Point of View SCADA/HMI

Point of View Software Reference Guide
PV-SWREF-M

⚡ WARNING ⚡

Thank you for purchasing automation equipment from **Automationdirect.com**®, doing business as, **AutomationDirect**. We want your new automation equipment to operate safely. Anyone who installs or uses this equipment should read this publication (and any other relevant publications) before installing or operating the equipment.

To minimize the risk of potential safety problems, you should follow all applicable local and national codes that regulate the installation and operation of your equipment. These codes vary from area to area and usually change with time. It is your responsibility to determine which codes should be followed, and to verify that the equipment, installation, and operation is in compliance with the latest revision of these codes.

At a minimum, you should follow all applicable sections of the National Fire Code, National Electrical Code, and the codes of the National Electrical Manufacturer's Association (NEMA). There may be local regulatory or government offices that can also help determine which codes and standards are necessary for safe installation and operation.

Equipment damage or serious injury to personnel can result from the failure to follow all applicable codes and standards. We do not guarantee the products described in this publication are suitable for your particular application, nor do we assume any responsibility for your product design, installation, or operation.

Our products are not fault-tolerant and are not designed, manufactured or intended for use or resale as on-line control equipment in hazardous environments requiring fail-safe performance, such as in the operation of nuclear facilities, aircraft navigation or communication systems, air traffic control, direct life support machines, or weapons systems, in which the failure of the product could lead directly to death, personal injury, or severe physical or environmental damage ("High Risk Activities"). **AutomationDirect** specifically disclaims any expressed or implied warranty of fitness for High Risk Activities.

For additional warranty and safety information, see the Terms and Conditions section of our catalog. If you have any questions concerning the installation or operation of this equipment, or if you need additional information, please call us at 770-844-4200.

This publication is based on information that was available at the time it was printed. At **AutomationDirect** we constantly strive to improve our products and services, so we reserve the right to make changes to the products and/or publications at any time without notice and without any obligation. This publication may also discuss features that may not be available in certain revisions of the product.

Trademarks

This publication may contain references to products produced and/or offered by other companies. The product and company names may be trademarked and are the sole property of their respective owners. **AutomationDirect** disclaims any proprietary interest in the marks and names of others.

No part of this manual shall be copied, reproduced, or transmitted in any way without the prior, written consent of **Automationdirect.com® Incorporated**. **AutomationDirect** retains the exclusive rights to all information included in this document.

ADVERTENCIA

Gracias por comprar equipo de automatización de **Automationdirect.com**®. Deseamos que su nuevo equipo de automatización opere de manera segura. Cualquier persona que instale o use este equipo debe leer esta publicación (y cualquier otra publicación pertinente) antes de instalar u operar el equipo.

Para reducir al mínimo el riesgo debido a problemas de seguridad, debe seguir todos los códigos de seguridad locales o nacionales aplicables que regulan la instalación y operación de su equipo. Estos códigos varian de área en área y usualmente cambian con el tiempo. Es su responsabilidad determinar cuales códigos deben ser seguidos y verificar que el equipo, instalación y operación estén en cumplimiento con la revisión mas reciente de estos códigos.

Como mínimo, debe seguir las secciones aplicables del Código Nacional de Incendio, Código Nacional Eléctrico, y los códigos de (NEMA) la Asociación Nacional de Fabricantes Eléctricos de USA. Puede haber oficinas de normas locales o del gobierno que pueden ayudar a determinar cuales códigos y normas son necesarios para una instalación y operación segura.

Si no se siguen todos los códigos y normas aplicables, puede resultar en daños al equipo o lesiones serias a personas. No garantizamos los productos descritos en esta publicación para ser adecuados para su aplicación en particular, ni asumimos ninguna responsabilidad por el diseño de su producto, la instalación u operación.

Nuestros productos no son tolerantes a fallas y no han sido diseñados, fabricados o intencionados para uso o reventa como equipo de control en línea en ambientes peligrosos que requieren una ejecución sin fallas, tales como operación en instalaciones nucleares, sistemas de navegación aérea, o de comunicación, control de tráfico aéreo, máquinas de soporte de vida o sistemas de armamentos en las cuales la falla del producto puede resultar directamente en muerte, heridas personales, o daños físicos o ambientales severos ("Actividades de Alto Riesgo"). **Automationdirect.com** específicamente rechaza cualquier garantía ya sea expresada o implicada para actividades de alto riesgo.

Para información adicional acerca de garantía e información de seguridad, vea la sección de Términos y Condiciones de nuestro catálogo. Si tiene alguna pregunta sobre instalación u operación de este equipo, o si necesita información adicional, por favor llámenos al número 770-844-4200 en Estados Unidos.

Esta publicación está basada en la información disponible al momento de impresión. En **Automationdirect.com** nos esforzamos constantemente para mejorar nuestros productos y servicios, así que nos reservamos el derecho de hacer cambios al producto y/o a las publicaciones en cualquier momento sin notificación y sin ninguna obligación. Esta publicación también puede discutir características que no estén disponibles en ciertas revisiones del producto.

Marcas Registradas

Esta publicación puede contener referencias a productos producidos y/u ofrecidos por otras compañías. Los nombres de las compañías y productos pueden tener marcas registradas y son propiedad única de sus respectivos dueños. Automationdirect.com, renuncia cualquier interés propietario en las marcas y nombres de otros.

AVERTISSEMENT

Nous vous remercions d'avoir acheté l'équipement d'automatisation de **Automationdirect.com®**, en faisant des affaires comme, **AutomationDirect**. Nous tenons à ce que votre nouvel équipement d'automatisation fonctionne en toute sécurité. Toute personne qui installe ou utilise cet équipement doit lire la présente publication (et toutes les autres publications pertinentes) avant de l'installer ou de l'utiliser.

Afin de réduire au minimum le risque d'éventuels problèmes de sécurité, vous devez respecter tous les codes locaux et nationaux applicables régissant l'installation et le fonctionnement de votre équipement. Ces codes diffèrent d'une région à l'autre et, habituellement, évoluent au fil du temps. Il vous incombe de déterminer les codes à respecter et de vous assurer que l'équipement, l'installation et le fonctionnement sont conformes aux exigences de la version la plus récente de ces codes.

Vous devez, à tout le moins, respecter toutes les sections applicables du Code national de prévention des incendies, du Code national de l'électricité et des codes de la National Electrical Manufacturer's Association (NEMA). Des organismes de réglementation ou des services gouvernementaux locaux peuvent également vous aider à déterminer les codes ainsi que les normes à respecter pour assurer une installation et un fonctionnement sûrs.

L'omission de respecter la totalité des codes et des normes applicables peut entraîner des dommages à l'équipement ou causer de graves blessures au personnel. Nous ne garantissons pas que les produits décrits dans cette publication conviennent à votre application particulière et nous n'assumons aucune responsabilité à l'égard de la conception, de l'installation ou du fonctionnement de votre produit.

Nos produits ne sont pas insensibles aux défaillances et ne sont ni conçus ni fabriqués pour l'utilisation ou la revente en tant qu'équipement de commande en ligne dans des environnements dangereux nécessitant une sécurité absolue, par exemple, l'exploitation d'installations nucléaires, les systèmes de navigation aérienne ou de communication, le contrôle de la circulation aérienne, les équipements de survie ou les systèmes d'armes, pour lesquels la défaillance du produit peut provoquer la mort, des blessures corporelles ou de graves dommages matériels ou environnementaux («activités à risque élevé»). La société **AutomationDirect** nie toute garantie expresse ou implicite d'aptitude à l'emploi en ce qui a trait aux activités à risque élevé.

Pour des renseignements additionnels touchant la garantie et la sécurité, veuillez consulter la section Modalités et conditions de notre documentation. Si vous avez des questions au sujet de l'installation ou du fonctionnement de cet équipement, ou encore si vous avez besoin de renseignements supplémentaires, n'hésitez pas à nous téléphoner au 770-844-4200.

Cette publication s'appuie sur l'information qui était disponible au moment de l'impression. À la société **AutomationDirect**, nous nous efforçons constamment d'améliorer nos produits et services. C'est pourquoi nous nous réservons le droit d'apporter des modifications aux produits ou aux publications en tout temps, sans préavis ni quelque obligation que ce soit. La présente publication peut aussi porter sur des caractéristiques susceptibles de ne pas être offertes dans certaines versions révisées du produit.

Marques de commerce

La présente publication peut contenir des références à des produits fabriqués ou offerts par d'autres entreprises. Les désignations des produits et des entreprises peuvent être des marques de commerce et appartiennent exclusivement à leurs propriétaires respectifs. **AutomationDirect** nie tout intérêt dans les autres marques et désignations.

Notes:

Point of View Software Reference Guide

AUTOMATIONDIRECT.com

Please include the Manual Number and the Manual Issue, both shown below, when communicating with Technical Support regarding this publication.

Manual Number:	**PV-SWREF-M**
Issue:	**1st Edition**
Issue Date:	**11/13**

Publication History

Issue	Date	Description of Changes
1st Edition	11/13	Original

Notes:

Table of Contents

Chapter 1

Introduction

Introduction

Point of View (or POV, for short) is a powerful, integrated tool that exploits key features of Microsoft operating systems and enables you to build fullfeatured SCADA (Supervisory Control and Data Acquisition) or HMI (Human- Machine Interface) programs for your industrial automation business. This Point of View Quick Start Guide is intended for individuals using POV for the first time. This publication will help you quickly familiarize yourself with the basic functions of POV.

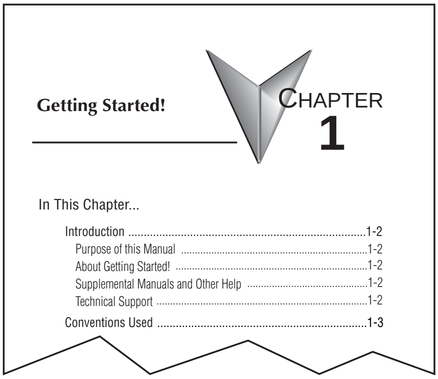
Conventions Used

When you see the "note pad" icon in the left-hand margin, the paragraph to its immediate right will be a special note. Notes represent information that may make your work quicker or more efficient. The word NOTE: in boldface will mark the beginning of the text.

When you see the "exclamation point" icon in the left-hand margin, the paragraph to its immediate right will be a warning. This information could prevent injury, loss of property, or even death in extreme cases. Any warning in this manual should be regarded as critical information that should be read in its entirety. The word WARNING in boldface will mark the beginning of the text.

Key Topics for Each Chapter

The beginning of each chapter will list the key topics that can be found in that chapter.



Getting Started!

CHAPTER 1

In This Chapter...

About This Software

Point of View is powerful software for developing HMI, SCADA, and OEE/Dashboard projects that can be deployed anywhere.

Each POV project consists of:

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

Product Features

Alarms:
In addition to all the alarm functions you'd expect, POV v7.1+SP2 also sends alarms using multi-media formats like PDF. Use remote notification to have alarms sent right to your inbox, a printer, or a smartphone. Alarms are real-time and historical, log data in binary format or to any database.

Animation:
POV 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 configurations. 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 any SQL database (MS SQL, MySQL, Sybase, Oracle), or MS Access, or Excel, and ERP/MES systems (including SAP). Flexible enough to have a built in interface without the need for knowing SQL (for trends, alarms/events, grid and other objects), or use any SQL statement you need anywhere you need it.

Drivers:
POV v7.1+SP2 contains 18 built-in drivers for most PLCs.

Email:
Send email using SMTP to desktop, email enabled phone, or any enabled device. Get real-time information on alarms, process values, and other events. Point of View v7.1+SP2 supports SSL encryption allowing the use of web based providers such as Gmail.

Events:
POV v7.1+SP2 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 functionality to create 21 CFR part 11 compliant projects with traceability and e-signatures. These features are often used for pharmaceutical and food applications, but also for any application where traceability is a must.

FTP:
Automatically upload or download files during runtime to/from remote storage locations using FTP protocol 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 library for future use, or 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 and 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 application in one of many development languages, including English, Portuguese, German and French.

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

OPC:
Drivers for several major brands of PLCs are built in, but any OPC server may optionally be used. POV supports OPC DA (Server/Client), OPC HDA(Server), UA (Client) and OPC .NET 3.0 (Client).

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, POV v7.1+SP2 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 everywhere. Take an application created for Windows XP and deploy it on any supported Microsoft operating system, including Windows 7, Windows 8, and Server editions.

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 Automationdirect.com 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:
POV includes support for group and user accounts, esignatures, and traceability, as well as support for the ADAM Server, in addition to standard LDAP Servers. Integrate your project to the Active Directory (Users and Groups).

SSL Support for Emails:
Native support for Secure Socket Layer (SSL), makes it easy and secure to send emails from Point of View using web based tools such as Gmail.

Standards:
Take advantage of common industry standards to develop applications that are compatible with any format. TCP/IP, .Net, ActiveX, OPC (client and server), ADO/ODBC, COM/DCOM, OLE, DDE, XML, SOAP, and HTML are 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:
Included library 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.

Tag Database:
POV features an object oriented database with boolean, integer, real, strings, arrays, classes (structures), indirect tags and included system tags.

Thin Clients:
Remotely view screens as web pages using Internet Explorer web browser, or Automationdirect.com Secure Viewer. Use SMA (Studio Mobile Access) to monitor or access process values and alarms with remote devices such as tablets and mobile phones. Enhanced SMA offers data in easy-to-read widgets that can be viewed on any WebKit (HTML5) based web browser found on iPads, and Android phones and tablets.

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 application.

Troubleshooting:
Quickly debug and verify a project using local and remote tools for troubleshooting, including status fields, DatabaseSpy 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.

Comparison of Point of View Software Components

The Point of View software suite actually comprises several individual components that can be installed on different computers to perform different functions.

Comparison of Point of View Software Components

Component	Functions	Platforms
Point of View	• Project development • Tag integration • Remote management • Project runtime server • Project runtime client	• Windows • Windows Server • Windows Embedded Standard
Web Tunneling Gateway (WTG)	• Enables a public-facing web server to pass data between your project runtime server and web thin clients, when your project runtime server is located on a secure, internal network	• Microsoft IIS for Windows
Mobile Access Runtime	• Enables your project runtime server to deliver HTML5-enhanced project screens to tablets and smartphones	• Microsoft IIS for Windows
Automationdirect.com Thin Client (a.k.a. Secure Viewer)	• Web thin client (standalone application)	• Windows • Windows Server
ISSymbol	• Web thin client (browser plug-in)	• Internet Explorer for Windows

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

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

There are separate, downloadable installers only for the web thin clients (i.e., Secure Viewer and ISSymbol).

Install the Full Point of View Software

Install the full Point of View software on your Windows computer in order to develop POV projects, or to use the computer as a project runtime server and/or client.
To install and run the full Point of View software, you must have:

1) A Windows-compatible computer with a standard keyboard, pointer input (e.g., mouse, trackpad, or touchscreen) and SVGA-minimum display
2) A Windows, Windows Server, or Windows Embedded Standard operating system that is currently supported by Microsoft, which at this time includes:
   - Microsoft Windows XP Service Pack
   - Microsoft Windows Vista Service Pack 2
   - Microsoft Windows 7 Service Pack 1
   - Microsoft Windows 8
   - Microsoft Windows Server 2003 Service Pack 2
   - Microsoft Windows Server 2008 Service Pack 2
   - Microsoft Windows Server 2008 R2 Service Pack 1
   - Microsoft Windows Server 2012
3) Microsoft .NET Framework 3.5.1 (see note below);
4) Microsoft Internet Explorer 6.0 or later;
5) 2 GB free hard drive space or non-volatile memory; and
6) An Ethernet or Wi-Fi network adapter, for TCP/IP networking or Serial COM ports and adapters to be used for direct communication with PLCs and other devices.
7) A USB port, to be used with hardkey licensing.

We recommend the Home Premium, Professional, Enterprise, and Ultimate editions of Windows because they include Microsoft Internet Information Services (IIS) as a pre-installed feature that may be turned on. IIS is used to make your projects accessible to web thin clients and mobile devices. We do not recommend the Starter and Home Basic editions because they do not include IIS, but you may still use them if you do not plan to use those features.

The following items are optional but recommended:

- A DVD-ROM drive, to install the software from disc. This is optional because you may also download the installer to your computer.
- Microsoft IIS installed and turned on, if you want to make your projects accessible to mobile devices. For more information, see the description of Mobile Access Runtime later in this chapter. This is optional because you may choose not to install the Mobile Access Runtime feature.

NOTE: *You must have Administrator privileges on the computer in order to install software.*

NOTE: *Only Windows 7, Windows 8, Windows Server 2008 R2, and Windows Server 2012 are under what Microsoft calls "mainstream support", which means they are actively maintained and additional service packs may be released for them in the future. The rest of the listed operating systems are under what Microsoft calls "extended support", which means they are no longer actively maintained.*

NOTE: *We do not support Point of View on Windows operating systems that Microsoft is not supporting or on XP 64 bit.*

NOTE: *Microsoft .NET Framework 3.5.1 is a pre-installed feature on newer versions of Windows, but it may not be turned on by default. To turn it on in Windows Vista, Windows Server 2008, Windows 7, or Windows Server 2008 R2, use the Programs and Features control panel (Control Panel > Programs > Programs and Features > Turn Windows features on and off).*
On older versions of Windows, Microsoft .NET Framework 3.5.1 must be installed separately. The Point of View installer will attempt to do this for you (see Step 2 below), but depending on your computer's security settings, the installation may fail without notice. If you experience problems later, while trying to run Point of View, then you should at least confirm that Microsoft .NET Framework 3.5.1 was installed successfully. To do this in Windows XP or Windows Server 2003, use the Add/Remove Programs control panel (Control Panel > Add/Remove Programs).
Please note that Microsoft .NET Framework 4 does not include Microsoft .NET Framework 3.5.1.

To install the full Point of View software:

1) Do one of the following:

   - Download the zipped installer to your computer, either from our website (www.automationdirect.com) or from another location on your network where you have previously saved it. Extract the zip archive, open the resulting folder, and then locate and double-click setup.exe.
   - Insert the installation disc into your DVD-ROM drive. If it does not autorun, then manually run D:\automationdirect.htm. When the page opens in your browser, follow the instructions for Product Installation.

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

2) Select a language from the list, and then click OK.

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

   The wizard prepares for installation. During this step, it automatically installs SafeNet's Sentinel drivers.

3) On the Welcome page of the wizard, click Next to proceed with the installation.

4) On the License Agreement page, click either Yes to accept the agreement and proceed or No to refuse the agreement and exit the wizard.

5) On the Customer Information page, enter your user name and company name, and then click Next.

6) 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: C:\Program Files\AutomationDirect\Point of View v7.1\ (Program Files X86 if 64 bit operating system).

7) On the Select Features page, select which software features and components you want to install on your computer, and then click Next. The following features are available:

   - Program Files - The main program files for the project development application, the project runtime server, and the project runtime client. **This feature cannot be deselected**.
   - Demos - Premade projects that demonstrate the capabilities of the Point of View software.
   - Hardkey Support - Additional drivers to support the use of hardkey (USB port) software licenses. **This feature cannot be deselected**.
   - OPC Components - Additional components to communicate with other OPC compatible devices. This includes OPC DA (a.k.a. OPC Classic), OPC UA, OPC .NET (a.k.a. OPC Xi), and OPC XML-DA.
   - PDF Printing - Software for printing run-time reports directly to PDF files.
   - Security System Device Driver - An additional keyboard driver to control user input for project security.
   - Symbol Library - A library of premade but configurable screen objects such as pushbuttons, toggle switches, gauges, dials, indicator lights, and so on.
   - Windows Mobile - Additional runtime software for older Windows Mobile devices.
   - Mobile Access Runtime - Additional software to automatically configure IIS to make your project runtime accessible to mobile devices such as tablets and smartphones. This feature requires that you have IIS turned on and configured with ASP, ASP.NET, and ISAPI Extensions enabled. The Point of View installation wizard will attempt to verify that you do, and if you do not, then it will not install this feature.
     For more information, see the "Turn on Microsoft IIS for thin client access over the Internet" help file topic.

     You do not need to install this feature at this time. You may install it later, after you have turned on IIS, or you may install it on another computer that is acting as your project runtime server. There is a separate Mobile Access Runtime installer (MobileAccessSetup.exe) that is unpacked with the rest of the software.

***NOTE:** To use Mobile Access Runtime, your Point of View software license must have the necessary add-on. You may still develop projects that include Mobile Access features, even without the license add-on, but mobile devices will not be able to access them.*

8) On the Ready To Install page, click Install and the installation proceeds.

9) On the final page, click Finish and the wizard closes.

When you have finished the installation, you should find the Point of View software in your Start menu at Start > All Programs > AutomationDirect > Point of View v7.1. It includes the following components:

- Point of View v7.1+SP2 - The project development application, project runtime server, and project runtime client.

- Register - A utility that manages your Point of View software license.
- StartUp - A shortcut that automatically starts the project runtime server and runs the most recent project.
- Remote Agent - A utility that allows Point of View running on other computers to connect to your computer and send projects to it.
- Quick Start Guide - A brief guide to installing and using the project development application, including a tutorial for building a simple project.
- Help Manual - A complete technical reference and user guide for the Point of View software.
- Release Notes - A list of changes in the the Point of View software.
- There should also be a shortcut icon on your desktop.

NOTE: *You may encounter 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: http://consumerdocs.installshield.com/selfservice/viewContent.do?externalId=Q108464*

To run the project development application, do one of the following:

- Double-click the shortcut icon on the desktop
- Click, Start > All Programs > AutomationDirect > Point of View v7.1 > Point of View v7.1.

Execution Modes

Evaluation Mode

NOTE: When Windows UAC (User Account Control) is enabled, you will need to run POV as administrator (at least once) to enable Evaluation Mode (otherwise, it will not have the rights to access the Windows Registry, which is required for the Evaluation Mode). Please right-click on the POV icon (from the Desktop) and select the option "Run as administrator".

Enables all of the product's engineering and runtime features. The first time you install Point of View 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 install a valid licensed hardkey. You cannot reactivate Evaluation Mode, even if you reinstall the software on your computer.

NOTE: Each version of Point of View has an evaluation period that is independent of every other version. For example, if your Point of View v7.0 evaluation period has expired and you are running in Demo Mode because you have not installed a license, when you install Point of View v7.1+SP2 on the same computer, the newer version will begin its own 40-hour evaluation period and the older version will continue running 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 workbench options for an unlimited time. This mode also allows you to execute the runtime tasks and debugging tools (Database Spy, Output window, and LogWin module) for 24 hours continuously. After the 24-hour period these tasks shut down, but you can restart them again and run for another 24 hours, and so on. You can use this license for development and testing only.

Licensed for Runtime Only

Enables you to run all runtime 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.

Licensed for Engineering + Runtime

Enables all engineering tools, runtime tasks, and debugging tools (Database Spy, Output window, and LogWin module) for an unlimited period of 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.

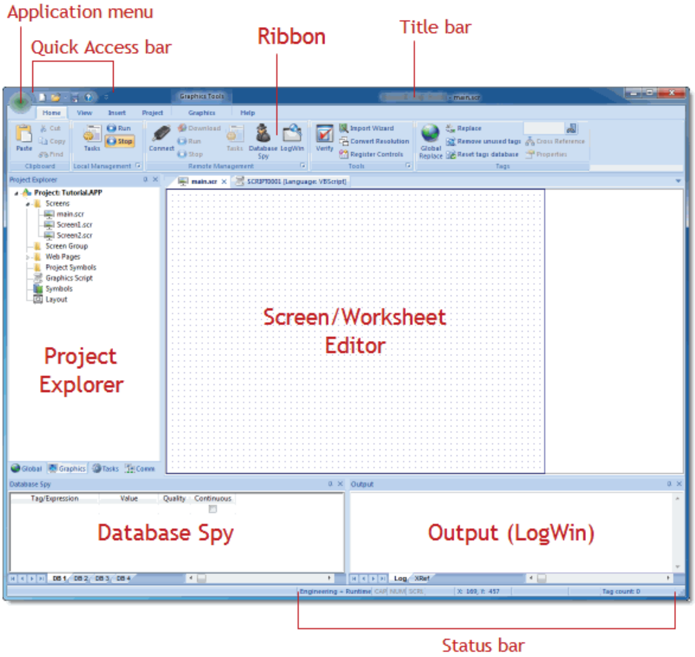
Notes:

The Development Environment

Chapter 2

The Development Environment

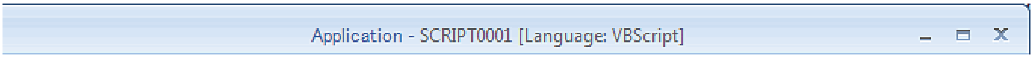
Point of View incorporates a modern, Ribbon-based Windows interface to provide an integrated and user-friendly development environment.



The POV Development Environment

Title Bar

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



Application - SCRIPT0001 [Language: VBScript]

Example of Title Bar

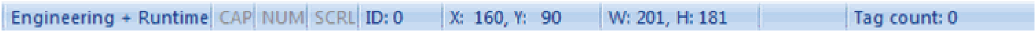
The Title Bar also provides the following buttons (from left to right):

- 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.



Engineering + Runtime CAP NUM SCRL ID: 0 X: 160, Y: 90 W: 201, H: 181 Tag count: 0

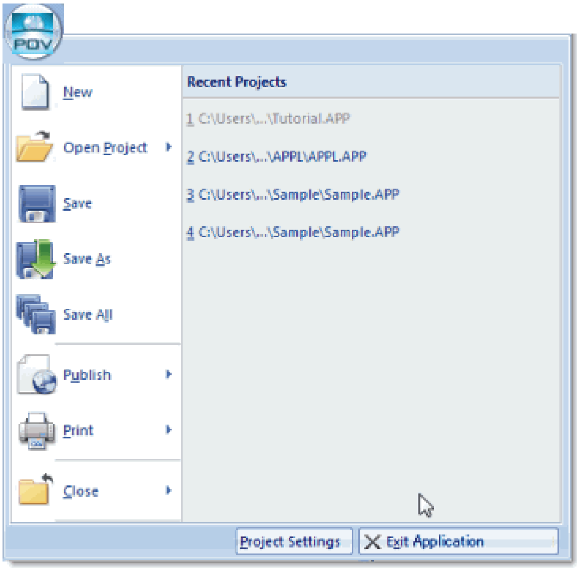
Example of Status Bar

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.


PDV
New
Open Project
Save
Save As
Save All
Publish
Print
Close
Recent Projects
1 C:\Users\...\Tutorial.APP
2 C:\Users\...\APPL\APPL.APP
3 C:\Users\...\Sample\Sample.APP
4 C:\Users\...\Sample\Sample.APP
Project Settings
Exit Application

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

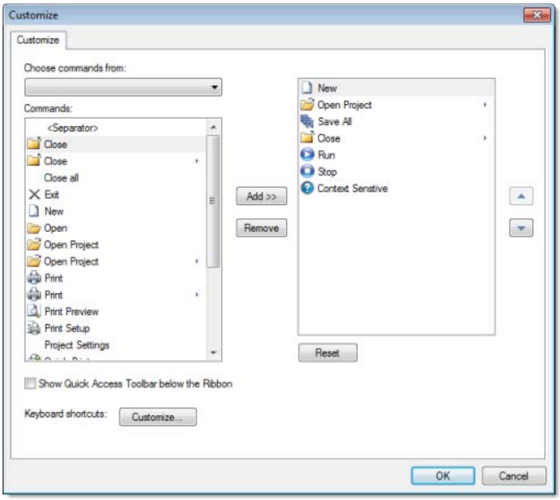
You can add a command to the Quick Access Toolbar directly from commands that are displayed on the ribbon:

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.
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.

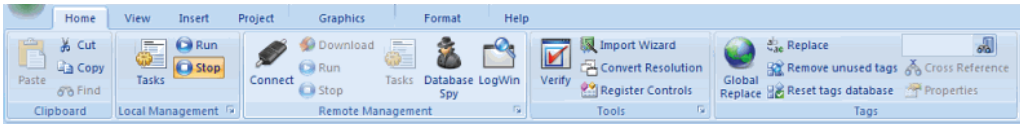
Only commands can be added to the Quick Access Toolbar. The contents of most lists, such as indent and spacing values and individual styles, which also appear on the ribbon, cannot be added to the Quick Access Toolbar.



Customize Quick Access Toolbar Dialog

Ribbon

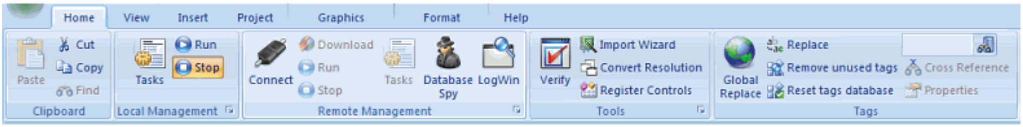
The ribbon combines the numerous menus and toolbars into a single, user-friendly interface. Almost all application commands are on the ribbon, organized into tabs and groups according to general usage.



The Ribbon Interface

Home Tab

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



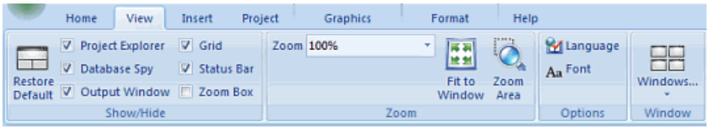
Home Tab of the Ribbon

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.
- **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 the "About remote management" help file topic.
- **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.



☑ Project Explorer
☑ Database Spy
☑ Output Window
☑ Grid
☑ Status Bar
□ Zoom Box
Zoom 100%

View Tab of the Ribbon

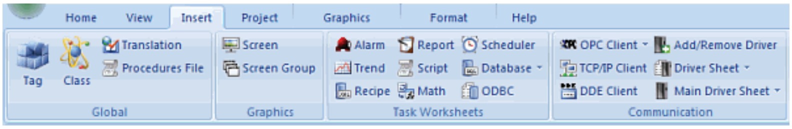
The tools are organized into the following groups:

- 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.



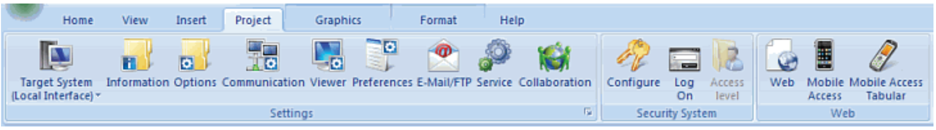
Insert Tab of the Ribbon

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.



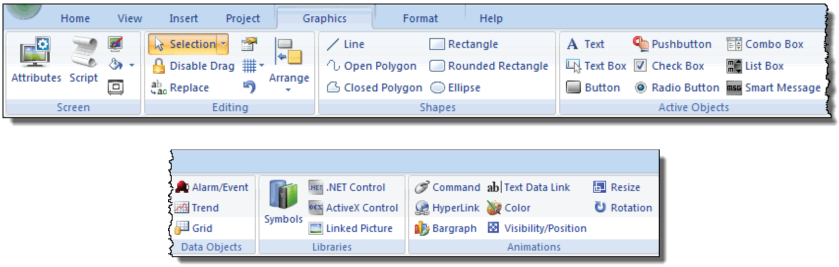
Project Tab of the Ribbon

The tools are organized into the following groups:

- **Settings:** Configure the general project settings or set the project to run as a Windows service.
- **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.



Graphics Tab of the Ribbon

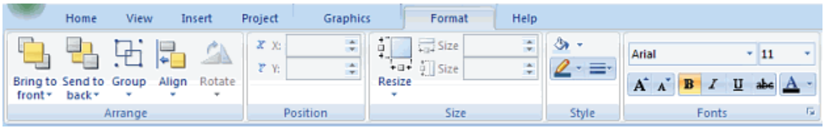
NOTE: This tab is available only when you have a project screen open for editing.

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, .NET and ActiveX controls, and external image files.
- **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.



Format Tab of the Ribbon

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

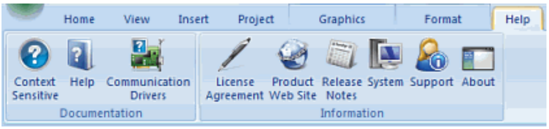
The tools are organized into the following groups:

- **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.



Help Tab of the Ribbon

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 Point of View, 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.

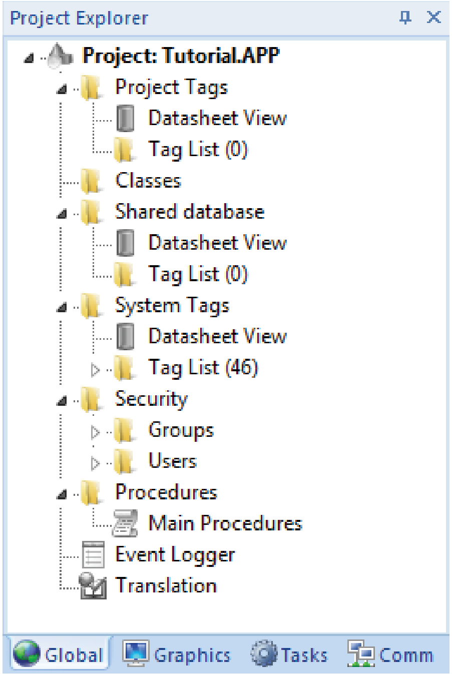
To close a folder, click the Collapse icon to the left of the folder.

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:

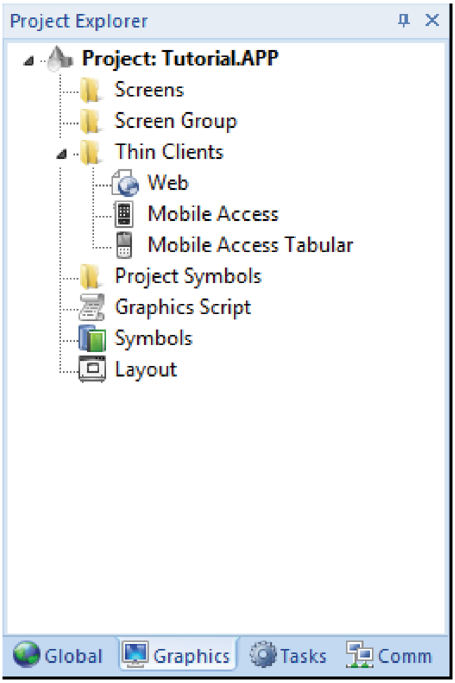
- **Project Tags:** The project tags database contains all of the data tags that you create during project development, such as screen tags (e.g., button1_state) or tags that read from / write to connected devices.
- **Classes:** Classes are compound tags that you can create to associate a set of values, 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.
  You cannot add, edit, or delete 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:

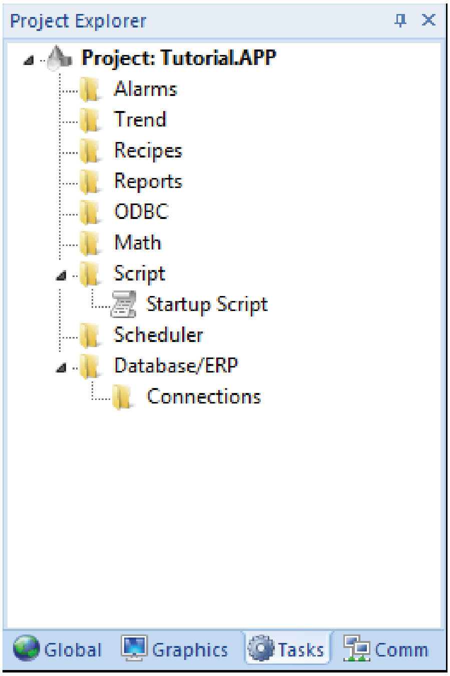
- **Screens:** You create screens to provide a graphical interface for your project. Each screen can contain many buttons, sliders, dials, indicators, graphs, and so on.
- **Screen Groups:** You can combine individual screens into screen groups, so that they all open together at the same time.
- **Thin 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.

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.)

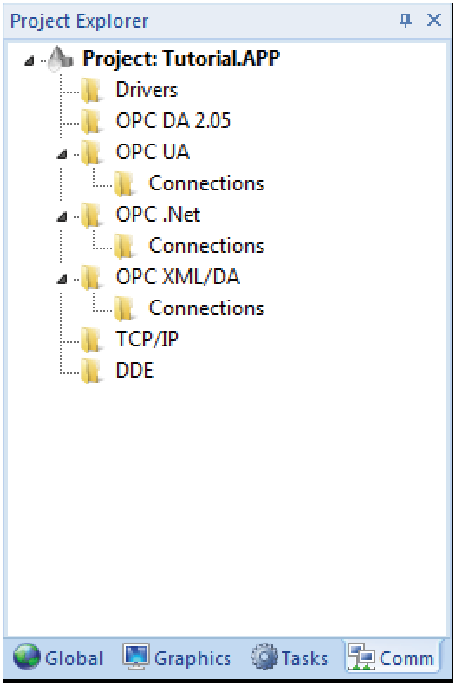


Tasks Tab of the Project Explorer

- 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.)
- **ODBC:** You can use ODBC worksheets to set up connections and exchange data with other ODBC-compliant databases.
- **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 (as an alternative to ODBC).

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.



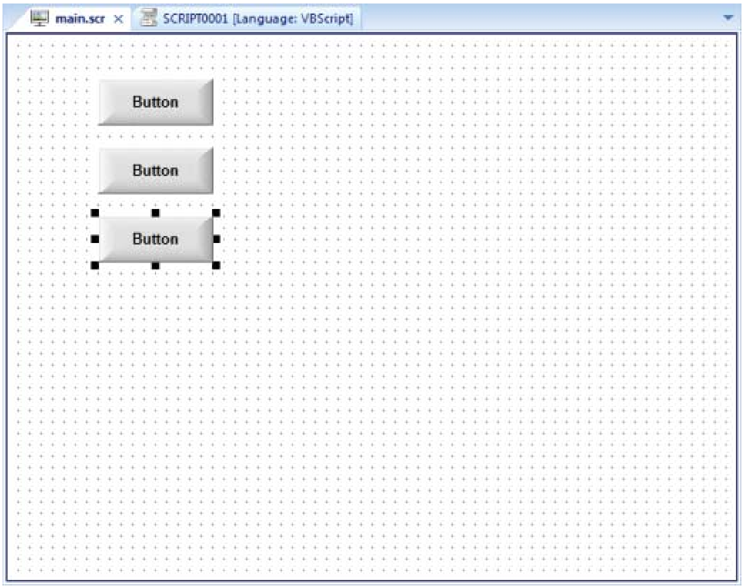
Comm Tab of the Project Explorer

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 .Net:** You can use OPC .Net worksheets to communicate with OPC servers via the new OPC .NET 3.0 protocol (formerly OPC Xi).
- **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 POV 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.
- **DDE:** You can use DDE worksheets to communicate with other Microsoft Windows applications, such as Microsoft Excel, that support the Dynamic Data Exchange 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

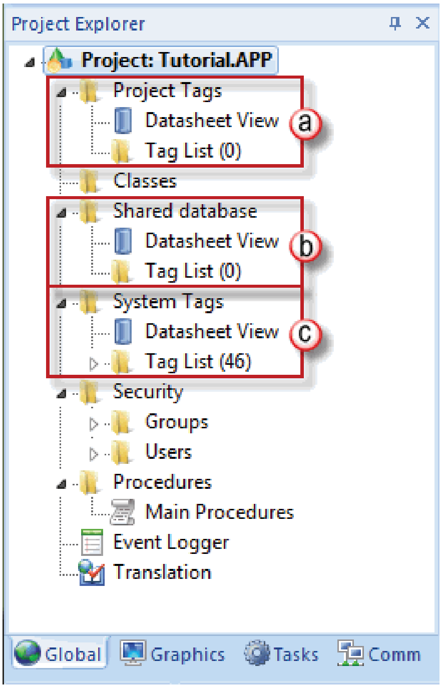
About Tags and the Project Database

Chapter 3

About Tags and the Project Database

Tags are a core component of any POV project. Simply put, tags are variables used by POV 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. POV includes a real-time database manager that provides a number of sophisticated functions such as timestamping of any value change, checking tag values against runtime minimum and maximum values, comparing tag values to alarming limits, and so on. A POV 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:

a. 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

b. Shared Database tags are created in a PC-based control program and then imported into POV's

tags database.
For example, you might create tags in Productivity Suite and import them into POV so POV can read/write data from a P3000 PAC based control product.
You cannot modify shared tags within POV — you must modify the tags in the original PC-based control program, and then re-import them into the Tags database.

c. System Tags are predefined tags with predetermined functions that are used for POV 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 delete 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. The figure, shown on previous page, shows a partial list of system tags.

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.

NOTE: *System tags do not count against your project tag count.*

Understanding the Tag Name Syntax

Observe the following guidelines when naming a tag:

- Your tag names must be unique — you cannot specify the same name for two different tags (or functions). If you type an existing tag name, POV recognizes that the name exists and will not create the new tag.
- You must begin each tag name with a letter. Otherwise, you can use letters, numbers, and the underscore character (_) in your tag name.
- You cannot use the following symbols in a tag name:
  ` ~ ! @ # $ % ^ & * ( ) - = \ + \ [ ] { } < > ?
- You can use a maximum of 255 characters for a tag name or a class member name.You can use uppercase and lowercase characters. Tag names are not case sensitive. Because POV does not differentiate between uppercase and lowercase characters, you can use both to make tag names more readable. (For example: TankLevel instead of tanklevel.)
- Tag names must be different from system tag names and math functions.

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

Some valid tag examples include:

- Temperature
- pressure1
- count
- x

Choosing the Tag Data Type

Another consideration when designing a tag is what type of data the tag will receive. POV 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.

See also: Understanding Tag Properties and Parameters help file topic

Using Array Tags

POV 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].

WARNING: 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].

NOTE: *When using the PAC3K Driver, the index minimum is 1.*

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 (use 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 POV 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. (See “Creating project database Tags” in help file).When you create an n-position array tag, POV 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 POV 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], POV regards a[second as the tag and considers it invalid because POV 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

POV 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 of the String type

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 X 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

Chapter 4

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

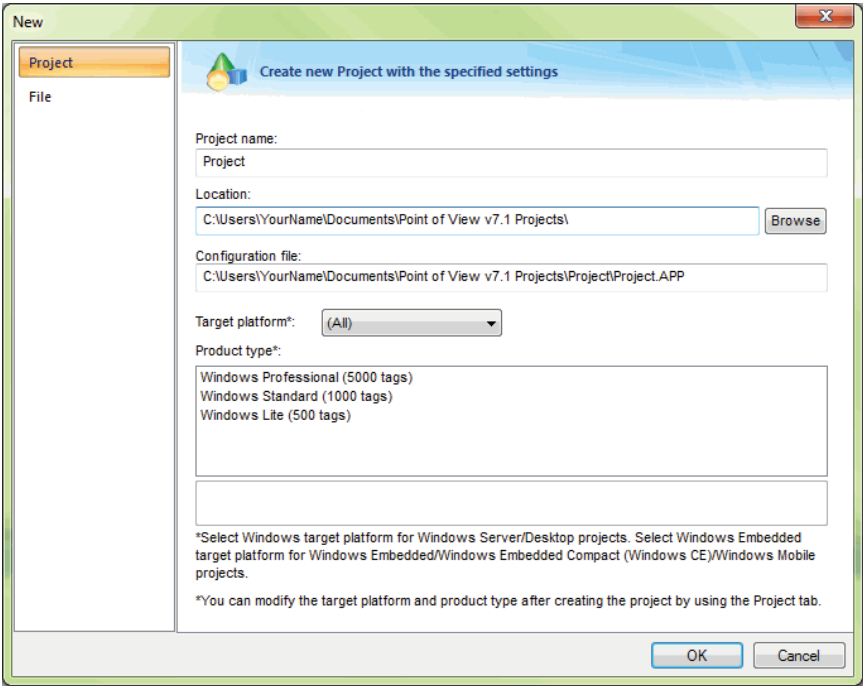
This part of the tutorial shows how to create a new project, including how to give it a name and select the target platform.

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. For this example, select Windows Standard. This is a tag and feature-limited product type that can be safely deployed on Windows PC Platforms.



New

Create new Project with the specified settings

Project name: Project

Location: C:\Users\YourName\Documents\Point of View v7.1 Projects\

Configuration file: C:\Users\YourName\Documents\Point of View v7.1 Projects\Project\Project.APP

Target platform*: (All)

Product type*:

Windows Professional (5000 tags)
Windows Standard (1000 tags)
Windows Lite (500 tags)

*Select Windows target platform for Windows Server/Desktop projects. Select Windows Embedded target platform for Windows Embedded/Windows Embedded Compact (Windows CE)/Windows Mobile projects.

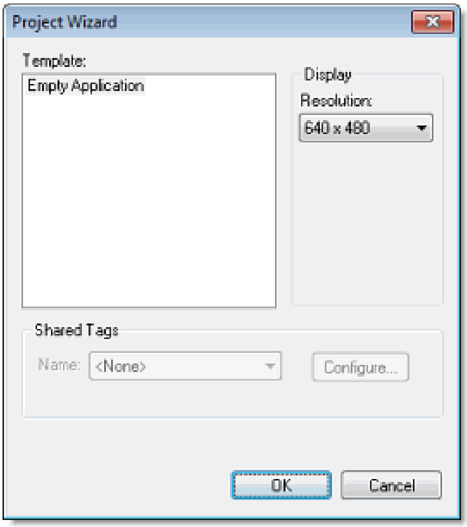
*You can modify the target platform and product type after creating the project by using the Project tab.

Selecting the target platform and product type

5) Click OK.
The New dialog is closed and the Project Wizard dialog is displayed.

6) In the Template list, select Empty Application.

7) In the Resolution list, select 640 x 480.


Project Wizard

Template: Empty Application

Display

Resolution: 640 x 480

Shared Tags

Name: <None>

Specifying an empty Application with 640x480 resolution

8) Click OK.
The Project Wizard dialog is closed and the new project is created in the development environment.

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:

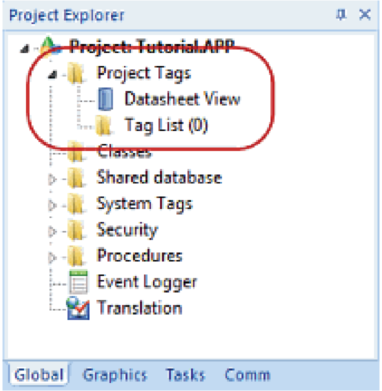
1) On the Project tab of the ribbon, in the Settings group, click Viewer.
The Project Settings dialog box is displayed with the Viewer tab selected.

2) In the Startup screen box, type main.scr.
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 an 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.

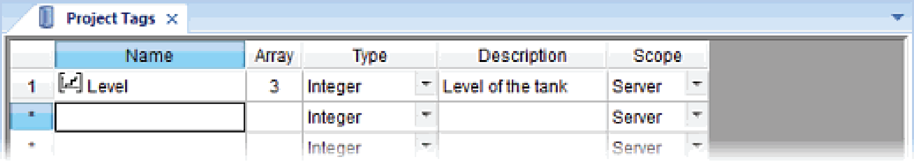
c. **Type:** Specify the data type of the tag: Boolean, Integer, Real, String, or Class. For this tutorial, select Integer.

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.



	Name	Array	Type	Description	Scope
1	Level	3	Integer	Level of the tank	Server
*			Integer		Server
*			Integer		Server

Creating the Level tag

5) Save and close the Project Tags datasheet.

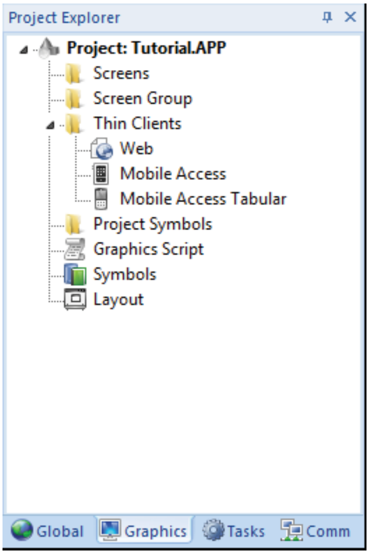
You will create additional tags as you build the project.

NOTE: *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

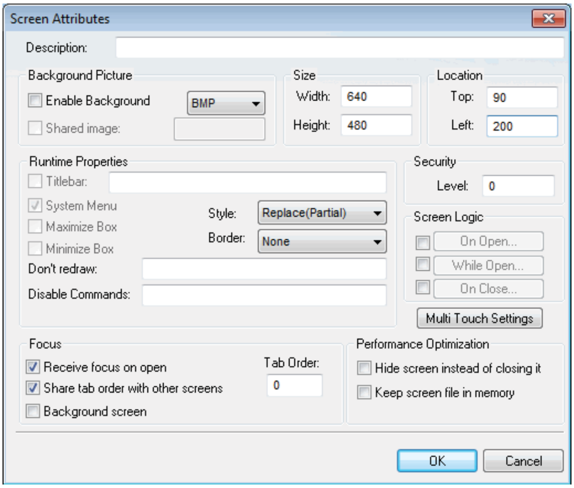
This part of the tutorial shows how to create your first screen, which will contain a single button that opens another screen.

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 an project in this Screens folder.

The Screen Attributes dialog box is displayed.



Screen Attributes

Description:

Background Picture
□ Enable Background BMP
□ Shared image:

Size
Width: 640
Height: 480

Location
Top: 90
Left: 200

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

Security
Level: 0

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

Multi Touch Settings

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

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

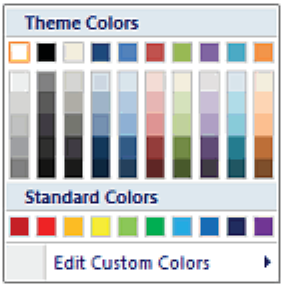
OK Cancel

Screen Attributes dialog box

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 box 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.

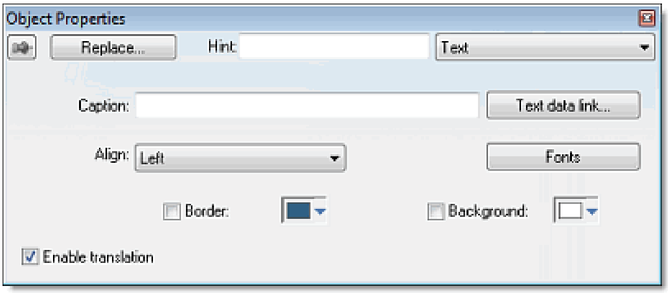
Drawing the Main Screen's Title

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 box.



Hint:

Text

Caption:

Align: Left

□ Border:

□ Background:

☑ Enable translation

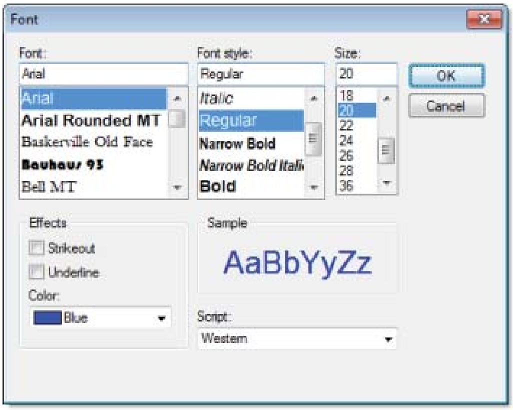
Object Properties: Text dialog box

- Double-clicking on any screen object opens an Object Properties dialog box containing the properties for that object. The properties shown in the dialog change depending on the type of object.
- The Object Properties dialog box 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 box 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 box, and then specify the font settings.
For this tutorial…

- Font is Arial
- Font style is Regular
- Size is 20
- Color is Blue

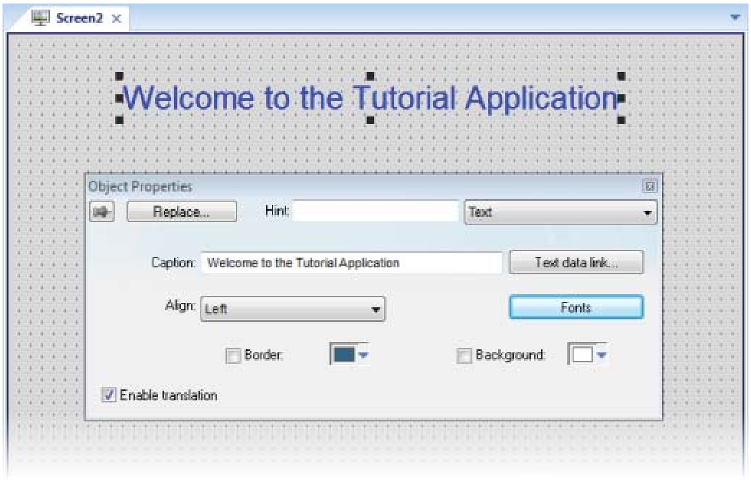


Font

Font: Arial
Font style: Regular
Size: 20
Effects: □ Strikeout □ Underline
Color: Blue
Sample: AaBbYyZz
Script: Western

Specifying the font settings

5) Click OK to close the Font dialog box.
The font settings are applied to the Text object.



Welcome to the Tutorial Application

Object Properties

Hint:
Caption: Welcome to the Tutorial Application
Align: Left
□ Border
□ Background
☑ Enable translation

Font settings applied to Text object

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

Drawing a Button to Open Another Screen

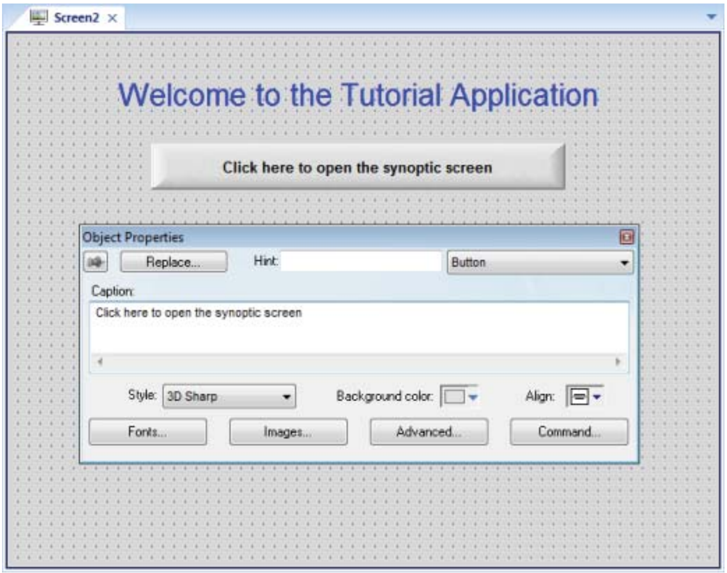
This part of the tutorial shows how to draw and configure a button that will open another screen.

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 box.

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



Welcome to the Tutorial Application

Click here to open the synoptic screen

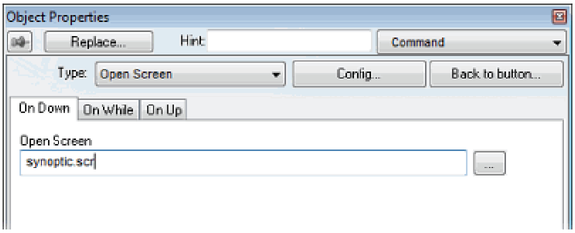
Hint:
Button
Caption:
Click here to open the synoptic screen
Style: 3D Sharp
Background color:
Align:

Adding a caption to the button

5) Click Command.
The Object Properties dialog box 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.scr.



Hint:
Command
Type: Open Screen
Open Screen
synoptic.scr

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 box.

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 box is displayed.

2) In the File name box, type main.

3) Click Save.
The file is saved in your project folder (at […]\Documents\Point of View v7.1 Projects\projectname\Screen\main.scr), and the Save dialog box 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 Project Explorer, click the Graphics tab.
2) Right-click the Screens folder, and then click Insert on the shortcut menu.
   The Screen Attributes dialog box is displayed.
3) Use this dialog box to set attributes such as size and type.
   For this tutorial, click OK to accept the default settings.
4) 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 box is displayed.
5) In the File name box, type synoptic.
6) Click Save.
   The file is saved in your project folder (at […]\Documents\Point of View v7.1 Projects\project_name\Screen\synoptic.scr), and the Save dialog box is closed.

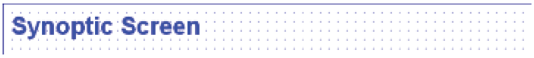
Do not close the screen like you did the main screen when you saved it. You still need to draw the synoptic screen.

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 box.
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) Close the Object Properties dialog box.
6) Move the Text object to the top left corner of the screen.
7) 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 screen title.



Synoptic Screen

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.

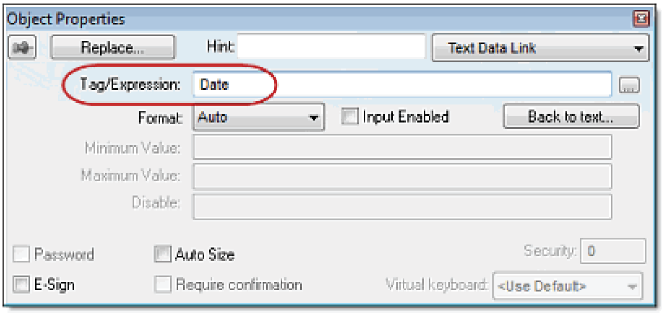
1) On the Graphics tab of the ribbon, in the Active Objects group, click Text.

2) Click on the screen, type Date: ##########, and then press Return.

3) Double-click the object to open its Object Properties dialog box.

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.



Hint:
Text Data Link
Tag/Expression: Date
Format: Auto
□ Input Enabled
Minimum Value:
Maximum Value:
Disable:
□ Password
□ Auto Size
Security: 0
□ E-Sign
□ Require confirmation
Virtual keyboard: <Use Default>

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 box.

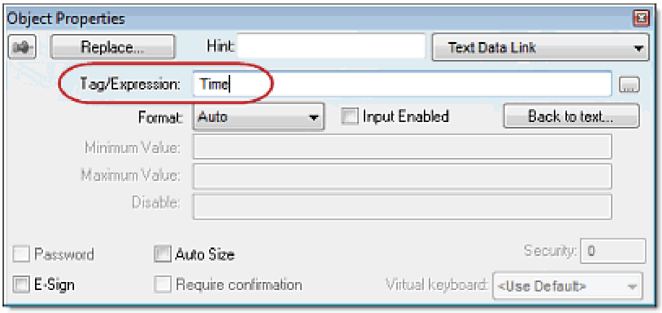
7) On the Graphics tab of the ribbon, in the Active Objects group, click Text.

8) Click on the screen, type Time: ##########, and then press Return.

9) Double-click the object to open its Object Properties dialog box.

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.



Hint:
Text Data Link
Tag/Expression: Time
Format: Auto
□ Input Enabled
Minimum Value:
Maximum Value:
Disable:
□ Password
□ Auto Size
Security: 0
□ E-Sign
□ Require confirmation
Virtual keyboard: <Use Default>

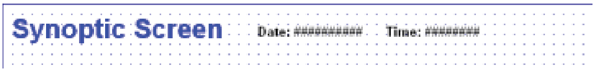
Specifying the Time system tag

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 box.

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've created the date and time objects.



Synoptic Screen Date: ########## Time: ########

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 runtime project.

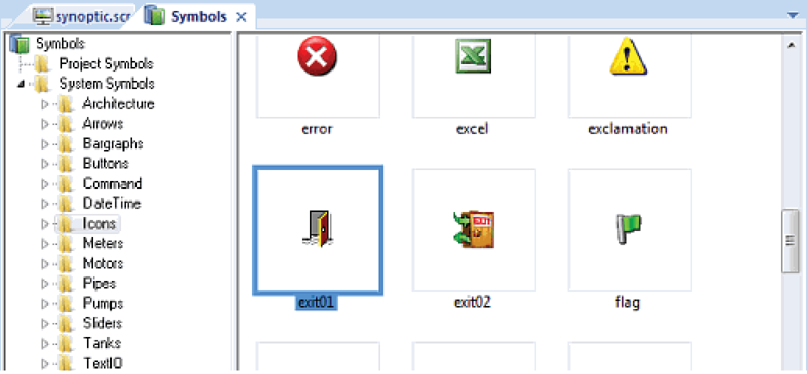
NOTE: If any part of the project doesn't work as expected (or to exit the runtime application), switch back to the development application (ALT+TAB) and then click Stop on the Home tab of the ribbon. Then if needed, begin troubleshooting.

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.

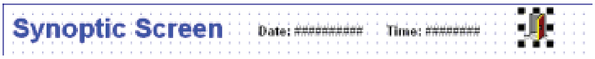
3) In the Icons sub-folder, select exit01.
The symbol will be displayed in the symbol viewer to the right of the menu tree.

4) Click on the symbol.
The mouse cursor will change to show that the symbol is ready to be placed in a screen.



Selecting the "exit01" symbol

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.



Synoptic Screen Date: ########## Time: ########

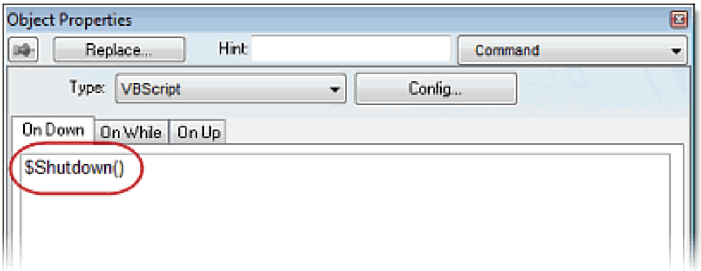
Placing the Linked Symbol object

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 box.

8) In the Type list, select VBScript.

9) In the On Down box, type $Shutdown().
Shutdown is one of Point of View's built-in scripting functions, but it can be used within VBScript by prefacing it with a dollar sign ($).



Type: VBScript

```
$Shutdown()
```

Specifying the Shutdown command on the symbol

10) Close the Object Properties dialog box.

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 show how to test the project so far.

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) Click the button to open the synoptic screen.
The synoptic screen is displayed.

3) Click the exit icon to shut down the project.

If any part of the project doesn't 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

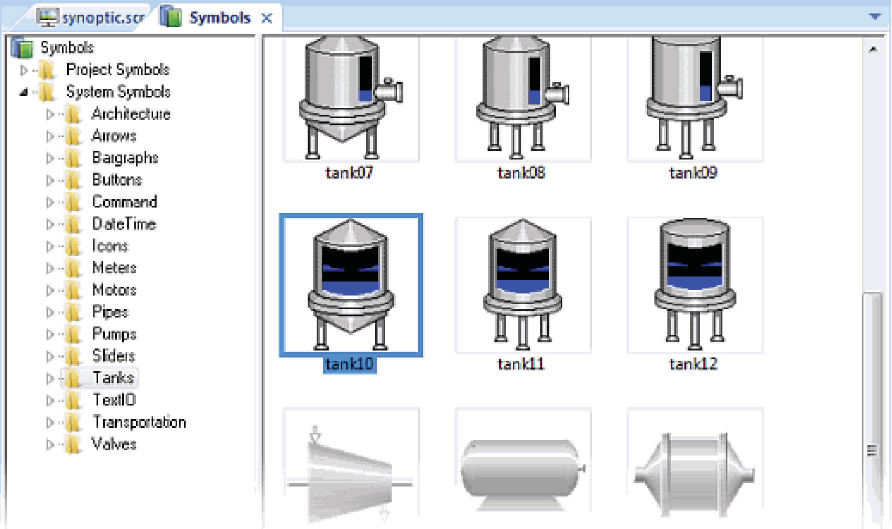
This part of the tutorial shows how to select an animated tank from the Symbol Library and place it on the screen (similar to how you selected and placed the "Exit" icon), then associate some project tags with the tank's properties.

1) 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 Tanks

sub-folder.

3) In the Tanks sub-folder, select a tank symbol.
You may select any tank you like; they all function basically the same way.

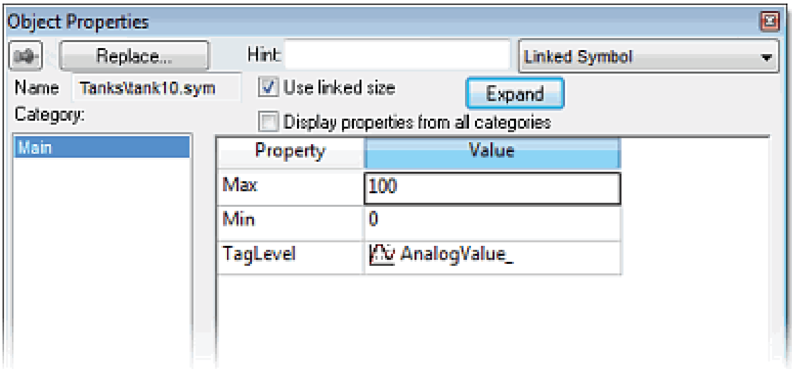


Selecting a tank symbol

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 click in it.
The symbol is placed as a Linked Symbol object.

6) Double-click the object to open its Object Properties dialog box.



Object Properties

Replace... Hint: Linked Symbol

Name Tanks\tank10.sym ☑ Use linked size Expand

Category: ☐ Display properties from all categories

Main

Property	Value
Max	100
Min	0
TagLevel	AnalogValue_

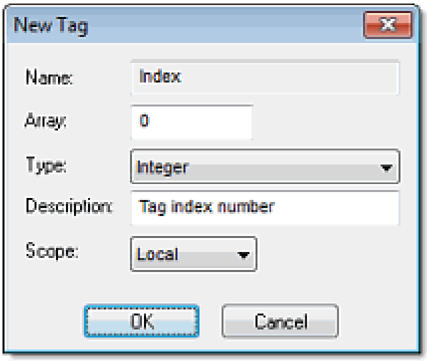
The tank symbol's properties

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.

7) 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.

8) Click Yes.
A New Tag dialog is displayed.

9) Configure the new tag with Array as 0, Type as Integer, and Scope as Local.



New Tag

Name: Index

Array: 0

Type: Integer

Description: Tag index number

Scope: Local

Configuring a new tag

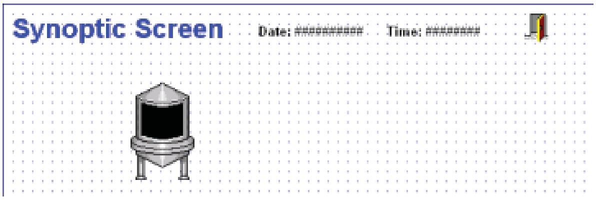
10) Click OK to close the New Tag dialog.
You can use the tag Index to set the array position of the tag Level, and show the level for any of the three tanks in the same object:

- 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.

11) Close the Object Properties dialog box.

12) 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.



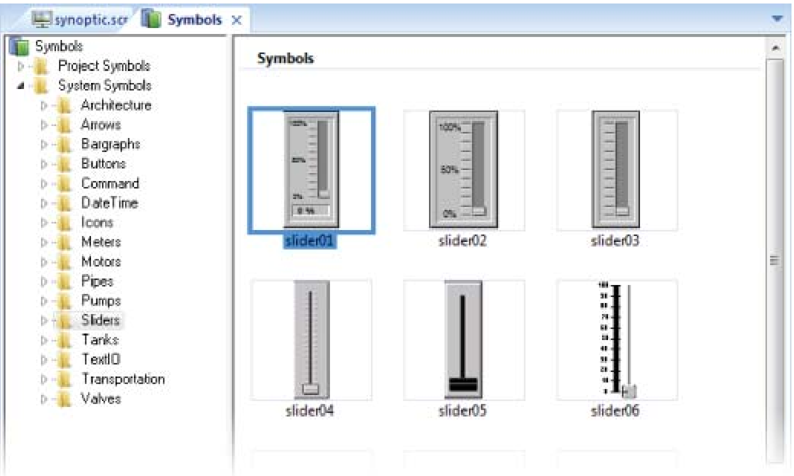
Finished tank object

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.

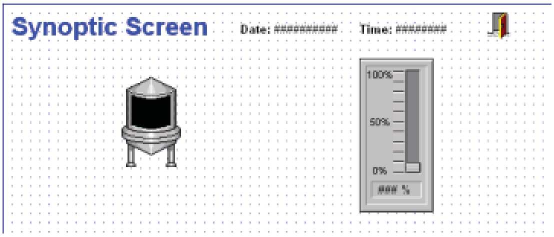
5) Switch back to the screen where you want to place the symbol and click in it.
The symbol is placed as a Linked Symbol object.

6) Double-click the object to open its Object Properties dialog box.

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 box.

9) Click the Application button at the top-left of the development environment, and then click Save on the Application menu.

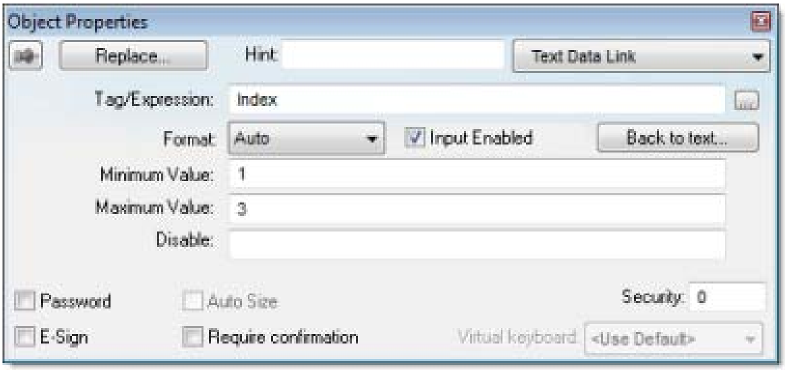


Finished level slider object

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 box.
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.



Hint:

Text Data Link

Tag/Expression: Index

Format: Auto

☑ Input Enabled

Minimum Value: 1

Maximum Value: 3

Disable:

☐ Password

☐ Auto Size

Security: 0

☐ E-Sign

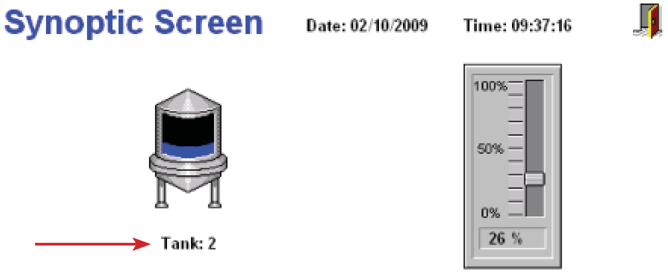
☐ Require confirmation

Virtual keyboard: <Use Default>

Configuring the "Tank" text input

9) Close the Object Properties dialog box.
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 (shown with a red arrow).



Finished tank selector object during runtime

Testing the Project

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

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) Click the button to open the synoptic screen.
   The synoptic screen is displayed.
3) 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.
4) Click the exit icon to shut down the project.

If any part of the project doesn't 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

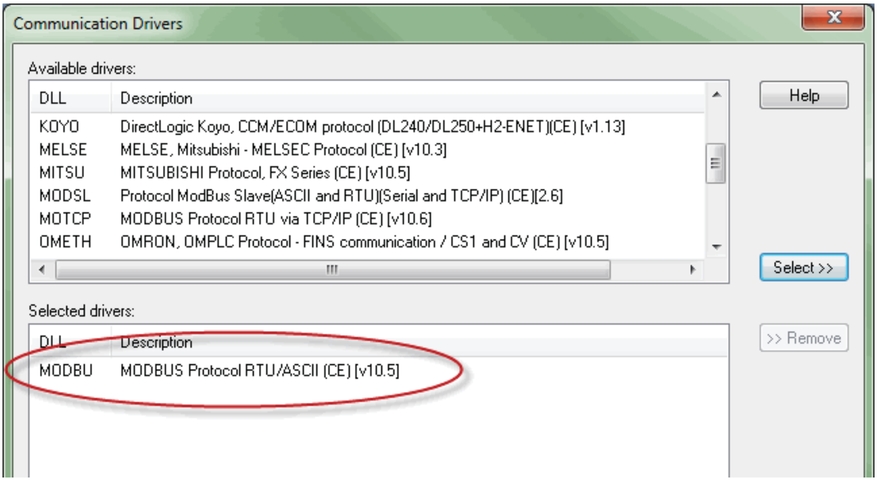
This part of the tutorial shows how to select and configure a driver to communicate with an external I/O device.

***NOTE:** If you are using a DirectLOGIC or P3000 controller, see the Quick Start Guide for AutomationDirect Drivers manual (PV-ADCDRV-M) for information pertaining to those controllers.*

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.



Available drivers:

DLL	Description
KOYO	DirectLogic Koyo, CCM/ECOM protocol (DL240/DL250+H2-ENET)(CE) [v1.13]
MELSE	MELSE, Mitsubishi - MELSEC Protocol (CE) [v10.3]
MITSU	MITSUBISHI Protocol, FX Series (CE) [v10.5]
MODSL	Protocol ModBus Slave(ASCII and RTU)(Serial and TCP/IP) (CE)[2.6]
MOTCP	MODBUS Protocol RTU via TCP/IP (CE) [v10.6]
OMETH	OMRON, OMPLC Protocol - FINS communication / CS1 and CV (CE) [v10.5]

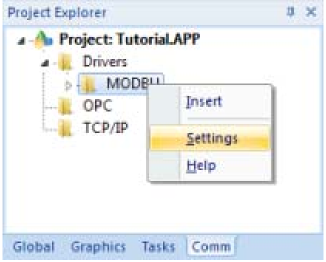
Selected drivers:

DLL	Description
MODBU	MODBUS Protocol RTU/ASCII (CE) [v10.5]

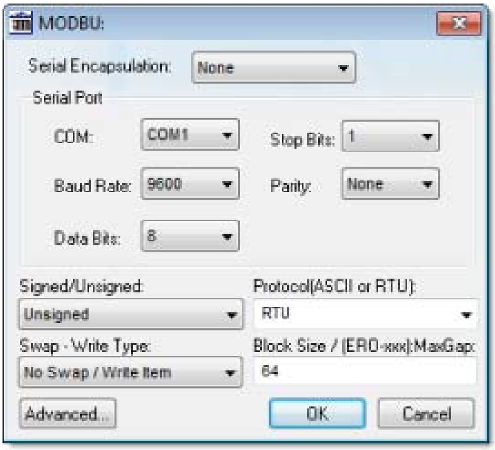
MODBU driver selected

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.



MODBU:

Serial Encapsulation: None

Serial Port

COM: COM1

Stop Bits: 1

Baud Rate: 9600

Parity: None

Data Bits: 8

Signed/Unsigned: Unsigned

Protocol(ASCII or RTU): RTU

Swap - Write Type: No Swap / Write Item

Block Size / (ERO-xxx):MaxGap: 64

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.

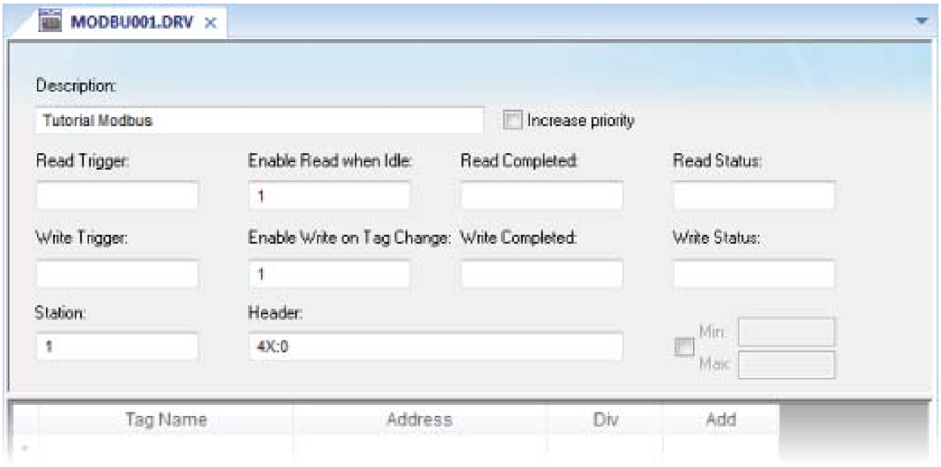
b. In the Enable Read When Idle box, type 1.
This setting is a trigger that takes a Boolean value. A value of 1 — either entered manually as above or evaluated from a tag/expression — forces your project to continue reading tag values from the target device even when there are no changes in value.

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



MODBU001.DRV

Description: Tutorial Modbus

☐ Increase priority

Read Trigger:	Enable Read when Idle:	Read Completed:	Read Status:
	1		
Write Trigger:	Enable Write on Tag Change:	Write Completed:	Write Status:
	1		

Station: 1

Header: 4X:0

☐ Min: Max:

Tag Name	Address	Div	Add

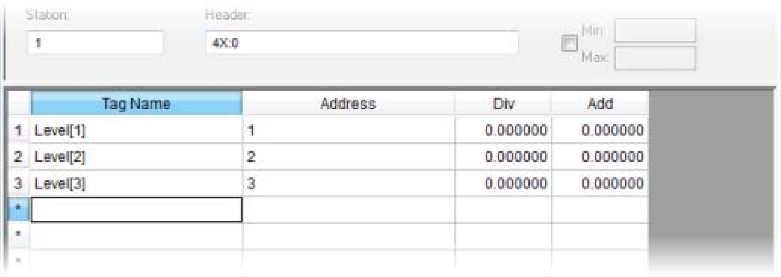
Completed worksheet header

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)
Level[3]	3	4X:3 (Holding Register 3)



Station: 1

Header: 4X:0

☐ Min: Max:

	Tag Name	Address	Div	Add
1	Level[1]	1	0.000000	0.000000
2	Level[2]	2	0.000000	0.000000
3	Level[3]	3	0.000000	0.000000
*				

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.

Deploying Your Project as a Web Application

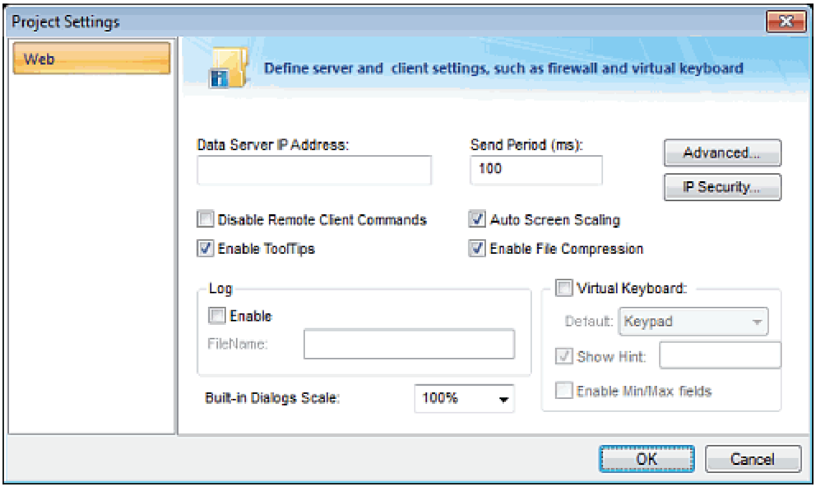
This part of the tutorial shows how to deploy your project as a web application, to which remote users can connect with Internet Explorer.

For Internet Explorer to work as a web thin client, it must install an ActiveX control that "plays" POV project screens. If your computer is connected to the Internet, then IE will automatically download the control from AutomationDirect.com's public server when you connect to a project runtime server for the first time.

To deploy your project as a web application:

1) Configure the IP address of the data server.

a. On the Project tab of the ribbon, in the Thin Clients group, click Web.
The Project Settings dialog box is displayed with the Web tab selected.



Project Settings

Web

Define server and client settings, such as firewall and virtual keyboard

Data Server IP Address:

Send Period (ms): 100

□ Disable Remote Client Commands

☑ Auto Screen Scaling

☑ Enable ToolTips

☑ Enable File Compression

Log

□ Enable

FileName:

□ Virtual Keyboard:

Default: Keypad

☑ Show Hint:

□ Enable Min/Max fields

Built-in Dialogs Scale: 100%

Web tab of Project Settings dialog box

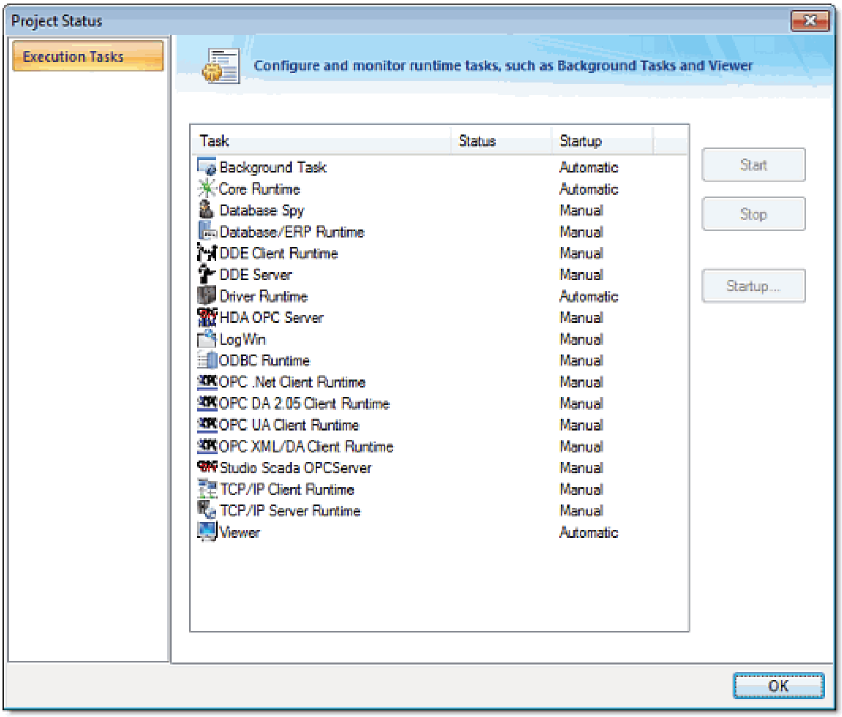
b. In the Data Server IP Address box, type the IP address of the web server.
For this tutorial, type 127.0.0.1, which is the standard loopback address (a.k.a. "localhost").

c. Click OK to close the dialog.

2) Make sure the data server is set to start up when you run your project.

a. On the Home tab of the ribbon, in the Local Management group, click Tasks.

The Execution Tasks dialog box is displayed.

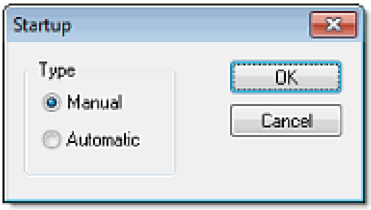


Configure and monitor runtime tasks, such as Background Tasks and Viewer

Task	Status	Startup
Background Task		Automatic
Core Runtime		Automatic
Database Spy		Manual
Database/ERP Runtime		Manual
DDE Client Runtime		Manual
DDE Server		Manual
Driver Runtime		Automatic
HDA OPC Server		Manual
LogWin		Manual
ODBC Runtime		Manual
OPC .Net Client Runtime		Manual
OPC DA 2.05 Client Runtime		Manual
OPC UA Client Runtime		Manual
OPC XML/DA Client Runtime		Manual
Studio Scada OPCServer		Manual
TCP/IP Client Runtime		Manual
TCP/IP Server Runtime		Manual
Viewer		Automatic

Execution Tasks dialog box

b. In the list of tasks, select TCP/IP Server Runtime, and then click Startup.
The Startup dialog box is displayed.



Type
● Manual
○ Automatic

Startup dialog box

c. Select Automatic, and then click OK.

d. Click OK to close the Execution Tasks dialog box.

3) Save and close all open screens and worksheets.

4) Click the Application button at the top-left of the development environment, and then click Publish > Save All As HTML on the Application menu.

Your project screens are saved as HTML files in the Web sub-folder of your project folder (i.e., \project_name\Web).

5) Configure a web server to make the Web sub-folder available to the network.
   For this tutorial, use NT Web Server, the free, lightweight web server program that is included with POV. Simply copy the program file from […]\AutomationDirect\Point of View v7.1\Bin\NTWebServer.exe to the Web sub-folder, and then double-click the file to run it. It automatically serves the contents of whatever folder it is in, without any further configuration.

WARNING: NT Web Server is provided for testing purposes only. It should never be used in a real production environment or on a secure network.

   You can also configure the "root directory" or "home directory" setting of some other web server program (e.g., Microsoft IIS) to point to the Web sub-folder, or you can copy the Web sub-folder to an existing web server on your network. The web server (which makes the HTML files available to clients) and the data server (which actually runs your POV project and exchanges data with the clients) do not need to be the same computer.

6) On the Home tab of the ribbon, in the Local Management group, click Run.

7) Open a web browser (e.g., Microsoft Internet Explorer), and then enter the URL address of the synoptic screen on the web server. For this tutorial, type http://127.0.0.1/synoptic.html and then press Return. (127.0.0.1 is the standard IP address for the loopback network interface, a.k.a. "localhost.")
   After a few moments, during which the browser downloads and installs the ActiveX control, the synoptic screen is displayed in the browser.

Notice that you can modify the level of any tank either locally using the project viewer or remotely using the web browser, and changes on one client appear immediately on the other. They work equally well.