, click Attributes.
- Right-click anywhere in the Screen worksheet and then click Screen Attributes on the shortcut menu.

The Screen Attributes dialog is used to configure runtime settings — such as size, location, title bar, security level, and screen logic — for a specific project screen.

Group / Setting		Description
Description		A brief description of the project screen. This is not shown anywhere during runtime.
Background Picture	**Enable Background**	Enables the background picture layer and specifies the file type of the picture. When this option is selected, a new BMP file with the same name as the screen is automatically saved in the Screen sub-folder of your project folder (e.g., screen_name.BMP).

Group / Setting		Description
	Shared Image	Uses the specified image file located in the Screen sub-folder of your project folder. Only BMP files are supported in projects developed for Windows Embedded target systems.
Size	**Width**	The default width of the screen (in pixels) when it is initially displayed. The user can change the size during runtime if the screen is set to be resizable; see **Border** below.
	Height	The default height of the screen (in pixels) when it is initially displayed. The user can change the size during runtime if the screen is set to be resizable; see **Border** below.
Location	**Top**	The default distance (in pixels) between the top of the computer display and the top of the screen when the screen is initially displayed. The user can change the location during runtime if the screen is set to have a border and title bar; see **Border** below.
	Left	The default distance (in pixels) between the left side of the computer display and the left side of the screen when the screen is initially displayed. The user can change the location during runtime if the screen is set to have a border and title bar; see **Border** below.
Runtime Properties	**Style**	The general runtime behavior of the screen: • **Overlapped**: Opens the screen without closing any other screens. • **Popup**: Forces the screen in front of all other screens but does not close them. • **Replace (Partial)**: Opens the screen and closes all other Replace screens that it partially covers. This is the default. • **Dialog**: Similar to **Popup**, except that the other screens are also disabled until the dialog is closed by the user. • **Replace (Complete)**: Similar to **Replace (Partial)**, except that it closes only other Replace-type screens that it completely covers.
	Border	The type of border around the screen: • **None**: No border; the screen is a flat, immovable rectangle on the computer display. This is the default. • **Thin**: A thin border that makes the screen a movable window. Includes title bar. • **Resizing**: A thick border that makes the screen a movable, resizable window. Includes title bar.
	Titlebar	Shows the window's title bar with the specified text.
	System Menu	Provides a menu of basic window commands at the left end of the title bar.
	Maximize Box	Shows the Maximize button at the right end of the title bar.
	Minimize Box	Shows the Minimize button at the right end of the title bar.
	Don't Redraw	While this tag/expression evaluates as TRUE, the screen's graphics are not updated.

Group / Setting		Description
	Disable Commands	While this tag/expression evaluates as TRUE, the screen is locked against user interaction but the graphics continue to be updated.
Security		The minimum security level that a user must have to access this screen.
Screen Logic	**On Open**	Lists expressions to be evaluated once when the screen is opened, similar to a Math worksheet.
	While Open	Lists expressions to be continuously evaluated while the screen is open, similar to a Math worksheet.
	On Close	Lists expressions to be evaluated once when the screen is closed, similar to a Math worksheet.
Multi-Touch Settings		Customizes the Multi-Touch settings for this screen.
Focus	**Receive focus on open**	When the screen is opened, the focus will automatically go to the first object in the screen (according to Object ID) that can receive focus, as if the user tabbed into the screen.
	Share tab order with other screens	When the user tabs through the last object in the screen, the focus will go to the next open screen (according to Tab Order below) rather than back to the first object in the current screen.
	Tab Order	Similar to Object ID for screen objects, this determines the tab order between screens when multiple screens are open. When the user tabs through the last object in a screen, the focus will go to the open screen with the next higher Tab Order number. Each screen should have a unique Tab Order number between 0 and 32767.
	Background screen	When the user clicks on the screen, it remains in the background and is not brought in front of the other open screens. If more than one screen has this option selected, then the screens are arranged in tab order with the greatest Tab Order number being the farthest back.
Performance Optimization	**Hide screen instead of closing it**	Closing the screen (by any user action or system process) only makes it hidden, and reopening the screen makes it visible again. This makes the screen appear to open very quickly.
	Keep screen file in memory	When the screen is closed, the screen file is kept in memory so that it does not need to be reloaded from the hard drive when the screen is reopened. The screen still needs to be redrawn, however. This is not as fast as **Hide screen instead of closing it** above, but it still makes the screen appear to open quickly.

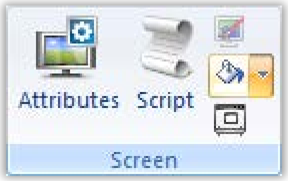
Modifying a screen's background color or image

A project screen can have either a solid background color or an editable background image.

Selecting a screen's background color

By default, a newly created project screen has a solid white background. You can change the screen background color on a screen that is open for editing using the Background Fill color picker by doing one of the following:

- On the Graphics tab of the ribbon, in the Screen group, click Background Color.



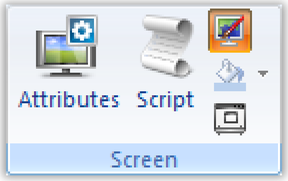
- Right-click anywhere in the Screen worksheet and then click Background Color on the shortcut menu.

Note: You cannot select a background color if the screen has a Background Picture.

Modifying a screen's background image

If you enabled the Background Picture in the Screen Attributes, then make sure the screen file is open for editing and do the following:

- On the Graphics tab of the ribbon, in the Screen group, click Background Image.



Microsoft Paint is run automatically and the image file is opened for editing.

Screen Groups

The Screen Group folder combines individual Screen worksheets from the Screens folder into more manageable groups.

To create a new Screen Group, do one of the following:

- On the Insert tab of the ribbon, in the Graphics group, click **Screen Group**.

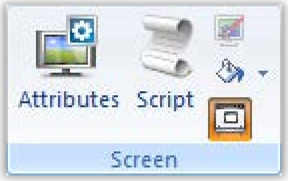
- On the Graphics tab of the Project Explorer, right-click the **Screen Group** folder and click **Insert Screen Group** on the shortcut menu.

Pro Tip: You cannot rename or change a screen group using the development environment. If you want to rename or change a screen, then you must delete the existing screen group and recreate it.

Layout Tool

To open the Layout tool, do one of the following:

- On the Graphics tab of the ribbon, in the Screen group, click **Layout**.



- On the Graphics tab of the Project Explorer, right-click the Layout tool and click **Open** on the shortcut menu.

The Layout tool displays all screens currently open in the Workspace and allows you to:

- Modify the Screen Attributes
    - Right-click on the screen displayed on the Layout tab and use the alignment options or the Screen Attributes link to modify the screen position.
    - Click and drag the screen to change its position or resize it.
- Visualize how the screens fit together during runtime. This option is especially useful when creating pop-up/dialog screens or groups of screens.

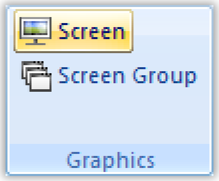
LAB 8 - CREATE SCREENS AND SCREEN GROUP

In this lab you will create screens called Header, Navigation, Main, and Template. You will also create a screen group called StartUp that opens the Header, Navigation, and Main screens when the project runtime client is started.

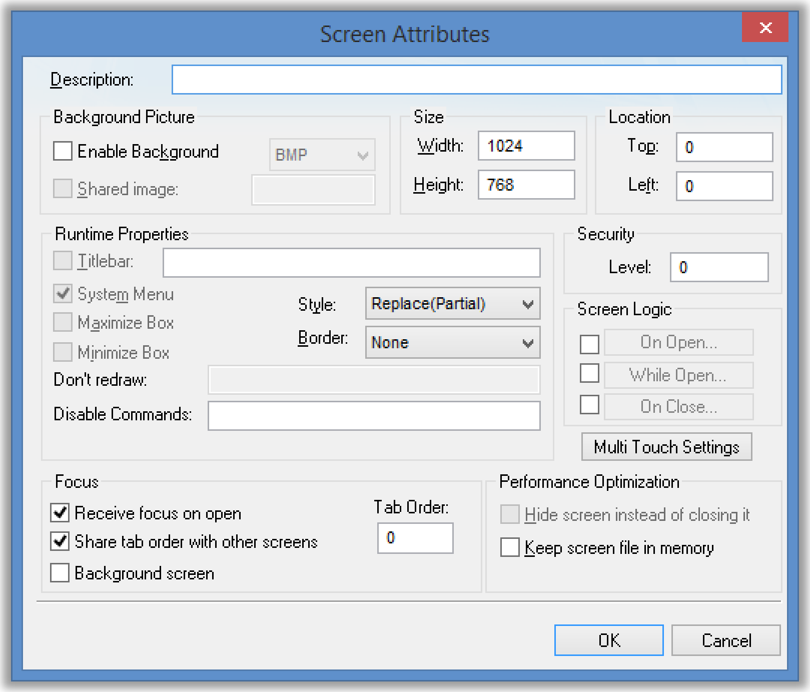
Create Screens

Header screen

1. On the Insert tab of the ribbon, in the Graphics group, click **Screen**.



The Screen Attributes dialog is opened.



Screen Attributes

Description:

Background Picture
□ Enable Background BMP
□ Shared image:

Size
Width: 1024
Height: 768

Location
Top: 0
Left: 0

Runtime Properties
□ Titlebar:
☑ System Menu
□ Maximize Box
□ Minimize Box
Style: Replace(Partial)
Border: None
Don't redraw:
Disable Commands:

Security
Level: 0

Screen Logic
□ On Open...
□ While Open...
□ On Close...

Multi Touch Settings

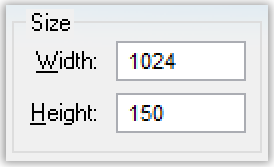
Focus
☑ Receive focus on open
☑ Share tab order with other screens
□ Background screen
Tab Order: 0

Performance Optimization
□ Hide screen instead of closing it
□ Keep screen file in memory

OK Cancel

Pro Tip: You can also right-click the Screens folder on the Graphics tab of the Project Explorer and the click Insert on the shortcut menu.

2. In the Size group, in the **Width** and **Height** fields, enter `1024` and `150`, respectively.

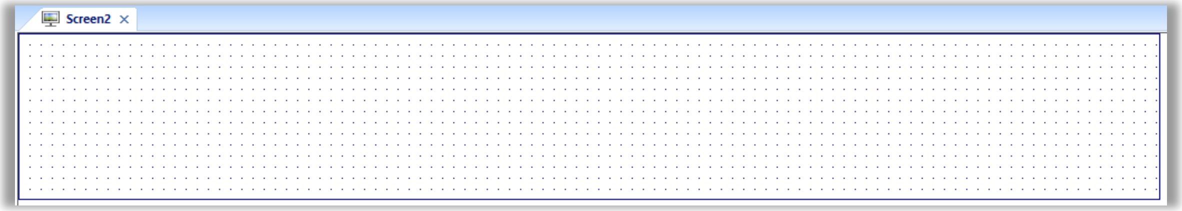


Size

Width: 1024

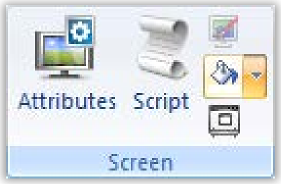
Height: 150

3. Click **OK** to close the Screen Attributes dialog. The new screen is opened in the Workspace.

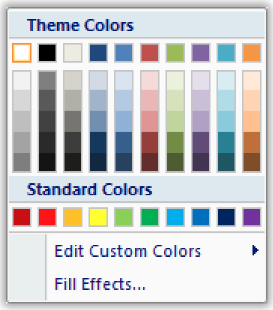


Note: The name of your screen may be different from that shown in the picture above.

4. On the Graphics tab of the ribbon, in the Graphics group, click **Background Color**.

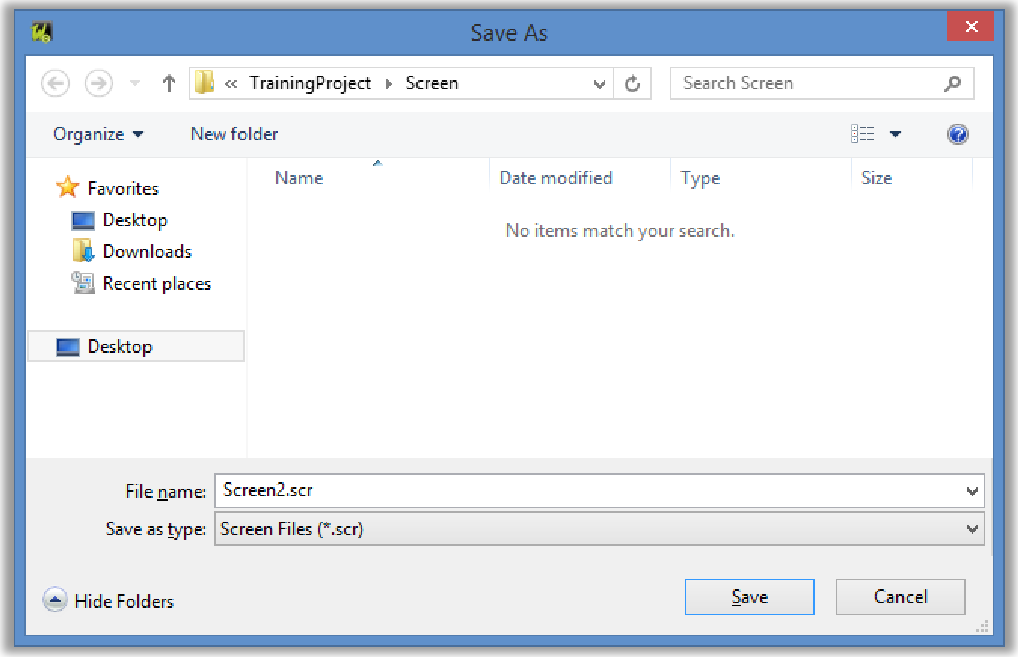


The Theme Colors dialog is opened.



5. Select the medium Olive Green color.

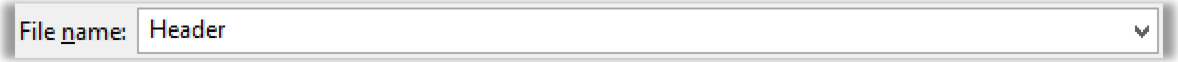
6. Click the Application button and click **Save As** on the Application menu. The Save As dialog is opened.



File name: Screen2.scr

Save as type: Screen Files (*.scr)

7. In the **File name** field, enter `Header`.



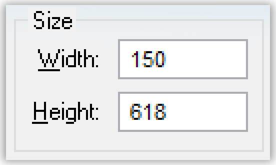
File name: Header

8. Click **Save**. The Save As dialog is closed.

Navigation screen

1. On the Insert tab of the ribbon, in the Graphics group, click **Screen**. The Screen Attributes dialog is opened.

2. In the Size group, in the **Width** and **Height** fields, enter `150` and `618`, respectively.

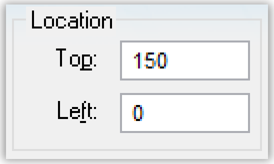


Size

Width: 150

Height: 618

3. In the Location group, in the **Top** field, enter `150`.



Location

Top: 150

Left: 0

4. Click **OK** to close the Screen Attributes dialog. The new screen is opened in the Workspace.

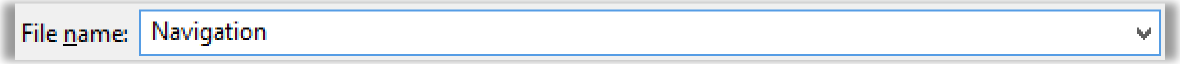
5. Click the tab on the Workspace corresponding to the new screen.

6. On the Graphics tab of the ribbon, in the Graphics group, click **Background Color**. The Theme Colors dialog is opened.

7. Select the light Olive Green color.

8. Click the Application button and click **Save As** on the Application menu. The Save As dialog is opened.

9. In the **File name** field, enter `Navigation`.



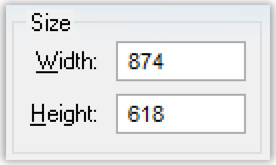
File name: Navigation

10. Click **Save**. The Save As dialog is closed.

Template screen

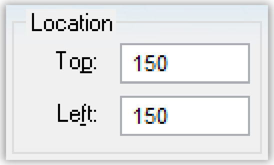
1. On the Insert tab of the ribbon, in the Graphics group, click **Screen**. The Screen Attributes dialog is opened.

2. In the Size group, in the **Width** and **Height** fields, enter `874` and `618`, respectively.



Size
Width: 874
Height: 618

3. In the Location group, in the **Top** and **Left** fields, enter `150`.



Location
Top: 150
Left: 150

4. Click **OK** to close the Screen Attributes dialog. The new screen is opened in the Workspace.

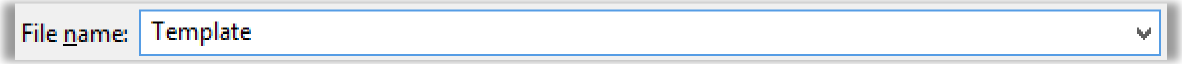
5. Click the tab on the Workspace corresponding to the new screen.

6. On the Graphics tab of the ribbon, in the Graphics group, click **Background Color**. The Theme Colors dialog is opened.

7. Select the light Gray color.

8. Click the Application button and click **Save As** on the Application menu. The Save As dialog is opened.

9. In the **File name** field, enter `Template`.

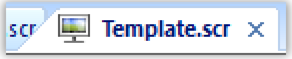


File name: Template

10. Click **Save**. The Save As dialog is closed.

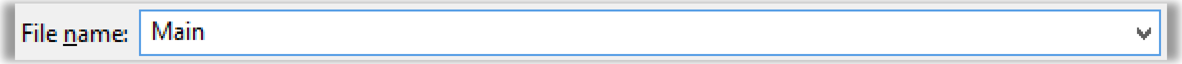
Main screen

1. Click the **Template.scr** tab on the Workspace.



2. Click the Application button and click **Save As** on the Application menu. The Save As dialog is opened.

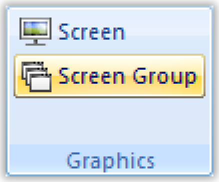
3. In the **File name** field, enter `Main`.


File name: Main

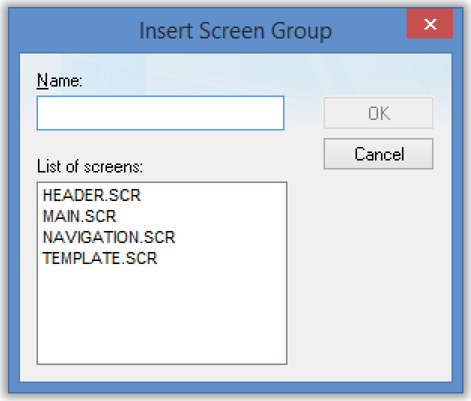
4. Click **Save**. The Save As dialog is closed.

Create Screen Group

1. On the Insert tab of the ribbon, in the Graphics group, click **Screen Group**.



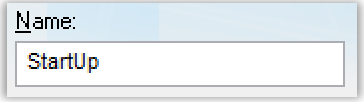
The Insert Screen Group dialog is opened.


Insert Screen Group

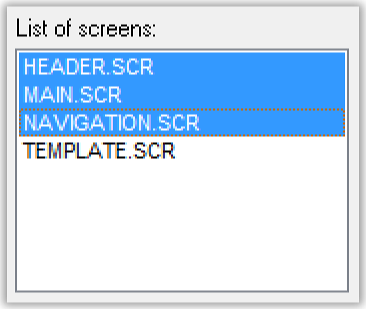
Name:

List of screens:
HEADER.SCR
MAIN.SCR
NAVIGATION.SCR
TEMPLATE.SCR

2. In the **Name** field, enter `StartUp`.

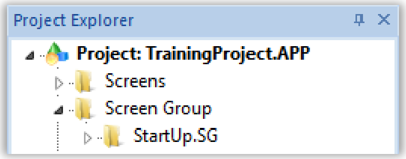

Name: StartUp

3. In **List of screens** list box, select the screens **HEADER**, **MAIN**, and **NAVIGATION** screens.

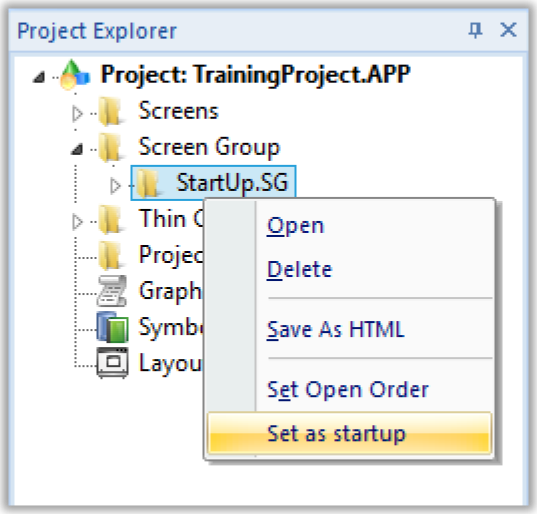


4. Click **OK**. The Insert Screen Group is closed.

5. On the Graphics tab of the Project Explorer, click the chevron to the left of the **Screen Group** folder. The Screen Group folder is expanded.



6. On the Graphics tab of the Project Explorer, right-click the **StartUp.SG** screen group folder and click **Set as startup** on the shortcut menu.



Pro Tip: You can view and change the screen that will be opened when the projet runtime client starts by selecting Viewer in the Settings group on the Project tab of the Ribbon.

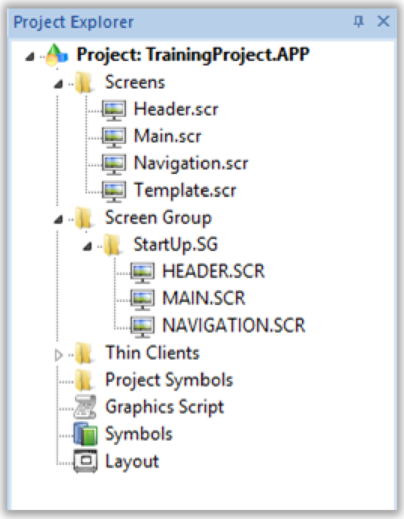
Verification and Test

Do the following to verify that you successfully completed this lab:

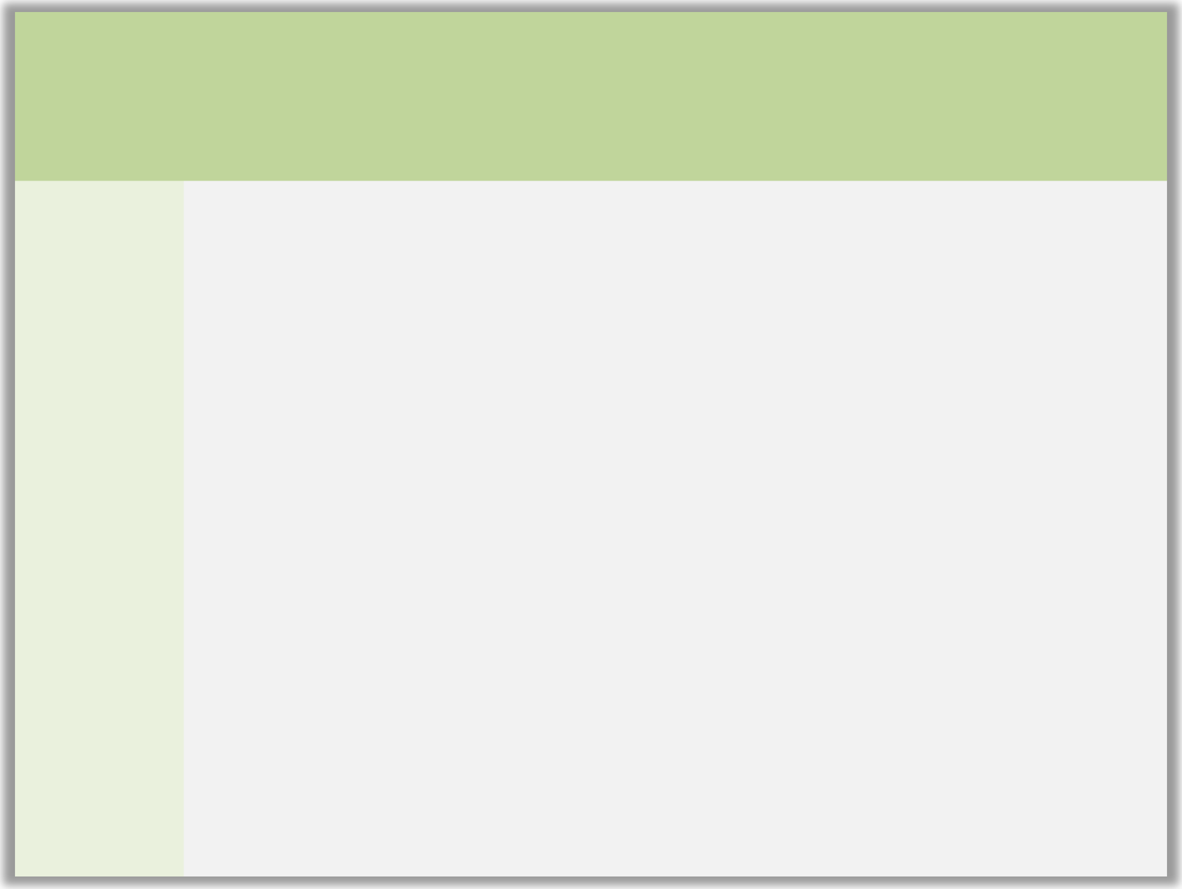
1. On the Graphics tab of the Project Explorer, click the chevron to the left of the **Screens** folder. The Screens folder is expanded.

2. On the Graphics tab of the Project Explorer, click the chevron to the left of the **Screen Group** folder. The Screen Group folder is expanded. Click the chevron to the left of the **StartUp.SG** folders. The StartUp.SG screen group folder is expanded.

Your Project Explorer should look similar to the following:



If your runtime is on a remote machine, then use the the Remote Management tools on the Home tab of the Ribbon to download your project to the remote machine. Start the runtime and test. The project runtime client should look similar to the following:



SECTION 2 - FORMATTING

This section explains how to format screen objects using standard graphics options.

Concept

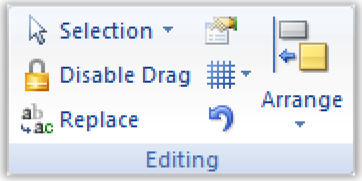
Formatting allows the user to format and organize the objects and shapes on the screen of their project, as well as set different editing properties of a screen. This is to make creating and editing objects on the screen as effortless as possible.

Editing and formatting screens and objects can be done through the Editing group on the Graphics tab of the Ribbon, as well as the Format tab of the Ribbon.

Overview

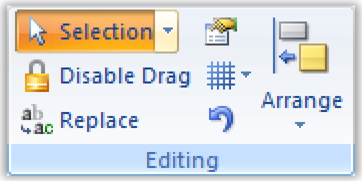
Editing group

The Editing group on the Graphics tab of the Ribbon provides tools for general screen editing.

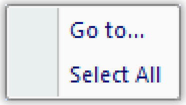


Selection

Click **Selection** to display a mouse cursor that you can use to select and move objects on the screen.



Click the chevron to the right of **Selection** to open the shortcut menu.

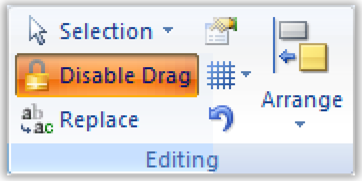


Click **Go To** to select a specific object on the screen, designated by Object ID number.

Click **Select All** to select all objects on a screen.

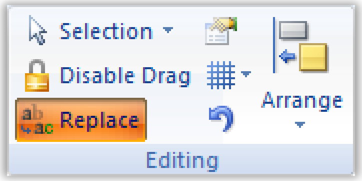
Disable Drag

Click **Disable Drag** to disable the dragging of objects in the screen editor, to prevent accidental moves after you've layed out the screen exactly as you want it.

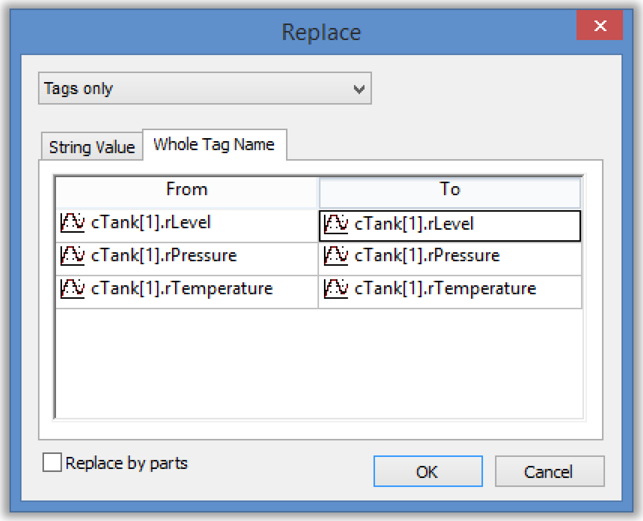


Replace

Click **Replace** to replace all occurences of a tag or text string in the current screen.



The Replace dialog is opened.



From	To
cTank[1].rLevel	cTank[1].rLevel
cTank[1].rPressure	cTank[1].rPressure
cTank[1].rTemperature	cTank[1].rTemperature

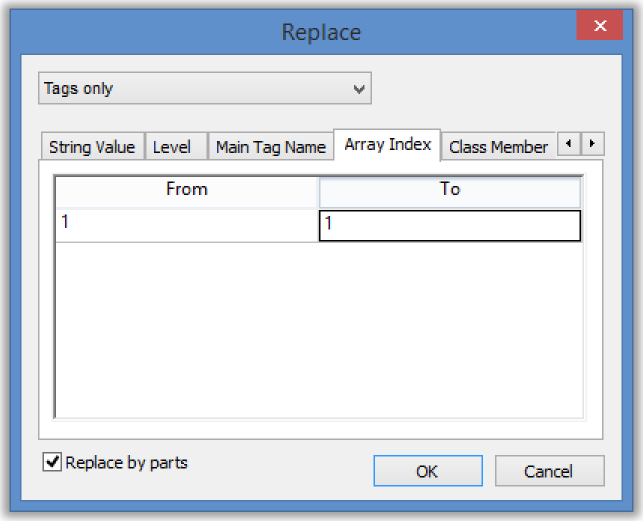
Note: You can also open the Replace dialog in the Tags group on the Home tab of the Ribbon.

You can replace one or more tags by clicking the **Whole Tag Name** tab. Current tags used are displayed. The original tag names are shown in the **From** column on the left, and you can enter your new tag names in the **To** column on the right.

This does not rename or delete any tag — it only replaces the tags used in the object with other tags from the project tags database.

If you have complex tag names using classes or arrays and you only want to change a portion of the tag name, then click **Replace by parts**.

The **Whole Tag Name** tab will be replaced by separate tabs for **Level**, **Main Tag Name**, **Array Index**, **Class Member**, and **Tag Field**.



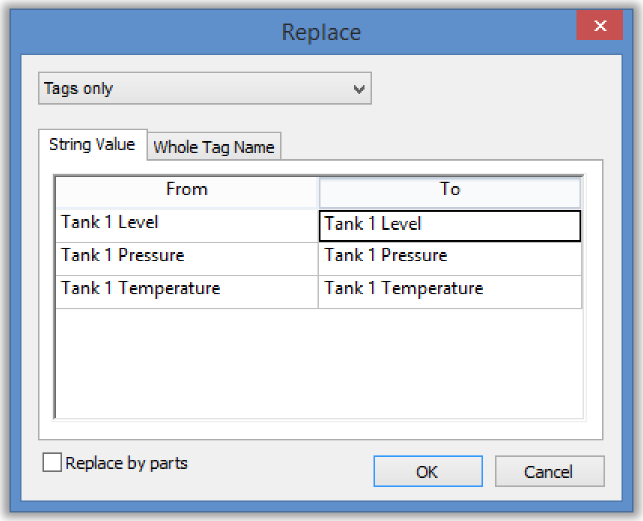
Tags only

From	To
1	1

☑ Replace by parts

Pro-Tip: To replace all occurances of a tag in your project, you can open Global Replace in the Tags group on the Home tab of the Ribbon.

You can also replace one or more text strings (e.g., button captions, descriptive text) by clicking the **String Value** tab.



Tags only

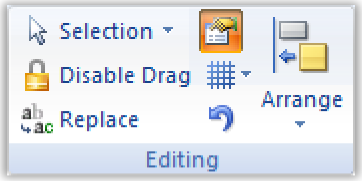
From	To
Tank 1 Level	Tank 1 Level
Tank 1 Pressure	Tank 1 Pressure
Tank 1 Temperature	Tank 1 Temperature

□ Replace by parts

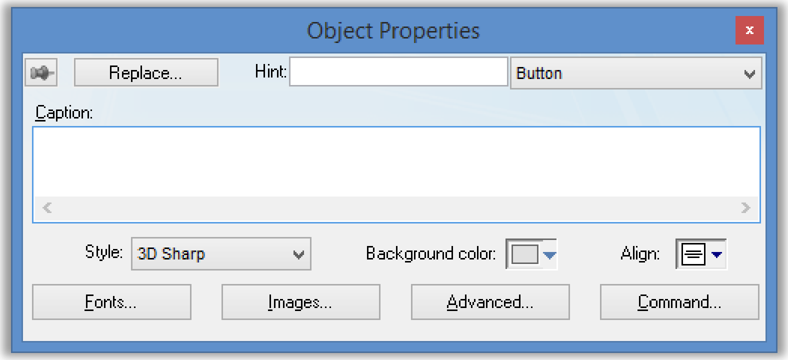
When you are done, click **OK**. The Replace dialog is closed.

Object Properties

Click Properties to show the configurable properties of a screen object or animation.



The Object Properties dialog is opened.



Hint:

Button

Caption:

Style: 3D Sharp

Background color:

Align:

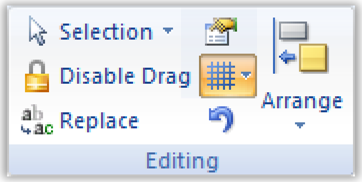
Note: You can also open the Object Properties dialog by right-clicking a screen object clicking Properties on the shortcut menu, or simply double-click a screen object.

This dialog shows the configurable properties of a screen object or animation. Each type of screen object and animation has its specific properties, but all types have a few properties in common.

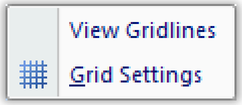
Area / Element Name	Description
Pushpin	When the pin button is released, the focus is passed to the object on the screen as soon as that object is selected. When the pin button is pressed, the focus is kept on the Object Properties window even when you click the objects on the screen.
Replace	Launches the Replace dialog, where you can replace strings, tags or properties for the selected object or group of objects.
Hint	Tooltip displayed during runtime, when the user hovers the mouse cursor over the object. This can be used to provide quick-help to the user. The text in the Hint field is also temporarily written to the Hint system tag, so that you can trigger actions based on the value of this tag when the mouse cursor is moved over a specific object. You can enable/disable this feature separately for full project viewers (Viewer in the Settings group on the Project tab of the ribbon) and for thin clients (Web in the Web group on the Project tab of the viewer).
Combo Box Button	The combo-box at the right side of the dialog allows you to select the specific object, group of objects, or animation that must be edited.
Close ✕	The Object Properties dialog is closed.

Grid Settings

Click **Grid Settings** to show/hide/edit the grid in the screen editor.

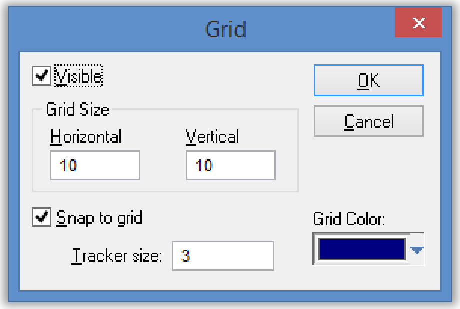


The Grid Settings shortcut menu is opened.



Click **View Gridlines** to show/hide the grid in the screen editor.

Click **Grid Settings** to edit the grid settings. The Grid dialog is opened.



Grid

☑ Visible

OK

Cancel

Grid Size

Horizontal: 10

Vertical: 10

☑ Snap to grid

Tracker size: 3

Grid Color: ■

Note: You can also open the Grid dialog by right-clicking anywhere in the Screen worksheet is Grid on the shortcut menu.

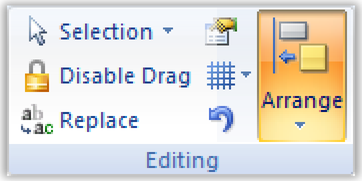
When you are done, click **OK**. The Grid dialog is closed.

Undo

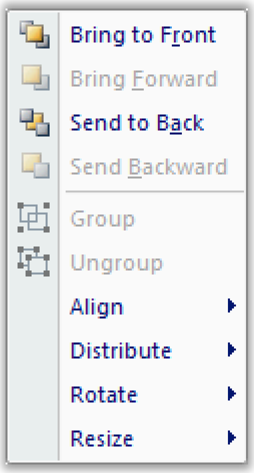
Click **Undo** to cancel the last action performed (and up to 20 actions taken prior to the last action) while working on a screen.

Arrange

Click Arrange to format and arrange objects in a project screen.



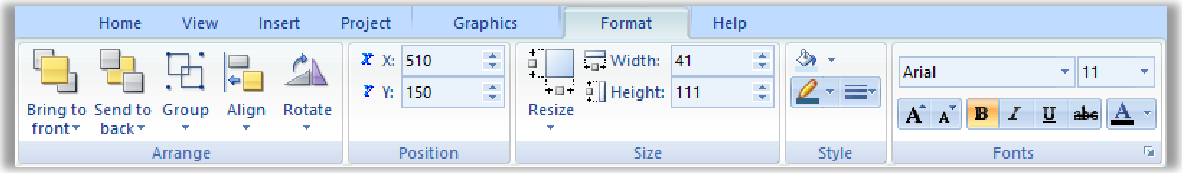
The Arrange shortcut menu is opened.



The Arrange shortcut menu provides the same tools as those found in the Arrange group on the Format tab of the Ribbon. See below for more information.

Format tab of the Ribbon

The Format tab of the Ribbon is used to format and organize the objects on a project screen.



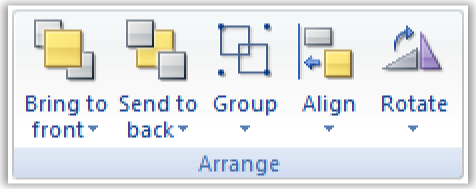
Note: This tab is available only when you have selected one or more objects on a project screen.

The formatting tools are organized into the following groups:

- **Arrange**: Arrange objects in a project screen, including bring to front and send to back, group, align, and rotate.
- **Position**: Precisely adjust the position of a screen object in a project screen.
- **Size**: Precisely adjust the size of a screen object.
- **Style**: Change the fill and line color of a screen object.
- **Fonts**: Change the caption font of a screen object.

Arrange

The Arrange group on the Format tab of the Ribbon includes tools used to arrange objects in a project screen.



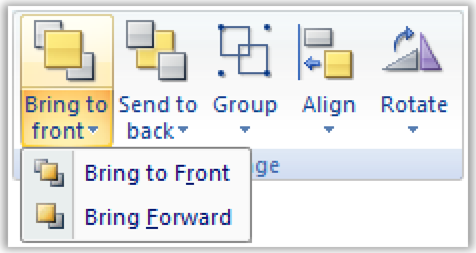
Bring to front / Send to back

Screen objects are automatically assigned ID numbers, starting with ID 0, as you add the objects to a project screen. (To check the ID number of an object, simply select it. The ID number is displayed on the status bar at the bottom of the development application window.) The object with ID 0 is the furthest back, behind all other objects on the screen, and each additional object is displayed in front of it. In other words, objects with higher ID numbers are displayed in front of objects with lower ID numbers.

On a finished screen, you may have objects arranged in overlapping layers that you want to rearrange so that they are displayed correctly. You can bring an object to the front, so that it is displayed in front of all other objects, or you can send an object to the back, so that it is displayed behind all other objects. You can also move an object forward or backward one layer at a time, if necessary.

Moving objects forward

On the Format tab of the ribbon, in the Arrange group, click the text area of the **Bring to front** control to open the shortcut menu.

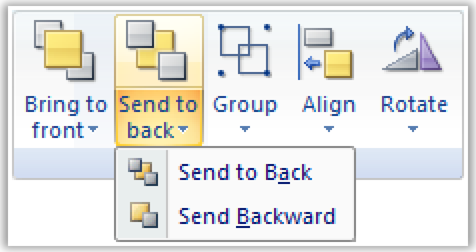


To bring the object all the way to the front (i.e., increase it to the highest ID number), then click **Bring to Front** on the shortcut menu.

To bring the object one layer forward (i.e., to increase its ID number by 1), then click **Bring Forward** on the shortcut menu.

Moving objects backward

On the Format tab of the ribbon, in the Arrange group, click the text area of the **Send to back** control to open the shortcut menu.



To send the object all the way to the back (i.e., decrease it to ID 0), then click **Send to Back** on the shortcut menu.

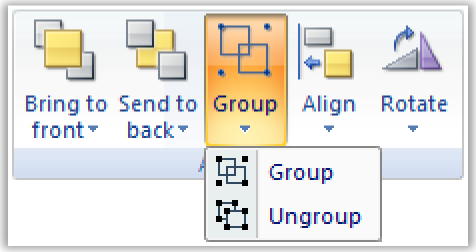
To send the object one layer backward (i.e., to decrease its ID number by 1), then click **Send Backward** on the shortcut menu.

Note: You can change the layering order of objects by right-clicking the selected objects in the screen, and then click Bring to Front, Bring Forward, Send to Back, and Send Backward on the shortcut menu.

Group

A group of objects can be selected, moved, copied, and in some cases modified as if it is a single object, which makes it easier to manage in a busy project screen. Moreover, a group can be grouped with other groups to create increasingly complex groups.

On the Format tab of the ribbon, in the Arrange group, click the **Group** control to open the shortcut menu.



To group two or more screen objects, then click Group on the shortcut menu.

To ungroup a group of objects, then click Ungroup on the shortcut menu.

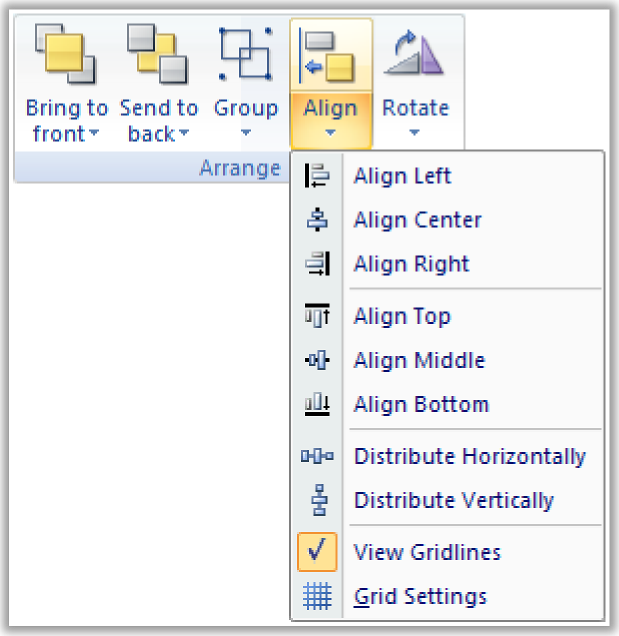
Note: You can group or ungroup objects by right-clicking the selected objects or group or objects in the screen, and then click Group on the shortcut menu.

Once you have created a group, you may choose to save a master of it in your Symbols Library and then reuse it elsewhere in your project. Each copy will be linked to the master so that if you change the master, all of the linked copies will also be changed. See the section called Creating your Own Symbols for more information.

Align

When you select two or more objects, you can align those objects based on the location of the last object selected. As you select objects, solid handles display on the last object selected, and the handles on all previously selected objects become empty (unfilled) boxes.

On the Format tab of the ribbon, in the Arrange group, click the text area of the **Align** control to open the shortcut menu.



Click the **Align Left** tool to align all selected objects to the left edge of the last object selected.

Click the **Align Center** tool to align all selected objects to the vertical center of the last object selected.

Click the **Align Right** tool to align all selected objects to the right edge of the last object selected.

Click the **Align Top** tool to align all selected objects to the top edge of the last objected selected.

Click the **Align Middle** tool to align all selected objects to the horizontal center of the last object selected.

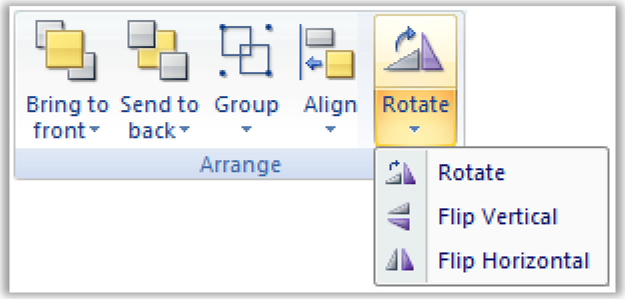
Click the **Align Bottom** tool to align all selected objects to the bottom edge of the last object selected.

Click the **Distribute Horizontally** tool to put an equal amount of horizontal space between a series of objects (two or more).

Click the **Distribute Vertically** tool to put an equal amount of vertical space between a series of objects (two or more).

Rotate

On the Format tab of the ribbon, in the Arrange group, click the text area of the **Rotate** control to open the shortcut menu.



Click the **Rotate** tool to rotate the selected object 90 degrees (a quarter turn) clockwise.

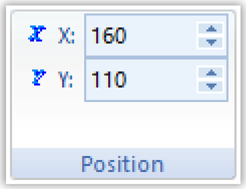
Click the **Flip Vertical** tool to invert the selected object vertically. The object rotates around an imaginary line through its vertical center until it is a mirror image of the original object.

Click the **Flip Horizontal** tool to invert the selected object horizontally. The object rotates around an imaginary line through its horizontal center until it is a mirror image of the original object.

Position

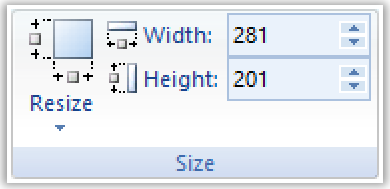
Moving a screen object by selecting and dragging it can be imprecise — especially if you have "snap to grid" enabled — so you can use the Position tools to precisely adjust the position of an object.

On the Format tab of the ribbon, in the Position group, use the X and Y controls to adjust the horizontal and vertical position of the object.

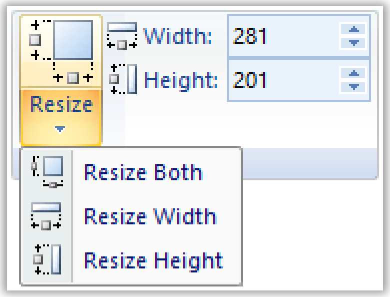

X: 160
Y: 110

Size

Use the Size group on the Format tab of the Ribbon to precisely adjust the size of a screen object.


Width: 281
Height: 201

Enter a specific width or height in the **Width** and **Height** Fields, respectively. Alternatively, click the text area of the **Resize** control to open the shortcut menu.



Click the **Resize Both** tool to set the width and height of all selected objects to the width and height of the last object selected, or to resize one selected object so that its width equals its height.

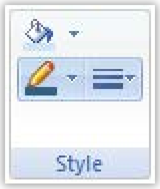
Click the **Resize Width** tool to set the width of all selected objects to the width of the last object selected, or to resize one selected object so that its width equals its height.

Click the **Resize Height** tool to set the height of all selected objects to the height of the last object selected, or to resize one selected object so that its height equals its width.

You also can use the mouse pointer. When you select an object (or group of objects) with the pointer, handles are displayed at each corner and at the midpoint of each side. To resize an object, drag a handle in the direction you want to resize the object. Dragging a side handle resizes the object in one direction only (height only or width only). Dragging a corner handle resizes the entire object (height and width). When you drag a corner handle, the object's proportions are constrained by default. To freely resize the object, hold down the SHIFT key as you drag the handle.

Style

Use the Style group of the Format tab on the Ribbon to change the fill and line color and width of a screen object.



Click the Fill Color tool to specify a default fill color for the following objects:

- Closed Polygons
- Ellipses
- Rounded Rectangles
- Rectangles

Click the Line Color tool to specify a line color for the following objects:

- Open Polygons
- Closed Polygons

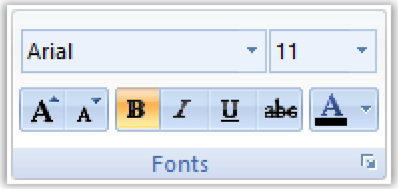
- Lines
- Ellipses
- Rounded Rectangles
- Rectangles

When you click the Line Color tool, the Line Selection dialog displays. Use this dialog to specify line styles and color for the selected objects.

Pro Tip: To save development time, select several objects and use Fill Color, Line Color, or Line Style to specify a default style for all of them at once.

Fonts

Use the Fonts group of the Format tab on the Ribbon to change the caption font of a screen object.



Click the **Fonts** tool to specify the font and color for selected Text objects, or to specify a default font and color for new Text objects.

Pro Tip: To save development time, select several Text objects and use the Fonts tool to specify font and color settings for all of the objects at once. (You cannot use this function for grouped Text objects however.)

SECTION 3 - THE SYMBOL LIBRARY

Using a step-by-step tutorial, this section explains how to insert system symbols to be used by the project.

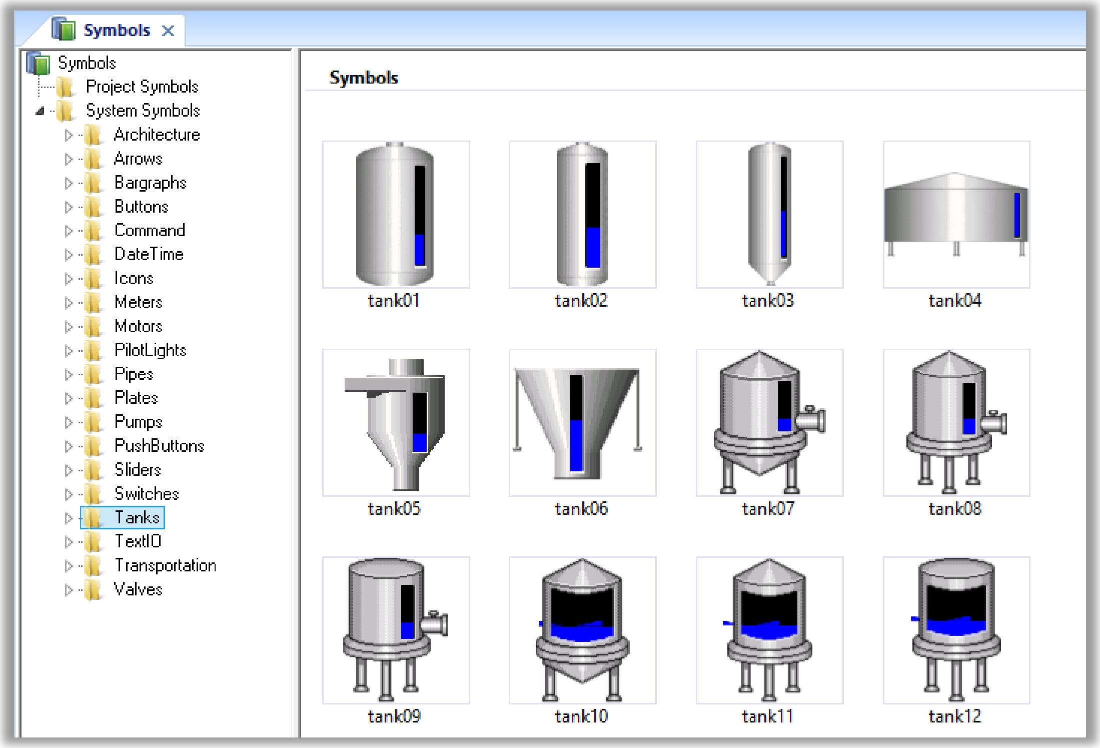
Concept

The symbols library contains a large selection of premade symbols that are installed with the BLUE Open Studio development environment, such as pushbuttons, toggle switches, gauges, dials, indicator lights, and so on.

Overview

The Symbols library is a visual browser for all of the symbols that are available to be used in your project screens. To open the library: on the **Graphics** tab of the Ribbon, in the **Libraries** group, click **Symbols**.

Expand the System Symbols and click any folder to see the available symbols.



The library is divided into two main folders: the Project Symbols folder contains your user-made symbols for the current project, and the System Symbols folder contains all of the premade symbols (sorted by type) that are installed with the BLUE Open Studio software.

To select a symbol and place it in a project screen:

1. Find the symbol you want in the library, and then double-click it. The mouse cursor will change to indicate that you have a symbol waiting to be placed.
2. Return to the project screen where you want to place the symbol.

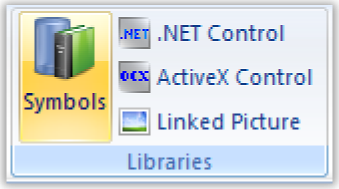
3. Click anywhere in the project screen to place the selected symbol.
4. Edit the symbol's object properties as needed, including any custom properties.

LAB 9 - USING THE SYMBOL LIBRARY

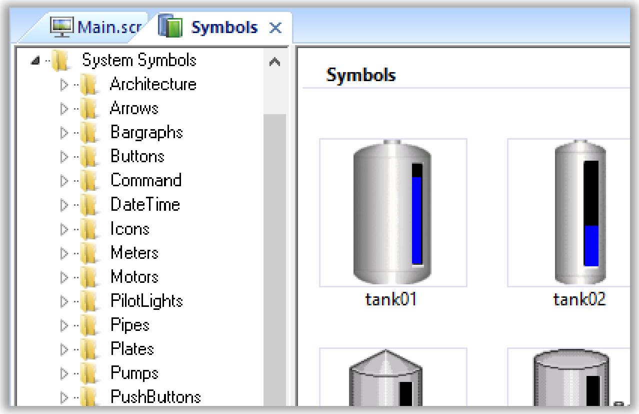
In this lab you will add and configure various symbols from the Symbols Library to create an HMI to monitor the tank farm.

Add Symbols to the Main screen

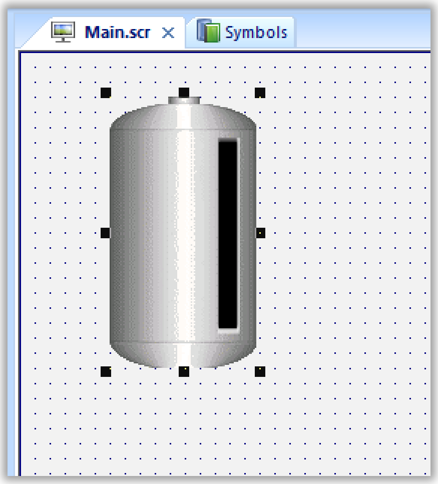
1. Open the Main screen.
2. In the Libraries group of the Graphics tab of the Ribbon, click on **Symbols**.



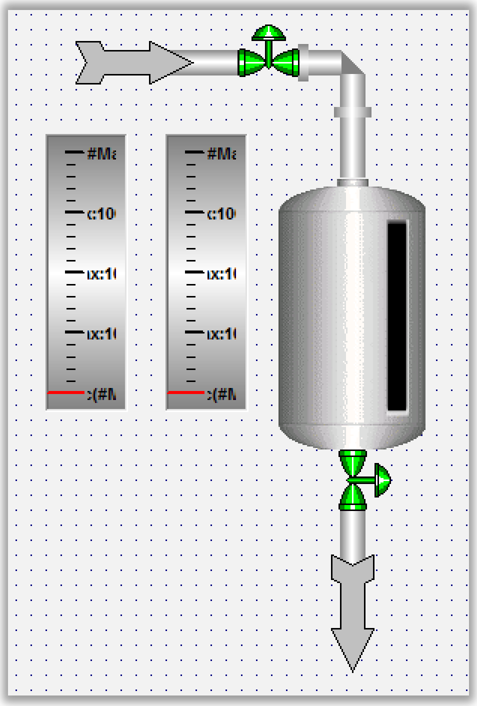
3. The Symbols tab will be displayed in the Workspace.
4. Expand the **System Symbols** folder of the System in the menu on the left.
5. Click on the **Tanks** folder in the menu on the left.
6. Click **tank01** to select it. The cursor will change, indicating the symbol is now ready to be placed.



7. Click on the tab of the Main screen in the Workspace.
8. Click on any point to insert the Symbol.

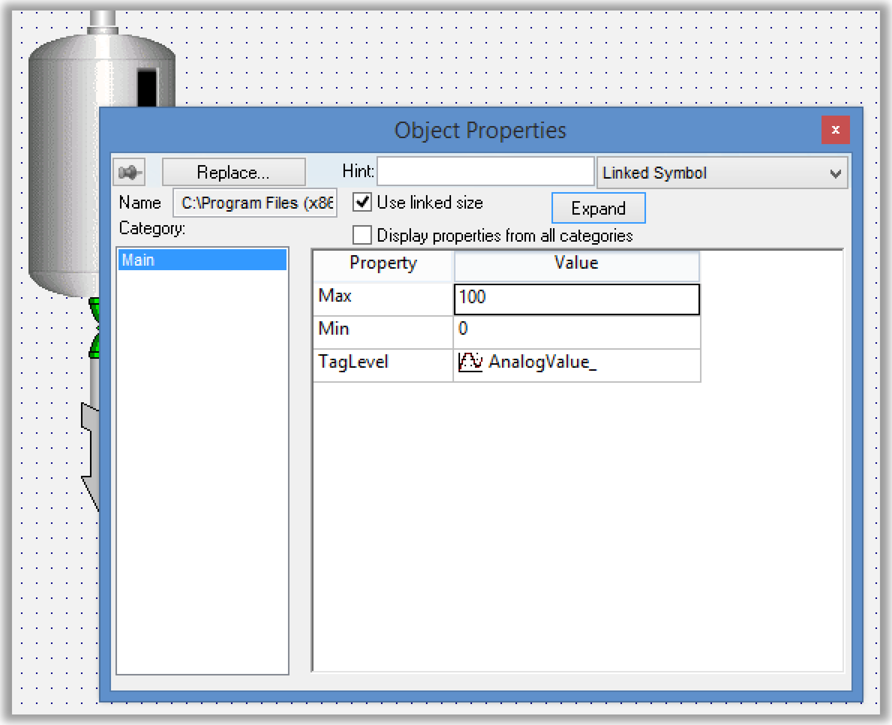


9. Repeat these steps for the following symbols:
    - Arrows\Arrow05
    - Arrows\Arrow07
    - Meters\Meter07 (insert two copies of this symbol)
    - Pipes\Pipe01 (insert two copies of this symbol)
    - Pipes\Pipe16
    - Pipes\Pipe02
    - Valves\Valve05
    - Valves\Valve06
10. Create the following layout on the Main screen using the added symbols.



Remember to use the tools on the Format tab of the Ribbon to move and align objects.

11. Double-click on the **Tank01** symbol to open the Object Properties Window.


Object Properties

Replace...

Hint:

Linked Symbol

Name: C:\Program Files (x86

☑ Use linked size

Expand

Category:

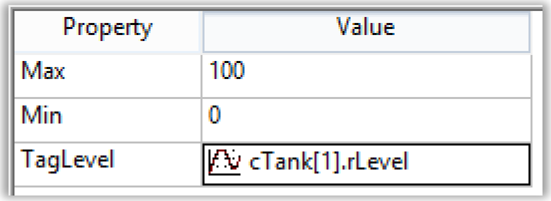
☐ Display properties from all categories

Main

Property	Value
Max	100
Min	0
TagLevel	AnalogValue_

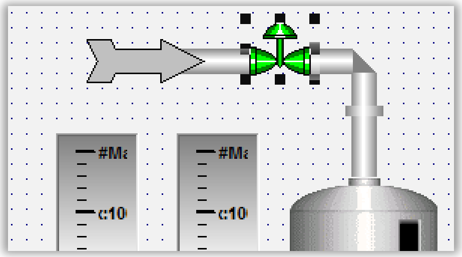
12. The Properties listed are preconfigured **Custom Properties** for the objects in the **Linked Symbol**. These can be assigned to Captions or Animations within the **Symbol**, and can reference static values or even tags.

13. Configure the property values as shown below.

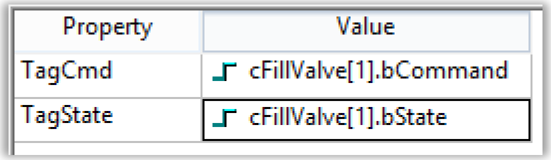


Property	Value
Max	100
Min	0
TagLevel	cTank[1].rLevel

14. Double-click on the **Valve05** symbol to open the Object Properties Window.

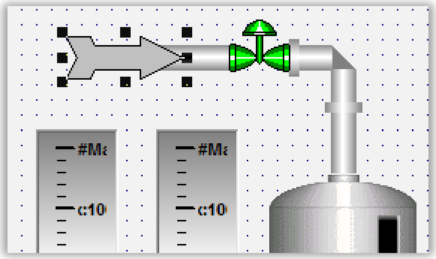


15. Configure the property values as shown below:

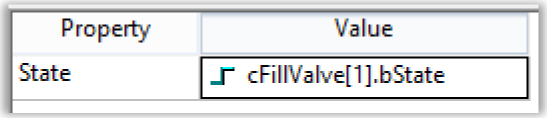


Property	Value
TagCmd	cFillValve[1].bCommand
TagState	cFillValve[1].bState

16. Double-click on the **Arrow05** symbol to open the Object Properties Window.

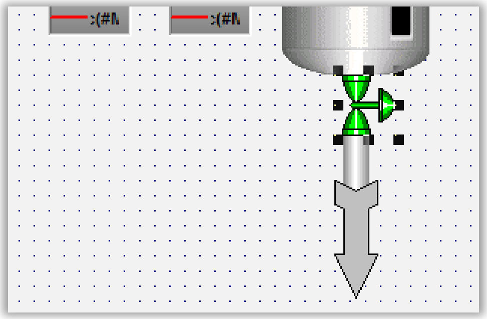


17. Configure the property values as shown below:

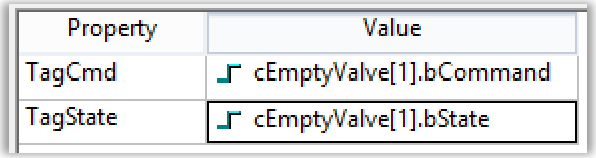


Property	Value
State	cFillValve[1].bState

18. Double-click on the **Valve06** symbol to open the Object Properties Window.

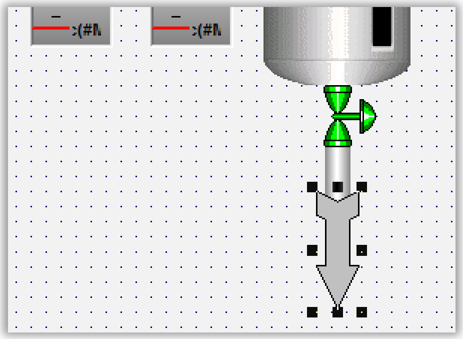


19. Configure the property values as shown below:

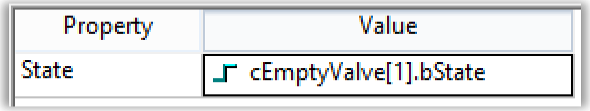


Property	Value
TagCmd	cEmptyValve[1].bCommand
TagState	cEmptyValve[1].bState

20. Double-click on the **Arrow07** symbol to open the Object Properties Window.

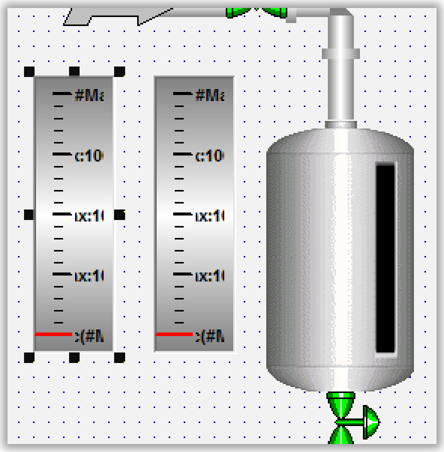


21. Configure the property values as shown below:

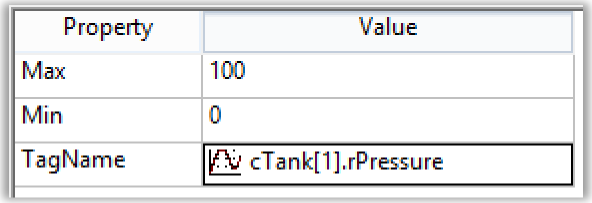


Property	Value
State	cEmptyValve[1].bState

22. Double-click on the first **Meter07** symbol to open the Object Properties Window.

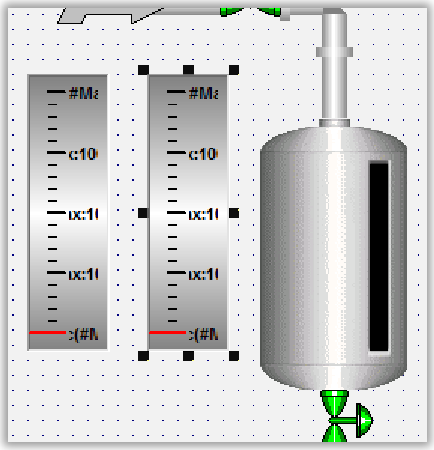


23. Configure the property values as shown below:

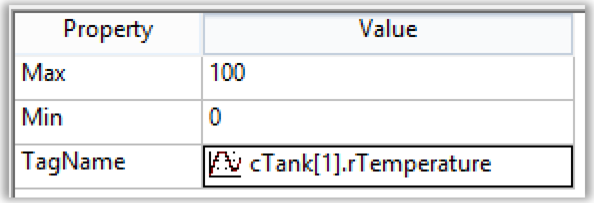


Property	Value
Max	100
Min	0
TagName	cTank[1].rPressure

24. Double-click on the second **Meter07** symbol to open the Object Properties Window.

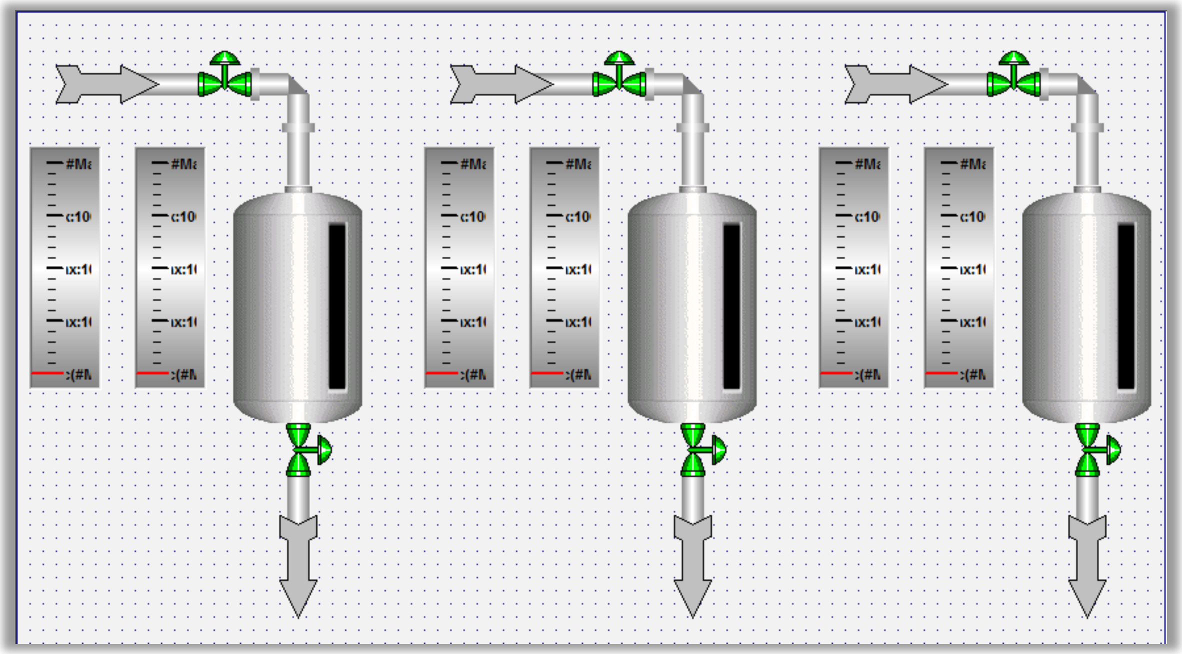


25. Configure the property values as shown below:



Property	Value
Max	100
Min	0
TagName	cTank[1].rTemperature

26. Repeat steps 2 through 25 to create similar layouts for Tanks 2 and 3. When you are done, then the Main screen should look similar to the following:

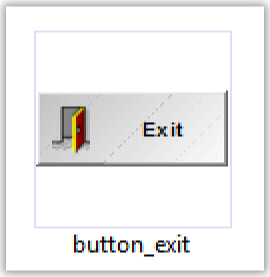


Pro Tip: Use the tools on the Format tab of the Ribbon to group and copy the layout you created. Use the Replace tool to change the tag used from cTank[1] to cTank[2] and cTank[3], cFillValve[1] to cFillValve[2] and cFillValve[3], and cEmptyValve[1] to cEmptyValve[2] and cEmptyValve[3], respectively.

27. Save and close the Main screen.

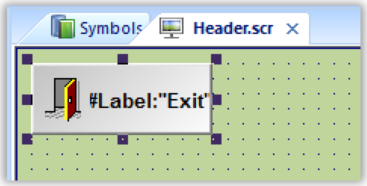
Add Symbols to the Header screen

1. Open the Header screen.
2. In the Libraries group of the Graphics tab of the Ribbon, click on **Symbols**.
3. The Symbols tab will be displayed in the Workspace.
4. Expand the **System Symbols** folder of the System in the menu on the left.
5. Click on the Buttons folder.
6. Click **button_exit** to select it. The cursor will change, indicating the symbol is now ready to be placed.



7. Click on the tab of the Header screen in the Workspace.

8. Click on a point near the upper left of the screen to insert the Symbol.



9. Save and close the Header screen.

10. If your runtime is on a remote machine, then use the the Remote Management tools on the Home tab of the Ribbon to download your project to the remote machine. Start the runtime and test.

SECTION 4 - SHAPES

Using a step-by-step tutorial, this section explains how to insert active objects, insert shapes, and formatting options that will be used for the training course.

On the HEADER screen, you will place title text objects for the project and image with the company's logo. You will also use the Rectangle Shape to display Date and Time.

Concept

The Shapes allow the user to create polygons, rectangles, lines and other objects for their screen, personalizing it.

BLUE Open Studio supports the following shape types:

- Line: draws a single line.
- Open Polygon: draws an open polygon.
- Closed Polygon: draws a closed polygon, with borders and/or fill.
- Rectangle: creates rectangles which supports fill, borders and/or captions.
- Rounded Rectangle: draws rounded rectangles.
- Ellipse: draws ellipses, circles, rings, arcs and chords.

SECTION 5 - ACTIVE OBJECTS

This section defines Active Objects and explains how to insert and configure those objects onto a screen.

Concept

Active Objects allow you to create interactive objects on the screen, with which you can add an interactive button, display text, make selections, and a variety of other options. Most options allow you to read and write values from tags as well as expressions.

Overview

BLUE Open Studio has nine basic Active Objects:

- **Text**: Creates text objects in order to display a single line of text. This object also supports the Text Data Link animation, allowing you to display tag/expression values, as well as write tag values.
- **Text Box**: A text input/output box in OS style that can be configured to show several lines.
- **Button**: Creates custom buttons specifically defined for use with the command animation.
- **Pushbutton**: Creates an object with pre-configured Momentary, Maintained or Locked events defined as an On-Down and On-Up event.
- **Checkbox**: Creates a checkbox that indicates a TRUE/FALSE value.
- **Radio Button**: Creates interfaces from which users can view and select between several options on the screen.
- **Combo Box**: Creates a drop-down combobox, allowing you to select an item from a list.
- **List box**: Displays a list of messages tied to a tag/expression value, or may be used to write a value to a tag by selecting a message.
- **Smart Message**: Displays messages that can be changed by the user or by value during execution, and can even update a tag’s value. Message Viewer. The Smart Message may be configured as a Message Display, Multi-state Indicator, or Multi-state Push Button.

LAB 10 - USING ACTIVE OBJECTS

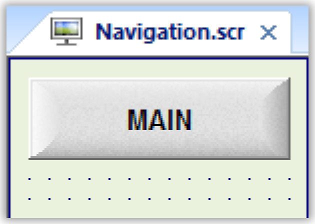
In this lab you will create control to navigate to other screens that will be created in subsequent labs.

Preparation

1. Open the Template screen.
2. Click the Application button and click **Save As** on the Application menu. The Save As dialog is opened.
3. In the **File name** field, enter `Active Objects`.
4. Click **Save**. The Save As dialog is closed.
5. Close the Active Objects screen.

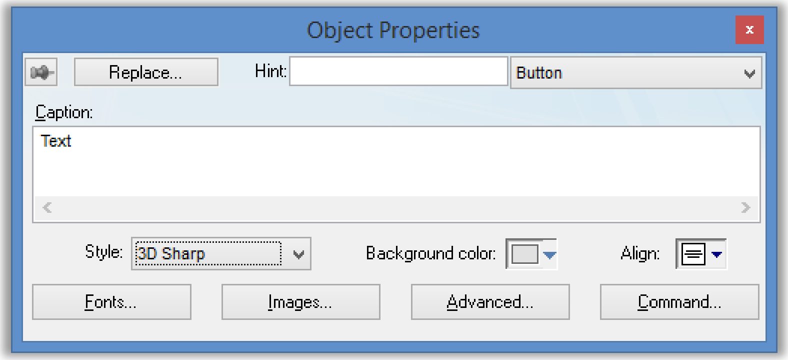
Button

1. Open the NAVIGATION screen.
2. On the Graphics tab of the Ribbon in the Active Objects group, select **Button**.
3. Click and drag on the NAVIGATION screen to shape the button as shown below:



The button should have a height of not more than 41 pixels.

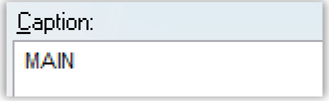
4. Double-click on the button to open the Object Properties window.



Caption:
Text

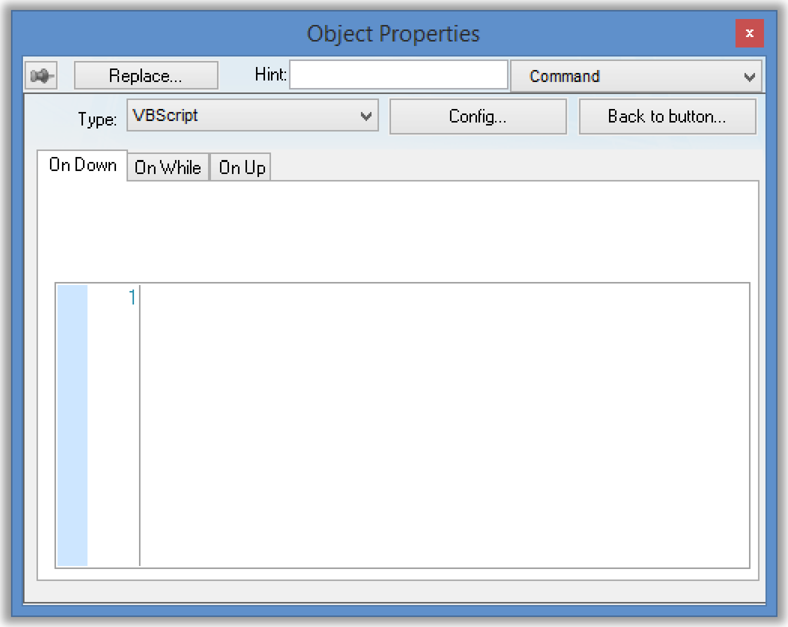
Style: 3D Sharp

6. In Caption field, type MAIN.

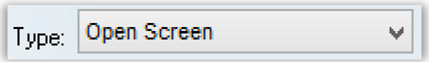


Caption:
MAIN

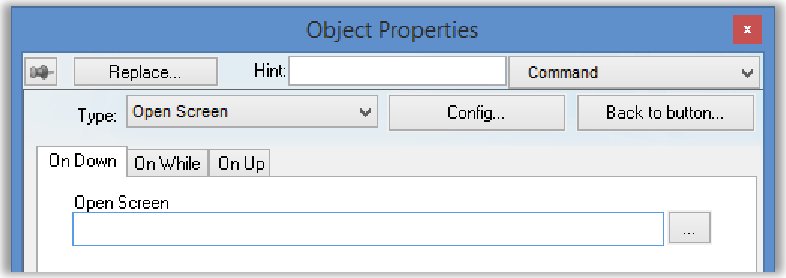
7. Click Command. The Command object properties is opened.



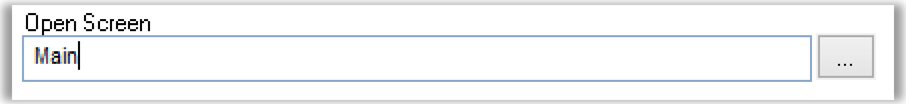
8. In the **Type** field, select Open Screen.



The Comand object properties changes as shown below.



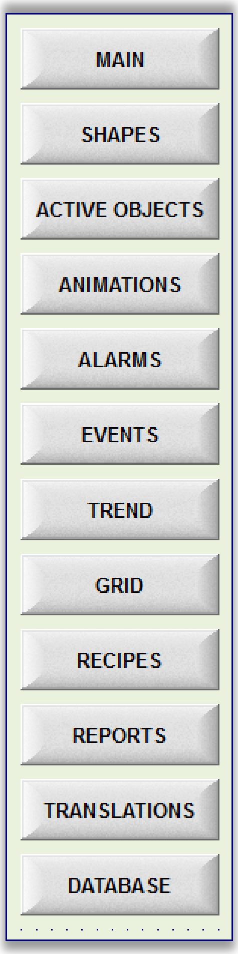
9. In **Open Screen**, enter `Main`.



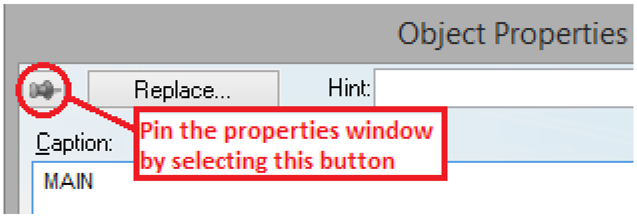
10. Repeat steps 2 through 6 and changing MAIN to SHAPES, ACTIVE OBJECTS, ANIMATIONS, ALARMS, EVENTS, TREND, GRID, RECIPES, REPORTS, TRANSLATIONS, and DATABASE, respectively.

11. You can insert new buttons, repeating stages 2-6; or copy the object, using **Ctrl+C** and **Ctrl+V** option, or press and hold the Ctrl key, then click and drag.

12. At the end of the process, you will have the following buttons on the NAVIGATION screen.



Pro Tip: When creating multiple copies of an object, you can pin the properties window to the workspace to keep the dialog open. This will always display the properties of the currently selected object, allowing you to switch quickly between objects.



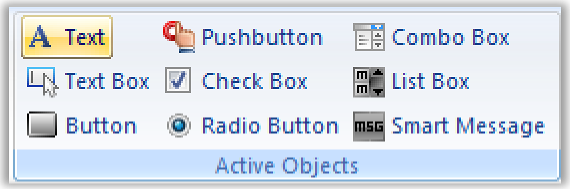
Pin the properties window by selecting this button

13. Save and close the NAVIGATION screen.

Text Object

1. Open the ACTIVE OBJECTS screen.

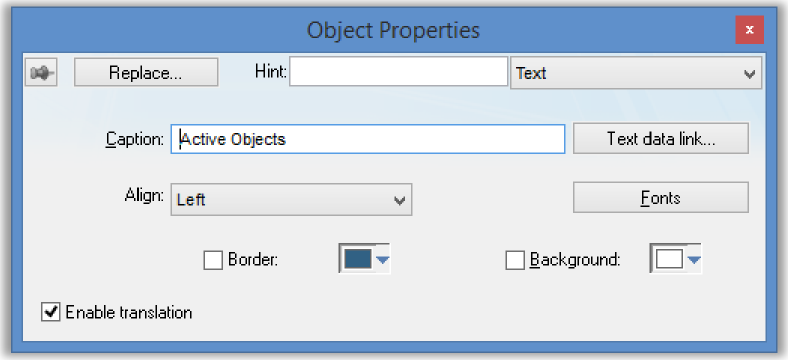
2. In the Graphics tab of the Ribbon menu, click on the **Text** option in the Active Objects group.



3. Click on a point on the screen near the top of the screen to start writing. A flashing cursor will appear.

4. Type `Active Objects`.

5. Double-click on the text to open the Object Properties window.



Hint:
Text

Caption: Active Objects

Align: Left

☐ Border:

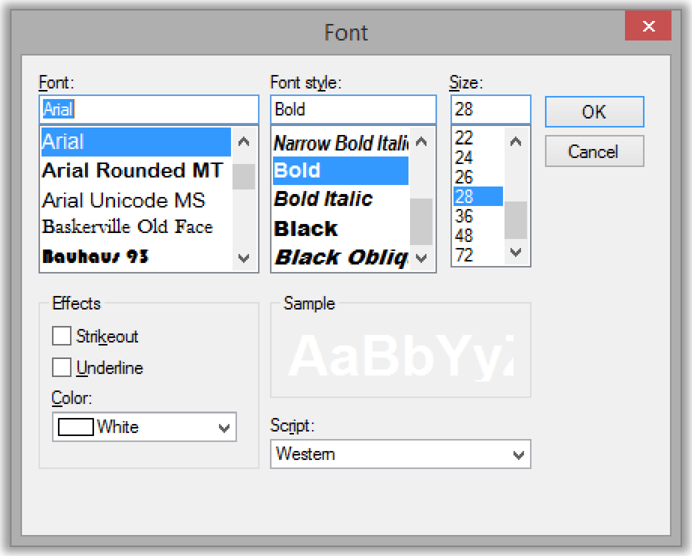
☐ Background:

☑ Enable translation

6. Click on the **Fonts** button.

7. The Fonts window will be displayed.

8. Use the dialog to configure the font to Arial, Bold, size 28, color White.



Font: Arial

Font style: Bold

Size: 28

☐ Strikeout

☐ Underline

Color: White

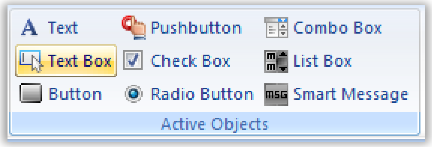
Script: Western

9. Click **OK**.

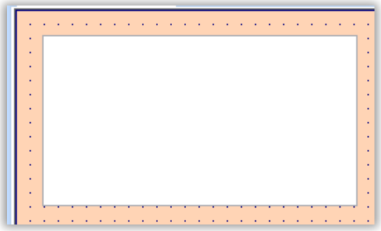
10. Close the Object Properties window.

Text Box

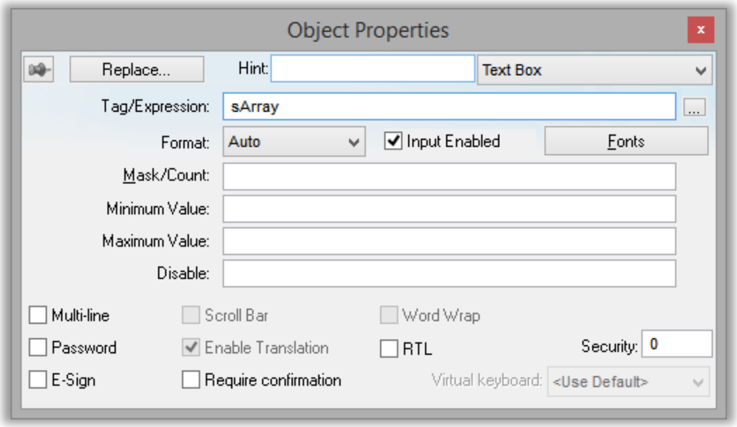
1. In the Graphic Ribbon, select **Textbox** from the Active Objects group.



2. Click and drag on the ACTIVE OBJECTS screen to shape the Textbox.



3. Double-click on the Text Box to open the Object Properties window.



Object Properties

Hint:
Text Box

Tag/Expression: sArray

Format: Auto

☑ Input Enabled

Mask/Count:

Minimum Value:

Maximum Value:

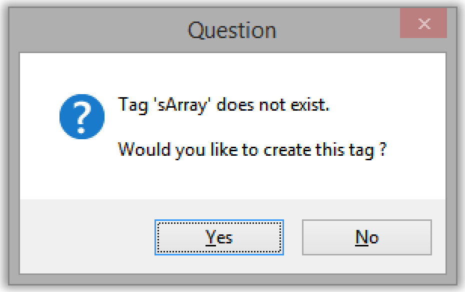
Disable:

☐ Multi-line ☐ Scroll Bar ☐ Word Wrap

☐ Password ☑ Enable Translation ☐ RTL Security: 0

☐ E-Sign ☐ Require confirmation Virtual keyboard: <Use Default>

4. Under Tag/Expression, type "**sArray**" as above and click another location on the dialog. Since this tag name does not exist, BLUE Open Studio will prompt you to create the tag.



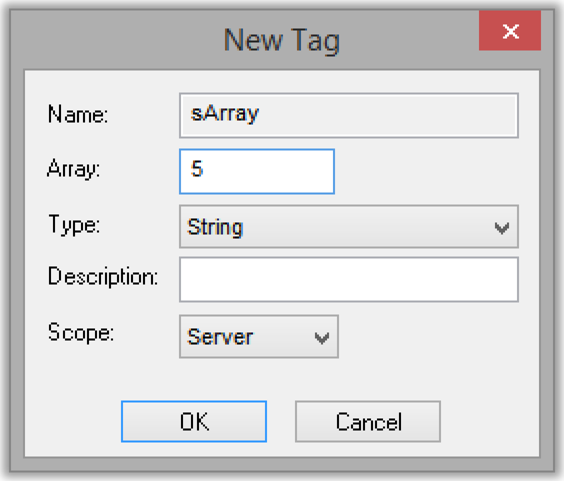
Question

Tag 'sArray' does not exist.

Would you like to create this tag ?

5. Select **Yes**.

6. The Create New Tag dialog is opened.


New Tag

Name: sArray

Array: 5

Type: String

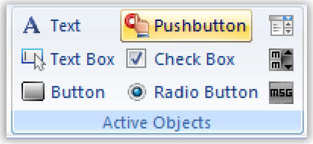
Description:

Scope: Server

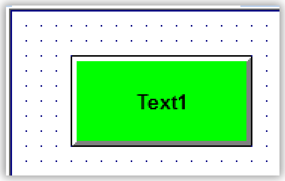
7. Assign the Array Size of **5** and the Type to **String**.
8. Select **OK** to create the tag.
9. Back on the Object Properties dialog, select the **Multi-line** and **Word Wrap** checkboxes.
10. Close the Object Properties window.

Pushbutton

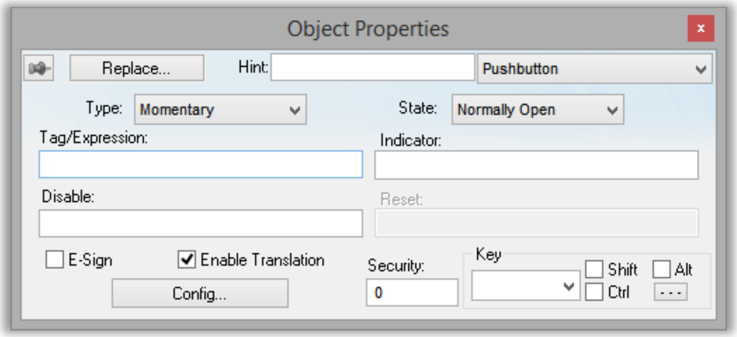
1. On the Graphics tab of the Ribbon in the Active Objects group, select **Pushbutton**.



2. Click and drag on the ACTIVE OBJECTS screen to shape the Pushbutton as shown below.



3. Double-click on the Pushbutton to open the Object Properties window.



Hint:

Pushbutton

Type: Momentary

State: Normally Open

Tag/Expression:

Indicator:

Disable:

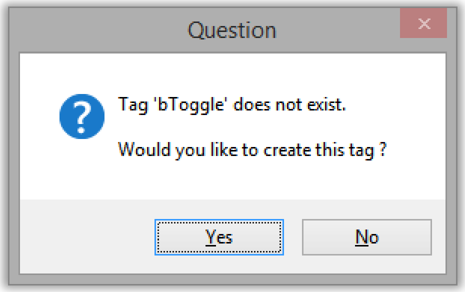
Reset:

☐ E-Sign ☑ Enable Translation

Security: 0

☐ Shift ☐ Alt ☐ Ctrl

4. Under Tag/Expression, type “**bToggle**” and click another location on the dialog. Since this tag name does not exist, BLUE Open Studio will prompt you to create the tag.

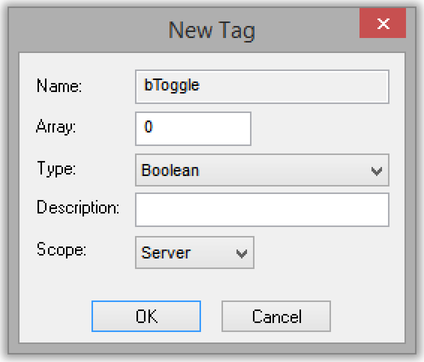


Tag 'bToggle' does not exist.

Would you like to create this tag ?

5. Select **Yes**.

6. The Create New Tag dialog is opened.



Name: bToggle

Array: 0

Type: Boolean

Description:

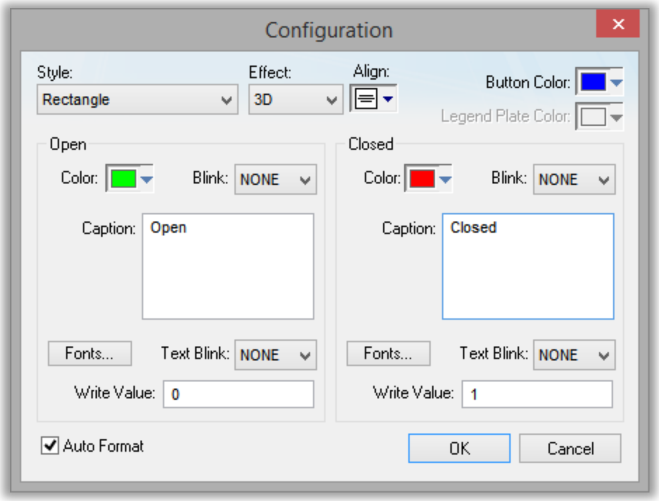
Scope: Server

7. Assign the Type to **Boolean**.

8. Select **OK** to create the tag.

9. Back on the Object Properties dialog, add the tag **bToggle** in the Indicator field.

10. Click the Config button to open the Configuration dialog.



Configuration

Style: Rectangle
Effect: 3D
Align:
Button Color: 🟦
Legend Plate Color:

Open
Color: 🟩 Blink: NONE
Caption: Open
Fonts... Text Blink: NONE
Write Value: 0

Closed
Color: 🟥 Blink: NONE
Caption: Closed
Fonts... Text Blink: NONE
Write Value: 1

☑ Auto Format

OK Cancel

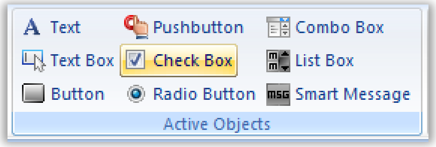
11. Change the Caption in the Open section to “**Open**” and the Caption on the Closed section to “**Closed**” as above.

12. Click **OK** to return to the Object Properties window.

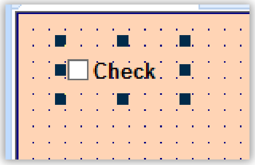
13. Close the Object Properties window.

Checkbox

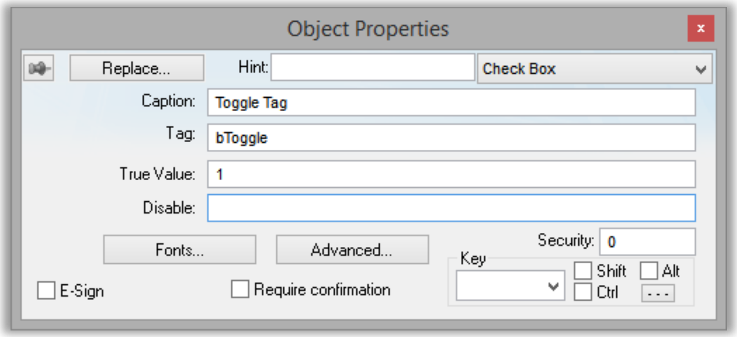
1. On the Graphics tab of the Ribbon in the Active Objects group, select **Checkbox**.



2. Click and drag on the ACTIVE OBJECTS screen to shape the Checkbox.



3. Double-click on the object to open the Object Properties window.



Hint:

Check Box

Caption: Toggle Tag

Tag: bToggle

True Value: 1

Disable:

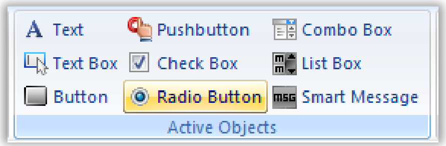
Security: 0

4. Under Caption, type “**Toggle Tag**” and under the Tag field type “**bToggle**”.

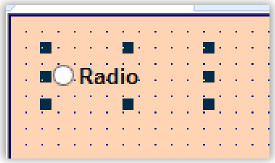
5. Close the Object Properties window.

Radio button

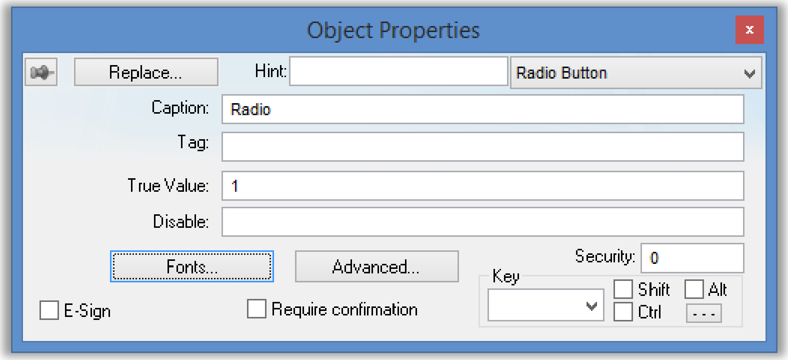
1. On the Graphics tab of the Ribbon in the Active Objects group, select **Radio Button**.



2. Click and drag in the ACTIVE OBJECTS screen to shape the Radio button.



3. Double-click on the object to open the Object Properties window.



Hint:

Radio Button

Caption: Radio

Tag:

True Value: 1

Disable:

Security: 0

4. Under Caption, type “**Integer Value 1,**” under the Tag field type “**iVar1**” and for the True Value select **1** as above.

5. Close the Object Properties window.

6. Create four additional radio buttons, with **iVar1** as the Tag, but with the Caption and True Value showing values for the numbers 2-5 instead. For example, the second radio's caption will be "**Integer Value 2**" and the True Value will be **2**. Repeat with:

   Caption: **Integer Value** [n]

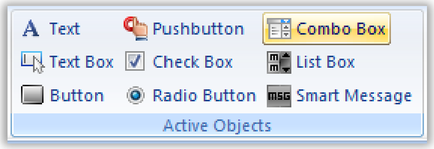
   True Value: [n]

   With [n] being 2 through 5. The Tag field will not change.

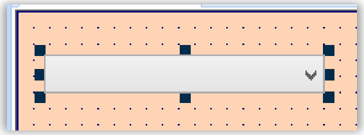
7. Close the Object Properties window.

Combo Box

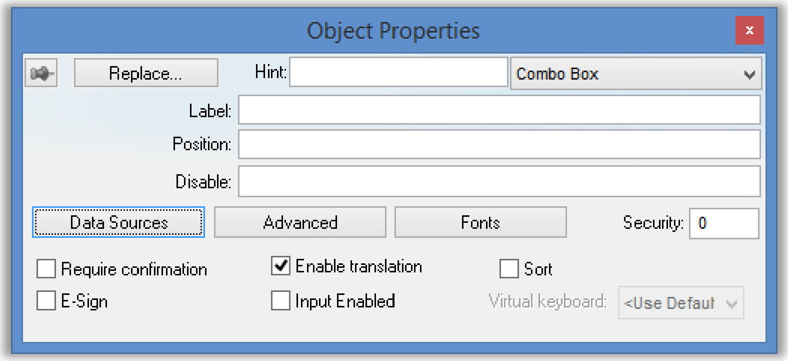
1. On the Graphics tab of the Ribbon in the Active Objects group, select **Combo Box**.



2. Click and drag on the ACTIVE OBJECTS screen to shape the Combo Box.



3. Double-click on the Combo Box to open the Object Properties window.


Object Properties

Replace... Hint: Combo Box

Label:

Position:

Disable:

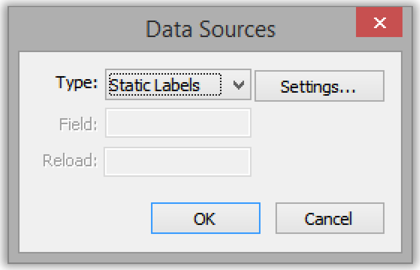
Data Sources | Advanced | Fonts | Security: 0

□ Require confirmation ☑ Enable translation □ Sort

□ E-Sign □ Input Enabled Virtual keyboard: <Use Defaut

4. In the field called Position, enter **iVar1**.

5. Click the Data Sources button. This will open the Data Sources dialog.



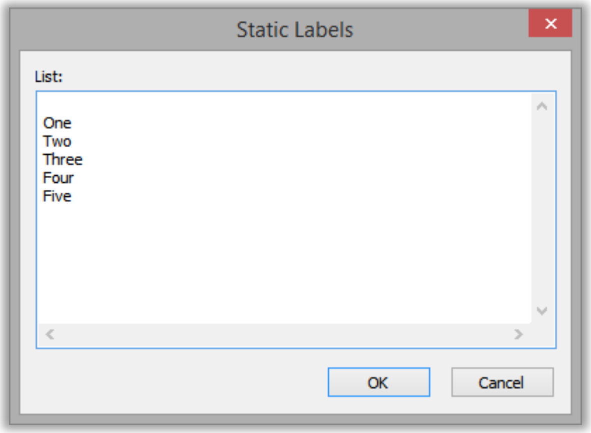
Data Sources

Type: Static Labels

Field:

Reload:

6. For Type, type **Static Labels.** Click the Settings button to enter the text that will be displayed in the combobox object. Type the numbers one through five as shown below, leaving the first line blank.



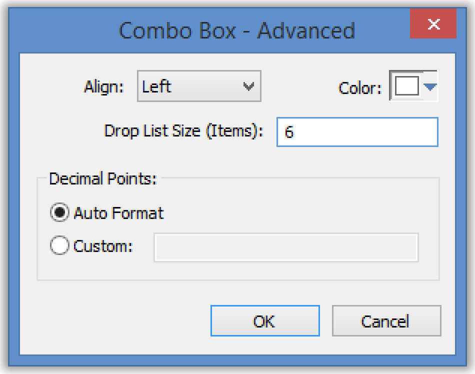
Static Labels

List:

One
Two
Three
Four
Five

7. Click **OK** to return to the Data Sources dialog, then click **OK** to return to the Object Properties window.

8. Click the button called Advanced to open the advanced object properties and change the field called Drop List Size to 6 as shown below.



Combo Box - Advanced

Align: Left

Color: □

Drop List Size (Items): 6

Decimal Points:

● Auto Format

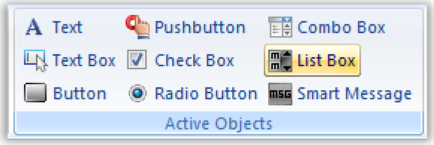
○ Custom:

9. Click **OK** to return to the Object Properties window

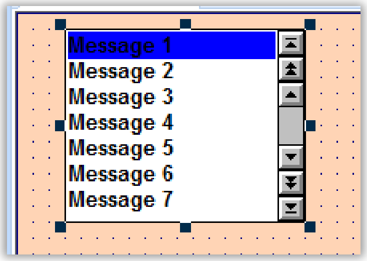
10. Close the Object Properties window.

List Box

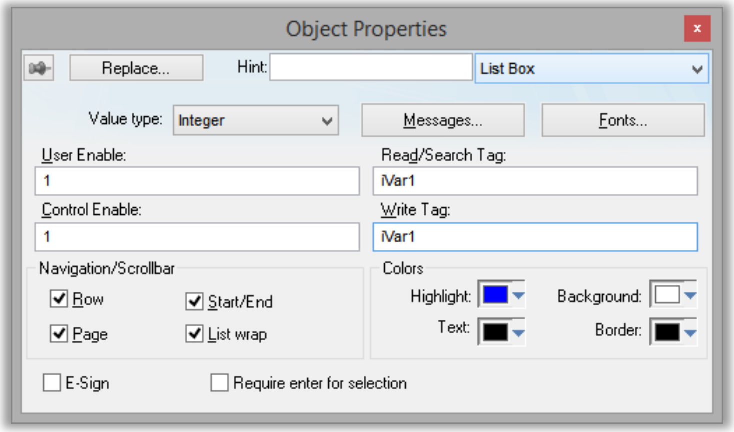
1. On the Graphics tab of the Ribbon in the Active Objects group, select **List Box**.



2. Click and drag on the ACTIVE OBJECTS screen to shape the List Box.



3. Double-click on the List Box to open the Object Properties window.



Object Properties

Replace... Hint: List Box

Value type: Integer Messages... Fonts...

User Enable: 1

Read/Search Tag: iVar1

Control Enable: 1

Write Tag: iVar1

Navigation/Scrollbar

☑ Row ☑ Start/End

☑ Page ☑ List wrap

Colors

Highlight: 🟦 Background: □

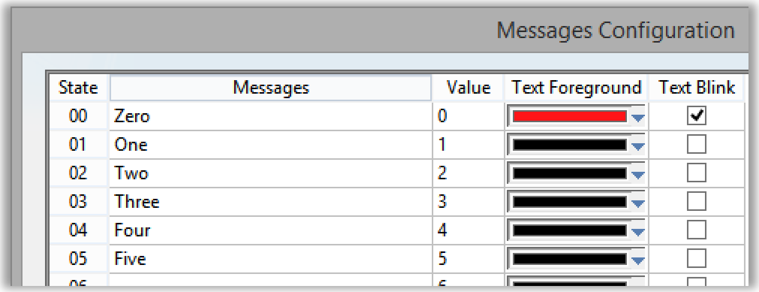
Text: ■ Border: ■

□ E-Sign □ Require enter for selection

4. Under both the Read/Search Tag and the Write Tag fields, enter the **iVar1** tag as above.

5. For Value Type, select **Integer**.

6. Click the Messages button. This will open the Messages Configuration dialog.



Messages Configuration

State	Messages	Value	Text Foreground	Text Blink
00	Zero	0	🟥	☑
01	One	1	■	□
02	Two	2	■	□
03	Three	3	■	□
04	Four	4	■	□
05	Five	5	■	□
06		6	■	□

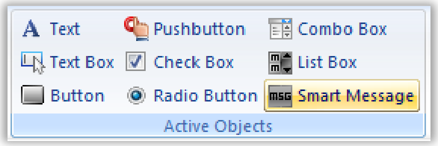
7. For State “00”, type **Zero** as the Message, **“0”** as the Value, set the Text Foreground to Red, and select the Text Blink checkbox. For States 01 through 05, under Messages type the name of the number as above.

8. Click OK to return to the Object Properties window.

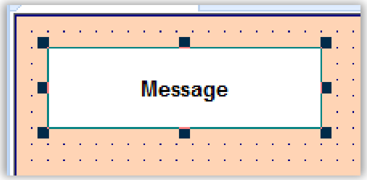
9. Close the Object Properties window.

Smart Message

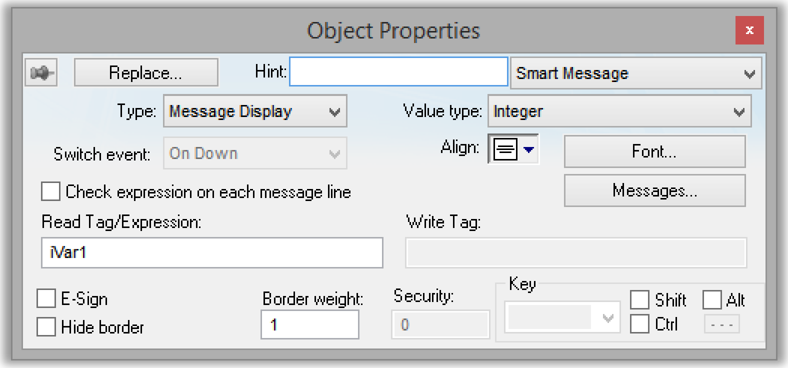
1. On the Graphics tab of the Ribbon in the Active Objects group, select **Smart Message**.



2. Click and drag on the ACTIVE OBJECTS screen to shape the Smart Message.



3. Double-click on the Smart Message to open the Object Properties window.



Object Properties

Replace... Hint: Smart Message

Type: Message Display

Value type: Integer

Switch event: On Down

Align:

Font...

□ Check expression on each message line

Messages...

Read Tag/Expression: iVar1

Write Tag:

□ E-Sign

□ Hide border

Border weight: 1

Security: 0

Key

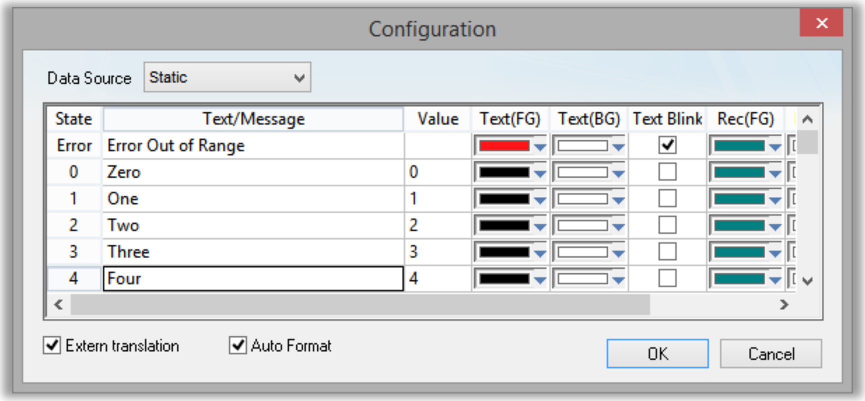
□ Shift □ Alt

□ Ctrl

4. For Type, select **Message Display** and for Value Type, select **Integer**.

5. Under the Read Tag/Expression, enter the **iVar1** tag as above.

6. Click the Messages button. This will open the Configuration dialog.



Data Source: Static

State	Text/Message	Value	Text(FG)	Text(BG)	Text Blink	Rec(FG)
Error	Error Out of Range		🟥	⬜	☑	🟩
0	Zero	0	⬛	⬜	☐	🟩
1	One	1	⬛	⬜	☐	🟩
2	Two	2	⬛	⬜	☐	🟩
3	Three	3	⬛	⬜	☐	🟩
4	Four	4	⬛	⬜	☐	🟩

☑ Extern translation ☑ Auto Format

7. For State “Error”, type **“Error Out of Range”** as the Text/Message, set the Text(FG) to Red, and select the Text Blink checkbox. For States 0 through 5, under Text/Message type the name of the number as above.

8. Click OK to return to the Object Properties window.

9. Close the Object Properties window.

10. Save and close the ACTIVE OBJECTS screen.

11. If your runtime is on a remote machine, then use the the Remote Management tools on the Home tab of the Ribbon to download your project to the remote machine. Start the runtime and test.

SECTION 6 - ANIMATIONS

This section explains, using a step-by-step tutorial, how to insert animations on screen objects.

Concept

Animations can be applied to a screen object or even to a group of objects. Animations allow you to modify the properties of the object during execution according to the tag or expression values. Multiple animations may be added onto the same object, provided the object supports the desired animation.

Overview

BLUE Open Studio has the following 8 types of animations:

- **Command:** allows you to use click or key events on the object to execute a command in runtime.
- **Hyperlink**: allows you to click on the object during execution to execute the standard browser and to open the configured URL, or to open a file in the default application for that filetype.
- **Bargraph**: allows users to configure an object as a bargraph filling the object with color either horizontally or vertically.
- **Text Data Link**: an animation to text objects that allow users to insert values to a tag and display tag/expression values in runtime.
- **Color**: allows you to modify the color of an object during runtime based on the value of a tag or expression.
- **Visibility/Position**: allows users to move an object horizontally and/or vertically based on a tag value, or to even use the object's position to write values to a tag. This animation also allows you to display or hide an object dynamically during runtime.
- **Resize**: allows users to increase or decrease the size of an object based on a tag/expression value.
- **Rotation**: allows users to rotate the object on the screen at a defined pivot point based on a tag/expression value.

LAB 11 - USING ANIMATIONS

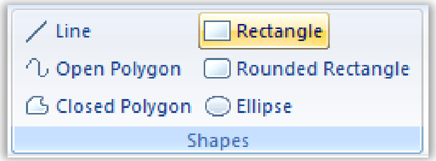
In this lab you will configure the browse buttons of the NAVIGATION screen to open the screens relating to the captions. You will also configure animations on other objects to visualize tag data.

Preperation

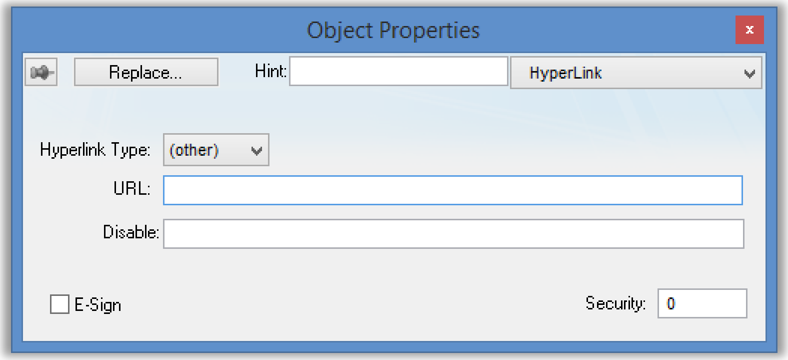
1. Open the Template screen.
2. Click the Application button and click **Save As** on the Application menu. The Save As dialog is opened.
3. In the **File name** field, enter `Animation`.
4. Click **Save**. The Save As dialog is closed.

Hyperlink

1. On the Graphics tab of the Ribbon in the Shapes group, select **Rectangle**.



2. Click and drag on the ANIMATION screen to shape the object.
3. Double-click on the Rectangle to open the Object Properties window.
4. In the Caption, type **HYPERLINK**.
5. With the object selected, on the Graphics tab of the Ribbon under the Animations group, select **Hyperlink**.
6. Double-click on the rectangle to open the Hyperlink animation properties window.



Hint:

HyperLink

Hyperlink Type: (other)

URL:

Disable:

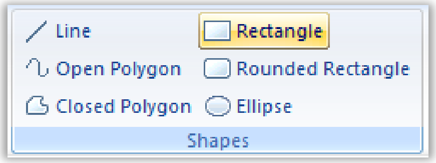
□ E-Sign

Security: 0

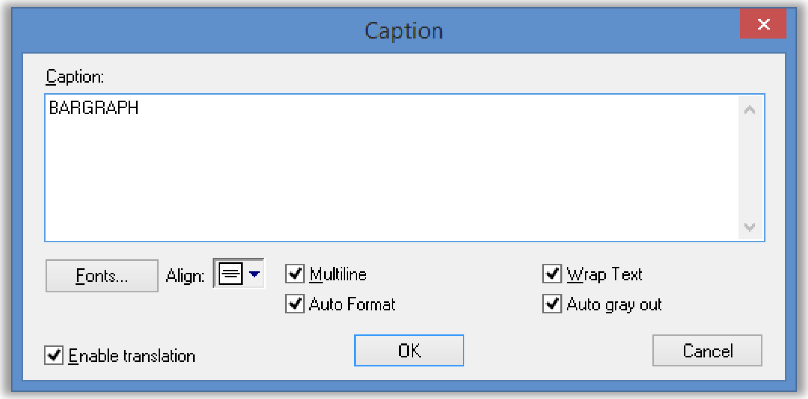
7. Under Hyperlink Type, select **http:**
8. Under URL, type some URL address.
9. Close the Object Properties window.

Bargraph

1. On the Graphics tab of the Ribbon in the Shapes group, select **Rectangle**.



2. Click and drag on the ANIMATION screen to shape the object.
3. Double-click on the Rectangle to open the Object Properties window.
4. In the Caption, type **BARGRAPH** as shown below.



Caption

Caption:

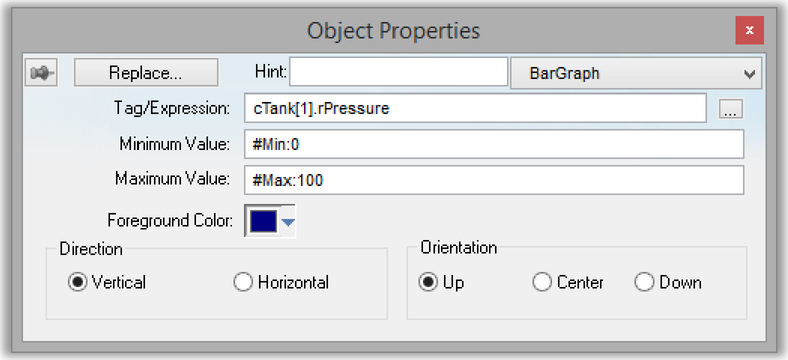
BARGRAPH

Fonts... Align:

☑ Multiline
☑ Wrap Text
☑ Auto Format
☑ Auto gray out
☑ Enable translation

OK Cancel

5. With the object selected, on the Graphics tab of the Ribbon under the Animations group, select **Bargraph**.
6. Double-click on the rectangle to open the Bargraph animation properties window.
7. Under Tag/Expression, use the tag **cTank[1].rPressure**
8. Change Foreground color to **Red.**
9. Under Direction and Orientation, select **Vertical** and **Up.**



Object Properties

Replace... Hint: BarGraph

Tag/Expression: cTank[1].rPressure

Minimum Value: #Min:0

Maximum Value: #Max:100

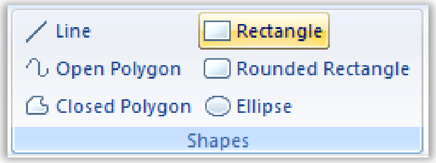
Foreground Color:

Direction: ● Vertical ○ Horizontal

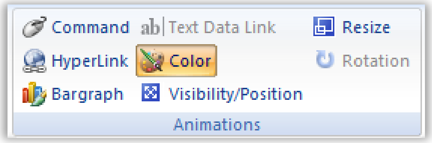
Orientation: ● Up ○ Center ○ Down

Color

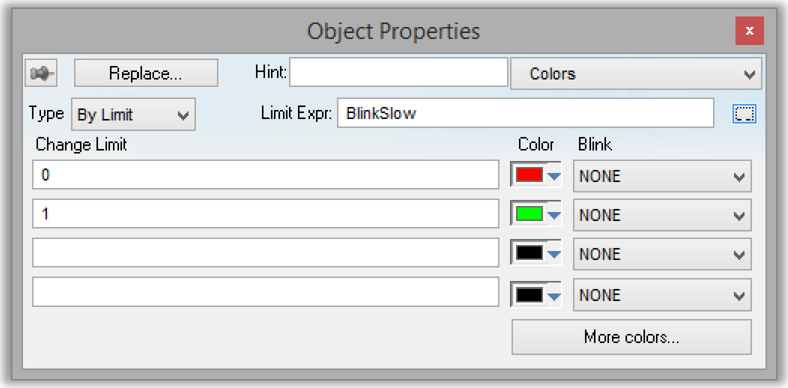
1. On the Graphics tab of the Ribbon in the Shapes group, select **Rectangle**.



2. Click and drag on the ANIMATION screen to shape the object.
3. Double-click on the Rectangle to open the Object Properties window.
4. In the Caption, type **COLOR**.
5. With the object selected, on the Graphics tab of the Ribbon under the Animations group, select **Color**.



6. Double-click on the rectangle to open the Color animation properties window.
7. Select Type **By Limit**.
8. Under Limit Expr:, use the system tag **BlinkSlow**.



Object Properties

Replace... Hint: Colors

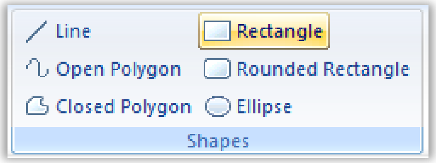
Type: By Limit Limit Expr: BlinkSlow

Change Limit	Color	Blink
0	🟥	NONE
1	🟩	NONE
	⬛	NONE
	⬛	NONE

More colors...

Visibility

1. On the Graphics tab of the Ribbon in the Shapes group, select **Rectangle**.

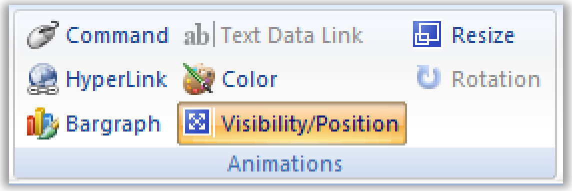


2. Click and drag on the ANIMATION screen to shape the object.

3. Double-click on the Rectangle to open the Object Properties window.

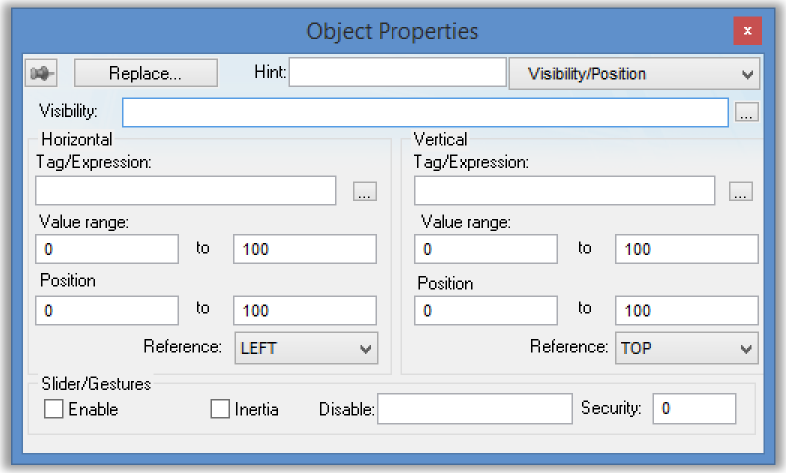
4. In the Caption, type **VISIBILITY**.

5. With the object selected, on the Graphics tab of the Ribbon under the Animations group, select **Visibility/Position**.



6. Double-click on the rectangle to open the Visibility/Position animation properties window.

7. Under Visibility, use the System Tag **BlinkSlow** as shown below.



Object Properties

Replace... Hint: Visibility/Position

Visibility:

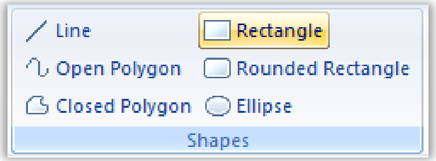
Horizontal
Tag/Expression:
Value range: 0 to 100
Position: 0 to 100
Reference: LEFT

Vertical
Tag/Expression:
Value range: 0 to 100
Position: 0 to 100
Reference: TOP

Slider/Gestures
□ Enable □ Inertia Disable: Security: 0

Position

1. On the Graphics tab of the Ribbon in the Shapes group, select **Rectangle**.

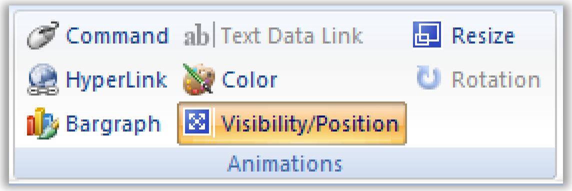


2. Click and drag to create a Rectangle with a width and height of 201.

3. Double-click on the Rectangle to open the Object Properties window.

4. In the Caption, type **POSITION**.

5. With the object selected, on the Graphics tab of the Ribbon under the Animations group, select **Visibility/Position**.



6. Double-click on the rectangle to open the Visibility/Position animation properties window.

7. Under Horizontal, configure the fields as follows:

8. Tag/Expression: **cTank[2].rTemperature**

9. Value Range: **0** to **100**

10. Position: **0** to **200**

11. Reference: **LEFT**

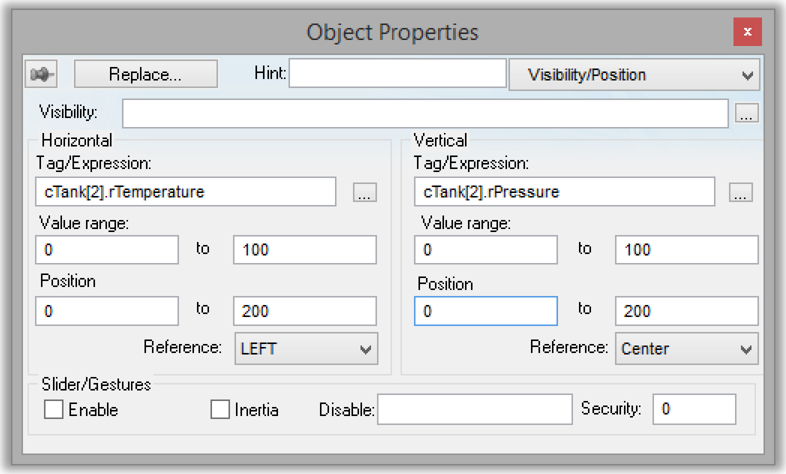
12. Under Vertical, configure the fields as follows:

13. Tag/Expression: **cTank[2].rPressure**

14. Value Range: **0** to **100**

15. Position: **0** to **200**

16. Reference: **CENTER**



Hint:
Visibility/Position
Visibility:

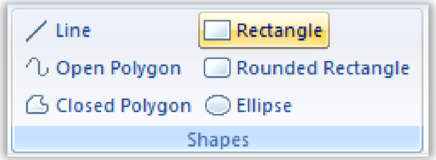
Horizontal
Tag/Expression: cTank[2].rTemperature
Value range: 0 to 100
Position 0 to 200
Reference: LEFT

Vertical
Tag/Expression: cTank[2].rPressure
Value range: 0 to 100
Position 0 to 200
Reference: Center

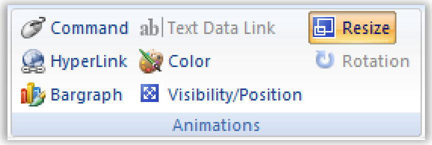
Slider/Gestures
□ Enable □ Inertia Disable: Security: 0

Resize

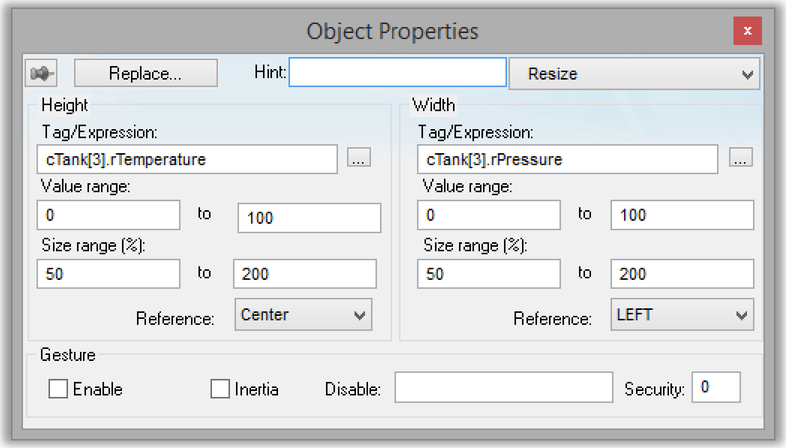
1. On the Graphics tab of the Ribbon in the Shapes group, select **Rectangle**.



2. Click and drag on the ANIMATION screen to shape the object.
3. Double-click on the Rectangle to open the Object Properties window.
4. In the Caption, type **RESIZE**.
5. With the object selected, on the Graphics tab of the Ribbon under the Animations group, select **Resize**.



6. Double-click on the rectangle to open the Resize animation properties window.
7. Under Height, configure the fields as follows:
8. Tag/Expression: **cTank[3].rTemperature**
9. Value Range: **0** to **100**
10. Size Range (%): **50** to **200**
11. Reference: **CENTER**
12. Under Width, configure the fields as follows:
13. Tag/Expression: **cTank[3].rPressure**
14. Value Range: **0** to **100**
15. Size Range (%): **50** to **200**
16. Reference: **LEFT**



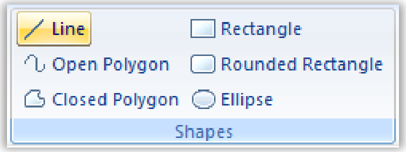
Height
Tag/Expression: cTank[3].rTemperature
Value range: 0 to 100
Size range (%): 50 to 200
Reference: Center

Width
Tag/Expression: cTank[3].rPressure
Value range: 0 to 100
Size range (%): 50 to 200
Reference: LEFT

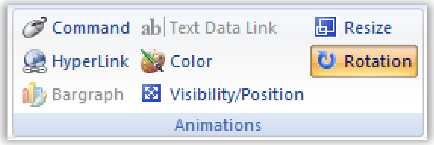
Gesture
□ Enable □ Inertia Disable: Security: 0

Rotation

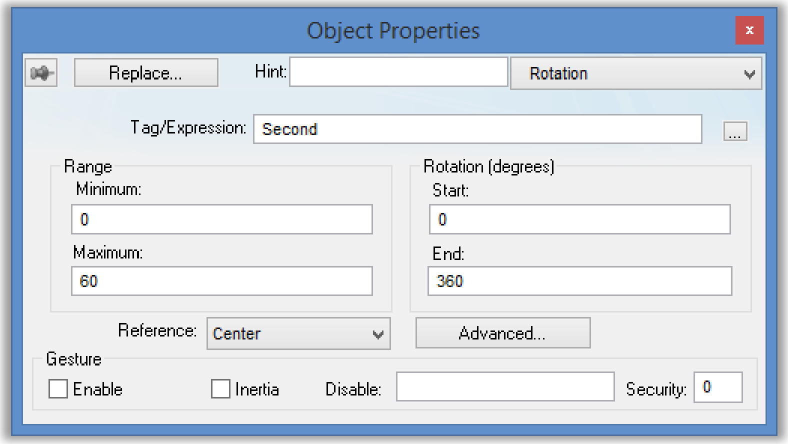
1. On the Graphics tab of the Ribbon in the Shapes group, select **Line**.



2. Click on the ANIMATION screen to start the line. Right-click another point on the screen to complete the object.
3. Double-click on the Line to open the Object Properties window.
4. Under Color, select **Red.**
5. Under weight, change the value to **3**.
6. With the object selected, on the Graphics tab of the Ribbon under the Animations group, select **Rotation**.



7. Double-click on the line to open the Rotation animation properties window.
8. For Tag/Expression, use the System Tag **Second**.
9. Under Range, configure a Minimum of **0** and a Maximum of **60**.
10. Under Rotation (degrees), configure a Start of **0** and a End of **360**.
11. Under Reference, select **Center.**


Object Properties

Hint: Rotation

Tag/Expression: Second

Range
Minimum: 0
Maximum: 60

Rotation (degrees)
Start: 0
End: 360

Reference: Center

Gesture
□ Enable □ Inertia Disable: Security: 0

12. Save and close the ANIMATION screen.
13. If your runtime is on a remote machine, then use the the Remote Management tools on the Home tab of the Ribbon to download your project to the remote machine. Start the runtime and test.

SECTION 7 - LIBRARIES

Using a step-by-step tutorial, this section explains how to insert images and create your own symbols to be used by the project.

Concept

Libraries are built-in, user-created, or installed objects, such as symbols or ActiveX controls, and allows users to add external images (linked picture).

Overview

BLUE Open Studio has the following 3 Library options:

- **Symbol Library:** A visual browser for all the built-in or user-created symbols available to be inserted in a screen.
- **ActiveX:** You can use most ActiveX components installed in your computer.
- **Linked picture:** Allows the user to associate an external image with the project so that you can easily reuse the image in your project and/or dynamically change the image during execution.

SECTION 8 - PROJECT SYMBOLS

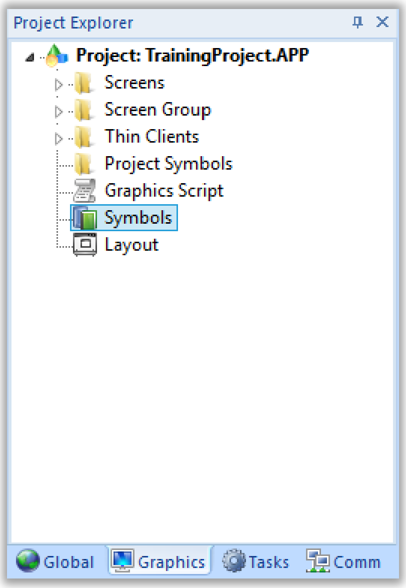
Using a step-by-step tutorial, this section explains how to create your own project symbols.

Concept

As discuss previously, the symbols library contains a large selection of premade symbols that are installed with the BLUE Open Studio development environment. When you use a symbol from the system library, then a local copy of that symbol is stored with your project. You can also create your own symbols.

Overview

A Project Symbol is as a set of one or more objects that you can save and reuse in your projecy. They are organized into the Symbols folders on the Graphics tab of the Project Explorer.



You can also access your Project Symbols by selecting Symbols on the Libraries group of the Graphics tab of the Ribbon.

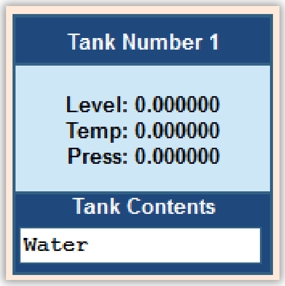
LAB 12 - CREATING YOUR OWN PROJECT SYMBOLS

In order to monitor the tank values, you can create your own **Linked Symbol** that can be used in multiple screen locations. The symbol can be configured with multiple **Custom Properties** to hold static values or even tags. A custom property can be placed anywhere that a value or tag can be placed, such as in a caption or animation.

The syntax for a custom property would be (Items within brackets “[ ]” optional):

#[Category.]PropertyName:[DefaultValue]

You will start by configuring a set of objects as follows:


Tank Number 1

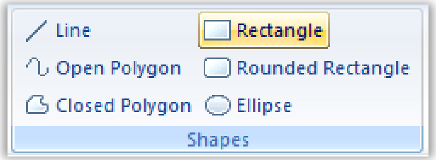
Level: 0.000000
Temp: 0.000000
Press: 0.000000

Tank Contents

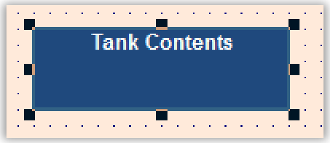
Water

This symbol contains three Rectangles and one Text object

1. Open the Main screen.
2. On the Graphics tab of the Ribbon in the Shapes group, select **Rectangle**.

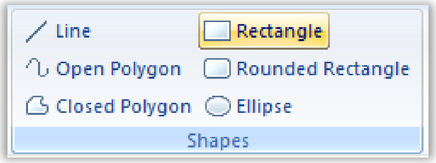


3. Click and drag on the screen to shape the object.
4. Double-click on the rectangle to open the Object Properties window.
5. In the Background section, select Dark Blue.
6. Click on the **Caption** button.
7. Type **Tank Contents** into the field.
8. Under **Align**, select Top Center
9. Click on **Fonts**.
10. In **Font Style**, select Bold.
11. In **Color**, select White.



12. Click **OK**.

13. On the Graphics tab of the Ribbon in the Shapes group, select **Rectangle**.

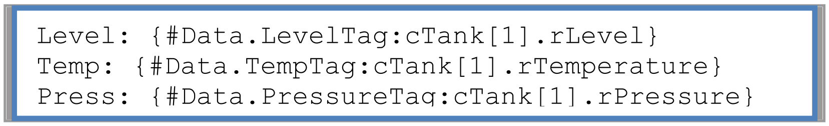


14. Click and drag on the Main screen to shape the object.

15. Double-click on the Rectangle to open the Object Properties window.

16. Click on the **Caption** button.

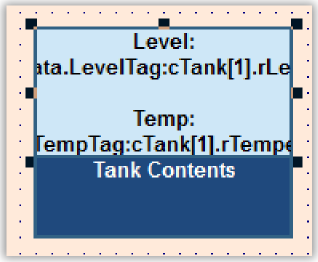
17. Type the following text into the field:



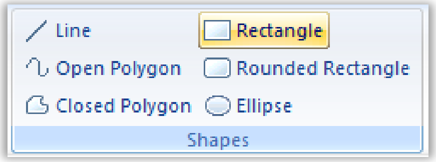
```
Level: {#Data.LevelTag:cTank[1].rLevel}
Temp: {#Data.TempTag:cTank[1].rTemperature}
Press: {#Data.PressureTag:cTank[1].rPressure}
```

18. Click **OK**

19. Position this rectangle above the first rectangle.



20. On the Graphics tab of the Ribbon in the Shapes group, select **Rectangle**.



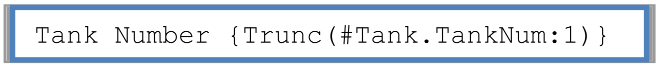
21. Click and drag on the Main screen to shape the object.

22. Double-click on the Rectangle to open the Object Properties window.

23. In the Background section, select Dark Blue.

24. Click on the **Caption** button.

25. Type the following text into the field:



```
Tank Number {Trunc(#Tank.TankNum:1)}
```

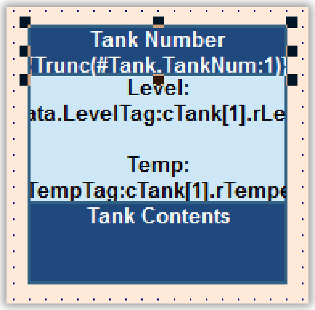
26. Click on **Fonts**.

27. In **Font Style**, select Bold.

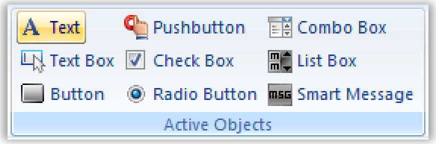
28. In **Color**, select White.

29. Click **OK**.

30. Position this rectangle above the second rectangle.



31. On the Graphics tab of the Ribbon in the Active Objects group, select **Text**.



32. Click on the Main screen to add the object. A cursor will appear.

33. Type **####################** at the cursor and click Enter.

34. Double-click on the Text object to open the Object Properties window.

35. Select the Border and Background boxes (default colors).

36. Select **Fonts**.

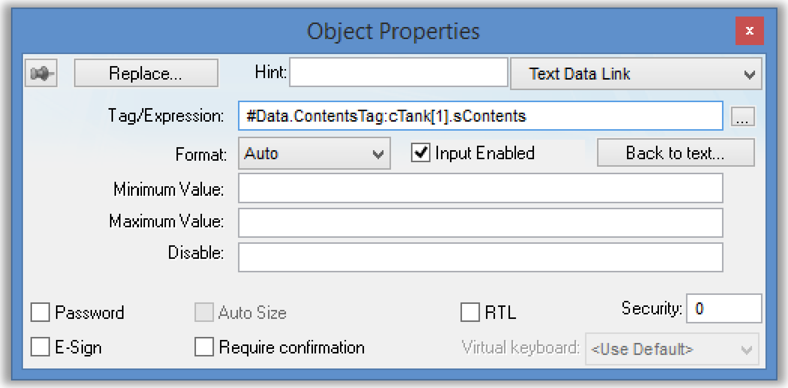
37. Configure the Font as **Courier New,** Style **Bold**.

38. Click **OK**.

39. Click the **Text data link…** button.

40. In the Tag/Expression, configure **#Data.ContentsTag:cTank[1].sContents**.

41. Click the box for Input Enabled.



Hint: Text Data Link

Tag/Expression: #Data.ContentsTag:cTank[1].sContents

Format: Auto ☑ Input Enabled

Minimum Value:

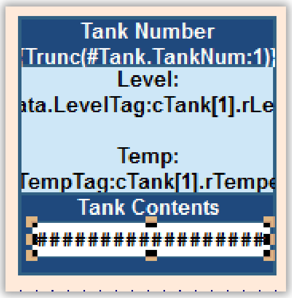
Maximum Value:

Disable:

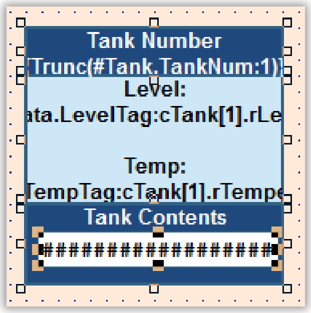
□ Password □ Auto Size □ RTL Security: 0

□ E-Sign □ Require confirmation Virtual keyboard: <Use Default>

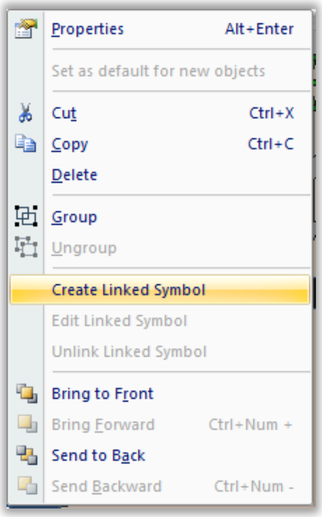
42. Position the Text object on top of the lower rectangle.



43. Select all four objects by drawing a box around the objects with your cursor



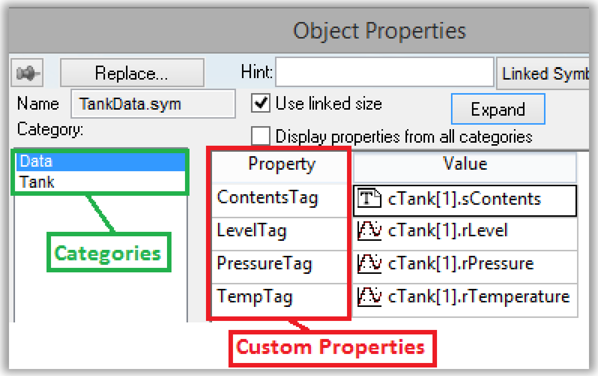
44. Right-click the selected objects and click **Create Linked Symbol.**



45. A Save As dialog is opened. For File name:, type **TankData** and click **Save.**

Pro-Tip: By default, the symbol will be saved in your project's **\Symbol** folder. If you wish to have this symbol available to all projects, you can save the **.sym** file in your installation folder (typically `C:\Program Files (x86)\BLUE Open Studio v8.0\Symbol`). Saving the **.sym** file will make the symbol file available by accessing the Symbols library.

46. The objects are now a linked symbol. Back on the Main screen double-click the symbol to open the Object Properties window.


Object Properties

Replace... Hint: Linked Symb

Name TankData.sym ☑ Use linked size Expand

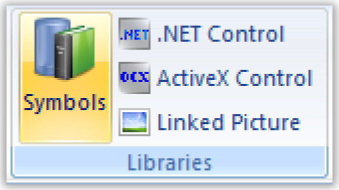
Category: ☐ Display properties from all categories

Data
Tank

Property	Value
ContentsTag	cTank[1].sContents
LevelTag	cTank[1].rLevel
PressureTag	cTank[1].rPressure
TempTag	cTank[1].rTemperature

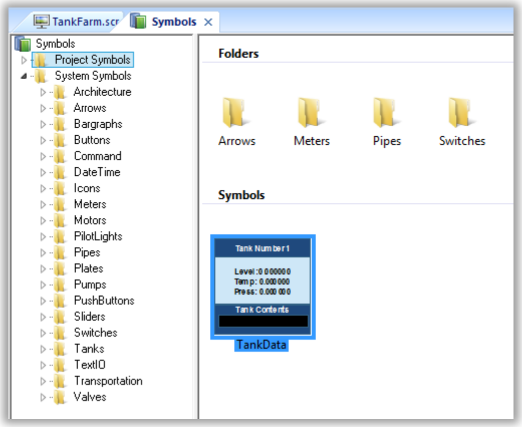
47. Observe the defined **Categories** and **Custom Properties**.

48. In the Libraries group of the Graphics tab of the Ribbon, click on **Symbols**.



49. The Symbols tab will be displayed in the Workspace.

50. Click on Project Symbols on the left. Select the TankData symbol from the Symbols group.

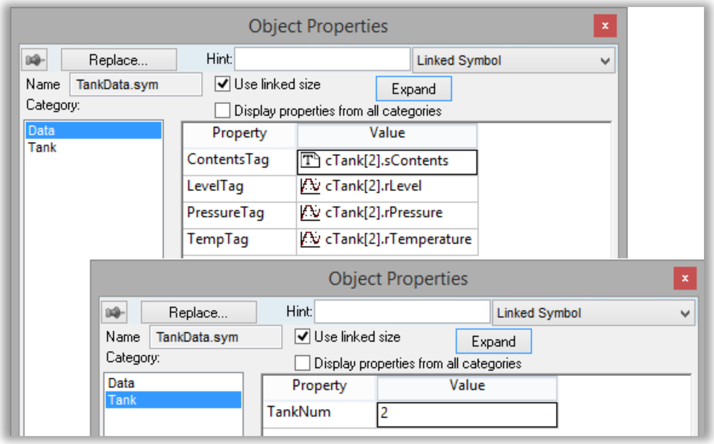


51. Select the tab in the Workspace for the Main screen.

52. Click on the screen to place the symbol.

53. Double-click the symbol to open the Object Properties window.

54. Configure the properties to access Tank 2 as below.



Hint: Linked Symbol

Name TankData.sym

☑ Use linked size

☐ Display properties from all categories

Property	Value
ContentsTag	cTank[2].sContents
LevelTag	cTank[2].rLevel
PressureTag	cTank[2].rPressure
TempTag	cTank[2].rTemperature

Hint: Linked Symbol

Name TankData.sym

☑ Use linked size

☐ Display properties from all categories

Property	Value
TankNum	2

55. Repeat Steps 50-56 for Tank 3.

56. Save and close Main.

57. If your runtime is on a remote machine, then use the the Remote Management tools on the Home tab of the Ribbon to download your project to the remote machine. Start the runtime and test.

SECTION 9 - PROJECT LOCALIZATION

You can quickly translate your project's user interface to multiple languages, using either machine translation (e.g., Google Translate) or a human translator, and then you can switch your project's language during runtime with a simple function call.

Concept

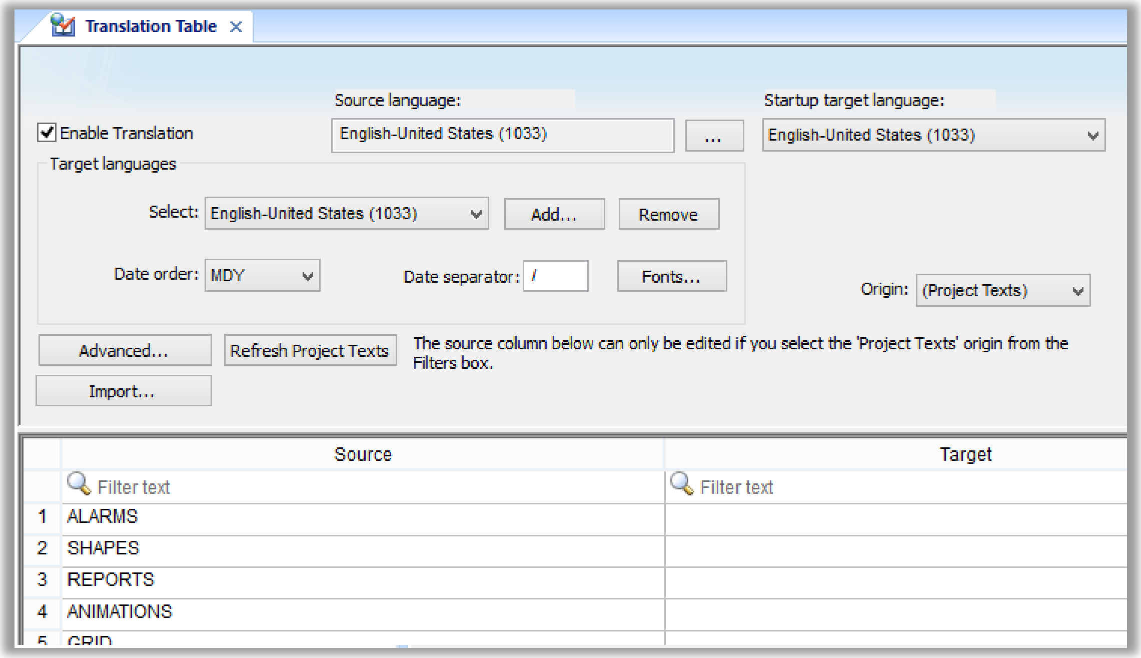
The Translation Table is a worksheet that you can use to create a multilingual user interface (MUI) for your project. (This is different from changing the language of the development environment itself; that is done with the Language command on the View tab of the ribbon.) The worksheet is divided into a Source column, which contains original pieces of text from your project screens, and a Target column, which contains the translated equivalents of the items in the Source column.

Overview

To open the translation table, do one of the following:

- On the Insert tab of the ribbon, in the Global group, click **Translation.**
- On the Global tab of the Project Explorer, right-click the **Translation** folder and click **Open** on the shortcut menu.

The Translation Table is opened.



☑ Enable Translation

Source language: English-United States (1033)

Startup target language: English-United States (1033)

Target languages

Select: English-United States (1033)

Date order: MDY

Date separator: /

Origin: (Project Texts)

The source column below can only be edited if you select the 'Project Texts' origin from the Filters box.

	Source	Target
	Filter text	Filter text
1	ALARMS	
2	SHAPES	
3	REPORTS	
4	ANIMATIONS	
5	GRID	

The Translation Table is used to manage the languages into which you want to translate your project. Adding a language to the table is as simple as selecting it from a list and then automatically translating your project texts.

Make sure the Enable Translation option is selected. If this option is cleared, then the entire Translation Table worksheet is disabled and the language cannot be changed during run time.

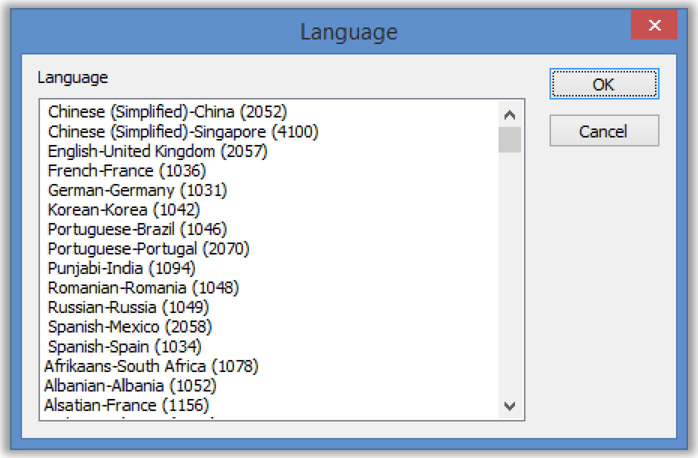
If the source language of your project is other than the language of the development environment itself, set the source language:

- To the right of the Source language box, click the browse button. The Languages dialog box is displayed.
- In the Languages dialog box, select the language for which your project was originally developed, and then click OK.

The selected language is set in the Source language box.

Add a target language to the Translation Table

In the Target languages group, click Add. The Languages dialog box is displayed.



Chinese (Simplified)-China (2052)
Chinese (Simplified)-Singapore (4100)
English-United Kingdom (2057)
French-France (1036)
German-Germany (1031)
Korean-Korea (1042)
Portuguese-Brazil (1046)
Portuguese-Portugal (2070)
Punjabi-India (1094)
Romanian-Romania (1048)
Russian-Russia (1049)
Spanish-Mexico (2058)
Spanish-Spain (1034)
Afrikaans-South Africa (1078)
Albanian-Albania (1052)
Alsatian-France (1156)

In the Languages dialog box, select the language to which you want to translate your project, and then click OK.

The selected language is added to the Select list, and a new worksheet is created for the language. The Source column of the worksheet is automatically populated with all of the translatable text strings in your project, and the Target column is blank.

For each text item in the Source column, enter the translation in the Target column. You can manually translate the items one by one, or you can automatically translate multiple items using Google Translate.

- Highlight the cells in the Source column to select them, and then press Ctrl-C to copy those items to the clipboard. You can also click the column header to select the entire column.
- In your web browser, go to translate.google.com.
- Select the From and To languages.
- Click in the text box on the left, and then press Ctrl-V to paste the items from the clipboard. The items are pasted as separate lines.
- Click Translate. The translation appears on the right. Again, the items are translated as separate lines.

- Highlight the translated items to select them, and then press Ctrl-C to copy the items to the clipboard.
- In BLUE Open Studio, highlight the empty cells in the Target column to select them. You can also click the column header to select the entire column.
- Press Ctrl-V to paste the translated items from the clipboard into the Target column. Because the translated items were copied as separate lines, they should be correctly pasted into the rows of the worksheet.

Review the translated items in the Target column, to confirm the translations and to make sure they match the untranslated items in the Source column.

Pro Tip: If you want to keep the original, untranslated text, then leave the Target column blank.

About the date format and how to change it

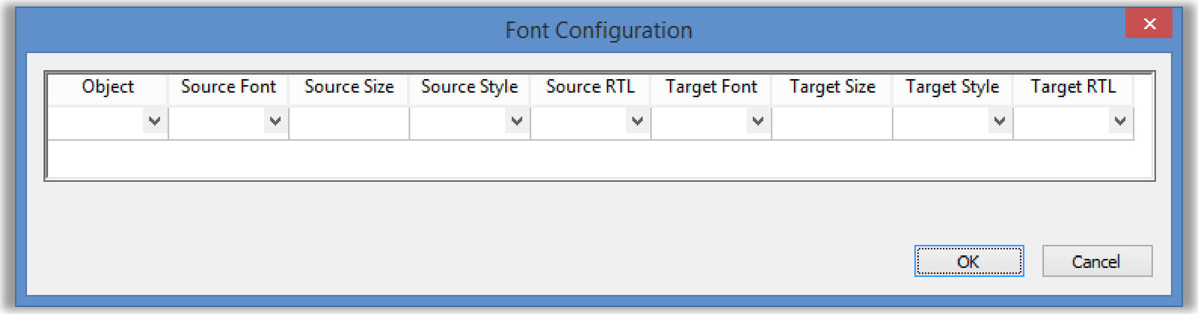
The date format determines how dates are displayed throughout the BLUE Open Studio software, as well as how date strings are handled by the project during run time. You can change the date format, either as part of localizing your BLUE Open Studio project or simply to suit your personal preference.

Configure Date order and Date separator as needed for the target language. For example, for English-United States (i.e., American English), Date order is typically MDY and Date separator is typically /, resulting in a date format of MM/DD/YYYY. In many European languages, however, Date order is typically YMD and Date separator is typically ., resulting in a date format of YYYY.MM.DD.

Configure fonts for a target language

Configure the fonts for a target language so that translated text items are displayed in another font, size, and/or style that is more appropriate to the language.

In the Target languages area, select the language for which you want to configure fonts, and then click Fonts. The Font Configuration dialog box is displayed.


Font Configuration

Object	Source Font	Source Size	Source Style	Source RTL	Target Font	Target Size	Target Style	Target RTL

In the Object column, select the type of screen objects that the configuration will apply to. To have it apply to all types, select (All).

In the Source Font column, select the font that the configuration will apply to. To have it apply to all fonts, select (All). The list of selectable fonts includes all fonts that are installed on your computer, not just the fonts that are used in your project.

In the Source Size column, type the font size (in points) that the configuration will apply to. To have it apply to all sizes, type (All).

In the Source Style column, select the font styles that the configuration will apply to. To have it apply to all styles, select (All).

In the Source RTL column, select the reading direction (left-to-right or right-to-left) that the configuration will apply to. To have it apply to both, select (All).

In the Target Font column, select the font to which the Source Font should be changed. If you want to leave the Source Font unchanged, select *. The list of selectable fonts includes all fonts that are installed on your computer.

Pro Tip: Make sure that the font you select is also installed on any computers or devices that will run as project runtime clients. If you cannot do this — in other words, if you do not control all of the potential clients — then select a "standard" font that is commonly installed on all platforms.

In the Target Size column, type the font size (in either points or percentage) to which the Source Size should be changed. If you want to leave the Source Size unchanged, type *.

In the Target Style column, select the font style to which the Source Style should be changed. If you want to leave the Source Style unchanged, select *.

In the Target RTL column, select the reading direction (left-to-right or right-to-left) to which the Source RTL should be changed. If you want to leave the Source RTL unchanged, select *.

Repeat this procedure for additional rows, if necessary.

Note: The font configuration applies only to the selected target language.

Set the project's language at startup

Even when you have multiple languages configured for your project, you must still specify which language you want your project to start in at runtime.

In the Startup target language list, select the language in which you want your project to start. The list of available languages includes the source language in which you developed the project and any target languages that you have added.

Set the project's language during run time

You can set your project's language during run time by using the function SetLanguage anywhere that an expression can be configured.

Module 6 - Scripting

SECTION 1 - CLIENT-SIDE SCRIPTS

Screen Script

Each project screen has an associated screen script. To edit the screen script for a given project screen, open the screen worksheet for editing and then do one of the following:

- On the Graphics tab of the ribbon, in the Screen group, click Script; or
- Right-click in the screen worksheet, and then click Screen Script on the shortcut menu.

This interface can be used to execute scripts on the following events, based on preconfigured sub-routines:

- **Screen_OnOpen()**: The code configured within this sub-routine is automatically executed just once when the screen is opened.
- **Screen_WhileOpen()**: The code configured within this sub-routine is automatically executed continuously while its screen is open. The rate in which this sub-routine is called depends on the performance of the platform where the project is running.
- **Screen_OnClose()**: The code configured within this sub-routine is automatically executed just once when the screen is closed.

The variables and procedures declared in this interface are available for the VBScript interfaces of the screen where the Screen Script is configured.

Note: Do not change the names of the preconfigured sub-routines described above. If you do, then the system will not be able to call them.

Screen Logic

Each project screen has associated Screen Logic that can be configured from the Screen Attributes dialog.

- **On Open**: Lists expressions to be evaluated once when the screen is opened, similar to a Math worksheet.
- **While Open**: Lists expressions to be continuously evaluated while the screen is open, similar to a Math worksheet. If you also configure a tag/expression in Trigger, then instead of being continuously evaluated, the listed expressions will be evaluated once each time the value of the trigger changes while the screen is open.
- **On Close**: Lists expressions to be evaluated once when the screen is closed, similar to a Math worksheet.

Graphics Script

The Graphics Script interface can be edited by its icon from the Graphics tab of the Project Explorer:

- On the Graphics tab of the Project Explorer, right-click **Graphics Script**, and click **Open** on the shortcut menu.

This interface can be used to execute scripts on the following events, based on pre-defined sub-routines:

- **Graphics_OnStart()**: The code configured within this sub-routine is automatically executed just once when the graphic module is started. This interface is useful for initializing variables or executing logics that must be implemented when running the project.
- **Graphics_WhileRunning()**: The code configured within this sub-routine is automatically executed continuously while the graphic module is running. The rate in which this sub-routine is called depends on the performance of the platform where the project is running.
- **Graphics_OnEnd()**: The code configured within this sub-routine is automatically executed just once when the graphic module is closed.

Important: Do not change the name of the predefined sub-routines. If you do, then the system will not be able to automatically execute them.

SECTION 2 - BACKGROUND TASKS

Math Worksheets

A Math worksheet is used to implement program logic (using the Built-in Scripting Language) that should be continuously executed during runtime, rather than on specific actions like the user pressing a button on a screen.

Note: The Math worksheet is functionally similar to the Script worksheet, except that it uses the Built-in Scripting Language instead of VBScript.

To create a new Math worksheet, do one of the following:

- On the Insert tab of the ribbon, in the Task Worksheets group, click Math;
- Right-click the Math folder in the Project Explorer, and then click Insert on the shortcut menu; or
- Click New on the Application menu, click the File tab, and then select Math Worksheet.

To edit an existing Math worksheet, double-click it in the Project Explorer

The Math worksheet is divided into two areas:

- **Header area** (top section), which contains information for the whole group; and
- **Body area** (bottom section), where you define each tag, expression, and Programming Lines (logical routines and mathematical calculations through functions and logical operations) in the group.

Use the **Header** parameters on this worksheet as follows:

- **Description field:** Type a description of the worksheet for documentation purposes.
- **Execution field**: Type an expression, a single tag, or a constant value to determine when the worksheet should execute.

  Note: The project executes the worksheet only when the Execution field result is True (not zero). If you always want the worksheet to execute constantly, type a **1** (static True value) in the Execution field.

Use the **Body** parameters on this worksheet as follows:

- **Tag Name:** Type a tag to receive a return value from the specified calculation in the Expression column.
- **Expression:** Type an expression to return a return value to the specified tag in the Tag Name column.

To execute a Math worksheet at a specific time, separate from whatever is configured in the Execution field, use the Math function anywhere an expression is allowed.

Script Worksheets

A Script worksheet is used to implement program logic (using VBScript) that should be continuously executed during runtime, rather than on specific actions like the user pressing a button on a screen.

> Note: The Script worksheet is functionally similar to the Math worksheet, except that it uses VBScript instead of the Built-in Scripting Language.

To create a new Script worksheet, do one of the following:

- On the Insert tab of the ribbon, in the Task Worksheets group, click Script;
- Right-click the Script folder in the Project Explorer, and then click Insert on the shortcut menu; or
- Click New on the Application menu, click the File tab, and then select Script Worksheet.

To edit an existing Script worksheet, double-click it in the Project Explorer

The Script worksheet is divided into two areas:

- **Header area** (top section), which contains information for the whole group; and
- **Body area** (bottom section), where you write a script using VBScript syntax.

Use the **Header** parameters on this worksheet as follows:

- **Description field:** Type a description of the worksheet for documentation purposes.
- **Execution field**: Type an expression, a single tag, or a constant value to determine when the worksheet should execute.

  > Note: You must use the syntax supported by the Built-in Scripting Language in the Execution field. Only the body of the worksheet supports VBScript.

In the **Body,** write script using VBScript and Built-In using the following syntax:

- **VBScript:** Type in any valid VBScript code supported by the Runtime device’s Operating system.
- **Built-In:** Precede any Tag or Built-In function with a Dollar sign (**“$”**) to integrate into your script. (e.g. `$MyTag, $GetComputerIP()`).

Variables declared in the worksheet have local scope for that specific group only. They are not available for any other VBScript interface.

Startup Script worksheet

The Startup Script worksheet is a VBScript interface that is automatically executed when the project is run.

To edit the Startup Script worksheet, double-click it in the Project Explorer. (It is located on the Tasks tab, in the Script folder.) The worksheet is displayed.

The code configured in this worksheet is executed just once when the Background Task module (BGTask) is started. This interface is useful for initializing variables or executing logics that must be implemented when the project is run.

You can declare and initialize variables and define procedures. However, variables or procedures declared in this interface will be available ONLY to the Script worksheets executed by the Background Task module — they are not available to any VBScript interface from the Graphic Module.

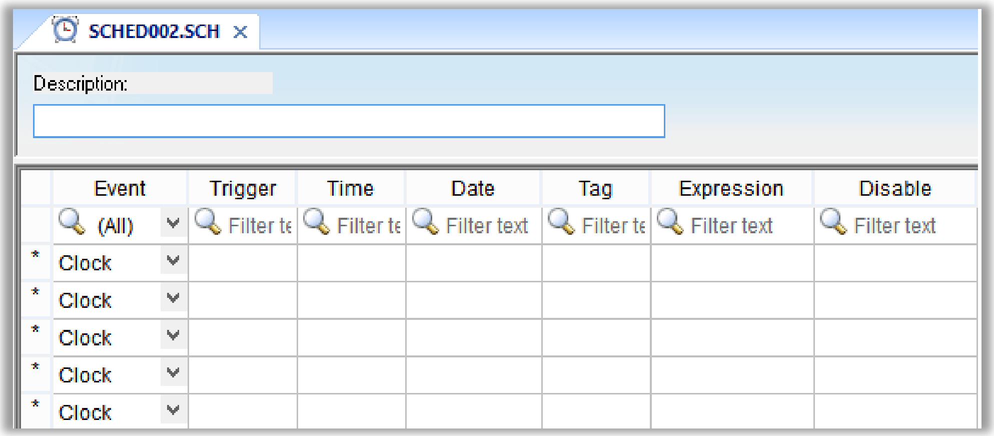
Scheduler

A Scheduler worksheet is used to execute program logic (using the Built-in Scripting Language) at a specific date/time, on a regular time interval, or upon a triggering event.

To create a new Scheduler worksheet, do one of the following:

- On the Insert tab of the ribbon, in the Task Worksheets group, click Scheduler;
- Right-click the Scheduler folder in the Project Explorer, and then click Insert on the shortcut menu; or
- Click New on the Application menu, click the File tab, and then select Scheduler Worksheet.

To edit an existing Scheduler worksheet, double-click it in the Project Explorer



Description:

	Event	Trigger	Time	Date	Tag	Expression	Disable
	(All)	Filter te	Filter te	Filter text	Filter te	Filter text	Filter text
*	Clock						
*	Clock						
*	Clock						
*	Clock						
*	Clock						

The Scheduler worksheet is divided into two areas:

- **Header area** (top section), which contains the description for the whole worksheet.
- **Body area** (bottom section), where you define each tag, expression, and condition for the event.

The parameters on this worksheet are as follows:

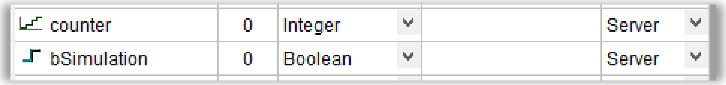
- **Description:** A description of the worksheet for documentation purposes.
- **Event:** Click to select an event type from the following:
  - **Calendar:** Generates time bases greater than 24 hours. For example, you can define an event that prints a report every Friday at a specific time.

  - **Clock:** Generates time bases smaller than 24 hours (intervals in minutes or seconds). This function is frequently used with trend graphics. For example, you can define a tag that will be incremented each hour.
  - **Change:** Event related to the change of a tag in the Trigger field.

- **Trigger:** This field is used only with the Change Event type. The event of this type will be triggered when the tag entered into this field changes value.
- **Time:** This field is used with the Calendar and Clock Event types.
  - If the Event type is **Calendar**, then Time is a specific time of the day on Date using the **HH:MM:SS** format as representation of a time on a 24-hour clock. When that Date and Time occurs, the expression is executed.
  - If the Event type is **Clock**, then Time is a time interval using the **HH:MM:SS** format starting from when the project is launched. Every time the interval occurs, the expression is executed.
- **Date:** An optional field used only with the Calendar Event type. Type a specific date using the **MM/DD/YYYY** format, and when that **Date** and **Time** occurs, the expression is executed.
  - Valid values are from 01/01/1900 to 12/31/2099. If the field is left blank, then the event occurs daily at the specified Time.
- **Tag:** Type a tag that will receive the value, if any, returned by the process in the Expression field.
- **Expression:** Type an expression to be executed when the event occurs. This field is used by all events.
- **Disable:** Contains a disable condition for the specified function, based on a True/False expression. By leaving this field blank or having a False expression (i.e. any zero value), this event will execute. Any True (i.e. non-zero) value will disable the event.

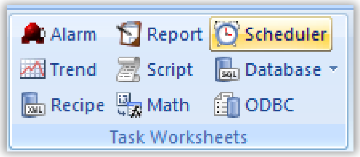
LAB 13 - USING BACKGROUND SCRIPTS

1. Open the Project Tags Datasheet View and create the tags as shown below.



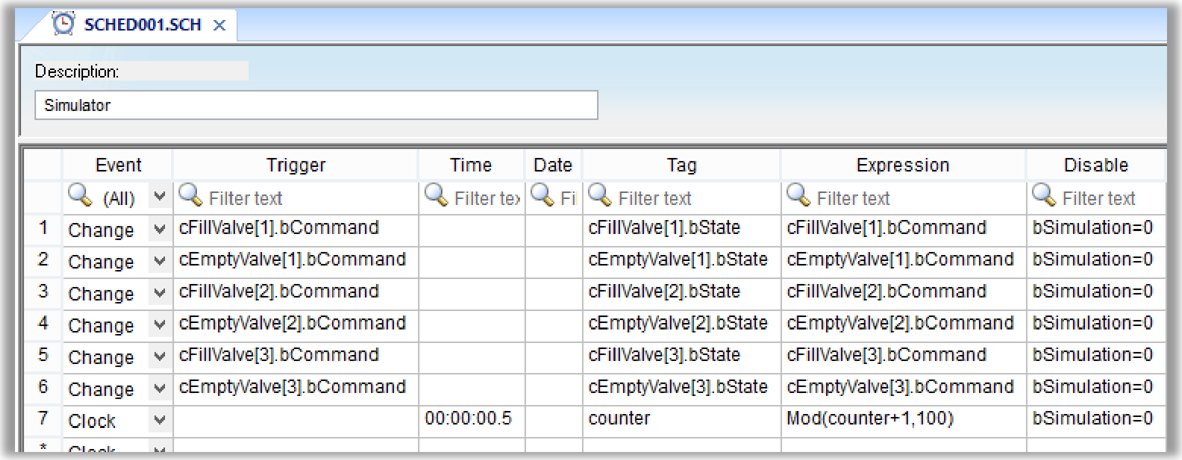
counter	0	Integer		Server
bSimulation	0	Boolean		Server

2. In the **Insert tab of the Ribbon**, select **Scheduler** from the **Task Worksheets** group.



3. The SCHED001 tab will be displayed in the Workspace.

4. Configure the worksheet as shown below,



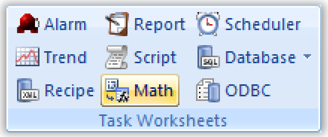
Description:

Simulator

	Event	Trigger	Time	Date	Tag	Expression	Disable
	(All)	Filter text	Filter tex	Fi	Filter text	Filter text	Filter text
1	Change	cFillValve[1].bCommand			cFillValve[1].bState	cFillValve[1].bCommand	bSimulation=0
2	Change	cEmptyValve[1].bCommand			cEmptyValve[1].bState	cEmptyValve[1].bCommand	bSimulation=0
3	Change	cFillValve[2].bCommand			cFillValve[2].bState	cFillValve[2].bCommand	bSimulation=0
4	Change	cEmptyValve[2].bCommand			cEmptyValve[2].bState	cEmptyValve[2].bCommand	bSimulation=0
5	Change	cFillValve[3].bCommand			cFillValve[3].bState	cFillValve[3].bCommand	bSimulation=0
6	Change	cEmptyValve[3].bCommand			cEmptyValve[3].bState	cEmptyValve[3].bCommand	bSimulation=0
7	Clock		00:00:00.5		counter	Mod(counter+1,100)	bSimulation=0
*	Clock						

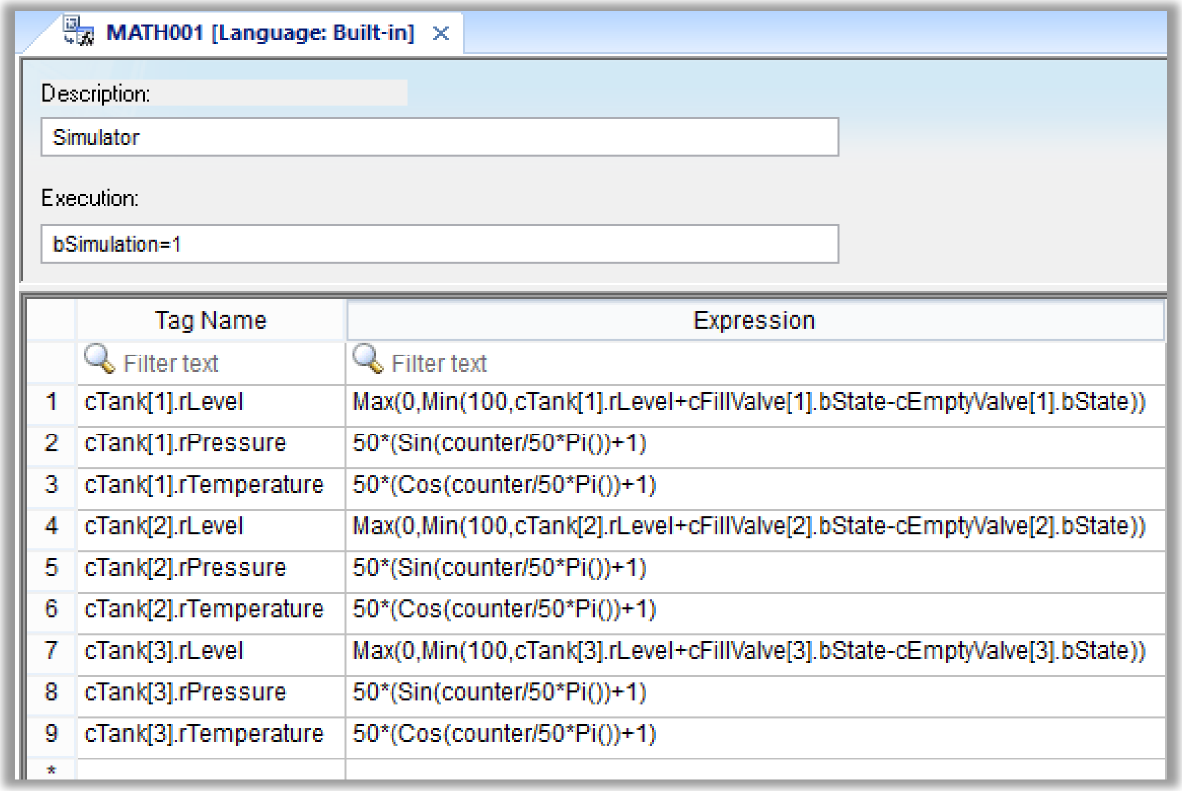
5. Save and close the SCHED001 worksheet.

6. In the **Insert tab of the Ribbon**, select **Math** from the **Task Worksheets** group.



7. The MATH001 tab will be displayed in the Workspace.

8. Configure the worksheet as shown below,



Description:

Simulator

Execution:

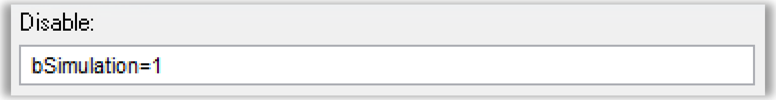
bSimulation=1

	Tag Name	Expression
	Filter text	Filter text
1	cTank[1].rLevel	Max(0,Min(100,cTank[1].rLevel+cFillValve[1].bState-cEmptyValve[1].bState))
2	cTank[1].rPressure	50*(Sin(counter/50*Pi())+1)
3	cTank[1].rTemperature	50*(Cos(counter/50*Pi())+1)
4	cTank[2].rLevel	Max(0,Min(100,cTank[2].rLevel+cFillValve[2].bState-cEmptyValve[2].bState))
5	cTank[2].rPressure	50*(Sin(counter/50*Pi())+1)
6	cTank[2].rTemperature	50*(Cos(counter/50*Pi())+1)
7	cTank[3].rLevel	Max(0,Min(100,cTank[3].rLevel+cFillValve[3].bState-cEmptyValve[3].bState))
8	cTank[3].rPressure	50*(Sin(counter/50*Pi())+1)
9	cTank[3].rTemperature	50*(Cos(counter/50*Pi())+1)
*		

9. Save and close the MATH001 worksheet.

10. Open the MAIN DRIVER SHEET of the MOTCP communication driver.

11. Configure the field called Disable as shown below.



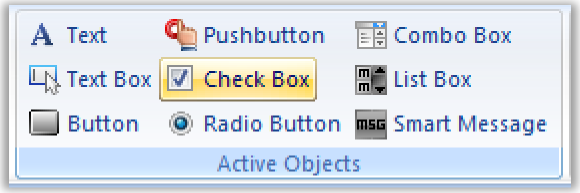
Disable:

bSimulation=1

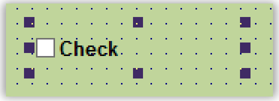
12. Save and close the MAIN DRIVER SHEET.

13. Open the HEADER screen.

14. On the Graphics tab of the Ribbon, select **Checkbox** from Active Objects.



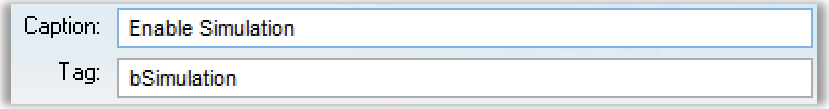
15. Click and drag on the HEADER screen to shape the object.



16. Double-click on the object to open the properties.

17. In **Caption**, configure **Enable Simulation**.

18. In **Tag**, add the **bSimulation** tag.


Caption: Enable Simulation

Tag: bSimulation

19. Save the changes made and close the HEADER screen.

Verification and Test

1. If your runtime is on a remote machine, then use the the Remote Management tools on the Home tab of the Ribbon to download your project to the remote machine.

2. Start the runtime.

3. Click the Enable Simulation checkbox in the header.

4. Verify that that you can read tank analog values and control valves independent of the PLC.

Learning Services

Pro-face
by Schneider Electric

Module 7 - Alarms and Events

SECTION 1 - ALARMS

This section uses a step-by-step tutorial to explain how to insert the Alarm task and the Alarm/Event Object, which will be used for training.

In the Alarm task, you can configure the high and low alarms for the Level and Temperature variables.

We can use the Alarm/Event Object to display the alarms in real and historical time.

Concept

The main purpose of the Alarm is to inform the operator of any problem or abnormal situation during the project runtime so that they may take corrective action. It runs in the background, sounds alarms, and saves alarm messages to a history but it does not display them.

The Alarm/Event Control Object does not generate alarms, but displays them either from the Alarm background task (Live alarms) or the alarm log/database (historical alarms). The Alarm/Event Control Object can also display messages from the Event Logger (See Section 2 of this module).

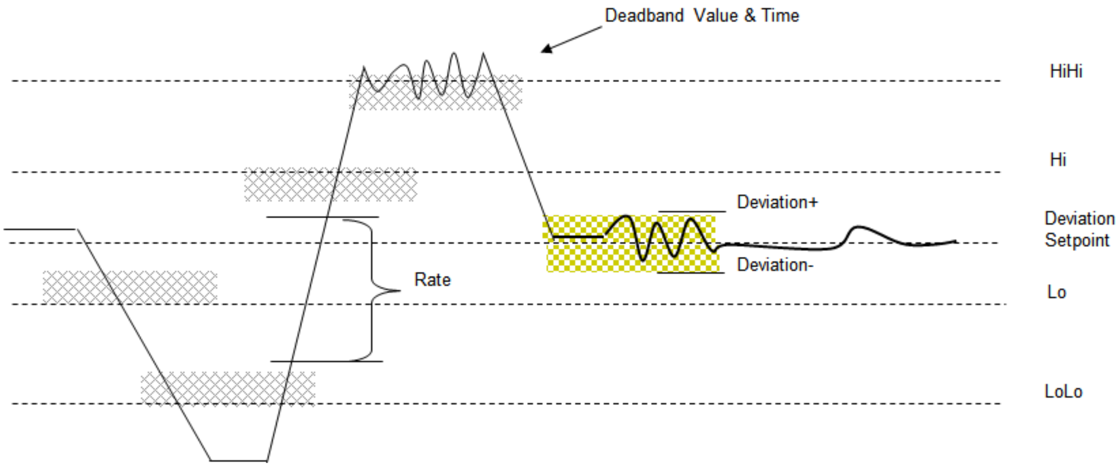
Overview

The Alarms Task allows for configuring alarm groups and tags related to each tag. The Alarm worksheet defines the alarm messages generated by the project. The primary purpose of an alarm is to inform the operator of any problems or abnormal condition during the process so he can take corrective action(s).

The Alarm worksheet is executed by the Background Task module. It handles the status of all alarms and optionally saves the alarm messages to a history file, but it does not display the alarm messages to the operator. The Alarm/Event Control screen object, available on the Graphics tab of the Ribbon, must be created and configured in a screen in order to display alarms.

The Alarm task has 3 main Alarm types for each numeric tag (**Real** or **Integer** datatype):

- **Level Alarms (HiHi, Hi, Lo, LoLo):** Triggers alarm if the value is equal to or higher than the limit (Hi, HiHi); or equal to or lower than the limit (Lo, LoLo)
- **Rate Alarms (Rate):** Triggers the alarm if the value rises or falls by a particular amount within a designated time frame
- **Deviation Alarms (Dev+, Dev-):** Triggers the alarm when the value deviates by a designated amount above (Dev+) or below (Dev-) a specified setpoint.



The Alarm task has 2 main Alarm types for each bit tag (**Boolean** datatype):

- **Level Alarms (Hi, Lo):** Triggers alarm if the value of the tag changes to True (Value = **1**, **Hi Alarm**); or changes to False (Value=**0**, **Lo Alarm**).
- **Rate Alarms (Rate):** Triggers the alarm if the value of the tag changes (either to True or False) within a designated time frame.

No alarms are available for tags of type string.

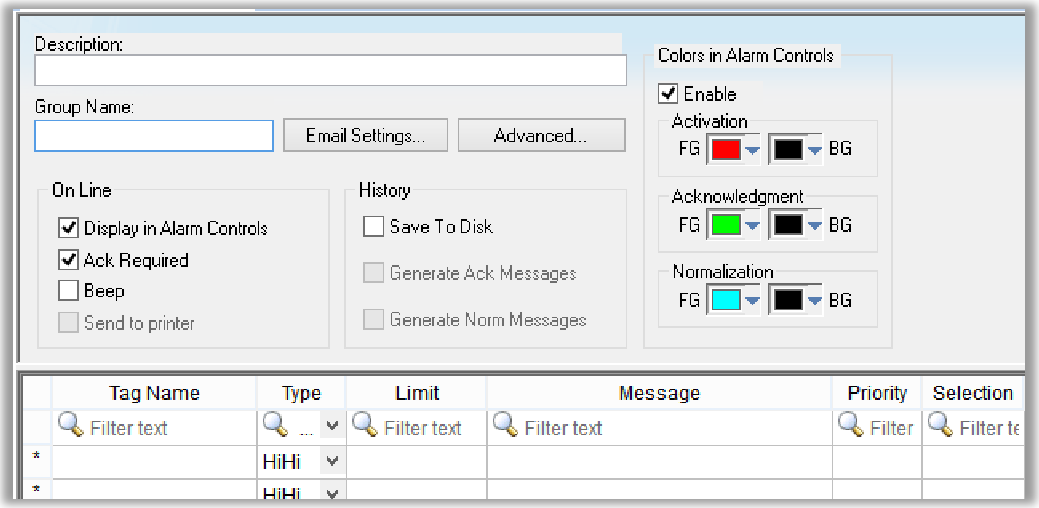
The Alarms for each tag are stored in the Tags Database as properties of that tag. Therefore, only one alarm of each type is available within the project for each tag. However, the alarms on a particular tag are not limited to a single Group (Alarm worksheet). So for example, even though there may be only one Hi and one Lo alarm available on each tag, the Hi and the Lo alarm may be on separate worksheets.

Each Alarm type may belong to one of three State types:

- **Active:** means that the alarm is active and has not been acknowledged;
- **Acknowledged:** means that the alarm is still in an active state but has been acknowledged;
- **Normalized:** means that the condition for the alarm is no longer active, but that the alarm has not yet been acknowledged

Alarm Worksheet

You can create multiple Alarm groups (worksheets) and each group can be configured with independent settings.



Description:

Group Name:

On Line
☑ Display in Alarm Controls
☑ Ack Required
☐ Beep
☐ Send to printer

History
☐ Save To Disk
☐ Generate Ack Messages
☐ Generate Norm Messages

Colors in Alarm Controls
☑ Enable
Activation FG 🟥 ■ BG
Acknowledgment FG 🟩 ■ BG
Normalization FG 🟦 ■ BG

	Tag Name	Type	Limit	Message	Priority	Selection
*		HiHi				
*		HiHi				

Each Alarm worksheet is composed into two areas:

- **Headers:** Settings applied to all tags and alarms configured in the same alarm group. These settings allow you to configure the formatting of the message and the actions that must be triggered based on alarm events.

- **Body:** Configure alarm messages and associate them to conditions linked to tags.

The parameters on this worksheet are as follows:

- **Description:** Description of the alarm worksheet as it will be displayed in the Project Explorer, used for documentation only.

- **Group Name:** Name of the Alarm group. During runtime, the operator can filter alarms based on the Group Name by the built-in Filters dialog (Advanced settings) of the Alarm/Event control object.

- **Email Settings:** Launches the Email Settings dialog, where you can configure settings for emails sent automatically based on alarm conditions.

- **Advanced:** Launches the Advanced Settings dialog, where you can configure advanced features, such as disabling the group, remote acknowledgement triggers, feedback values and deadband configuration.

- **On Line**
  - **Display in Alarm Controls:** When checked, the configured alarms are available to be displayed on the Alarm/Event Control object.
  - **Ack Required:** When checked, the alarms require acknowledgment, and will be displayed on the Alarm/Event Control object (Online mode) until they are acknowledged AND normalized.
  - **Beep:** When checked, the computer keeps beeping while there are alarm(s) requiring acknowledgement.

  - **Send to Printer:** When checked, the alarm messages are sent to the printer at LPT1 as soon as the alarm event occurs.

Note: When using this option, you must use a line-feed matrix printer in order to print the message(s). Otherwise, each alarm will be printed in a different sheet of paper. In order to print multiple alarms on a single page instead, use the Print trigger in the Alarm/Event Control Object's Advanced dialog.

- **History**
  - **Save to Disk:** When checked, the alarm messages are stored in the history log when they become active.
  - **Generate Ack Messages:** When checked, the alarm messages are stored in the history log when they are acknowledged.
  - **Generate Norm Messages:** When checked, the alarm messages are stored in the history log when they become normalized.
- **Colors in Alarm Controls:** When checked, the alarms configured in this group will be displayed with the colors assigned to each state (Activation, Acknowledgement or Normalization) according to the colors configured in the worksheet. If disabled (unchecked), the Alarm Control Object uses the colors set in the object's properties.
- **Tag Name:** Name of the tag associated with the alarm.
- **Type:** Type of alarm (Hi, HiHi, LoLo, Lo, Rate, DevP, and DevM)

Note: When using the types **Rate**, **DevP** and **DevM**, it is necessary to configure the Rate interval and/or Deviation Setpoint by the Tag Properties dialog (Properties under the Home tab of the Ribbon, Tags group).

- **Limit:** Limit associated with each alarm.
- **Message:** Message associated to the alarm.
- **Priority:** Priority number associated to the alarm, from 0-255; used for filtering alarms in runtime. Also prioritizes alarms in the tags database.
- **Selection:** Alias associated to the alarm (e.g., AreaA, AreaB, etc), for displaying, filtering and sorting purposes.

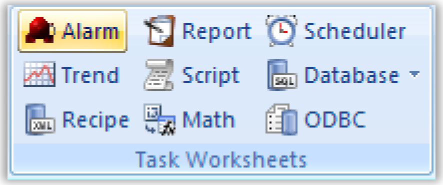
Pro-Tip: You can configure up to 10 additional custom fields that will be saved to the alarm history. The number of custom fields (up to 10) can be set in Project Settings: Options.

LAB 14 - WORKING WITH ALARMS

In this lab you will configure alarms on the Tank level tags to alert you when your tanks are getting full.

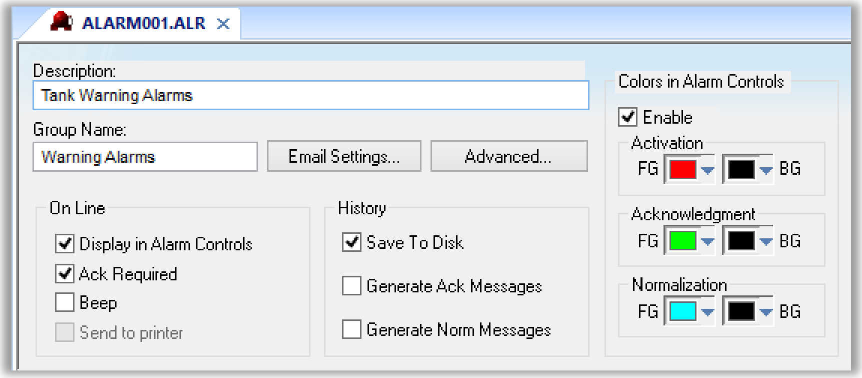
Alarm Task

1. In the Insert tab of the Ribbon select **Alarm** from the Task Worksheets group.



2. The ALARM001 tab is opened in the Workspace.

3. Configure the header of the ALARM001 worksheet as shown below,



Description: Tank Warning Alarms

Group Name: Warning Alarms

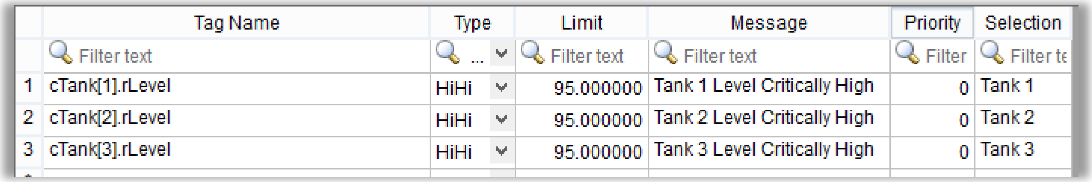
On Line
☑ Display in Alarm Controls
☑ Ack Required
□ Beep
□ Send to printer

History
☑ Save To Disk
□ Generate Ack Messages
□ Generate Norm Messages

Colors in Alarm Controls
☑ Enable
Activation: FG 🟥 ⬛ BG
Acknowledgment: FG 🟩 ⬛ BG
Normalization: FG 🟦 ⬛ BG

The FG color should be red.

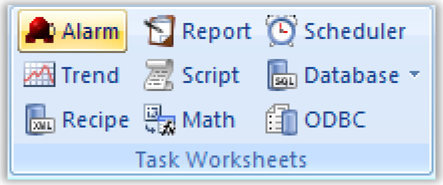
4. Configure the body of the ALARM001 worksheet as shown below.



	Tag Name	Type	Limit	Message	Priority	Selection
	Filter text	...	Filter text	Filter text	Filter	Filter te
1	cTank[1].rLevel	HiHi	95.000000	Tank 1 Level Critically High	0	Tank 1
2	cTank[2].rLevel	HiHi	95.000000	Tank 2 Level Critically High	0	Tank 2
3	cTank[3].rLevel	HiHi	95.000000	Tank 3 Level Critically High	0	Tank 3

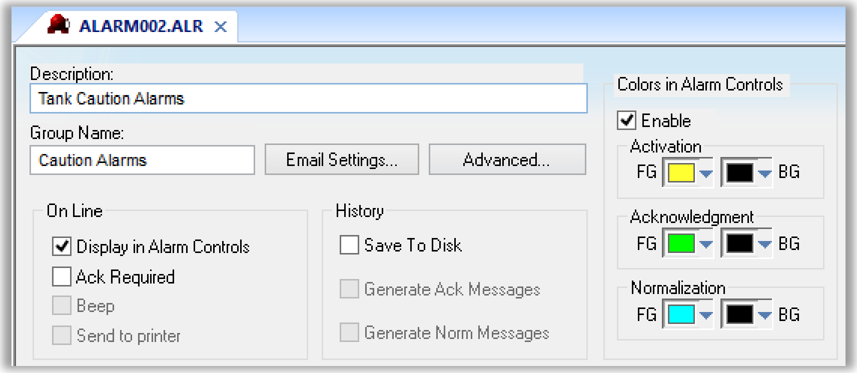
5. Save and close alarm worksheet ALARM001.

6. In the Insert tab of the Ribbon select **Alarm** from the Task Worksheets group.



7. The ALARM002 tab is opened in the Workspace.

8. Configure the header of the ALARM002 worksheet as shown below,



ALARM002.ALR ×

Description: Tank Caution Alarms

Group Name: Caution Alarms

Email Settings... Advanced...

On Line

☑ Display in Alarm Controls

□ Ack Required

□ Beep

□ Send to printer

History

□ Save To Disk

□ Generate Ack Messages

□ Generate Norm Messages

Colors in Alarm Controls

☑ Enable

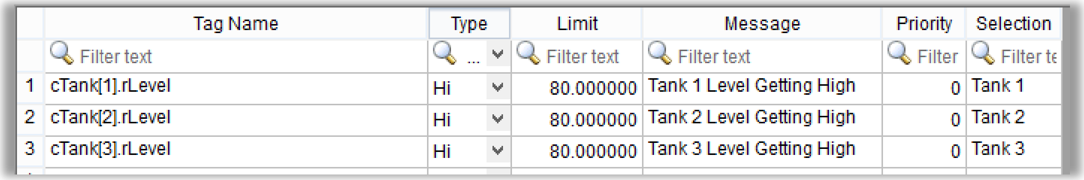
Activation: FG 🟨 BG ■

Acknowledgment: FG 🟩 BG ■

Normalization: FG 🟦 BG ■

The FG color should be yellow.

9. Configure the body of the ALARM002 worksheet as shown below.

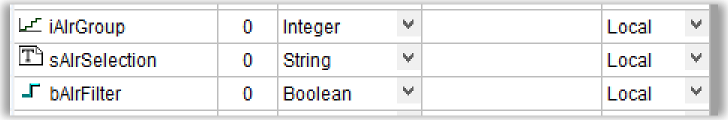


	Tag Name	Type	Limit	Message	Priority	Selection
	Filter text	...	Filter text	Filter text	Filter	Filter te
1	cTank[1].rLevel	Hi	80.000000	Tank 1 Level Getting High	0	Tank 1
2	cTank[2].rLevel	Hi	80.000000	Tank 2 Level Getting High	0	Tank 2
3	cTank[3].rLevel	Hi	80.000000	Tank 3 Level Getting High	0	Tank 3

10. Save and close alarm worksheet ALARM002.

Alarm Object

1. Open the Project Tags Datasheet View and create the tags as shown below.



iAlrGroup	0	Integer		Local
sAlrSelection	0	String		Local
bAlrFilter	0	Boolean		Local

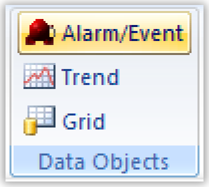
2. Open the Template screen.

3. Click the Application button and click **Save As** on the Application menu. The Save As dialog is opened.

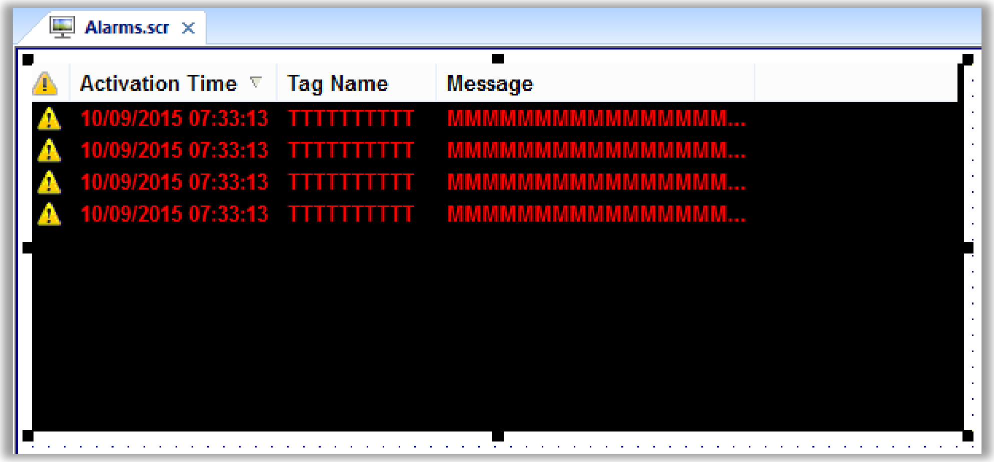
4. In the **File name** field, enter `Alarms`.

5. Click **Save**. The Save As dialog is closed.

6. In the Graphics tab of the Ribbon, select **Alarm/Event** under Data Objects.

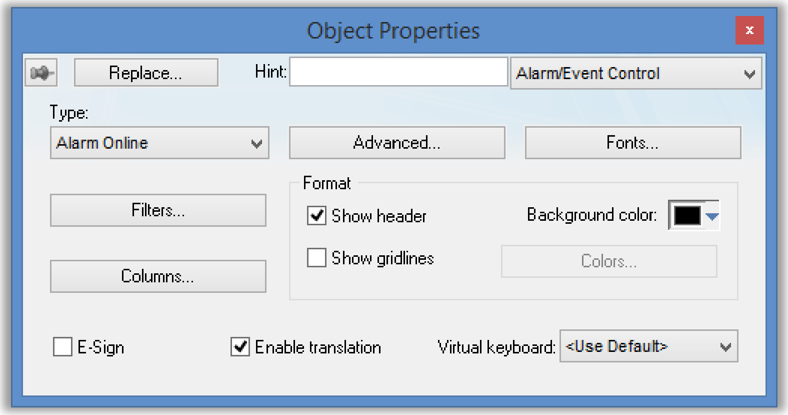


7. Click and drag on the screen to shape the object as shown below.



⚠	Activation Time ▽	Tag Name	Message
⚠	10/09/2015 07:33:13	TTTTTTTTTT	MMMMMMMMMMMMMMMMM...
⚠	10/09/2015 07:33:13	TTTTTTTTTT	MMMMMMMMMMMMMMMMM...
⚠	10/09/2015 07:33:13	TTTTTTTTTT	MMMMMMMMMMMMMMMMM...
⚠	10/09/2015 07:33:13	TTTTTTTTTT	MMMMMMMMMMMMMMMMM...

8. Double-click on the object to open the properties.



Hint: Alarm/Event Control

Type: Alarm Online

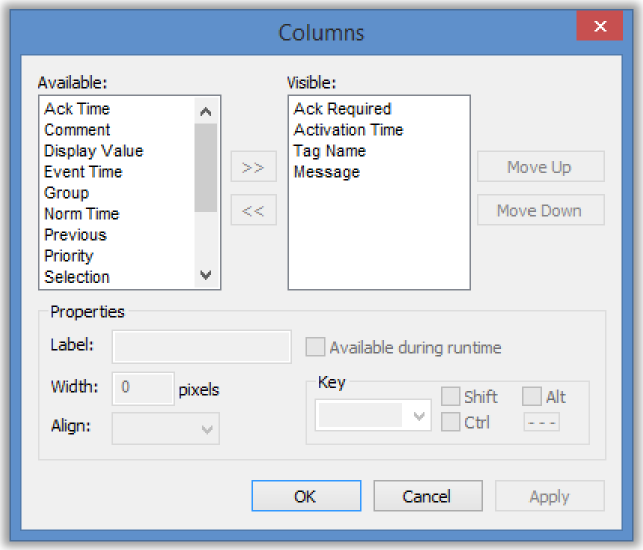
Format

☑ Show header Background color: ■

□ Show gridlines

□ E-Sign ☑ Enable translation Virtual keyboard: <Use Default>

9. Click on **Columns**. The Columns dialog is opened.



Available:
Ack Time
Comment
Display Value
Event Time
Group
Norm Time
Previous
Priority
Selection

Visible:
Ack Required
Activation Time
Tag Name
Message

Properties

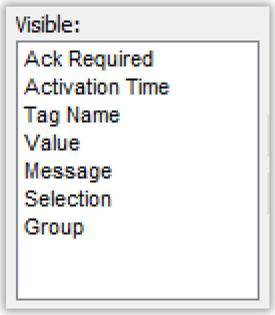
Label: □ Available during runtime

Width: 0 pixels

Key □ Shift □ Alt □ Ctrl - - -

Align:

10. Use the left and right double arrow controls and Move Up and Move Down buttons to move Ack Required, Activation Time, Tag Name, Value, Message, Selection and Group from the Available list to the Visible list as shown below.

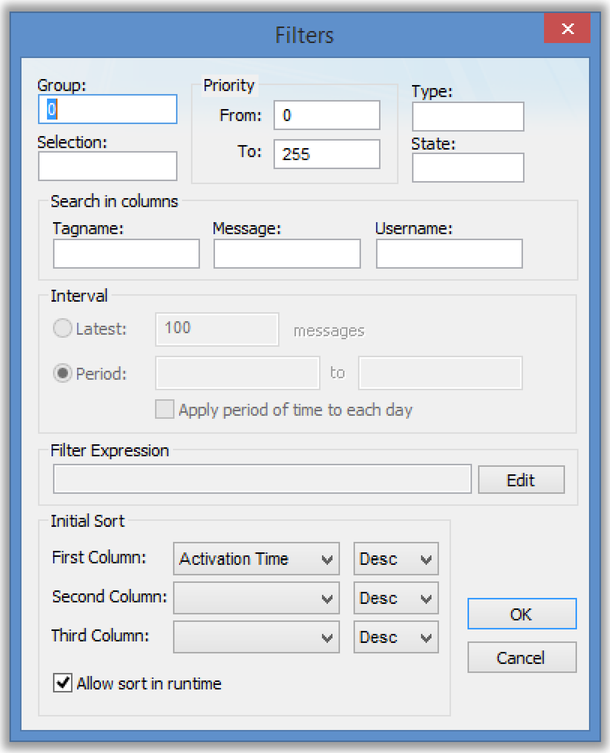


Visible:

- Ack Required
- Activation Time
- Tag Name
- Value
- Message
- Selection
- Group

11. Select **OK** to close the Columns dialog.

12. Click on **Filters**. The Filters dialog is opened.



Filters

Group: 0

Selection:

Priority

From: 0

To: 255

Type:

State:

Search in columns

Tagname:

Message:

Username:

Interval

○ Latest: 100 messages

● Period: to

□ Apply period of time to each day

Filter Expression

Initial Sort

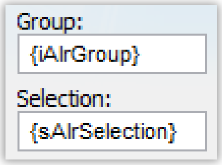
First Column: Activation Time Desc

Second Column: Desc

Third Column: Desc

☑ Allow sort in runtime

13. Configure the fields called **Group** and **Selection** as shown below.



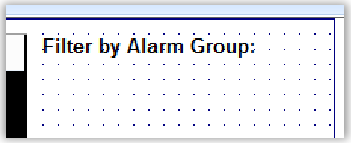
Group: {iAlrGroup}

Selection: {sAlrSelection}

14. Select **OK** to close the Filters dialog.

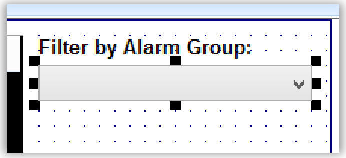
15. In the Graphics tab of the Ribbon, select **Text** under Active Objects.

16. Click on the screen and enter **Filter by Alarm Group:** as shown below.

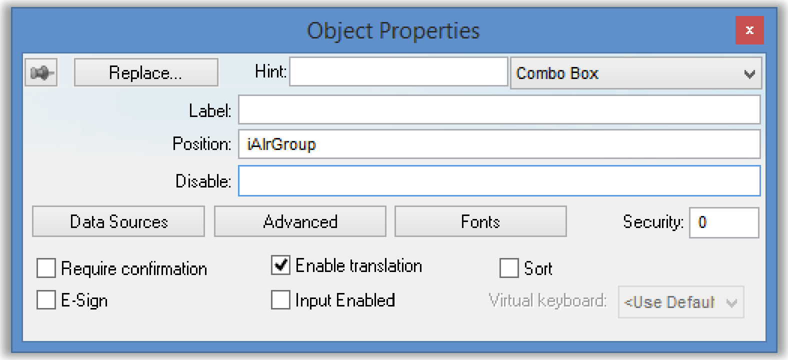


17. In the Graphics tab of the Ribbon, select **Combo Box** under Active Objects.

18. Click and drag on the screen to shape the object as shown below.

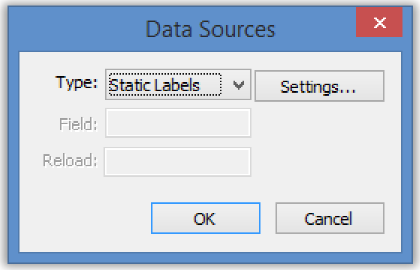


19. Open the Object Properties of the Combo Box and set the field called **Position** to the tag called iAlrGroup as shown below.



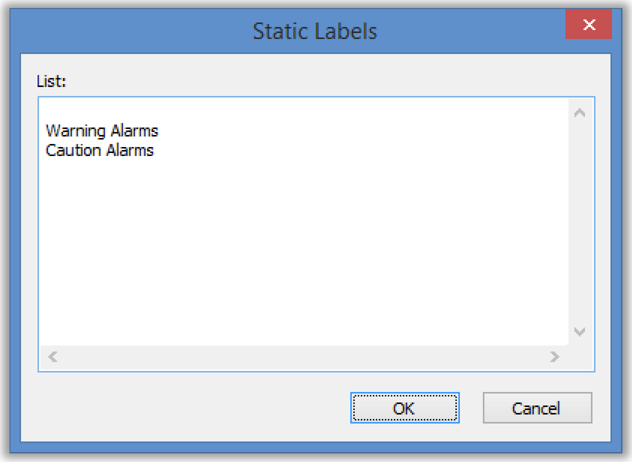
Hint:
Combo Box
Label:
Position: iAlrGroup
Disable:
Security: 0
□ Require confirmation ☑ Enable translation □ Sort
□ E-Sign □ Input Enabled Virtual keyboard: <Use Defaut

20. Click **Data Sources** to open the Combo Box data sources settings as shown below.



Type: Static Labels
Field:
Reload:

21. Click **Settings** and set the Static Labels as shown below.



Warning Alarms
Caution Alarms

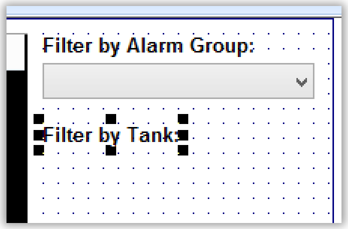
Make sure that the first line is a blank line.

22. Click **Ok** to close the Static Labels settings.

23. Click **Ok** to close the Data Source settings.

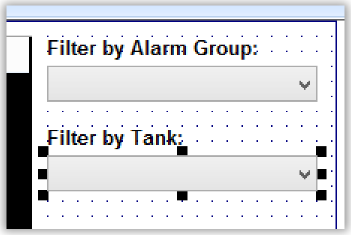
24. In the Graphics tab of the Ribbon, select **Text** under Active Objects.

25. Click on the screen and enter **Filter by Tank:** as shown below.

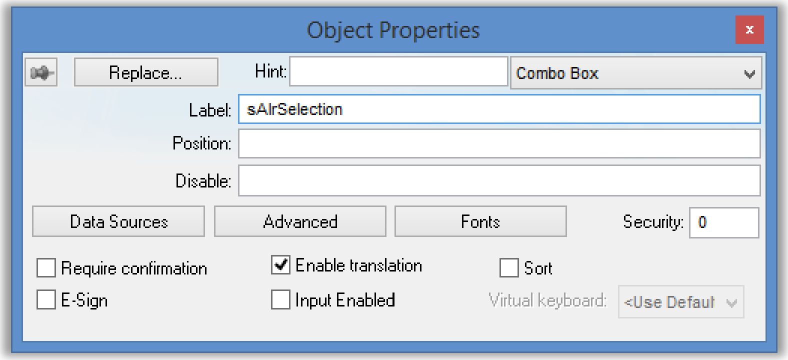


26. In the Graphics tab of the Ribbon, select **Combo Box** under Active Objects.

27. Click and drag on the screen to shape the object as shown below.



28. Open the Object Properties of the Combo Box and set the field called **Label** to the tag called sAlrSelection as shown below.



Hint:

Combo Box

Label: sAlrSelection

Position:

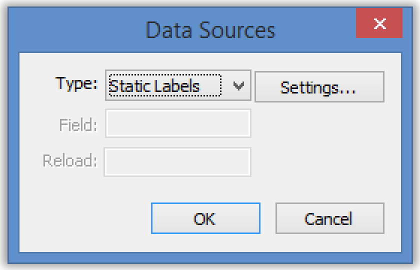
Disable:

Security: 0

□ Require confirmation ☑ Enable translation □ Sort

□ E-Sign □ Input Enabled Virtual keyboard: <Use Defaul

29. Click **Data Sources** to open the Combo Box data sources settings as shown below.

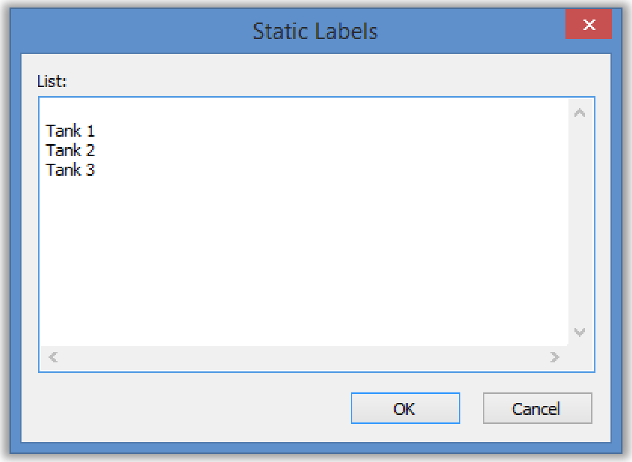


Type: Static Labels

Field:

Reload:

30. Click Settings and set the Static Labels as shown below.



List:

Tank 1
Tank 2
Tank 3

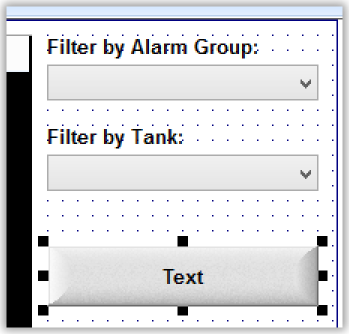
Make sure that the first line is a blank line.

31. Click **Ok** to close the Static Labels settings.

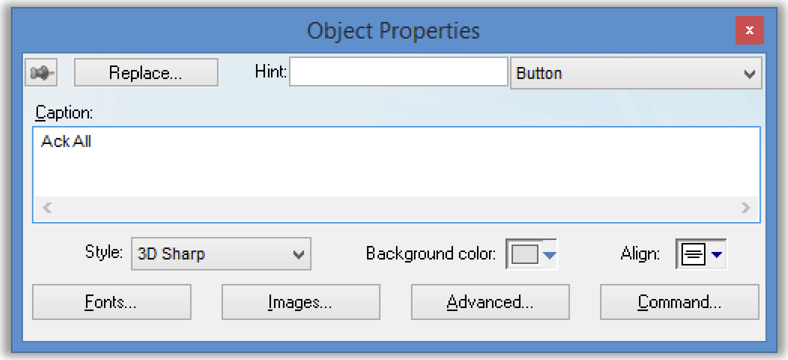
32. Click **Ok** to close the Data Source settings.

33. In the Graphics tab of the Ribbon, select **Button** under Active Objects.

34. Click on the screen and draw the button as shown below.



35. Open the Object Properties of the Button and set the field called **Caption** as shown below.


Object Properties

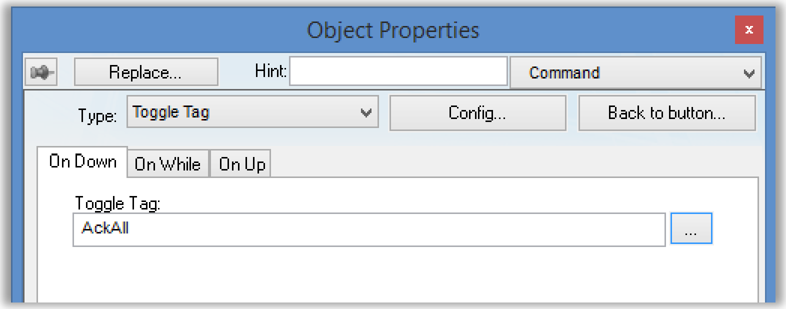
Replace... Hint: Button

Caption:
Ack All

Style: 3D Sharp Background color: Align:

Fonts... Images... Advanced... Command...

36. Click Command to open the Command object properties and set them as shown below.


Object Properties

Replace... Hint: Command

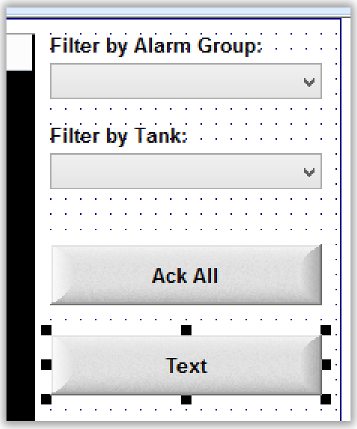
Type: Toggle Tag Config... Back to button...

On Down On While On Up

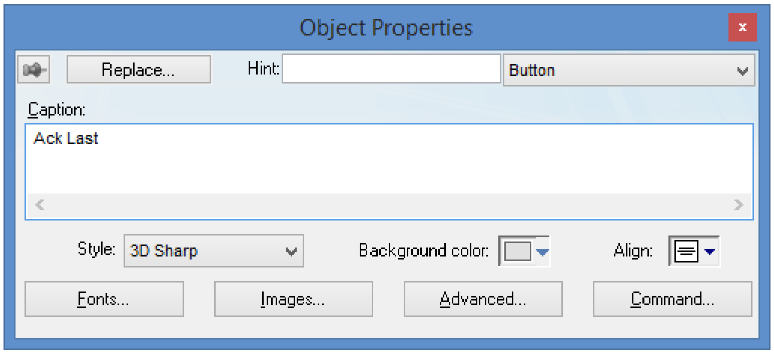
Toggle Tag:
AckAll

37. In the Graphics tab of the Ribbon, select **Button** under Active Objects.

38. Click on the screen and draw the button as shown below.



39. Open the Object Properties of the Button and set the field called **Caption** as shown below.

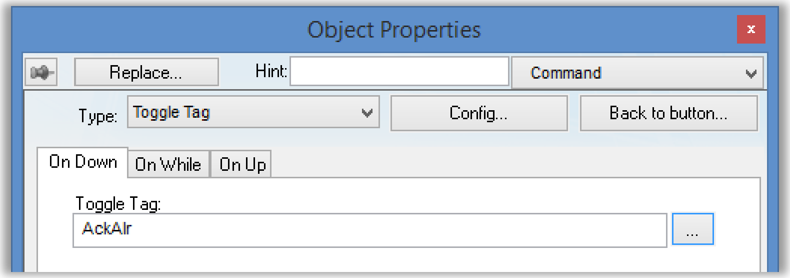


Button

Caption:
Ack Last

Style: 3D Sharp

40. Click Command to open the Command object properties and set them as shown below.

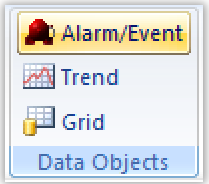


Command

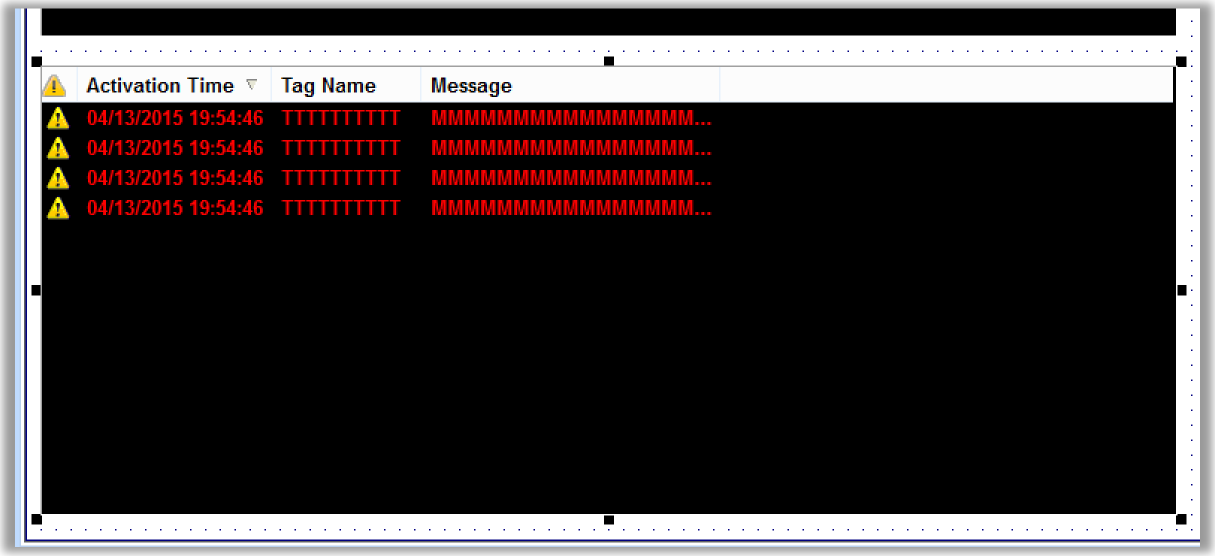
Type: Toggle Tag

Toggle Tag:
AckAlr

41. In the Graphics tab of the Ribbon, select **Alarm/Event** under Data Objects.

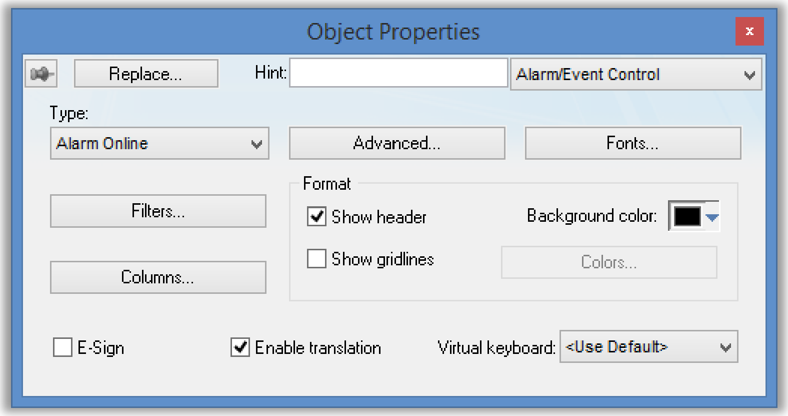


42. Click and drag on the screen to shape the object as shown below.



⚠	Activation Time ▽	Tag Name	Message
⚠	04/13/2015 19:54:46	TTTTTTTTTT	MMMMMMMMMMMMMMMMM...
⚠	04/13/2015 19:54:46	TTTTTTTTTT	MMMMMMMMMMMMMMMMM...
⚠	04/13/2015 19:54:46	TTTTTTTTTT	MMMMMMMMMMMMMMMMM...
⚠	04/13/2015 19:54:46	TTTTTTTTTT	MMMMMMMMMMMMMMMMM...

43. Double-click on the object to open the properties.



Object Properties

Replace... Hint: Alarm/Event Control

Type: Alarm Online

Advanced... Fonts...

Filters...

Columns...

Format

☑ Show header

Background color:

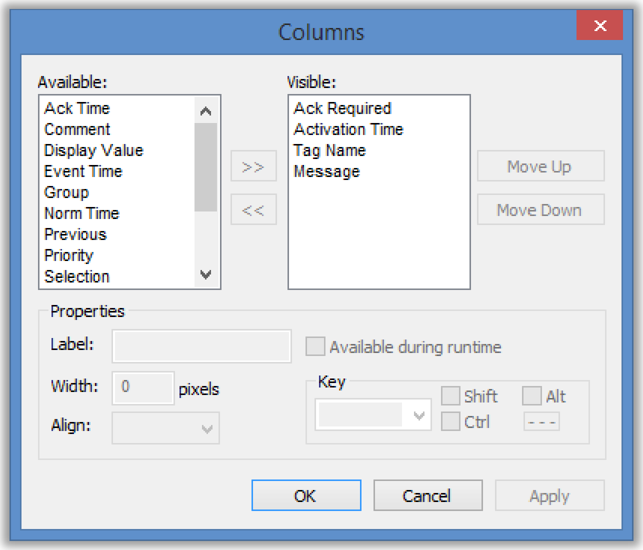
☐ Show gridlines

Colors...

☐ E-Sign ☑ Enable translation Virtual keyboard: <Use Default>

44. In **Type**, configure **Alarm History**.

45. Click on **Columns**. The Columns dialog is opened.



Columns

Available:

Ack Time
Comment
Display Value
Event Time
Group
Norm Time
Previous
Priority
Selection

Visible:

Ack Required
Activation Time
Tag Name
Message

Properties

Label:

□ Available during runtime

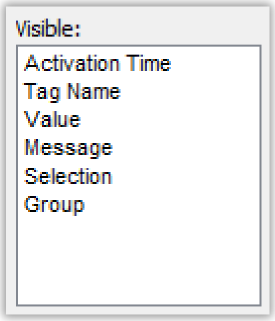
Width: 0 pixels

Align:

Key

□ Shift □ Alt
□ Ctrl - - -

46. Use the left and right double arrow controls and Move Up and Move Down buttons to move Activation Time, Tag Name, Value, Message, Selection and Group from the Available list to the Visible list as shown below.

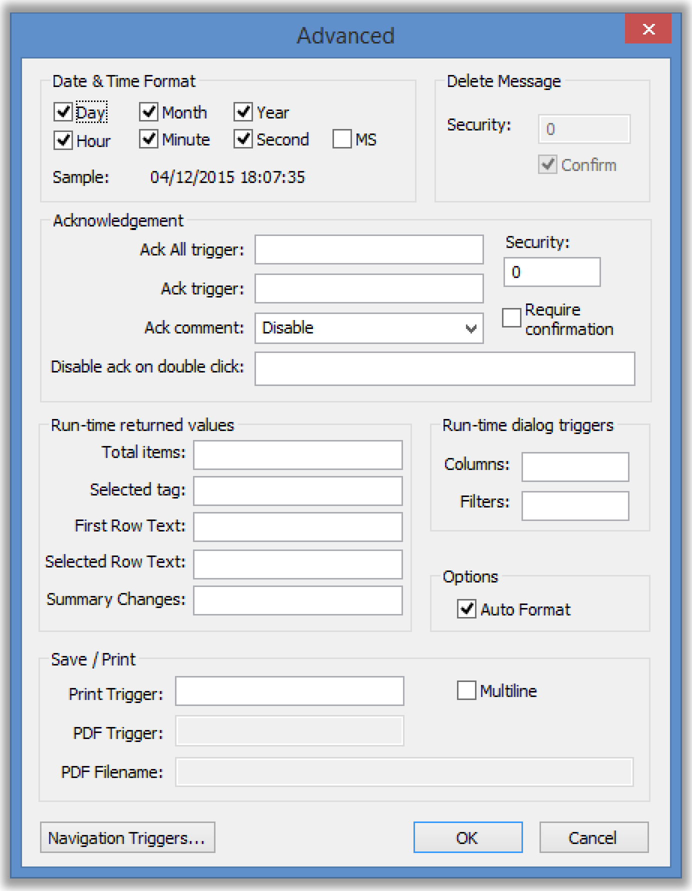


Visible:

Activation Time
Tag Name
Value
Message
Selection
Group

47. Select **OK** to close the Columns dialog.

48. Click on **Advanced**. The Advanced settings is opened.



Advanced

Date & Time Format

☑ Day ☑ Month ☑ Year

☑ Hour ☑ Minute ☑ Second □ MS

Sample: 04/12/2015 18:07:35

Delete Message

Security: 0

☑ Confirm

Acknowledgement

Ack All trigger:

Ack trigger:

Ack comment: Disable

Disable ack on double click:

Security: 0

□ Require confirmation

Run-time returned values

Total items:

Selected tag:

First Row Text:

Selected Row Text:

Summary Changes:

Run-time dialog triggers

Columns:

Filters:

Options

☑ Auto Format

Save / Print

Print Trigger:

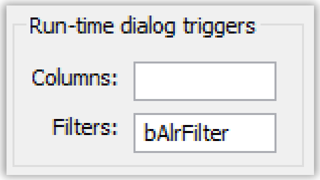
PDF Trigger:

PDF Filename:

□ Multiline

Navigation Triggers... OK Cancel

49. Configure the Run-time dialog triggers section as shown below.



Run-time dialog triggers

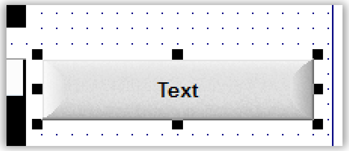
Columns:

Filters: bAlrFilter

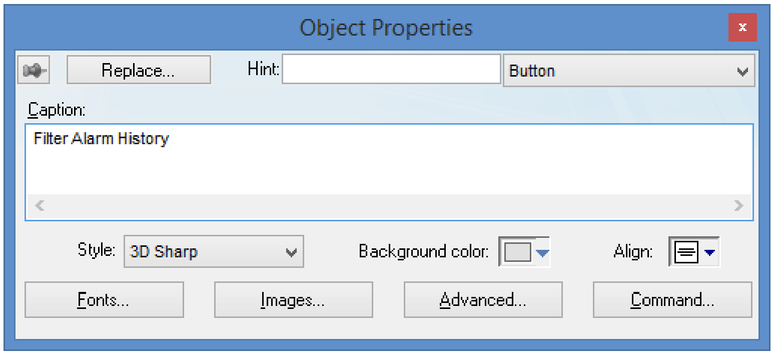
50. Select **OK** to close the Advanced settings.

51. In the Graphics tab of the Ribbon, select **Button** under Active Objects.

52. Click on the screen and draw the button as shown below.



53. Open the Object Properties of the Button and set the field called **Caption** as shown below.



Hint:

Button

Caption:

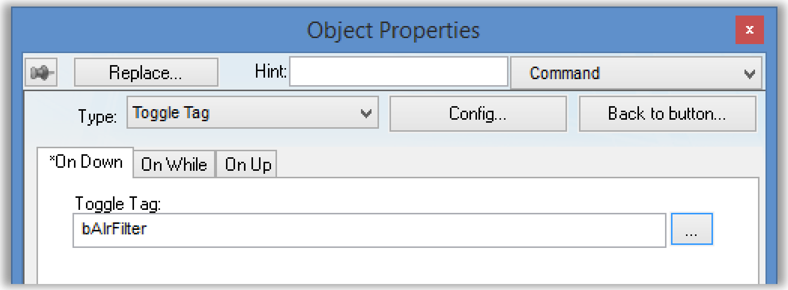
Filter Alarm History

Style: 3D Sharp

Background color:

Align:

54. Click Command to open the Command object properties and set them as shown below.



Hint:

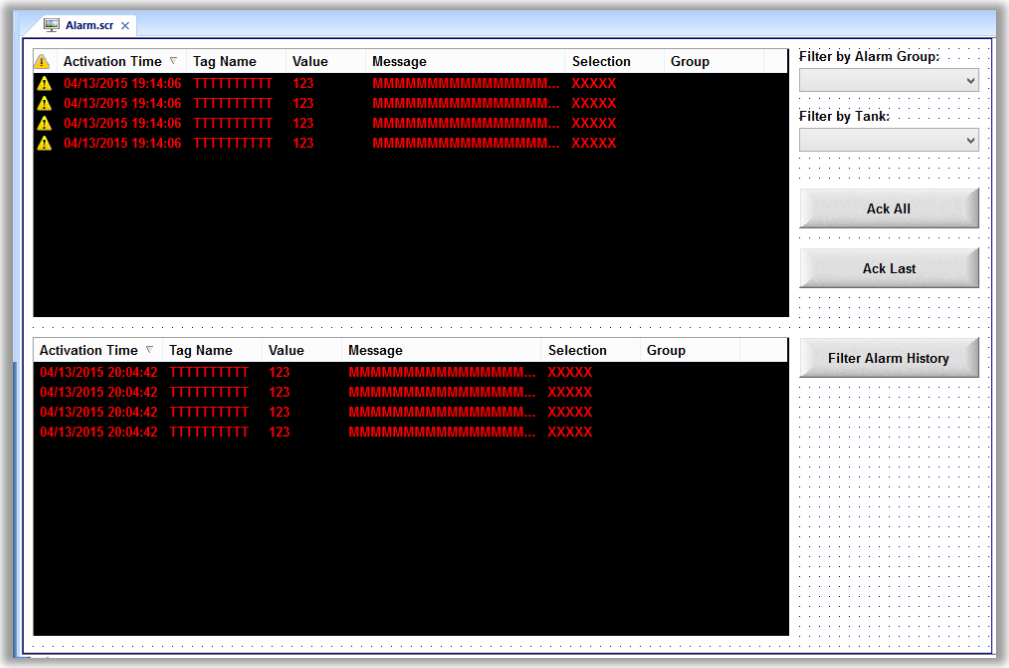
Command

Type: Toggle Tag

Toggle Tag:

bAlrFilter

55. When you are done the Alarm screen should look similar to the following.



⚠	Activation Time	Tag Name	Value	Message	Selection	Group
⚠	04/13/2015 19:14:06	TTTTTTTTTT	123	MMMMMMMMMMMMMMMMM...	XXXXX	
⚠	04/13/2015 19:14:06	TTTTTTTTTT	123	MMMMMMMMMMMMMMMMM...	XXXXX	
⚠	04/13/2015 19:14:06	TTTTTTTTTT	123	MMMMMMMMMMMMMMMMM...	XXXXX	
⚠	04/13/2015 19:14:06	TTTTTTTTTT	123	MMMMMMMMMMMMMMMMM...	XXXXX	

Filter by Alarm Group:

Filter by Tank:

Activation Time	Tag Name	Value	Message	Selection	Group
04/13/2015 20:04:42	TTTTTTTTTT	123	MMMMMMMMMMMMMMMMM...	XXXXX	
04/13/2015 20:04:42	TTTTTTTTTT	123	MMMMMMMMMMMMMMMMM...	XXXXX	
04/13/2015 20:04:42	TTTTTTTTTT	123	MMMMMMMMMMMMMMMMM...	XXXXX	
04/13/2015 20:04:42	TTTTTTTTTT	123	MMMMMMMMMMMMMMMMM...	XXXXX	

56. Save your changes and close the ALARMS screen.

Verification and Test

1. If your runtime is on a remote machine, then use the the Remote Management tools on the Home tab of the Ribbon to download your project to the remote machine.

2. Start the runtime and use the tank valves to fill the tanks to create some alarms.
3. Open the ALARMS screen and verify that it is displaying the expected alarms.

SECTION 2 - EVENTS

This section describes the logging and event-retrieval features of BLUE Open Studio.

Concept

Event logging is disabled by default to conserve runtime resources. To enable the saving of events to the history file, use the Event Logger in the Project Explorer.

When you enable the Event Logger you must configure the types of events to log.

Event Type	Description
Security System	Events generated by your project's security system, including: • Log On / Log Off users • User created/removed by calling the CreateUser or RemoveUser functions • User blocked/unblocked by calling the BlockUser or UnblockUser functions • User blocked by the security system after several attempts to enter an invalid password • Password expired • Password modified • Invalid Log On attempt
Display	Open Screen and Close Screen events.
Recipe	Recipes loaded, saved, initialized, or deleted.
Report	Reports saved to disk or sent to printer.
Custom Messages	Events generated by calling the SendEvent function.
System Warning	Various runtime warnings and errors, including: • Errors that occur when sending alarms by email • Tag was blocked/unblocked • Division by zero • Connection/Disconnection of the remote security system
Tag Change	Log changes to a tag’s value.

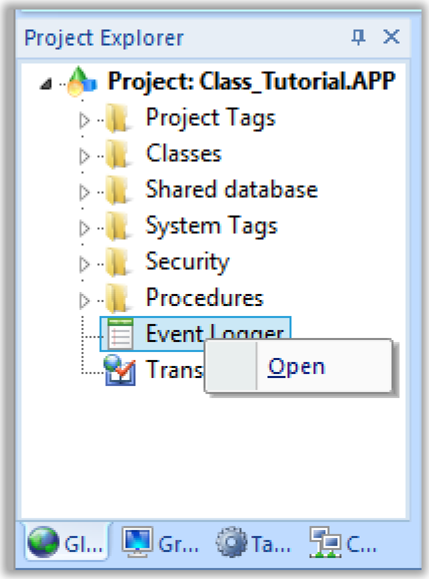
The Alarm/Event Control screen object, available on the Graphics tab of the ribbon, must be created and configured in a screen in order to display events.

LAB 15 - WORKING WITH EVENTS

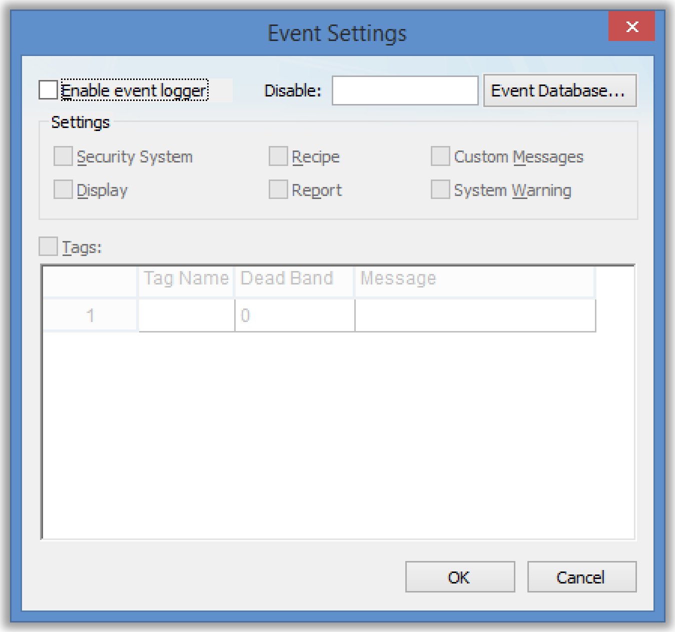
In this lab you will configure the Event Logger to log system events and tag change events when the tank valve controls are opened or closed.

Event Logger

1. On the Global tab of the Project Explorer, right-click on **Event Logger**, and then select **Open** on the shortcut menu.



2. The Event Settings dialog is opened.



3. Select **Enable event logger**.

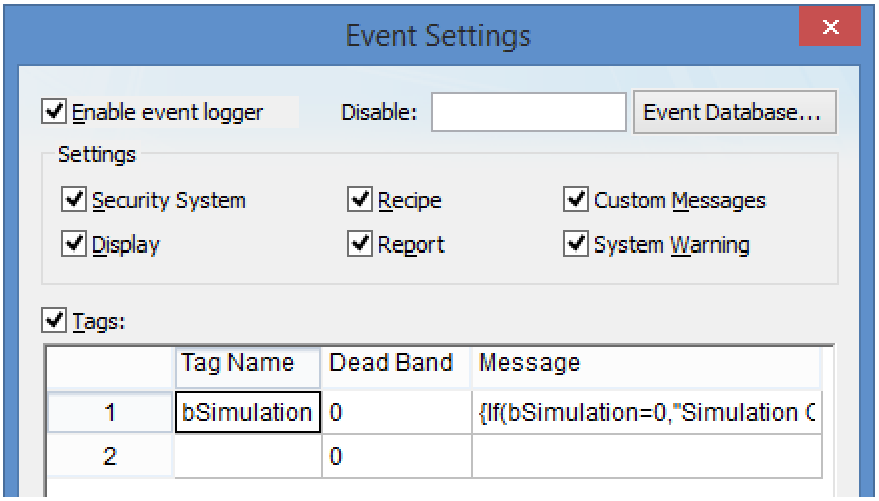
4. In the Settings group, select **Security System**, **Display**, **Recipe**, **Report**, **Custom Messages**, and **System Warning**.

5. Select **Tags**.

6. In the **Tag Name** column, enter `bSimulation`.

7. In the **Message** column, enter `{If(bSimulation=0,"Simulation Off","Simulation On")}`.

8. The Event Settings should look similar to the following,



Event Settings

☑ Enable event logger

Disable:

Settings

☑ Security System ☑ Recipe ☑ Custom Messages

☑ Display ☑ Report ☑ System Warning

☑ Tags:

	Tag Name	Dead Band	Message
1	bSimulation	0	{If(bSimulation=0,"Simulation C
2		0	

9. Click **OK** to close the Event Settings dialog.

Alarm/Event Object

1. Open the Template screen.

2. Click the Application button and click **Save As** on the Application menu. The Save As dialog is opened.

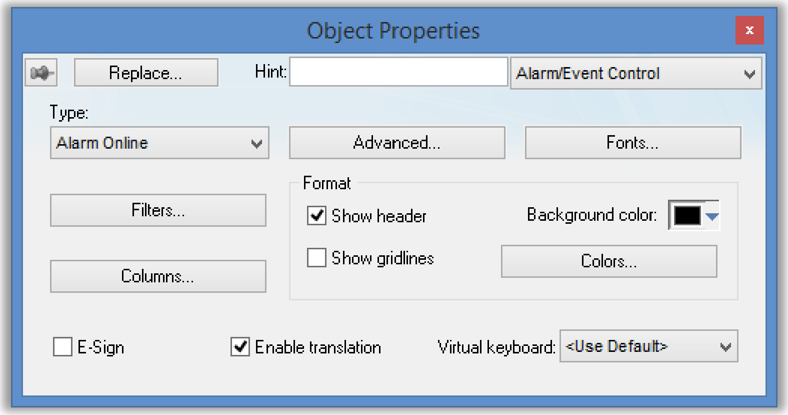
3. In the **File name** field, enter `Events`.

4. Click **Save**. The Save As dialog is closed.

5. On the Graphics tab of the Ribbon in the Data Objects group, click **Alarm/Event**.

6. Click and drag the mouse on the EVENTS screen to draw the Alarm/Event object.

7. Double-click on the Alarm/Event object to open the object properties dialog.



Hint:
Alarm/Event Control

Type: Alarm Online

☑ Show header

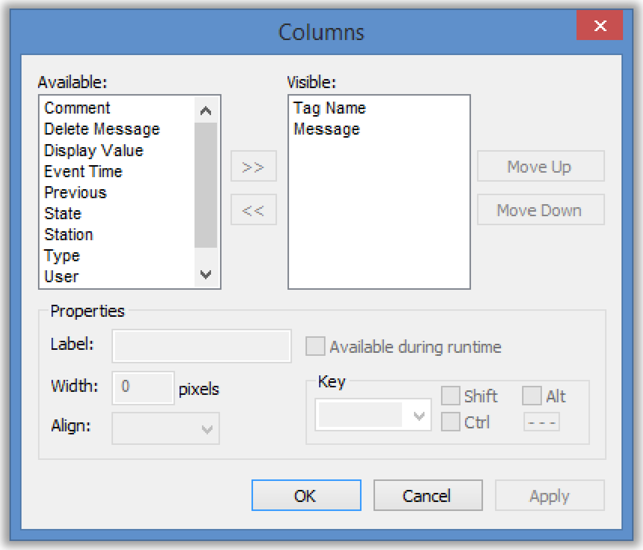
□ Show gridlines

Background color: ■

□ E-Sign ☑ Enable translation Virtual keyboard: <Use Default>

8. In the **Type** field, select **Event**.

9. Click **Columns**. The Columns dialog is opened.



Available:
Comment
Delete Message
Display Value
Event Time
Previous
State
Station
Type
User

Visible:
Tag Name
Message

Label:

□ Available during runtime

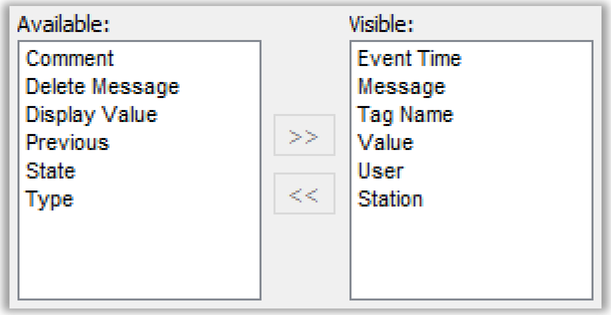
Width: 0 pixels

Align:

□ Shift □ Alt

□ Ctrl - - -

10. Use the left and right double arrow controls and Move Up and Move Down buttons to move Event Time, Value, User, and Station from the Available list to the Visible list as shown below.



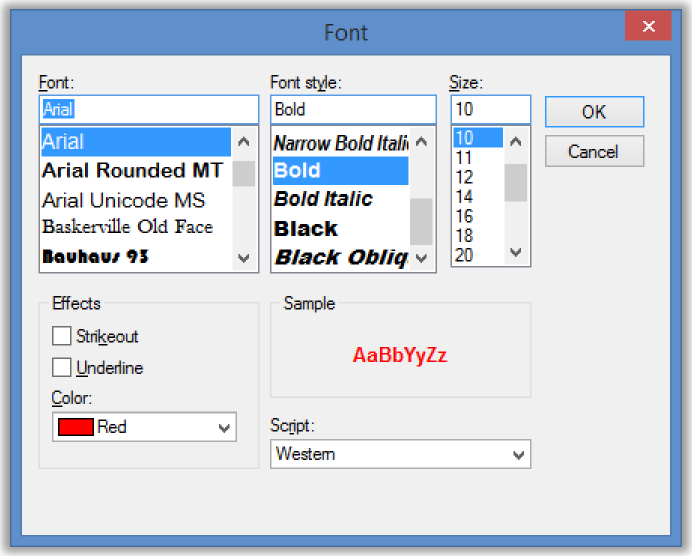
Available:
Comment
Delete Message
Display Value
Previous
State
Type

Visible:
Event Time
Message
Tag Name
Value
User
Station

11. Click **OK** to close the Columns dialog.

12. In **Background Color** field, select White.

13. Click **Fonts**. The Font dialog is opened.



Font: Arial

Font style: Bold

Size: 10

Color: Red

Script: Western

Sample: AaBbYyZz

14. In the **Color** field, select Black.

15. Click **OK** to close the Font dialog.

16. Close the Alarm/Event object properties dialog

17. Save and close the EVENTS screen.

Verification and Test

1. If your runtime is on a remote machine, then use the the Remote Management tools on the Home tab of the Ribbon to download your project to the remote machine.

2. Start the runtime and exercise your project.

3. Open the EVENTS screen and verify that it is displaying the expected events.

Module 8 - Data Logging and Display

SECTION 1 - TREND

This section uses a step-by-step tutorial to explain how to insert the Trend task and Trend object.

In the Trend task, you will configure the automatic data logging of tag values.

The Trend Object will graphically show the real-time, as well as stored values.

Concept

The Trend Control object displays data points (values) from different sources of data in a graphical format.

The key features offered by the trend control object are:

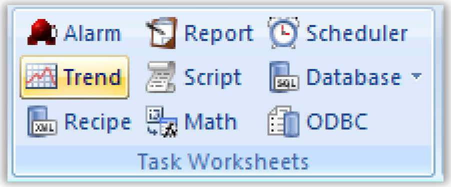
- Display of several panels simultaneously.
- Support for different data sources, such as Tag, Batch, Database and text file.

LAB 16 - WORKING WITH TRENDS

In this lab you will configure the Trend task to save your tank analog values every second and display them to the screen using the Trend object.

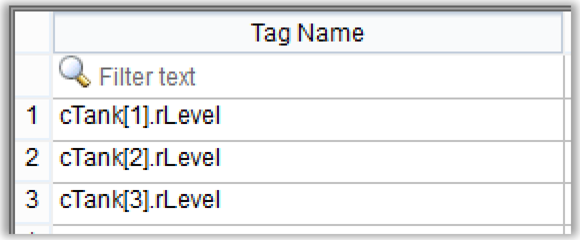
Trend Task – Proprietary Format

1. In the Insert tab of the Ribbon select **Trend** under Task Worksheets.



2. The TREND001 worksheet is opened in a new tab in the workspace.

3. On the **Tag Name**, configure the Tags **cTank[1].rLevel**, **cTank[2].rLevel** and **cTank[3].rLevel**. Leave everything else as default.



	Tag Name
	Filter text
1	cTank[1].rLevel
2	cTank[2].rLevel
3	cTank[3].rLevel

4. Save the Worksheet as 1.

Trend Object

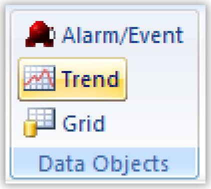
1. Open the Template screen.

2. Click the Application button and click **Save As** on the Application menu. The Save As dialog is opened.

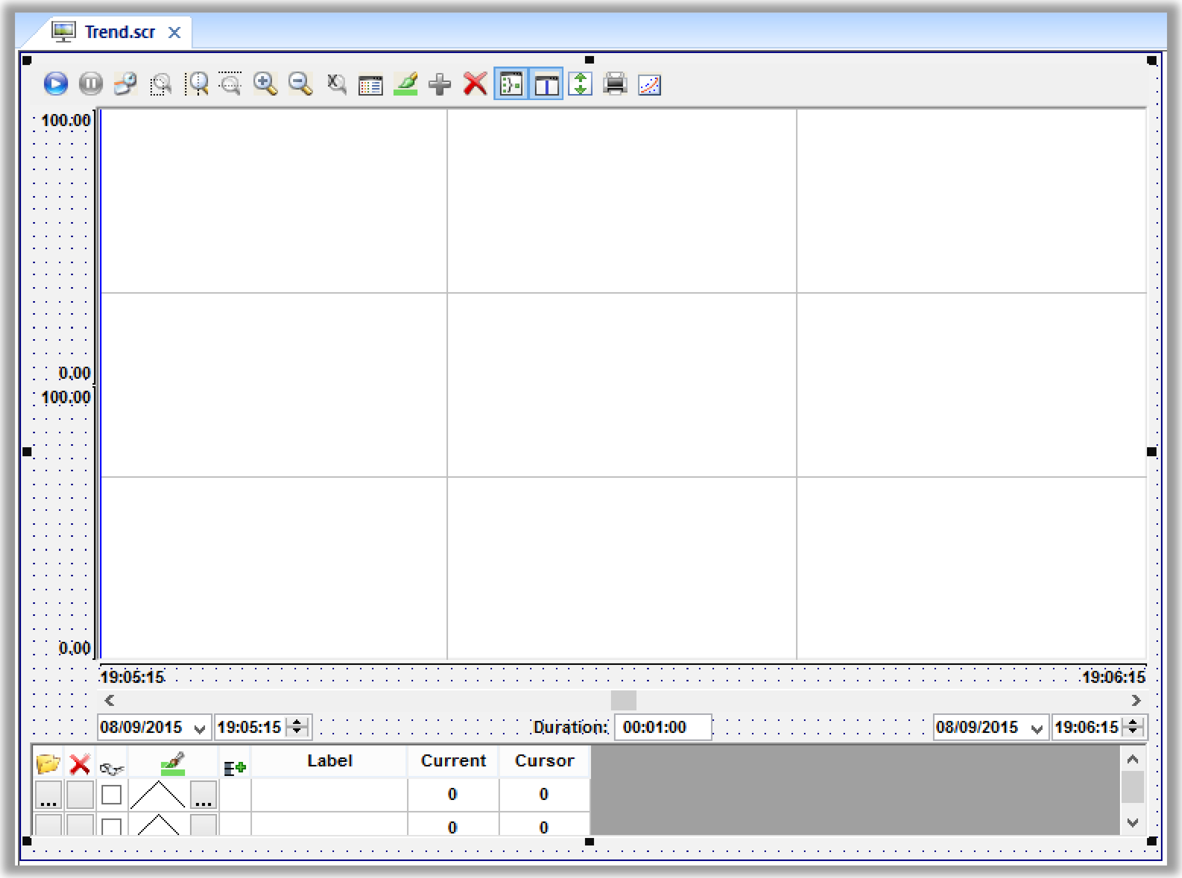
3. In the **File name** field, enter `Trend`.

4. Click **Save**. The Save As dialog is closed.

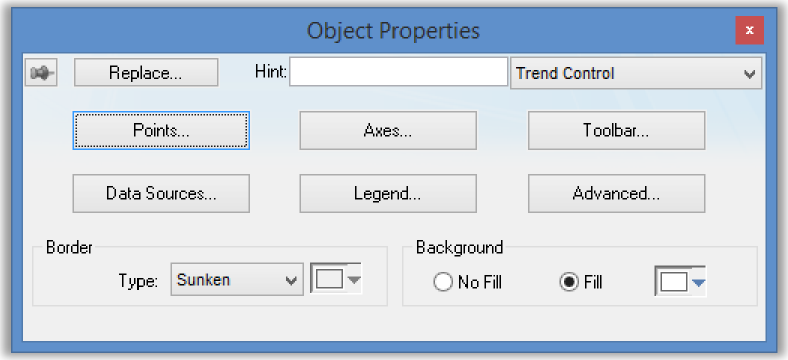
5. In the Graphic Ribbon, select **Trend** under Data Objects.



6. Click and drag on the TREND screen to shape the object.



7. Double-click on the object to open the properties.



Hint:

Trend Control

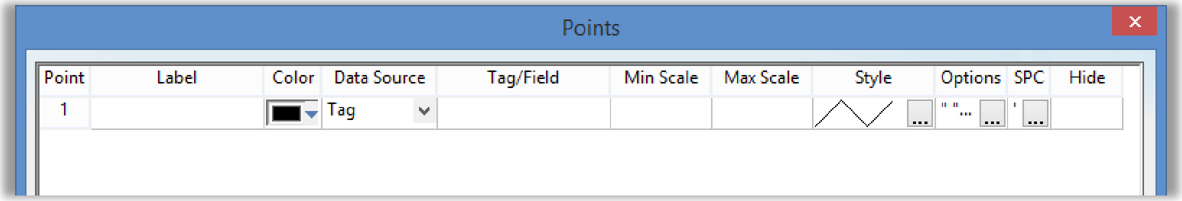
Border

Type: Sunken

Background

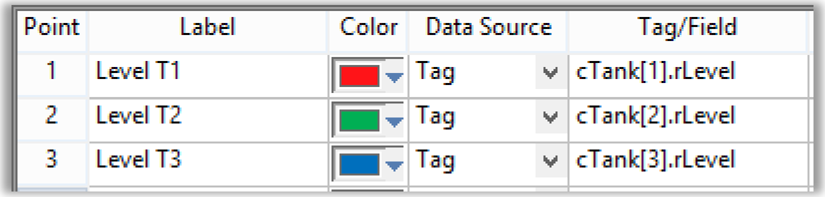
○ No Fill ● Fill

8. Click Points. The Points dialog is opened.



Point	Label	Color	Data Source	Tag/Field	Min Scale	Max Scale	Style	Options	SPC	Hide
1		■	Tag				...	" "... ...	' ...	

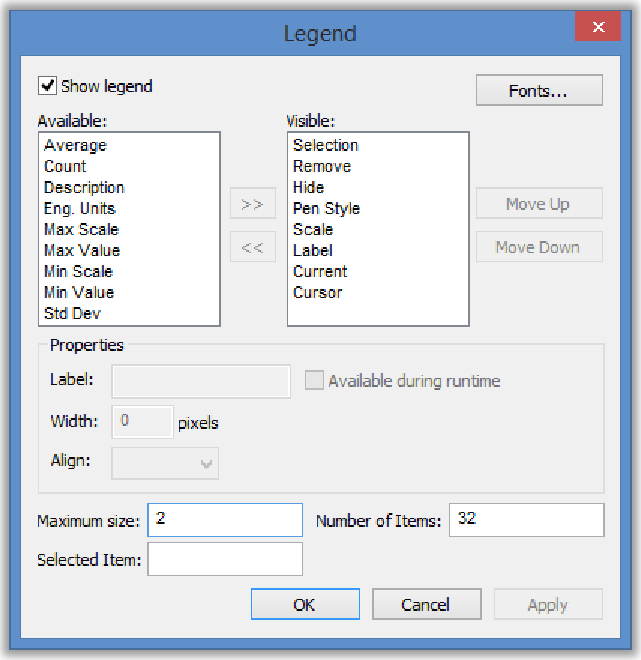
9. Configure the Points dialog according to the figure below.



Point	Label	Color	Data Source	Tag/Field
1	Level T1	🟥	Tag	cTank[1].rLevel
2	Level T2	🟩	Tag	cTank[2].rLevel
3	Level T3	🟦	Tag	cTank[3].rLevel

10. Click **OK**.

11. Click **Legend**. The Legend dialog is opened.



Legend

☑ Show legend

Fonts...

Available:
Average
Count
Description
Eng. Units
Max Scale
Max Value
Min Scale
Min Value
Std Dev

>>
<<

Visible:
Selection
Remove
Hide
Pen Style
Scale
Label
Current
Cursor

Move Up
Move Down

Properties

Label:

□ Available during runtime

Width: 0 pixels

Align:

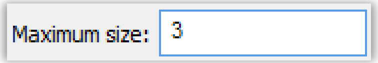
Maximum size: 2

Number of Items: 32

Selected Item:

OK Cancel Apply

12. Change the Maximum size to 3 since we configured three points.



Maximum size: 3

13. Click **OK**.

14. Save your changes and close the TREND screen.

Verification and Test

1. If your runtime is on a remote machine, then use the the Remote Management tools on the Home tab of the Ribbon to download your project to the remote machine.

2. Start the runtime and use the tank valves to fill and empty the tanks.

3. Open the TREND screen and verify that it is displaying the expected data.

SECTION 2 - GRID

This section explains, using a step-by-step tutorial, how to configure a Grid object that will be used for training.

You will configure a Grid data object to read Access Database values, which was configured in prior chapter.

Concept

The Grid object allows you to read/write data in tab format of the data source configured in the object itself.

Sources may be:

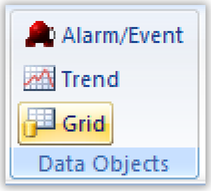
- Text file
- Class Tag
- Database

LAB 17 - USING THE GRID OBJECT

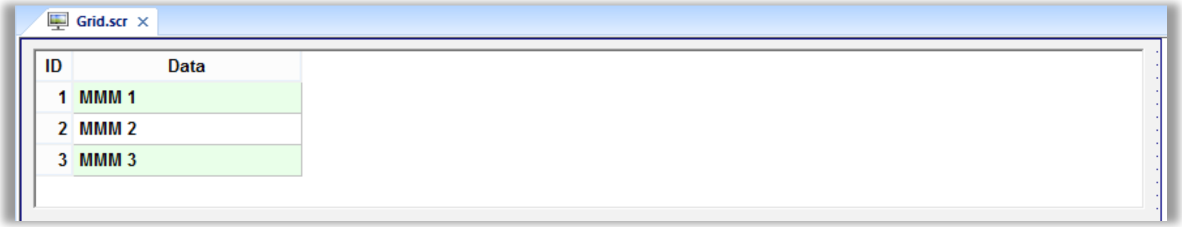
In this lab you will create a Grid object to disaply the historic values stored by the Trend task configured in the previous lab.

Grid Object

1. Open the Template screen.
2. Click the Application button and click **Save As** on the Application menu. The Save As dialog is opened.
3. In the **File name** field, enter `Grid`.
4. Click **Save**. The Save As dialog is closed.
5. On the Graphics tab of the Ribbon in the Data Objects group, select **Grid**.

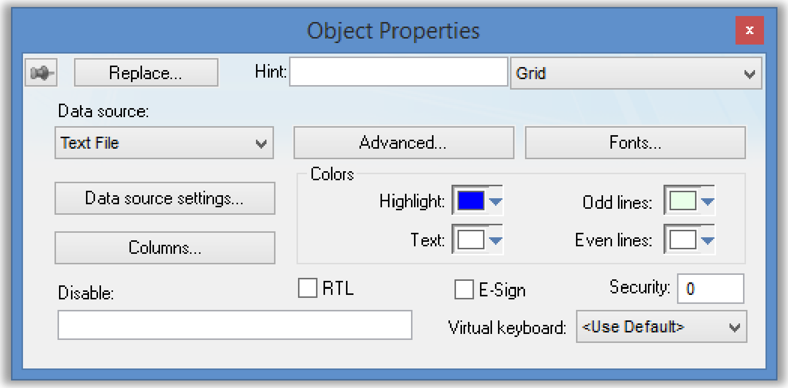


6. Click and drag on the GRID screen to shape the object.



ID	Data
1	MMM 1
2	MMM 2
3	MMM 3

7. Double-click on the object to open the properties.



Hint: Grid

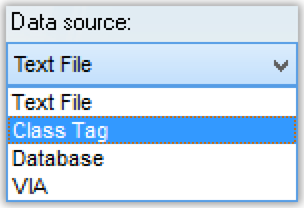
Data source: Text File

□ RTL □ E-Sign Security: 0

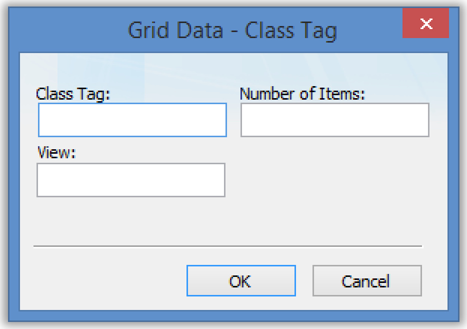
Disable:

Virtual keyboard: <Use Default>

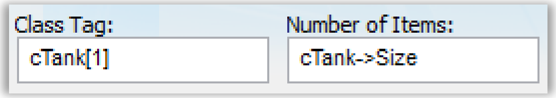
8. Click the **Data Source** field, select **Class Tag**.



9. Click the **Data Source Settings** button. The Grid Data - Class Tag dialog is opened.



10. Configure the fields called **Class Tag** and **Number of Items** as shown below.

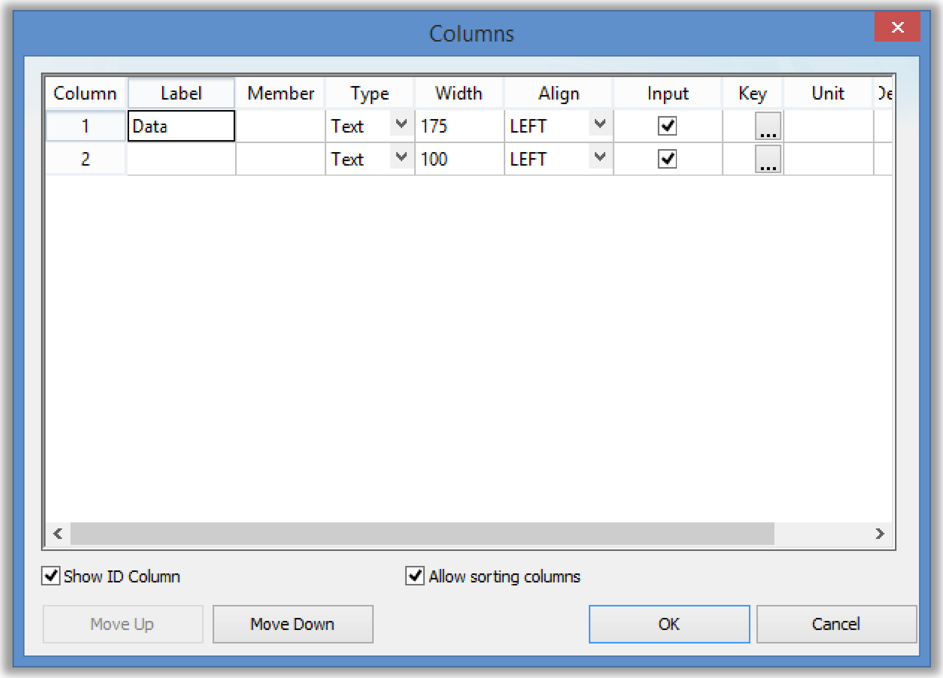


Class Tag: cTank[1]

Number of Items: cTank->Size

11. Click **OK** to close the Grid Data - Class Tag dialog

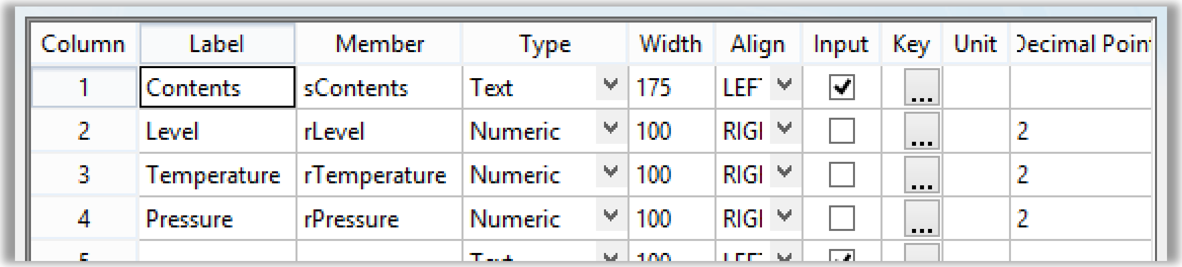
12. Click **Columns**. The Columns dialog is opened.



Column	Label	Member	Type	Width	Align	Input	Key	Unit	De
1	Data		Text	175	LEFT	☑	...		
2			Text	100	LEFT	☑	...		

☑ Show ID Column ☑ Allow sorting columns

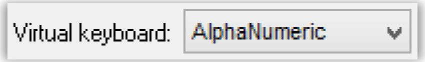
13. Configure the Columns dialog as shown below.



Column	Label	Member	Type	Width	Align	Input	Key	Unit	Decimal Poin
1	Contents	sContents	Text	175	LEF	☑	...		
2	Level	rLevel	Numeric	100	RIGI	□	...		2
3	Temperature	rTemperature	Numeric	100	RIGI	□	...		2
4	Pressure	rPressure	Numeric	100	RIGI	□	...		2
5			Text	100	LEF	☑			

14. Click **OK** to close the Columns dialog.

15. Click the **Virtual keyboard** combo box and select **AlphaNumeric**.



Virtual keyboard: AlphaNumeric

16. Save changes and close GRID screen.

Verification and Test

1. If your runtime is on a remote machine, then use the the Remote Management tools on the Home tab of the Ribbon to download your project to the remote machine.

2. Start the runtime and use the tank valves to fill and empty the tanks.

3. Open the TREND screen and verify that it is displaying the expected data.

4. Change the Contents column for one or more of your tanks and go back to the Main screen to verify that it was changed in the class tag.

Learning Services

Module 9 - Recipes and Reports

SECTION 1 - RECIPES

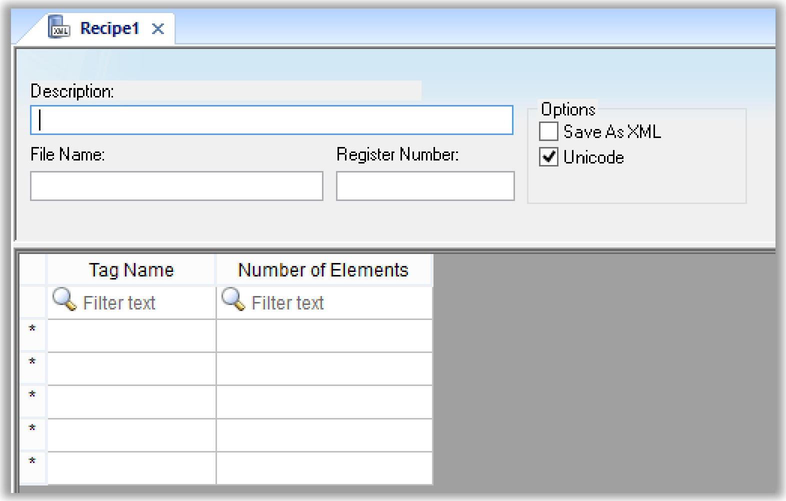
A Recipe worksheet is used to load tag values from or save tag values to an external file during runtime. It is typically used to execute process recipes, but you can store any type of information (such as operation logs, passwords, and so on) in the external file.

Concept

To create a new Recipe worksheet, do one of the following:

- On the Insert tab of the ribbon, in the Task Worksheets group, click Recipe;
- Right-click the Recipes folder in the Project Explorer, and then click Insert on the shortcut menu; or
- Click New on the Application menu, click the File tab, and then select Recipe Worksheet.

To edit an existing Recipe worksheet, double-click it in the Project Explorer.



Options
☐ Save As XML
☑ Unicode

Note: When saving a Recipe worksheet, you can save it using any name you choose (you are not required to use a predefined file name like Recipe1). A configuration file using the extension `.RCP` is created containing the recipe configuration.

The worksheet is divided into two areas:

- Header area (top section), which contains information for the whole group
- Body area (bottom section), where you define each tag in the group.

Use the Header parameters on this worksheet as follows:

- **Description** field: Type a description of the worksheet, for documentation purposes.
- **File Name** field: Type the name of the external file, using static text (e.g., `File1`) or a tag (e.g., `{FileNameTag}`).

- **Register Number** field: Type a tag to define the register number to be read or written into a DBF file. (This field is for legacy purposes only and is no longer used.)
- **Save As XML** checkbox: Select (check) to save information in XML format, or deselect (uncheck) to save in DAT format.

  You can load information in a .DAT file into different tags using a second Recipe worksheet, but you must load information in an .XML file into tags with the same name as the tag from which the data originated.
- **Unicode** checkbox: Click (enable) to save the recipe in Unicode format (two bytes per character) or (disable) to save the recipe in ANSI format (one byte per character).

Use the Body parameters on this worksheet as follows:

- **Tag Name** field: Type tag names to update with file contents or with values to write to a file. If the tag is an array, you must specify the first position to use.

  If the tag is an Array or a Class (or both), then the development application automatically enables every array position and class member by default. To configure a specific array position and/or class member, type it in the Tag Name field as normal (e.g., `level[3].member`).
- **Number of Elements** field: Specify how many positions of the array tag are in use.

  > Pro Tip: You can configure a tag name between curly brackets (`{`*`tagname`*`}`) in this field, allowing the user to dynamically change the Number of Elements configured in the Recipe for each array tag, during runtime.

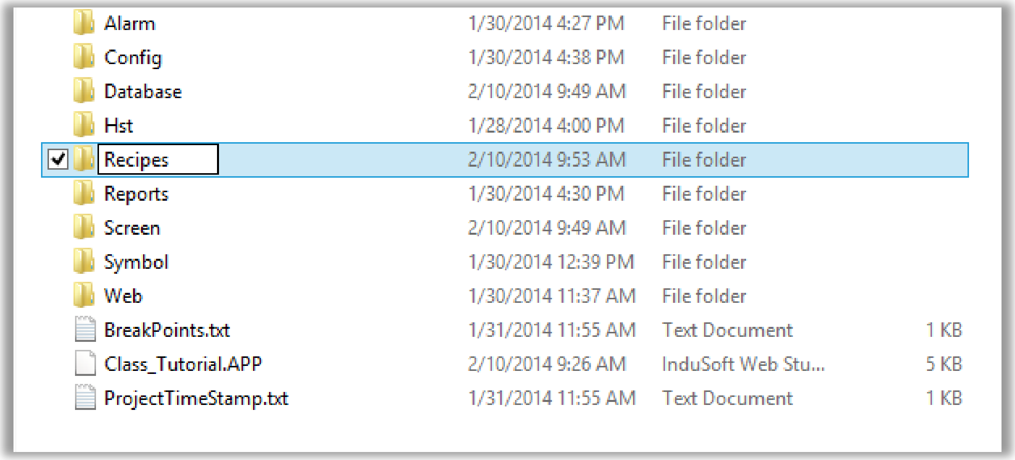
To execute a Recipe worksheet, use the **Recipe()** function anywhere an expression is allowed.

LAB 18 - USING RECIPES

In this lab you will create a Recipe.

Preparing the environment

1. Open Windows Explorer, navigate to your project folder, and create a folder called `Recipes`.

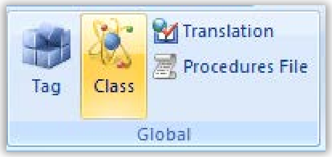


Alarm	1/30/2014 4:27 PM	File folder	
Config	1/30/2014 4:38 PM	File folder	
Database	2/10/2014 9:49 AM	File folder	
Hst	1/28/2014 4:00 PM	File folder	
☑ Recipes	2/10/2014 9:53 AM	File folder	
Reports	1/30/2014 4:30 PM	File folder	
Screen	2/10/2014 9:49 AM	File folder	
Symbol	1/30/2014 12:39 PM	File folder	
Web	1/30/2014 11:37 AM	File folder	
BreakPoints.txt	1/31/2014 11:55 AM	Text Document	1 KB
Class_Tutorial.APP	2/10/2014 9:26 AM	InduSoft Web Stu...	5 KB
ProjectTimeStamp.txt	1/31/2014 11:55 AM	Text Document	1 KB

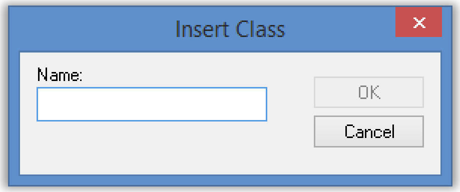
Pro Tip: Alternatively, you could use the built-in function called DirCreate() in the Startup Script to create the Recipes folder when the project is started.

Tags

4. On the Insert tab of the Ribbon, in the Global group, click on **Class**.



The Insert Class dialog is opened.

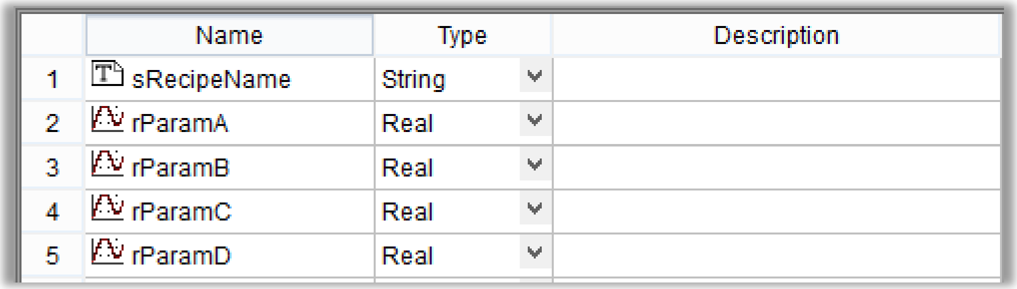


5. In the field called **Name**, enter `RecipeClass`.

Name: RecipeClass

6. Click **OK**. The Insert Class dialog is closed and the Class: RecipeClass worksheet is opened in the Workspace.

7. Configure the class as shown below.

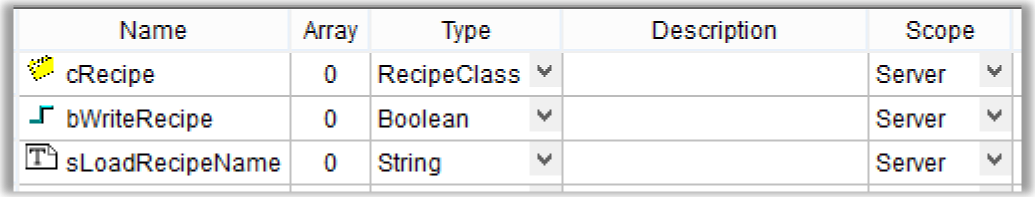


	Name	Type	Description
1	sRecipeName	String	
2	rParamA	Real	
3	rParamB	Real	
4	rParamC	Real	
5	rParamD	Real	

8. Close the Class: RecipeClass worksheet.

9. On the Global tab of the Project Explorer, expand the Project Tags folder, right-click Datasheet View, and select Open on the shortcut menu. The Project Tags worksheet is opened in the Workspace.

10. Create the tags shown below.

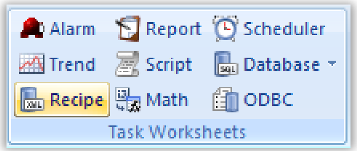


Name	Array	Type	Description	Scope
cRecipe	0	RecipeClass		Server
bWriteRecipe	0	Boolean		Server
sLoadRecipeName	0	String		Server

11. Close the Project Tags worksheet.

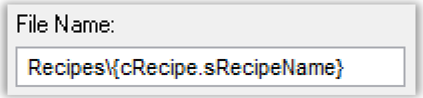
Recipe Task

1. On the Insert tab of the Ribbon, in the Task Worksheets group, click **Recipe**.



A blank Recipe worksheet is opened in the Workspace.

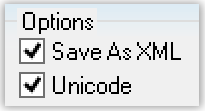
2. In the **File Name** field, enter `Recipes\{cRecipe.sRecipeName}` as shown below.



File Name:

Recipes\{cRecipe.sRecipeName}

3. Enable the options called **Save as XML** and **Unicode** as shown below.

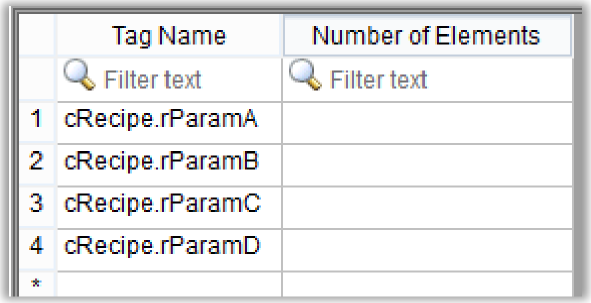


Options

☑ Save As XML

☑ Unicode

4. Add the four Real-type members of the **cRecipe** tag to the Tag Name column of the recipe worksheet as shown below.



	Tag Name	Number of Elements
	Filter text	Filter text
1	cRecipe.rParamA	
2	cRecipe.rParamB	
3	cRecipe.rParamC	
4	cRecipe.rParamD	
*		

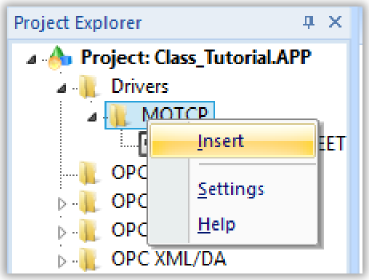
Pro Tip: If you use a class tag to store your recipe tags, then you could simply specify the name of the class tag (e.g., cRecipe) in the Tag Name column instead of enumerating each class member.

5. Save the worksheet and name it `Recipe1`.

6. Close the Recipe1 worksheet.

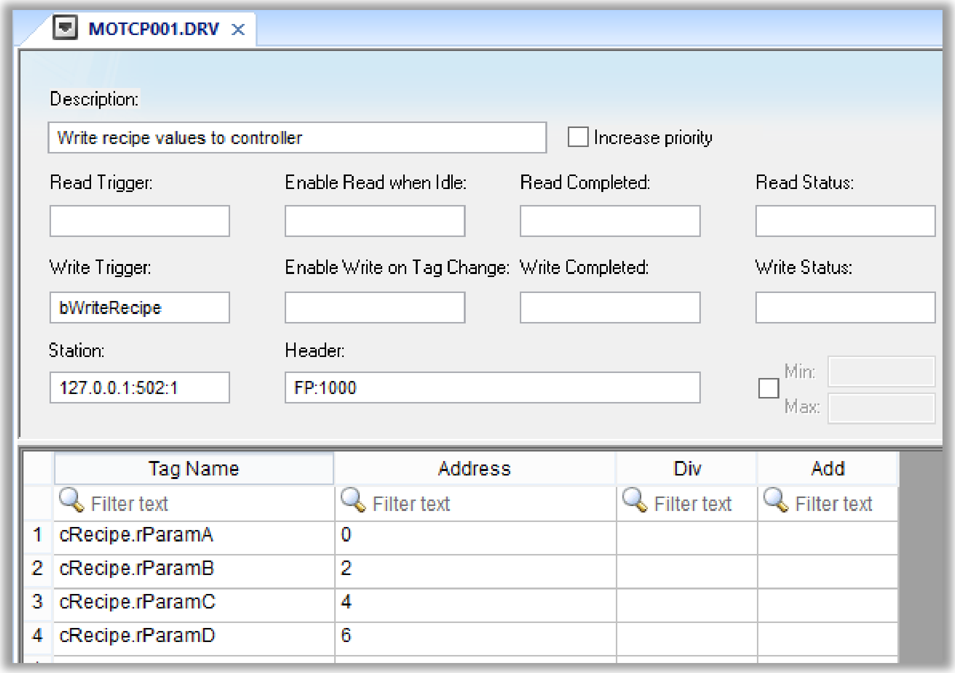
Create a Driver Worksheet

1. On the Comm tab of the Project Explorer, expand the **Drivers** folder, right-click on **MOTCP**, and select **Insert**.



A new standard driver worksheet is opened in the Workspace.

2. Configure the worksheet as shown below or as instructed by the instructor.



Description: Write recipe values to controller

□ Increase priority

Read Trigger:

Enable Read when Idle:

Read Completed:

Read Status:

Write Trigger: bWriteRecipe

Enable Write on Tag Change:

Write Completed:

Write Status:

Station: 127.0.0.1:502:1

Header: FP:1000

□ Min: Max:

	Tag Name	Address	Div	Add
	Filter text	Filter text	Filter text	Filter text
1	cRecipe.rParamA	0		
2	cRecipe.rParamB	2		
3	cRecipe.rParamC	4		
4	cRecipe.rParamD	6		

3. Save the worksheet as sheet number 1 and close the worksheet.

Recipe Screen

1. Open the Template screen.

2. Click the Application button and click **Save As** on the Application menu. The Save As dialog is opened.

3. In the **File name** field, enter `Recipes`.

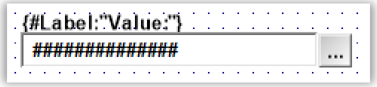
4. Click **Save**. The Save As dialog is closed.

5. On the Graphics tab of the Ribbon, in the Libraries group, click **Symbols**. The Symbols library is opened on the Workspace.

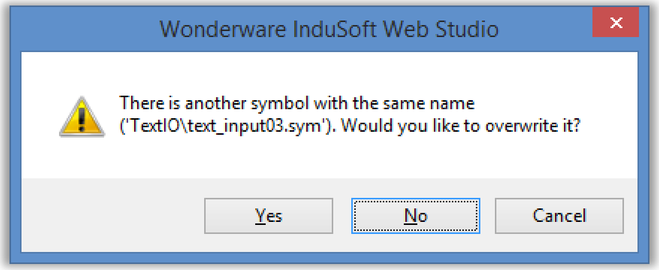
6. Expand the **System Symbols** folder, and then click on the folder called **TextIO**.

7. Select the symbol called **text_input03**.

8. Click on the **Recipes.scc tab** on the Workspace to select the Recipes screen, and then click near the upper left corner on the Recipes screen to paste the symbol.

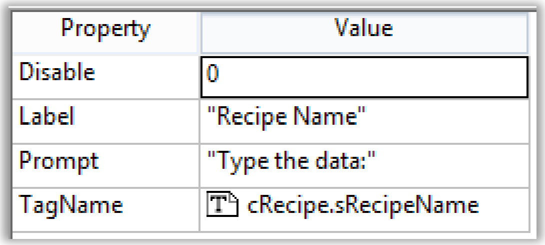


If you see the following message, then click **Yes**.



There is another symbol with the same name ('TextIO\text_input03.sym'). Would you like to overwrite it?

9. Open the Object Properties of the text_input03 symbol and configure them as shown below.

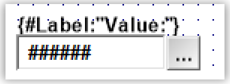


Property	Value
Disable	0
Label	"Recipe Name"
Prompt	"Type the data:"
TagName	cRecipe.sRecipeName

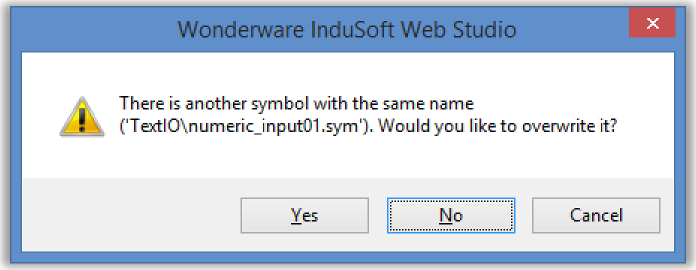
10. Close the Object Properties dialog.

11. Click on the **Symbols** tab on the Workspace, and select the symbol called **numeric_input1**.

12. Click on the **Recipes.scc** tab on the Workspace to select the Recipes screen, and then click below the previous symbol on the Recipes screen to paste the symbol.

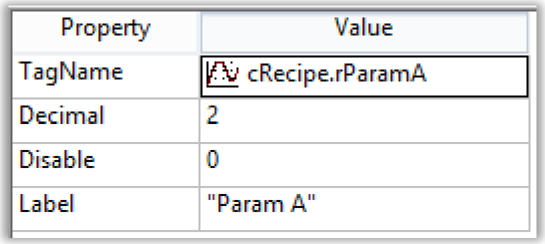


If you see the following message, then click **Yes**.



There is another symbol with the same name ('TextIO\numeric_input01.sym'). Would you like to overwrite it?

13. Open the Object Properties of the symbol and configure them as shown below.



Property	Value
TagName	cRecipe.rParamA
Decimal	2
Disable	0
Label	"Param A"

14. Close the Object Properties dialog.

15. Repeat steps 10 through 14 for the tags called cRecipe.rParamB, cRecipe.rParamC, and cRecipe.rParamD.

16. Close the Symbols tab on the Workspace.

17. On the Graphics tab of the Ribbon, in the Active Objects group, click **Button** and add a button to the Recipes screen to the right of the upper most symbol.

18. Open the Object Properties of the button.

19. In the field called **Caption**, enter `Save`.

20. Click **Command**. The Command animation object properties are displayed.

21. Set **Type** to **VBScript** and, enter the following in the body of the On Down event.

```
$Recipe("Save:Recipe1.rcp")
```

22. Close the Object Properties dialog.

23. On the Graphics tab of the Ribbon in the Active Objects group, select **Button** and add a button to the Recipes screen below the previous button.

24. Open the Object Properties of the button.

25. In the **Caption** field, enter `Load`.

26. Click **Command**.

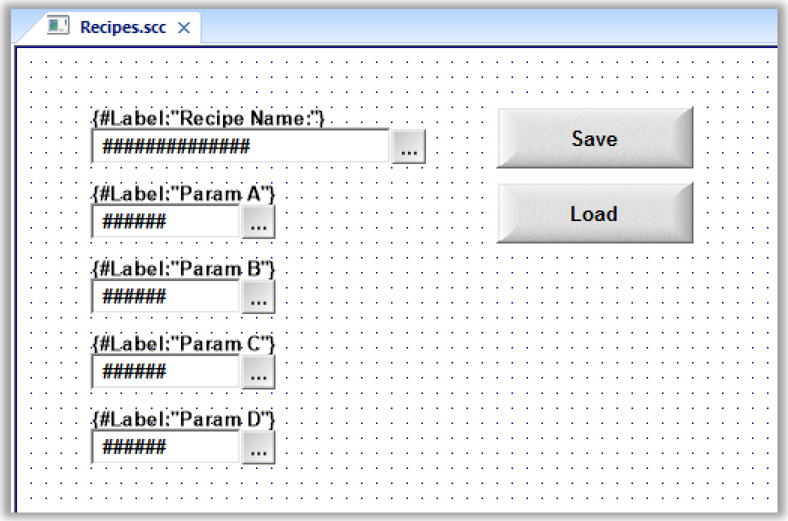
27. Set **Type** to **VBScript**, and enter the following in the body of the On Down event.

```
If $RdFileN("sLoadRecipeName","Recipes","*.xml",0) = 0 Then
  $sLoadRecipeName = Replace($sLoadRecipeName,"Recipes\","")
  $cRecipe.sRecipeName = Replace($sLoadRecipeName,".xml","")
  $Recipe("Load:Recipe1.rcp")
  $bWriteRecipe = $Toggle($bWriteRecipe)
End If
```

Note: You can clopy/paste this script from a softcopy of the training manual to your project to reduce the amount of typing and potential typing mistakes.

28. Close the Object Properties dialog.

29. The Recipes screen should like similar to the one shown below.



30. Save and close the RECIPES screen.

Verification and Test

1. If your runtime is on a remote machine, then use the the Remote Management tools on the Home tab of the Ribbon to download your project to the remote machine.

2. Start the runtime.

3. Open the RECIPES screen.

4. Fill in the text fields and save some recipes.

5. Load some saved recipes.

6. Use Windows Explorer to verify that recipe files are being stored in the folder called Recipes that you created in your project folder.

SECTION 2 - REPORTS

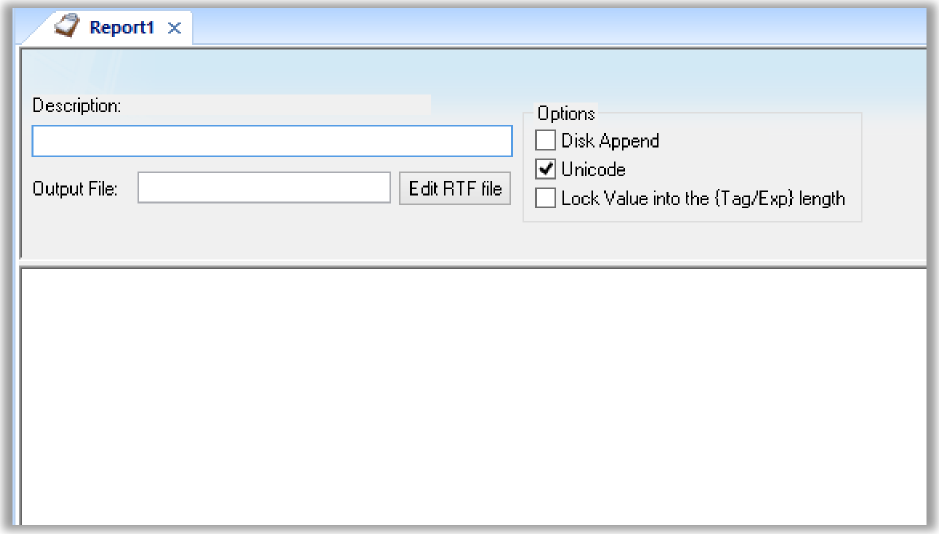
A Report worksheet is used to design a report that is dynamically generated during runtime (using the current values of the included tags) and then either sent to a printer or saved to a file.

Concept

To create a new Report worksheet, do one of the following:

- On the Insert tab of the ribbon, in the Task Worksheets group, click **Report**;
- Right-click the **Reports** folder in the Project Explorer, and then click **Insert** on the shortcut menu; or
- Click **New** on the Application menu, click the **File** tab, and then select **Report**.

To edit an existing Report worksheet, double-click it in the Project Explorer.



☐ Disk Append
☑ Unicode
☐ Lock Value into the {Tag/Exp} length

Note: When saving a Report worksheet, you can save it using any name you choose (you are not required to use a predefined file name like Report1). A configuration file using the extension `.REP` is created containing the report configuration.

The Report worksheet is divided into two areas:

- Header area (top section), which contains information for the whole group; and
- Body area (bottom section), where you define each tag in the group.

Use the Header parameters on this worksheet as follows:

- **Description** field: Type a description of the worksheet for documentation purposes.
- **Output File** field: Type a tag name for the output file (using the `{tag}` syntax) where data is stored when you are printing to a file. Where the tag value is part of the file name.

For example: `report{Day}.out`. Where the generated file might be `report1.out`, `report2.out`, `report3.out`, and so on, according to the tag day value.

- **Edit RTF** file button: Click to access the report as an RTF file, which you can edit for layout modification and so forth.
- **Disk Append** checkbox: Click (enable) the box to add (append) the new report to the end of an existing file or (disable) the box to replace the existing report in that file with the new report.
- **Unicode** checkbox: Click (enable) to save the report in Unicode format (two bytes per character) or (disable) to save the report in ASCII format (one byte per character).
- **Lock Value into the {Tag/Exp} length** checkbox: Click (enable) to automatically truncate the values of Tags/Expressions in the report to fit between the curly brackets, as they are positioned in the Body of the report (see below). This helps to preserve the layout of the report. If this option is left unchecked, then the full values of Tags/Expressions in the report will be displayed.

Use the Body portion of this worksheet for report formatting. You can configure a report using data in the system and indicating where to print the tag values. Each tag name will replace the `{tag_name}` tag name. For Real type tags, use the following syntax: `{tag_name n}`, where `n` is the number of decimal places you want printed.

Note: If you are using the standard report editor (text only: ASCII or Unicode), then the number of characters reserved for the tag value will be equal to the number of characters used to type the tag name (including the two "curly" brackets). For example, if you configure `{TagA}` in the report body, reserve six characters for the tag value in the report file. This behavior is not valid for reports in RTF format.

To execute a Report worksheet, use the **Report()** function anywhere an expression is allowed.

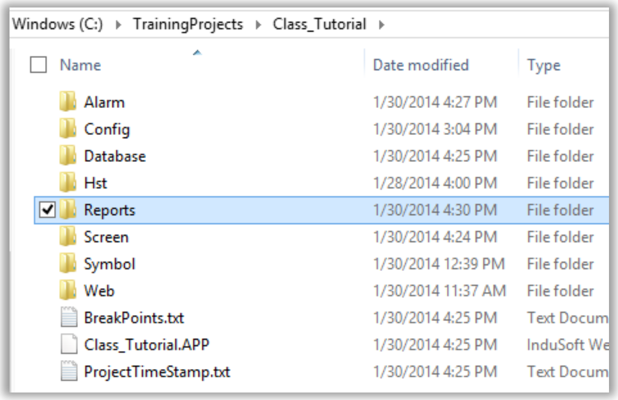
Pro-Tip: Because the Report task supports almost all text files, you can use this worksheet to generate more than just **.txt** files, such as **.readme**, **.html**, **.xml**, and **.csv** files.

LAB 19 - USING REPORTS

In this lab you will create a simple text report, RTF report, HTML report, and CSV report.

Preparing the environment

1. Open Windows Explorer, navigate to your project folder, and create a folder called `Reports`.



Windows (C:) ▸ TrainingProjects ▸ Class_Tutorial ▸

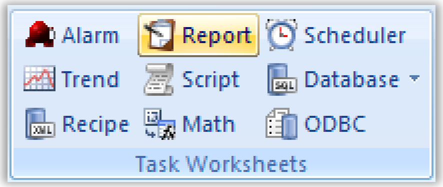
□ Name	Date modified	Type
Alarm	1/30/2014 4:27 PM	File folder
Config	1/30/2014 3:04 PM	File folder
Database	1/30/2014 4:25 PM	File folder
Hst	1/28/2014 4:00 PM	File folder
☑ Reports	1/30/2014 4:30 PM	File folder
Screen	1/30/2014 4:24 PM	File folder
Symbol	1/30/2014 12:39 PM	File folder
Web	1/30/2014 11:37 AM	File folder
BreakPoints.txt	1/30/2014 4:25 PM	Text Docum
Class_Tutorial.APP	1/30/2014 4:25 PM	InduSoft We
ProjectTimeStamp.txt	1/30/2014 4:25 PM	Text Docum

Pro Tip: Alternatively, you could use the built-in function called DirCreate() in the Startup Script to create the Reports folder when the project is started.

Create Report Task Worksheets

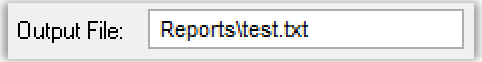
Text Report

1. On the Insert tab of the Ribbon, select **Report** under Task Worksheets.



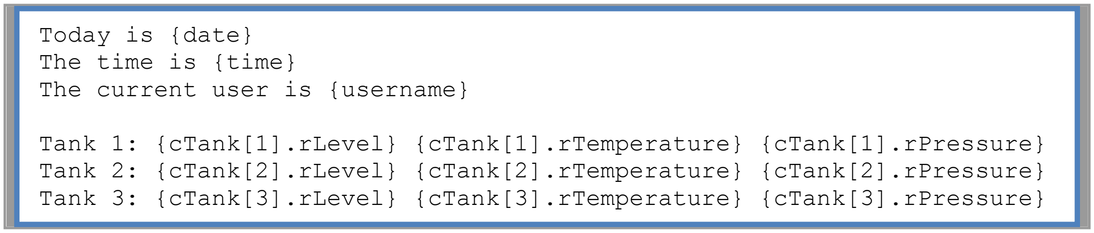
2. A blank report worksheet will be created and displayed in the Workspace.

3. In the **Output File** field, enter `Reports\test.txt`.



Output File: Reports\test.txt

4. Enter the following into the body of the worksheet.



```
Today is {date}
The time is {time}
The current user is {username}

Tank 1: {cTank[1].rLevel} {cTank[1].rTemperature} {cTank[1].rPressure}
Tank 2: {cTank[2].rLevel} {cTank[2].rTemperature} {cTank[2].rPressure}
Tank 3: {cTank[3].rLevel} {cTank[3].rTemperature} {cTank[3].rPressure}
```

5. Save the report worksheet and call it `TextReport`.

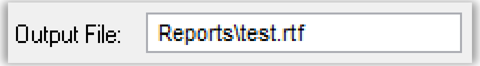
6. Close the TextReport worksheet.

RTF Report

7. On the Insert tab of the Ribbon, select **Report** under Task Worksheets.

8. A blank report worksheet will be created and displayed in the Workspace.

9. In the **Output File** field, enter `Reports\test.rtf`.

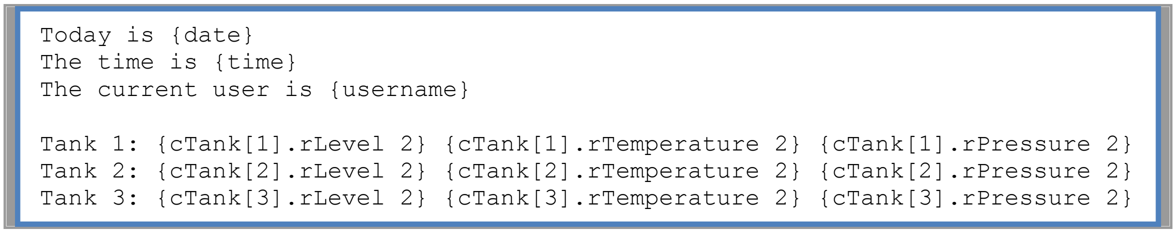


Output File: Reports\test.rtf

10. Save the report worksheet and call it `RTFReport`.

11. Click the button called **Edit RTF File**. The ReportWriter window is opened.

12. Enter the following into the ReportWriter window.

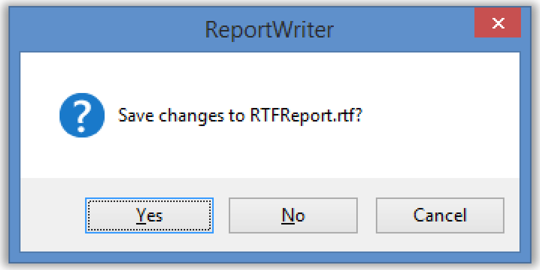


```
Today is {date}
The time is {time}
The current user is {username}

Tank 1: {cTank[1].rLevel 2} {cTank[1].rTemperature 2} {cTank[1].rPressure 2}
Tank 2: {cTank[2].rLevel 2} {cTank[2].rTemperature 2} {cTank[2].rPressure 2}
Tank 3: {cTank[3].rLevel 2} {cTank[3].rTemperature 2} {cTank[3].rPressure 2}
```

13. Use the features of the ReportWriter to choose different fonts and colors for your RTF report.

14. Close the ReportWriter window. Click **Yes** when you see the following window.



Save changes to RTFReport.rtf?

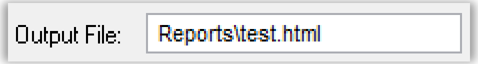
15. Save and close the RTFReport worksheet.

HTML Report

16. On the Insert tab of the Ribbon, select **Report** under Task Worksheets.

17. A blank report worksheet will be created and displayed in the Workspace.

18. In the **Output File** field, enter `Reports\test.html`.



Output File: Reports\test.html

19. Enter the following into the body of the worksheet.

```
<html>
<body>
Today is <b>{date}</b><br />
The time is <b>{time}</b><br />
The current user is <b>{username}</b><br />
<hr>
<table border=1>
 <tr>
  <td></td>
  <td align="center">Level</td>
  <td align="center">Temperature</td>
  <td align="center">Pressure</td>
 </tr>
 <tr>
  <td>Tank 1:</td>
  <td align="right">{cTank[1].rLevel 2}</td>
  <td align="right">{cTank[1].rTemperature 2}</td>
  <td align="right">{cTank[1].rPressure 2}</td>
 </tr>
 <tr>
  <td>Tank 2:</td>
  <td align="right">{cTank[2].rLevel 2}</td>
  <td align="right">{cTank[2].rTemperature 2}</td>
  <td align="right">{cTank[2].rPressure 2}</td>
 </tr>
 <tr>
  <td>Tank 3:</td>
  <td align="right">{cTank[3].rLevel 2}</td>
  <td align="right">{cTank[3].rTemperature 2}</td>
  <td align="right">{cTank[3].rPressure 2}</td>
 </tr>
</table>
</body>
</html>
```

20. Save the report worksheet and call it `HTMLReport`.

21. Close the HTMLReport worksheet.

CSV Report

22. On the Insert tab of the Ribbon, select **Report** under Task Worksheets.

23. A blank report worksheet will be created and displayed in the Workspace.

24. In the **Output File** field enter `Reports\{Str(Year)+ Format("%02d",Month)+Format("%02d",Day)}.csv`.

25. Clear the option called **Unicode**.

26. Enter the following into the body of the worksheet.

```
"Date","Time","Tank 1 Level","Tank 2 Level","Tank 3 Level"
```

27. Save the report worksheet and call it `CSVReportHeader`.

28. Close the CSVReportHeader worksheet.

29. On the Insert tab of the Ribbon, select **Report** under Task Worksheets.

30. A blank report worksheet will be created and displayed in the Workspace.

31. In the **Output File** field enter `Reports\{Str(Year)+Format("%02d",Month)+Format("%02d",Day)}.csv`

32. Clear the option called **Unicode** and select the option called **Disk Append**.

33. Enter the following into the body of the worksheet.

```
"{Date}","{Time}",{cTank[1].rLevel},{cTank[2].rLevel},{cTank[3].rLevel
```

34. Save the report worksheet and call it `CSVReportBody`.

35. Close the CSVReportBody worksheet.

Create Reports Screen

1. Open the Template screen.

2. Click the Application button and click **Save As** on the Application menu. The Save As dialog is opened.

3. In the **File name** field, enter `Reports`.

4. Click **Save**. The Save As dialog is closed.

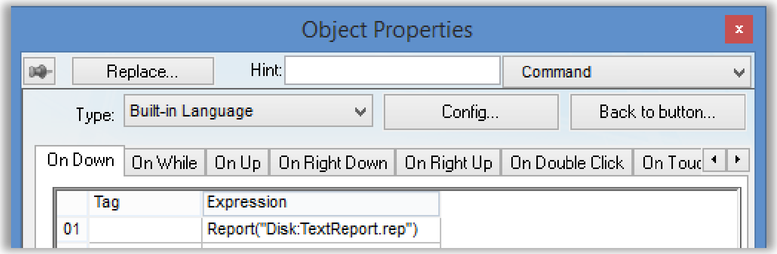
5. On the Graphics tab of the Ribbon in the Active Objects group, select **Button** and add a button to the screen.

6. Open the Object Properties of the button.

7. In **Caption**, type `CREATE TEXT REPORT`.

8. Click **Command**.

9. Set **Type** to **Built-In Language**, and then in the **Expression** column, enter `Report("Disk:TextReport.rep")` as shown below.



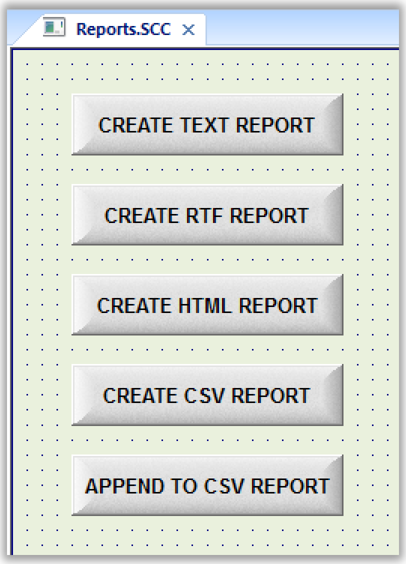
Command

Type: Built-in Language

	Tag	Expression
01		Report("Disk:TextReport.rep")

10. On the Graphics tab of the Ribbon in the Active Objects group, select **Button** and add a button to the screen.

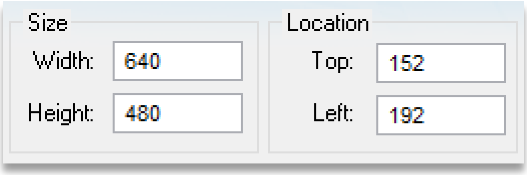
11. Open the Object Properties of the button.
12. In **Caption**, type `CREATE RTF REPORT`.
13. Click **Command**.
14. Set **Type** to **Built-In Language**, and then in the **Expression** column, enter `Report("Disk:RTFReport.rep")`.
15. On the Graphics tab of the Ribbon in the Active Objects group, select **Button** and add a button to the screen.
16. Open the Object Properties of the button.
17. In **Caption**, type `CREATE HTML REPORT`.
18. Click **Command**.
19. Set **Type** to **Built-In Language**, and then in the **Expression** column, enter `Report("Disk:HTMLReport.rep")`.
20. On the Graphics tab of the Ribbon in the Active Objects group, select **Button** and add a button to the screen.
21. Open the Object Properties of the button.
22. In **Caption**, type `CREATE CSV REPORT`.
23. Click **Command**.
24. Set **Type** to **Built-In Language**, and then in the **Expression** column, enter `Report("Disk:CSVReportHeader.rep")`.
25. On the Graphics tab of the Ribbon in the Active Objects group, select **Button** and add a button to the screen.
26. Open the Object Properties of the button.
27. In **Caption**, type `APPEND TO CSV REPORT`.
28. Click **Command**.
29. Set **Type** to **Built-In Language**, and then in the **Expression** column enter `Report("Disk:CSVReportBody.rep")`.
30. When you are done the Reports screen should like similar to the one shown below.



31. Save and close the REPORTS screen.

Create HTML Reports Viewer

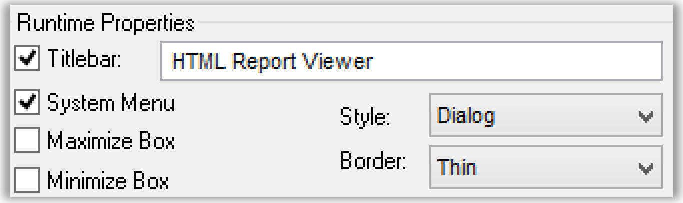
1. Insert a new screen with the following size and location,



Size
Width: 640
Height: 480

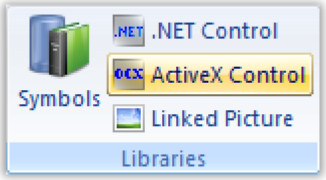
Location
Top: 152
Left: 192

and the following Runtime properties,



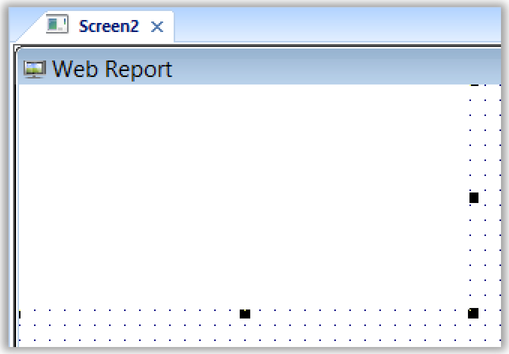
Runtime Properties
☑ Titlebar: HTML Report Viewer
☑ System Menu
☐ Maximize Box
☐ Minimize Box
Style: Dialog
Border: Thin

2. On the Graphics tab of the Ribbon in the Libraries group, select **ActiveX Control**.

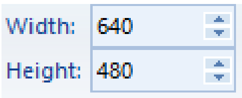


3. The Insert ActiveX Control dialog is opened with the list of available ActiveX controls.

4. Select the control called **Microsoft Web Browser** and click **OK**. The control is added to your screen.



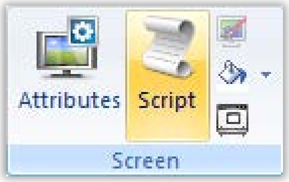
5. Select the Format tab on the ribbon, and then set the size of the control to have a width of 640 and a height of 480 as shown below.



Width: 640

Height: 480

6. Select the Graphics tab on the ribbon, and then select **Script** on the Screen group.



7. The Screen script is opened.

8. Change the subroutine called Screen_OnOpen as shown below.

```
Sub Screen_OnOpen()
    MicrosoftWebBrowser1.Navigate2 $GetAppPath() & "Reports\test.html"
End Sub
```

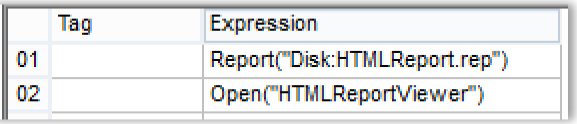
9. Save and close the Screen Script.

10. Save and close the screen and name it `HTMLReportViewer`.

11. Open the screen called Reports.

12. Open the Object Properties for the button called CREATE HTML REPORT.

13. Append the expression `Open("HTMLReportViewer")` to the Command as shown below.



	Tag	Expression
01		Report("Disk:HTMLReport.rep")
02		Open("HTMLReportViewer")

14. Save and close the Reports screen.

15. Run the project and click the button called CREATE HTML REPORT.

Verification and Test

1. If your runtime is on a remote machine, then use the the Remote Management tools on the Home tab of the Ribbon to download your project to the remote machine.
2. Start the runtime.
3. Open the REPORTS screen.
4. Click the button called CREATE TEXT REPORT.
5. Click the button called CREATE RTF REPORT.
6. Click the button called CREATE HTML REPORT.
7. Click the button called CREATE CSV REPORT.
8. Click the button called APPEND TO CSV REPORT a few times.
9. Use Windows Explorer to verify that the report files are being stored in the folder called Reports.
10. Open each report to confirm that they look as expected.

.

Module 10 - Project Security

SECTION 1 - PROJECT SECURITY

This section explains how to configure the security system that will be used for training.

You will enable the Security Settings and create two groups and users. One of the groups and users will have full access to the application (development and runtime) while the other group and user will have their actions in the project (development and runtime) limited.

Concept

BLUE Open Studio includes a security system that controls how users and user groups can access a project during development and execution time.

Almost all items in a project (screens, object animations, worksheets, and tasks) can have a security access level attributed. This access level determines which groups may edit the item during development and/or access the item during execution.

There are 255 possible access levels permitting a great amount of flexibility. Each user group is configured with security level ranges independently for development and execution, and group levels may overlap.

For a user that is able to edit and/or use an item, the item access level must be within the specified security range of the group.

Overview

Security Modes

In addition to managing users and groups locally, entirely within a single project, you can also get pre-defined users and groups from other BLUE Open Studio projects or from an LDAP-compliant domain server.

BLUE Open Studio supports four security modes:

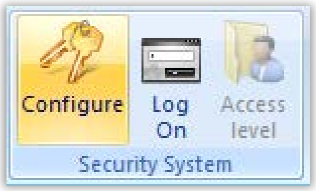
- **Local Only:** This is the standard mode for most projects: users and groups are created and apply only to the project for which they're created.
- **Distributed – Server:** This is similar to Local Only, except that the project's security system configuration is also made available to other BLUE Open Studio projects (that are set to Distributed – Client) on the same network. Furthermore, if the project loses its security system configuration for some reason, then it can reimport the configuration from one of its client projects.
- **Distributed – Client:** When this mode is selected, the project gets its entire security system configuration from another BLUE Open Studio project (that is set to Distributed – Server) on the same network. The project caches this configuration and can continue to run even if it loses communication with the server project.
- **Domain (LDAP):** The Lightweight Directory Access Protocol (LDAP) is a recognized standard for managing users and groups across many different applications on a network. When this mode is selected, the project gets its users and groups from an LDAP-compliant domain server, such as Microsoft Active Directory for Windows or OpenLDAP for Linux.

However, only the user names, passwords, and group memberships are taken from the domain; specific rights must still be configured within the project.

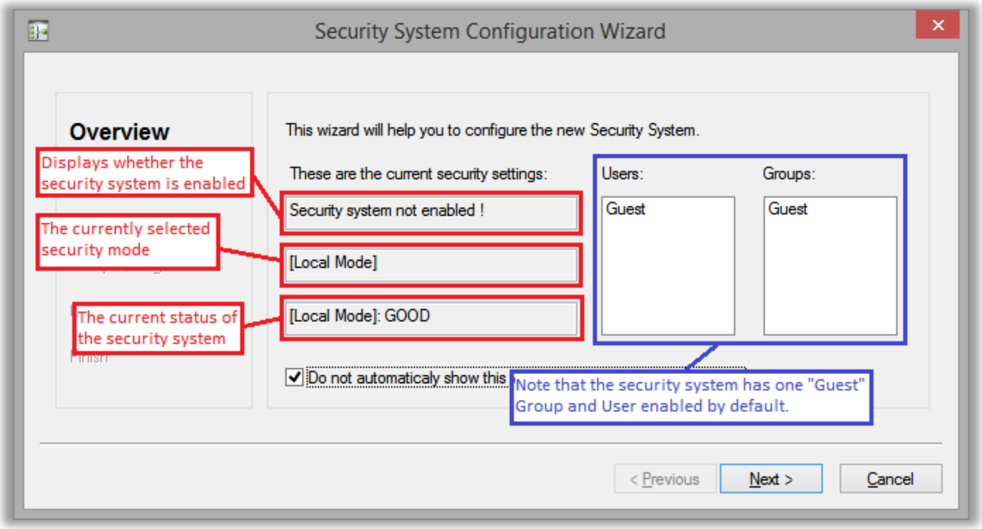
LAB 20 - CONFIGURE THE PROJECT SECURITY SYSTEM

Enable the Security Settings

1. In the Project tab of the Ribbon under the Security Settings group, select **Configure**.



2. The Security Settings Configuration Wizard is opened. This wizard will walk you through configuring the security step-by-step.



This wizard will help you to configure the new Security System.

These are the current security settings:

Security system not enabled !

[Local Mode]

[Local Mode]: GOOD

Guest

Guest

☑ Do not automaticaly show this

Displays whether the security system is enabled

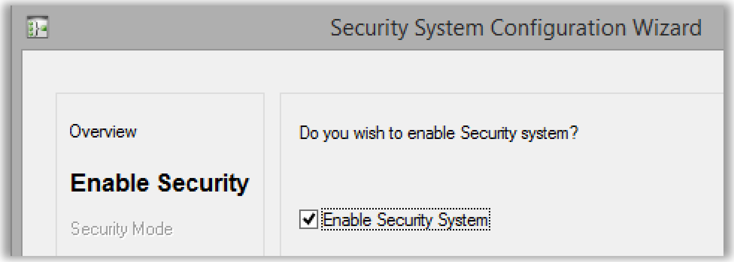
The currently selected security mode

The current status of the security system

Note that the security system has one "Guest" Group and User enabled by default.

3. Click on **Next**.

4. Select option **Enable the Security Settings**.

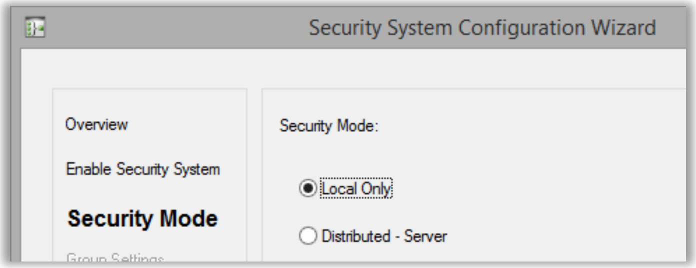


Do you wish to enable Security system?

☑ Enable Security System

5. Click on **Next**.

6. Select option **Local Only**.



Security System Configuration Wizard

Security Mode:

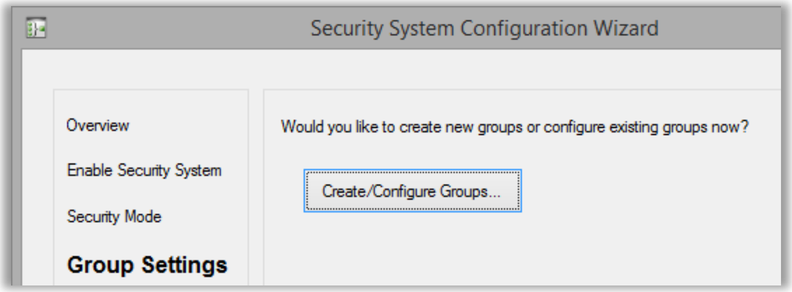
● Local Only

○ Distributed - Server

7. Click **Next.**

Configuration of Groups

1. Click the **Create/Configure Groups** button.

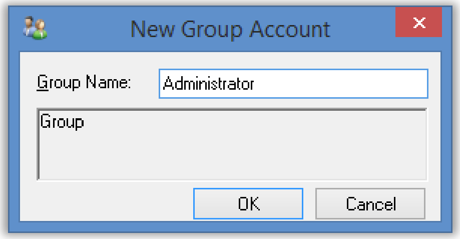


Security System Configuration Wizard

Would you like to create new groups or configure existing groups now?

2. Click on **New** to open the New Group Account dialog.

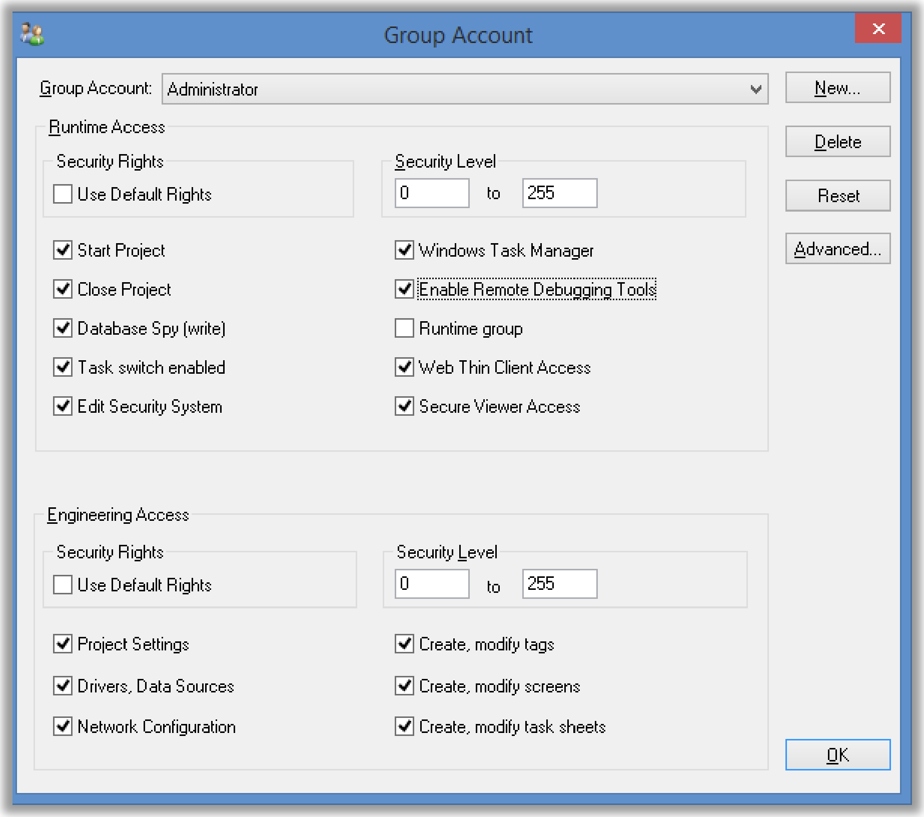
3. Enter `Administrator` into the Group Name field and select **OK**.



New Group Account

Group Name: Administrator

4. Configure the Administrator group as shown below.



Group Account: Administrator

Runtime Access

Security Rights

☐ Use Default Rights

Security Level: 0 to 255

☑ Start Project
☑ Close Project
☑ Database Spy (write)
☑ Task switch enabled
☑ Edit Security System
☑ Windows Task Manager
☑ Enable Remote Debugging Tools
☐ Runtime group
☑ Web Thin Client Access
☑ Secure Viewer Access

Engineering Access

Security Rights

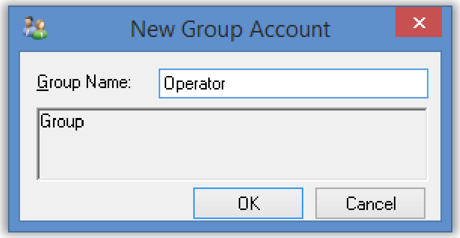
☐ Use Default Rights

Security Level: 0 to 255

☑ Project Settings
☑ Drivers, Data Sources
☑ Network Configuration
☑ Create, modify tags
☑ Create, modify screens
☑ Create, modify task sheets

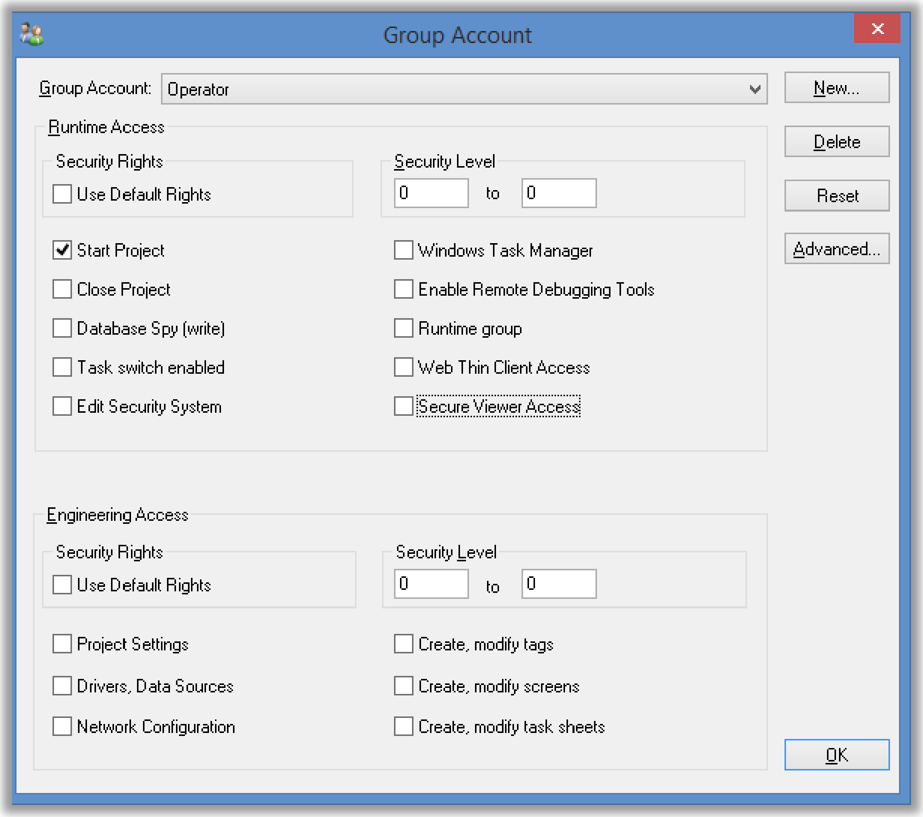
5. Click on **New**.

6. Enter `Operator` into the Group Name field and select **OK**.



Group Name: Operator

7. Configure the Operator group as shown below.



Group Account: Operator

Runtime Access

Security Rights
□ Use Default Rights

Security Level
0 to 0

☑ Start Project
□ Close Project
□ Database Spy (write)
□ Task switch enabled
□ Edit Security System
□ Windows Task Manager
□ Enable Remote Debugging Tools
□ Runtime group
□ Web Thin Client Access
□ Secure Viewer Access

Engineering Access

Security Rights
□ Use Default Rights

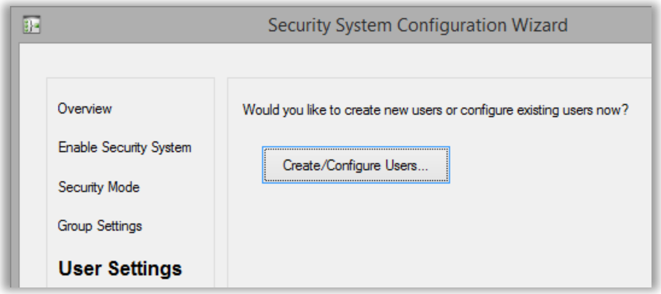
Security Level
0 to 0

□ Project Settings
□ Drivers, Data Sources
□ Network Configuration
□ Create, modify tags
□ Create, modify screens
□ Create, modify task sheets

8. Click **OK**.
9. Click on **Next**.

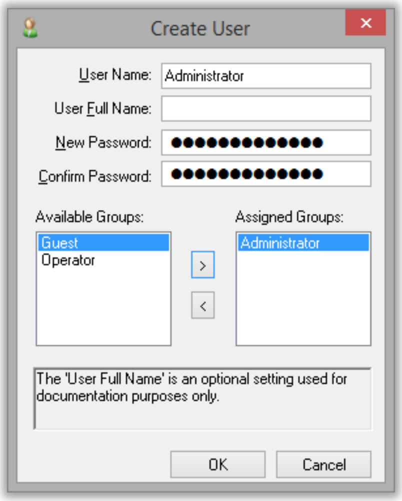
User Configuration

1. Select option **Create/Configure Users**.



Would you like to create new users or configure existing users now?

2. Click on **New**.
3. In **User Name**, add the `Administrator` name.
4. In **New Password**, type `Administrator`.
5. In **Confirm Password**, type `Administrator`.
6. Select the **Administrator** group.



Create User

User Name: Administrator

User Full Name:

New Password: ●●●●●●●●●●●●●

Confirm Password: ●●●●●●●●●●●●●

Available Groups: Guest, Operator

Assigned Groups: Administrator

The 'User Full Name' is an optional setting used for documentation purposes only.

7. Click **OK**.

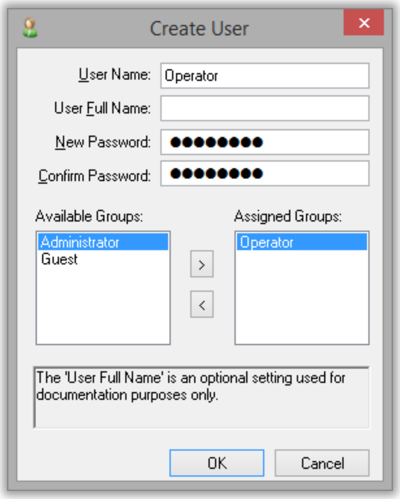
8. Click on **New**.

9. In **User Name**, add the `Operator` name.

10. In **New Password**, type `Operator`.

11. In **Confirm Password**, type `Operator`.

12. Select the **Operator** group.



Create User

User Name: Operator

User Full Name:

New Password: ●●●●●●●●

Confirm Password: ●●●●●●●●

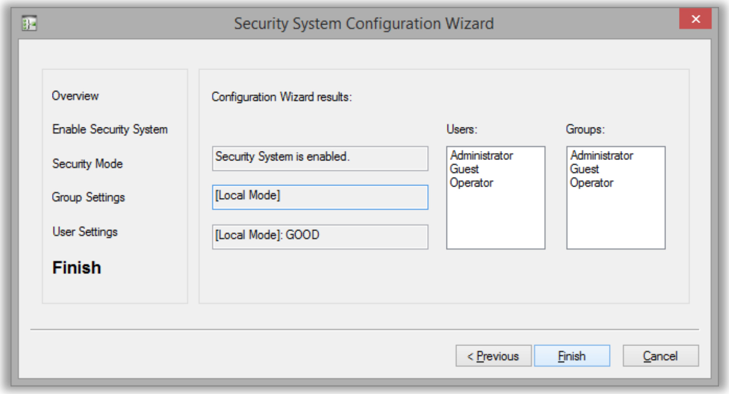
Available Groups: Administrator, Guest

Assigned Groups: Operator

The 'User Full Name' is an optional setting used for documentation purposes only.

13. Click **OK**, then **OK**

14. Click **Next** to display the status and changes made to the security system.



Configuration Wizard results:

Security System is enabled.

[Local Mode]

[Local Mode]: GOOD

Users: Administrator, Guest, Operator

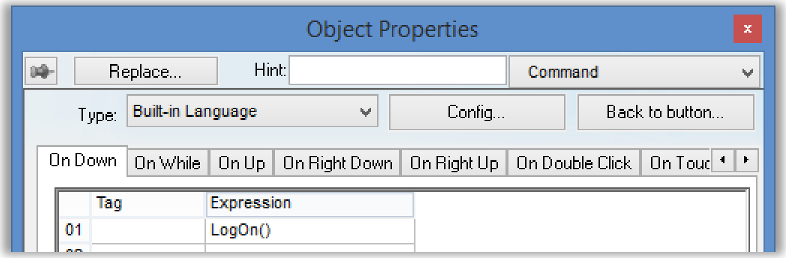
Groups: Administrator, Guest, Operator

15. Click **Finish** to exit the wizard.

Configuring your project to use security during runtime

Creating a Log On button

1. Open the HEADER screen.
2. Add a button to the screen with the caption `LOG ON`.
3. Double-click on the button to open the Object Properties window and click on **Command**.
4. The Command animation dialog is opened.
5. In **Type**, select **Built-In Language**.
6. In the first row, click in the **Expression** column and type `LogOn()`.



Hint:

Command

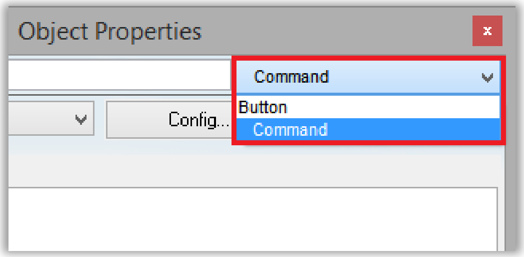
Type: Built-in Language

	Tag	Expression
01		LogOn()

7. Save and close the HEADER screen.

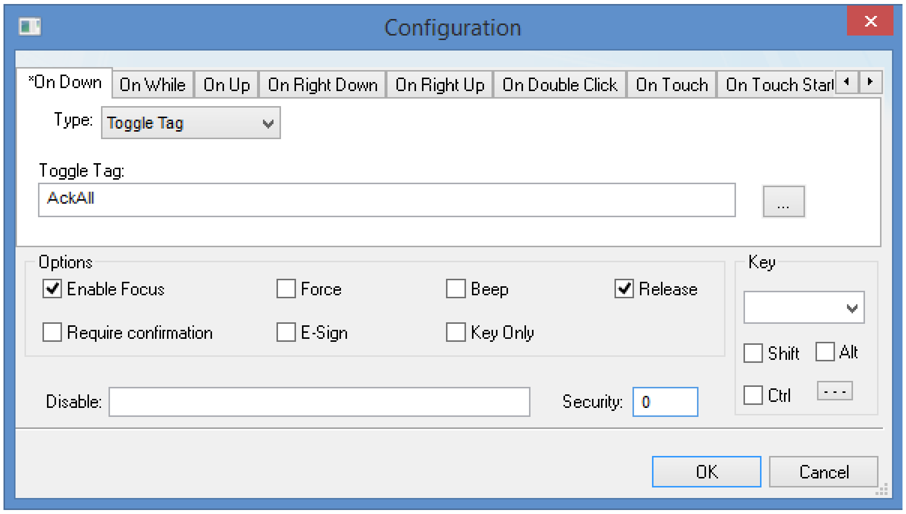
Limiting access to screen objects

1. Open the ALARMS screen.
2. Double-click on the **Ack All** button to open the Object Properties window
3. The Object Properties dialog should open to the **Command** animation. If it does not, use the object navigator dropdown to switch.



Command

4. Click the **Config** button to access the Configuration dialog.



Type: Toggle Tag

Toggle Tag: AckAll

☑ Enable Focus □ Force □ Beep ☑ Release

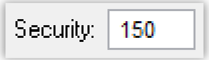
□ Require confirmation □ E-Sign □ Key Only

Disable:

Security: 0

□ Shift □ Alt □ Ctrl

5. Under the Security field (bottom right), enter **150**.



Security: 150

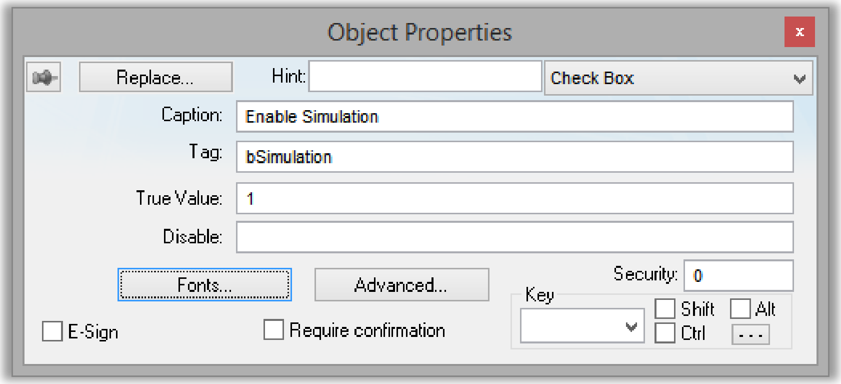
6. Click **OK**, then close the Object Properties dialog.

7. Save and close the ALARMS screen.

Configuring electronic signatures on screen objects

1. Open the HEADER screen.

2. Double-click the **Enable Simulation** checkbox object to open the Object Properties dialog.



Hint:

Check Box

Caption: Enable Simulation

Tag: bSimulation

True Value: 1

Disable:

Security: 0

□ Shift □ Alt
□ Ctrl

□ E-Sign □ Require confirmation

3. Check the **E-Sign** checkbox (bottom left).

☑ E-Sign

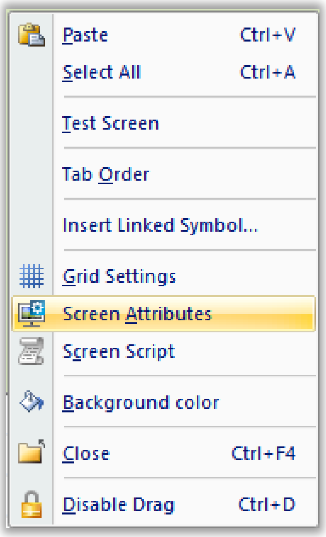
4. Click **OK,** then close the Object Properties dialog.

5. Save and close the HEADER screen.

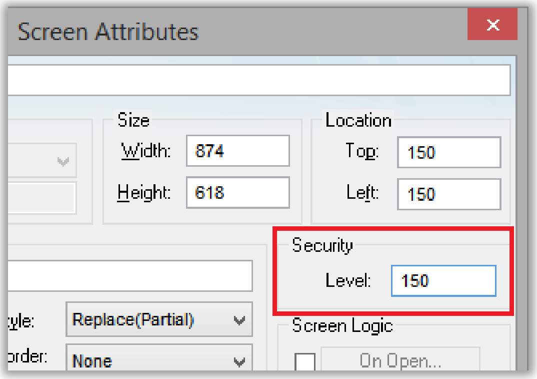
Limiting access to screens

1. Open the EVENTS screen.

2. Right-click on the screen and select **Screen Attributes**.



3. On the Screen Attributes dialog, under the Security Level field, enter `150`.



Width: 874

Height: 618

Top: 150

Left: 150

Level: 150

yle: Replace(Partial)

order: None

□ On Open...

4. Close the EVENTS screen.

Verification and Test

1. If your runtime is on a remote machine, then use the Remote Management tools on the Home tab of the Ribbon to download your project to the remote machine.

2. Start the runtime.

3. Log on as Operator.

4. Click the button called EVENTS. Observe that you get denied.

5. Open the ALARMS screen. Observe that the button called ACK ALL is disabled.

6. Click the Enable Simulation checkbox. Observe that you get prompted for your credentials.

7. Click the button called LOG ON and log on as Administrator.

8. Repeat steps 3 and 4 above and verify that they work as expected.

SECTION 2 - PROTECTING YOUR INTELLECTUAL PROPERTY

This section explains how to configure passwords on the screens and worksheets that you create.

Concept

Screens, symbols, and worksheets in the Project Explorer can be password-protected. You can assign individual passwords to each file, or you can assign a single password to all files in the project.

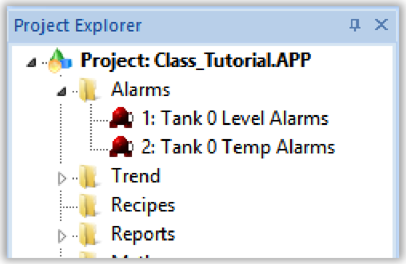
Almost all project files are encrypted to prevent unauthorized analysis by third-party tools. However, you can take the extra step of password-protecting your files to prevent unauthorized changes or re-use by other BLUE Open Studio project developers.

LAB 21 - ENCRYPTING YOUR PROJECT FILES

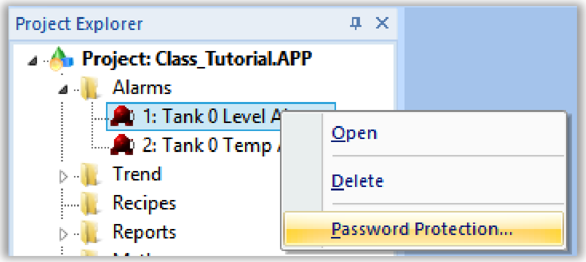
In this lab you will configure a password that is used to encrypt individual project files or all the project files.

Encrypting a single file

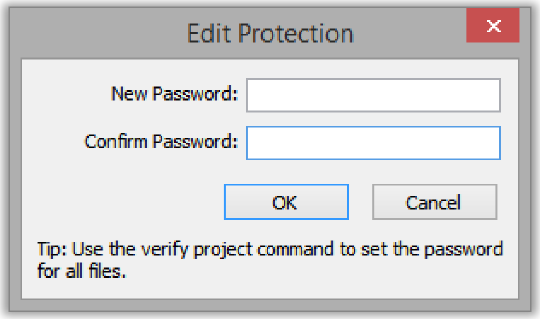
1. In the Project Explorer under the Tasks tab, expand the Alarms folder to display the two alarm worksheets.



2. Right click on Alarm worksheet **1: Tank 0 Level Alarms** and select Password Protection.



3. Doing so will open the Edit Protection dialog.



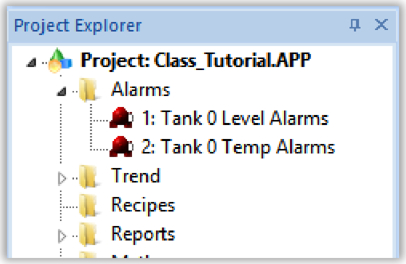
Tip: Use the verify project command to set the password for all files.

4. Under **New Password** and **Confirm Password**, type `Password`.

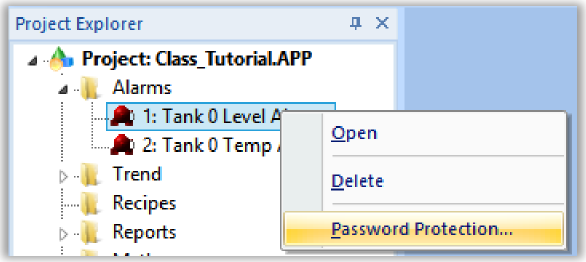
5. Click **OK**.

Removing Password Protection

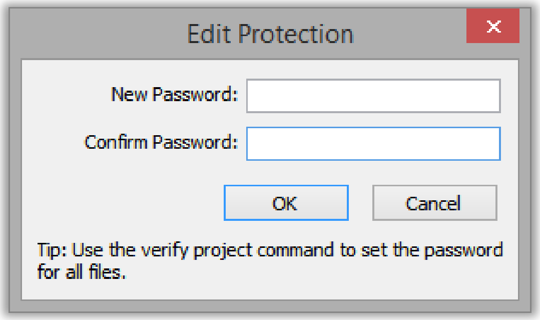
1. In the Project Explorer under the Tasks tab, expand the Alarms folder to display the two alarm worksheets.



6. Right click on Alarm worksheet **1: Tank 0 Level Alarms** and select Password Protection.



7. Doing so will open the Edit Protection dialog.



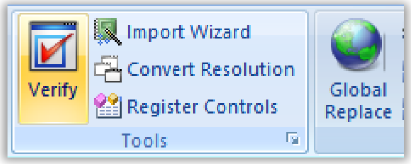
Tip: Use the verify project command to set the password for all files.

8. Under **New Password** and **Confirm Password**, leave the fields blank.

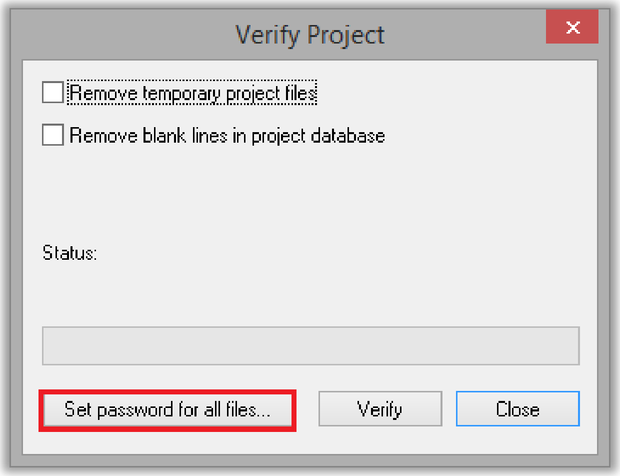
9. Click **OK**.

Encrypting all your project files

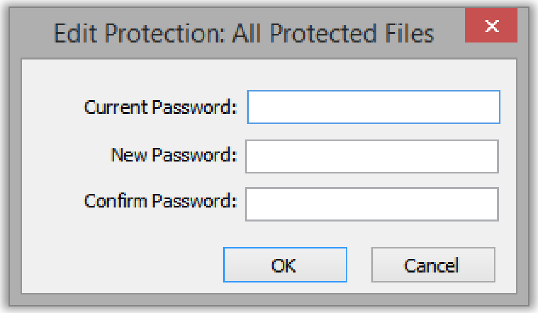
1. Open the Home tab of the Ribbon. Under the Tools group, select Verify to open the Verify Project tool.



2. The Verify Project tool is opened. Select the **Set Password for all files...** button.


Verify Project
□ Remove temporary project files
□ Remove blank lines in project database
Status:
Set password for all files...
Verify
Close

3. The Edit Protection: All Protected Files dialog is opened.



4. Under **New Password** and **Confirm Password**, type `Password`.

5. Click **OK**.

Note: This process will only protect existing project files. Any new files created will not contain this protection

Verification and Test

1. Try to open a screen or worksheet and verify that it prompts you for your password.

2. Enter your password and confirm that you can open the screen or worksheet.

Module 11 - Remote Clients

SECTION 1 - MOBILE ACCESS

You can use Mobile Access (also sometimes called Studio Mobile Access or SMA) to deploy an HTML5-enhanced web interface that presents alarms, trends, process values, and even project screens in a unified, easy-to-use "dashboard".

Concept

The web interface is specifically designed for smartphones and tablets, including iOS and Android devices, but it can be accessed from almost any computer using a modern web browser.

It is important to remember that although Mobile Access is part of the same project runtime and may be hosted on the same server that hosts the screens published for Mobile Access. The Mobile Access web interface is based on HTML5, so it can run in most web browsers on most computers and devices.

Overview

To use Mobile Access, you must have the Mobile Access Runtime software installed on your project runtime server. Either you selected the Mobile Access Runtime feature when you installed the full BLUE Open Studio software, or you ran the separate, included Mobile Access Runtime installer (MobileAccessSetup.exe) afterward.

Also, your software license must include enough SMA Clients to accommodate all of the users that you expect to access the Mobile Access web interface at the same time.

Note: The Mobile Access Runtime software cannot be installed on Windows Embedded, and Mobile Access cannot be configured in projects that are set to run on Windows Embedded target systems.

Configuring the Mobile Access web interface

Use the Mobile Access Configuration worksheet to configure the Mobile Access web interface for your BLUE Open Studio project. To open the Mobile Access Configuration worksheet, do one of the following:

- On the Project tab of the ribbon, in the Web group, click **Mobile Access**.
- On the Graphics tab of the Project Explorer, in the Thin Clients folder right-click **Mobile Access**, and click **Open** on the shortcut menu.

The Mobile Access configuration consists of one or more "areas", which are typically organized by location, system, or machine. Each area has an Alarm control, a Process Values control, a Trend control, and a Screens control.

- **Alarm:** The Alarm control is similar to the Alarm/Event Control screen object. It displays active alarms and allows the user to acknowledge them.
- **Process Values:** The Process Values control is similar to the Symbols library, in that it uses various pre-made widgets (i.e., gauges and switches) to graphically represent project tag values. It can also allow the user to change the values during run time, depending on how you configure the widgets.

- **Trend:** The Trend control is similar to the Trend Control screen object. It graphs the changes in process values over a specified duration, and it also saves a history of those changes.
- **Screens:** The Screens control presents project screens that you have selected to include in the web interface.

These controls appear in the web interface as green tiles, and when you click/tap one of these tiles, it opens a new page for that control.

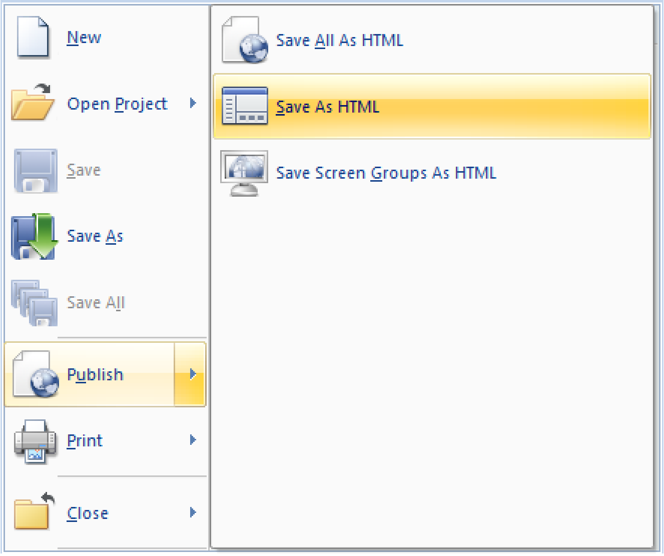
LAB 22 - CONFIGURING AND USING MOBILE ACCESS

In this lab you will perform all the steps necessary to configure Mobile Access.

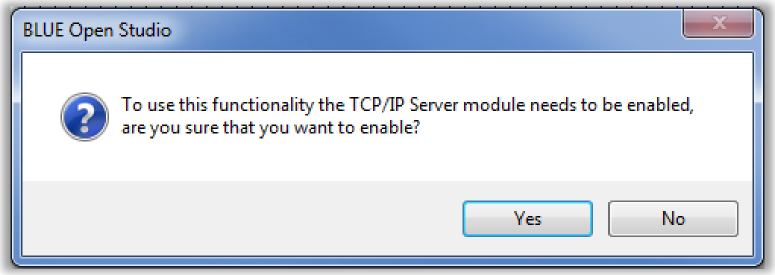
Note: Mobile Access requires that Microsoft Internet Information Services (IIS) be installed on the runtime machine, but there is no specific IIS configuration that must be performed to use Mobile Access.

Publish Main screen as HTML

1. Open the screen called Main.
2. Click the Application button, click **Publish** on the Application menu, and then click **Save As HTML**.

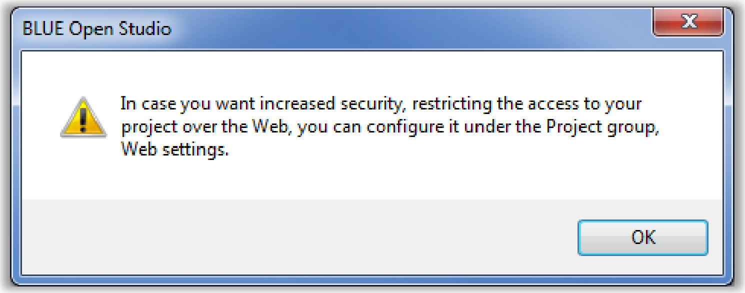


3. The following message may be displayed:


BLUE Open Studio

To use this functionality the TCP/IP Server module needs to be enabled, are you sure that you want to enable?

4. Click on **Yes** to enable the TCP/IP Server task.
5. If IP Security is not enabled, the following message will be displayed:



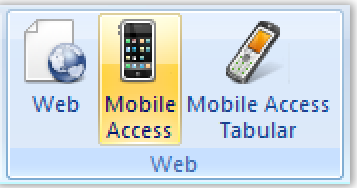
BLUE Open Studio

In case you want increased security, restricting the access to your project over the Web, you can configure it under the Project group, Web settings.

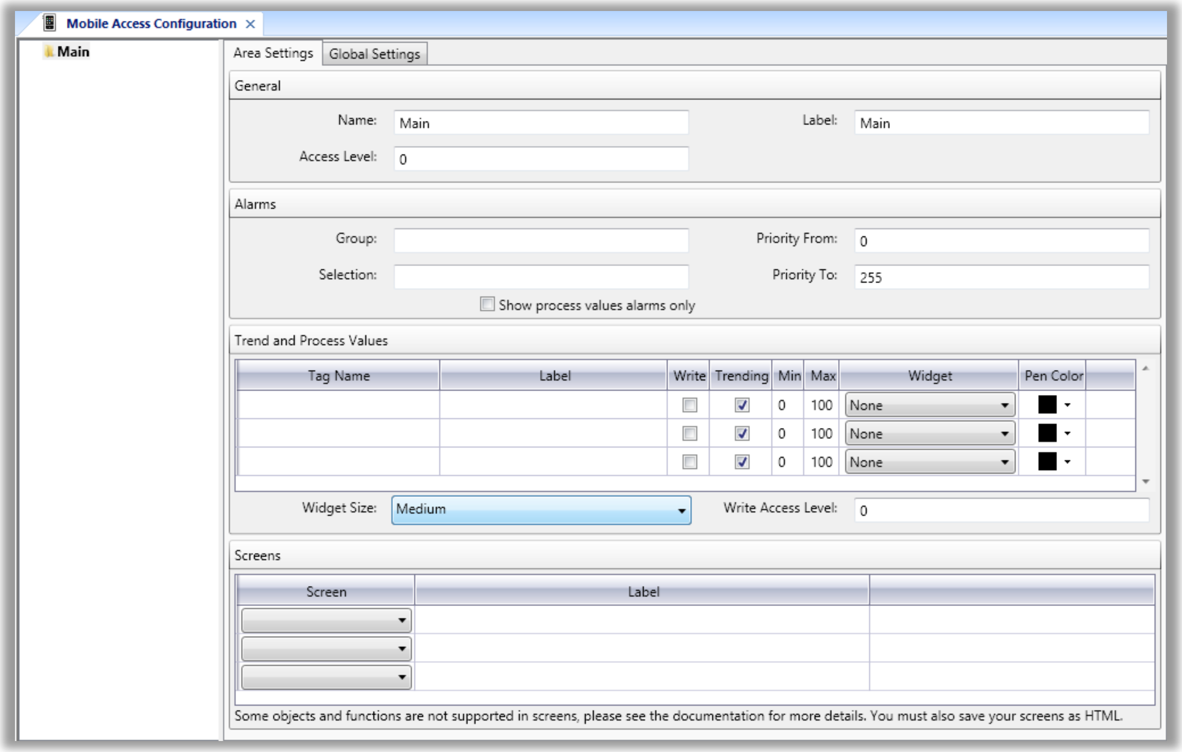
6. Click on **OK**

Configure Mobile Access

1. On the Project tab of the ribbon, in the Web group, click **Mobile Access**.



2. The Mobile Access Configuration is opened in the Workspace.



Mobile Access Configuration ×

Main

Area Settings | Global Settings

General

Name:	Main	Label:	Main
Access Level:	0		

Alarms

Group:		Priority From:	0
Selection:		Priority To:	255

□ Show process values alarms only

Trend and Process Values

Tag Name	Label	Write	Trending	Min	Max	Widget	Pen Color
		□	☑	0	100	None	■
		□	☑	0	100	None	■
		□	☑	0	100	None	■

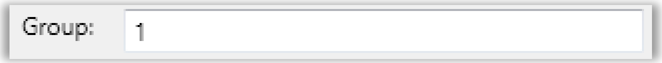
Widget Size: Medium   Write Access Level: 0

Screens

Screen	Label	

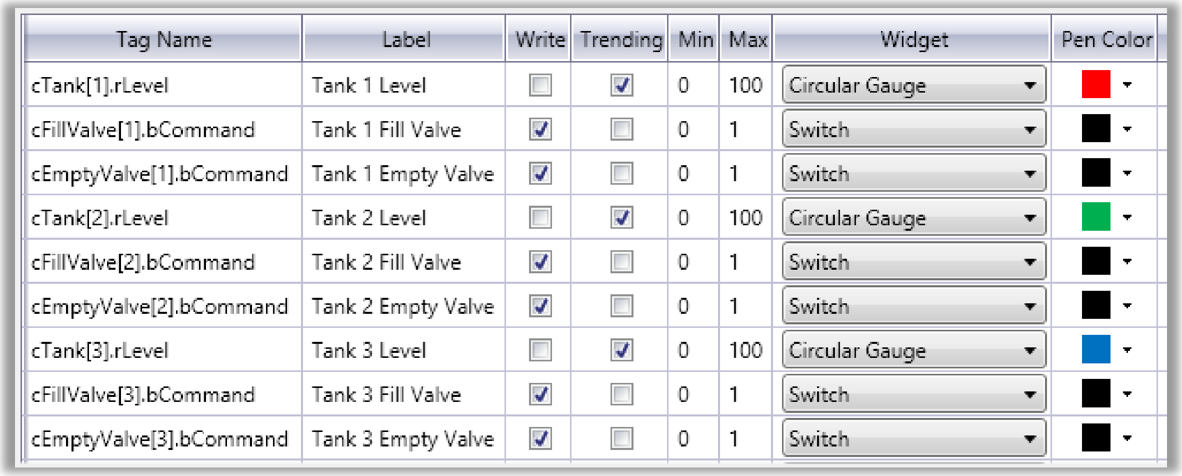
Some objects and functions are not supported in screens, please see the documentation for more details. You must also save your screens as HTML.

3. In the **Alarms** group, enter the value `1` in the **Group** field.



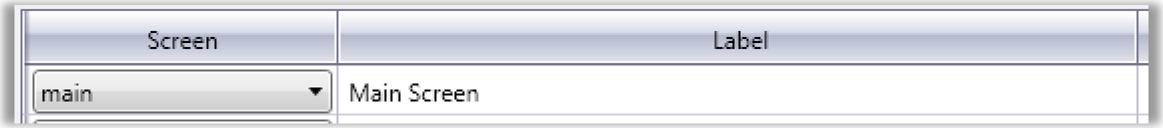
Group: 1

4. In the **Trend and Process Values** group, fill in the worksheet as shown below.



Tag Name	Label	Write	Trending	Min	Max	Widget	Pen Color
cTank[1].rLevel	Tank 1 Level	☐	☑	0	100	Circular Gauge	🟥
cFillValve[1].bCommand	Tank 1 Fill Valve	☑	☐	0	1	Switch	⬛
cEmptyValve[1].bCommand	Tank 1 Empty Valve	☑	☐	0	1	Switch	⬛
cTank[2].rLevel	Tank 2 Level	☐	☑	0	100	Circular Gauge	🟩
cFillValve[2].bCommand	Tank 2 Fill Valve	☑	☐	0	1	Switch	⬛
cEmptyValve[2].bCommand	Tank 2 Empty Valve	☑	☐	0	1	Switch	⬛
cTank[3].rLevel	Tank 3 Level	☐	☑	0	100	Circular Gauge	🟦
cFillValve[3].bCommand	Tank 3 Fill Valve	☑	☐	0	1	Switch	⬛
cEmptyValve[3].bCommand	Tank 3 Empty Valve	☑	☐	0	1	Switch	⬛

5. In the **Screens** group, fill in the worksheet as shown below.



Screen	Label
main	Main Screen

6. Save and close the Mobile Access Configuration.

Verification and Test

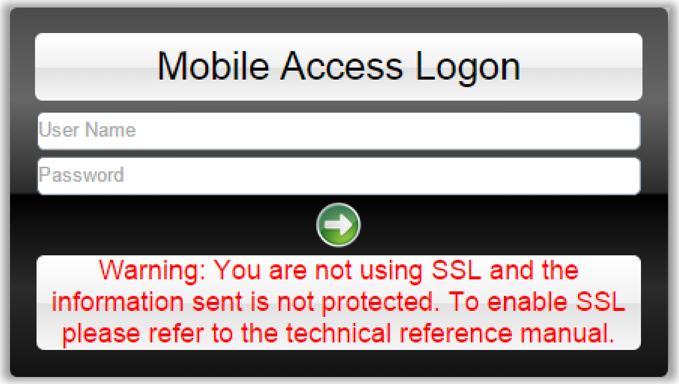
1. If your runtime is on a remote machine, then use the Remote Management tools on the Home tab of the Ribbon to download your project to the remote machine.

2. Start the runtime.

3. Open any HTML5 compatible browser like Chrome or Internet Explorer on some machine (could be your development machine).

4. In the **URL** field, enter `http://<host-name-or-IP-address>/MA`, where `<host-name-or-IP-address>` is the name or IP address of your runtime machine.

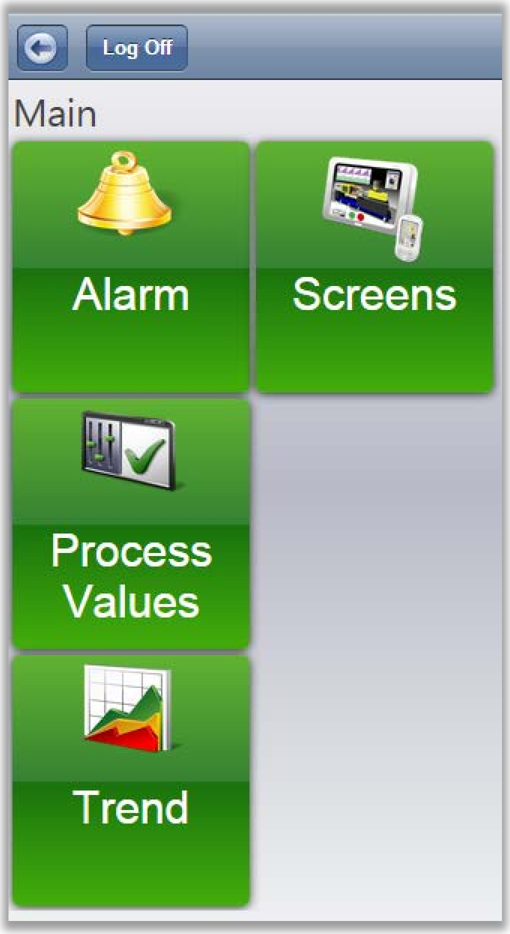
5. The Mobile Access Logon is opened.



Mobile Access Logon

Warning: You are not using SSL and the information sent is not protected. To enable SSL please refer to the technical reference manual.

6. Log on as Administrator.
7. The Mobile Access tiles are displayed.



8. Click on the different tiles and observe that you can view and modify alarms, process values, trends, and screens.

Pro Tip: You can open any screen directly by entering a URL similar to `http://<host-name-or-IP-address>/MA/index.html?screen=<screen-name>`.

Module 12 - Using Databases

SECTION 1 - THE PROJECT DATABASE

This section explains using a step-by-step tutorial how to configure a project database for Alarms, Events, and Trends.

Concept

Using the database interface, the project can easily get or set data to a third party system.

Tasks that support Database:

- Alarm
- Event
- Trend
- Viewer (Grid, Combination Box, ...)
- Web

Overview

Configuring a database interface with BLUE Open Studio is basically linking tasks from BLUE Open Studio (Alarms, Events or Trends) to tables of external databases via a specific Database Provider that supports the database you have chosen.

Each history task (Alarm, Events or Trend) can be configured to save data either to files with the proprietary format from BLUE Open Studio, the Wonderware Historian or to external SQL Relational Databases. Use the Options tab to configure the database to save Alarm and Event history. (See the Trend Folder for instructions for saving history for the trend tasks.)

BLUE Open Studio supports ADO.NET to provide an intuitive, simple, flexible and powerful interface with standard technologies from MDAC (Microsoft Data Access Components) such as OLE-DB (Object Linking Embedded — Database) and ODBC (Open Database Connectivity). By using this capability, you can connect to any database that is MDAC compatible.

The following tasks support the database interface:

- **Alarms:** The project can save and/or retrieve the alarm history messages in a relational database.
- **Events:** The project can save and/or retrieve the event messages in a relational database.
- **Trends:** The project can save and/or retrieve the Trend history values in a relational database.
- **Viewer:** Database information can be displayed both in table format (Alarm/Event Control and Grid objects) or in a graphical format (Trend Control object).
- **Web:** Because the items listed below are already available in BLUE Open Studio Web interface, you can deploy a project that stores/saves data in a relational database and have it working over the Web.

Using its included database interface, BLUE Open Studio can easily provide data from the plant floor to third-party systems (e.g., ERP) or get data from them.

BLUE Open Studio can interface with any relational database supported by a valid ADO.NET Provider, OLE DB provider or ODBC driver.

SQL Relational Databases

A SQL Relational Database is a set of information stored in tables with fields and registers, which support SQL commands.

Each database can have one or more tables. Each table is composed of fields (columns) and registers (rows). Typically, the fields are pre-defined and the project adds or reads one or more registers, according to the query condition.

BLUE Open Studio uses Database Providers (ADO.NET) to interface with SQL Relational Databases. Database Providers are libraries developed to access data from different databases through SQL commands. The ADO.NET Provider for a specific database can be supplied by the operating system or by the database manufacturer.

Note: BLUE Open Studio provides the interface for ADO.NET Providers. However, the ADO.NET Providers and/or the ODBC Driver/OLE DB Provider must be supplied either by the operating system or by the database manufacturer. If your Connection String does not refer to a valid ADO.NET Provider, the OLE.DB Provider will be used.

Although most projects typically link to only one type of database, BLUE Open Studio gives you the flexibility to link each task to a specific database supported by a Database Provider. Furthermore, by using this architecture, you do not need to worry about the specific characteristics of each database (it is mostly handled by the Database Provider for each database or by the BLUE Open Studio Database Gateway interface). Therefore, the project settings are mostly uniform, regardless of the specific database chosen by you.

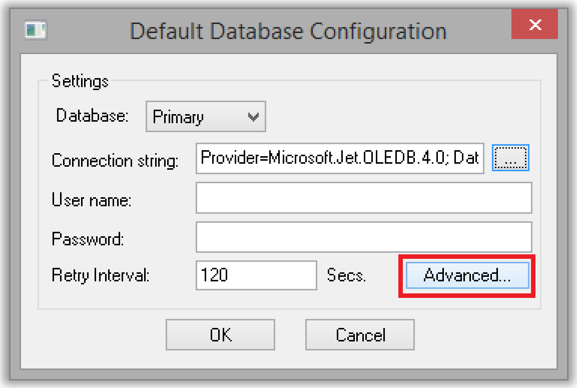
Linking the Database Through a Remote DB Provider

Depending on the architecture of your project, the ADO.NET Provider for the SQL Relational Database may not be available in the same stations where BLUE Open Studio is running. This scenario is especially common when the project is run on a Windows Embedded Compact target system, as most providers are not compiled for embedded systems. In order to solve this problem, BLUE Open Studio supports a flexible solution that allows you to configure distributed systems.

The project can communicate with the BLUE Open Studio Database Gateway (running in a remote computer) via TCP/IP. The Gateway implements the interface with the Database through the Database Provider available in the computer where it is running.

The Studio Database Gateway does not require complex configuration. Simply locate the files STADOSvr.exe and StADOSrv.ini from the \BIN sub-folder of the BLUE Open Studio installation folder (typically C:\Program Files (x86)\BLUE Open Studio v8.0) and paste them into any folder of the computer that will be used as the Gateway station. Once the files have been placed on the remote computer, execute the STADOSvr.exe program.

Configuring the project to communicate through the remote gateway computer instead of running the gateway requires only one step. When configuring the database connection, such as when configuring the default database, click the Advanced button.



Database: Primary

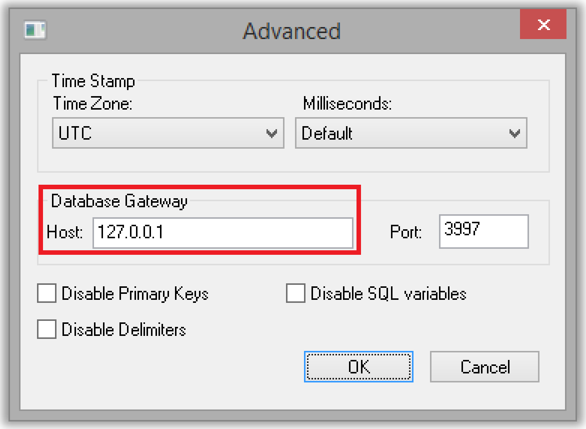
Connection string: Provider=Microsoft.Jet.OLEDB.4.0; Dat

User name:

Password:

Retry Interval: 120 Secs.

On the Advanced dialog, simply change the Database Gateway Host field to the IP address of the remote gateway computer.



Time Zone: UTC

Milliseconds: Default

Host: 127.0.0.1

Port: 3997

☐ Disable Primary Keys

☐ Disable SQL variables

☐ Disable Delimiters

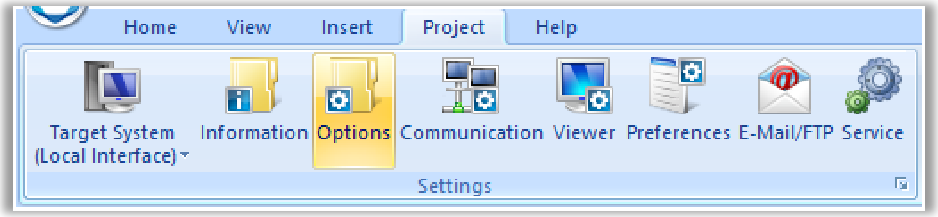
Pro Tip: The Database Gateway is a TCP/IP Server for the BLUE Open Studio project and it uses the TCP Port 3997 by default. You can specify a different port number when executing the STADOSvr.exe program according to the following syntax: `STADOSvr.exe Port Number` (Example: `STADOSvr 3998`)

LAB 23 - CONFIGURING A PROJECT DATABASE

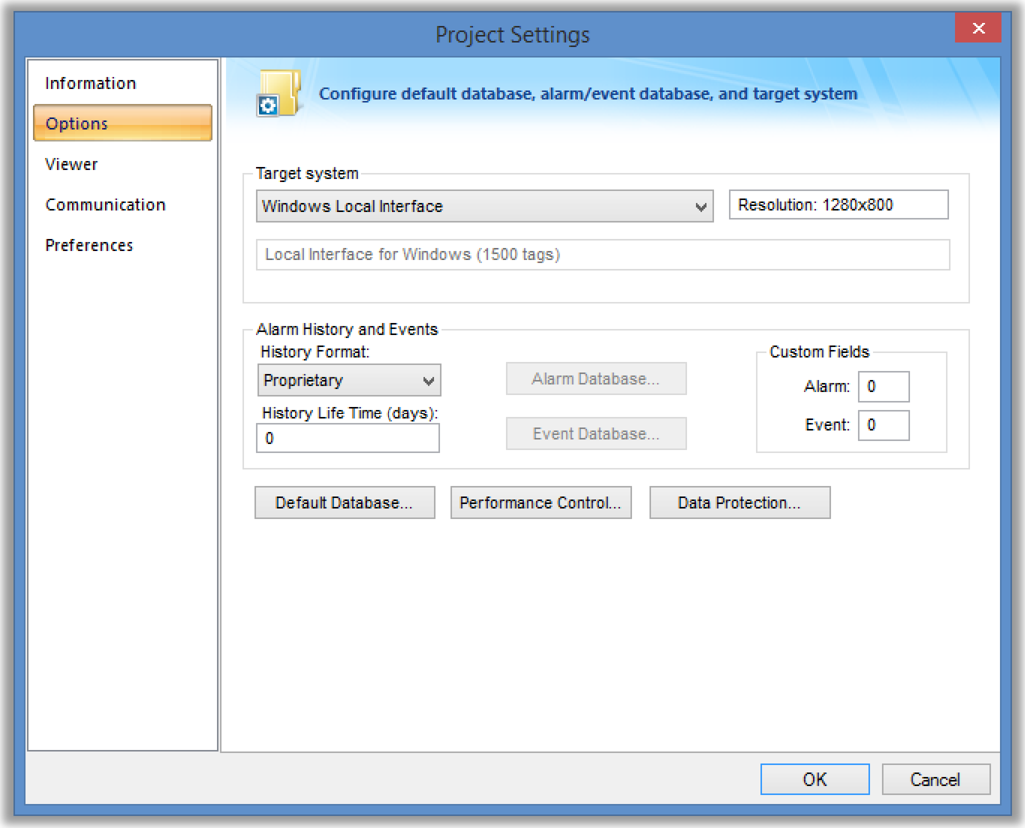
In this lab you will configure a default database connection to SQL Server Express and configure the Alarm task, Event Logger, and Trend task to write to the database.

Configuring the database connection

1. In the Project tab of the Ribbon, select **Options** located in the Settings group.



2. The Project Settings dialog is opened.



Target system: Windows Local Interface

Resolution: 1280x800

Local Interface for Windows (1500 tags)

Alarm History and Events

History Format: Proprietary

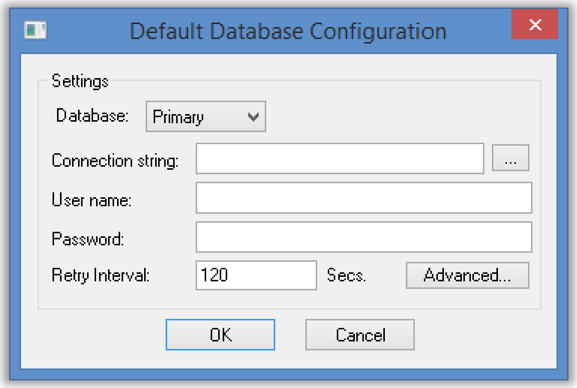
History Life Time (days): 0

Custom Fields

Alarm: 0

Event: 0

3. Click the button called **Default Database**. The Default Database Configuration dialog is opened.



Default Database Configuration

Settings

Database: Primary

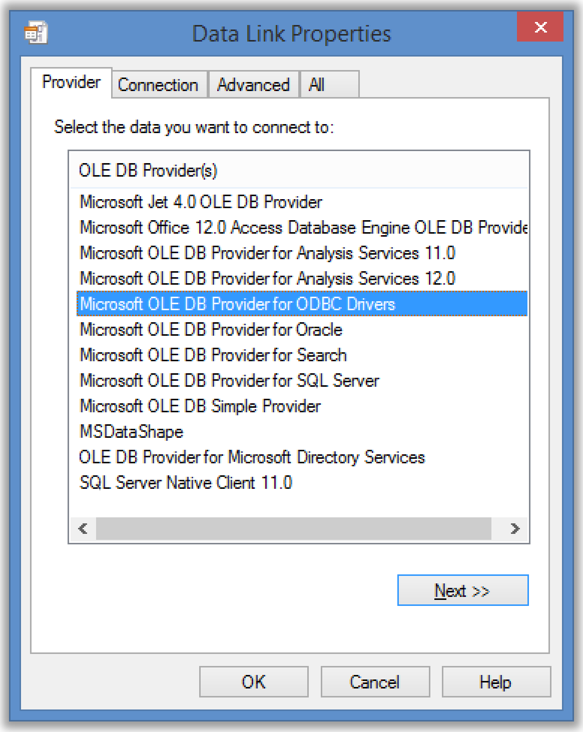
Connection string:

User name:

Password:

Retry Interval: 120 Secs.

4. To the right of the **Connection String** field, select the **Browse** (…) button. The Data Link Properties dialog is opened.



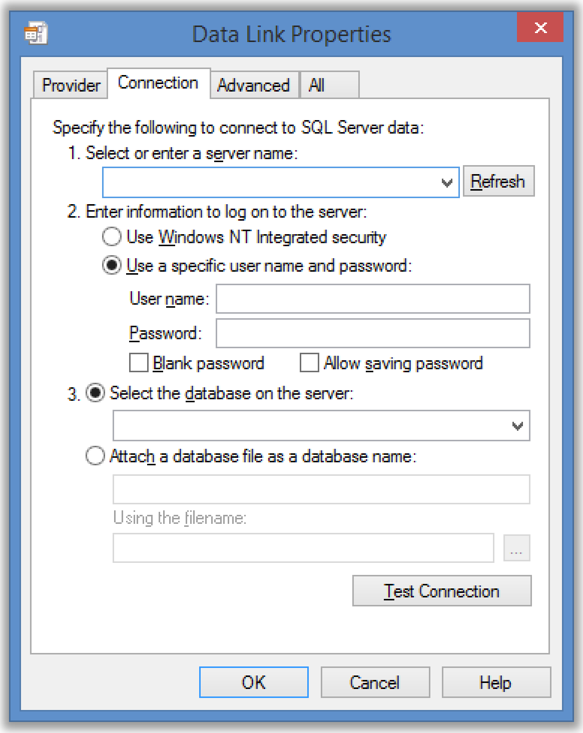
Data Link Properties

Select the data you want to connect to:

OLE DB Provider(s)
Microsoft Jet 4.0 OLE DB Provider
Microsoft Office 12.0 Access Database Engine OLE DB Provide
Microsoft OLE DB Provider for Analysis Services 11.0
Microsoft OLE DB Provider for Analysis Services 12.0
Microsoft OLE DB Provider for ODBC Drivers
Microsoft OLE DB Provider for Oracle
Microsoft OLE DB Provider for Search
Microsoft OLE DB Provider for SQL Server
Microsoft OLE DB Simple Provider
MSDataShape
OLE DB Provider for Microsoft Directory Services
SQL Server Native Client 11.0

Note: The list of OLE DB Provider(s) is derived from the list of providers on your machine and may be different than those shown.

5. Select **Microsoft OLE DB Provider for SQL Server** and click **Next**. The Connection tab of the Data Link Properties dialog is opened.



Data Link Properties

Provider | Connection | Advanced | All

Specify the following to connect to SQL Server data:

1. Select or enter a server name:

   Refresh

2. Enter information to log on to the server:

   ○ Use Windows NT Integrated security

   ● Use a specific user name and password:

   User name:

   Password:

   □ Blank password □ Allow saving password

3. ● Select the database on the server:

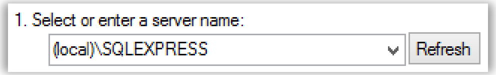
   ○ Attach a database file as a database name:

   Using the filename:

Test Connection

OK Cancel Help

6. In the field number 1, enter `(local)\SQLEXPRESS`.

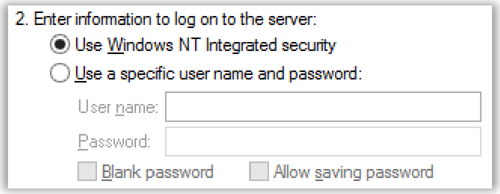


1. Select or enter a server name:

(local)\SQLEXPRESS

Refresh

7. In the field number 2, select **Use WindowsNT integrates security**.



2. Enter information to log on to the server:

● Use Windows NT Integrated security

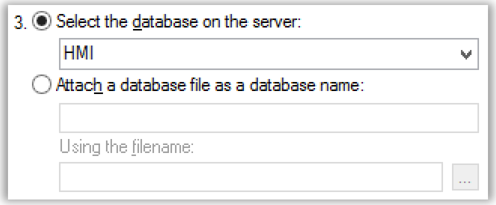
○ Use a specific user name and password:

User name:

Password:

□ Blank password □ Allow saving password

8. In the field number 3, select **Select the database on the server**, and then use the combobox to select the database called **HMI**.



3. ● Select the database on the server:

HMI

○ Attach a database file as a database name:

Using the filename:

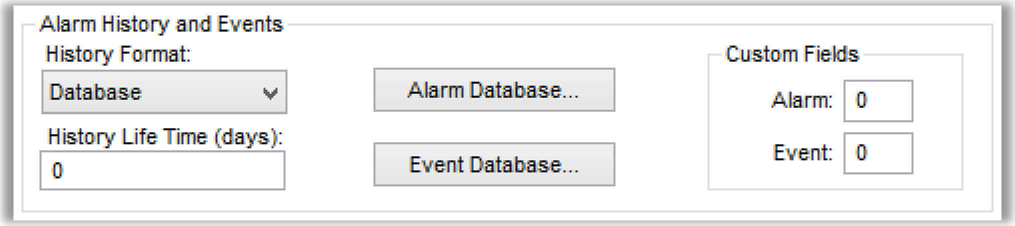
9. Click **OK** to close the Data Link Properties dialog.

10. Click **OK** to close the Default Database Configuration dialog.

Configure the project tasks to use the project database.

1. The Project Settings dialog should be open from the previous steps. If not, then open it and select the **Options** tab.

2. In the Alarm History and Events group, change the **History Format** field to **Database**.



Alarm History and Events

History Format: Database

History Life Time (days): 0

Custom Fields

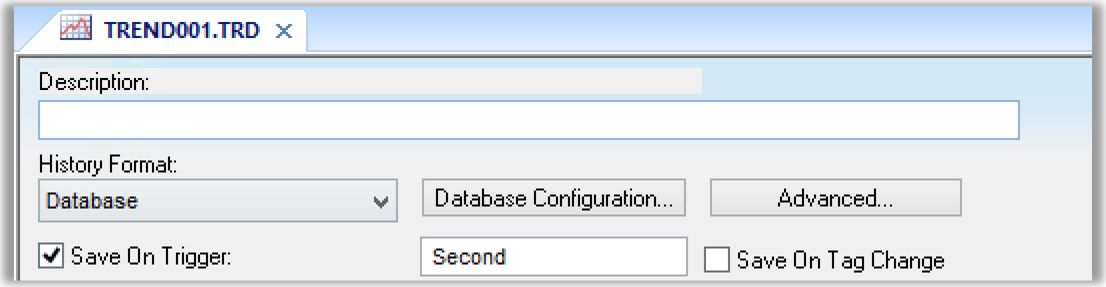
Alarm: 0

Event: 0

3. Click **OK** to close the Project Settings dialog.

4. Open the Trend task worksheet number 1.

5. Change the **History Format** field to **Database**.



Description:

History Format: Database

☑ Save On Trigger: Second

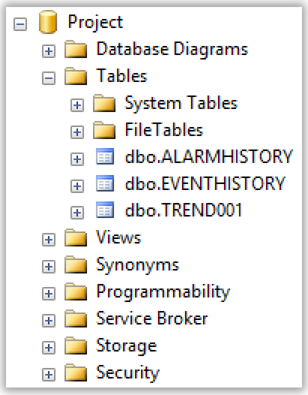
□ Save On Tag Change

6. Save and Close the Trend worksheet.

7. If your runtime is on a remote machine, then use the the Remote Management tools on the Home tab of the Ribbon to download your project to the remote machine. Start the runtime.

8. Open and close your tank valves to cause alarms to be activated.

9. Open SQL Server Management Studio and observe that tables have been created and populated for Alarms, Events, and Trends.



```
⊟ Project
  ⊞ Database Diagrams
  ⊟ Tables
    ⊞ System Tables
    ⊞ FileTables
    ⊞ dbo.ALARMHISTORY
    ⊞ dbo.EVENTHISTORY
    ⊞ dbo.TREND001
  ⊞ Views
  ⊞ Synonyms
  ⊞ Programmability
  ⊞ Service Broker
  ⊞ Storage
  ⊞ Security
```

SECTION 2 - DATABASE/ERP TASK

This section explains, using a step-by-step tutorial, how to configure a connection with an Access Database using the Database/ERP task preconfigured for use with this tutorial.

You will configure a connection with Access Database to read/write the Tank's values.

Concept

The Database/ERP task is used to associate Project tags with external database fields. First, a connection with the database must be configured. Once this connection is defined, commands can be easily sent to the database using built-in functions or a worksheet.

Database/ERP worksheets permit the asynchronous execution of database operations and offer a friendly interface for building SQL commands.

Overview

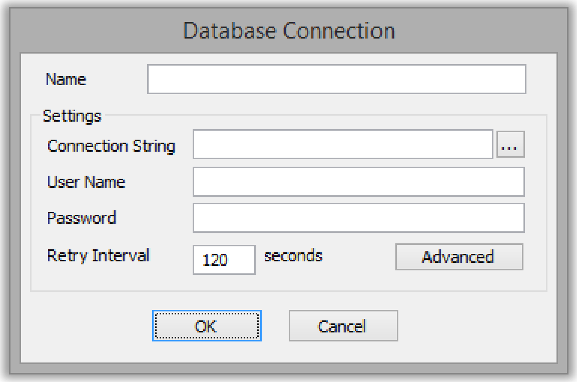
Just as in the Database connections are configured for connecting the BLUE Open Studio tasks to a SQL Relational database, the Database/ERP connections allow you to connect to, and run SQL queries on any other database to retrieve and store data.

Note: The Database/ERP functions and worksheet will create the SQL Connection Strings and convert simple-language requests into SQL for you. However, it is important to be familiar with how SQL commands are formed and executed before using this feature.

In order to utilize the Database/ERP functionality, you must first create a Connection and then send commands to the database, either by using the Database Functions or the Database/ERP worksheet.

Database/ERP Connection

The Database/ERP Connection dialog allows you to create named connections that can be called at will by the Database functions or worksheet.



Database Connection

Name

Settings

Connection String

User Name

Password

Retry Interval 120 seconds

Similar to the Default Database Configuration dialog used in the previous section, the Database Connection dialog allows you to connect to any SQL database accessible by the runtime device. By naming the connections, it allows for easy access to a set database. This only requires one connection per database, so one named DB/ERP connection can access all tables within that one database.

Once a connection is created, it is stored as XML files in the project folder in the \Config sub-folder. Each file is given the same name as the name of the connection with the .XDC file extension.

Note: Database worksheets are executed under the Database Client Runtime task. However, creating a new worksheet does not automatically enable the task; you must use the Execution Tasks dialog (Tasks on the Home tab of the Ribbon) to configure the task to start at runtime.

Database Functions

The Database/ERP functions are commands in the BLUE Open Studio built-in language to send SQL commands directly to the named database. Since these are functions using the built-in language,they can be used practically anywhere within your project (such as a command animation, scheduler worksheet, graphics script, etc.).

The available functions are listed in the table below:

Function	Description
DBCursorClose	Closes the cursor and releases the result set.
DBCursorCurrentRow	Returns the row number of the current row (i.e., the cursor position) in a SQL result set.
DBCursorGetValue	Gets the value of the specified column of the current row (i.e., the cursor position) in a SQL result set.
DBCursorMoveTo	Moves the cursor to the specified row in a SQL result set and copies that row's values to the mapped tags.
DBCursorNext	Moves the cursor to the next row in a SQL result set and copies that row's values to the mapped tags.
DBCursorOpen	Selects a set of rows and columns in a database table, initializes the cursor at the first row of the result set, copies that row's values to mapped tags, and then returns a cursor handle that can be referenced by other DB/ERP functions.
DBCursorOpenSQL	Selects a set of rows and columns in a database table, initializes the cursor at the first row of the result set, copies that row's values to mapped tags, and then returns a cursor handle that can be referenced by other DB/ERP functions. (This function is equivalent to a SQL SELECT statement.)
DBCursorPrevious	Moves the cursor to the previous row of the result set and copies that row's values to the mapped tags.
DBCursorRowCount	Gets the total number of rows in a SQL result set.
DBDelete	Deletes selected rows from a database table. (This function is equivalent to a SQL DELETE statement.)
DBExecute	A function that executes a custom SQL statement on a database. If the statement is a query (e.g., SELECT), then the database values are copied to specified array tags.

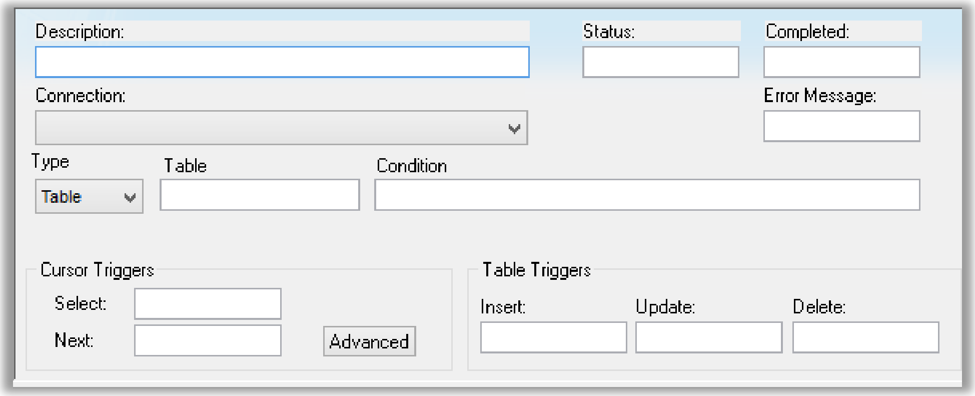
Function	Description
DBInsert	Inserts one new row into a database table. (This function is equivalent to a SQL INSERT statement.)
DBSelect	A function that selects a result set from a database (equivalent to a SQL SELECT statement), maps the columns to array tags in your project, and copies the values from the result set to the array tags.
DBUpdate	Selects a result set and then writes the same value to all rows of a specified column. (This function is equivalent to a SQL UPDATE statement.)

Database/ERP Worksheet

Database worksheets allow asynchronous execution of database operations, and they offer a user-friendly interface for building SQL commands. Once a Database/ERP connection is made, the worksheet links the BLUE Open Studio tag with a field in the database, and the most commonly used commands can be executed in the database by use of trigger tags.

The worksheet has two parts: the header and body.

Header: The worksheet header identifies the database and subsequent table being connected to, any necessary conditional statements, as well as the triggers for the most common queries, such as Select, Insert, Delete, and Update.



Header

The header of the database worksheet is configured as follows:

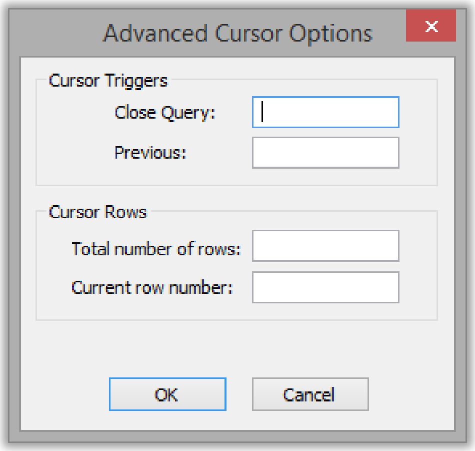
- **Description**: Enter a description of the worksheet, for documentation purposes.
- **Status:** Enter the name of a numeric tag that will receive status codes for database operations during runtime
- **Completed:** Enter the name of a numeric tag that will be toggled when database commands are successfully executed.

- **Error Message:** Enter the name of a string tag that will receive detailed error messages, if errors occur during runtime.
- **Connection combo-box:** Click to select a connection to the target database. All available connections are listed, as configured with the Database Connection dialog described above.
- **Type combo-box:** Click to specify how the result set will be selected for the worksheet:
  - **Table:** Enter a table name and an optional filter condition. (The filter condition is equivalent to the SQL "Where" clause.) All rows of the table that match the filter condition are selected.
  - **SQL:** Enter a custom SQL "Select" statement.

Note: For Table, Condition and SQL Statement, you can enter the names of project tags that contain the desired information by enclosing a tag name within curly brackets (**{}**). This lets you programmatically change the selection during runtime.

Cursor Triggers

- **Select:** When the tag configured in this field changes, a new cursor opens the first row of the result set and copies those values to the tags configured in the worksheet body.
- **Next:** When the tag configured in this field changes, the cursor moves to the next row of the result set and copies those values to the tags configured in the worksheet body.
- **Advanced:** Click to open the Advanced Cursor Options dialog…



  - **Close Query:** When the tag configured in this field changes, the cursor releases the current result set.
  - **Previous:** When the tag configured in this field changes, the cursor moves to the previous row of the result set and copies those values to the tag configured in the worksheet body.
  - **Total number of rows:** Enter a numeric tag that will receive the total number of rows in the result set.

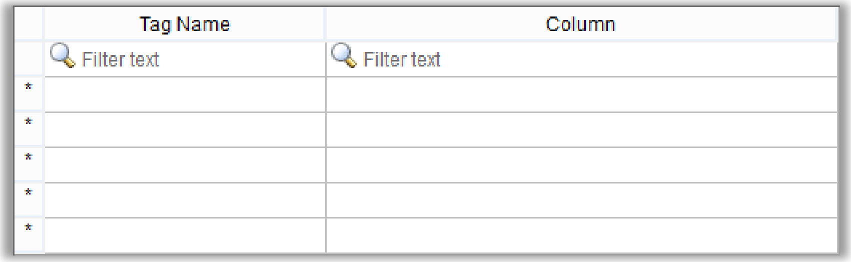
  - **Current row number:** Enter a numeric tag that will receive the number of the current row (i.e., the position of the cursor). When a result set is first opened using the **Select** trigger, this number is 1. Each **Next** trigger increments this number, and each **Previous** trigger decrements it.

Table Triggers

- **Insert:** When the tag configured in this field changes, a new row is inserted with the current values of the tags configured in the worksheet body.
- **Update:** When the tag configured in this field changes, all rows of the result set are overwritten with the current values of the tags configured in the worksheet body.
- **Delete**: When the tag configured in this field changes, all rows of the result set are deleted.

Body

The body links the BLUE Open Studio tag with a column (field) in the database.



	Tag Name	Column
	Filter text	Filter text
*		
*		
*		
*		
*		

In the body of the worksheet, you can map project tags to the columns (fields) of the result set. For each row of the body, enter a Tag Name and its corresponding Column. Which columns are available depends on how the result set is selected, and how it is selected may change during runtime, so be sure to map all necessary columns.

Note: When using the Database/ERP worksheet, remember that before you create a new selection, you must first release the existing one by using the Close Query trigger configured in the Advanced dialog

LAB 24 - USING THE DATABASE/ERP TASK

In this lab you will use the Database/ERP task to store and manage recipes in a Microsoft SQL Server database.

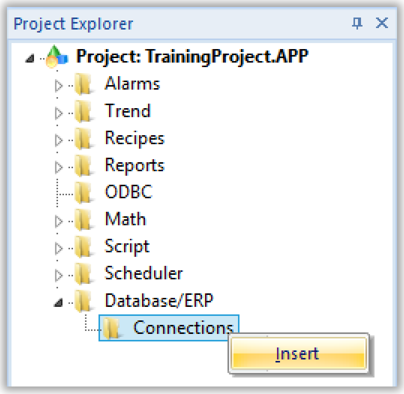
Prepare the SQL Server Database

This lab requires that you have SQL Server Express installed with the default options on the machine where BLUE Open Studio is running and on the runtime machine, if different. SQL Server Express should have a database called HMI with a table called Recipes created as follows:

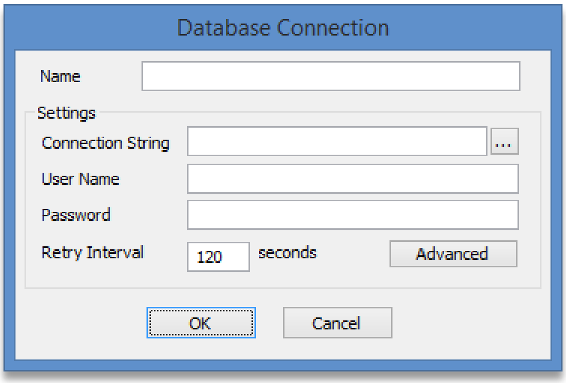
```
USE HMI
CREATE TABLE Recipes(
      Name nvarchar(50) NOT NULL,
      A real NOT NULL,
      B real NOT NULL,
      C real NOT NULL,
      D real NOT NULL
)
```

Create a Database/ERP Connection

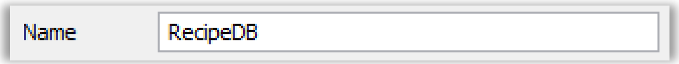
1. On the Tasks tab of the Project Explorer, expand the **Database/ERP** folder, right-click the **Connections** folder, and click **Insert** on the shortcut menu.



2. The Database Connection dialog is opened.

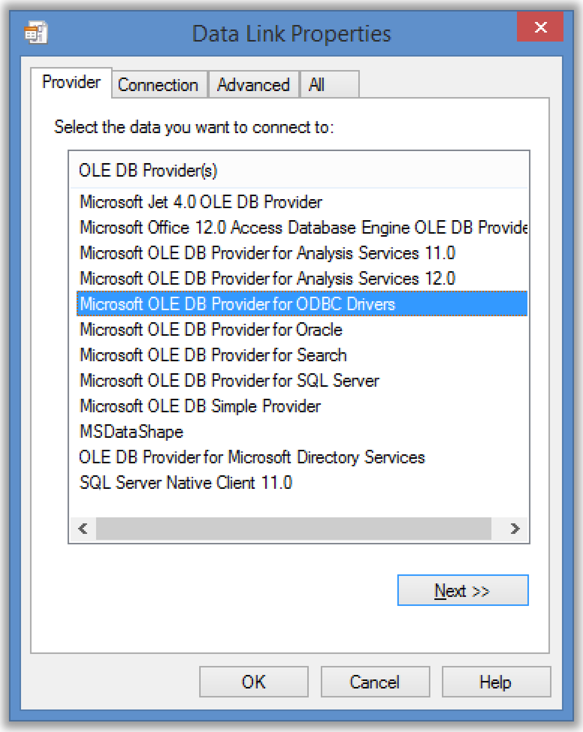


3. In the **Name** field enter `RecipeDB`.



Name RecipeDB

4. To the right of the **Connection String** field, select the browse **(…)** button. The Data Link Properties dialog is opened.



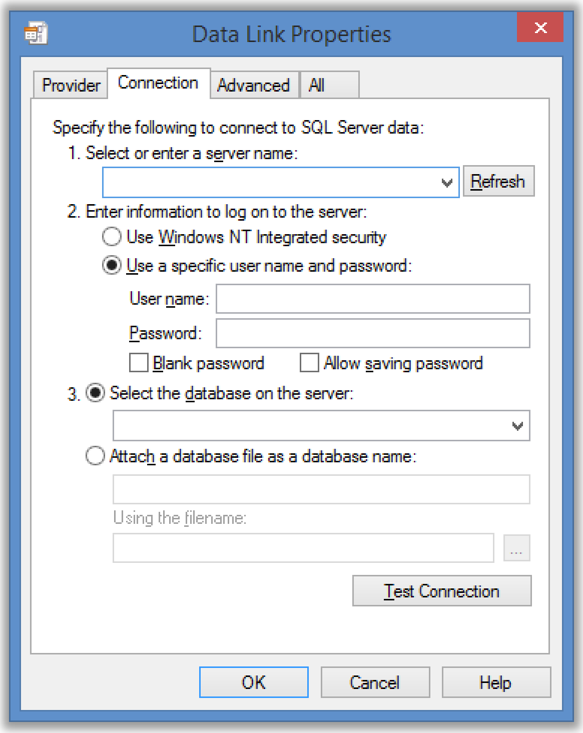
Select the data you want to connect to:

OLE DB Provider(s)

Microsoft Jet 4.0 OLE DB Provider
Microsoft Office 12.0 Access Database Engine OLE DB Provide
Microsoft OLE DB Provider for Analysis Services 11.0
Microsoft OLE DB Provider for Analysis Services 12.0
Microsoft OLE DB Provider for ODBC Drivers
Microsoft OLE DB Provider for Oracle
Microsoft OLE DB Provider for Search
Microsoft OLE DB Provider for SQL Server
Microsoft OLE DB Simple Provider
MSDataShape
OLE DB Provider for Microsoft Directory Services
SQL Server Native Client 11.0

Note: The list of OLE DB Provider(s) is derived from the list of providers on your machine and may be different than those shown.

5. Select **Microsoft OLE DB Provider for SQL Server** and click **Next**. The Connection tab of the Data Link Properties dialog is opened.



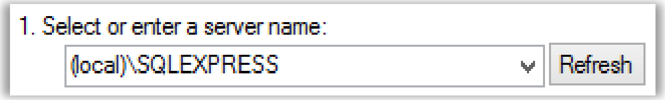
Specify the following to connect to SQL Server data:

1. Select or enter a server name:

2. Enter information to log on to the server:
   - ○ Use Windows NT Integrated security
   - ● Use a specific user name and password:
   - User name:
   - Password:
   - □ Blank password □ Allow saving password

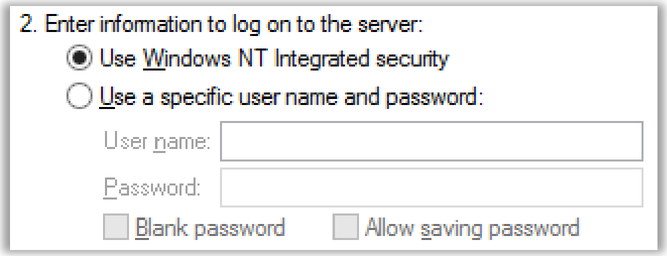
3. ● Select the database on the server:
   - ○ Attach a database file as a database name:
   - Using the filename:

6. In the field number 1, enter `(local)\SQLEXPRESS`.



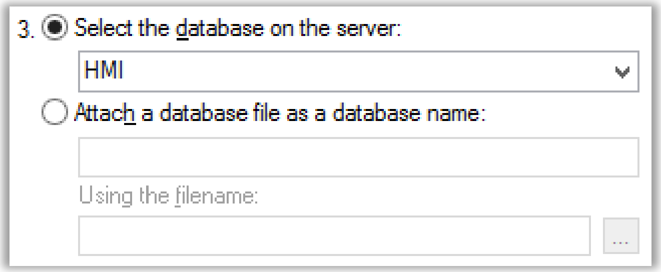
1. Select or enter a server name:
(local)\SQLEXPRESS

7. In the field number 2, select **Use WindowsNT integrates security**.



2. Enter information to log on to the server:
   - ● Use Windows NT Integrated security
   - ○ Use a specific user name and password:
   - User name:
   - Password:
   - □ Blank password □ Allow saving password

8. In the field number 3, select **Select the database on the server**, and then use the combobox to select the database called **HMI**.



3. ● Select the database on the server:

HMI

○ Attach a database file as a database name:

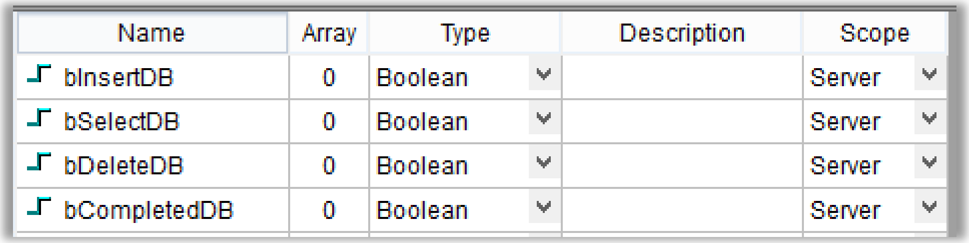
Using the filename:

9. Click **OK** to close the Data Link Properties dialog.

10. Click **OK** to close the Database Configuration dialog.

Create a Database/ERP Worksheet

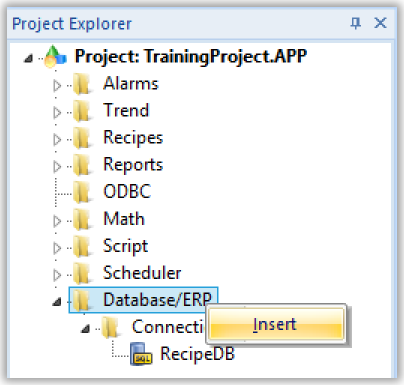
1. Open the Project Tags Datasheet View and create the tags shown below.



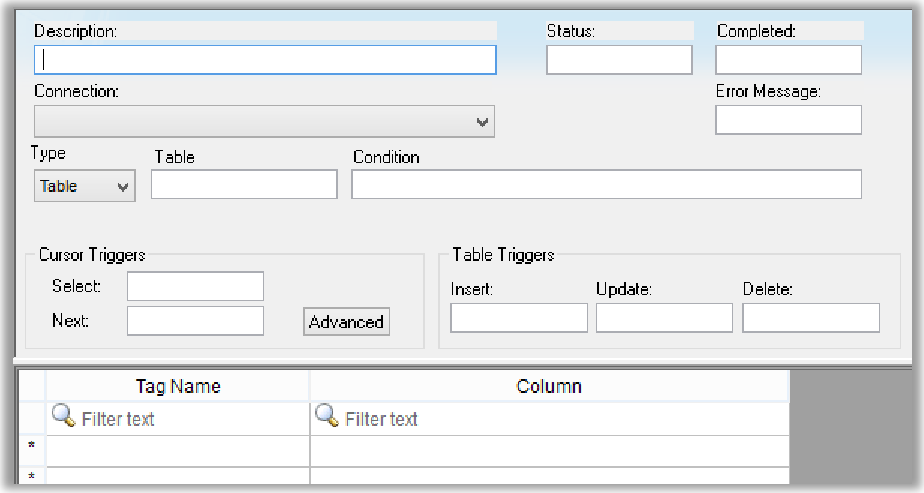
Name	Array	Type	Description	Scope
bInsertDB	0	Boolean		Server
bSelectDB	0	Boolean		Server
bDeleteDB	0	Boolean		Server
bCompletedDB	0	Boolean		Server

Note: This lab also uses the tags called cRecipe and sLoadRecipeName created in the Using Recipes lab of this training manual.

2. On the Tasks tab of the Project Explorer, right-click the **Database/ERP** folder, and click **Insert** on the shortcut menu.

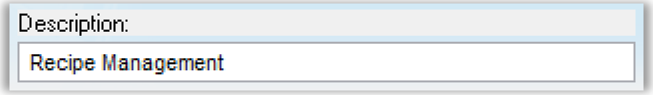


3. A blank Database/ERP worksheet is created and displayed in the Workspace.



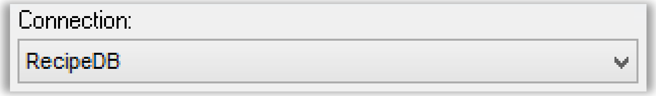
Type: Table

4. In the field called **Description**, enter `Recipe Management`.



Description: Recipe Management

5. Use the combobox beneath the field called **Connection** and select **RecipeDB**.



Connection: RecipeDB

6. In the field called **Table**, enter `Recipes`.

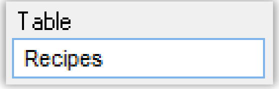
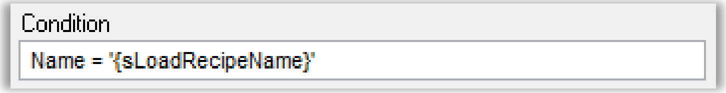


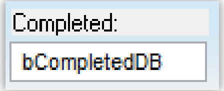
Table: Recipes

7. In the field called **Condition**, enter `Name = '{sLoadRecipeName}'`.



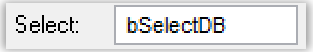
Condition: Name = '{sLoadRecipeName}'

8. In the field called **Completed**, enter `bCompletedDB`.



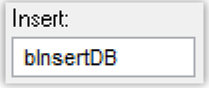
Completed: bCompletedDB

9. In the field called **Select**, enter `bSelectDB`.



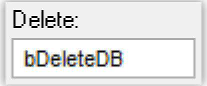
Select: bSelectDB

10. In the field called **Insert**, enter `bInsertDB`.



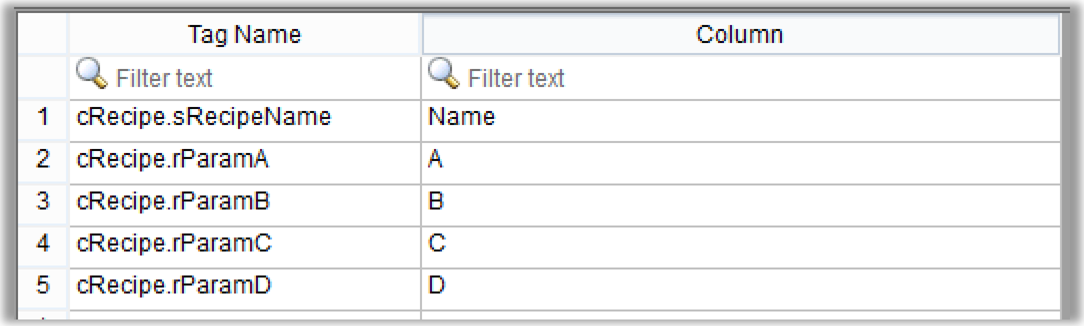
Insert:
bInsertDB

11. In the field called **Delete**, enter `bDeleteDB`.



Delete:
bDeleteDB

12. Configure the columns called **Tag Name** and **Column** in the body of the worksheet as shown below.

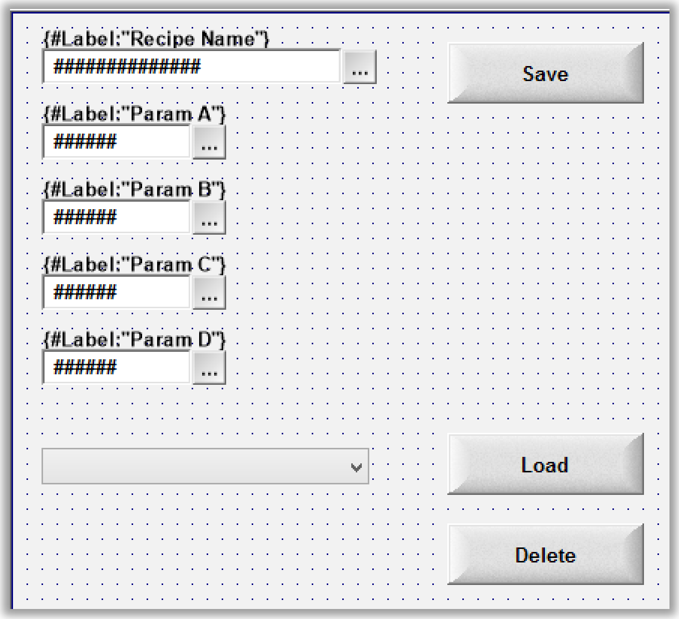


	Tag Name	Column
	Filter text	Filter text
1	cRecipe.sRecipeName	Name
2	cRecipe.rParamA	A
3	cRecipe.rParamB	B
4	cRecipe.rParamC	C
5	cRecipe.rParamD	D

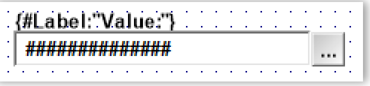
13. Save the worksheet as sheet number 1 and close worksheet DB001.

Create Database Screen

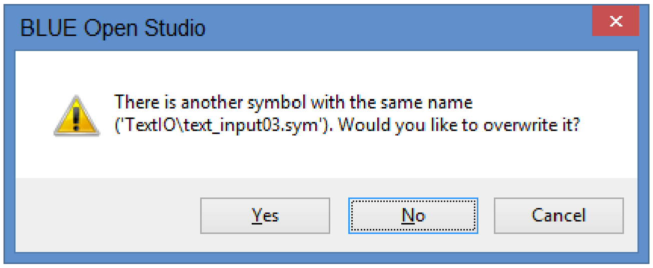
In this portion of this lab you will create a screen that looks similar to the picture below.



1. Open the Template screen.
2. Click the Application button and click **Save As** on the Application menu. The Save As dialog is opened.
3. In the **File name** field, enter `Database`.
4. Click **Save**. The Save As dialog is closed.
5. On the Graphics tab of the Ribbon, in the Libraries group, select **Symbols**. The Symbols library is opened on the Workspace.
6. Expand the **System Symbols** folder, and then click on the folder called **TextIO**.
7. Select the symbol called **text_input03**.
8. Click on the **Database** tab on the Workspace to select the Database screen, and then click anywhere on the Database screen to paste the symbol.

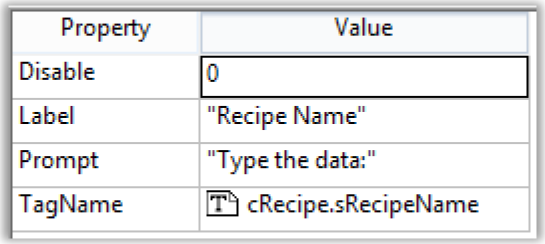


If you see the following message, then click **Yes**.



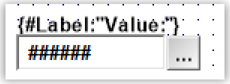
There is another symbol with the same name ('TextIO\text_input03.sym'). Would you like to overwrite it?

9. Open the Object Properties of the symbol and configure them as shown below.

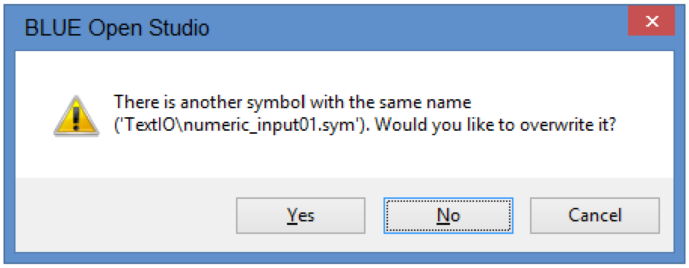


Property	Value
Disable	0
Label	"Recipe Name"
Prompt	"Type the data:"
TagName	cRecipe.sRecipeName

10. Click on the **Symbols** tab on the Workspace, and select the symbol called **numeric_input1**.
11. Click on the **Database** tab on the Workspace to select the Database screen, and then click anywhere on the Database screen to paste the symbol.

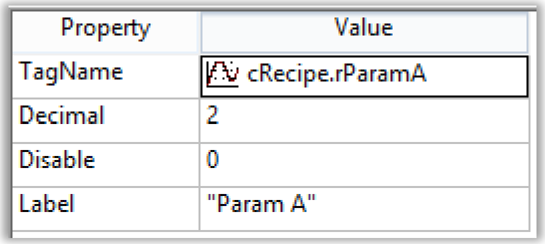


If you see the following message, then click **Yes**.


BLUE Open Studio

There is another symbol with the same name ('TextIO\numeric_input01.sym'). Would you like to overwrite it?

12. Open the Object Properties of the symbol and configure them as shown below.



Property	Value
TagName	cRecipe.rParamA
Decimal	2
Disable	0
Label	"Param A"

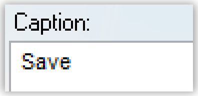
13. Repeat steps 10 through 13 for the tags called cRecipe.rParamB, cRecipe.rParamC, and cRecipe.rParamD.

14. Close the Symbols tab on the Workspace.

15. On the Graphics tab of the Ribbon, in the Active Objects group, select **Button** and add a button to the screen.

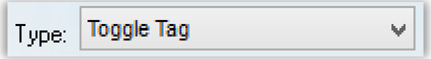
16. Open the Object Properties of the button.

17. In the field called **Caption**, enter `Save`.

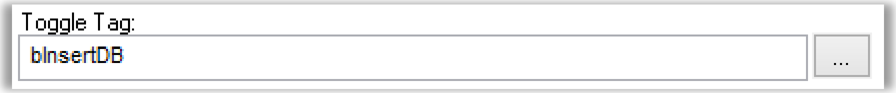

Caption:
Save

18. Click **Command**. The Command animation object properties are displayed.

19. Change the field called **Type** to **Toggle Tag**.


Type: Toggle Tag

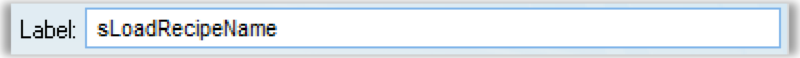
20. In the field called Toggle Tag, enter `bInsertDB`.


Toggle Tag:
bInsertDB

21. On the Graphics tab of the Ribbon, in the Active Objects group, select **Combo Box** and add a combo box to the screen.

22. Open the Object Properties of the combo box.

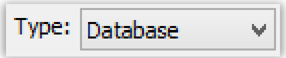
23. In the field called **Label**, enter sLoadRecipeName.



Label: sLoadRecipeName

24. Click **Data Sources**. The Data Sources dialog is opened.

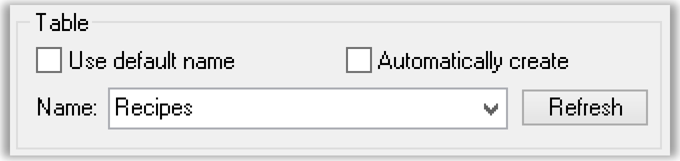
25. Change the field called **Type** to **Database**.



Type: Database

26. Click **Settings**. The Database Configuration dialog is opened.

27. In the section called **Table**, clear the checkbox called **Use default name**, and in the field called **Name** enter `Recipes`.



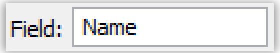
Table

☐ Use default name ☐ Automatically create

Name: Recipes

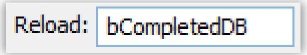
28. Click **OK** to close the Database Configuration dialog.

29. In the field called **Field**, enter `Name`.



Field: Name

30. In the field called **Reload**, enter `bCompletedDB`.

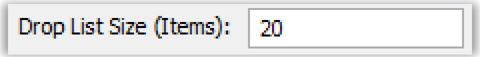


Reload: bCompletedDB

31. Click **OK** to close the Data Sources dialog.

32. Click **Advanced**. The Combo Box - Advanced dialog is opened.

33. In the field called **Drop List Size (Items)**, enter `20`.



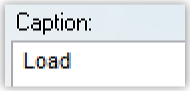
Drop List Size (Items): 20

34. Click **OK** to close the Combo Box - Advanced dialog.

35. On the Graphics tab of the Ribbon, in the Active Objects group, select **Button** and add a button to the screen.

36. Open the Object Properties of the button.

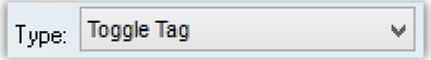
37. In the field called **Caption**, enter `Load`.



Caption:
Load

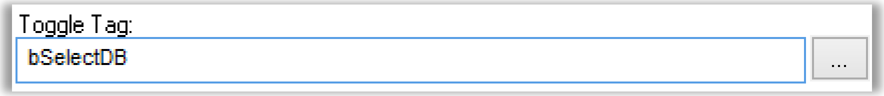
38. Click **Command**. The Command animation object properties are displayed.

39. Change the field called **Type** to **Toggle Tag**.



Type: Toggle Tag

40. In the field called Toggle Tag, enter `bSelectDB`.

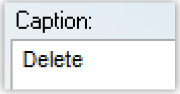


Toggle Tag:
bSelectDB

41. On the Graphics tab of the Ribbon, in the Active Objects group, select **Button** and add a button to the screen.

42. Open the Object Properties of the button.

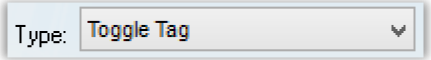
43. In the field called **Caption**, enter `Delete`.



Caption:
Delete

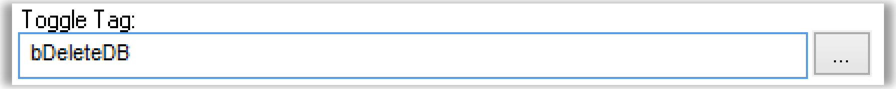
44. Click **Command**. The Command animation object properties are displayed.

45. Change the field called **Type** to **Toggle Tag**.



Type: Toggle Tag

46. In the field called Toggle Tag, enter `bDeleteDB`.



Toggle Tag:
bDeleteDB

47. Save and close the Database screen.

Verification and Test

1. If your runtime is on a remote machine, then use the the Remote Management tools on the Home tab of the Ribbon to download your project to the remote machine.

2. Start the runtime.

3. Open the DATABASE screen.

4. Fill in the text fields and save some recipes.

5. Load some saved recipes.
6. Delete some saved recipes.
7. Use Microsoft SQL Server Management Studio to query the Recipes table to verify that recipe files are being saved and deleted.

Appendicies

Appendix 1 - Keyboard Shortcuts

Global Shortcuts

Key Combination	Action
F1	Opens the Technical Reference Help file
Shift+F1	Enables context sensitive Help
F2	Toggles the status bar
F4	Opens the zoom window
F5	Runs the currently loaded project
Shift+F5	Stops the local project runtime server and client
F10 or Alt	Enables navigational hotkeys
Ctrl+N	Create New Project
Ctrl+F4	Closes the current workspace tab
Ctrl+O	Open project file
Ctrl+S	Saves the current workspace tab
Ctrl+C	Copy
Ctrl+X	Cut
Ctrl+F	Find
Ctrl+V	Paste
Ctrl+P	Prints the current worksheet or screen
Alt+0	Toggles the Project Explorer
Alt+1	Toggles the Output Window
Alt+2	Toggles the Database Spy
Alt+F	Opens the Application (File) menu
Alt+H	Opens the Home tab of the Ribbon
Alt+V	Opens the View tab of the Ribbon
Alt+I	Opens the Insert tab of the Ribbon
Alt+P	Opens the Project tab of the Ribbon
Alt+G	Opens the Graphics tab of the Ribbon
Alt+E	Opens the Help tab of the Ribbon

Workspace Shortcuts (applies to screen/screen objects only)

Key Combination	Action
Ctrl+A	Select all screen objects in current screen
Ctrl+Z	Undo the previous action
Ctrl+G	Toggles the grid
Ctrl+Num+	Bring selected object forward by one (affects object ID)
Ctrl+Num-	Send selected object backward by one (affects object ID)
Ctrl+D	Disable object dragging on the screen editor
Ctrl+<drag>	Create a copy of the selected object(s)
Shift+<arrow>	Moves the selected object one pixel or grid in the direction of the arrow
Tab	Change selection to the next screen object (based on object ID)
Shift+Tab	Change selection to the previous screen object (based on object ID)
Alt+Enter	Opens the Object Properties dialog for the currently selected object

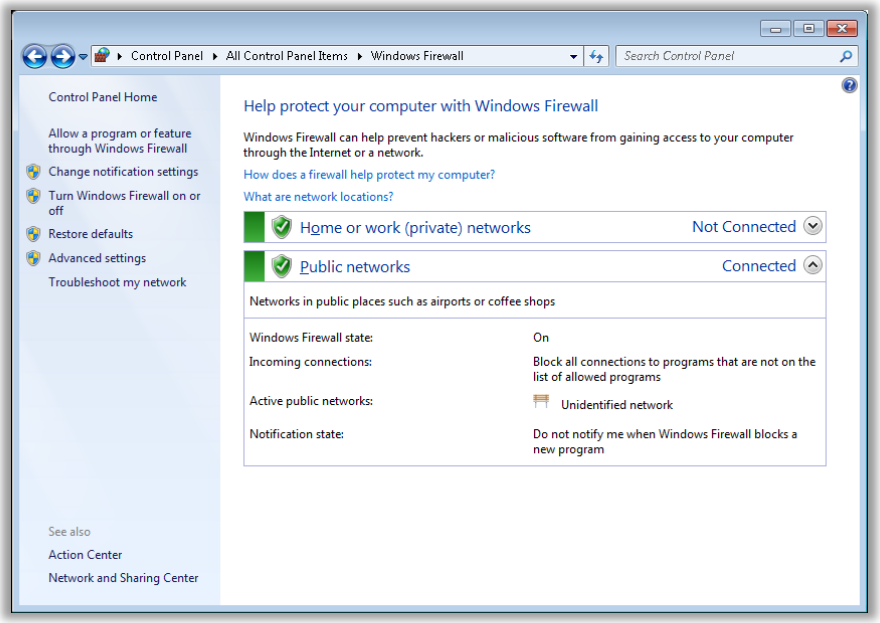
VBScript Debug Mode Shortcuts

Key Combination	Action
F9	Set/Reset a VbScript debug breakpoint
Ctrl+F5	Starts or continues the runtime in debug mode
Ctrl+Scroll Lock	Breaks the VbScript execution
F11	Step into the VbScript debug execution
Shift+F11	Step out of the VbScript debug execution
Shift+F10	Step over the VbScript debug execution

Appendix 2 - Configure Windows Firewall for Remote Access

Firewall configuration for Windows Embedded Standard 7 and above.

1. Open the Control Panel and select **Windows Firewall**.



Control Panel ▸ All Control Panel Items ▸ Windows Firewall
Search Control Panel
Control Panel Home
Allow a program or feature through Windows Firewall
Change notification settings
Turn Windows Firewall on or off
Restore defaults
Advanced settings
Troubleshoot my network
See also
Action Center
Network and Sharing Center

Help protect your computer with Windows Firewall

Windows Firewall can help prevent hackers or malicious software from gaining access to your computer through the Internet or a network.

How does a firewall help protect my computer?

What are network locations?

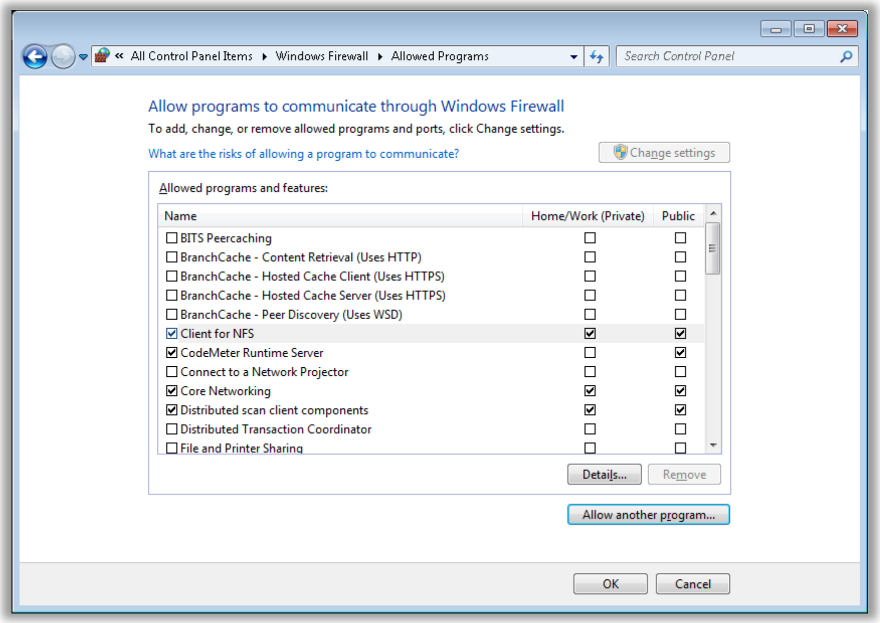
Home or work (private) networks — Not Connected

Public networks — Connected

Networks in public places such as airports or coffee shops

Windows Firewall state:	On
Incoming connections:	Block all connections to programs that are not on the list of allowed programs
Active public networks:	Unidentified network
Notification state:	Do not notify me when Windows Firewall blocks a new program

2. Click the link on the left called **Allow an app or feature through Windows Firewall.**



Allow programs to communicate through Windows Firewall

To add, change, or remove allowed programs and ports, click Change settings.

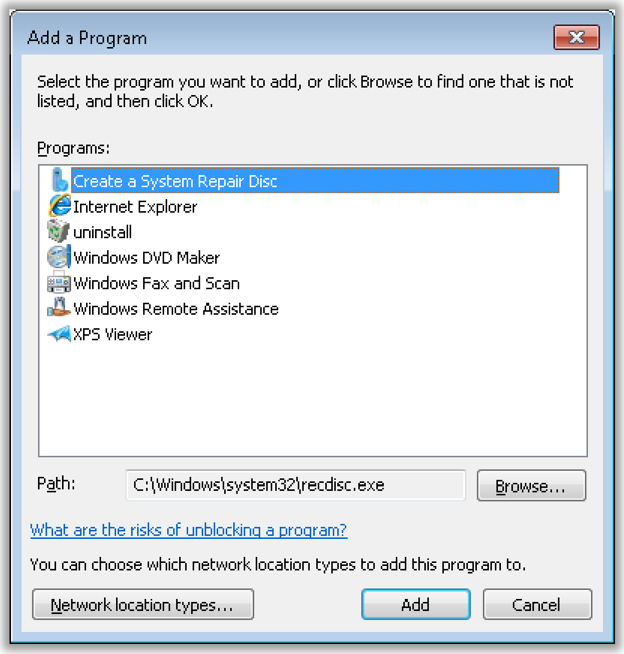
What are the risks of allowing a program to communicate?

Allowed programs and features:

Name	Home/Work (Private)	Public
□ BITS Peercaching	□	□
□ BranchCache - Content Retrieval (Uses HTTP)	□	□
□ BranchCache - Hosted Cache Client (Uses HTTPS)	□	□
□ BranchCache - Hosted Cache Server (Uses HTTPS)	□	□
□ BranchCache - Peer Discovery (Uses WSD)	□	□
☑ Client for NFS	☑	☑
☑ CodeMeter Runtime Server	□	☑
□ Connect to a Network Projector	□	□
☑ Core Networking	☑	☑
☑ Distributed scan client components	☑	☑
□ Distributed Transaction Coordinator	□	□
□ File and Printer Sharing	□	□

If you will have Mobile Access or anticipate the need to connect using the remote database spy or logwin, then proceed to Step 3, otherwise proceed to Step 5.

3. Click the button called **Allow another app…**



Select the program you want to add, or click Browse to find one that is not listed, and then click OK.

Programs:

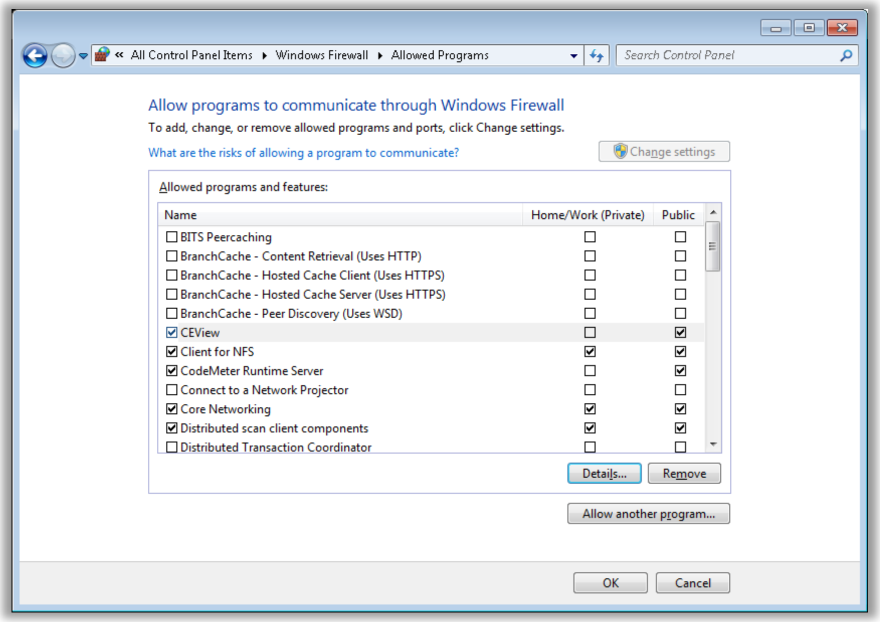
- Create a System Repair Disc
- Internet Explorer
- uninstall
- Windows DVD Maker
- Windows Fax and Scan
- Windows Remote Assistance
- XPS Viewer

Path: C:\Windows\system32\recdisc.exe

What are the risks of unblocking a program?

You can choose which network location types to add this program to.

4. Click on the **Browse** button, browse to the BLUE Open Studio runtime installation folder (e.g., C:\STUDIO\Bin), open CEView.exe, and click on the button called **Add**.



« All Control Panel Items ▸ Windows Firewall ▸ Allowed Programs
Search Control Panel

Allow programs to communicate through Windows Firewall

To add, change, or remove allowed programs and ports, click Change settings.

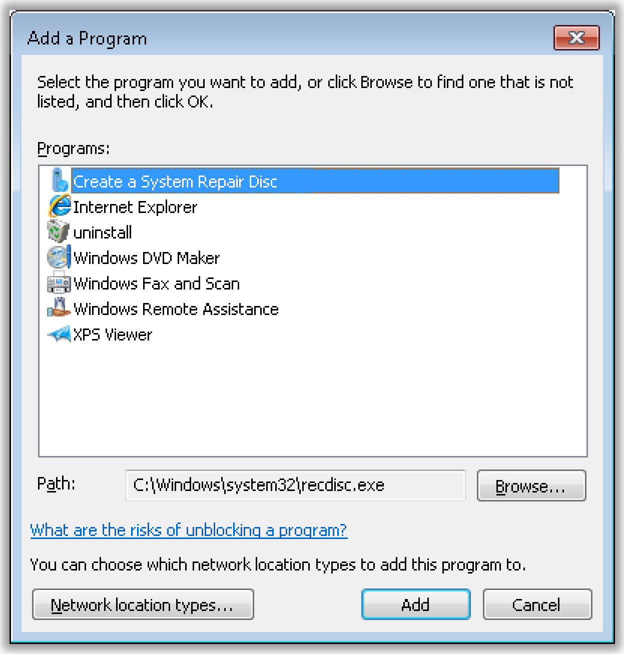
What are the risks of allowing a program to communicate?

Allowed programs and features:

Name	Home/Work (Private)	Public
☐ BITS Peercaching	☐	☐
☐ BranchCache - Content Retrieval (Uses HTTP)	☐	☐
☐ BranchCache - Hosted Cache Client (Uses HTTPS)	☐	☐
☐ BranchCache - Hosted Cache Server (Uses HTTPS)	☐	☐
☐ BranchCache - Peer Discovery (Uses WSD)	☐	☐
☑ CEView	☐	☑
☑ Client for NFS	☑	☑
☑ CodeMeter Runtime Server	☐	☑
☐ Connect to a Network Projector	☐	☐
☑ Core Networking	☑	☑
☑ Distributed scan client components	☑	☑
☐ Distributed Transaction Coordinator	☐	☐

Enable **Private and/or Public access** through the firewall as desired.

5. Click the button called **Allow another app…**



Add a Program

Select the program you want to add, or click Browse to find one that is not listed, and then click OK.

Programs:

- Create a System Repair Disc
- Internet Explorer
- uninstall
- Windows DVD Maker
- Windows Fax and Scan
- Windows Remote Assistance
- XPS Viewer

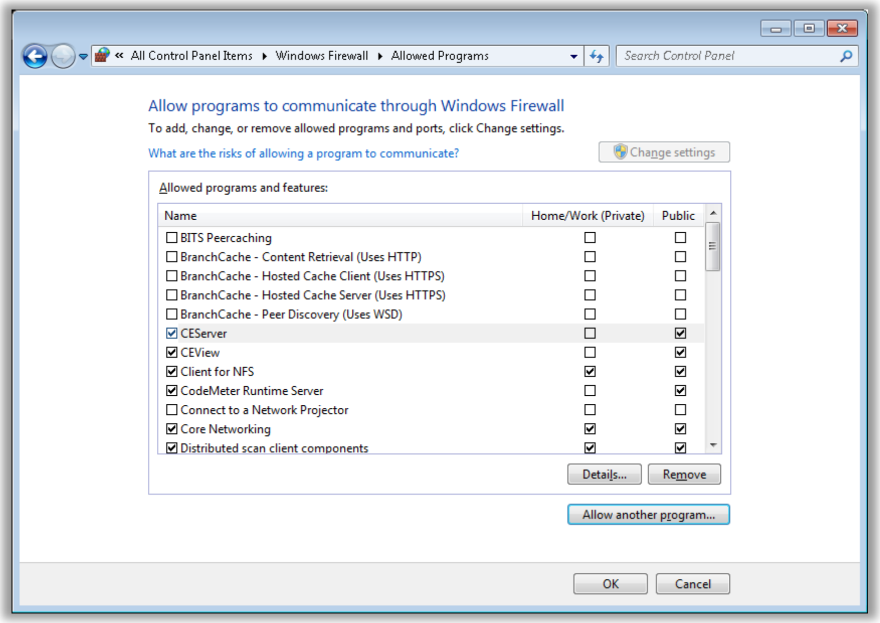
Path: C:\Windows\system32\recdisc.exe

Browse...
What are the risks of unblocking a program?

You can choose which network location types to add this program to.

Network location types...
Add
Cancel

6. Click on the **Browse** button, browse to the BLUE Open Studio installation folder (e.g., C:\STUDIO), open CEServer.exe, and click on the button called **Add**.



« All Control Panel Items ▸ Windows Firewall ▸ Allowed Programs
Search Control Panel

Allow programs to communicate through Windows Firewall

To add, change, or remove allowed programs and ports, click Change settings.

What are the risks of allowing a program to communicate?
Change settings

Allowed programs and features:

Name	Home/Work (Private)	Public
□ BITS Peercaching	□	□
□ BranchCache - Content Retrieval (Uses HTTP)	□	□
□ BranchCache - Hosted Cache Client (Uses HTTPS)	□	□
□ BranchCache - Hosted Cache Server (Uses HTTPS)	□	□
□ BranchCache - Peer Discovery (Uses WSD)	□	□
☑ CEServer	□	☑
☑ CEView	□	☑
☑ Client for NFS	☑	☑
☑ CodeMeter Runtime Server	□	☑
□ Connect to a Network Projector	□	□
☑ Core Networking	☑	☑
☑ Distributed scan client components	☑	☑

Enable Private and/or Public access through the firewall as desired

7. Click **OK** to close the Allowed apps dialog.

Appendix 3 - Run Studio Database Gateway as a Service

The Studio Database Gateway is a network-based application that interacts with databases and the Wonderware Historian using the Microsoft .NET Framework. If you are using the Gateway to communicate with the Wonderware Historian, then it should be run on the same computer that is running the Historian.

Copy the Studio Database Gateway to the Remote Computer

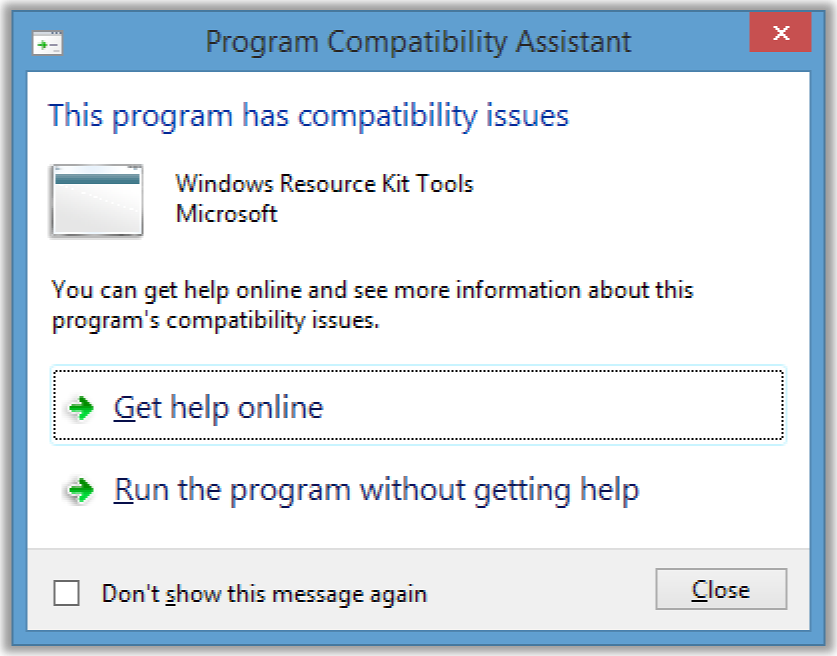
In order to run the Studio Database Gateway on a remote computer, you must copy the gateway executable and supporting files to the remote computer. Copy the files called StADOSvr.exe, StADOSvr.ini, StudioDataProtection.dll, and aahClientManaged.dll from the Bin folder of the BLUE Open Studio installation folder on your development computer (e.g., C:\Program Files (x86)\BLUE Open Studio v8.0\Bin or C:\Program Files\BLUE Open Studio v8.0\Bin) to the remote computer. You can put these files anywhere on the remote computer, but all of them must be in the same folder. Our Best Practice recommendation would be to create a folder in the Program Files folder called Studio Database Gateway (e.g., C:\Program Files (x86)\Studio Database Gateway or C:\Program Files\Studio Database Gateway).

You are now ready to configure the Studio Database Gateway to run as a Windows Service. These instructions use a program called SRVANY.EXE from the Windows Resource Kit Tools and the Windows Service Control program called SC.EXE. These instructions are for Windows 7 and higher and for Windows Server 2008 R2 and higher.

> Note: The Studio Database Gateway requires .NET Framework 4.0. Please ensure that it is installed prior to starting the gateway.

Install the Windows Resource Kit Tools

Download and install the Windows Resource Kit Tools from http://www.microsoft.com/en-us/download/details.aspx?id=17657. When you launch the installer, you may get a popup that looks similar to the following,



Program Compatibility Assistant

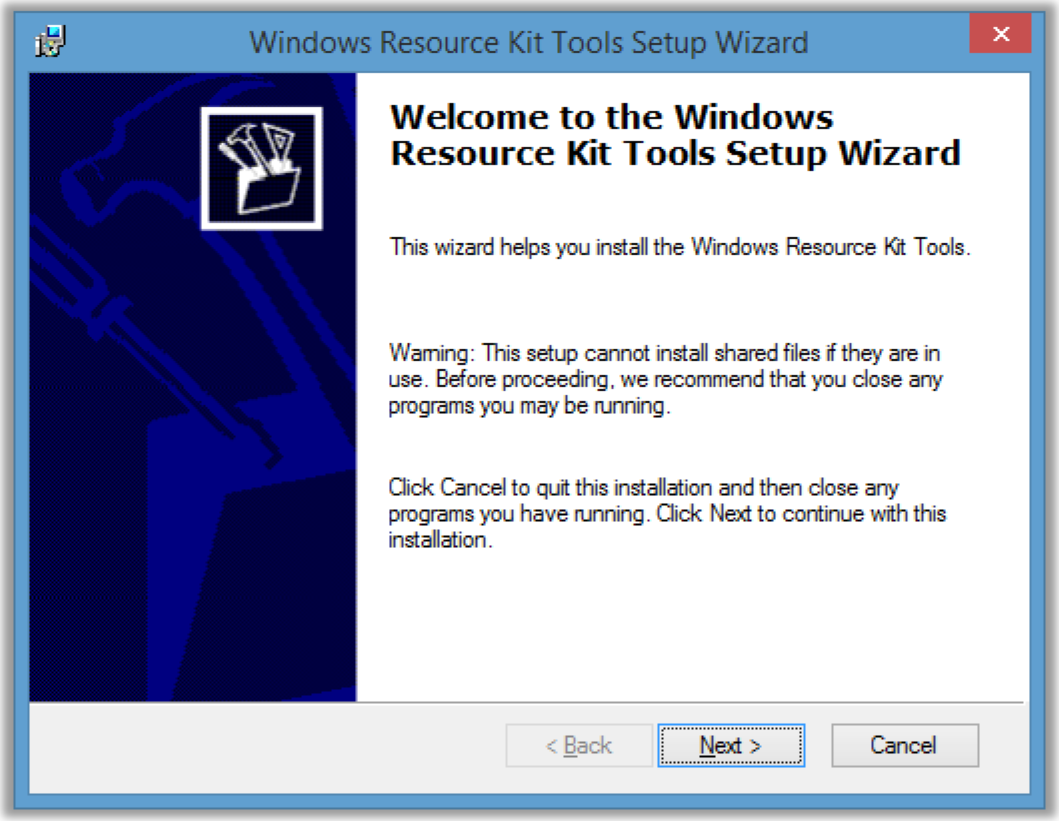
This program has compatibility issues

Windows Resource Kit Tools
Microsoft

You can get help online and see more information about this program's compatibility issues.

Get help online
Run the program without getting help
□ Don't show this message again
Close

If so, then click the link called Run the program without getting help. The installer should launch and you should see something similar to the following,



Windows Resource Kit Tools Setup Wizard

Welcome to the Windows Resource Kit Tools Setup Wizard

This wizard helps you install the Windows Resource Kit Tools.

Warning: This setup cannot install shared files if they are in use. Before proceeding, we recommend that you close any programs you may be running.

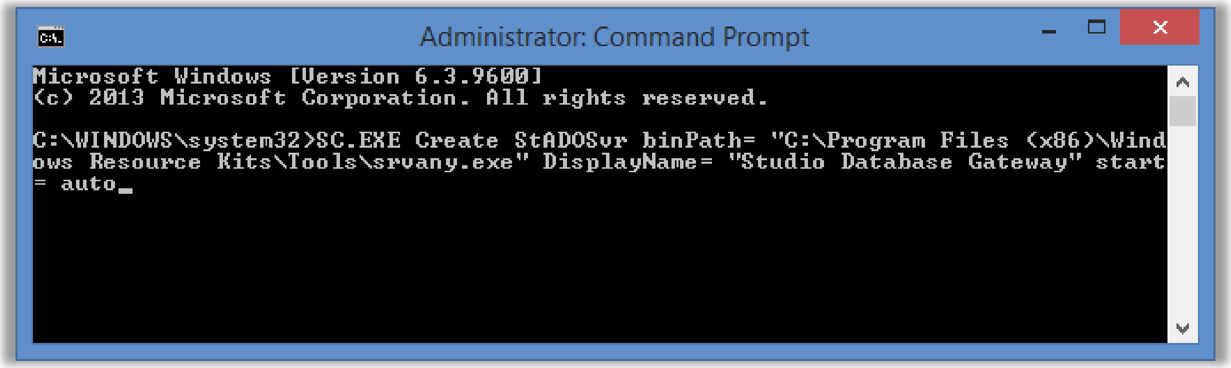
Click Cancel to quit this installation and then close any programs you have running. Click Next to continue with this installation.

Click the button labeled Next and follow the installation prompts to complete the installation. This will install the Windows Resource Kit Tools in your Program Files folder (e.g., C:\Program Files (x86)\Windows Resource Kits\Tools or C:\Program Files\Windows Resource Kits\Tools). The remainder of these instructions presume that that Windows Resource Kit Tools are installed in C:\Program Files (x86)\Windows Resource Kits\Tools. Please make the appropriate changes if they were installed in a different folder on your computer.

The next section will create a Windows Service using the Windows Resource Kit Tool called SRVANY.EXE to launch the Studio Database Gateway.

Create the Windows Service

Open a Windows Command Prompt with elevated permissions. Do this by right-clicking on the Command Prompt application in the Windows Start menu (or CMD.EXE) and selecting Run as administrator. Once the Command Prompt opens, run the SC.EXE command exactly as shown below,



```
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\WINDOWS\system32>SC.EXE Create StADOSvr binPath= "C:\Program Files (x86)\Wind
ows Resource Kits\Tools\srvany.exe" DisplayName= "Studio Database Gateway" start
= auto_
```

The command is one command that will wrap depending on the width of your Command Prompt window. Also, note that the spaces following binPath=, DisplayName=, and start= are required. For your convenience, you can copy/paste the following command to your Command Prompt window and run it.

```
SC.EXE Create StADOSvr binPath= "C:\Program Files (x86)\Windows Resource
Kits\Tools\srvany.exe" DisplayName= "Studio Database Gateway" start= auto
```

Remember to change the path if the Windows Resource Kit Tools are installed in a different location on your computer.

This creates a Windows Service called StADOSvr that launches SRVANY.EXE. Next, you need to configure the service to launch the Studio Database Gateway. This step requires that you edit the Windows Registry. To quote Microsoft,

WARNING: Using Registry Editor incorrectly can cause serious problems that may require you to reinstall your operating system. Microsoft cannot guarantee that problems resulting from the incorrect use of Registry Editor can be solved. Use Registry Editor at your own risk.

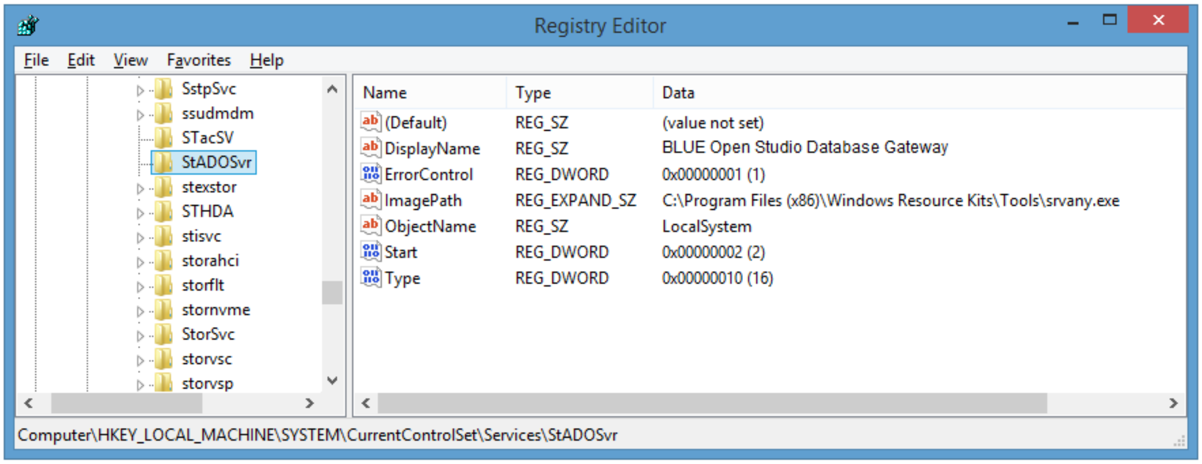
NOTE: You should back up the registry before you edit it.

Configure the Windows Service

Launch the Registry Editor (regedit.exe) and locate the following key:

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\StADOSvr
```

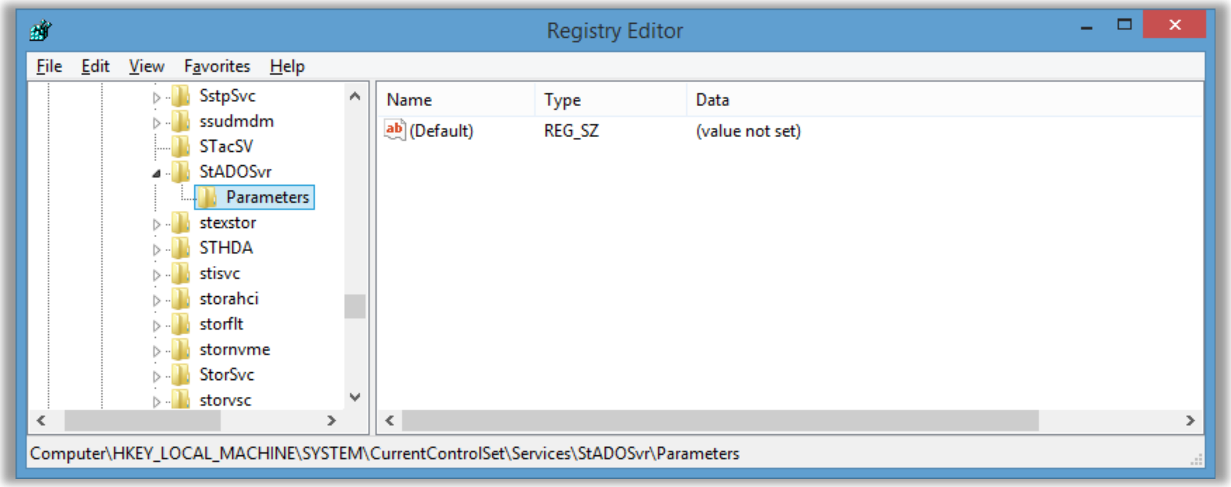
Click the key called StADOSvr, then you should see something similar to the following,



Name	Type	Data
(Default)	REG_SZ	(value not set)
DisplayName	REG_SZ	BLUE Open Studio Database Gateway
ErrorControl	REG_DWORD	0x00000001 (1)
ImagePath	REG_EXPAND_SZ	C:\Program Files (x86)\Windows Resource Kits\Tools\srvany.exe
ObjectName	REG_SZ	LocalSystem
Start	REG_DWORD	0x00000002 (2)
Type	REG_DWORD	0x00000010 (16)

Computer\HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\StADOSvr

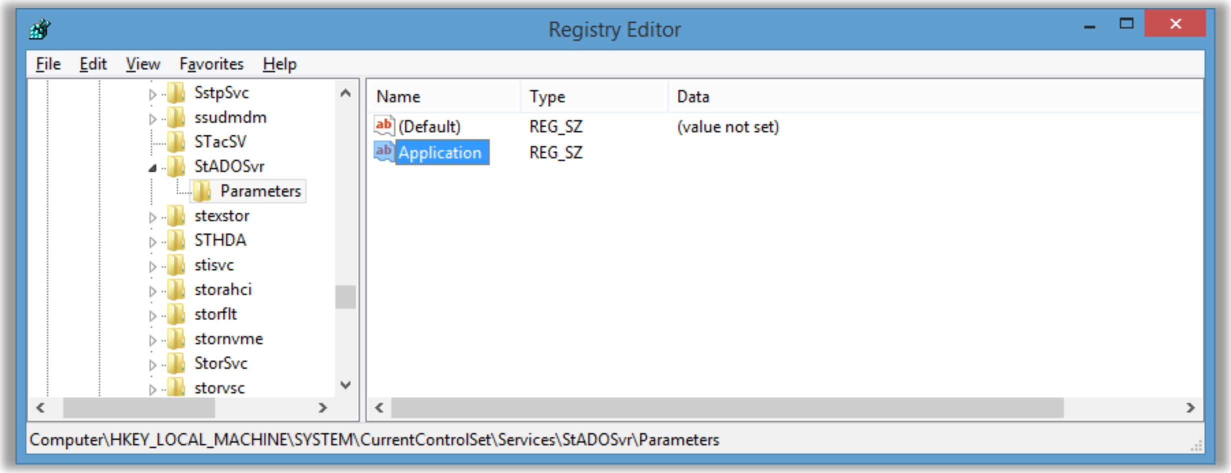
Right-click the key called StADOSvr, click New >> Key, and create a key called Parameters. You should see something similar to the following,



Name	Type	Data
(Default)	REG_SZ	(value not set)

Computer\HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\StADOSvr\Parameters

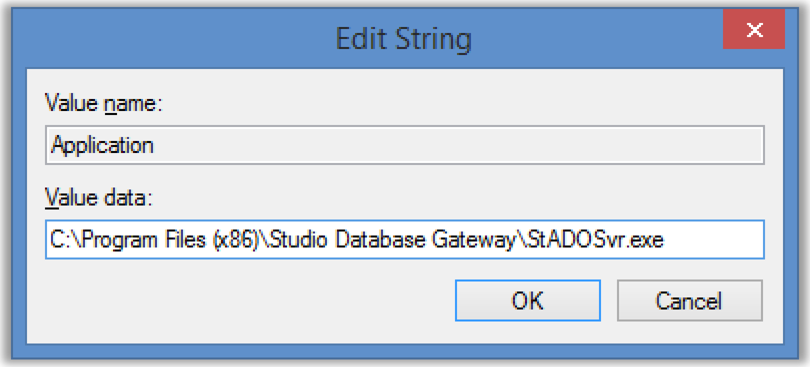
Right-click the new key called Parameters, click New >> String Value, and create a new string value called Application. You should see something similar to the following,



Name	Type	Data
(Default)	REG_SZ	(value not set)
Application	REG_SZ	

Computer\HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\StADOSvr\Parameters

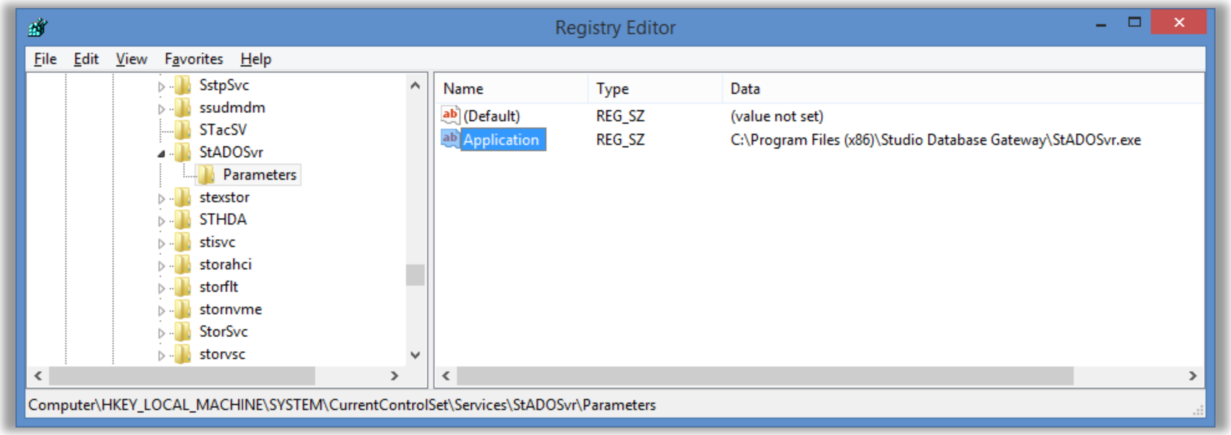
Right-click the new string value called Application and change the field called Value data to be the full path name of the Studio Database Gateway as shown below,



Value name: Application

Value data: C:\Program Files (x86)\Studio Database Gateway\StADOSvr.exe

If you installed the Studio Database Gateway files in a different location, then change the path as required. Click the button labeled OK and you should see something similar to the following,



Name	Type	Data
(Default)	REG_SZ	(value not set)
Application	REG_SZ	C:\Program Files (x86)\Studio Database Gateway\StADOSvr.exe

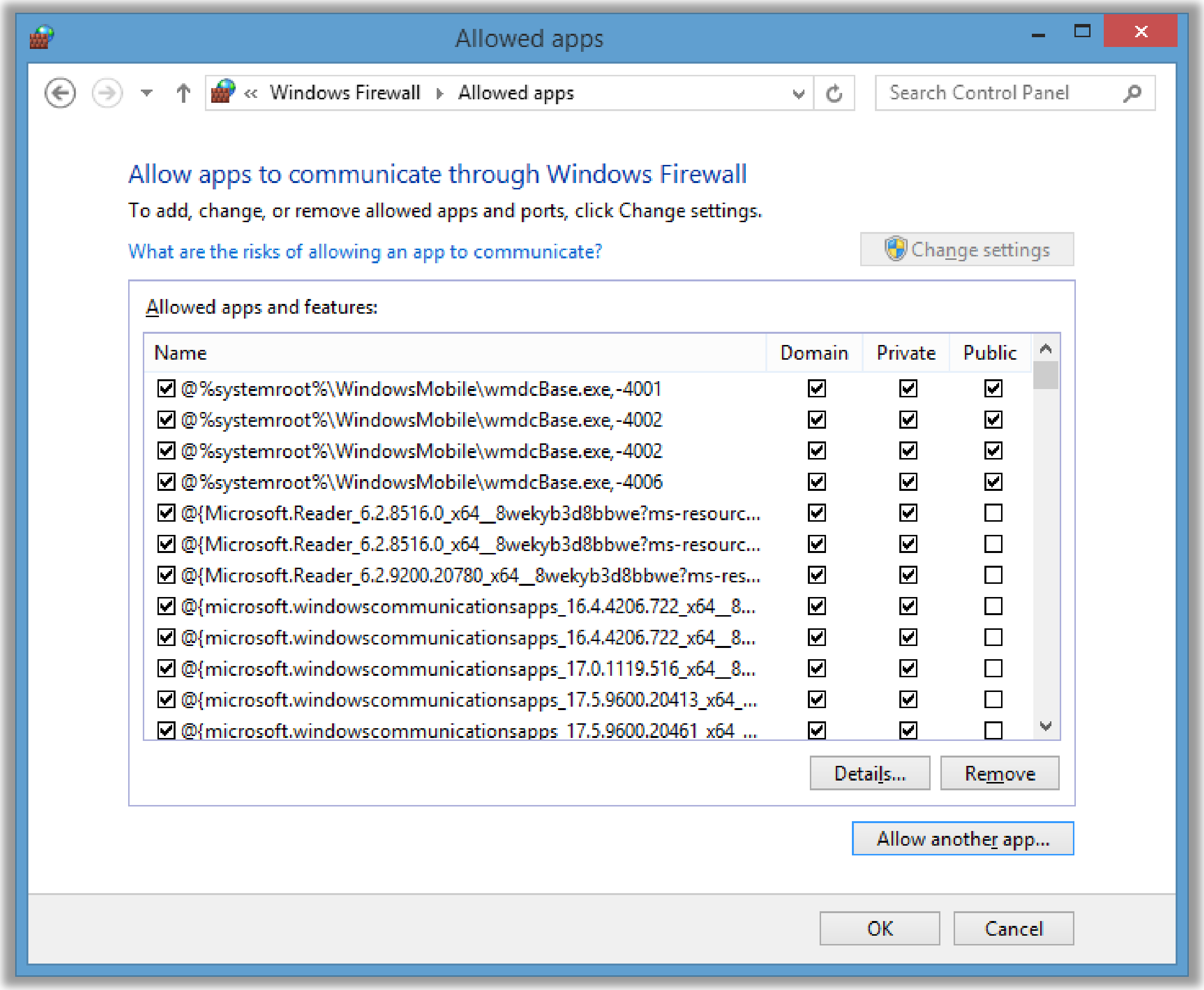
Computer\HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\StADOSvr\Parameters

Close the Registry Editor.

You should now be able to start the Studio Database Gateway service like you would other Windows Services. The service is also configured to automatically start whenever the computer starts. The final step is to configure the Windows Firewall to allow remote access to the gateway.

Configure the Windows Firewall

Open the Windows Firewall from the Windows Control Panel and click the link called Allow an app or feature through the Windows Firewall. You should see something similar to the following,



Allow apps to communicate through Windows Firewall

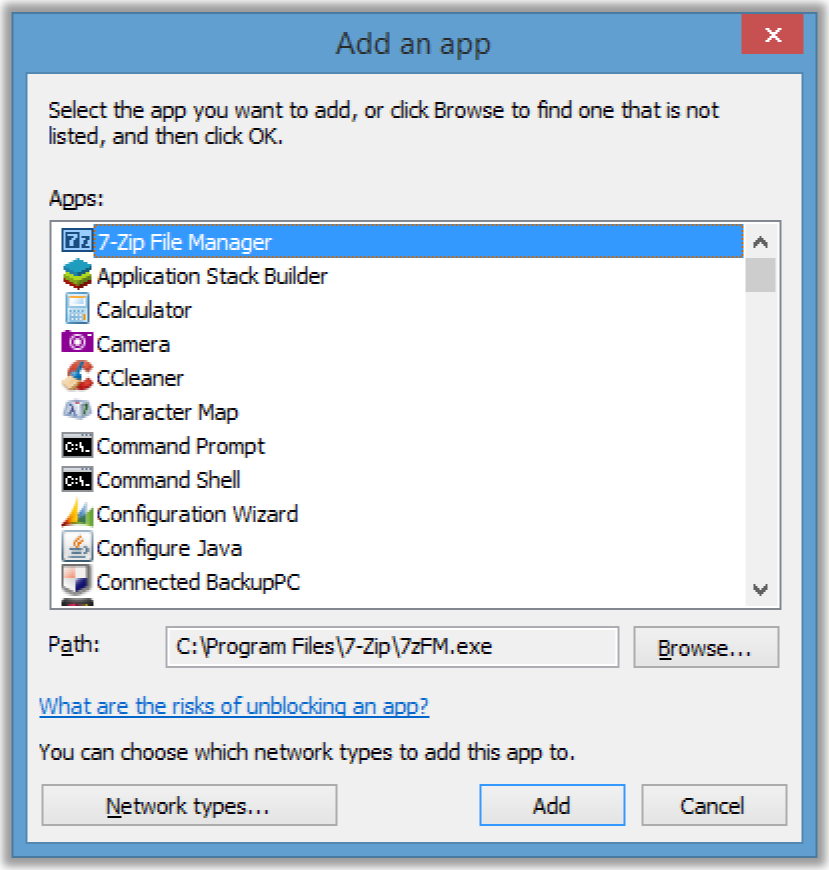
To add, change, or remove allowed apps and ports, click Change settings.

What are the risks of allowing an app to communicate?

Allowed apps and features:

Name	Domain	Private	Public
☑ @%systemroot%\WindowsMobile\wmdcBase.exe,-4001	☑	☑	☑
☑ @%systemroot%\WindowsMobile\wmdcBase.exe,-4002	☑	☑	☑
☑ @%systemroot%\WindowsMobile\wmdcBase.exe,-4002	☑	☑	☑
☑ @%systemroot%\WindowsMobile\wmdcBase.exe,-4006	☑	☑	☑
☑ @{Microsoft.Reader_6.2.8516.0_x64__8wekyb3d8bbwe?ms-resourc...	☑	☑	□
☑ @{Microsoft.Reader_6.2.8516.0_x64__8wekyb3d8bbwe?ms-resourc...	☑	☑	□
☑ @{Microsoft.Reader_6.2.9200.20780_x64__8wekyb3d8bbwe?ms-res...	☑	☑	□
☑ @{microsoft.windowscommunicationsapps_16.4.4206.722_x64__8...	☑	☑	□
☑ @{microsoft.windowscommunicationsapps_16.4.4206.722_x64__8...	☑	☑	□
☑ @{microsoft.windowscommunicationsapps_17.0.1119.516_x64__8...	☑	☑	□
☑ @{microsoft.windowscommunicationsapps_17.5.9600.20413_x64_...	☑	☑	□
☑ @{microsoft.windowscommunicationsapps_17.5.9600.20461_x64_...	☑	☑	□

Click the button labeled Allow another app. If the button is not selectable, then you may need to click the button called Change Settings to enable the ability to allow an application through the firewall. You should see something similar to the following,



Select the app you want to add, or click Browse to find one that is not listed, and then click OK.

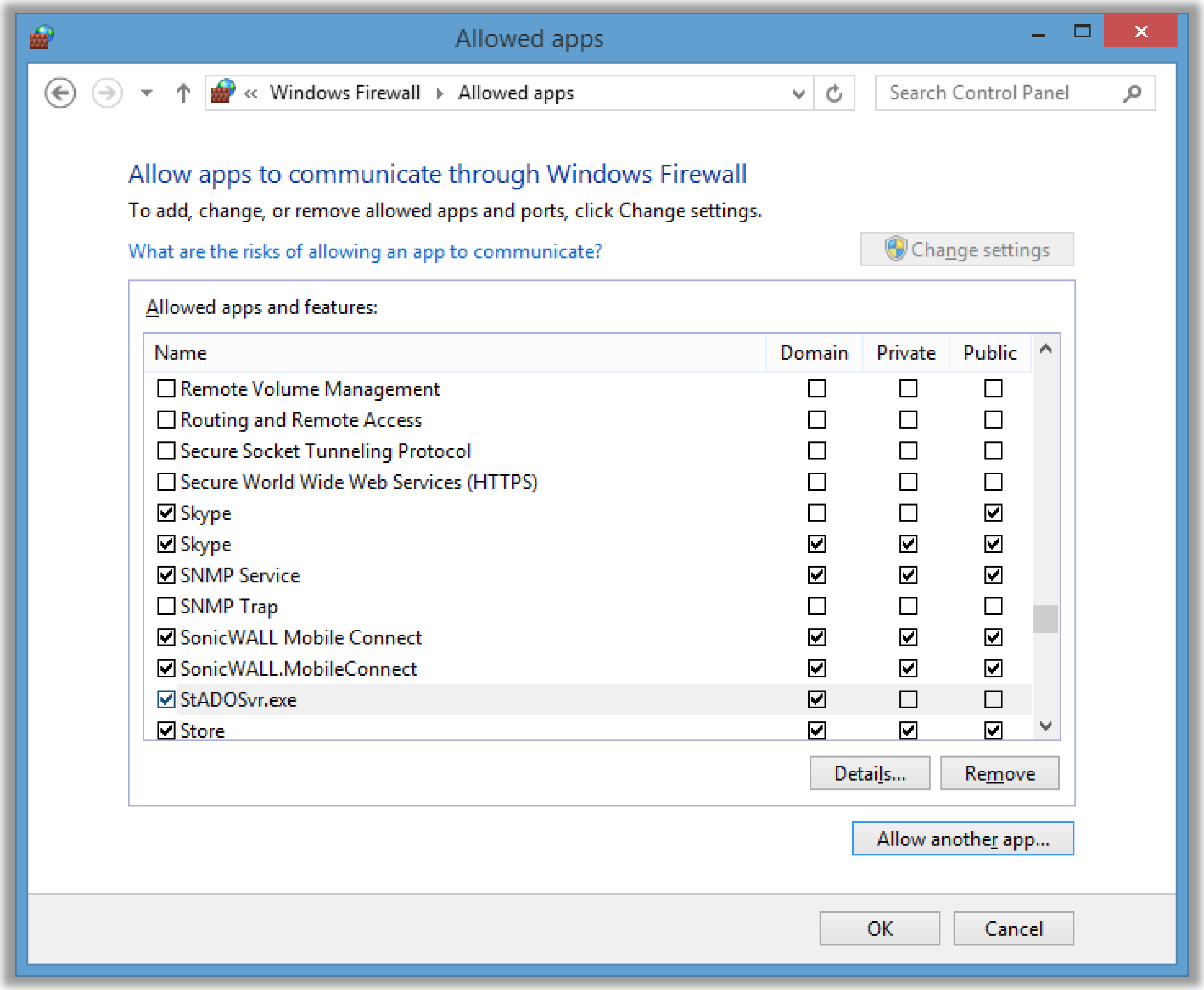
Apps:

- 7-Zip File Manager
- Application Stack Builder
- Calculator
- Camera
- CCleaner
- Character Map
- Command Prompt
- Command Shell
- Configuration Wizard
- Configure Java
- Connected BackupPC

Path: C:\Program Files\7-Zip\7zFM.exe

You can choose which network types to add this app to.

Click the button labeled Browse, browse for the file called StADOSvr.exe and click the button labeled Add. You should see something similar to the following



Allow apps to communicate through Windows Firewall

To add, change, or remove allowed apps and ports, click Change settings.

What are the risks of allowing an app to communicate?

Allowed apps and features:

Name	Domain	Private	Public
□ Remote Volume Management	□	□	□
□ Routing and Remote Access	□	□	□
□ Secure Socket Tunneling Protocol	□	□	□
□ Secure World Wide Web Services (HTTPS)	□	□	□
☑ Skype	□	□	☑
☑ Skype	☑	☑	☑
☑ SNMP Service	☑	☑	☑
□ SNMP Trap	□	□	□
☑ SonicWALL Mobile Connect	☑	☑	☑
☑ SonicWALL.MobileConnect	☑	☑	☑
☑ StADOSvr.exe	☑	□	□
☑ Store	☑	☑	☑

Tic the checkboxes that apply to the networks for which you want to grant access. Click OK when you are done.