EcoStruxure™ Machine SCADA Expert Quick Start Guide

12/2019

EIO0000002447_06

www.schneider-electric.com

Schneider Electric™

Contents

INTRODUCTION..... 4

Conventions used in this documentation.....5
About this software.....7
About the EcoStruxure Machine SCADA Expert software components.....10
Install the full EcoStruxure Machine SCADA Expert software..... 11
Execution Modes..... 16

THE DEVELOPMENT ENVIRONMENT..... 18

Title Bar..... 19
Status Bar.....20
Application button.....21
Quick Access Toolbar.....22
Ribbon.....24
Home tab.....24
View tab.....24
Insert tab.....25
Project tab..... 26
Graphics tab..... 26
Format tab..... 27
Help tab..... 28
Project Explorer..... 29
Global tab..... 29
Graphics tab..... 30
Tasks tab..... 32
Comm tab..... 33
Screen/Worksheet Editor.....35

ABOUT TAGS AND THE PROJECT DATABASE.....36

Naming the Tag..... 38
Choosing the Tag Data Type..... 39
Using Array Tags.....40
About indirect tags..... 43

TUTORIAL: BUILDING A SIMPLE PROJECT..... 44

Creating a new project.....45
Specifying the startup screen.....47
Creating tags..... 48
Creating the main screen.....50
Drawing the main screen's title..... 52
Drawing a button to open another screen.....54
Saving and closing the main screen..... 56

Creating the synoptic screen..... 57
Drawing the synoptic screen's title..... 57
Drawing "Date" and "Time" displays..... 57
Placing an "Exit" icon..... 59
Testing the project..... 60
Placing an animated tank..... 61
Placing a level slider..... 63
Drawing a tank selector..... 64
Testing the project..... 65
Configuring the communication driver..... 67
Monitoring device I/O during runtime..... 70

Introduction

EcoStruxure Machine SCADA Expert is a powerful, integrated tool that exploits key features of Microsoft operating systems and enables you to build full-featured SCADA (Supervisory Control and Data Acquisition) or HMI (Human-Machine Interface) programs for your industrial automation business.

This *EcoStruxure Machine SCADA Expert Quick Start Guide* is intended for individuals using EcoStruxure Machine SCADA Expert for the first time. This publication will help you quickly familiarize yourself with the basic functions of EcoStruxure Machine SCADA Expert.

Conventions used in this documentation

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

Text conventions

This documentation 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 you must enter are indicated using **`monospace`** text (for example, **`D:\Setup.exe`**).
- Buttons, menu options, and keyboard keys are indicated using a **bold** typeface (for example, **File** menu).

In addition, this documentation segregates some text into **Tip**, **Note**, and **Caution** boxes:

- **Tips** provide useful information to save development time or to improve the project performance.
- **Notes** provide extra information that may make it easier to understand the nearby text, usually the text just before the note.
- **Cautions** provide information necessary to prevent errors that can cause problems when running the project, and may result in damage.

Mouse and selection conventions

Because most PCs used for project development run a version of Microsoft Windows with a mouse, this documentation 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 documentation 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.

Windows conventions

This documentation 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.

About this software

EcoStruxure Machine SCADA Expert is powerful software for developing HMI, SCADA, and OEE/Dashboard projects that can be deployed anywhere.

Each EcoStruxure Machine SCADA Expert project consists of:

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

After you develop your project, you can either run it locally on your development workstation or download it to a remote computer and run it there. The project runtime server processes I/O data from connected devices according to your project parameters and then reacts to, displays, and/or saves the data.

Product features

ActiveX and .NET

Use third-party controls to enhance your project. This software is a container for ActiveX and .NET controls. Add functionality such as browsers, media players, charting, and other tools that support the ActiveX and .NET interface standards.

Alarms

In addition to all of the alarm functions you would expect, this software also sends alarms using multi-media formats like PDF. Use remote notification to have alarms sent right to your email inbox, a printer, or a smartphone! Alarms are real-time and historical, log data in binary format or to any database.

Animation

This software gives you great command over graphics. Paste images, and even rotate them dynamically. Fill bar graphs with color, or adjust the scale of objects with easy-to-use configuration. Other animations include "command" (for touch, keyboard and mouse interaction), hyperlink, text data link, color, resize (independent height and width), position, and rotation (with custom rotation point).

Database

Connect to SQL databases (MS SQL, MySQL, Sybase, Oracle), MS Access and Excel, and ERP/MES systems (including SAP), even from Windows Embedded. Flexible enough to have a built-in interface without the need to know SQL (for trends, alarms/events, grid and other objects), or use any SQL statement you need anywhere you need it.

Drivers

This software includes over 240 built-in communication drivers for most PLCs, temperature controllers, motion controllers, barcode/RFID readers, and other devices. Use these built-in drivers without the need for OPC servers (but are an optional connection method).

Email

Send email via SMTP to any desktop or mobile device. Get real-time information on alarms, process values, and other events. This software supports SSL encryption allowing the use of third-party providers such as Gmail.

Events

This software offers traceability for operator initiated actions or internal system activity. Log events such as security system changes (user logon or off), screen open/close, recipe/report

operations, custom messages and system warnings. Also any tag value changes including custom messages.

FDA Traceability

Take advantage of built-in traceability and e-signature features to create projects that fully comply with U.S. Food and Drug Administration regulations (Title 21 CFR Part 11). These features are often used in food and pharmaceutical applications, but they can be used in any application where traceability is required.

FTP

Automatically upload or download files during run time to/from remote storage locations using FTP and flexible scripting functions. Configure FTP via scripting or the included configuration interface.

Graphics and Design Tools

Create powerful screens to meet any application need using the improved tools in our graphic interface. Combine built-in objects to create any functionality required. Store graphics in the symbol library for future use. Easily make projects across a product line share a consistent "look and feel".

Historical Performance

We have optimized the trend history module and designed it to load millions of values from SQL relational databases with high performance, with built-in data decimation in the Trend Control. Easy-to-use tools provide quick access to Statistical Process Control (SPC) values without any need for programming.

Intellectual Property Protection

Screens, documents, scripts and even math worksheets can be individually password protected. This prevents unauthorized viewing or editing of your corporate custom functionality. Protect the entire project with just a few mouse clicks.

Multi-Language

Develop your project in one of many development languages, including English, Portuguese, German, and French.

OPC

As an alternative to the built-in drivers for direct communication with PLCs, you can also use any of several different versions of OLE for Process Control (OPC) to manage your devices. This software includes support for "classic" OPC DA (client or server), OPC UA (client or server), OPC XML-DA (client only), and OPC HDA (server only).

PDF Export

Send Alarms, Reports, or any file (including .doc or .txt) to a production supervisor, quality manager, or maintenance staff using the included PDF writer.

Recipes

Save time and maintain consistency by automating part parameters or productions quantities with any triggering event.

Redundancy

For critical applications where data is vital, this software supports web server, database and overall system redundancy.

Reports

Create clear, concise reports in text format, graphical RTF, XML, PDF, HTML, and CSV, or integrate with Microsoft Office. Get the data you need, in the format you need it, to make informed decisions, fast.

Scalable

Develop once and deploy anywhere, on any currently supported version of Microsoft Windows.

Scheduler

Schedule custom tag changes on date/time, frequency, or any trigger. Use this for simulation, to trigger reports or other functionality at a particular time of day, or even to trigger driver worksheets to read/write at a scan rate you choose.

Scripting

Two powerful scripting languages are supported. Use built-in functions or use standard VBScript to take advantage of widely available resources. Both can be used simultaneously to give you the functionality you need.

Security

This software provides support for group and user accounts, e-signatures, and traceability, as well as support for Lightweight Directory Access Protocol (LDAP). Integrate your project with your Active Directory, including Active Directory Application Mode (ADAM).

SSL Support for Emails

Native support for Secure Socket Layer (SSL), which makes it easy and secure to send emails from this software using third-party tools such as Gmail!

Standards

Take advantage of common industry standards to develop projects that are compatible with any format. TCP/IP, ActiveX/.NET, OPC (client and server), COM/DCOM, OLE, XML, SOAP, and HTML are all supported.

SNMP

Easily configure managed networked devices on IP networks (such as switches and routers) using incorporated SNMP configuration commands and an easy-to-use configuration interface.

Symbols

An extensive library of pre-made symbols features push buttons, pilot lights, tanks, sliders, meters, motors, pipes, valves and other common objects. Use the included symbols in your project, modify existing symbols to suit your needs, or create your own from scratch. Plus support for third-party symbol libraries and graphic tools.

Tags Database

This software features an object-oriented tags database with boolean, integer, real, strings, arrays, classes (structures), indirect tags, and included system tags.

Thin Clients

Remotely view project screens on several different types of thin clients. Use the standalone Secure Viewer to acheive the greatest security on plant-floor stations. Or use the HTML5-enabled Mobile Access to access your project from almost any other browser or mobile device.

Trends

Real-time and Historical trends are supported. Log data in binary format or to any database locally and remotely. Color or fill trends with graphic elements to enhance clarity of data. Date/Time based or numeric (X/Y plot) trends give you the flexibility to display information that best suits your project.

Troubleshooting

Quickly debug and verify a project using local and remote tools for troubleshooting, including status fields, Database Spy and LogWin. Capture screen open and close times, see communications in real-time, and messages related to OPC, recipes/reports, security, database errors and even custom messages. Quickly get your project finished using these powerful tools.

About the EcoStruxure Machine SCADA Expert software components

The EcoStruxure Machine SCADA Expert software suite comprises several individual components that can be installed on different platforms to perform different functions. The architecture of your finished EcoStruxure Machine SCADA Expert project depends on which components you install, where you install them, and how you connect them to each other.

The following table lists all of the available components.

Component	Features	Platforms
EcoStruxure Machine SCADA Expert	• Project development environment • Tag integration • Remote management of project runtimes • Project runtime • Agent to allow remote management • Project viewer	• Windows • Windows Server • Windows Embedded Standard
Studio Database Gateway (StADOSvr)	Enables communication between the project runtime and external databases, including Historian and most ADO.NET-compatible databases.	• Windows • Windows Server
Mobile Access Runtime	Enables the project runtime to serve HTML5-enhanced project screens to web browsers and mobile devices.	• Internet Information Services (IIS) for Windows • any CGI-enabled web server (e.g., Apache)
Secure Viewer	Project viewer, as a standalone program. (See note below.)	• Windows • Windows Server • Windows Embedded Standard

It is important to distinguish between the project development environment and the project runtime. You can use the project development environment to design, develop, troubleshoot, deploy, and monitor projects. In contrast, the project runtime actually runs your project, communicates with external databases and devices, and serves screens to project viewers.

The full EcoStruxure Machine SCADA Expert software for Windows includes both the project development environment and the project runtime. Your software license determines which parts of the software you can use on any given computer or device. For more information, see Execution Modes on page 16.

In most cases, the first thing you should do is install the full EcoStruxure Machine SCADA Expert software on your primary workstation, because it not only sets up the project development environment for you, it also unpacks the rest of the components so that they can be installed on other computers and devices.

> **Note:** We recommend that you use Mobile Access instead of Secure Viewer or Web Thin Client whenever possible. Secure Viewer and Web Thin Client depend on legacy, Windows-only technologies, while Mobile Access allows you to use any HTML5-compatible browser running on any platform as a project viewer. Mobile Access does not yet support all of the features that Secure Viewer and Web Thin Client do, but we are continuing to improve Mobile Access with every new release. For more information, see Mobile Access.

Install the full EcoStruxure Machine SCADA Expert software

Install the full EcoStruxure Machine SCADA Expert software on your Windows computer in order to develop EcoStruxure Machine SCADA Expert projects, or to use the computer as a project runtime server and/or thin client.

To install and run the full EcoStruxure Machine SCADA Expert software, you must have the following:

- A Windows-compatible computer with a standard keyboard, a pointer input (i.e., a mouse, trackpad, or touchscreen), and an SVGA-minimum display;
- One of the following Windows operating systems:
  - Windows:
    - Windows 10 (including LTSC/LTSB versions)
    - Windows 8.1
    - Windows 7 Service Pack 1
  - Windows Server:
    - Windows Server 2016
    - Windows Server 2012 R2
    - Windows Server 2008 R2 Service Pack 1
  - Windows Embedded Standard:
    - Windows Embedded Standard 7 Service Pack 1
- .NET Framework 3.5 and .NET Framework 4.7.2 (see note below);
- Internet Explorer 11 (not Microsoft Edge);
- Minimum 2 GB available storage (hard drive or non-volatile);
- Minimum 1 GB available memory (RAM); and
- An Ethernet or Wi-Fi network adapter.

We recommend the "Pro" and "Enterprise" editions of Windows, because they include Internet Information Services (IIS) as a pre-installed feature that can be turned on. You can use IIS to make your projects accessible to thin clients and mobile devices. We do not recommend the "Home" and "Education" editions of Windows, because they hide or disable many important features.

Only Windows 10 and Windows Server 2016 are under what Microsoft calls "mainstream support", which means they are actively maintained and additional service packs might be released for them in the future. Windows 8.1, Windows 7, Windows Server 2012 R2, and Windows Server 2008 R2 are under what Microsoft calls "extended support", which means they are no longer actively maintained. For more information, go to: https://support.microsoft.com/en-us/help/13853/windows-lifecycle-fact-sheet

Regardless of which version or edition of Windows you are using, you should make sure it is fully updated before you install EcoStruxure Machine SCADA Expert. Updating Windows ensures that it has all of the latest security fixes and system components.

The operating system, storage, and memory requirements will necessarily increase for larger projects; the minimum requirements listed above are only for projects of up to 4,000 tags. The following table shows the complete requirements:

Project Size	Operating System	Storage	Memory
up to 4,000 tags	Windows, Windows Server, Windows Embedded Standard	2 GB available	1 GB available
up to 64,000 tags	Windows, Windows Server	4 GB available	2 GB available

Your computer needs to meet only the minimum requirements when you first install the software and begin to develop your project, but the requirements will increase as your project grows. Furthermore, every computer or device that you plan to use as a runtime station must meet the same requirements.

The following items are optional but recommended:

- A USB port or memory card slot, to be used for hardkey licensing of the software.

  This is optional because softkey licensing is also available.

- Serial COM ports and adapters, to be used for direct communication with PLCs and other devices.

  This is optional because many newer device protocols use Ethernet communication (i.e., TCP/IP or UDP/IP) instead of serial communication.

- Internet Information Services (IIS) installed and turned on; for more information, see the description of the **Mobile Access Runtime** option below.

  This is optional because you may choose not to install the Mobile Access Runtime feature at this time, as part of the full EcoStruxure Machine SCADA Expert software. You can install it at a later time, for either IIS or CGI.

- Visual Studio 2010 or Team Explorer 2010 installed on the same computer; for more information, see the description of the **Collaboration** option below.

  This is optional because you may choose not to install the Collaboration feature.

Finally, you must have Administrator privileges on the computer in order to install any software.

> **Note:**
>
> You must have both .NET Framework 3.5 and .NET Framework 4.7.2 installed and turned on in order to use all of the features of EcoStruxure Machine SCADA Expert.
>
> If Windows is fully updated, as described above, then the latest versions of .NET Framework should be installed. They might not be turned on, however, so use the *Windows Features* control panel to confirm that they are.
>
> In some cases, it might not be possible to keep Windows fully updated through normal means. (For example, if your computer is on a private or secure network, then it might not be able to contact the Windows Update service.) The EcoStruxure Machine SCADA Expert software installer will check for both versions of .NET Framework, and if it does not find them, it will attempt to install them for you. Depending on your computer's security settings, however, the installation(s) might fail without notice. Therefore, if you experience issues later while trying to use EcoStruxure Machine SCADA Expert — in particular, while trying to communicate with external databases — use the **Windows Features** control panel to confirm that both versions of .NET Framework are installed and turned on.
>
> In Windows Server 2012 R2, .NET Framework 3.5 is not pre-installed and it cannot be installed by the EcoStruxure Machine SCADA Expert software installer. You must use the Server Manager tool to install it. For more information, see Install .NET Framework 3.5 in Windows Server 2012 R2.
>
> For more information about .NET Framework, go to: https://docs.microsoft.com/en-us/dotnet/framework/index

To install the full EcoStruxure Machine SCADA Expert software:

1. Close all other running programs, if possible.

   We recommend you do this because those programs can use a significant amount of system resources and therefore cause this installation to take longer to finish. Windows services (e.g., Windows Defender, Windows Update) can have the same effect, but we do not recommend you stop or disable those services.

2. Do one of the following:

   - Download the zipped installer to your computer, either from our website (www.schneider-electric.com) or from another location on your network where you have previously saved it. Extract the files, open the resulting folder, and then locate and run the setup program (`setup.exe`).

   The installation wizard runs and asks you to select a language for the installation.

3. Select a language from the list, and then click **OK**.

   This selection determines the language of the user interface for both the installation wizard and the project development environment. You can change the language for the project development environment later, after the software has been installed.

   The wizard prepares for installation. During this step, it automatically installs SafeNet's Sentinel drivers (a part of the software licensing mechanism), .NET Framework 3.5, and .NET Framework 4.7.2, if necessary.

4. On the **Welcome** page of the wizard, click **Next** to proceed with the installation.
5. On the **License Agreement** page, click **Yes** to accept the agreement and proceed, or click **No** to refuse the agreement and exit the wizard.
6. On the **Customer Information** page, type your user name and company name, and then click **Next**.
7. On the **Choose Destination Location** page, select the folder where the software should be installed, and then click **Next**.

   By default, the software will be installed at the following location:

   ```
   C:\Program Files (x86)\Schneider Electric\EcoStruxure Machine SCADA Expert v8.1\
   ```

8. On the **Select Features** page, select the specific features and components that you want to install, and then click **Next**.

Feature	Description
Program Files	The main program files for the project development environment, the project runtime, and the project viewer. This feature cannot be deselected.
Demo Projects	Premade projects that demonstrate the capabilities of the EcoStruxure Machine SCADA Expert software.
Hardkey Support	Additional drivers to support the use of hardkey licenses.
OPC Components	Additional components required for communication with other OPC-compatible devices. This includes OPC DA (a.k.a. OPC Classic), OPC XML-DA, and OPC UA.
PDF Printing	Additional software that allows run-time reports to be saved as PDF files.
Security System Device Driver	An additional keyboard driver that enforces project security during project run time by controlling user input.
Symbol Library	A library of premade but configurable screen objects such as pushbuttons, toggle switches, gauges, dials, indicator lights, and so on.
BDE for PanelMate™ Import Wizard	Borland Database Engine (BDE), which is required to import a PanelMate Plus or PanelMate Power Pro program into a new project.
Mobile Access Runtime	Add-on software for web servers that allows mobile devices (i.e., smartphones and tablets) to access your projects. This software is available for both Internet Information Services (IIS) and other, CGI-based web servers like Apache. You do not need to install this feature at this time. Regardless of whether you select this installation option, a separate Mobile Access Runtime software installer (`MobileAccessSetup.exe`) will be unpacked with the rest of the EcoStruxure Machine SCADA Expert software, and you can choose to run that installer at a later time. To use this feature, your software license must include the Mobile Access Runtime add-on. To purchase this add-on, contact your EcoStruxure Machine SCADA Expert software distributor.

Feature	Description
Collaboration Tools	Additional tools for workgroup collaboration and source control within the project development environment. This feature requires that you have Visual Studio Team Explorer 2010 installed on the same computer. The EcoStruxure Machine SCADA Expert software installer will attempt to verify that you do, and if you do not, it will not install this feature. To download Team Explorer 2010 for free, go to: http://www.microsoft.com/en-us/download/details.aspx?id=329 The full version of Visual Studio 2010 includes the Team Explorer module. **Note:** Team Explorer 2012, Team Explorer 2013, and Team Explorer 2017 are not supported as collaboration clients at this time. You should also have Visual Studio Team Foundation Server 2010 or 2012 running somewhere on your network, but if you do not, it will not prevent you from installing the Collaboration feature at this time. **Note:** Team Foundation Server 2013 and Team Foundation Server 2017 are not supported as collaboration servers at this time. To use this feature, your software license must include the Collaboration add-on. To purchase this add-on, contact your EcoStruxure Machine SCADA Expert software distributor.
Wonderware Historian	Additional software that is required to save historical data (e.g., from Trend worksheets) to Wonderware Historian databases and AVEVA Insight. To use this feature, you must have .NET Framework 4.7.2 installed and turned on.
Custom Widget Framework	Additional software that is required to develop HTML5-based widgets and then use them in project screens.
OPC Factory Server	A standalone OPC server that your projects can use to communicate with Schneider Electric PLCs (e.g., M340, M580, Quantum, TSX Compact, TSX Micro, TSX Momentum, TSX Series 7, TSX S1000, TSX/PCX Premium, Unity Momentum). For more information, see the documentation that is installed with OPC Factory Server.

9. On the **Ready To Install** page, click **Install**.

Note: You might receive the following error message during installation: "Error 1628: Failed to complete script based install." For more information about this error and how to resolve it, go to: https://flexeracommunity.force.com/customer/articles/en_US/ERRDOC/Error-1628-Failed-To-Complete-Script-Based-Install

> **Note:** If you try to install an earlier version of the EcoStruxure Machine SCADA Expert software on a computer that already has a later version installed, you might receive the following message during installation: "Version x.x.x.x of CodeMeter Development Kit is already installed. Downgrading to Version x.x.x.x is not possible, installation will be aborted." CodeMeter is supplemental software used by EcoStruxure Machine SCADA Expert to manage hardkey licenses. To resolve this issue, you must use Task Manager in Windows to stop CodeMeter Runtime Server (`CodeMeter.exe`) before you install the earlier version of EcoStruxure Machine SCADA Expert.

The software is installed, and then when the installation is finished, the last page of the wizard is displayed.

10. Click **Finish** to close the installation wizard.

When you have finished the installation, you can find the software in your Windows Start menu at: **Start** > **EcoStruxure Machine SCADA Expert v8.1**

> **Note:**
>
> In Windows 8 / Windows Server 2012, the software should be located at: **Start** > **Apps** > **Schneider Electric** > **EcoStruxure Machine SCADA Expert v8.1**
>
> In Windows 7 / Windows Server 2008, the software should be located at: **Start** > **All Programs** > **Schneider Electric** > **EcoStruxure Machine SCADA Expert v8.1**

The software includes the following "apps" (applications):

EcoStruxure Machine SCADA Expert v8.1

The project development environment, project runtime, and project viewer. Its capabilities are determined by your software license.

EcoStruxure Machine SCADA Expert v8.1 Help Manual

A complete technical reference and user guide for all of the EcoStruxure Machine SCADA Expert software.

EcoStruxure Machine SCADA Expert v8.1 Quick Start Guide

A brief guide to installing and using the project development environment, including a tutorial for developing a simple project.

EcoStruxure Machine SCADA Expert v8.1 Register

A utility program that manages your EcoStruxure Machine SCADA Expert software license.

EcoStruxure Machine SCADA Expert v8.1 Release Notes

A list of changes in the EcoStruxure Machine SCADA Expert software.

EcoStruxure Machine SCADA Expert v8.1 Remote Agent

A utility program that allows other stations to remotely manage EcoStruxure Machine SCADA Expert as a project runtime.

EcoStruxure Machine SCADA Expert v8.1 StartUp

A shortcut that automatically starts the project runtime and then runs the most recent project.

There should also be a shortcut icon on your desktop.

To run the software, do one of the following:

- Double-click the shortcut icon on your desktop; or
- Click **Start** > **EcoStruxure Machine SCADA Expert v8.1** > **EcoStruxure Machine SCADA Expert v8.1** .

If the installation failed for any reason, you can use System Restore to restore the computer to the restore point that was created at the beginning of the installation. For more information about System Restore, go to: https://support.microsoft.com/help/17127/windows-back-up-restore

Execution Modes

EcoStruxure Machine SCADA Expert support the following execution modes:

Execution Mode	EcoStruxure Machine SCADA Expert
Evaluation Mode	Supported
Demo Mode	Supported
Licensed for Engineering Only	Supported
Licensed for Runtime Only	Supported
Licensed for Engineering + Runtime	Supported

Evaluation Mode

Enables all of the product's engineering and runtime features.

The first time you install EcoStruxure Machine SCADA Expert 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 program automatically converts to Demo Mode until you apply a valid license. You cannot reactivate Evaluation Mode, even if you reinstall the software on your computer.

> **Note:** Each version of EcoStruxure Machine SCADA Expert has an evaluation period that is independent of every other version. For example, if an earlier version is running in Demo Mode because its evaluation period has expired, and then you install the latest version on the same computer, the latest version will begin its own 40-hour evaluation period and the earlier version will continue to run in Demo Mode.

Demo Mode

Allows you to download projects to remote stations and to run projects for testing or demonstration purposes. You can execute runtime tasks and use the debugging tools (*LogWin* and *Database Spy*), but they shut down automatically after running for two hours continuously. You can restart the Demo Mode again and run for another two hours, and so on.

You cannot create or modify screens, worksheets, or project settings in Demo Mode.

Licensed for Engineering Only

Enables all development options for an unlimited time.

This mode also allows you to continuously run the runtime tasks and debugging tools (*Database Spy*, *Output* window, and *LogWin* module) for 72 hours. After that period, these tasks shut down, but you can restart them and run for another 72 hours, and so on. You can use this license for development and testing only.

Licensed for Runtime Only

Enables all runtime tasks and debugging tools (*Database Spy*, *Output* window, and *LogWin* module) for unlimited time, but you cannot create or modify screens and/or worksheets.

The menu options available in Runtime Only mode are the same as the options listed for Demo Mode (see previous table).

Licensed for Engineering + Runtime

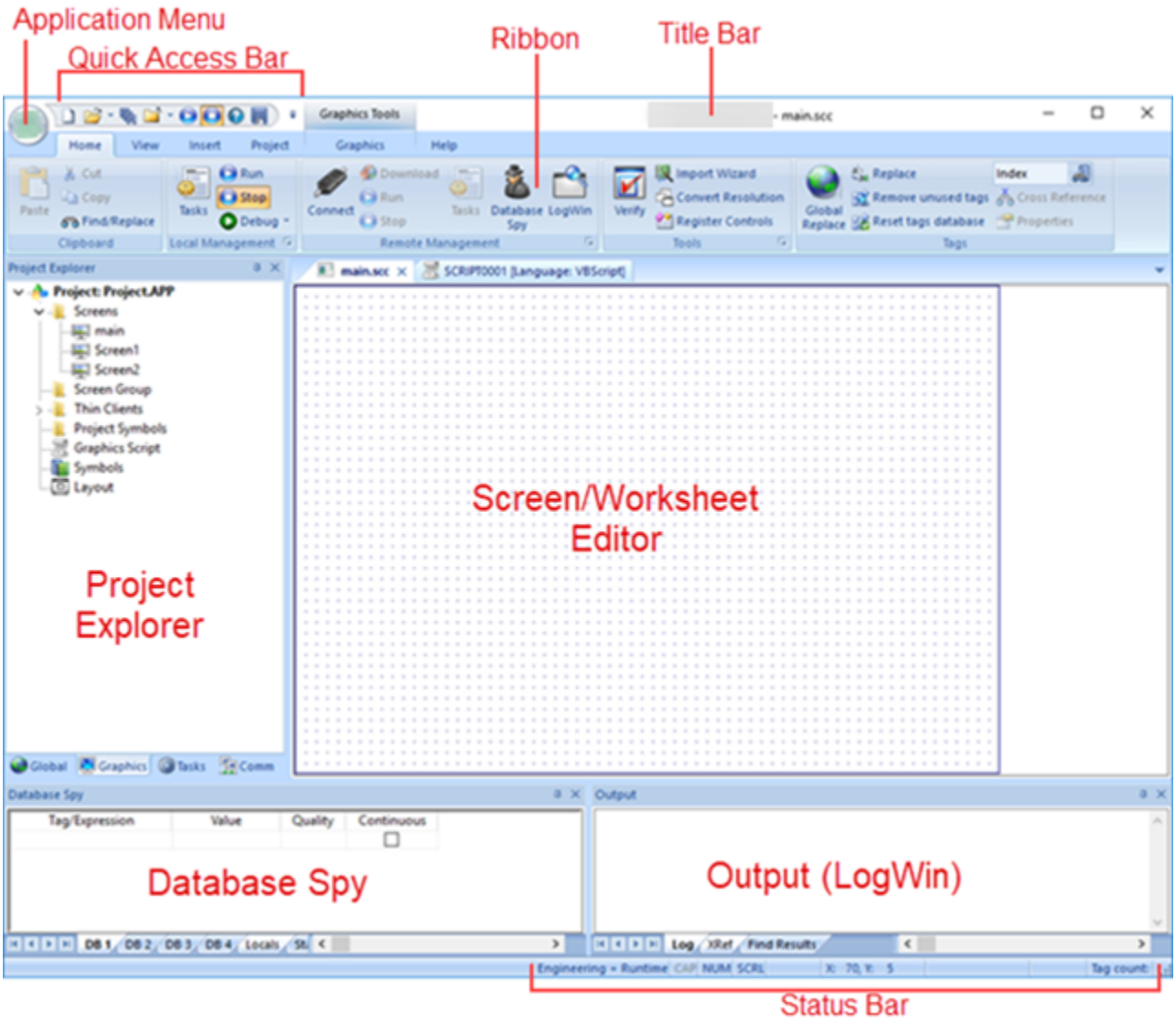
Enables all development options, runtime tasks, and debugging tools (*Database Spy*, *Output* window, and *LogWin* module) for an unlimited time.

> **Note:** The Remote Management tool is always available, regardless of the execution mode, so that you can upload files from or download files to remote stations.

To see which execution mode you are currently running, click **About** on the Help tab of the ribbon; the *About* dialog shows the execution mode, including the time remaining if you are in Evaluation Mode.

The Development Environment

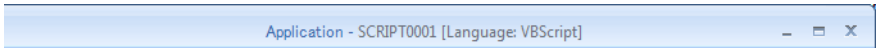
EcoStruxure Machine SCADA Expert incorporates a modern, Ribbon-based Windows interface to provide an integrated and user-friendly development environment.



The EcoStruxure Machine SCADA Expert Development Environment

Title Bar

The Title Bar located along the top of the development environment displays the application name (e.g., EcoStruxure Machine SCADA Expert) followed by the name of the active screen or worksheet (if any).



Example of Title Bar

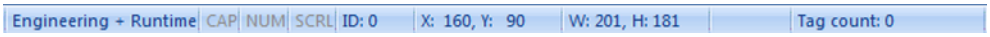
The Title Bar also provides the following buttons (from left to right):

- **Minimize** button : Click to minimize the development environment window to the Taskbar.
- **Restore Down** / **Maximize**: Click to toggle the development environment window between two sizes:
  - **Restore Down** button  reduces the window to its original (default) size.
  - **Maximize** button  enlarges the window to fill your computer screen.
- **Close** button : Click to save the database and then close the development environment. If you modified any screens or worksheets, the application prompts you to save your work. This button's function is similar to clicking **Exit Application** on the Application menu.

> **Note:** Closing the development environment does *not* close either the project viewer or the runtime system, if they are running.

Status Bar

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



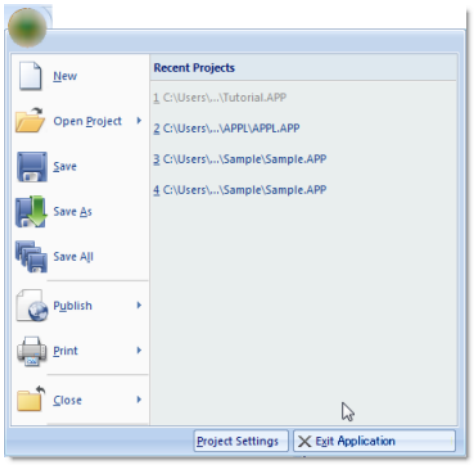
Example of Status Bar

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

Field	Description
Execution Mode	The current execution mode of the application.
CAP	Indicates whether the keyboard **Caps Lock** is on (black) or off (grey).
NUM	Indicates whether the keyboard **Num Lock** is on (black) or off (grey).
SCRL	Indicates whether the keyboard **Scroll Lock** is on (black) or off (grey).
Object ID	The ID number of a selected screen object.
Cursor Position	The location of the cursor on the active screen or worksheet. If it's a screen, then the position of the *mouse* cursor is given as X,Y coordinates, where X is the number of pixels from the left edge of the screen and Y is the number of pixels from the top edge of the screen. If it's a worksheet, then the position of the *text* cursor is given as Line and Column.
Object Size	The size (in pixels) of a selected screen object, where W is the width and H is the height.
No DRAG	Indicates whether dragging is disabled (**No DRAG**) or enabled (empty) in the active screen.
Tag Count	The total number of tags used so far in the project.

Application button

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



Application button opens menu of commands

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.

Move the Quick Access Toolbar

The Quick Access Toolbar can be located in one of two places:

- Upper-left corner next to the Application button (default location); or
- Below the ribbon, where it can run the full length of the application window.

If you don't want the Quick Access Toolbar to be displayed in its current location, you can move it to the other location:

1. Click **Customize Quick Access Toolbar** .
2. In the list, click **Show Below Ribbon** or **Show Above Ribbon**.

Add a command to the Quick Access Toolbar

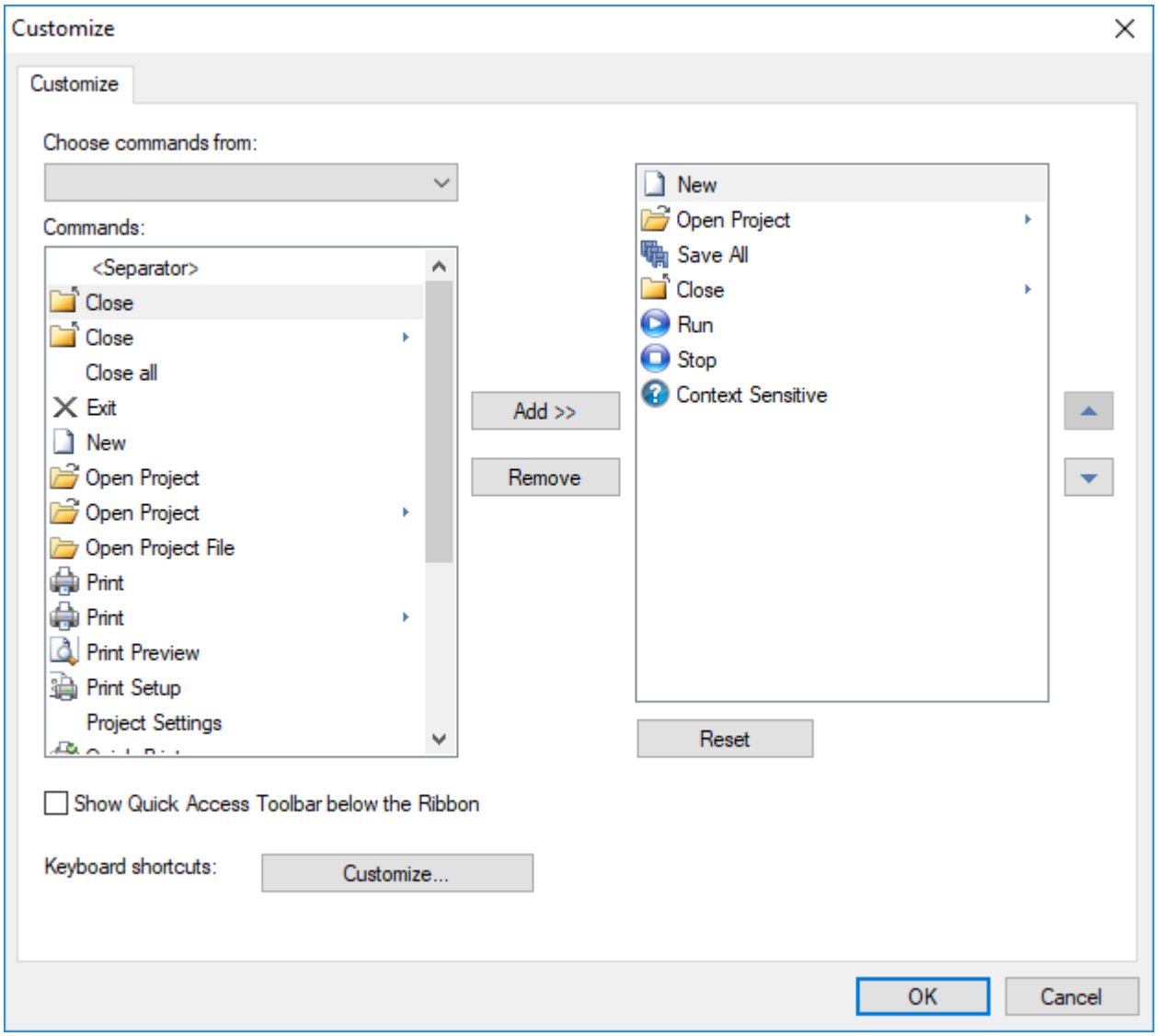
You can add a command to the Quick Access Toolbar directly from commands that are displayed on the ribbon:

1. On the ribbon, click the appropriate tab or group to display the command that you want to add to the Quick Access Toolbar.
2. Right-click the command, and then click **Add to Quick Access Toolbar** on the shortcut menu.

You can also add and remove commands — as well as reset the toolbar to its default — using the *Customize* dialog:

1. Click **Customize Quick Access Toolbar** .

2. In the list, click **More Commands**. The *Customize* dialog is displayed.



Customize Quick Access Toolbar dialog

3. In the **Choose commands from** menu, select the appropriate Ribbon tab. The commands from that tab are displayed in the **Commands** list.
4. In the **Commands** list, select the command that you want to add to the Quick Access Toolbar.
5. Click **Add**.

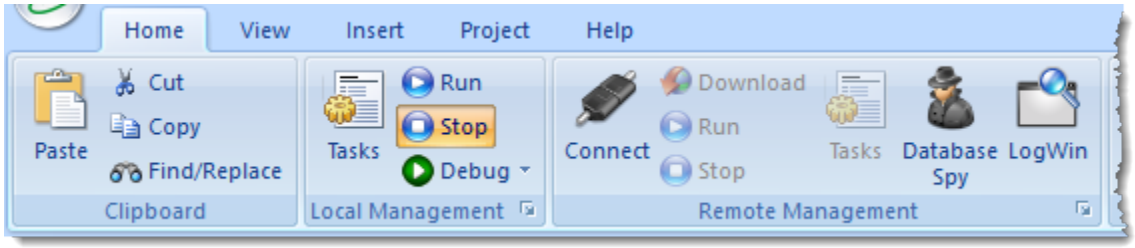
Only commands can be added to the Quick Access Toolbar. The contents of most lists, such as indent and spacing values and individual styles, which also appear on the ribbon, cannot be added to the Quick Access Toolbar.

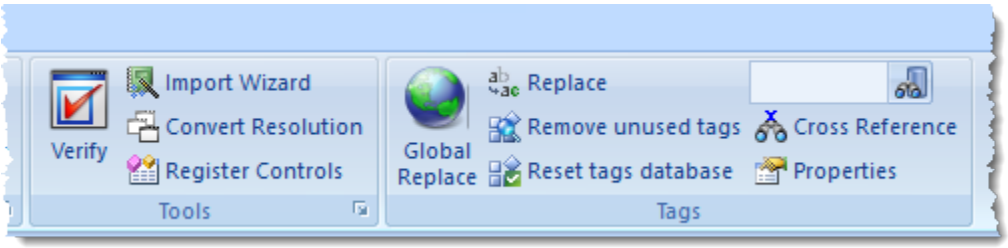
Ribbon

The new ribbon combines the numerous menus and toolbars from the previous versions of this software into a single, user-friendly interface. Almost all application commands are now on the ribbon, organized into tabs and groups according to general usage.

Home tab

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



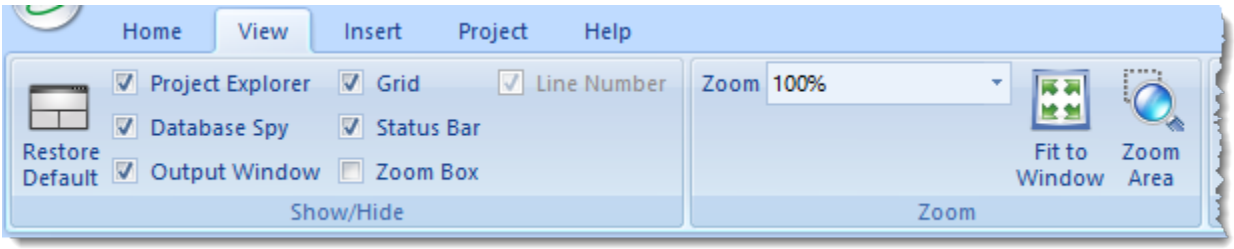


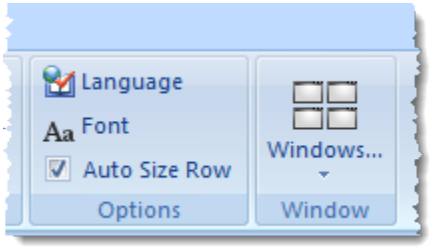
The tools are organized into the following groups:

- **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 application 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 so that you can download the project to it, and then run, stop, and troubleshoot the project on that station. For more information, see Remote Management.
- **Tools**: Miscellaneous tools to verify the project, import tags from other projects, convert screen resolutions, and register ActiveX and .NET 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.



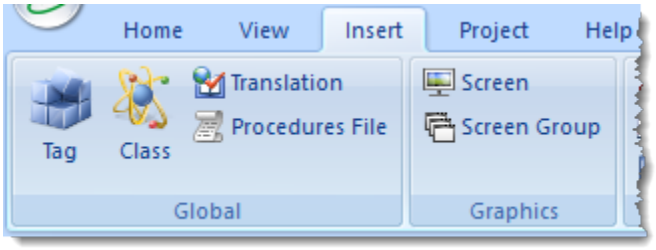


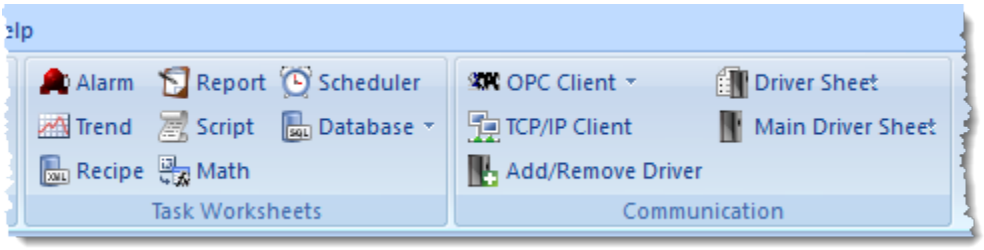
The tools are organized into the following groups:

- **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.



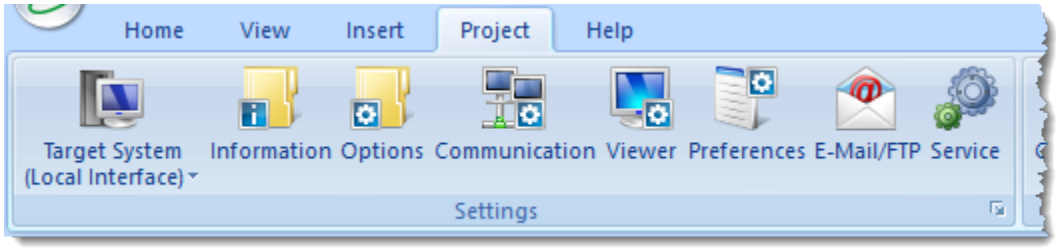


The tools are organized into the following groups:

- **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.



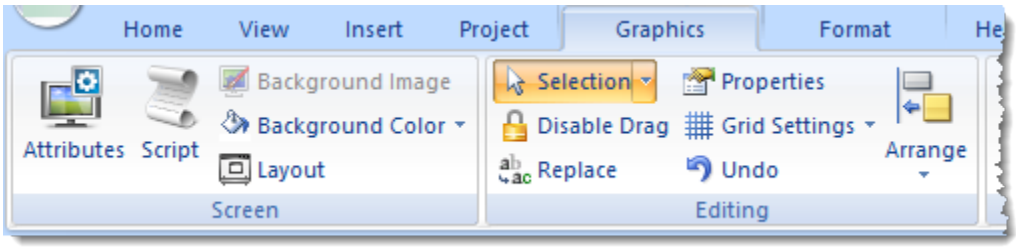


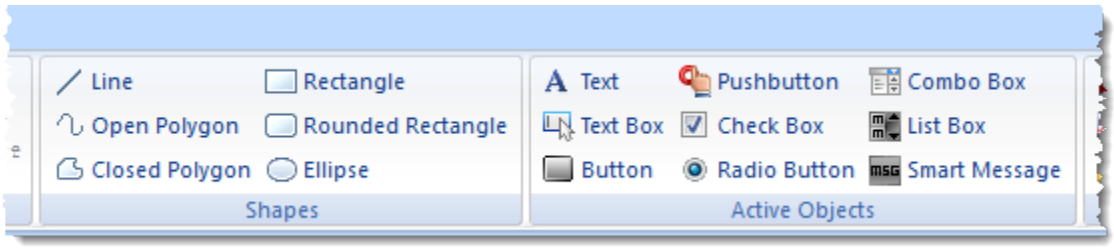
The tools are organized into the following groups:

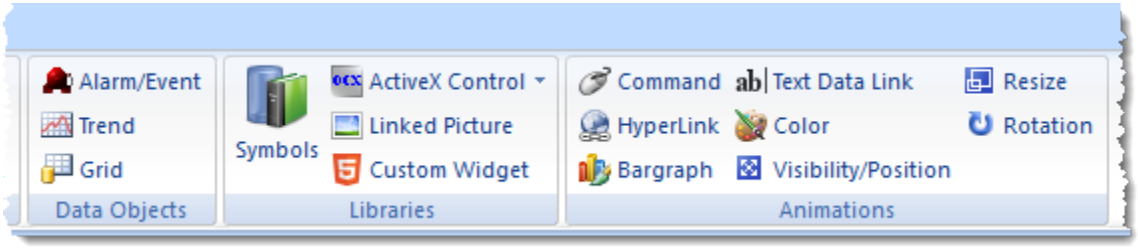
- **Settings**: Configure the general project settings, set the project to run as a Windows service, or enable workgroup collaboration and version control.
- **Security System**: Enable and configure the project security system.
- **Web**: Configure the project to accept connections from a variety of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

Graphics tab

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







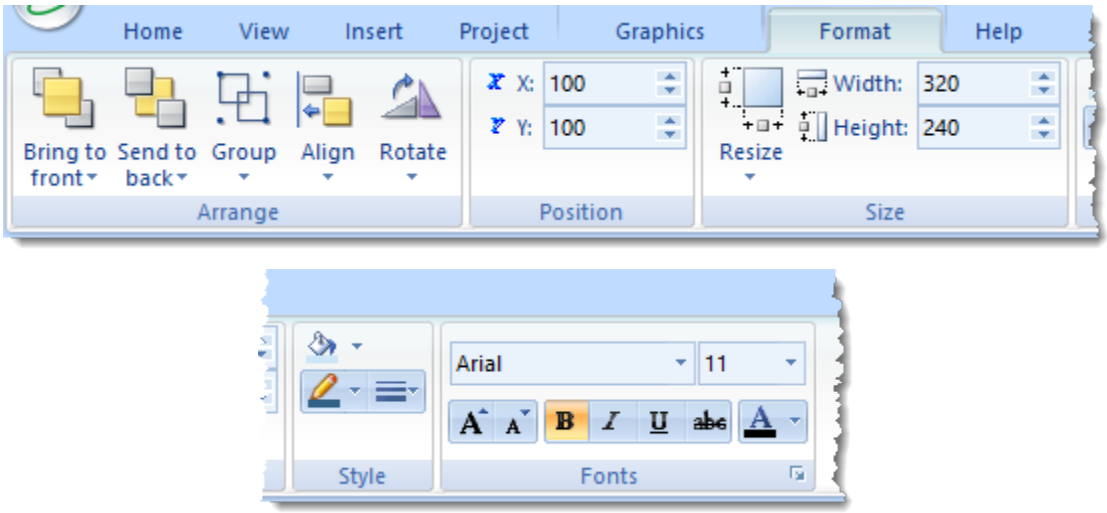
> **Note:** This tab is available only when you have a project screen open for editing.

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 and .NET controls , external image files, and HTML5-based custom widgets.
- **Animations**: Apply animations to other screen objects.

Format tab

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



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

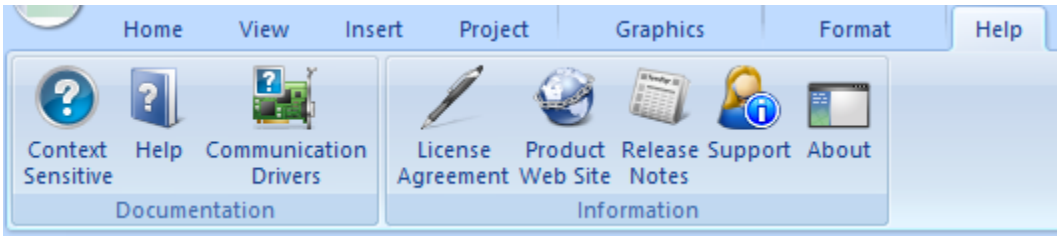
The tools are organized into the following groups:

- **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.

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 application, including this help file / technical reference and notes for the individual communication drivers.
- **Information**: Access other information about EcoStruxure Machine SCADA Expert, 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 organizes all of the screens, worksheets, and other items that comprise your project and presents them in an expandable tree-view.

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

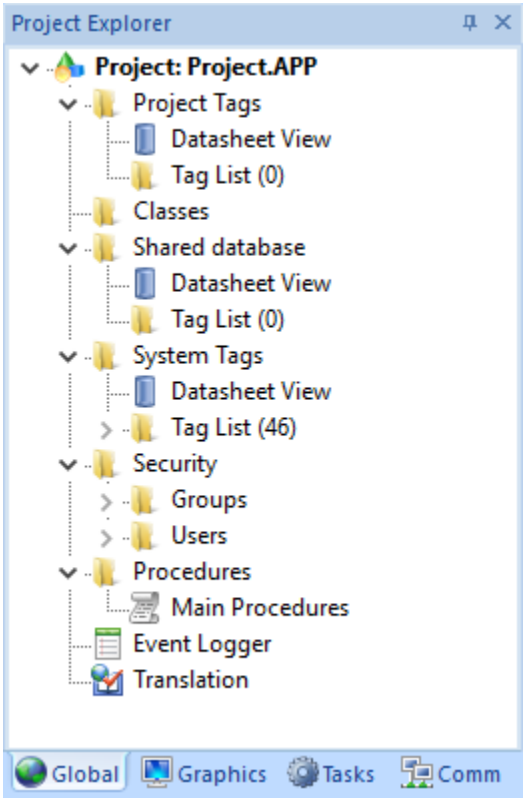
To close a folder, click the Collapse icon ◢ to the left of the folder.

If you right-click any item in the Project Explorer, then a shortcut menu will appear 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 UI translation.



Global tab of the Project Explorer

The folders on the Global tab are described in the following sections:

Project Tags

The project tags database contains all of the data tags that you create during project development, such as screen tags (e.g., **`button1_state`**) 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

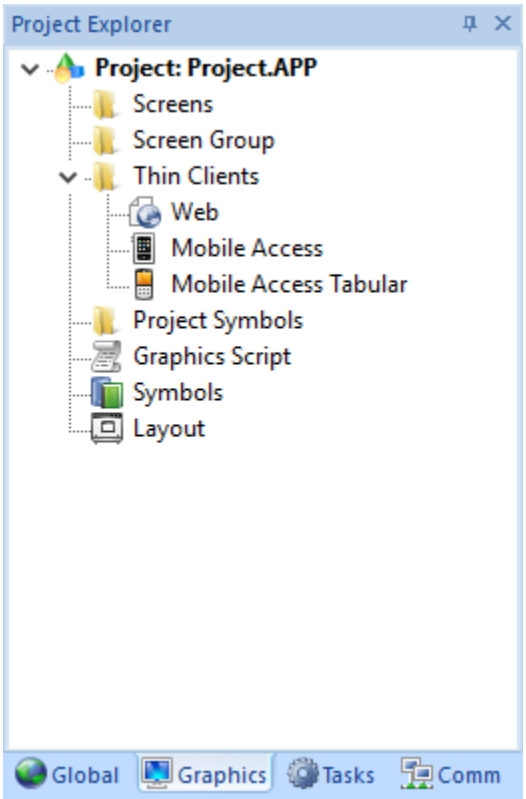
The event logger saves important runtime messages and task results to an external database.

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.



Graphics tab of the Project Explorer

The folders on the Graphics tab are described in the following sections:

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 desktop web browsers, 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

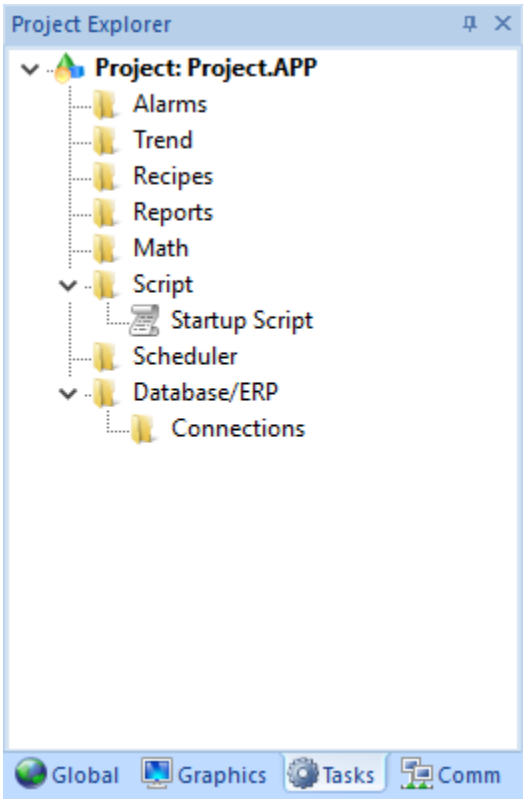
The symbols library contains not only the custom symbols that you create (see Project Symbols above), but also a large selection of premade symbols that are installed with the development application.

Layout

The layout editor displays all of the screens the are 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.



Tasks tab of the Project Explorer

The folders on the Tasks tab are described in the following sections:

Alarms

You can use Alarm worksheets to define when alarms are trigged, how they must be handled, and what messages they generate.

(You can then use the Alarm/Event Control screen object to display your alarms on screen, but that is a separate procedure.)

Trends

You can use Trend worksheets to select project tags that should be displayed as data trends and/or saved as historical data.

(You can then use the Trend Control screen object to actually display your trends on screen, but that is a separate procedure.)

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.

(You can then call the `Recipe` function to actually run a configured Recipe worksheet, but that is a separate procedure.)

Reports

You can use Report worksheets to design runtime reports that are either sent to a printer or saved to disk.

(You can then call the `Report` function to actually run a configured Report worksheet, but that is a separate procedure.)

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

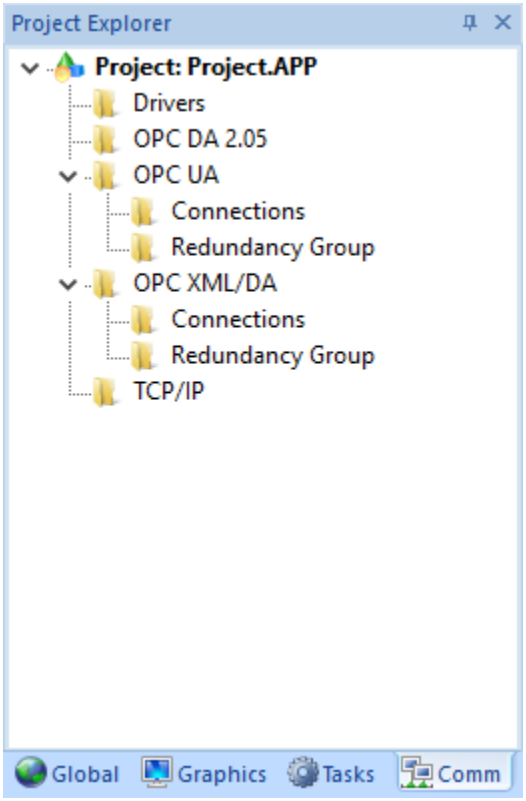
You can use Scheduler worksheets to run commands at specified times, dates, or trigger events.

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.



Comm tab of the Project Explorer

The folders on the Comm tab are described in the following sections:

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.

OPC XML/DA

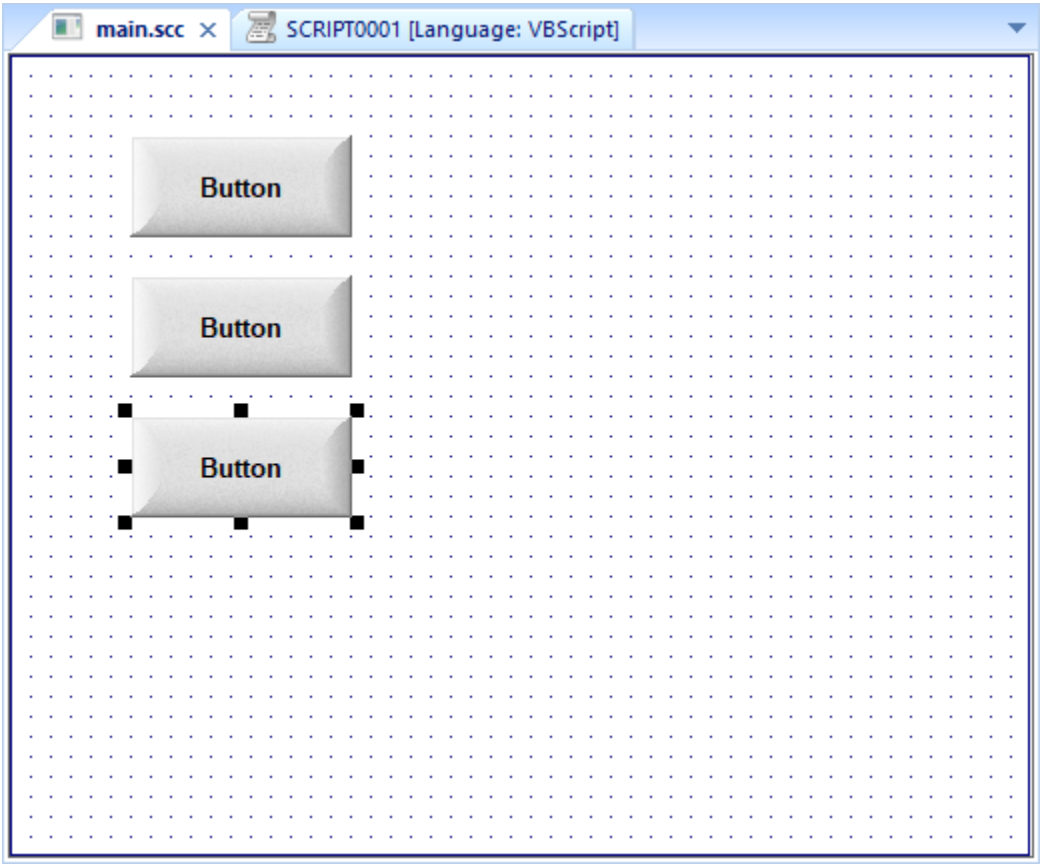
You can use OPC XML/DA worksheets to communicate with OPC servers via the new OPC XML-DA protocol.

TCP/IP

You can use TCP/IP worksheets to configure communication between your own project and other EcoStruxure Machine SCADA Expert 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.

Screen/Worksheet Editor

Use the powerful, object-oriented screen 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.



Screen/Worksheet Editor

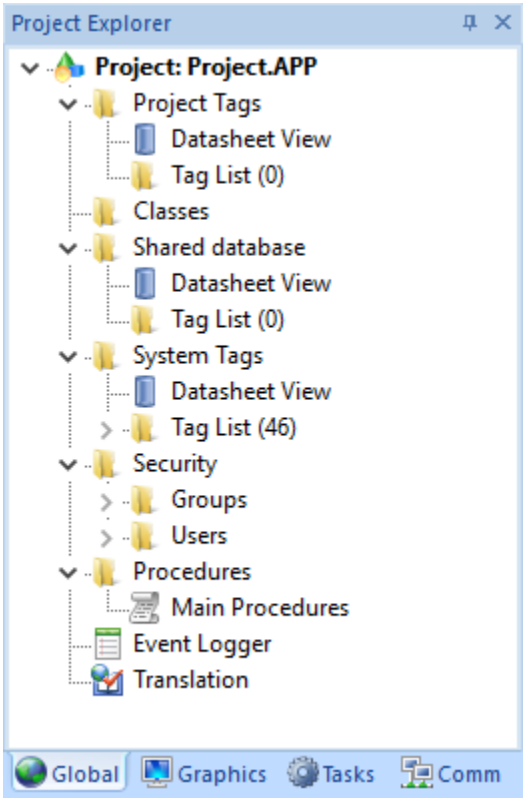
Other screen editor features include:

- 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

About Tags and the Project Database

Tags are a core component of any EcoStruxure Machine SCADA Expert project. Simply put, tags are variables used by EcoStruxure Machine SCADA Expert to receive and store data obtained from communication with plant floor devices, from the results of calculations and functions, and from user input. In turn, tags can be used to display information on screens (and Web pages), to manipulate screen objects, and to control runtime tasks.

But tags are more than simple variables. EcoStruxure Machine SCADA Expert 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 EcoStruxure Machine SCADA Expert tag has both a value and various properties that can be accessed, some at development and others only at runtime.



All tags are organized into one of the following categories, which are represented by folders on the **Global** tab of the *Project Explorer*:

- **Project Tags** are 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
- **Shared Database** tags are created in a PC-based control program and then imported into EcoStruxure Machine SCADA Expert's tags database.

For example you might create tags in SteepleChase and import them into EcoStruxure Machine SCADA Expert so EcoStruxure Machine SCADA Expert can read/write data from a SteepleChase PC-based control product.

You cannot modify shared tags within EcoStruxure Machine SCADA Expert — you must modify the tags in the original PC-based control program, and then re-import them into the Tags database.

- **System Tags** are predefined tags with predetermined functions that are used for EcoStruxure Machine SCADA Expert supervisory tasks. For example,
  - Date tags hold the current date in string format
  - Time tags hold the current time in string format

  Most system tags are *read-only*, which means you cannot add, edit, or remove these tags from the database.

  To see a list of the system tags, select the **Global** tab in the *Project Explorer*, open the **System Tags** folder, and open the **Tag List** subfolder.

After creating a tag, you can use it anywhere within the project, and you can use the same tag for more than one object or attribute.

Naming the Tag

Observe the following guidelines when you name a tag or class member:

- Each name must be unique — you cannot specify the same name as another user-created tag or class member, an imported tag, a system tag, or a built-in function. If you enter an existing name, the project development environment will recognize that name and it will not prompt you to create a new tag.
- The name can be composed of uppercase and lowercase letters (**A–Z**, **a–z**), the accented forms of those letters (e.g., **é**, **ü**, **ç**), standard numerals (**0–9**), and the underscore character (_). All other punctuation, special characters, mathematical symbols, and non-Latin alphabets are not allowed.
- The name must begin with a letter.
- The name can be up to 255 characters long.
- Even though the name can be composed of both uppercase and lowercase letters, it is not actually case-sensitive. It will be recognized as long as it is spelled correctly. Therefore, you can use uppercase and lowercase letters to make the name more readable to you. For example, **TankLevel** and **tanklevel** both refer to the same tag.

> **Tip:** To indicate a tag will be used as an indirect tag, insert the "at" sign (@) at the beginning of the tag name.

Choosing the Tag Data Type

Another consideration when designing a tag is what type of data the tag will receive. EcoStruxure Machine SCADA Expert recognizes the following, standard tag *data types*:

- **Boolean** (*one bit*): Simple boolean with the possible values of 0 (false) and 1 (true). Equivalent to the "bool" data type in C++. Typically used for turning objects off and on or for closing and opening objects.
- **Integer** (*four bytes*): Integer number (positive, negative, or zero) internally stored as a signed 32-bit. Equivalent to the "signed long int" data type in C++. Typically used for counting whole numbers or setting whole number values. Examples: `0`, `5`, `-200`.
- **Real** (*floating point, eight bytes*): Real number that is stored internally as a signed 64-bit. Equivalent to the "double" data type in C++. Typically used for measurements or for decimal or fractional values.
- **String** (*alphanumeric data, up to 1024 characters*): Character string up to 1024 characters that holds letters, numbers, or special characters. Supports both ASCII and UNICODE characters. Examples: `Recipe product X123`, `01/01/90`, `*** On ***`.

You can also assign a new tag to a class that you have previously created.

You can find these tag types (and their respective icons) in the **Global** tab of the *Project Explorer*.

Using Array Tags

EcoStruxure Machine SCADA Expert tags can consist of a single value or an array of values.

> **Note:** The maximum array size is 16384 as long as it does not exceed the maximum number of tags supported by the license (Product Type) selected for the project. Each array position (including the position 0) counts as one tag for licensing restrictions, because each position has an independent value.

An array tag is a set of tags with the same name, which is identified by indexes (a matrix of ***n*** lines and 1 column). The maximum array size depends on the product specification. You can use the following syntax to access an array tag:

```
ArrayTagName[ArrayIndex]
```

For example: **tank[0]**, **tank[1]**, **tank[2]**, and **tank[500]**.

> **Note:** You must specify a maximum index for each array tag in the **size** column of any datasheet. You can specify ***n*** to indicate the array tag has positions from 0 to *n*. For example, if the size of TagA is 3, the tag elements could be **TagA[0]**, **TagA[1]**, **TagA[2]**, and **TagA[3]**.

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

An array index can be a tag, a numeric value, or an expression with the arithmetic operator "**+**".

> **Note:** When you refer to an array with an index using the **+** arithmetic operation, you must use the following syntax:
>
> ```
> ArrayTagName[NumValue1+NumValue2]
> ```
>
> Where ***NumValue1*** and ***NumValue2*** can be an integer tag or a numerical constant. For example: **temperature[tk+2]** or **temperature[tk+6]**.

Using array tags in any EcoStruxure Machine SCADA Expert task can save a signification amount of project development time. For example, if you needed tag points related to the temperature of four tanks. The conventional configuration method is the following:

- **temperature1**: high temperature on tank 1
- **temperature2**: high temperature on tank 2
- **temperature3**: high temperature on tank 3
- **temperature4**: high temperature on tank 4

Using array tags simplifies this task, as follows:

- **temperature[j]**: high temperature on tank {j}

> **Note:** When you create a four-position array tag, the system creates five positions (from 0 to 4). For example:
>
> ```
> tag_example[15] //start position=0, end position=15
> ```
>
> Therefore, the **tag_example[15]** array has 16 elements.

When using another tag to reference the index of an array, if the value of the tag is outside the size of the array, then the following results are given:

- If ***IndexTag*** is greater than the size of the array, then **MyArray[*IndexTag*]** will point to the end position of the array; and
- If ***IndexTag*** is less than 0, then **MyArray[*IndexTag*]** will point to the start position of the array.

Array Tags

An *array* tag consists of a set of tags that all have the same name, but use unique array indexes (a matrix of *n* lines and one column) to differentiate between each tag. An *array index* can be a fixed value, another tag or an expression. Maximum array sizes are determined by product specifications.

You can use array tags to:

- Simplify configurations
- Enable multiplexing in screens, recipes, and communication interfaces
- Save development time during tag declaration

You specify array tags in one of two formats:

- For a simple array tag, type:

  ***ArrayTagName*[*ArrayIndex*]**

- For a complex array tag (where the array index is an expression consisting of a tag and an arithmetic operation), type:

  ***ArrayTagName*[*ArrayIndex*+*c*]**

  Where:

  - ***ArrayTagName*** is the tag name;
  - **[*ArrayIndex*]** is the unique index (fixed value or another tag);
  - **+** is an arithmetic operation; and
  - ***c*** is a numerical constant.

> **Note:**
>
> - You must specify a maximum index for each array tag by typing a value (*n*) in the Array Size column of an *Project Tags* datasheet or in the Array Size field on a *New Tag* dialog.
>
> When you create an *n*-position array tag, EcoStruxure Machine SCADA Expert actually creates ***n*+1** positions (from 0 to *n*). For example, if you specify **ArrayTag[15]**, the array will have 16 elements, where 0 is the start position and 15 is the end position.
>
> - You must not use spaces in an array tag.
>
> When EcoStruxure Machine SCADA Expert reads a tag it begins with the first character and continues until it finds the first space or null character. Consequently, the system does not recognize any characters following the space as part of the array tag.
>
> For example, if you type **a[second + 1]**, EcoStruxure Machine SCADA Expert regards **a[second** as the tag and considers it invalid because EcoStruxure Machine SCADA Expert does not find (recognize) the closing bracket. However, if you type **a[second+1]**, this is a valid array tag.

You can specify an array tag wherever you would use a variable name. Also, because array tags greatly simplify configuration tasks and can save development time, we suggest using them whenever possible.

For example, suppose you want to monitor the temperature of four tanks. The conventional configuration method is:

- **temperature1** — high temperature on tank 1
- **temperature2** — high temperature on tank 2

- **`temperature3`** — high temperature on tank 3
- **`temperature4`** — high temperature on tank 4

You can use array tags to simplify this task as follows (where [*n*] represents the tank number):

- **`temperature[`*`n`*`]`** — high temperature on tank [*n*]

The following table contains some additional examples of an array tag:

Array Tag Examples

Array Tag Example	Description
`Tank[1]`, **`Tank[2]`**, **`Tank[500]`**	Simple arrays, where the array indexes (**`1`**, **`2`**, and **`500`**) are numerical constants. For example, tank numbers.
`Tank[tk]`	A simple array, where the array index (**`tk`**) is a tag. For example, a tag representing the tank number.
`Tank[tk+1]`	A complex array, where the array index (**`tk+1`**) is an expression. For example, the value of **`tk`** (tank number) plus 1.



Note: When using another tag to reference the index of an array, if the value of the tag is outside the size of the array, then the following results are given:

- If ***`IndexTag`*** is greater than the size of the array, then **`MyArray[`*`IndexTag`*`]`** will point to the end position of the array; and
- If ***`IndexTag`*** is less than 0, then **`MyArray[`*`IndexTag`*`]`** will point to the start position of the array (i.e., **`MyArray[0]`**).

About indirect tags

EcoStruxure Machine SCADA Expert supports indirect access to tags in the database. For example, consider a tag ***X*** of the String type. This tag can hold the name of any other tag in the database (that is, it can provide a pointer to any other type of tag, including a class type). The syntax for an indirect tag is straightforward: ***@IndirectTagName***. For example, assume that a tag named ***X*** holds a "**TEMP**" string. Reading and/or writing to ***@X*** provides access to the value of the **TEMP** variable.

> **Note:** 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. For example:

Class	**TANK** with members **Level**
Tag	**TK** of the class **TANK**
Tag	**XCLASS**

To access the **TK.Level** value, you must store the " **TK.Level**" value within the **XCLASS** tag and use the syntax, **@XCLASS**. You can also refer to a member of a class-type tag directly; identifying a class-type that points to a class member.

For example:

Class	**TANK** with members **Level**
Tag	**TK** of the class **TANK**
Tag	**XCLASS** of the class **TANK**

To access the **TK.Level** value, you must store the "**TK**" value within the **XCLASS** tag and use the syntax, **@XCLASS.Level**.

When creating tags for indirect use, place an **@** in the tag column rather than creating them as strings. For the type, write the type of tag for which you are creating a reference. Follow the **XCLASS** example: **@Z Integer**, **@X Class:TANK**.

Indirect Tags

Indirect tags "point" to other database tags (including class-type tags). Using indirect tags can save development time because they keep you from having to create duplicate tags (and the logic built into them).

You create an indirect tag from any string-type tag simply by typing the **@** symbol in front of the tag name ***@TagName***.

- To reference a simple tag, assume the **strX** tag (a string tag) holds the value **"Tank"**, which is the name of another tag, then reading from or writing to **@strX** provides access to the value of the **Tank** tag.
- To reference a class-type tag and member, you simply create a string tag that points to the class tag and the member. For example, if a tag **strX** (a string tag) holds the value **"Tank.Level"**, which is the name of the class tag, then reading from or writing to **@strX** provides access to the value of the **Tank.Level** member.
- You can also point directly to a class-type tag member; by identifying a class-type that points to a class member. For example: to access the **Tank.Level** member of the class, you must store the **"Tank"** value within the **strX** tag and use the syntax, **@strX.Level**.

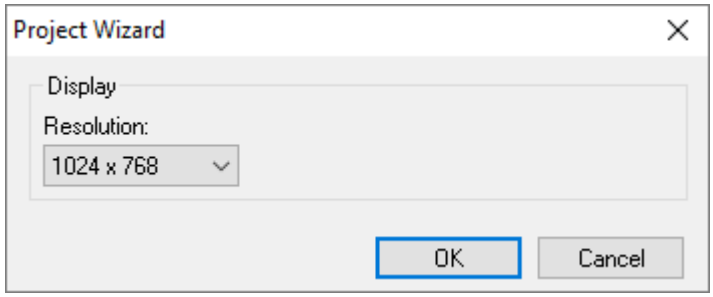
Tutorial: Building a Simple Project

This section explains, using a step-by-step tutorial, how to build a simple project, as well as how to select and configure an I/O driver.

Creating a new project

This part of the tutorial shows how to create a new project, including how to give it a name and then select the target platform and system.

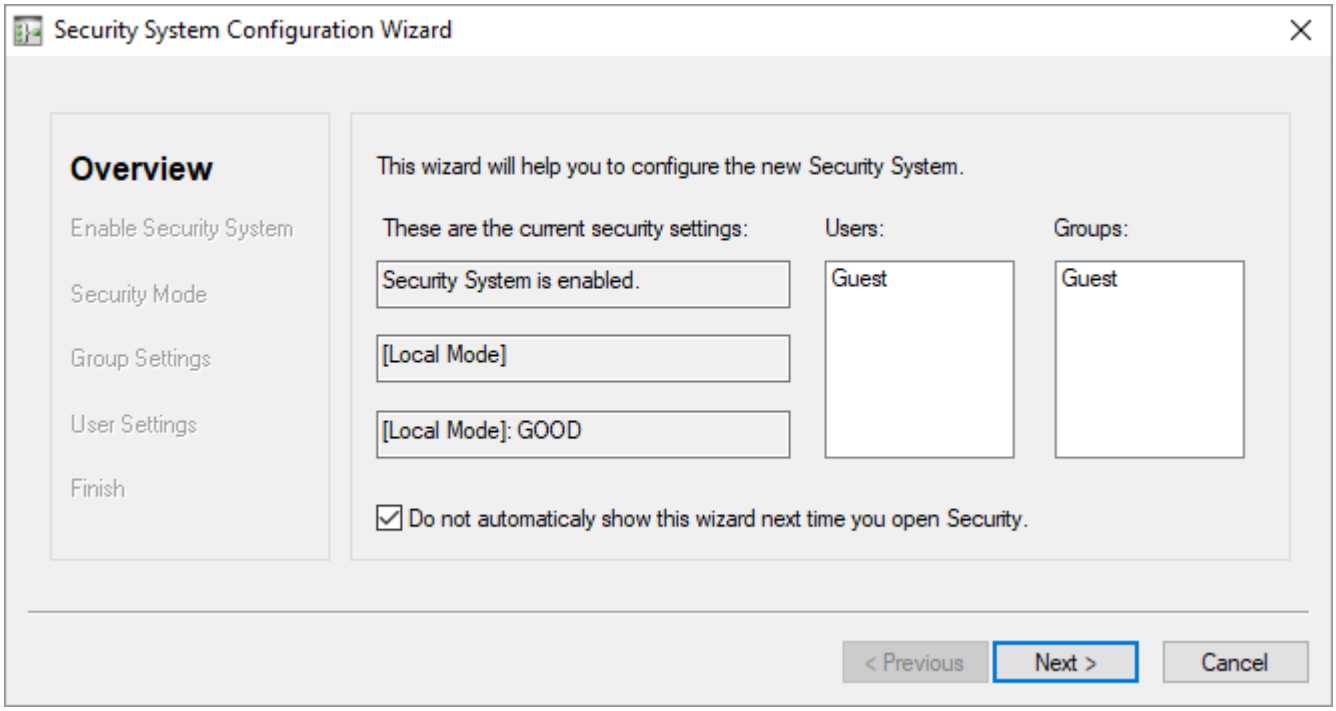
1. Click the Application button in the top-left corner of the development environment, and then click *New* on the Application menu.
   The *New* dialog is displayed.
2. Click the **Project** tab.
3. In the **Project name** box, type the name of your project.
   For this tutorial, type `Tutorial`.
   The development application automatically creates a new directory of the same name and assigns your project file to that directory. (Notice the **Configuration file** text box in the figure.) To put your project file somewhere other than in the default projects folder, click **Browse** and navigate to the preferred location.
4. In the **Product type** list, select the type of project that you want to build.
5. Click **OK**.
   The *New* dialog is closed and the *Project Wizard* dialog is displayed.
6. In the **Resolution** list, select **1024 x 768**.



Specifying an empty Application with 1024x768 resolution

7. Click **OK**.

The *Project Wizard* dialog is closed, the project is created in the development environment, and the *Security System Configuration Wizard* is displayed.



Security System Configuration Wizard

8. Use the *Security System Configuration Wizard* to set a Main Password for your project.
   The security system is enabled by default for all new projects.

When you finish the *Security System Configuration Wizard*, your new project is ready for development.

Specifying the startup screen

This part of the tutorial shows how to open the project settings and then specify which screen should be displayed on startup.

- Use the **Information** tab to provide information that identifies the project (such as project description, revision number, Company name, Author's name, field equipment, and general notes).
- Use the **Options** tab to specify generic settings for the project, such as the Target System, Automatic Translation, Alarm history and Events, Default Database and Shared Tags.
- Use the **Viewer** tab to enable/disable the runtime desktop parameters.
- Use the **Communication** tab to specify communication parameters relating to the project in general.
- Use the **Preferences** tab to enable/disable warning messages when using the development application.

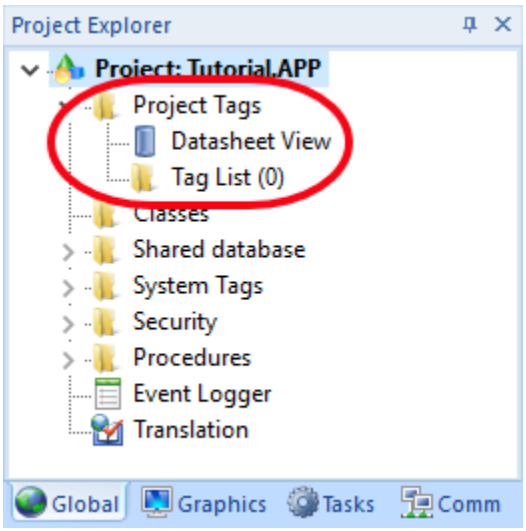
To specify the startup screen:

1. On the **Project** tab of the ribbon, in the **Settings** group, click **Viewer**.
   The *Project Settings* dialog is displayed with the **Viewer** tab selected.
2. In the **Startup screen** box, type `main.scc`.
   When you run the project, it will automatically display the main screen (or whichever screen you specify) first. You can specify a screen before you create it, but if the screen has been created, then you can also select it from the list.
3. Click **OK**.

Creating tags

This part of the tutorial shows how to create new tags by adding them to the Project Tags datasheet.

A tag is any variable that holds a value. All tags created in a project are stored in the Project Tags folder, on the Global tab of the Project Explorer.

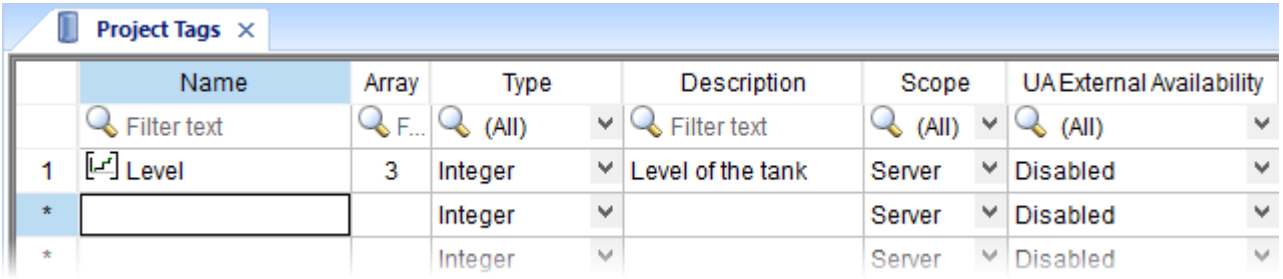


Project Tags folder

1. In the Project Explorer, click the **Global** tab.
2. Double-click **Project Tags** to expand the folder.
3. Double-click **Datasheet View** to open the *Project Tags* datasheet.
4. Use the following parameters to create a tag for the sample project.
   a) **Name**: Specify a unique tag name. For this tutorial, type `Level`.
   b) **Array**: Specify the top array index of the tag. (Simple tags have an Array of `0`.) For this tutorial, type `3`.
      Each array index relates to one of the three tanks:
      - **`Level[1]`** is the level of Tank #1
      - **`Level[2]`** is the level of Tank #2
      - **`Level[3]`** is the level of Tank #3

      You will not use **`Level[0]`** in this tutorial, even though it is a valid tag.
   c) **Type**: Specify the data type of the tag: Boolean, Integer, Real, String, or Class. For this tutorial, select **Integer**.
   d) **Description** (optional): Type a description of the tag for documentation purposes only.
   e) **Scope**: Specify how the tag is managed between the Server and the Thin Client stations.
      - Select **Local** if you want the tag to have independent values on the Server and Client stations.
      - Select **Server** if you want the tag to share the same value on the Server and Client stations.

For this tutorial, select **Server**.



Creating the Level tag

5. Save and close the *Project Tags* datasheet.

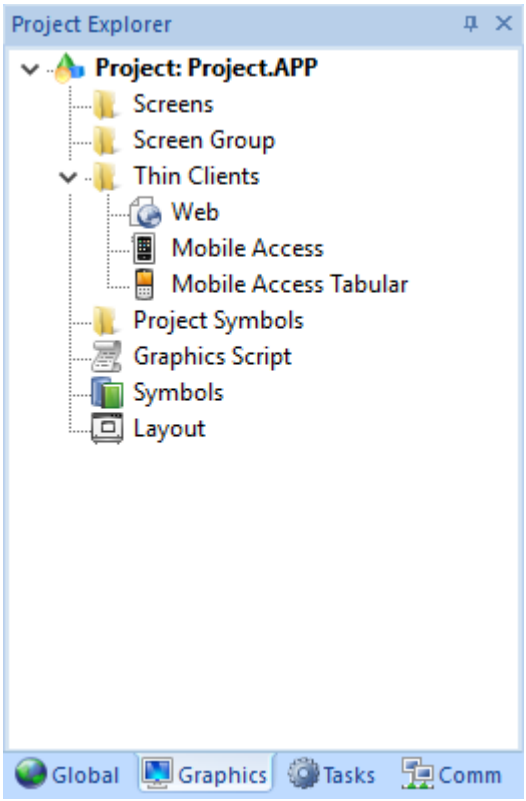
You will create additional tags as you build the project.

> **Tip:** You can sort the data in the *Project Tags* datasheet or insert/remove additional columns by right-clicking on it and then choosing the applicable option from the pop-up menu.

Creating the main screen

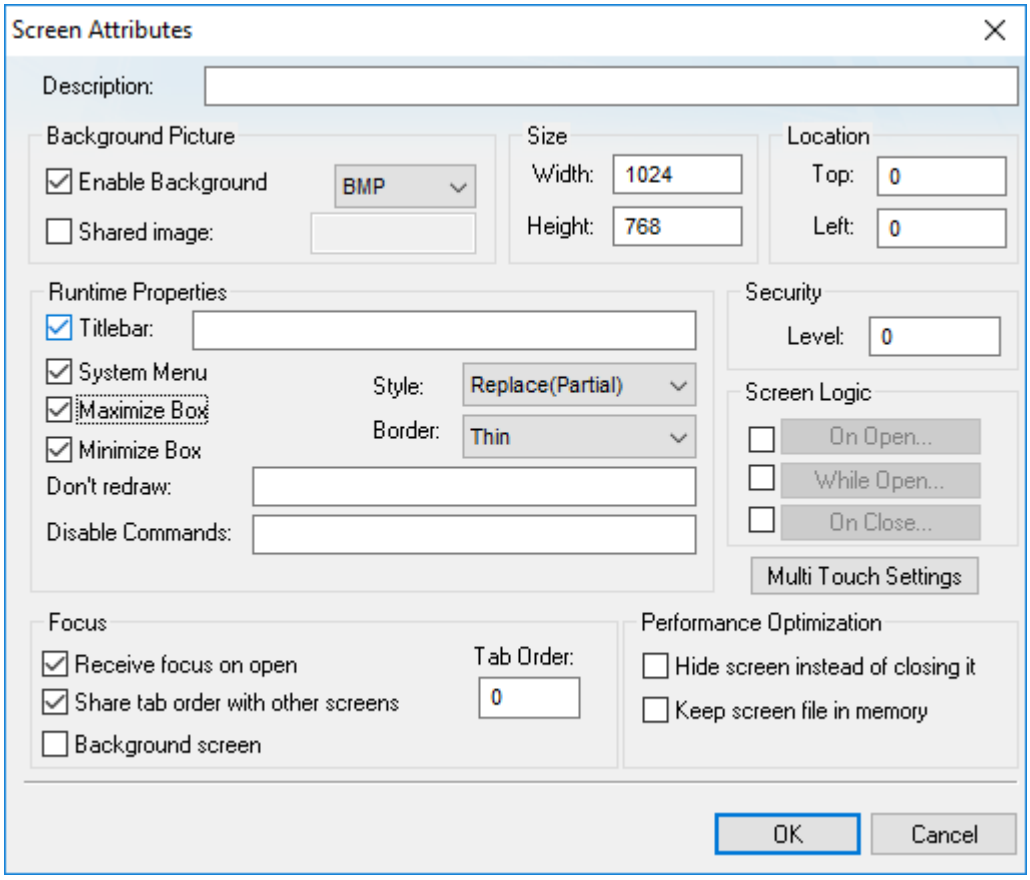
This part of the tutorial shows how to create your first screen, which will contain a single button that opens another screen.

1. In the *Project Explorer*, click the **Graphics** tab.



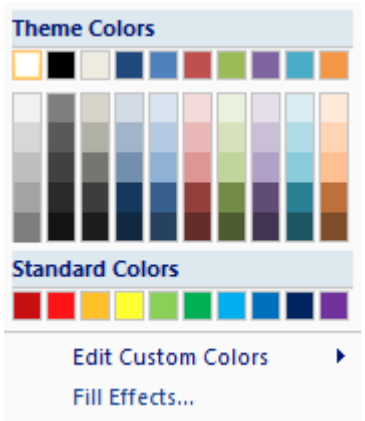
2. Right-click **Screens**, and then click **Insert** on the shortcut menu.
   The development application stores all screens created for a project in this **Screens** folder.

The *Screen Attributes* dialog is displayed.



Screen Attributes dialog

3. Use this dialog to set screen properties such as size and type.
   For this tutorial, click **OK** to accept the default settings.
   The *Screen Attributes* dialog is closed, and the new screen is opened in the workspace for editing.
4. On the **Graphics** tab of the ribbon, in the **Screen** group, click **Background Color**.
   A standard color picker is displayed.
5. In the color picker, select a light gray color.



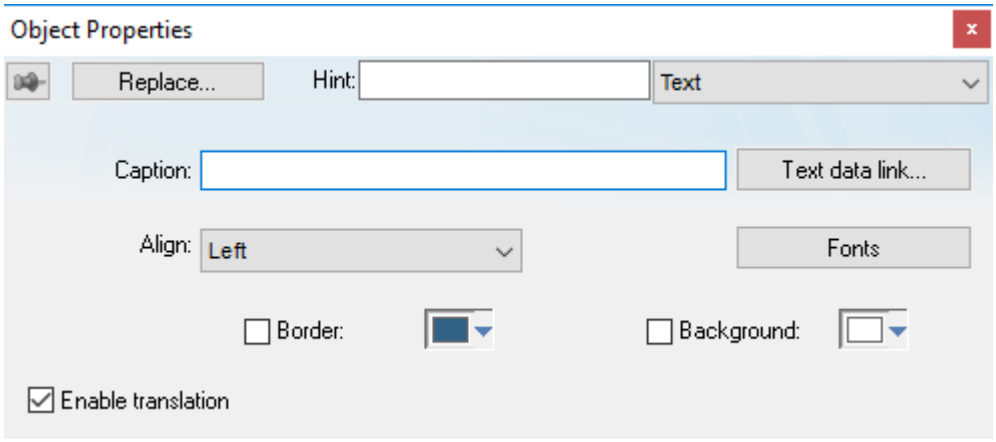
Color picker

That color is applied to the screen.

Drawing the main screen's title

This part of the tutorial shows how to draw the main screen's title using a Text object.

1. On the **Graphics** tab of the ribbon, in the **Active Objects** group, click **Text**.
   Your mouse cursor changes from an arrow to a crosshair.
2. Click on the screen, type `Welcome to the Tutorial Application`, and then press Return.
   This creates a new Text object with the specified text.
3. Double-click the object to open its *Object Properties* dialog.

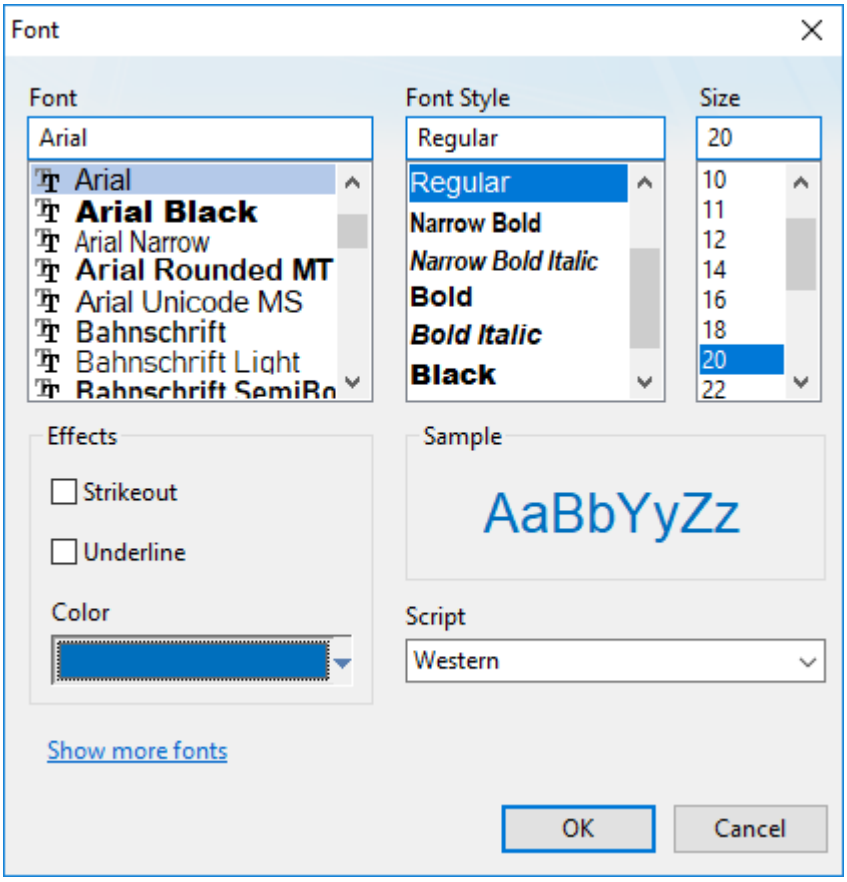


Object Properties: Text dialog

- Double-clicking on any screen object opens an *Object Properties* dialog containing the properties for that object. The properties shown in the dialog change depending on the type of object.
- The *Object Properties* dialog also contains a pin button that controls whether this dialog remains open. The button changes state (and function) each time you click on it, as follows:
  - When the pin button is released, the focus is passed to the object on the screen as soon as it is selected. It is recommended that this button is kept released when you want to manipulate the objects (Copy, Paste, Cut, or Delete). Although the *Object Properties* dialog is on the top, the keyboard commands (**Ctrl+C**, **Ctrl+V**, **Ctrl+X**, or **Del**) are sent directly to the objects.
  - When the pin button is pressed, the focus is kept on the *Object Properties* dialog, even when you click the objects on the screen. We recommend you keep this button pressed when you want to modify the settings of the objects. You can click an object and type the new property value directly in the *Object Properties* dialog (it is not necessary to click on the window to bring focus to it). Also, when the pin button is pressed, the *Object Properties* dialog does not automatically close when you click on the screen.

4. Click **Fonts** to open *Font* dialog, and then specify the font settings.
   For this tutorial...
   - Font is **Arial**
   - Font style is **Regular**
   - Size is **20**

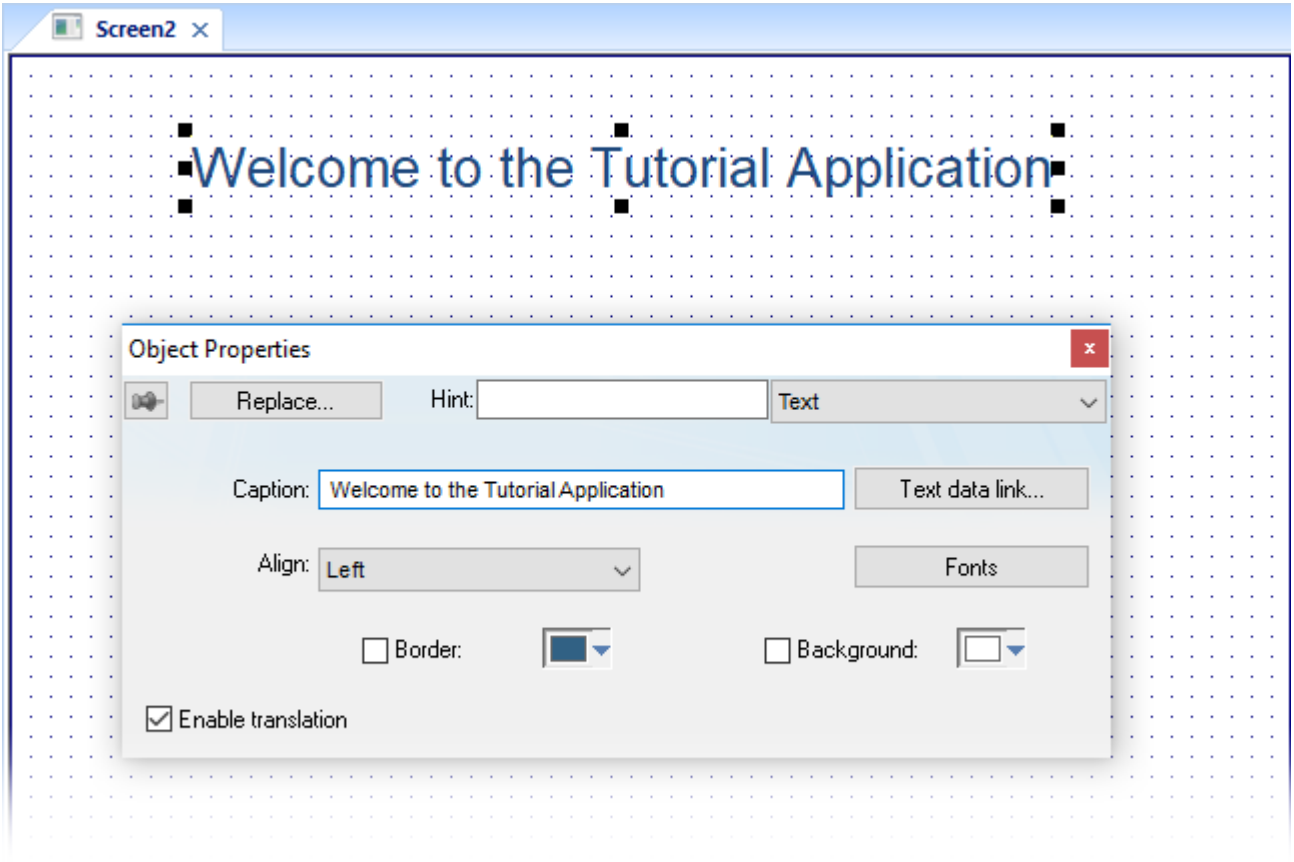
- Color is **Blue**



Specifying the font settings

5. Click **OK** to close the *Font* dialog.

The font settings are applied to the Text object.



Font settings applied to Text object

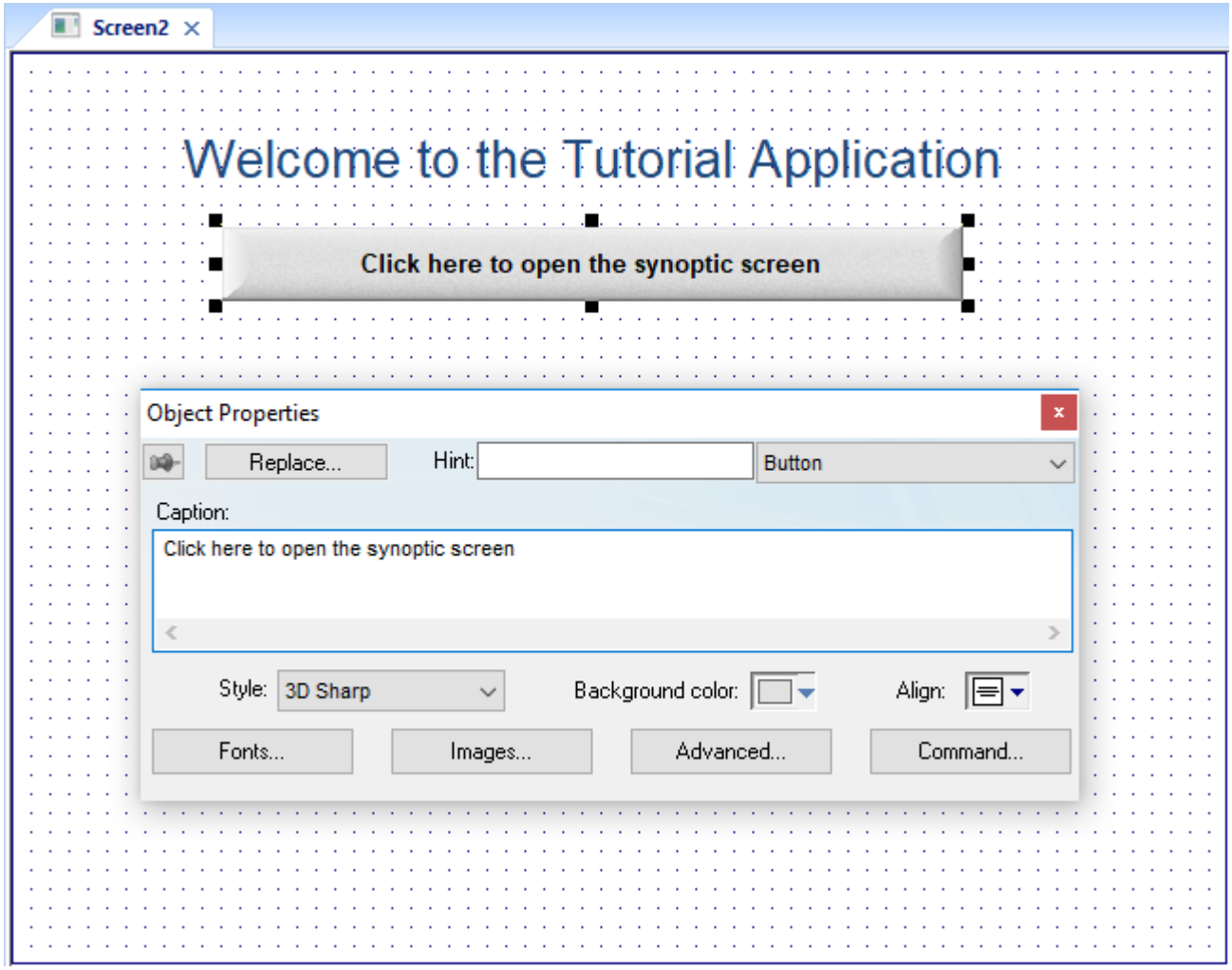
6. Close the *Object Properties* dialog (i.e., click the Close button in the dialog box's top-right corner).

Drawing a button to open another screen

This part of the tutorial shows how to draw and configure a button that will open another screen.

1. On the **Graphics** tab of the ribbon, in the **Active Objects** group, click **Button**.
   Your mouse cursor changes from an arrow to a crosshair.
2. Click and hold on the screen, and then drag the cursor to draw the Button object.
3. Double-click the object to open its *Object Properties* dialog.

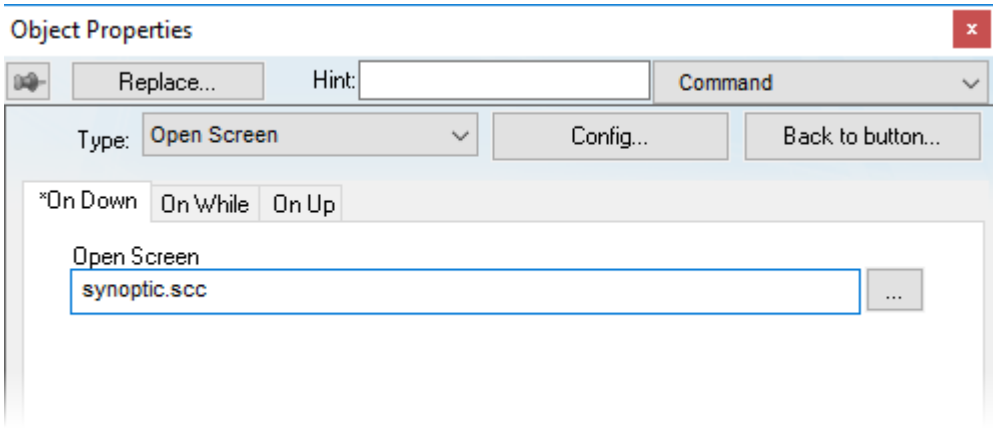
4. In the **Caption** box, type the following text: `Click here to open the synoptic screen`.



Adding a caption to the button

5. Click **Command**.
   The *Object Properties* dialog changes to show the properties for the Command animation.
6. In the **Type** list, select **Open Screen**.

7. In the **Open Screen** box, type `synoptic.scc`.



Configuring an Open Screen command on the button

   You can specify a screen that you have not yet created.

8. Close the *Object Properties* dialog.

Saving and closing the main screen

This part of the tutorial shows how to properly save and close a screen.

1. Click the Application button at the top-left of the development environment, and then click **Save** on the Application menu.
   A standard Windows *Save* dialog is displayed.
2. In the **File name** box, type `main`.
3. Click **Save**.
   The file is saved in your project folder (at `<project name>\Screen\main.scc`), and the *Save* dialog is closed.
4. Click the Application button at the top-left of the development environment, and then click **Close** on the Application menu.

Creating the synoptic screen

This part of the tutorial show how to create your second screen, which will include an animated tank of liquid and some basic controls for that tank.

1. In the **Graphics** tab of the *Project Explorer*, right-click the **Screens** folder, and then click **Insert** on the shortcut menu.
   The *Screen Attributes* dialog is displayed.
2. Use this dialog to set attributes such as size and type.
   For this tutorial, click **OK** to accept the default settings.
3. Click the Application button at the top-left of the development environment, and then click **Save As** on the Application menu.
   A standard Windows *Save As* dialog is displayed.
4. In the **File name** box, type `synoptic`.
5. Click **Save**.
   The file is saved in your project folder (at `<project name:gt;\Screen\synoptic.scc`), and the *Save* dialog is closed.

Do not close the screen like you did the main screen when you saved it. You still need to draw the synoptic screen.

Drawing the synoptic screen's title

As in a previous part, this part of the tutorial shows how to draw the synoptic screen's title using a Text object.

1. On the **Graphics** tab of the ribbon, in the **Active Objects** group, click **Text**.
2. Click on the screen, type `Synoptic Screen`, and then press Return.
3. Double-click the object to open its *Object Properties* dialog.
4. Click **Fonts** to open *Font* dialog, and then specify the font settings.
   For this tutorial...
   - Font is **Arial**
   - Font style is **Bold**
   - Size is **20**
   - Color is **Blue**
5. Click **OK** to save the font settings and close the dialog.
6. Close the *Object Properties* dialog.
7. Move the Text object to the top left corner of the screen.
8. Click the Application button at the top-left of the development environment, and then click **Save** on the Application menu.

This figure shows how your screen should look after you have drawn the screen title.



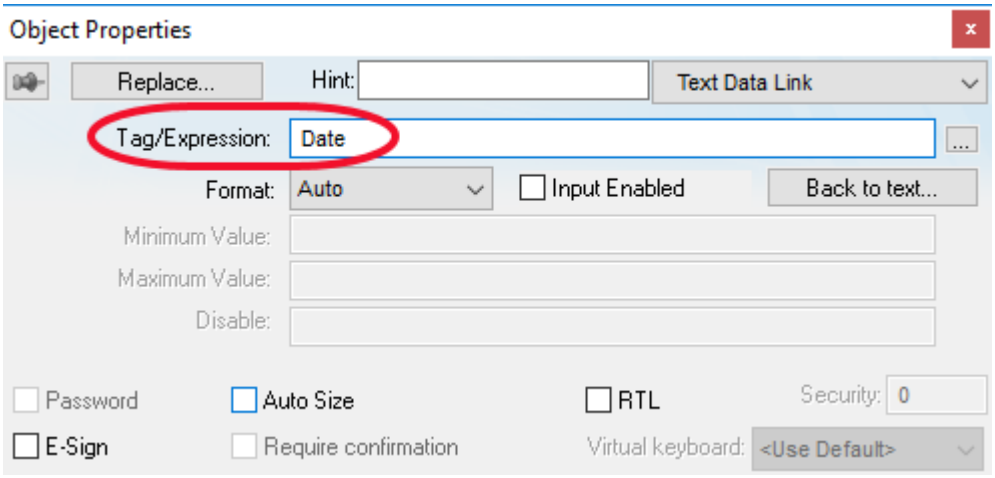
Finished screen title

Drawing "Date" and "Time" displays

This part of the tutorial shows how to draw "Date" and "Time" displays by linking Text objects to system tags.

`Date` and **`Time`** are system tags that hold the current date and time of the local station. These tags are available to any project.

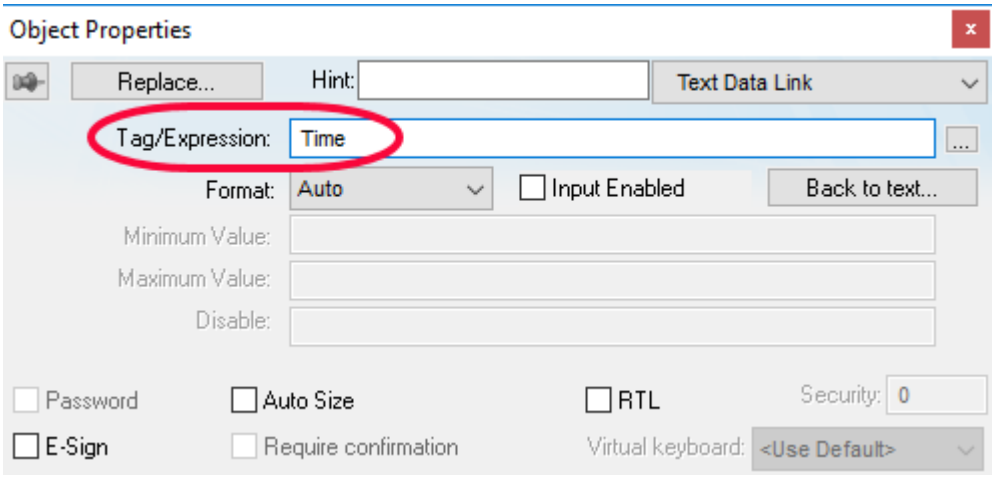
1. On the **Graphics** tab of the ribbon, in the **Active Objects** group, click **Text**.
2. Click on the screen, type `Date: ##########`, and then press Return.
3. Double-click the object to open its *Object Properties* dialog.
4. Click **Text Data Link**.
   The *Object Properties* dialog changes to show the properties for the Text Data Link animation.
5. In the **Tag/Expression** box, type `Date`.



Specifying the Date system tag

During runtime, the project replaces the **##########** characters of the Text object with the value of the system tag **Date**.

6. Close the *Object Properties* dialog.
7. On the **Graphics** tab of the ribbon, in the **Active Objects** group, click **Text**.
8. Click on the screen, type `Time: ##########`, and then press Return.
9. Double-click the object to open its *Object Properties* dialog.
10. Click **Text Data Link**.
    The *Object Properties* dialog changes to show the properties for the Text Data Link animation.
11. In the **Tag/Expression** box, type `Time`.



Specifying the Time system tag

During runtime, the project replaces the **##########** characters of the Text object with the value of the system tag **Time**.

12. Close the *Object Properties* dialog.
13. Click the Application button at the top-left of the development environment, and then click **Save** on the Application menu.

This figure shows how your screen should look after you have created the date and time objects.

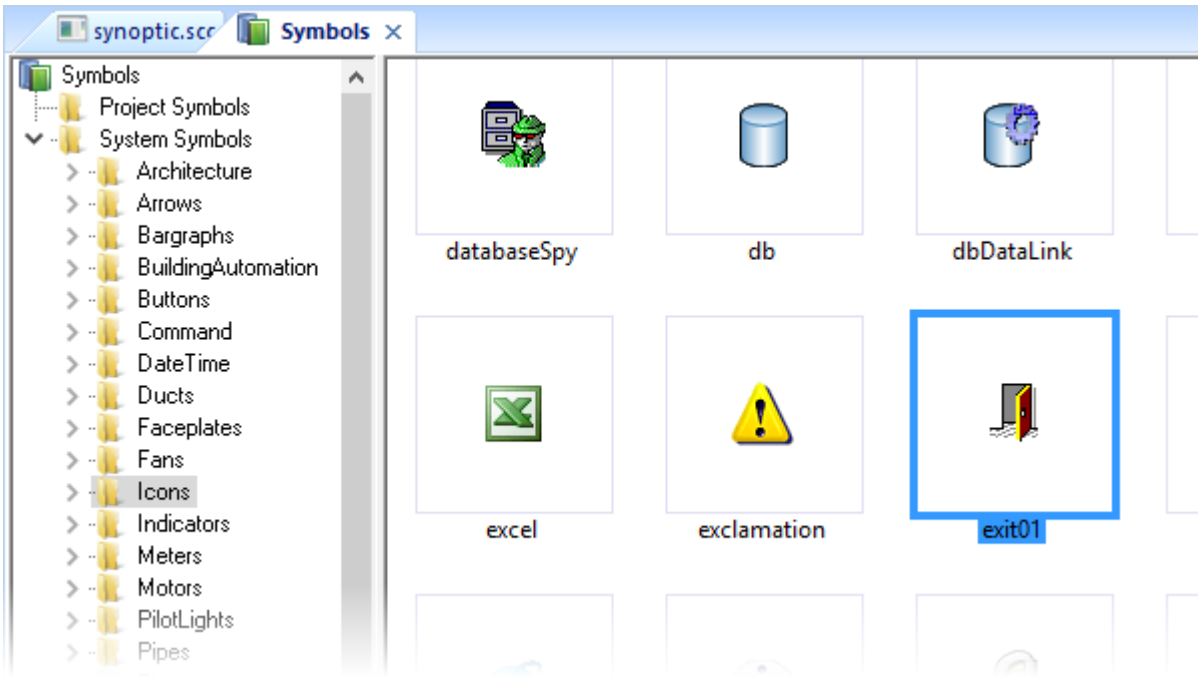


Finished date and time objects

Placing an "Exit" icon

This part of the tutorial shows how to place an icon (by selecting and configuring a Linked Symbol) that allows the user to exit the project, .

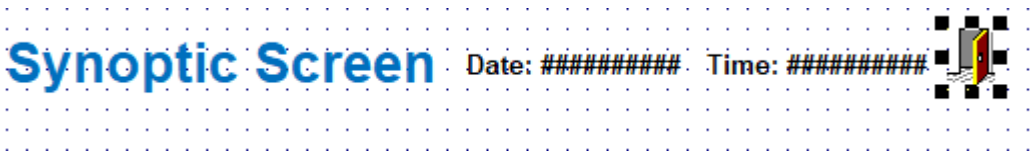
1. On the **Graphics** tab of the ribbon, in the **Libraries** group, click **Symbols**.
   The symbols library is displayed.
2. In the Symbols menu tree, open the **System Symbols** folder and then open the **Icons** sub-folder.
3. In the Icons sub-folder, select **exit01**.
   The symbol will be displayed in the symbol viewer to the right of the menu tree.



Selecting the "exit01" symbol

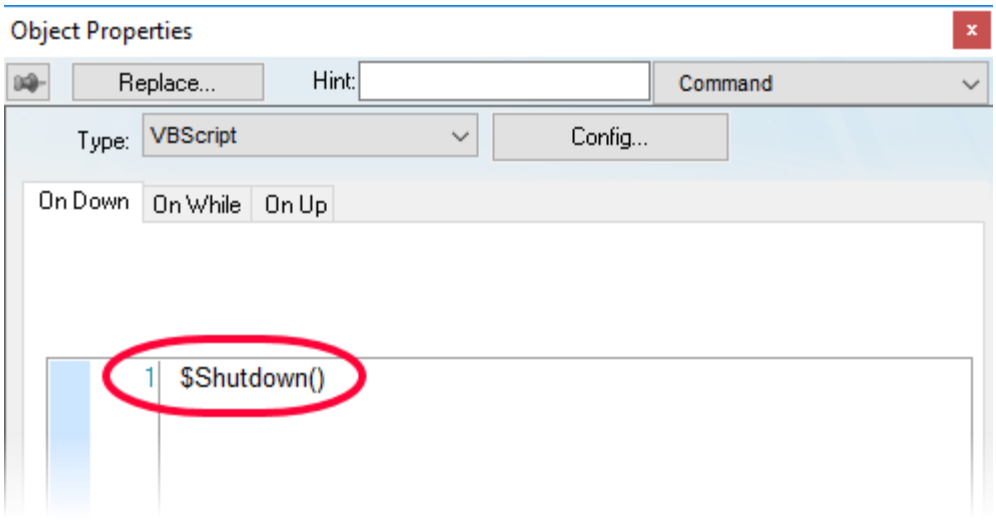
4. Click on the symbol.
   The mouse cursor will change to show that the symbol is ready to be placed in a screen.
5. Switch back to the screen where you want to place the symbol and then click in it.

The symbol is placed as a Linked Symbol object.



Placing the Linked Symbol object

6. With the object still selected, click **Command** (on the **Graphics** tab of the ribbon, in the **Animations** group) to apply this animation to the object.
7. Double-click the object to open its *Object Properties* dialog.
8. In the **Type** list, select **VBScript**.
9. In the **On Down** box, type `$Shutdown()`.
   `Shutdown` is one of EcoStruxure Machine SCADA Expert's built-in scripting functions, but it can be used within VBScript by prefacing it with a dollar sign (**$**).



Specifying the Shutdown command on the symbol

10. Close the *Object Properties* dialog.
11. Click the Application button at the top-left of the development environment, and then click **Save** on the Application menu.

Now, when a user clicks this icon during runtime, the project will stop and exit to the station's desktop.

Testing the project

This part of the tutorial shows how to test the project so far.

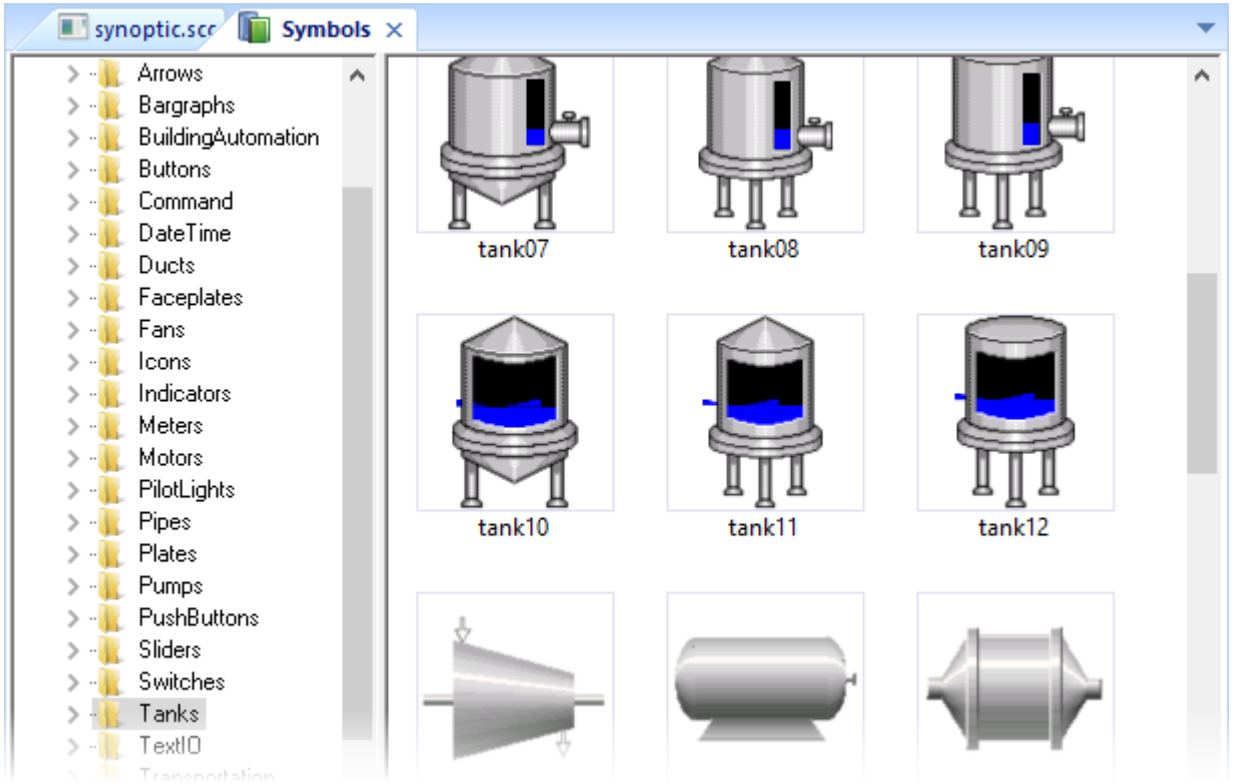
1. Click the Application button at the top-left of the development environment, and then click **Close** > **Close All** on the Application menu.
   All open worksheets are closed.
2. On the **Home** tab of the ribbon, in the **Local Management** group, click **Run**.
   The project runs and the startup screen is displayed.
3. Click the button to open the synoptic screen.
   The synoptic screen is displayed.
4. Click the exit icon to shut down the project.

If any part of the project does not work as expected, switch back to the development application (**ALT+TAB**) and then click **Stop** on the Home tab of the ribbon.

Placing an animated tank

This part of the tutorial shows how to select an animated tank from the Symbol Library and place it on the screen (similar to how you selected and placed the "Exit" icon), then associate some project tags with the tank's properties.

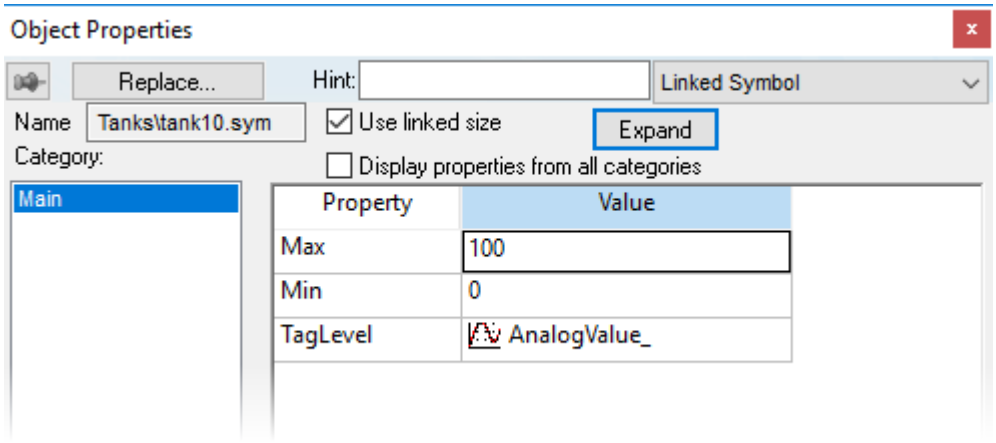
1. In the **Graphics** tab of the *Project Explorer*, expand the **Screens** folder.
2. Double-click **synoptic.scc**.
   The synoptic screen worksheet is reopened for editing.
3. On the **Graphics** tab of the ribbon, in the **Libraries** group, click **Symbols**.
4. In the Symbols menu tree, open the **System Symbols** folder and then open the **Tanks** sub-folder.
5. Browse the tank symbols and choose one.
   You may choose any tank symbol that you like; they all function basically the same.



Choosing a tank symbol

6. Click the symbol.
   The mouse cursor will change to show that the symbol is ready to be placed in a screen.
7. Switch back to the screen where you want to place the symbol and click in it.
   The symbol is placed as a Linked Symbol object.

8. Double-click the object to open its *Object Properties* dialog.



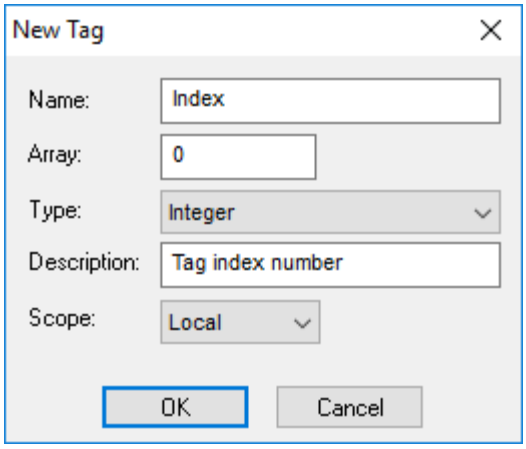
The tank symbol's properties

A tank is an arrangement of different objects and animations (for example a rectangle, a bar graph, etc.), all combined together as a Linked Symbol. You can modify the properties of this symbol by editing the properties list. For this tutorial, you will modify the tag associated with the tank level.

9. For the property **`TagLevel`**, delete the existing value and then type `Level[Index]`.
Note that you do not need to reopen the Project Tags datasheet to create tags as you develop the project. Because you have not previously created the tag **`Index`** in the Project Tags database, an alert message asks you if you would like to create it.

10. Click **Yes**.
A *New Tag* dialog is displayed.

11. Configure the new tag with **Array** as `0`, **Type** as `Integer`, and **Scope** as `Local`.



Configuring a new tag

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

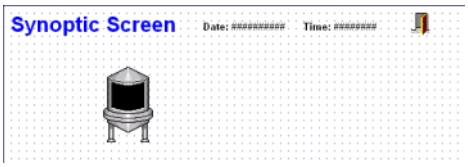
You can use the tag **`Index`** to set the array position of the tag **`Level`**, and show the level for any of the three tanks in the same object:

- When **`Index`** equals **`1`**, the tank object shows the level of Tank #1 (i.e., **`Level[1]`**);
- When **`Index`** equals **`2`**, the tank object shows the level of Tank #2 (i.e., **`Level[2]`**); and
- When **`Index`** equals **`3`**, the tank object shows the level of Tank #3 (i.e., **`Level[3]`**).

Also, because the tag scope is local, the tag can have different values for the Server and Client stations at the same time. Consequently, the local user (i.e., the Server station) can be monitoring the level of Tank #1 while the remote user (i.e., the Client station) is monitoring the level of Tank #2.

13. Close the *Object Properties* dialog.
14. Click the Application button at the top-left of the development environment, and then click **Save** on the Application menu.

This figure shows how your screen should look after you've created the tank object.

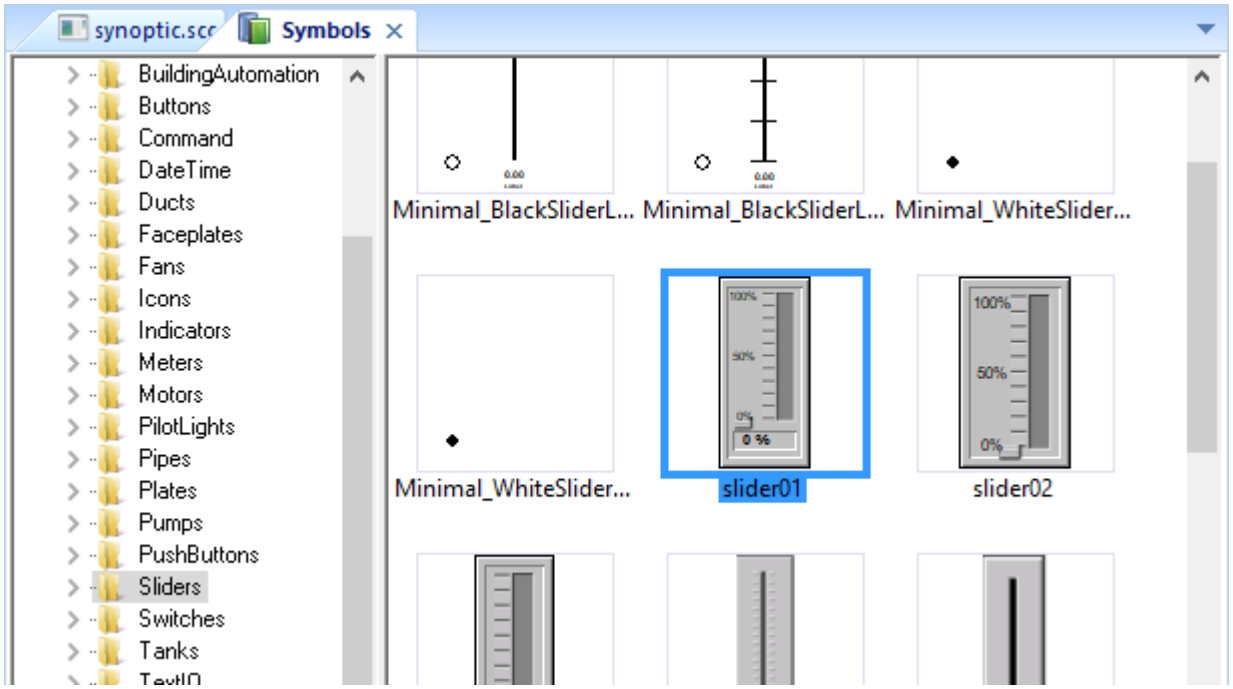


Finished tank object

Placing a level slider

This part of the tutorial shows how to select a slider control from the Symbol Library and then connect it to the animated tank.

1. On the **Graphics** tab of the ribbon, in the **Libraries** group, click **Symbols**.
2. In the Symbols menu tree, open the **System Symbols** folder and then open the **Sliders** sub-folder.

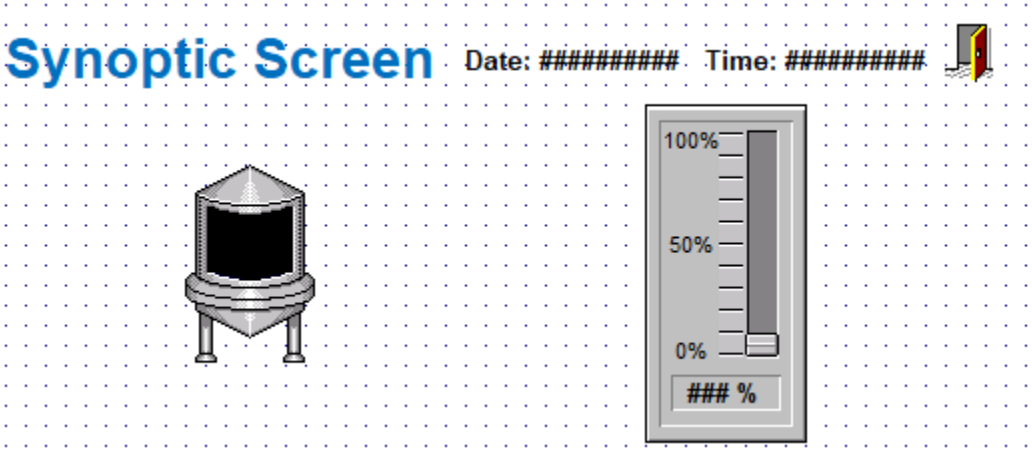


Selecting a slider symbol

3. In the Sliders sub-folder, select a slider control.
   You may select any slider you like; they all function basically the same way.
4. Click on the symbol.
   The mouse cursor will change to show that the symbol is ready to be placed in a screen.
5. Switch back to the screen where you want to place the symbol and click in it.
   The symbol is placed as a Linked Symbol object.

6. Double-click the object to open its *Object Properties* dialog.
7. For the property **TagName**, delete the existing value and then type `Level[Index]`.
   Just as with the tank, you need to modify the symbol property associated with the slider level.
8. Close the *Object Properties* dialog.
9. Click the Application button at the top-left of the development environment, and then click **Save** on the Application menu.

This figure shows how your screen should look after you've created the level slider object.



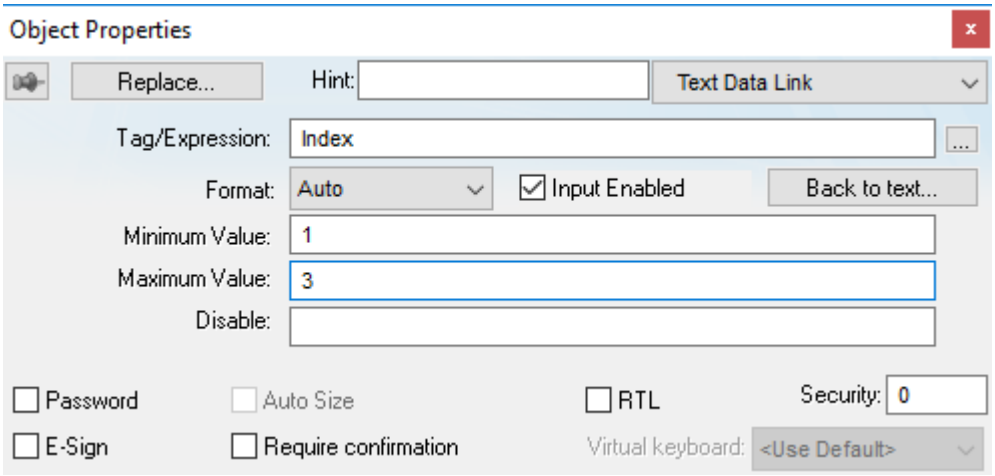
Finished level slider object

Drawing a tank selector

This part of the tutorial shows how to draw a text input box that can be used to change which real-world tank is represented by the animated tank on the screen.

1. On the **Graphics** tab of the ribbon, in the **Active Objects** group, click **Text**.
2. Click on the screen, type **`Tank: #`**, and then press Return.
3. Double-click the object to open its *Object Properties* dialog.
4. Click **Text Data Link**.
   The *Object Properties* dialog changes to show the properties for the Text Data Link animation.
5. In the **Tag/Expression** box, type `Index`.
6. Select the **Input Enabled** option.
   This allows the operator to enter a new value for the tag during runtime.
7. In the **Minimum Value** box, type `1`.

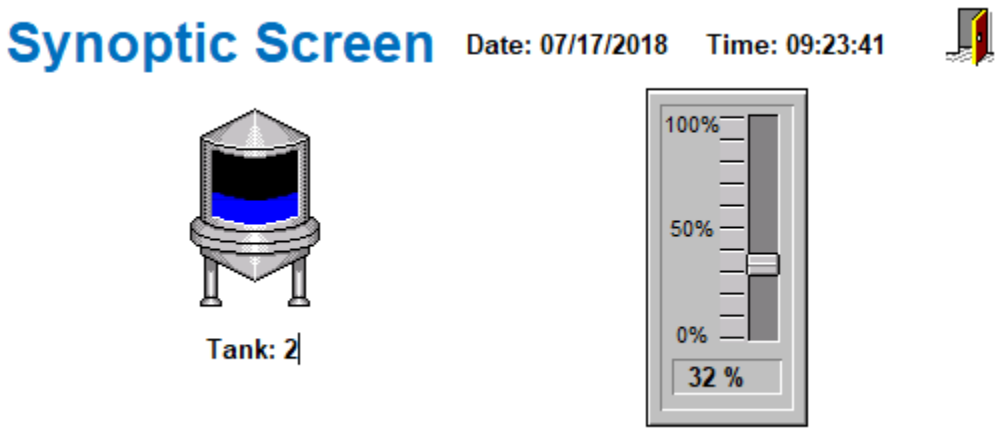
8. In the **Maximum Value** box, type `3`.



Configuring the "Tank" text input

9. Close the *Object Properties* dialog.
10. Click the Application button at the top-left of the development environment, and then click **Save** on the Application menu.

This figure shows how your screen should look after you've created the tank selector object.



Finished tank selector object during runtime

Testing the project

This part of the tutorial shows how to test the project again with the animated tank, the level slider, and the tank selector.

1. Click the Application button at the top-left of the development environment, and then click **Close** > **Close All** on the Application menu.
   All open worksheets are closed.
2. On the **Home** tab of the ribbon, in the **Local Management** group, click **Run**.
   The project runs and the startup screen is displayed.
3. Click the button to open the synoptic screen.
   The synoptic screen is displayed.
4. Type the tank number (1, 2, or 3) in the Tank label, and then use the slider to adjust the tank level.

Note that you can view/adjust the level of each tank independently.

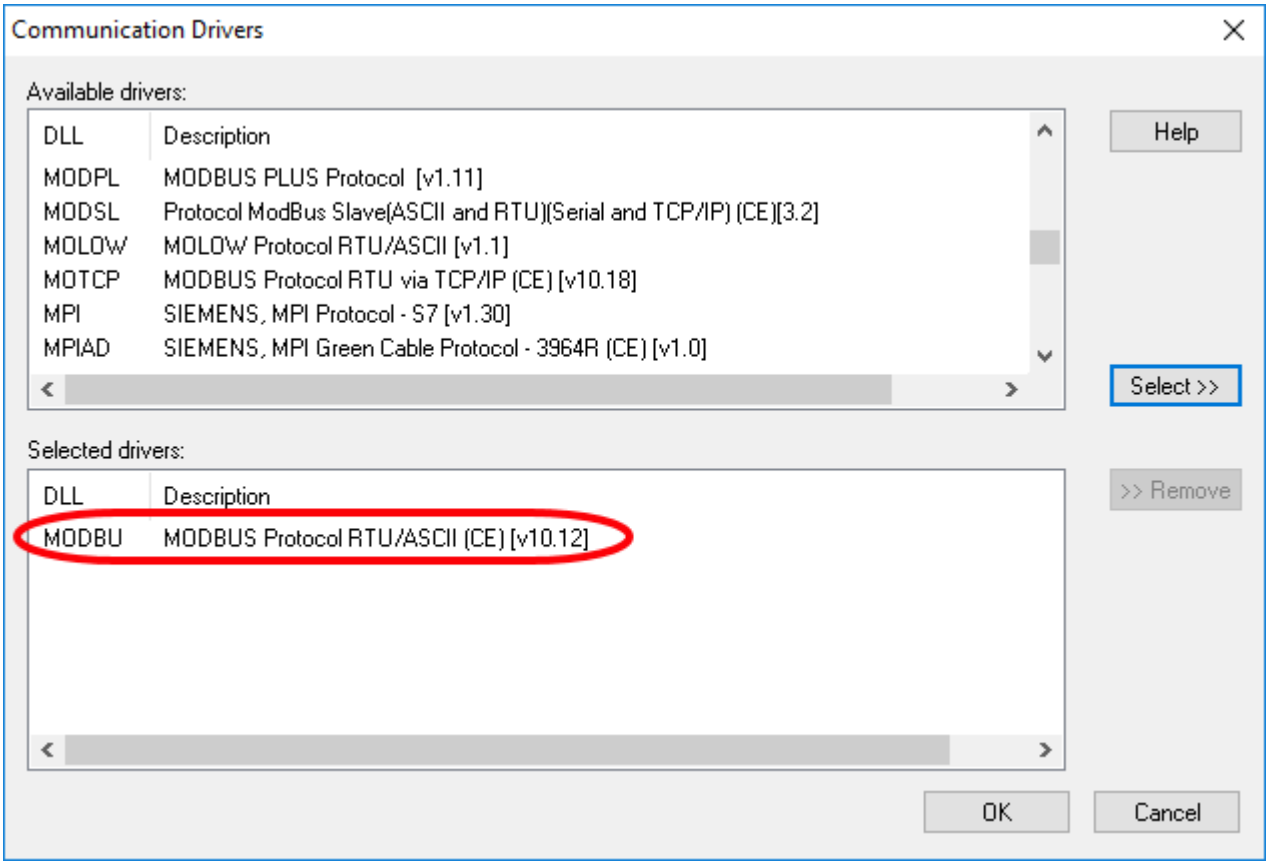
5. Click the exit icon to shut down the project.

If any part of the project does not work as expected, switch back to the development application (**ALT+TAB**) and then click **Stop** on the Home tab of the ribbon.

Configuring the communication driver

This part of the tutorial shows how to select and configure a driver to communicate with an external I/O device.

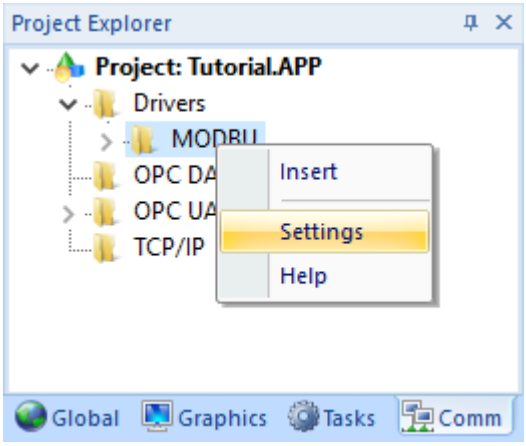
1. In the *Project Explorer*, click the **Comm** tab.
2. Right-click the **Drivers** folder, and the click **Add/Remove Drivers** on the shortcut menu.
   The *Communication Drivers* dialog is displayed.
3. Select a driver from the **Available drivers** list, and then click **Select**.
   For this tutorial, select MODBU.
   The driver is moved to the **Selected drivers** list.



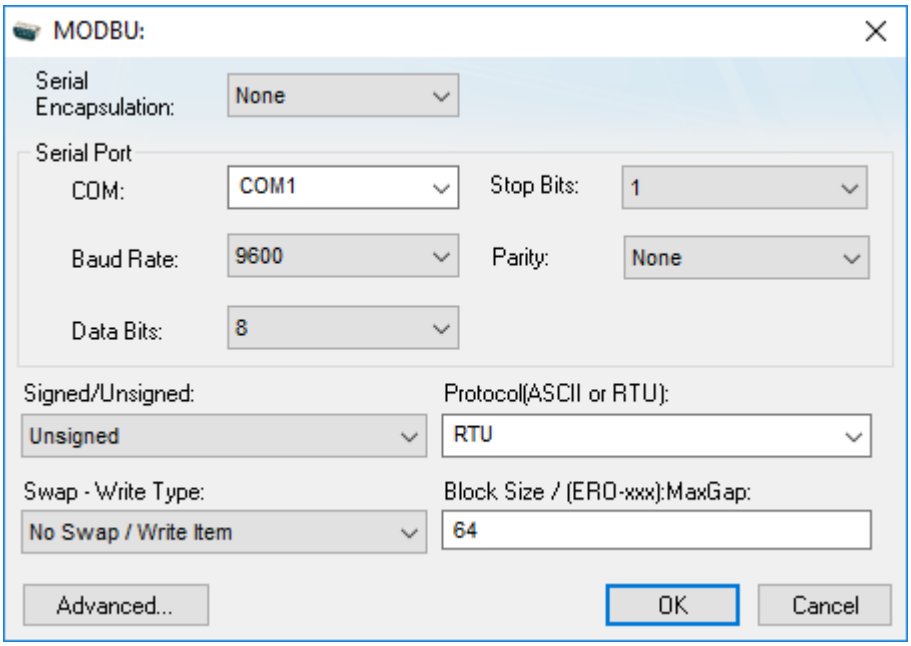
MODBU driver selected

4. Click **OK**.
   The *Communication Drivers* dialog is closed, and the driver is added to the Drivers folder in the *Project Explorer*.

5. In the Project Explorer, right-click the **MODBU** folder, and then click **Settings** on the shortcut menu.



The *Communication Settings* dialog is displayed.



Communication Settings dialog for MODBU driver

6. Configure the communication settings as needed for the target device.
   For this tutorial, accept the default settings.

> **Note:** For more information about a specific driver, click **Communication Drivers** on the **Help** tab of the ribbon.

7. Click **OK** to close the dialog.
8. In the Project Explorer, right-click the **MODBU** folder and then click **Insert** on the shortcut menu.
   A new driver worksheet named `MODBU001.drv` is created and opened for editing.
9. Configure the worksheet header:
   a) In the **Description** box, type `Tutorial Modbus`.
      This setting is for documentation only; it does not affect the runtime project in any way.
   b) In the **Enable Read When Idle** box, type `1`.

This setting is a trigger that takes a Boolean value. A value of 1 — either entered manually as above or evaluated from a tag/expression — forces your project to continue reading tag values from the target device even when there are no changes in value.

c) In the **Enable Write On Tag Change** box, type `1`.
This setting is also a trigger. A value of 1 forces your project to write tag values to the target device only when those values change, rather than continuously. This saves system resources and improves performance during rutime.

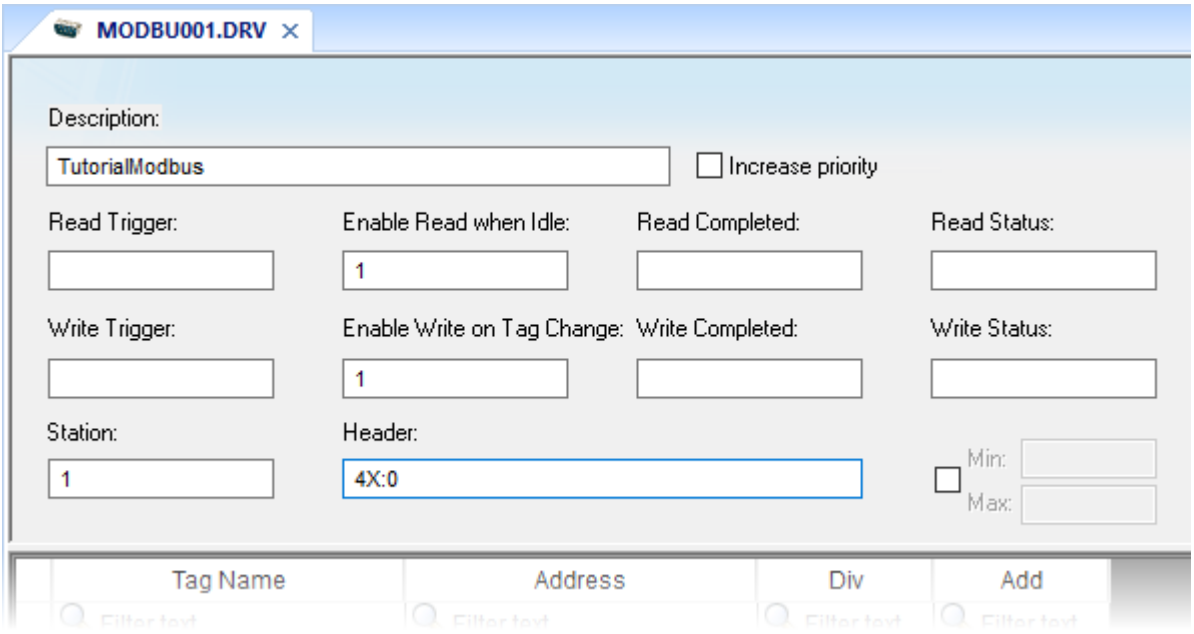
d) In the **Station** box, type `1`.
This indicates the I/O device number to be accessed by this driver. Typically, the PLC is specified as Device #1.

e) In the **Header** box, type `4X:0`.

You must use a driver-specific format. The format for the MODBU driver is:

register_type*:*initial_offset

Register Type	Description
0X	Coil Status
1X	Input Status
3X	Input Register
4X	Holding Register
ID	Slave ID Number



Completed worksheet header

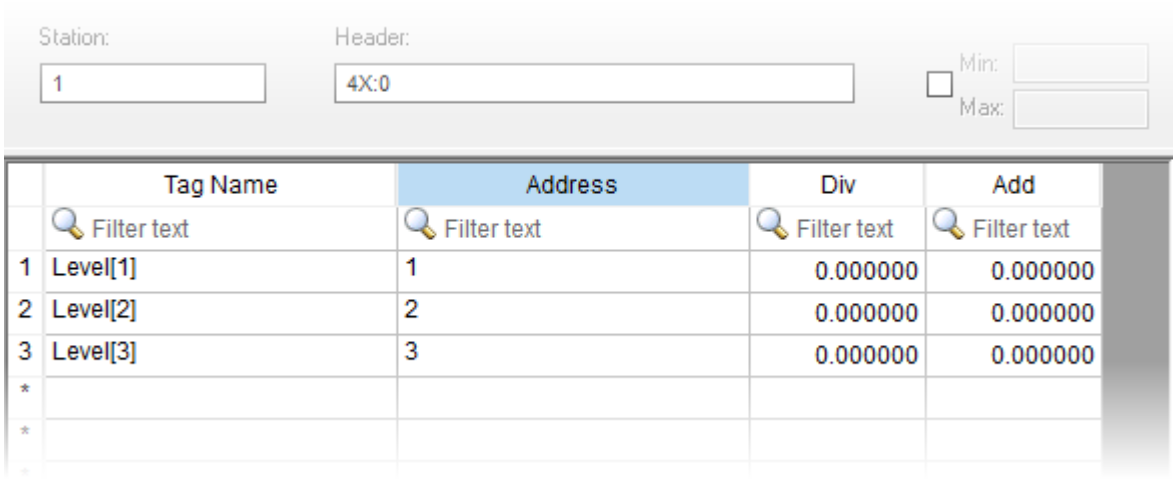
10. In the worksheet body, enter the tags and their associated device addresses — for each tag:

a) In the **Tag Name** field, type the name of the project tag.

b) In the **Address** field, type the value to be added to the header to form the complete device address.

Tag Name	Address	Complete Device Address
`Level[1]`	1	4X:1 (Holding Register 1)
`Level[2]`	2	4X:2 (Holding Register 2)

Tag Name	Address	Complete Device Address
Level[3]	3	4X:3 (Holding Register 3)



Completed worksheet body

11. Click the Application button at the top-left of the development environment, and then click **Save** on the Application menu.
12. When prompted to choose the driver sheet number, type `1` and then click **OK**.

Monitoring device I/O during runtime

This part of the tutorial shows how to monitor device I/O during runtime by using the *Log* window.

1. On the **Home** tab of the ribbon, in the **Local Management** group, click **Run**.
   The project runs and the startup screen is displayed.
2. Press **ALT+TAB** to switch back to the development application.
3. Right-click in the *Output* window, and then click **Settings**.
   The *Log Settings* dialog is displayed.
4. Select the **Field Read Commands**, **Field Write Commands**, and **Protocol Analyzer** options.
5. Click **OK** to close the *Log Settings* dialog.

You can now monitor the device I/O during runtime.