GENESIS64 Standard Training Manual

Version 10.7

Rev. B

© 2013 ICONICS, Inc. All rights reserved worldwide. This document may not be reproduced or copied in any manner without written authorization from ICONICS, Inc.

The information contained within this document is subject to change without notice. ICONICS, Inc. does not guarantee the accuracy of the information.

GENESIS64, Hyper Historian, Pocket GENESIS, BizViz and their respective modules, OPC-To-The-Core, and Visualize Your Enterprise are trademarks of ICONICS, Inc.

Windows, Windows XP, Windows 2000, Windows 2003 Server, Windows CE, Windows NT, Windows XP-64, Windows 2003-64, Windows Vista, Windows Server 2008, Windows 7, Windows Server 2008 R2, Windows 8 and Windows Server 2012 are registered trademarks of Microsoft Corp. in the United States and other countries.

Other product and company names mentioned herein may be trademarks or registered trademarks of their respective owners.

Document Number: TM-GEN64-107B

ICONICS LICENSE AGREEMENT

YOU SHOULD CAREFULLY READ THE FOLLOWING TERMS AND CONDITIONS!

Opening and using the enclosed software for any purpose indicates your acceptance of the terms and conditions of this license agreement. If you do not agree with them you should return all software, documentation and copy protection keys within seven days of shipment unopened and your money will be refunded.

ICONICS provides this program and licenses its use in the United States, Puerto Rico, or internationally. You assume the responsibility for the selection of the program to achieve your intended results, and for the installation, use and results obtained from this program.

LICENSE

You are granted a personal license to use this program under the terms stated in this Agreement. You may: 1) Install and use the program on a single machine. 2) Make a single archival back-up copy of the program for the sole purpose of supporting your use of the single program on a single machine. 3) You may not use, copy, modify, or transfer the program, or transfer any copy, in whole or in part, except as expressly provided in this license, or with a written contractual agreement with **ICONICS,** Inc.

TERM

The license is effective until terminated. It will terminate if you fail to comply with any term or condition of this Agreement. You agree, upon such termination, to destroy the program and all copies that were made from it and to promptly return any copy protection key(s) to **ICONICS.**

LIMITED WARRANTY

ICONICS WARRANTS THE CD-ROM OR PHYSICAL DISKETTES, ON WHICH THE PROGRAM IS FURNISHED, AND PHYSICAL DOCUMENTATION TO BE FREE OF DEFECTS IN MATERIALS AND WORKMANSHIP UNDER NORMAL USE FOR A PERIOD OF NINETY (90) DAYS FROM THE DATE OF DELIVERY TO YOU AS EVIDENCED BY YOUR RECEIPT. THIS PROGRAM IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE ENTIRE COST OF ALL NECESSARY SERVICING, REPAIR, OR CORRECTION.

ICONICS SPECIFICALLY DISCLAIMS ALL OTHER WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO, IMPLIED WARRANTIES OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. IN PARTICULAR, WITH RESPECT TO ANY PARTICULAR APPLICATION, USE OR PURPOSE, IN NO EVENT SHALL **ICONICS** INC. BE LIABLE FOR ANY OTHER COMMERCIAL DAMAGE, INCLUDING BUT NOT LIMITED TO SPECIAL, INCIDENTAL, CONSEQUENTIAL OR OTHER DAMAGES. **ICONICS** DOES NOT WARRANT THAT THE FUNCTIONS CONTAINED IN THE PROGRAM WILL MEET YOUR REQUIREMENTS OR THAT THE OPERATION OF THE PROGRAM WILL BE UNINTERRUPTED OR ERROR FREE.

LIMITATION OF REMEDIES

ICONICS' entire liability and your exclusive remedy shall be the replacement of any CD-ROM or diskette not meeting **ICONICS**' "Limited Warranty" which is returned to **ICONICS** with a copy of your receipt within the warranty period. The remedy for breach of this warranty shall be limited to replacement and shall not encompass any other damages including but not limited to loss of profit, special, incidental, consequential, or other similar claims arising out of the use or inability to use such program even if **ICONICS** has been advised of the possibility of such damages, or for any claim by any other third party.

GENERAL

This Agreement will be governed by the laws of the Commonwealth of Massachusetts. Should you have any questions concerning this Agreement, you may contact **ICONICS** in writing at:

ICONICS, Inc.
100 Foxborough Boulevard
Foxborough, MA 02035

You acknowledge that you have read this agreement, understand it and agree to be bound by its terms and conditions. You further agree that it is the complete and exclusive statement of the Agreement between you and **ICONICS** which supersedes any proposal or prior Agreement, oral or written, and any other communications between you and **ICONICS** relating to the subject matter of the Agreement.

ICONICS RETURN POLICY

All sales are final (NO RETURNS) unless one of the following conditions is applicable: Program is returned in its original packaging material within 7 days after shipment. In such a case, a full refund is provided, less all incurred shipping and handling costs. Program is returned within 30 days from the original shipment date. In such a case a 20% restocking fee is applicable. No returns are allowed on OPC ToolWorX or ActiveX ToolWorX.


ICONICS

Contents

1. Introduction to GENESIS64

1.1. GENESIS64 Architecture 1-3
1.2. PLC, OPC, and GENESIS64 1-7
1.3. Data Connections 1-14
1.4. The Data Browser 1-15
1.5. Microsoft Silverlight and Windows 7 Features 1-19
LAB: OFFSITE TRAINING INSTALLATION AND CONFIRMATION L1-1
REVIEW: INTRODUCTION TO GENESIS64 ..... R1-1

2. Workbench

2.1. Introduction to the Workbench 2-1
2.2. Customizing the Workbench 2-6
2.3. Creating a Project 2-9
2.4. Configuration and Layout Management 2-13
2.5. Project Management Tools 2-24
2.6. Deployment 2-28
2.7 Workbench Runtime 2-29
LAB: DATA CONNECTIVITY L2-1
REVIEW: WORKBENCH R2-1

3. Getting to Know GraphWorX64

3.1. GraphWorX64 Layout 3-2
3.2. The Object Explorer 3-3
3.3. The Properties Explorer 3-6
3.4. Customizing the Workspace Using the View Ribbon 3-7
3.5. Templates and Transitions 3-11
3.6. Symbols Library 3-13
LAB: ICTONICS GRAPHICAL TEMPLATE L3-1
REVIEW: GETTING TO KNOW GRAPHWORX64 ..... R3-1

4. GraphWorX64 Basics

4.1. Basic 2D Shapes and Objects 4-1
4.2. Dynamics 4-4
4.3. Object Count 4-9
4.4. Application Mode 4-10
4.5 Display Password Protection 4-12
LAB: ICTONICS MONITOR DISPLAY L4-1
REVIEW: GRAPHWORX64 BASICS R4-1

5. GraphWorX64 Tools

5.1 Objectives 5-1
5.2 GENESIS64 Controls 5-1
5.3 Search and Replace 5-6
5.4 Arrange Controls and Objects 5-8
5.5 Views 5-11
5.6 The GraphWorX64 Runtime Ribbon and Menus 5-14
LAB: LAYERS AND POPUP DISPLAYS L5-1
REVIEW: GRAPHWORX64 TOOLS R5-1

6. AlarmWorX64 Server

6.1. AlarmWorX64 Server 6-1
6.2. Alarm Tags 6-3
6.3. Configuring Alarm Tags 6-4
6.4. Group Alarm Tags by Area 6-19
6.5. Importing and Exporting Components 6-22
LAB: ALARMWORX64 SERVER L6-1
REVIEW: ALARMWORX64 SERVER R6-1

7. AlarmWorX64 Viewer

7.1 Configuring the Viewer 7-3
7.2 Alarm Viewer Runtime Options 7-21
LAB: VIEWING ALARMS L7-1
REVIEW: ALARMWORX64 VIEWER R7-1

8. AlarmWorX64 Logger

8.1. Starting the AlarmWorX64 Logger Configurator 8-2
8.2. AlarmWorX64 Logger Configuration 8-2
8.3. Column Definitions 8-10
8.4. Nodes Configuration 8-11
LAB: ALARM LOGGER L8-1
REVIEW: ALARMWORX64 LOGGER R8-1

9. TrendWorX64 Logger

9.1. TrendWorX64 Logger in the Workbench 9-2
9.2. Database Configurations 9-3
9.3. Database Groups 9-4
9.4. Adding Logging Groups 9-5
9.5. Add Tags to a Logging Group 9-10
LAB: TRENDWORX64 LOGGER L9-1
REVIEW: TRENDWORX64 LOGGER R9-1

10. TrendWorX64 Viewer Control

10.1. TrendWorX64 Viewer Modes 10-2
10.2. TrendWorX64 Viewer Properties 10-3
10.3. Tabs 10-5
10.4. Chart View 10-6
10.5. Plots and Plot Types 10-11
10.6. Pen Configuration 10-13
10.7. The Side Panel 10-16
10.8. Runtime Operations 10-17
10.9. Workbench vs. GraphWorX64 10-21
LAB: TREND VIEWER L10-1
REVIEW: TRENDWORX64 VIEWER R10-1

11. Unified Data Manager

11.1 Working with Objects in the UDM 11-2
11.2 Expressions 11-4
11.3 Groups 11-8
11.4 Value Sets 11-10
11.5 Registers 11-13
11.6 Triggers 11-19
LAB: RECIPES L11-1
REVIEW: UNIFIED DATA MANAGER R11-1

12. Global Aliases

12.1 Architecture 12-1
12.2 Definition of Global Alias 12-3
12.3 Alias Resolution 12-5
12.4 Using Themes and Theme Values 12-6
12.5 Configuring Themes and Global Aliases 12-8
12.6 Using Global Aliases 12-12
LAB: GLOBAL ALIASING L12-1
REVIEW: GLOBAL ALIASES R12-1

13. Language Aliases

13.1 Background 13-2
13.2 Configuring Language Aliasing 13-3
13.3 Language Alias Switching 13-12
LAB: A MULTI-LINGUAL SYSTEM 13-1
REVIEW: LANGUAGE ALIASING 13-1

14. GenTray and FrameWorX64 Server

14.1. Interfacing with GenTray 14-1
14.2. FrameWorX64 Server Configuration 14-3
14.3. Changing the Default FrameWorX64 Server 14-8
14.4. Remote Access 14-9
14.5. Web Publishing 14-16
LAB: REMOTE CONFIGURATION L14-1
REVIEW: GENTRAY AND FRAMEWORX64 ..... R14-1

15. GridWorX64

15.1 GridWorX64 Server Configurator 15-2
15.2 Create a Grid Using GridWorX64 15-11
15.3 Create a Grid in a GraphWorX64 Display 15-12
LAB: GRIDWORX64 ..... L15-1
REVIEW: GRIDWORX64 ..... R15-1

16. Security

16.1 Security System Components 16-2
16.2 Global Settings 16-3
16.3 Security Objects and Their Relations 16-8
16.4 Security Privileges 16-14
16.5 Logging-in and Logging-out 16-22
16.6 Advanced Settings 16-25
LAB: SECURITY L16-1
REVIEW: SECURITY R16-1

A. GraphWorX64 Shortcuts A-1
B. Delimiters and Syntaxes B-1
C. Glossary C-1
D. Test Your Knowledge - Answers D-1

Introduction to GENESIS64 1

Look around. In any modern industrial economy, the water you drink, the food you eat, the clothes you wear, the energy you use, and the myriad other necessities and amenities of life are made possible through computer-aided observation and control of automated systems used to process them. Industrial automation takes data issued by devices, puts them into a form that can be sent over a network, and delivers that data to the computer that has the software necessary to process it. Process control, in turn, takes commands issued by automation software on a computer and sends them to the correct device(s) where each command is translated into action. For this scheme to work in an open system (multi-vendor environment), devices and systems must be discoverable, communication must use known transport protocols, and data must be in a predictable format– all of which require standards.

The software you will be learning in this course, the ICONICS GENESIS64 automation suite, is an open system software suite based on Microsoft Windows and industry standardized networking protocols. Listed below are some applications and functions within the GENESIS64 suite:

- Displays process data in near real-time within GraphWorX64
- Allows hierarchical organization with AssetWorX
- Monitors events and alarm conditions with AlarmWorX64
- Logs historical data for analysis with TrendWorX64 or with the unique high speed Hyper Historian
- Displays geographical information for your assets with EarthWorX64
- Displays database data from various sources with GridWorX64

GENESIS64 allows you to construct Graphical User Interfaces (GUIs) in a visual object-oriented development environment. Your displays can model your processes or systems such that controls look and feel just like real devices. GENESIS64 masks the complexity of the underlying data in a way that it displays only the information that an operator needs, which uses configuring controls and presents them in a meaningful, instantly recognizable way. You can change flows with simulated dials, observe temperatures with simulated thermometers, check tank levels with a cut-a-way figure showing the water level in the tank – practically anything you can think of!

With scripting, aliasing (variables), reusable Microsoft .NET Framework components, open standards, and being Web-enabled, GENESIS64 works seamlessly in a multi-vendor environment. Once you learn how to use GENESIS64 products, you can reuse components, apply complex shapes and controls with mouse click, and rapidly develop the displays you need.

Consider the Industry Case Studies in the GenDemo application (Figure 1.1) as an example of the possibilities with GENESIS64. Each case study was built using GENESIS64 technology. To see these case studies in GENESIS64, click Start → All Programs → ICONICS → GENESIS64 → GenDemo. Click on the text for *Examples* on the left to load the Industry examples. Select GENESIS64 or GENESIS32 in the options that load on the left (GENESIS64 examples can also be accessed from the Navigation Menu → Industries tab). Then click on an image to open a display with sample data.



Figure 1.1 - GenDemo: Industry Case Studies

After completing this chapter, you should:

- Understand the GENESIS64 architecture
- Know how to open a display
- Distinguish Configuration mode from Runtime mode
- Know how to connect a process point or dynamic to device data
- Understand OPC UA and other OPC specifications
- Know the different applications in the GENESIS64 Suite
- Understand the benefits of using GENESIS64 on Microsoft Silverlight and with Windows Operating System features

1.1. GENESIS64 Architecture

GENESIS64 is a massive suite of applications. Naturally you are wondering, "Where do I start?" and "How do the applications communicate?". Luckily, we have created a single location to manage all your project details; the Workbench.

You can perform all system configurations in the Workbench. From your Alarm Server configuration, to your system security, to configuring your HMI displays, you can do it all within the Workbench. The GENESIS64 application architecture is shown in Figure 1.2 below. The Workbench offers a centralized configuration environment for layers labeled Application Servers and Platform Services.

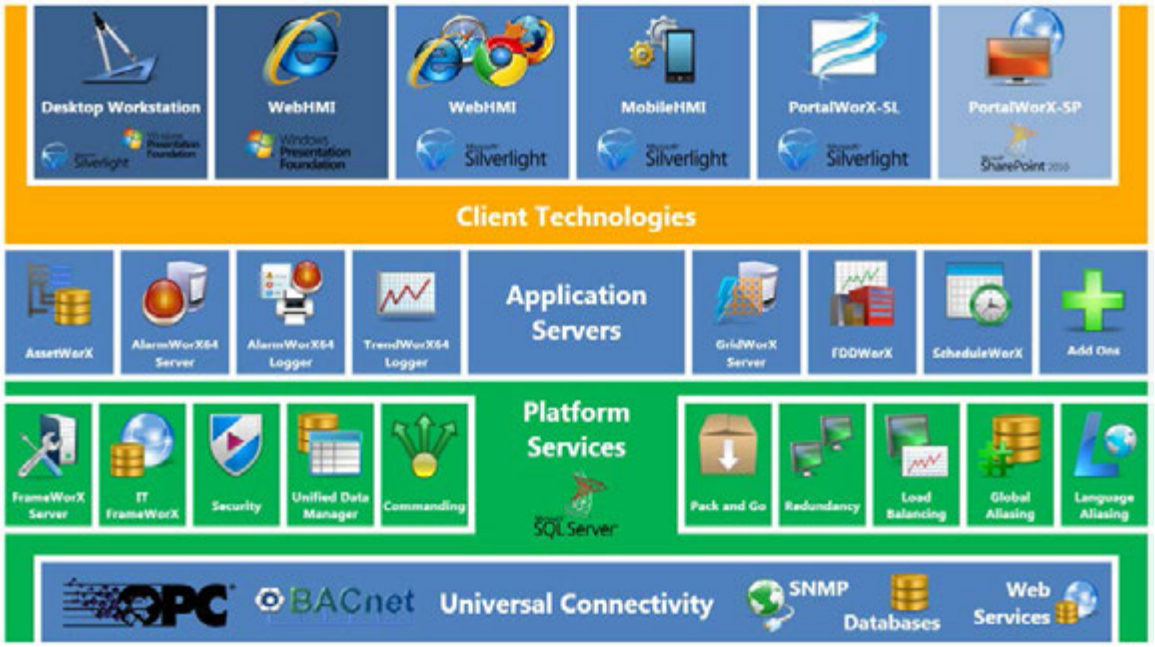


Figure 1.2 - Architecture

As you see from Figure 1.2, you can connect to and configure many different GENESIS64 modules within the Workbench. Some of the applications that you can access include:

- ICONICS viewers (.NET controls) for GraphWorX64, TrendWorX64, AlarmWorX64, EarthWorX, Push Pin, Smart Pin, and Pipe
- Server configuration utilities such as the AlarmWorX64, Security, Licensing, and many others
- ICONICS utilities such as Global Aliasing, Language Aliasing, Unified Data Manager and others, which are front ends to SQL Server data stores
- Communication server configurations for the FrameWorX (OPC UA) server, GenTray (for GenBroker), BACnet, SNMP, and more

Aside from the utilities and the configurators, we have containers for viewing different types of data in the Workbench. These containers are also controls that you can add into GraphWorX64 displays. Unlike GENESIS32, where containers are standalone applications, most containers in GENESIS64 can only be configured and run within the Workbench. The GraphWorX64 container is the only one in the GENESIS64 suite that allows for configuration and runtime without going into the Workbench. Like the Workbench, you can configure and even go into runtime mode to view data using these containers.

1.1.1. Configuration vs. Runtime

When you create or edit a display, you are in configuration mode where you may see a button in the top right hand corner of the display that says Runtime. Runtime is the mode in which data is accessed, while Configuration mode allows you to modify the display or environment while viewing sample data only. In runtime, you may see a button on the display that says Configuration that lets you switch to configuration mode. However, in a display that has been opened in Runtime Only Mode for operators, you won't see a button in runtime that would let you switch to configuration mode.

In Figure 1.3, the Workbench is shown in runtime mode; to open this display in configuration mode you would go to Configuration mode and use the navigation tree on the left to right-click the display and choose an Edit Display menu item.

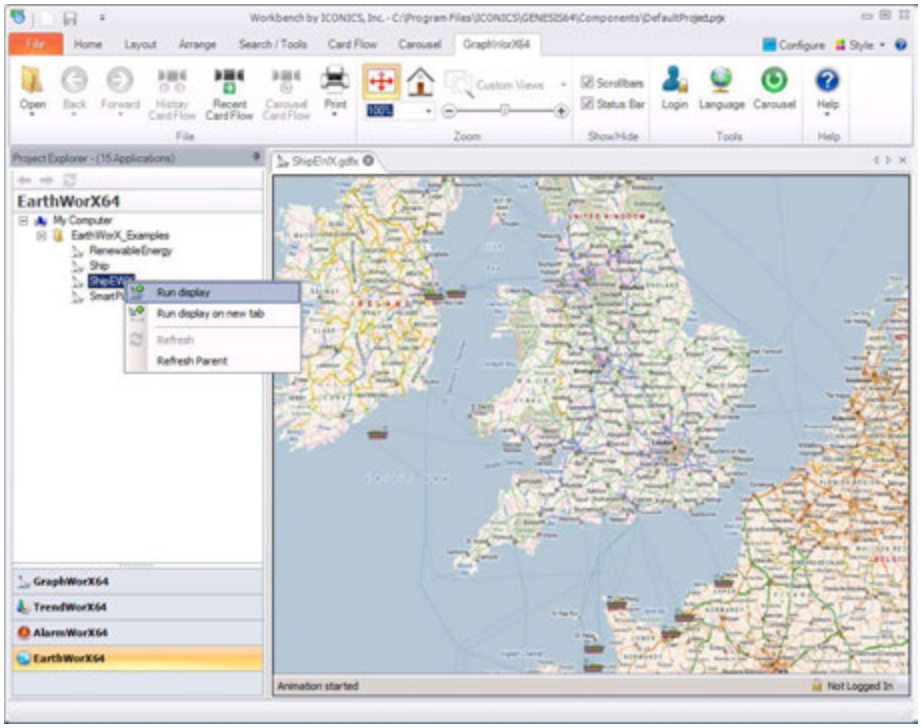


Figure 1.3 - Workbench: EarthWorX64 Display in Runtime

A GENESIS64 application is created and edited in Configuration mode only. In Configuration mode, you have access to all of the objects in the viewers, their properties, and the methods (commands or operations) you can apply to each object. The object hierarchy for the viewer objects uses the PEM model (properties, events, and methods) that characterizes visual development environments such as Visual Basic. A gauge is shown in Configuration mode within the work area in Figure 1.4.

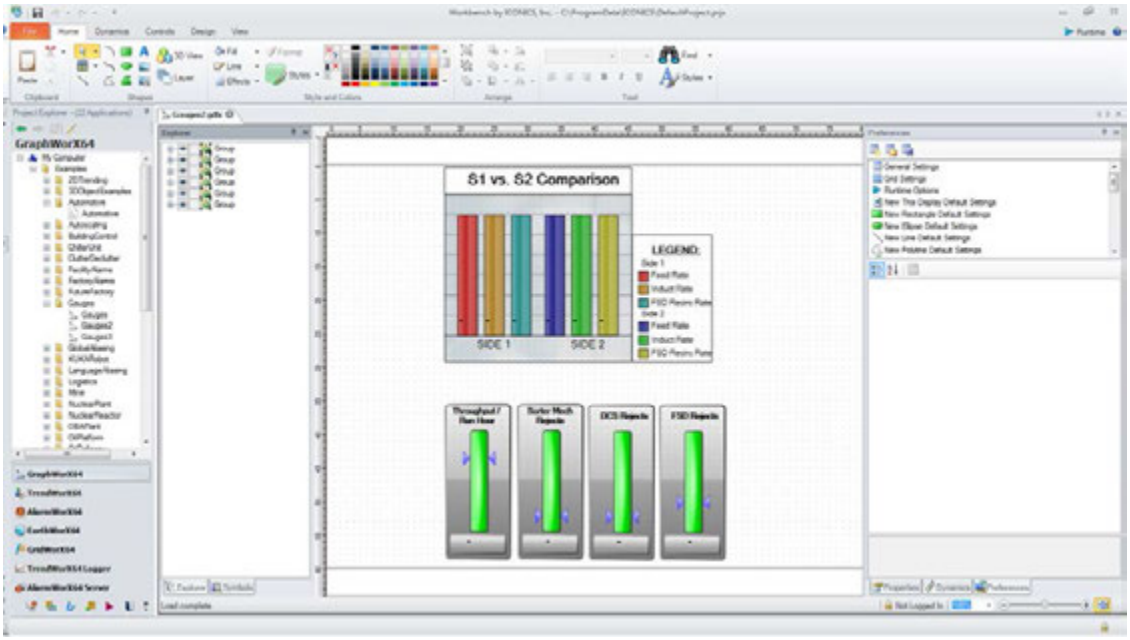


Figure 1.4 - Workbench: Gauges in GraphWorX64

Some applications in the Workbench, such as AlarmWorX64, have a preview mode that allows you to view sample data. However, the data shown are for display purposes and do not match data coming from your OPC server. When you want to see your display with real data, you click the Runtime button (located at the right of the menu bar) to switch the display into Runtime mode. To make additional changes you switch back into Configuration mode by clicking the Configuration button.

Servers and configuration utilities in the Workbench use online forms for enforcing settings on the servers and services. If you are familiar with the Microsoft Management Console (MMC), then the forms used by the Workbench should seem familiar to you.

Figure 1.5 shows a form for the default account policy in the Security Server. ICONICS security provides access to user and group accounts using policies applied to these security objects.

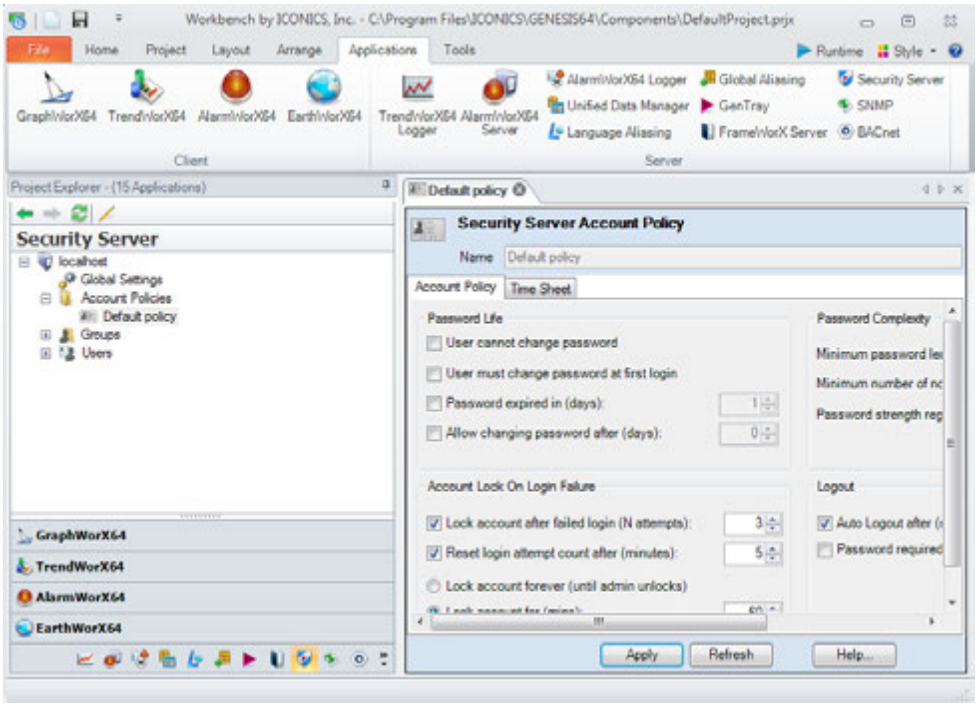


Figure 1.5 - The Security Configuration

The unification of applications into the Workbench environment means that the tools used can be common to all applications. You will find that templates, layouts, shapes, colors, and many other facilities are manipulated identically in GENESIS64 applications, making it easier to learn and use.

The complete collection of server settings, application defaults, opened displays, and other aspects of a working environment are saved into a GENESIS64 *project file* (.PRJX). A project file can be packaged (zipped) and that package can be deployed in additional locations with minimal effort.

1.2. PLC, OPC, and GENESIS64

Consider a simple device like a resistance thermometer that is used to measure temperatures. Resistance thermometers use platinum films or wires because platinum exhibits a linear resistance to temperature ratio. A highly accurate four wire Kelvin connection thermometer (the additional wires are used to cancel out lead resistances) like the one in Figure 1.6 has the property that, at a constant voltage, the resistance measured changes linearly with temperature.

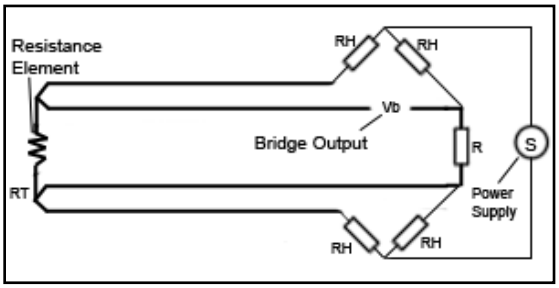


Figure 1.6 - A Four-wire Resistance Thermometer

A computing device can sample the value of the resistance, and that value is then converted from the value in ohms to a temperature measurement based on a standard physical property. While this device has one measurement of interest (the temperature as converted from the resistance), in order for the information to be useful, the temperature value needs to be packaged with additional pieces of information, including:

- The address and identification of the thermometer
- The time that the sample was collected
- The units of measure (Fahrenheit or Celsius, for example)
- The precision of the measurement

Even though the field data sought is a single value measurement, at least three or four related values must be transmitted with it to make the data useful; and they must be packaged in such a way (addressed) that allows the data to be found (browsed) and understood.

1.2.1. Programmable Logical Devices (PLC)

At the lowest level, sensors and other devices are wired into collection devices where data sources are identified (tagged), aggregated (collected), and transmitted (when required). Originally, large automation systems were proprietary hardware/software solutions referred to as Distributed Control Systems (DCS). But the advent of Ethernet and server-based network operating systems (Windows, Netware, and UNIX) led to the development of aggregation devices called Programmable Logic Controllers (PLC). Figure 1.7 shows an example of a Programmable Logic Controller (PLC).



Figure 1.7 - PLC Modules

A PLC is a small computer industrially hardened against dust, moisture, heat, cold, etc. The PLC's embedded operating system is an example of a real-time operating system since input conditions are expected to result predictably in corresponding outputs within a defined period and in a particular order. Programs to control the inputs and outputs (I/O) are typically stored in non-volatile, battery-backed memory. These programs coordinate the reading of inputs and writing of outputs to and from digital and analog sources such as temperature and pressure sensors, limit switches, actuators, electric motors, pneumatics, relays and solenoids. Some even use machine vision. PLCs can have simple I/O built into a single unit or may have external I/O modules attached to a network.

1.2.2. OPC Servers

The appearance of PLC devices led to the specification of some large industrial projects in the late 1980's and early 1990's, particularly in the automobile industry in which Ford Motors was a prominent example. In order for vendors' products to interoperate, several vendors got together to create a standard based on PLC communications. One of these efforts was the industry task force that eventually became the OPC Foundation (http://www.opcfoundation.org/). The OPC Foundation released the first set of standards in 1994 based on Windows as the operating system and Ethernet networks as the transport.

OPC originally stood for Object Linking and Embedding for Process Control; a name chosen because the technology was based on Microsoft's inter-application communication (IAC) standard at the time, OLE. The rise of the Internet led Microsoft to update OLE with first, the Common Object Model (COM), and in rapid fashion DCOM (Distributed COM). DCOM added the ability to communicate with remote networked systems.

You can recognize software built using DCOM because it loads ActiveX components into a browser or into a container application. GENESIS32 versions 5 through 9 all used ActiveX technology inside container applications.

The OPC standard specifies a set of objects, interfaces (APIs), and methods that are used by automation software in an open system. OPC bridges devices like PLCs and DCS to client applications using applications called *OPC servers* that communicate using the OPC protocol. An OPC server can be a local service running on the same system as the client (also known as the *localhost*), on a server on the LAN, or somewhere in the clouds (WAN).

OPC communication eliminates the need for developers to write individual device drivers to connect clients to individual devices. As long as a client understands the correct flavor of OPC, the connection or "subscription" may be made. OPC ended up publishing a number of different specifications based on DCOM as they embraced different classes of devices. Eventually OPC included:

- OPC Data Access (OPC DA). This set of standards connects real time data acquisition devices to applications.
- OPC Alarm and Events (OPC AE). These standards provide event-based alarm communication.
- OPC Historical Data Access (OPC HDA). Historical data access specifies how to log data to a data store and retrieve the data for later analysis.
- OPC Security. This specification sets the protocol for applications to securely communicate with OPC devices and systems.
- OPC Batch. The batch standard offers communication methods useful for batch processing needs.
- OPC XML-DA. XML DA extended the Data Access specification so that it communicates using XML over Microsoft Simple Object Access Protocol (SOAP).
- OPC Complex Data. This specification extended OPC to include binary data, complex data objects (like BLOBs), and XML documents.

Of the different standards it is only the first three – OPC DA, OPC AE and OPC AE – that find widespread use.

You can now see the reasons for the origin of individual ICONICS GENESIS applications to support each of the different standards:

- GraphWorX supports OPC DA access
- AlarmWorX is used for OPC AE access
- TrendWorX is used for OPC HDA access

Each application leverages the different capabilities of the individual types of OPC data. The GenBroker service (introduced into GENESIS32 version 6) facilitates the need for remote OPC communication, overcoming communication hardship via DCOM.

1.2.2.1. OPC UA

ICONICS GENESIS64 (version 10) is based on the newest OPC standard, called OPC Unified Architecture (OPC UA). OPC UA is based on the Microsoft .NET Framework and is meant to leverage the capabilities of object oriented languages such as C#. OPC UA specifies a Service Oriented Architecture (SOA) with multi-platform implementation using ANSI C, Java, or .NET. OPC UA data sources communicate through the ICONICs FrameWorX64 server in GENESIS64.

OPC UA defines a classic object-oriented address space. Items exist in an object hierarchy and can contain objects of other types. Objects based on another object have the property of inheritance, they start with all of the properties, events, and methods (PEM) of the parent and add additional properties to that. Figure 1.8 shows a view of part of the OPC UA address space.

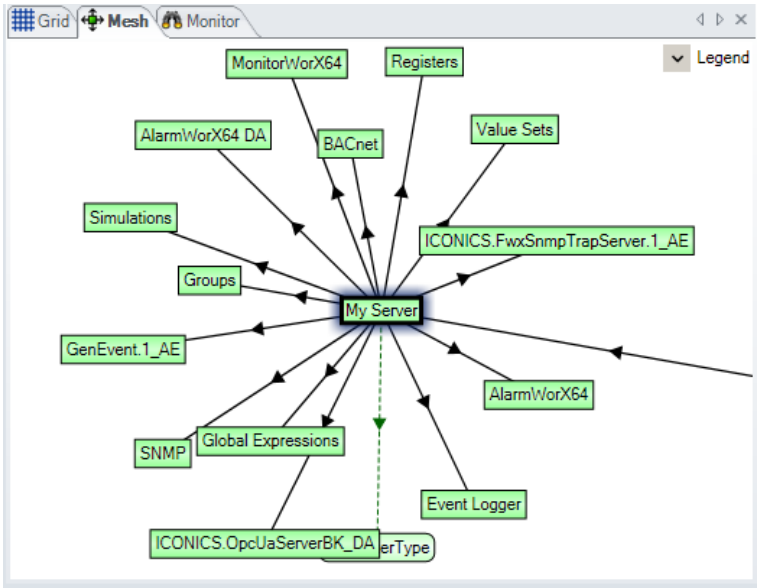


Figure 1.8 - Mesh Browser

In a process control system, a boiler, a sensor, or valve can be an object, and so forth. It is only necessary that the address of the object be unique so that you can differentiate sensor 1 from sensor 2 regardless of the boiler. The data associated with any object is expressed as the *value that is assigned to a property* of that object. Therefore, the assignment of an object property to a process point in GENESIS64 represents the data connection.

Objects have one-way relationships to other objects, and the relationships themselves are objects with properties. In the OPC UA namespace, you can define hierarchical and structural relationships, as well as mesh-line and free form relationships. Vendors may define their own address spaces as long as they follow the rules of OPC UA. Eventually OPC UA will replace OPC based on DCOM, but GENESIS64 still contains GenBroker to allow older protocol devices to be observed and controlled.

1.2.3. Data Displays

Figure 1.9 shows a display of a building of HVAC sensors for a well-known casino. This display was built using the GENESIS automation suite. Notice that the display contains a combination of digital signals (states and alarms) and analog signals (values) from multiple devices. On the display, flow data is shown numerically, in white text boxes which are READ only, as well as graphically.

A display can also show alarms, maps, and trends, which can be added to a GraphWorX64 display as TrendWorX64, AlarmWorX64, and EarthWorX64 viewers.

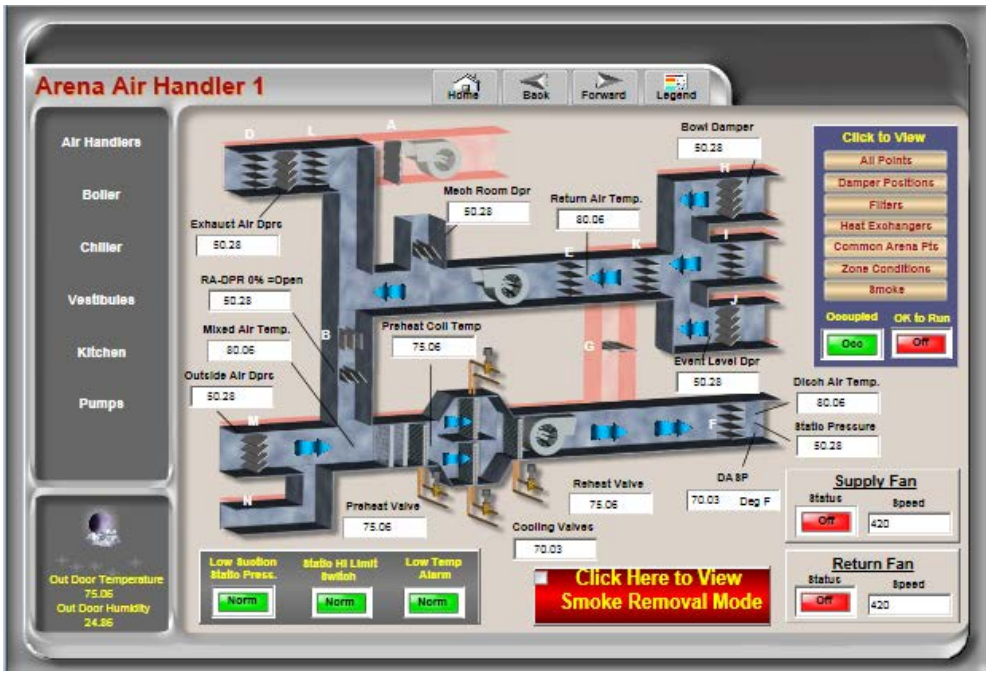


Figure 1.9 - An HVAC Display with Multiple Process Points

This type of display is commonly referred to as a Human Machine Interface (HMI); or alternatively the display portion of a SCADA or Supervisory Control and Data Acquisition system. HMI or SCADA is just industry jargon for a GUI display.

The displays generally show only the data of interest, which represents only a small fraction of the data that the underlying devices are transmitting and which are received and/or stored. Each switch can be drawn to show its state: closed, open. Underlying the display is process logic that the developer creates to control behavior. For an alarm condition, perhaps the developer will choose to draw the affected portion of the system in red, or flash an alarm. Or maybe the developer will include a button on the screen that can initiate an action. All of these options are made possible by monitoring a point coming from the OPC Server. Such points are called *OPC Tags* or *process points*.

1.2.3.1. OPC Tags

Consider the schematic of a boiler shown in Figure 1.10 with three different process points:

- the top temperature sensor outputs the temperature of the return fluid,
- the middle sensor indicates whether an alarm condition exists, and
- the bottom temperature sensor outputs the temperature of the supply fluid.

These three sensors can all be wired into a single PLC device and transmitted to a single OPC server where the data is aggregated into a data set. Then that data set can be transmitted as a unit or single transaction. Alternatively, the two real-time data sensor temperature outputs could be sent to one OPC server that is a process system while the alarms are sent to another OPC server for events. OPC UA is flexible enough to handle either scenario.



Figure 1.10 - GenDemo: Boiler Display

From the standpoint of a client, the connection to meaningful data values defines a data point as opposed to the method used to package the data and communicate it. If an OPC server exposes the OPC data as a single object (all three values) – for the sake of simplicity let's call the object "Boiler" – GENESIS64 would require you to browse the OPC UA data as a single object that shows all of the different individual values. If instead there were two objects, one called "Flows" (with temperatures) and another called "Alarms" (with events), then GENESIS64 would require that you browse two different OPC UA objects in order to assign a data source to a process point on a display.

In either scenario, there are three values connected to the display and GENESIS64 treats both scenarios as if there were three individual data points, also called tags. A tag is therefore considered to be the connection of a GENESIS64 display to a single useful value.

Note that a tag was defined as the connection to a *meaningful data point*. A temperature value or alarm condition is meaningful data. However, when you are dealing with data that is multi-valued, the entire definition of the data point is part of the same tag. In EarthWorX64, GPS locations are used, which include values for latitude, longitude, and altitude. Each value would be meaningless without the other two, so the connection to all three values as a unit is considered to be a single tag.

To take this one step further, if you have a truck moving on your EarthWorX64 map and you show location, load condition, and direction, then you consume two tags. The first tag is for the GPS location, as described in previously. The second tag is for the load condition: whether it is empty, full, or in a partially loaded state. The third displayed metric for direction is not assigned a tag value because it is based on a calculation between the last two GPS locations – with the vector connecting the two – defining a direction in your display's coordinate system.

Since ICONICS GENESIS64 uses the number of tags as part of its licensing scheme, you can see that the number of tags used has real-world financial implications. So keep in mind that objects in OPC UA can be complex data types, while usage defined by their tag values can be either simple (single-valued) or complex (multi-valued).

1.3. Data Connections

GENESIS64 requires that you establish a data connection when you want to display time-varying data from a data source. GENESIS64 provides the tools needed to connect to a wide variety of data sources.

Data sources can be:

- Variables
- Simulation functions
- Real time OPC servers with tags that can be READ-only, WRITE-only, or READ/WRITE
- Historical data that has been stored in a memory cache or saved in a disk file or log
- Custom data stored in a database or provided by a web-service for use when required

Consider the simple dial with its corresponding process point data value shown below in Figure 1.11. The dial works by having a defined rotation angle that starts at the beginning of the range (the 0 point) and ends at the end of the range (the 100 point). It's the relationship between the process point value (here having a value of 53) and the rotational angle of the indicator shape (the chevron) that is important.

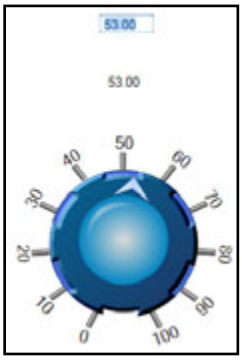


Figure 1.11 - A Dial

Other properties determine how many increments the knob has, what the labels of each increment are, whether the knob is analog or digital (has steps), and other properties. The data point might have a one-to-one correspondence of each process point's value to the labels of the knob, but the relationship can be linear, exponential, or whatever you define.

Both the text box and the rotation dynamic of the chevron object in the dial are connected to the same data source. You assign the data source for each of these two objects by entering the address of the variable, simulation, expression, or tag into the DataSource property.

As long as the value you enter into the DataSource property is syntactically correct, it will be allowed; it is up to the developer to determine the correctness of the assignment. At runtime, the value of the data source refreshes at an interval that is another property of the display.

The knob and the process point in Figure 1.11 display READ-only data, which allows for observation but no operator input. However, if the OPC tags are writable, you can easily make this into a device where you can control the value by turning the knob or writing a value in the process point. Conceptually, control is as simple to perform as observation in the OPC model.

1.4. The Data Browser

As you can see from the previous knob example, you can connect graphical objects to OPC points to display the value. You can either type in the address of the point or select it in the Data Browser in the DataSource field. In Configuration mode, when a developer clicks in the DataSource entry box, an ellipsis button appears to the right of the DataSource property box as shown in Figure 1.12. An ellipsis button in GENESIS64 indicates that more choices are available through a utility, menu, or some other interface element.

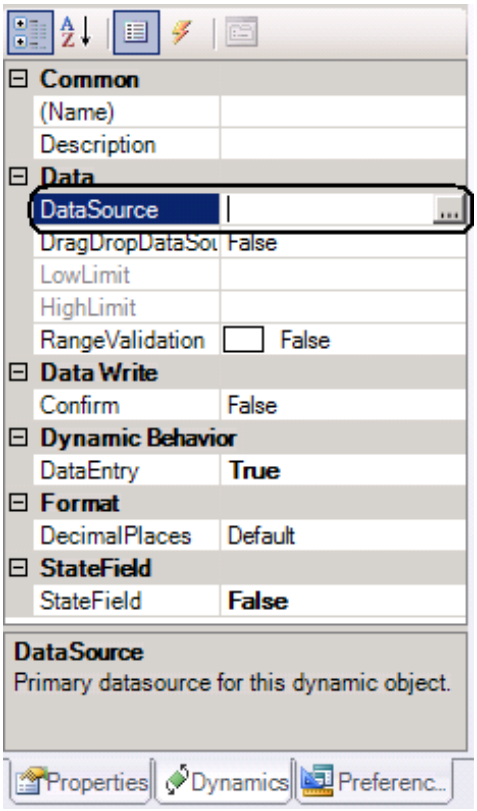


Figure 1.12 - The Data Source Property

For the DataSource property, the ellipsis button opens the ICONICS Data Browser utility, shown in Figure 1.13. (You saw the mesh view of the Data Browser for OPC UA address space previously in Figure 1.8.) The Data Browser has many functions, each of which is isolated on the individual tabs of the dialog box.

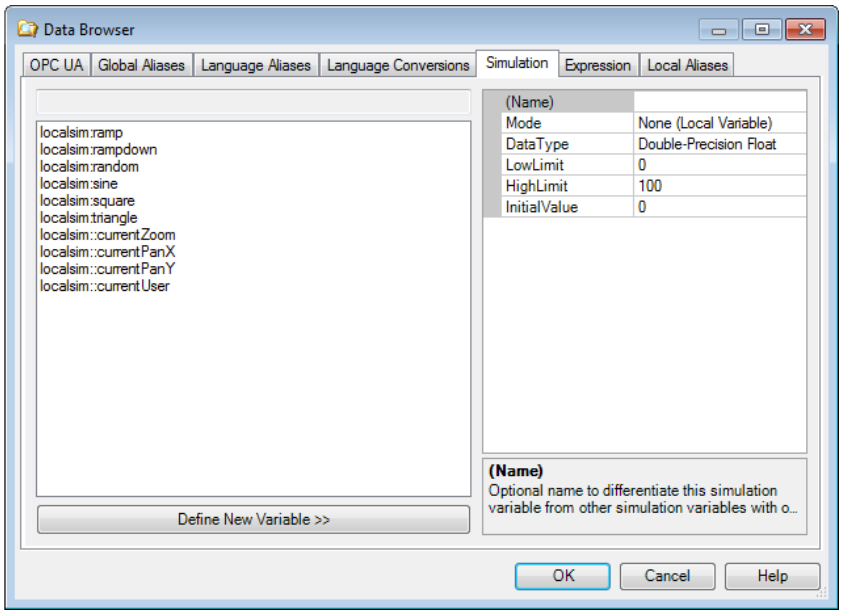


Figure 1.13 - Data Browser: Simulation Tab

For example, the Simulation tab allows you to select one of ICONICS' GENESIS64 simulator functions as the data source: a ramp function, sine wave, random number generator, square function, and so forth. These simulators enter their expressions in the form *localsim::sine* to indicate that the simulator is on the local system; *sine* is the actual name of the function. More formally, *localsim::sine* is an address that GENESIS64 knows how to parse and when you select this function and click the OK button, the Data Browser enters the correct address and syntax back into the Data Source property.

The Data Browser gives you a set of tools for navigating the OPC address space, browsing data sources, and selecting the exact data source (tag) that you require. In Figure 1.14, the OPC UA tab is shown with a tag already selected. Notice that the address to the device tag is displayed in the address bar. (When the address bar isn't selected as it was here, it appears in a breadcrumb style.) When you click OK the Data Browser enters the address in the DataSource property in the correct form using the appropriate syntax.

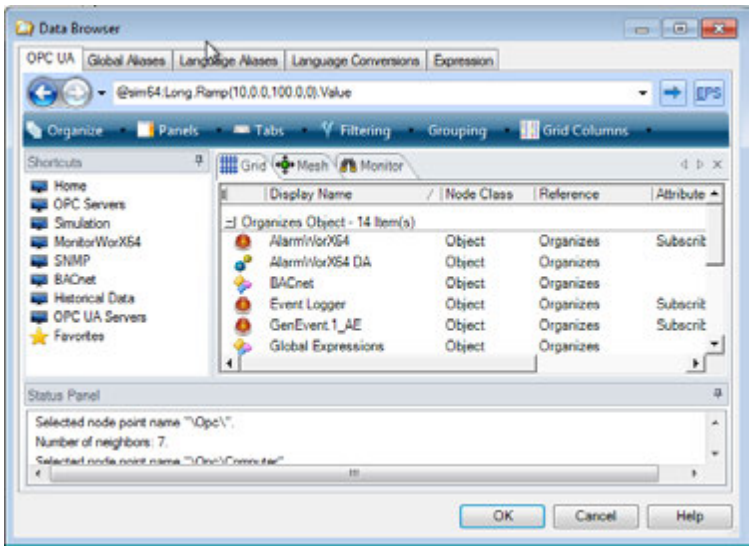


Figure 1.14 - Data Browser: Grid View

The power of the Data Browser is that it lets you know what data sources are actually available and correctly defines the addresses using the appropriate syntax. All data source requests go through the FrameWorX64 Server. However, to keep backward compatibility, the FrameWorX64 server uses GenBroker to communicate to legacy systems (e.g., OPC DA) and expose these data in the OPC UA address space. Obviously, the syntax for a true OPC UA tag is different from a tag coming from a legacy system. The Data Browser frees you from having to know the individual syntax coming from the different servers. You can simply browse for tags; the addresses are entered back into the display configuration in the correct form. Now when you save your configuration and go into runtime, the needle is connected to the tag you just assigned to it and it changes appropriately.

1.5. Microsoft Silverlight and Windows Features

With GENESIS64, you can exploit the features of Microsoft Silverlight and Windows Operating Systems.

1.5.1. Microsoft Silverlight

Microsoft Silverlight is an implementation of the .NET Framework that runs on most browsers. In GENESIS64, you can save files such as GraphWorX64 displays, AlarmWorX64 alarms, GridWorX data, and TrendWorX64 trends in Silverlight format so that they can be deployed on clients running Silverlight. Because not all GENESIS64 features are supported on Silverlight, you should be aware of what is and is not supported before you develop GENESIS64 files that are intended for use on a Silverlight platform. Most importantly, 3D objects, scripting and Smart Symbols are not supported. A full list is provided in the GENESIS64 online Help.

Benefits of the Silverlight version include:

- Easy thin-client deployment. You do not need .NET Framework on client machines, and you do not need to change security settings on clients.
- Faster initialization. For example, a Silverlight GraphWorX64 viewer starts very quickly.
- Files load faster. For example, Silverlight GraphWorX64 displays load significantly faster than in the desktop version of GraphWorX64.
- You can use non-IE (Internet Explorer) browsers, such as Firefox and Safari.
- You can use non-Windows operating systems, such as Macintosh.

The Silverlight plug-in is required. ICONICS can provide a link to download and install this if you do not have the Silverlight plug-in (approximately 5MB download).

1.5.2. Windows Operating System Features

GENESIS64 is supported on even the latest Windows Operating Systems, which offers some added benefits when using GENESIS64 applications, including the task bar, jump lists, and Multi-touch capabilities.

The Windows **task bar** is located along the bottom of your Windows desktop in which GENESIS64 icons appear. Other notification icons in the task bar icon let you know if there is an error message, a security violation, or an informational message. Windows **jump lists** give you one more way to find GraphWorX64 files and related applications.

The GraphWorX64 jump list is available from the GraphWorX64 entry in the Start menu. Figure 1.15 shows examples of a jump list and task bar.

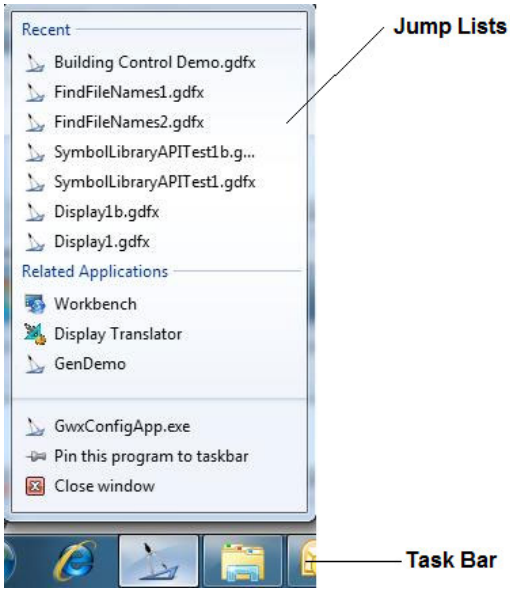


Figure 1.15 - Jump List and Task Bar Examples

Multi-touch capabilities allow runtime users to interact with GraphWorX64, AlarmWorX64, and TrendWorX64 displays directly on a monitor using hand-on-monitor gestures to manipulate objects on the screen. This is especially helpful when a GraphWorX64 display contains an EarthWorX map or a 3D GraphWorX64 display. When a runtime user touches a touch-sensitive Windows enabled device, the Windows Multi-touch Platform sends gesture messages to GENESIS64, allowing the user to **scroll**, **zoom**, **pan**, **orbit or rotate**, **drag-and-drop**, **select** items, and **right-click** to open pop-up menus. Figure 1.16 shows a user zooming in a 3D GraphWorX64 display.

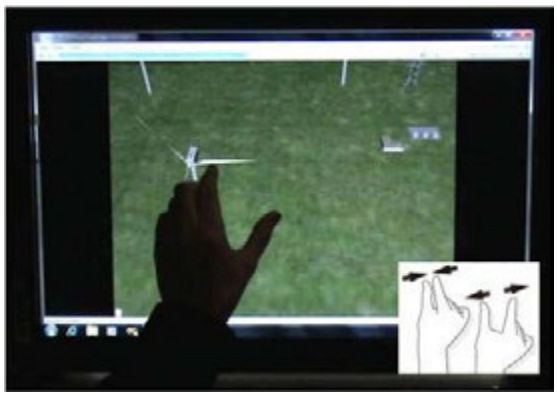


Figure 1.16 – Zooming Using Multi-touch

Lab: Offsite Training Installation and Confirmation *L1*

In this lab, we will install GENESIS64 and apply the necessary files for the rest of the training. If you are attending this training course at one of our training facilities and are using one of our provided machines, you may read this section to familiarize yourself with it, but you do not need to perform the lab.

Estimated time to complete this lab: 90 minutes

1.1. Task Checklist

In this lab, you will complete the following tasks:

- Install GENESIS64
- Install the ICONICS OPC Server Suite
- Apply the training support files
- Create necessary database files

1.2. Support Files

You will need the following file(s) to complete this lab:

- GENESIS64 Installation Disk
- ICONICS OPC Servers Installation Disk
- GENESIS64TrainingSymbol.sdfx
- ButtonsAndSwitches.sdfx
- icTonics.opf
- icTonics-logo.png
- icTonicsUDMConfig.xml
- Translations.txt

1.3. Installation

In this section, we will walk you through the installation of GENESIS64 as well as list software that you should have on the system before you begin the install process.

- Other goals for this lab are to:
    - Install GENESIS64 and SQL Server on the training machine.
    - Understand the privileges inherent to local user "admin" (which you are automatically logged in as) on each training machine.
    - Review and/or adjust user permissions through SQL Server.

1.3.1. Installation Prerequisites

Before we install any ICONICS applications, it is generally a good idea to make sure that the required software has already been installed on the computer. Below is a list of what is recommended you already have installed.

- Internet Information Services (IIS) with FTP components
    - FTP is required when you intend to publish displays for use with WebHMI clients.
- The latest .Net Framework – the GENESIS64 installation will try to detect this component and install it if it is not there. This can take up to an hour to install.
- SQL Server 2005 SP4 Express or higher – configuration database in GENESIS64 can only be an MS SQL 2005 SP4 or higher database.
- User Account Control (UAC) – Although GENESIS64 works with UAC, its best to have it turned off during installation.

Depending on your operating system, the IIS installation may require your Operating System's installation disk. You should obtain this from your IT personnel. Additionally, the path where IIS needs to be installed varies by Operating System. You may want to ask your instructor its required location on the Operating System you will be using.

1.3.2. Installing GENESIS64

First, Insert the GENESIS64 installation disk.

When the splash screen comes up, click on the "GENESIS64 Installation" link, which is circled below. In case the splash screen does not pop up (you may have auto-run disabled), you can go to your DVD drive and double-click on the "Default.hta" file to bring up the splash screen.

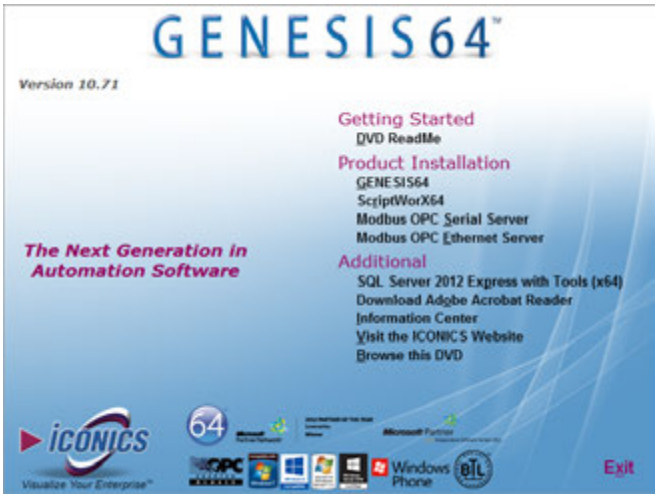


Figure L1.1 - Installation Splash Screen

1. You should then see the **InstallationShield Wizard** as shown below. Click on "Install" to continue.

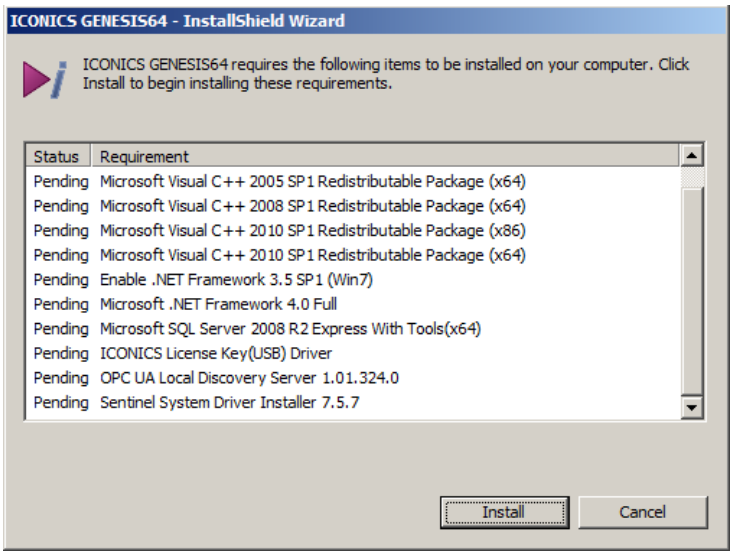


Figure L1.2 - Installation Shield Wizard

Note: If you get any popup security warning messages, just click on "Run" to continue the installation.

2. The installation may take a few minutes and you may see some command prompt windows opening. This is normal.

3. If you don't have SQL Server pre-installed on your machine, you will get a message as seen below. If you are only doing a local installation, just click on "yes" to proceed. If you are doing a corporate installation, you may not need to install the SQL Server now. But you need to ask your IT personnel to provide you with a domain, as well as SQL server user account.

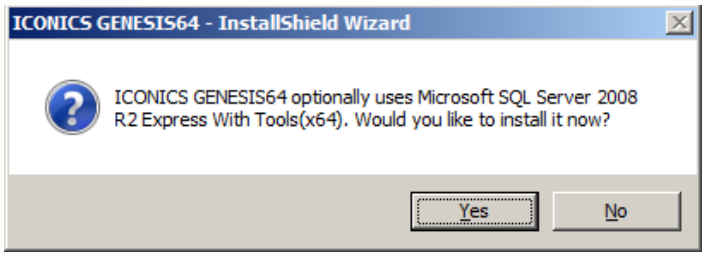


Figure L1.3 – SQL Server Installation

4. Depending on your hardware, you may also come across some prompt windows which ask you to restart your computer. Follow its instructions.

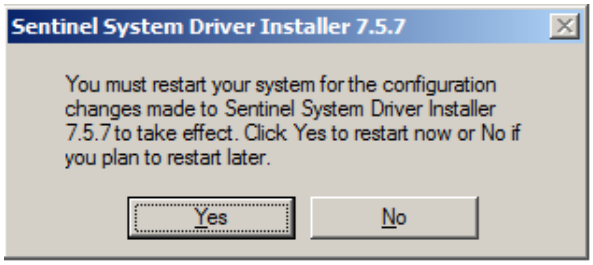


Figure L1.4 – Restart Request from New Driver Installation

5. When the installation pre-requisites complete, you should see the installation **Welcome** screen.



Figure L1.5 - Installation: Welcome Screen

6. Click on "Next" to begin the installation.

7. On the **License Agreement** screen, choose the "I accept the terms in the license agreement" option and click on "Next".

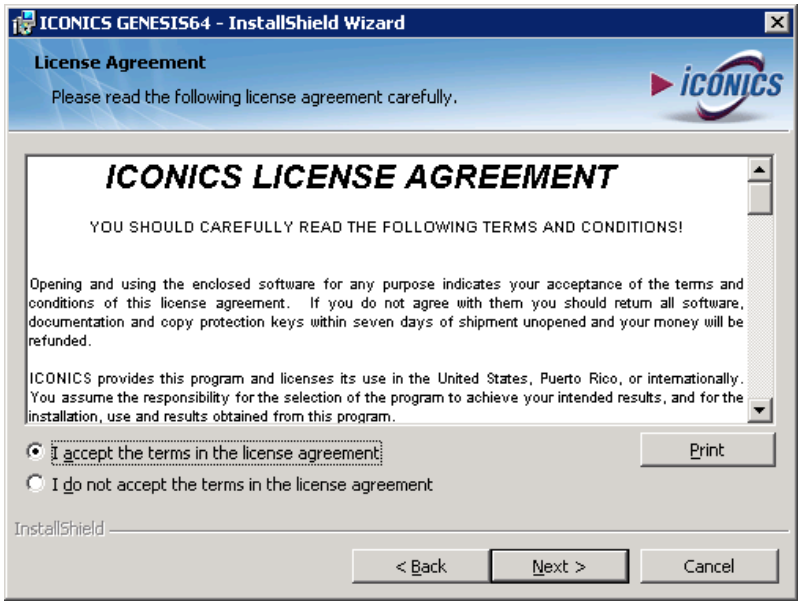


Figure L1.6 - Installation: License Agreement

8. On the **Setup Type** screen, leave the selection at "Typical" and click on "Next".

9. On the **Database Server configuration** screen, your screen should like Figure L1.7. If it's not, you can choose the SQL Server from the dropdown list. If your SQL Server Browser services are not running, you may not be able to see a list of installed SQL Servers, in this case you will need to type in the SQL Server name. If you installed the SQL Server with the GENESIS64 Installation disk the instance name is *hostname*\SQLEXPRESS where *hostname* is the name of your computer.

   Note: You can open up the command prompt from the Windows start menu, and type "hostname" (without quotes) to verify your hostname.

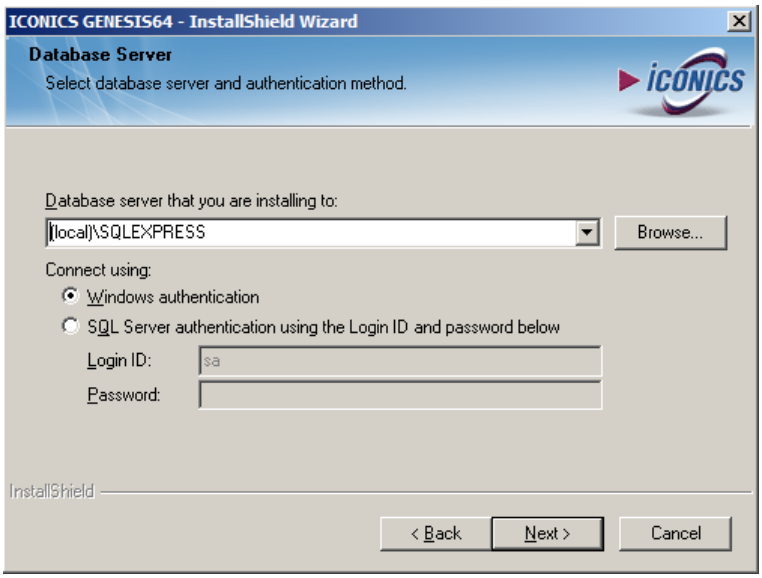


Figure L1.7 Installation: Database Server Configuration

10. On the **Setup type** screen, check GenDemo.

11. On the **Ready to Install** screen, click on "Install" to begin.

12. The installation will take a few minutes. During installation, you will see the **User Setup Wizard**. Click on "Next" to continue.



Figure L1.8 - User Setup: Welcome Screen

13. On the next screen, you will choose a user to run the ICONICS Services. This user must have a password. You can choose an existing local computer user or a Domain user. In this course, we will use the existing user "admin".

14. Give this user a password that conforms to the password policy on the computer and click on "Finish". Note, if you're attending training in Foxborough, please ask the instructor for the username and password.

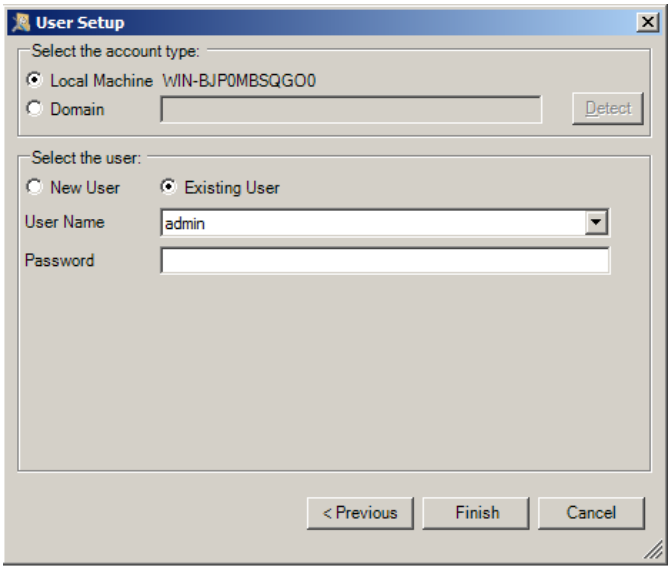


Figure L1.9 - User Setup: User Selection

15. You will then see the **GENESIS64 SQL Databases Installation Wizard**. Click on "Next" in the **Welcome** screen.



Figure L1.10 - GENESIS64 SQL Databases Installation: Welcome Screen

16. Leave the default selection in the **Change common database options** screen and click on "Install".

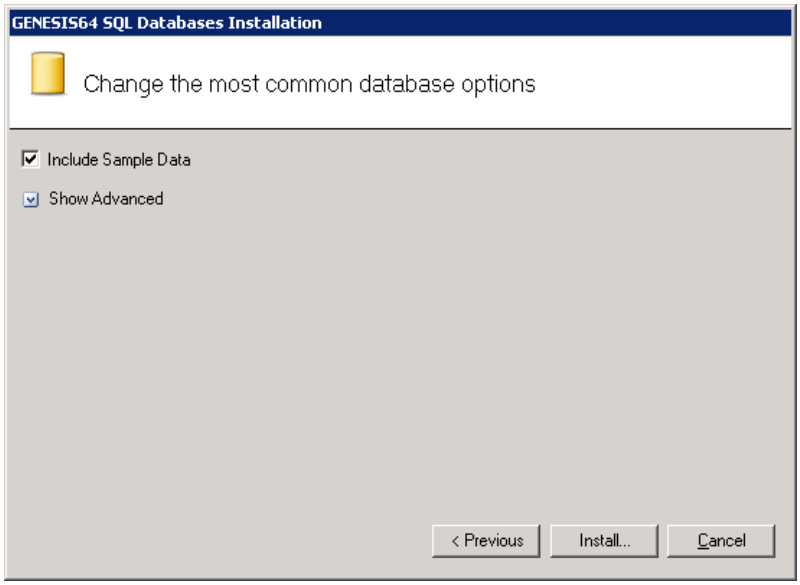


Figure L1.11 - GENESIS64 SQL Database Installation: Change Common Database Options

17. The installation continues and this takes a few minutes.

18. When the installation completes, you will receive a confirmation message as shown in Figure L1.12.

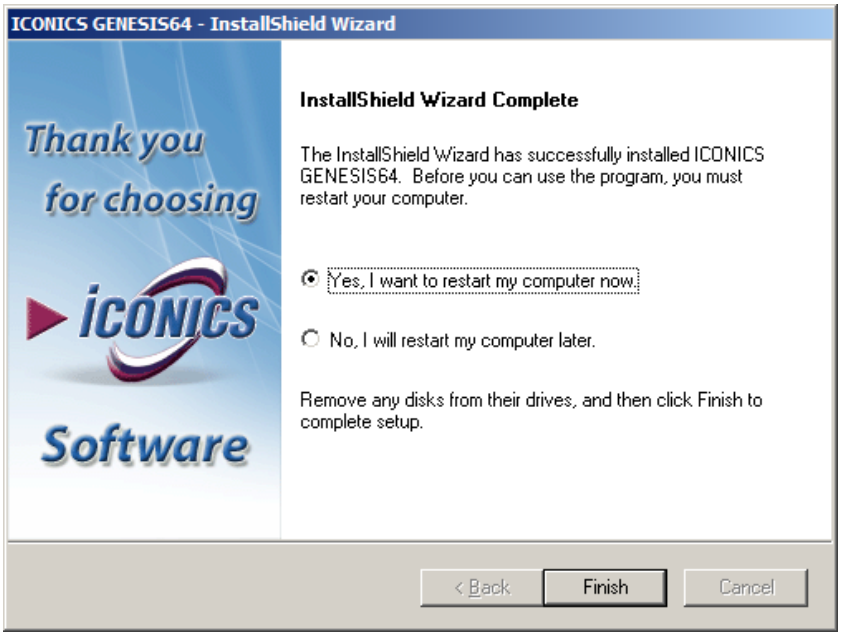


Figure L1.12 Installation: Completed

19. Click on "Finish" to close the dialog box and reboot your computer.

1.3.3. Installing the OPC Server Suite

1. Insert the OPC Servers installation disk.

2. When the splash screen comes up, click on the "OPC Standard Server Suite" link, which is circled in Figure L1.13. In case the splash screen does not pop up (you may have auto-run disabled), you can go to your DVD drive and double-click on the "Default.hta" file to bring up the splash screen.



Figure L1.13 - OPC Servers Installation Splash Screen

3. You should see the **InstallShield Wizard** come up. Click on "Install" to being the pre-requisite installation.

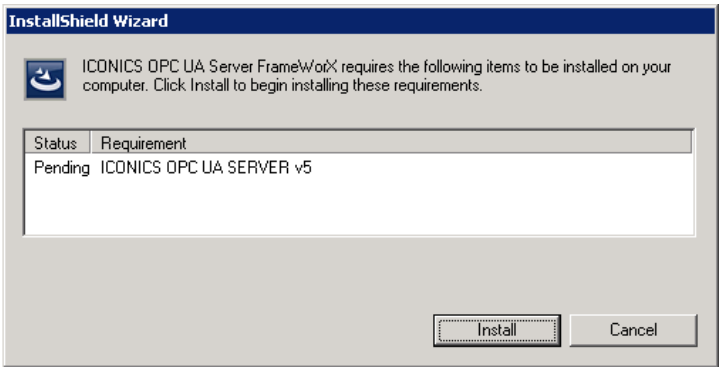


Figure L1.14 - Prerequisites for OPC Server Installation

4. Once the prerequisites are installed, you should see the OPC Server Suite 5 Setup Welcome screen. Click on "Next" to continue.

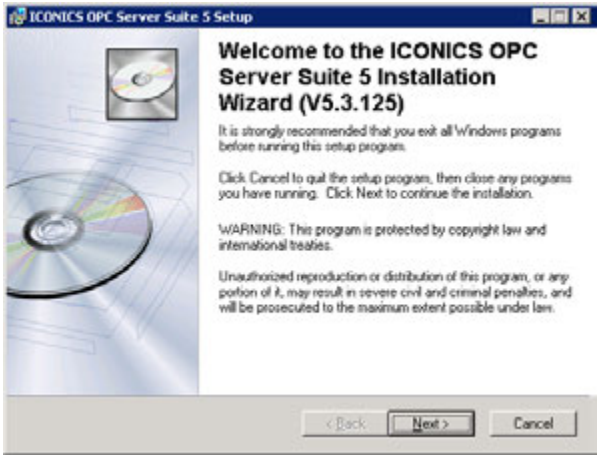


Figure L1.15 - ICONICS OPC Server Suite 5 Setup: Welcome Screen

5. On the Licensing Agreement screen, choose the "I accept the license agreement" option and click on "Next".

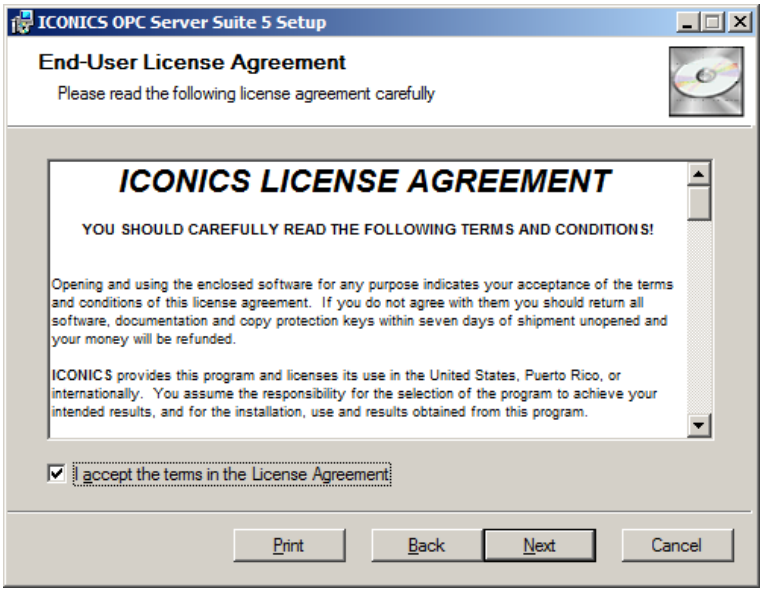


Figure L1.16 - ICONICS OPC Server Suite 5 Setup: License Agreement

6. Leave the Destination folder as default on the next screen and click on “Next” to continue.

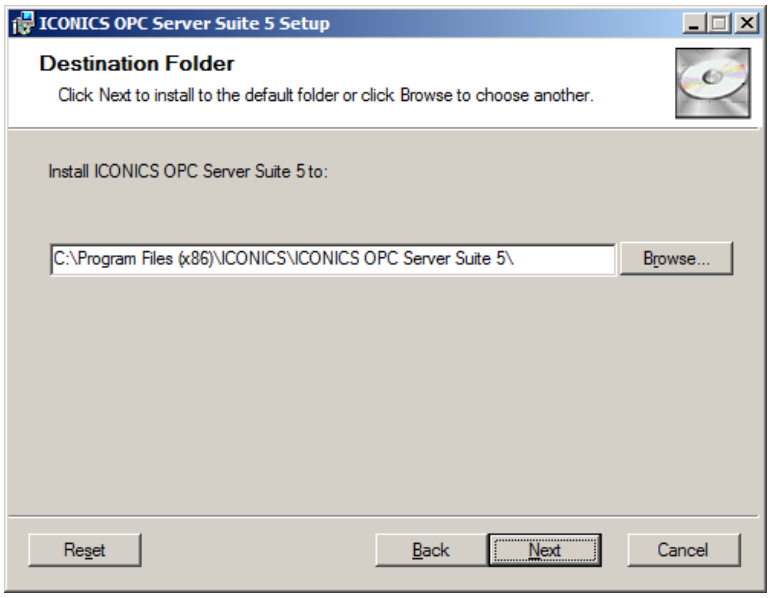


Figure L1.17 - ICONICS OPC Server Suite 5 Setup: Destination Folder

7. Leave the application data folder as default and click on “Next”.

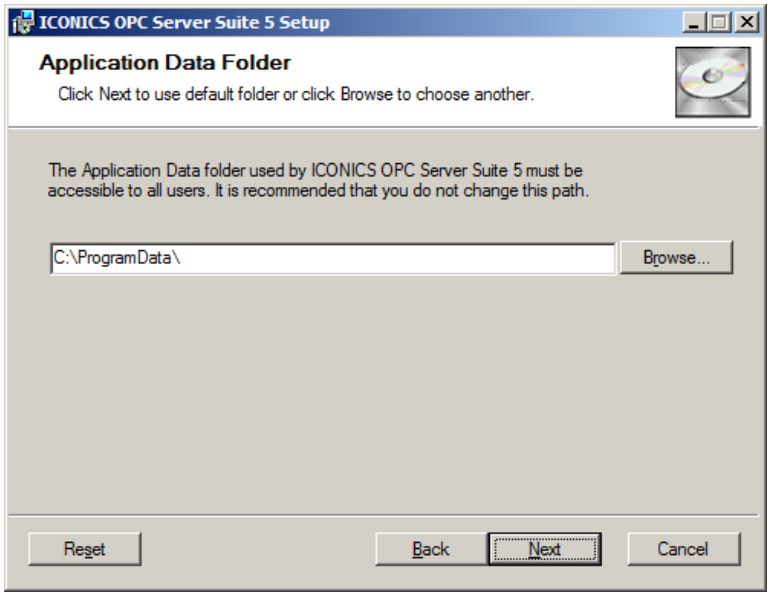


Figure L1.18 - ICONICS OPC Server Suite 5 Setup: Specify Application Data Folder

8. Leave the Shortcut folder as default and click on “Next”.

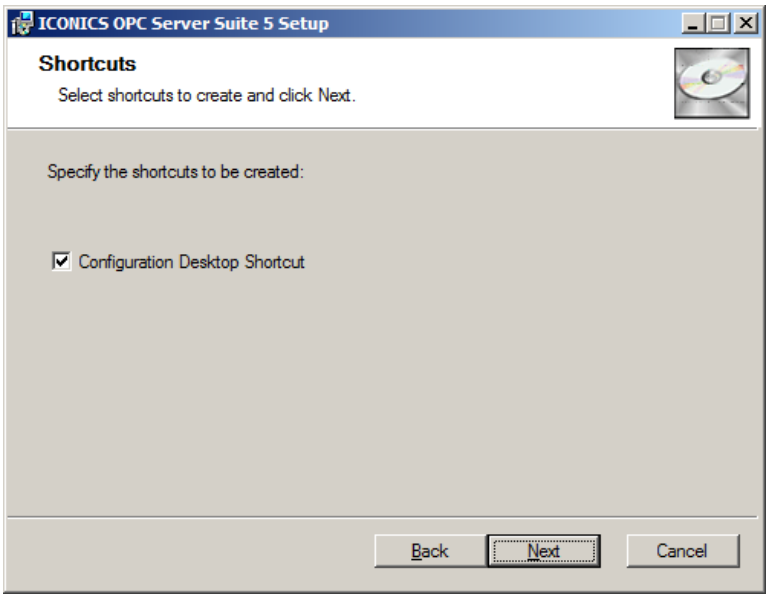


Figure L1.19 - ICONICS OPC Server Suite 5 Setup: Shortcuts Folder

9. On the Select Feature Screen, make sure that all features are selected and click on “Next”

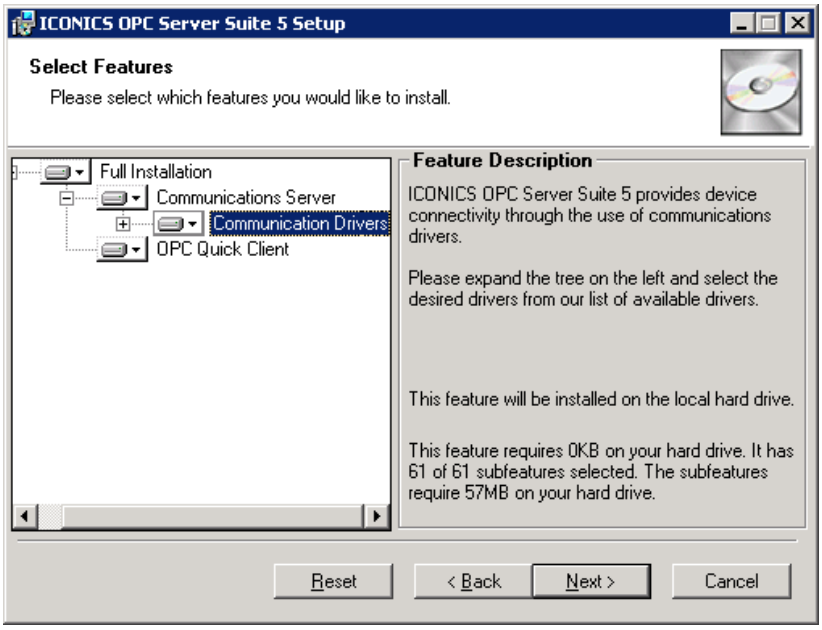


Figure L1.20 - ICONICS OPC Server Suite 5 Setup: Select Features

10. Click on "Next" on the Ready to Install the Application screen to begin installation. This may take a few minutes.

11. When the installation completes, you will receive a successes screen. Click on "Finish" to complete the installation.

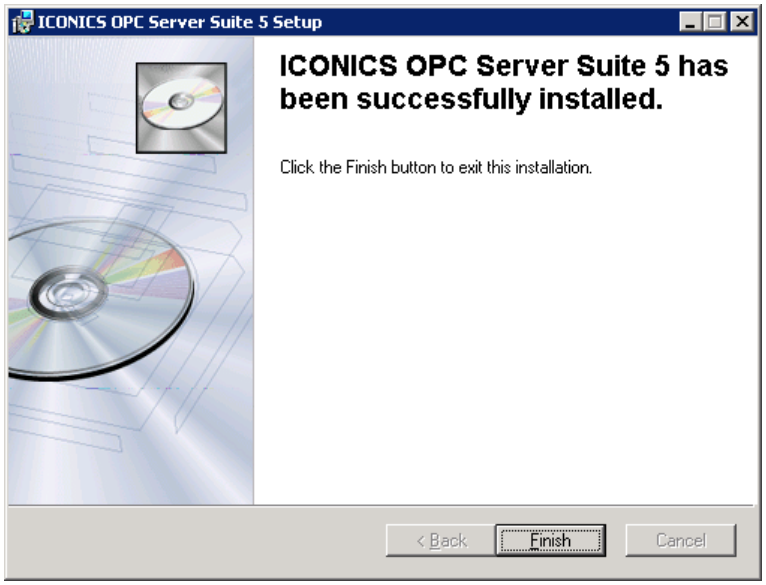


Figure L1.21 - ICONICS OPC Server Suite 5 Setup: Successfully Installed

12. You will then see the **OPC UA Server FrameWorX Installation Wizard** welcome screen. Click on "Next" to continue.

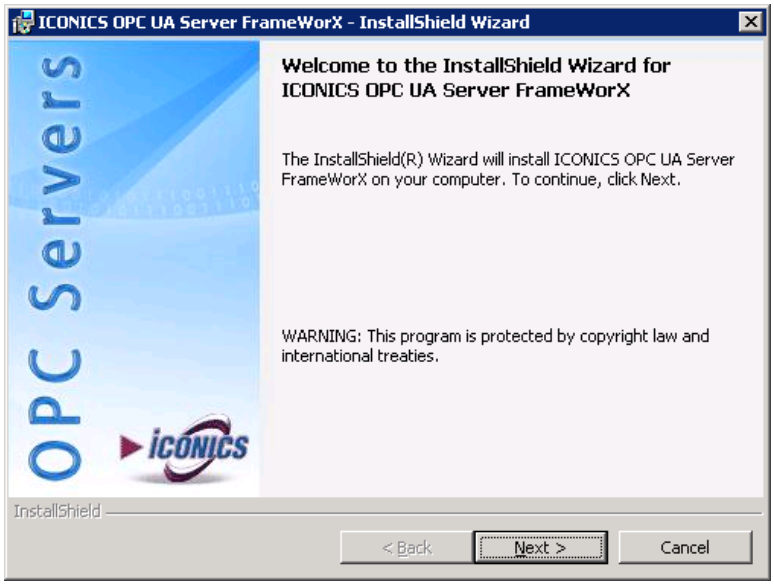


Figure L1.22 - ICONICS OPC UA Server FrameWorX: Welcome Screen

13. Click on "Install" on the Ready to Install the Program screen to begin installation for the OPC UA Server FrameWorX. This will take a few minutes.

14. When the installation completes, you should see a confirmation screen. Click on "Finish" to close the wizard.

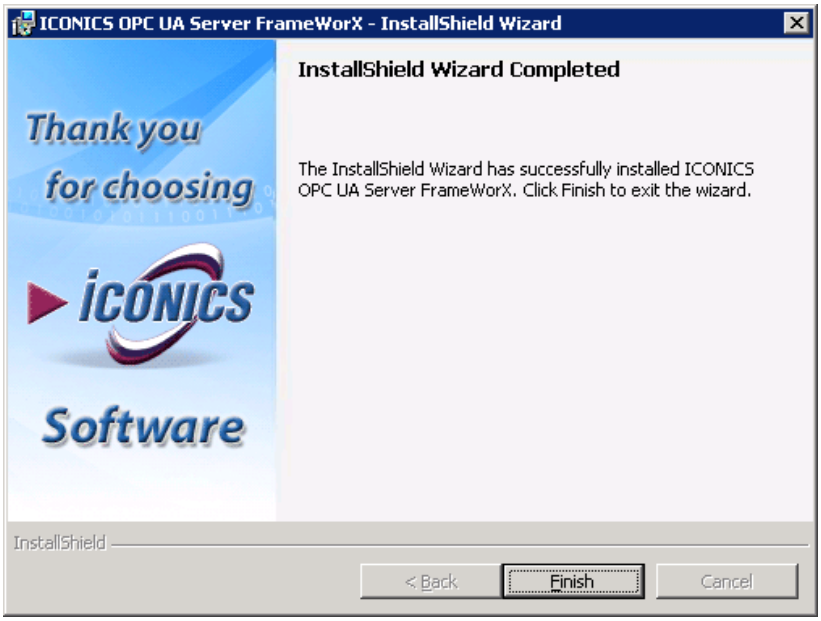


Figure L1.23 - ICONICS OPC UA Server FrameWorX: Installation Completed

15. Click on "Exit" on the splash screen to close it.

1.3.4. Configuring the OPC Server for Training

16. Go to Start → Run and type in "dcomcnfg" (without quotes) in the **Run** dialog box and click on "OK".

17. In the **Component Services** Window, expand Component Services → Computers → My Computer → DCOM Config on the left-hand pane.

18. Find the "ICONICS OPC Server Suite 5.5" component in the list, right-click on it and select "Properties".

19. In the **ICONICS OPC Server Suite Properties** dialog box, select the Security tab.

20. Change the Access Permissions to “Use Default”.

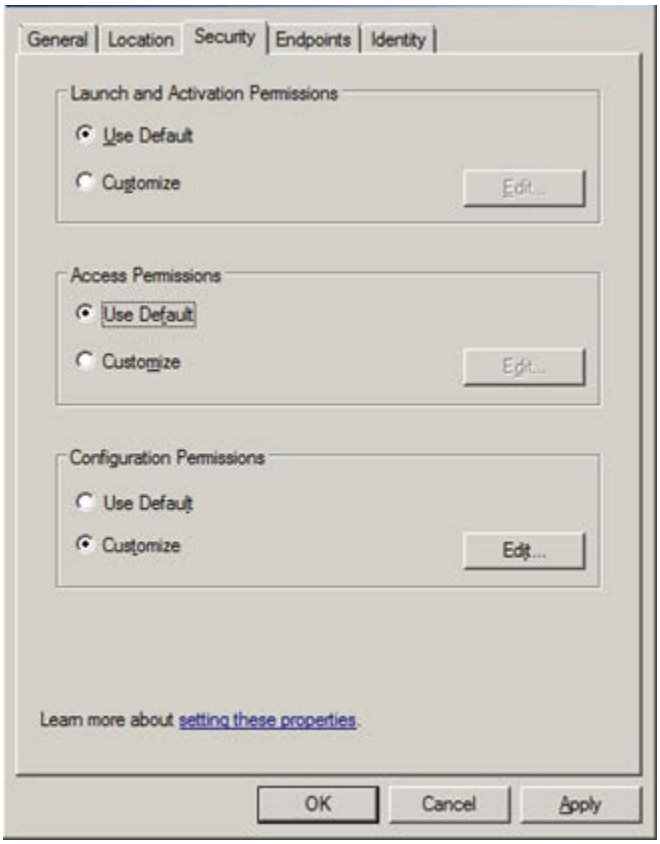


Figure L1.24 - ICONICS OPC Server Suite Properties: Security

21. Close the **Component Services** window.

1.4. Training Support Files

There are a few support files that you will need in order for the training to run smoothly. Again, if you are using one of the provided training computers, you do not need to do the steps in this section.

1. Open Windows Explorer and navigate to C:\

2. Create a folder here called “Training”. This folder will be your project folder and you will store all of the training files into this folder.

3. Copy the following files into the C:\Training directory:

- GENESIS64TrainingSymbol.sdfx
- icTonics.opf
- icTonics-logo.png
- icTonicsUDMConfig.xml
- Translations.txt

Note: Ask your instructor for these files if you do not have them.

- **Symbol Library**

4. Start GraphWorX64 by going to Start → All Programs → ICONICS → GENESIS64 → GraphWorX64 → GraphWorX64.

5. Once GraphWorX64 opens, you should see the Explorer to the left-hand side. At the bottom of the explorer, you should see a tab that says "Symbols". Select this tab.

6. Click on the "Symbol Library commands menu"  icon at the top of the explorer and select "Open Existing Category" from the popup menu.

7. In the **Open** dialog box, look for the GENESIS64TrainingSymbols.sdfx in C:\Training. Click on "Open" to add the library.

8. If you click on the dropdown list of the Symbol library now, you should see the GENESIS64TrainingSymbols category listed.

9. Perform steps 7 and 8 again to also add the category ButtonsAndSwitches.sdfx.

- **Unified Data Manager**

10. Start Workbench64 by going to → All Programs → ICONICS → Workbench.

11. Once Workbench is opened, click on the "Unified Data Manager"  icon to load it's configuration in the Project Explorer.

12. Right-click on "localhost" and select "New Database".

13. In the **Create Database Wizard** pop-up box**,** click on "Next" to get to the change common database options screen.

14. Change the Database Name to "**icTonicsUDMConfig**".

15. Uncheck the "Include Sample Data" checkbox. Your **Create Database Wizard** should look similar to Figure L1.25.

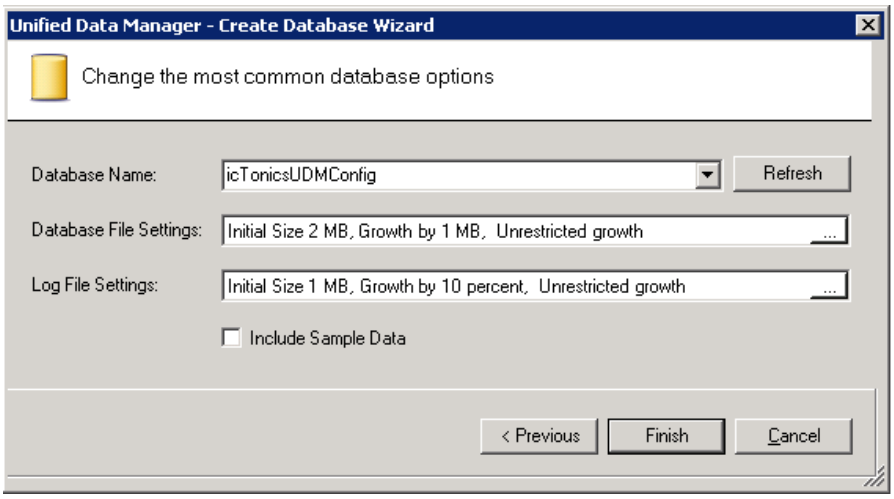


Figure L1.25 - Create Database Wizard: Common Database Options

16. Click on "Finish" to create the database. This may take a moment.

17. Once the database is created, right-click on "icTonicsUDMConfig" and select "Import".

18. In the **Open** dialog box, go to the C:\Training folder and look for the "icTonicsUDMConfig.csv" then click on "Open" to import the file. Leave the default settings in the "Import from" dialog box and click on "Start".

19. Once the file imports, right-click on "icTonicsUDMConfig" and select "Make Active."

20. You should see icTonicsUDMConfig (Active) in the navigation tree and the icon to the left should contain a green element.

21. Close the Workbench.

- **OPC Server Configuration**

22. Start the ICONICS OPC UA Server 5 Configurator by going to Start → All Programs → ICONICS → ICONICS OPC Server Suite 5 → **ICONICS OPC Server 5 Configuration.**

23. Go to File → Open and browse to C:\Training.

24. Select the "icTonics.opf" file and click on "Open".

25. Select "Yes, Update" if you see the following screen in Figure L1.26.

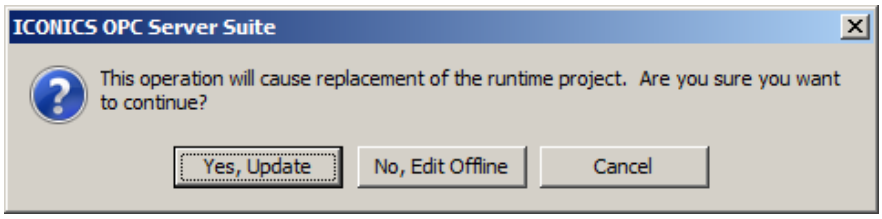


Figure L1.26 – Update OPC Server

26. Close the **ICONICS OPC Server Suite Configurator**.

27. Open the **OPC UA Configuration** by going to Start → All Programs → ICONICS → ICONICS OPC Server Suite 5 → OPC UA Configuration.

28. On the Server Endpoints, select the "opc.tcp://127.0.0.1:49460" endpoint and uncheck the "Enable" checkbox near the bottom of the dialog box.

29. Select the "opc.tcp://*ComputerName*:49460"endpoint where *ComputerName* is the name of your computer.

30. Click on the "Edit" button near the bottom of the dialog box to bring up the **Endpoint Definition** dialog box.

31. Uncheck the "Basic128Rsa15" and "Basic256" options. The dialog box should now look like Figure L1.27.

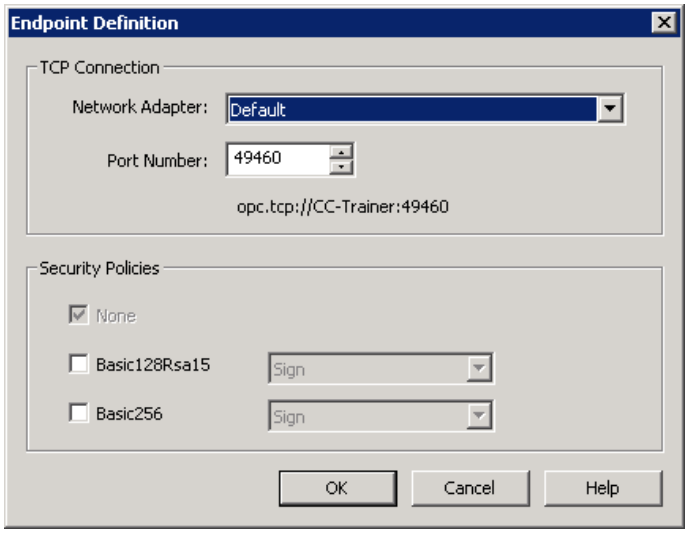


Figure L1.27 - Endpoint Definition

32. Click on "OK" to close the dialog box.

33. Check the "Enable" checkbox at the bottom of the **OPC UA Configuration Manager** and click on "Close" to save the settings and close the dialog box.

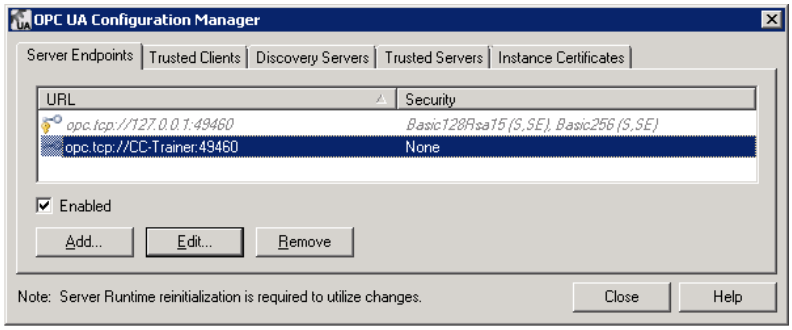


Figure L1.28 - OPC UA Configuration Manager

1.5. Creating Necessary Logging Databases

In this section, we will create the necessary databases that you will use in the course to log Trend and Alarm data. In this document, we will assume that you are using SQL 2008. The steps for creating databases in SQL 2005 should be similar.

1. Open the SQL Server Management Studio by going to Start → All Programs → Microsoft SQL Server 2008 R2 → SQL Server Management Studio.

2. Once the Management Studio opens, you will receive a dialog box that asks you to connect to a SQL Server. If this is the first time you are using the SQL Management Studio, you may receive a **welcome** and **setup** dialog box prior to the **connection** dialog box. You can use the default credentials that are already filled in the dialog box. Take note of these credentials. Click on "Connect" to connect to the server.

Note: If the server name is blank, enter: "(local)\SQLEXPRESS" (without quotes) into it.

3. Once you are connected, expand your server instance → Databases in the Object Explorer on the left-hand side.

4. Right-click on Database in the Object Explorer and select "New Database".

5. In the New Database dialog box, type in "icTonicsAlarmData" in the Database name field.

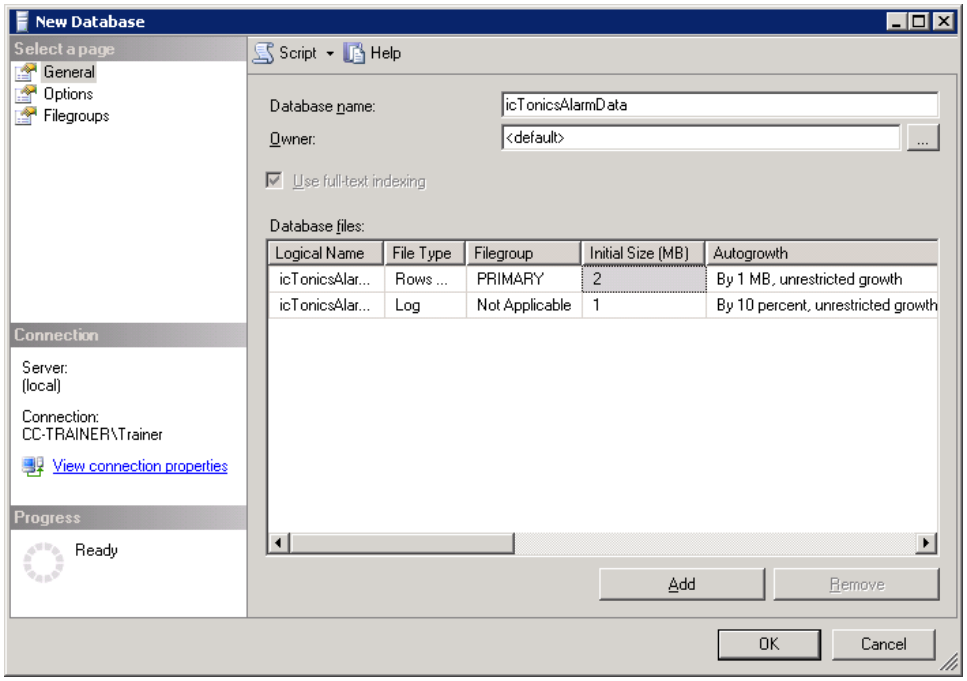


Figure L1.29 - Creating a New SQL Database

6. Click on "OK" at the bottom of the dialog box to create the database. This may take a few seconds.

7. Repeat steps 4-6 to create another database called "**icTonicsTrendData**".

1.6. SQL Server Permissions Review

1. Go to Start → All Programs → Microsoft SQL Server 2008 R2 → SQL Server Management Studio.

2. Within the management studio, expand Security → Logins, then double-click on *Hostname\Username* (in your case, the username should be "admin") and then navigate to User Mapping.

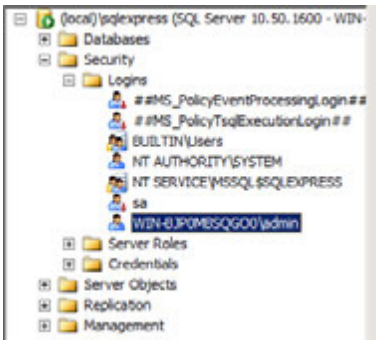


Figure L1.30 – Admin User Account in SQL

3. Make sure the following databases have been checked: AwxLogger, AwxServer, TwxConfig, and icTonicsTrendData. Also, you need to confirm that the role membership for each one of the databases is db_owner, seen in the figure below.

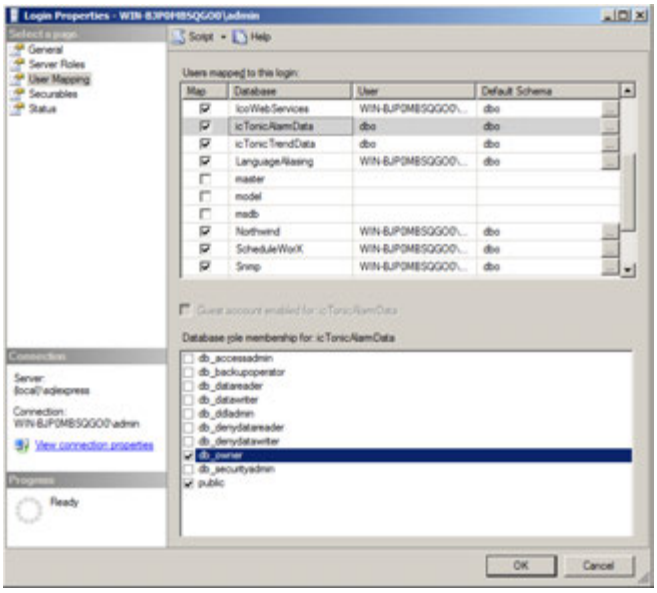


Figure L1.31 – Authorize Admin User Permission

Review: Introduction to GENESIS64

R1

1.1. Objectives Review

You should now know and understand the following:

- What GENESIS64 is and how it is used to create automation control systems for observing and controlling equipment
- What the different component applications in GENESIS64 are and where they are used
- What are the different types of OPC specified data

1.2. Summary

GENESIS64 is automation observation, analysis, and control software that allows you to build HMI and SCADA systems. Based on open standards developed by the OPC Foundation for Windows systems, GENESIS64 works with the newest standard of data called OPC UA and maintains backwards compatibility to the older standards such as OPC DA, OPC AE, OPC HDA, and others.

GENESIS64 is based on the Microsoft .NET Foundation and utilizes the Windows Communication Foundation (WCF) and Windows Presentation Foundation (WPF) to create state-of-the-art Windows Service Oriented Architecture (SOA) solutions.

All GENESIS64 applications can be configured within a framework application called the Workbench. You can display process data in a graphical display in real time using GraphWorX64, organize assets hierarchically with AssetWorX, monitor events and alarm conditions with AlarmWorX64, log data for analysis with TrendWorX64, show database data with GridWorX64, and display geographical information in a map based format using EarthWorX64.

The GenDemo provides industry case studies and other examples so you can envision ways in which GENESIS64 can be used. A Learning Center lets you familiarize yourself with some of the tools that are available.

1.3. Terms

Address Space	AlarmWorX64	AssetWorX
Configuration Mode	EarthWorX64	GenBroker
GenTray	GraphWorX64	GridWorX64
HMI	OPC AE	OPC Client
OPC DA	OPC HDA	OPC Server
OPC and PLC	Project File	Runtime Mode
SCADA	Tags	TrendWorX64
Unified Data Browser	Workbench	

1.4. Test Your Knowledge

In this chapter, we learned some basic concepts of PLC, OPC, and GENESIS64. Before you continue to the next chapter, be sure that you can answer the following questions.

1. OPC refers to all of the following except:

   a. A data communications protocol
   b. OLE for Process Control
   c. A standards body
   d. A Microsoft networking protocol

2. All of the following are true except:

   a. The Workbench is used to configure displays and services.
   b. GraphWorX64 can only be run in the Workbench.
   c. FrameWorX64 is an OPC UA server.
   d. AssetWorX, TrendWorX64, AlarmWorX64, GridWorX64, and EarthWorX64 viewers can be placed into a GraphWorX64 display.

3. True or False: A tag allows you to display OPC data in a client application.

4. During configuration, which of the following is NOT true?

   a. A display can be modified or edited.
   b. Sample data can be displayed so that the display can be previewed.
   c. A runtime button is available to go into the runtime mode.
   d. You can display live data for short periods of time.

5. Which of the following is true of OPC UA?

   a. It can use COM/DCOM in its legacy mode.
   b. It is a hierarchically organized object environment only.
   c. The object model supports one and only one namespace.
   d. All of the above
   e. None of the above.


ICONICS

Workbench 2

GENESIS64 comes with a centralized project management tool: the Workbench. This tool is the single application that you need to configure and run the different aspect of your HMI project. Furthermore, this application is based on .NET technology, allowing you to make changes to your project over the internet. In this chapter, we are going to introduce you to Workbench. You will learn about the different providers available through Workbench as we progress through the course. After completing this chapter, you should be familiar with:

- The basic features of the GENESIS64 Workbench
- How to customize the Workbench
- How to create an ICONICS project using the Workbench
- Techniques for arranging and working with files in the Workbench
- Ways to share a project among development team members
- Deploying a project for a customer

2.1. Introduction to the Workbench

The Workbench is the centralized Web-based environment for GENESIS64 products. It provides a configuration environment that runs multiple applications and customized third-party environments. Using the Workbench, you will be able to:

- Configure web-based components. Multiple users can configure databases and application servers at the same time.
- Launch runtime mode for simulating and testing and running project components
- Manage your projects
- Deploy your projects

The Workbench, shown in Figure 2.1 in its basic form, provides a frame in which you can use every component of GENESIS64. In addition to being a configuration tool, the Workbench can also act as an operator interface and project management tool.

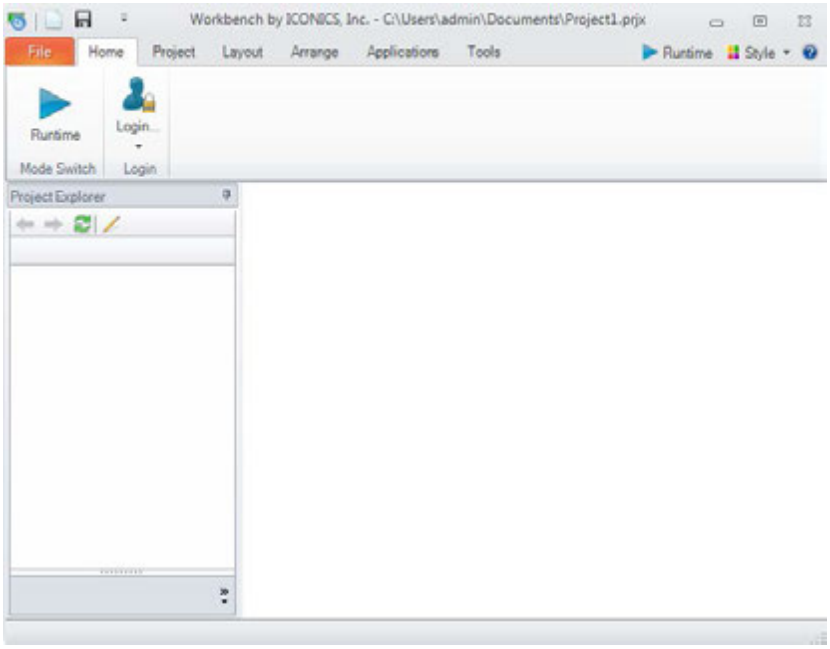


Figure 2.1 - Empty Workbench

All GENESIS64 configurations can be done from within the Workbench, allowing you to learn common techniques that you can apply to set up servers, tools, utilities, and even applications. This design economizes your engineering learning time and overhead costs. The ability to configure remotely lets multiple users configure multiple application servers concurrently and administer multiple databases simultaneously.

Using GENESIS64 applications in the Workbench, you will use its configuration and runtime operations. The Workbench:

- Provides a framework for project management and security (security will be covered later in this course),
- Can load various providers or plug-ins that read from and write to graphics files,
- Can read and write to SQL or system databases,
- Can load configuration forms that start and stop services, set environmental variables, and more,
- Provides a browsing function, a display or design function, and a command set through the ribbon and menu commands that change with the application you're working with.

When you open the Workbench for the first time you see all of the applications, tools, utilities, and services listed in the Project Explorer that you can possibly open within the Workbench. Figure 2.2 illustrates what is available. (Note that if you have a Hyper Historian installation, the set of components is slightly smaller than the one you see listed here.)



Figure 2.2 - Workbench Components

As shown in Figure 2.3, the Workbench is made up of three general areas: the Project Explorer, the ribbon menu bar, and the content area.

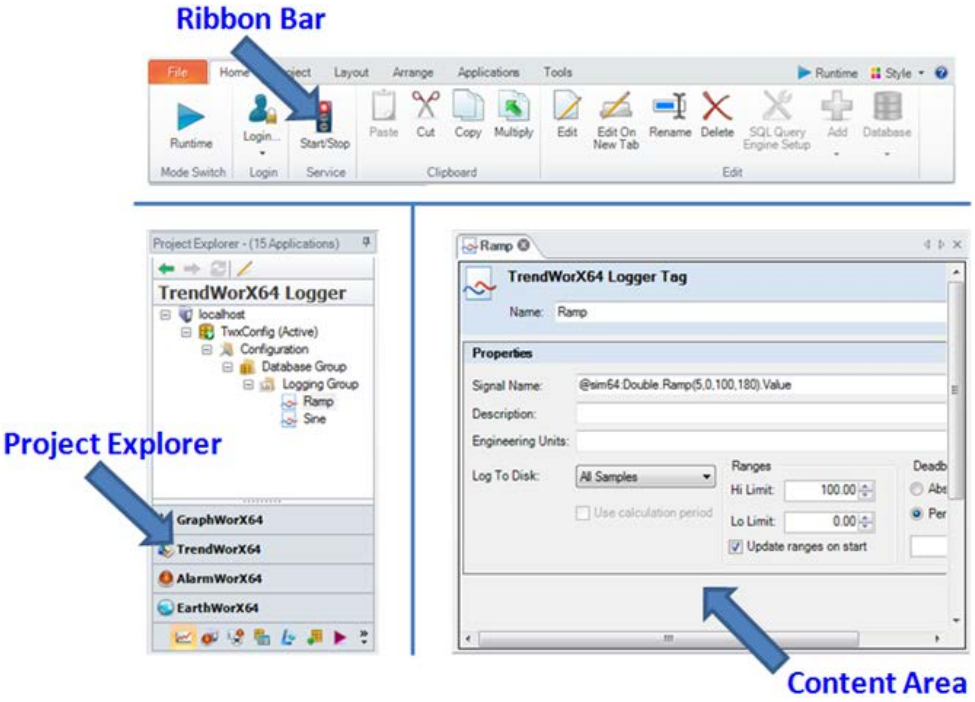


Figure 2.3 - Different Sections of the Workbench

2.1.1. Project Explorer

The Project Explorer encapsulates the ICONICS project that you are working on and its content defines the breadth of a project. The Project Explorer is also the navigational component you use to browse the project's content. It has a navigation tree for quick access to data as well as quick access to different applications. You can select any application that is in the project and navigate its data. Button bars and icons also make it easy to access applications from inside the Workbench.

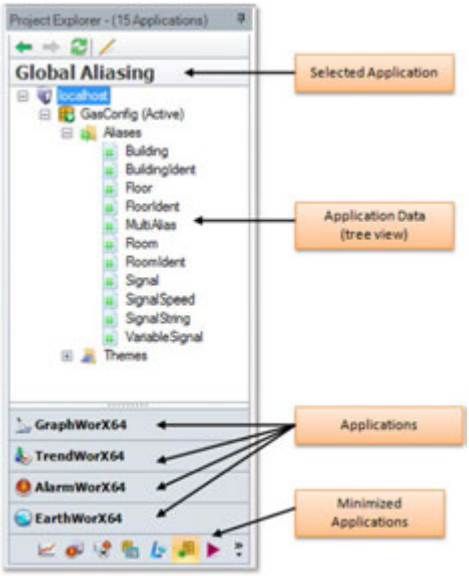


Figure 2.4 - The Project Explorer

2.1.2. The Ribbon

The Ribbon is designed to help you quickly find the commands you need to complete a task. Commands are organized in logical groups under tabs. Each tab relates to a type of activity, such as project management or page layouts.

Each ICONICS application or tool that you open in the Workbench has its own ribbon; as you progress through this course, you will see each GENESIS64 application's ribbon commands. As you work in the Workbench, you will learn how to integrate custom actions and tools inside the ribbon.

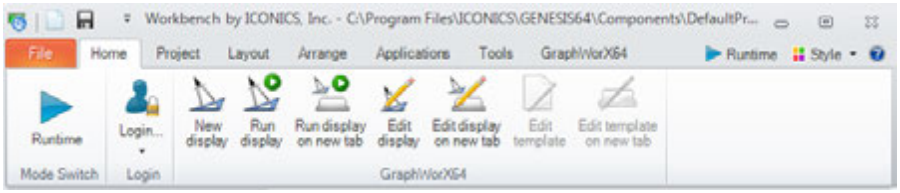


Figure 2.5 - TrendWorX64 Ribbon in the Workbench

For each configuration in providers you are allowed to Export and Import settings in every level whole configuration or just part. When you have opened some provider's configuration, you can see Import and Export button in Home Ribbon menu, like is showed in Figure 2.6.



Figure 2.6 - TrendWorX64 Ribbon in the Workbench

Above the ribbon bar is a quick access toolbar containing, by default, icons for the New and Save commands. By using the drop-down menu located directly above the Project menu (shown in Figure 2.7), you can add icons to the quick-access toolbar and place the quick-access toolbar below the ribbon bar.

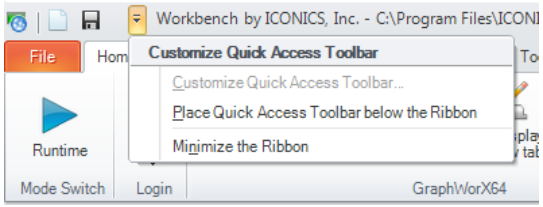


Figure 2.7 - Quick Access Toolbar

Using this same menu, you can also minimize the ribbon so that only its menu items display. You can also do this by double-clicking on the ribbon tabs.

2.1.3. Content Area

The Content area is your workspace. As you proceed through this course, you will make heavy use of the content area when working with GENESIS64 applications. We will cover the content area in detail as you proceed. In this chapter, we will focus on how you can organize your work inside the content area.

2.2. Customizing the Workbench

Some customization options for the content area are available from the "Options" button on the File menu. You can identify startup options: a project to be loaded into the content area when you start the Workbench, and can also specify the layout. You can also configure a **Home page**, to display in the Content area.

The Home page can be displayed only from the startup options. Because the Workbench has a browser-type interface, you can configure the Home page to connect to any site you like, such as your company's RSS, and specify a refresh rate. Maybe you have your own site that provides shortcuts or puts information at your fingertips. You can also select a background color or image for the content area. These settings are stored locally, and are applied to your machine, not to any specific project.

You can also choose various color schemes using the Style button at the top right of the ribbon bar.

2.2.1. Moving Explorers and Other Tools

While working with ICONICS applications in the Workbench, you will see that there may be additional tools and Explorers available. You may want to reorganize tools for your preferences to maximize productivity. In this section, we will detail some methods that you can use to organize and customize where your tools reside.

2.2.1.1. Dragging and Dropping a Tool

You can drag any tool or Explorer around by its title bar. Drop it wherever you like in the content area and it will stay there, as shown in Figure 2.8, as a tool that floats freely over the Content area.

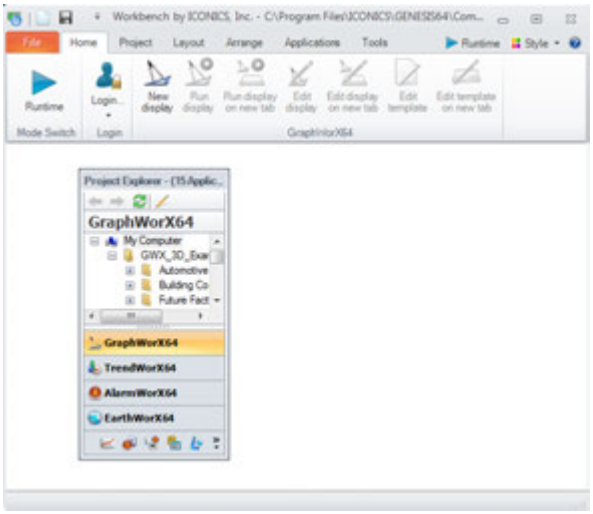


Figure 2.8 - Free Floating Project Explorer

2.2.1.2. Docking a Tool

You can also dock the tool, which locks it into its own space adjacent to the Content area. As you move a tool around, a diamond dock appears, showing you the docking stations in the center and in the north, south, east, and west positions in the Content area as shown in Figure 2.9. (If files are open in the Content area, these docking stations may be difficult to see.)

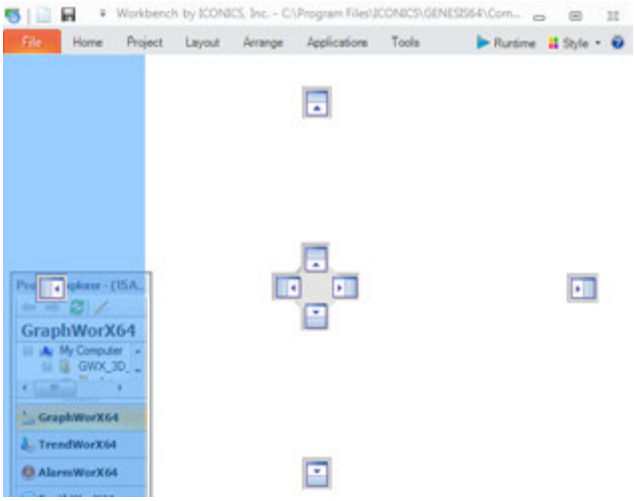


Figure 2.9 - Docking a Tool

As you move the object and your cursor closer to a docking station, a blue area appears (on the left in Figure 2.9) in the Content area, showing you where the tool you are dragging will display if you release the mouse button.

2.2.1.3. Auto-hide a Tool

Tools like the Project Explorer have an auto-hide feature that lets you tuck them into the margin of the Workbench until you want to use them again. To auto-hide the Explorer, click on the pin in its title bar. To expand a hidden Explorer or other tool, click on it where it is hiding in the margin. To lock the tool open, click the pin while it is expanded.

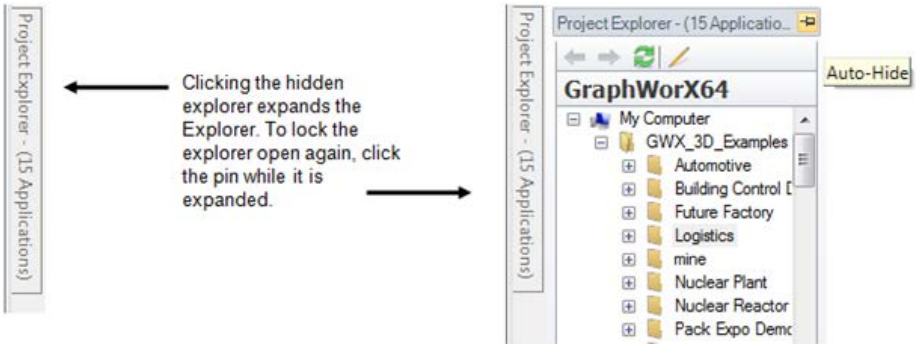


Figure 2.10 - Auto Hide (Left) and Pinned Down (right) Project Explorer

2.2.2. TraceWorX64 and Licensing Tools

Also available from the Workbench's Tools ribbon and from its File menu → Tools item, there are additional tools: Licensing, Redundancy, TraceWorX, and Default FrameWorX Server Location. The **Licensing** tool displays all ICONICS licenses currently in place and in use at the site. **Redundancy** lets you display redundant servers that are configured for your system, and see the servers that are currently on line alongside their stand-by servers.

You can use **TraceWorX64** to log trace files and configure collection parameters on any application if you suspect malfunction.

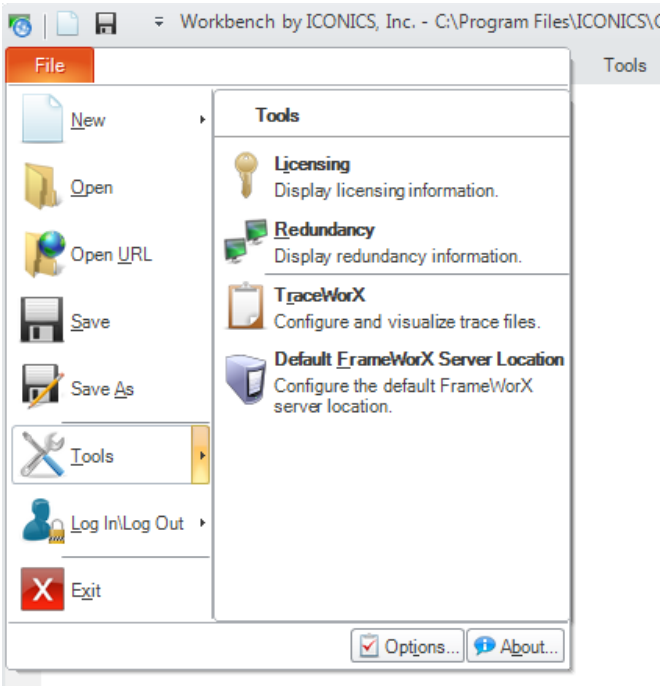


Figure 2.11 - Accessing TraceWorx64 and Licensing

2.3. Creating a Project

The project is one of the fundamental concepts of the Workbench. A project is a set of applications and related resources that execute inside the Workbench.

The best way to understand a project is to create one. You can do this by launching the Workbench from the Windows **Start → Programs → ICONICS GENESIS64 → Workbench**. In this course, the Workbench launches with the default ICONICS project, which includes all GENESIS64 components, along with sample files.

When you create a new project, you can create it using the ICONICS default project as a starting point, or from scratch. To create the project from scratch, click the File menu and select **New → Empty Project**, which is shown in Figure 2.12.

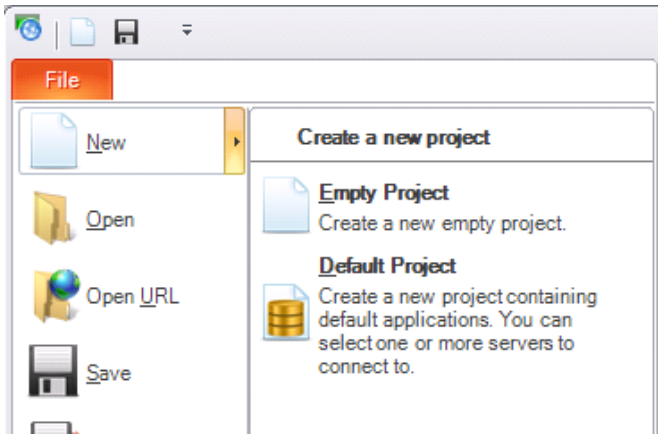


Figure 2.12 - Creating a New Project

2.3.1. Adding Applications for the Project

Once you have selected to create a new Project, the Add/Remove Project Application dialog box appears. Here, you can add any ICONICS tools and applications required by the project by clicking the Add button. You can incorporate any application you need into the project. For example, if you need to define alarms, you would add the AlarmWorX64 application to the project; it appears in the Applications pane on the left.

When you select an application from this pane, the application's properties appear on the right, as shown in Figure 2.13. You can modify the properties of the application to your preference. For help with a property, select it and read the tip that appears below it.

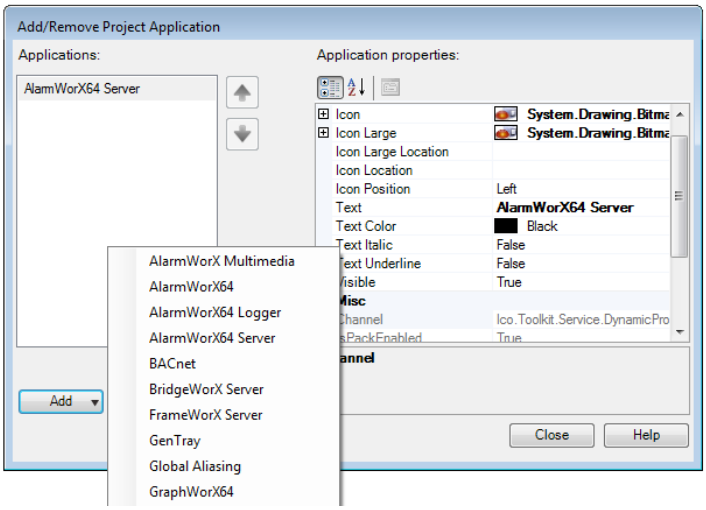


Figure 2.13 – Adding GENESIS64 Applications to a Project

Once you have added all the applications you want to use for the project, click "Close" to exit the dialog and begin working in the project.

2.3.2. Changing Applications in the Project

You can add or remove applications in a project or change their properties at any time by reopening the Add/Remove Project Application dialog box. Click on one of the Edit Project buttons show in Figure 2.14.

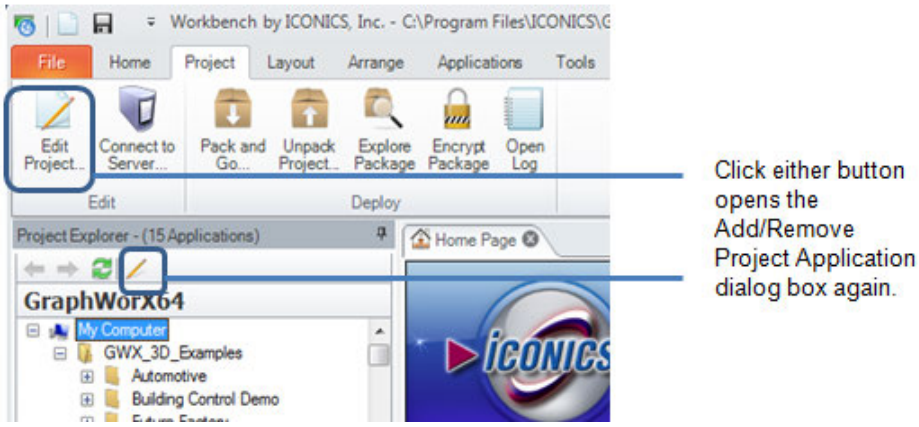


Figure 2.14 - Edit Application List

Once the project's applications are listed in the Project Explorer, you can get to work. But first, let us go over some ways you can manage the applications within the Project Explorer.

2.3.3. Reducing/Enlarging the Application Stack

Applications are stacked as buttons in the Project Explorer. You can move the button stack out of your way by grabbing the top of the stack (circled in Figure 2.15) and dragging the stack up or down. Notice how the icon tray (below the stack) expands and shrinks as you do this.

Note: If your computer is configured to display text and objects at 125% or 150% you may be unable to interact with the stack layout (you can check this in the Control Panel under Appearance and Personalization options for the Display).

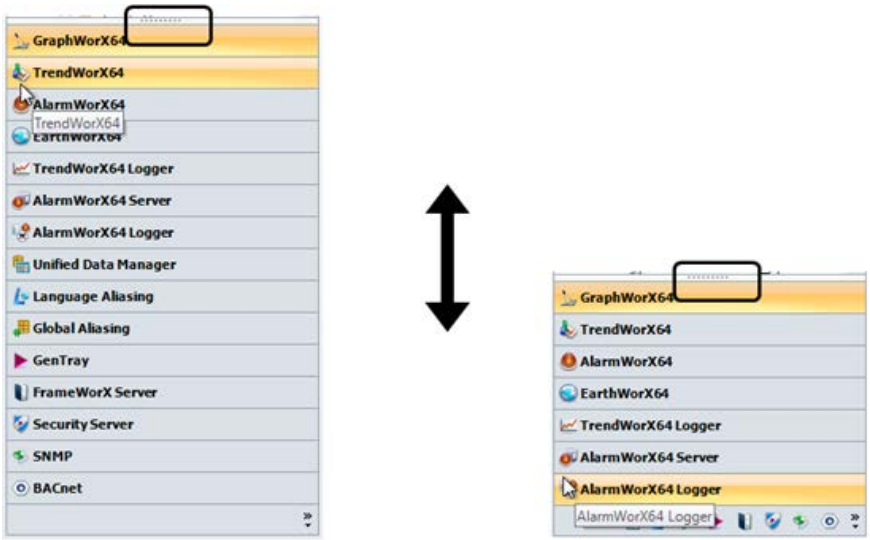


Figure 2.15 - Application Stack in the Project Explorer

2.3.4. Organizing the Application Buttons

If you use some buttons more than others, you can change their sequence in the stack by clicking the menu button in the bottom tray and selecting **Navigation Pane Options**. On the dialog box that appears, you can re-sequence the buttons and even remove them. Notice there are other options on the popup menu for organizing the buttons.

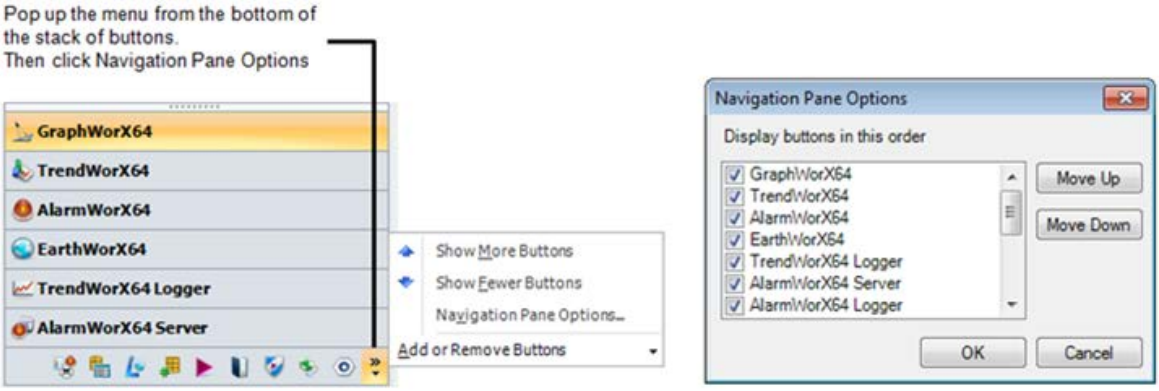


Figure 2.16 - Re-sequencing the Application Buttons

2.3.5. Removing Application Buttons

You can remove applications from the Project Explorer by clicking the popup menu in the icon tray and selecting "Add or Remove Buttons", and then disabling the checkboxes next to their names in the Button bar menu. The unchecked items disappear from the Project Explorer. You can always return an application's visibility as long as the application is in the project.

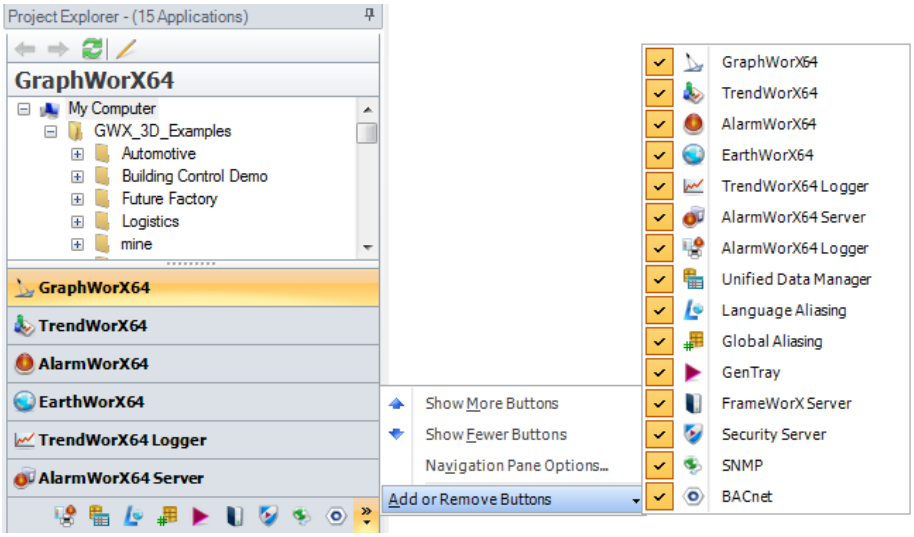


Figure 2.17 - Set Visibility on an Application

Selecting "Show More Buttons" from the popup menu puts all application icons in the icon tray. Selecting "Show Less Buttons" puts only the selected icons in the icon tray.

2.4. Configuration and Layout Management

Now that you know how to manage the applications in the project, let us look at how you use the Workbench to configure and lay out your projects. The Workbench is the tool you will work with most in GENESIS64. In it, you will:

- Configure applications where you choose options and save settings
- Design and test interactive or dynamic graphical interfaces

2.4.1. Working with Application Files

To create a new configuration or file, in the Project Explorer click on the application you want to work with. For example, if you are creating an alarm display, click on the AlarmWorX64 button. Expand the host and follow the application's instructions for creating a new configuration or display object.

If you find yourself unsure of what to do next, you can find instructions in GENESIS64 online help, which is available at any time by clicking on the help button to the right of the ribbon bar.

Applications can be grouped into two types:

- *Configuration forms* have prompts that help you to set up the application. The applications that have this type interface include Security Server, Global Aliasing, and Hyper Historian.
- *Runtime available* applications such as AlarmWorX64, TrendWorX64, EarthWorX, GraphWorX64, GridWorX64, and many others have a configuration mode and a runtime mode. You can set up the files in configuration mode. Once you have a file, you can *open it in configuration mode and runtime mode* depending on how you save and open the file.

2.4.2. Configuration Forms

With a few exceptions (such as the FrameWorX64 Server Location Setup dialog) nearly all GENESIS64 configurators and applications open in the Workbench. Configurators display forms in the Content area. Figure 2.18 shows a configuration form for the Security Server.

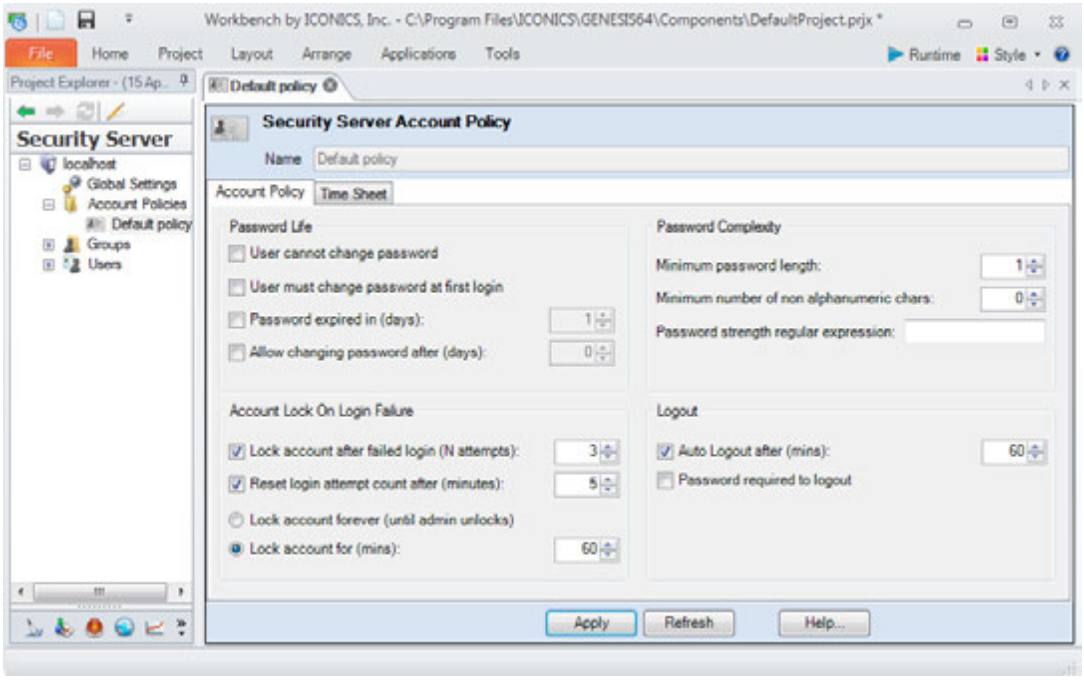


Figure 2.18 - Security Server Default Account Policy

2.4.2.1. Design Surfaces

When an application such as GraphWorX64, AlarmWorX64, TrendWorX64, GridWorX64, or EarthWorX64 loads in the Workbench, the Content area becomes a design surface upon which you can place different objects. Configuration of a display, trend, alarm, or map is done by configuring objects in the design surface in configuration mode. An example of a display with a design surface is the gauge shown in GraphWorX64 in Figure 2.19.

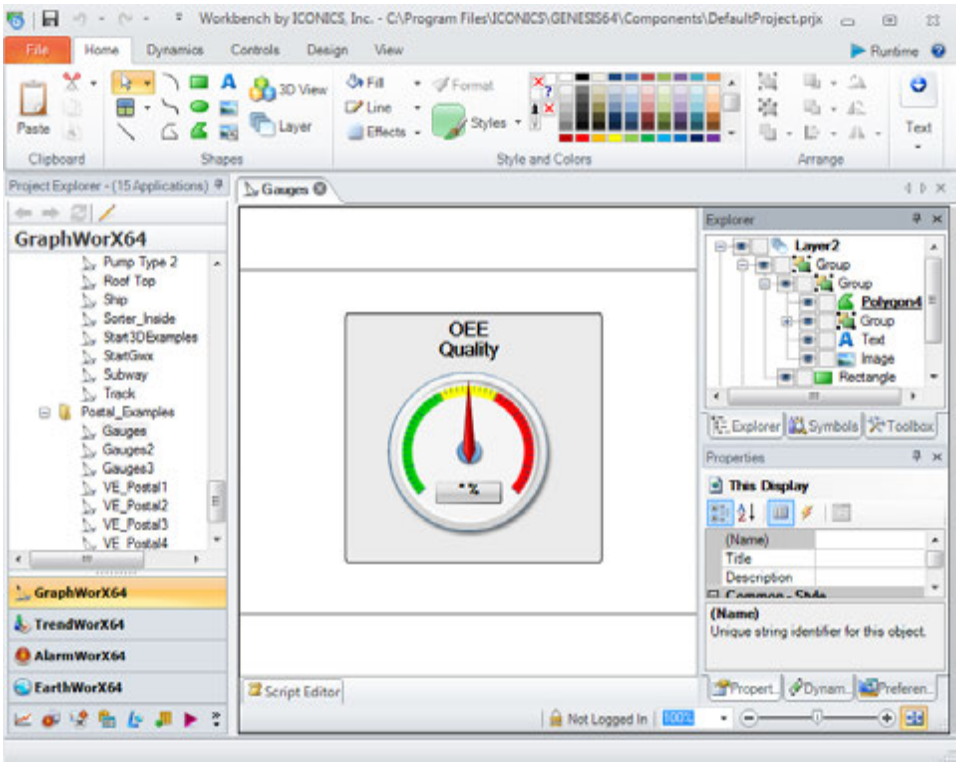


Figure 2.19 - GraphWorX64 Design Surface

Notice in Figure 2.19 that the object hierarchy is displayed in the Explorer navigation tree on the right, and the Properties pane exposes the object's Properties, Dynamics (Events and Methods), as well as the Preferences for the application and objects.

2.4.2.2. Runtime Mode

When the GraphWorX64 display is switched to runtime mode, all configuration information is removed. The same is true of any of the ICONICS viewers. The runtime version of Figure 2.19 is shown in Figure 2.20.

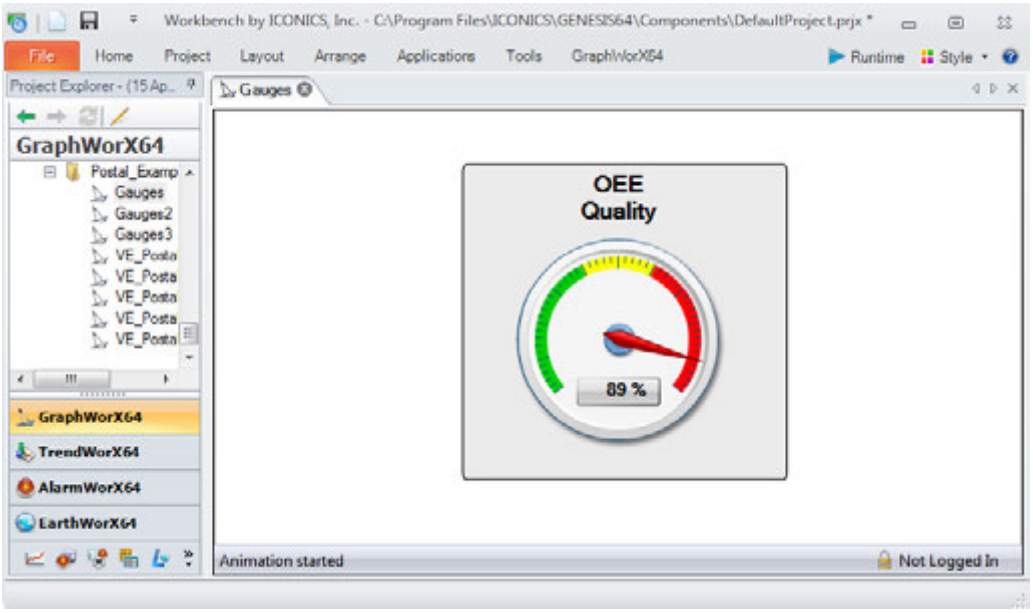


Figure 2.20 - GraphWorX64 Display in Runtime

We discussed the difference between runtime and configuration mode in the previous chapter, so here we'll focus on one major point: You can open a file in either configuration mode or runtime mode, but…

- A file opened in **runtime mode** cannot be toggled into configuration mode,
- While a file opened in **configuration mode** can be toggled between runtime and configuration mode without limit.

This is a design consideration for the Workbench to prevent users or customers from editing the display after deployment of the project. Since you cannot edit the display in runtime mode, it acts as a second security line to prevent those who lack permission from editing it.

2.4.3. Layout Management

It is important to realize that you are not restricted to displaying one form or one design surface at a time. The Workbench is designed to let you handle a myriad of files all at once. You can work with several files at the same time, arranging them in the content area in ways that you can see all files at once. You can even save an entire file layout for easier access when configuring on a later day. Instead of opening individual files and re-arranging them each morning when you begin, you can reopen a configured layout containing all of the files in the same arrangement in which you have previously saved.

2.4.3.1. Ways to Open a File

First, let us take a detour to learn how to open a file. You can open and manage all of the project's files from the Project Explorer in the Workbench. In here you can open a file in runtime mode, in configuration mode, and/or in a new tab. You can:

- Double-click a file to open it in runtime mode *in the current tab* of the content area
- Wheel-click (or middle-mouse click) on a file to open it in runtime mode *in a new tab* of the content area
- Right-click on a file to popup a menu of options on how to opening the file. Typically, popup menu options let you open the file in runtime mode in the same tab or a different tab, and edit the file in the same tab or a different tab.
- The Home ribbon provides these same options.

2.4.3.2. Arrange Files in the Content Area

Now that we know the different ways to open a file, let us get back into managing your content area. You can configure and run files for any GENESIS64 products in a complex layout that you create using the Workbench.

The default layout when you have multiple files open is the tabbed format, where each file has its own tab in the content area. You can rearrange the layout of the files on the Home tab of the ribbon. Use the buttons in the Arrange section to organize the different files in stacked  , tabbed  , side-by-side  , or table  format.

The tabular format has a drop-down box that, starting in the top-left corner, allows you to choose the number of cells you want the content area to divide into; the format chosen in Figure 2.21 would display two rows of three files.

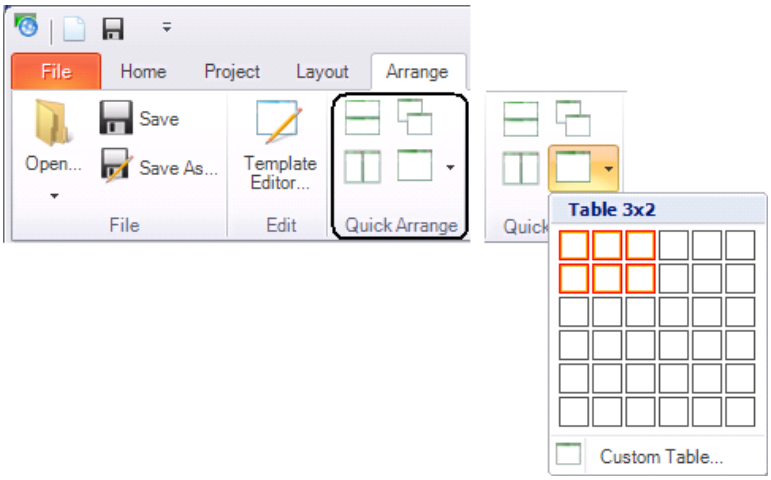


Figure 2.21 - Changing Content Area Layout

2.4.3.3. Maximize and Full Screen Options

Also on the Arrange tab of the ribbon are two options that let you expand your workspace area. You can click on the Maximize button to maximize the active document in the Content area. When the screen is maximized, clicking on (Close) returns you to the original screen.

You can click on the Full Screen button to expand the Workbench across all available screens; the Full Screen button is useful only if you are working with a configuration of multiple monitors. Clicking on (Close) returns you to normal.

2.4.3.4. Template Arrangements

We have researched and created a several popular templates for your use, saving you from the need to create your own arrangement. Figure 2.22 shows the Template Gallery on the Arrange ribbon, with some of its preconfigured template formats. You can select a template from the Template Gallery, or click the Open button to select a recently used one. If you already have files open when you apply a template, the files will be arranged automatically into a section of the chosen template.

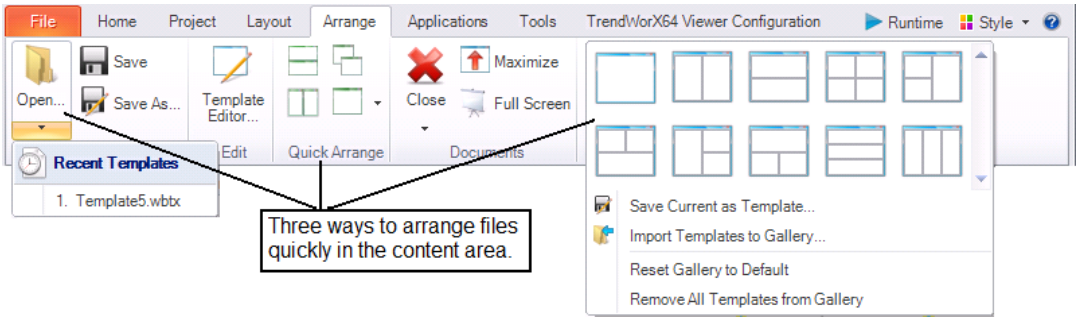


Figure 2.22 - Arrange Ribbon

2.4.3.5. The Template Editor

You are not limited to the predefined templates; you can save any Content area arrangement as a template by clicking the "Save" button on the Arrange ribbon. You can also design your own templates by using the Template Editor, shown in Figure 2.23. You can open the editor by clicking the "Template Editor" button on the Arrange ribbon.

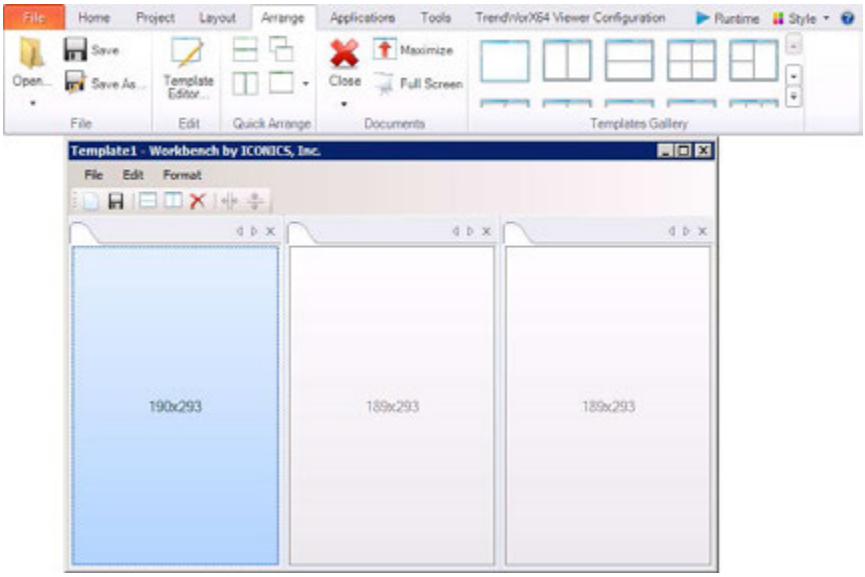


Figure 2.23 - Template Editor

To use the Template Editor, you can open an existing template or start from scratch. Select a tab area in the window, and then click a button to split the tab horizontally or vertically. You can also change the size of any tab area by grabbing the edge (the cursor will change to or) with the mouse, dragging the cursor, and then releasing the button when the tab is at the desired size. Save the template when you are done.

> ***Tip!*** If you are working with a configuration of multiple monitors, create templates that align to your monitor's configuration when you work in Full Screen mode.

2.4.3.6. Layouts

Once you have opened a template in the content area of the Workbench, any file you open opens in the currently-selected section in the content area. Therefore, before you open a file, click in the section you want to open file in first, and then open the file. You will notice that each section's file opens in a tab. You can, of course, rearrange the files in stacked or side-by-side format within the content area.

Once you have opened files in the sections you want, you have a *layout*. Whereas a *template* is a simple arrangement of sections into which you can insert files, a *layout* is an arrangement of individual files inside a template.

Figure 2.24 shows a sample layout with three sections. Note that the top-left section has two alarm displays; the top-right has an alarm display and a trend display; and the bottom contains only one trend display.

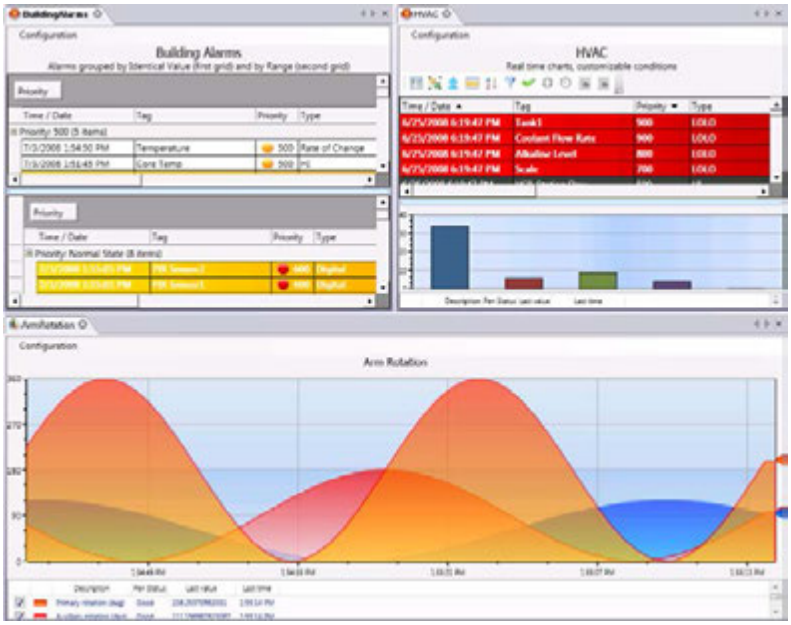


Figure 2.24 - Sample Layout

2.4.3.7. Saving Layouts

When you are working in a layout, you can preserve it for later use. Opening a layout opens the same files in the same arrangement, saving you development time in having to reconstruct the layout from scratch every time you want to work with it.

On the Layout tab's **File** section, you'll find buttons to create a new layout, open an existing layout, and save a layout. The ribbon also has a button for a carousel of layouts, and a gallery of layouts.

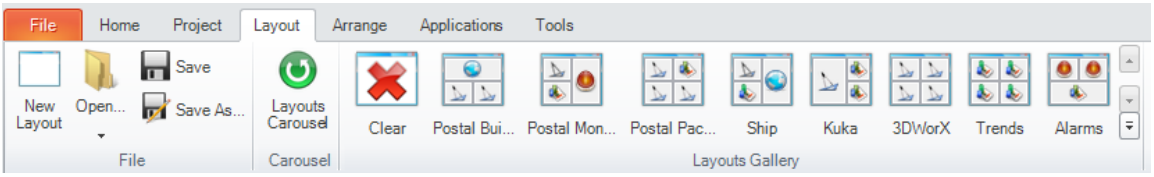


Figure 2.25 - Layout Tab of the Ribbon

Saved layouts do not automatically appear in the Layout Gallery, which we will discuss next. If you want to add the new layout to the Gallery, you can either import the layout after saving it, or simply use the "Save Current Layout To Gallery" option in the Layouts Gallery section.

2.4.3.8. Layouts Gallery

In the Layouts Gallery, you can see the many sample layouts provided in GENESIS64. If you hold your cursor over a sample in the Layouts Gallery, you'll get a preview, like the one shown Figure 2.26. Clicking a layout on the Layouts tab in the ribbon opens the layout in the content area.

You can look at these samples and use them as a starting point for your own layouts; or you can remove the samples and create your own. Note that you can import layouts from other projects, too. The layouts are local; they are not saved with the project and are not available with the project when you package it and deploy it.

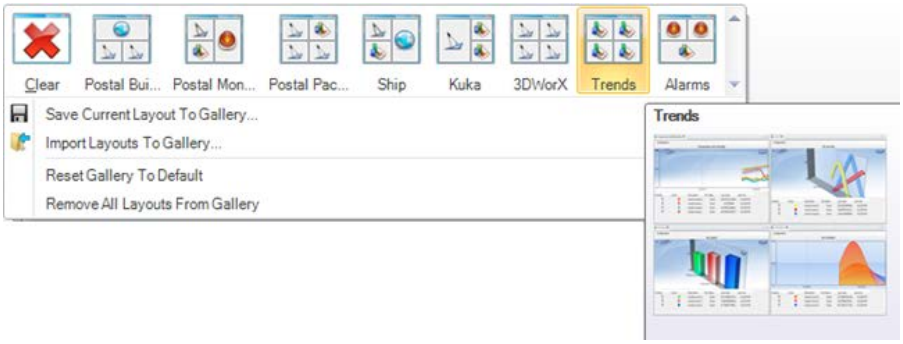


Figure 2.26 - Layout Gallery

> ***Tip!*** If you save a layout file on your Windows desktop, double-clicking it launches the Workbench and loads the layout.

2.4.3.9. Layouts Carousel

One additional feature is the Layouts Carousel, primarily used as a play-list in Runtime mode. In the carousel, you can queue files to "play" in a slideshow format during runtime.

For example, you have two displays of one facility – one display monitors pipes and the other monitors circuitry – and you want the runtime display to toggle between the two. You can use carousel to do this.

You can also use the carousel for your own convenience during configuration. In the carousel, you can list any files you want available from a particular tab of the layout; those files are listed in the carousel for easy access.

To start the carousel, make sure you've selected the tab where you want the carousel to be used, then click on the Layouts Carousel button on the Layout ribbon. The Carousel dialog box appears and looks something like Figure 2.27.

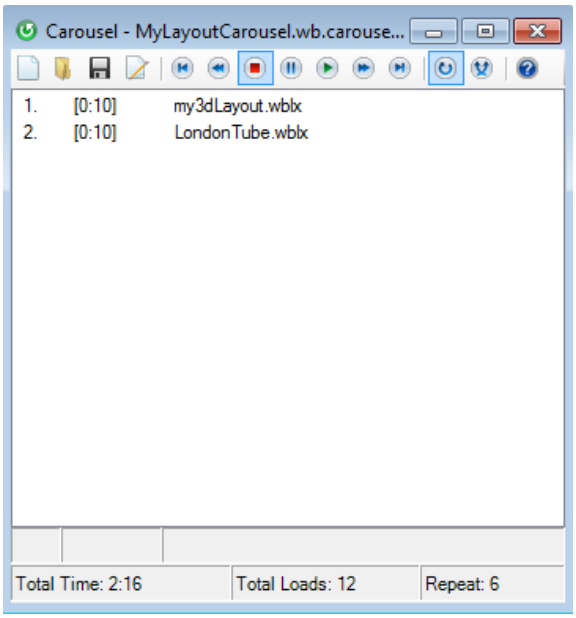


Figure 2.27 - Layout Carousel Playlist

In this dialog box, you can create a playlist consisting of Workbench layout files (.wblx) by clicking the button. The playlist is saved with the extension .wb.carousel.xml.

You can edit the list by clicking the button to open Edit Playlist dialog box, shown in Figure 2.28. Here you can add and remove layouts. You can also put the layout files into the sequence in which you want them played.

As you add files to a playlist using the Edit Playlist dialog, you can click on the Options button to display the Playlist Options dialog, also shown in Figure 2.28, where you can enter specifications for playing the list.

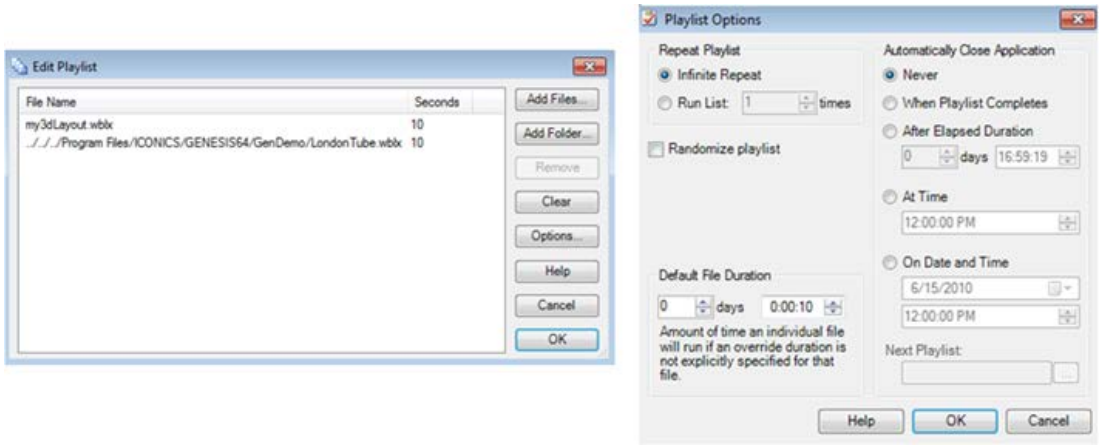


Figure 2.28 - Playlist Editor (left) and Playlist Options (right)

See how you can limit the number of times a playlist repeats; play files at random; limit how long individual files can run; automate the time at which the playlist stops; and queue up the next playlist to play within the options.

2.5. Project Management Tools

GENESIS64 saves all of your tools, applications, and settings to the project file so that you can reopen and reuse the project. In each project, the following are stored:

- Applications
- Database configurations such as the Global Aliasing, Language Aliasing, and other SQL databases
- Network services tools such as GenTray, SNMP, the FrameWorX Server, and the Security Server (see “Working with Services” on the next page)
- Workbench configuration settings that you apply to the Workbench window position, styles, modifications made to the design surface, layouts, and templates

A project is normally stored on the system where the servers are running. In actuality, there may be several servers that need to be accessed in order to run the different displays that an enterprise might require. As long as you have defined their connection details, those connection details are saved, too.

The Workbench is designed for concurrent configuration by multiple development team members over the internet.

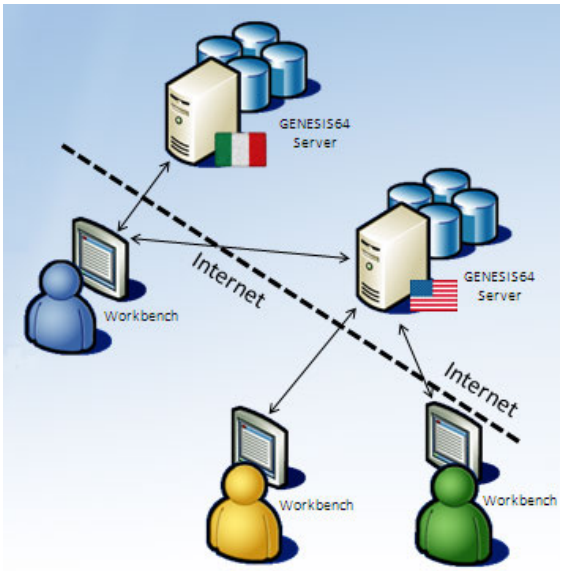


Figure 2.29 - Concurrent Web Development by Multiple Team Members

It follows the *optimistic concurrency model*; there are no database locks. Multiple application servers and databases can be configured simultaneously. An additional ICONICS Workbench Toolkit can be used to create plug-ins from third-party and OEM content providers.

Centralized security allows all GENESIS64 operations to be secured, which can be adopted by OEM customers. Security Server limits access permissions to projects and plug-ins, including file access to create, modify, or read only. Historical data are secured with an audit trail. It also allows you to secure OEM operations.

There are two security models: ICONICS Security Server database and Windows Active Directory. We will discuss security in more details in a later chapter.

2.5.1. Working with Services

A number of GENESIS64 components are servers or services that you can control and modify within the Workbench. They include:

- **GenTray** starts and stops any service, including ICONICS and Windows services.
- **FrameWorX64 Server** allows configuring GenBroker, GenEvents, and web services. You can also configure the databases that support these different methods of communications.
- **AlarmWorX64 Server** starts and stops the alarm server, as well as provides a configuration interface to configure alarms.
- **TrendWorX64 Server** starts and stops the historical trend server, as well as provides a configuration interface for adding historical trend tags.
- **SNMP** configures and manages the SNMP (Simple Network Management Protocol) service that is used to discover SNMP devices and their properties over a network.
- **BACnet** communications can be integrated into GENESIS64 for use with HVAC controls, fire detection and alarm, lighting control, security, "smart" elevators, and utility company interfaces.
- **Unified Data Manager (UDM)** is a central location for storing expressions, subscriptions, groups, values, registers, triggers, and other data that can be used by any GENESIS64 application. You can start and stop the service and create and modify all of these data types and data instances from within the UDM in the Workbench.
- **Security Server** stores accounts and policies in a database organized by users and groups. You can start, stop, modify, and create security policies from within the Security Server.
- **Global Aliasing System (GAS)** stores variable names and their values within an SQL database so that these variables can be used throughout GENESIS64 applications. Start, stop, create, and modify global aliases using this interface.
- **Language Aliasing System (LAS)** makes substitutions for language aliases whenever the assigned language in the Language Selector dialog box is changed.
- **ScheduleWorX64** starts and stops the ScheduleWorX64 server and run your created configuration containing holidays or schedules, that can be triggered different ways.

2.5.2. Edit a Project Remotely

The Workbench service management reports back to the remote client the health of the service. It gives you fast access to all running and installed Windows services and you can inspect their properties, start them, and stop them.

To open and review a remote system, click on the **Connect to Server** button in the Project tab of the ribbon and set up access to the remote system; it then becomes available from the Workbench.

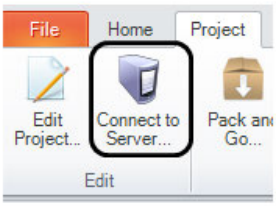


Figure 2.30 - Connecting to a Remote Server

Allowing team members to connect to your server remotely using the Connect to Server button on the Project ribbon lets them open the project and edit it as necessary. Alternatively, you can copy the project using the tools described in the "Deployment" section.

The Connect to Server button is handy with deployed projects as well. If you are supporting a customer project remotely, and the customer gives you access to its server, you can connect and edit the customer's deployed project without having to make an onsite visit.

2.6. Deployment

With a large project, there are typically several configuration files, databases, and resources. Configurations and files may even be spread across several machines among the team members working on a project. Sometimes the real challenge is not the configuration itself, but how to copy all of those resources already configured on a target machine to share them with a team member on a separate network. An even bigger challenge can be deploying the project at the customer's site when you are done.

The Workbench simplifies the task of packing and deploying configuration databases and runtime-related files so that developers do not have to create custom tools for deploying existing configurations and resources files.

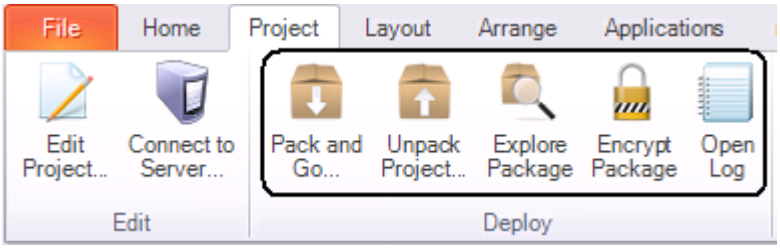


Figure 2.31 - Deploy Section of the Project Ribbon

Use the built-in capabilities on the Workbench's Project tab of the ribbon:

- The **Pack and Go** wizard copies your project's resources into a file, making the project portable. The wizard gives you the options to compress the project, encrypt the project, and open the log file when the wizard is done.
- The **Open Log** tool lets you see the status of a project you just packed, along with project statistics. You can open the log from the Pack and Go wizard or at any time by clicking the Open Log button on the Project ribbon.
- The **Encrypt Package** tool encrypts a packed project; an encrypted project requires a password to unpack it. The Pack and Go wizard prompts you to encrypt the project, or you can encrypt a packed project at any time using the Encrypt Package button on the Project ribbon.
- The **Explore Package** tool lets you look at the content of a packed project.
- The **Unpack Project** wizard extracts the project onto a remote target machine, whether you are simply making the project available to development team members or deploying it for users. If the project is encrypted, the password entered when it was encrypted will be required in order to unpack the project.

2.7. Workbench Runtime

When you configure your Workbench project you can always go into Runtime mode and see how it looks. To switch into Runtime mode you need to press the Runtime button in the upper right-hand corner of the window.



Figure 2.32 – Runtime Button

In Runtime mode you have a Ribbon menu similar to configuration mode just with runtime functions. For viewers like GraphWorX, TrendWorX, AlarmWorX and EarthWorX you have the possibility of browsing folders you added to your current configuration.

In the Workbench Runtime it is possible to use the Search Tool which can search your entire project and include sortable results for the server side searches. This search result dialog allows you to sort items found.

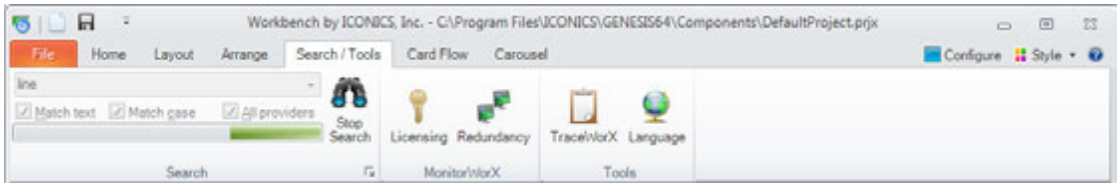


Figure 2.33 – Workbench Search

The other useful thing you can use in the Workbench Runtime is the Card Flow view (shown below) used in runtime to preview other displays in the file history before opening one. This is a handy alternative to the Back and Forward buttons, which provide lists of files in file history but no ability to preview the files you're about to open.

In Workbench Runtime you can preview configurations and load examples. You can access the Card Flow view from the Runtime ribbon's Back Forward button and Recent button.

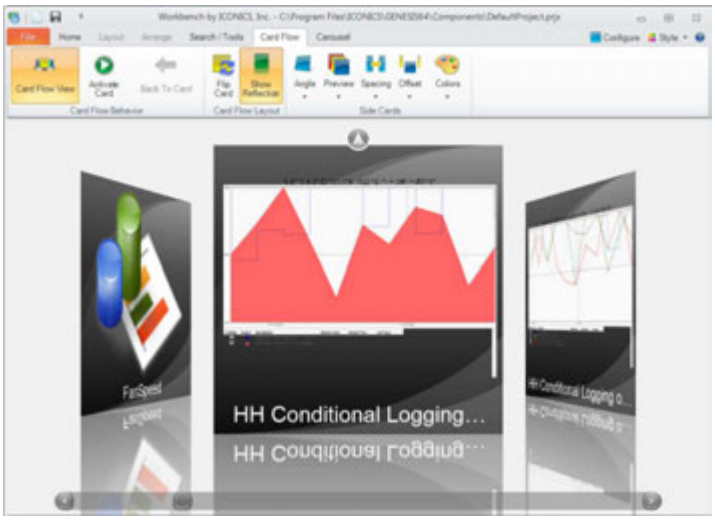


Figure 2.34 – Workbench Card Flow View

Note: You can encrypt or view contents after creation.

Lab: Data Connectivity L2

2.1. Lab Overview

In this section, you will explore GenDemo and become familiar with it so that you can use it effectively going forward. We will also give you a chance to practice inserting OPC DA tags into GraphWorX64 and view live data.

Estimated time to complete this lab: 20 minutes

2.1.1. Task Checklist

In this lab, you will complete the following tasks:

- Explore GenDemo
- Connect your PLC to an OPC server
- Connect to OPC tags
- View and manipulate OPC tags within GraphWorX64

2.1.2. Support Files

You do not need any additional supporting files to complete this lab.

2.2. Exploring GenDemo

There are many helpful displays within GenDemo that can serve as examples when needed. In this section, we will explore a few of these displays.

1. Open GenDemo by going to Start → All Programs → ICONICS → Tools → Edit GenDemo. Opening this version of GenDemo allows us to go into GraphWorX64 configuration mode.

2. Once the display opens, hover over the objects on the right and click on the **Navigation Menu** button. Select the **GraphWorx64** selection. To pause, click on the arrow in the circle in the upper left. Select **Architecture** from the list. This display shows an overall view of how different components, such as your OPC server, Databases, and different GENESIS64 components, all work together.

3. Hover your mouse over the icons on the right and select the **Tutorials** button. We've included the Tutorials in GenDemo to demonstrate the different tools available in configuration mode. Click on a selection from the list on the left and then click on the items inside the display to see how they react.

2.3. Workbench Navigation

Let's take a look at some of the Workbench features.

1. Since the Workbench acts as a centralized location to access most of the applications, we will use it throughout this course to manage our training course project. Open the Workbench from Start → All Programs → ICONICS → Workbench.



Figure L2.1 - The Workbench Ribbon Bar

2. Figure L2.1 above shows the Workbench ribbon. As described in Figure 2.3 on page 2-3 in the Chapter 2 lecture, there are three sections in the Workbench. The Ribbon Bar (shown above), the Project Explorer (which lists the providers) and the Content Area.

2.4. Workbench Project Folder

The GENESIS64 Providers are exposed in the Project Explorer located on the left side of the Workbench. The list of providers can be expanded or retracted by selecting the bar above the GraphWorX64 provider.

1. To add the C:\Training folder to the list of GraphWorX64 folders. Click on the GraphWorX64 provider. Expand the GraphWorX64 navigation tree by clicking on the plus sign to the left of "My Computer". Right click on "My Computer" and select "Add/Remove folders" in the drop-down list.

2. From the Folder Collection Editor dialog box, click on the "Add Folder" button. Browse to and select the C:\Training folder to add this folder to the Folder Collection list. Click on "OK".

2.5. Workbench Providers

1. The project folder is the default location where all the project files will reside after being saved. Create a folder called **C:\Training**. This is the location where we will store our training files.

2. As seen in the figure above, the File selection is above the Runtime icon in the Ribbon Bar. Click on File → Save As. If this is the first time after the GENESIS64 installation, the default path and project name will be similar to "C:\ProgramData\ICONICS\DefaultProject.prjx".

3. Browse to the C:\Training folder and change the File Name to icTonicsProject.

2.6. Data Connectivity

We will now try to connect to a simulation tag in GraphWorX64 so you can view and control it within the application.

1. Since the Workbench acts as a centralized location to access most of the applications, we will use it throughout this course to manage our training course project. Open the Workbench from **Start → All Programs → ICONICS → Workbench.**

2. GraphWorX64 opens by default in the Workbench. (If it doesn't, go to the Applications ribbon and click the GraphWorX64 button.) In the Home ribbon on top, click the **New Display** button.

3. Add a process point by going to the **Dynamics** ribbon, and click the **Process Point** button. Then use your mouse to draw a square for the button onto the display canvas.

4. In the **Data Browser** that comes up, click on the **Simulation** tab, and select the **localsim:ramp** tag. Notice the tag properties on the right-hand pane, and then click "**OK**".

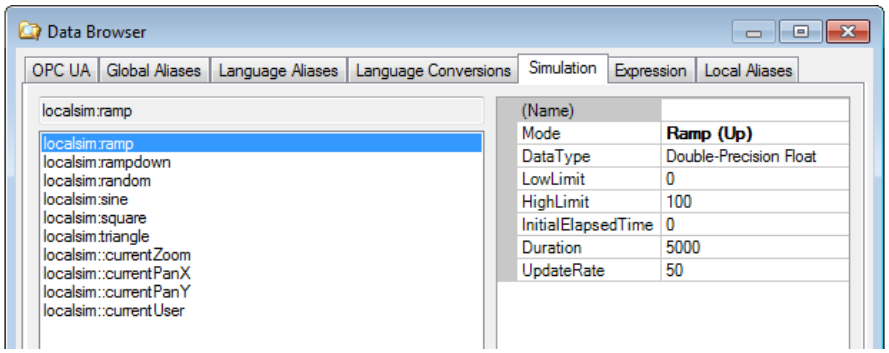


Figure L2.2 - Data Browser

5. Click on the "**Runtime**" button in the upper-right corner to go into runtime and watch the data change.

6. Select the **GraphWorX64** tab and then the **Configure** button to go back into configuration mode. Select the process point on the display canvas. In the Dynamics explorer on the right-hand side, notice that the Data Source property says only **localsim:ramp**. This is because we are using a GraphWorX64 local simulation tag. You will not be able to use this tag outside of GraphWorX64.

7. Now we need to set some options to be able to see other tags. Got to **Start → All Programs →ICONICS → ICONICS OPC Server Suite 5 → ICONICS OPC Server 5 Administration**. This launches the "**ICONICS OPC Server Suite – Runtime**" dialog.

8. **Right-click on the ICONICS OPC Server Suite 5.5 icon** (located in the lower right hand section of the Windows task bar (**near the clock**) as seen in the figure below and select **Settings**. This launches the "**ICONICS OPC Server Suite Settings**" dialog box.

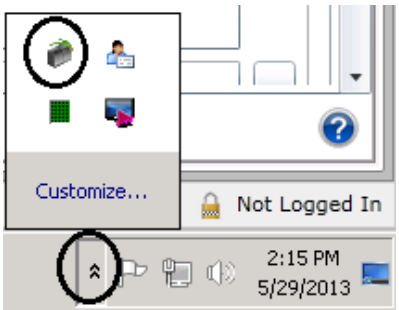


Figure L2.3 – ICONICS OPC Server Suite 5.5 Icon

9. Click on **Runtime Process** tab and from the drop down menu select **Interactive**. Click on **Configuration** tab and check the box **"Allow runtime to accept remote connections**". Click **OK** to save changes. You may receive notification that this operation may cause clients to be disconnected. Click **Yes**.

10. From the "**ICONICS OPC Server Suite – Runtime**" dialog, you opened previously, reinitialize the server by clicking on menu **Runtime → Reinitialize**.

11. Return to GraphWorX64 add another process point to your display. Then in the **Data Browser** select the **OPC UA** tab, then select **OPC Classic Servers → My Server → ICONICS.IconicsOPCUAServer.V5 → Channel_0_User_Defined → Ramp → Ramp1**, and click on "**OK**". Now notice the syntax for the data source property looks like this:

    @ICONICS.IconicsOPCUAServer.V5\Channel_0_User_Defined.Ramp.Ramp1.Value

12. The prefix **@** and suffix ***.Value*** indicates that **this is a tag from an OPC DA server and not an OPC UA server**.

13. Add one final **Process Point** and click **Cancel** to close the **Data Browser**.

14. This time, we will type in a portion of the path and then browse. In the **Dynamics** Explorer on the right, find the **DataSource** field and type in:

    opc.tcp://ComputerName:Port

Notes:

- Replace ***ComputerName*** with the name of your computer or IP address.
- ***Port* 49460**. The port number 49460 is fine for a training computer however it's worthwhile noting, the OPC UA server port number is defined through the OPC UA Configurator you launched in lab one.

15. Click on the **Ellipse** button, at the end of the DataSource field, to launch the **Data Browser** and locate the tag:

    Server.:ServerStatus.:BuildInfo.:ManufacturerName

16. Click **OK** and go **Runtime** mode to see that all of your data shows up with some value. The UA field should show the word **ICONICS**.

Review: Workbench R2

2.1. Objectives Review

You should now know and understand the following topics:

- The basic features of the GENESIS64 Workbench
- How to customize the Workbench
- How to create an ICONICS project using the Workbench
- Techniques for arranging and working with files in the Workbench
- Ways to share a project among development team members
- Deploying a project for a customer

2.2. Summary

In this chapter, we learned that the GENESIS64 Workbench is a centralized Web-based work environment for GENESIS64 products. It provides a configuration environment that runs multiple applications and customized third-party environments. In addition to being a configuration tool, the Workbench can act as an operator interface and project management tool.

We looked at three distinct parts of the Workbench: the Project Explorer, the ribbon bar, and the content area and reviewed ways to customize each. In creating a project, we start by adding ICONICS applications to the project. There are several ways to manage applications in the project. We looked at sample configuration forms and display files in the Workbench in configuration mode and in runtime mode, and then reviewed the ways in which you can work with those files in the Workbench: by arranging files, saving an arrangement as a template, and saving an arrangement of files as a layout.

We then looked at overall project management of services and working remotely, and then reviewed ways to pack a project for sharing with remote team members or for deploying at a customer site.

2.3. Terms

Content Area	Configuration Mode	Home Page
Pack and Go Wizard	Project	Project Explorer
Ribbon	Runtime Mode	Workbench
Window Layout	Workbench Template	

2.4. Test Your Knowledge

In this chapter, we discussed the GENESIS64 Workbench. Before continuing to the next chapter, make sure you can answer the questions below:

1. True or False: Workbench can run inside Internet Explorer.

2. True or False: You can use Workbench to edit multiple servers at the same time.

3. Is Workbench a runtime or configuration environment?

   a. Configuration Environment
   b. Runtime Environment
   c. Both

4. Can you plug-in third party applications?

   a. No
   b. Yes, all .net controls, WPF and Active-X applications
   c. Only *.EXE applications

5. Can GraphWorX64 run inside Workbench?

   a. Yes
   b. No
   c. Only through an import filter

6. True or False: Workbench is a .NET application?

7. True or False: You can start and stop only local GENESIS64 services from the Workbench.

8. True or False: You can secure the Workbench.

Getting to Know GraphWorX64 3

GraphWorX64 is the ICONICS application you use to create dashboards and interfaces for your runtime operators. GraphWorX64, a human-machine interface (HMI) software package for process control, creates astonishing interactive, animated graphics. GraphWorX64's design surface is a canvas on which you can display meaningful real-time or historical data from any OPC-UA compliant data source. This allows an operator to easily recognize the state of your equipment, processes, conditions, and anything else you want to quantify in the display.

You can open GraphWorX64 as a standalone application or open it in the Workbench. Regardless where you open the application, GraphWorX64 comes with a ribbon that is similar to the one in the Workbench. When you open the application in the Workbench, the GraphWorX64 ribbon replaces the Workbench one.

In the upcoming sections and chapters, we will discuss options in the GraphWorX64 ribbon, as well as other dialogs and toolbars within the application.

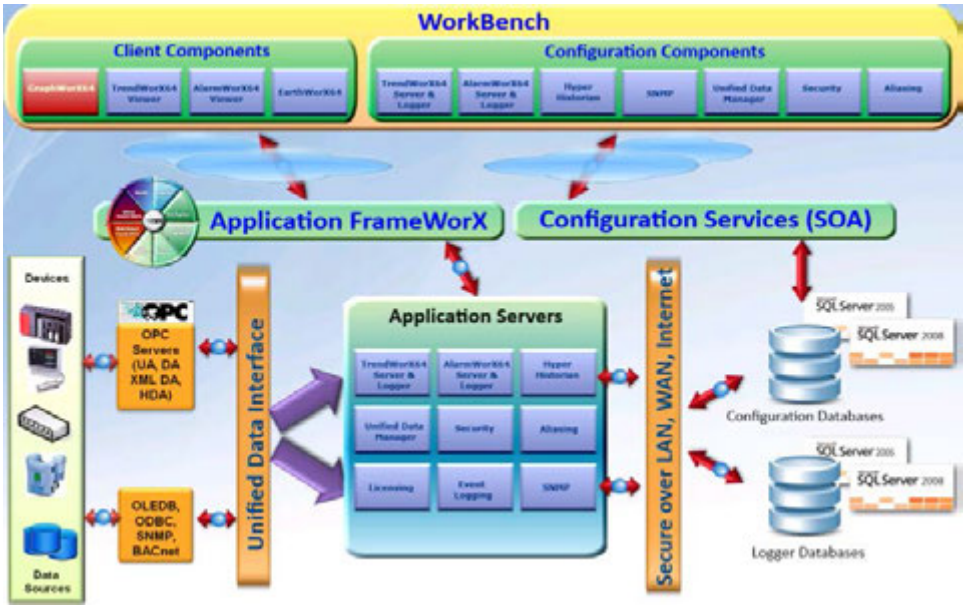


Figure 3.1 - GraphWorX64 Architecture

After completing this chapter, you should know how to:

- Work with the GraphWorX64 Object Explorer
- Work with the GraphWorX64 Properties and Dynamics Explorers
- Set up preferences in the Preferences explorer
- Customize your workspace
- Use pre-made images and save new ones to the Symbols Library
- Use templates and add transitions for opening and closing displays
- Preview a runtime version of a display

3.1. GraphWorX64 Layout

Once you open the GraphWorX64 application in the Workbench, you will see a screen similar to Figure 3.2. On the left is the Workbench's now-familiar Project Explorer. Let us take a moment to review the GraphWorX64 areas of what you see here.

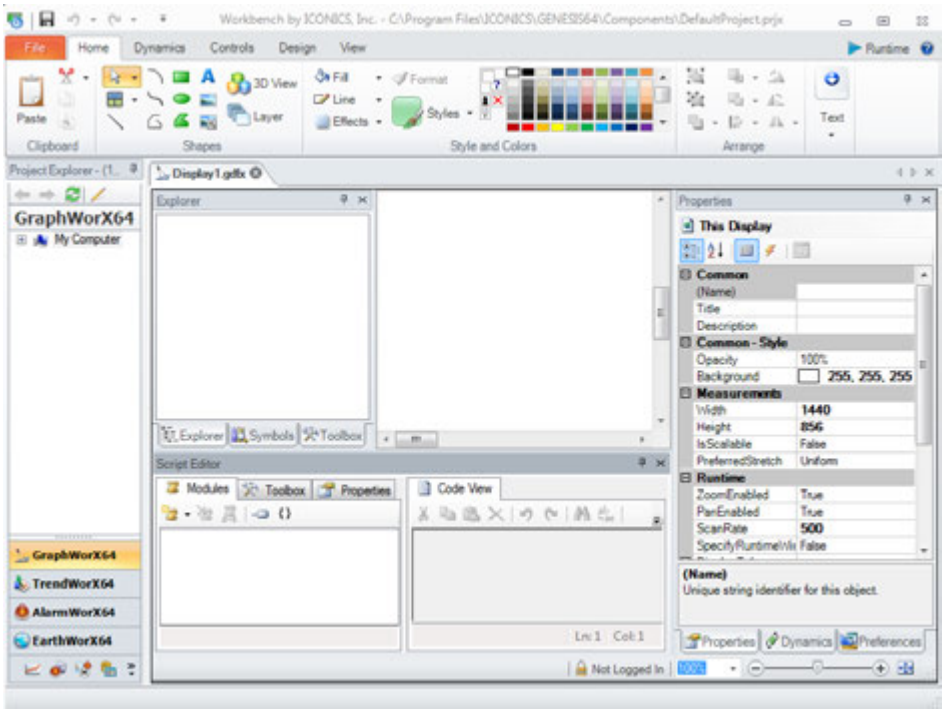


Figure 3.2 - GraphWorX64 Application Interface

Figure 3.2 divides the GraphWorX64 interface into five different areas. At the top, there is the ribbon. In the middle is the content area in which you can create your graphical interface by adding shapes, process points, controls, etc. Surrounding the contents area are places where you can dock explorers. The explorers include the Object Explorer as well as the Dynamics, Properties, Preferences, Toolbox, and Symbols explorers.

At the bottom is the Script Editor, which has its own additional components. Once you dock these explorers, you have the option to hide them when they are not in use.

We will discuss these explorers throughout the GraphWorX64 chapters. Let us first look at the Object Explorer, which lists all of the objects in the display.

3.2. The Object Explorer

In GraphWorX64, an object can be a shape, a layer, a group, a panel, or some other entity. GraphWorX64 uses a hierarchical object scheme for object creation and inheritance. When you create a new display, you begin with the *This Display* object, in which all other GraphWorX64 objects are placed.

The Object Explorer, also referred to as *the Explorer*, is a tree control that lists all of the objects currently defined on the GraphWorX64 display. This includes all of the objects and groups that are visible and editable, as well as those that are defined but hidden.

> **NOTE**: If you do not see some of the Explorer, you can bring it back into view by enabling the Explorer checkbox in the Show/Hide section of the ribbon's View tab.

The Explorer, shown in Figure 3.3, provides a hierarchical tree-like overview of the objects that are currently on the display surface, removing your need to drill down through several levels to get an object within a grouped object. You can also reorganize objects by dragging and dropping them inside the Explorer tree.

The Explorer can tell you:

- The order (Z-order) of objects in the display, which is their sequence from front to back.
- Whether an object is selected or active. Selected objects are highlighted in blue, while active objects are highlighted in gray.
- If an object has a dynamic action associated with it. Dynamic actions are underlined.
- Whether a user-assigned name has been given to the object. Generic, default names appear in normal text, while user-assigned names appear in **bold text**.
- An object's type, using icons for text , rectangles , panels , etc.

- Whether an object is visible in the display, and whether it is locked from being edited .

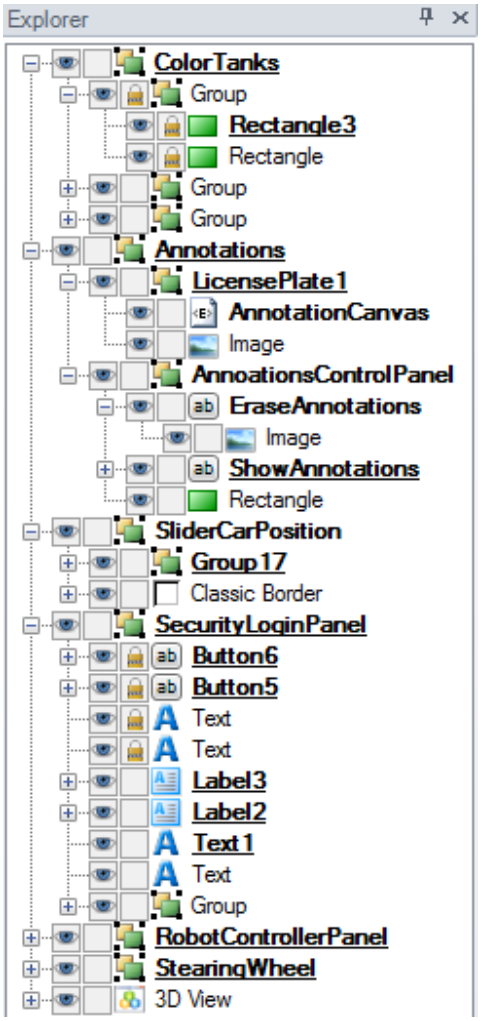


Figure 3.3 - Sample Explorer Tree

3.2.1. Selecting Objects

The objects shown in the Explorer are synchronized with the objects that you see in the content area of the display. As you select an object in either location, the object is also selected in the other.

An object's icon in the Object Explorer indicates what type of object it is. You can change the size of the icons in the tree by clicking on the Application Style button in the Misc section on the View tab and selecting Icon Sizes, as shown in Figure 3.4. Select the size that suits you.

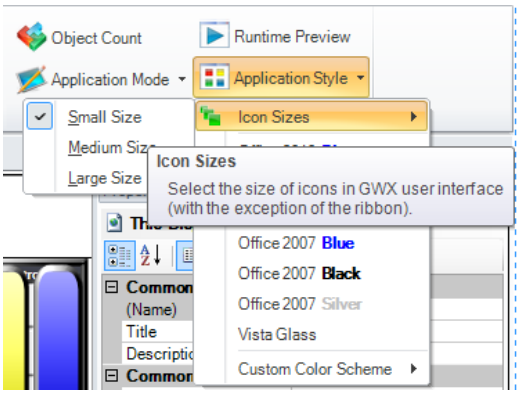


Figure 3.4 - Changing Explorer Icon Size

3.2.2. Renaming Objects

Each object added to the display content area appears in the Explorer with a generic name that GraphWorX64 provides by default. For example, a rectangle is given the name Rectangle. You can change the name of each individual object to something you can easily identify in the Explorer tree. For example, after adding three radio buttons, you can change their names from *Radio button* to the names of their states: *High*, *Medium*, and *Low*. The default name appears in the Explorer in normal text, but a user-assigned name appears in **bolded text**. To rename an object in the display, just click on the object name in Explorer, then type the new name.

3.2.3. Manipulating Objects

You can move an object up or down in the hierarchy, changing its front-to-back assignment. Moving an object to another branch in the tree changes its location in the display and makes it part of another object or group.

The eye icon indicates that the object is visible in the display. Clicking the object's Show/Hide property makes it visible or invisible.

The Explorer's lock icon indicates that the object is non-editable or locked. Clicking an object's Lock/Unlock property locks or unlocks it, making it non-editable (locked) or editable (unlocked).

3.3. The Properties Explorer

You can define the properties for the current display by selecting the Properties tab. If you see the properties of an object, then maybe the Properties tab is selected. Click anywhere on the background of your display to view the This Display object's properties, as shown in Figure 3.5.

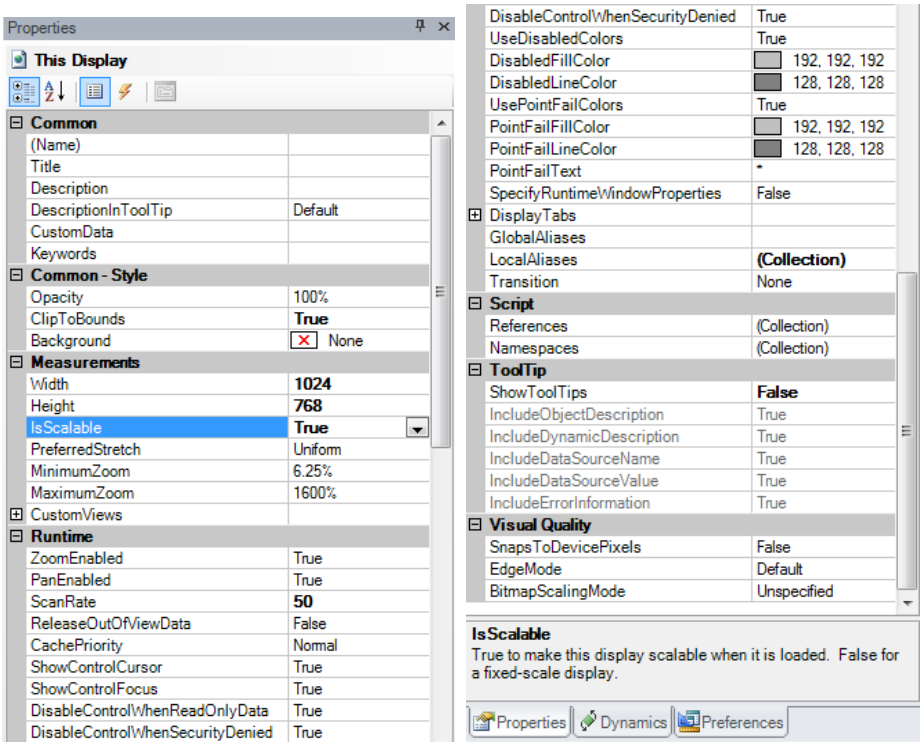


Figure 3.5 - This Display Properties

In the Properties Explorer, you can configure some options for the display. We will not review all of the available properties because there are too many, but we will cover some of the more useful or commonly-used properties.

To begin with, in the Measurements section, you can configure the size of the display. Figure 3.6 shows the measurement of the display to be 1024 x 768. You can change your display to be bigger or smaller depending on your preference. You can also set the display to be scalable or not by turning setting the IsScalable field to true or false.

Measurements	
Width	**1024**
Height	**768**
IsScalable	**True**
PreferredStretch	Uniform
MinimumZoom	6.25%
MaximumZoom	1600%

Figure 3.6 - This Display Properties: Measurements

Another important section is in the Runtime section. You can set the default scan rate for any process points in the ScanRate field. The default is fifty milliseconds. You may want to slow this rate to reduce the load on your OPC server.

3.4. Customizing the Workspace Using the View Ribbon

The View ribbon, shown in Figure 3.7, is where you customize the layout of the GraphWorX64 workspace, allowing you to tailor it to your own work habits.



Figure 3.7 - View Tab of the Ribbon

3.4.1. The Grid Section

Using the **Grid/Ruler** section of the View ribbon, shown in Figure 3.8, you can put a grid into the content area of your display; the grid can guide you in the placement of objects. You can use the X and Y fields, also shown in Figure 3.8, to change the distance between grid points, and the Snap to Grid feature makes sure objects you place on the surface align directly (or snap) to the grid points. Showing the ruler gives you guidelines so that you see the actual placement on the ruler; the ruler measures grid points; the X and Y values of the grid determine the granularity on the ruler.

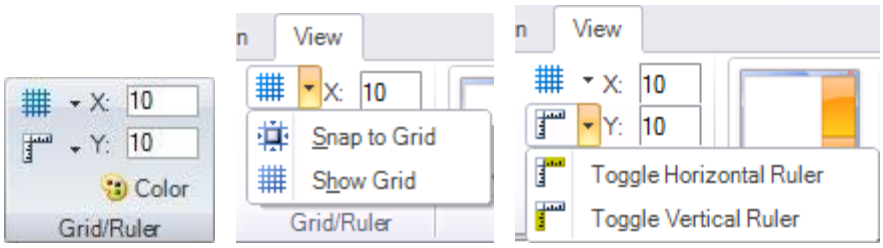


Figure 3.8 – The Grid/Ruler Section (left) with Grid (center) and Ruler (right) Options

3.4.2. The Window Layout Section

In the **Window Layout** section of the View tab, shown in Figure 3.9, are several layouts you can use for organizing your GraphWorX64 workspace. A layout consists of a pre-determined arrangement of the workspace. It hides, shows, and arranges the workspace's panes in a prescribed way. For example, you can use a drawing layout, a scripting layout, or a symbols layout, to name a few.

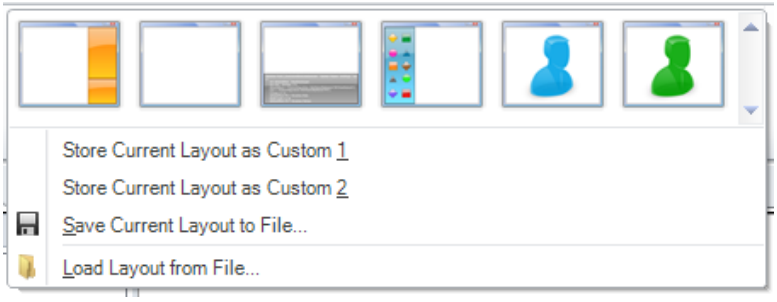


Figure 3.9 - Layouts Section

You do not have to use a layout, but if there is a workspace layout that you prefer, you can save it and select it any time you wish to use it. You can store any number of new layouts and choose the layout that suits you at any time while you are working.

3.4.3. The Show/Hide Section

Use the **Show/Hide** section, shown in Figure 3.10, to control which explorers appear on the GraphWorX64 workspace. You can show and hide the Properties, Dynamics, Preferences, (Object) Explorer, Toolbox, Symbols Library, Scripts, Scrollbars, and Status Bar. You can move the explorers around your work area by dragging them by their title bars, and dropping them where you prefer.

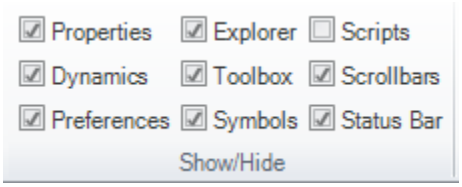


Figure 3.10 - The Show/Hide Section

You might show the Preferences explorer, shown in Figure 3.11, to change the default settings of new objects that you are adding to the workspace. If you find that you are constantly resetting an object's values, consider changing the default preferences for that object.

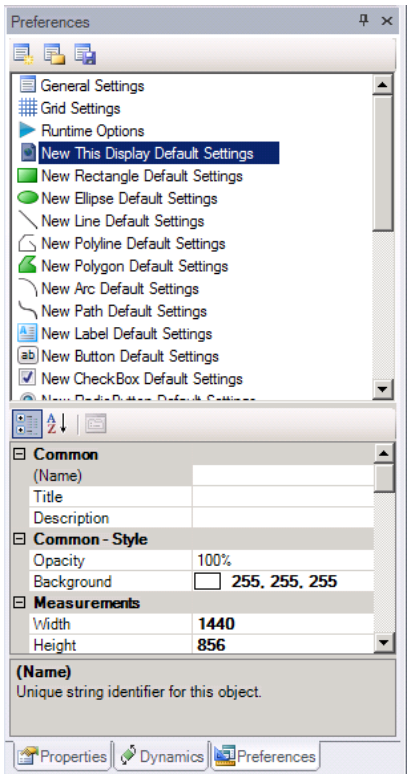


Figure 3.11 - Preferences Explore

3.4.4. Application Style

Using the **Application Style** button in the Misc section, you can change the color scheme of your workspace and you can modify the icon sizes that display in the toolboxes. Figure 3.12 shows the variety of options that are available.

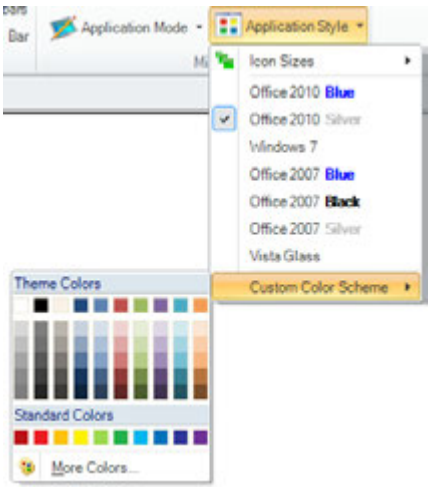


Figure 3.12 - Changing Color Scheme

3.4.5. Runtime Preview

At any time, you can preview exactly how a display will look during runtime by using the Runtime Preview feature in the Misc section of the View tab. When you use this feature, the display remains in configure mode but looks as if it is in runtime. While viewing a display in Runtime Preview mode, you retain all configuration capabilities, which you can get to by using your right-click popup menu. You can also:

- Zoom in and out of the display
- Load a file from a URL or the file system
- Browse backward and forward
- Print the display
- Open the Diagnostics utility, which displays all error information for the current display in runtime. This dialog displays status information for current errors.
- Log in or log out of a display
- Open the Carousel to view a sequence of displays

To exit Runtime Preview mode, right-click the display and choose “Exit Runtime Preview Mode” from the popup menu.

3.5. Templates and Transitions

GraphWorX64 has design tools that help you quickly develop the look and feel of your displays. Using the template and transition design tools can save you time and effort, and can provide an important quality-control feature because you can use them to impose a consistent look and feel throughout your business. Both of these tools are available on the Design tab on the ribbon.

3.5.1. Templates

One way to control a design's layout is by using a template as a starting point. A template can include colors, backgrounds, objects, and any .NET controls. A template can also save an arrangement of display panels in prescribed positions in the content area so you can quickly and easily return to it later. Typically, templates are used as tools for enforcing a uniform look-and-feel among displays.

Figure 3.13 shows some of the preconfigured templates you can use, as well as other features that are available with templates.

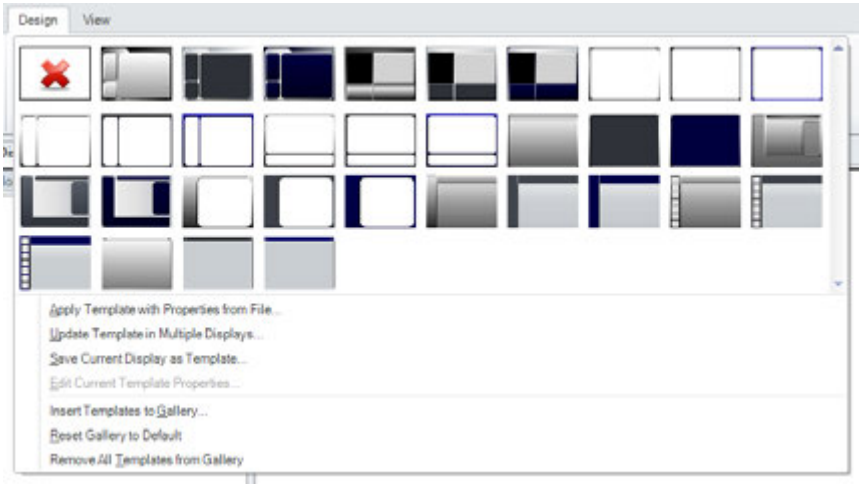


Figure 3.13 - Templates in the Layout Tab of the GraphWorX64 Ribbon

Any GraphWorX64 display can be saved as a template with the .TDFX file extension. Templates can be saved to the gallery or in a folder of your choice. Figure 3.14 compares a display *with* a template and *without*.

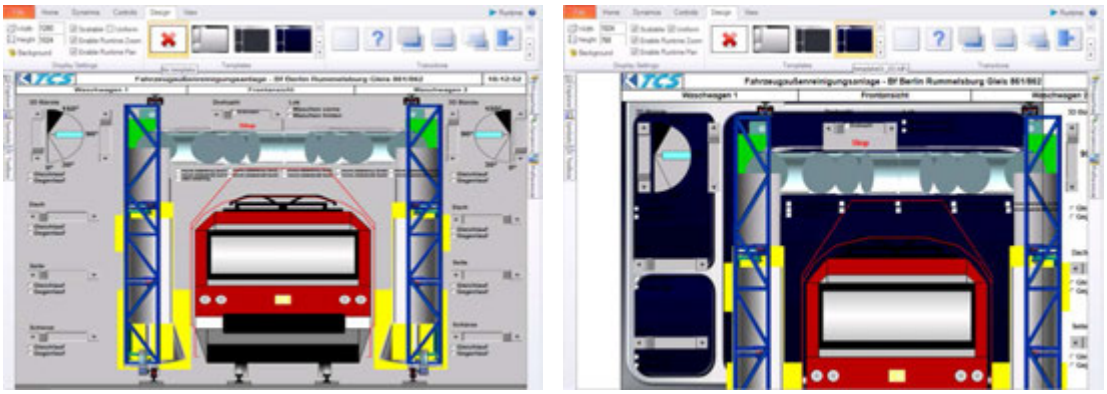


Figure 3.14 - Display with No Template (Left) and Display with Template (Right)

3.5.2. Transitions

Transitions are special effects applied to a display as it closes or loses focus when another display is opened in the same window. The effects work in a similar way to Microsoft Office PowerPoint presentations. You can use them to fade the display out, fade to a blur, or wipe away in any direction or in a rotation motion, to name just a few of the effects. Experiment with them to see which ones you like to use.

It is easy to apply a transition. Simply select the transition from the Transitions section. Transition options are shown in Figure 3.15.

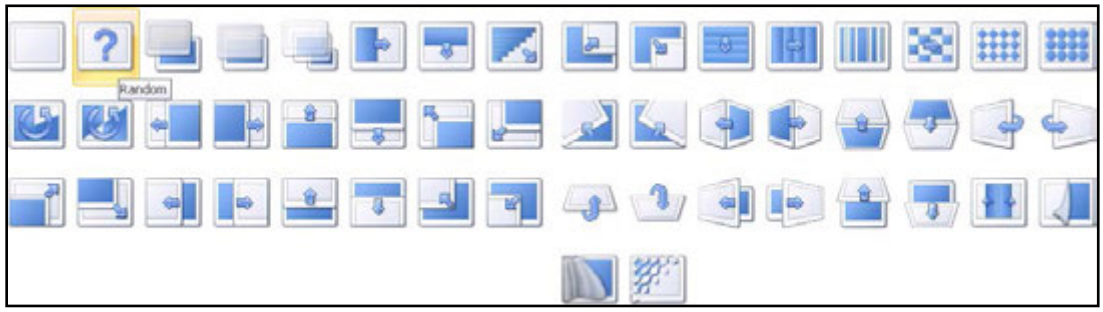


Figure 3.15 - Transition Options

3.6. Symbols Library

A *symbol* is a group of objects that is treated as one single object. Symbols can be user-created or can be loaded from the Symbols Library, available in GENESIS64. The Symbols Library provides a wide assortment of 2D and 3D objects which saves you the time and cost of creating them from scratch, lets you reuse objects from one display to the next, and reduces the need to write scripts for such objects. Think of them as dynamic clipart. Using both static and dynamic pre-made symbols, you can quickly create animated displays without having to start from scratch.

Before creating a symbol from scratch, look in the Symbols Library for a pre-built object that you can use instead. Even if you do not see the exact symbol you are looking for in the library, you can use one as a starting point; modify it, then save the modified version back to the library. Figure 3.16 shows a handful of the wide assortment of symbols that are available to you.

The Symbols Library has its own explorer in GraphWorX64. To display it, put a checkmark in the Symbols checkbox in the Show/Hide section on the View tab. To use a symbol from the library, click and drag the symbol from the Symbols explorer tree onto the GraphWorX64 content area, then use the symbol for your own purposes by changing whatever properties you need.



Figure 3.16 - Sample Symbols in the Symbols Library

Among the symbols you can choose from are automotive, building control panels, gauges, knobs, standard symbols for various machinery, oil and gas, operator control panels, pharmaceutical, food processing and manufacturing operations, water and wastewater operations, in both 2D and 3D.

You can even add your own folders and symbols to the library. Any animated object that you create, along with any attached scripts, is saved automatically and categorized for you so that the next time you use the animated object, you can quickly bind a new data source to the object.

Lab: icTonics Graphical Template *L3*

3.1. Lab Overview

Starting with this lab, we will build a project for the icTonics beverage company. In the next few labs, we will create a series of displays for icTonics. These displays will include a Welcome screen and a Tank Monitor screen. IcTonics has requested that all displays use the same banner to keep a uniform look throughout.

Estimated time to complete this lab: 20 minutes

3.1.1. Task Checklist

In this lab, you will complete the following tasks:

- Explore the GraphWorX64 ribbon
- Explore the Symbols library
- Change the window layout of your tools
- Create a template for icTonics
- Create a Home display for icTonics

3.1.2. Support Files

To complete this lab, you will need the following file(s):

- icTonics-logo.png

3.2. icTonics' Template

We will start by creating a template on which to build our displays.

1. Open the Workbench if you closed it after the last lab.
2. Create a new GraphWorX64 display and save it as icTonicsTemplate.tdfx

3. In the Properties Explorer for the display, change the Measurements to 1024 x 768. Your display will now be at this resolution size.

> ***Tip!*** If you want to maximize your design surface area, you can change the view to the Drawing Layout. To do this, select the **View** tab in the ribbon, in the Window Layout section, select the "Drawing Layout".

4. Add a rectangle that fills the top of the screen. This will be the basis of our icTonics banner.

5. Right-click on the rectangle and select "Object Properties". The Properties Explorer should expand on the right-hand side.

6. Click on the "Fill" area, and then click on the ellipsis button (...) to bring up the fill color dialog box.

7. Select the Gradient tab. Change the first gradient color to green (RGB: 0, 255, 0) and the last gradient color to sky blue (RGB: 0, 212, 255).

8. Delete one of the gradients in the middle by selecting it and hitting the "Delete" key.

9. You should now only have three gradients. Change the middle gradient color to cyan (RGB: 0, 255, 170). Your final gradient color should look like Figure L3.1. Click on "OK" to apply and close the dialog box.

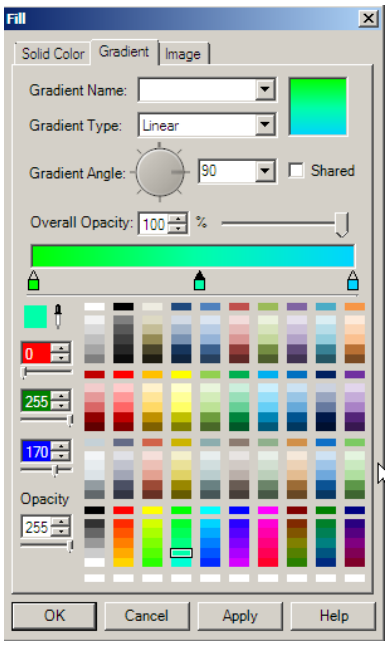


Figure L3.1 - Fill Gradients

10. Change the line color to "none" by selecting the line area, clicking on the ellipse and clicking the red X near the eyedropper in the **Line** Color dialog box.

11. Change the measurement of the rectangle to 1028 x 93. You should now have a rectangle that spans your display. To place it at the top of the display, you can either drag it or, to be more precise, set the left and top measurements to zero.

12. Insert the icTonics-logo.png into the display. You can locate this file in C:\Training, and drag the image onto the design surface. Resize the logo to 283 x 93 and align the left and top of this image to your rectangle.

13. Insert a Time Date Dynamic by going to the **Dynamics** tab of the ribbon and, in the Text Objects section, select "Time Date", and then click on the design surface.

14. Change the properties of the Time Date *object* so that it has a size of 300 x 31. Put this in the upper right-hand corner of the square you made.

15. Set the TextAlignment to "Right".

16. On the Dynamic Explorer, change the DateFormat property to "Current Locale (long)".

17. Next, we are going to insert the Backward, Forward, and Home button from the Symbol library. You can open the symbol library by clicking on the Symbols tab on the left-hand side of the display screen.

18. Once the library opens, click on the drop-down list and select Buttons & Switches.

19. Insert the "Menu Button Back", "Menu Button Forward", and "Menu Button Home" objects onto your display. We will need to make some edits to these objects to match our color scheme and style.

20. Open the object Explorer. You should notice that you have three objects in bold starting with "MenuButton". These are bolded because they are grouped options.

21. Select the "MenuButtonBack1" object and expand its tree. Select its "Classic Border" object and delete this. You should notice that you no longer have a background square for the arrow.

22. Expand the MenuButtonBack object's Group. There should be two more groups in the next level. Expand each of these groups as well.

23. Select the second item in each of the groups and delete it. The arrow should no longer have shadows.

24. Repeat steps 21-23 for your MenuButtonHome and MenuButtonForward objects.

25. We will now change the color for the MenuButtonForward and MenuButtonBack objects.

26. In the MenuButtonBack object, drill down to the properties of the Rectangle and change the fill color to a Gradient. There are only two colors for this Gradient; a light orange (RGB: 255, 170, 0) and a dark orange (RGB: 255, 127, 0).

27. Set the line color to "none".

28. Repeat steps 26-27 for the Polygon in MenuButtonBack as well as the Rectangle and Polygon in MenuButton Forward.

> ***Tip!*** Try using the "Format" painter in the Style and Colors Section of the Home ribbon tab. You can also try saving a Style and applying it.

29. Arrange the three symbol objects from left to right starting with MenuButtonBack, followed by MenuButtonHome, and then MenuButtonForward.

30. We will need to add dynamics to the different buttons.

31. Select the MenuButtonBack object. Click on the Dynamics tab and click on the "Pick" icon in the Buttons section.

32. In the Dynamics explorer, change the Command property to "Display Back".

33. Select the MenuButtonForward object. Click on the Dynamics tab and click on the "Pick" icon in the Buttons section.

34. In the Dynamics explorer, change the Command property to "Display Forward".

35. Select the MenuButtonHome object. Click on the Dynamics tab and click on the "Pick" icon in the Buttons section.

36. In the Dynamics explorer, leave the Command property as "Load Display".

37. In the FileName property, type in "icTonicsHome.gdfx" (without quotes). We will create this display next.

38. Select all three objects by holding down the shift key and then clicking on each. Then click on the Home tab of the ribbon and, from the Arrange section, select the group button. You can give this group a name of "Navigations," if you like.

39. Then place the Navigations object in the lower right corner of the rectangle you created earlier. (You may want to unpin any explorers that are pinned open.) You should now have a banner that looks similar to Figure L3.2



Figure L3.2 - Template Banner

40. Save and close your display.

3.3. icTonics' Home Display

1. Open a new GraphWorX64 display and save it as icTonicsHome.gdfx
2. Go to the Design tab in the ribbon and click on the down arrow in the Templates section, and then select "Insert Templates to Gallery".
3. Click on the "Add Files" button in the **Edit File List** dialog box.
4. Browse to your icTonicsTemplate.tdfx file, select it, and click on "OK" twice. You will now see the icTonicsTemplate template in the Templates section; it looks like a small green bar.
5. Click on the icTonicsTemplate template to apply it to the display.
6. Insert a new rectangle. Change the size to 1028 x 678. The top should be at 92 and the left at 0.
7. Change the color of the rectangle to sky blue (RGB: 0, 212, 255 and Opacity to 255) with no borders.
8. Add a text object by going to the Home tab of the ribbon and then, in the Shapes section, click on "Draw Label" A, and then click on the design surface.
9. Type in the text "Welcome" (without quotes); then change the font family to Lucida Handwriting and the font size to 100.
10. Drag the text label to the center of the display.
11. Add a button to the display by going to the Dynamics tab of the ribbon, clicking on the "Button" ab icon, and then clicking in the display.
12. Change the text of the button to read "Login" (without quotes).

13. Set the size of the button to be 150 x 50 and move it to just below the “Welcome” label.

14. In the Dynamics explorer, change the Command property to “Login/Logout” dialog box.

15. Save the display. We will work on this display some more in the upcoming labs. Your current display should look similar to Figure L3.3.



Figure L3.3 - icTonics' Home Page

Review: Getting to Know GraphWorX64 R3

3.1. Objectives Review

You should now know and understand the following topics:

- GraphWorX64 Object Explorer
- GraphWorX64 Properties and Dynamics Explorer
- Setup preferences in the Preferences explorer
- Customizing the workspace
- Symbols Library
- Templates and Transitions
- Runtime preview

3.2. Summary

The Object Explorer is a tree control that lists all of the objects currently defined in the GraphWorX64 display. Because of its hierarchical nature and its heavy use of icons, the Explorer can tell you several things about each object at a glance.

You can customize your GraphWorX64 workspace by first configuring your content area's properties using the Properties Explorer. You can also use features on the View tab to customize your workspace. And you can preview how your display will look in Runtime mode with Runtime Preview, which presents the display as if in runtime while retaining configuration abilities.

Using the template and transition features can save you time and effort, providing an effective quality control tool that lets you impose a consistent look and feel throughout your displays.

The Symbols Library is a set of folders that contain pre-built design objects that you can incorporate into your displays. Even if you do not see the exact symbol you are looking for in the library, you can start with a predefined symbol, modify it, and then save it into your library for later use.

3.3. Terms

Content Area	Dynamics Explorer	GraphWorX64 Template
Object Explorer	Preferences Explorer	Properties Explorer
Runtime Preview Mode	Symbol	Symbols Library
Transition	Window Layouts	Z-order

3.4. Test Your Knowledge

In this chapter, we discussed some GraphWorX64 options you can use to customize the workspace and to create displays. Before continuing to the next chapter, make sure you can answer the questions below.

1. Using the Object Explorer, what is an easy way to make an object invisible?

    a. Clicking on the eye icon of an object in the Explorer
    b. Right-clicking the object and selecting Show/Hide
    c. Writing a script
    d. All of the above
    e. Both A and B
    f. None of the above

2. How do you add the Object Explorer to your workspace or remove it from your workspace?

    a. Click the Explorer's minimize icon
    b. In the View tab, check Explorer in the Show/Hide section
    c. Right-click the Explorer title bar and select Show or Hide
    d. The Explorer displays at all times

3. What can you tell about an object or group by looking at it in the Explorer?

    a. Whether the object is dynamic
    b. Whether the object is editable
    c. Whether the object is selected in the display
    d. The type of object
    e. All of the above
    f. None of the above

4. Which GraphWorX64 features provide predefined designs that you can use in your displays?

   a. Transitions
   b. Templates
   c. Monitoring utilities
   d. Symbols library
   e. All of the above
   f. A, B, and C
   g. A, B, and D
   h. B, C, and D
   i. None of the above

5. True or False: You can make changes to a display when previewing it in Runtime Preview mode?

6. What are some ways in which you can customize your workspace?

   a. Edit simultaneously with another user
   b. Change the size of icons in the Object Explorer
   c. Save your own workspace layouts
   d. Change your preferred application mode
   e. All of the above
   f. A, B, and C
   g. B, C, and D
   h. A, C, and D
   i. None of the above

7. In configuration mode, what is the best way to prevent an object from being accidently modified?

   a. Hide the object by clicking the object's Show/Hide property to make it invisible
   b. Hide the object by using the View ribbon's Show/Hide section
   c. Lock the object by clicking its Lock/Unlock property
   d. Include it in a grouping
   e. Any of the above

8. How do you change the default settings for an object?

   a. Use the Default Settings explorer
   b. Use the Preferences explorer
   c. Use the Properties explorer
   d. Right-click the object in the Explorer, then click Set as Object Default
   e. You cannot change the default settings for an object

GraphWorX64 Basics 4

As you know by now, GraphWorX64 is a platform for you to develop HMI. Being such a platform, it offers some basic drawing capabilities as well as dynamics and pick actions that animate displays. In this chapter, we will explore this aspect of GraphWorX64. After completing this chapter, you should know how to do the following:

- Work with 2D shapes and objects
- Work with text objects
- Work with 2D color and dimension dynamics
- Work with pick actions, buttons, and the Function Key action
- Count objects in your display
- Change the application mode

4.1. Basic 2D Shapes and Objects

GraphWorX64 uses vector graphics for designing animated objects in its displays. It also supports data display and processing of OPC data from a variety of data sources. When you open a GraphWorX64 display in Configuration mode, you can add shapes and other objects including bitmap images, labels, and other static objects.

Some objects (such as buttons) come with actions, called *dynamics*. By default, buttons perform a dynamic called the Load Display action, which means that at runtime a click of the button animates the loading of a display. But you can change that action to any of a number of other possibilities. You can create a button that performs a sequence of actions when clicked. But you can change the trigger from clicking the mouse to something else, like moving the cursor over the object.

First, let's start with the basic shapes in GraphWorX64, which are:

- Line 
- Arc 
- Spline 
- Polyline 
- Rectangle, square 
- Ellipse, circle 
- Polygon 

Once you have drawn basic shapes, you can combine them into new, more complex shapes by using the Combine Objects button in the Home ribbon's Arrange section. Figure 4.1 shows an example of the different commands that you can use to create a complex shape.

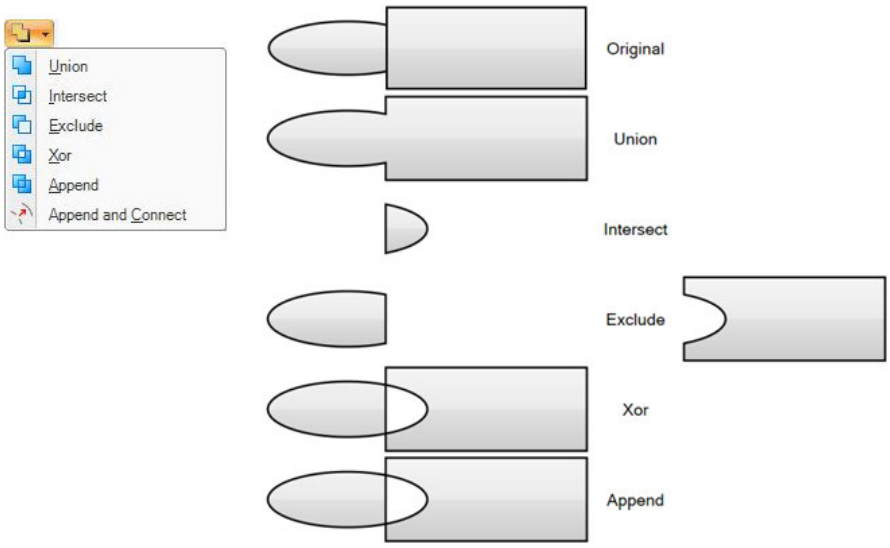


Figure 4.1 - Combine Basic Shapes into Complex Shapes

Using these and other shape manipulation methods, you can create complex and dazzling objects from basic shapes. You can flip shapes horizontally, vertically, or rotate them in 90° increments using options in the Arrange pane on the Home ribbon, shown in Figure 4.2.

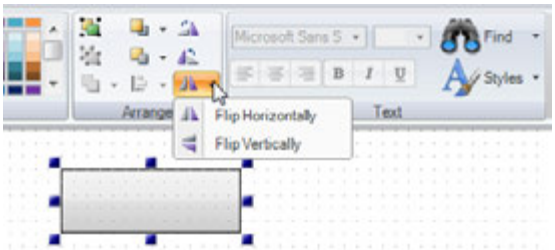


Figure 4.2 - Flip and Rotate

You can freely rotate the shape by dragging the corner when you see the arrows connected by a curved line:  . This handle appears when you hold the cursor on the outside of a shape's *corner* handle. Grab the corner to rotate the shape freely or, if you hold down the Shift key, the rotation happens in 15° increments.

Skewing the shape is simple, too. Just hold the cursor on the outside of a *side* handle until you see the mouse change into two arrows separated by a line, similar to:  . Then grab the handle and move it using your mouse to skew the shape.

Experiment on your own. Try out different shape combinations to get astonishing new shapes like the one shown in Figure 4.3. It is made by using a circle in the middle, crossed by four ellipses, and then applying the XOR arrangement. Then we applied a style from the Style and Colors tab on the Home ribbon to the shape.



Figure 4.3 - Complex Shape Created Using Ellipses with a Predefined Style

Combining these shapes with smart symbols and controls will give you the ability to create easily dashboards (like the one shown in Figure 4.4), processing and electrical systems, facilities, campuses, and much, much more.

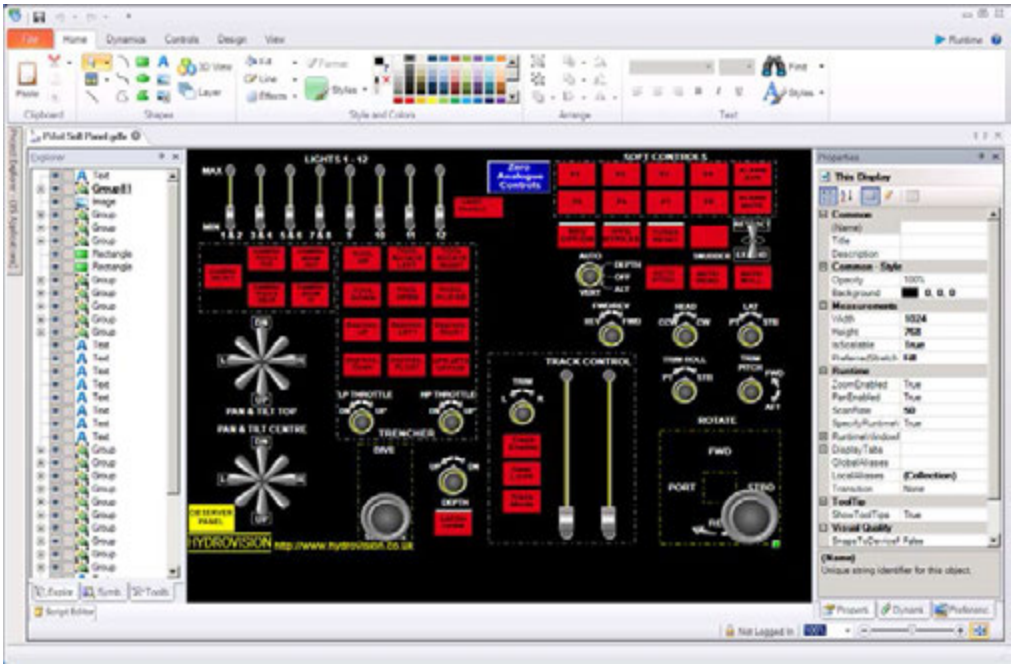


Figure 4.4 - Example of a Display in Configuration Mode

In Figure 4.4, you can see sliders, dials, buttons, lights, and switches. You could add temperature gauges, a clock, indicators, pressure gauges, and perhaps a dozen other active data points. This GraphWorX64 display has, among its many object types, push buttons that perform actions such as turning a pump on or off. We could add thumbnails of other displays that open additional GraphWorX64 displays for TrendWorX64, AlarmWorX64, and EarthWorX64. GraphWorX64 puts the power of the entire GENESIS64 programming suite at your disposal.

4.2. Dynamics

As suggested by their name, *dynamics* are objects that show changing information. A *process point* is one example of a dynamic. The value of a process point changes as it reads the changing value from the OPC server. Some objects are dynamics. But you can also apply dynamics to shapes and objects to show process information graphically. For example, you can attach a dynamic to a rectangle to create a slider. In this section, we will discuss the different types of dynamics available in GraphWorX64.

4.2.1. Text Objects

Process Points, Data Entry, and Time Date objects are three variants of a dynamic object type called *Text Objects*. You can create any of these three objects by using the Text Objects section in the Dynamics tab on the ribbon, as shown in Figure 4.5.

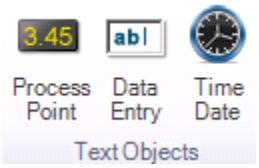


Figure 4.5 - Text Objects

A **Process Point object** is a text object that you can associate a data source to and display dynamic or static values in a GraphWorX64 display. By default, a Process Point is a read-only object; runtime users typically don't need to update the value of a Process Point, they just want to see its value. You would use a Process Point to display OPC-UA tag values, results from expressions, or any other time-dependent data.

A **Data Entry object** is a Process Point object but with different default property settings. The most striking difference is that, by default, it can be updated and it has a textbox that allows runtime users to enter values. Use a Data Entry object for an OPC point that runtime users need to be able to set or change the value of. Where you use a Data Entry object in a GraphWorX64 display, runtime users can change a state, assign a value to a variable, or make a selection that controls a process, changes other objects, or provides other forms of user-generated input – depending on how you configure the object.

Advanced settings in both of these object types can make a Process Point writeable and a Data Entry object read-only, essentially interchanging their default behavior leaving only slight visual clues that their original identities were something different. Typically you'll have no need to do this though.

The third Text Label object is the **Time Date object**, which displays the time and date. The Time Date object's properties are similar to those of Process Point and Data Entry objects.

4.2.2. Dimensions and Color Dynamics

Besides text objects, there are other dynamics such as Buttons, Dimensions, colors, etc., that you can use to create a spectacular display. In particular, the dimensions and color dynamics are most often used to add animation to a display. Some objects in the Symbols Library, such as knobs and gauges, come with dynamics that animate them.

To use these dynamics, you start by connecting a shape to an OPC tag so that the shape can receive the tag's values. Then you apply the dynamic to the object. This lets you use the value of the OPC tag to set the size, color, location, etc., of the object. Figure 4.6 illustrates a list of the simple dynamics you can apply to a shape or object. You can change an object's size, location, and color, and you can rotate it, hide it, and disable it.



Figure 4.6 - Dimensions and Color Dynamics

Also, you can combine any of these dynamics into one object. For example to create a ventilator like the one in Figure 4.7, you can use the *rotation dynamic* to spin it and the *color dynamic* to make it change colors depending on the speed. Notice that the object's dynamics are listed at the top of the Dynamics explorer, where there is also a dynamics toolbar so you can add and remove the object's dynamics if you like.

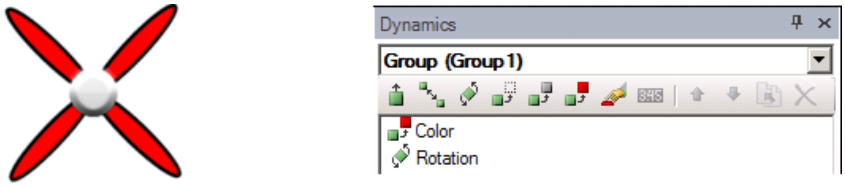


Figure 4.7 – A Ventilator (Left), with its Dynamics in the Dynamics Explorer (Right)

4.2.3. Pick Actions

A Pick action is a dynamic that is triggered by an action, such as a mouse click, taken by a runtime user. You define the user action and what happens in response to the user's action. Most commonly, a Pick action opens a display or a popup window, downloads a value to an OPC Tag, or runs a script. You can add a Pick action to any shape by using the Pick button on the Dynamics ribbon. Then, in the Pick action's Dynamic properties, shown in Figure 4.8, you specify the user action that will *trigger* the action, and choose the *command* that the action will perform. Notice that when you click on a property, a helpful tip appears at the bottom of the Dynamics explorer.

There are two different types of Pick actions in GraphWorX64: Local Pick Actions and Global Pick Actions. Local Pick Actions refer to actions within the GraphWorX64 display while Global Pick Actions utilize AssetWorX Commanding to affect modules outside of the GraphWorX64 display. While some Pick Actions are duplicated and may seem similar between Local and Global types, there are differences in the properties.

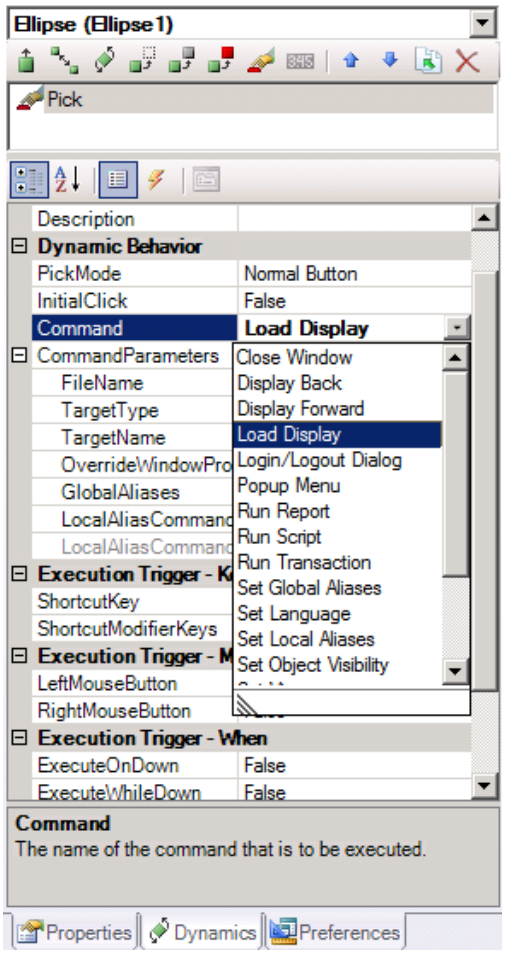


Figure 4.8 - Pick Action Dynamic Properties

Let us look at each Pick action command below.

- The **Close Window** pick action closes the Pick action's window or display.
- GraphWorX64 maintains a history of the last 50 displays. You can navigate through the display history using the pick actions **Display Back** and **Display Forward** command. These commands are also available as OLE Automation methods. The Display History feature is similar to Microsoft Internet Explorer's Back/Forward feature.
- The **Load Display** pick action opens a window of the type that you specify in the dynamic's properties. Windows can be dependent or independent popups, modal, or embedded. They can even refer to the current display (which is a *refresh* operation). You can specify other parameters, such as what loads in the display. Load Display is the default command for a Pick action.
- The **Login/Logout Dialog** command opens a dialog in which the runtime user can log in or log out.
- The **Popup Menu** pick action allows you to display a collection of pick actions that the runtime user can choose from. You can configure the position of the popup menu.
- The **Run Report** pick action allows you to execute a ReportWorX report during runtime.
- The **Run Script** pick action runs a script that you have created using JScript.NET in the display.
- The **Run Transaction** pick action allows you to execute a BridgeWorX transaction during runtime.
- The **Runtime Menu Command** allows users to access the Ribbon menu when it is not visible. Displays can be configured to embed the commands available in the Ribbon to customize operator options and view.
- The **Set Global Aliases** pick action allows you to set the global alias theme at runtime for the This Display object when the pick dynamic is triggered. This is especially helpful if global aliases are used in your OPC tags.
- The **Set Language** pick action sets the runtime language to a language of your choosing.
- The **Set Local Aliases** pick action sets aliases for data connections in runtime mode.
- The **Set Object Visibility** pick action applies a Hide/Show dynamic to an object based on the condition and using the trigger you specify.

- The **Set View** pick action takes the runtime user to a specified view in the display. Think of a view as a location or detailed close-up in an especially large display. To help runtime users navigate large or complex displays, you can create views that animate a zoom or scroll to areas of detail that runtime users need access to. This command lets runtime users access those views quickly and easily. We will talk more about views in the next chapter of this course.
- The **Start Application** pick action launches a specified application or document in runtime mode. In addition to launching executables, you can launch various documents, including Microsoft Word documents, Microsoft Excel spreadsheets, HTML files, and text files. The registered application is automatically started when the file is launched.
- The **Toggle Value** pick action is used to toggle between two specified values or states during runtime mode using an action that you specify.
- The **Update Database** pick action updates an OLE DB data source with the current values of tags that you specify based on the condition or action that triggers the pick action.
- The **Write Value** pick action writes a value that you specify or the value of one tag to another tag.

4.2.4. Button Objects Preloaded with the Pick Action

The Pick action is a dynamic action that can be applied to any shape. However, GENESIS64 gives you four button objects that are pre-loaded with the Pick action so that you do not have to create buttons from scratch. The same commands that are available for the Pick action are also available with the button objects.

To use these objects, go to the Buttons section of the Dynamics ribbon. Figure 4.9 shows them on the ribbon along with an example of each type. The simple, clickable *button* object displays a text prompt. The *checkbox* object lets runtime users click to add a checkmark. The *radio button* object can be lined up with other radio button objects to give runtime users a choice among multiple options. A *graphical button* displays the miniaturized image of another GraphWorX64 display. All allow you to specify the trigger that executes the Pick action command that you configure for it.

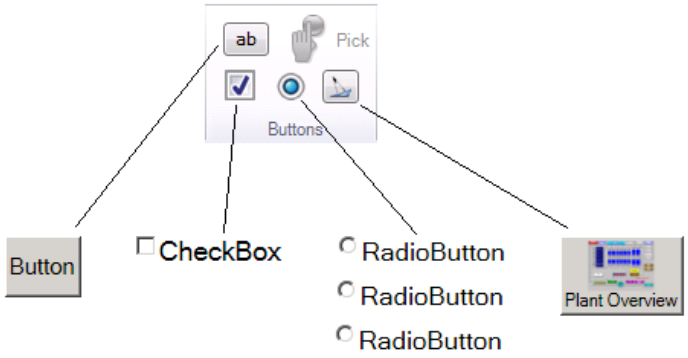


Figure 4.9 - Button Objects that Are Preloaded with a Pick Action

4.2.5. Function Key Action for the This Display Object

The Function Key action, available in the This Display section of the Dynamics ribbon, is nearly identical to a Pick Action, but it is applied to the This Display object only and is initiated by a keystroke. Function Key actions are similar to application keystroke macros that you assign in a document. The same commands that are available for the Pick action are also available for use with the Function Key action.

4.3. Object Count

As you create a display, you'll find that some objects seem to hide when you look at the display because objects overlay each other or they are grouped. If you want to see the count of objects in a display, their types, the number of data connections, or the number of dynamic animation types defined, you can click the Object Count button, which is in the Misc section of the View ribbon.

If no object is selected when you click on the button, you will see a list of all objects in the display. As you can see in Figure 4.10, the Object Count dialog counts the objects, as well as giving you other more information about your display.

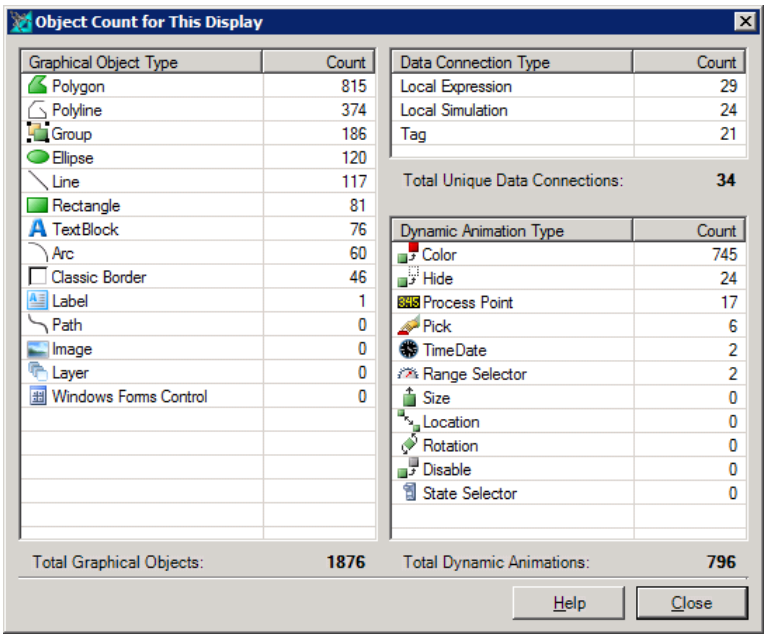


Figure 4.10 - Object Count Dialog

4.4. Application Mode

GraphWorX64 comes in three different user-selectable configuration modes: Basic, Standard, and Advanced. Depending on your comfort level and engineering requirements, you can select different modes that best suit your needs and preferences while in configuration mode.

- **Basic Configuration Mode** limits the features of GraphWorX64 to a "beginner-level" configuration for relatively simple applications. In basic mode, most advanced features are hidden from view.
- **Standard Configuration Mode** is appropriate for configuration and development of most typical applications. In standard mode, some advanced features are hidden from view.
- **Advanced Configuration Mode** is recommended for high-level configuration and powerful development capabilities for the most complex applications. In advanced mode, all GraphWorX64 features are available. As a power user of GraphWorX64 you will most probably use the advanced mode.

To change modes, click the Application Mode button on the View ribbon.

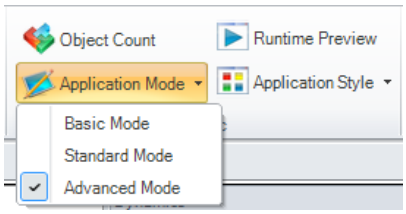


Figure 4.11 – Choose a Configuration Mode

To gain an appreciation of the differences between the three modes, consider Figure 4.12, which shows the Properties explorer for a Button object in each mode. *Basic* mode hides all properties except a small collection with the key properties of Text (label), Fonts, Colors, and Scaling. In *Standard* mode, you see about 80% of all the properties, which is sufficient in most cases. Finally, *Advanced* mode shows all properties that you can control in configuration mode.

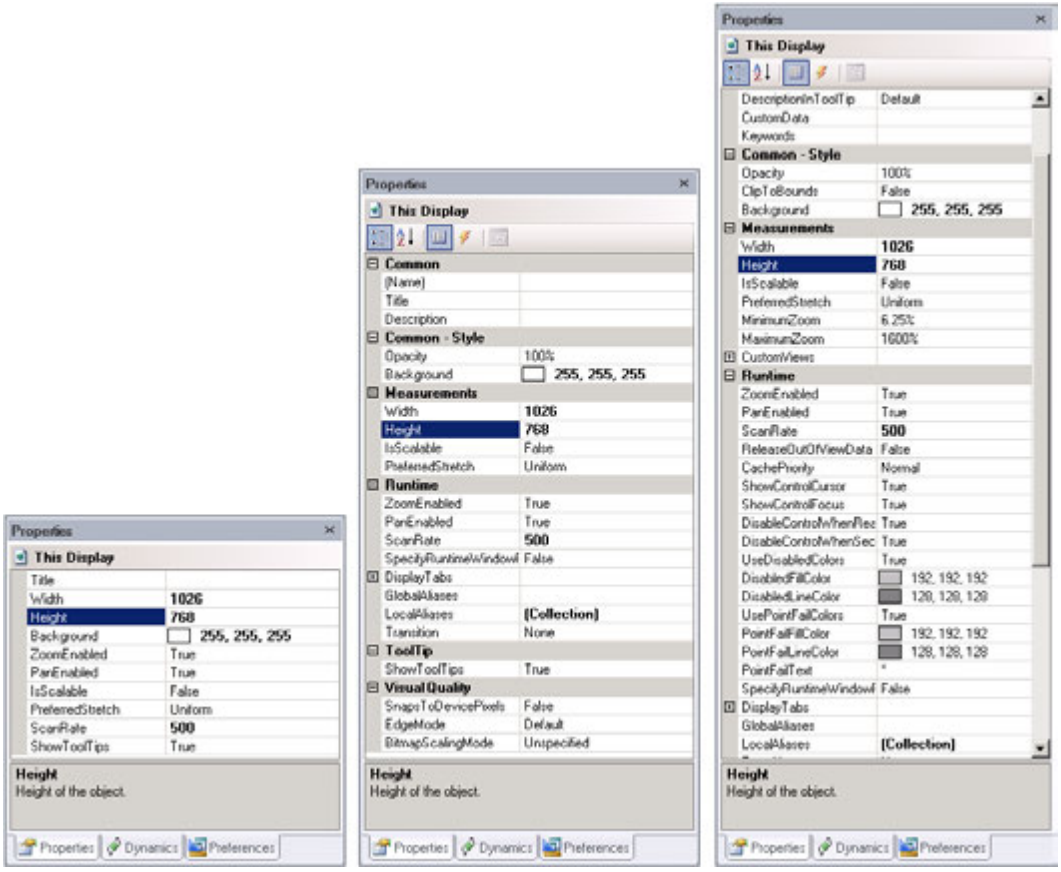


Figure 4.12 – Basic (Left), Standard (Middle), and Advanced (Right) Modes

4.5. Display Password Protection

This application note provides a guide to password protect your screens so no one else can make changes to them in design mode. This can only be done by selecting Application \ Advanced Mode when designing screens in GraphWorX64 like was described in previous chapter.

When you have set the Advanced mode of your GraphWorX64 and go to properties of created display, you should see Protection property in Properties tab, like is showed below.

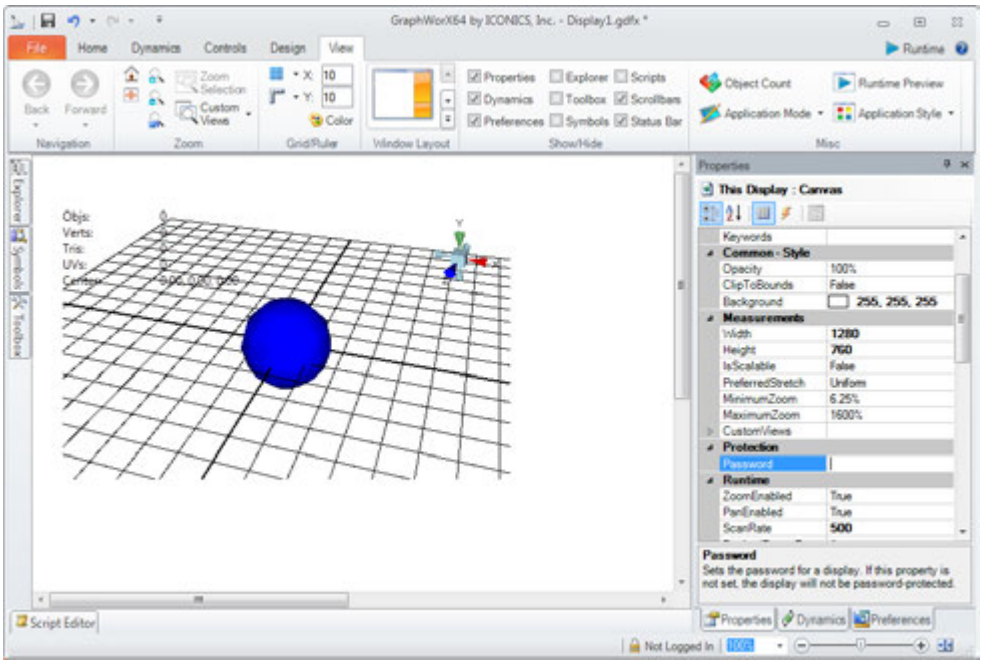


Figure 4.13 – Protection Property - Password

There you simply write your password by which you wish to protect your display. When you open now your display in configuration mode, the Display Password window will be opened and you will not be allowed to do anything without putting right password on it. In case you forgot the password the Challenge tool is placed in same dialog to you will be able to reach display using technical support.

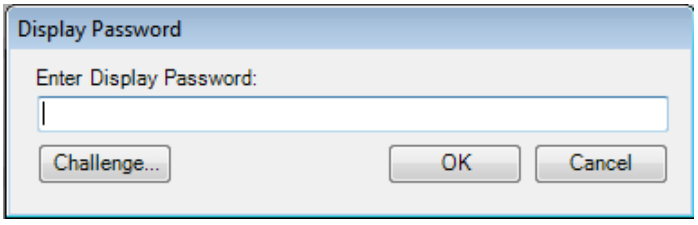


Figure 4.14 – Display Password

Lab: icTonics Monitor Display L4

4.1. Lab Overview

In this lab, we will create a display for icTonics' operators to use to control the mixer tank as well as provide some graphics that indicate what is happening on the plant floor. This will allow the operators to control what to make and how to make it with a few easy clicks.

Estimated time to complete this lab: 40 minutes

4.1.1. Task Checklist

In this lab, you will complete the following tasks:

- Create a display that monitors the drink mixer tank
- Use Location dynamics to indicate the direction of flow
- Use a Size dynamic to show the tank level

4.1.2. Support Files

To complete this lab, you will need the following file(s):

- icTonicsTemplate.gdfx

4.2. Monitor Display

For the tank monitoring display, you need to build a mixer tank. This tank has two valves to allow water and certain syrups to flow into the tank for mixing. We will build this tank, as well as add animations to it to make the display more dynamic.

1. Open the Workbench if you have closed it after finishing the previous lab.
2. Create a new GraphWorX64 display and apply the icTonics template to it. Save the display in the GENSIS64Training directory as "icTonicsTankMonitor.gdfx".
3. From the GENESIS64TrainingSymbols Symbol library, insert a "BaseTank" and two "Pipes". Also, from the **_General → Tanks → 2D_Tanks** library, insert the "Jagged Cut-Away 2". Arrange the objects to look like Figure L4.1.

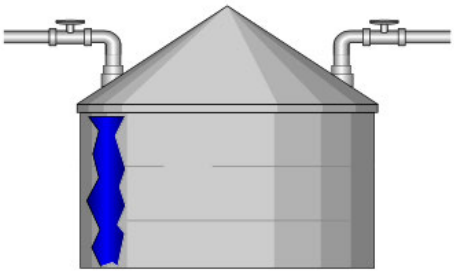


Figure L4.1 - Mixer Tank

4. On the left, select the Explorer tab next to the Symbols tab. In the Tree Explorer, you should see a "JaggedCutAway2" group. Drill into this group and select "Jaggedcutaway4" text. This is the blue polygon and if you look in the Dynamics explorer, on the right, it already has a size dynamic attached to it.
5. Change the DataSource of the Size dynamic to our Tank1 Level tag. You can do this by clicking in the DataSource field, then click on the "Browse" (ellipse) button to bring up the Data Browser.
6. In the Data Browser, select the OPC UA tab if it is not already selected. Then go to OPC Classic Servers → My Server → ICONICS.IconicsOPCUAServer.V5 → icTonics → Tanks → USA → Foxboro → Tank1 → TankLevel. Click on "OK" to choose the TankLevel tag and close the Data Browser.
7. Set the LowLimit in the Size dynamic to 0 and the HighLimit to 100. Place the mouse cursor into any white space in the display and click the left mouse key until the JaggedCutAway is de-selected.

8. From the **Dynamics** tab on the ribbon, in the **Text Objects** section, select a **Data Entry**. Place the data entry to the right of the JaggedCutAway symbol. Link the DataSource to the same Tank Level tag as you selected previously.

9. In the Dynamics properties of the object, find the **PostfixLabel** property (*If you can't see it in the Dynamics, go to the View tab in the Ribbon, click Application Mode in the Misc section, and select Advanced Mode*) and add the percentage sign (%) in the field.

10. Select everything that you have created thus far then click on the Group Objects button in the Home ribbon's Arrange section.

    Note: You can still edit the individual pieces of the object by double-clicking on the pieces. You can also select the objects that you want to edit in the Object Explorer.

11. In the Symbols library, browse to GENESIS64TrainingSymbols and drag the "RightArrow" onto the display.

12. Resize the arrow so that it fits inside the pipe on the left. In the Dynamics toolbar, select the Hide dynamic. Click on the ellipse to the right side of the DataSource field and use the Data Browser to define the tag below as the data source:

    *@ICONICS.IconicsOPCUAServer.V5\icTonics.Tanks.USA.Foxboro.Tank1.**SyrupValveStatus**.Value*

13. Using the drop down list, change the **HideWhen** property to "data == 0".

    Select the Location dynamic, and then click on the DataSource field to set it to *@ICONICS.IconicsOPCUAServer.V5\icTonics.Tanks.USA.Foxboro.Tank1.**SyrupValveStatus**.Value*

14. Go to the Path property. Using the drop down list select the "**Edit Path On-Screen**" option. Notice that a red path appears on the pipe.

15. Drag the end of the path to the end of the pipe and drag a point in the middle of the path to the elbow of the pipe. Your path should look similar to Figure L4.2.

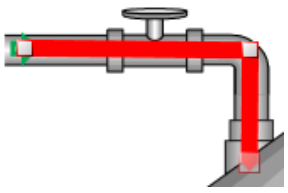


Figure L4.2 - Location Dynamic

16. Set the Location dynamic's **AutoRotate** property to "True". Change the Duration to "3000" (3 seconds) and the **RepeatCount** to "Infinite".

17. With only the arrow selected, use Ctrl-D to duplicate the arrow. Place the duplicate on the pipe on the right. Change the DataSource in the Hide and Location dynamics to use the **WaterValveStatus** tag and reset the Path of the Location dynamic.

18. Select the Explorer tab on the left. Drill into the Group object to select the valve on the left. With the valve selected, add a pick action by clicking on the "Pick" icon under the Dynamics tab.

19. From the Properties tab on the right, change the **Command** for the pick action to "Popup Menu".

20. Scroll down to the MenuItems property, and click on the "Browse" button to bring up the **Edit Menu Items** dialog box.

21. Click on the "Add" button to add a menu item. You will then see a properties menu similar to the one in GraphWorX64.

22. Change the **Command** to "Write Value". **Expand** the **CommandParameters** section and change the OnUpValue to 1.

23. Change the **MenuText** property value to "Open" and set the data source to the SyrupValveStatus.

24. Add a second Menu Item. This second one will be similar to the first except the MenuText should be set to "Close", and then change the OnUpValue to 0. Your finished configuration should look similar to Figure L4.3.

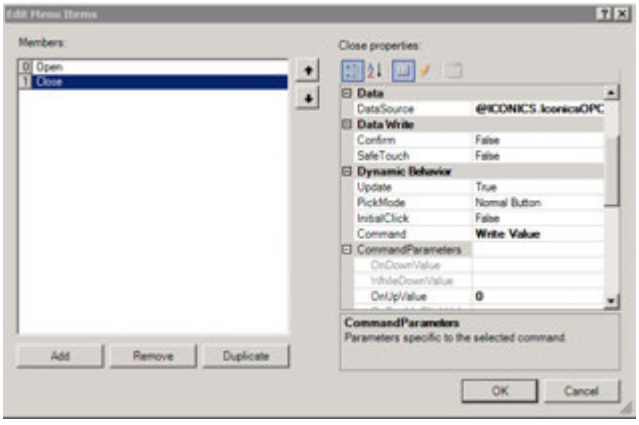


Figure L4.3 - Popup Menu Configuration

25. Click on "OK" to save the configuration and close the dialog box.

26. Repeat steps 18-25 for the valve on the right-hand side of the tank. (Except change the data source to **WaterValveStatus**.)

> ***Tip!*** You can copy and an object and right-click on a destination option to choose "Paste Dynamic" to paste dynamics from one object to another.

27. Save the display and go into runtime. You can click on your Valves to open and close them to test the display. Your finished display will look similar to Figure L4.4.

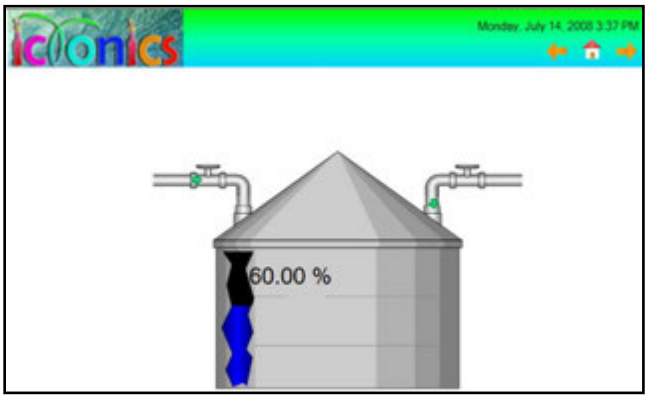


Figure L4.4 - icTonics Tank Monitor Display

Review: GraphWorX64 Basics R4

4.1 Objectives Review

You should now know and understand the following topics:

- 2D shapes and objects
- Text objects
- 2D color and dimension dynamics
- Pick actions, buttons, and the Function Key action
- Counting objects in your display
- Application modes

4.2 Summary

GraphWorX64 is an application that supports data display and processing of OPC data from data sources. In this chapter, we reviewed the objects you can use to create dashboards, consoles, and other displays, and we looked at some ways to animate your displays and make them interactive.

With shapes, you can create visual objects that you need for a display. Some objects (such as buttons) come with actions called *dynamics*. Other objects don't have any dynamics, but you can add dynamics to them to animate them. A library of smart symbols makes it easy to add an object such as a dynamic knob or slider, change a property such as the data source, and create highly effective HMI displays.

Depending on the nature of the OPC tag, you can give its object changeable characteristics, such as a changing size or location or color, or give it motion. You can also hide it, disable it, or use a pick action that lets you open windows and popup menus, run scripts, write values, and more.

When you have many objects in a display, you can use the object count tool to see the total number of objects on the display. This can be very useful if you have a limited license that restricts the number of data sources allowed on the display

If you are an advanced user, you can change the application mode to show more options in the Properties explorer, giving more flexibility in controlling your objects on the display.

4.3 Terms

Application Mode	Data Entry Object	Dynamic
Function Key Action	Pick Action	Process Point Object
Text Object	Time Date Object	

4.4 Test Your Knowledge

In this chapter, we discussed how to use GraphWorX64 to create a display containing shapes and dynamics. We added some dynamics to the shapes and looked at dynamic objects. Before continuing to the next chapter, make sure you can answer the questions below

1. True or False: Creating a complex object is easy in GraphWorX64.
2. True or False: You can combine more than one dynamic into one object.
3. True or False: A Process Point, being a read-only object, can be changed to a writable object.
4. True or False: The Date Time object can show the date and time appropriately in accordance with local preferences (for example, US 3/24/2008, EUR 24.03.2008).
5. True or False: Clicking the "Object Count" button in the Misc section of the View tab gives you a count of the selected objects in the display.
6. True or False: The configuration mode hides or shows object properties on the basis of one's level of expertise.
7. True or False: You can use a dynamic to disable an object in the display, but you can't use a dynamic to hide it.
8. True or False: You can use a Pick action dynamic in a display to close a window, open a window, start an application, change the language of the display, run a report in ReportWorX, pop up a menu, update the database, toggle a value, open the Login/Logout dialog, navigate back and forth in display history, run a script in the display, set the display's global alias theme, set local aliases for data connections, change the display's view, and write a value to a tag in runtime.
9. True or False: Function keys can be configured using hidden button objects in the display.

GraphWorX64 Tools 5

5.1 Objectives

In the last chapter, we discussed some basic dynamics available in GraphWorX64. Besides visible animations and dynamics on an object, you can also use controls. In this chapter, we will concentrate on how GraphWorX64 use controls to retrieve data.

After completing this chapter, you should know how to:

- Add controls
- Arrange objects in a display
- Work with layers
- Work with the Search and Replace feature
- Add views
- Use the Runtime ribbon and menus

5.2 GENESIS64 Controls

If you are familiar with GENESIS32, you would be familiar with the term ActiveX. In GENESIS64, there are no more ActiveX. Instead, we have controls, which are containers that you can insert into your GraphWorX64 display to show different information such as alarms, trends, etc. There are several controls in the Controls ribbon and the Toolbox panel, as shown below, which you can insert into a display.

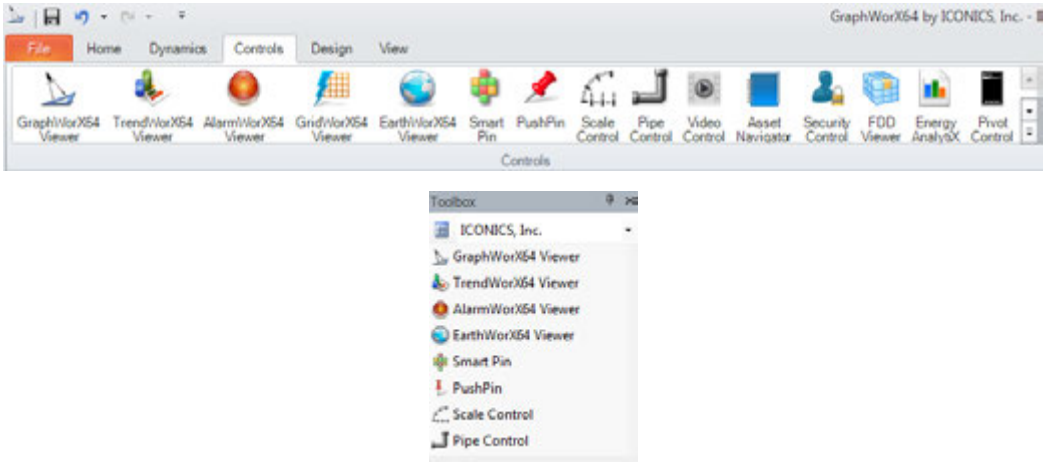


Figure 5.1 - Controls Ribbon (Above) and the Toolbox (Below)

From the Toolbox, you can drag a control to the content area. From the Controls ribbon, you can click on the, and then click on the content area. This creates a control at the default size, which you can resize in the content area.

> **Note:** The Toolbox is often docked at the left side of the configuration window. You can show or hide it using the Show/Hide tab group on the View ribbon. The Toolbox can be pinned as a tab to a side of the display or it can float freely when you unpin it from its docked state.

Some of the controls are ICONICS viewers and others are complex objects that serve a specific purpose. One control lets you add third-party control. All can be added to and used in a display.

5.2.1 ICONICS Viewer Controls

Viewers for ICONICS containers provide a portal into viewing information that comes from files that are external to the display. By adding a GraphWorX64 Viewer, you can a portal to a separate GraphWorX64 display inside the display you are configuring. By adding an EarthWorX64 Viewer, you can add a portal to EarthWorX64 and create a map inside the display. The same is true for including a TrendWorX64 Viewer, and an AlarmWorX64 Viewer. Once you add a viewer to a display, you can work inside the viewer to create a new display in the portal.

5.2.2 Smart Pins

The Smart Pin is a collection of squares called SmartIcons that form a set of indicators. Each square's location in the grid is meaningful. You strategically arrange SmartIcons and attach them to data sources to create a color scheme that allows runtime users to monitor the status of multiple data sources, all at the same time.

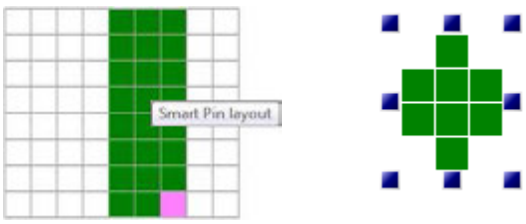


Figure 5.2 - Smart Pin Configuration Layout (Left) and a Selected Smart Pin in a Display

A Smart Pin provides an effective visual indicator for dense data in an overview for an EarthWorX64 map or a GraphWorX64 display. For example, you could use Smart Pins throughout a geographically dispersed system; by making use of color and blink control, you can give users instant access to changing conditions.

When you enable a Smart Pin's tooltips, it becomes possible for users to easily drill down to underlying data and address time-sensitive issues quickly. Smart Pins are ideal for monitoring underlying conditions in a dense area. Examples include monitoring patients in a hospital, water pipes in a vast water system, or toxic gas levels at numerous isolated locations.

5.2.3 PushPins

The PushPin is a simple shape with complex features, which you can add to an EarthWorX map or a GraphWorX64 display for identifying locations or assets. Runtime users can click a PushPin to zoom to a different level of the display. A PushPin can have any number of data values associated with it. Each value can come from a data source, and if you add multiple data values, each value has its own key. The key can be accessed by Pick actions or scripts in the display, allowing you to use PushPins for a purpose of your own design. For example, you can use data values and aliases in the PushPin's tooltips so that runtime users can easily access information about the location that the PushPin identifies.

PushPins can be created using different colors and shapes (shown in Figure 5.3), and you can even use your own image files. Your ability to customize the look makes it easy for your runtime users to find and understand the PushPin images in a map or display.

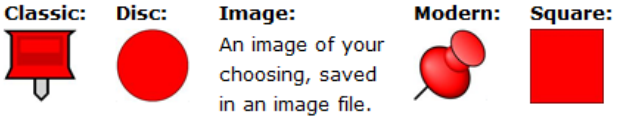


Figure 5.3 - A Few PushPin Shapes

5.2.4 The Scale Control

When drawing a gauge or a scale, arranging the face of its dial can be cumbersome: spacing the tick marks evenly, and adjusting the marks and numbers can be tricky. GraphWorX64 now provides a preset Scale Control that you can use instead. The purpose of a scale control is to provide a shortcut to creating a symbol or object that has a dial or ruler, such as a gauge, a thermometer, a scale, a clock or watch, a meter, a speedometer, or a visual counter.

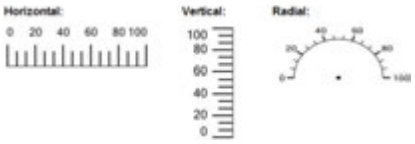


Figure 5.4 - The Scale Control

5.2.5 The Pipe Control

The Pipe Control makes it easy to create pipes and elaborate plumbing schemes in a GraphWorX64 2D display by allowing you to add sections of pipe to a display, and then bend and cap those sections. A handy feature of the Pipe Control is that you can easily animate an object so that during runtime, it moves along the path of a 2D pipe.

5.2.6 Pivot Control

The Pivot Control creates a simple menu infrastructure that includes *pages* of information. Each *page* is a GraphWorX64 canvas and can include any other control, object or dynamic. *Pages* can be added via the contextual ribbon menu items or from the buttons at the bottom of the Property Grid.

5.2.7 Video Control

Video control is able to handle streaming or recorded video. As you can see below, you just simply add the Video Control from Controls Ribbon Menu and place it to your display. In Video Control's properties you are able to browse for video source.

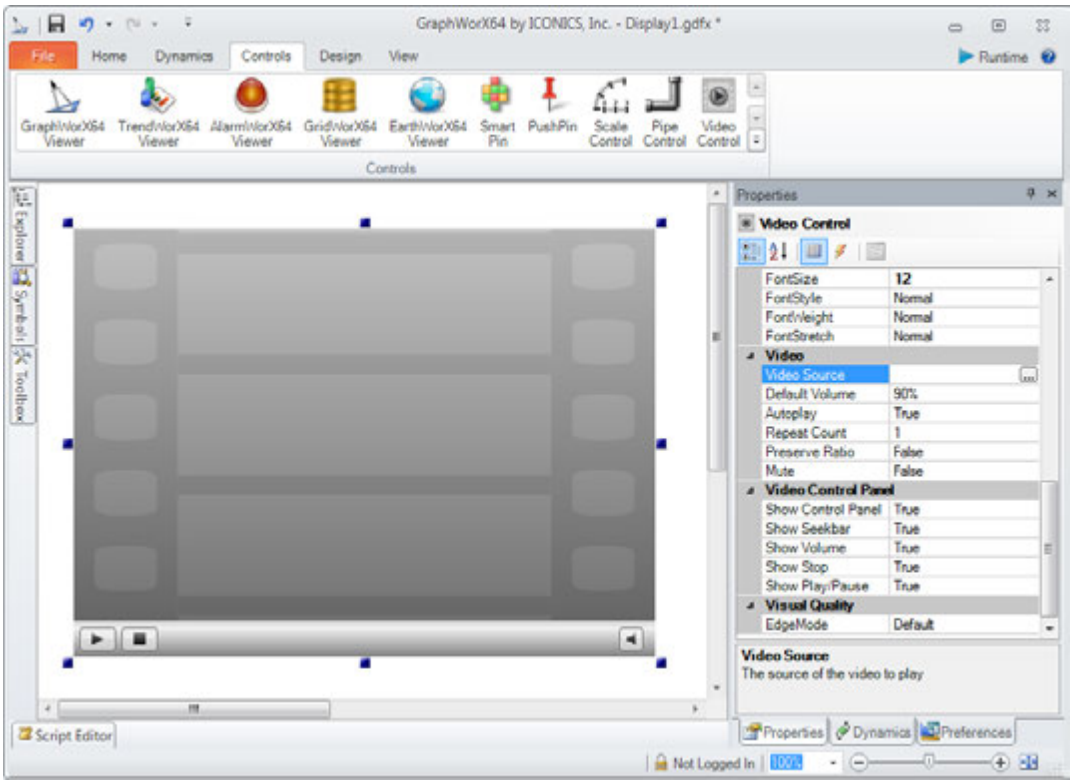


Figure 5.5 - The Video Control

You can view two different types of video. First is already existing video saved in memory, for which you are able to simply browse by button in video source. You need to have selected Filesystem like a Source type and click to Browse button to select your video.

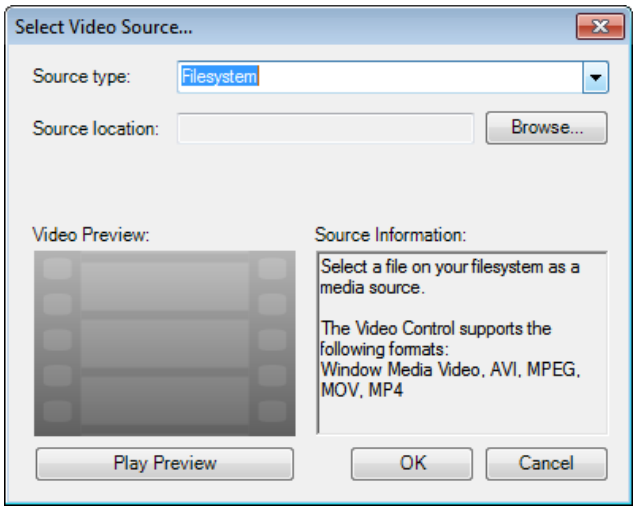


Figure 5.6 - The Saved Video From File

The second type of video you can play with GraphWorX Video Control is Streaming video. With this you can connect to a URL address where a stream resides and easily play it inside this control. For streaming video select the Streaming source type and write the stream URL into source location.

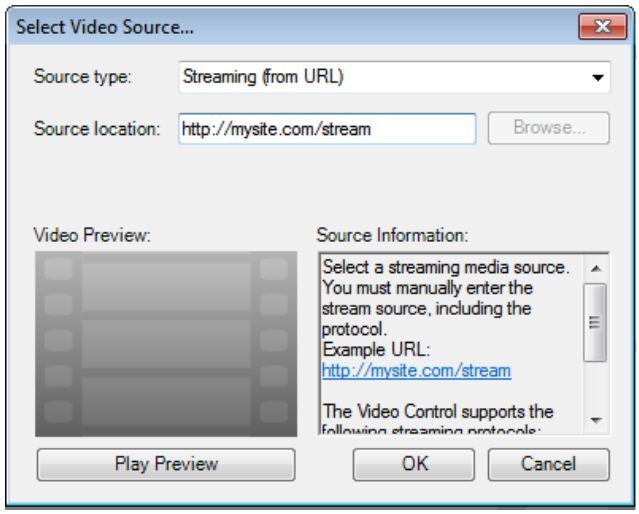


Figure 5.7 - The Saved Video From File

For both options you can use Play Preview to see how your video will look like in Runtime mode and check if you filled right path for your video source.

5.2.8 Third-Party Controls

In addition to the ICONICS controls, you can use any other control that someone else creates if these controls are in the following categories:

- Windows Presentation Foundation (WPF) Controls
- Windows Forms .Net Controls
- Web Browser Controls
- 64-bit ActiveX

To insert a third-party control, click on the Toolbox drop-down list and select the type of control you want to use. Notice that this option is available only in the Toolbox.

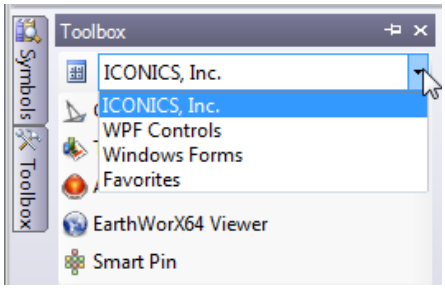


Figure 5.8 - Select Between Different Control Types

5.3 Search and Replace

While creating a display in GraphWorX64, or after creating one, you might want to change some text fields or the OPC tag connected to an object. This can easily be done by using the Replace option; you can find it in the Home tab inside the Text section.

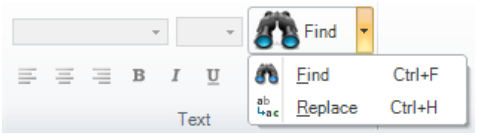


Figure 5.9 - Find/Replace

With this option, you can replace text, data sources, names of objects, titles, descriptions, and more. You can also replace text in the current display, or in multiple displays. If you find a typing error or you need to change the name of a certain object throughout many displays, this handy tool will do the trick. You can even use wildcard characters in this dialog.

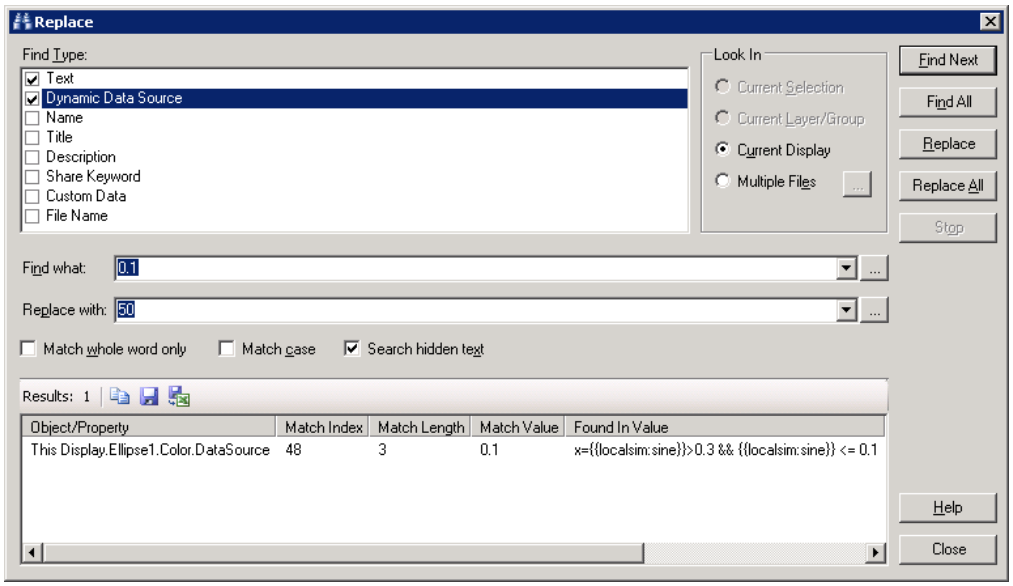
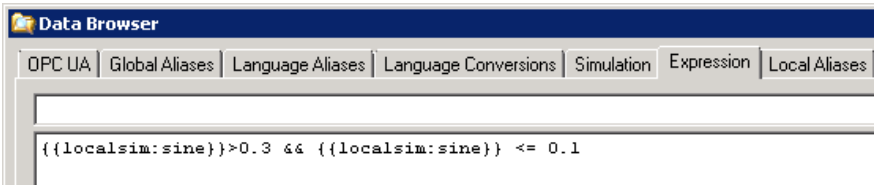


Figure 5.10 - Replace Dialog

Figure 5.11 shows an expression before and after the replacement illustrated in Figure 5.10. The expression originally evaluated values greater than 0.3 and less than 0.1. After a few replacements, the expression evaluates values greater than 30 and less than 50.

Before:



After:

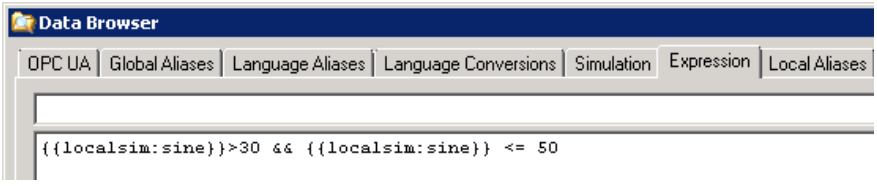


Figure 5.11 - Expression Before (Top) and After (Bottom) Replacement

5.4 Arrange Controls and Objects

When you have many objects and controls in a display, you'll probably want to arrange them in an orderly way that's visually appealing. There are tools to help you arrange objects in the 2D space and in the 3D space. And keep in mind that controls like the Alarm Viewer or the Trend Viewer shown in Figure 5.12 are objects, too. They, too, can be arranged.

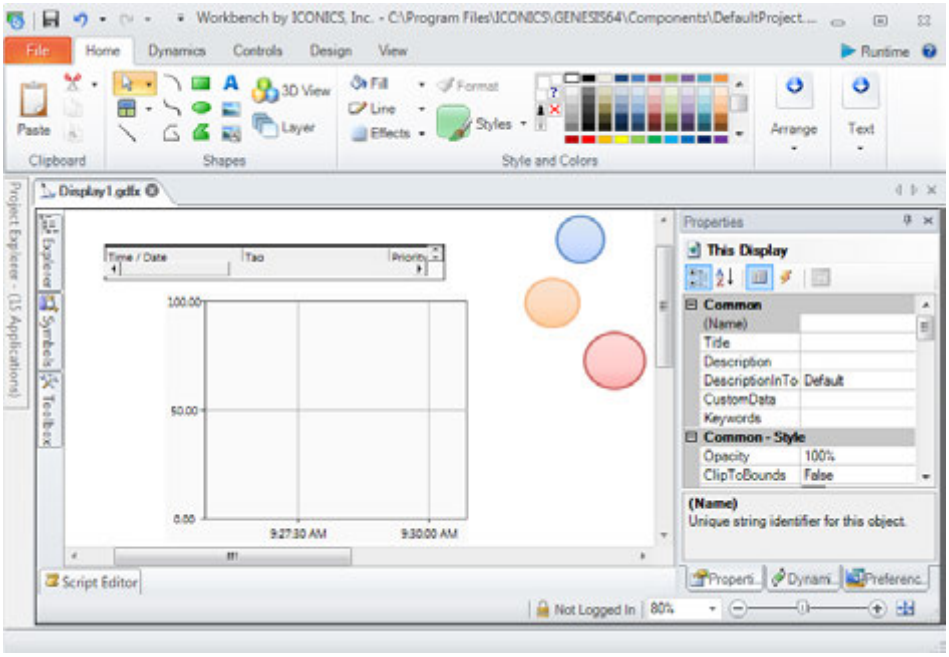


Figure 5.12 - A Display with Objects and Controls

To arrange objects in the active layer, simply select the objects you want to arrange, then select how you want to align them. Note that the last object you select becomes the *reference object*, as shown in Figure 5.13. All other objects are arranged in relation to the reference object.

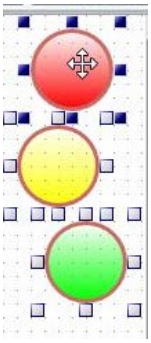


Figure 5.13 - The Top-Most Circle is the Reference Object

5.4.1 Aligning Objects

In the Arrange section of the Home tab, you can choose the alignment you want to use. When using options in the Arrange section, you can actually *change* some attributes, such as size, in relation to another object, and align objects to each other.

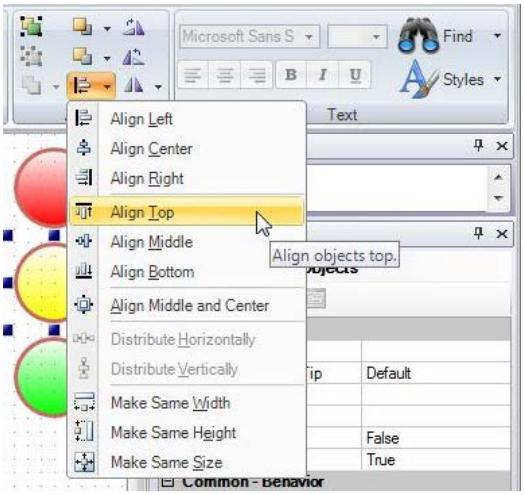


Figure 5.14 - Align Options

When working in a 3D workspace, you can also align objects to each other, and even stack them along an axis you want.

5.4.2 Changing an Object's Z-Order

As you add objects to your display, those objects are added in order from back to front – this is their Z-order. The object drawn last is the top-most object, both in the content area and in the Object Explorer, as shown in Figure 5.15 where the Rectangle object is on the bottom (or in back of the other objects).

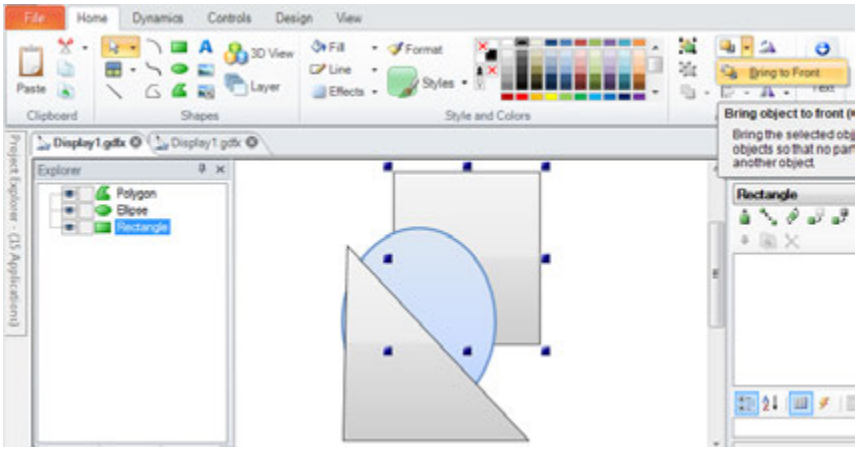


Figure 5.15 - Z-Order of Objects in a Layer

You can change the Z-order of the objects by using options in the Arrange section of the Home tab. In the following example, the selected rectangle is being brought to the front from the back. A different way of changing the Z-order would be to cut and paste the Rectangle in the Object Explorer, or drag and drop the Rectangle up to a higher spot in the Object Explorer. Instead of using the Home ribbon, you can right-click on the object and select the appropriate function from the pop-up menu, as shown in Figure 5.16.

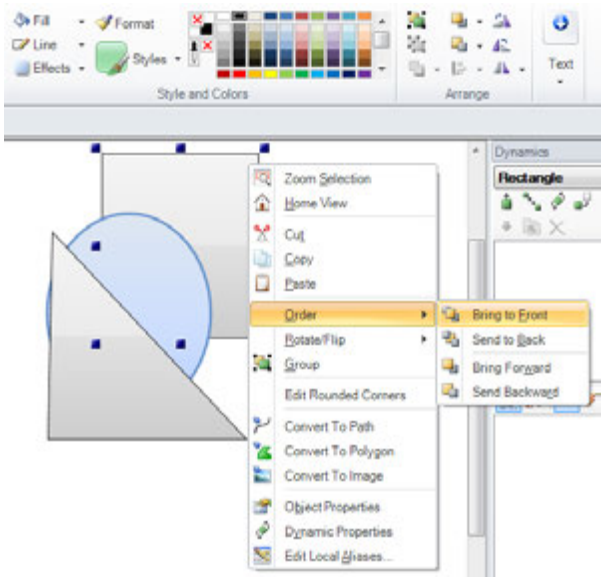


Figure 5.16 - Arrange Objects with Right-click Options

5.5 Views

Once you have completed a GraphWorX64 display, you may find that it is too large or complex, or difficult to navigate easily in runtime. Maybe a display shows a worldwide map created inside an EarthWorX64 Viewer that runtime users need to be able to move around on easily; or a display shows a large campus with detail that extends inside buildings; or a dashboard has detailed areas that runtime users want to be able to zoom easily.

To help runtime users navigate through displays, you can create *views*, which are pre-set scenes or snapshots inside a GraphWorX64 display. In configuration mode, you can navigate to desired destinations within the display and create a view for each destination. As a result, runtime users can access these views, giving them easy access to these areas of detail.

To create views in 2D, you go to the View ribbon and use the options in the Zoom section, shown in Figure 5.17. To create views in a 3D viewport, you go to the Home ribbon and use options in the View section. In this section we'll focus on 2D views.

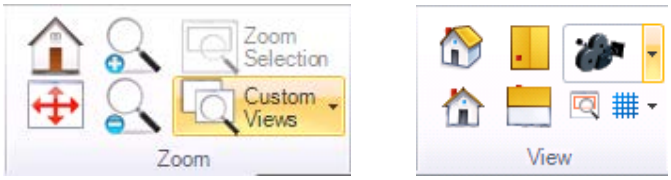


Figure 5.17 – Tools for Creating Views in 2D (Left) and 3D (Right)

To create a 2D view, you must position the GraphWorX64 display so that your work surface shows exactly what you want the view to look like. This may involve zooming in or out, or scrolling up, down, left, or right to position the display. From the View ribbon, click the "Custom Views" button, then select "Add Current View" from the drop-down menu. The **Custom Views** dialog box displays, as shown in Figure 5.18. You can give the view a name that runtime users can easily recognize in the dialog.

During runtime, users can right-click the 2D display to see the list of views and then click on a view to go directly to it. Alternatively, you can create buttons or other objects that use a Pick action, and add them to the display for easy access at runtime.

These same pre-set views that you create for runtime users are also available to you during configuration. In configuration mode, you can see a 2D display's views by right-clicking the display and selecting Custom Views, or by clicking the Custom Views button on the View ribbon.

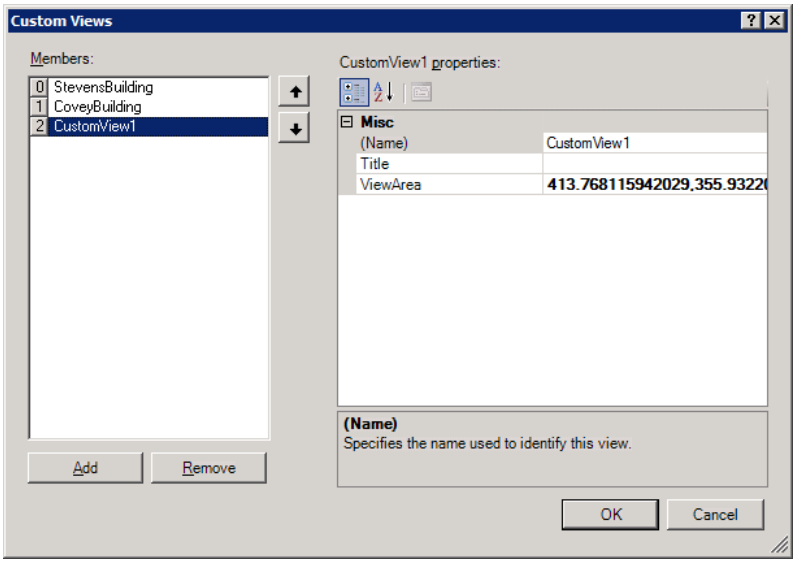


Figure 5.18 - Naming a View

5.6 The GraphWorX64 Runtime Ribbon and Menus

Runtime mode is an environment that populates the GraphWorX64 display objects in memory, draws them on your screen, and updates them with data in accordance with the properties you set in Configuration mode. There is also error tracking in Runtime mode, which can help you troubleshoot your display if needed, and a separate collision detection test for 3D displays.

You can preview Runtime mode using Runtime Preview, which creates and draws the objects but which uses sample data in place of real or historical data drawn from data sources. Or you can enter Runtime mode by clicking the Runtime button above the Configuration toolbar, or pressing Ctrl+M.

In Runtime mode, GraphWorX64 provides runtime menus and runtime ribbons. Using these, runtime users can:

- Magnify areas of the display by zooming in and out
- Access predefined views
- Load a file based on a URL or open a file using the file system
- Browse forward or back in the browse history
- Print a display
- Display interface features such as: the menu, scroll bars, status bar, different sized icons
- Open the Diagnostics utility
- Login or logout of a display
- Open the Carousel to view a sequence of displays
- Go to pre-set views (refer to Animating 3D Views at Runtime and Creating 2D Views for Runtime Users)
- Go to the Card Flow view to preview previously-opened displays

During GraphWorX64 runtime, the **File**, **View**, and **Tools menus** are visible if you choose to show them. You must hide the runtime ribbons in order to gain access to the menus.

> **Note:** The menus are not available if the display has an EarthWorX map in it.

The **ribbon** may be easier to use, though. There are several tabs on the Runtime ribbon. A GraphWorX64 display shows its own GraphWorX64 runtime ribbon, but if the display has Viewer controls for EarthWorX64, TrendWorX64, or AlarmWorX64, *additional* ribbons may also be available. The GraphWorX64 runtime ribbon looks something like Figure 5.19.



Figure 5.19 – GraphWorX64 Runtime Ribbon

To show the runtime ribbons:

- Right-click the display to show the pop-up menu and then select Show Ribbons. This shows the GraphWorX64 runtime ribbon and, if there is an EarthWorX map in the display, the EarthWorX64 ribbon.
- To show the TrendWorX64 ribbon, click on a trend chart.
- To show the AlarmWorX64 ribbon, click on an alarm or a chart of alarms.

Lab: Layers and Dynamics L5

5.1. Lab Overview

Now that we have created a Welcome page, icTonics would like everyone to see this page first. Personnel with proper credentials should be able navigate to other pages as well. Furthermore, they should have the ability to look at detailed trend information for the mixer tank. We will create a popup display that exhibits such information.

Estimated time to complete this lab: 20 minutes

5.1.1. Task Checklist

In this lab, you will complete the following tasks:

- Create a secured layer
- Add a Scale Control
- Add a SetPoint data entry field
- Add a Draining hide dynamic

5.1.2. Support Files

To complete this lab, you will need the following file(s):

- icTonicsTemplate.tdfx
- icTonicsHome.gdfx

5.2. Adding a Layer

We will add a secured layer in this section so that once a user with the proper credentials logs in, the user can navigate to other displays from the home display. The layer you will add in this section will not actually be secure until we incorporate a security configuration in Chapter 14.

1. Open Workbench, if you closed it from the previous lab.
2. Open the icTonicsHome.gdfx display in a new tab for editing. You can do this from within GraphWorX64 by selecting the File menu tab. Then you can select it from the Recent Documents in the menu, or click on "Open" to navigate to it and select it.
3. In the ribbon, go to the Home tab. In the Shapes section, click on the "Layer" button.
4. In the Object Explorer, you should now see a new object called "Layer1". Rename this layer "SecuredLayer".
5. Go to the Properties explorer of the SecuredLayer, and find the HideWhenSecurityDenied property in the Declutter section and change this property value to "True".
6. Go to the Dynamics ribbon. In the Buttons section, click on the "Button"  icon and then click on the GraphWorX64 design surface to add a button to the lower left-hand corner of the display.
7. Leave the button's Command property as "**Load Display**".
8. In the Dynamics explorer, under the **CommandParameters** section, in the **FileName** property field, click the down arrow to show the drop-down list and select the "Browse Files" option.
9. In the **Open File** dialog box browse to C:\Training and select the icTonicsTankMonitor.gdfx file. Click on "Open" to select the file and close the dialog box.
10. Since we want to use relative path, delete the file path in the FileName field so that it only reads "icTonicsTankMonitor.gdfx".
11. Go to the button's Properties explorer and, in the **Content** section, change the Text of the button to "Tank Monitor".

12. In the **Measurements** section, set the button Width to 170 and the Height to 70. Confirm the button is placed in the viewable area in the lower left part of the display.

13. Save the display.

5.3. Add a Scale Control

1. Open the icTonicsTankMonitor display in configuration mode.

2. In the GraphWorX64 ribbon go to the **Controls** tab and select **Scale Control**. The cursor will turn to cross hairs. While holding the left mouse key down, place the scale control to the right of the tank. Go to the Properties explorer of the scale control and locate the **ScaleType** property in the **Scale** section. Set it to **Radial**. Your display should now look like Figure L5.1.

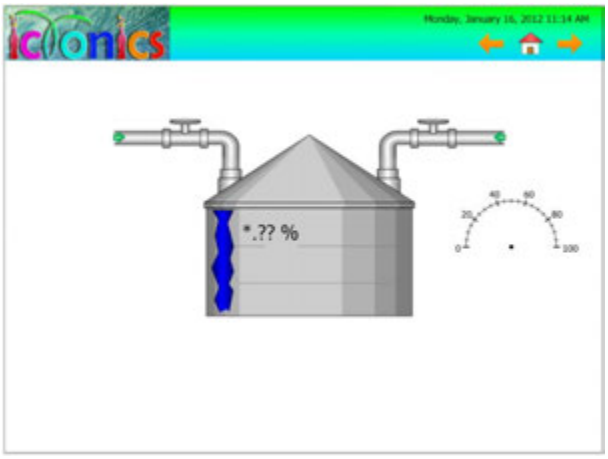


Figure L5.1 - Scale Control

3. Now we will add a pointer to this gauge. Go to the Home tab in the ribbon. Add a **Rectangle** in the gauge and resize it to make it look like a pointer.

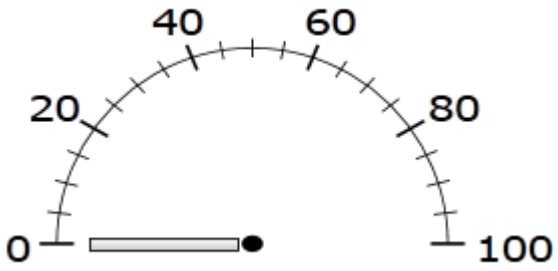


Figure L5.2 - Gauge with hand

4. Add a **Rotation** dynamic to the pointer. Click on the ellipse in the DataSource field to launch the Data Browser. Define the DataSource as **@ICONICS.IconicsOPCUAServer.V5\icTonics.Tanks.USA.Foxboro.Tank1.TankLevel.Value.**

5. In the **Data** section, set the **LowLimit** to 0 and the **HighLimit** to 100. In the **Dynamic Behavior** section, change the **StartAngle** to 0 and the **EndAngle** to 180.

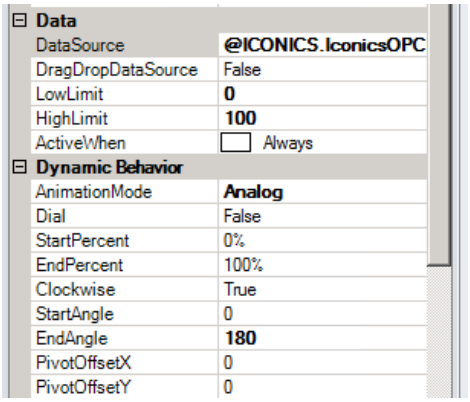


Figure L5.3 - Rotation Dynamic Settings

6. With the pointer selected, click on the "**Edit Dynamic On-Screen**" link at the bottom of the Dynamics explorer. Move the red pointer to the center pivot point of the gauge.

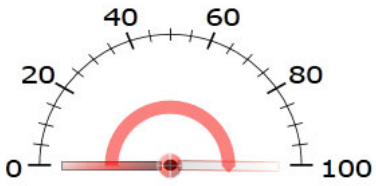


Figure L5.4 - Rotate Path

7. In the Object Explorer Tree on the left-hand side, hold the "Ctrl" key and select Rectangle1 and Scale Control, in the Arrange Group under the Home Tab, then click "Group Objects".

5.4. Add a Radio Button

1. Now we'll add a Radio button to toggle the churning blade on and off. From the Dynamic tab in the Buttons section, select a Radio button.

2. Place the Radio button above the tank, and center it on the two pipes. Name it "Churning Blade On" in the Content section of the properties explorer.

3. In the Dynamics explorer, under the **Data** section, set the **DataSource** to

   @ICONICS.IconicsOPCUAServer.V5\icTonics.Tanks.USA.Foxboro.Tank1.ChurningBladeStatus.Value.

4. Set the **Command** to **Toggle Value** and set **ValueOne** to 1 and **ValueTwo** to 0.

5.5. Add a State Field

1. From the Dynamics tab in the Text Objects section, select the Process Point. Place the process point below the tank.

2. From the Data Browser select:

   @ICONICS.IconicsOPCUAServer.V5\icTonics.Tanks.USA.Foxboro.Tank1.ChurningBladeStatus.Value

3. In the **Format** section, type "Churning blade is" in the **PrefixLabel** field. (Make sure you turn on the "Advanced Mode" in the Application Mode under View Tab).

4. In the **Measurements** section, set the Width to 320 and Height to 40.

5. In the **StateField** section set **StateField** to "True". Click on the ellipsis in the States field. In the "**Edit Process Point States**" dialog box, click Add. Under the **Data** section set the **LowLimit** to 0 and the **State Text** to "Off". Click on "OK".

6. Click Add again, under the **Data** section set the **LowLimit** to 1 and the **State Text** to "On". Click on "OK".

7. Your finished display should be similar to the Figure L5.5.

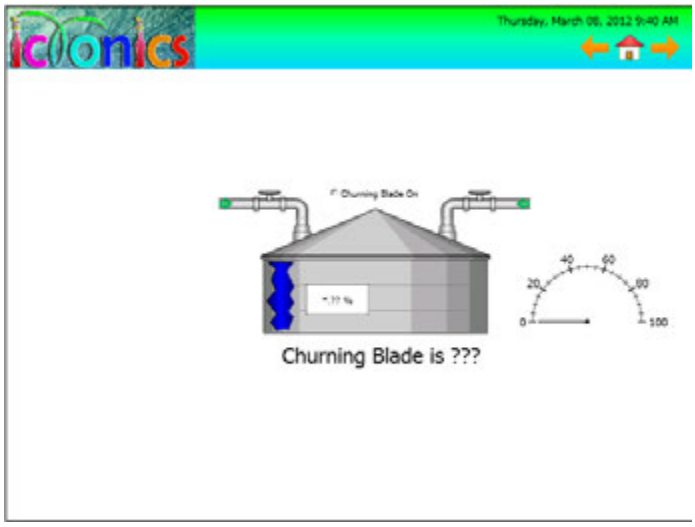


Figure L5.5 - icTonicsTankMonitor

5.6. Add a SetPoint Data Entry Field

8. From the Dynamics tab in the Text Objects section, select the **Data Entry**. Place the data entry above the scale control.

9. From the Data Browser select:

   @ICONICS.IconicsOPCUAServer.V5\icTonics.Tanks.USA.Foxboro.Tank1.LevelSetPoint.Value

10. To keep the look consistent, select the "**Churning blade is ???**" field.

11. Perform a **Ctrl-D** which will make a copy the selected object.

12. By selecting the copied field with the left mouse key, drag and drop the field above the SetPoint data entry field.

13. Notice the handles of the copied object are red. This indicates there is a dynamic associated with this object. We're going to delete the dynamic.

14. Click on the **Dynamics Tab** in the dynamics explorer.

15. Select the **Process Point** dynamic at the top of the explorer and hit the **Delete** key on the keyboard. This will remove the dynamic property.

16. Notice the handles turn blue. This indicates the dynamic property has been removed.

17. Select the **Properties Tab** in the properties explorer.

18. In the Content section change the **Text** field to **Level SetPoint**.

5.7. Add a Text Object with a "Hide" Dynamic

1. Next we will add text object with an associated "Hide" dynamic.

2. From the GraphicWorX ribbon click on the **Home Tab**.

3. Under the **Shapes** section select the **Text** (Draw Label) tool. This is represented by a capital A.

4. By holding the left mouse key, **draw and size** the text object in the lower right hand corner of the tank. Size this object however you see fit. We will be adding a Hide dynamic so this text will only be visible when the tank is draining.

5. Click on the **Properties Tab**. From within the **Content Section** change the **Text** to "**Draining**".

Note: Verify from the View menu on the Ribbon in the Misc section that the Application Mode is set to Advanced Mode.

6. Click on the **Dynamics Tab** and select the **Hide** dynamic from the top of the Dynamics explorer.

7. From the Data Browser set the **DataSource** to:

   @ICONICS.IconicsOPCUAServer.V5\icTonics.Tanks.USA.Foxboro.Tank1.PumpStatus.Value

8. In the **Data** section set the **HideWhen** field to **data == 0**.

9. Your final configuration should resemble the figure below.

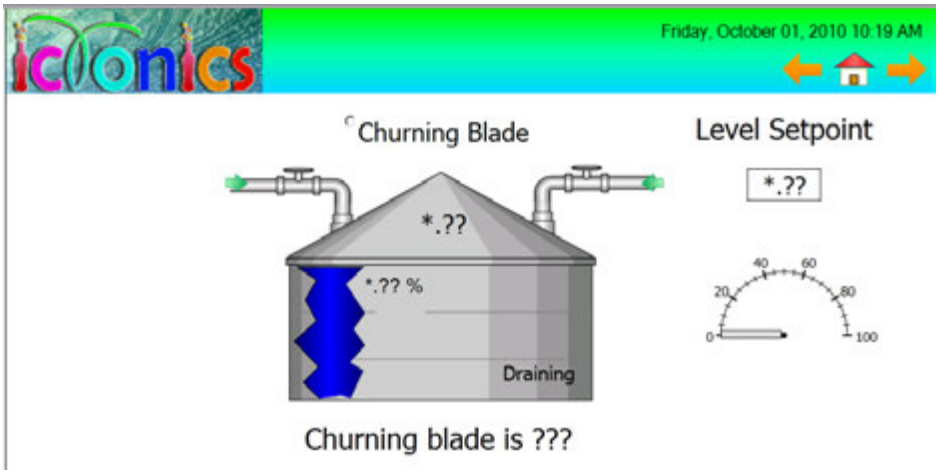


Figure L5.6 - icTonicsTankMonitor with Level SetPoint and Draining Dynamics

Review: GraphWorX64 Tools R5

5.1 Objectives Review

You should now know and understand the following topics:

- Adding controls
- Arranging objects in a display
- Working with layers
- Working with the Search and Replace feature
- Adding views
- Web-publishing your displays
- Using the Runtime ribbon and menus

5.2 Summary

You can use controls to add Viewers to a GraphWorX64 display for working in EarthWorX64 maps, TrendWorX64 trend charts, the AssetWorX Navigator, AlarmWorX64 grids and charts, and even other GraphWorX64 displays. You can also use GraphWorX64 as the container application for WPF (Windows Presentation Foundation) or Windows Forms .NET controls, or even third-party controls.

Smart Pins and PushPins are powerful tools for enhancing the runtime visibility of assets in a crowded or high-level display while putting drill-down capabilities into the hands of your runtime operators. The Scale and Pipe controls make it easier for you to create objects that have face dials and pipe schemes. Menus can be easily generated using the Pivot Control containing any of the other viewers in a GraphWorX64 canvas.

In case you need to modify your project, you may benefit from the Search and Replace feature, which is not limited to one-file jobs, but can search multiple files at once.

The ease with which you can align objects and change their Z-order speeds up your development. The use of layers helps you distribute information into levels in a 2D display. You can also take advantage of hiding a layer under specific zoom or security conditions to rapidly develop your application. Views, which are available in 2D and 3D, are an ideal tool for helping runtime users navigate large, complex displays; but they're also helpful in configuration mode.

Use the Wizard to web-publish your work, eliminating the need to write any HTML code yourself and preventing others from seeing your efforts through a thin client. Runtime ribbons let operators zoom in and out, access views, browse through display history, hide objects in the display, log in, open a Carousel of displays, and more.

5.3 Terms

Alignment	Decluttering	GENESIS64 Controls
Layers	Pipe Control	Pivot Control
PushPins	Reference Object	Runtime Ribbon
Scale Control	Smart Pins	Views
Web Publishing Wizard	WPF	

5.4 Test Your Knowledge

In this chapter, we discussed how to use GraphWorX64 to edit displays in the 2D workspace. Before continuing to the next chapter, make sure you can answer the questions below:

1. True or False: A GraphWorX64 display can hold one or more viewers (for example, an AlarmWorX64 Viewer or TrendWorX64 Viewer).
2. True or False: The Object Explorer shows the This Display object.
3. True or False: It is a requirement that you create a layer for arranging objects or controls in a display.
4. True or False: A layer can be dynamically hidden and shown.
5. True or False: Multiple layers inside the same display can have the same name.
6. True or False: A sub-layer can be moved inside the hierarchy of layers in the Project Explorer by using drag-and-drop.
7. True or False: An object and a control can be aligned to each other in the same layer.
8. True or False: Objects can be put on top of each other using the Align feature in the Arrange pane.
9. True or False: Search-and-replace applies to text inside one display only.

10. True or False: GENESIS64 uses ActiveX for making displays web enabled.

11. Under which of the following circumstances will Views be available to users?

    a. In configuration mode only
    b. In runtime mode only
    c. In both configuration and runtime modes
    d. In neither configuration and runtime modes

12. True or False: Strategic arrangement of SmartIcons in a Smart Pin lets runtime users easily monitor the status of multiple data sources, all at the same time.

13. True or False: For a toolbar to appear at runtime, you must configure a toolbar into a template that you use for creating all of your GraphWorX64 displays.

14. What control could you use to configure a 2D object easily to move along a path in a 2D display?

    a. The Pipe control
    b. A third-party tool
    c. The Pipe primitive
    d. The Move dynamic
    e. All of the above
    f. None of the above

AlarmWorX64 Server 6

AlarmWorX64 is a distributed enterprise-wide alarm and events management system. Available in the standard GENESIS64 suite of applications, or as a stand-alone Open Series component, AlarmWorX64 offers the tools you need to deliver real-time alarm information throughout your system. AlarmWorX64 is a family of modular alarming products, including the Alarm Server, the Alarm Logger, the Alarm Viewer Control, and the Multimedia Server.

In the next few chapters, we will discuss the major parts of AlarmWorX64. Before we can discuss any other components in AlarmWorX64, we must first understand and configure an alarm server. This will be the source from which we obtain alarm and event information. After completing this chapter, you should be familiar with the following:

- AlarmWorX64 Server and how to launch it
- The different types of alarms
- Configuring alarm tags individually or using the Tag Import Wizard
- Configuring and using an alarm tag template
- Grouping alarm tags by area

6.1. AlarmWorX64 Server

The AlarmWorX64 Server receives field data from your OPC-compliant Data Access servers and performs alarm detection and reporting on that data. You use the AlarmWorX64 Server interface to create and maintain a database configuration of alarm tags.



Figure 6.1 - Overview of AlarmWorX64

All alarm detection and reporting conforms to OPC Alarm and Events standards. Event notifications generated by the AlarmWorX64 Server are sent to OPC UA clients (such as, AlarmWorX64 Viewer and AlarmWorX64 Logger) that subscribe to it.

AlarmWorX64 Server can be run as a service. The server reads its configuration information from a Microsoft SQL Server database file (which by default is AWXSVRCONF.MDF). During runtime, the server polls the configuration database for changes, allowing on-the-fly configuration change without stopping the server.

To start AlarmWorX64 Server, go to **Start → All Programs → ICONICS GENESIS64 → AlarmWorX64 → AlarmWorX64 Server Configurator**. The AlarmWorX64 Server Configurator opens in the Workbench. You can also launch the configurator in the Workbench from the "AlarmWorX64 Server" program icon at the bottom of the Project Explorer as shown in Figure 6.2.

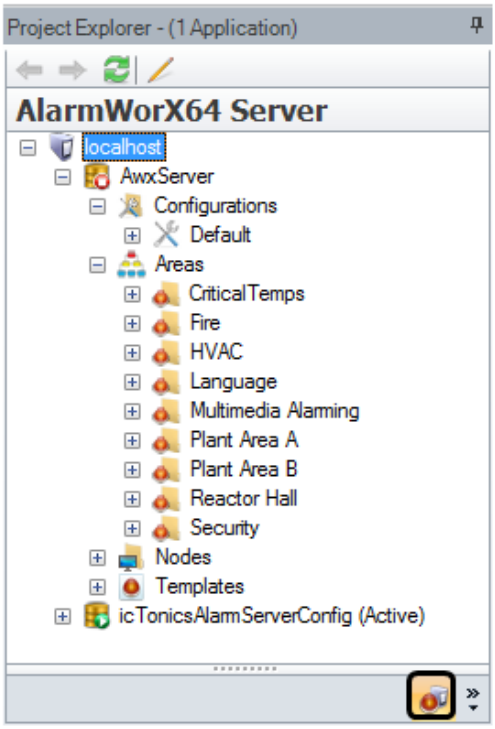


Figure 6.2 - AlarmWorX64 Server Configurator

6.2. Alarm Tags

An alarm tag is an OPC tag to be monitored along with the value ranges at which alarms should trigger. In a typical scenario, the AlarmWorX64 Server receives a value periodically for each OPC tag and determines if the incoming OPC value is in an alarm condition. If it is, the AlarmWorX64 server generates an alarm. The alarm client will then be able to display the condition so that nearby operators can rectify the situation.

For example, you might want to create an alarm tag for an OPC tag for the speed of a cooling fan so that an alarm is triggered when the cooling fan's speed rises above or falls below an acceptable range. As a result, an operator can investigate or make adjustments to rectify the problem before damage can occur to goods or to the equipment. You might also want an alarm to be triggered when a secured door is opened or when a pressure gauge reading drops rapidly.

For each alarm that you configure, you may require an operator to acknowledge the alarm. You can even require an operator to acknowledge an alarm that has returned to its normal state.

Remember that you do not have to limit your alarms to adverse or critical conditions. For example, you can configure an alarm that provides instructions for a manual step at a key point in a process, or you can configure a series of alarms on a piece of equipment to provide automated training for new operators.

AlarmWorX64 provides six types of alarms:

- A **Limit Alarm** occurs when the incoming reading exceeds a ceiling or floor limit that you configure.
- A **Digital alarm** occurs when the value of the OPC tag or expression equals the Alarm State Value. Since this is a digital alarm, the alarm state values can only be 1 or 0; the other value is the normal state.
- A **Deviation alarm** occurs when the absolute value of the difference between two OPC tags exceeds a ceiling or floor limit that you configure.
- A **Rate of Change alarm** occurs when the amount of change between the last two readings exceeds a limit that you configure.
- A **Rate Limit alarm** occurs when the amount of change exceeds the high or low limits specified in the alarm tag. A **Trigger Limit alarm** reports when the tag's value changes, and will not reset until after a runtime operator acknowledges it. With this alarm, there is no normal condition; any change in the value triggers the alarm. This type of alarm is useful for a tag whose value rarely changes.

These types of alarms are available for each alarm tag that you create. You can also configure any or all of these alarms in one alarm tag. Each type comes with its own conditions that you can choose from in order to configure it for an OPC tag. Each type has its own distinct configuration.

6.3. Configuring Alarm Tags

You can configure any number of alarms for an OPC tag, and in this section we will discuss how you can configure them.

> **NOTE:** Each alarm that you create resides in a configuration database. Therefore, you must create a configuration database for the alarms before you can add alarms. You must also make sure that the configuration database is active before running the alarm server.

In AlarmWorX64, you can configure alarms one at a time, or you can import several OPC tags and generate alarm tags for them *en masse* using an import wizard. In either case, you can create alarm tags from scratch or you can create templates in advance which you can use as the basis for creating new alarm tags.

6.3.1. Creating Alarm Tags One at a Time

Let's start by looking at how you create an alarm tag. At the most basic level, you can create alarms one at a time. To do so, in AlarmWorX64 Server right-click on your database configuration, then select **Tag** from the menu. This opens the alarm tag configuration form with the **General** tab on top, as shown in Figure 6.3.

Give the alarm a **name** and, in the **OPC Input** field, specify the tag whose value you want to use for the alarm. For example, if the alarm is monitoring water pressure, you want to use pressure readings from the pressure gauge. You can click on the ellipsis button to the right of the OPC Input field; this opens the Data Browser where you can navigate to and select the appropriate tag.

You can also specify a **Template** to be used for prefilling some or all of the tag's fields. Templates are a useful tool in both creating and maintaining your alarm tags. We will talk about templates later in this chapter. But for now, it is important that you know that you can change the template to be used for a tag at any time. If you are not sure about the content of a template, clicking the **Jump to Template** button will open the template for you so you can review it before using it for a tag.

The Name, OPC Input, and Template fields always appear regardless the tab you are on in an alarm tag configuration. There are five different tabs in the alarm tag configuration: General, Alarms Limit/Digital, Alarms Deviation/Rate of Change, Alarms Rate Limit/Trigger Limit, and Related Values. We are going start by looking at the General and Related Values tabs. Then we'll look at the six types of alarms that are available to you on the remaining three tabs.

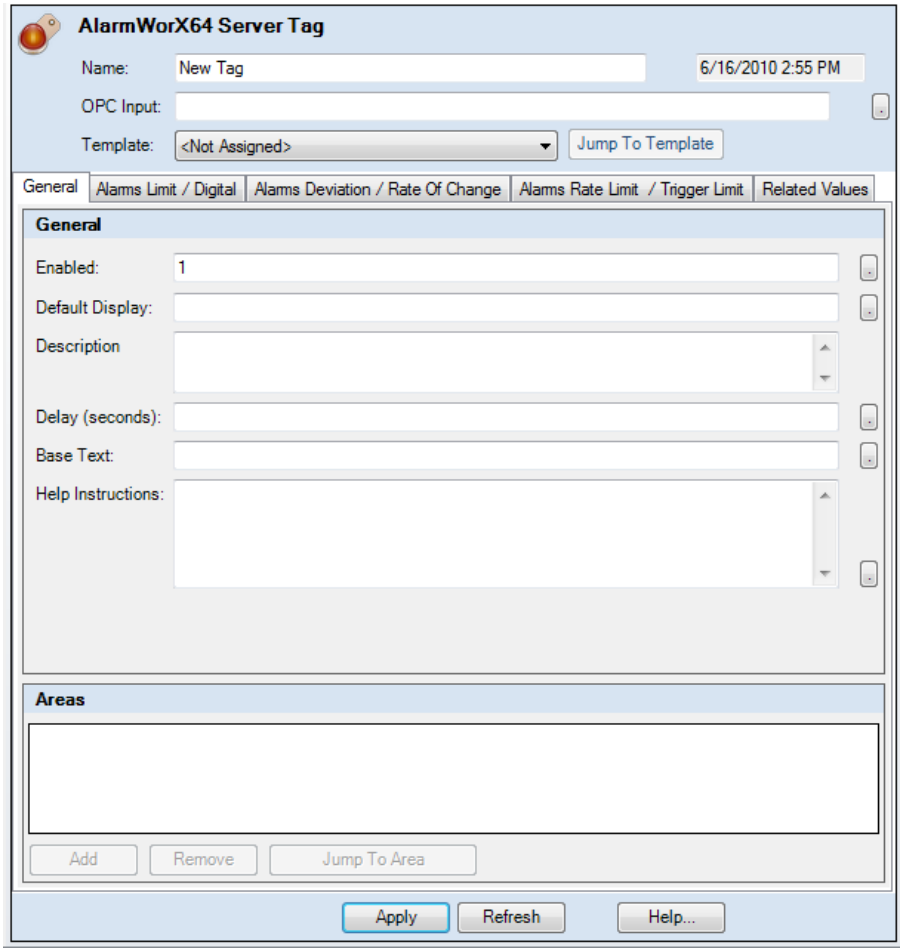


Figure 6.3 - Alarm Tag Configuration

6.3.1.1. General Tab

On the General tab, shown in Figure 6.3, you can configure some basic information for the alarm. Most of the fields in this tab are optional, except for the Enabled field, which is set to true by default. You have the following fields in the General tab:

- **Enabled**: By default, an alarm is enabled when it is initially created (a 1 in the Enabled field means that the alarm is enabled). However, you may prefer to enable and disable the alarm based on a condition. For example, you may not want to monitor water pressure when the valve is shut off because the drop in pressure would trigger the alarm. In this case, you can specify the tag for the valve's reading in the Enabled field such that when the valve is open (1) the alarm is enabled, but when the valve is closed (0) the alarm is disabled.
- **Default Display**: Once you have set up an AlarmWorX64 Viewer or GraphWorX64 display where the alarm has been added so that it can be monitored by an operator, you can identify that display as the **Default Display** for the alarm. Click the ellipsis button to navigate to the file and select it. Note that the alarm can be added to other displays, too.
- **Description**: Add a description for the alarm.
- **Delay (seconds)**: When an alarm condition exists, the server generates the alarm immediately. You can enter a **delay** that postpones the alarm's generation for a few seconds to wait for conditions to return to normal. If the alarm condition still exists after the delay, the server generates the alarm. You may not want to use a delay for every alarm, but it is helpful for offsetting readings that fluctuate wildly or sensors that sometimes return faulty readings. You can also use it with equipment that typically returns to normal quickly.
- **Base Text**: For each alarm condition, you may want to add a message that indicates the alarm state. In the Base Text field, enter information that is required for each of the states; the text in this field is added before the message text for each alarm condition.
- **Help Instructions**: Additional aid for your users comes from any text you enter in the **Help Instruction** field where you can provide the steps to resolve or acknowledge the alarm; you can enter information directly into this field or paste it from the Clipboard. Information entered in this field can display as a tooltip in the alarm viewer when users select and mouse-over the alarm.
- **Areas**: Once you have saved a tag, you can use the bottom of the General tab to add the tag to areas quickly and easily. We'll talk about areas later in this chapter.

6.3.1.2. Related Values Tab

The **Related Values** tab, shown in Figure 6.4, identifies as many as twenty additional values you want to capture when an alarm condition occurs. Related values can be static text strings, OPC tags, or expressions. Capturing additional readings can help you diagnose a problem later.

For example, following an alarm that reported a drop in water pressure on an outtake line, you might want to review the tank's water level and the intake valve. As long as you have added the OPC tags for the water level and intake valve as related values for the alarm, you may log the data and review it at a later day.

Now that we have reviewed the general options available for any alarm tag that you configure, let us look at the details of each of the six types of alarms (Limit, Deviation, Digital, Rate of Change, Rate Limit, and Trigger Limit) you can configure in an alarm tag. Note that you can configure any of the alarms described here, and can even combine them into one alarm tag configuration.

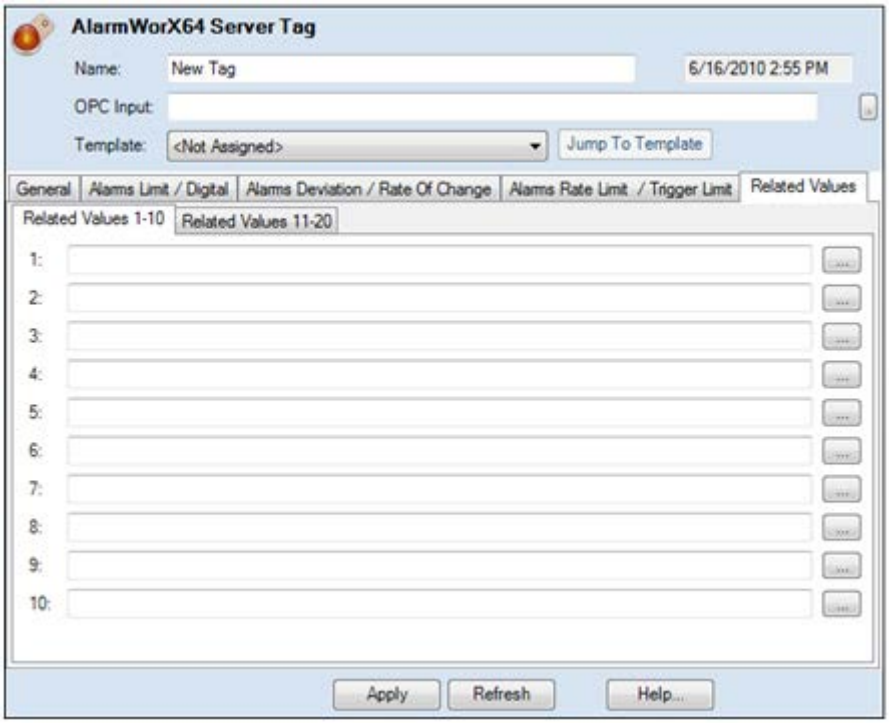


Figure 6.4 - Related Values tab

6.3.2. Limit Alarm

If you go to the alarm tag's Alarms Limit/Digital tab, in the **Limit** pane, shown in Figure 6.5, you can configure an analog **Limit alarm** for the OPC tag. A Limit alarm reports an incoming reading that exceeds a high or low threshold that you have configured. Use this type of alarm for OPC tags whose readings have static limits that you do not want exceeded such as temperature, counters, pressure gauges, etc.

By default, this alarm reads the values of the OPC tag you specified in the OPC Input field. For the purpose of setting an alarm limit, you can override this tag by adding a different tag to the **OPC Override Input** field in the Limit pane.

Let's skip the Deadband value for now, and take a look at the **Value** fields in which you provide the acceptable limits of the tag's readings. You can set one, two, three, or four limits, each with its own warning or alert text, severity level, and acknowledgement requirements. The values that you enter are readings from the tag in the **OPC Override Input** field, if any, or the **OPC Input** field at the top of the configurator.

You can add **Lo** (low value), **LoLo** (very low value), **Hi** (high value), or **HiHi** (very high value) limits. A tag value that is at or below the LoLo value is in an alarm condition; as a result, the alarm generates with the message and severity configured for it, and an operator's acknowledgement of the alarm is required if the **Req. Ack.** value is **1**. Values that are at or below the Lo value do the same, as do values that are at or above the Hi and HiHi values.

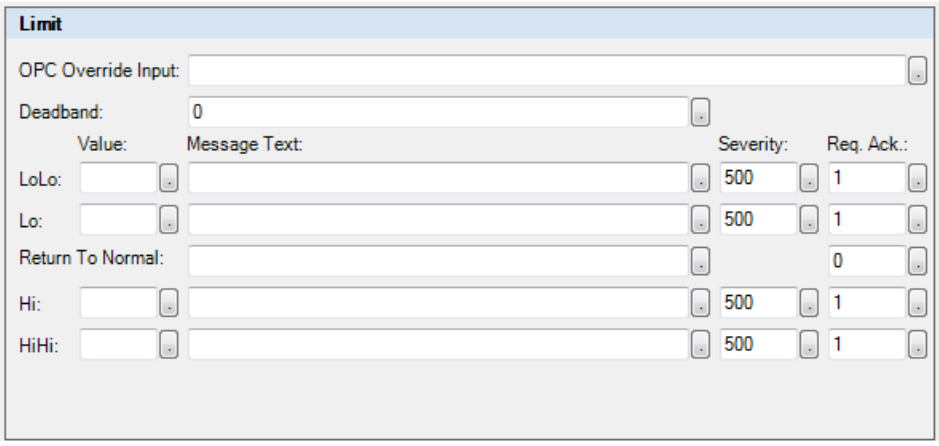


Figure 6.5 - Limit Alarm Configuration - Alarms Limit/Digital Tab

There's a lot of flexibility in your ability to configure any of these limits. For example, you can use the Lo limit as a warning level and LoLo as an alert. You can tie **Message Text** directly to each value that creates an alarm condition. The **Severity** is the OPC-defined value for alarm priority; the OPC severity range is 0 (lowest) to 1000 (highest).

On the **Return to Normal** line, you can also provide the message text that appears when the tag's value returns to normal from an alarm condition and you can require the operator's acknowledgement of a return-to-normal condition. All values between the Lo and Hi limits, exclusively, are assumed to define the normal range.

Let us go back to the deadband value. By providing a value in the **Deadband** field, you identify a buffer zone around alarm limit values that you have entered. Applied to converted analog values, the deadband value is germane to borderline cases in which readings *seesaw* or *blink* into and out of an alarm condition. The alarm server applies the deadband to each limit value; it ignores any reading inside the deadband limits and will not generate an alarm. This is helpful especially if the operator is required to acknowledge any change in conditions. You certainly do not want operators distracted with repeatedly having to acknowledge a condition that is toggling in a borderline condition.

Take a look at the example below in which the high limit is 101. A Deadband of 1 prevents an alarm from being reported on readings of 100 and 102. Were the Deadband 0, these readings would report an alarm.

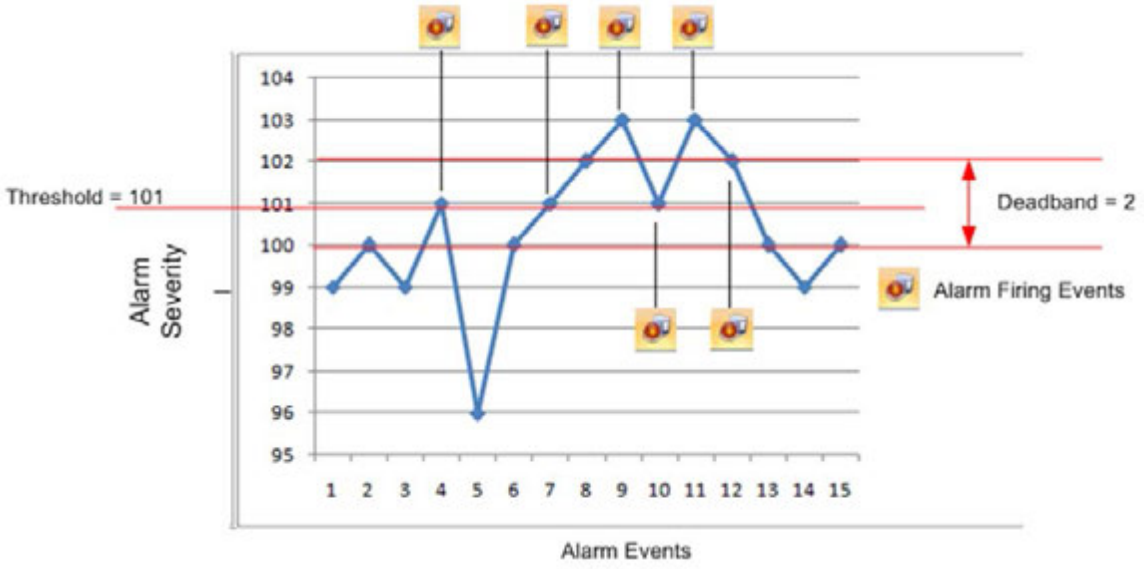


Figure 6.6 - Deadband for Alarms

6.3.3. Deviation Alarm

Similar to the Limit alarm is the **Deviation alarm**, which you configure using the **Deviation** pane on the Alarms Deviation/Rate of Change tab, shown in Figure 6.7. The analog Deviation alarm occurs when the absolute value of the difference between the **OPC Input** tag and the **OPC Input 2** tag falls below or exceeds predefined values. The **OPC Input 2** field in the Deviation pane is required for setting up a Deviation alarm, and is used to calculate the deviation from the OPC Input field. The alarm is triggered if the difference between the two values falls above the high limits or below the low limits specified in the configuration.

All other fields in the **Deviation** pane work in the same way they work for the Limit alarm, but with *one significant exception – the **Value** field*. Instead of entering the *value* of a tag that will set off an alarm, you enter the ***delta*** (the difference) between the two input values that will set off an alarm.

For example, you are creating a Deviation alarm that compares today's ongoing water level readings to yesterday's final water level reading. You have a tag that holds the value of yesterday's final water level, in centimeters; you enter that tag in the **OPC Input 2** field. The **OPC Input** tag at the top of the configurator contains the tag of today's current water level. These two tags are compared.

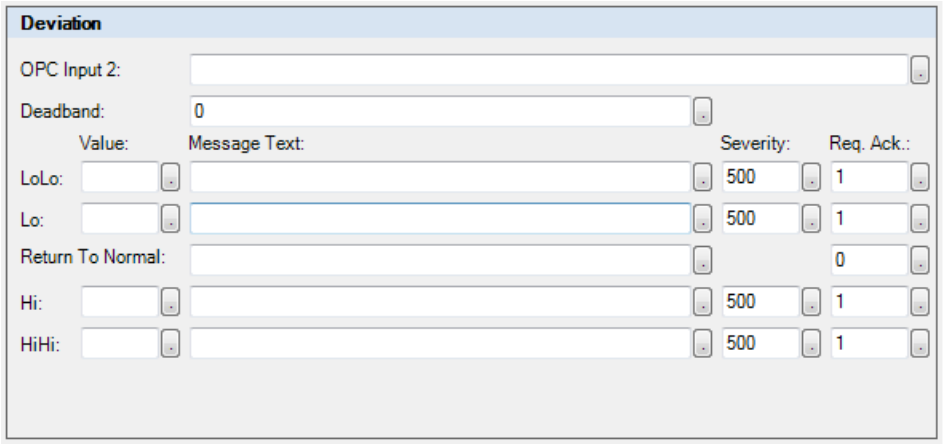


Figure 6.7 - Deviation Alarm Configuration - Alarms Deviation/Rate of Change Tab

Then you enter the delta values at which you want an alarm to generate. For example, you want to be notified if the water level drops or rises more than 2 centimeters compared to yesterday's final value. You would enter a Hi value of 2. You also want to know if it drops or rises 5 centimeters or more. In this case, you would enter a HiHi value of 5. When the difference between the two OPC tags is 2 to 4 it will generate a Hi alarm, and when it is 5 or more it will generate a HiHi alarm.

If yesterday's reading was 50 centimeters, and today the water falls to 48, the server generates a Hi alarm. When the water drops to 45, the alarm server generates the HiHi alarm.

6.3.4. Digital Alarm

If you go to the alarm tag's Alarms Limit/Digital tab, you can configure a **Digital alarm** for the OPC tag in the **Digital** pane, which is shown in Figure 6.8. A Digital alarm occurs when the tag's value goes into a state that you specify.

This alarm reads the values of the OPC tag specified in the **OPC Override Input** field if supplied in the Digital pane, or the OPC Input field at the top of the configuration.

In the **Alarm State Value** field, type the value of the state that triggers the alarm. Since this is a digital alarm, the valid state values for this alarm are limited to 1 and 0. If there are multiple states that can trigger the alarm, you can incorporate those states into an expression. However, the result of this expression must equal 1 or 0. The value that is not the alarming state is considered to be the normal state.

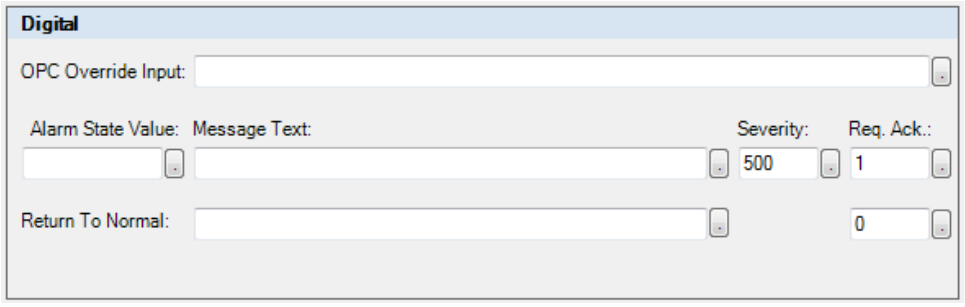


Figure 6.8 - Digital Alarm Configuration - Alarms Limit/Digital Tab

Like all other alarms, you can specify a message and a severity and you can require an operator to acknowledge the alarm. In the **Return to Normal** field, you can specify the message to display when the alarm returns to normal and indicate whether an operator is required to acknowledge a return-to-normal condition.

6.3.5. Rate of Change Alarm

If you go to the alarm tag's Alarms Deviation/Rate Of Change tab, you can configure a **Rate of Change alarm** for the OPC tag in the **Rate of Change** pane, which is shown in Figure 6.9. A Rate of Change alarm occurs if the rate of change in the OPC tag's values is more severe than allowed by the alarm's configuration.

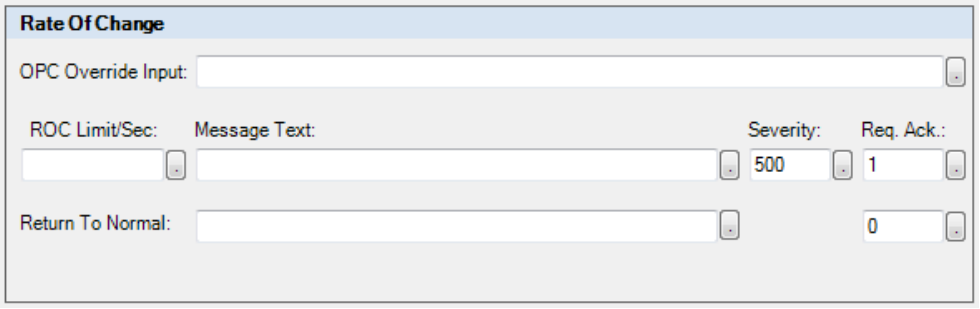


Figure 6.9 - Rate of Change Configuration - Alarms Deviation/Rate of Change Tab

This alarm reads the values of the OPC tag you specified in the **OPC Override Input** field in the Rate of Change pane (if any), or the **OPC Input** field at the top of the configuration.

In the **ROC Limit/Sec** field, specify the rate of change per second that must trigger the alarm. Following our earlier example, you can set up a Rate of Change alarm so that if the water level drops more than 2 centimeters per second, the alarm will report.

The rate of change is calculated by taking the *absolute* value of the last two OPC values and dividing it by the difference of the timestamp from these two values. When entering a rate of change, keep this calculation method in mind.

Like all other alarms, you can provide message text, a severity level, and you can require operator acknowledgement of the alarm. In the **Return to Normal** field, you can specify the message to display when the alarm returns to normal and indicate whether an operator is required to acknowledge a return-to-normal condition.

6.3.6. Rate Limit Alarm

A **Rate Limit alarm** is much like the Rate of Change alarm except that the change is not an absolute number; it is expressed as a positive or negative value. The rate of change is calculated *per second* by taking the increase or decrease in value of the last two OPC values. Using the two values' timestamps, the increase or decrease over the length of a second (that is, the rate) is calculated, then compared to the high and/or low limits defined in the alarm tag. If the rate exceeds one of the limits, the alarm reports.

You configure this alarm using the **Rate Limit (RLM)** pane on the Alarms Rate Limit/Trigger Limit tab, shown in Figure 6.10.

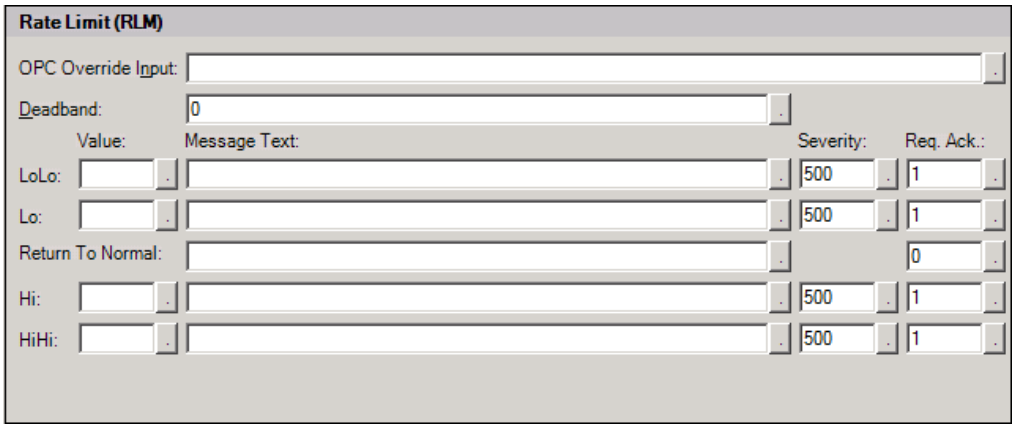


Figure 6.10 - Rate Limit Alarm Configuration - Alarms Rate Limit/Trigger Limit Tab

This alarm reads the values of the OPC tag you specified in the **OPC Override Input** field in the Rate Limit pane (if any), or the **OPC Input** field at the top of the configuration.

In the **Deadband** field, specify a buffer amount surrounding a limit in which the alarm will not signal. The deadband value is required It works the same way here as it does for the Limit and Deviation alarms; it is calculated on borderline alarming limit values to prevent repeated alarm cycles.

Enter values in the **Value** fields of **LoLo**, **Lo**, **Hi**, and/or **HiHi**. The values that you enter are compared to the rate. The alarm is triggered when the OPC tag's rate exceeds the high or low limits. Unlike the Rate of Change alarm, this alarm's rate is measured as a *real number*, not as the rate's absolute. This lets you set LoLo, Lo, Hi, and HiHi limits for the alarm.

For example, the value -1 is returned and then .5 seconds later a value of 1 is returned for a difference of +2. The difference is applied to the entire second, therefore the rate of change per second is +4. The alarm's Hi limit is set to 5, so no alarm is reported.

Then .5 seconds later a value of -1 is returned for a half-second difference of -2, making the rate of change per second -4. The alarm's Lo limit is set to -2, so an alarm is reported.

As with other alarms, for each level you can provide message text, a severity level, and you can require operator acknowledgement of the alarm. Furthermore, in the **Return to Normal** field, you can specify the message to display when the alarm returns to normal and indicate whether an operator is required to acknowledge a return-to-normal condition.

6.3.7. Trigger Limit Alarm

The **Trigger Limit** alarm reports when the OPC tag's value changes and requires a runtime operator's acknowledgement. This type of alarm is useful for an OPC tag whose value rarely changes, such as tag that reports a state. When the value changes, the alarm will report and will not reset until after a runtime operator acknowledges it. With this alarm, there is no normal condition; any change in the value triggers the alarm.

You configure this alarm using the **Trigger Limit (TLA)** pane on the Alarms Rate Limit/Trigger Limit tab, shown in Figure 6.11. By default, this alarm reads the values of the OPC tag you specified in the **OPC Input** field. You can override this tag by specifying a different tag in the **OPC Override Input** field in the Trigger Limit (TLA) pane.

This alarm is different in that you can disable and activate it. To activate it, put a check mark in the **Enable** check box.

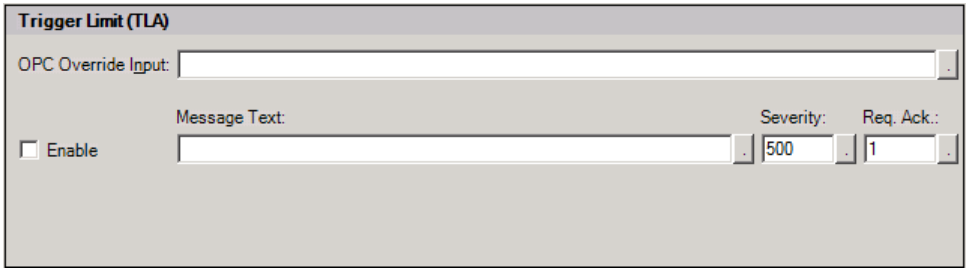


Figure 6.11 - Trigger Limit Alarm Configuration - Alarms Rate Limit/Trigger Limit Tab

Provide message text and a severity level. Then enter **1** in the **Req. Ack.** field. Because by definition this alarm requires an operator to acknowledge it, ***1** is the only value you should be entering in this field*. (Although you can enter other values, no other value is supported at this time.)

6.3.8. Alarm Tag Templates

A configuration in the AlarmWorX64 Server database exists for the purpose of holding alarm tags defined for OPC tags. There can be hundreds or thousands of alarm tags for a configuration, and creating them one by one takes time as well as attention to detail.

You can, however, create these tags using templates. Templates contain the settings that are typically used by certain types of OPC tags. You can use templates to define individual alarm tags (which we just looked at), or for defining multiple alarm tags using a wizard (which we'll look at next).

Not only does this save you time in creating alarm tags, it also makes it easier to maintain alarm tags as time passes. Why? Because after you use a template to create an alarm tag, when you change the template, the alarm tags continue to reference the template; changes that you make to the template are also made to the alarm tags that reference the template.

6.3.8.1. Creating a Template for OPC Alarm Tags

The steps for defining a template are the same as for defining an individual alarm tag. The form for creating a template is nearly identical to the form for creating an alarm tag.

To create a template for an alarm tag, in AlarmWorX64 Server, in the Project Explorer, right-click **Templates** in the AlarmWorX64 Server navigation tree, then select **Template**. The form that opens is nearly identical to the form for creating an alarm tag but with one major difference – an additional tab that we will look at soon. Use this form to define the values that are shared by the OPC tags you will be creating alarm tags for. The fields are nearly identical to those we looked at for individual alarm tags, earlier in this chapter.

6.3.8.2. Creating Alarm Tags Using a Template

To use a template in the creation of alarm tags, select the template's name from the **Template** field. Then create the tag and save it.

When you create or maintain an alarm tag created using a template, you'll see that these tags look different in the AlarmWorX64 Server Tag configurator than alarm tags created without one. Values that the template populates are highlighted with a pink background. If you type a different value, the highlighting disappears. Fields highlighted in pink can be maintained from the template and will change if they are changed in the template. Once you change a tag's value causing the pink highlighting to disappear, the field can no longer be maintained from the template.

6.3.8.3. Maintaining Alarm Tags from a Template

If OPC alarm tags were created using a template, the alarms still reference the template. Instead of modifying each OPC alarm tag individually, you can maintain the alarms uniformly from the template. When you change a value on the template, the value is essentially "pushed" to the alarms that reference the template, allowing you to maintain uniformity and making it easy to update alarm tags.

Before making changes, you may want to remind yourself of the alarms that refer to the template. To do this, go to the template form's **Alarms Assigned** tab, shown in Figure 6.12 where you can see a list of all the alarm tags that reference the template.

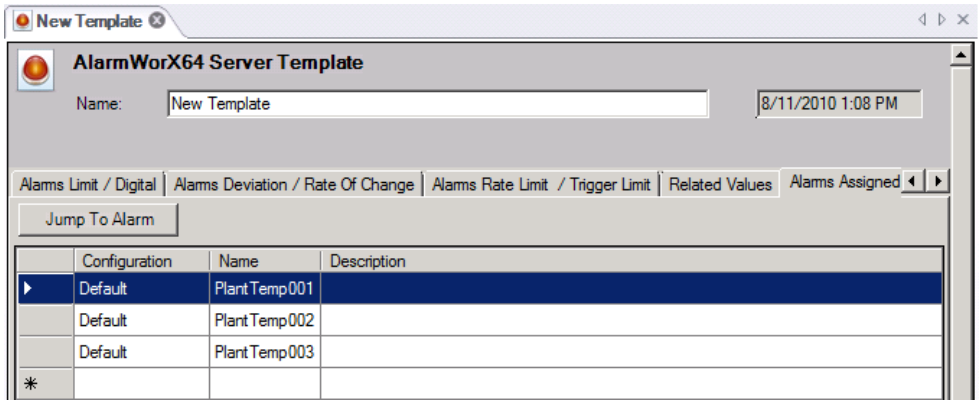


Figure 6.12 – Alarms Assigned Tab for a Template

It is important to note that *not all alarm tag fields can be maintained from the template*. Instead, you must open the alarm tag itself to maintain these fields:

- The tag's **name** and **OPC Input**
- The tag's **Enabled** checkbox
- **Deadband** values in the tag's alarms
- **Severity levels** and **Required acknowledgement** options in the tag's alarms

When maintaining a template's values, keep in mind the following:

- Only those alarm tags that were created using the template reference the template
- Once an alarm tag's field is changed manually, the field's link to the template is broken and that field can no longer be maintained from the template. Even if you restore the field to its original value, the link is broken. However, other unmodified fields on the alarm tag can still be maintained from the template.

6.3.9. Creating Several Alarm Tags at Once

At this point, we now know that an alarm tag is an OPC tag that is configured with one or more alarms with limits that specify when an alarm for the tag will report. We know that using templates helps us to create and maintain alarm tags. And we know how to define alarms one at a time. But it's likely that you have hundreds or even thousands of OPC tags that you need alarms tags defined for, and defining them one at a time will be too time-consuming. When this is the case, you can use the Tag Import Wizard to create alarm tags for multiple OPC tags.

Before you can use the Tag Import Wizard, you must have a template that you can use as the basis for all of the alarm tags you are about to define. Because templates are defined with specific types of OPC tags in mind, you'll want to plan ahead to make sure you import the right OPC tags using templates that are appropriate for the alarm tags you are creating.

In AlarmWorX64 Server right-click on your database configuration, then select **Tag Import Wizard** from the menu. This opens the AlarmWorX64 Server Import Tags form, as shown in Figure 6.13.

From the **Alarm Template** drop-down list, select the template you want to use for creating alarms for the multiple OPC tags. Optionally, if you want to assign the new alarm tags that you create to one particular area, you can select the area from the **Alarm Area** dropdown list.

Then enter the **Base Item Name**. This is the name that will be given to all of the alarm tags that are created as part of this procedure; for example, you might use PlantTemp as the base name for OPC tags that register temperature readings. A number will be appended to the base name for each alarm tag so that each alarm tag's name is unique. Numbers are increment by 1. If you leave this field empty, the name that gets assigned to each alarm tag is its OPC tag path. Note that you can rename tags later, if you want to.

If you have already created alarm tags using this base name, you can use the **First Number** field to indicate the starting number of the alarm tags that are defined. You can also specify the **number of places** that the number will occupy.

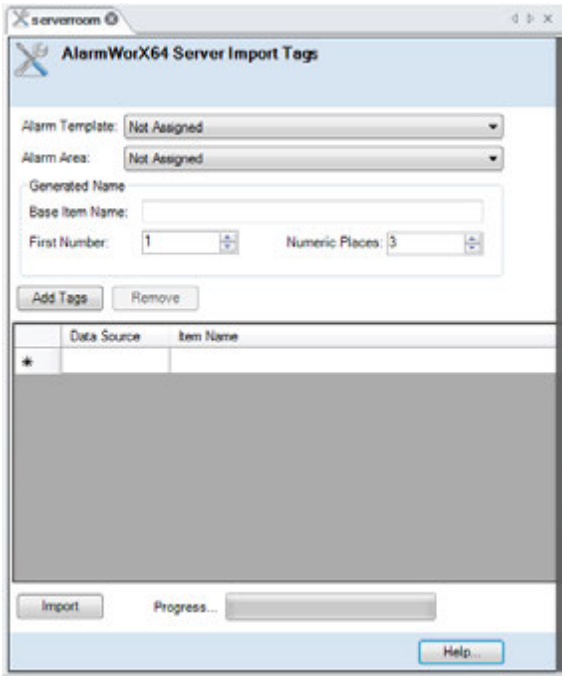


Figure 6.13 - Alarm Tag Import Wizard

Click the **Add Tags** button to open the Data Browser. This is where you will navigate to the OPC tags you are defining alarms for, select them, and click OK. You can do this several times if necessary to add all of the tags you want to include in the creation of alarm tags.

Once you add them to the AlarmWorX64 Server Import Tags form, the tags appear in the **Data Source** column. Each alarm tag's **Item Name** is the Base Item Name plus a number. Be aware that after you have added OPC tags to this list, any changes you make to the Base Item Name will not be reflected in the list; the list shows the alarm tag names exactly as they will be created. You can remove items from the list by selecting them and clicking the **Remove** button.

When you are ready to create the alarm tags, click the **Import** button. When the progress meter at the bottom of the form is full, all of the alarm tags have been created. The Results dialog box opens and lists the tags that failed because of errors and the tags that were successfully crated. You can review the new alarm tags in the Project Explorer. At this point, you can modify any of the tags or use them in AlarmWorX64 Viewer and GraphWorX64 displays. Remember to set up protection for alarms by assigning them to users or groups in Security Server, which we will be looking at later in this course.

6.4. Group Alarm Tags by Area

To organize the myriad of alarm tags that you create, you can group alarm tags into areas. An area is nothing more than a folder that you create in AlarmWorX64 for a physical area in a plant, a category of sensor, or a grouping of machines or types of equipment. It can be any category into which you want to classify your alarm tags. Not only do areas provide a way to organize, they can be used by clients for filtering.

You can include an alarm tag in any area you like and multiple areas if appropriate. Each area can have sub-areas, and sub-areas can overlap into other areas. For example, you can create an area for Alkaline Sensors, another for Security Alarms, and another for the Baltimore Plant that has sub-areas defined within it. Tags for security alarms in the Baltimore Plant area are also categorized into the Security Alarms area. Tags for some of the Baltimore Plant's alkaline sensors are categorized into the Alkaline Sensors area.

Figure 6.14 provides an example of areas in the Project Explorer pane; the Critical Temps area is expanded and one of its alarms displays in the pane on the right.

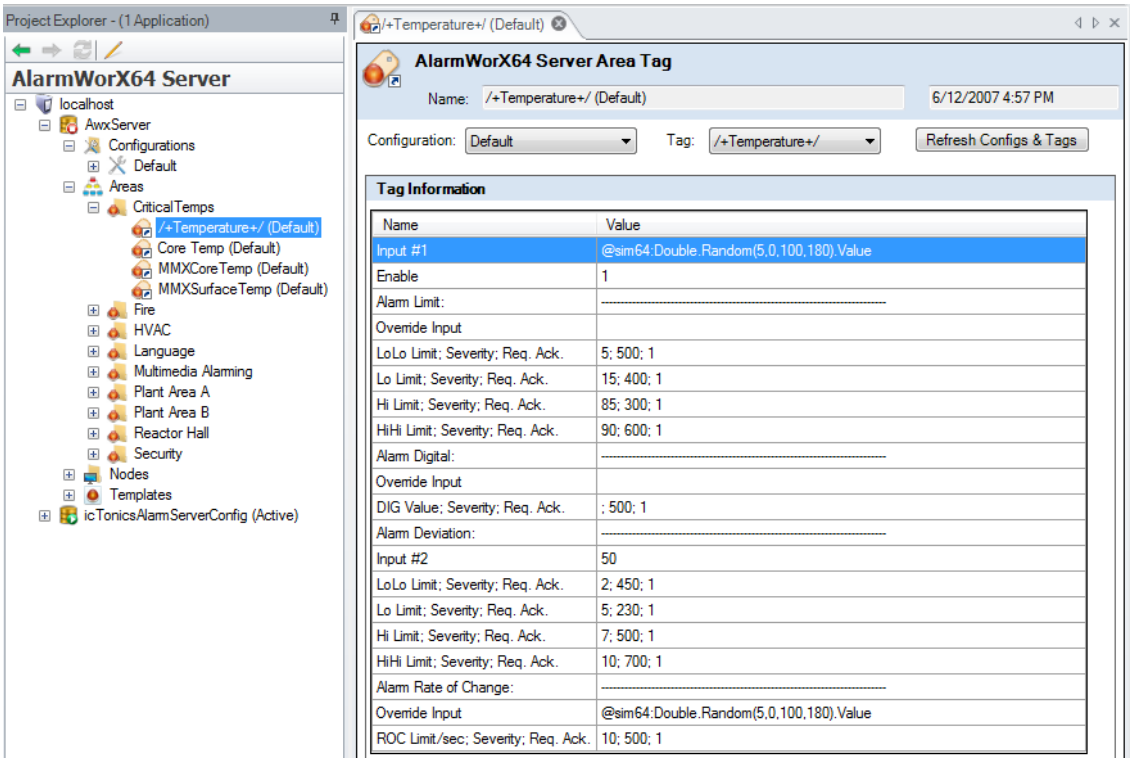


Figure 6.14 - Areas for Grouping Alarms

6.4.1. Adding an Area

To add an area, right-click on the **Areas** node, then select **Area** from the popup menu. This provides a very simple form in which you provide a name for the area and a description.

You can also add a sub-area using a similar method. Right-click on the area where you want to create the sub-area, then select Area from the popup menu. You will get the same configuration interface.

6.4.2. Adding an Alarm Tag

Once you define the new area or sub-area, you can add tags to it: right-click on the area and select **Area Tag** from the popup menu, which displays the configurator illustrated in Figure 6.15. After you pick the tag's database configuration, you can select the alarm tag from the **Tag** dropdown list.

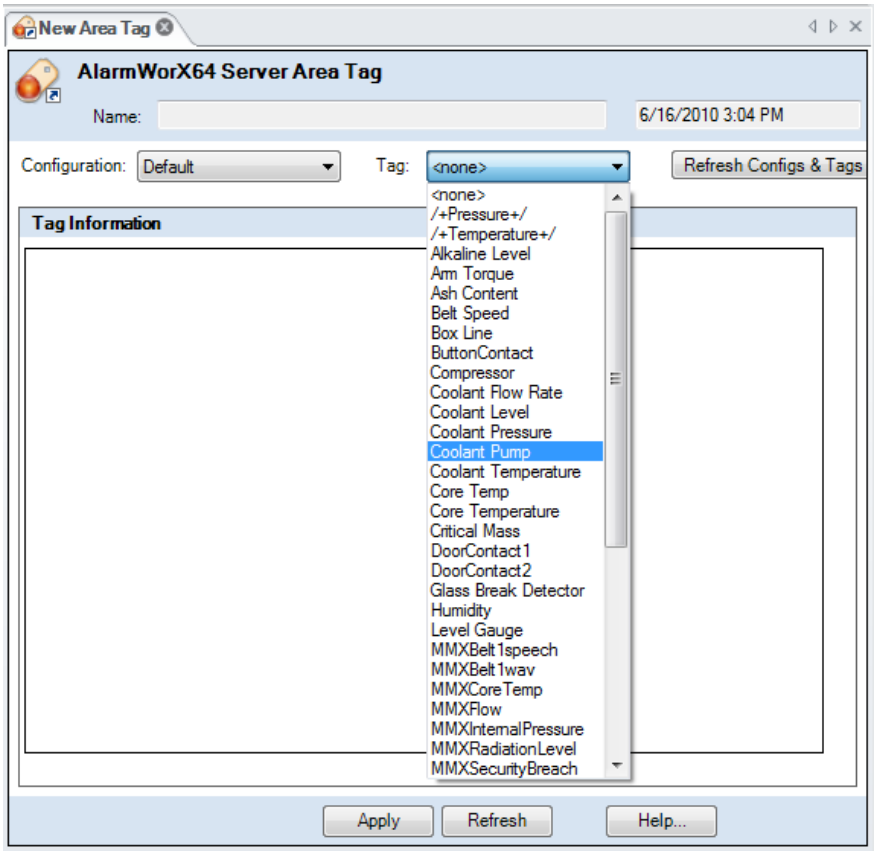


Figure 6.15 - Adding an Alarm to an Area

Once you select the alarm tag, the alarm's information displays in the **Tag Information** box.

If you browse back to the alarm, you should see the area listed in the **Areas** section of the alarm tag. You can also add or remove areas to the alarm in this section by clicking on the Add or Remove buttons.

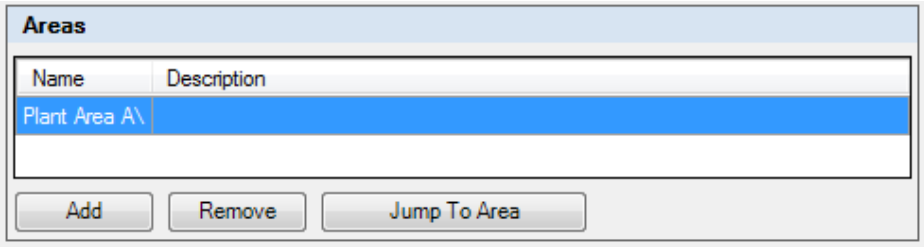


Figure 6.16 - Areas Section in an Alarm Tag

If you click on the Add button, the Select or Enter Area name dialog comes up. You can browse for the area and select it in this dialog to add it. Make sure that the area you want to add is in the text field at the top of the dialog, otherwise, the OK button will be grayed out.

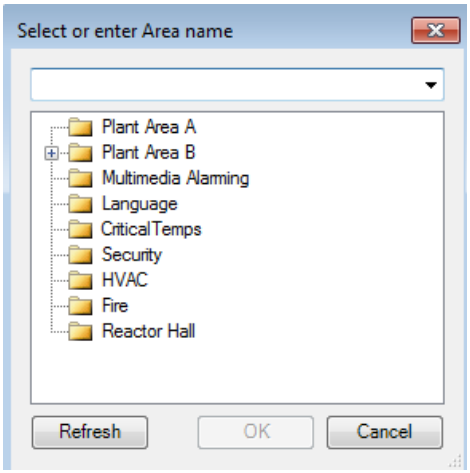


Figure 6.17 - Select or Enter Area Name Dialog

You can also remove an area from the alarm by selecting the area, and click on the Remove button.

6.5. Importing and Exporting Components

All of your alarm tags are created inside an AlarmWorX64 Server database configuration. In the lab, you will see how easy it is to create a database configuration, and then you will add AlarmWorX64 Server components to it. Alternatively, you do have the option of using an existing configuration database as a starting point if you have such a configuration database available. GENESIS64 provides the ability to export an Alarm Server database or individual components to .XML, .CSV or Excel Workbook files. This enables you to import those components into other AlarmWorX64 servers or new configurations. You can use the import to create new components or to update existing components. You can also use the export feature for the purpose of backing up an Alarm Server or its components.

In AlarmWorX64 Server, the components you can export and/or import are: a database, the Configurations folder, an individual configuration, the Areas folder, an individual area, the Nodes folder, and the templates folder. But AlarmWorX64 Server is not the only GENESIS64 product in which you can import and export components.

As you proceed through this course, keep a watch for the Import and Export options in components' right-click menus. The options are available throughout the GENESIS64 Workbench. The import and export features work the same for all components in the Workbench, and are available to help you expedite the configuration of your data. Whether you are using these features to move data, to copy existing data and then update it, or to back up data, importing and exporting components can make your data creation quicker and easier.

6.5.1. Exporting Alarm Server Components

The export copies a specified component and all of its sub-components to any of the supported file types. For example, if you export the Area folder, all areas and the alarm tags that make up that folder are exported. You might export a component so that you can import it elsewhere; or you might export simply to make a backup copy.

To export a component of the AlarmWorX64 server, right-click the component you want to export (for example, right-click the Configurations folder), then select **Export** from the menu. In the dialog box that opens, navigate to the destination or folder where you want the exported (e.g., .XML) file to be saved, and provide a name for the file. Specify a file name, and click OK. The component and all levels beneath it are exported to the file.

6.5.2. Importing Alarm Server Components

When you import components, you copy them from a file to a location of your choice. When you import a file into AlarmWorX64 Server, you can import new items only, or existing items only, or both. This lets you use the import not only *to create new items* in the alarm server, but to *update existing items*, too.

To import a component from an exported (e.g., .XML) file, right-click the component you want to import, then select **Import** from the menu. In the Import dialog box that opens, specify how you want import to happen. If there are components in the file that don't exist in the destination location, specify whether you want these components to be added. To add them, put a checkmark next to **Insert New Items**.

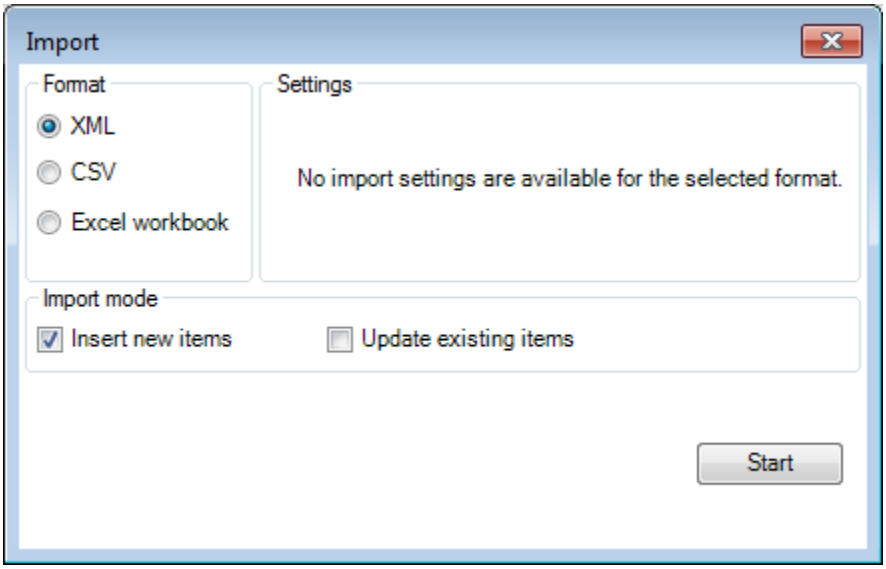


Figure 6.18 - Import Options

If there are components in the file that already exist in the destination location, specify whether you want the components to update the components in the destination location. To update them, put a checkmark next to **Update Existing Items**.

Click **Start Import**. When the import is done, you can look in the destination location for the imported component or components and confirm that they were imported as planned. Once components are imported, you can modify them in accordance with the alarm requirements of the configuration you imported them to.

Lab: AlarmWorX64 Server L6

6.1. Lab Overview

In this lab, we will create alarms that will alert us when the drink-mixer-tank level is too high or too low. We will also create an alarm that alerts us when the turner-blade in the tank stops turning. We will view these alarms in the AlarmWorX64 Viewer in the next section.

Estimated time to complete this lab: 20 minutes

6.1.1. Task Checklist

In this lab, you will complete the following tasks:

- Create a new alarm server configuration
- Create limit and digital alarms
- Create areas for different alarms
- Associate your alarms to the appropriate areas

6.1.2. Support Files

You do not need any supporting files to complete this lab.

6.2. Alarm Configuration Database

1. In order to receive alarm notifications, we will need to first configure our alarm server. In this section we will go through the steps to create a new configuration file for our icTonics beverage company project.

2. Open the Workbench if you closed it since the previous lab.

3. Expand the provider list in the left-hand tree control and select the AlarmWorX64 Server. You could also select the AlarmWorX64 Server by clicking on this icon at the bottom of the tree control window.

   Note: When your mouse hovers over each icon, you should see a tooltip saying which provider the icon relates to.

4. Expand the AlarmWorX64 Server tree under the localhost. Notice the green icon next to the (Active) configuration database. We will create our own configuration database by right-clicking on localhost → New Database.

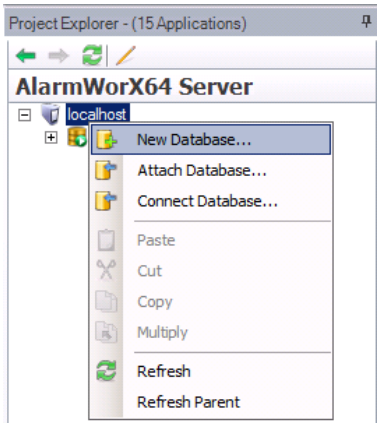


Figure L6.1 - Creating New Alarm Server Configuration Database

5. The SQL AlarmWorX64 Server - Create Database Wizard comes up. With the “Use Default SQL Server (recommended)” check box selected, click on "Next".

6. In the **Common Database Option** screen, uncheck the "Include Sample Data" checkbox, and change the Database Name to **icTonicsAlarmServerConfig**, uncheck "Include Sample Data", then click on "Finish".

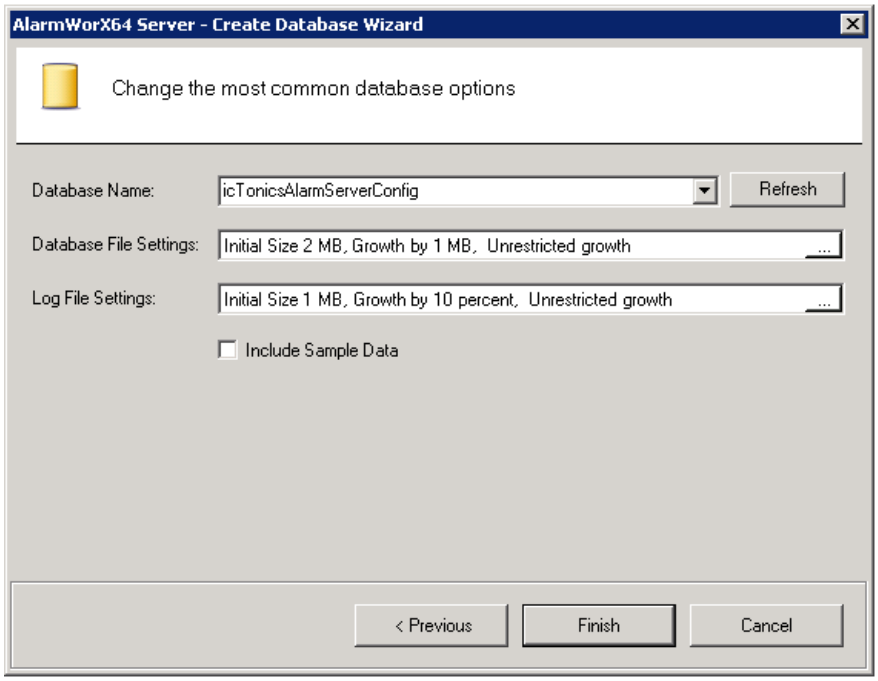


Figure L6.2 - Configuration Database Options

7. Once the database is successfully created, you will see the icTonicsAlarmServerConfig below the AlarmWorX64 Server localhost. Notice the icon to the left of the icTonicsAlarmServerConfig database is red, indicating that it is not currently an active configuration database.

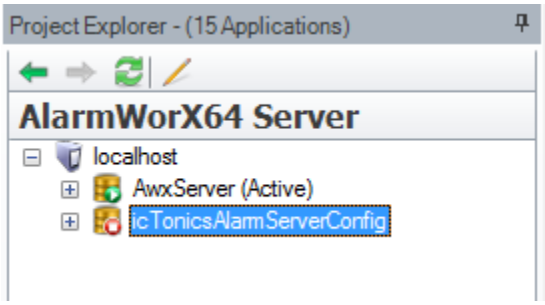


Figure L6.3 - New Deactivated Configuration

8. Expand the icTonicsAlarmServerConfig database. You will see four folders underneath it: Configurations, Areas, Nodes, and Templates. Since everything will be local to this computer, we will not worry about the Nodes. However, in the next few sections, we will create different Areas and a Configuration that will contain your alarms. As we proceed with our configuration, adding description information is optional.

6.2.1. Alarm Areas

1. Right-click on the Areas folder and select “+ Area”.

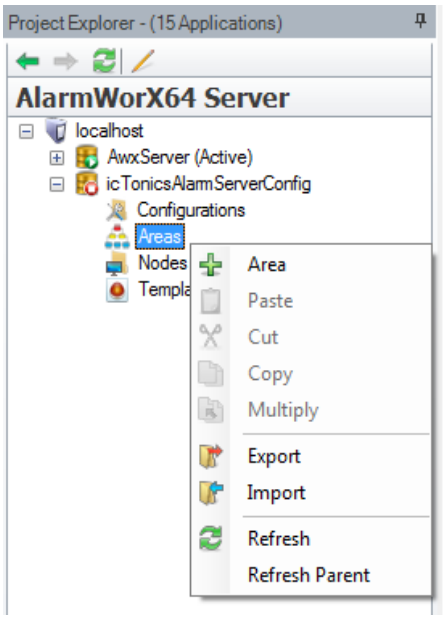


Figure L6.4 - Adding an Area

2. A new workbench tab opens that allows you to create a new Area. Name the Area, “USA”. Click Apply.

3. Create a sub-area by right-clicking on the area called USA and select “+ Area”. We’ll name this area “Foxboro”, Click on "Apply".

4. Create another sub-area under Foxboro called “Tank Levels”. Click on "Apply". When you are done, your areas should look like Figure L6.5.

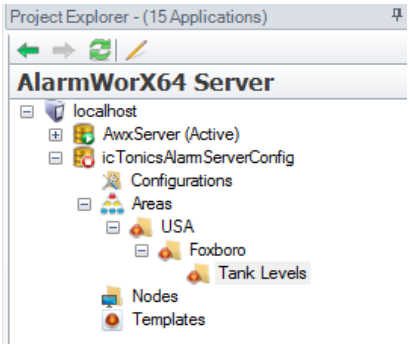


Figure L6.5 - Alarm Areas

6.2.2. Alarm Configurations

1. Right-click on Configurations and select "+ Configuration" to create a new alarm configuration.

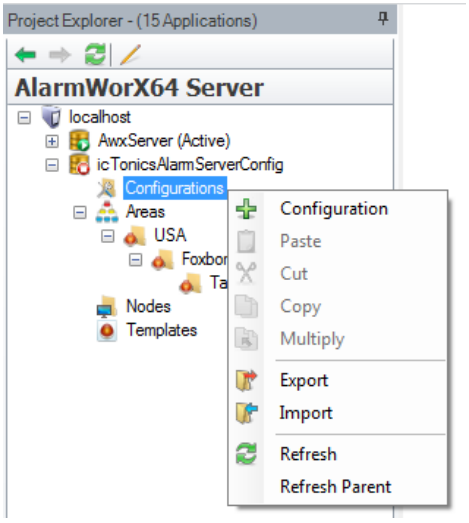


Figure L6.6 - Inserting Alarm Configuration

2. Name this configuration "icTonicsAlarms" and put a checkmark in the "**Default (Active) Configuration**" checkbox to make this configuration active upon startup. Click on "**Apply**" to save changes.

3. Right-click on the icTonicsAlarms, under the Configuration tab, and select "+ Tag" to add an alarm tag.

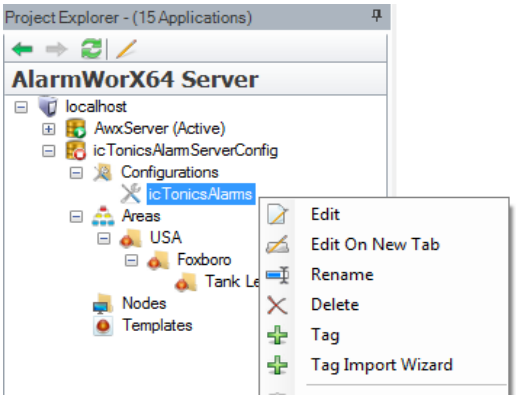


Figure L6.7 - Inserting Alarms

4. A new tab opens with alarm tag properties. Give this alarm the name "Tank1 Level".

5. In the next blank, click on the ellipse button to the right of the OPC Input text box. In the Data Browser we will look for the tag called **@ICONICS.IconicsOPCUAServer.V5\icTonics.Tanks.USA.Foxboro.Tank1.TankLevel.Value.**

6. With the **General** tab selected enter the text "**Tank1 level is**" for the **Base Text** field. Then click "Apply" to save the tag.

7. In the **Areas** pane near the bottom of the box, click on the "**Add**" button, and add the **Tank1 Levels** area that you created in the previous section.

8. Your General tab should look like Figure L6.8. Click on "Apply" to save the changes.

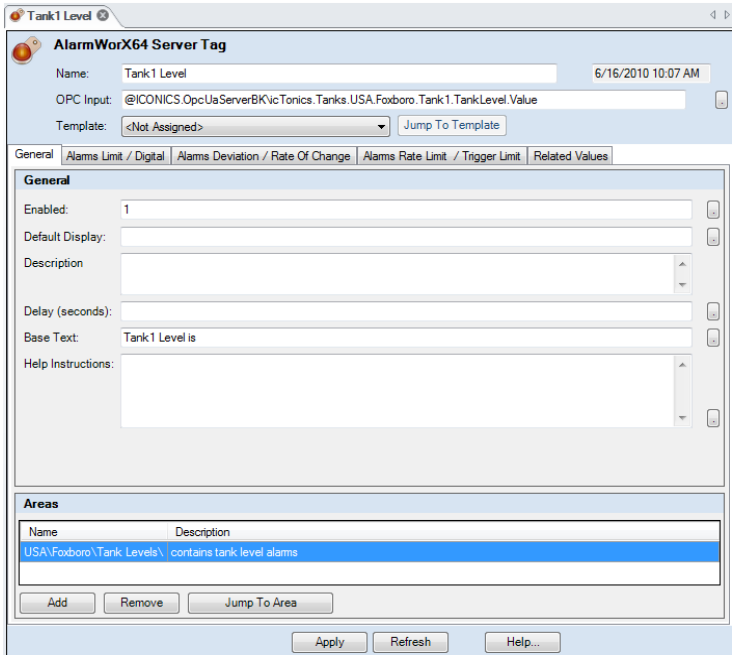


Figure L6.8 - Alarm Configuration General Tab

9. Select the **Alarms Limit / Digital** tab. In the Limit section, configure the alarm properties as shown in Table L6.1.

Table L6.1 - Limit Alarm Value

Condition	Value	Message	Severity
LoLo	10	very low	800
Lo	20	low	500
Hi	80	high	500
HiHi	90	very high	900

10. In the **Return to Normal**: message field enter "**normal**". Your Limit section should look like Figure L6.9. Click on "Apply" to save changes.

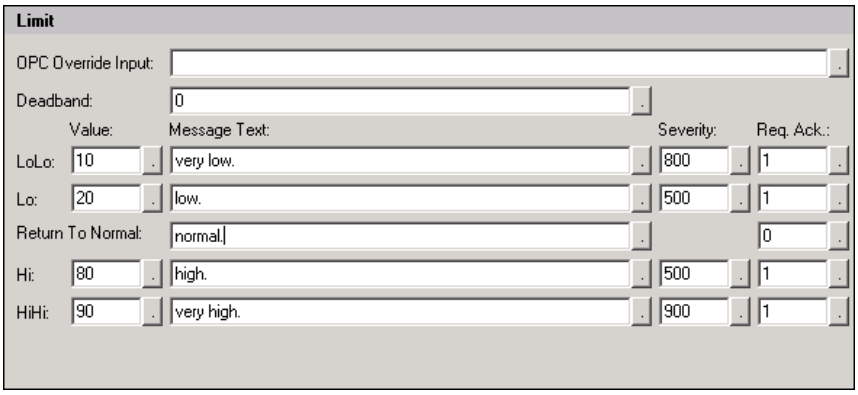


Figure L6.9 - Limit Alarm Configuration

11. Using a similar procedure, create another New Tag. The configuration will differ slightly since this will be a Boolean tag. Under the **Alarms Limit / Digital** tab, in the **Digital** section of the same screen, fill in "Alarm State Value", "Message Text" and "Return To Normal" according to the chart below. When configured completely, this tag will alarm if the churning blade1 has stopped moving. Table L6.2 shows a few details about this alarm.

Table L6.2 - Digital Alarm Details

Name:	Churning Blade1 Status
OPC Input:	@ICONICS.IconicsOPCUAServer.V5\icTonics.Tanks.USA.Foxboro.Tank1.ChurningBladeStatus.Value
Base Text:	The churning blade in Tank1
Alarm State Value:	0
Message Text:	Has stopped moving
Return To Normal:	Is moving

12. We are now done configuring the alarm serve however there is one Important step left. We need to make icTonicsAlarmServerConfig the Active database and confirm the alarm Service starts properly.

13. As we perform the activation procedure, monitor the Start/Stop light on the side menu. Right-click on the **icTonicsAlarmServerConfig** database and select "**Make Active**". Make sure that you have activated your configuration database so that it shows (Active) and there is a green indicator. Also, the Start/Stop light on the ribbon should be green.

6.3. Alarm Tag Multiply

1. Now we will add similar alarms for tank 2 and tank 3. For this, we can use the Multiply function which makes it easier to create similar alarms.

2. Right click on the "**Tank1 Level**" alarm tag and select "Multiply" to launch the **Multiply Options** dialog box. You could alternatively select "Tank1 Level" and click on Multiply icon in the Clipboard section of the ribbon.

3. Set the **Number of items** to "2", the **First number** to 2 and the **Numeric places** to 1.

4. Replace the **Tank1** value in all of the **Base Value** fields with **Tank{0}**.

5. Uncheck all the fields that have an empty Base Value.

6. When your dialog resembles the figure below, Click on "**OK**".

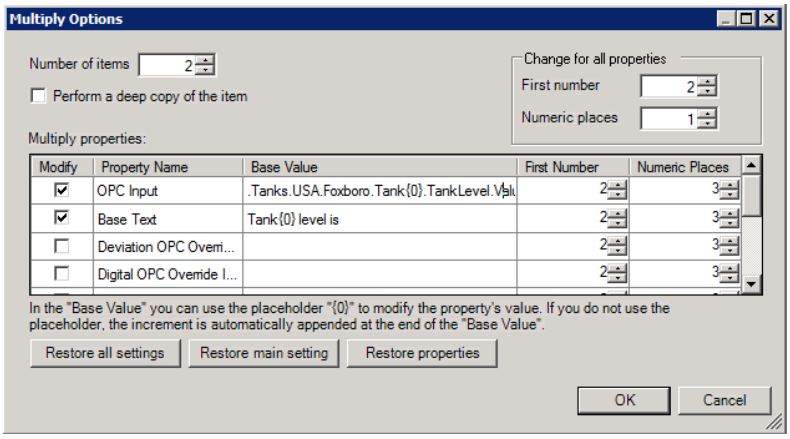


Figure L6.10 - Multiply Options Dialog Box

Note: As an alternative method you could Copy and Paste to create multiple alarm tag entries.

7. With a left mouse click, select "Churning Blade1 Status" alarm tag in the tree control. Click on Copy icon in the Clipboard section of the ribbon. Notice that the Paste icon is grayed out.

8. Select "icTonicsAlarms" (under Configurations) in the tree control. When you do this, notice that the Paste icon is now visible and selectable. Click on the "Paste" icon in the Clipboard section of the ribbon.

9. Repeat steps 5 and 6 to create another copy. Your icTonicsAlarmServerConfig should look similar to the figure below.

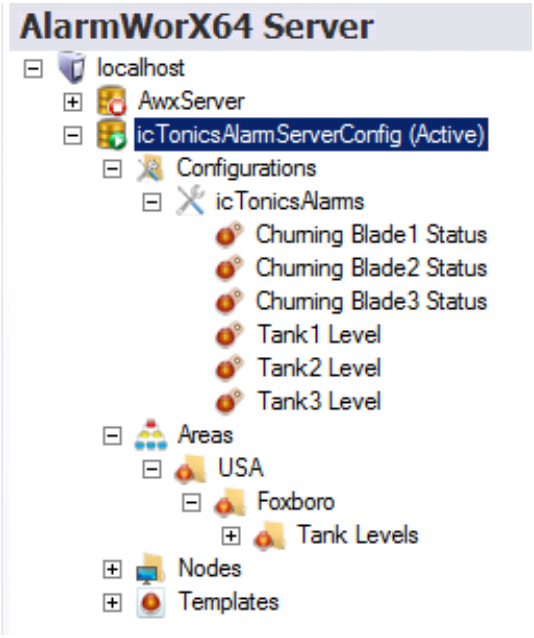


Figure L6.11 - IcTonicAlarms after Multiply and Copy / Paste

10. *This step is very important as it will affect your alarm functionality for the rest of the training session.* Carefully review and edit each of the newly created alarm tags and make the appropriate changes on both the "**General**" and "**Alarms Limit/Digital**" tabs.

11. In the upper section of the AlarmWorX64 Server Tag configuration be sure the "**Name**:" and the "**OPC Input**:" sections are **Unique** for each tag. Each of the newly created tags will have the same OPC Input values. As in the example below for the multiplied tags, change the OPC Inputs from Tank1 to Tank2 and Tank3.
@ICONICS.IconicsOPCUAServer.V5\icTonics.Tanks.USA.Foxboro.Tank1.TankLevel.Value
@ICONICS.IconicsOPCUAServer.V5\icTonics.Tanks.USA.Foxboro.Tank2.TankLevel.Value
@ICONICS.IconicsOPCUAServer.V5\icTonics.Tanks.USA.Foxboro.Tank3.TankLevel.Value

    You should also check our churning blade tags as well.

12. Your finished configuration should look similar to the following:

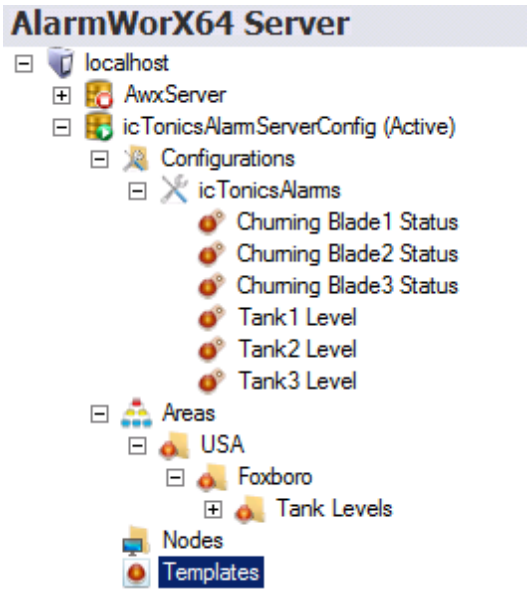


Figure L6.12 - Advanced Alarm Server Configuration

Review: AlarmWorX64 Server *R6*

6.1. Objectives Review

You should now know and understand the following topics:

- AlarmWorX64 Server and how to launch it
- The different types of alarms
- Configuring alarm tags individually or using the Tag Import Wizard
- Configuring and using an alarm tag template
- Grouping alarm tags by area

6.2. Summary

The AlarmWorX64 Server receives field data from your OPC compliant Data Access servers and detects alarms and reports on that data. Event notifications generated by the AlarmWorX64 Server are sent to the OPC UA clients (such as, AlarmWorX64 Viewer and AlarmWorX64 Logger) that subscribe to it. During runtime, the server polls the configuration database for changes, allowing for on-the-fly configuration edit without stopping the server.

You can use alarms to alert operators to adverse or critical conditions, prompt operators to perform manual procedures or equipment reviews, assist in training new operators, and more. In general, AlarmsWorX64 Server takes each reading that it gets for the OPC tag and compares it to another value, then determines if the incoming OPC value is in an alarm condition. If it is, the alarm server generates an alarm and sends it to subscribed alarm viewers where nearby operator can view the alarm.

Tools can help you define the myriad alarm tags you need to create for your OPC tags. Alarm tag templates define common values shared by the alarm tags that are created from it. Furthermore, you can use a template to "push" changed values to the tags created from it. The Tag Import Wizard lets you quickly and easily create an unlimited number of alarms tags with the use of one common template.

There are six types of that you can configure and combine in an alarm tag alarms– Limit, Deviation, Digital, Rate of Change, Rate Limit, and Trigger Limit. A *Limit* alarm occurs when a tag's value exceeds a high or low threshold that you have configured. A *Deviation* alarm occurs when the absolute value of the difference between two input values exceeds a limit that you specify. A *Digital* alarm occurs when the OPC Input value equals the alarm state value that you specify. A *Rate of Change* alarm is generated if the rate of change is higher than the limit you set. A *Rate Limit* alarm is like the Rate of Change alarm except that its rate is not an absolute number– instead, it is expressed as a positive or negative value. A *Trigger Limit* alarm occurs when the tag's value changes; the alarm resets only after a runtime operator acknowledges it.

You can take advantage of deadbands and alarm delays to handle tags that tend to flicker in and out of an alarm condition. You can tailor message text to alarm levels, and you can provide instructions for the operator to follow when responding to an alarm. You can require an operator to acknowledge an alarm and a return to normal. Furthermore, you can collect data from *other* tags during an alarm condition for the purpose of reviewing their captured readings later, for troubleshooting, or for diagnosing a problem. Finally, you organize alarm tags into folders called areas which can also be used by clients for filtering data.

6.3. Terms

Alarm Area	Alarm State Value	Alarm Tag
Alarm Template	AlarmWorX64 Server	Deviation Alarm
Digital Alarm	Limit Alarm	Rate of Change Alarm
Rate Limit Alarm	Tag Import Wizard	Trigger Limit Alarm

6.4. Test Your Knowledge

In this chapter, we discussed how to use AlarmWorX64 Server to create a database of alarm tags and organize them into areas. Before continuing to the next chapter, make sure you can answer the questions below:

1. What AlarmWorX64 clients can subscribe to the AlarmWorX64 Server?

    a. Alarm Viewer
    b. Alarm Logger
    c. Historic Alarm Viewer
    d. All of the above
    e. Both A and B
    f. Both A and C
    g. Both B and C
    h. None of the above

2. Which database type can you use to create and AlarmWorX64 configuration database? (Circle all that applies).

    a. MSDE
    b. MS SQL 2005 Express
    c. MS SQL 2008 Express
    d. Oracle
    e. MySQL
    f. All of the above
    g. Only A and B
    h. Only B and C
    i. Only D and E
    j. None of the above

3. True or False: You can have multiple alarm configurations within a single configuration database, but you can only run one active configuration at a time.

4. True or False: Alarm areas are a way to organize your alarms into logical folders that represent your alarm locations or structure.

5. True or False: The only way to disable an alarm tag once you have created it is to delete it or delete its alarm definitions.

6. When might you use the Tag Import Wizard?

    a. When you need to acknowledge a multitude of alarms all at once
    b. When you are initially defining alarms for your OPC tags
    c. When changing the Return to Normal message on an alarm
    d. All of the above

7. What are some benefits in using a template to define alarm tags?

    a. You can use a template with the Tag Import Wizard for defining multiple alarm tags uniformly
    b. By changing values on a template, you push changes to all alarms created from the template
    c. Alarm tag fields that reference the template are easy to identify
    d. All of the above
    e. None of the above

8. True or False: When importing AlarmWorX64 components, you must make sure that you do not overwrite any existing components.

For the remaining the questions, choose one of the answers below:

    a. Digital alarm
    b. Limit alarm
    c. Rate of Change alarm
    d. Deviation alarm
    e. Trigger Limit alarm
    f. Rate Limit alarm

9. If you want an alarm to trigger when a sensor's readings are changing too quickly, which type of alarm would you configure?

10. If you want to make sure a door or valve is closed under certain conditions, which type of alarm would you configure?

11. If you want an alarm to generate when a temperature reading rises above 28 degrees and again if it reaches 32 degrees, which type of alarm would you configure?

12. If you want the levels of two reservoirs to remain relatively equal, which type of alarm would you configure?

13. If you want to know if a reading ever changes no matter what it changes to, which type of alarm would you configure?

AlarmWorX64 Viewer 7

The AlarmWorX64 Viewer is client application that provides real-time and historical alarm information within the GENESIS64 product family. Alarm information displays in a grid-like view similar to how it worked in GENESIS32.

The viewer connects to OPC UA servers and the legacy OPC AE servers to provide alarm display and management features based on the connections to the data source that you specify.

In AlarmWorX64 Server, you control how often a client can access the server for data (polling rate). In the AlarmWorX64 Viewer, you determine:

- Which alarms display on the screen
- What the alarm display looks like
- How the alarms are grouped and sorted

The AlarmWorX64 viewer, as indicated by its name, can be used to view alarms, like those shown in Figure 7.1. You can also use it to acknowledge alarms and do simple alarm analysis. You can add an AlarmWorX64 viewer inside the Workbench or you can add the AlarmWorX64 control onto a GraphWorX64 design surface. You can also insert the AlarmWorX64 viewer into any .NET Framework form.

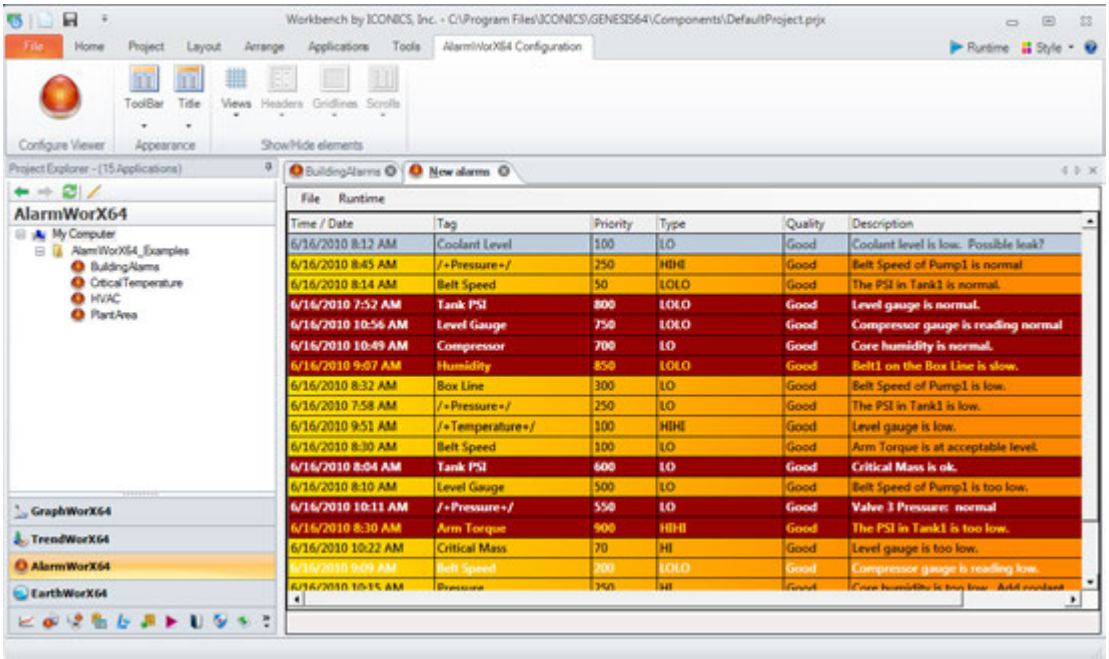


Figure 7.1 - AlarmWorX64Viewer Inside the Workbench

If you like, you can display alarms in a chart or grid, like the one shown in Figure 7.2, which provides a summary or aggregate view of alarms organized by the criteria you specify. Each AlarmWorX64 viewer can have multiple tabs and each tab can be connected to multiple grids and charts.

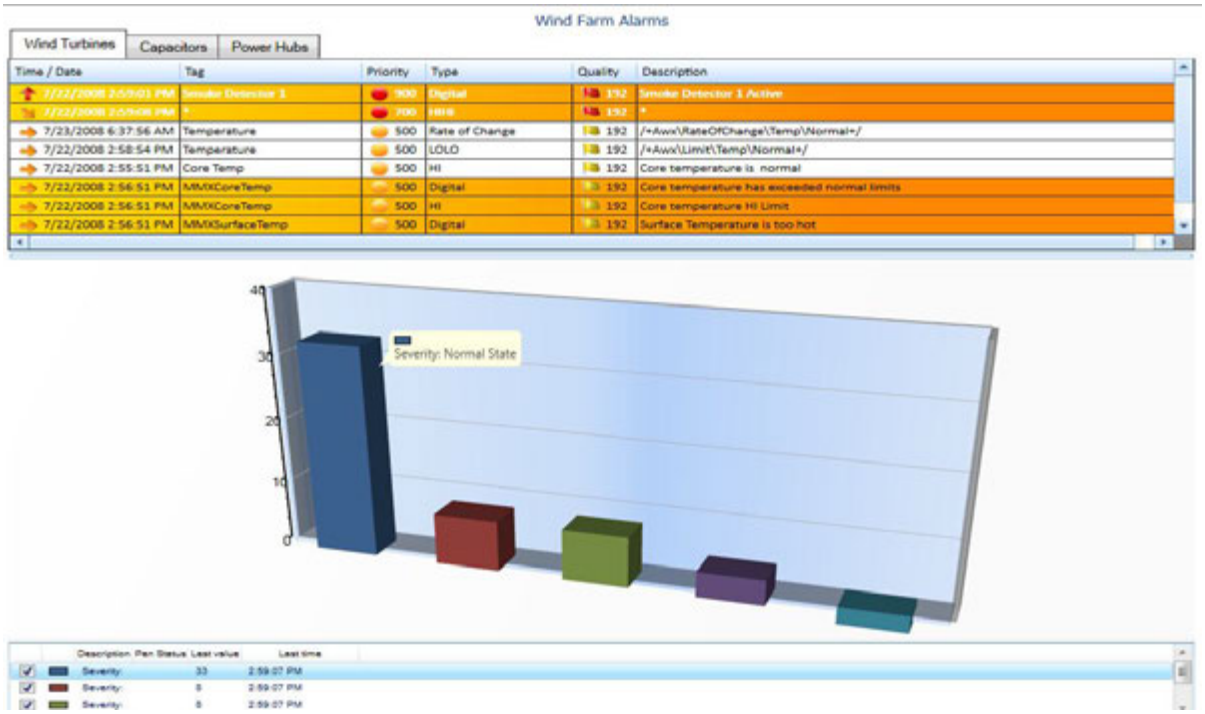


Figure 7.2 - AlarmWorX64 Viewer with a Summary Chart

By the end of this chapter, you will be familiar with the following:

- The alarm viewer control
- Tabs, grids, and charts in the alarm viewer
- Grid Configuration control
- Sources, appearance, and behavior
- Status indicators and text translators
- Grouping ranges and grouping
- Columns and conditions
- Alarm viewer runtime options

7.1 Configuring the Viewer

AlarmWorX64 Viewer can be configured in the Workbench. To open the configurator, you click the Configure Viewer button on the AlarmWorX64 Configuration tab or double-click on the AlarmWorX64 object in a GraphWorX64 display.

The AlarmWorX64 Viewer configurator appears as shown in Figure 7.3 below. The configurator is divided into two parts: on the left is a tree view of tabs, grids, and charts; on the right are properties for the selected item. AlarmWorX64 automatically adds at least one tab and one grid to the display. You can alter the display of the viewer, tab, or grid objects; but only the grid object and chart object can connect (or *subscribe*) to an alarm event source.

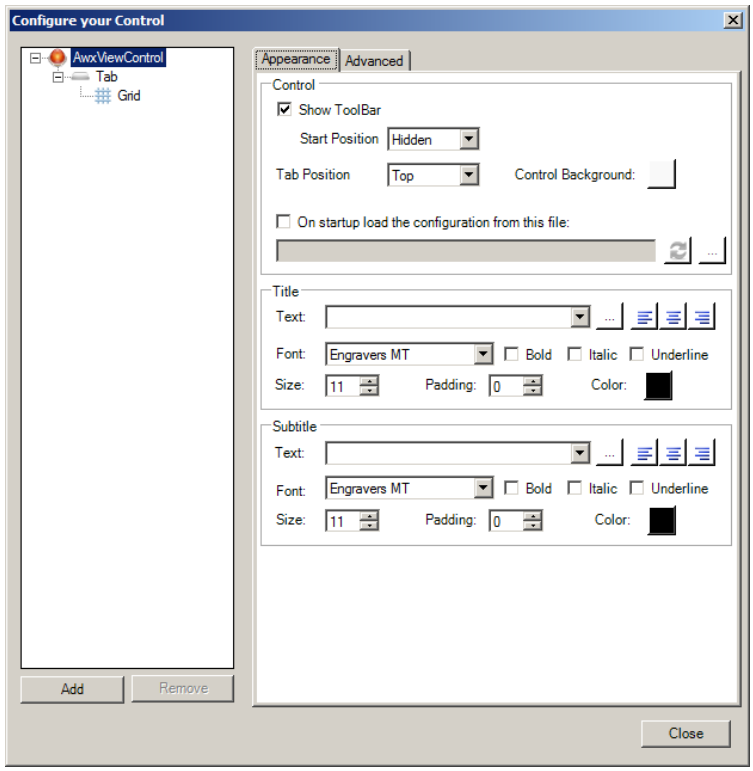


Figure 7.3 - Configuring the Viewer's Appearance Tab

In Figure 7.3, the viewer is selected in the navigation tree, making its Appearance and Advanced tabs available on the right. On the viewer's Appearance tab, you can:

- Control the viewer's appearance
- Dock the toolbar to the top, bottom, right, and left, as well as float the toolbar (these operations are available in runtime as well)
- Enter and format a name for the viewer
- Add a subtitle as a second line below the title and format it

The Advanced tab lists other properties for the viewer. Note that all items in the navigation tree have an **Advanced** tab that lists additional properties for the item.

7.1.1 Adding a Tab

To add a tab to the viewer, select the AlarmWorX64 Viewer in the tree control, and click on the "Add" button at the bottom. (Alternatively, you can right-click on the AlarmWorX64 Viewer, and choose Add → Tab as shown in Figure 7.4.) Then you can edit the tab properties in the tab's Appearance and Advanced tabs on the right.

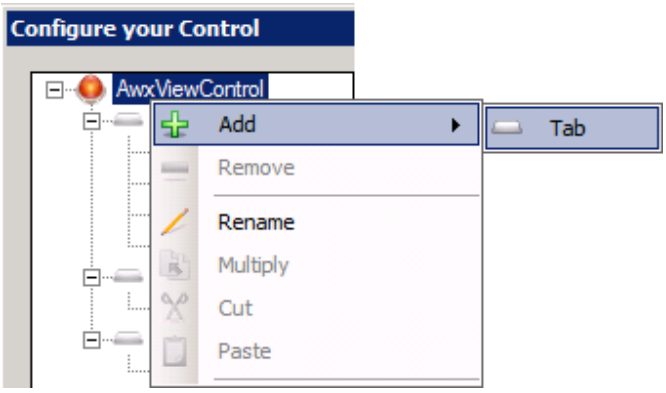


Figure 7.4 - Configuring AlarmWorX64 Viewer

Using the tab's Appearance tab, shown in Figure 7.5, you can customize the title that appears in the tab at runtime and specify the tab's properties when it is selected and unselected. Notice the ellipsis button next to the title's Text field. Clicking the ellipsis button opens the Data Browser where you can assign a language alias (letting you translate tabs for a multi-lingual user base) or expression. You can also apply these property values to all of the tabs in the AlarmWorX64 Viewer.

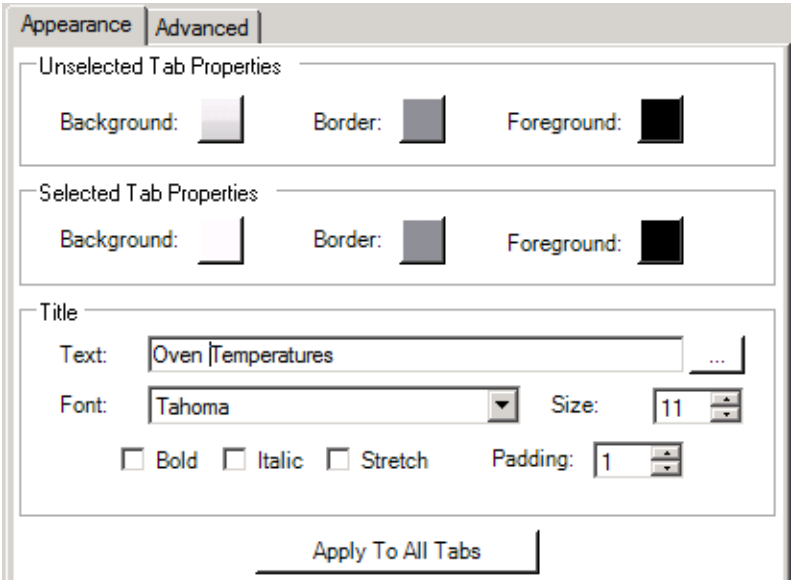


Figure 7.5 - Configuring AlarmWorX64 Viewer

7.1.2 Adding a Grid or Chart to a Tab

To add a grid or chart, in the navigation tree you can select the tab that will contain the grid or chart. Then, click on the "Add" button at the bottom and select "Grid" or "Chart" from the dropdown list. Alternatively, you can right-click on the tab in the navigation tree, and select Add → Grid or Add → Chart as shown in Figure 7.6.

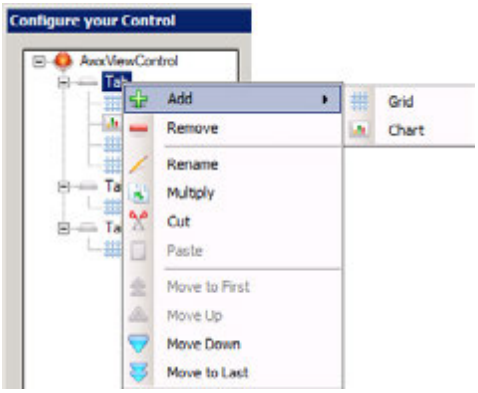


Figure 7.6 - Adding a Grid or Chart to a Tab

Adding a chart is like adding a grid, but with fewer options. The chart properties are shown in Figure 7.7. Because charts and grids are similar, and because grids offer many more options, this chapter will describe AlarmWorX64 Viewer using grids as an example. However, keep in mind that grids and charts are configured in much the same way.

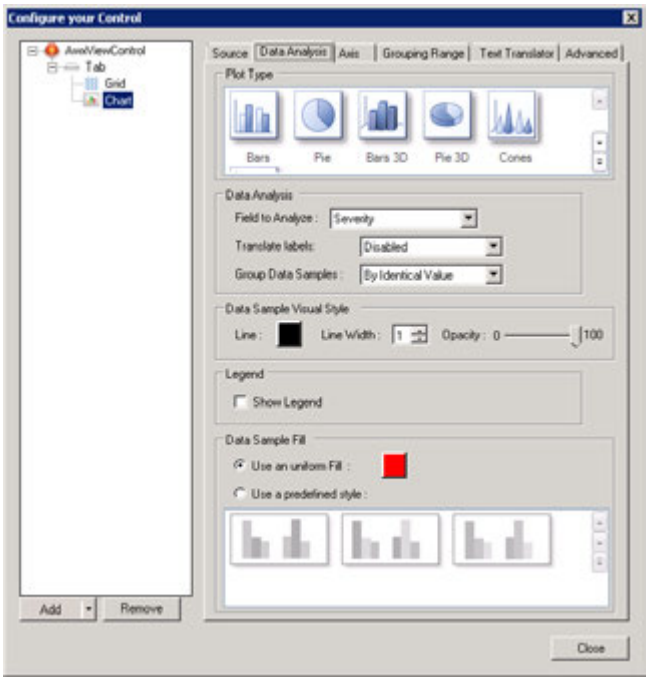


Figure 7.7 - Chart Added to a Tab in AlarmWorX64 Viewer

7.1.3 Adding an Alarm Subscription to a Chart or Grid

Charts and grids are the AlarmWorX64 Viewer component where you add the OPC tag alarms that you want the display to monitor. When you add an alarm, the chart or grid is said to *subscribe* to the alarm. Once you have added a subscription, the AlarmWorX64 Viewer gets alarm updates from real-time alarms connected to an alarm server, which is the source of our alarms.

To make this connection, you must specify the alarm source. To configure the alarm source, click on the grid or chart object in the tree control. As a result, the **Source** tab, shown in Figure 7.8, displays in the source properties where you define the alarm source.

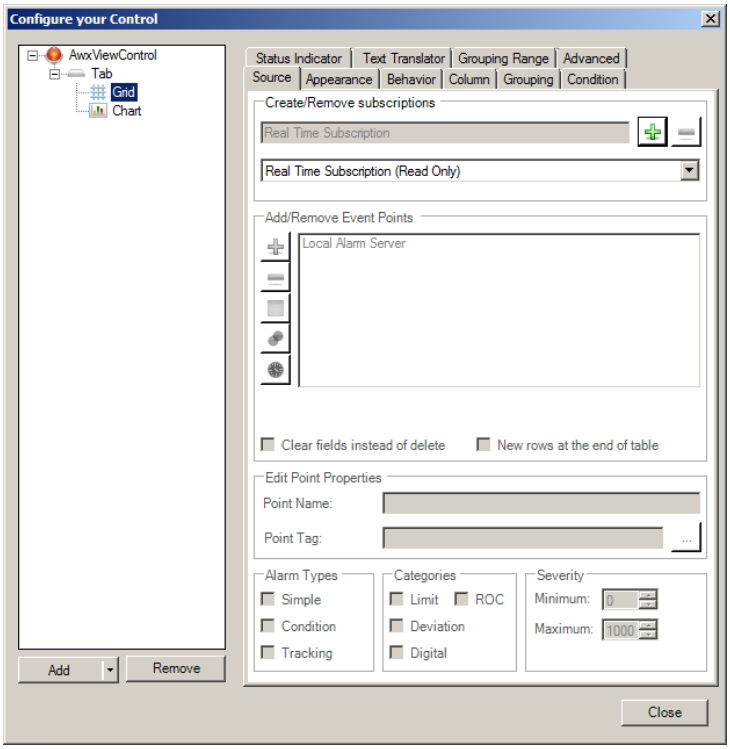


Figure 7.8 - Real Time Alarm Subscription

To add an alarm subscription, first use the **Create/Remove Subscriptions** section at the top of the Source tab to create a new subscription list. Click the "Add" button, you will get a popup menu list beside the button. From the menu list, you can choose "real Time Subscription", "Historical Subscription" or "Alarm Subscription Wizard". Next to this selection, you can name your new subscription in the upper text bar. You will notice that the lower the content in the lower text bar will change correspondingly when you rename your subscription. You can also click the down arrow to change your subscription type.

Then select the OPC servers that you want to subscribe to in the **Add/Remove Event Points** section. Click on the "Add" button in this section to bring up the Data Browser as shown in Figure 7.9.

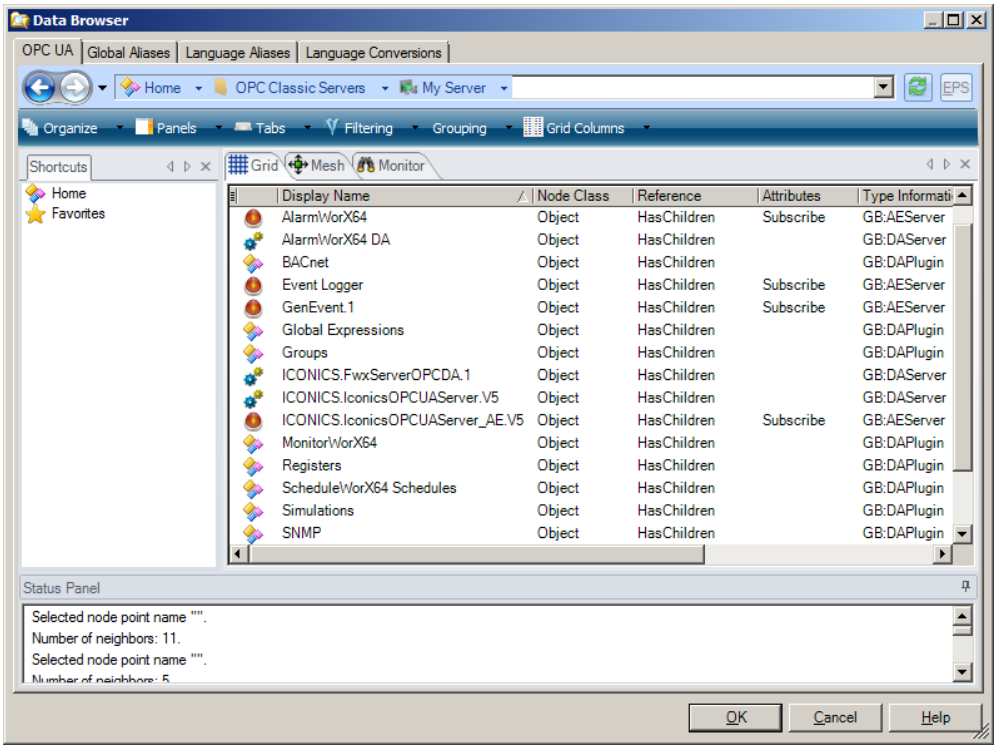


Figure 7.9 - Browsing for an Alarm Subscription

As you can see, there are many OPC servers in the list. You will be able to subscribe to a legacy alarm server (AlarmWorX32 Server) if you have it installed. In addition, as shown in Figure 7.9, part of the GENESIS64 Alarm Server is also a DA server. Make sure you select the AE Server (the one that reads AlarmWorX64 instead of AlarmWorX64 DA) for this subscription.

> **Note:** You can also browse for a historical alarm that will replay a timeframe of your choosing. If you select the Historical option, you will also need to specify the time range for the data to be replayed.

Once you have added a subscription, you will see it in the Add/Remove Event Points section. You can then select the subscription and click on the "Edit" button to add or remove fields for the alarm subscription, using the **Add/Remove fields from the subscription** dialog box shown in Figure 7.10.

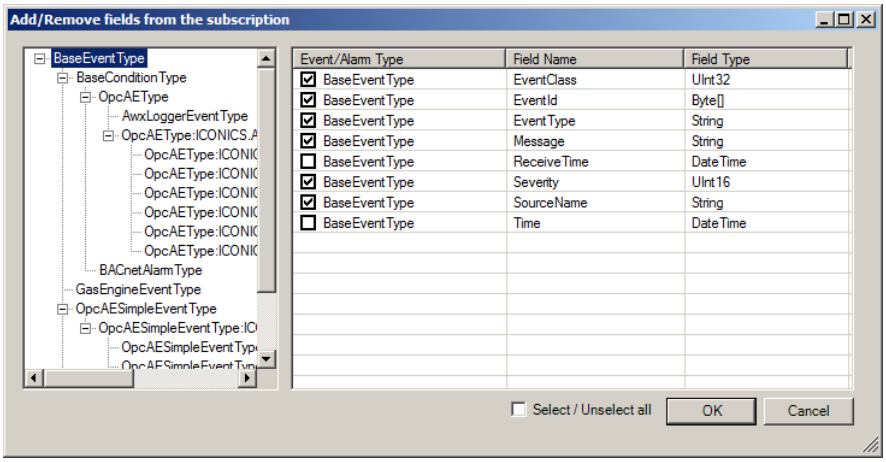


Figure 7.10 - Add or Remove Fields in an Alarm Subscription

7.1.4 Configuring a Grid

The remainder of this chapter describes how to format a grid on an AlarmWorX64 Viewer tab. There are many tabs containing different options that allow you to personalize the alarm grid. We will go through most of these tabs and their properties.

7.1.4.1 Appearance Tab

Use the grid's **Appearance** tab, shown in Figure 7.11, to configure the appearance of the grid. Each grid can have its own style and features. You can also select the elements to show or hide in the grid. To achieve uniformity in appearance among all of your grids, you can even apply your style choices to all grids in the tab, or all grids in the viewer.

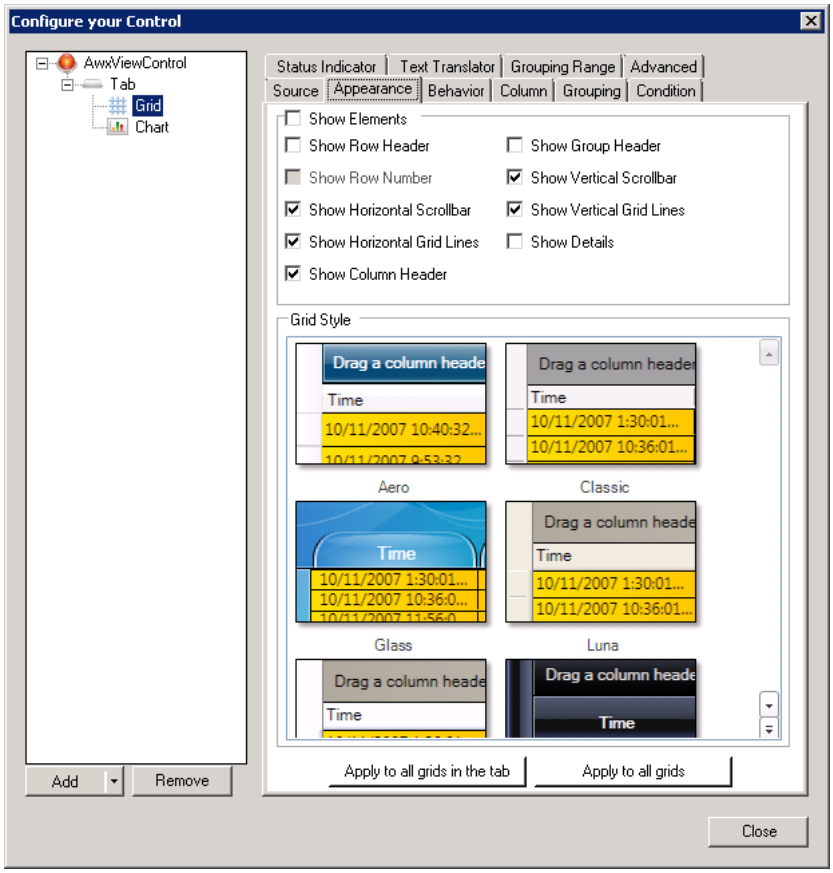


Figure 7.11 - Appearance Tab for a Grid

7.1.4.2 Behavior Tab

The grid's **Behavior** tab is shown in Figure 7.12. On this tab are options for runtime features.

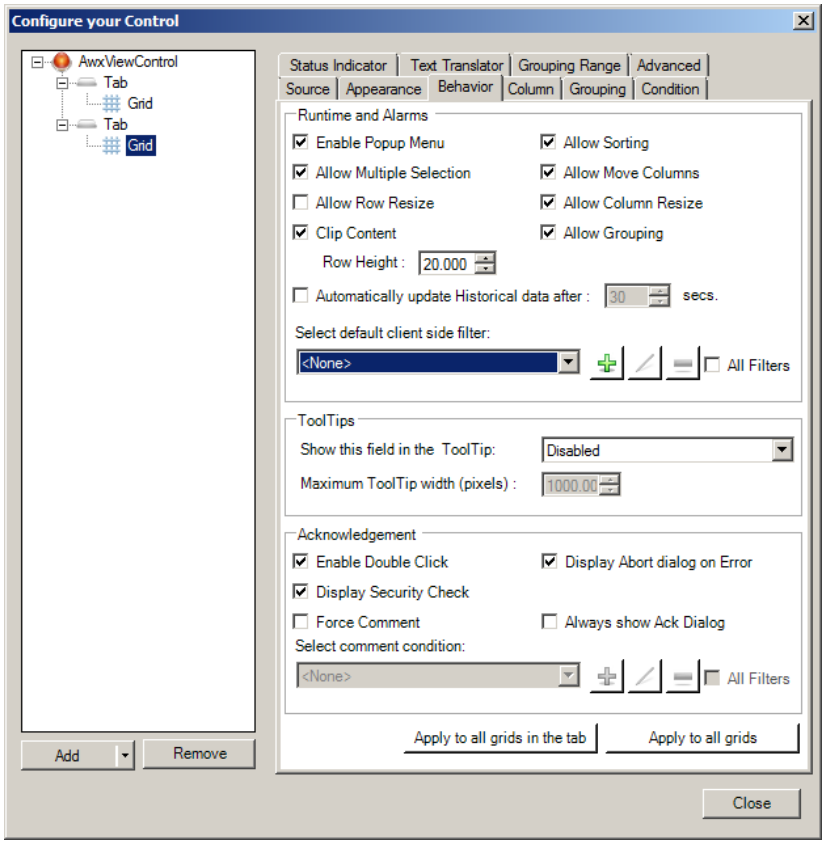


Figure 7.12 - Behavior Tab for a Grid

In the **Runtime and Alarms** section, use the options to configure what runtime users are allowed to do in the viewer. You can also create client side filters, which help you to limit the alarms shown in the grid, making it easier for users to find a particular alarm. For example, you can limit the display to alarms that have higher priority or alarms within a certain area.

You can create your client-side filters in the Behaviors tab. Click on the "Add" button next to the "Configure Client side filter" label to bring up the **Configure Alarm Filter** dialog. Click on button to edit the existing filter. Click to delete selected filter. In the Configure Alarm Filter dialog, you can create your own client side filter. The dialog is intuitive. To create a filter, you need to first specify the name of the filter in the top textbox. Then, you can specify your alarm state, type, sub conditions and category in the radio box.

You can also customize the alarm severity, tag, message, etc by clicking the check box first. The  button beside each row can help you define the global aliases and the language aliases. Figure 7.13 shows an example of a filter. Any filters you create within this dialog are usable throughout this instance of the control.

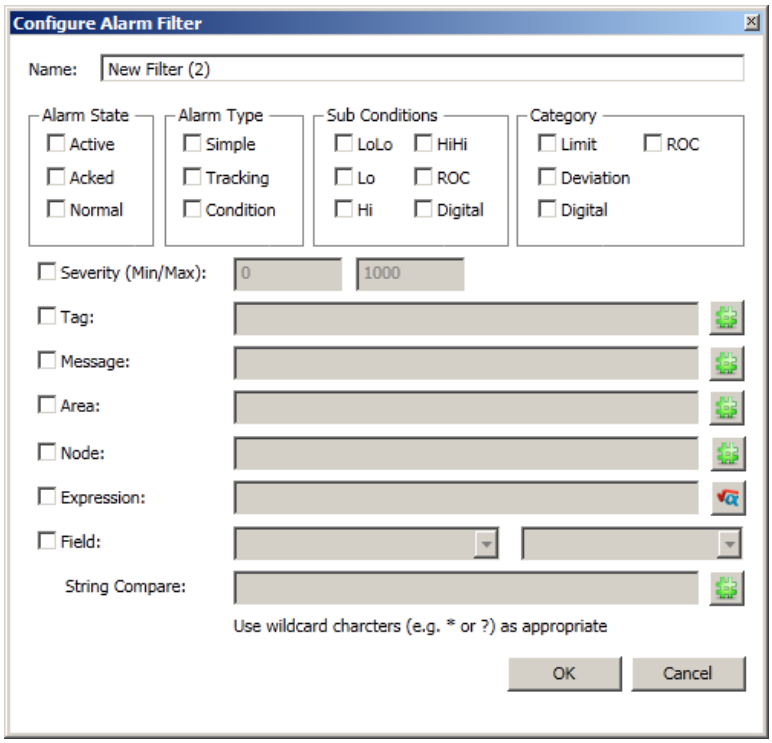


Figure 7.13 - Filter in the Edit Shared Filter Dialog

You can also enable **tooltips** for alarms. Remember that you can create help text in the alarm server. You can enable tooltips that display the help text configured in the alarm server to guide the operator when an alarm occurs.

You also specify alarm **acknowledgement** controls, including whether comments are required when an operator acknowledges an alarm. You can also apply the properties that you set for this grid to other grids.

7.1.4.3 Status Indicator Tab

The **Status Indicator** is a special feature of AlarmWorX64 Viewer that shows key performance indicators (KPIs) in the AlarmWorX64 Viewer grid. Status icons are a good way of providing a quick way for operators to gauge the alarm conditions that they see on a display.

A status indicator consists of a set of icons you configure for the indicator, such as a set of arrows, flags, faces, symbols, and anything else you want. During runtime, these icons swap in and out to give the operator an indication of an alarm's state. For example, as an alarm becomes more severe, the display shows a red arrow. As the alarm becomes less severe, the arrow displays first a yellow arrow and then a green one.

Each icon set is called a **dictionary**. In addition to the predefined dictionaries of status indicators, you can import additional sets.

As shown in Figure 7.14, each status indicator consists of a set of filters that define the levels of severity. Each filter is assigned its own image (that is, an icon from the selected dictionary) and a Show Value checkbox which, when checked, shows the original value of the cell together with the image.

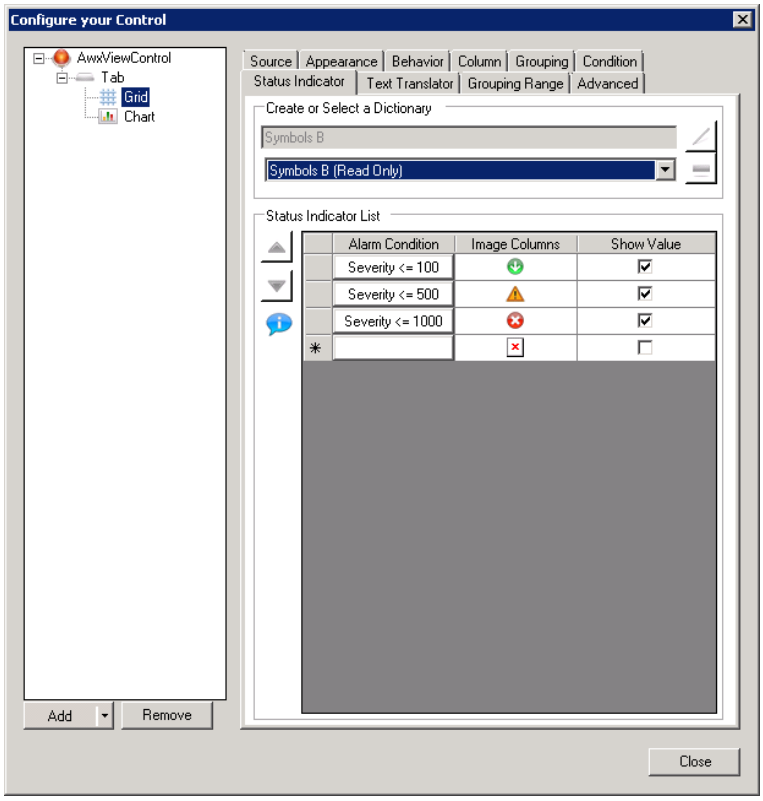


Figure 7.14 - Status Indicator Tab for a Grid

You can define different images to display using different conditions. You can define a condition by clicking on a cell in the **Alarm Condition** column. This brings up the **Edit Shared Filter** dialog as shown in Figure 7.13. In here, you can define the conditions required to display the indicator. When you are defining a condition for a status indicator, keep in mind that you can use any alarm attribute. For example, one filter could be **Severity > 100 AND Tag = 'Temperature'**. In this case, the filter's image will display for any alarm whose severity is higher than 100 and whose alarm tag name is 'Temperature'.

Once you have created a status indicator, you can go to the **Column** tab, as shown in Figure 7.15, to apply the status indicator to any column.

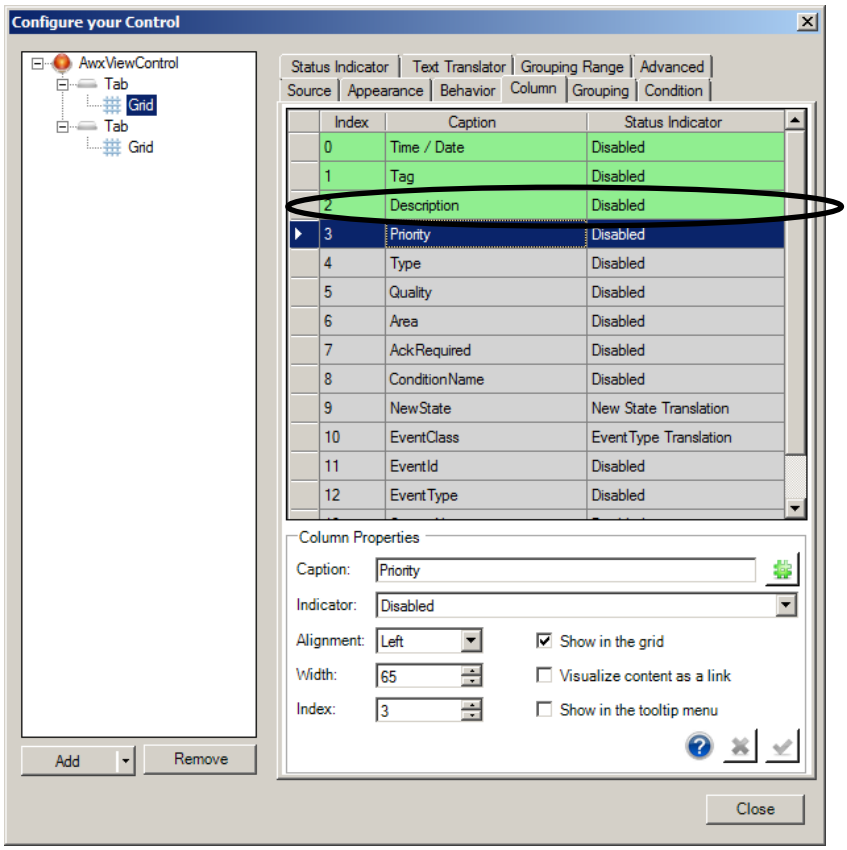


Figure 7.15 - Column Tab Showing a Column's Status Indicator

The result is shown in Figure 7.16. The images are applied to the cells of the column when the condition is true for the whole row.

Time / Date	Tag	Priority	Type	Quality	Description
10/4/2010 3:24 PM	Tank PSI	625	LOLO	Good	The PSI in Tank1 is too low.
10/4/2010 3:23 PM	Scale	700	LOLO	Good	Scale: Reading is low.
10/4/2010 3:23 PM	Radiation Level	800	HI	Good	No Radiation detected. System ok.
10/4/2010 3:23 PM	ROCTrigger	500	Rate of Change	Good	ROC on Analog1
10/4/2010 3:23 PM	Pump1	450	LOLO	Good	Pressure in pump1 is too low
10/4/2010 3:24 PM	Pressure	600	HI	Good	Valve 3 Pressure: normal
10/4/2010 3:24 PM	MMXVatLevel	500	HI	Good	fluid in vat number 1 HI Limit
10/4/2010 3:24 PM	MMXVatLevel	500	Digital	Good	fluid in vat number 1 is too high
10/4/2010 3:24 PM	MMXTankOverflow	500	Digital	Good	The main storage tank is overflowing
10/4/2010 3:24 PM	MMXSurfaceTemp	500	Digital	Good	Surface Temperature is too hot
10/4/2010 3:24 PM	MMXStructuralIntegrity	500	HI	Good	Structural Integrity HI Limit
10/4/2010 3:24 PM	MMXStructuralIntegrity	500	Digital	Good	Structural Integrity has failed
10/4/2010 3:24 PM	MMXSecurityBreach	500	HI	Good	HI Limit
10/4/2010 3:24 PM	MMXSecurityBreach	500	Digital	Good	Possible Intruder in section 7-G
10/4/2010 3:24 PM	MMXRadiationLevel	500	HI	Good	Radiation Level is HI Limit
10/4/2010 3:24 PM	MMXRadiationLevel	500	Digital	Good	Radiation Level is too high

Figure 7.16 - Status Indicator in Runtime

7.1.4.4 Column Tab

The grid's **Column** tab, shown in Figure 7.17, allows you to select the columns or alarm attributes to display in the grid. When you select a data source for a grid, you automatically populate the grid with all of the attributes that are defined for that data source in the tab. Notice the labels listed in the Caption column – if you click on any of these fields, an ellipsis button becomes visible so you can browse for language aliases to be used, if you need to make the grid language sensitive for runtime operators.

The checkbox in the first column lets you show or hide the column in the grid. You can also change the order that the columns that appear in the grid. The top-to-bottom sequence of rows in this dialog box indicates their left-to-right sequence as columns in the grid. You can select a row, then click the up or down arrow to move it to its new position.

Also in the Column tab, you can edit any column's title, width, and status indicator. When you are done editing, be sure to click on the "Apply" button to update the grid. If at any point you are not sure of what to do, you can mouse over the "information" icon to get help.

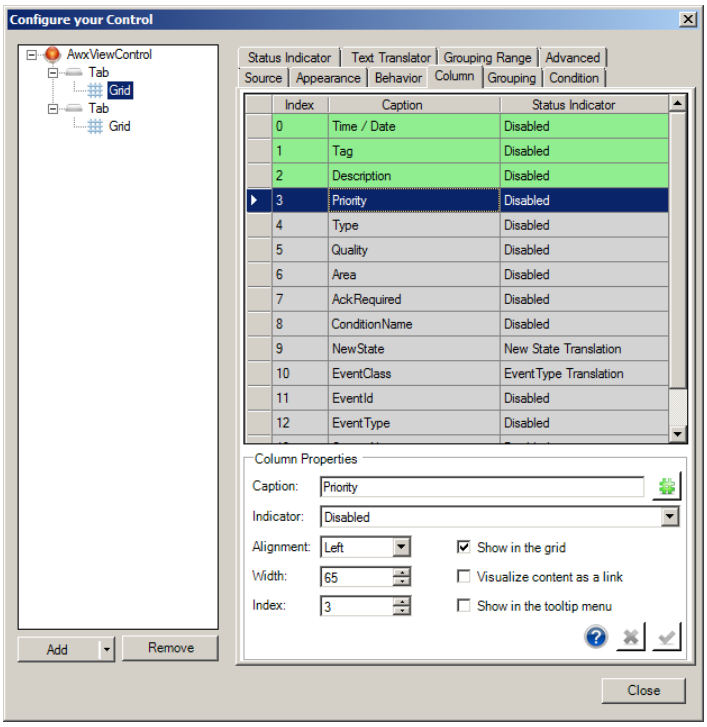


Figure 7.17 - Column Tab for a Grid

7.1.4.5 Grouping Range Tab

When there is much information in the viewer, you may want to group similar data so that it can be found it easily. To apply grouping, you will first need to set up how you want to group the alarms. The grid's **Grouping Range** tab (shown in Figure 7.18) allows you to set up a set of filters that you can apply to a column later, for proper grouping.

By default, there is one group range configured in the grid called "Severity Range" – it is a read-only range that you cannot modify. The items in this grouping range are filters that determine the severity of the alarms. In it are five different filters for alarms with different severity ranges to accommodate the different needs.

To create your own group range, you must select "New" in the dropdown list, and then you can type the name of the range in the Create or Select a Range textbox and click on the "Add" button. In the new Criteria List section, you can click in the Alarm Condition column to open the Filter dialog. You can use this dialog to create your filter; it is the same filter dialog used for status indicators, as shown back in Figure 7.13.

Once you create each filter, in the Caption column you can give a descriptive name for the filter. Add as many filters as you need.To apply your grouping range, you will need to go to the Grouping tab, which we will discuss next.

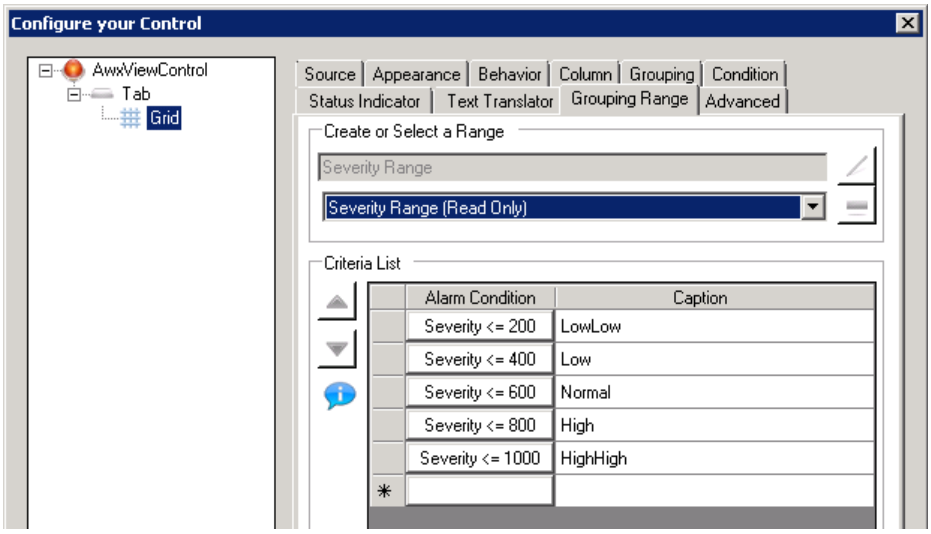


Figure 7.18 - Grouping Range Tab

7.1.4.6 Grouping Tab

Options on the grid's **Grouping** tab organize your alarms by values contained in the column that you apply the group function. Grouping allows you to view your alarms in a hierarchical view, which is helpful when analyzing alarms.

It is possible to create groups on more than one grid column. Grouping applies from the top down of the list on this tab. Following the example in Figure 7.19, the grid will first group by the Source Name of the alarm, then by the Time/Date, and lastly by the Event ID. You can change the order of the grouping by selecting an item in the list, then clicking on the up or down arrow.

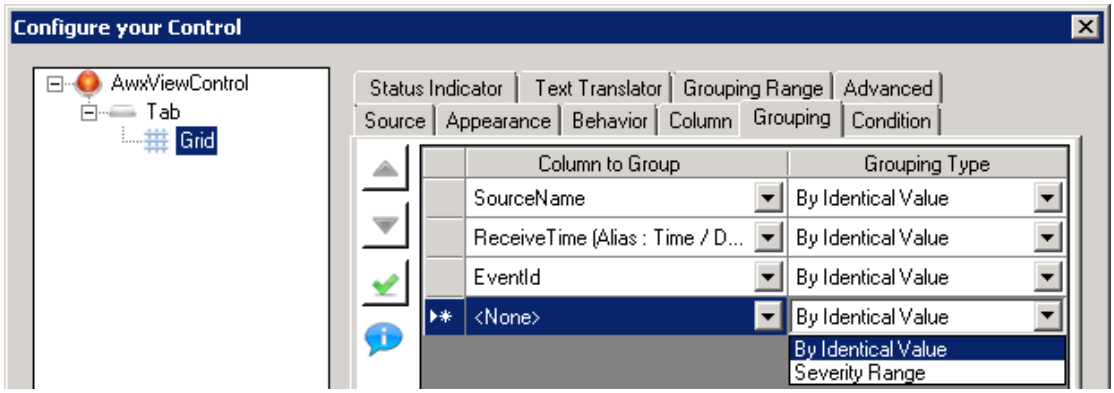


Figure 7.19 - Grouping Tab

Once you decide on *which* columns to group, you need to decide *how* to group the columns. In the Grouping Type section, you can choose how to group the columns. For example, in Figure 7.19, we are grouping the SourceName column by Identical Value (which is a default grouping type). Therefore, by default you have two grouping types: By Identical Value and Severity Range. You can also select other grouping types that were defined using Grouping Range tab.

When you are done editing, be sure to click on the "Apply" button to update the grid. If at any point you are not sure of what to do, you can mouse over the information icon to get help.

7.1.4.7 Condition Tab

In the grid's **Condition** tab, you can configure each alarm based on a condition. The most popular way to use this is to filter on the alarm priority (which is OPC-defined to have a value from 0 to 1000). An alarm that rises above a certain priority may require immediate action and you might want such an alarm to appear with a bold color, while other alarms that fall below a certain priority can be ignored and you might make the background of those alarms use lighter colors.

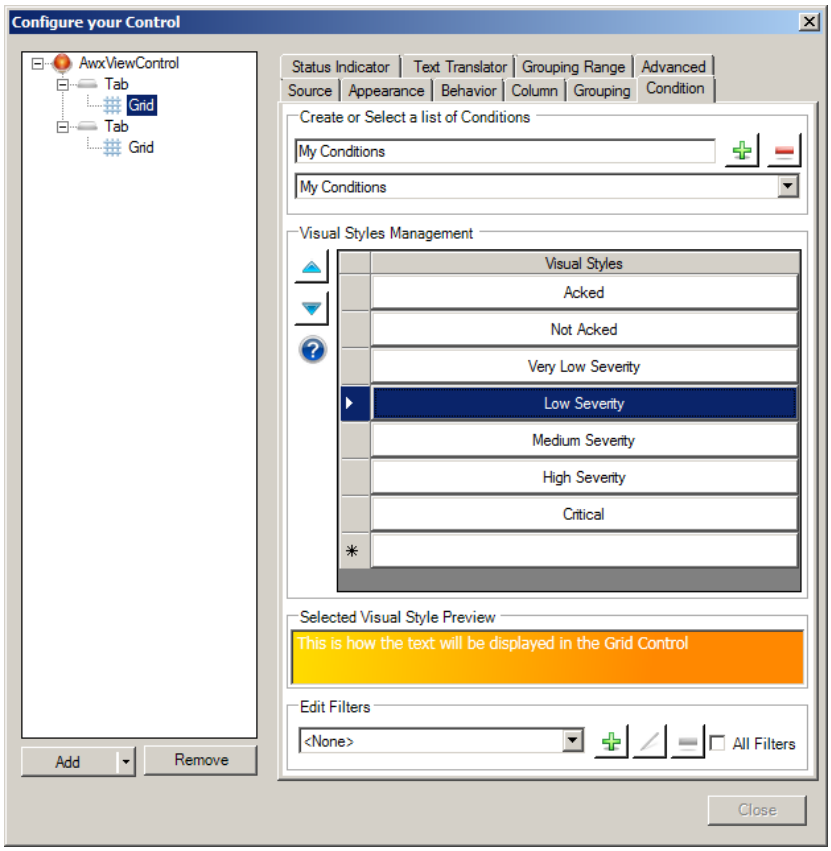


Figure 7.20 - Condition Tab

A condition is made up of a filter and a visual style. In runtime, if the filter is true for a row, then the related visual style is applied to the row. You can create as many conditions as you have filters and visual styles. You choose what the values are, what the condition level is named, and how alarms are displayed in the grid at runtime.

You can select a list of conditions or create a new list of conditions. Once you select a condition, the **Visual Styles Management** section lists the different conditions. If you create a new condition list, this section will be empty.

Clicking on a cell in the Visual Styles column brings up the Configure Styles dialog. There are two tabs in here: the Alarm Condition tab and the Visual Style tab. We will not show the Alarm Condition tab here for it looks exactly like Figure 7.13. The Visual Style tab, shown in Figure 7.21, provides options for color, blinking, and beeping – all ways to get the runtime operator's attention when an alarm condition occurs.

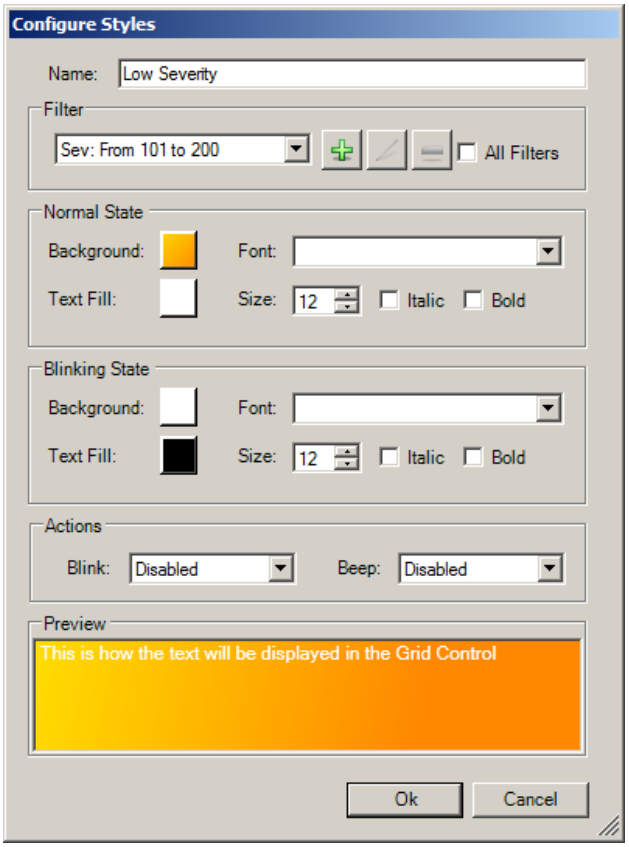


Figure 7.21 - Configure Styles Dialog for Conditions

7.1.4.8 Text Translator Tab

Alarm grids and charts created using AlarmWorX64 Viewer display values that may or may not make sense to the runtime operators who will be working with alarms. To make values more meaningful, you can translate them into words using the **Text Translator** tab, shown below. A set of translated values is called a *dictionary*. You can use translation dictionaries so that translated text appears in alarm grid columns and in chart labels.

ICONICS provides predefined, read-only translation dictionaries that define meanings for New State field values, Event Type field values, Category field values, and Quality field values. For any other field values, you must define your own dictionary using the grid's Text Translator tab, shown in Figure 7.22.

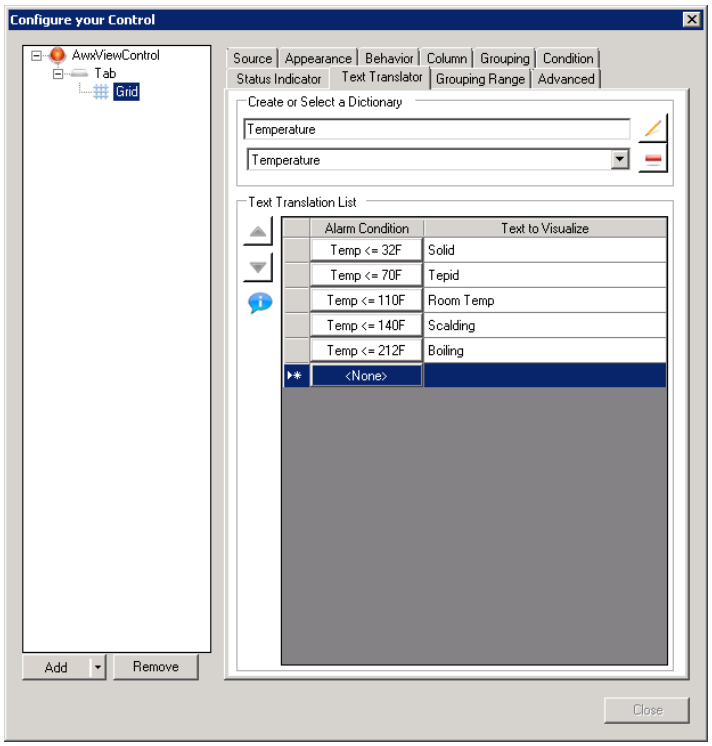


Figure 7.22 - Text Translator Tab

When creating a new dictionary, plan the values you will be creating with the dictionary. Then in the AlarmWorX64 Viewer Configurator, select the grid or chart from the tree and go to the Text Translator tab. You can select an existing dictionary, but you cannot modify the read-only dictionaries that are provided to you by ICONICS. To create a new dictionary, in the Create or Select a Dictionary box, select "New" from the dropdown list. In the same box, go to the field above your New selection and type the name of the new dictionary. Give it a name that aptly describes its use and the values it translates.

Now you can enter translated text for each alarm condition you want to put into the dictionary. A condition is a value or range of values that you want to translate into a word or words. To add a condition you click the empty box under Alarm Condition, which opens the Filter dialog box that you are now so familiar with (the one shown in Figure 7.13). When you return to the Text Translator tab, next to the filter you can enter text that the runtime operator will see.

7.2 Alarm Viewer Runtime Options

Once you are done configuring the AlarmWorX64 Viewer, you can go into runtime and view your alarms with all of the conditions, behaviors, styles, and other properties you have applied. There are a few tools that runtime operators can use to control which alarms display.

If you right-click on the alarm viewer in runtime, you will see a menu like the one shown in Figure 7.23. Depending on your alarm subscription of real-time alarms or historical alarms, different options may be available for use.

At a basic level, this menu lets runtime operators acknowledge alarms, sort columns, and filter data if the grid is displaying real-time data. If you are viewing historical data, you may have options to refresh the data, create a condensed alarm report or a detailed alarm report, and change the time range of the data that shows in the grid.

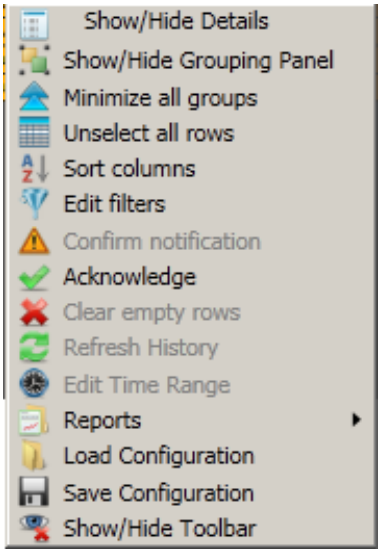


Figure 7.23 - Alarm Viewer's Right-click Runtime Menu

At the top of this menu, you have an option to Toggle Toolbar Visibility. If you click on this option when the toolbar is invisible, a runtime toolbar that comes with the alarm viewer will appear.

This toolbar, shown in Figure 7.24, can be free floating or docked within the alarm viewer. The toolbar gives many options such as sorting and filtering alarms like the right-click menu.



Figure 7.24 - Alarm Viewer Runtime Toolbar

Runtime operators can use either the right-click runtime menu or the runtime toolbar to configure the view of the alarm viewer.

Lab: Viewing Alarms L7

7.1 Lab Overview

In the previous lab, you created some alarms that indicate the status of the churning blade and the tank level for each of the tanks. These alarms are meaningless unless someone can see them when they go into an alarm state. IcTonics would like a display in which an operator can easily spot important alarms and manipulate (or acknowledge) them.

Estimated time to complete this lab: 20 minutes

7.1.1 Task Checklist

In this lab, you will complete the following tasks:

- Set up a simple alarm file in the Alarm Container
- Set up an alarm control in GraphWorX64
- Configure an alarm subscription
- Create and apply a client-side filter
- Add flash to alarms with a condition

7.1.2 Support Files

To complete this lab, you will need the following file(s):

- icTonicsAlarmServerConfig database
- icTonicsTankMonitor.gdf

7.2 Alarm Container

GENESIS64 provides a place where you can view alarms. We call this the AlarmWorX64 Viewer container. You can open this container within the Workbench or GraphWorX64.

1. Open the Workbench if you closed it from the previous lab. In the Project Explorer tree control on the left, select GraphWorX64. Expand the Training folder and double-click on the **icTonicsTankMonitor** to open the display in the configurator. Click on "Runtime" in the upper right corner.

2. In the Project Explorer tree control on the left, select **AlarmWorX64** (not the AlarmWorX64 Server or the AlarmWorX64 Logger).

3. Create a new **Alarm View** by right clicking on "**My Computer**" in the tree control and selecting "**New Alarms**" in the menu. This should create a tab to the right of the icTonicsTankMonitor.gdfx tab called "New Alarms".

4. The ribbon should now reflect the AlarmWorX64 Configuration menus. Click on the "**Configure Viewer**" button in the upper left-hand corner. 

This will launch the "Configure your Control" dialog box as seen in Figure L7.1.

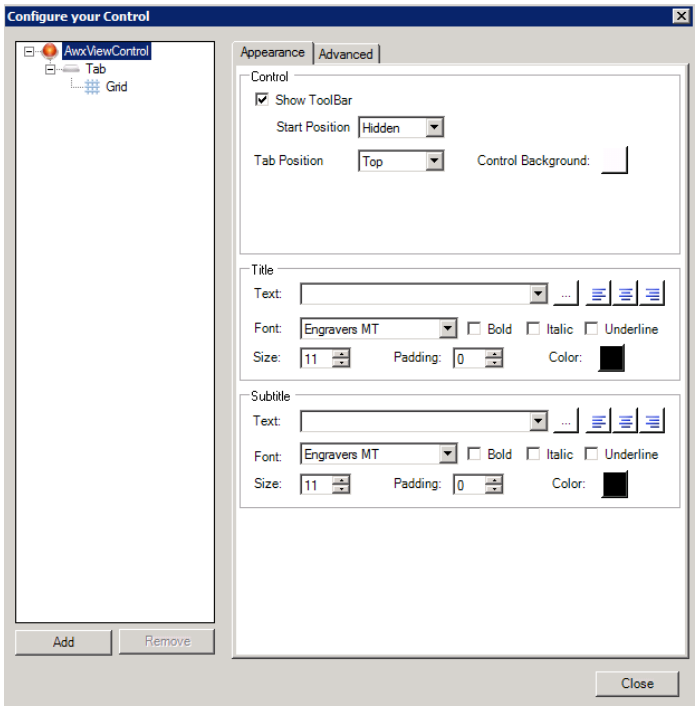


Figure L7.1 - Alarm Viewer Control Configure Dialog box

5. In the **Configure your Control** dialog box, click on the Grid selection in the left-hand pane. In the right-hand pane, under the **Create/Remove subscriptions** section, click on green plus sign (add button) to create a new real-time subscription. For this exercise, in the field just below the header. We can leave the name "New Subscription". Click on "Close".

6. Click on “Runtime” in the upper left-hand corner of the alarm view tab to see your alarms. You can easily go back to configuration mode by clicking the “Configuration” button in the same place.

7. We will generate alarms by clicking on the **icTonicsTankMonitor.gdfx** tab and clicking on the churning blade radio button. This in turn will place the churning blade in and out of alarm.

8. We will view the alarms by selecting the **New alarms** tab. Right click on the alarms in the viewer. Feel free to experiment by selecting various options (Acknowledge for instance) in the drop down. Please note, you can select multiple alarms while holding the ctrl or shift key and selecting the alarms with the mouse. Check out the reactions of the alarms as you turn the churning blade on and off.

9. The configuration that you created can be saved by clicking on the “Save” button in the ribbon or by clicking the “File” button left to the “Runtime” button on the bar right above the alarm viewer. The file extension will be “.awxx” or “.awxxs” (for Silverlight files).

> **Note:** Do **not** click on the “Runtime” button in the upper right-hand corner of the Workbench because this button is designed for the whole Workbench. It will change the layout of the working environment when it goes into runtime. So it is strongly recommend that you always use the provider’s runtime rather that Workbench runtime for your convenience.

7.3 Alarm Control in GraphWorX64

The alarm control is what you use to view alarms in GraphWorX64. It has the same properties as the Alarm Container. We will talk more about the alarm control in a later chapter. For now, think of it as an alarm container that you can use in GraphWorX64. In this section, you will create a display to view alarms as well as customize the control to suit the project.

1. Close any of the AlarmWorX64 files that are open in the Workbench. But leave the **icTonicsTankMonitor.gdfx** tab open.

2. From the Project Explorer, select GraphWorX64. Right-click on "My Computer" and select New Display.

3. Save the display in the C:\Training folder and name the display **icTonicsAlarms.gdfx**. You should save your work periodically as you progress through the lab.

4. Apply the template to this display by clicking on the Design tab and selecting the template you added in a previous lab.

5. Go to the **Controls** tab of the ribbon. Click on the **AlarmWorX64 Viewer** and draw the viewer into the GraphWorX64 workspace. You can resize the viewer to take up the majority of the empty space in the display.

6. Double-click on the AlarmWorX64 viewer control to bring up the **Configure Your Control** dialog box.

7. In the **Configure your Control** dialog box, click on the **Grid** selection in the left-hand pane. In the right-hand pane, under the "**Create/Remove subscriptions**" section, click on the green plus sign (add button) to create a new, real-time subscription. In the field just below the header, replace the "New Subscription" text with "icTonicsAlarms".

8. Notice the "Local Alarm Server" definition in the "**Add/Remove Event Points**" section. This confirms that the Local Alarm Server event point is already inside this subscription.

9. Select the Appearance tab, and then in the **Grid Style** section, select the "**Glass style**".

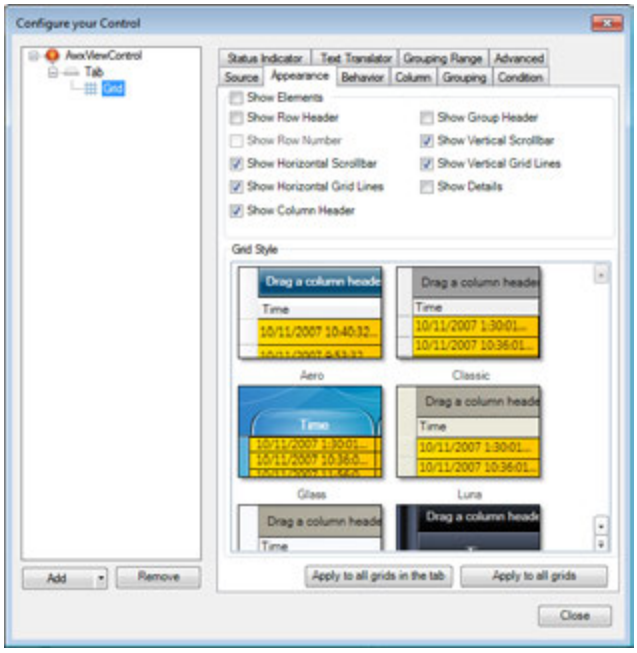


Figure L7.2 - Selecting a Style for the Grid

10. Go to the **Column** tab and move the **Description** field to just below the **Tag** field. You can do this by selecting the Description field row, and changing the **Index** number to 2. Also change the **Width to 250** and click on the "**Apply the changes**" button to save the changes you have made.

11. For the **Type** and **Quality** fields uncheck the property "**Show in the grid**" and apply these changes. Your Column tab should look similar to Figure L7.3.

Note: Until you apply the changes, the "Close" button at the bottom of the dialog box may be grayed out.

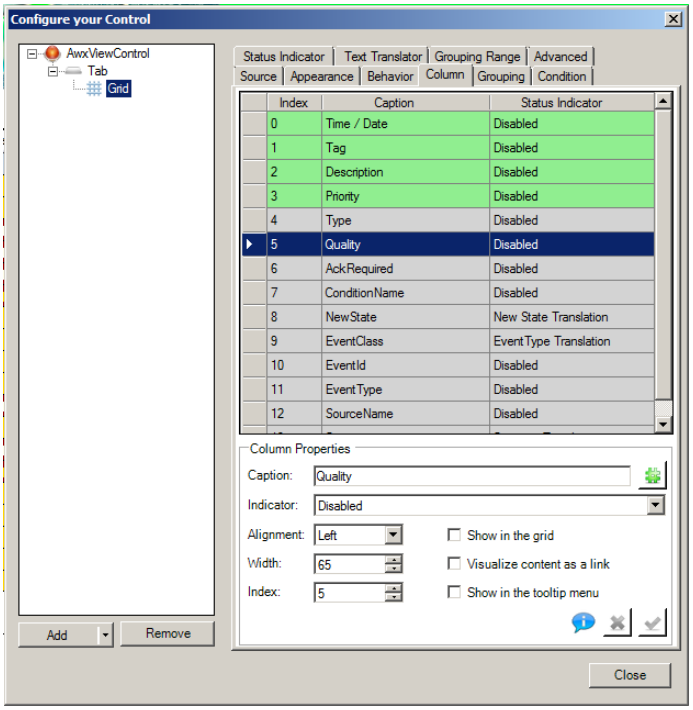


Figure L7.3 - Alarm Viewer Configuration: Column

12. Click on the Condition tab. Notice that there are preconfigured conditions here; we cannot modify these because they are read-only. We will create a new list, and then add a new condition for alarms with a severity of 900 or higher. The conditions are applied from the top down.

13. Click on top right "Add" (green plus sign) button to add your list of conditions. In the Name: field (top textbox), type in "**icTonics Conditions**".

14. Click on the blank alarm condition at the bottom of the list (this is the row with the * in the first column) to bring up the **Configure Styles** dialog box. Give the filter a name of "SeverityBelow900".

15. Click on the "Add" (green plus sign) button in the **Filter** section. The **Configure Alarm Filter** dialog box will now be displayed.

16. Name the filter "**SeverityBelow900Filter**", select the **Severity** checkbox and in the second data entry, enter a maximum value of "**900**".

17. Your "Configure Alarm Filter" tab should look similar to Figure L7.4. Then click on "OK" to confirm this filter.

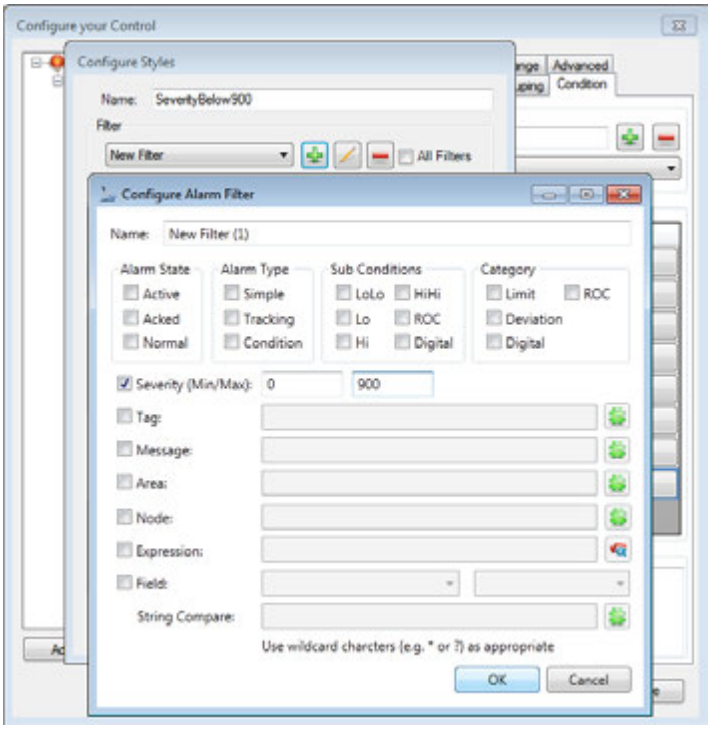


Figure L7.4 - Configure Style: Alarm Condition

18. In the **Blinking State** section, change the **Text Fill** color to Yellow and the **Background** color to a gradient. (Or you can pick your own colors.)

19. In the **Actions** section, change **Blink** to **Slow**. Click on "OK" to close out of the configure style dialog box.

20. Using arrows on the left side, move the new condition that you have just created to the one below the High Severity condition. If you check (edit) the High Severity and Critical conditions, you can see that their severity levels are set to 800 and 1000 respectively.

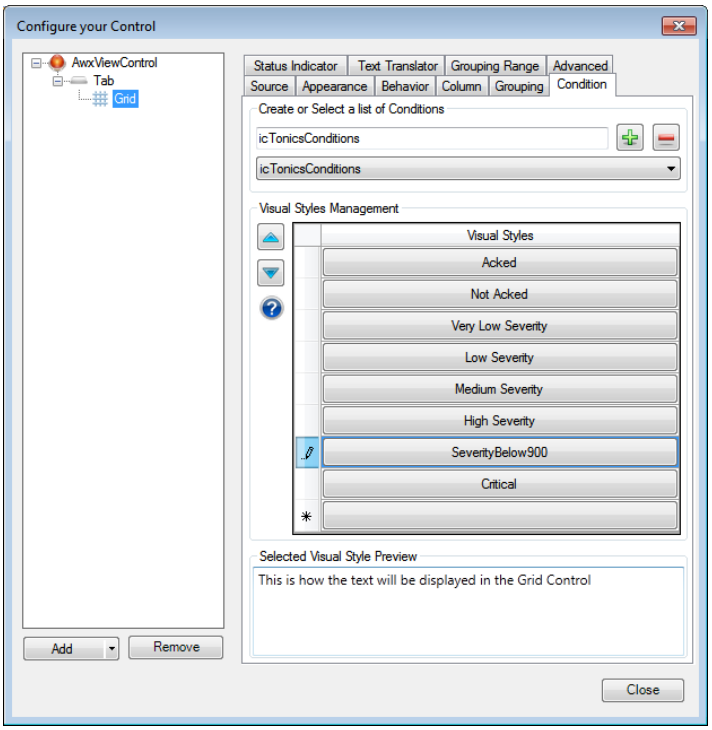


Figure L7.5 - Configure Your Control: Condition

21. Click "Close" to close out of the configuration dialog box.

22. Save the display and go into runtime. You may notice that you already have alarms.

23. Right-click on the viewer and select "Show/Hide Toolbar" to bring up the toolbar. Click on the "Filter Data" icon. We will now apply a filter to view the Tank level alarms only.

24. Click on "**Add**" button (plus sign) to create a new filter. In the "**Configure Alarm Filter**" dialog box, change the Name to "Tank Levels".

25. Check the Tag checkbox and write inside "***Level***" without quotes. (The word Level is surrounded by the wildcard *.) When this filter is applied, only alarms with the word "Level" as part of the tag name will be visible.

26. Your filter should look similar to Figure L7.6. Click on "OK" when you are done.

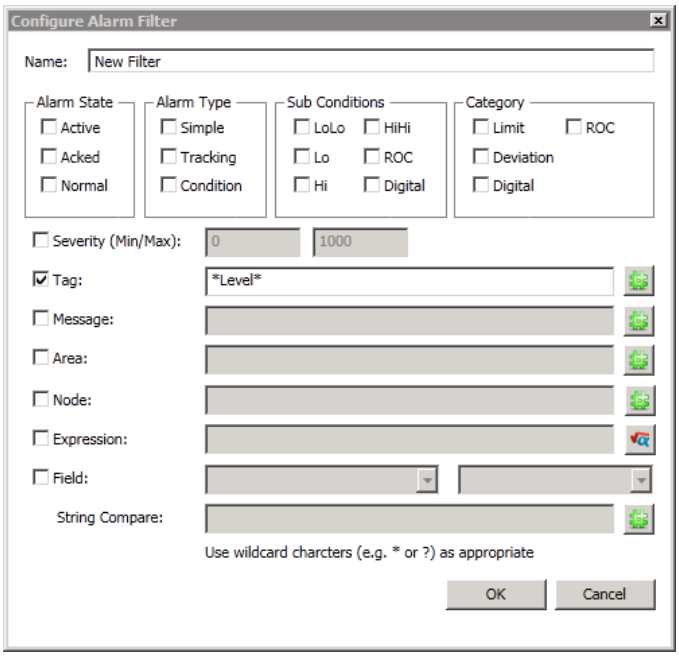


Figure L7.6 - Alarm Filter

27. Select the "**Tank Levels**" filter and click on "OK".

28. Return to the **icTonicsTankMonitor.gdfx** tab and set the tank level to 99. This will generate a HiHi alarm.

29. Click on the **Real Time Alarms** tab to view your alarms.

30. Take a minute to experiment and familiarize yourself with the alarm viewer control.

7.3.1 Tank Monitor Alarm Viewer

In this section, we will add an alarm control to the Tank Monitor display.

1. If you closed the icTonicsTankMonitor.gdfx open it in another instance of GraphWorX64, leaving the icTonicsAlarm.gdfx open. Enter configuration mode on both displays.

2. Within the icTonicsAlarms.gdfx select the Alarm viewer control. Using the Ctrl-C key sequence, copy the alarm viewer control. You could also use the right mouse key and select Copy.

3. Select the icTonicsTankMonitor.gdfx. With the display visible use the Ctrl-V key sequence to paste the alarm viewer into this display. You could also click the right mouse key and select Paste.

4. Resize the alarm viewer control so that it fits below the mixer tank. Double-click on the control it to open the properties.

5. Select "Grid" on the left-hand pane and click on the "**Behavior**" tab.

6. Click on the "Add" button next to the "**Select default client side filter**" field to bring up the **Configure Alarm Filter** dialog box.

7. Name this filter "**Tanks**".

8. Check the Tag checkbox and write inside "***<#AlarmFilter#>***" with quotation marks. The syntax signifies that whatever is between the delimiters <# #> is a global alias. We'll elaborate on this in the global aliasing chapter.

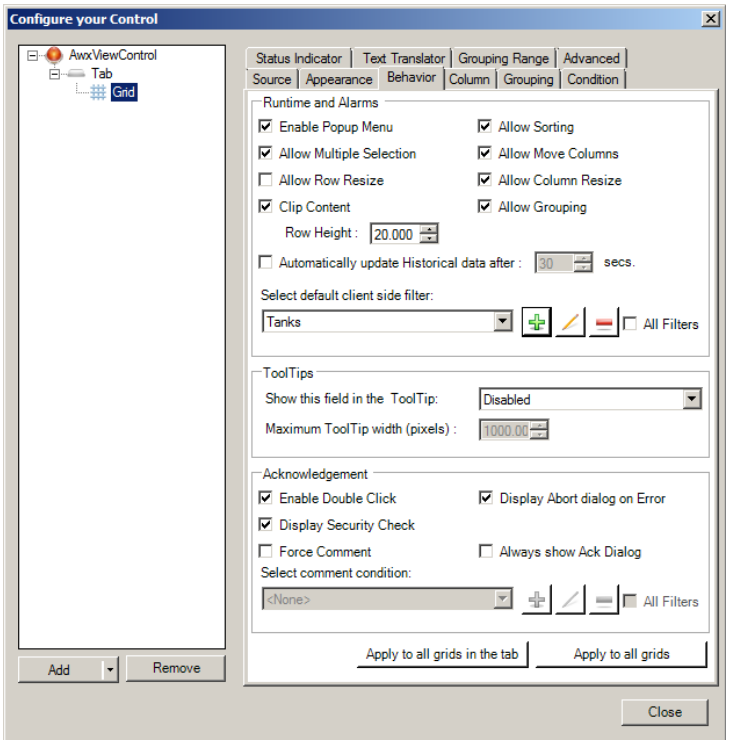


Figure L7.7 - Tank Filter with Global Alias

9. Click on "Close" to close the **Configure Your Control** dialog box.

10. Save the display. If you go into runtime now, no alarms will show because we have not defined the global alias <#AlarmFilter#> in the alarm filter yet. We will do this in a later lab.

Review: AlarmWorX64 Viewer R7

7.1. Objectives Review

You should now know and understand the following topics:

- The alarm viewer control
- Tabs, grids, and charts in the alarm viewer
- Grid Configuration control
- Sources, appearance, and behavior
- Status indicators and text translators
- Grouping ranges and grouping
- Columns and conditions
- Alarm viewer runtime options

7.2. Summary

The AlarmWorX64 Viewer is a client application that provides real-time and historical alarm information within the GENESIS64 product family. The viewer connects to OPC UA and OPC AE servers to provide alarm display and management features based on the connections to the data source that you specify.

In the AlarmWorX64 Viewer, you determine which alarms display, what the display looks like, and how alarms are grouped and sorted.

The viewer can be used inside the Workbench application or you can drop the AlarmWorX64 control onto a GraphWorX64 canvas. The viewer comes with many personalization tools in configuration mode for you to set up runtime alarm behavior based on conditions, groupings, and status indicators. In runtime mode, a toolbar and right-click menu allow runtime operators to turn on filters, sort alarms, and acknowledge alarms, as well as other options.

7.3. Terms

Alarm Filter
Alarm Indication Dictionary
Alarm Subscription
AlarmWorX64 Viewer
KPI

7.4. Test Your Knowledge

In this chapter, you learned how to configure an alarm viewer and its options. Before you proceed to the next chapter, make sure you can answer the following questions.

1. What alarm server(s) can be subscribed to AlarmWorX64 Viewer?

    a. OPC UA servers
    b. OPC AE servers
    c. OPC DA servers
    d. All of the above
    e. A and B
    f. A and C
    g. B and C
    h. None of the above

2. True or False: AlarmWorX64 Viewer is an ActiveX application.

3. You can use the AlarmWorX64 Viewer in which of the following?

    a. GraphWorX64
    b. Alarm Container
    c. AlarmWorX64 in the Workbench
    d. All of the above
    e. A and B
    f. A and C
    g. B and C
    h. None of the above

4. True or False: The AlarmWorX64 Viewer can display only real-time alarms.

5. True or False: You can add client-side filters in the alarm viewer to limit the alarms that displays.

6. True or False: When adding a chart or grid to an AlarmWorX64 Viewer tab, you must start by adding an alarm subscription using the Status Indicator tab.

AlarmWorX64 Logger 8

The AlarmWorX64 Logger provides a permanent copy (database storage or printed copy) of alarm and event notifications. The alarm and event notifications are generated from any OPC Alarm and Events server, including the AlarmWorX64 Server. AlarmWorX64 Logger subscribes to an OPC Alarm and Events server and stores the information in a database. You can configure the AlarmWorX64 Logger within the Workbench, which also allows starting and stopping of the Logger service. AlarmLogger64 configuration information is stored in SQL Server database.

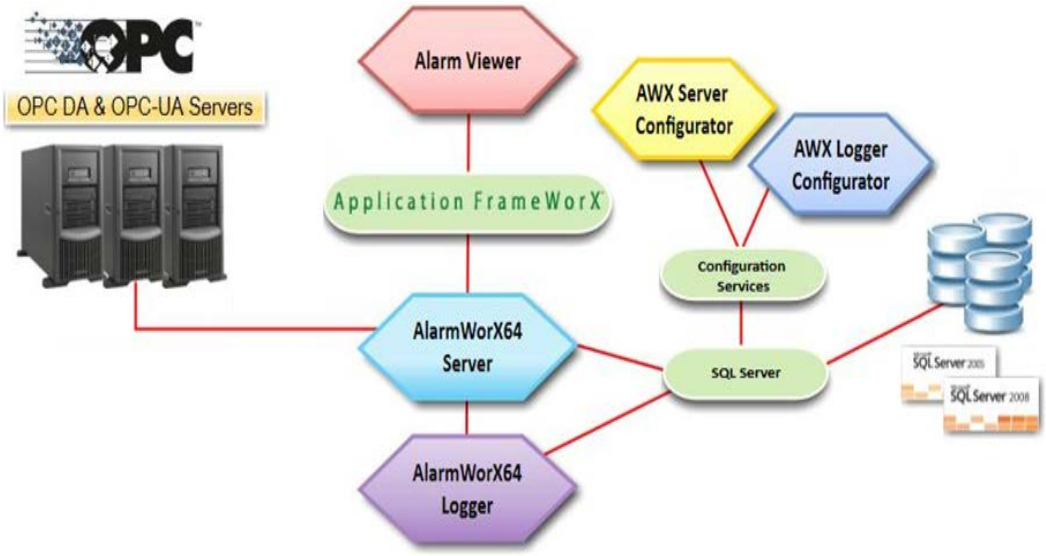


Figure 8.1 - Alarm Logger Architecture

After completing this chapter, you should know how to:

- Start up AlarmWorX64 Logger Configurator and services
- Configure an AlarmWorX64 Logger database connection to log alarms
- Add server-side and client-side filters for alarm logging and column definitions
- Create a node configuration

8.1. Starting the AlarmWorX64 Logger Configurator

There are two ways to start the AlarmWorX64 Logger configurator:

- From the Windows desktop, click on Start → All Programs → ICONICS GENESIS64 → AlarmWorX64 → AlarmWorX64 Logger Configurator.
- From within the GENESIS64 Workbench, select the AlarmWorX64 Logger button or icon.

Once the AlarmWorX64 Logger configurator and you will see a screen like Figure 8.2.

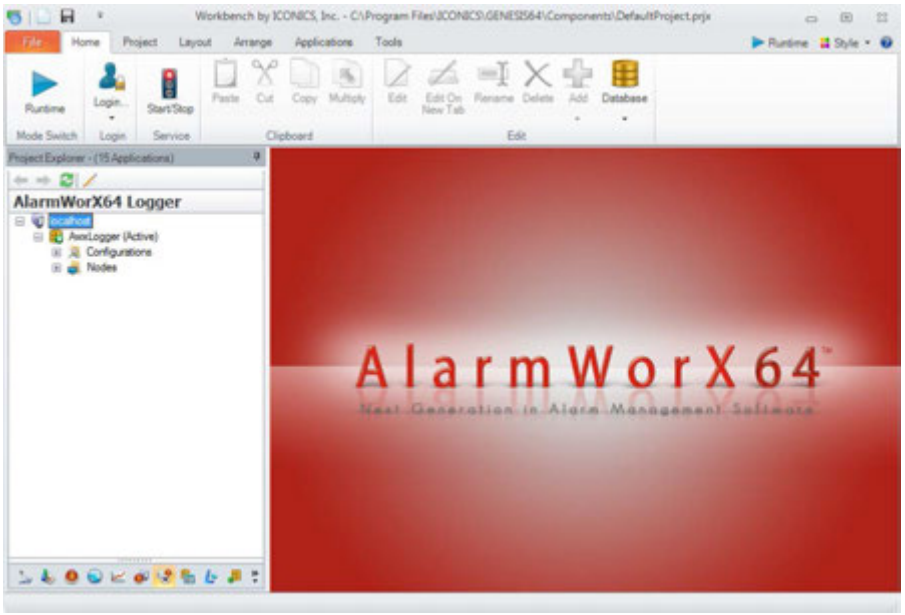


Figure 8.2 - AlarmWorX64 Logger Configurator

8.2. AlarmWorX64 Logger Configuration

Like many of the GENESIS64 applications, the configuration for the AlarmWorX64 logger is stored in an MS SQL database. You can create a new configuration database from the Home ribbon by clicking on the Database drop-down menu and selecting New Database. Alternatively, you can right-click on Localhost in the AlarmWorX64 Logger tree in the Project Explorer and select New Database.

The now-familiar Database Wizard will come up and guide you through the steps of creating your new alarm logger database. During setup of your new configuration database, there is an option in the wizard that allows you to include sample data. This option is checked by default. If you wish to exclude examples from this configuration database, uncheck the option before proceeding with the database creation.

8.2.1. Configurations

The configuration database can host many logging configurations. A logging configuration stores information, such as the database connection, to store the alarm and event data and the subscription to the OPC AE server.

You can create a logging configuration by right-clicking on **Configurations** under your database and then select the Configuration from the pop-up menu. Alternatively, on the Home tab of the ribbon, you can click on the Add button's down arrow and select **Configuration**.

When you do this, AlarmWorX64 Logger opens the Alarm Log Configuration tab, which displays the configuration properties, as shown in Figure 8.3.

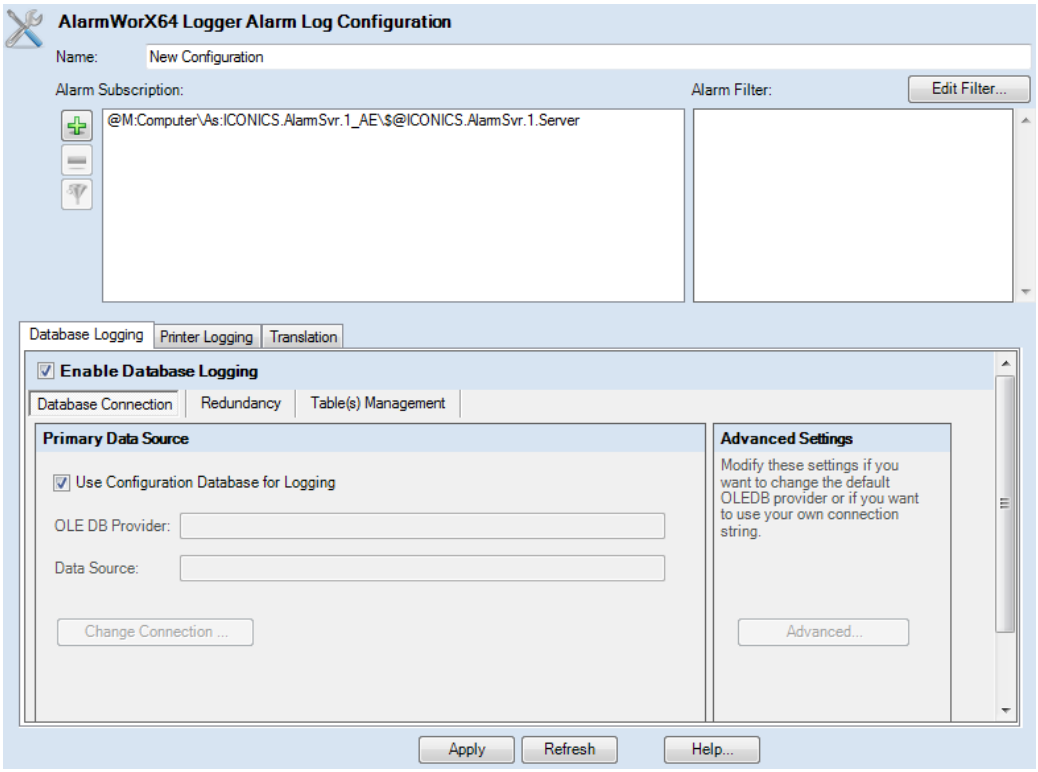


Figure 8.3 - New Configuration in AlarmWorX64 Logger

8.2.2. AlarmWorX64 Logger Subscriptions

Once you create a configuration, by default it is automatically subscribed to the local Alarm Server, as shown in Figure 8.4. If an additional subscription is needed, simply click on the "Add" button to create a new alarm subscription. The Data Browser window will appear, allowing you to select another alarm server and/or area.

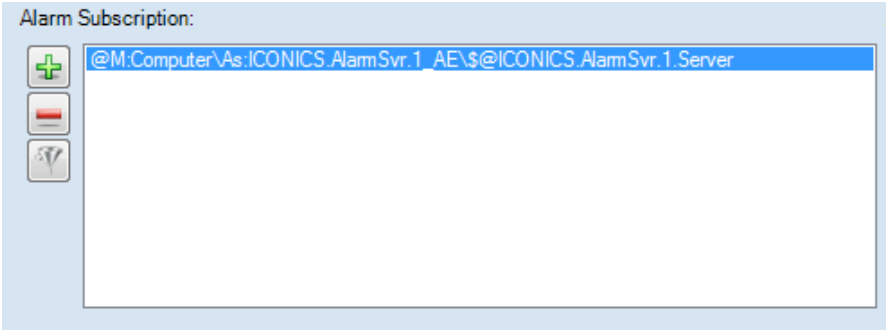


Figure 8.4 - Alarm Subscription

You can delete a subscription by highlighting it and clicking on the "Remove" button.

8.2.3. AlarmWorX64 Server-Side Filter

Once a subscription is created, you may need to edit it so that the subscription can get the data you want. This is called *server-side filtering*. A server-side filter sorts alarm data coming from the alarm server into the logger. Choose the alarm subscription you want to edit, and then click on the "Edit Categories Filter" button to open the subscription's properties dialog, which is shown in Figure 8.5.

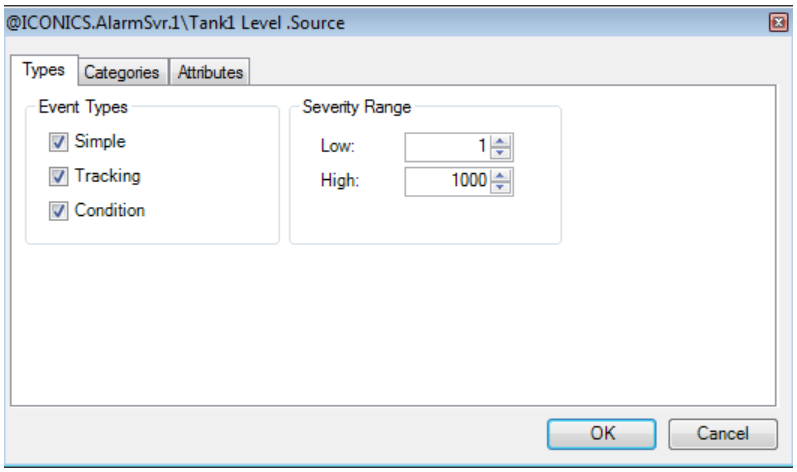


Figure 8.5 - Alarm Subscription Property Window

There are three tabs that are available here: Types, Categories, and Attributes. The **Types** tab allows you to choose or filter the event types and severity range of the alarms and events. The **Categories** and **Attributes** tabs allow you to subscribe to custom categories and to pick the attributes that you want logged. A general rule in the Categories tab is that if nothing is in the subscribed section, you will be subscribed to the available categories. However, if you add any categories to the subscribed side, then you will only receive alarms and event in those subscribed categories.

8.2.4. AlarmWorX64 Client-Side Filter

While server-side filtering restricts the information coming from the alarm server that the logger configuration subscribes to, *client-side filtering* limits the data that is logged into the database once you have data from the alarm server. In AlarmWorX64 Logger, this filter is located next to the alarm subscription, in the text box shown in Figure 8.6.

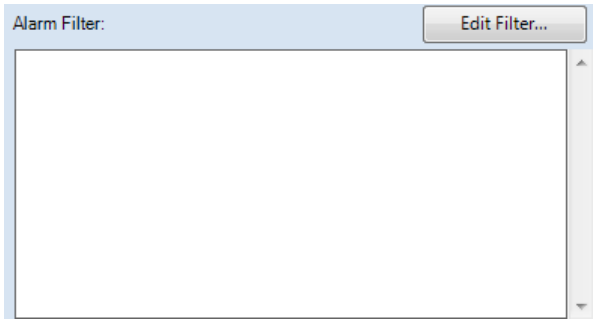


Figure 8.6 - AlarmWorX64 Logger Filter

You can start typing in the **Alarm Filter** box to use the Advanced Expression engine. Or you can click on the **Edit Filter** button to use a Filter Wizard or the Expression Editor.

Why two different types of filters? Use the *server-side filter* to save traffic resources between the Alarm Server and Alarm Logger. Use the *client-side filter* to save space and to restrict how quickly the log database grows. These filters become more important when the alarm server, the logger, and the database are on different machines and the end-user or runtime operator does not want to see all types of data from the alarm server getting logged.

8.2.5. Configuring the Database Connection

When you first create a configuration, the logger is automatically set to log all alarms into its own configuration database file, as shown on the Alarm Log Configuration's Database Logging tab. Figure 8.7 shows this tab with the Database Connection sub-tab on top. Notice that there are two additional sub-tabs: Redundancy and Table(s) Management.

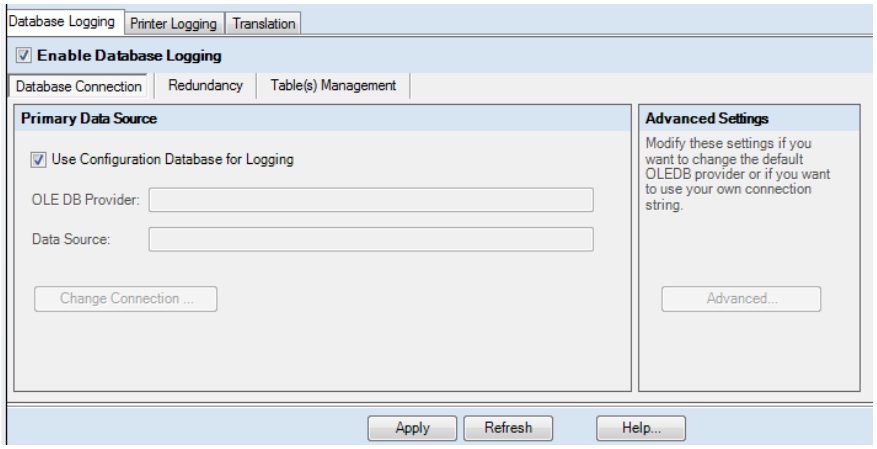


Figure 8.7 - AlarmWorX64 Logger Database Connection

If you want data logged to a different database, you will first need to uncheck the "Use Configuration Database for Logging" checkbox and configure the database connection by clicking on the "Change Connection" button. This opens the **OLE DB Connection** dialog box as shown in Figure 8.8, where you simply select the OLE DB Provider you want to use, its location, and the database to which the alarms will be logged. Click on the "Test Connections" button to make sure you can connect to the destination database successfully. When you have confirmation that the connection is good, click on "OK" to automatically populate the necessary fields on the OLE DB Provider and Data Source.

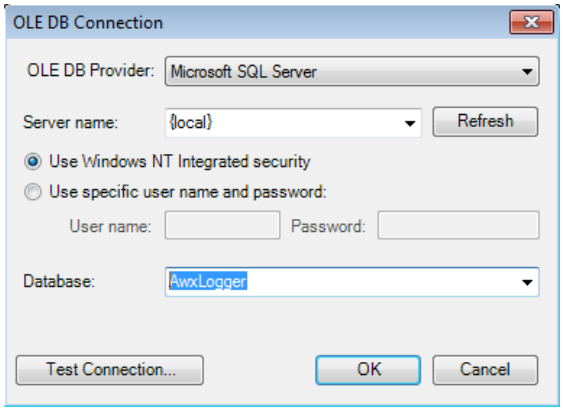


Figure 8.8 - OLE DB Connection

Although the connection string is automatically created when you configure the OLE DB Provider, you can modify it. If you need to modify it or manually create your own connection string to the database where the alarms will be logged, you can click on the "Advanced" button in the **Advanced Settings** section.

8.2.6. Table Management

Alarm data are logged to one master, active table in the database. If there are no restrictions placed on the size of the active table, data are logged to the active table as long as the Alarm Logger is running and there is disk space on the hard drive. You can, however, place a restriction on the size of the active table by using the **Database Logging** tab's **Table(s) Management** sub-tab, shown in Figure 8.9. The active table can be restricted to a maximum size determined by either a number of records or a time interval.

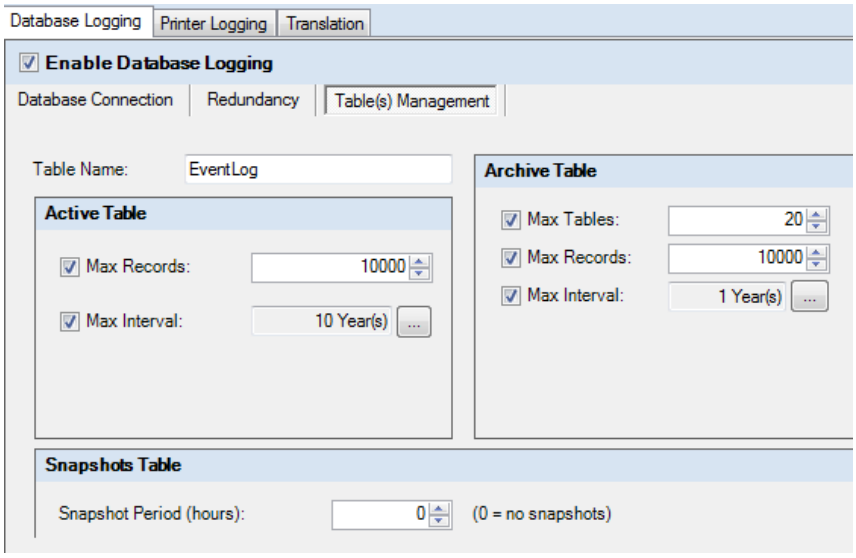


Figure 8.9 - AlarmWorX64 Logger Table Management

If the active table's size is restricted, all incoming data continues to be logged to the active table, but the oldest data that exceeds the restriction will be moved from the active table into an archive table. Archive tables can also have a maximum size determined by either a number of records or a time interval. When one archive table is full, data will be logged to the next archive table. When all archive tables are full, the oldest archive records will be replaced by the newer information. The name of each archive table is the name of the active table followed by the date and time of the first alarm in the table. With this system, the active table always has the most recent alarm data.

During runtime, the Alarm Logger will create the active table if it does not exist and will add any columns to the table if they are missing. The Logger will not create any indexes, but you can use a database administration tool to add indexes to columns that are searched and/or sorted by queries.

You can use the Table Management section of the Database Logging tab to configure the following parameters for the active table:

- **Table Name:** Enter the name of the active table.
- **Active Table:** Specify a limit on the size of the active table by putting a checkmark in one or both of the checkboxes. The limit can be a maximum number of records or a time limit on the age of data in hours, days, months, or years (the interval refers to the time span between the oldest and most-recent items logged). If *both* conditions are checked, when one of the conditions is met, the table is full. If neither item is checked, then archiving is disabled and will never take place.
- **Archive Table:** Specify the quantity of archive tables and a limit on the size for each archive table. These fields work the same way as they do for the active table, but with the following exception: If the **Max Records** and **Max Interval** fields are both left *unchecked*, all archiving will go to *one* archive table of a (theoretically) unlimited size, because the table size is not restricted. In essence, the **Max Tables** field will be ignored.
- **Snapshots Table:** This section allows you to create a "snapshot" table that you can use to replay alarm data. In the **Snapshots Period** field, specify how often (in hours) a snapshot table can be created. A snapshot period value of **0** (zero) means that no snapshot table will be created. For data replay to work with alarms, the value of the snapshot period must be greater than **0**.

  Note: You can print incoming alarm and event messages to a printer in addition to storing them to a database. For configuring the printer setup, see the "Printer Logging" tab. For additional details, please refer to the AlarmWorX64 documentation.

8.2.7. Printer Logging

On the Printer Logging tab, you can configure options for sending alarm events to a printer so that a written record is made of the log files. This works best with a dedicated ink jet or dot matrix printer, which prints one line at a time, rather than a page-oriented printer like a laser printer. Laser printers will still work, but the page will not be ejected until it is full.

Putting a checkmark in the Enable Printing checkbox makes the options available.

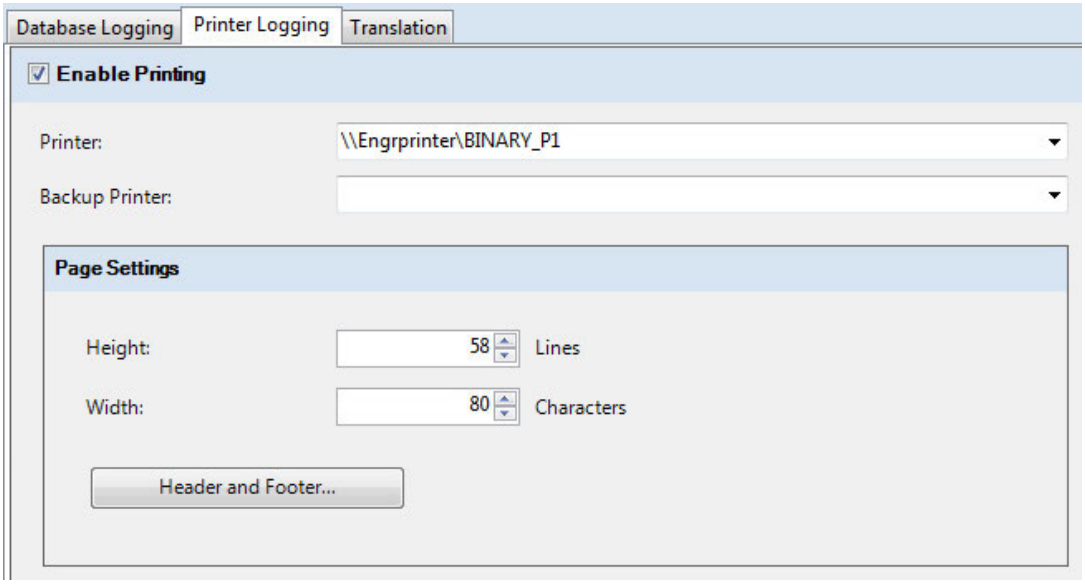


Figure 8.10 - AlarmWorX64 Logger: Printing Logger Events

You can select the primary printer and a back-up to be used if the primary becomes unavailable. Event notifications will print on the primary printer unless an error is detected (for example, if the printer goes offline or is out of paper), in which case the back-up printer will be used. A switchover from the backup to the primary occurs automatically when the error condition is cleared on the primary printer.

You can click the Heading and Footer button to specify a header for the top and/or footer for the bottom of each printed page. You can print using character sets other than the standard 7-bit ASCII codes appropriate for English, too. Instructions for setting up other character sets are provided in the online Help for GENESIS64.

8.3. Column Definitions

In the context of AlarmWorX64 Logger, alarm column definitions are the alarm attributes that the logger should take information from and put into the database. The alarm column definitions that you configure dictate the column headings in the table as well as which attributes of each alarm or event received will be saved in the database.

You can configure columns by right-clicking on the configuration in which you want to add a column for and select the " Column" option. This opens the Column Definition configuration as shown in Figure 8.11.

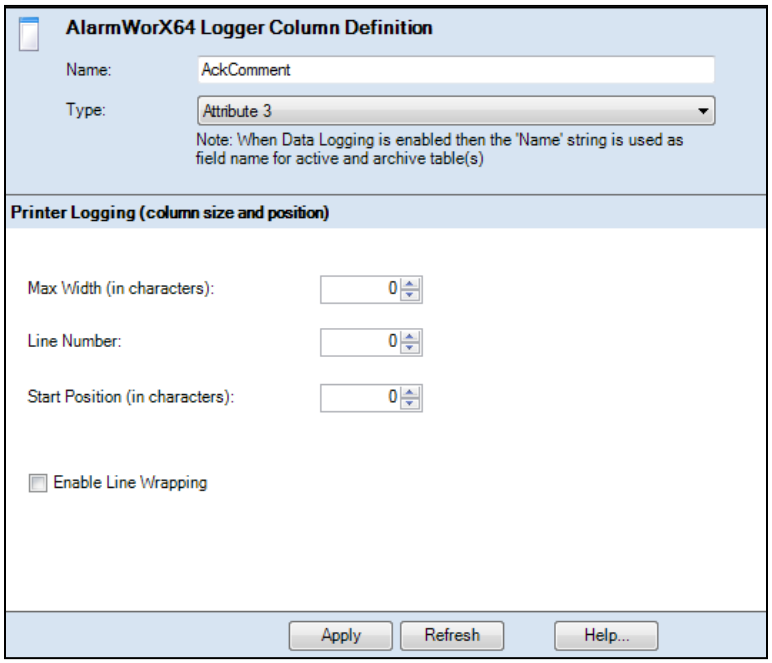


Figure 8.11 - Column Definition

In this interface, you can specify the name of the column, which will become the column name in the logging table. Then choose the alarm attribute from the Type field.

8.4. Nodes Configuration

A node is the PC network name associated with a particular configuration. When the alarm server starts, it checks the database for its node name. Unlike most other GENESIS64 applications, you can run multiple configurations at a time. However, you can have only one active configuration database at a time. It is because of the use of nodes that the alarm logger can run multiple configurations at a time. Any configuration you want to log data to must be associated with a node. You must add the node, and then you can associate the configuration(s) with it.

You can add a node, by right-clicking on **Nodes** and then selecting " Node" from the popup menu, as shown in Figure 8.12.

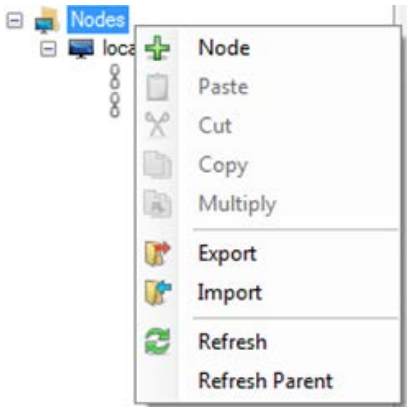


Figure 8.12 - Node Configuration

The New Node tab appears. Type the **Name** of the computer in which you want to use the configuration (which in most cases is your local computer name) or click the ellipsis button to choose it from a list, and then click on "Apply".

To associate a configuration with the node, right-click the node and select " Node Configuration". In the dialog box that appears, select the configuration you want to attach to the node and click on "Apply".

Note that when working locally on your GENESIS64 server, instead of specifying the name of the actual machine, you can use the neutral node name "anonymous" to run one or more alarm logger configurations on the local GENESIS64 node.

Lab: Alarm Logger *L8*

8.1. Lab Overview

Logging generated alarms can help in pinpointing problem areas on the production operation. Therefore, icTonics has decided to log the alarms for future evaluation and analysis purposes. This will also help icTonics keep track of the operators that acknowledge the alarms when we implement security.

Estimated time to complete this lab: 20 minutes

Task Checklist

In this lab, you will complete the following tasks:

- Create a new alarm logger configuration
- Create and connect to a database to log alarms
- Configure active and archive tables
- Associate the configuration to a node
- View historical alarms in the Alarm Viewer

Support Files

To complete this lab, you will need the following file(s):

- icTonicsAlarmServerConfig database
- icTonicsAlarms.gdfx

8.2. Alarm Logger Configuration

In order to log the alarms or events generated by an alarm server or event server, you will need to subscribe to an alarms or events OPC Server and configure the attributes that need to be saved in the logging database.

1. Open the Workbench if you closed it from the previous lab and select the **AlarmWorX64 Logger** from the Project Explorer.
2. Right-click on the **localhost** and select **New Database** to bring up the **AlarmWorX64 Logger – Create Database Wizard**. Click on "Next" to get to the common database options window.
3. Select the "**Include Sample Data**" checkbox and change the **Database Name** to "**icTonicsAlarmLoggerConfig**" and click on "**Finish**" to create the database.
4. Expand Configurations under icTonicsAlarmLoggerConfig. You should see two configurations. This is because when we initially created the database we left the "Include Sample Data" checkbox checked.
5. Then, expand **Nodes**, and also expand **localhost**. Depending on how your computer is setup, you may see your own computer name in the list instead of localhost. Notice the two sample configurations are associated with the local computer already.
6. Expand the **Configurations**, right-click on "Alarms", and select "Edit on New Tab". This will open the configuration options for the logger in the content area on the right.

> ***Tip!*** You can open anything from the Project Explorer by clicking on the item with the middle mouse (scroll wheel) button.

7. Notice that there is already an Alarm Subscription connected to the local alarm server. Change the configuration name to "**icTonicsAlarms**".
8. In the Database Logging tab, select the **Database Connection** page.
9. Uncheck the option **Use Configuration Database for Logging**, and then click on the **Change Connection** button.

10. The OLE DB Connection dialog box comes up. Leave the **Microsoft SQL Server** as the OLE DB Provider. Enter a Server name of "**(local)\SQLEXPRESS**" (without double quotes). From the **Database:** drop down list select "icTonicsAlarmData" as the database. Click on the **Test Connection** button. If the test is unsuccessful, review this configuration again. Your finished configuration should look similar to Figure L5.1.

    Note: Depending on how your machine is configured, the (local) string may need to be replaced with either a period "." or the computer hostname.
    E.g. .\SQLEXPRESS or hostname\SQLEXPRESS.

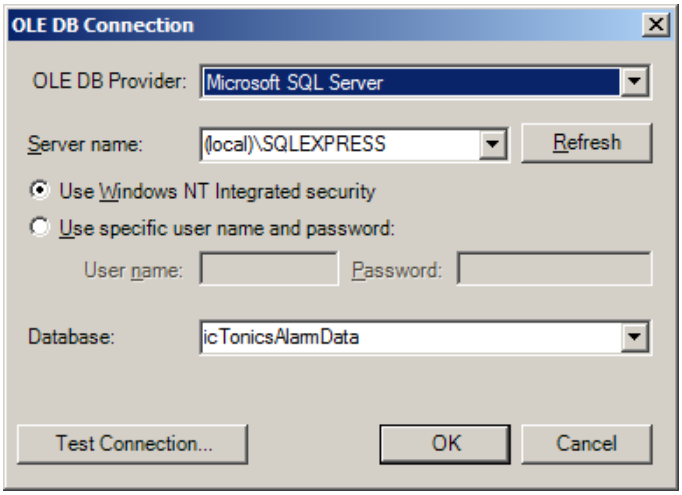


Figure L5.1 - OLE DB Connection

11. Click on "OK" to close the dialog box, and then click on the **Table(s) Management** page.

12. In the Active Table section, put a checkmark in the "**Max Interval**" checkbox and change the Max Interval to "**1 Day(s)**".

13. In the Archive Table section, put a checkmark in the "**Max Interval**" checkbox and change the interval to "**7 Day(s)**". This configuration will allow us to keep only one day's record in the active logging table, but archive seven days of alarm records into one table. Your Tables Management sub-tab should look similar to Figure L5.2 when you are done.

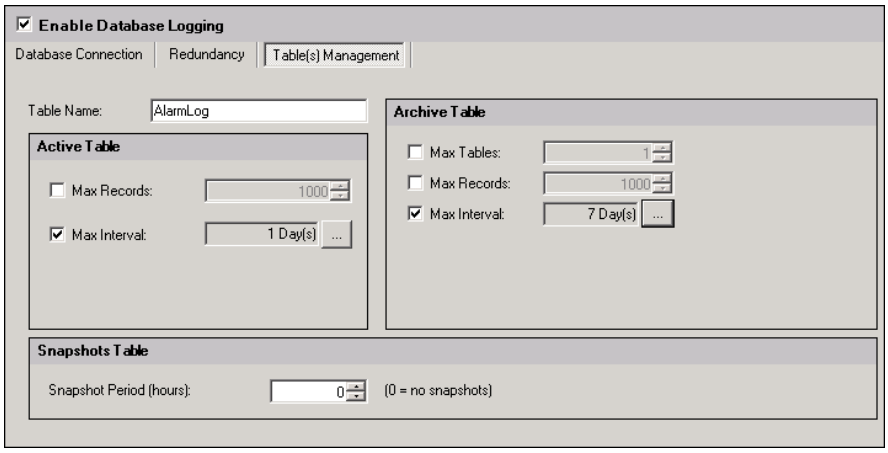


Figure L5.2 - Table(s) Management

14. Click on "Apply" to save your changes.

15. In the project explorer, expand the Node again. Select your localhost (or computer) name, right-click on it and select "**Edit on New Tab**". Change the localhost so that it uses your computer name. Click on "Apply" to save your changes.

16. We will now change the active configuration database. Stop the data logging service by clicking on the Start/Stop traffic light button in the Home ribbon. Wait for the light to turn red.

17. To make icTonicsAlarmLoggerConfig the active configuration database, right-click on the icTonicsAlarmLoggerConfig in the project explorer. From the drop down list select "**Make Active**". Now start the data logging service by clicking on the Start/Stop traffic light button in the Home ribbon.

8.3. Viewing Historical Alarms

Once you have logged your alarms into a database you can view these logged entries. Of course you can view it through a database tool, or you can use the alarm viewer control.

1. Open the icTonicsAlarms.gdfx display that you created in the previous lab.

2. Double-click on the **Alarm Viewer control** to open the configuration dialog box.

3. Select **Tab** in the navigation tree on the left. In the Title section, change the Text: to "**Real Time Alarms**".

4. Right-click on **AwxViewControl** on the left pane and select Add → Tab.

5. Select the new tab and change the Title text to "**Historical Alarms**". Notice that there is already a grid under this tab.

6. Select the **Grid** under **Historical Alarms**. We are going to set this grid up similarly to the one that you setup in the previous lab.

7. In the **Create/Remove Subscriptions** section, click the green plus button to create a new subscription. Select Historical Subscription. In the top textbox, replace "New Subscription" with the text "**icTonicsHistoricalAlarms**".

8. Notice that the Local Alarm Server is already in the **Add/Remove Event Points** section. Click on "**Local Alarm Server**" to select it, the detailed information of this Event point will be displayed in the Point Tag field.

9. Click on the **ellipsis** button next to the Point Tag to bring up the **Data browser**.

10. Click on **Home** and select **Historical Data → AlarmWorX64 Logger → icTonicsAlarms** and then double-click on the tag that starts with **@** as shown in Figure L5.3. Click on "**OK**" to exit out of the browser.

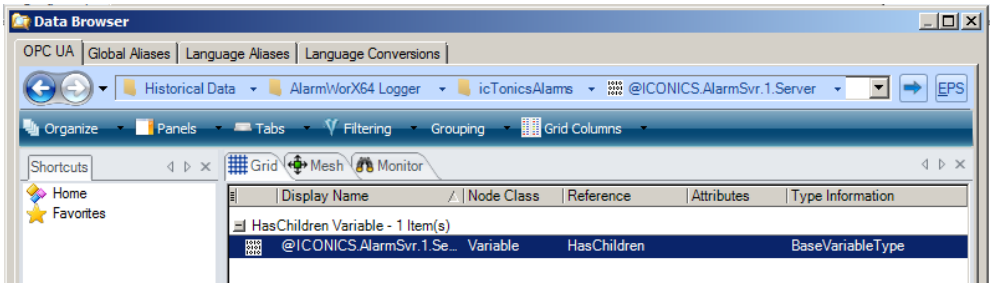


Figure L5.3 - OPC UA Tag for Historical Alarm Server

11. In the **Edit Point Properties** section, change the **Point Name** to "**Local Alarm Logger**". Note this also changes the name under the Add/Remove Event Points Section.

12. In the Add/Remove Event Point section Click the "Clock"  button. In the "**Edit the time range"** dialog box box, select "**Today**" from the drop down list. This will limit the alarms to show only today's records.

13. Your Source tab should look similar to Figure L5.4 when you are done.

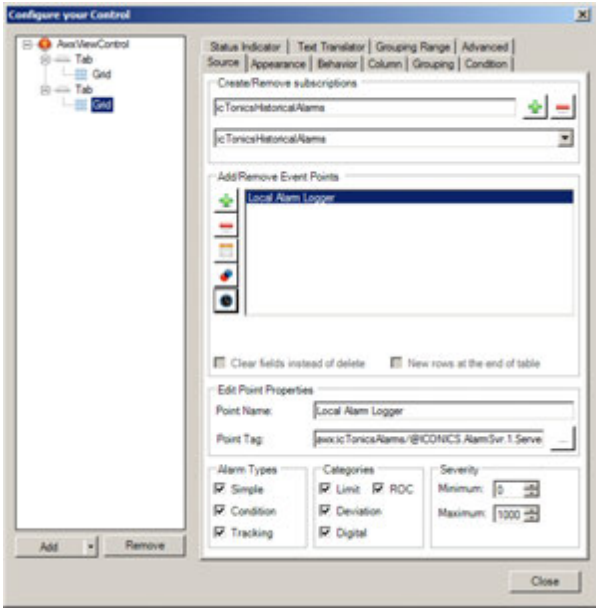


Figure L5.4 - Historical Alarm Sources

14. Select the **Column** tab and uncheck the **Show** in the grid option of the **Priority** and **Quality** columns. Click on "**Apply**" (the green checkmark button) at the left of the table to save your changes. Then click on "**Close**" to exit the configuration dialog box.

15. Go into **Runtime**. You may get a message saying that there are not any historical alarms for that time range. This is ok, there probably were not any alarms triggered since we started logging alarms.

16. By viewing the traffic light for each, verify that your alarm server and logger has started and triggered a new alarm by changing a tag value in the icTonicsTankMonitor display. See that there are alarms in the Historical Alarms tab of the viewer, like the example in Figure L5.5.

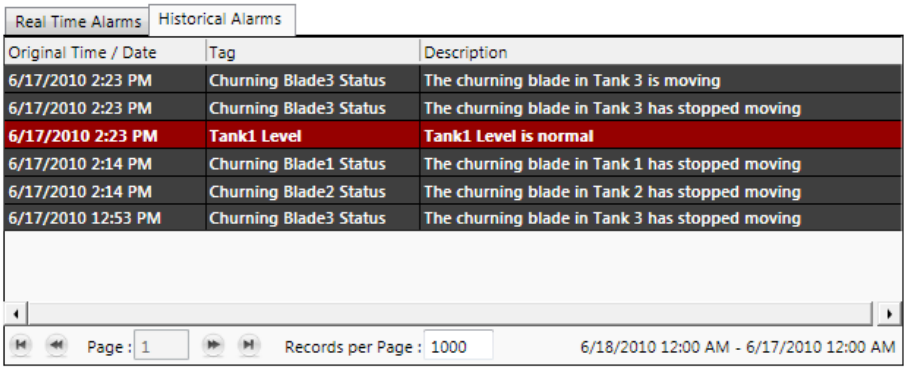


Figure L5.5 - Historical Data

Review: AlarmWorX64 Logger *R8*

8.1. Objectives Review

You should now know and understand the following topics:

- AlarmWorX64 Logger and how to start the service
- Configuring an AlarmWorX64 Logger database connection to log alarms
- Adding server-side and client-side filters for alarm logging and column definitions
- Creating node configurations in AlarmWorX64 Logger

8.2. Summary

The AlarmWorX64 Logger provides a permanent copy (database storage or printed copy) of alarm and event notifications produced by any OPC AE servers. AlarmWorX64 Logger subscribes to an OPC AE server and then stores the information in a database. You can configure the AlarmWorX64 Logger within the GENESIS64 Workbench.

In the AlarmWorX64 Logger, you can create a configuration that subscribes to an alarm server (e.g., the AlarmWorX64 Server) as the source of alarms you want to log. You can also add filters to limit the information that is logged into the database.

There are two types of filters: server-side and client-side. The server-side filter sorts alarm data coming from the alarm server into the logger, and the client-side filter filters data coming from the logger into the database.

The subscription to the alarm server ensures that the logger receives any alarm triggered in the alarm server. The received alarm is then logged into the destination database specified in the configuration settings. The database itself can be the configuration database for the AlarmWorX64 Logger (default) or any other MS SQL database.

You can also have multiple configurations in the same machine and use those configurations for multiple machines by configuring the nodes. After creating the configurations you want to use, you can add nodes and link them to those configurations. You can also link the same configurations to different machines or link multiple configurations to the same machine. This is very useful because if you have different subscriptions to different alarm (and/or event) servers but would like to log data to one machine, you will be able to do that as well.

8.3. Terms

Alarm Logger Columns	Alarm Logger Configuration	Alarm Logger Node
AlarmWorX64 Logger	Client-side Filtering	Printer Logging
Redundancy	Server-side Filtering	Table Management

8.4. Test Your Knowledge

In this chapter, you learned how to create an alarm logger configuration and set it up to log alarm information. Before you continue, make sure you can answer the following questions.

1. How do you connect to an alarm server to receive the information to log?

    a. Subscribe to an OPC AE server
    b. Draw a connection between the alarms and the logger
    c. Filter the alarms that you want to log
    d. Add alarm columns to log the data

2. If you want the alarm logger in *Machine A* to subscribe to an alarm server in *Machine B* and you want to limit bandwidth usage, what would you do in the logger?

    a. No changes are needed in the configuration
    b. Set up client-side filtering
    c. Set up server-side filtering
    d. Log into a different database

3. If you want to keep data for a year and store each month's data in a new table, what do you need to do?

   a. Nothing, it does that automatically
   b. Set up the active table
   c. Set up the archive tables
   d. Set up the active table and archive tables

4. True or False: You will need to associate an Alarm Configuration with a node for the configuration to work.

5. AlarmWorX64 Logger can send alarms and events to which of the following?

   a. A database
   b. A printer
   c. A fax
   d. An e-mail
   e. All of the above
   f. Both A and B
   g. Both C and D
   h. None of the above

TrendWorX64 Logger 9

TrendWorX64 Logger is a server application that is installed as part of the GENESIS64 suite. It handles data logging and data retrieval as well as multiple types of databases. The logger collects data from an OPC UA and OPC DA real-time data source and sends that data to a SQL server database such as Microsoft SQL Server 2008 or SQL Server 2008 Express Edition. Its basic functions are to collect and log data for analysis and to provide optional data redundancy.

The TrendWorX64 HDA server will connect to a TrendWorX64 Logger database configuration and serve the data contained in that database to HDA clients such as the TrendWorX64 Viewer. You can deploy the TrendWorX64 Logger using the same computer on which the database resides or connect to a database on a different computer on the network. You can configure TrendWorX64 Logger using configuration forms in the Workbench.

Figure 9.1 shows a diagram of how TrendWorX64 fits in with the GENESIS64 product family.

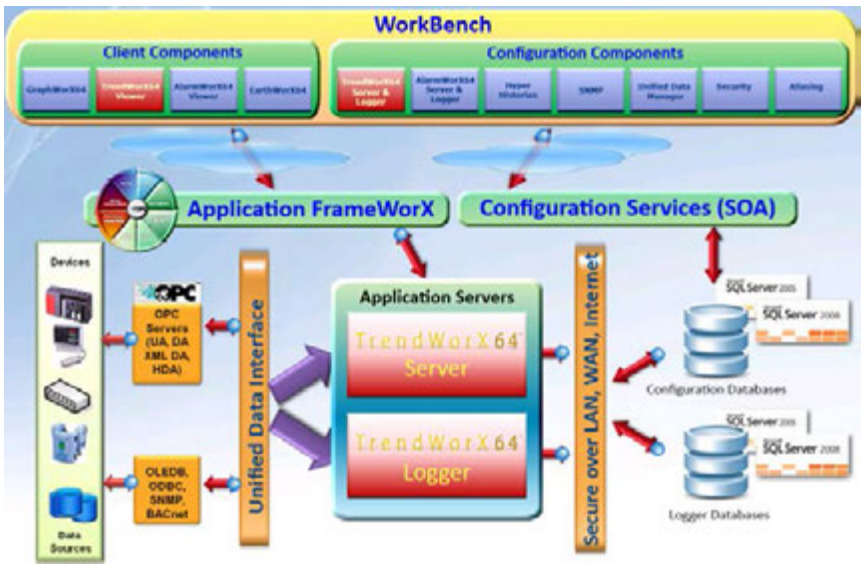


Figure 9.1 - TrendWorX64 Overview

After completing this chapter, you should be familiar with:

- TrendWorX64 Logger and how to launch it
- Database configurations
- Database groups and logging groups
- Tags that collect trend data

9.1. TrendWorX64 Logger in the Workbench

You can control various aspects of the TrendWorX64 Logger's behavior within a set of configuration forms in the Workbench. Using these forms, you can select the database that TrendWorX64 uses, create new databases, modify a database, and create and store configuration(s) that support your trending displays. In the Workbench, you can start and stop logging activities as needed.

To start TrendWorX64 Logger Configurator, go to **Start → All Programs → ICONICS → GENESIS64 → TrendWorX64 → TrendWorX64 Logger**. TrendWorX64 Logger opens in the Workbench as depicted in Figure 9.2. In this configuration, you can set up your database groups, logging groups, and tags, which determines where and what to log when the service starts.

The logging service relies on the FrameWorX64 server to send logging data to the TrendWorX64 Viewer client. In the Windows Task manager, you should see that the FrameWorX64 server process (IcoUAServer.exe) is running.

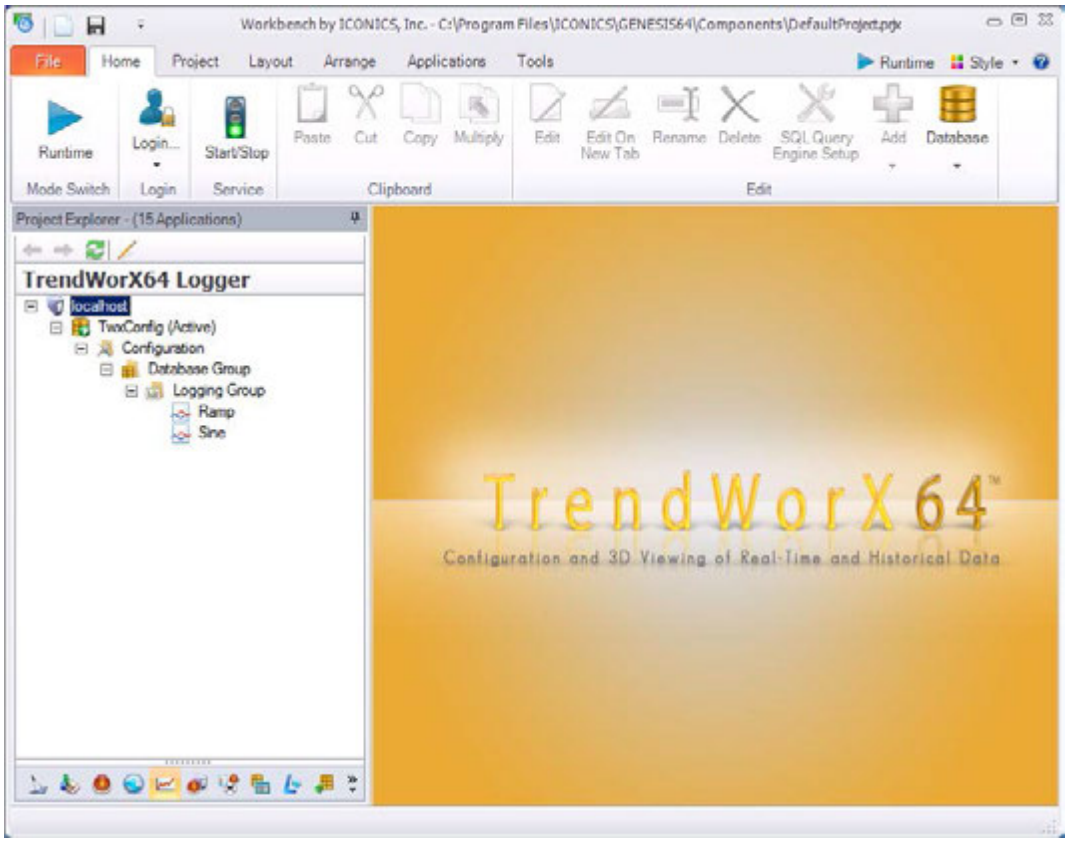


Figure 9.2 - TrendWorX64 Logger in the Workbench

9.2. Database Configurations

To configure the TrendWorX64 logger in the configurator, you need to make sure that you have a configuration database first. The GENESIS64 installation comes with a default configuration database depending on the options you choose during installation. However, you may create additional configuration databases as you wish.

You can create a configuration database by right-clicking on localhost in the Project Explorer and selecting the "New Database" menu item. Alternatively, you can click on the "Database" button in the Home ribbon and select "New Database".

The database creation wizard will come up and assist you through creating the configuration database. During this setup, you will be given the choice of including sample data. This option is selected by default. If you do not want to include sample data, you can uncheck this option before creating the configuration database.

Once you complete the steps in the wizard, you will see the database in the Project Explorer, with the database's Configuration node below it in the tree (if you included sample data).

To change the configuration's settings, double-click on the Configuration node in the Workbench's Project Explorer. The figure below shows the settings that are available for the configuration.

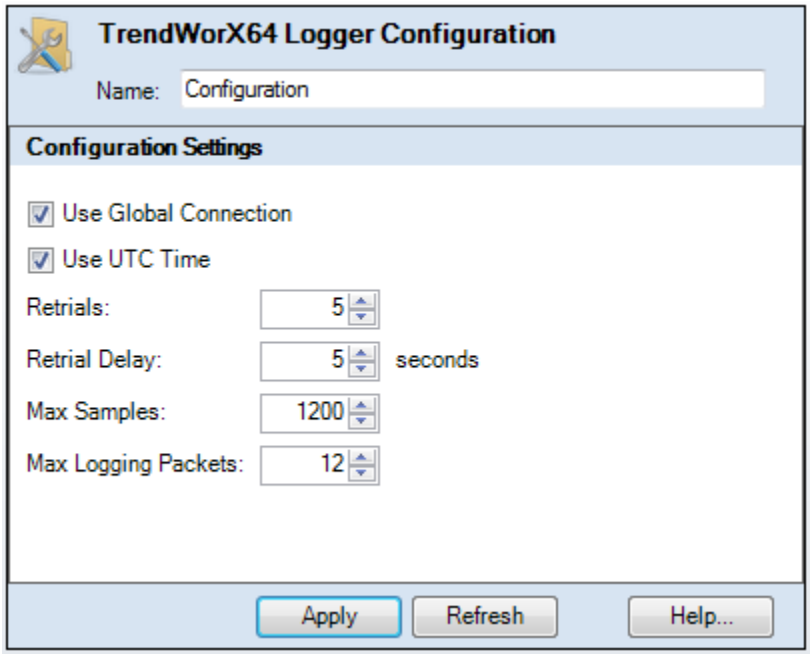


Figure 9.3 - TrendWorX64 Configuration Settings

Only one configuration per computer can be active at any given time; you must make the database active to use its configuration. To make the configuration database active, right-click on the database in the Project Explorer and select "Make Active" from the popup menu.

9.3. Database Groups

In order to log data, you must create a database group that specifies how to connect a database to store the information. You can have one or more database groups for a trend configuration. Each database group acts as a physical connection to the database, and manages data updates and logging activities. It is also a grouping of logging groups.

To create a database group, you can right-click on the database configuration, and then select "Database Group". The Database Group configurator appears with the Database Connection tab displayed, where you can specify the database connection, as illustrated in Figure 9.4.

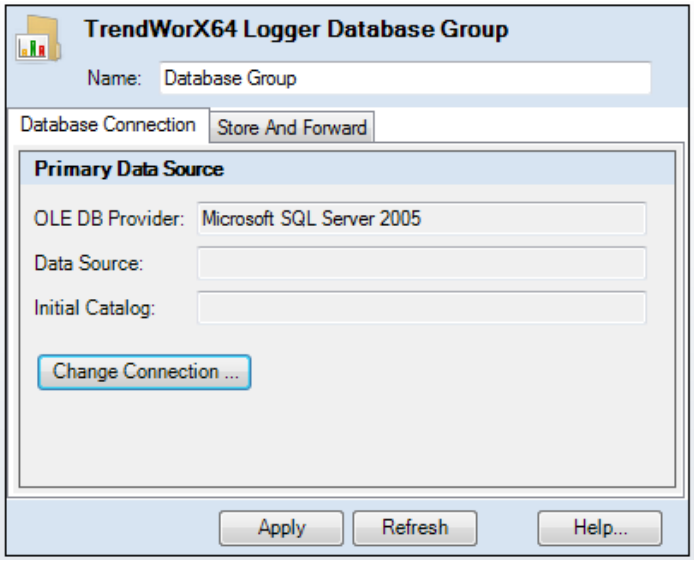


Figure 9.4 - TrendWorX64 Logger Database Group - Database Connection Tab

When creating a database group, enter a name. Then on the **Database Connection** tab click on the "Change Connection" button to open the OLE DB Connection dialog. Enter the connection information for the primary data source, and then click on the "OK" button.

9.4. Adding Logging Groups

It is likely that the TrendWorX64 Logger will be storing data for many different kinds of tags from the data source. Some tags may collect data from light sensors, other tags may collect data from security sensors, and still other tags monitors or control the physical plant. To organize the various types of tags that collect data, you can create as many logging groups as you need for the database group.

Not only does a logging group serve as a folder for tags, it also holds important settings for those tags. With these settings, you can:

- Control data collection rates and refresh rates for the tags
- Automate data collection by specifying the time spans when you want data to be collected from the logging group's tags
- Indicate when you want the collected data to be logged (that is, stored) in the database
- Manage the database tables that contain the logged data

To create a new logging group, right-click on the database group and select "Logging Group" from the popup menu. This displays the logging group configurator with the **Data Collection** tab on top, as shown in Figure 9.5. Let us talk about your configuration options.

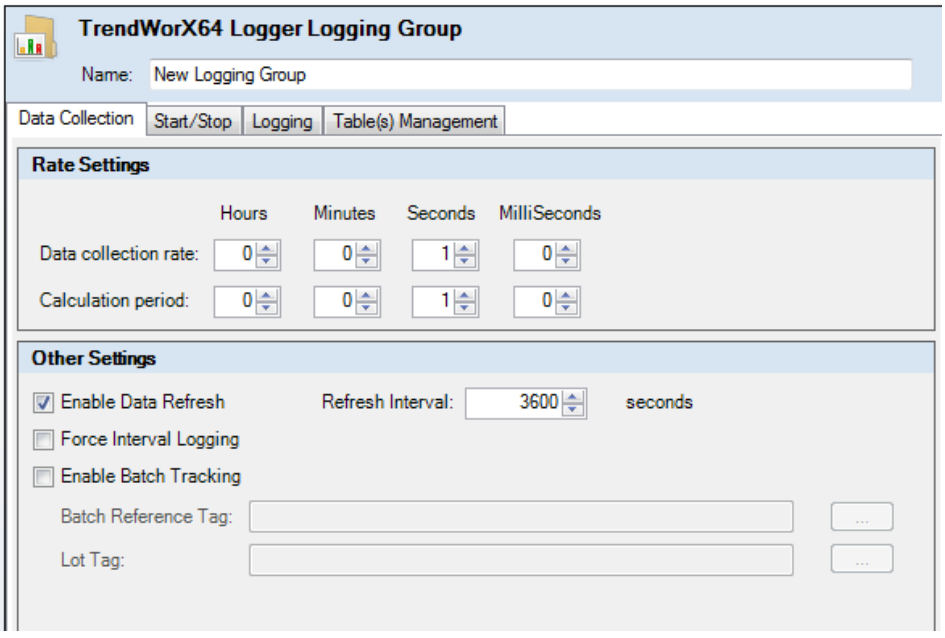


Figure 9.5 - Data Collection Tab for a Logging Group

9.4.1. Data Collection

On the **Data Collection** tab, the **Data Collection Rate** sets the frequency of how often to collect samples for the tags that you will be adding to the logging group. The default is 200 milliseconds, or 5 samples per second. The **Calculation Period** determines how often trending data are calculated for an aggregate function; by default, one update is calculated each second. The data collection rate and calculation period play a major role in consuming TrendWorX64 Logger and data source (SQL server) system resources. If your servers become sluggish, you may have to consider collecting fewer samples or lengthening the calculation period for fewer updates.

Use the "Enable Data Refresh" checkbox to refresh the data used for historical trend logging. This option makes a request to read (refresh) data to get an updated sample timestamp with the most recent value. The **Refresh Interval** must be longer than the data collection rate; this longer interval is necessary for getting responsive historical replay performance.

"Force Interval Logging" uses the TrendWorX64 Logger to force logging even when there is no change in data. Because OPC is change based, if values do not change between collection intervals, the OPC server may not send update to the trend server, causing no logged data. This option allows the server to actively poll data from the OPC server instead, making sure that there are data to log into the database even if there is no change in the tag's value. Having this type of data may be important to you later when you are tracking down values for critical OPC points.

TrendWorX64 supports batch and lot tracking. You can **Enable Batch Tracking** for the logging group and define OPC string signals that store the current **Batch** and **Lot** information. Batch and lot numbers can be retrieved automatically using a bar scanner or they can be entered manually using a display created in GraphWorX64.

9.4.2. Automated Data Collection Times

On the logging group's **Start/Stop** tab, shown in Figure 9.6, you can automate the times or conditions under which you want the logging group's data collection to begin and end. Using this method, you can automatically start capturing data at a certain time or stop collecting data when collection or data no longer needed. Although you are not required to end data collection for a logging group, preserving resources when no data collection is required is a good habit.

> **Note**: This tab dictates times for data collection only. Collected data held in memory is *logged to disk* in accordance with options on the Logging tab.

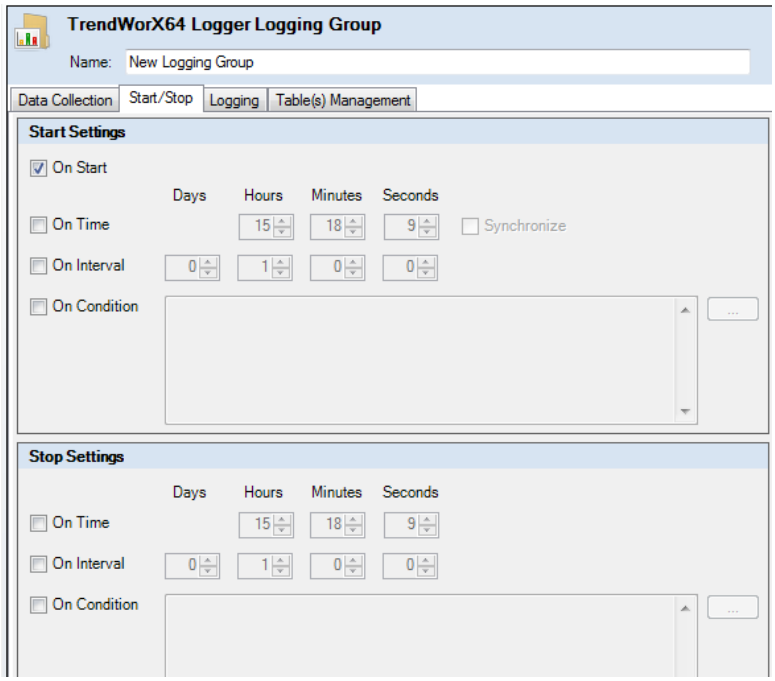


Figure 9.6 - Automated Start and Stop for a Logging Group

In automating start and stop times for data collection, you can:

- Start data collection immediately when the logger starts up by selecting **On Start**.
- Define the exact time at which you want data collection to start and/or stop each day by selecting **On Time**.
- Specify a length of time after which you want data collection to start after stopping, or stop after starting, by selecting **On Interval**
- Create a Boolean expression that evaluates to *True* or *False* to define the start or stop of data collection, in which case you select **On Condition**.

You can use more than one of these options for starting data collection as well as for stopping it. You can also use different methods for starting than you use for stopping; for example, you can start every day at 8:00 AM (**On Time**) and stop when a key piece of equipment shuts off (**On Condition**).

9.4.3. Your Data Logging Strategy

On the logging group's **Logging** tab shown below, you set the times or conditions under which you want all collected data in memory to be logged (that is, stored) in the database.

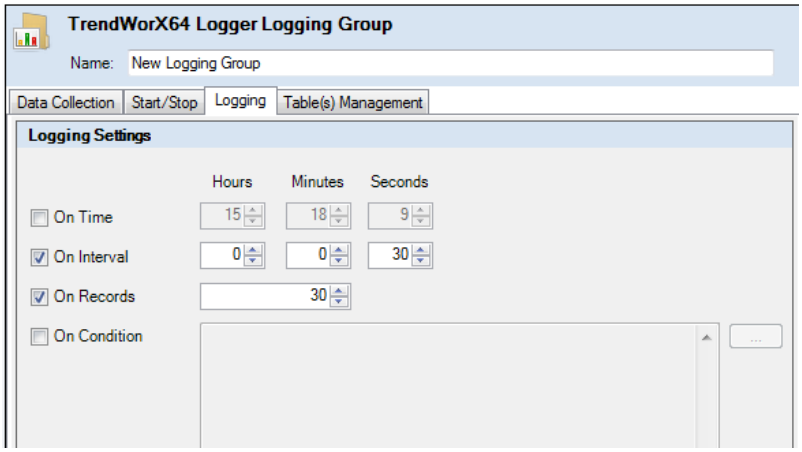


Figure 9.7 - Data Logging Times

The **On Time** option lets you specify the exact time at which data is to be logged each day. All data in memory at that time will be logged. With **On Interval**, you can specify the interval at which data logging occurs (for example, every 30 minutes). With **On Condition**, you can enter a *Boolean expression* that causes data to be logged when the expression becomes true. Finally, with **On Records**, each tag's data will be logged when the specified number of records has been collected. This last option gives you some control over the number of records being held in memory.

9.4.4. Your Table Management Strategy

When data is logged, it is stored in tables. You can set up data logging so that data is stored to multiple tables in a sequential order. Doing this provides you with:

- **Faster access to trend data**. Trend clients can be configured in a way that they can retrieve data from the most-recently-created tables, while the other tables are maintained for reporting and comparison to the data in the most-recently-created table.
- **Database Optimization**. Although historical information is retrieved by combining information from various historical tables, the ability to store data in multiple sequential tables optimizes performance by removing older, unneeded tables.
- **Improved Load Balancing**. It also helps in designing a configuration for system load balancing, as you can write tables to different SQL servers. By controlling the size of each table, you avoid creating extremely large tables that may degrade overall performance. To accelerate data retrieval, SQL Server creates the necessary database schema to maintain updated information with respect to the time interval at which each table logs data.

Use the Table Management tab, shown in Figure 9.8, to specify the number of tables in the database and when to create a new table. Start by giving the table a name.

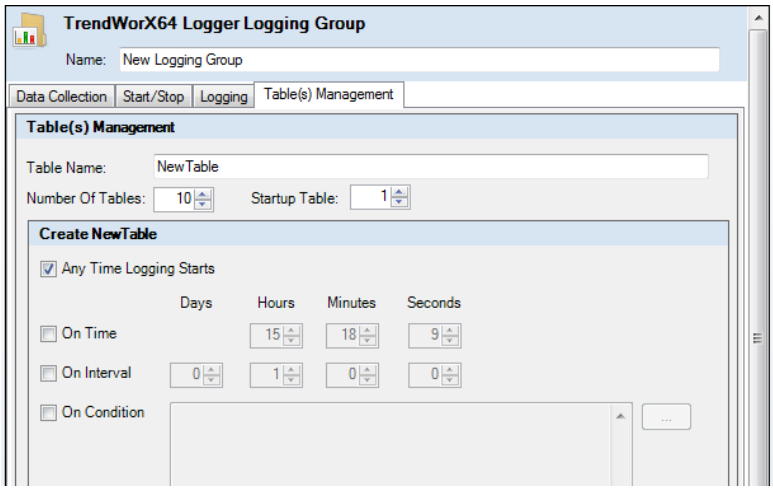


Figure 9.8 - Managing a Logging Table

Start by giving the tables a basic table name. You should also enter a **Startup Table** number to tell the logger how to create the first logging table. The name of the table is the table name appended by the table number starting with the startup table number.

The **Number of Tables** field determines the maximum number of tables that will be created (the maximum is 3,000). This gives you the ability to manage the capacity of the hard drive by limiting the number of tables in the database. When that number of tables is reached and filled, the oldest table will be overwritten with most recent data.

You must also specify the occasion for creating a new table. To create a new table every time data logging begins, select **Any Time Logging Starts**. To create a new table at a set time each day, select **On Time** and specify the time. To create new tables at a predetermined interval, such as every 5 days, select **On Interval** and specify the interval. To create a new table whenever a Boolean expression that you have created evaluates to true, select **On Condition** and then specify the expression.

9.5. Add Tags to a Logging Group

In addition to collection, logging, and table specification, the logging group is a container for the tag or tags it collects and logs data. Tags contain the data that you are collecting and logging for trend analysis. Tag data is collected into memory, and then stored in the active database.

You can add tags one at a time. Or you can add multiple tags using the Tag Import Wizard.

9.5.1. Adding One Tag at a Time

To add a tag to a logging group, right-click on the logging group, and select **Tag** from the popup menu. Then, you can configure the tag in the Tag configurator, shown in below.

Each tag is made up of a data source and the attributes or fields that are collected by that data source. You have control over the attributes and fields that are collected, but the path to the OPC UA or OPC DA tag is fixed to reference the particular tag of interest. In building the tag path, which is referred to as the *signal name*, you can use global and language aliasing to make it easier to create and work with tags. With this in mind, enter the **Signal Name** and a description of the tag.

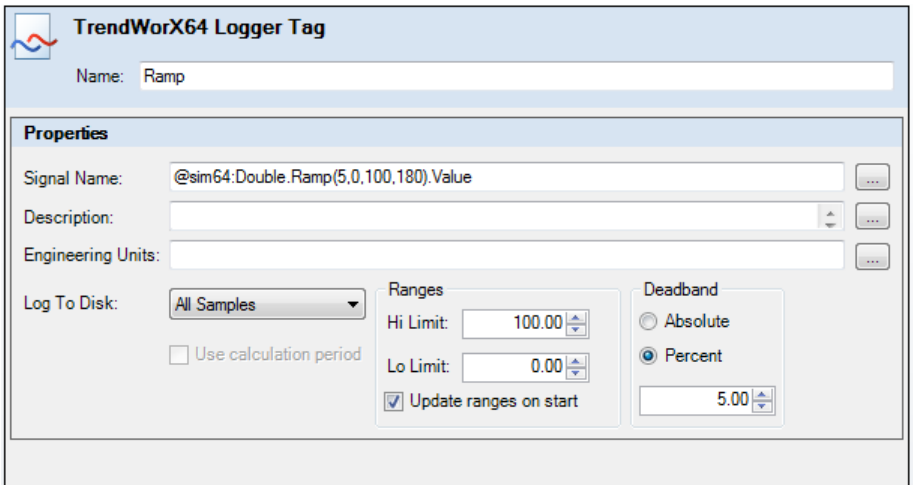


Figure 9.9 - Adding a Tag to a Logging Group

You must also specify which collected samples you want to log to disk. You can log all samples or specify a selected value (such as the minimum, the maximum, the average, and so on) from the **Log to Disk** dropdown list. Selecting the "Use Calculation Period" checkbox forces the database to recalculate aggregates at sub-intervals defined on the logging group's Logging Settings tab and applies to only the Max, Min, Avg, and Std. Dev. functions. Leaving the "Use Calculation Period" checkbox empty means that all aggregate functions are calculated based on the sample values as samples are written to disk.

Other options are also available for limiting the data that is logged. You can set a **range** for the samples and you can also use a **deadband** filter. The online help for TrendWorX64 Logger provides a good explanation of these options; you can navigate to the "Tags" online help topic by clicking on the **Help** icon at the top-right of the Workbench, then from the help's **Content** tab navigate to **TrendWorX64 → TrendWorX64 Logger → Tags**.

9.5.2. Creating Several Trend Tags at Once

TrendWorX64 Logger has a Tag Import Wizard which you can use to create multiple trend tags from your OPC tags. Like the Tag Import Wizard in AlarmWorX64 Server, the TrendWorX64 Tag Import Wizard lets you specify a common naming system for all generated trend tags that is incremented by 1 for each tag. Default values are used for each generated tag, which you can modify after its creation.

To generate multiple trend tags, right-click on the logging group in the TrendWorX64 Logger in the Project Explorer, then select **Tag Import Wizard**. Alternatively, on the Home ribbon click the Add button and select **Tag Import Wizard** from the dropdown menu.

This opens the **TrendWorX64 Logger Import Tags** form shown in Figure 9.10, which is similar to, but simpler than, the form used, by AlarmWorX64 Server. Provide a base name, the first number, and the number of numeric places. Then add all of the tags you want to add to the logging group, and then click the "Import" button.

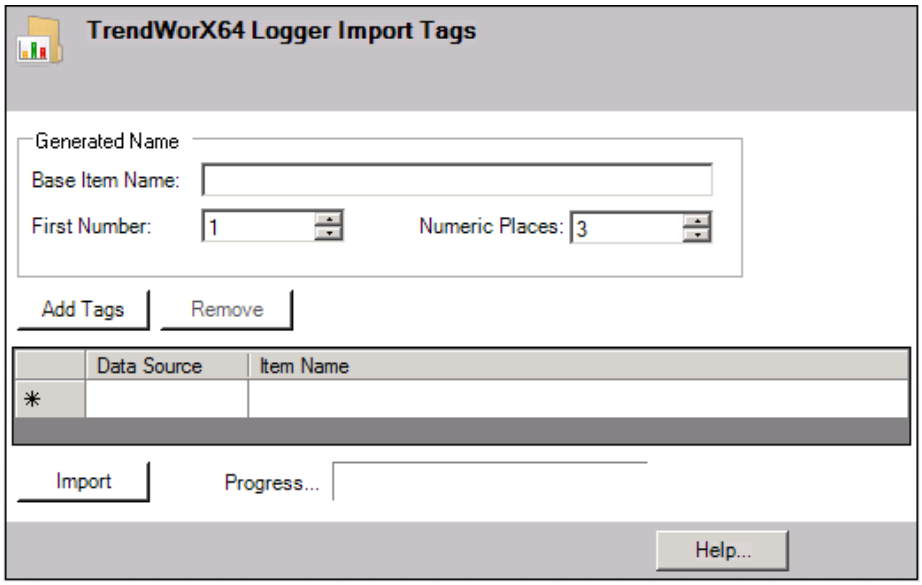


Figure 9.10 - Tag Import Wizard in TrendWorX64 Logger

Lab: TrendWorX64 Logger *L9*

9.1. Lab Overview

Many companies want to log plant floor data for analysis and icTonics is no different. They have requested that we set up a logging mechanism that allows them to look at older data and analyze it in the future.

Estimated time to complete this lab: 20 minutes

9.1.1. Task Checklist

In this lab, you will complete the following tasks:

- Create a new TrendWorX64 configuration database
- Create a new trend configuration
- Create a new database group
- Create a new logging group
- Make a connection to a logging database
- Add tags to log
- View the logged data for the tag

9.1.2. Support Files

You do not need any supporting files to complete this lab.

9.2. Configure TrendWorX64 Server

The TrendWorX64 configuration allows you to set up OPC tags whose values you want to log. We will create a configuration that specifies where to log data, how fast to collect data, and how often to write data into the database.

1. Open the Workbench if you have closed it since the previous lab and select TrendWorX64 Logger in the Project Explorer.
2. In the Home ribbon, click on the "Database" button and select "**New Database**".
3. The **Create Database Wizard** should come up. Click on "**Next**" to get to the Database option screen.
4. **Uncheck** the Include **Sample Data** option and name the database "**icTonicsTrendLoggerConfig**".
5. Click on "**Finish**" to create the new configuration database.
6. You should see your configuration in the Project Explorer. Expand icTonicsTrendConfig to expose the Configuration folder. You should have an empty Configuration folder.
7. Right-click on Configuration and select "+ **Database Group**", as shown Figure L9.1.

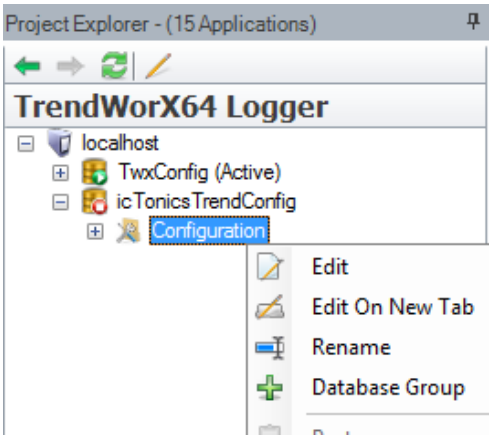


Figure L9.1 - Adding a Database Group

8. A new tab will open showing configuration properties for the new database group. Name this database group "**icTonicsTrends**".
9. In the **Database Connection** tab, click on the "**Change Connection**" button to bring up the **OLE DB Connection** dialog box.

10. Leave the OLE DB Provider as Microsoft SQL Server. Change the Server Name to "(local)\SQLEXPRESS" and select "**icTonicsTrendData**" as the database. The finished configuration should look similar to Figure L9.2.

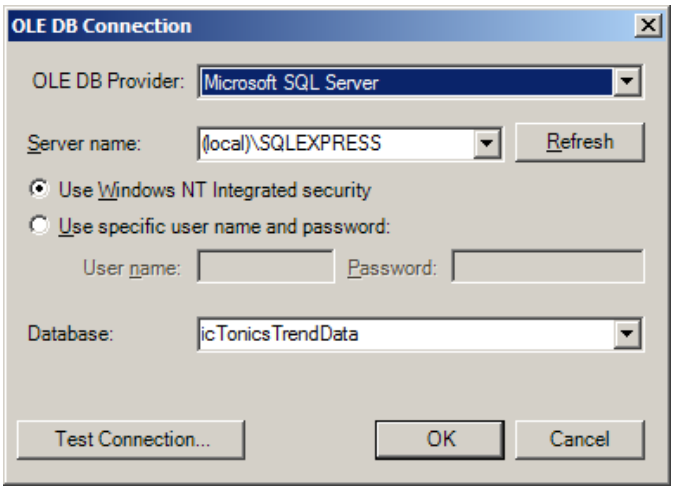


Figure L9.2 - Database Connection Configuration

11. Click on the "**Test Connection**" button. After a successful test, click on "OK" to close out of this dialog box.

12. Click on "**Apply**" to save the changes. You may get a message telling you that changes will not take effect until the logger restarts, just click on "OK" to close the message box.

13. Right-click on the **icTonicsTrend** database group and select "+ **Logging Group**".

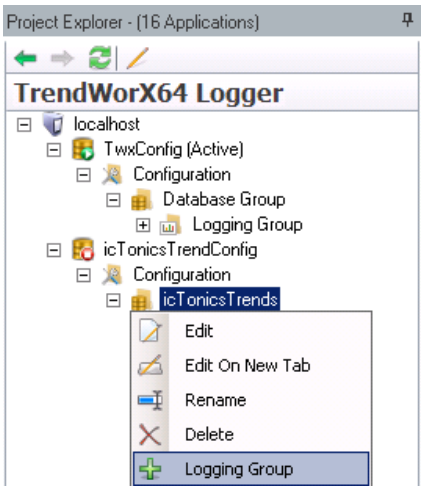


Figure L9.3 - Adding a Logging Group

14. A new tab should open called the Logger Logging Group. Rename this logging group "**Tank Levels**".

15. In the Data Collection tab, change the **Data collection rate** and **Calculation period** to **5** seconds.

16. Since our data does not change often, let us put a checkmark in the "**Force Interval Logging**" check box. Your finished configuration should look similar to Figure L9.4. Click on "**Apply**" to save the changes.

    Note: You can hover the mouse cursor over the various checkboxes to review the associated Tool Tips. This will help you understand each fields function.

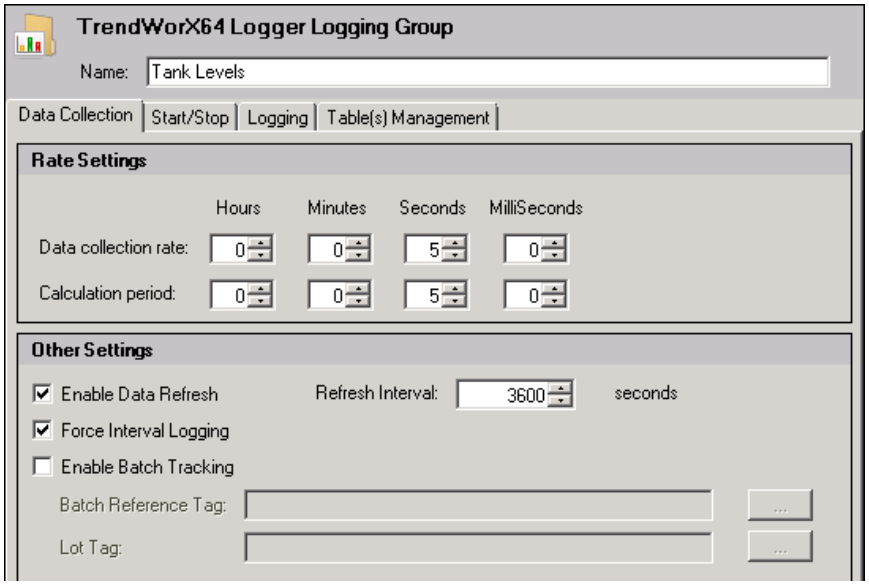


Figure L9.4 - Data Collection Tab

17. Select the **Logging** tab. Uncheck the "**On Records**" option. We will leave the default logging condition at "**On Interval every 30 seconds**".

18. Select the **Table(s) Management** tab. Change the Table Name to "**TankLevels**". Uncheck the option to create new table **Any Time Logging Starts** and check the option to create a new table "**On Time**". Set it to 11:00 PM (23 Hours, 0 Minutes and 0 Seconds). Your finished configuration should look similar to Figure L9.5.

    Note: In the **Table Name**: field if you enter **Tank Levels** (with a space), data validation will fail. The proper entry is **TankLevels** (no space).

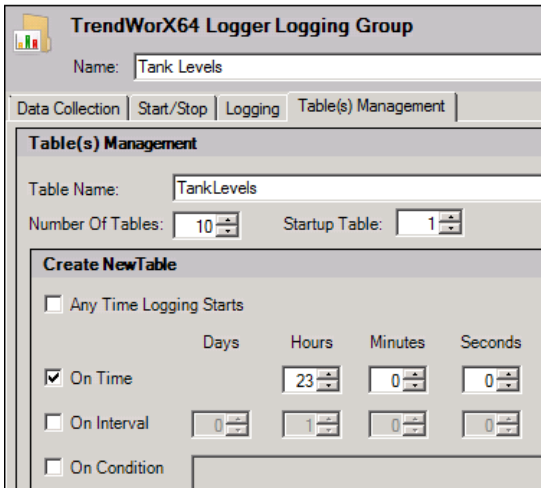


Figure L9.5 - Table(s) Management Tab

19. Click on "Apply" to save the configuration. We will now add the tags to log the data.

20. Right-click on "Tank Levels" and select "+ **Tag**". A new tab should open with tag properties.

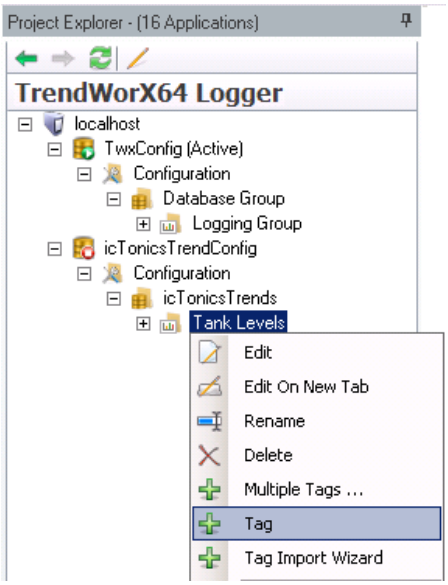


Figure L9.6 - Adding Trend Tags

21. Name this tag "**Tank1 Level**". Click on the ellipsis button next to the Signal Name field to bring up the **Data Browser**.

22. In the Data Browser, we will look for the following tag:
@ICONICS.IconicsOPCUAServer.V5\icTonics.Tanks.USA.Foxboro.Tank1.TankLevel.Value

23. We can leave everything else as the default for the tag configuration. Click on "**Apply**" to save the changes. Your finished tag configuration should look similar to Figure L9.7.

24. Right-click on "Tank Levels" and select "+ **Tag**". A new tab should open with tag properties.

25. Name this tag "**Tank1 LevelSetpoint**". Click on the ellipsis button next to the Signal Name field to bring up the **Data Browser**.

26. In the Data Browser, we will look for the following tag:
@ICONICS.IconicsOPCUAServer.V5\icTonics.Tanks.USA.Foxboro.Tank1.LevelSetPoint.Value

27. Select it and then click on "**OK**" to go back to the **Tag Configuration** screen.

28. We can leave everything else as the default for the tag configuration. Click on "**Apply**" to save the changes. Your finished tag configuration should look similar to Figure L9.7.

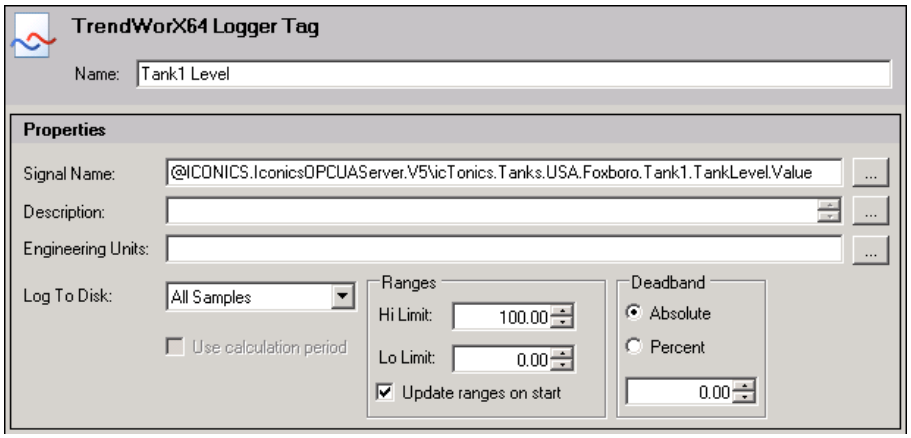


Figure L9.7 – Tank1 Level Tag Configuration

29. Click on the Start/Stop light in the ribbon to stop the data logging service. The light should turn red.

30. Right click on the icTonicsTrendConfig database and select "Make Active". This will make the configuration database you created active.

31. Click on the Start/Stop light in the ribbon again and start data logging service. The light should turn green.

9.3. Viewing Logged Data

You can view the trend data by using the Trend Viewer. We will go into more details about the trend viewer in the next chapter. For now, it is just a container for viewing logged data.

1. In the Workbench64 Project Explorer, select TrendWorX64 and right-click "New Trend" from the tool bar across the top of the screen to create a new trend.
2. Double-click on the white space in the work area to bring up the "**Configure your Control**" dialog box.
3. Right-click on "**Chart**" in the left-hand pane of the dialog box, and select + **Add** → **Plot**.
4. Right-click on "Plot", and select + **Add** → **Pen**.
5. Select the Pen you just added on the left-hand pane. In the **General** tab on the right, click on the ellipsis button next to Data Source to bring up the Data Browser.
6. Click "Home" on the left-hand side and double-click on "Historical Data" on the right-hand side.
7. Double-click on TrendWorX64 → icTonicsTrends → Tank Levels → Tank1 Level and then click on "OK".
8. Click on "Close" in the configuration dialog box.
9. Save the configuration in the c:\training folder with the name "icTonicsTrend64" by clicking on the "Save" button in the ribbon or by clicking the "File" button left to the "Runtime" button on the bar right above the alarm viewer. The file extension will be ".twxx" or ".twxxs" (for Silverlight files).
10. You can now go into runtime and see your trend. Feel free to change the values of the Tank1 Level tag a few times to see the value on your graph change.

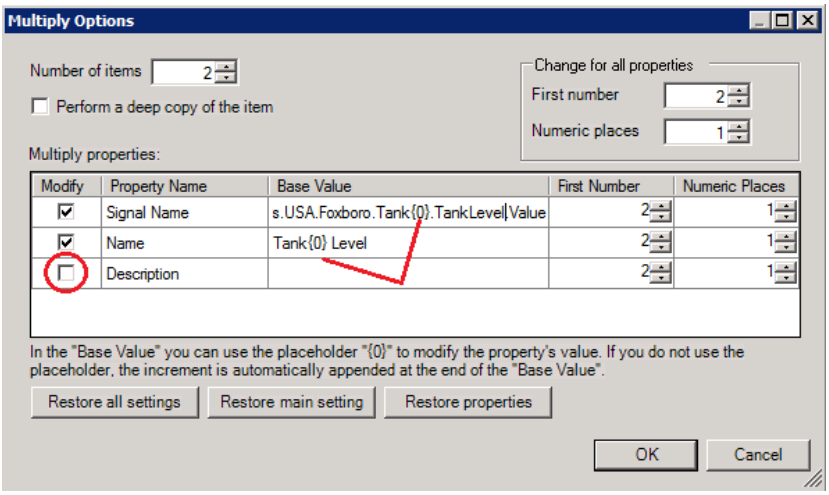
9.4. Trend Logger Tag Multiply

1. Now we will add similar loggers for Tank2 and Tank3. For this we can use the Multiply function which makes it easier to create similar alarms.

2. Right click on the "Tank1 Level" logger tag and select Multiply to launch the **Multiply Options** dialog box. You could alternatively select "Tank1 Level" and click on Multiply icon in the Clipboard section of the ribbon.

3. Set the **Number of items** to "**2**", **First Number** to "**2**" and the **Number of places** to "**1**".

4. Let's use the **placeholder {0}** to modify the appropriate property values. See the figure below for more information. Also note that within the Multiply Options dialog there is a description of the placeholder {0} functionality.



L9.8 - Multiply Options Dialog Box

5. Deselect the **Description** checkbox.

6. When your configuration resembles the figure below, select **OK**.

   This step is very important as it will affect your alarm functionality for the rest of the training session. Carefully review, and edit if necessary, each of the newly created logger tag's **Name** and **Signal Name** properties.

   Take a moment and check to make sure they resemble the following tag syntax:
   @ICONICS.IconicsOPCUAServer.V5\icTonics.Tanks.USA.Foxboro.Tank1.TankLevel.Value
   @ICONICS.IconicsOPCUAServer.V5\icTonics.Tanks.USA.Foxboro.Tank2.TankLevel.Value
   @ICONICS.IconicsOPCUAServer.V5\icTonics.Tanks.USA.Foxboro.Tank3.TankLevel.Value

7. Repeat steps you've learned to add the following three tags to be logged.

@ICONICS.IconicsOPCUAServer.V5\icTonics.Tanks.USA.Foxboro.Tank1.LevelSetPoint.Value
@ICONICS.IconicsOPCUAServer.V5\icTonics.Tanks.USA.Foxboro.Tank2.LevelSetPoint.Value
@ICONICS.IconicsOPCUAServer.V5\icTonics.Tanks.USA.Foxboro.Tank3.LevelSetPoint.Value

8. Your finished configuration may look like the following:

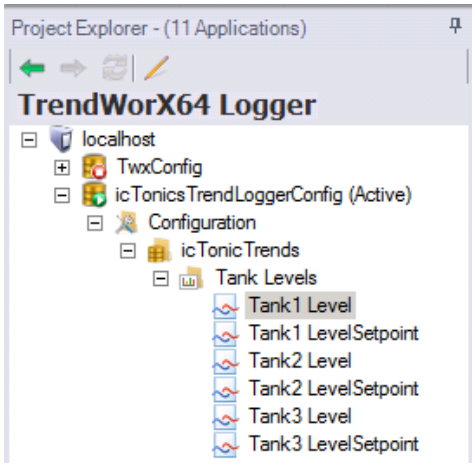


Figure L9.9 - Advanced TrendWorX64 Configuration


ICONICS

Review: TrendWorX64 Logger R9

9.1. Objectives Review

You should now know and understand the following topics:

- TrendWorX64 Logger and how to launch it
- Database configurations
- Database groups and logging groups
- Connecting to a logging database
- Adding tags for logging trend data
- Viewing trend data

9.2. Summary

TrendWorX64 Logger handles data logging and data retrieval among multiple types of databases. The logger collects data from OPC UA and OPC DA real-time data sources and sends that data to a SQL server database such as Microsoft SQL Server 2005 or SQL Server 2005 Express Edition.

You can control various aspects of the TrendWorX64 Logger's behavior from within a set of configuration forms in the Workbench. Using these forms, you can select the database that TrendWorX64 Logger uses, create new databases, modify databases, and create and store configuration(s) that support your trending displays.

During installation of TrendWorX64, you can install a default configuration database. The TrendWorX64 Logger configurator provides a wizard for creating new SQL Server 2005 or SQL Server 2005 Express Edition configuration databases.

By creating database groups, you identify the primary data sources and can configure redundant databases. Each database group acts as the physical connection to the database, and it manages data updates and logging activities for one or more logging groups.

Tags contain the data that you are collecting and logging. Tag data is collected into memory and then stored in the active database table during logging. To organize the various types of tags that collect data, you can create as many logging groups as you need for each database group.

A logging group serves as a folder for tags, and it holds tag settings that control data collection rates. The logging group also determines the times at which data will be collected and the rate at which collected data will be logged to the database, and it manages the database tables that contain the logged data.

9.3. Terms

Database Group	Logging Group	Signal Name
TrendWorX64 Configuration	TrendWorX64 Logger	

9.4. Test Your Knowledge

In this chapter, we discussed how to use TrendWorX64 Logger. Before continuing to the next chapter, make sure you can answer the questions below:

1. Where do you define the database in which you want to log data?

    a. In the configuration properties form
    b. In the database group
    c. In the logging group
    d. In the individual tag

2. Where do you add tags?

    a. In the database group
    b. In the trend configuration
    c. In the logging group
    d. None of the above

3. What steps might you take if data collection becomes sluggish?

    a. Collect fewer samples
    b. Lengthen the calculation period
    c. Log data more frequently
    d. All of the above
    e. Both A and B
    f. Both A and C
    g. Both B and C
    h. None of the above

4. Which of the following statements are true when configuring a logging group to start data collection?

   a. Data collection can be configured to start at the same time each day
   b. Data collection can be configured to occur when a Boolean expression evaluates to true
   c. You can specify how soon you want data collection to start after stopping, or stop after starting
   d. No options are available; data collection always starts when the logger service starts and always stops when the logger service stops.
   e. All above
   f. A, B, and C
   g. None of the above

5. Which of the following statements are true when configuring a logging group for data logging?

   a. Logging can be configured to log data at the same time each day
   b. Logging can be configured to occur when a Boolean expression evaluates to True
   c. You can specify a time interval for logging data
   d. Logging can be configured to log data after a certain number of records have been collected
   e. All above
   f. B, C, and D
   g. None of the above

6. True or False: It is possible to configure a logging group to log into one or more data tables.

TrendWorX64 Viewer Control 10

The ICONICS TrendWorX64 Viewer, shown in Figure 10.1, is a client application that provides real-time and historical data trend displays within the GENESIS64 product family. The viewer communicates to OPC UA, OPC DA, and OPC UA HDA servers to provide trend displays based on the connections to the data source that you specify.

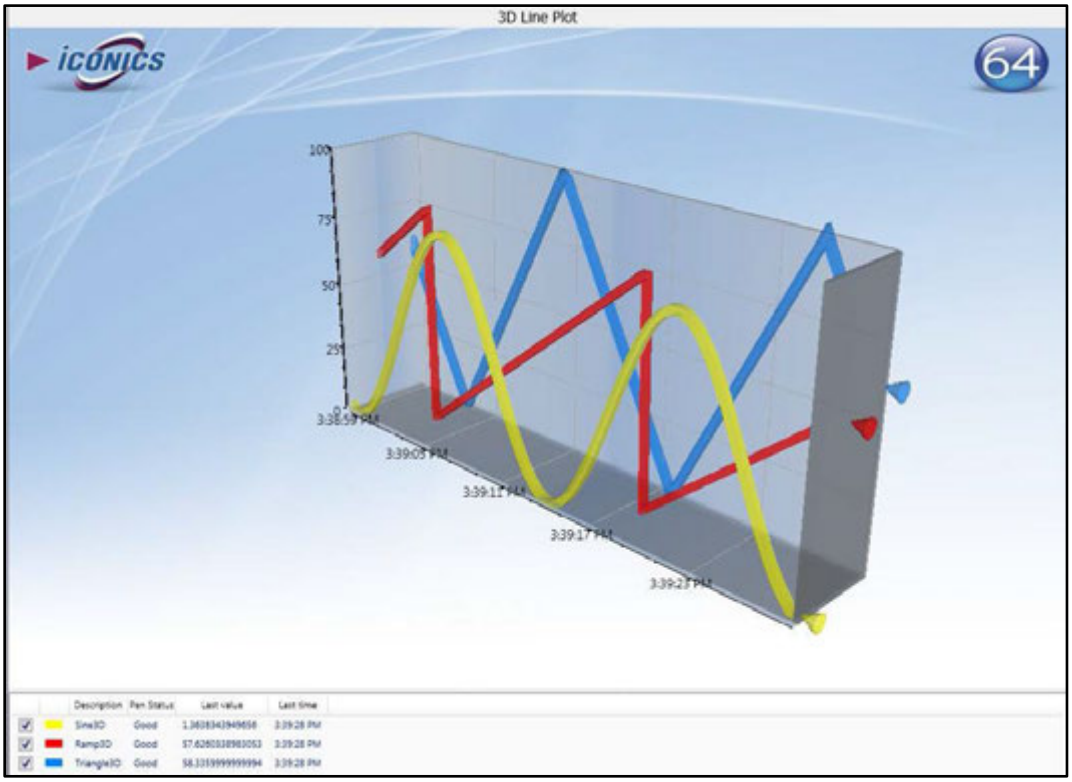


Figure 10.1 - TrendWorX64 Viewer

Features of the TrendWorX64 Viewer include:

- Real-time data displays
- Historical data displays
- Data utility functions
- Sophisticated, multi-threaded data buffering to guarantee data integrity with support for customer-configurable data collection rates

With TrendWorX64 Viewer you can use the following features and more:

- 3D visualization
- A side panel
- Multiple tabs with multiple charts
- Multiple pens and a runtime legend that identifies them
- A Toolbar for changing the Trend Period
- Plots that can swap the Y and X axes
- A summary view
- Ability to modify and customize the display at runtime

By the end of this chapter, you will be familiar with the following concepts:

- TrendWorX64 Viewer properties and different kinds of modes
- Chart options
- The side panel
- Tabs, plots, and pens
- Time and rates
- Runtime operations
- Workbench vs. GraphWorX64 displays

10.1. TrendWorX64 Viewer Modes

The TrendWorX64 Viewer has three different modes: Configuration mode, Runtime mode, and Freeze mode.

Configuration mode is design for changing the properties. In Runtime mode the data is collected and displayed and the chart updates according to the data received from the OPC servers. During runtime, you can freeze the display and examine the data in it. The data still collects, but the display stops updating with current data until you leave freeze mode.

10.2. TrendWorX64 Viewer Properties

The TrendWorX64 Viewer Properties dialog, shown in Figure 10.3, is the trend configuration interface, which is accessible from Workbench or GraphWorX64. It is more appropriate to describe it as a "control", which is itself a container that can contain additional objects. To open this dialog, simply click on the "Configure" button in TrendWorX64 Viewer Configuration ribbon.

When you first open the dialog for a new trend, the Viewer control contains two configurable items in the tree structure on the left: a side panel and a tab that has a chart in it. It looks something like the tree in Figure 10.2.

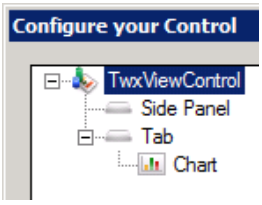


Figure 10.2 - Default Structure of Objects in the TrendWorX64 Viewer

Clicking on TrendWorX64Viewer (or TwxViewControl) in the tree offers general options such as whether there is a border or if the runtime toolbar appears as well as header, title and subtitle options. Figure 10.3 shows the Appearance tab (with the Advanced tab behind it) for the TrendWorX64Viewer tree items.

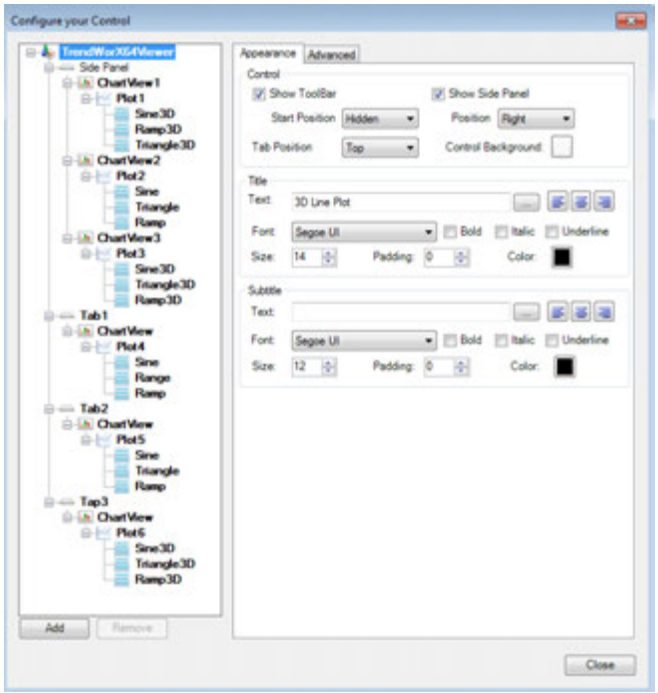


Figure 10.3 - TrendWorX64 Viewer Properties

Note: Not all of these configuration options are within the scope of this class. For more information about TrendWorX64 Viewer properties, please refer to the TrendWorX64 Viewer Help documentation.

10.2.1. Appearance Settings

The **Appearance** tab, shown previously in Figure 10.3, appears by default when you select the viewer object. This tab sets the appearance properties for the trend viewer's toolbar, side panel, title, and subtitle.

The "Show Toolbar" checkbox lets you prevent the toolbar from appearing with the trend, but if you choose to show it you use the "Start Position" option to choose the toolbar's initial position. You can dock the toolbar at the top, bottom, right, left, or let it float. The toolbar appears only in Runtime or Freeze mode. If the toolbar is hidden, runtime operators can double-click on the display to show the toolbar.

The "Show Side Panel" checkbox and "Position" option control the display of the viewer's side panel at runtime and specify the position of the side panel at the top, bottom, left, or right of the trend. The side panel is a powerful feature, which allows you to switch between the different types of charts predefined in TrendWorX64 Viewer. If you choose to show the side panel, make sure you configure it.

Although the Side Panel object appears just below the Viewer object in the configurator tree, we are going to side step it for the moment and focus on the Viewer's other objects. After that, we will return to the topic of the side panel.

10.2.2. Advanced Settings

The Advanced tab lets you customize the TrendWorX64 Viewer's appearance, behavior, header, measurements, text, etc. Properties on the Advanced tab come with helpful tips that appear for the selected property at the bottom of the tab.

10.3. Tabs

TrendWorX64 Viewer allows you to create multiple tabs inside the Viewer. These tabs are visible in Configuration mode as well as in Runtime mode. Figure 10.4 illustrates that a TrendWorX64 Viewer with multiple tabs and a side panel that provides a consolidated view of each tab. If you use these two features together in a Viewer, your runtime users will be able to switch easily between graphs and gain access to certain plots quickly.

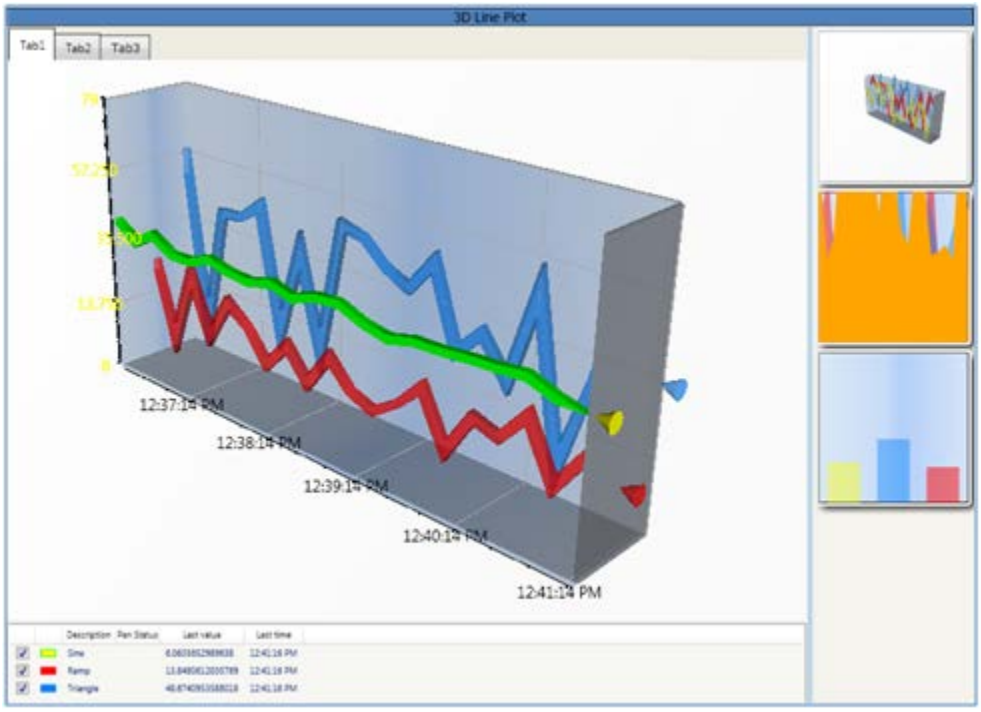


Figure 10.4 - TrendWorX64 Viewer with Side Panel and Multiple Tabs

You can add a new tab to the Viewer by selecting the Viewer and then clicking on the Add button at the bottom of the control's configuration dialog. Or you can right-click on the TrendWorX64Viewer in the tree control and select Add → Tab.

Selecting the tab in the tree control gives you access to its appearance tab where you give the tab a title, and can set a color scheme for the tab when it is selected and not selected. To create a uniform appearance among the tabs in the Viewer, you can also apply that scheme to all other tabs simply by clicking the "Apply To All Tabs" button.

At times, you may want to copy a fully-configured tab along with all of its charts. After copying it, you can edit the tab and its charts to create a new graph. To do this, you right-click the tab in the tree control and select the "Multiply" option. This lets you make one or more copies of the tab and all of its components.

10.4. Chart View

A chart is a container that holds one or more plots. Each tab in the Viewer can have any number of charts in it, but of course, you will limit the number of charts to keep the tab from becoming unusable.

Charts come with several options, as shown in Figure 10.5. You will also notice in the figure that charts can also be added to the side panel.

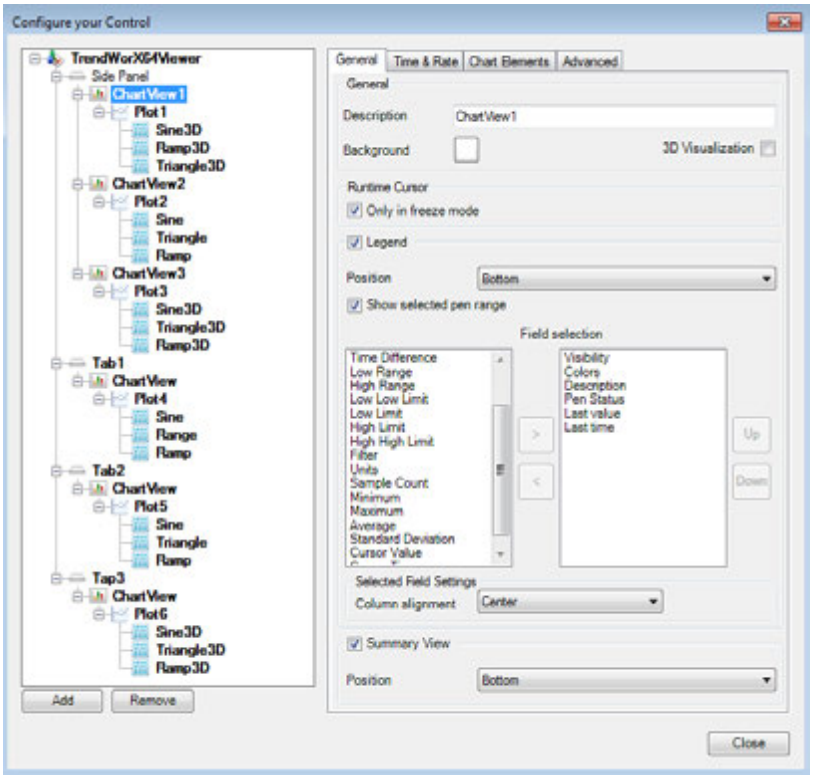


Figure 10.5 - Chart Options

> **Note:** You can add a plot to the chart by selecting the chart object, then clicking the Add button at the bottom of the screen.

You add a chart to a tab by right clicking the tab object in the tree and selecting Add → Chart. You configure a chart by selecting it in the tree view to bring up its options, which are spread over four tabs: the General tab, the Time & Rate tab, the Charts Elements tab, and the **Advanced** tab. We will look at the first three tabs in this course.

10.4.1. General Tab

On the chart's **General** tab shown in Figure 10.6, you can set the following:

- The general appearance
- Runtime cursor
- Legend
- Summary view

On the General tab, you can start by giving the chart a **Description** and choosing a **Background** color or image.

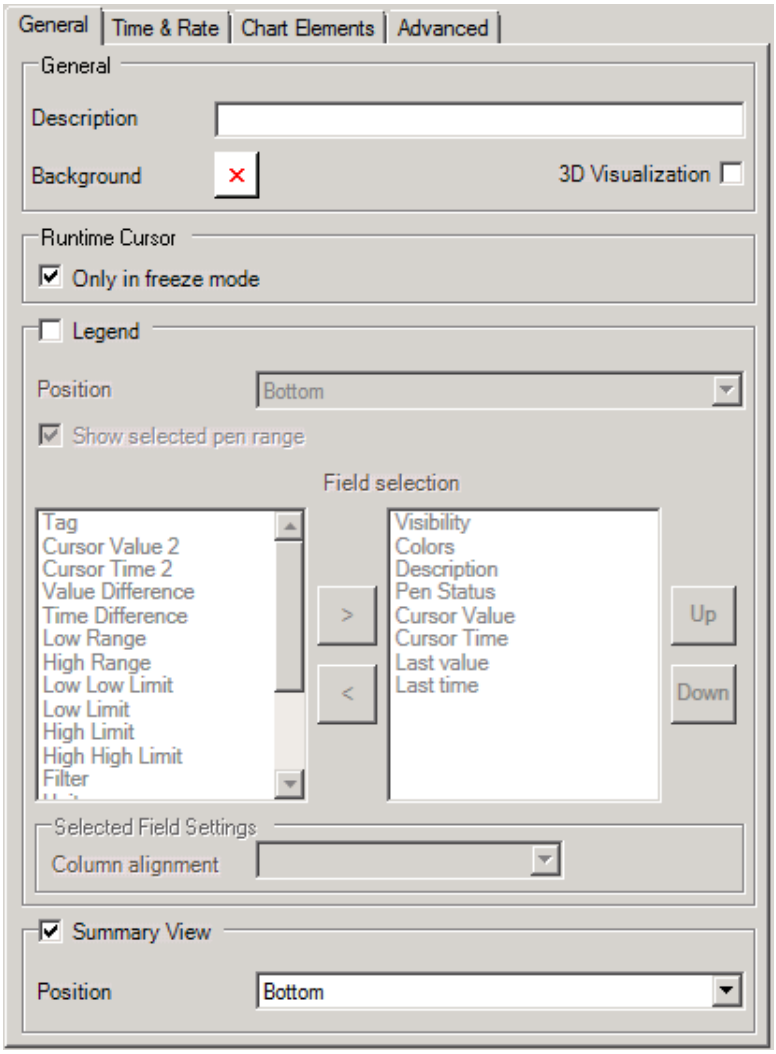


Figure 10.6 – The Chart's General Options

The “3D Visualization” checkbox gives the trend chart a third dimension, shown in the example in Figure 10.7.

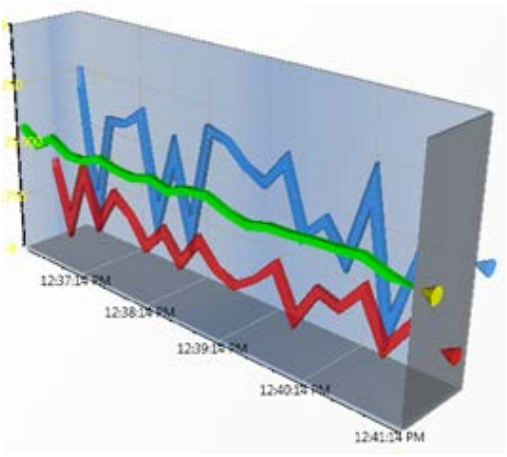


Figure 10.7 - 3D Visualization View

The cursor is always available to runtime users unless you put a checkmark in the “Only in Freeze Mode” checkbox. The “Legend” checkbox allows you to show or hide the legend in both runtime and configuration mode, and you can specify the content of the legend. Runtime users can manipulate pens during runtime if the chart has a legend that includes certain options. For example, if the legend includes the Visibility field, runtime users can show and hide individual pens in the chart. Selecting the “Summary View” checkbox displays a plot window to show the entire range of data that is collected; the Summary View shows only in Freeze mode.

> **Note:** More information about options on the **General tab** is available in the TrendWorX32 Viewer’s online Help. In the Workbench, click the Help icon in the upper right of the Workbench and then in the Contents tab navigate to the TrendWorX64 Viewer topics.

10.4.2. Time & Rate Options Tab

The chart's **Time & Rate** tab, shown in below, determines the data collection rate, data display rate, chart refresh rate, the Trend Period Tool Bar, and other factors that control what is stored in the TrendWorX64 Viewer data buffer and what is displayed. Alter these settings in configuration and runtime modes using the runtime toolbar.

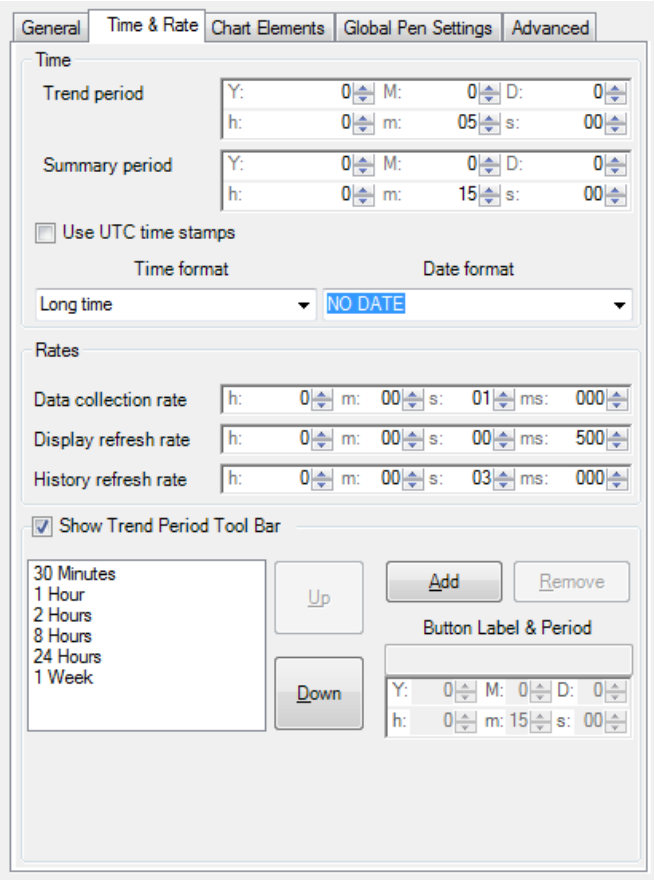


Figure 10.8 - Time and Rate Tab

The properties available that affect time are:

- **Trend Period**: Describes the duration of the interval shown in the Chart object display.
- **Summary Period**: Is the amount of data stored in the data buffer and that can be displayed in Freeze mode in the Summary window.
- **Use UTC time stamps**: If this option is checked, all time stamps stored in the database will be stored as UTC time stamps. This is to accommodate automatic daylight savings time switching in Windows when the TrendWorX64 Logger is logging data. All historical clients and reporting tools will always retrieve historical data in their local (non-UTC) time.

- You can select the desired Data Collection Rate, among other variables. Currently the fastest data-collection rate is 10 milliseconds (ms). The update rate is given by inserting the variables on the **Rates** tab. Therefore, if you change one of these variables, all other variables will be updated automatically.

Notice that the data collection rate is configured independently of the drawing action. The fastest display update rate is 50 ms, with the exception of bar plots, in which the display rate matches the data-collection period.

The **Display Refresh Rate** controls the speed at which the signal curves will be drawn. If, for example, the Data Collection Rate is set to 250 ms and the **Display Refresh Rate** is set to 1 sec, the samples will be collected at 250 ms, but the signal curve will be updated every second without loss of data. This is a great option to minimize system-resource use.

The **History Refresh Rate** controls the frequency at which the historical pens will be refreshed with new historical data from the TrendWorX64 historical database. This can be used as an optimization and load-balancing tool. For example, if the TrendWorX64 Logger is configured to log to the database every minute, then the **History Refresh Rate** should match this rate.

The **Trend Period Tool Bar** can be used to show buttons for dynamically switching the Trend Period by operators during Runtime. The number of buttons and order can be altered using the Up, Down, Add, and Remove buttons on the side. A name and trend period can be set with the time picker that activates when the Add button is clicked.

10.4.3. Chart Elements Tab

Chart Elements options (shown in Figure 10.9) include the color of lines and backgrounds, as well as how the different axes display. You can also alter the ranges that a multi-plot display uses. In runtime, when you click on the Show/Hide Axes button, this same tab displays as a dialog box so that the runtime user can change them. These settings apply to the configuration, runtime, and freeze mode.

You can set the following properties on the Chart Elements tab:

- Background and Line Color
- Show Grid
- X-Axis and Y-Axis
- Fill Style
- Global Range

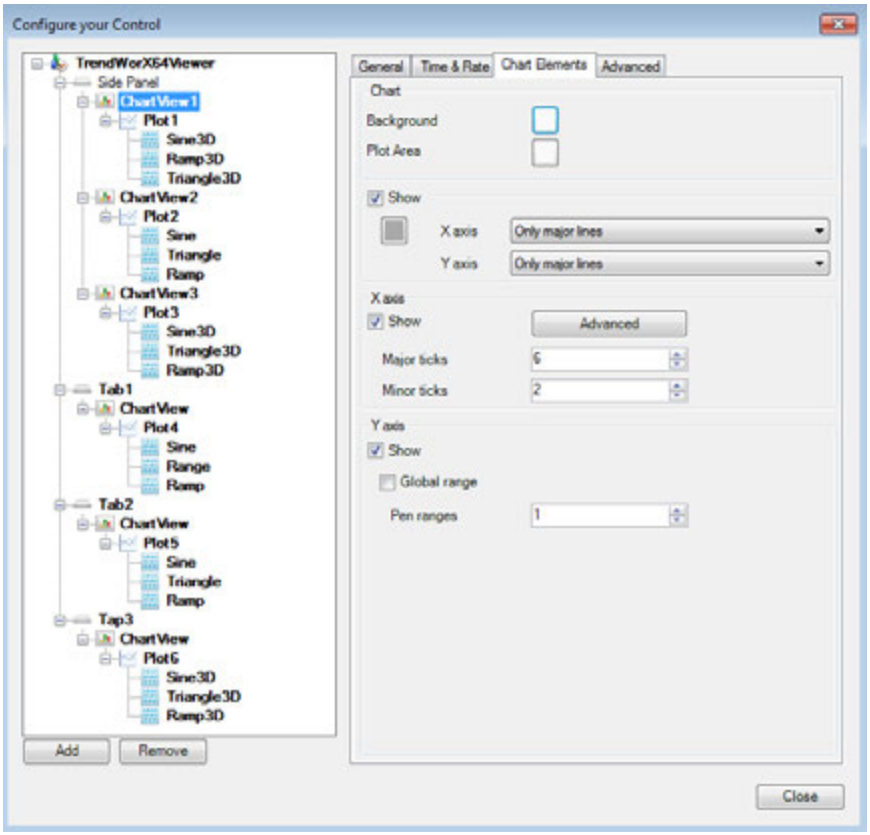


Figure 10.9 - Chart Elements Tab

10.5. Plots and Plot Types

Each chart is "drawn" by pens. There several ways a pen can draw a chart – each way is called a plot type. To choose the plot type a pen will use, you must first add a plot to the chart, and then choose its plot type. You can add any number of plots (i.e., plot types) to a chart. A *plot* is sometimes referred to as a *pen group*. It is a collection of pens that share a plot type. To understand this, you have to understand the plot types that are available.

You can see them on the Plot tab, shown in Figure 10.10. The supported plot types are:

- Line
- Area
- Bars
- Circular
- Pie

If you want to have five pens, three of which are lines (that is, a *line* plot type) and two of which are areas (that is, an *area* plot type), then you would need to create two plots: one for the three line pens and one for the two area pens. Then you would add their respective pens to the plots.

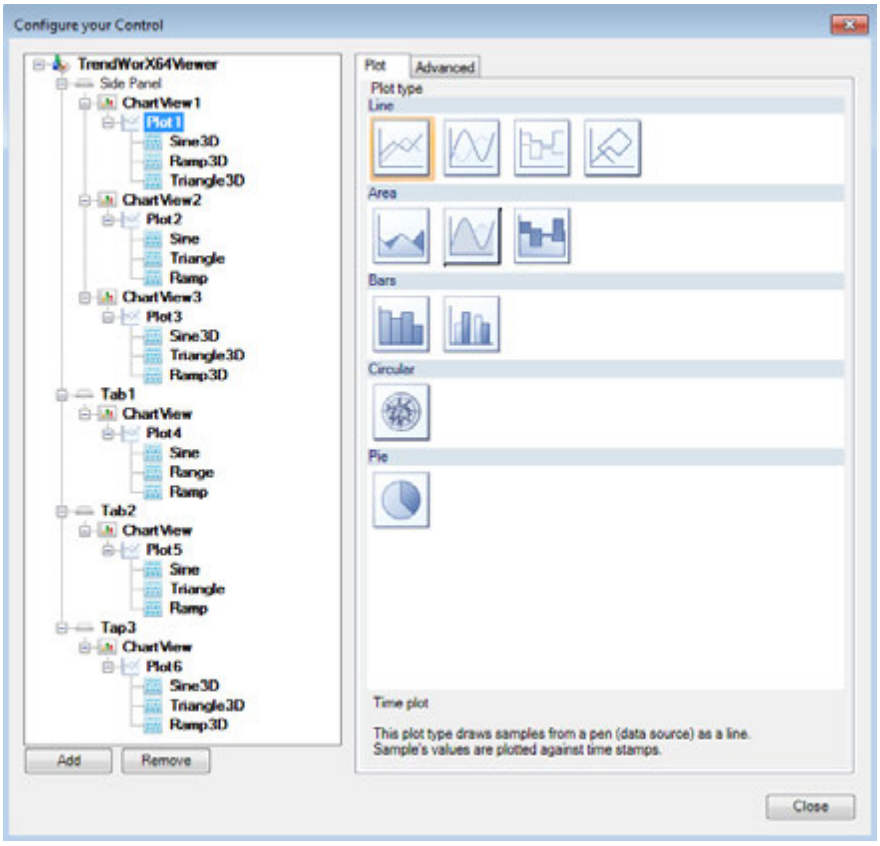


Figure 10.10 - Different Plot Types in TrendWorX64 Viewer

To add a plot (that is, a pen group) to a chart, right-click the chart object and select Add → Plot. Alternatively, you can select the chart object and click the Add button at the bottom of the dialog box. Then simply choose the plot type that you want the pens to use. After that, you can add pen to the plot object – pens are what we will cover next.

10.6. Pen Configuration

A pen is the point that puts values on a chart; it is the connection between the data source and your trend display. A pen writes the data point as well as the connection properties between data points. The following types of signals can be viewed in TrendWorX64 Viewer:

- OPC UA (Unified Architecture)
- OPC DA (Real-time data)
- OPC HDA (Historical data)

Runtime users can manipulate pens during runtime if the chart has a Legend that includes certain options. For example, if the legend includes the Visibility field, runtime users can show and hide individual pens in the chart.

You can have any number of pens of the same and differing styles on a chart. The pen takes on the style of the plot you add it to. To add a pen to a plot, right-click the plot and select Add → Pen, or select the plot object and click the "Add" button at the bottom of the dialog box. The **Pen Configuration** dialog box, shown below, has four tabs of options: the General tab, the Alarm tab, the Ideal Pen tab, and the Advanced tab.

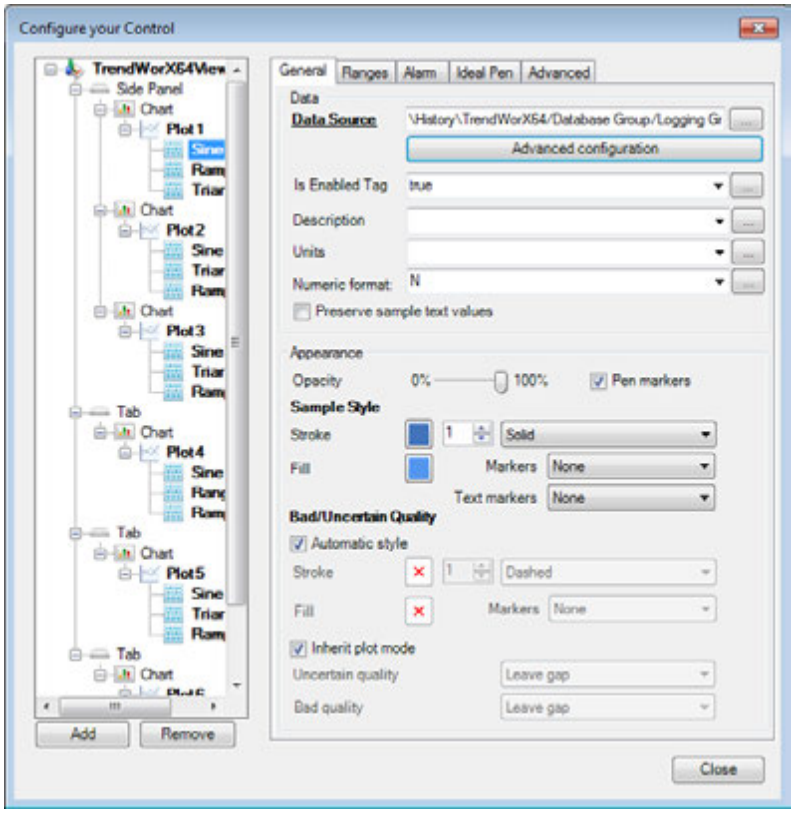


Figure 10.11 - Pen Configuration

10.6.1. General Tab

It is the pen configuration that handles the signal connection editing and configuration for the data displayed in the trend. When configuring a pen, based on what **Data Source** you provide, **Use DA Connection** and **Use HDA Connections** are automatically set; freeing you from worrying about which one to choose. However, you can use the Advanced Configuration button on the General tab to set filters, as shown in Figure 10.12. The **Aggregate Type** option in can be used to filter the samples logged into the logging database, as can a deadband filter for a DA connection.

Note: You can use aliases to construct the path and name of a pen's data source.

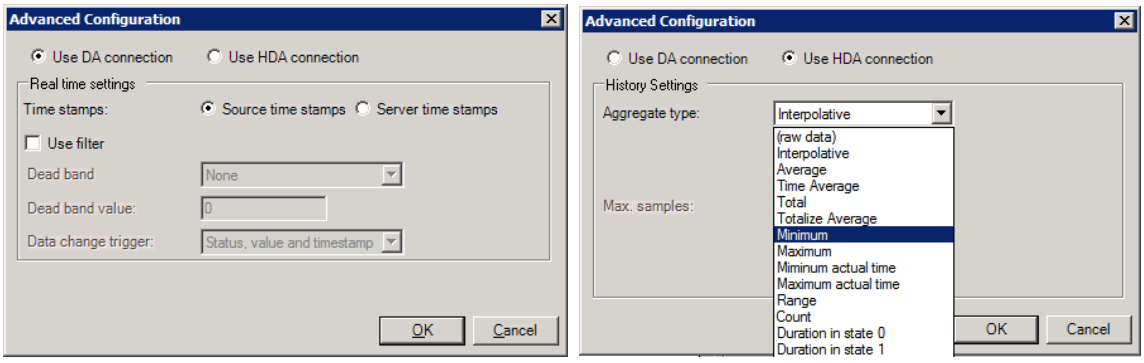


Figure 10.12 - Setting a Real-Time (left) or Historical (right) Data Source

Also on the General tab, a pen's appearance can be modified using "Stroke" and "Fill" color parameters. The "Opacity" setting allows the pen line or area to show the background through it. Setting the "Markers" option to None, Circle, Square, or Triangle specifies the symbol displayed at each data point in the display, which is helpful if you have multiple pens drawing lines. The pen's **Range** can be read directly from the server or automatically scaled. You can also set a fixed range for any pen. The "Configuration Axis" button opens an additional dialog where you can add labels, tick marks and other options to the chart for this particular pen.

10.6.2. Ranges Tab

On the Ranges tab the scale for the pen can be set to **Auto**, **AutoScale** or **Fixed**. Auto will automatically fit the trend to the viewer window. AutoScale will use the marginal percentage to scale the trend in the trend window. Fixed will allow the user to enter in their own values. The Configure Axis button opens up a new dialog for configuration of background color, title, tick marks, labels and base type (for certain plot types).

10.6.3. Alarm Tab

You can optionally set boundaries for the values that the pen writes. (You can choose to display the numerical values of these limits in the chart's legend by using the chart's General tab.) Note that you can click a field's ellipsis button to open the Data Browser where you can select an alias or tag that sets an alarm limit.

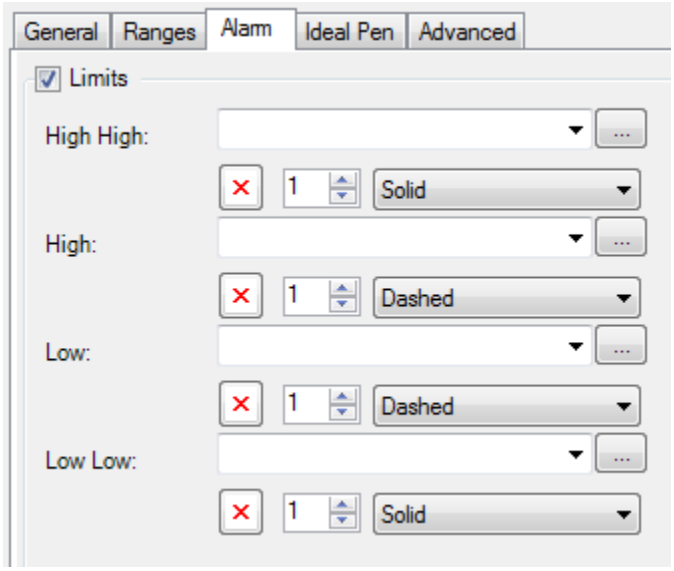


Figure 10.13 - The Pen's Alarm Tab

10.6.4. Ideal Pen Tab

You can configure any historical pen as an ideal pen. This feature allows for powerful comparisons of past historical data to current historical data for the same historical pen. There is no limit on the number of ideal pens in a TrendWorX64 Viewer.

As shown in Figure 10.14, you can use the Ideal Pen tab to make a pen an ideal pen and set its starting time and date.

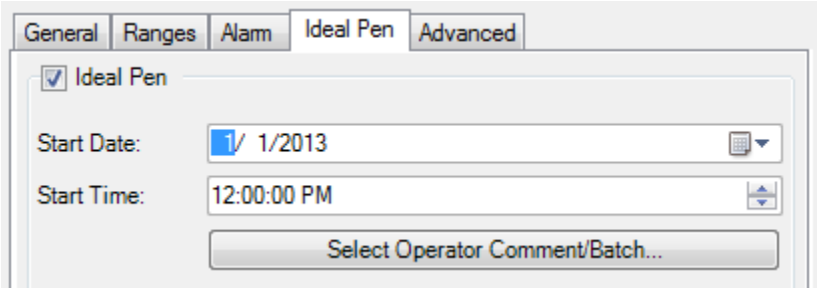


Figure 10.14 - The Ideal Pen Tab

Upon entering runtime, the TrendWorX64 Viewer populates the ideal pen with historical data starting at the selected time and date, while non-ideal historical pens update with current data from the historical database. In runtime, right clicking lets you know if a pen is an ideal pen. The legend Description field will also tell you if a pen is an ideal pen. You can set (or reset) any historical pen as an ideal pen at any time during runtime, and you can modify its starting time and date.

10.7. The Side Panel

One side panel object is created automatically for any Viewer that you create. You cannot remove the side panel or rename it. A side panel that displays beside the tabs you created in your display, as shown in Figure 10.15. Although the side panel has no properties of its own, you can add charts to it and can design those charts in the same way you design them for the viewer's tabs. The trends in the side panel are miniaturized charts.

By default, the side panel appears in runtime if you have selected the Show Side Panel check box in the viewer's Appearance tab, but a runtime user can choose to show the side panel at any time during runtime. If you have not configured a side panel, it will simply be blank.

> **Note:** Not all GENESIS64 features are supported in Microsoft Silverlight edition. Side panels are one of those features. If you are running Microsoft Silverlight edition, refer to the online Help for a list of features that are not supported.

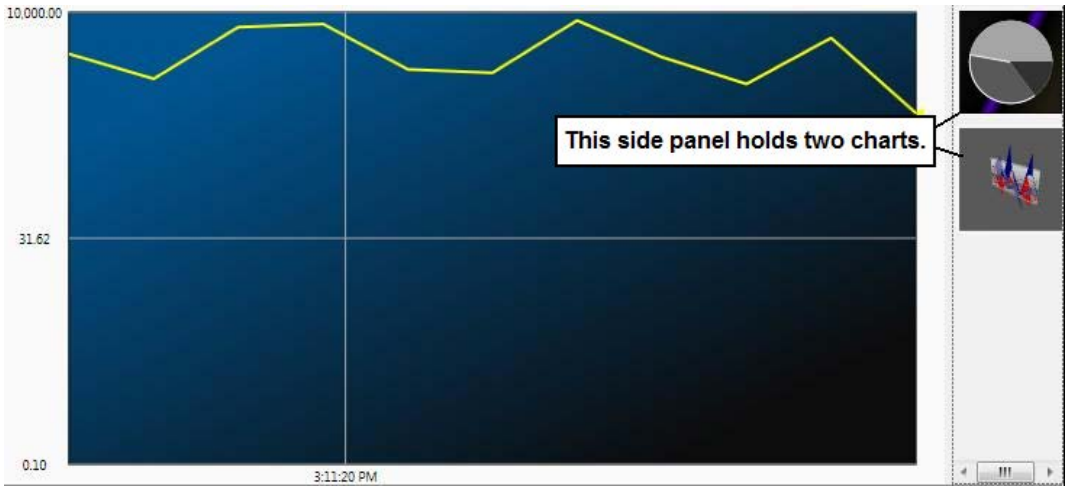


Figure 10.15 - Sample Side Panel

Adding charts to the side panel in the TrendWorX64 Viewer is similar to adding charts to a tab in the TrendWorX64 Viewer. Right-click Side Panel in the navigation tree and select Add, and then Chart. Configure the chart, and then add and configure the plot and the plot's pens.

10.8. Runtime Operations

The Trend Viewer **Runtime toolbar**, is the runtime user interface for the TrendWorX64 Viewer. The toolbar offers you access to various runtime functions and operations. To see the toolbar, you need to have the Enable Toolbar checkbox enabled in your TrendWorX64 Viewer's properties. Unless you set the Initial Position property to Hidden, the toolbar should appear at runtime in the selected position. To show or hide the toolbar, the runtime operator can double-click on the chart area. Notice that you cannot view the toolbar in configuration mode.

If the runtime operator freezes (or pauses) the trend display, the runtime toolbar expands to show additional options. Figure 10.17 shows the toolbar in normal mode while shows the runtime toolbar in freeze mode.



Figure 10.16 - Runtime Toolbar in Unfrozen Mode

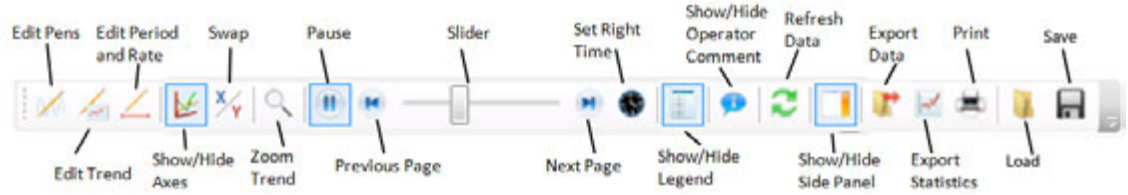


Figure 10.17 - Runtime Toolbar in Freeze or Pause Mode

Many of the buttons on the Runtime Toolbar provide access to some properties configurable in both Configure and Runtime modes, such as **Edit Pens**, **Edit Trend**, and **Edit Period**.

Some options, such as Slider and Previous/Next Page, Zoom Trend, Set Right Time, and Buffer Summary are only available when the viewer is in Freeze mode.

"Edit Pens" opens the chart's **Properties** dialog, allowing you to modify the chart, plot, and pen display. Modifications appear immediately in the runtime display, as soon as you close the Properties dialog.

The “Edit Trend” button opens the **Edit Trend** dialog box shown in Figure 10.18. This dialog gives the runtime user the same options that appear in configuration mode for the chart’s General tab, so that the general appearance, the runtime cursor, the legend, and the summary view can all be customized at runtime.

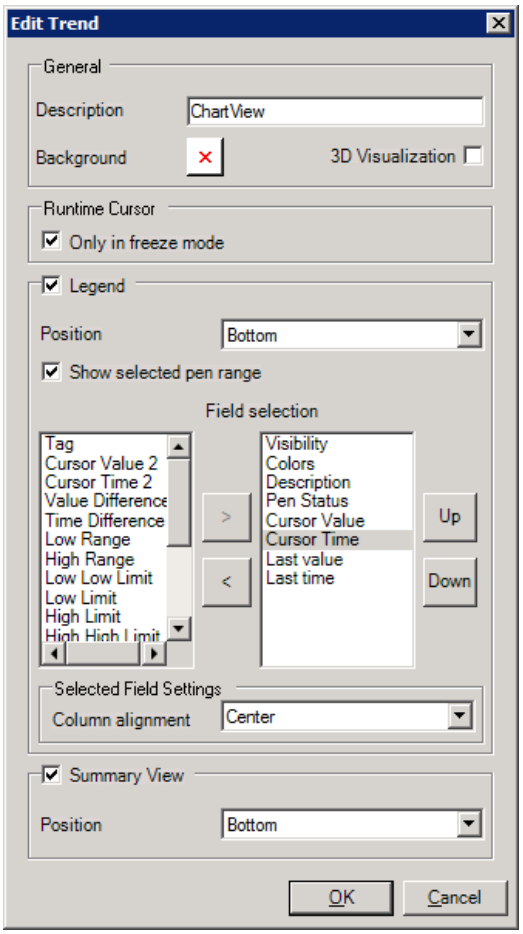


Figure 10.18 - Edit Trend

Clicking on the "Edit Trend Period and Rates" button opens **Edit Period and Rates** dialog box as shown in Figure 10.19.

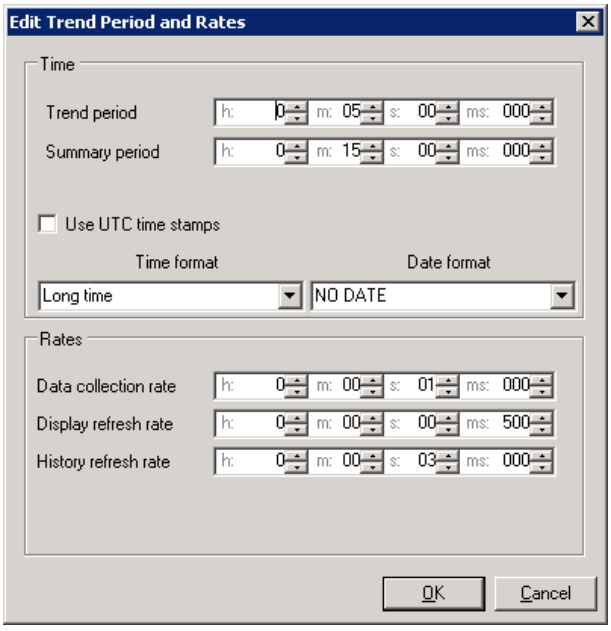


Figure 10.19 - Edit Trend Period and Rates

"Show/Hide Axes" button shows or hides the X- and Y-axes, giving your chart more room to display.

"Swap Y/X Axes" button lets the user switch the X-axis with the Y-axis, which can make visualization of the chart easier in some instances.

Clicking on the "Zoom" button and dragging a rectangle on the part of the chart lets the runtime user zoom in on a selected segment of the trend. The viewer resets the trend period to the boundaries set by the rectangle. This option is more effective when used in Freeze mode because incoming data does not cause the zoom frame to reset.

> **Note:** When used in "real-time mode" – the animated mode – the zoom along the time axis quickly resets (and disappears) as the animation frame is refreshed.

The "Freeze" button stops the trend from updating with incoming data. Freeze mode helps you look more closely at the data for a specific time from. You can use the **slider** at the right of the Freeze button to move forward and backward in time. Clicking the Freeze button a second time exits Freeze mode and returns you to real-time mode.

Clicking the "Set Right Time" button opens the **Set Right Time** dialog, shown in Figure 10.20. This dialog, allows you to enter the time at the end (on the right hand side) of the viewer and move to the data for that time. This option is very effective for reviewing historical data.

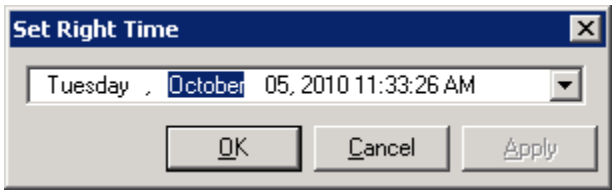


Figure 10.20 - Set Right Time

The **Legend** shows the assignment of pens, as shown in Figure 10.21. You can add and remove columns in the legend by clicking the Edit Trend button on the runtime toolbar.

Visibility	Colors	Description	Pen Status	Last value	Last time
☑		sim64:Float.Rand...	Good	99.67956	9:42:25 AM
☑		sim64:Float.Sine(...	Good	-50	9:42:25 AM
☑		Triangle	Good	14.42	9:42:25 AM

Figure 10.21 - Legend in TrendWorX64 Viewer

Additional options on the runtime toolbar and on the TrendWorX64 Runtime ribbon are described in detail in the TrendWorX64 Viewer's online Help.

The **Summary View** is a powerful feature located at the bottom of the TrendWorX64 Viewer; it is visible only in **Freeze mode** by clicking on the arrow pointing upwards. The Summary pane lets you view all data collected, but lets you choose the time span that displays above, giving you detailed access to collected data. Figure 10.22 shows how the summary view can be manipulated to show a specific timeframe of data.

Note: For more information about how to view trend data at runtime, please refer to the "Runtime Operations" topic in the TrendWorX64 Viewer online Help.

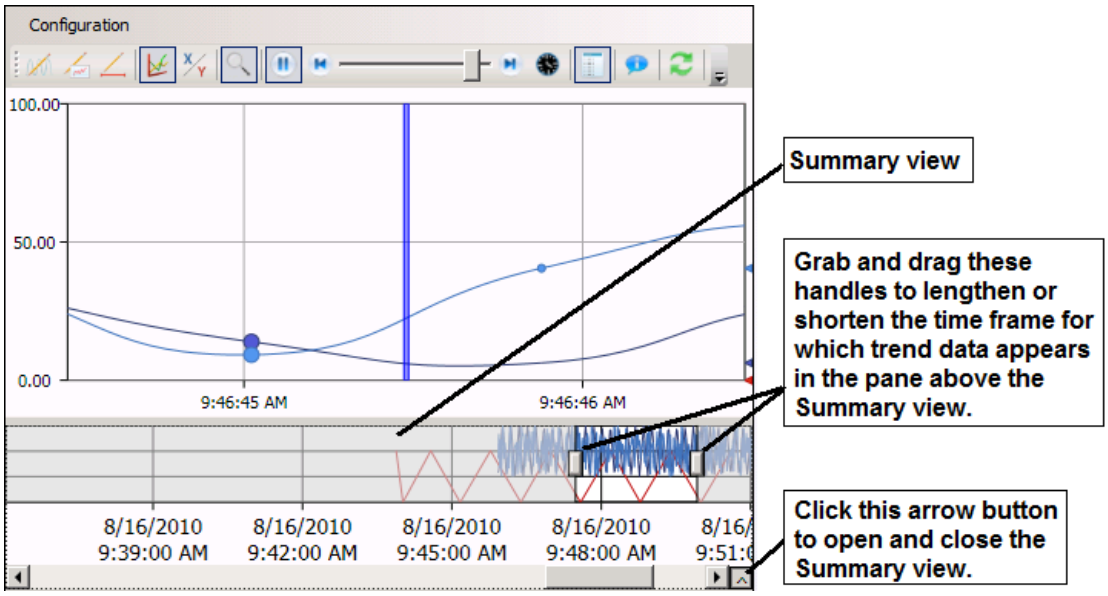


Figure 10.22 - Summary View

10.9. Workbench vs. GraphWorX64

The TrendWorX64 Viewer control can be dragged and dropped into GraphWorX64, where it has the same functionality as it does in Workbench. There are some additional settings found in the GraphWorX64 property sheet.

You can add the TrendWorX64 Viewer control into the GrapWorX64 display by clicking on the **Controls** tab of the GraphWorX64 ribbon and then clicking on the TrendWorX64 Viewer control button as shown in Figure 10.23. Alternatively, you can click on the **Toolbox** and select the TrendWorX64 Viewer control within the toolbox.



Figure 10.23 - TrendWorX64 Viewer Control in the Ribbon

Then the initial Trend Viewer Area can be drawn by clicking and dragging the crosshairs to the desired size that you want the trend to be. The TrendWorX64 Viewer control appears, as shown in Figure 10.24, with its selection handles visible.

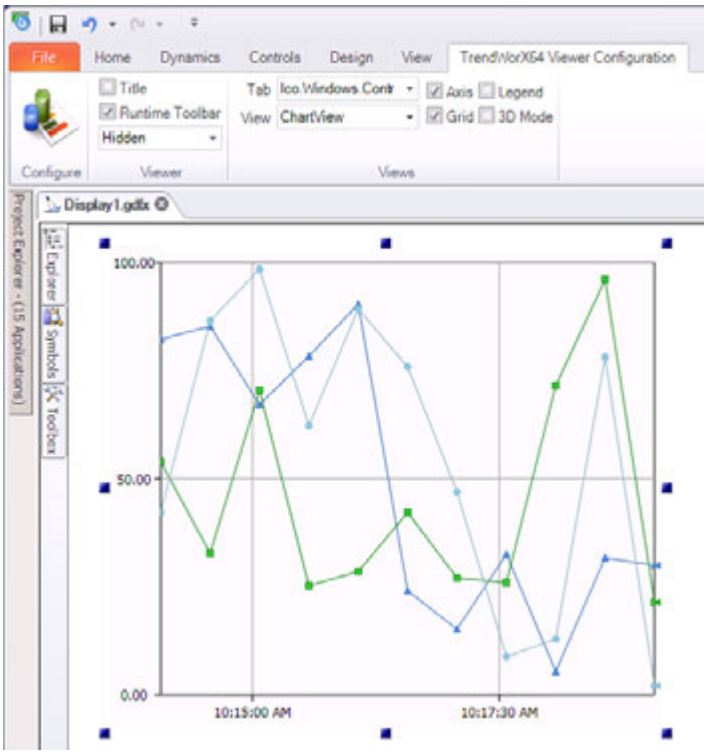


Figure 10.24 - TrendWorX64 Viewer

Note: The control can be resized by grabbing a handle and dragging the control to its new size. In configuration mode, the viewer is simply a graphic and does not display data from the data sources.

A few selected operations can be performed by using the TrendWorX64 Viewer Configuration ribbon shown in Figure 10.24. You can use the ribbon's options to open the TrendWorX64 Viewer configurator, change some of the viewer's settings, and modify options that display in the viewer during runtime.

Lab: Trend Viewer L10

10.1. Lab Overview

IcTonics wants to view trend data that they deem important and have logged, as well as those that they did not want to log, in a graphical format. In this lab, you will customize a TrendWorX64 Viewer for icTonics, as well as explore some options in the control.

Estimated time to complete this lab: 20 minutes

10.1.1. Task Checklist

In this lab, you will complete the following tasks:

- Add a real-time pen
- Add a historical pen
- Change the color of a pen
- Add markers to a pen line
- Create a chart view
- Set up Time & Rate on a chart view
- Multiply a plot

10.1.2. Support Files

To complete this lab, you will need the following file(s):

- icTonicsTrendConfig
- icTonicsTankMonitor.gdfx

10.2. TrendWorX64 Control

You can use the TrendWorX64 control to view your real-time or historical trends in GraphWorX64. Of course, you can do the same with TrendWorX64 container. In this lab, we will create a GraphWorX64 display that shows the proper trend.

1. Open Workbench if you closed it after the previous lab.
2. Create a new GraphWorX64 display and save it as “icTonicsTrends.gdfx”.
3. Insert a TrendWorX64 Viewer onto the design surface by going to the Controls tab of the ribbon, selecting TrendWorX64 Viewer, and then clicking on the design surface.
4. Double-click on the control to bring up the configuration dialog box. (This is the same dialog box that you saw in the previous lab.)
5. Click on "Tab" and change the Title text to “icTonics Trends”.
6. Click on "Chart", and change the Description in the General tab to “Real Time Trends”.
7. Under chart, add a Plot, and then add a Pen. In the tree view, click on the Pen and confirm the General tab is selected. Click on the ellipsis button to the right of the Data Source textbox.
8. In the Data Browser dialog box, browse to the following tag: @ICONICS.IconicsOPCUAServer.V5\icTonics.Tanks.USA.Foxboro.Tank1.TankLevel.Value tag.
9. Click on “OK” to exit out of the Data Browser and return to your TrendWorX64 Viewer configuration dialog box.

10. On the pen's General Tab, click the "**Advanced configuration**" button under the Data Source field. In the popup dialog box, make sure "Use DA connection" is selected.

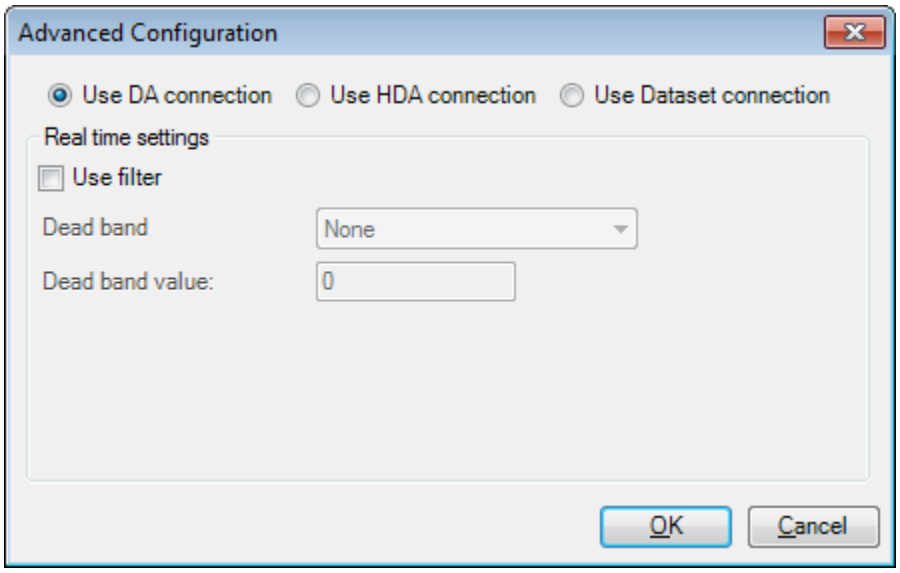


Figure L10.1 - Advanced Configuration

11. On the pen's General tab, change the description of the pen to "Real Time Tank1 Level".

12. In the Appearance section, change the **Stroke Thickness** to 3 and the color to Purple (R:112, G:48, B:160).

13. Your finished pen configuration should look similar to Figure L10.2.

Note: You may have to click the close button in order to save and reflect your changes as seen below.

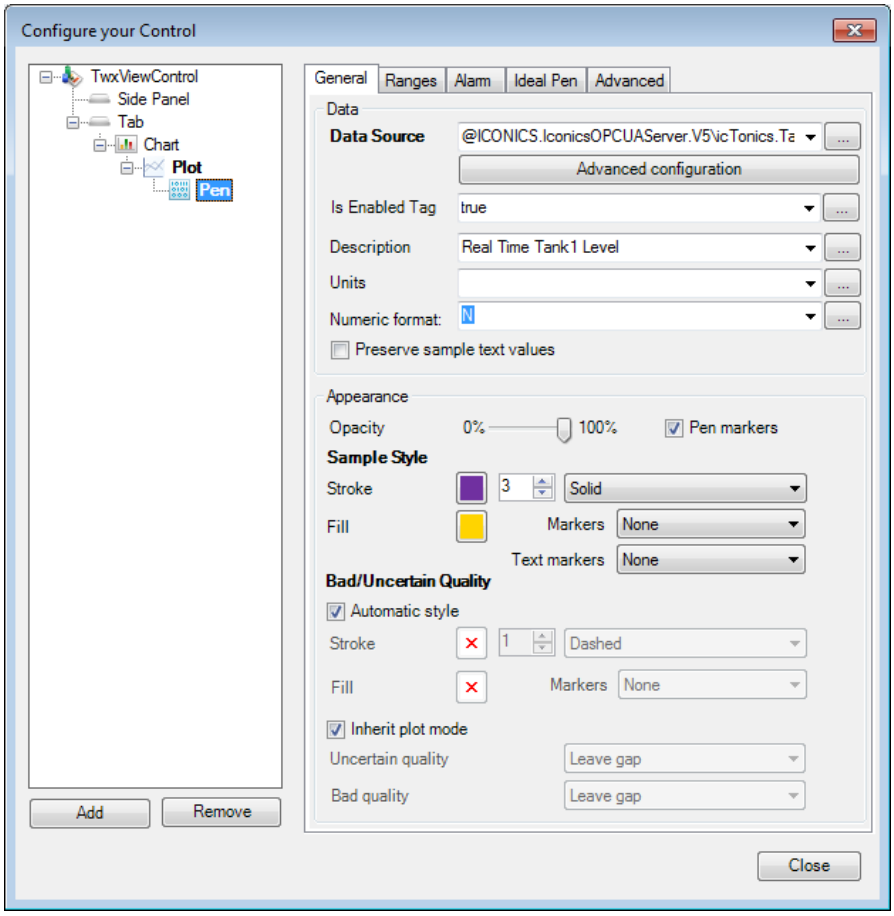


Figure L10.2 - Real-time Pen Configuration

14. Right-click on **Plot** in the left-hand pane and select "Multiply". In the dialog box that comes up, leave the number of items as 1 and click on "Apply" (the green checkmark button).

15. Select the newly created plot (called Plot1) and select the first Area plot as its plot type.

16. Select Plot 1's pen. Notice that the data source for the pen is the same data source you chose for the first plot's pen. Change the Stroke and Fill color to yellow (R:255, G:255: B:0). You are now done setting up this pen.

17. Right-click on Tab and select Add → Chart. Give this new chart the description "Historical Trends".

18. Select its **Time & Rate** tab. In the Rates section, change the "**Display refresh rate**" and "**Data collection rate**" to 5 seconds and change the **History refresh rate** to 30 seconds. As you may recall, we set the Trend Logger Data collection rate to 5 seconds also. Your finished configuration should look similar to Figure L10.3 below.

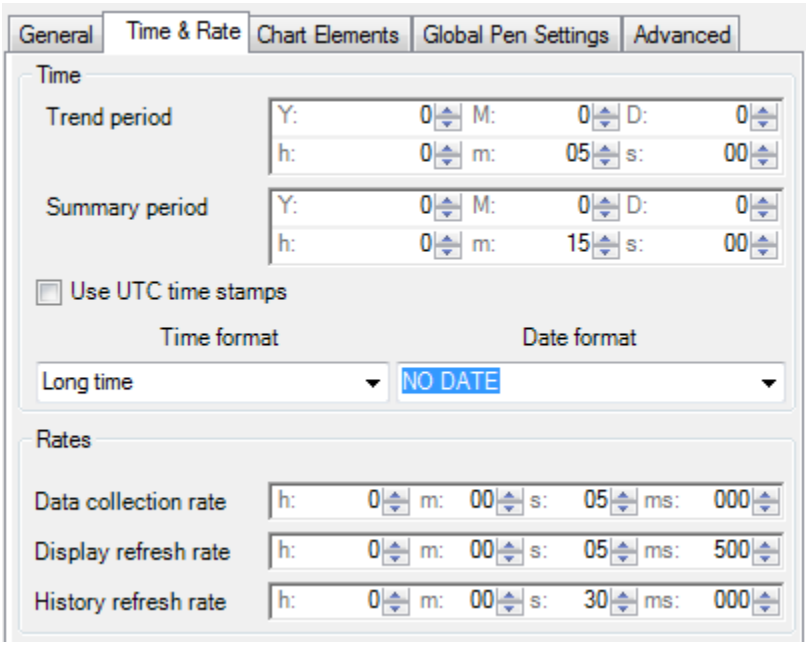


Figure L10.3 - Time & Rate Configuration

19. Repeat the same steps that you executed in the previous lab to add a historical pen. Using the ellipse and the data browser, connect to historical data, tank levels.tank1 level.value. You can refer to section 9.3 of Lab 9 for more details.

20. Give this pen the description "Tank1 Level". In the General tab, change the Stroke to 3 and set Fill and Stroke color to Dark Blue (R:0, G:112, B:192).

21. Select “Circle” as the **Markers** type. Your finished historical pen configuration should look similar to Figure L10.4. Click on “Close” to close the configuration dialog box.

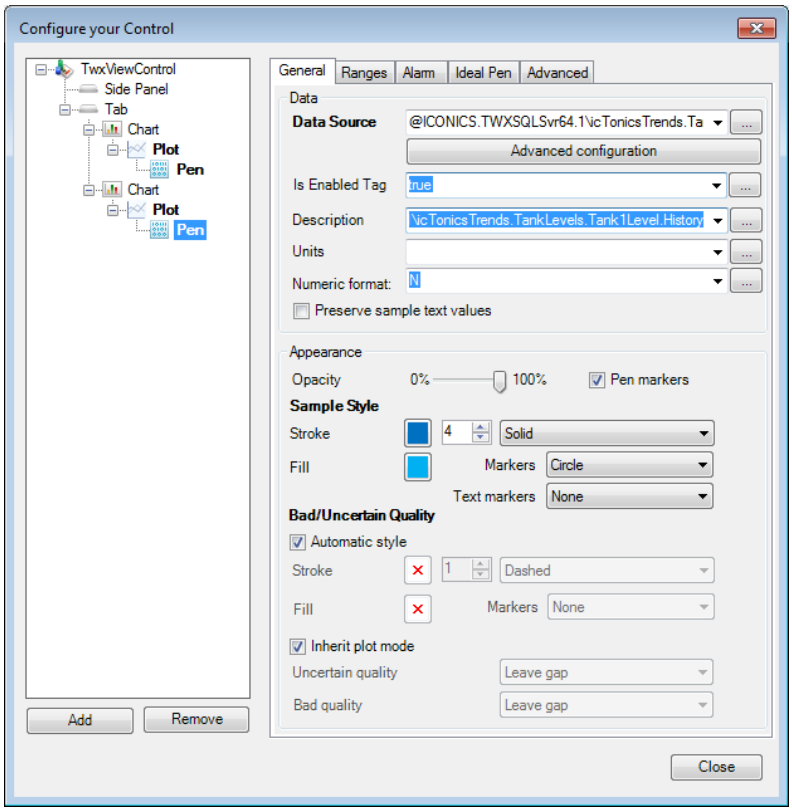


Figure L10.4 - Historical Pen Configuration

22. Notice that your historical trend has circles indicating data collection points, as shown in Figure L10.4.

23. Change the Tank1 Level tag a few times. See that your real time trend shows these changes immediately, but it takes about 30 seconds for the historical trends to refresh. If you leave the trend running long enough, your curve from the historical trend should look like your real-time trend.

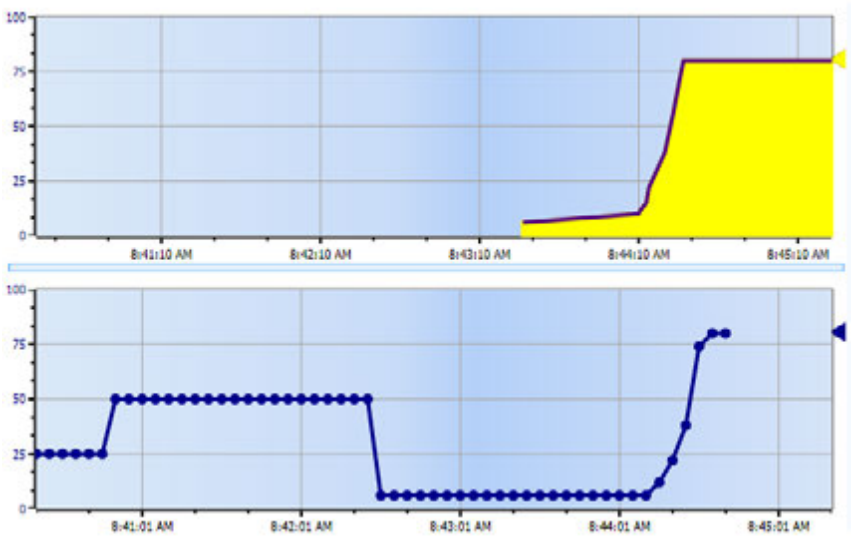


Figure L10.5 - TrendWorX64 Viewer

10.2.1. Trend Popup Display

We will now create a popup window using the trend display we created earlier.

1. In the icTonicsTankMonitor.gdfx display, add a dynamic **Pick Action** to the BaseTank. Hint: locate and select the BaseTank object under **Group** in the Explorer. We are going to configure the pick action to open a popup window.

2. In the Pick action's properties, under CommandParameters set the FileName to "icTonicsTrends.gdfx" and the TargetType to "Popup Window (Display Dependent - Modal)".

3. Set the OverrideWindowProperties to **true** and set the Windows properties as shown in the following Figure.

Note: You can double click in a field and toggle between values.

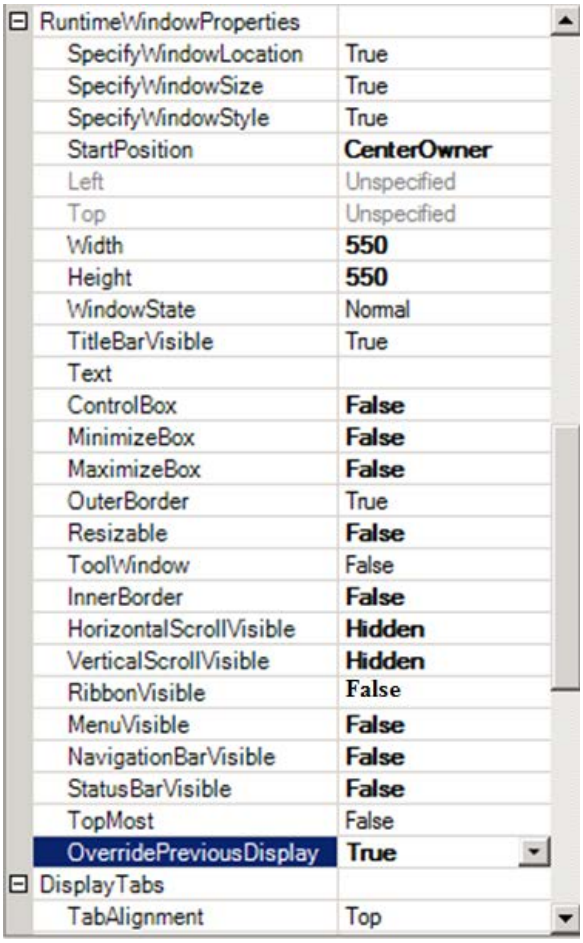


Figure L10.6 - Popup Window Properties

4. Save the icTonicsTankMonitor display and open the icTonicsTrends.gdfx display if you closed it. Apply the template to the display.

5. Select **Display: Canvas**. You can see when you have the display canvas selected by viewing the top of the properties explorer on the right. Change the display size height and width to 550 x 550 and resize the template and the controls to fit within this screen.

> ***Tip!*** You can group everything in the template together to make it easier to resize everything. Remember that you can hold down the "Shift" key to preserve aspect ratio as you resize an object.

Note: You may need to enter your username and password to unlock (click in the Explorer on the left side) the template before you can edit it.

6. Add a text label to the display and change the label text to "Tank 1". Put this label in the middle of the Banner of the template.

7. Add a button to the display. Change the button's Command to "Close Window", Double click on the button and change the button label to "Dismiss".

8. Place the button at the bottom-right corner of the display. Your finished configuration may look similar to Figure L10.7. Save this file.

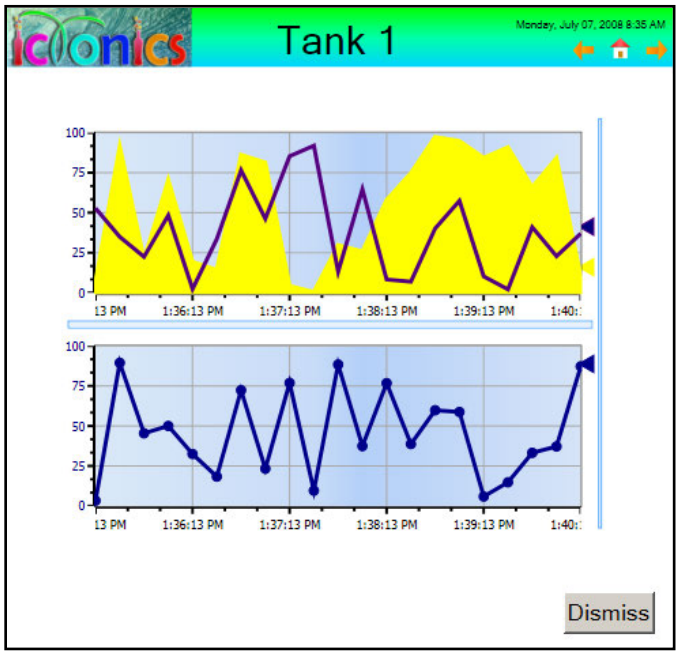


Figure L10.7 - Popup Trend Display

9. Go back to your icTonicsTankMonitor display. Click on the tank to see that your trend window pops up.

Note: You can click the Show/Hide button or right click in the header to turn runtime options on and off.


ICONICS

Review: TrendWorX64 Viewer R10

10.1. Objectives Review

You should now know and understand the following topics:

- TrendWorX64 Viewer properties and different kinds of modes
- Chart options
- The side panel
- Tabs, plots, and pens
- Time and rates
- Runtime operations
- Workbench vs. GraphWorX64 displays

10.2. Summary

TrendWorX64 Viewer is a client application, which lets the user display real-time and historical OPC data. It can be run from the ICONICS Workbench or GraphWorX64 in Configuration, Runtime and Freeze modes.

The TrendWorX64 Viewer contains one or more tabs and a side panel. A tab can contain one or more charts. The same is true for the side panel. A chart can contain one or more plots, and a plot can contain one or more pens.

You can access the TrendworX64 Viewer properties in Configuration mode by clicking on the Configure button on the TrendWorX64 Viewer Configuration ribbon. Each object in the viewer has its own configuration interface. For example, a Chart is edited using the General, Time & Rate, Chart Elements, and Advanced tabs.

You can change the general appearance of the viewer by selecting the TrendWorX64 Viewer Control object in the navigation tree. In a chart's General tab, you can check the 3D Visualization checkbox to show the chart in 3D. You can also enable and customize the Legend to make the visual analysis easier.

A chart's options on the Time & Rate tab determine the data collection rate, data display rate, history refresh rate, and other factors. Plot types can be added and changed, and pens are added to handle the signal connection for the data display.

The runtime user can access the Runtime Toolbar. Many of the buttons on the Runtime Toolbar provide access to properties that are configurable in Configuration mode and Runtime mode, such an Edit Pens, Edit Trend, and Edit Trend Period. The Summary View, which can only be visible in Freeze mode, lets the runtime user view all data collected while moving and customizing the detailed view's time window.

10.3. Terms

3D Visualization	Chart	Data Collection Rate
Display Refresh Rate	Freeze Mode	GENESIS64 Controls
History Refresh Rate	Ideal Pen	Legend
Markers	OPC HDA	Pen
Plot and Plot Type	Side Panel	Summary View
Tab	TrendWorX64 Viewer	UTC Time
X-Axis	Y-Axis	

10.4. Test Your Knowledge

In this chapter, we discussed the TrendWorX64 Viewer. Before continuing to the next chapter, make sure you can answer the questions below:

1. The TrendWorX64 Container is used to view and analyze trends in which of the following?

    a. OPC UA (Unified Architecture) data
    b. Real-time OPC Data Access (DA) data
    c. OPC Historical Data Access (HDA) data
    d. All of the above
    e. Both A and B
    f. Both B and C
    g. Both A and C
    h. None of the above

2. The TrendWorX64 Viewer supports which of the following views:

    a. 2 dimensional
    b. 3 dimensional
    c. Hyper dimensional
    d. All of the above
    e. Both A and B
    f. Both B and C
    g. None of the above

3. The TrendWorX64 Viewer container can be placed into which of the following?

   a. The Workbench
   b. GraphWorX64
   c. Hyper Historian
   d. All of the above
   e. Both A and B
   f. Both A and C
   g. None of the above

4. True or False: TrendWorX64 Viewer supports batch and lot tracking comments.

5. True or False: A plot is a collection of pens that use the same plot type.

6. Which of the following tabs is part of the chart options?

   a. General
   b. Time & Rate
   c. Chart elements
   d. Advanced
   e. All above
   f. None of the above

7. True or False: The runtime toolbar provides access to some properties that are configurable in Configuration mode.

8. True or False: The Swap Y/X Axes button allows you to switch the X-axis with the Y-axis.

9. The Summary view can be seen only in which of the following?

   a. Runtime mode
   b. Configuration mode
   c. Freeze mode
   d. None of the above

10. True or False: The Side Panel allows you to switch between the different types of charts predefined in TrendWorX64 Viewer.

11. True or False: You can set boundaries for the values that a pen writes by using a different plot type.

12. What handles the signal connection for the data displayed in the trend?

    a. The pen
    b. The Side Panel
    c. The tab
    d. The plot
    e. The chart
    f. The Viewer
    g. The plot type

Unified Data Manager 11

The Unified Data Manager (UDM) provides a centralized approach for managing and configuring commonly used items. You can configure expressions, value sets, tag groups, triggers, and registers by using the Unified Data Manager configuration interface, shown in Figure 11.1, in the Workbench. All of the items and their settings are stored in a dedicated SQL database. These items are exposed as OPC items and can be accessed from ICONICS applications via the Data Browser. By using the UDM, you can reduce development time and eliminate redundant work.

You can open the UDM within the Workbench by clicking the UDM icon in the Project Explorer or from the Start menu command in the GENESIS64 Tools folder.

The purpose of this chapter is to familiarize you with the capabilities of the Unified Data Manager. You will start by learning the operations that are common to all objects, such as creating items, folders, editing, etc. Then this lesson will proceed to each node in the UDM navigation tree. In turn, you will learn what expressions, groups, value sets, registers, and triggers are; what capabilities each has; and how you can use each in your work. After completing this chapter, you should know how to:

- Create items in the Unified Data Manager that you can use throughout GENESIS64
- Simplify application development using expressions, value sets, groups, registers, and triggers.

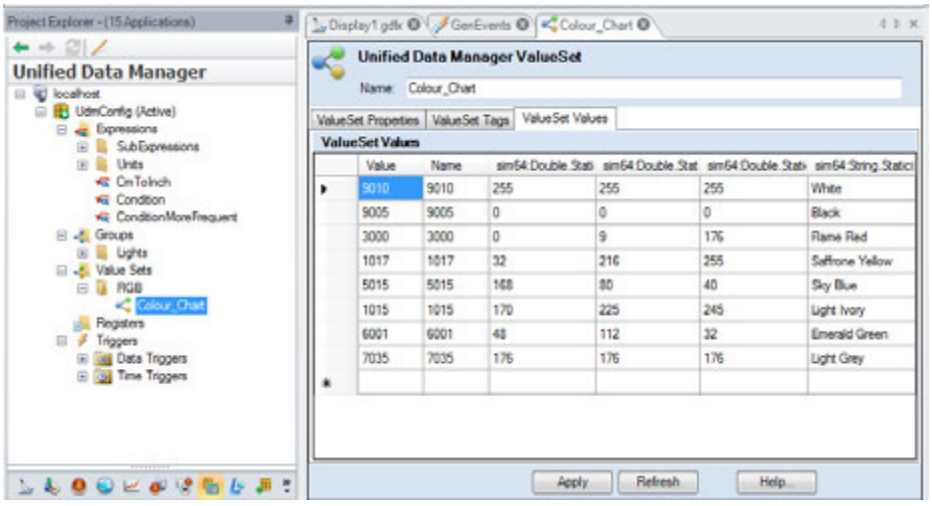


Figure 11.1 - Unified Data Manager Configuration

11.1 Working with Objects in the UDM

The layout of the Unified Data Manager configuration dialog consists of a tree control on the left and a property dialog that opens on the right when you edit an item from the tree. The hierarchy of the tree explorer starts at the level of your host machine, and then proceeds down from the database object to the nodes representing your item collections. Item collections can contain both folders and items.

As mentioned earlier, the configuration is stored in a SQL database. This configuration database can be created using the **SQL Create Database** dialog. You can also use the **SQL Attach Database** dialog to attach additional UDM SQL databases. Both of these dialog boxes can be launched using the "Database" button in the Edit section of the Home tab in the ribbon. Alternatively, you can launch them by right-clicking on the host machine item in the Project Explorer, and then selecting "New Database" or "Attach Database" from the popup context menu, as shown in Figure 11.2. Keep in mind that although you may see additional databases, only one database may be active at any time. Use the "Database" button on the Home ribbon or the popup menu to work with databases.

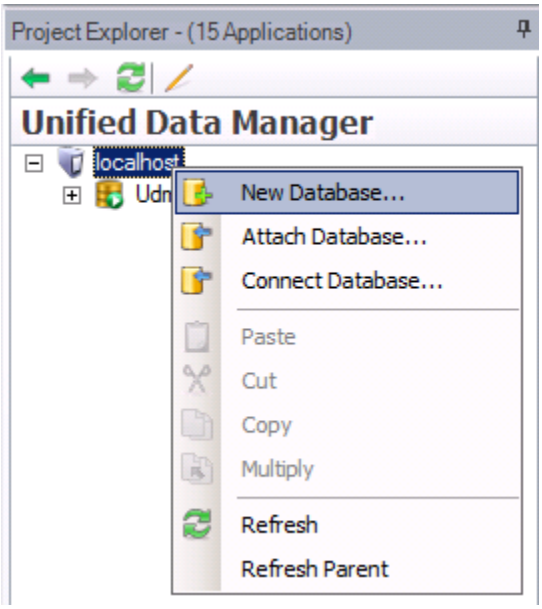


Figure 11.2 - Launching the Database Wizard

In the UDM navigation tree, the nodes **Expressions**, **Groups**, **Value Sets**, **Registers**, and **Triggers** are all at the same organizational level. You cannot add any new type of node at their level, but you can expand them and add sub-items to them. You can create as many folders as you wish, and give them names that are meaningful to you. Folders are listed alphabetically, followed by items that are contained within a folder. You can also create as many items as you wish, up to the limits of your database.

To create an item in the navigation tree hierarchy, right-click on the object you are adding the item from the menu:

- To **create a folder**, select the **Folder**
- To **create an item**, select **Expression**, **Register**, **Value Set**, **Register**, or **Trigger**. Your item choices vary depending on to which folder you are adding the item.
- To **edit or modify an item**, right-click on the item and select **Edit**, **Edit On New Tab**, **Rename**, or any other edit command.

Figure 11.3 shows the popup menu on a folder and on a specific item.

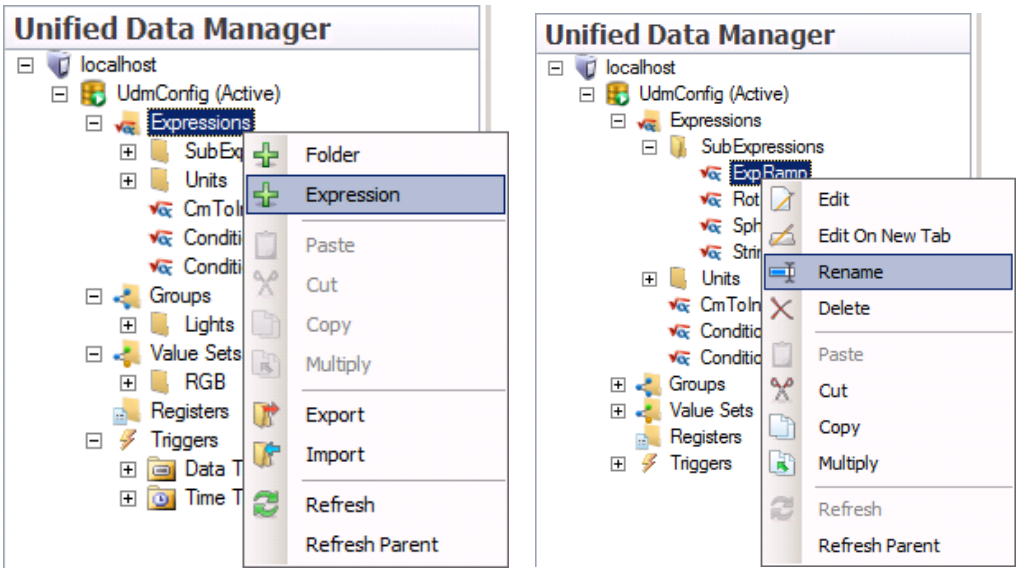


Figure 11.3 - Right-click Menus for Folders (Left) and Items (Right)

Nearly all of the work done in the UDM is found in configuration forms. The remainder of this chapter focuses on these different forms. Note that the configuration forms for folders have one and only one text field in which you can assign or edit the name.

11.2 Expressions

An expression is a function that takes input, performs a specified calculation, and returns an output. The calculation is executed periodically, at a user-configured rate. In expressions, you can use parameters and make the expressions writable. Such expressions can be accessed from a large variety of ICONICS products as single tags.

Thanks to their writable nature, expressions can be used for unit conversions. Consider the user-created function called *celsius_to_fahrenheit,* shown in Figure 11.4. The expression takes as its input a thermometer reading in degrees Celsius and converts the value to degrees Fahrenheit. This function will read every input and trigger tag every 100 milliseconds, and will return a Short data type. Because this function might be used with different thermometers, there is a parameter, named *Thermometer* defined, which stands for the currently selected thermometer.

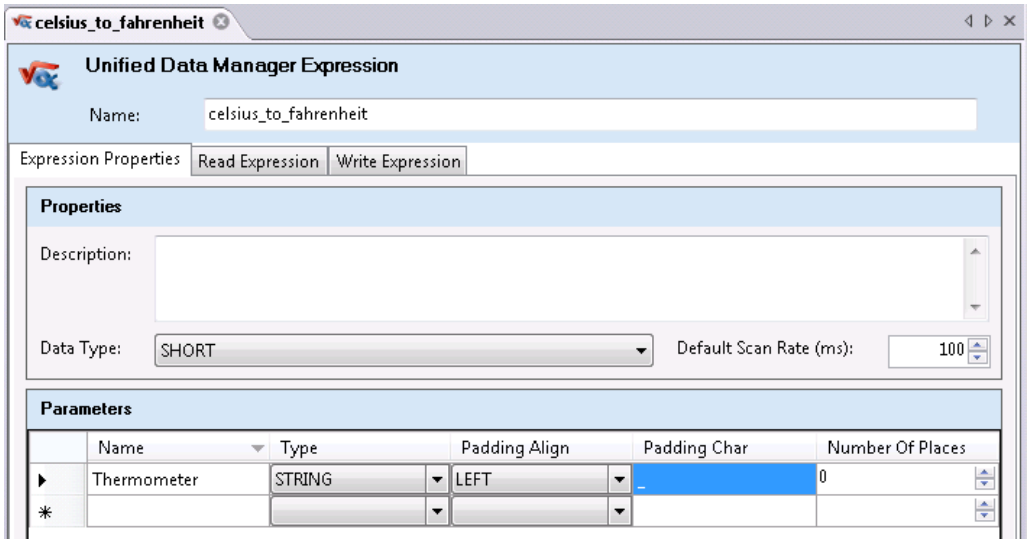


Figure 11.4 - Creating an Expression

Note: Parameters can be easily discerned in an expression formula because they are enclosed in double carats (for example, **<<Thermometer>>**), as you will see later.

The essential actions of an expression are contained on the **Read Expression** and **Write Expression** tabs. There are some minor but important differences between the options that you see on these tabs, both of which are shown in Figure 11.5. In this figure, note the different content of the Read and Write expressions.

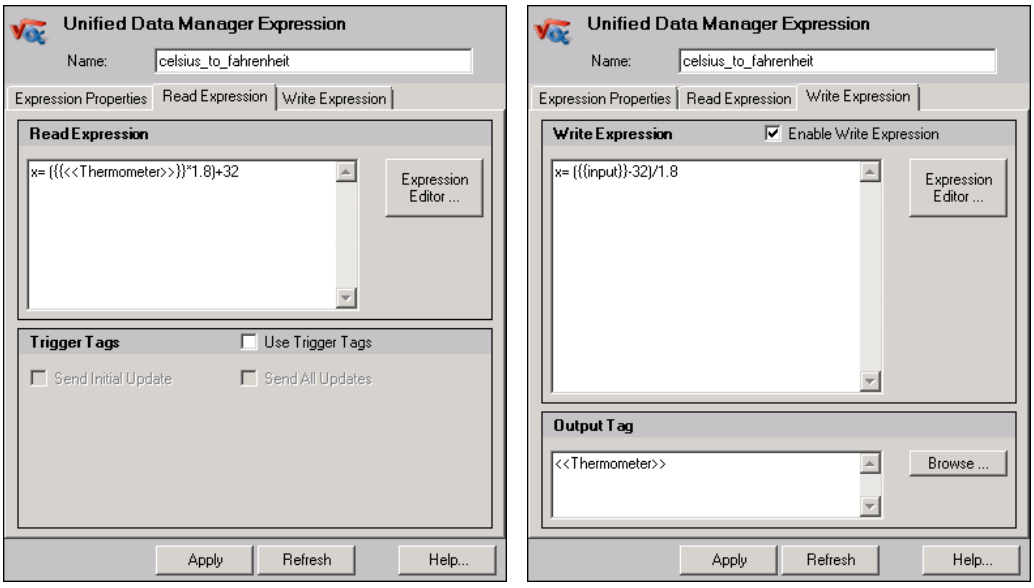


Figure 11.5 - The Read Expression (Left) and Write Expression (Right)

It is important to understand that you do not need to configure the **Write Expression** tab at all unless you intend to write values to the expression and perform a calculation with the written value. If you do need to perform such a calculation, you should remember that the input value, which an operator writes into the expression tag, is represented by a special string called **{{input}}**.

> **Note:** This tag is case sensitive. This means that if you type in a tag that says {{Input}} or {{inPut}} will not work.

In the Read Expression tab is the expression for calculating the conversion, which reads:

x= (<<Thermometer>>*1.8) + 32

Double triangular brackets "<<" and ">>" surround the Thermometer parameter. This parameter will be substituted with actual OPC tag (or other values) when a client application uses the expression. In this case, the substituted value would be a read/write OPC data point (this is because in the example we are also going to write into the point, as you will see later).

Notice that each tab contains a button called **Expression Editor**. The Expression Editor is a dialog that allows you to create expressions (and other functions), then checks for the validity of the expression.

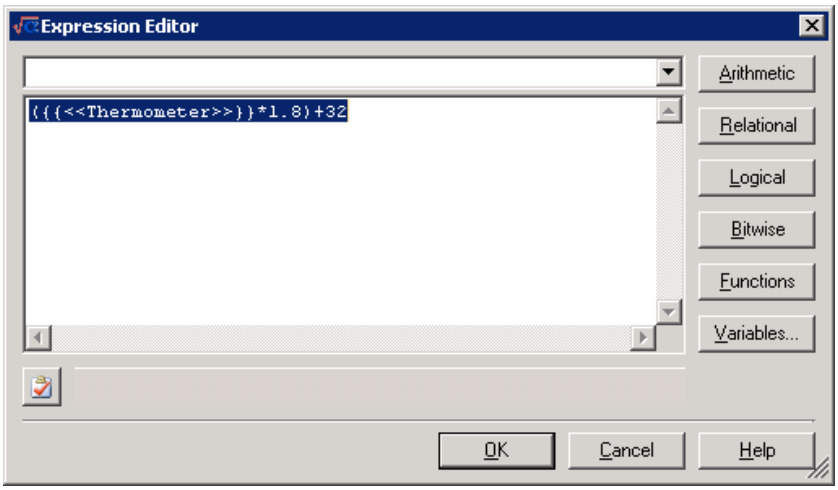


Figure 11.6 - Expression Editor

Each button in the Expression Editor displays a list of available functions. In cases such as the **Variables** button, the OPC tabs of the Unified Data Browser are opened to allow you to make an assignment based on tag values. The important thing to remember about the Expression Editor is that although it checks your expression syntax, it does not check whether the expression is correctly applied.

You do not have to use the Expression Editor at all. Instead, you can write your expression directly into the text box on the Read Expression tab. The expression will be evaluated whenever one of its inputs changes; or you can put a checkmark in the checkbox "Use Trigger tags" so that the expression will be calculated whenever one of the trigger tags becomes true, becomes false, or simply changes value as specified by the user.

Write expressions require that you check the "Enable Write Expression" checkbox to have the values calculated. They are configured similar to Read expressions with two exceptions. First, you should mark the place in your expression where you expect an operator's input. This place is represented by the special string **{{input}}**. For instance the write expression of the thermometer is:

x= ({{input}} - 32) / 1.8

Second, the **Write Expression** tab contains an "Output Tag" textbox where you should specify a data point where the result of the write expression will be stored. In the case of a conversion, it often makes sense to use the same tag as the one used in the read expression. In the case of our example, it will be our parameter.

It is relatively easy to create expressions, and the tools provided allow expressions to be as complex as you need. Using the expressions that you created is even easier.

As an example of using an expression, consider a process point on a GraphWorX64 display whose data source is set to the *celsius_to_fahrenheit* expression. Clicking on the ellipsis button in the DataSource property of a process point opens the Data Browser. The UDM's expressions can be found under the **Global Expressions** folder on the server that holds the Unified Data Manager configuration. In Figure 11.7, you can see the *celsius_to_fahrenheit* expression selected in Data Browser.

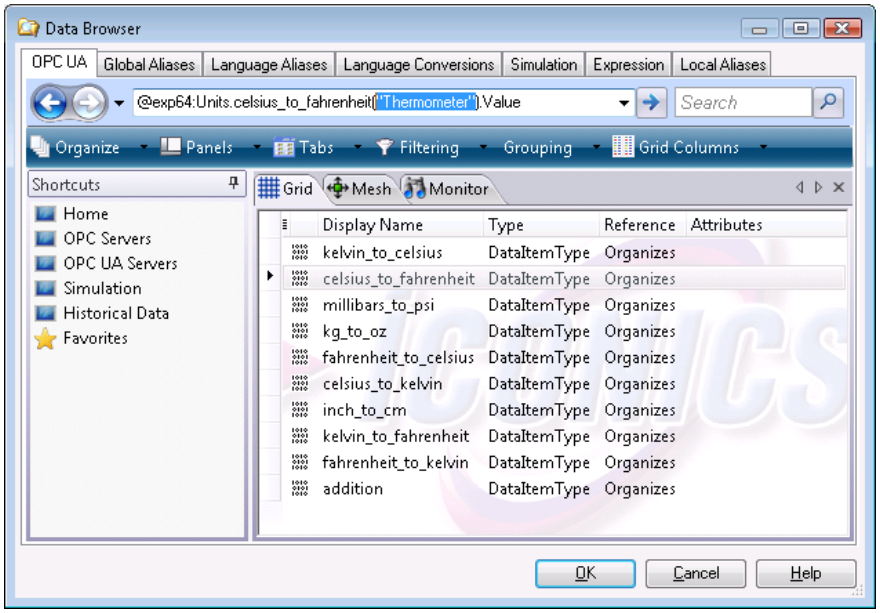


Figure 11.7 - Selecting Expression as a Data Source in the Data Browser

Note the string **"Thermometer"** enclosed by parenthesis is the data point name. You should replace the parameter with the name of the real thermometer OPC data point enclosed in quotes. You can do this by highlighting the parameter (Thermometer in this case) and browse to the OPC point to select it. The resulting tag path may look similar to this:

@exp64:Units.celsius_to_fahrenheit("@KEPware.KEPServerEx.V4\Channel_1.Device_1.Tag_1.Value").Value

11.3 Groups

A group is a collection of data tags that are manipulated as a single item. It is represented by a single group tag. Each group tag is treated as a READ/WRITE data type. Any value written to a group tag is written to all the tags included in the group, so that all of the tags in the Group have the same value. For this reason, use Groups for tags you want to write a uniform value. For example, this feature can be deployed for controlling lights in a building. You can switch on or off all lights on a particular floor by changing only one single value. The figures in this section show a group tag that controls lights on floor one of a building.

Group data tags are organized in a tree structure of folders and group items. Since each group is represented as a data tag, a group data tag may also be included in other groups. Group items have two configuration forms.

On the **Group Properties** tab, you can change the data type of the tag, add its description, and create parameters that can be used in the same way as parameters in expressions or other UDM items.

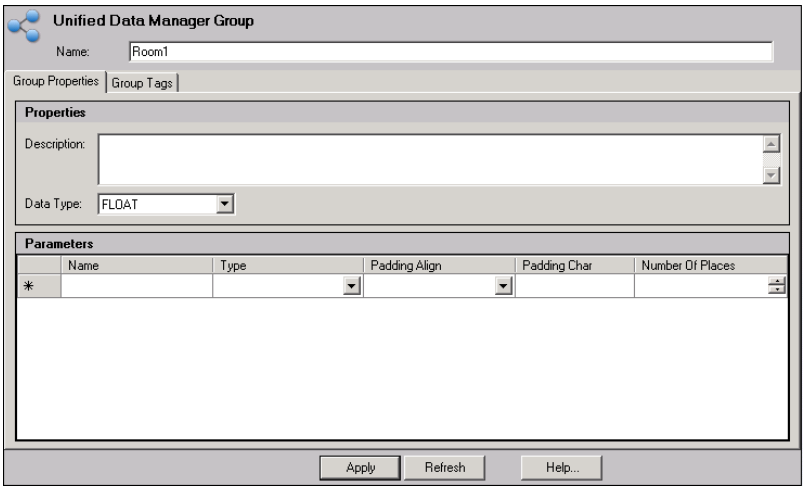


Figure 11.8 - Group Properties Dialog of Group Tag Item

On the **Group Tags** tab, you can add tags to the group tag. Whenever a value is written to the group tag, it is automatically written to all tags in this list.

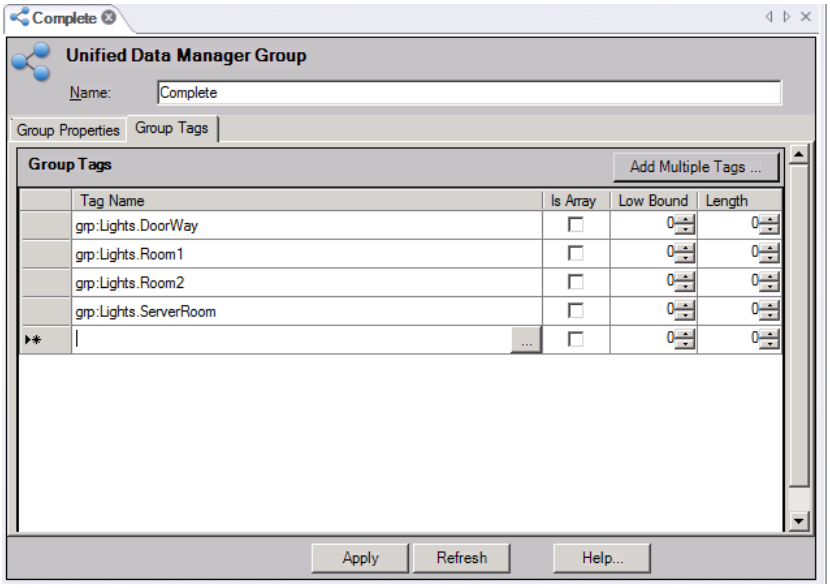


Figure 11.9 - Group Tags Dialog of Group Tag Item

A group can contain an unlimited number of data tags. Each data tag consists of the following attributes:

- **Tag Name:** A fully qualified data tag path (parameters or expressions can also be used) represented as a string. You can type in the tag path or select the source by using the Data Browser, which opens when you click on the ellipsis […] button next to the selected field. You can also use the "Add Multiple Tags" button to add in many tags at the same time.
- **Is Array:** If the tag is an array, put a checkmark in this column to indicate that the tag is an array.
- **Lower Bound:** specifies the lower limit of the tag if it is an array.
- **Length:** specifies the length of the array if the tag is an array.

GraphWorX64 provides an interface for writing values into a group tag. You just need to insert a Data Entry Point into a display and then browse for your group tag in Data Browser by clicking the ellipsis […] button to open the UDM and browsing in the **Groups** folder (highlighted in Figure 11.10) on the server that holds the Unified Data Manager configuration.

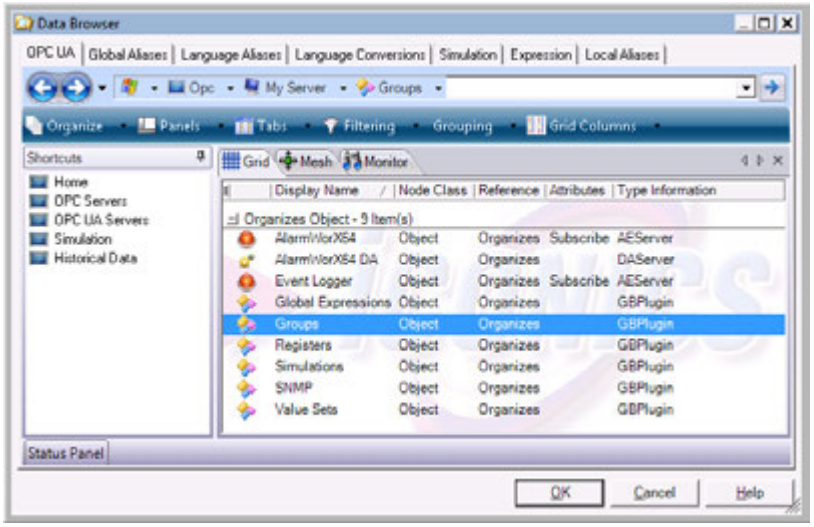


Figure 11.10 - Selecting Groups in Data Browser

11.4 Value Sets

A value set allows you to write pre-configured sets of values into a group of data tags. Each set of values has a reference value. Writing a reference value to a value set data tag will distribute pre-defined values to all data tags in the value set. A value set is different from a group in that each tag in a value set is updated with a preset value that can be different from tag to tag. Value set data tags are of *string* data type and are available as READ/WRITE data tags. You can also use parameters with value set data tags, just like you can use them with expressions and groups, which you have already seen.

Anything that requires multiple values to be written to set a meaningful collection of data is a candidate for a value set. One example is a GPS location, which requires two or three values: longitude, latitude, and altitude. Another use for a value set is for setting a color. Individual colors require three values to be written: one for red, one for green, and one for blue, in the RGB color space that computers use. Let us see how color value sets work.

Value sets are created and configured using a three-tabbed configuration form. On the **ValueSet Properties** tab, you can enter a description and add parameters.

Add members to the value set by using the **ValueSet Tags** tab. For any individual color, the color name or ID is written to one tag, then red is written to a second tag, green is written to a third tag, and blue is written to the fourth. You can see the configuration on the ValueSet Tags tab in the image below.

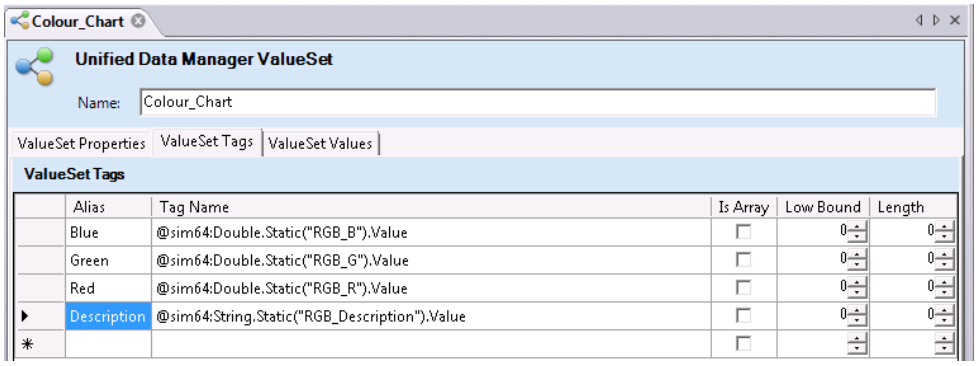


Figure 11.11 - Associate Tags with a Value Set

Once you associate tags with the value set, you need to add the colors that are part of the value set to the **ValueSet Values** tab as shown below.

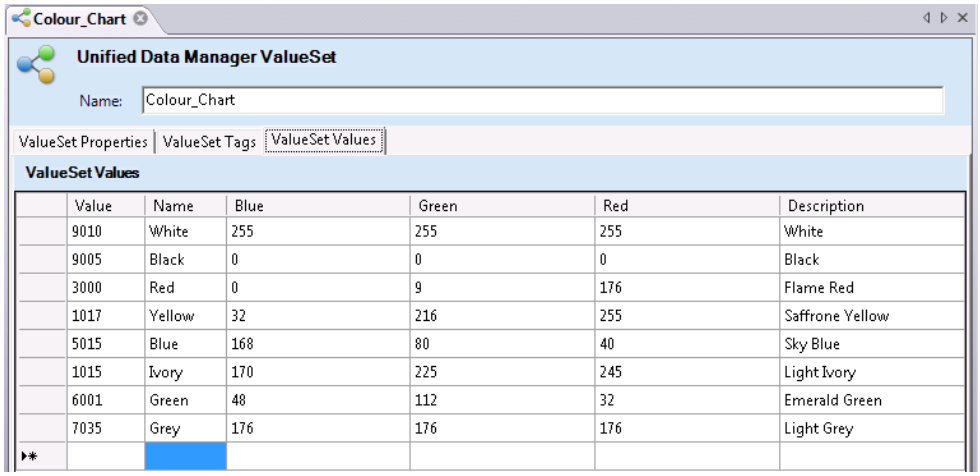


Figure 11.12 - Define Values for Tags

You can use, for instance, GraphWorX64 to write values to a value set tag. You just need to insert a Data Entry point into a display and browse for the value set tag in the Data Browser in the same way you browse for an expression or group, except that you select the **Value Sets** folder, which is highlighted in the next image. When you want to distribute the value set value for, for example, the color white, you would write to the tag "9010", which is the value specified in the Value column of the ValueSet Values tab.

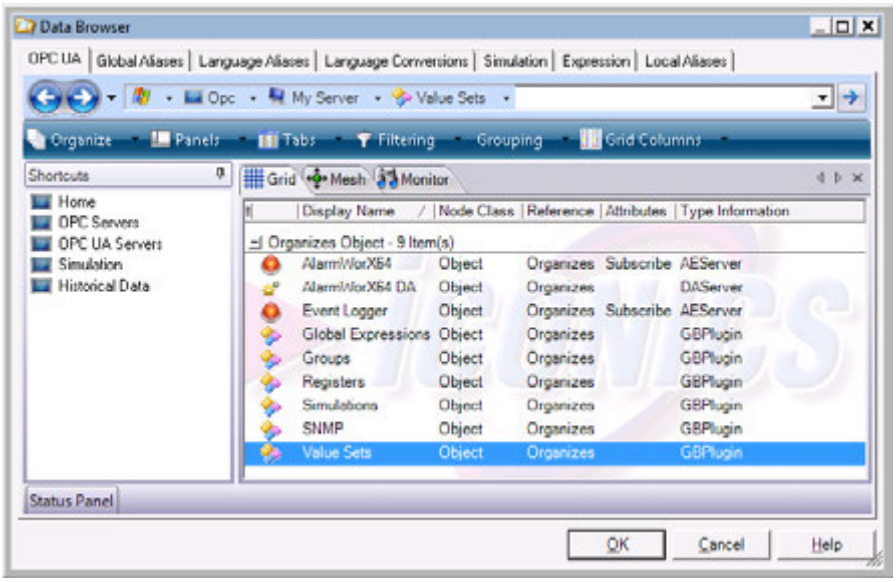


Figure 11.13 - Selecting the Value Sets Folder in the Data Browser

11.5 Registers

A register is a named address into which you can store values. In general, registers are named variables that all GENESIS64 clients can use. In the Unified Data Manager, a register consists of two configuration tabs.

In the **Register Properties** tab, shown below, you define the **Data Type** and a **Description**, and you decide whether the register is read only or writable. You can also enable or disable "Input Updates Propagation", which indicates whether values changed on input tag will be sent to an output tag.

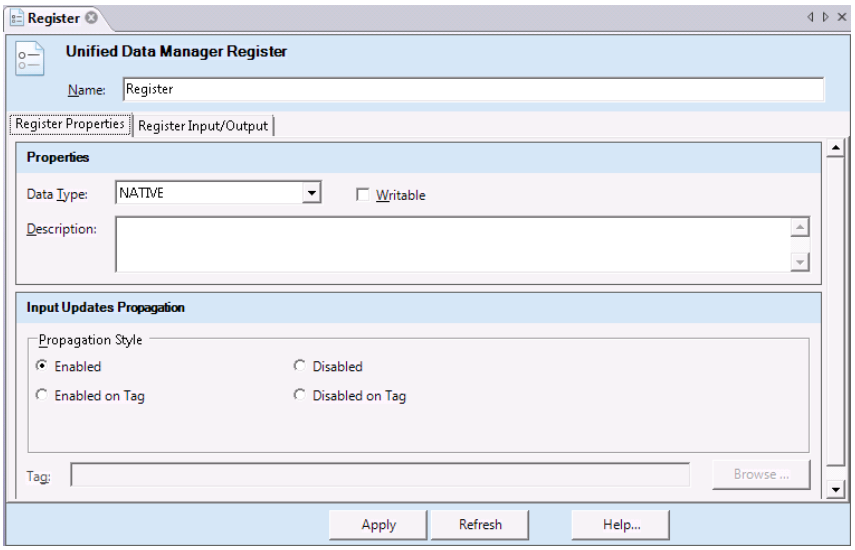


Figure 11.14 - Register Properties Configuration

How you configure the **Register Input/Output** tab depends on the type of the register you need. The options in the "Input Tag" section are designed for definition of an input signal for the register. You can choose to read an OPC Data Tag if you want to use this register to aggregate data to other OPC clients in GENESIS64. If you choose this option, you will need to specify the input tag path in the "Input OPC Data Tag" field. The default data type for this input tag is "Native" but you can choose an appropriate data type in the dropdown list. You can also specify the Scan Rate, or rate at which the register will poll the OPC tag.

Alternatively, if this is simulation variable, for example, you can choose “None” in the Input Tag section. You do not need to specify an input tag if you select this option, but you do have to provide the register with an Initial Value.

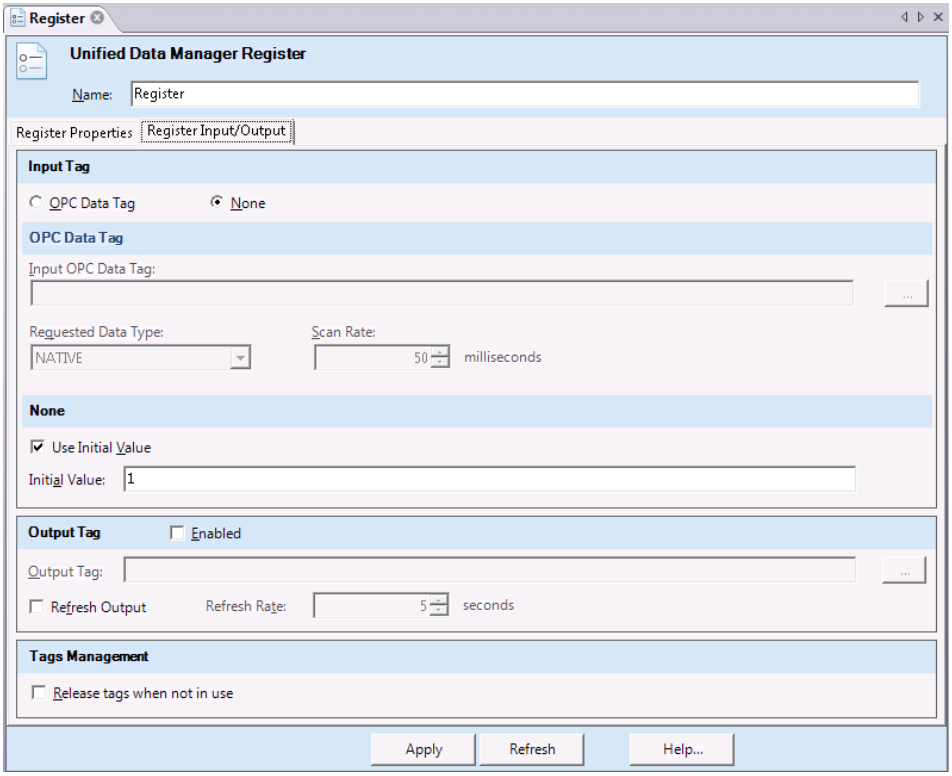


Figure 11.15 - Configuring Input/Output Properties of Register

If you need to bridge the data from the register to another OPC tag, check the “Enabled” checkbox in the **Output Tag** section. You can then specify the output tag path by typing it in or browsing for it. You can further indicate a refresh rate for the output tag by checking the “Refresh Output” option and giving a refresh rate.

The **Tags Management** section contains only the “Release tag when not in use” checkbox which, when checked, disconnects the input tag as well as the output tag when no client is connected to the register. This may save points counting toward your license. On the other hand, it will cost time when a client tries to request the point again. Having this option checked also disallow bridging.

As mentioned before, there are different types of registers. You can configure a register to achieve **data aggregation**, or provide a mechanism for data **bridging** where the value of one tag is written though the intermediate register to another tag. You can also use the register to **pre-fetch data** or to **hold** a value you enter into it. Let’s look at each of these four types of registers.

11.5.1 Bridging Registers

If you need a data point from one hardware device to be sent to a data point on another hardware device, use ***data bridging***. Data bridging maps the output of a device as the input to another device using the UDM Register as a bridge.



Figure 11.16 - Data Bridging

For bridging, you only need to specify an "Input OPC Point Tag" and an "Output Tag". To select a tag, click the field's ellipsis […] button to open the Data Browser.

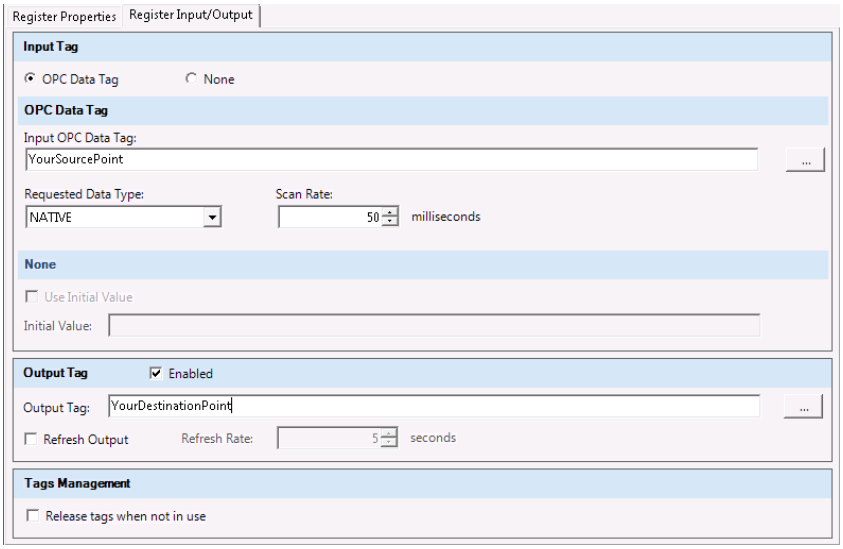


Figure 11.17 - Bridging configuration

Note: Do **not** check the "Release tags when not in use" checkbox, and do **not** change **Input Update Propagation** to "Disabled". If you do, bridging will not work.

11.5.2 Aggregation Registers

You create an aggregation register by specifying the same data source for the Input OPC Point Tag and the Output Tag. An aggregation register is usually used when you want all clients connected to only one server even when you have more OPC servers. By connecting them all to one server, you can aggregate data at that server.

You may find that you have hardware devices that cannot handle the volume of requests that are being sent to them. This happens if a device's tags are requested frequently or from several OPC clients, or if the device has a limited ability to cope with I/O requests. If the device cannot handle the requests, data becomes bottlenecked, and clients must wait for data to be returned.

One way you can overcome this problem is by using ***data aggregation***. Data aggregation collects multiple requests for a tag into one request, reducing network traffic.

In the example shown in Figure 11.18, there are three OPC clients that request the same temperature reading from a device. You can create a UDM register that handles those requests. The register acts as an intermediary that aggregates the three incoming client requests, sends one request to the device, gets one return from the device, which the register then forwards back to the three clients.



Figure 11.18 - Data Aggregation

Using the UDM register, you define one register for each tag whose requests you want to aggregate. Once you set up the register to aggregate tag requests, the register intercepts all client requests for the tag and pools them into one request that it sends to the device. In turn, the device returns one reply, which the register then broadcasts to the OPC clients that requested the tag.

In order to use this feature enter the OPC tag for the data point you are aggregating in both the "Input OPC Data Tag" field and the "Output Tag" field. Make sure you also **disable** input updates propagation on the **Register Properties** tab; otherwise, it may happen that your values could be running in circles.

> **Note**: You can also enable/disable the propagation based upon an OPC data point value.

11.5.3 Data Prefetch Registers

Another way to address problems related to I/O communication and volume is with ***data prefetch***. Data prefetch is a way of preloading data in anticipation of client requests.

Data prefetch loads data from a device onto the server even if no clients are requesting the data. Because the data is preloaded onto the server, the data is available when a client requests it; the client doesn't have to wait for the I/O from the device.

Data prefetch is useful with slower devices or communication networks where I/Os between clients and devices are typically delayed because, for example:

- Your older or slower devices do not have the capacity to fulfill their I/O requirements.
- Geographical distance separates clients from the server, or the server from the device.

For example, a client requests second-by-second polling data from a device that is located on the other side of the continent, and your limited hardware device cannot accommodate the requests in a timely way. In this case, data pre-fetch might provide the solution you are looking for.

Note: There is a downside to using data prefetch. The ongoing loading of data from the device to the server creates a continuous stream of traffic between the device and the server. For that reason, it is important that you use restraint when choosing the data points you use prefetch for; this is not a solution you want to overuse. You will have to determine whether the volume of traffic that prefetch creates works to your advantage. For example, even if a hardware device has 30 data points, there may be only one data point on that device where prefetch benefits you.

Also, keep in mind that any register that does data prefetch consumes a license point because the input/output tag is in constant use by the register.

The configuration of a Data Prefetch register is identical to that of an Aggregation register except make sure the "Release tags when not in use" checkbox remains *empty*.

11.5.4 Input Registers

If not combined with an output tag, registers usually serve as holding registers or read only registers connected to another OPC point. In order to get such a register, you must uncheck the "Enabled" checkbox in the Output section of the **Register Input/Output** tab.

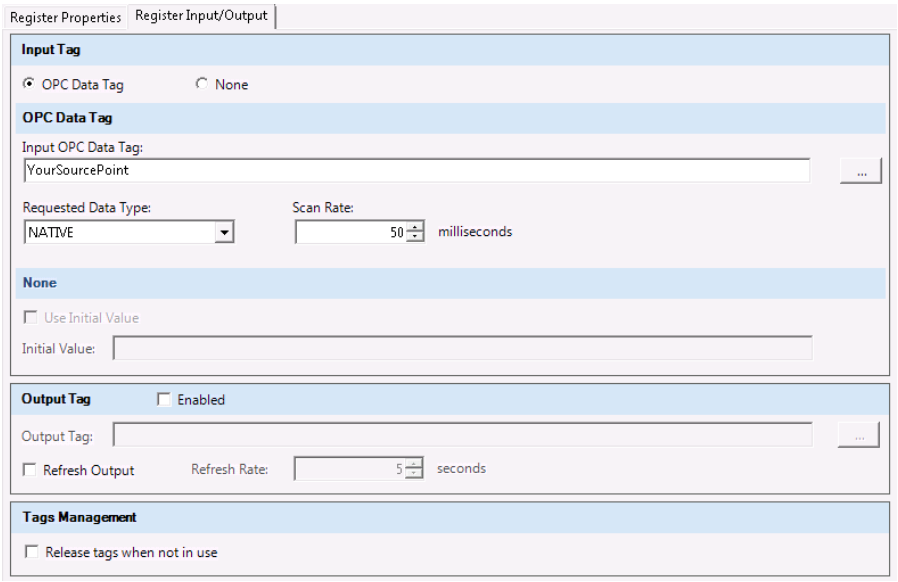


Figure 11.19 - Disabled Output Tag

Now that you have a register that you can use to read an OPC point defined as "Input OPC point" or you can create a holding register by selecting the "None" radio button. If you create a holding register, you can choose to define an "Initial value". For the holding register, you should also go to the **Register Properties** tab and make the register Writeable.

> **Note**: It is not recommended that you make a register writeable if it has an input OPC Data Point but no output tag. If you enter a value in such a register, the value is stored in the register but the input OPC Data Point value does not change. The entered value is propagated to nowhere because there is no tag defined as the register's output.

You can use the UDM registers in the same way as any other OPC points. You can select them using the Data Browser, as highlighted in Figure 11.20.

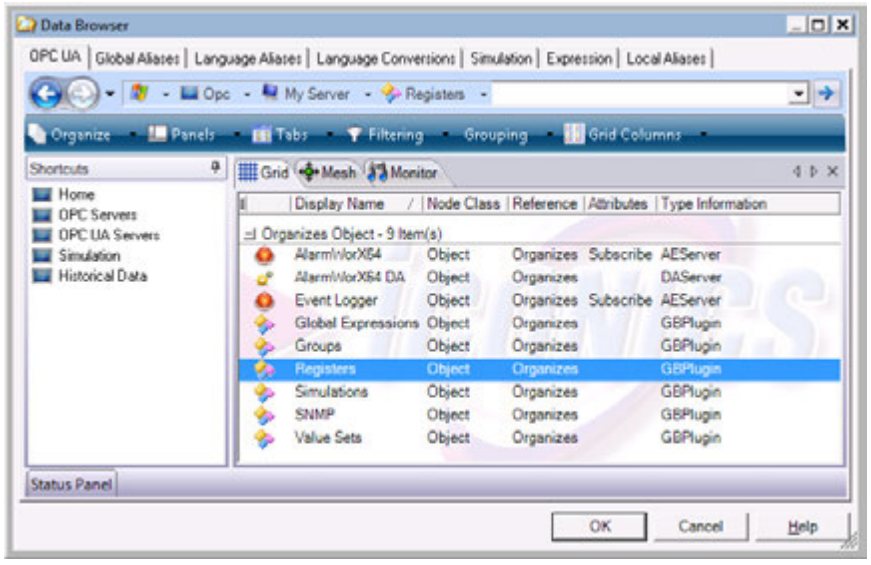


Figure 11.20 - Selecting Register in Data Browser

11.6 Triggers

A trigger is a rule that defines the conditions under which an event is to be executed. There are two types of triggers: data triggers and time triggers. *Data triggers* execute an event based on a specified condition that is defined in the trigger, while *time triggers* execute an event based on the time or sets of times that are part of the time trigger's definition.

Furthermore, each trigger can be scheduled to trigger by using the trigger's **Advanced** tab, shown below.

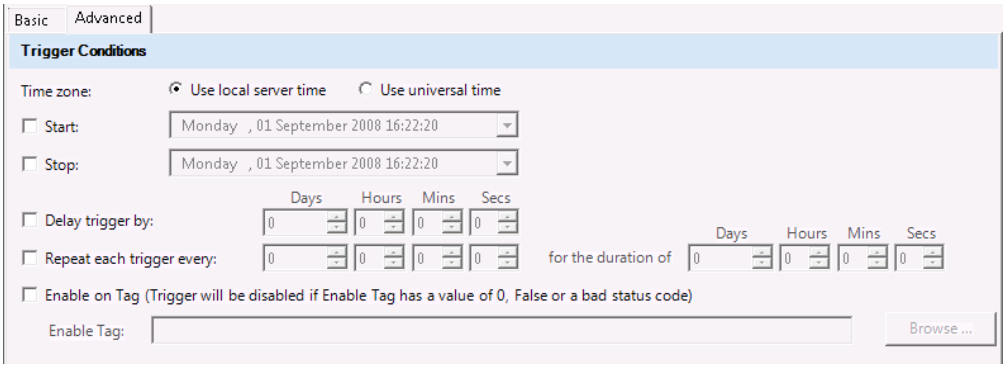


Figure 11.21 - Advanced Tab for a Trigger

To create a trigger, you expand the UDM's Triggers node, then either right-click "Data Triggers" and select the Data Trigger menu item (as shown below), or right-click "Time Triggers" and select the Time Trigger menu item.

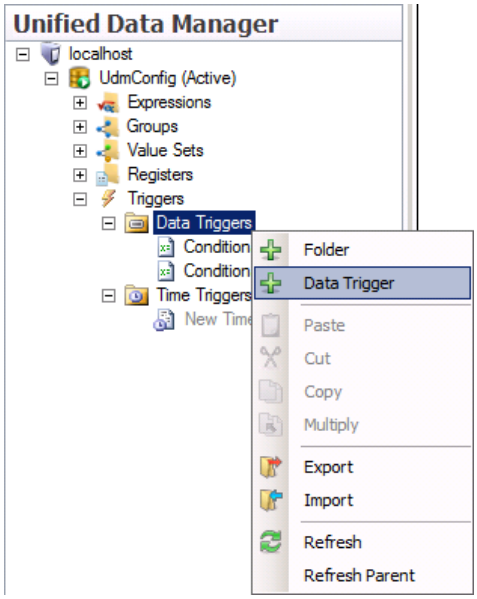


Figure 11.22 - Creating a New Trigger

Triggers are divided into condition triggers and event triggers. An *event trigger* is a one-shot trigger used for triggering an immediate action. However, if you need a continuous rather than single-shot action (for instance, for logging purposes), you would use a *condition trigger*. The type used is a function of what you want the trigger to do. Let us take a closer look at each type of trigger.

11.6.1 Data Triggers

A data trigger fires on any of the following conditions:

- When data changes (*event* trigger)
- When a data tag changes to TRUE (*event* trigger)
- When a data tag changes to FALSE (*event* trigger)
- When a data tag is TRUE (*condition* trigger)
- When a data tag is FALSE (*condition* trigger)

To select the data source that must meet the condition so that the trigger fires, click on the "Browse" button next to the "Data Tag" field.

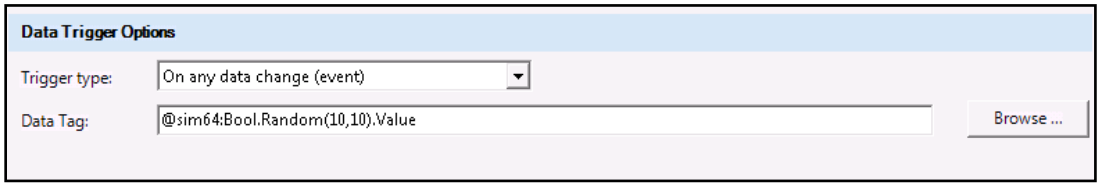


Figure 11.23 - Data Trigger Options with Selected Data Tag

11.6.2 Time Triggers

A time trigger executes an event or condition periodically, in accordance with a user-specified recurring timeframe. The recurring pattern can be a one-time event, can occur at different time intervals, or can be defined by specific dates and times. The Unified Data Manager's time trigger options change according to your selection and give you fine control over when the time trigger evaluates to **True**. Shown in Figure 11.24 is the trigger configuration dialog for a specific date and time trigger.

A time trigger event can be set to fire based on the following conditions:

- Once, at a set time
- After an interval of time has passed
- Recurring at specific dates and times

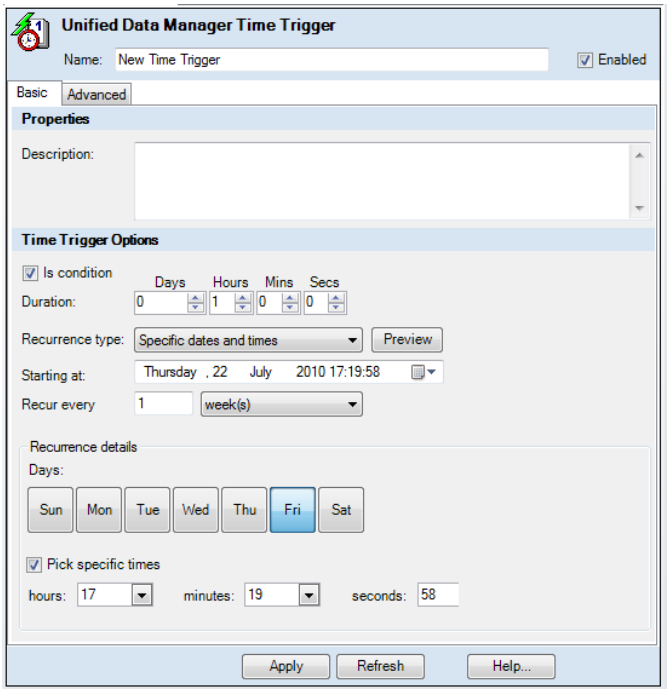


Figure 11.24 - Time Triggers Configuration

To set a time trigger *event*, you should leave the "Is Condition" checkbox unchecked; but if it is a time trigger *condition*, go ahead and put a checkmark in the "Is Condition" checkbox.

Select a "Recurrence type" of One time only, Time interval, or Specific date and time. Depending upon the time trigger recurrence type you select, the Options section changes to allow you to set appropriate parameters.

11.6.2.1 One Time Only Recurrence

To set a **onetime only** event, select the "One time only" Recurrence type, select the date and time that the trigger fires in the "Trigger at" box.

11.6.2.2 Time Interval Recurrence

To set a **time interval** at which an event fires, select the "Time interval" Recurrence type (as shown in Figure 11.25), then in the "Starting at" box enter the date and time that the time interval starts its counter, and then in the "Recur every" boxes specify the interval between time trigger events (the time span cannot be zero).

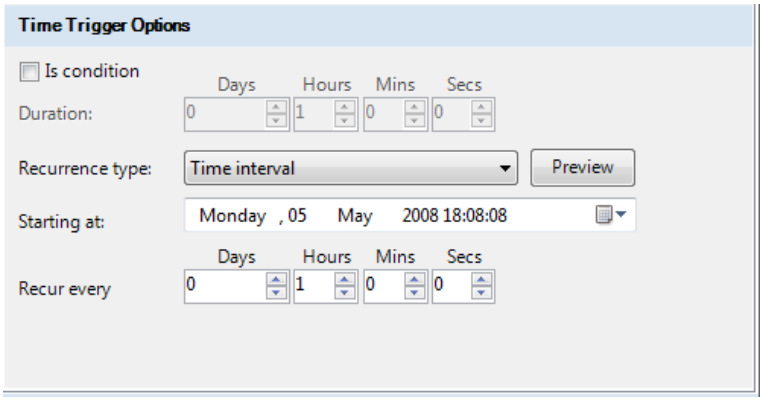


Figure 11.25 - Time Interval Selection

You can preview the times that the trigger will fire by clicking the "Preview" button. The figure below shows a sample of the type of preview you might see.

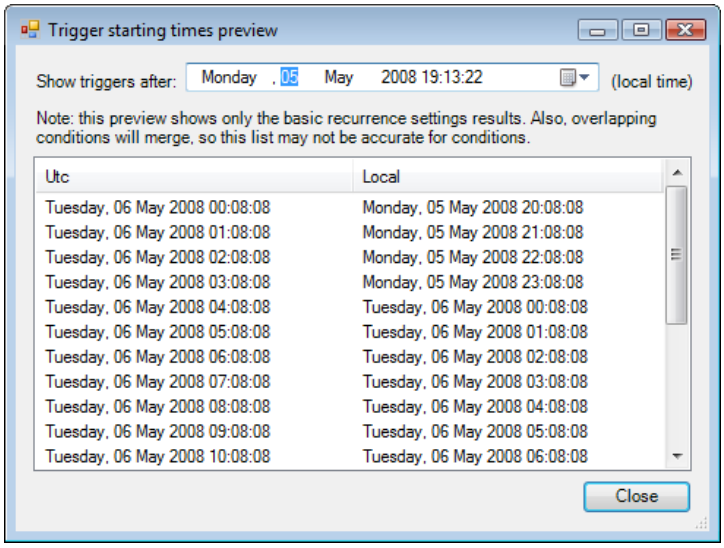


Figure 11.26 – Previewing a Trigger's Starting Times

11.6.2.3 Specific Date and Times Recurrence

To set **specific dates and times** when events fire, select the "Specific date and times" in the Recurrence type dropdown list. In the "Starting at" box, enter the time that the time interval starts its counter. In the "Recur every" boxes, specify the interval between time trigger events, and select the time period (seconds, minutes, hours, days, weeks, months, or years) from the dropdown list.

The Recurrence Details section changes depending on the selected interval. (Shown in below is the *Recur Every # Week(s)* selection.) Select the different days, months or times in the Recurrence details section.

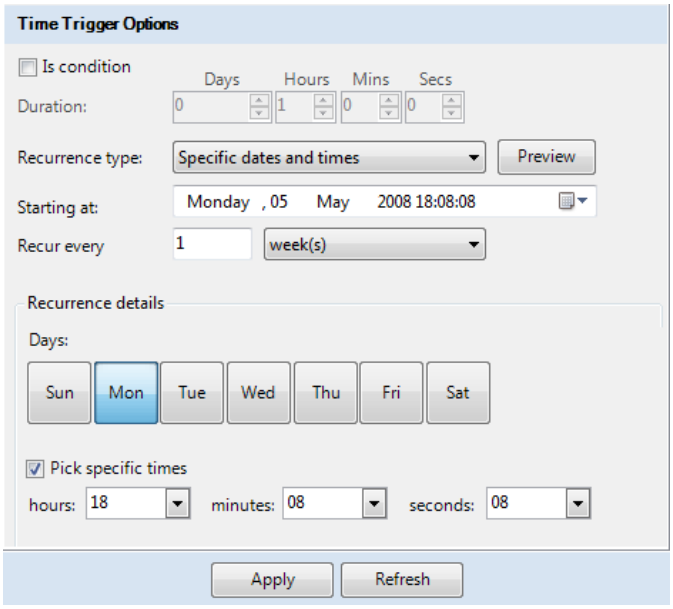


Figure 11.27 – Setting a Trigger's Specific Dates and Times

To select *specific times*, put a checkmark in the "Pick specific times" checkbox and set the time in the Hours, Minutes, and Seconds boxes. In the example shown above, the event will trigger at 4:08:08 PM every Monday. Remember you can see the schedule by clicking the "Preview" button.

11.6.2.4 Time Trigger Conditions

Some GENESIS64 applications can use a condition status to perform certain actions instead of relying on an event to be fired. For example, you can set a conditional time trigger in the Unified Data Manager so that the condition changes only once, changes in a repeating or recurring fashion, or changes at specific dates and times. All of the options we looked at in the section above apply to *conditional* time triggers, but now instead of an event, a Boolean value or flag is stored so that a program can READ the value and act *according to its own programming*. The condition time trigger appears in the Unified Data Browser so that it can be selected by an application. The time trigger condition is set so that it switches like this:

- From inactive to active once; the only recurring type.
- From active to inactive; and then inactive to active on a recurring basis.

The GENESIS64 online Help describes this type of trigger in more detail.

11.6.3 Using Triggers

Data triggers and time triggers are used in Hyper Historian. When you browse for triggers in Hyper Historian, the dialog box looks like below.

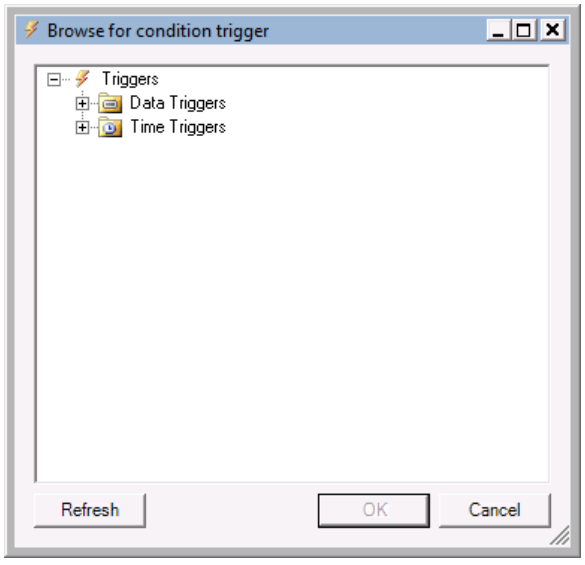


Figure 11.28 - Selecting a Trigger in Hyper Historian

Lab: Recipes (Value Sets) L11

11.1. Lab Overview

As you can imagine, icTonics makes a few different beverages, each with a different recipe. They have given us these recipes and would like us to create a scenario where the operator can change the production recipe with the click of a button. This will help them with transitions from making one beverage to another.

Estimated time to complete this lab: 20 minutes

11.1.1. Task Checklist

In this lab, you will complete the following tasks:

- Create a value set folder
- Create value sets that act as Recipes
- Edit the displays to allow for setting Recipes

11.1.2. Support Files

To complete this lab, you will need the following file(s):

- icTonicsTankMonitor.gdfx

11.2. Value Sets

Value Sets allow you to write different preset values to different OPC tags with a click of a button. In this section, we will add recipes as value sets and allow for change in recipes during runtime.

1. Open the Workbench if you closed it from the previous lab.
2. Select the Unified Data Manager (UDM) in the Project Explorer.
3. You should see a UDM configuration database called "**icTonicsUDMConfig** (Active)" in the Project Explorer. Expand this configuration and right-click on the "**Value Sets**" folder.
4. Add a folder and name it "**icTonicsRecipes**" then click on "**Apply**" to save it.
5. Right-click on the **icTonicsRecipes** folder in the Project Explorer and add a "**Value Set**". Name this new item "**Mixers**".
6. Click on the **ValueSet Tags** tab. Double click under **Alias** and type in **Water**. Double-click in the Tag Name field and click on the "Ellipses" (…) button to bring up the Data Browser and select the **Water tag**.
7. Repeat step 6 for the **Producing**, **TankLevel** and **Syrup** tags. Your finished configuration should look similar to Figure L11.1.

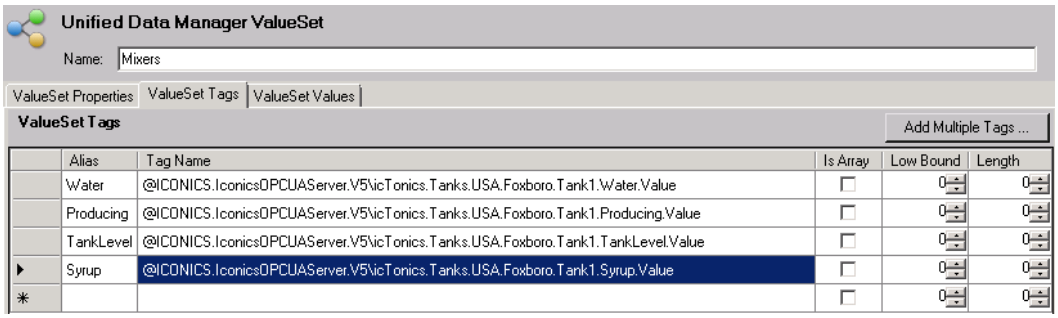
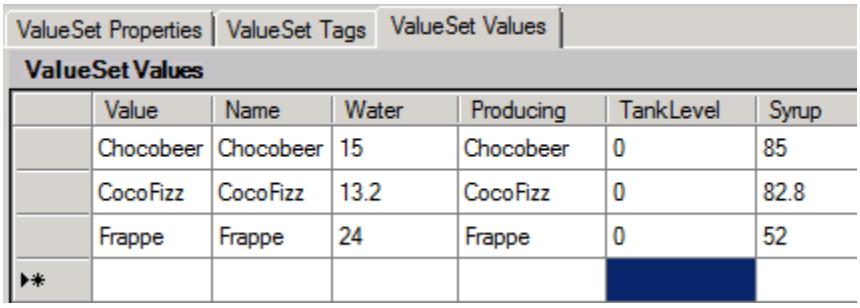


Figure L11.1 - ValueSet Tags

8. Click on the **ValueSet Values** tab. Under **Value**, type in "**Chocobeer**", and give it the same value in the **Name Field** (notice that this happens for you) and **Producing** field. Under **Water**, type in a value of **15** and, under **Syrup**, type a value of **85**.

9. Repeat the above steps to insert two more value sets so that your finished configuration looks like Figure L11.2.


ValueSet Properties | ValueSet Tags | ValueSet Values

ValueSet Values

	Value	Name	Water	Producing	TankLevel	Syrup
	Chocobeer	Chocobeer	15	Chocobeer	0	85
	CocoFizz	CocoFizz	13.2	CocoFizz	0	82.8
	Frappe	Frappe	24	Frappe	0	52
*						

Figure L11.2 - ValueSet Values

10. Click on "**Apply**" to save the value set.

11. Make sure the icTonicsUDMConfig configuration is active.

11.3. Changing Recipes in GraphWorX64

1. Open the **icTonicsTankMonitor.gdfx** and add a Process Point near the top of the display.

2. Select the **Tank1 Producing** tag in the Data Browser and click on "**OK**".

3. Add a **Text** object to the left of the process point and add the text "**Now Producing**".

4. Add a button to the right of the process point and select **Popup Menu** as its Command.

5. In the button's **MenuItems** property, click on the ellipsis button to bring up the **Edit Menu Items** dialog box.

6. Add a menu item for each of the recipes you created and enter the recipe's name as each item's **MenuText**. Keep the **Edit Menu Items** dialog box open.

7. Change the **Menu Items** command to "**Write Value**" for each menu item and set each one's **DataSource** to point to **@vset64:icTonicsRecipes.Mixers.Value**.

   Note: From within the **DataSource** field, you can click on the ellipse to launch the **Data Browser** and select **Home → OPC Classic Servers → My Server → Value Set** and the item you want to connect to.

8. Under the **CommandParameters**, enter the appropriate **OnUpValue** for each one.

   Note: Since the values we are writing to the tag are strings, be sure to enclose them in double quotes (e.g. **"Frappe"**) so that the values are properly written to the destination.

9. You should have three menu items, one for each recipe, when you are done. Click on "**OK**" to close the dialog box.

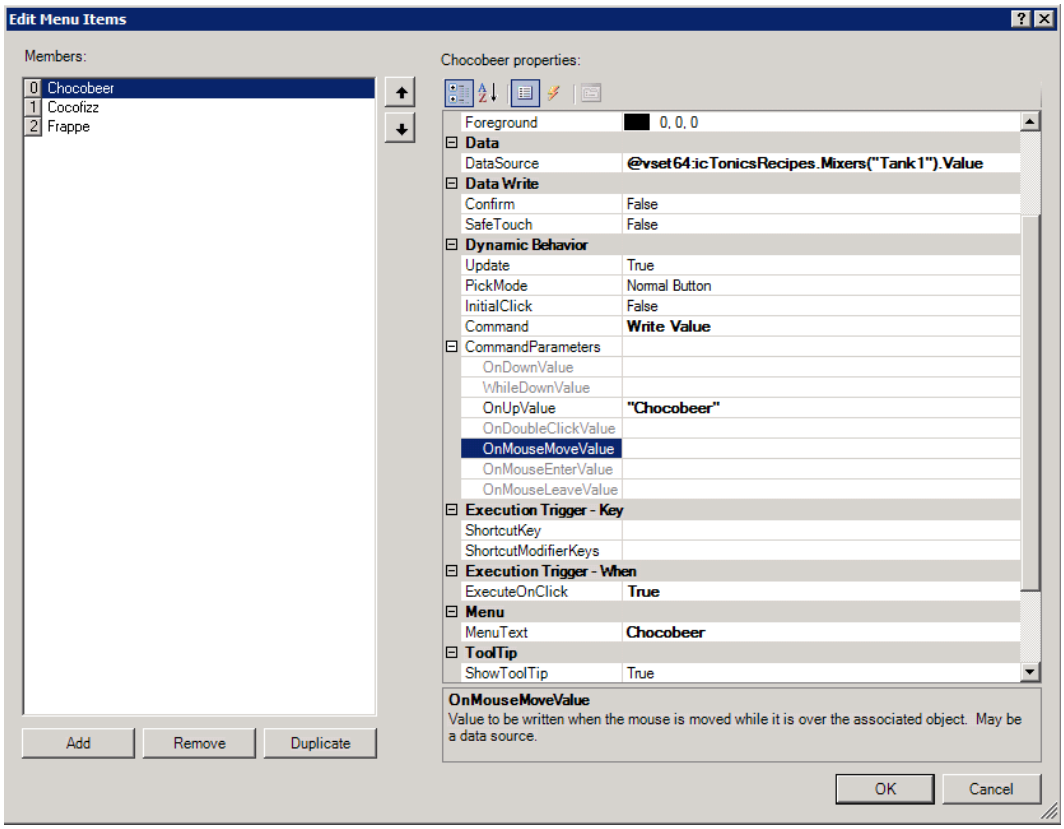


Figure L11.3 - Value Set: Write Value and Menu Text

10. Go to the button's properties and change the **Text** to "**Select Recipe**". Make your finished display look like the one shown in Figure L11.4, and save it.

11. Add two more **Process Points**: one to show the **Syrup** and the other to show the **Water**.

12. Now you can go into runtime and try changing the recipe. As you change the recipe, the tag should change to show the beverage type that is currently in production.

    Note: The Alarm Viewer in your display may be different from the following figure, depending on what filter you're currently using. Try double clicking on the viewer and clicking button in the popup tool bar to manipulate your filter.

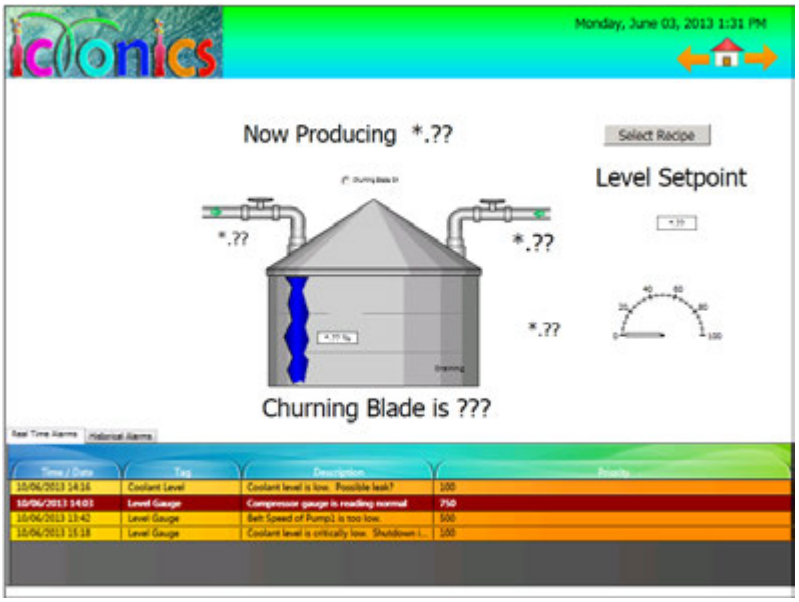


Figure L11.4 - icTonicsTankMonitor Display

11.4. Using Parameters

If you have some time, you can add a parameter to the Value Set. This way, when we try to change tanks later, we will be able to use the parameter to set a dynamic tag value.

1. In the value set your created previously, from the **ValueSet Properties** tab, add a parameter named **Tank**, and change its Type to **String**.

Parameters

	Name	Type	Padding Align	Padding Char	Number Of Places
▸	Tank	STRING	LEFT	_	0
*					

Figure L11.5 - Parameters

2. On the **ValueSet Tags** tab, change the part of the tag where it says "**Tank1**" to **"<<Tank>>"** to use the parameter we set in the Properties tab.

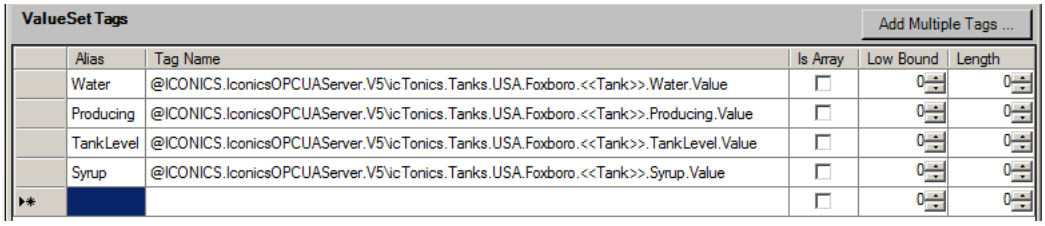


Figure L11.6 - ValueSet Tags with Parameter

3. We will need to change the tags in the GraphWorX64 display to give a parameter value.

4. In the **icTonicsTankMonitor.gdfx** display, select the "**Select Recipe**" button, and click on the **Menu Items** collection to bring up the **Edit Menu** Items dialog box.

5. Change the data source in each of the menu items to include a parameter value. You can either reselect the tag, or simply add "**("Tank1")**" (without outside quotes) after "**Mixers**" in the **DataSource** field. The syntax should look like this: **@vset64:icTonicsRecipes.Mixers("Tank1").Value**

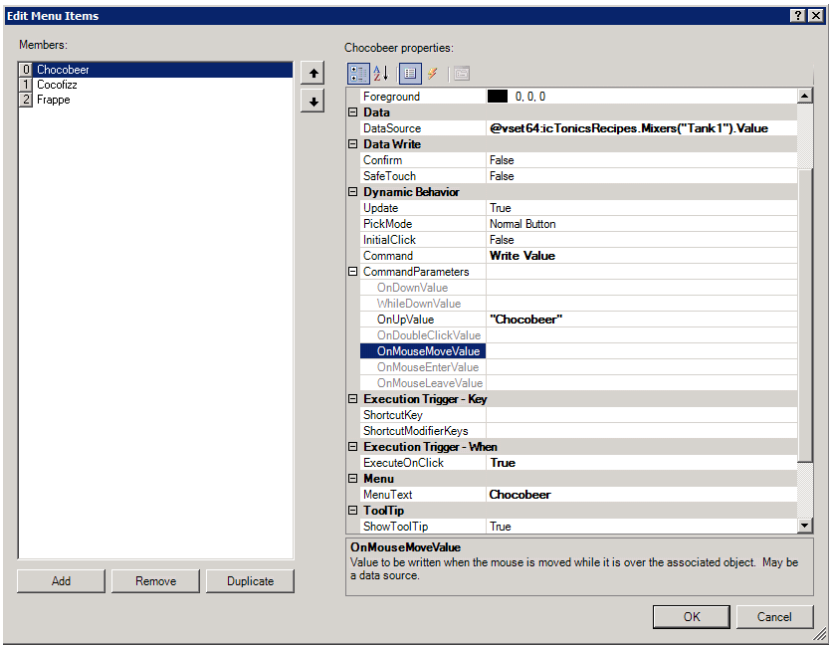


Figure L11.7 - Change Tag Name

6. Try running the display again, it should work just the same as if you did not have parameters.

Review: Unified Data Manager *R11*

11.1 Objectives Review

You should now know and understand the following topics:

- How to create items in the Unified Data Manager that you can use throughout GENESIS64
- Why and how to use expressions, value sets, groups, registers, and triggers

11.2 Summary

The Unified Data Manager (UDM) provides centralized storage of data such as expressions, groups, value sets, registers, and triggers, that you can benefit from when developing a project. UDM configuration is stored in a SQL database and its management is done in the Workbench.

You create items in the UDM by right-clicking nodes in the Project Explorer and selecting options from a popup menu. Each node has its own hierarchical tree into which you can add folders and items. The UDM has a node each for expressions, value sets, groups, registers, and triggers. An expression is a function that takes input, performs a specified calculation, and returns an output. A group is a collection of data tags that are manipulated as a single item. A value set allows you to write pre-configured sets of values into a group of data tags. A register is a named address into which you can store values. A trigger is a rule that defines the conditions under which an event is to be executed.

Items are modified using forms that appear in the UDM when the object is selected. When you save your changes to the UDM configuration database, those settings are immediately available to anyone connected to the database.

11.3 Terms

Data Aggregation
Data Bridging
Data Prefetch
UDM Expression
UDM Group
UDM Register
UDM Trigger
UDM Value Set
Unified Data Manager

11.4 Test Your Knowledge

In this chapter, we discussed how the Unified Data Manager is a centralized location where you configure expressions, registers, value sets, triggers, and groups. Before you continue to the next chapter, make sure you can answer the following questions.

1. The following data type is NOT configured in the UDM?

   a. Sets of values
   b. Groups of users
   c. Expressions and calculations
   d. Triggers that are based on conditions

2. A register can be used for which of the following purpose(s)?

   a. To store data
   b. To bridge data from one tag to another
   c. To aggregate data
   d. All of the above
   e. None of the above

3. True or False: You can use a UDM group in another UDM group.

4. True or False: You can make use of parameters in value sets, groups, and expressions.

5. Which of the following statements about triggers is true?

   a. A data trigger evaluates an expression periodically and fires when the condition is true
   b. A time trigger can fire at a specific date and time
   c. A time trigger can fire at a recurring set of dates and times
   d. A data trigger makes use of time trigger features
   e. All of the above
   f. None of the above

6. Which data type is used for data bridging?

   a. Expressions
   b. Value sets
   c. Groups
   d. Registers
   e. Triggers

Global Aliases 12

The Global Aliasing System (GAS) is a powerful referencing feature that can be used in any GENESIS64 applications to quickly switch between multiple data sources using a single reference by changing the value of the reference.

When you design a display, any process point can contain one or more global aliases. The Global Aliasing Server is able to resolve global alias references and provide the correct value for the tag at runtime. That tag is then used to display the current data being observed or controlled. In practice, you might define a button that opens a popup menu where you can select one out of ten different data sources. Each selection will change the value of the global alias reference in the displays. This in turn changes the data source for a process point in the GraphWorX64 display, so that in runtime, the data changes on your display. With this system in place, a single process point and a button can do the work of ten process points on a single display or alternatively the work of ten displays with a single process point. This is the power of global aliasing.

The Global Aliasing System follows a certain structure of values that groups common objects into categories. Consider Table 12.1, which demonstrates the global aliasing hierarchy in the left column on an example of building control in the right column.

Table 12.1 - Global Aliasing Mapping

Global Aliasing	Building Example
Theme	Buildings
Theme value	Building A, Building B
Alias	Lights, Heating
Alias Value	OPC points specific for Building A and B controlling Lights, Heating

After completing this chapter, you should be familiar with:

- General uses for global aliases
- How to create and use themes
- How to create and use global aliases

12.1 Architecture

Global aliases are stored and maintained in a SQL Server database and those aliases can be accessed over the network through the GAS Server. When you install GENESIS64, the installer creates a GAS server database with a default configuration. However, you may want to or need to create and configure your own GAS databases before you create global aliases configuration.

Figure 12.1 below shows the GAS architecture. Note the role of the FrameWorX server. If global aliases do not resolve or connections to the data sources are missing, make sure the FrameWorX server is functioning correctly and the service is running.

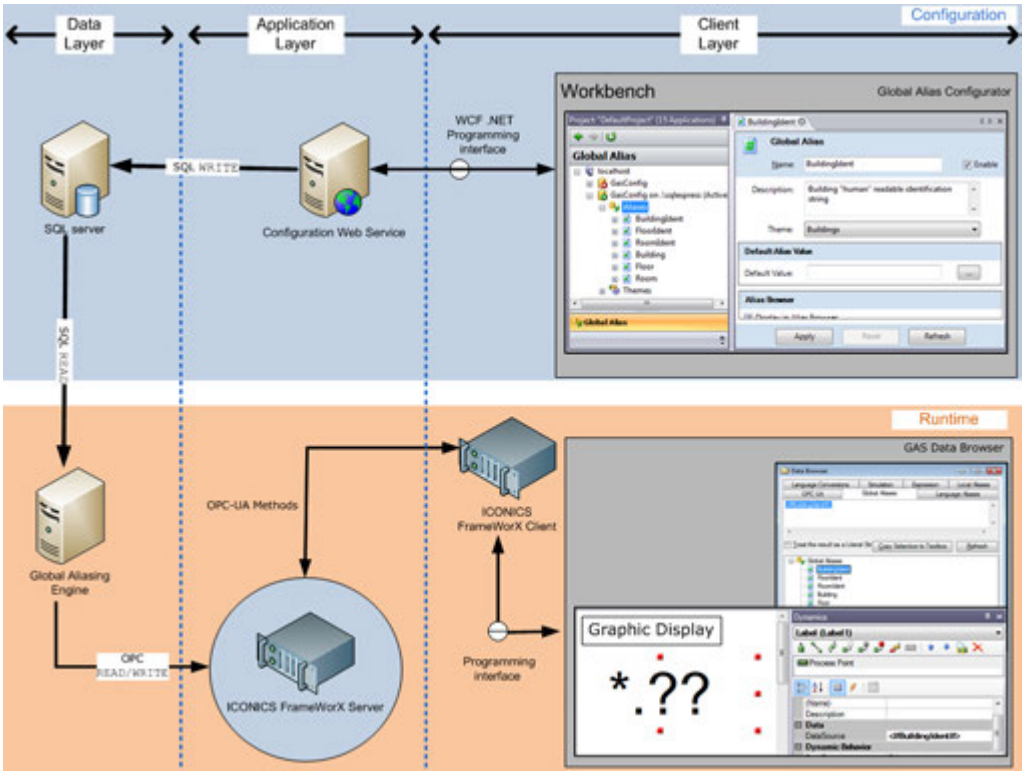


Figure 12.1 - Global Aliasing System (GAS) Architecture: Configuration (Top) and Runtime (Bottom)

You create global aliases and themes inside a configuration database. The GAS can have one *active* configuration database at a time. All aliases and themes used in your displays must be defined to that configuration. Any time you make a different configuration active, you must restart the GAS service so that it can recognize the change.

12.2 Definition of Global Alias

A global alias is a data string encapsulated between the delimiters <# and #> that enables you to create an object (such as a trend pen in TrendWorX64 or a process point in GraphWorX64) that can easily switch among multiple data sources using one unique name. Instead of creating multiple versions of an object (one for each tag), you can create one version and let global aliasing find the value for the object to be displayed. By using global aliases, you can reduce the overall number of objects in your display files as well as the number of tags that you have to manage.

The Global Aliasing system provides global storage for aliases and resolves aliases at runtime. The aliases can be used in OPC tags, menu values, file names, and almost any other place where you can insert text values into a display, map, trend, or alarm. Figure 12.2 shows how a global alias is used to switch between multiple OPC tags in a GraphWorX64 display. The values that a global alias resolves to are defined in the GAS as *alias values*. All global aliases and alias values are created and configured in the Global Aliasing interface in the Workbench.

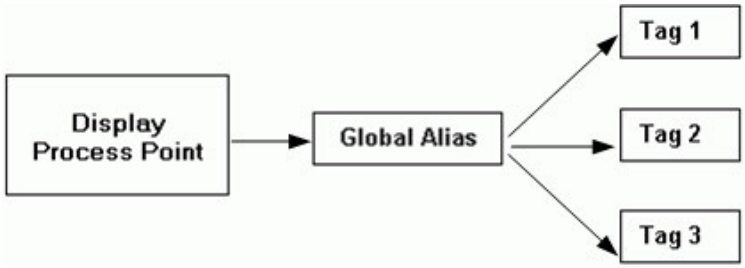


Figure 12.2 - A Single Global Alias for Multiple Data Sources

A collection of aliases with similar properties is called a theme. Aliases can be nested, eventually resolving to the value or property you wish.

12.2.1 A Time Saver

Consider lights in a building that you monitor, with each individual light reporting its current condition. Each light is assigned an alias, and that alias is globally available to GENESIS64 applications. In the scheme shown in the building example, buildings are divided into individual buildings, floors, rooms, and then the data sources you want to monitor. The address to a particular data source might be the following:

```
@Buildings.Administration.Floor_4.Room_403.Light_08
```

In this scheme, buildings are divided into individual buildings (`Administration`), floors (`Floor_4`), rooms (`Room_403`), and then the data sources you want to monitor (`Light_08`). Using this scheme, for 3 buildings, each with 6 floors, 34 rooms, and 20 lights you would have to identify 12,240 individual tags in a display to fully define the namespace.

With global aliases, this can be greatly simplified with the following: define a Building_ID alias for the building, substitution of which into your path would yield:

```
@Buildings.<#Building_ID#>.Floor_4.Room_403.Light_08
```

This single substitution reduces the number of unique tags by a factor of three, to 4,080 tags that you would have to identify in the display. Defining additional aliases could reduce the path to the following:

```
@Buildings.<#Building_ID#>.<#Floor_ID#>.<#Room_ID#>.<#Light_ID#>
```

This leaves you with *one* unique path to 12,240 individual tags that you can use anywhere! To accommodate the actual alias values, you would define values for: the three different buildings, floors 1 through 6, rooms 01 through 34, and lights 01 through 20.

That sums to 63 global alias values you need to define along with four global aliases. In your display, the process point would use only *four* global aliases. As you can reuse global aliases, this becomes a valuable, time-saving solution!

12.2.2 A Design Aid

Global aliases can also aid in display design. For the sake of this example, all three displays are of the same design. We will look at what you do as the user, and what happens behind the scenes.

First, at runtime, when you click on the Administration building (which assigns the value Administration to the Building_ID global alias), the display changes to show the label "Administration" in the title bar identifying the building you are looking at.

Then, on the same display, you select the fourth floor (which assigns the value Floor_4 to the Floor_ID alias and is used for selecting a floor diagram). The display then shows a diagram of the fourth floor with the rooms it contains. You only need to have one floor diagram for all of your buildings.

Finally, you click on room number 403 (which assigns the value Room_403 to the Room_ID alias and is used for selecting a room diagram) and another diagram appears, showing the lights in that room. Again, you only need to have one room diagram (as long as all the rooms are similar). The room diagram shows a status indicator showing each light's condition.

This solves your design problem. The complete path to each data source is built using the aliases you supplied. This automated construction of a path to the data source is a process that is referred to as *alias resolution*.

> **Note:** For examples of how Global Aliasing is used in GraphWorX64 displays, see the example files in GenDemo: FLOOR_CONTROL_GAS.GDFX and FLOOR_OVERVIEW_GAS.GDFX.

12.3 Alias Resolution

Alias resolution is the process of assigning a specific value to an alias.

Each global alias can be associated with a theme. A theme links a global alias to one or more data values (e.g. tags). The Global Aliasing engine allows you to assign different values to aliases based on the currently selected theme value. Theme values can be selected dynamically; either programmatically through an automation API or by a Pick action.

Figure 12.3 shows how a generic single alias can be used, for example, to link multiple tags to a single process point in GENESIS64. The Global Aliasing Server resolves the alias based on the selected theme value, which associates the alias to the data values (tags).

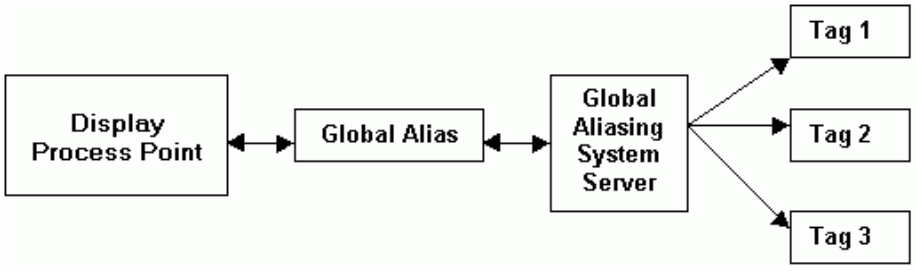


Figure 12.3 - Alias Resolution

Each theme can have one or more theme values associated with it, and each alias value is then associated with a theme value. A theme value allows you to switch between the data values that are linked to the theme, as shown in Figure 12.4. For example, if you have one theme with three theme values, and each theme value is associated with a tag, you can toggle between the theme values and therefore the tags.

Thus, you have a single process point associated with three different values, as shown in Figure 12.4. All themes and theme values are created and configured in the Global Aliasing configuration interface in the Workbench.

In addition, a single alias can be read by many different applications (for example, GraphWorX64, TrendWorX64, and AlarmWorX64), as shown in Figure 12.4. You can also associate multiple aliases with a single theme.

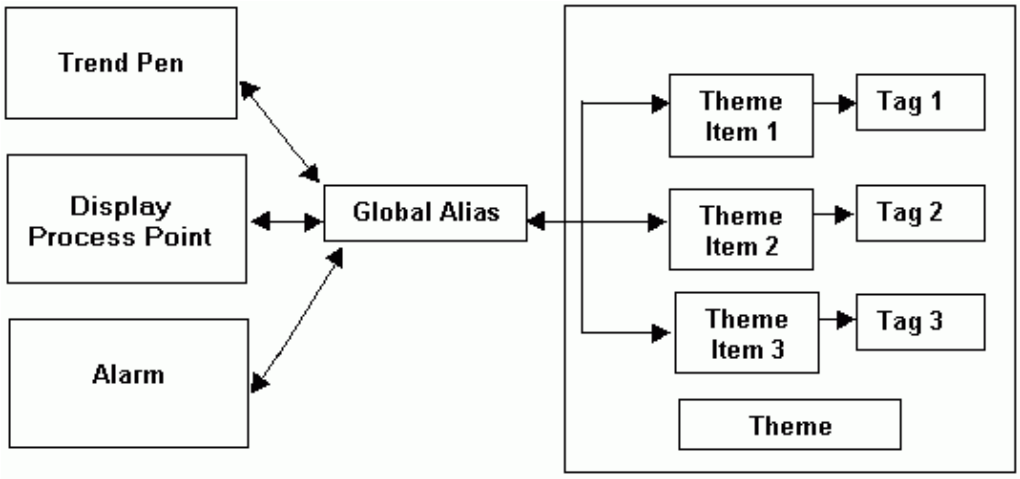


Figure 12.4 - Using Theme Values to Switch between Data Values

12.4 Using Themes and Theme Values

A theme can represent any general category, such as equipment, signals, files, regions, quality controls, departments, and much, much more. The sole purpose of a theme is to act as a category for the global aliases that use it. You assign alias values to each theme value when you associate a global alias with a theme.

The power of a theme is in the number of aliases that you associate with it. For example, the Buildings theme might also be associated with a second global alias called Floor_ID, a third called Room_ID, and a fourth called Light_ID. Plan the themes that you create so that by applying a theme at runtime you essentially enable an effective, efficient system for obtaining global aliases and their values. Note that one global alias can only be associated with one theme only.

The resolution of aliases is controlled by the theme scope (that is, on what level a theme is applied to your application). The Global Aliasing system supports three different levels of aliases:

- **Machine-level** applies to the entire computer.
- **Process-level** applies to the current process (such as, GraphWorX64, TrendWorX64, AlarmWorX64, or EarthWorX64.)
- **Document-level** applies only to the current document/display and its embedded controls (for example, a GraphWorX64 pop-up window).

  Note: Document level aliasing is available only in GraphWorX64. It does not apply to the TrendWorX64 Viewer and AlarmWorX64 Viewer.

In GraphWorX64 you can have multiple levels of documents or pop-up windows, as shown in Figure 12.5.

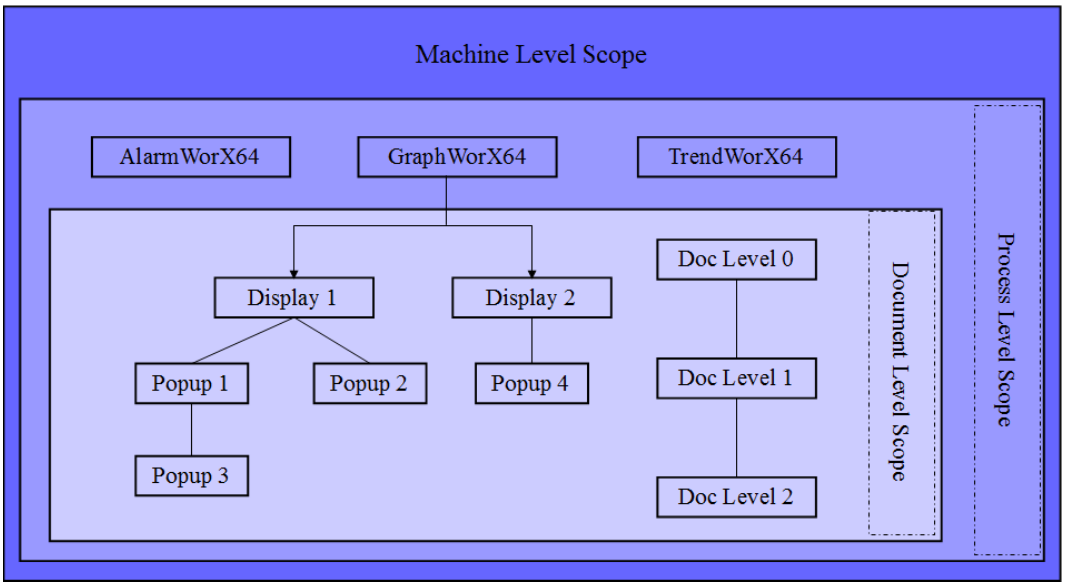


Figure 12.5 - Theme Scope

12.5 Configuring Themes and Global Aliases

To define themes and global aliases, open the Global Aliasing configuration interface in the Workbench by clicking the "Global Aliasing" icon in the Project Explorer, or go to **Start → All Programs → ICONICS → Tools → Global Aliasing**. After installation of GENESIS64, you will find a sample Global Aliasing configuration database, which demonstrates an example of themes and global aliases.

12.5.1 Configuration Database for Global Aliasing

When configuring global aliasing, first select the database in which you want themes and global aliases to be stored. Make sure you expand the database node in the Project Explorer to view existing global aliases and themes already created in the database.

If you modify your configuration database, you should reload it so that all connected clients are notified of the change and can use the latest values. In order to reload the database, right-click on the configuration database and select "Reload Configuration".

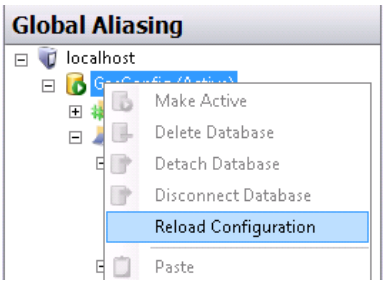


Figure 12.6 - Reload Configuration when Modified

12.5.2 Configuring Themes and Theme Values

In the Global Aliasing Configurator, expand the configuration database in the tree view so you can see the tree nodes called **Aliases** and **Themes**. Right-click **Themes** then select **Theme** from the popup menu. A form for defining a new theme opens, as shown in Figure 12.7.

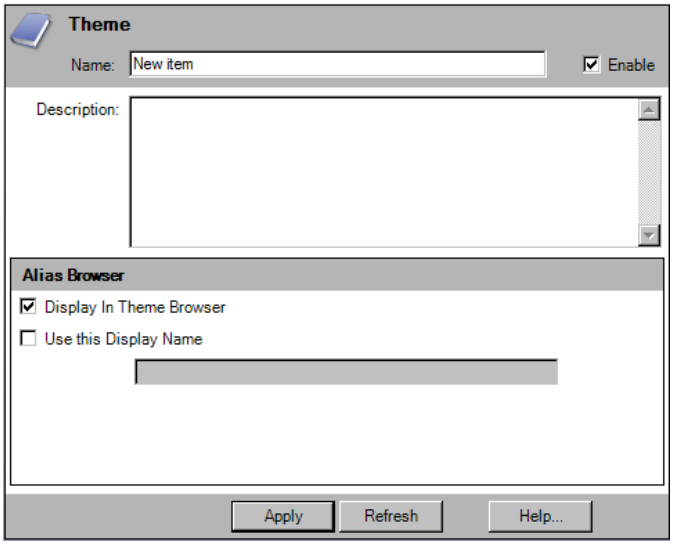


Figure 12.7 - Adding a New Theme

Once you have created the theme, you can define theme values for it. From the tree in the navigation pane, right-click the theme, then select **Theme Value** from the popup menu.

In Figure 12.8, a theme value of *Building C* is being added to the theme called *Buildings*. Note the **Default Theme Value** checkbox. A checkmark here indicates that the alias value that gets assigned to that particular theme value will be the default value for its global alias. For most themes, you will want to identify one theme value as the default so that, during runtime, when no value is selected, the global alias resolves to a value in all instances. Tips for easily identifying the default include:

- Always choosing the *first* theme value as the default, or
- Including the word DEFAULT in the theme value's name, or
- Using some other way to make it easy to identify the default value.

In Figure 12.8, the theme value is enabled and will be listed under the Theme tab that appears in the Data Browser.

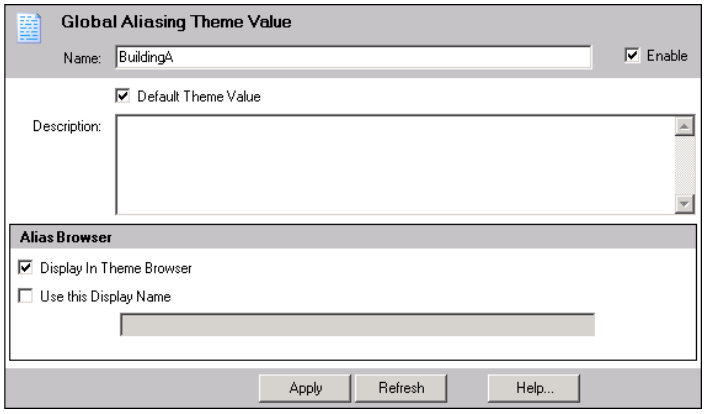


Figure 12.8 - Theme Values for the Theme Called Buildings

12.5.3 Configuring Global Aliases and Alias Values

Once you have defined a theme, you can define the global aliases that will associated with that theme, along with the alias values that get used with that theme. A theme, however, is not required for defining a global alias.

You can use folders to organize your global aliases. To create a folder, right-click **Aliases** in the navigation tree then select **Folder** from the popup menu, and define the folder. This adds the folder to the **Alias** node of the tree. Folders can also have subfolders.

Once you have a folder structure, you can add aliases to these folders. To create a global alias, right-click **Aliases** (or the alias folder you want to add it to) in the navigation tree then select **Alias**. The form for adding aliases, shown in Figure 12.9, appears.

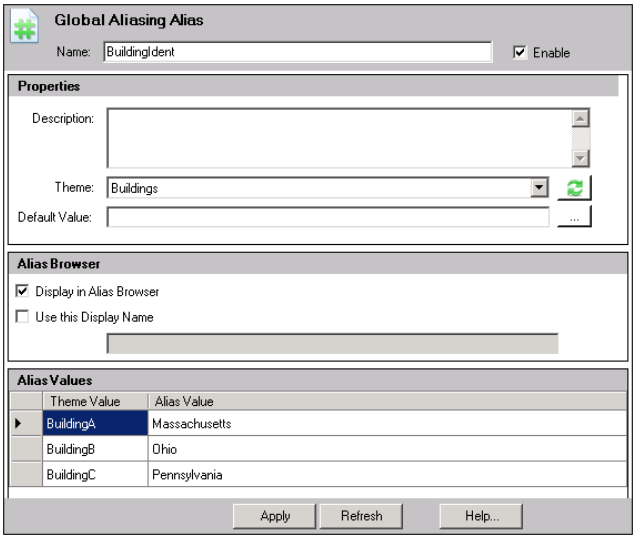


Figure 12.9 - Adding a Global Alias Definition

If you assign a theme, the theme's values appear in the **Alias Values** pane at the bottom of the form. In the **Alias Value** column, you provide the values that will be assigned to the global alias when the theme is in use. There are a few things you should keep in mind when entering alias values:

- You can type an alias value in the field, or choose a value such as a language alias from the Data Browser.
- One of the theme values may be the default value.
- You can nest a global alias in an alias value, as demonstrated in Figure 12.10. When adding a global alias to the alias value, use the following syntax: *<#GlobalAlias#>*.

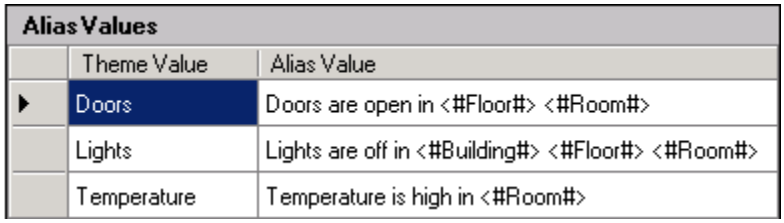


Figure 12.10 - Global Aliases Used in Alias Values

Once you have defined your themes, theme values, and global aliases, you can use them, for example, in GraphWorX64, AlarmWorX64, TrendWorX64, and EarthWorX64.

12.6 Using Global Aliases

Before you can use a global alias in a display, make sure the configuration database where it is defined is active. From the navigation tree in the Global Alias Configurator, right-click the configuration database then select Make Active from the popup menu.

12.6.1 Using a Global Alias in a GraphWorX64 Display

One way to show or use a Global Alias in a display is to insert a process point for a tag that uses a global alias in its path; when you browse for and select the tag, edit its path to replace the part that should be resolved dynamically with the global alias. For example, this data source:

```
@Buildings.Administration.Floor_4.Room_403.Light_08
```

Could become:

```
@Buildings.<#Building#>.Floor_4.Room_403.Light_08
```

When you go into runtime, you will see the default value of the Global Alias. However, GAS enables you to dynamically resolve an alias during runtime and to simultaneously apply one alias on different scopes.

You have two main options on how to set Global Aliasing themes or Aliases in GraphWorX64. First, you can automatically set them as your display enters runtime. Second, you can make use of Pick actions to give the operator control over current resolution of your Aliases.

12.6.2 Runtime Parameters of GraphWorX64 Display

You can set Global Alias Theme or Alias to be set automatically when your display enters runtime. Click in the blank area of the display to show the **This Display** property, shown in Figure 12.11.

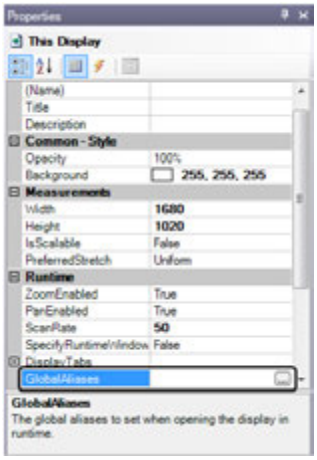


Figure 12.11 - Global Aliases Property for This Display

Put the cursor in the **Global Aliases** property, and then click the ellipsis button. The **Set Global Themes and Aliases** dialog box appears, as shown in Figure 12.12.

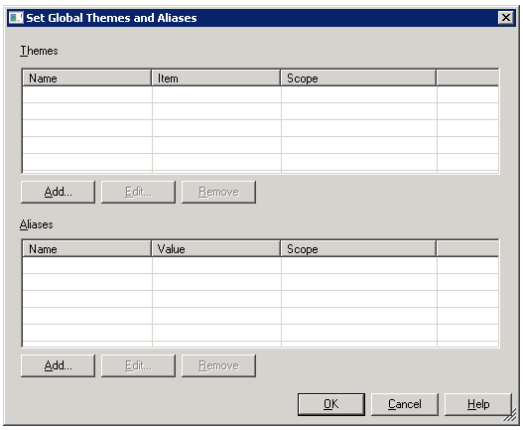


Figure 12.12 - Set Global Theme and Aliases Dialog

Use the Set Global Themes and Aliases dialog box to establish the initial setting of Global Themes and Aliases for the display when going into runtime.

To set the scope for a theme or theme value, click the "Add" button in the top **Themes** section of the dialog box. This displays the **Edit Global Themes** dialog box as shown in Figure 12.13 where you select a theme and its scope for the display.

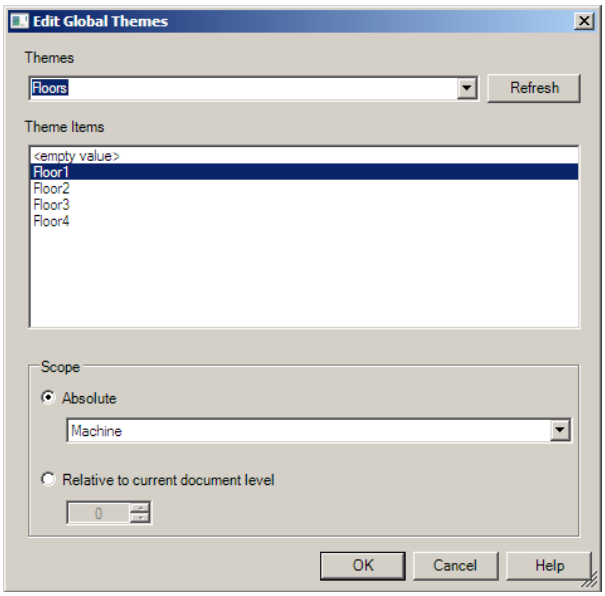


Figure 12.13 - Selecting the Display's Themes

You can apply the Alias theme to the machine, the process, a specified document level, or relative to the document level that is current during runtime.

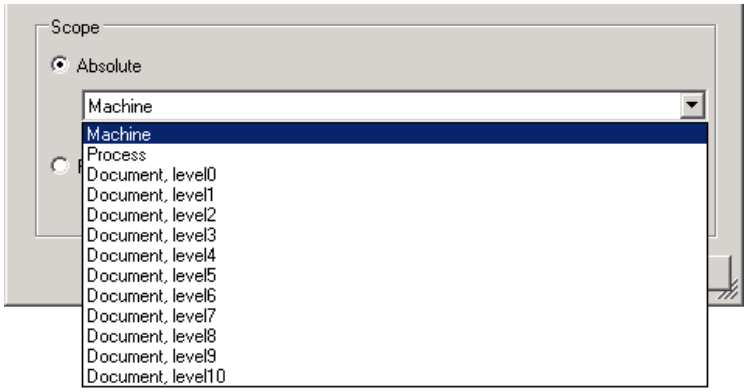


Figure 12.14 - Absolute Scope Options

Alternatively, you can dynamically define a global aliases and its value. That is, defining a global alias that is not in the GAS configuration database. To set the global alias or value, click the "Add" button in the bottom **Aliases** section of the dialog box.

This displays the **Edit Global Aliases** dialog box as shown in Figure 12.15 where you select a global alias, optionally a value, and the scope.

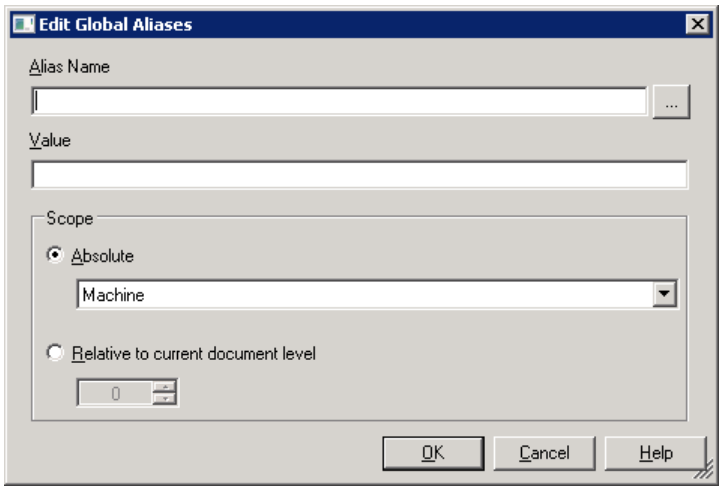


Figure 12.15 - Edit a Display's Global Aliases

12.6.3 Pick Actions in GraphWorX64

You can use global aliases in a Pick action dynamic to set a state, change a value, load a different display, or perform any action where variable substitution allows you to generalize the action at runtime. As a result, a runtime operator can set a global alias to a specific value with a click of a button.

Keep in mind that Pick actions can be added to nearly any object you include in a display. Furthermore, buttons and the Function Key dynamic have Pick actions already embedded in them.

Commands with which global aliases can be associated include:

- **Pick action-Load display:** Sets the global aliases in the newly loaded display.
- **Pick action-Set global aliases:** Sets the specified global alias theme value in the currently open display; can apply to the whole display or a specified object or symbol.
- **Display tabs:** Each tab can include a set of global aliases.

12.6.4 Global Aliasing with TrendWorX64

You can also use global aliases in OPC tags in TrendWorX64 Viewer when you create a new pen, using global aliases in the data source path just as you would in GraphWorX64.

12.6.5 Global Aliasing with AlarmWorX64

You can use global aliases in OPC alarms in AlarmWorX64 Viewer in the same way you might use them with displays in GraphWorX64. You can use a global alias to identify all or part of an OPC tag for an alarm.

An AlarmWorX64 configuration can be set to a particular theme as a runtime setting. In the AlarmWorX64 Server Configurator, use the database configuration, as shown in Figure 12.16 to specify the theme using the "GAS Theme" field. Clicking on the ellipsis button opens the **Set Global Themes and Aliases** dialog box, as shown in Figure 12.12, where you set Global alias themes and aliases the same way you did for GraphWorX64.

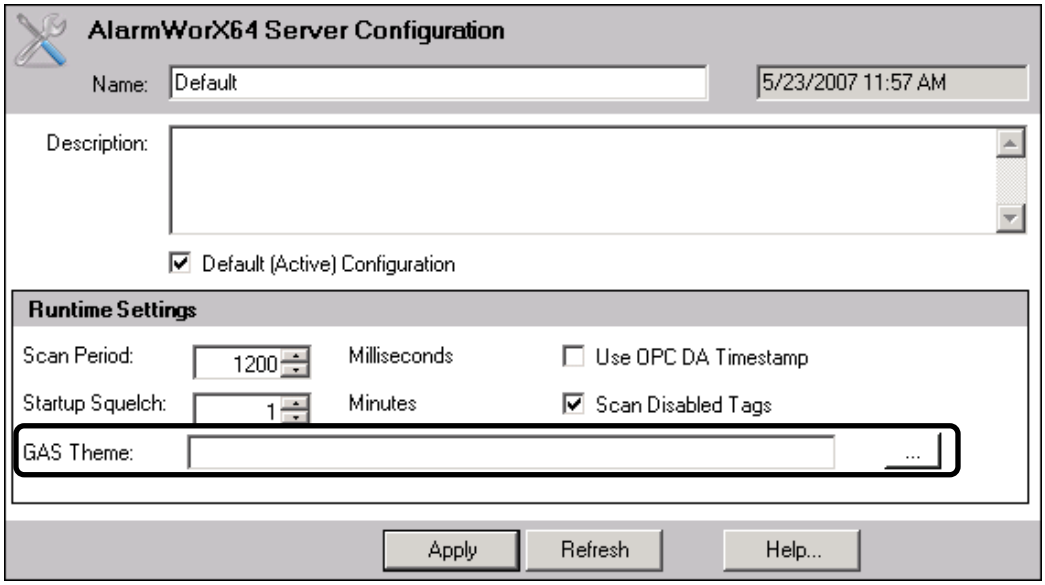


Figure 12.16 - AlarmWorX64 Server Configurator Sets the Theme

Lab: Global Aliasing L12

12.1. Lab Overview

On the production floor, there are many tanks running at the same time producing different types of drinks. However, we do not need to look at each of the mixer tanks at the same time, and therefore, we do not need to create different displays for each of the tanks. Instead, we will use Global Aliasing to create a dynamic display that can view each of the tanks.

Estimated time to complete this lab: 20 minutes

12.1.1. Task Checklist

In this lab, you will complete the following tasks:

- Create a global alias configuration database
- Create an alias theme
- Create theme items
- Create a global alias
- Assign values to different themes of the alias
- Activate the global aliasing configuration

12.1.2. Support Files

To complete this lab, you will need the following file(s):

- icTonicsHome.gdfx
- icTonicsTankMonitor.gdfx
- icTonicsTrends.gdfx

12.2. Global Alias Configuration

We will need to define the global alias and the corresponding value that the alias should resolve to during runtime. All this is done through a configuration file. In this section, we will configure the alias, the alias value, and a theme that we can switch to during runtime.

1. Open the Workbench if you closed it after finishing the previous lab.
2. Click the "Global Aliasing"  icon in the Project Explorer.
3. Right-click on localhost and select "**New Database**" to create a new configuration database. Leave "Use Default SQL Server" checked. Click on "**Next**".
4. In the **Create Database Wizard** dialog, change the database name to "**icTonicsGASConfig**".
5. **Uncheck** the "**Include Sample Data**" checkbox and click on the "**Finish**" button.
6. Expand the configuration once you see it in the Project Explorer. Right-click on **Themes** and select **"+ Theme**" to create a new theme.
7. In the **New Theme** tab, give name the theme "**TanksTheme**" and then click on "**Apply**".
8. In the Project Explorer, right-click on **TanksTheme** and select **"+ Theme Value**" to create a new theme value for **TanksTheme**.
9. **Name** this theme value "**Tank1Theme**" in the Project Explorer.
10. **Check** the box "**Default Theme Value**", then click "**Apply**".
11. Right-click on **Tank1Theme** and select "Multiply".
12. In the **Multiply Options** dialog box, change the "**Number of Items**" to **2**. Change the "**First Number**" to **2** and the "**Numeric Places**" to **1**.

13. Also, change the Base Value of the Name property to "Tank{0}". Your dialog box should look like Figure L12.1.

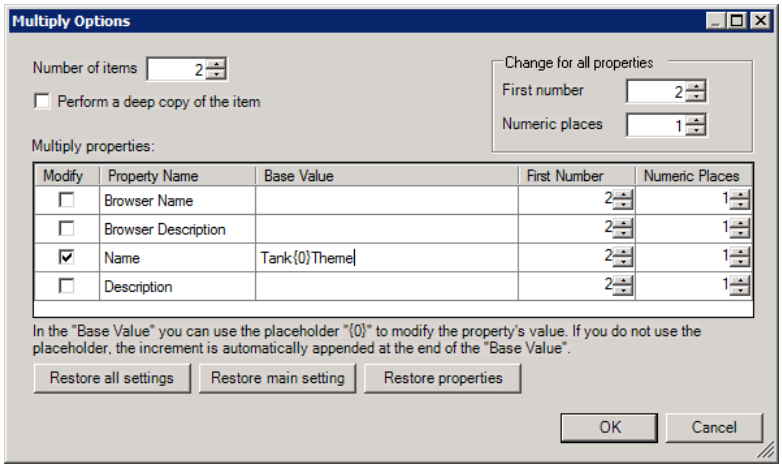


Figure L12.1 – Multiplying a Theme Value

14. Click on "OK" to start the multiplication. Name the themes Tank1Theme, Tank2Theme, and Tank3Theme.

15. Right-click on Aliases and select "+ Alias" to create a new global alias.

16. Call this Alias "TanksAlias" and choose "TanksTheme" as the theme in the dropdown list.

17. Notice once you choose the theme, the **Alias Values** section expands to display the theme's values.

18. Set the values in the Alias Value column as shown in Table L12.1, and then click on Apply to save the configurations.

Table L12.1 - Alias Values for Tanks

Theme Value	Alias Value
Tank1Theme	Tank1
Tank2Theme	Tank2
Tank3Theme	Tank3

19. Repeat steps 15-18 to create another alias called "**AlarmFilter**" with values as indicated in the table below.

Table L12.2 - Alias Values for AlarmFilter

Theme Value	Alias Value
Tank1Theme	Tank1 Level
Tank2Theme	Tank2 Level
Tank3Theme	Tank3 Level

20. Once you are done, make the database the active configuration by right-clicking on icTonicsGASConfig and selecting “Make Active”.

21. You will now have to reload the Global Aliasing configuration by clicking the “Reload Configuration” button on the ribbon for the new configuration to take effect.

12.3. GraphWorX64 Implementation

Now that we have a Global Alias configuration in place we will need to configure our displays to use these aliases.

1. Open the **icTonicsTrends** display and select the Tank1 label.

2. Add a process point dynamic to this label by selecting the Dynamics tab on the lower right corner and clicking on the process point icon  on the top of the explorer window. When the Data Browser comes up, select the Global Aliases tab, expand Global Aliases in the bottom pane, select TanksAlias, and click on the “**Copy Selection to Textbox**” button.

3. Check the “**Treat the Result as a Literal String**” checkbox. The Data Browser should look like the example in Figure L12.2. Click on “OK” to confirm the selection and close the Data Browser.

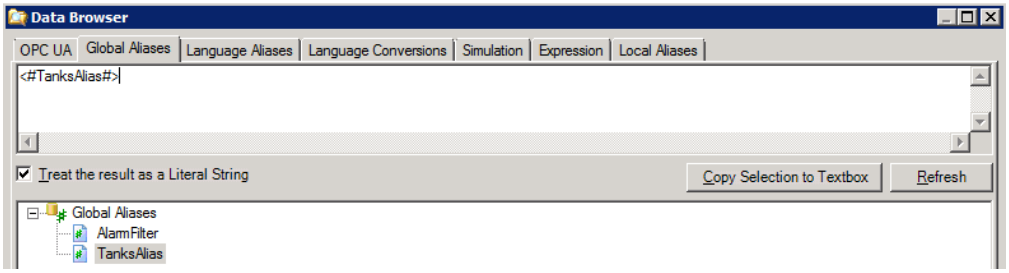


Figure L12.2 - Browsing for Global Alias

4. Double-click on the Trend Viewer control in the icTonicsTrends display to bring up the configuration dialog box. Click on your real-time pen, find “Tank1” in the data source, and change it to: <#TanksAlias#>. “<#” and “#>” are the delimiters for Global Aliases.

5. Change the description to read “Real Time <#TanksAlias#> Level” (include the quotes).

6. Do the same thing for the other real-time historical tags configured in the control.

7. Click on “Close” to close the control. Then save the display.

8. Open the icTonicsTankMonitor display.

9. In the Home tab of the ribbon, locate the Text section and click on the down arrow for the "Find" action. Select "Replace".

10. When the **Replace** dialog box comes up, in the Find Type section, check the "Dynamic Data Source" checkbox.

11. In the "**Find What**" field, type in "Tank1" and in the "**Replace With**" field type in **<#TanksAlias#>.** Your Replace dialog box should look like Figure L12.3.

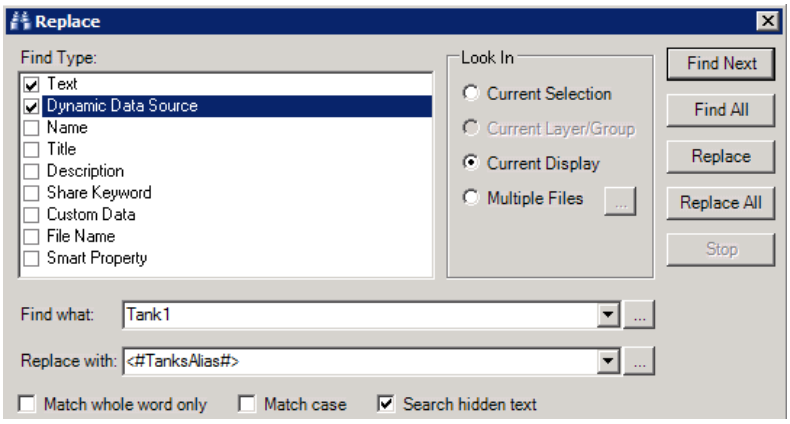


Figure L12.3 - Search and Replace

12. Click on the "**Replace All**" button to start the replacement. Once it is done, you should get a list of the objects that have been replaced in the results section. Close the dialog box and save the display.

13. Create a button called "**Mixer Tank 1**".

14. In the Dynamic properties of the button, find the "**Set Global Aliases**" in the Command field and click on the dropdown button in the GlobalAliases field under Command Parameters section. Select **Global Aliases**.

    Note: Make sure GraphWorX64 is in "Advanced Mode" or you won't be able to see all your options.

15. In the "**Select Global Themes and Aliases"** dialog box, click on the "**Add**" button in the Themes section to bring up the "**Edit Global Themes"** dialog box.

16. Under "**Scope**" select **Absolute** and in the dropdown list select **Machine.**

17. In the Themes dropdown list, select "**TanksTheme**". Select **Tank1Theme** in the Theme Items window and click on "OK" all the way back to the display.

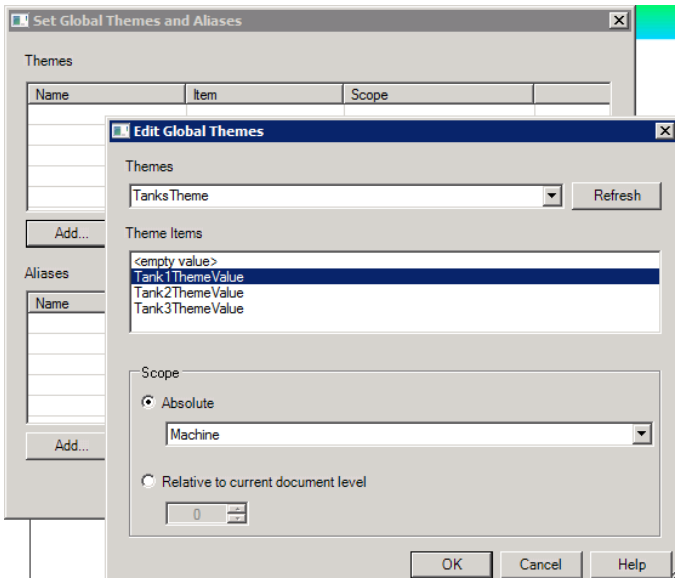


Figure L12.4 - Edit Global Themes Dialog Box

18. Now duplicate the "**Mixer Tank 1**" button twice. Repeat steps 14-16 for each duplicate, except set the global alias theme value to Tank2 on the second button and Tank3 on the third button, and change each button's text accordingly, too.

    Note: Beware of typos or extra spaces as any wrong character will prohibit your alias from working.

19. Your finished display should look like Figure L12.5. Go to runtime and load the icTonicsTankMonitor display with the different buttons. You should see your tags resolve to different paths and showing different values.

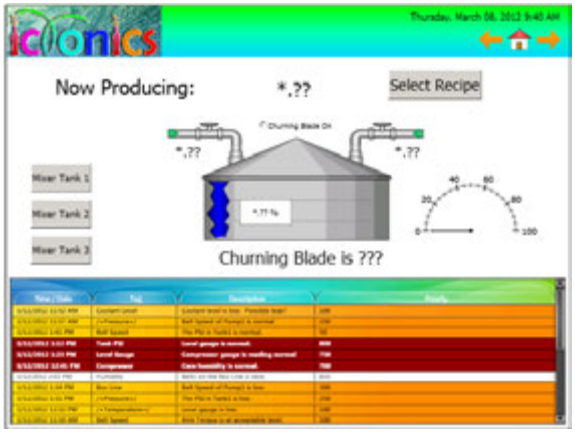


Figure L12.5 – Finished icTonicsTankMonitor Display

Review: Global Aliases R12

12.1 Objectives Review

You should now know and understand the following topics:

- General uses for global aliases
- How to create and use themes
- How to create and use global aliases

12.2 Summary

A global alias is a data string that enables you to reference multiple data sources with one unique name. Any process point can contain one or more global aliases. Global aliases are stored and maintained in a SQL database called the Global Aliasing System (GAS) database, and those aliases can be accessed over the network by clients. Global aliases, themes, and alias values are created and configured in the Global Aliasing interface of Workbench.

Each global alias can be associated with a theme. The power of a theme is in the number of aliases that you associate with it. A theme links Global Aliases together. The Global Aliasing database allows you to assign to aliases different values based on a theme so that, in runtime, when the theme changes, the values change. Themes can change dynamically; either programmatically through automation APIs or using a Pick action dynamic.

Alias resolution is the process of assigning a specific tag value to an alias. The resolution of aliases is controlled by the theme scope (that is, on what level a theme is applied to your application). The Global Aliasing system supports three different levels of aliases: machine-level, process-level, and document-level.

12.3 Terms

Alias Theme Alias Value Global Alias

12.4 Test Your Knowledge

In this chapter, we discussed global aliases, themes, and how they are used in GENESIS64 applications. Before continuing to the next chapter, make sure you can answer the questions below.

1. True or False: All global aliases must be configured and stored in the GAS database.
2. True or False: You can use a global alias anywhere in the path of an OPC tag.
3. True or False: Themes do not provide any runtime use.
4. True or False: An alias value can contain a global alias.
5. True or False: A dynamically-created global alias can be reused in any display created using any ICONICS viewer.

Language Aliases 13

The Language Aliasing System (LAS) provides a way to translate text strings and to convert measurements for any languages required for use in ICONICS applications:

- **Translation**: A specially delimited or tagged string (such as a word or phrase) is replaced by an appropriate, translated string for the selected language.
- **Conversion**: Different cultures use different units of measurement, so the conversion system also converts one type of measurement unit to another.

The LAS translates and converts using language aliases. An *alias* is a name for an object that is different from the object's "real" name. In English, a rose is "a rose", but in Italian a rose is "una rosa", in German it is "eine rose", in Dutch "een roos", and so forth. Using a language alias, you can define the text string for "a rose" in each language so that the displays you create can switch to the local client machine's language; your Italian users see "una rosa" and your German users see "eine rose".

There are two major components of the LAS: the **centralized database** and any **client** application that uses the language switching mechanism. All language variations and conversion expressions are stored in and retrieved from a centralized database configuration (Microsoft SQL Server or Microsoft SQL Server Express). The centralized database is maintained using the Language Aliasing configurator in the GENESIS64 Workbench. Language aliases are stored in the central global database and those aliases can be accessed over the network by clients. Changes you make to the LAS database are propagated to clients within roughly 1 to 3 seconds. In this chapter, you will learn to configure language aliases and use them in client applications.

After completing this chapter, you should know how to:

- Convert text strings for display in multiple languages
- Use language aliases for converting units of measure by language and locale

13.1 Background

The language/locale setting in Windows is based on an ISO standard. A locale is similar to a culture, differentiating English (United States) from English (United Kingdom). There are 264 defined settings available in Windows Vista, including language/locale. When a language is set in Windows on a client machine, Windows' user interfaces retrieve the language alias translations for that language and cache the information locally on the client displaying the translations in the local display.

In Vista, this language/locale setting establishes the language for the user account. GENESIS64 uses that same setting to determine the language/locale properties to be used in GENESIS64 applications at runtime. GENESIS64's LAS returns language settings for that language for any alias objects needing it for language translation. Similarly, GENESIS64 reads the unit binding for the current language to obtain unit conversions for values in the display.

Although GENESIS64 uses the language setting to determine the program action, it does not use any other Windows feature or dictionary to supply conversions. In order for your displays to have the correct language alias support, you must configure translations and conversions for the languages in which you need them, by using the LAS configurator.

Language aliases are applied to the client based on the active configuration. Only one configuration is active at any one time, and that database can use only one language setting for displaying data on the client. Languages cannot be mixed in a display. This means that when you use a Pick action to change the language/locale in a display, all of the language aliases defined in that display change to the currently selected language.

13.2 Configuring Language Aliasing

There are two ways to configure language aliasing: click the Language Aliasing icon at the bottom of the GENESIS64 Workbench's Project Explorer, or from the Windows desktop click **Start → All Programs → ICONICS GENESIS64 → Tools → Language Aliasing**. The Language Aliasing (LAS) configurator is shown in Figure 13.1.



Figure 13.1 - Language Aliasing Configuration Interface

The LAS configurator contains a tree control in the Project Explorer, where you select items you want to edit or add. As with all GENESIS64 applications, whenever you edit or view an item, a configuration form opens in the content area.

You can perform different actions for different types of items in the tree. All actions allowed for an item appear in the ribbon when you select an item. You can also right-click an item to pop up a menu, as shown in Figure 13.2.

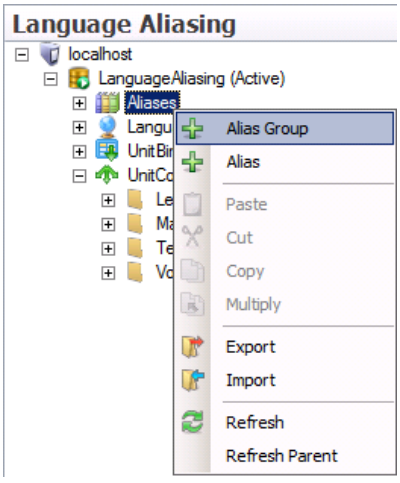


Figure 13.2 - Performing Actions in the Project Explorer

13.2.1 Configuring Languages

Start configuring the LAS by deciding which languages you are going to use, and then define them by right-clicking **Languages** in the Project Explorer tree, and add each language. These languages will be used for translating text, but will also be used for converting units of measure for the locale. Therefore, both the language and the locale are important for each language definition.

Adding a language is a simple process where you define the following information, as shown in Figure 13.3:

- **Language Info:** Select the **Culture Name** (that is, the language and locale) from the list provided by clicking the **Change Language** button. Give it the **Display Name** you want to use for its identification.
- **Properties:** Enable or disable the translation, enter a Description for it, and indicate whether the language should be the active default language for clients that access the LAS.

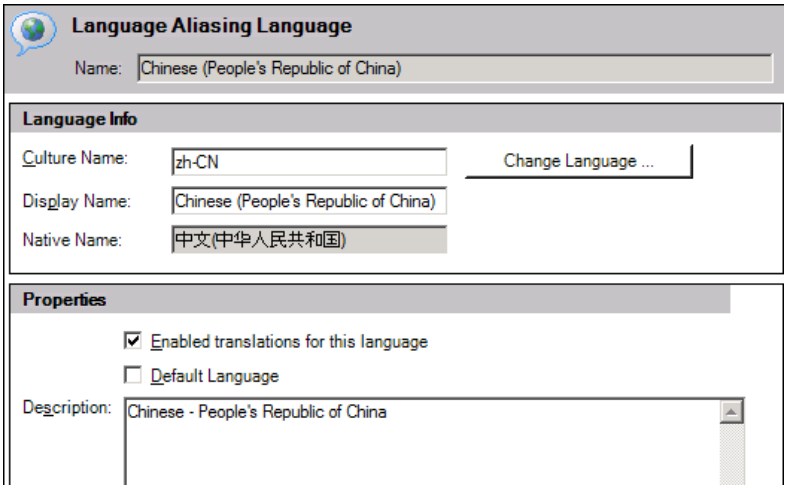


Figure 13.3 - New Language Definition

Once you have defined all of your languages, you can create language aliases for them. Remember there are two types of language aliases: translation aliases and conversion aliases. Each type is created in a different way. The following sections will walk you through both of them.

13.2.2 Configuring Language Aliases for Translating Text

Language alias translation takes an alias string in your text, process point, trend pen, etc., and converts it to a lookup value stored for different languages in the alias definition based on the current selection or a system's locale. This section describes how to create and use aliases for translating strings.

Translation Language Aliases are bundled in the **Aliases** folder in the Project Explorer. You can create language aliases in the Aliases folder or you can create subfolders for them. To add an alias, you need to specify the following information, as is shown in Figure 13.4:

- Give the alias a name that identifies the text that it translates. Describe the alias, and specify the Default Translation text that will be used if no language-specific translation is available for the language that is active during runtime.
- Then provide text to be used for each language you have created. A client will see only the translation text that corresponds to the currently active language for the client machine.

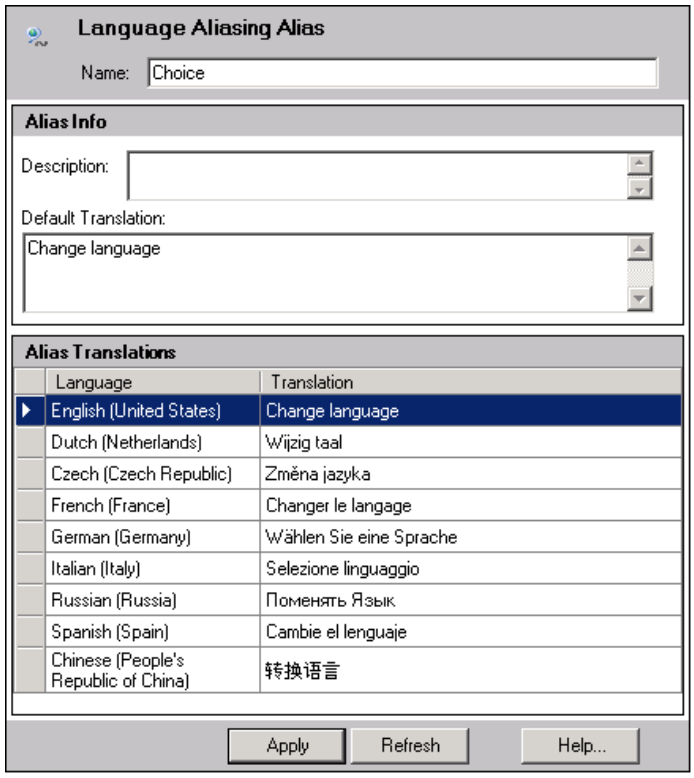


Figure 13.4 - Text Translation Alias Definition

13.2.3 Using Language Aliases

Once a language alias is created, you can use it in any GENESIS64 application. To use a translation alias, browse for it using the ICONICS Data Browser's **Language Aliases** tab, shown in Figure 13.5.

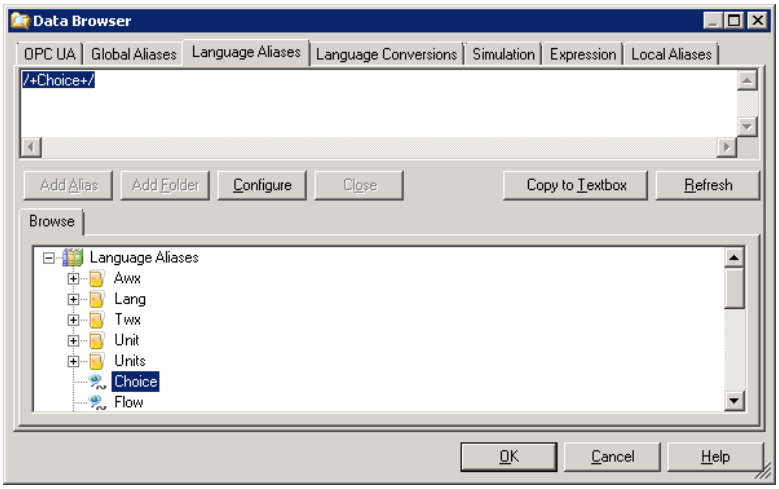


Figure 13.5 - Language Alias Selection in the Data Browser

In the field at the top of the tab, you can easily recognize the language alias by its special syntax. The language alias name is delimited by the characters /+ at the beginning and +/ at the end. For example, the alias name **Title** when used in your application will look like this: /+**Title**+/ *or* {{/+**Title**+/}}. A language alias is considered to be a data point and, like a global alias, can be used in the path of an OPC tag. When it is not being used as part of a data source, it must be enclosed in double braces {{ }}.

> **Note**: If you are planning a new project, consider the advantages of integrating translation capabilities as part of the initial project development. Integrating translation capabilities after a project is done will require a considerable amount of rework.

Also, notice the Configure button near the top of the Data Browser. Clicking this button opens the alias's definition in a tab next to the Browse tab so you can see it before selecting it. To close the alias's tab, you simply click the Close button.

13.2.4 Configuring Conversions for Units of Measure

Because languages and cultures use varying methods of measurements such as distance, temperature, and currency, the language aliasing system (LAS) provides a way to convert units of measure and their expressions.

You define **Unit Conversion** and **Unit Binding** objects in the **UnitConversions** and **UnitBinding** folders in the Project Explorer. You can create subfolders if necessary (as shown in Figure 13.6) to be able to find conversions later. Each is described in this section using the example of converting a temperature display in Celsius (formerly Centigrade) to Fahrenheit.

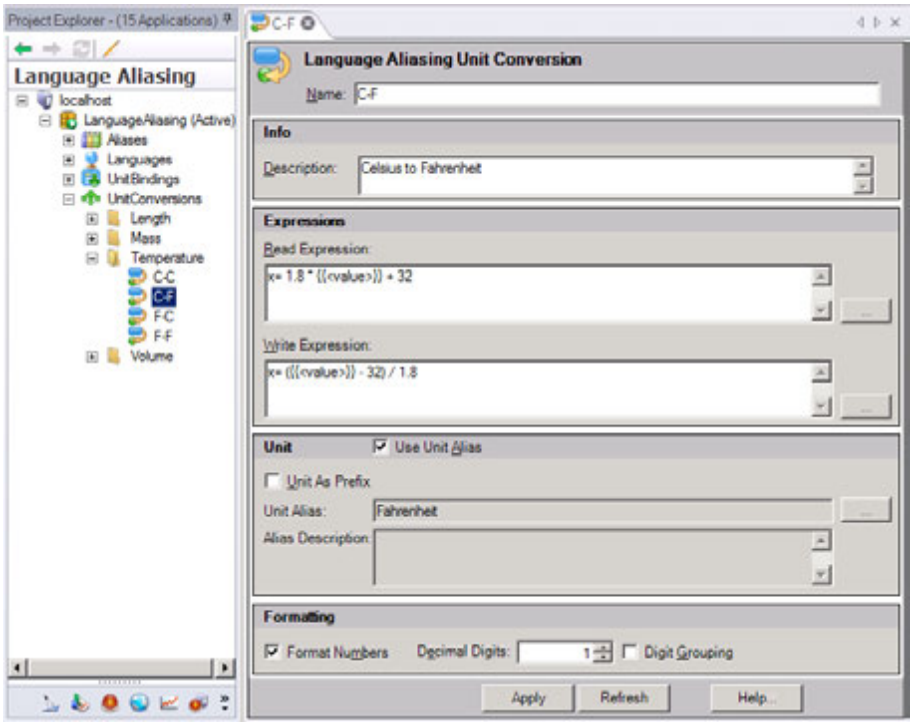


Figure 13.6 - Unit Conversion Definition

13.2.4.1 Unit Conversion

To define a conversion from Celsius to Fahrenheit, you start by defining unit conversions, as shown in Figure 13.6.

A Unit Conversion definition consists of a **Name** that describes the conversion direction (such as Celsius to Fahrenheit), **Info** that describes the conversion if you think it is needed, and **Expressions** that define the read and write expressions for converting the units. You can use the expression engine shown in Figure 13.7, which is available on the Expression tab in the Data Browser.

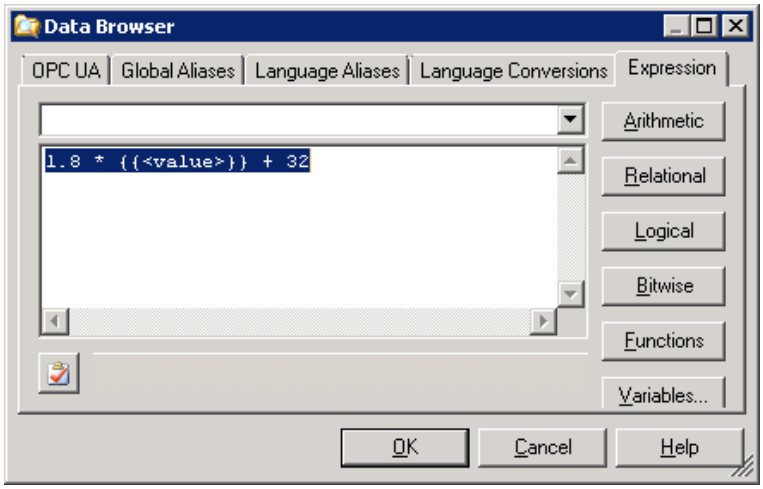


Figure 13.7 - Expression Tab in the Data Browser

- The **Read Expression** specifies the calculation that will be applied to the original read value – **{{<value>}}** – in order to display it in its converted form. For example, the calculation reads a value in Celsius and converts it to Fahrenheit for display.
- The **Write Expression** specifies the calculation that will be performed on a value – **{{<value>}}** – that a user manually inputs into a converted data point. For example, the calculation takes a user-provided value in Fahrenheit and converts the value so that it can be written in Celsius.

You also specify how the **unit** of measure will be labeled when it displays. If no label is needed for the unit, leave the "Use Unit Alias" checkbox empty; otherwise, put a check the checkbox. You can type a label in the "Unit Alias" field (especially if the label is a symbol that doesn't require string translation, such as €). Or you can click the ellipsis button to select a language alias to be used for the unit. This will allow the label to display translated for each language. If the label prefixes the unit value (as in €10), check the "Use as Prefix" checkbox. Otherwise, the label will be suffixed to the value (as in 10¢).

If you want to use any of the available formatting for a converted value, check the "Format Numbers" checkbox. Then, enter the number of Decimal Digits, and/or check the "Digit Grouping" checkbox. For example, specify that the unit is to appear with one decimal place (22.7); or, for example for large currency values, that you want them to appear with commas in the form: $1,500,000.00.

After you define a conversion, make sure you define its *reciprocal* conversion (that is, the conversion in the other direction). Using the Celsius-to-Fahrenheit (C-to-F) conversion as an example, you need to define C-to-F and F-to-C, but you also need to define F-to-F and C-to-C for situations where no conversion is performed.

> **Note:** The number of conversions you must define is n^2 where ***n*** is the number of different units of measure being converted. For example, if you are performing conversions among the three different types of tonnage – short tons (ST), long tons (LT), and metric tons (MT) – then you would need definitions for ST-to-ST, LT-to-LT, MT-to-MT, ST-to-LT, ST-to-MT, LT-to-ST, LT-to-MT, MT-to-ST, and MT-to-ST – or nine separate conversions.

13.2.4.2 Unit Binding

After defining all the necessary unit conversions, you must assign them to each of your languages using ***unit binding***. This determines which conversion is used for a given language. For example, which unit conversion, if any, should take place while the language Dutch is currently active?

Unit bindings provide the instructions for unit conversion based on locale. The unit binding takes values in the current unit of measure and assigns a language-specific unit conversion for the locale. With this, your unit conversions can change from locale to locale, if necessary. Unit bindings will set the data type.

For the following example, refer to Figure 13.8. In this example, the default unit conversion is C-to-C (Celsius to Celsius; no conversion is performed), which means that all languages/locales use this as default unless otherwise specified. You define two unit bindings called FromCelsius and FromFahrenheit. Each contains a table of languages/locales. For FromCelsius, the C-to-C default is overridden for only two language/locales: English (United States) and Chinese (People's Republic of China) where the temperature must be displayed in Fahrenheit.

When adding a new unit binding, as shown in Figure 13.8, you have to define the **Name** of this unit binding, an optional **Description**, and the **Data Type** of the alias and the **Default Unit Conversion** that will be used for all languages/locales unless specified otherwise in the Language Unit Conversions table. Then, in the **Language Unit Conversions** table, specify the unit conversion to be used if you do not want the default unit conversion to be used for a particular language/locale.

The unit conversion that will be performed during runtime will be determined by the language that is currently active.

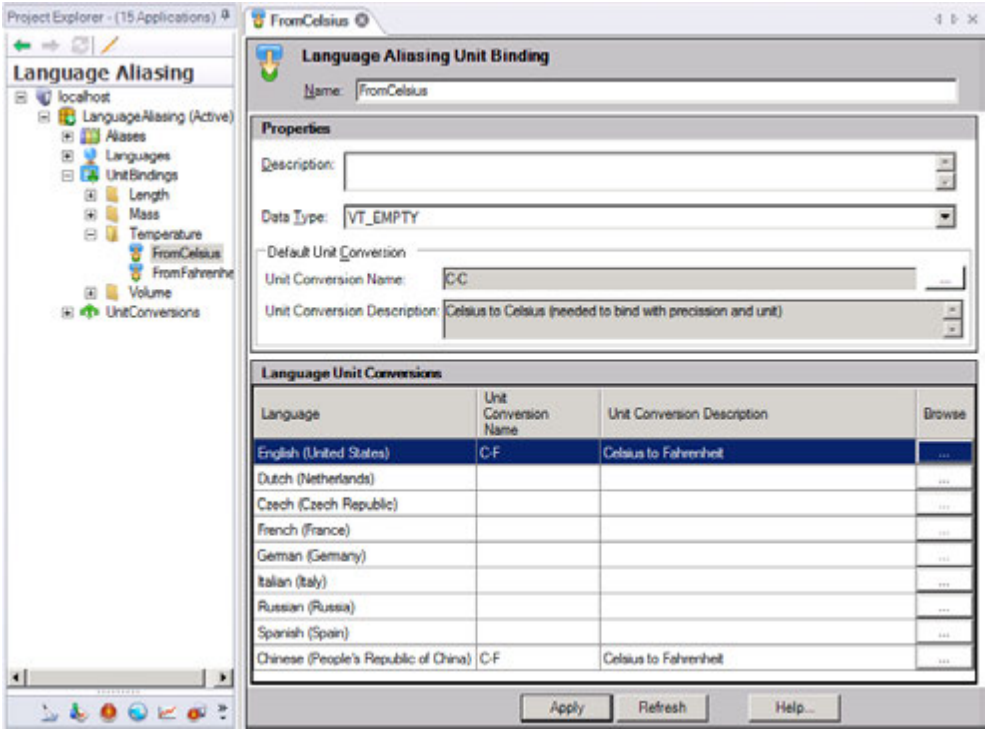


Figure 13.8 - Defining Unit Bindings

13.2.5 Using Unit Conversions

Similar to language aliases, you can use unit conversions across all GENESIS64 applications. However, unit conversions cannot be used on their own; they are always used with a data source.

For example, this is a data point:

@sim64:Long.Ramp(10,-50.0,50.0,140).Value

To convert its value using the **FromCelsius** unit binding conversions located in Temperature folder, you need to add the following prefix:

=Temperature\FromCelsius:Value=

The final data source will look like this:

=Temperature\FromCelsius:Value=@sim64:Long.Ramp(10,-50.0,50.0,140).Value

Fortunately, you do not have to remember this syntax. Instead, you can use the Data Browser to create it. To do so, first, select the data point you would like to use in your application. Next, keep the Data Browser open and go to the **Language Conversions** tab and select the conversion you would like to apply for this data point, as shown in Figure 13.9.

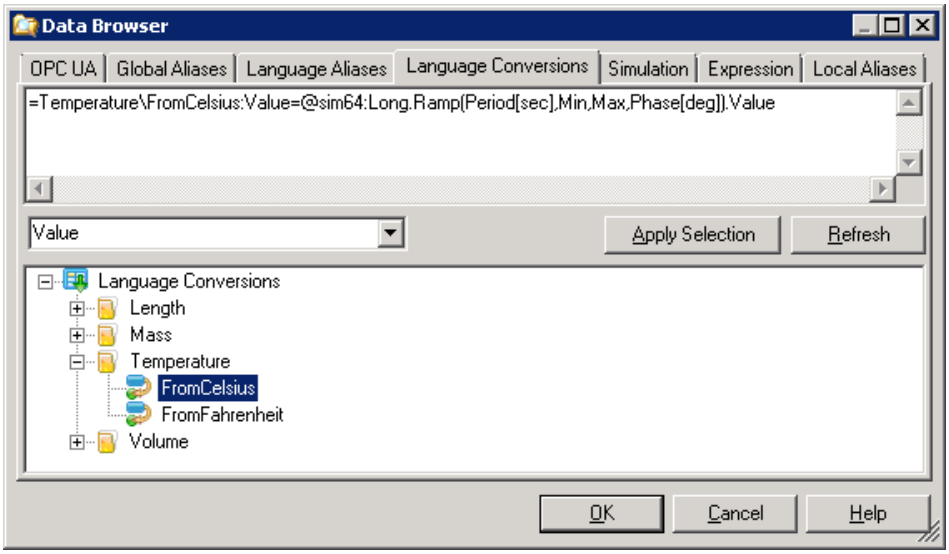


Figure 13.9 - Language Conversions Tab in the Data Browser

Figure 13.9 above has been enlarged so that you can see that the syntax of the conversion alias is made up of the path and name of the alias as well as a special string delimited by a colon (as in **:Value**). This string signifies the conversion output type. You can insert the output type manually or you can select it from the drop-down menu in the Data Browser, which is shown in Figure 13.10.

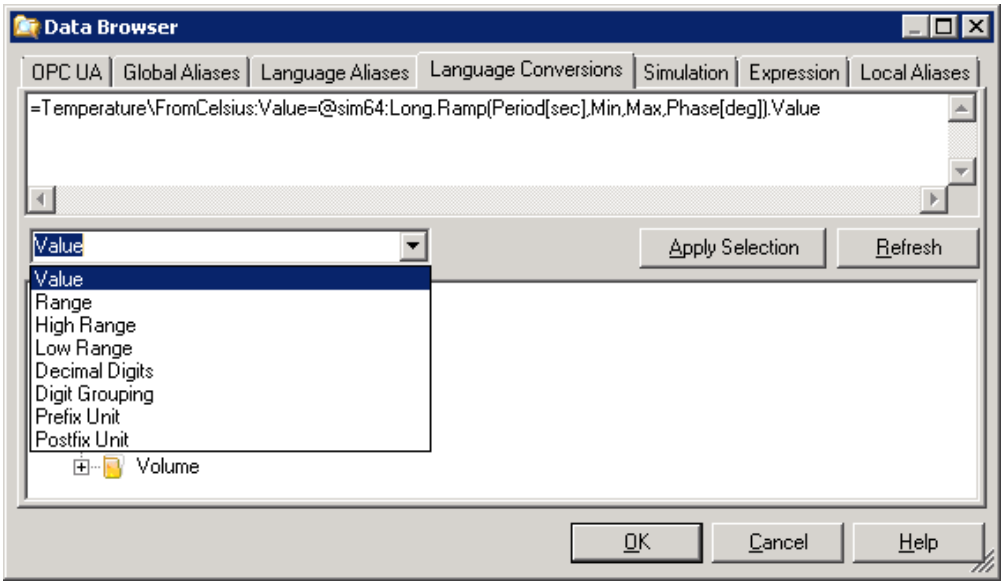


Figure 13.10 - Selecting a Conversion Output Type

13.3 Language Alias Switching

The easiest way to see a language alias at work in runtime is to use a button with the **Set Language** Pick action in a GraphWorX64 display. In order to create such a button, you either insert a button object into your GraphWorX64 display or create any new object and assign it a Pick action, which is one of GraphWorX64 dynamics.

Figure 13.11 shows a selected object (the flag surrounded by squares on the left) with the **Set Language** command assigned to it (shown at the right).

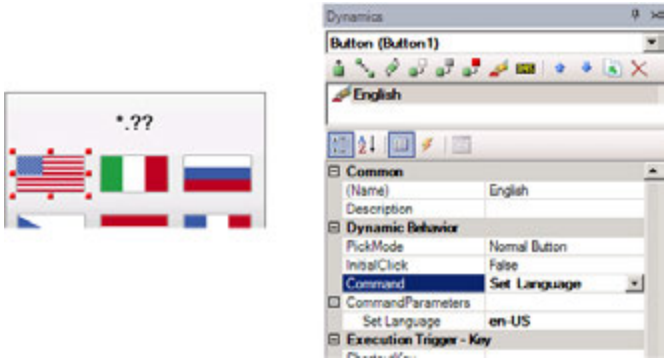


Figure 13.11 - Object with the Pick Action Set Language

The **Set Language** property is a Pick action command parameter. Here, you select a language from the list of the languages you defined earlier, as shown in Figure 13.12.

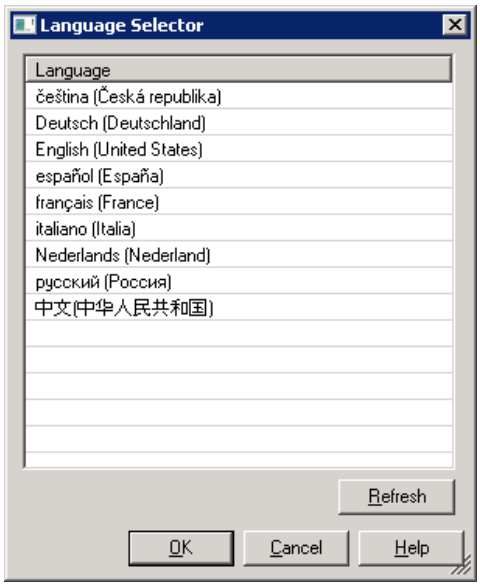


Figure 13.12 - Selecting a Language for Set Language Command Parameter

> **Note:** For demonstration and testing purposes, make sure you choose a language other than the language that will be active on the machine where you are testing.

Clicking on the button in runtime will change the current active language to the one defined in the **Set Language** box (which, in Figure 13.11, is en-US). As a result, the runtime display shows the translated value that is stored in the LAS database for that alias in the specified language.

Remember that language aliases are applied to the client based on the active SQL database. Because only one SQL database is active at any one time, and that database can use only one language setting for displaying data on the client, only one language is used in any given display. This means that when you use a Pick action to change the language/locale in a display, all of the language aliases defined in that display change to the currently selected language.

Lab: A Multi-Lingual System L13

13.1. Lab Overview

IcTonics is an international company with facilities in different locations around the world. This makes it crucial that they have dynamic displays that display in the local language for their operators. In this lab we will edit our displays and create translations to accommodate different languages.

Estimated time to complete this lab: 20 minutes

13.1.1. Task Checklist

In this lab, you will complete the following tasks:

- Create a new language configuration database
- Create alias groups
- Create language aliases
- Set translations for language aliases
- Activate the language configuration database

13.1.2. Support Files

To complete this lab, you will need the following file(s):

- icTonicsHome.gdfx

13.2. Language Configuration

In order to use language aliasing, we will need to provide translations to certain phrases that we will use in a configuration file. In this section, we will setup such a configuration.

1. If you closed the Workbench after the previous lab, go ahead and reopen it.
2. In the Project Explorer, select "Language Aliasing" to open it.
3. Right-click on localhost and select "New Database" to create a new configuration database. Click on "Next".
4. Change the Database name to "icTonicsLanguageConfig" and make sure that the "Include Sample Data" checkbox is checked. Click on "Finish" to create the new configuration database.
5. Once it finishes, expand icTonicsLanguageConfig's Languages folder.
6. By default, there are nine languages configured as shown in Figure L13.1. For our purposes, we will use Chinese, Czech, English, German, Italian, and Spanish.



Figure L13.1 - Default Languages

7. Right-click on Aliases and select + Alias Group to add a group; name this group "icTonics".
8. Right-click on the icTonics group and select "+ Alias"; name this alias Welcome. You may give it a description if you wish.

9. Give this a default translation of "Welcome". Then enter the translation strings listed in Table L13.1.

Table L13.1 - Translation of Welcome

Language	Translation
English	Welcome
Czech	Vítejte
German	Willkommen
Italian	Benvenuto
Spanish	Bienvenido
Chinese	□迎

Note: If you are using a training image, you can change the language to Chinese by pressing Left Alt + Shift+1 and type "HuanYing" to select the correct Chinese characters. Then use Left+Alt+0 to switch back to English. You can also copy and paste the values from Translations.txt in C:\Training.

10. Click on "Apply" and then repeat steps 8-9, only this time, name the alias "Login" and enter the translation strings listed in Table L13.2.

Table L13.2 - Translation for Login

Language	Translation
English	Login
Czech	Přihlášení
German	Anmelden
Italian	Accesso
Spanish	Acceso
Chinese	登□

Note: You can change the language to Czech by pressing Left Alt + Shift +2 and type in 5 for ř and 3 for š. You can also change to the Chinese keyboard and type in "denglu" to select the correct characters. You can also copy and paste the values from Translations.txt in C:\Training.

11. Apply the changes. Right-click on the icTonicsLanguageConfig and select "Make Active" to make this configuration database active.

13.3. Using Language Aliasing in GraphWorX64

In this section we will change some of our text objects to include language aliasing. We will also add buttons to our display for language switching.

1. Open the icTonicsHome.gdfx display.
2. Change the text of your "Welcome" text label to /+icTonics\Welcome+/. This is the Welcome alias we created in the previous section. The delimiters /+ and +/ indicate the beginning and end of a language alias.
3. Select the "Login" button and change the text of the button to /+icTonics\Login+/.
4. Add a new button and in its Pick action dynamic set its Command to "**Set Language**".
5. In the **Set Language** property in the Command Parameters section, click in the field and then click on the ellipse button. This brings up the **Language Selector** dialog box.
6. Select "English" and click on "OK" to close the dialog box. Change the text of the button to "English".
7. Duplicate this button five times and set the language and the text to each of the translated languages we are using. These buttons will allow us to change the language of the display during runtime.
8. Resize your buttons and align them on the bottom of the display. Then save the display. Your finished display should look similar to Figure L13.2.



Figure L13.2 - icTonicsHome Display

9. Go into runtime and switch to the different languages. You may need to resize your Welcome message or the login button to show the entire translated phrase.

   Note: Notice that the Time/Date dynamic in the template changes its format with the language automatically.

Review: Language Aliasing *R13*

13.1 Objectives Review

You should now know and understand the following topics:

- Converting text strings for display in multiple languages
- Converting units of measure by language and locale

13.2 Summary

Language Aliasing configuration is stored in one central MS SQL database and is maintained by a Language Aliasing interface that can be run from the Workbench. The configuration structure holds four items: Languages, Aliases, Unit Conversions, and Unit Bindings.

If you want to **translate** strings in your project, you have to add and edit **Aliases**. If you need to **convert** units, you must create new **Unit Conversions** and then **Bind** those conversions to your languages. Any change to the database is immediately propagated to all connected clients.

You can use Language Aliases all over the ICONICS product range. Usually you browse for your Aliases using the Data Browser. The Language Aliasing tab will appear if you are allowed to use **Language Aliases** or **Language Conversions**. The easiest way to change the language in runtime is to create a button whose Pick action command is set to **Set Language**.

13.3 Terms

Conversion Alias
Language Alias
Reciprocal Conversion
Set Language Command
Translation Alias
Unit Binding
Unit Conversion

13.4 Test Your Knowledge

In this chapter, we discussed Language Aliasing. Before you continue, make sure you can answer the following questions.

1. Where is the configuration of Language Aliasing stored?

    a. An SQL database
    b. An MS Access database
    c. A proprietary database

2. What do you need to do to notify clients of a LAS configuration change?

    a. Nothing
    b. Refresh the database
    c. Deactivate and reactivate the database

3. True or False: Two functions of Language aliasing are conversion of units and translation strings.

4. Where can you use LAS?

    a. All over the ICONICS product range
    b. Only in GraphWorX64
    c. Only in Process Points

5. How do you easily access the Aliases stored in your configuration when developing your project?

    a. Use the Data Browser
    b. Use the Language Alias configurator
    c. You have to type in the alias

6. Which is the correct format for a language alias?

    a. <#LanguageAlias#>
    b. “$LanguageAlias$”
    c. /+LanguageAlias+/
    d. <<LanguageAlias>>
    e. ~~LanguageAlias~~

GenTray and FrameWorX64 Server 14

Recently, GENESIS64 has taken a more service-oriented approach. The ICONICS GenTray has been improved to be a more service-focused monitoring application. The new GenTray controls the ICONICS services as well as the Windows services on a local or remote GENESIS64 system. This chapter discusses GenTray and the new benefits it provides.

In GENESIS64, the architecture for getting data from ICONICS servers and third-party OPC servers has been improved with a more centralized approach. The ICONICS FrameWorX64 server now mediates all data. This makes the ICONICS FrameWorX64 server a very important part of GENESIS64. Knowing how to configure it is vital to the success of a GENESIS64 project. This chapter discusses the various functions of the ICONICS FrameWorX64 server and outlines the areas that can be configured.

After completing this chapter, you should know how to:

- Use GenTray to control local and remote services
- Configure ICONICS FrameWorX64 server
- Set up redundancy and monitor redundant servers
- Web publish your displays

14.1. Interfacing with GenTray

GenTray is accessible through the Workbench, ICONICS' centralized configuration utility. In the Workbench, click on the "GenTray" icon to launch it, then in the Project Explorer expand localhost to view the services categorized as **Iconics Services** and **Others Services**, as shown in Figure 14.1. **Iconics Services** contains all ICONICS services while **Others Services** contains all non-ICONICS Windows services.

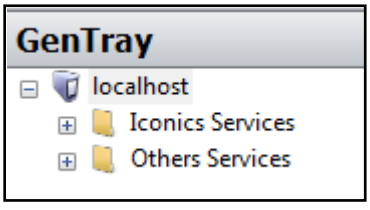


Figure 14.1 - GenTray

Expand either folder to see the services, as shown in Figure 14.2. Notice that an icon indicates each service's status. The running icon (with a green arrow) indicates that the service is running and the stopped icon (with a red square) indicates that it is stopped.

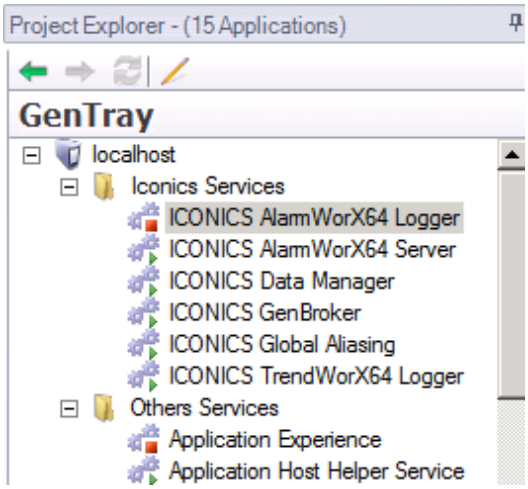


Figure 14.2 - Service Icons in GenTray

You can open a service's properties by double-clicking on the service. A tab that looks like Figure 14.3 appears on the right.

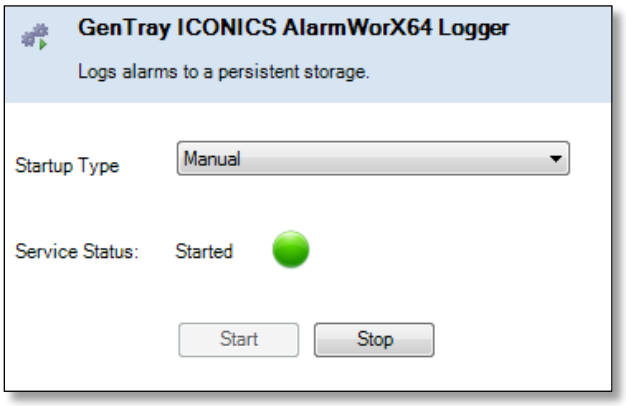


Figure 14.3 - Service control

In this dialog, the "Startup Type" determines how this service behaves. An **Automatic** service starts up when Windows starts up without requiring any login. A **Manual** service only starts when a user manually turns it on after logging onto the computer. A **Disabled** service is a service that is not needed; the system will prevent the service from starting up. You can start and stop the service by clicking on the "Start" and "Stop" buttons.

> **Note:** Depending on the operating system you are using, the Services Manager may contain more startup options.

14.2. FrameWorX64 Server Configuration

The ICONICS FrameWorX64 server is configured in the Workbench. The "FramWorX64 Server" icon launches the FrameWorX64 Server, which is shown with its nodes expanded in Figure 14.4.

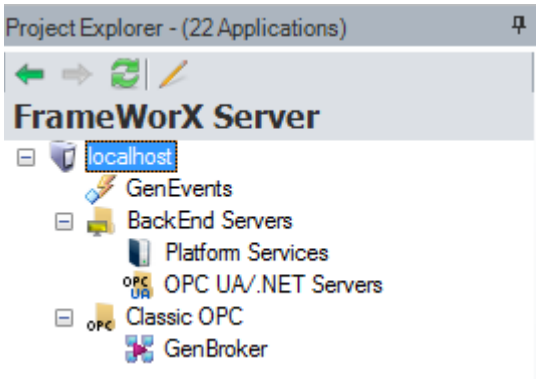


Figure 14.4 - FrameWorX64 Configuration

Notice there are three nodes defined in FrameWorX64 Server in the Project Explorer:

- **GenEvents:** exposes all events sent to it via the OPC Alarm and Events (AE) interface.
- **BackEnd Servers**, with a sub-node of **Platform Services**, which includes the option to add redundant Platform Services from other servers. Users must specify a Server Name as well as Primary and Secondary Machine Names. The second sub-node is the **OPC UA/.NET Servers**, where you configure remote OPC UA servers. You can use this as an alias for an OPC server with a long name as well.
- **Classic OPC**, with a sub-node of **GenBroker**, puts GENESIS64 in communication with legacy OPC servers residing on other machines in the same network or the internet. GenBroker also handles data redundancy for classic OPC servers.

14.2.1. GenEvents

GenEvents are messages generated by a GENESIS64 component. The GenEvent Server exposes all events sent to it via the OPC Alarm and Events (AE) interface. OPC AE clients, such as the AlarmWorX64 Viewer and Logger, can subscribe to the GenEvent Server to view and log these events. They can be error messages or general operation messages. These messages can be published to the NT Event Log on Windows and can be viewed using the Windows NT Event Viewer.

To configure GenEvents, double-click on **GenEvents** under **localhost** to open the GenEvents tab, shown in Figure 14.5.

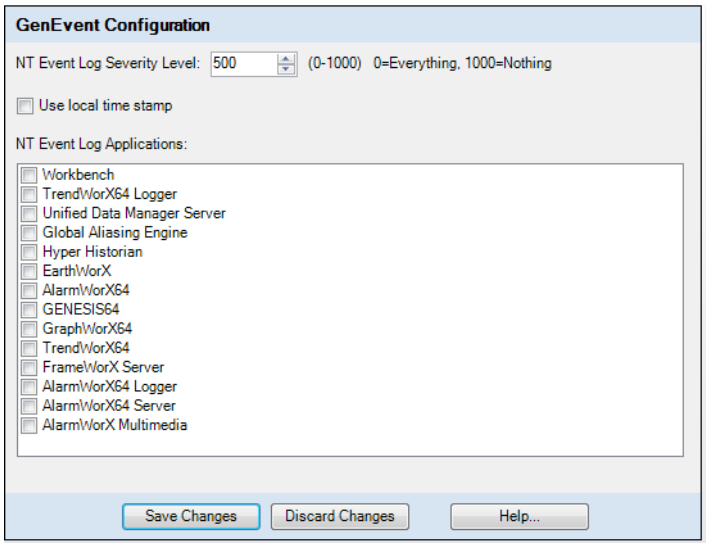


Figure 14.5 - GenEvents

Specify the **Severity Level** of the messages that you want written to the NT Application Event Log. The larger the number, the higher the message's severity must be in order for the event to enter into the NT Event Log. Setting it to zero (0) writes all event messages; setting it to 1000 writes no event messages.

Each event message has an associated timestamp. You can use the local time stamp by checking the "Use local time stamp" checkbox. If this checkbox is not checked, event messages are written using UTC time stamp.

In addition, in the "NT Event Log Applications" box, put a checkmark next to each component whose messages you want to include in the Application Event Log. Then save the configuration. Messages from the selected components that meet the severity criteria will be written to the NT Event Log as they happen.

14.2.2. Platform Services

Platform Services are the core services of GENESIS64. You can configure redundancy within the core by entering a Server Name, Primary Machine Name and Secondary Machine Name. Entering one pair will add a row to the configuration and allow you to add more. Click Apply to finish.

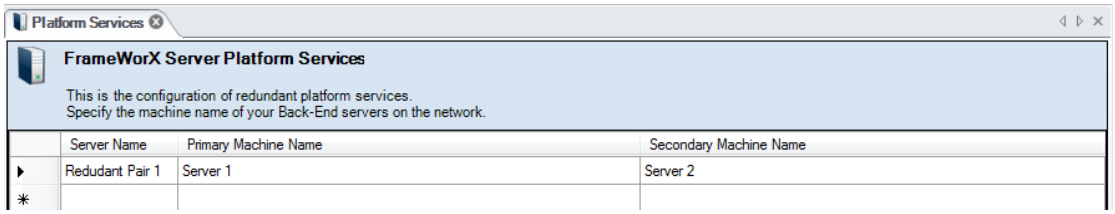


Figure 14.6 – Platform Service Redundancy Configuration in FrameWorX64 Server

14.2.3. OPC UA/.NET Servers

You can configure connection to your OPC UA/.NET Servers on a remote computer or assign an alias to that OPC UA server. To do this, open the FrameWorX64 Server, in the Project Explorer expand the "BackEnd Servers" node, and then double-click on "OPC UA/.NET Servers". The OPC UA/.NET Servers tab opens, as shown in Figure 14.7.

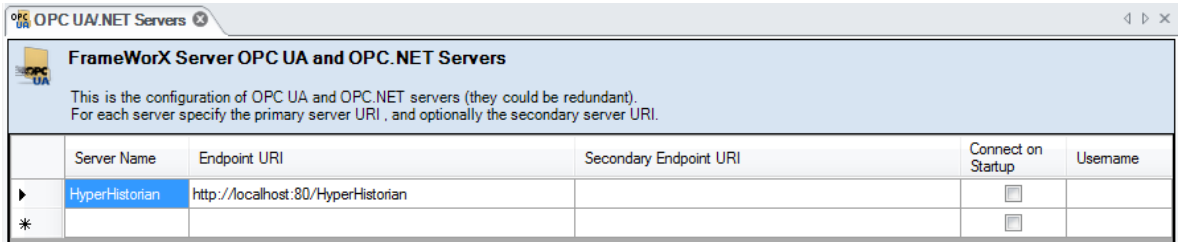


Figure 14.7 - OPC UA Servers Tab in FrameWorX64 Server

On each line, you can enter the following fields:

- **ServerName:** Name of the OPC Server or the alias you wish to give to the OPC Server.
- **Endpoint URI:** the access point for the OPC UA Server.
- **Secondary Endpoint URI:** the secondary access point for the OPC UA Server or a secondary OPC UA server.
- **Connect on Startup:** connects to the OPC UA Server on system startup when checked.

- **Username:** provide authentication credentials to the OPC UA Server. When you click on the "Browse" button in the field, the **Enter User Name and Password** dialog appears as shown in Figure 14.8.

  Fill in the username and password fields to provide credentials to use when connecting to the OPC UA Server.

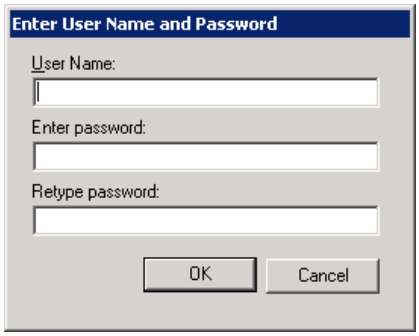


Figure 14.8 - Enter User Name and Password Dialog

Be sure to click on "Apply" to save your changes when you are done configuring your OPC UA/.NET Servers. You should know that if you need to setup OPC UA DA redundancy, this is where you would configure the OPC UA DA node pairs. For information about GENESIS64 Redundancy, please consult the GENESIS64 online Help, the respective application notes, and the Redundancy training course.

14.2.4. GenBroker (Classic OPC)

GenBroker puts GENESIS64 in communication with legacy OPC servers residing on other machines in the same local area network (LAN) or on the internet. To establish communication between legacy OPC Servers and clients that reside on different computers, you must set up and use GenBroker; GenBroker communication is not necessary if you are using OPC-UA servers.

GenBroker is a high-speed communications utility that you can use to designate network communications for a networked set of computers. Communications established using GenBroker are used for all GENESIS64 applications. You also use GenBroker to set up data redundancy for classic OPC servers. Since the topic of Redundancy is out of the scope of this class, we will not cover this tab on the GenBroker Configuration Properties. To find out more about ICONICS redundancy solution, please refer to the help files, appropriate application notes, as well as attend the redundancy training course.

To configure GenBroker, double-click on GenBroker under Classic OPC to open up the GenBroker tab, shown in Figure 14.9 with the **Channels** tab on top.

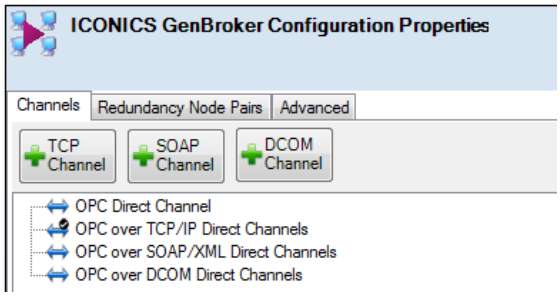


Figure 14.9 - GenBroker Configuration

14.2.4.1. Channels

By default, legacy OPC communication is done over the TCP/IP Direct channel. On the GenBroker **Channels** tab (see Figure 14.9), you can configure a different default channel by right-clicking on the desired channel and then selecting "Set as Default".

A new TCP, SOAP or DCOM channel can be added with a mediator node specified so that legacy OPC communication will be mediated through that node. To configure this, click on the "TCP Channel", "SOAP Channel", or "DCOM Channel" button to add the channel for a mediator node. When you click on any of the buttons to add a channel, the **Select a Node** dialog appears as shown in Figure 14.10. You can choose computer name from the dropdown list, type in the name, or the IP address and click on "OK" to add the node. When you complete the process, you will see a new channel with the specified computer as the mediator in your list of channels.

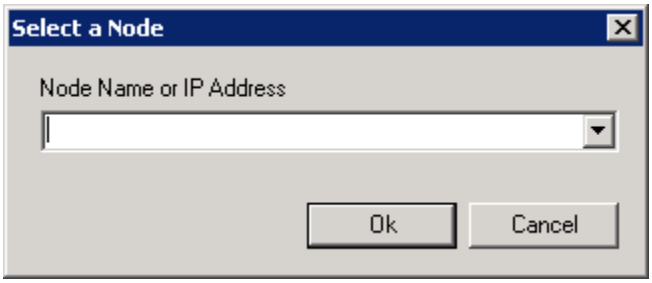


Figure 14.10 - Select a Node

You can add a specific node to use a particular channel so that all communication to that node goes through that channel. To configure this, right-click on the desired channel and then click on "Add Node". The same **Select a Node** dialog appears for you to select or type in the desire computer node name or IP address.

14.2.4.2. Advanced GenBroker Settings

The Advanced tab contains GenBroker settings for the advanced user. Some of these settings could affect system performance, and for that reason should be modified only by an expert user. Settings are shared among five tabs.

The Advanced **Tag Syntax** sub-tab is where tag extension syntax and XML DA syntax is configured. Fields on this tab are used for redefining separators for OPC item names, and are described in detail in the GENESIS64 online Help.

The Advanced **Miscellaneous** sub-tab contains various settings that affect performance of an OPC tag connection. **Client Side Cache** forces GENESIS64 to cache an OPC item for the specified amount of time. **Tag Retry Rate** specifies the rate at which GENESIS64 will retry an OPC connection when a previous connection attempt failed. And **Sync Call Timeout** specifies the time at which GENESIS64 will stop waiting for a response from an OPC connection and will decide that the connection attempt failed and will try to reconnect.

The Advanced **OPC Engine** sub-tab settings for allowing the use of different OPC versions, setting OPC DA group names, and some additional queue and status settings. **OPC Versions** determines which version of the OPC specification GENESIS64 is to use when connecting to a legacy OPC server. This setting may need to be tweaked for connections to legacy OPC servers that were created with a specific version of the OPC specification. **OPC DA group names** define the group name format for an OPC server. And the **Miscellaneous** section contains settings that affect the runtime queue and status.

The Advanced **Optimization** and **Connection** sub-tabs are advanced configurations whose default values have been carefully chosen for the best performance. To understand them better and to tweak them, refer to the GENESIS64 online Help.

14.3. Changing the Default FrameWorX64 Server

Because of its client-server architecture, a GENESIS64 system can get its data from a remote FrameWorX64 server. When a remote FrameWorX64 server is set as the *default* server, all GENESIS64 communication is mediated by the remote FrameWorX64 server. This means that all licensing, security, and data communication come from the remote FrameWorX64 server. Furthermore, all local connections are local to the FrameWorX64 server, not local to the client.

14.3.1. Setting a Remote FrameWorX64 Server

To set a remote FrameWorX64 server as the mediator server, while in the GENESIS64 Workbench, select **File → Tools → FrameWorX Server**. The FrameWorX Server Configuration dialog box appears, as shown in Figure 14.11. This is where you define the FrameWorX64 Server to be used. It is also here that you configure .NET client redundancy, WebHMI client redundancy (if different) and redundancy for silverlight clients (limited options and transport protocols).

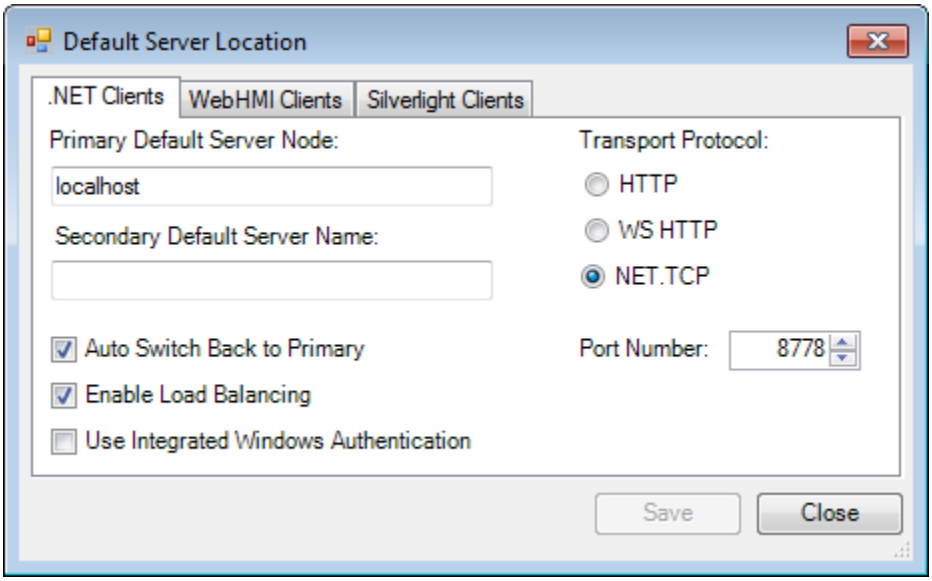


Figure 14.11 - Default FrameWorX Server Location Dialog Box

14.4. Remote Access

Since GENESIS64 uses the latest .NET technologies and comes with Web-enabled, the web clients need a way to access data remotely. You could also have GenBrowser station that needs to access date remotely as well. The applications that provide data access remotely are GenTray and FrameWorX64 Server and the clients need to access them to retrieve data.

There are two ways to access GenTray or FrameWorX64 server remotely. You can use the Workbench from an existing GENESIS64 system, or configure it using Internet Explorer. Each method is described here and both of them will allow you to manage a remote machine's configuration and services.

14.4.1. Using a Remote FrameWorX64 Server

If you have GenBrowser stations, you may need to set your FrameWorX64 Server to a remote computer that houses your license information and OPC Servers. In this case, since GENESIS64 are installed on the browser or client stations, you can use Workbench to configure the remote FrameWorX64 server.

To access GenTray or FrameWorX64 server remotely using the Workbench, a connection to the remote GenTray or FrameWorX64 server web service first needs to be made from the Workbench.

To add a remote Workbench server connection, in the Workbench, go to the Project tab of the ribbon and click on the "Connect to Server" button, shown in Figure 14.12.

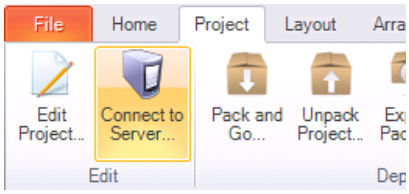


Figure 14.12 - Connect to a Server

In the **Connect To Server** dialog that opens, click on the "Servers" button. This will open the **Add/Remove Server** dialog. Click on the "Add" button in this dialog to open the **Server** dialog. Fill in the server name or IP address, display name, and the port to communicate on, as shown in Figure 14.13.

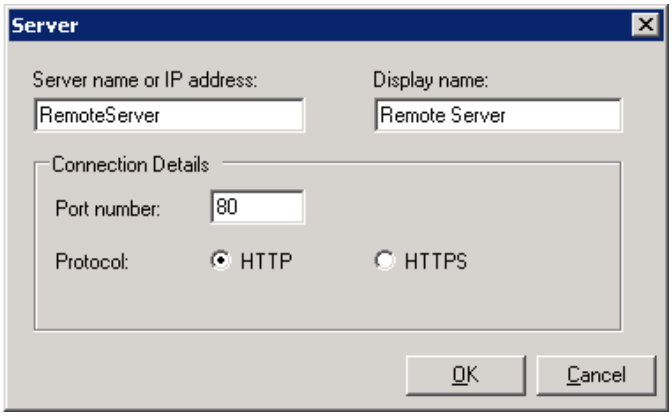


Figure 14.13 - Connecting to a Remote Server

Click on "OK" to return to the **Add/Remove** Server dialog. There, you will see the new server added to the servers list, as shown in Figure 14.14.

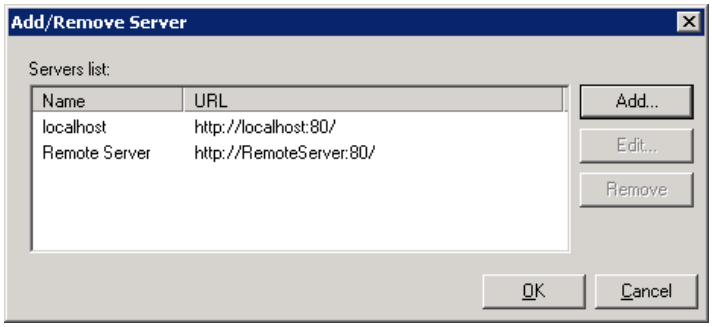


Figure 14.14 - Server List

Close the **Add/Remove Server** dialog by clicking on "OK". In the **Connect To Server** dialog, put a checkmark next to the server you wish to use, as shown in Figure 14.15.

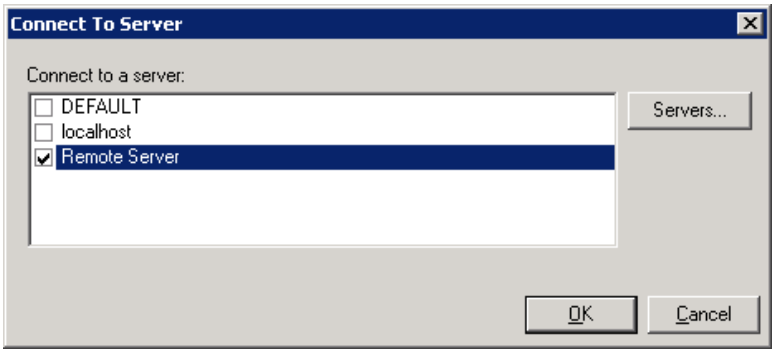


Figure 14.15 - Changing Project Server

Click on "OK". The next time that the FrameWorX64 Configuration is opened, it will connect to the remote server.

14.4.2. Using Internet Explorer

Accessing GenTray or FrameWorX64 server over Internet Explorer is possible because GENESIS64 is built using .NET and communication is based on web services. Internet Explorer client can either be 32-bit or 64-bit.

14.4.2.1. .NET Framework Installation and Security Settings on a Client

During installation, GENESIS64 adds some websites to IIS and sets all necessary security settings on the server to allow configuration over Internet Explorer on a client machine. In order to use Internet Explorer on a client to access your server, you must install the latest .NET Framework on the client computer. Once you install it, you will also need to configure .NET Framework permissions to allow remote access.

The latest version of .NET Framework that we support as of the date of this document is version 3.5 SP2. You can download and install it from Microsoft by going to: http://www.microsoft.com/downloads/en/details.aspx?familyid=5b2c0358-915b-4eb5-9b1d-10e506da9d0f&displaylang=en.

After .NET Framework is installed, you should open the command prompt by going to **Start → Run**. In the dialog box that opens, type in "cmd" (without quotes) and press the "Enter" key.

In the command window on your screen, change the directory to C:\Windows\Microsoft.NET\Framework\v2.0.50727 using the "cd" command. Once you are in the directory, execute the command: **caspol –cg Trusted_Zone FullTrust**. You will need to confirm the operation and it should return a success message. Figure 14.15 shows the command process in the command prompt.

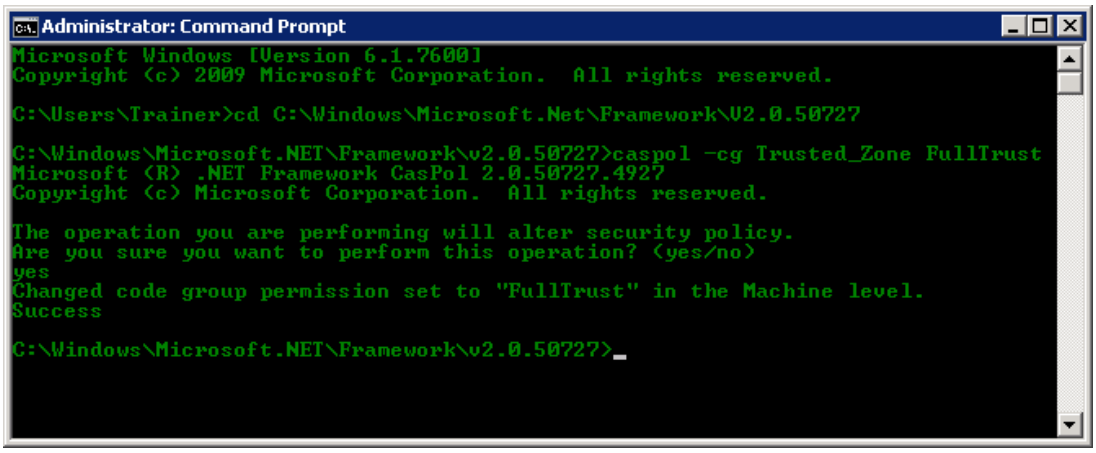


Figure 14.16 - Command Prompt

14.4.2.2. Setting Internet Explorer Security Settings on the Client

The website that provides remote configuration of a GENESIS64 system must be added to the Trusted Sites list.

Edit the Internet Explorer properties on the client computer by opening an instance of Internet Explorer and clicking on Tools → Internet Options. In the **Internet Options** dialog, select the Security tab. In the "Select a zone to view or change security settings" section, click on "Trusted sites". Click on the "Sites" button below the section to launch the **Trusted sites** dialog. Type in the http address of your GENESIS64 server machine and click on the "Add" button.

For example, type in "http://*GENESIS64Server*" where *GENESIS64Server* is the name of the machine that has GENESIS64 installed. Only the server name needs to be added, meaning that any Web site from *GENESIS64Server* is trusted.

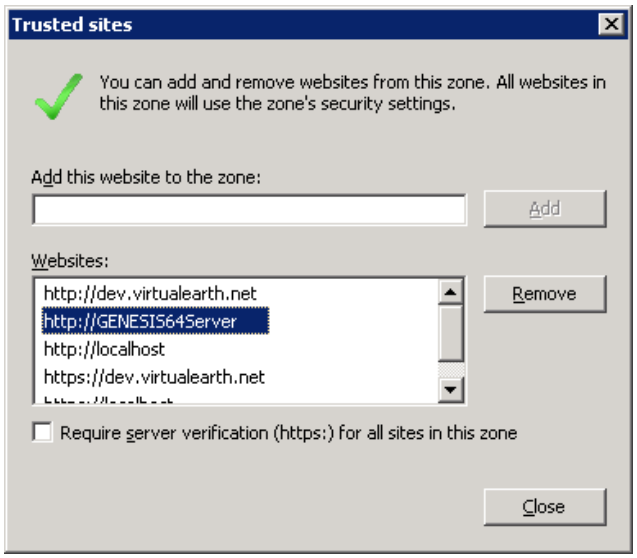


Figure 14.17- Adding a Trusted Site

Note: You may need to uncheck the "Require server verification (https for all sites in this zone" option at the bottom to add the site.

Once you have added all of the servers, click on “Close” to close the dialog. In the **Internet Options** dialog, lower the security level for the trusted sites to “Low” and click on “OK” to save the changes, and then close the dialog box.

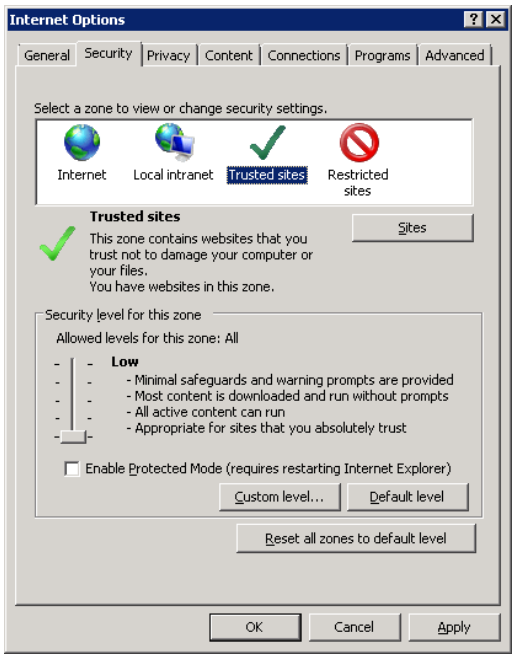


Figure 14.18 - Internet Options: Security for Trusted Sites

14.4.2.3. Navigate to Workbench from Internet Explorer

Once you have installed .NET Framework and configured Internet Explorer security, you are ready to connect to your GENESIS64 Server. However, first, we will need to make sure we can connect to the Web Server first.

You can check the connection to the web server by launching Internet Explorer and typing in the URL "http://*GENESIS64Server*" where *GENESIS64Server* is the name of the computer where you have installed GENESIS64. If connection to the web server is successful, you should see a page similar to the one shown in Figure 14.19.

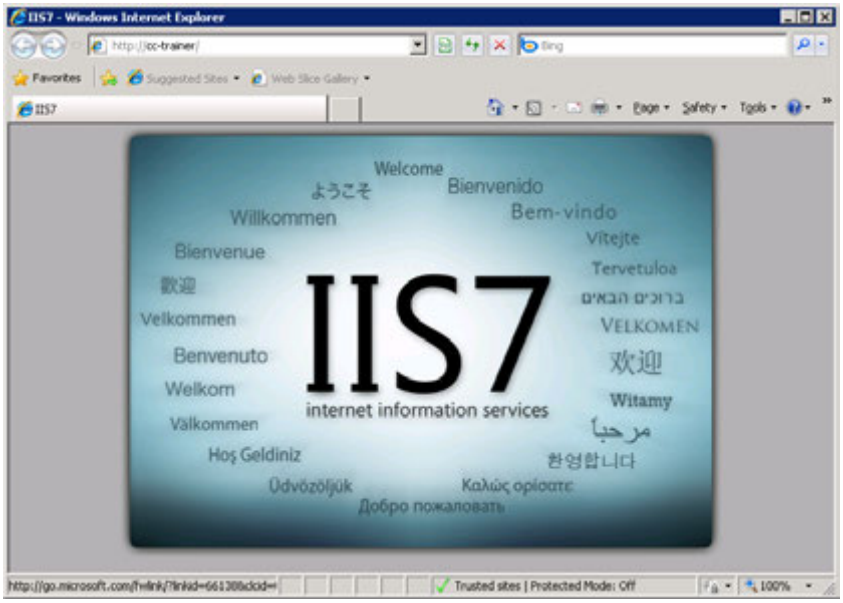


Figure 14.19 - Web Server Welcome Page

If you do not see this page, there are other problems with the Web Server and you should take care of that before continuing. If you do see the page, you can continue to try accessing the Workbench. The URL to the Workbench is http://*GENSIS64Server*/GENESIS64/WebSites/PubDisplay/Workbench.htm, where *GENESIS64Server* is the name of the machine where GENESIS64 is installed. The first time you access this page, it may take a few minutes for everything to load, subsequent access should be faster.

Once the page loads, you should see a page similar to the one shown in the following Figure.

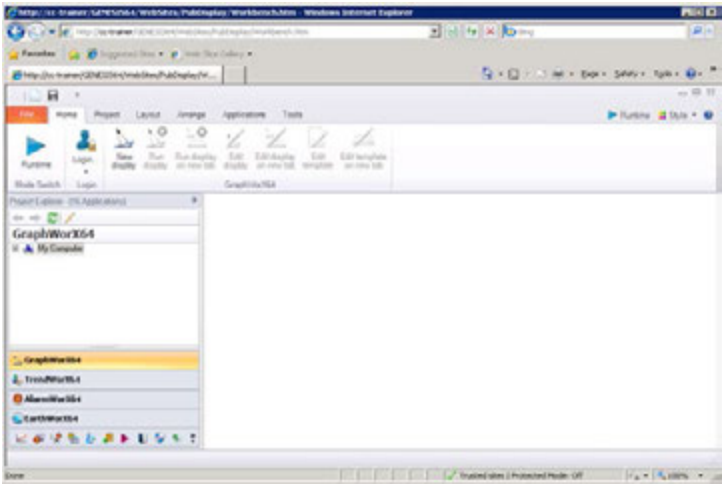


Figure 14.20 - Workbench64 in Internet Explorer

14.5. Web Publishing

In GraphWorX64, you will find a Web Publishing Wizard that enables you to convert your GraphWorX64 display files to HTML files and publish the HTML files to a web server (LAN or Internet). For displays created using MS Silverlight edition, the wizard publishes a Web page for Silverlight.

In publishing displays to a web server, the Web Publishing Wizard uses HTML to reference the files in an Internet-enabled format. Once a display is converted to an HTML file and then published to a Web server, client machines can browse it using Microsoft's Internet Explorer. The Web Publishing Wizard is accessed from the File menu's "Publish" button, shown in Figure 14.21.

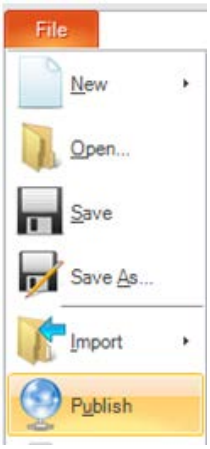


Figure 14.21 - Publishing a Display

After a display is published, it can be viewed as a web page. A GraphWorX64 display that is published to a Web server is configurable by a user who downloads the display from a remote client machine. The Web Publishing Wizard allows you to determine the level of configuration in the published display file that will be available to client-side users.

When you open the **Web Publishing Wizard**, you can set up your Publishing and Application Preferences; both are shown in Figure 14.22.

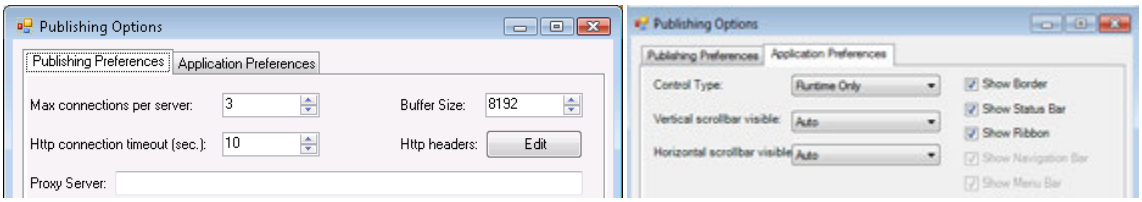


Figure 14.22 – Publishing (Left) and Application (Right) Preferences

The **Web Publishing Wizard** operates on the Zero Install and Thin Client philosophy. This means the client computer needs nothing but Windows and Internet Explorer installed. For this reason, all necessary web components must be remotely delivered and seamlessly installed. Therefore, it is critical that the HTML page be built in accordance with this. The Web Publishing Wizard looks inside the display to understand what components are necessary on the client-side and then builds the corresponding HTML code.

A GraphWorX64 display that is linked to other display files can include several .NET Windows forms or controls in the work areas of each dependent display file. Publishing such a file involves creating a single HTML file that triggers simultaneous download and installation of all .NET Controls embedded in all interlinked display files so that all displays work correctly. The Web Publishing Wizard's embedded .NET Control support makes this possible, allowing you to view the .NET Controls in all displays from a single client Web browser.

Many GraphWorX64 files are linked to other display files through pick actions such as a Load Display or Pop-up Window, display tabs, or display buttons. You would want all these links and references to work when the HTML file is downloaded to a client Web browser. The Web Publishing Wizard makes this possible by detecting all mutually linked displays, looping through all dynamic actions, and checking for pick actions in which a file name is specified as one of its parameters.

Putting a checkmark in the "Multiple Display Support" checkbox, shown in Figure 14.23, ensures that the web page will publish properly for the selected display files.

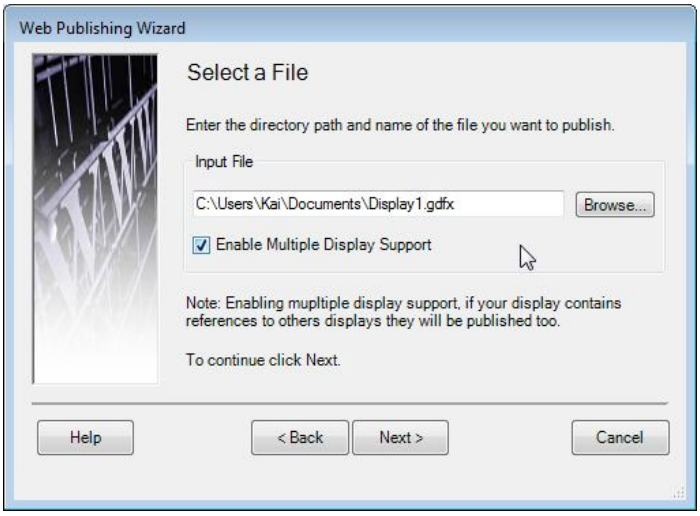


Figure 14.23 - Support for Multiple Displays

Next, the wizard presents a list of displays that are interlinked. Select the ones you want to include or exclude from the publishing process.

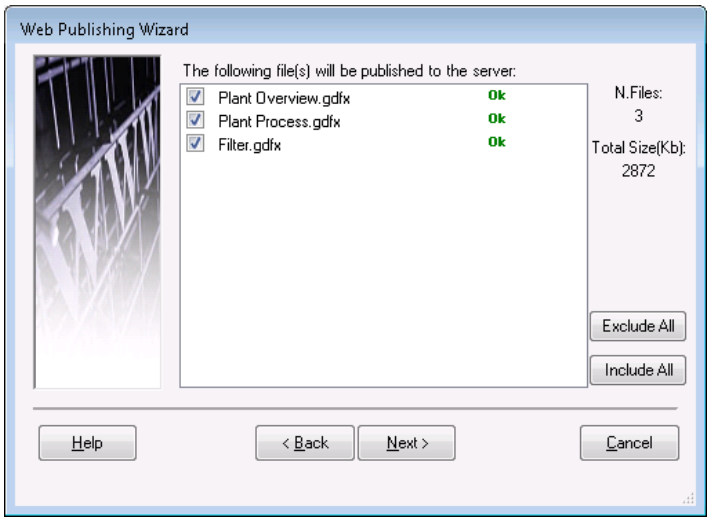


Figure 14.24 - Select Displays to Publish

The next steps of the wizard are dedicated to the Web server connection setting and bring the publishing procedure to an end. Once you have successfully published the display, make sure that GenBroker is running on your server machine for the client machines to get data.

Lab: Remote Configuration L14

14.1. Lab Overview

In today's globalized business environment, companies need to provide a window into production information that transcends geographical boundaries. It is extremely important for icTonics' personnel to be able to view and control the production from a remote location because their personnel travel frequently. They would also like to allow the managers to view and control the plant at home in case emergencies arise late at night. We will provide them with this ability by publishing our displays and allowing the displays to work on remote computers.

Estimated time to complete this lab: 20 minutes

14.1.1. Task Checklist

In this lab, you will complete the following tasks:

- Publish displays
- View displays in Internet Explorer

14.1.2. Support Files

To complete this lab, you will need the following file(s):

- icTonicsHome.gdfx
- icTonicsTankMonitor.gdfx
- icTonicsTrends.gdfx

14.2. Viewing Displays in Internet Explorer

In order to view displays remotely over the internet, we will need to publish the displays to a web server first. In this section, we will publish a display and then try to view it on the local training computer.

- **Test Website Functionality**

1. Before we start publishing the display, let us go a few steps further to verify that your machine is ready for publishing.

2. First, determine your hostname by going to Start → Command Prompt to launch a Windows command prompt. (Alternatively, you could go, Start → Run, and then type in "cmd" (without quotes) and hit enter.)

3. Type hostname and the enter key. The hostname below is HV-02. Leave the command prompt open for future testing.

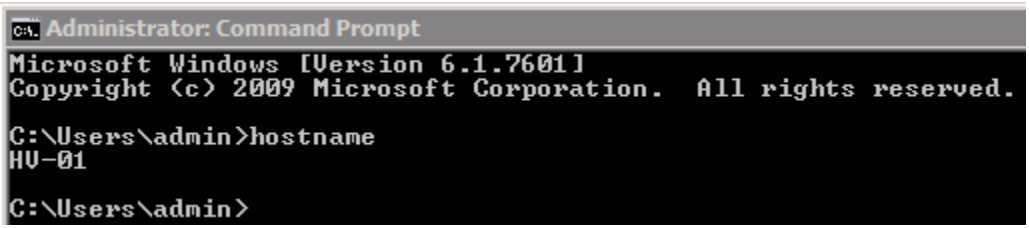


Figure L14.1 - Windows Command Prompt and Hostname Command

4. Launch the Internet Explorer and type http://hostname into the address field. If you see the following picture after you launch the URL, it means your Internet Information Services (IIS) is working. Close the Internet Explorer.



Figure L14.2 - IIS

- **Add FTP Publishing to the Default Web Site**

1. From Start → Administrative Tools → Internet Information Services (IIS), open the IIS Manager. If there are two selections open in the IIS manager, choose the one WITHOUT the 6.0 in its name.

2. Expand the Connections navigation tree on the left and then right-click on the "Default Web Site".

3. Select "**Add FTP Publishing**".

4. Select '**No SSL**" and click on "Next". Select **Anonymous** and **Basic**. From the Allow Access to: dropdown list select **All Users.** Select **Read** and **Write.** Then click on "Finish".

   NOTE: For testing purposes we have opened up the security. You may consider different settings in a production facility.

5. Under the "Manage Web Site" section on the right, click on "Restart".

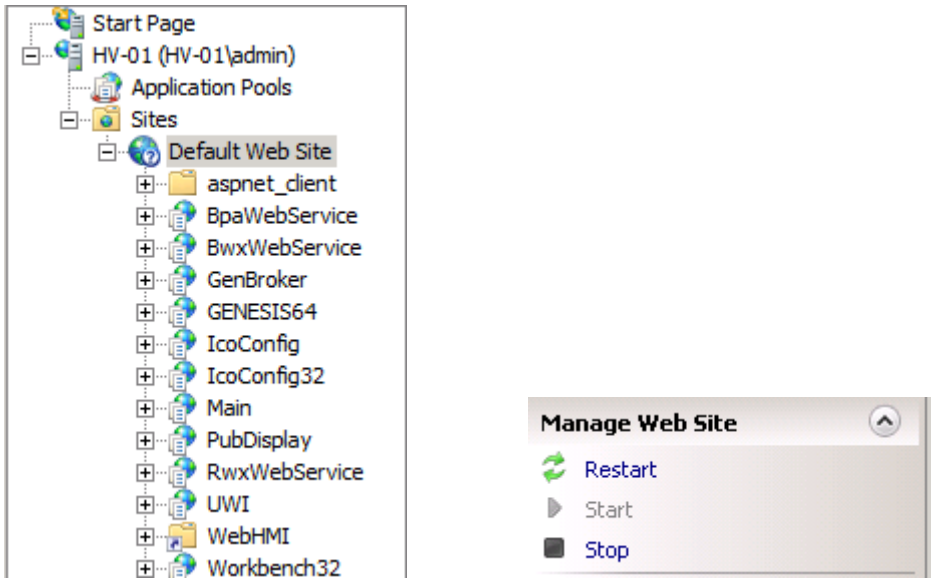


Figure L14.3 – Manage Web Site - Restart

- ## Test Ftp Connectivity

1. From the Command Prompt type in **ftp** *hostname* and press the enter key. A user login prompt should be displayed. Login with the user account that you use to login to your OS. If the command prompt returns this message "**230 User logged in**" after you enter your username and password it means your ftp connectivity is working. If any of the above steps do not work properly, ask our instructor for assistance. Close the Command Prompt window.

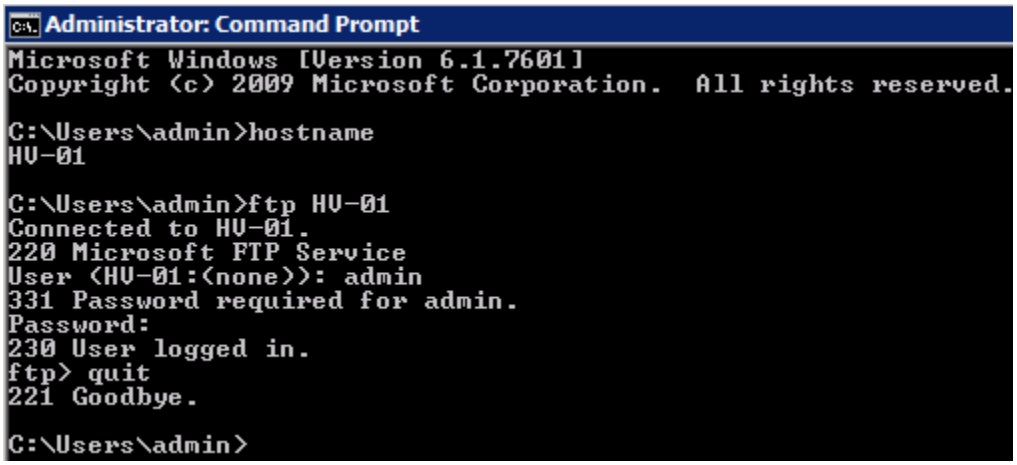


Figure L14.4 - Ftp Connectivity Test

- ## Publish a Display

1. Open Workbench if you closed it from the previous lab, then open the icTonicsHome display.

2. From the File menu, select "**Publish**".

3. In the **Web Publishing Wizard** that opens, click on "Next".

   In the "**Select a File"** dialog box, enter or browse for the "Input File". Because the display is already open in the Workbench, the icTonicsHome.gdfx display should appear as the "Input File". If it does not, click on the "Browse" button to navigate to it and select it.

4. Make sure there is a checkmark in the "**Enable Multiple Display Support**" checkbox and click on "Next".

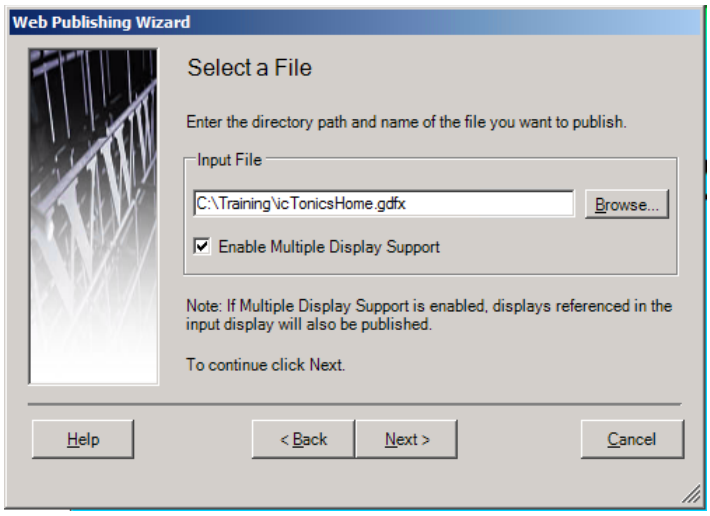


Figure L14.5 - Web Publishing Wizard: Select a File

5. You will now see a progress window. Notice that although we have only selected to publish the icTonicsHome.gdfx display, it published the icTonicsTankMonitor.gdfx and icTonicsTrends.gdfx displays, too. This is because we checked the "Enable Multiple Display Support" checkbox. This allows you to publish displays that have links (such as our Load Display pick action) to the files being published.

6. Click on "Next" to select the web server to publish the three files.

7. In the **Select a Publishing Location** dialog box, click on "New…" to create a new web server.

8. In the **Web Server Configuration** dialog box that opens, change the Protocol to "ftp" if it is not already selected and type in your computer name (hostname) in the "Host Name or IP Address" section.

9. In the Virtual Directory section, point it to "/PubDisplay"; the PubDisplay FTP site was setup during the GENESIS64 installation.

10. Put a checkmark in the "**Save This Web Server with a Friendly Name**" option and type in "icTonics Web Server" in the field below it. The next time we publish the displays, we can just choose this one. Your finished configuration should look similar to Figure L14.6.

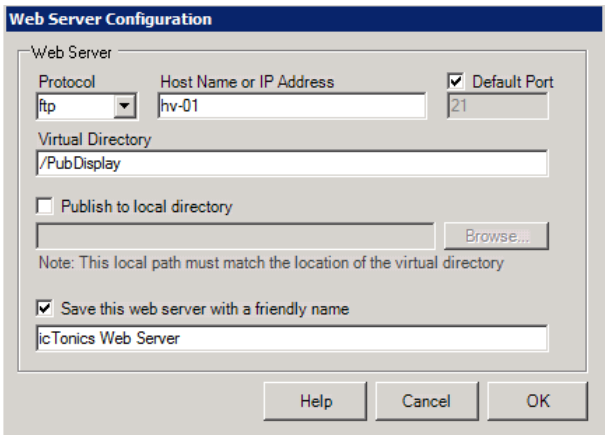


Figure L14.6 - Web Server Configuration

11. Click on "OK" to finish the server configuration and return to the "**Select a Publishing Location"** dialog box. Click on "Next" to proceed.

12. In the **User Name and Password** dialog box, enter a local administrator user and its password, and then click on the "**Publish**" button.

13. It may take awhile to publish, but once it finishes you should see a dialog box indicating the status of the publishing as shown in Figure L14.7.

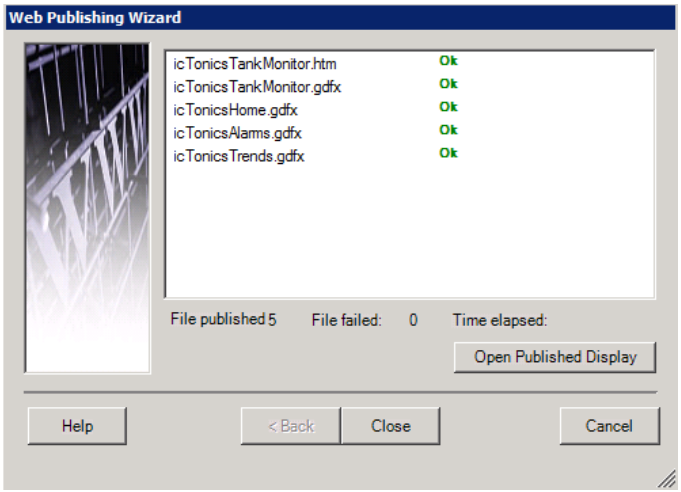


Figure L14.7 - Publishing Wizard

Note: If the publishing results give any warning or error, you can place your mouse over the warning or error and the tooltip will give you more details.

14. You can view your displays in Internet Explorer by going to http://*HostName*/PubDisplay/icTonicsHome.htm. Feel free to navigate it as you would a regular display.

14.3. Remote Workbench

In some cases, engineers may need to travel to different places, leaving the environment where they created the displays. In such cases the engineers can use the remote configuration capability to continue their work from a remote location. They can even use a 32-bit machine. This section will walk you through the necessary setup to configure a project on a remote 32-bit machine.

1. Check that you have .NET Framework 4.0 installed on the client machine. You can do this by going to the **Add/Remove Programs** (or Programs and Features) in the Control Panel. Depending on the OS, it could be under the installed program list or in the Windows Features list.

2. Launch an Internet Explorer session. Under Tools → Internet Options → Security → Trusted Sites add your ServerName to the list of trusted sites.

3. If you are using a 64-bit machine, you can go to http://*ServerName*/GENESIS64/websites/PubDisplay/Workbench.htm (where *ServerName* is the hostname of your server machine) to bring up the workbench and begin configuration.

Review: GenTray and FrameWorX64

R14

14.1 Objectives Review

You should now know and understand the following topics:

- Using GenTray to control local and remote services
- Configuring ICONICS FrameWorX64 server
- Setting up redundancy and monitoring redundant servers

14.2 Summary

ICONICS GenTray is a tool that can be used to control all ICONICS and non-ICONICS Windows services. Any Windows service can be started and stopped from GenTray. A service's startup type can also be modified from GenTray. GenTray is accessed from the Workbench.

FrameworX64 server is a central mediating server for all GENESIS64 communications. Configuration of the FrameWorX64 server is done using the Workbench. By default, the communication channel is TCP/IP, but this can be changed to SOAP or OPC Direct. The default FrameWorX64 server can be changed using the FrameWorX64 server location setup tool.

Furthermore, GENESIS64 supports redundancy to both 64-bit servers and 32-bit Servers. This means that in GENESIS64 you can set up redundant servers for: data redundancy, alarm redundancy, and historical redundancy using Hyper Historian. You can also configure redundancy for clients, licenses, and security.

Both GenTray and FrameWorX64 server can be accessed remotely through Workbench or Internet Explorer. A remote web service connection can be configured in the Workbench to access a remote GENESIS64 system. To access the Workbench remotely using Internet Explorer, .NET Framework 3.0 needs to be installed.

14.3 Terms

FrameWorX64 Server	GenBroker	GenEvent
GenTray	Node Pairs	Redundancy

14.4 Test Your Knowledge

In this chapter, you learned about GenTray and FrameWorX64 Server. Before you continue, make sure you can answer the following questions.

1. Which Windows services can ICONICS GenTray control?

   a. All Windows services
   b. Services that are running
   c. None

2. True or False: A service startup type can be changed from GenTray?

3. True or False: A service's logon user can be changed from GenTray?

4. How can you access GenTray remotely?

   a. Using Internet Explorer
   b. Using Workbench
   c. Both of the above
   d. None of the above

5. What communication does the ICONICS FrameWorX64 server mediate?

   a. All communication
   b. Local only
   c. Remote only

6. What is the default communication channel for ICONICS FrameWorX64 Server?

   a. TCP/IP
   b. UDP
   c. DCOM
   d. SOAP/SML

7. What tool should be used to change the default ICONICS FrameWorX64 Server?

   a. The Workbench
   b. FrameWorX64 Enterprise Workbench
   c. A command window
   d. FrameWorX64 Server Location setup tool

8. True or False: You do NOT need .NET framework to be installed to access Workbench using Internet Explorer.

9. Which version of .NET framework should be installed?

   a. .NET Framework 4.0
   b. .NET Framework 3.5
   c. .NET framework 3.0
   d. .NET framework 2.0
   e. Any version

10. Does Internet Explorer need to be 64-bit to access the GENESIS64 Workbench?

    a. Yes, a 64-bit Internet Explorer is required
    b. No, a 32-bit Internet Explorer will do
    c. It depends on your license

11. Classic OPC redundancy works with which of the following OPC servers?

    a. OPC DA Servers
    b. AlarmWorX32 Server
    c. TrendWorX32 HDA
    d. All of the above

12. What GENESIS64 tool do you use to monitor server health and to switch from one server to another?

    a. MonitorWorX
    b. Server Viewer
    c. Redundancy Viewer
    d. Node Pair Viewer


ICONICS

GridWorX64 15

This chapter provides end-users with a basic understanding of GridWorX64. For more detailed information, procedures, etc., please refer to the main GENESIS64 Help file included with the product installation.

GridWorX64 allows GENESIS64 clients to natively access database information through the FrameWorX Server. GridWorX64 can visualize database data in the form of OPC-DA data or through the GridWorX64 Viewer Control in GraphWorX64. You can create basic read and write actions along with stored procedures against the connected databases. The functionality is similar to how the Data Mining OPC Server works in GENESIS32.

GridWorX64 consists of a GridWorX64 Server Configurator, GridWorX64 Runtime and a GridWorX64 Viewer. The GridWorX64 Server Configurator stores information in a database that is queried by the GridWorX64 Runtime Server as during operation. GridWorX64 Viewer is a client and allows you to see data from different databases.

If you have experience with GENESIS32 you can compare the following products:

- GridWorX64 Server Configurator – v9 Data Mining Configurator
- GridWorX64 Runtime – v9 DBOPC Server
- GridWorX64 Viewer – v9 DB Mining ActiveX

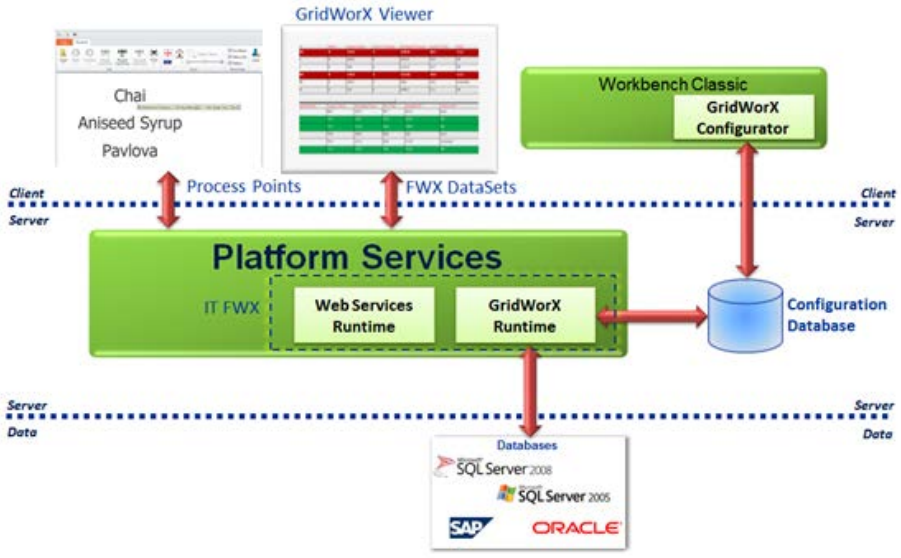


Figure 15.1 – GridWorX64 Architecture

Using GridWorX64 you can access all the database sources listed below:

- MS SQL Server
- Oracle
- SAP
- 64-bit OLEDB
- 64-bit ODBC

Since there are no 32-bit OLEDB providers available in GridWorX64, MS Access and 32 bit MS Excel are not supported. However, 64-bit is supported.

15.1. GridWorX64 Server Configurator

GridWorX64 Server configurator is integrated inside the Workbench and provides a configuration tool for connecting to various databases. Through the Workbench you also have the capability of launching the Advanced Query Wizard for easier configuration of database connections.

To open GridWorX64 Server Configurator, go to Windows Start Menu →All Programs → ICONICS →Workbench. Expand the provider list in the explorer tree on the left and click on the GridWorX64 Server icon to open the provider. Expand the GridWorX64 Server localhost branch on the left. You should see your GridWorX64 Server as shown in Figure 15.2.

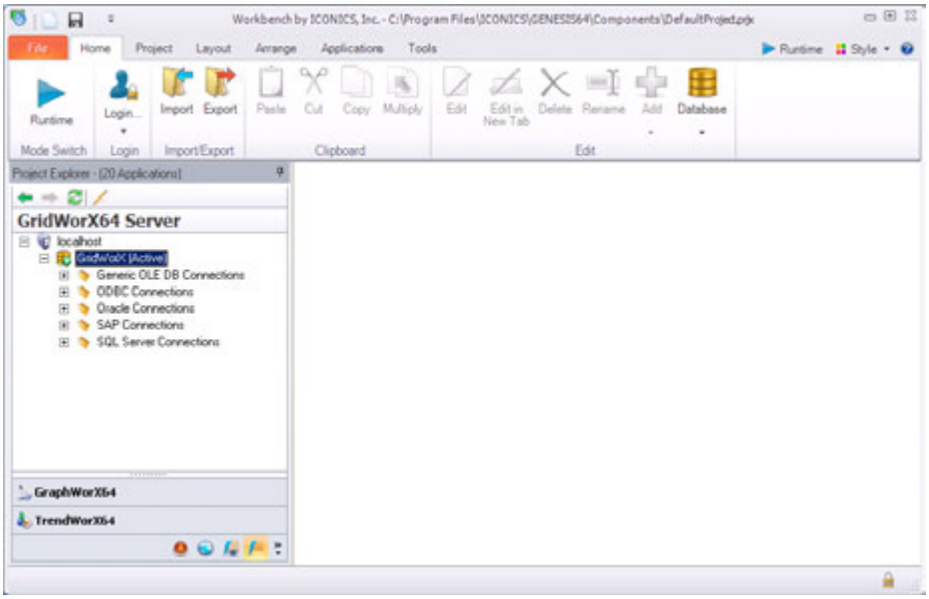


Figure 15.2 – GridWorX64 Server

Expand the SQL Server Connections → Northwind. Inside the configuration, you can find four types of GridWorX64 objects:

- Database Connections
  - Data Sources
    - Data Items
  - Data Manipulators

Data Items can be created when you specify any parameter for Data Source. We will describe each data type detailed in following chapters.

15.1.1. Connections

Database Connection is a top level object. It specifies the connection string to a target database. You can create as many database connections as you want. Multiple connections to the same database can be created, but they must all have a unique name. It's worth noting that if a functional connection fails during runtime, then there is a mechanism that automatically reestablishes the connection.

To Connect to a SQL Server Database in GridWorX64:

To create a new Database Connection, open your Workbench GridWorX64 Server provider and right-click on the connection type you wish to create. We will the select SQL Server Connections type and the Northwind database will be used. For the remainder of this document, the SQL Server Connection type and the Northwind database will be used. Select "New Database Connection" from the dropdown list. You can alternatively select the Add button from the Ribbon Menu, as is shown in the figure below.

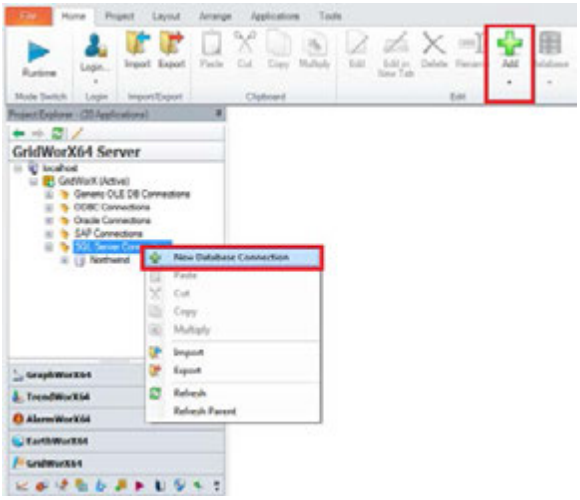


Figure 15.3 – Adding New Connection

To build a Connection String, click on the Change Connection button and select one of your existing databases from the **Connection Strings** dialog box (the Northwind test database was installed during the GENESI64 installation). In the "Connection Strings" dialog, double click on "SQL Server Connections". Click on the "Add Connection String" button to open up the "SQL Connection String" dialog box.

Enter your SQL Server name [for example, (localhost\sqlexpress)] and the proper login information. If done correctly, you will be able to retrieve a list of databases from the "Select the database on the server" dropdown list. If you choose the Northwind database, your finished connection string should look similar to the figure below. Be sure to click 'Apply' to save your changes.

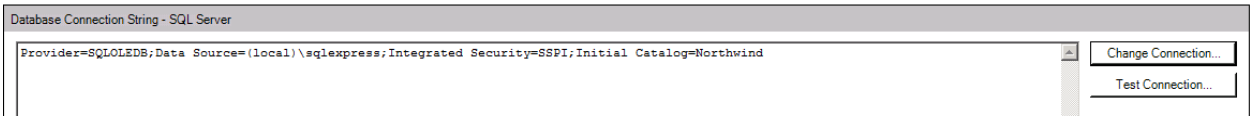


Figure 15.4 – Connection Strings

Verify you have a successful connection by clicking on the "Test Connection" button.

15.1.2. Creating a Data Source (Simple SQL Designer)

A Data Source is a handle provided for the Unified Data Browser that is used to connect to a database table. The Data Source stores SQL commands to select, insert, update, and delete entries in the database table. SELECT configuration is required. Insert, Update, and Delete configurations are optional. Follow these instructions to create a Data Source.

From the GridWorX64 Server provider, right click on the database connection and select 'New Data Source' from the drop down list. You have ability to specify different kinds of commands. However, the SELECT command is the minimum required to configure a data source.

A Data Source is used to retrieve data from a database. Data is queried from the database and then cached in the server's memory to increase performance. This means that when one or more client request data for a particular Data Source, they will receive the cached data (if present in memory) instead of the live data from the database. This behavior can be configured using three different options:

- **No refresh** - Data will be cached in the server's memory and kept indefinitely, unless manually refreshed from a client (see @@Refresh on next page).

- **Periodic refresh** - Data will be cached in the server's memory for the specified amount of time, after which it will be discarded. A subsequent request will cause a new query to the database to fetch the data.

- **UDM Trigger** - Data will be cached in the server's memory and it will be discarded when the specified UDM Trigger is fired. A subsequent request will cause a new query to the database to fetch the data

Clients can also force the server to discard cached data if needed. Data Sources expose a special DA Point (@@Refresh) that can be written to inform the server to discard its cached data for the Data Source. This will force the server to query the database again to fetch the data on the next request. You should consider the effect on resource usage when specifying a refresh rate. Refreshing can consume a lot of memory and CPU cycles. Because the same resources are shared by other providers and clients, such as the alarm server, take caution to consider the true importance of frequently retrieving database data. That is why you can set server or client refresh rates separately.

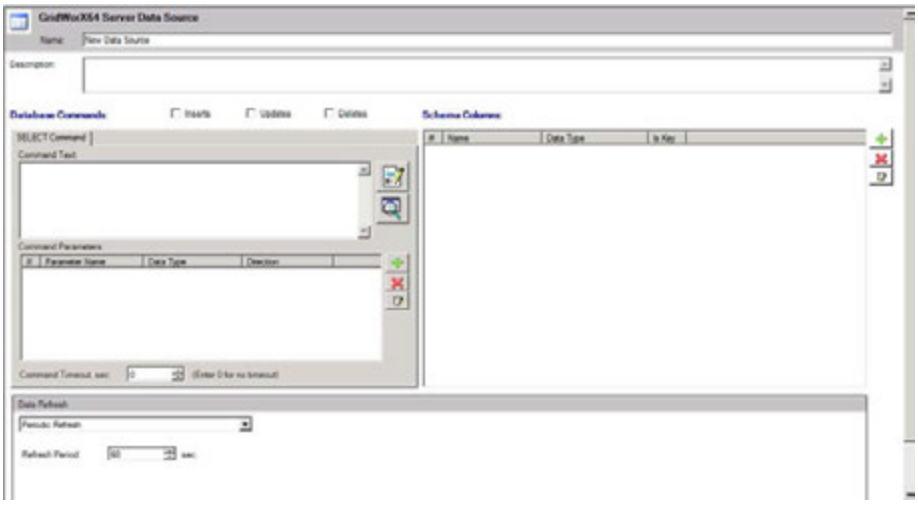


Figure 15.5 – New Data Source Dialog

There are four commands available with the GridWorX64 Server Data Source:

- **Select** – the select statement is required by every Data Source. Primary Key for this command is not required however if you intend to auto-generate updates, inserts or deletes the primary key is required. Generally, selecting a primary key is a good idea.

- **Insert** – for inserting items into your connected database. When this is configured, GridWorX64 Viewer in runtime allows writing. Primary Key for this command is required.

- **Delete** – for deleting items from your database. Primary Key for this command is required.
- **Update** – for updating items in your database. Primary Key for this command is required.

Upon selecting the checkbox to include other available commands, every additional command will add a tab to the Database Commands configuration dialog. Selecting the individual tabs allows the user to configure the selected command parameters. Within each tab, some additional configuration is required. However, most checkbox selections are optional.

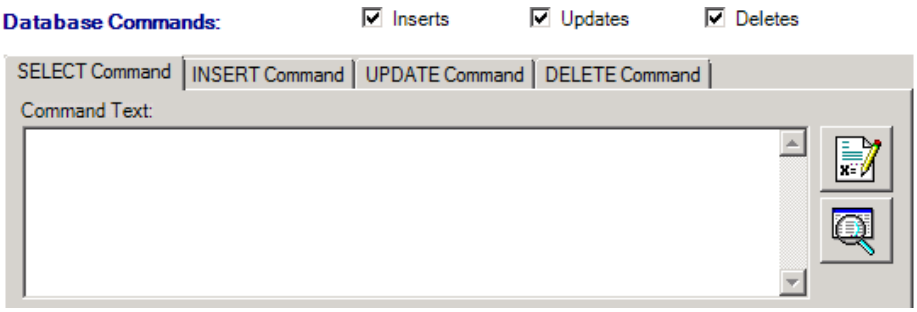


Figure 15.6 - All Database Commands Selected

There are four command text editors available: Simple SQL Designer, Advanced SQL Designer, Stored Procedure Call and Manual SQL Editor. For the scope of this class, we'll focus on the Simple SQL Designer. Please refer to the GENESIS64 Help documentation for information on the additional text editors.

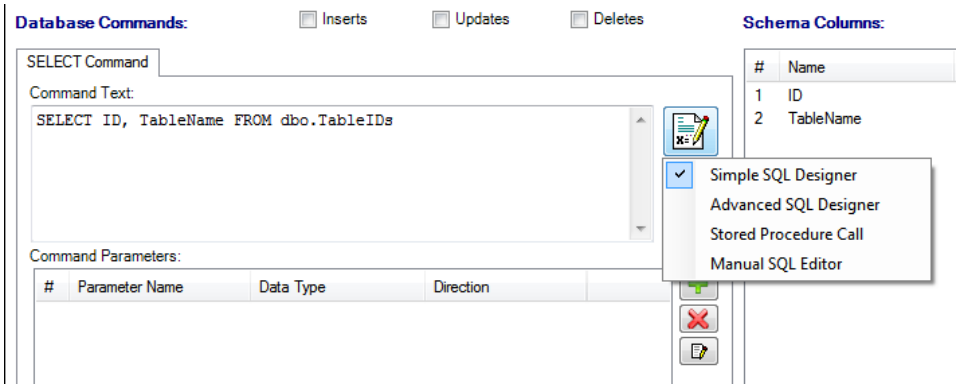


Figure 15.7 – Simple SQL Designer Dialog

Inside the Simple SQL Designer, from the Select Table dialog, clicking on the Table Name ellipse button allows you to browse for the table you would like to work with. After specifying a table and clicking on "OK", through a checkbox selection interface you can choose table columns.

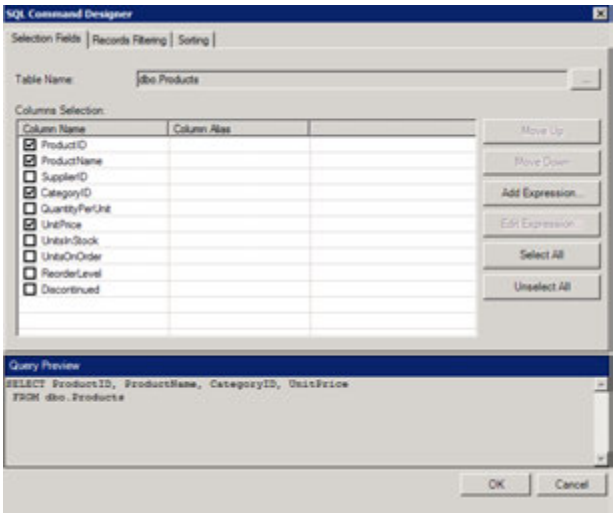


Figure 15.8 - Selecting Columns from the Products Table

From the SQL Command Designer dialog, you can Add Expressions to perform calculations on the data retrieved from the table columns. The Expressions that are available and their usage are explained in the system help file.

There are two more tabs (Records Filtering and Sorting) within the Simple SQL Designer dialog.

Records Filtering – Selecting this tab and clicking on the green plus sign launches the Filtering Criteria dialog box F, as seen in the figure below.

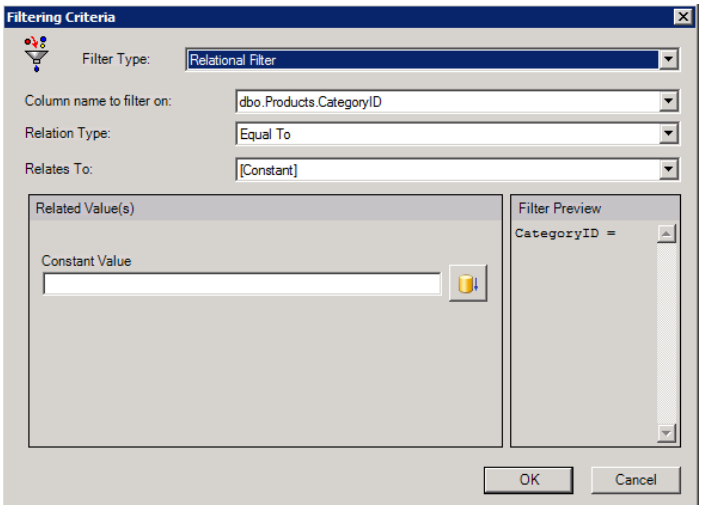


Figure 15.9 - Data Source Filter

GridWorX64 supports two Filter Types, Relational and Expression. Within the **Relational** filter, most all of the fields requiring configuration are selectable through mouse clicks however the Related Values criteria will vary depending on the Relates To value selection. It's worth noting, by utilizing parameters you are able to change values in runtime. On the other hand, constants are static so your query returns the data based on the constant value you defined during configuration.

Expression Filtering - is outside the scope of this class, as it requires an elevated knowledge of both SQL scripting and the database schema.

Sorting - The last tab, sorting, will sort the records that are returned by the query based on the sorting definition during this configuration. There are four levels of sorting and each characteristic has an ascending and descending selection.

Data Source Testing – When the data source configuration is complete and you have returned the Database Command dialog box, click on the magnifying glass icon  to execute the query you have created.

Data Items –are considered children of Data Source items and are used to define a list of preset parameter values. This gives you a way to assign a user-friendly name to customized data sources and use them without having to specify the parameter values. Setting of parameters on the data source is required in order for the data item selection to be available and selectable in the data item selection drop down list.

Creating a Data Item – after creating a data source with at least one parameter, you are able to right click on Data Source and select New Data Item from drop down list.

To Create a Data Item:

1. Right-click on the Data Source and select a new data item. You may want to change the name of this data item. In the figure below, it is changed to CategoryID.
2. Assign values in the Parameter Value field for all parameters.

Creating a Data Item



Figure 15.10 - GridWorX64 Data Item Dialog Box

3. Click **Apply**. Click the Test Data Source icon  to see your query results.

Data Manipulators - are objects in GridWorX64 that can manipulate database data and structure using stored procedures or simple SQL statements when your database server or engine does not support stored procedures. Data manipulation actions are typically executed manually by writing to the Data Manipulator's @@Execute DA process.

To Create a Data Manipulator:

1. Right-click on your database connection and select New Data Manipulator to open the GridWorX64 data manipulator screen.
2. Give a descriptive name to the data manipulator. For example, you might want to create a Data Manipulator which you can set to delete records from the Products Table.
3. Click the ellipsis button and chose Browse Table for Delete to open the Select Table dialog box.



Figure 15.11 – Creating a Data Manipulator

4. Select the table to be affected by this Data Manipulator. For example, to delete a record from the Products table, select Products in this dialog box.

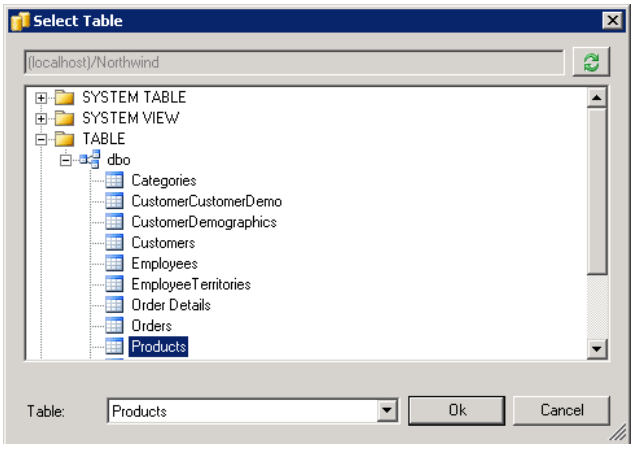


Figure 15.12 – Select Table Screen

5. Click on "**OK**" to create the Data Manipulator.

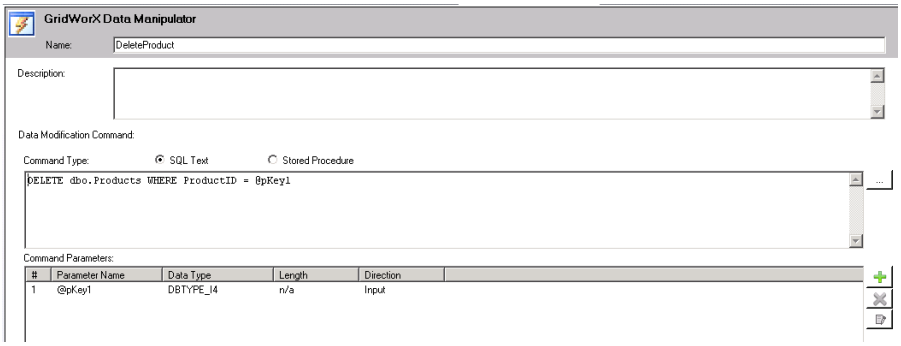


Figure 15.13 - Data Manipulator Screen

GridWorX64 Runtime

The GridWorX64 Runtime allows GENESIS64 clients to natively access database information through FrameWorX Server. When the GridWorX64 Viewer client requests some data, GridWorX64 Runtime starts processing this request by checking the GridWorX64 Server Configurator settings. Providing this is a legitimate request, GridWorX64 Runtime retrieves data from a database and returns the data back to the client.

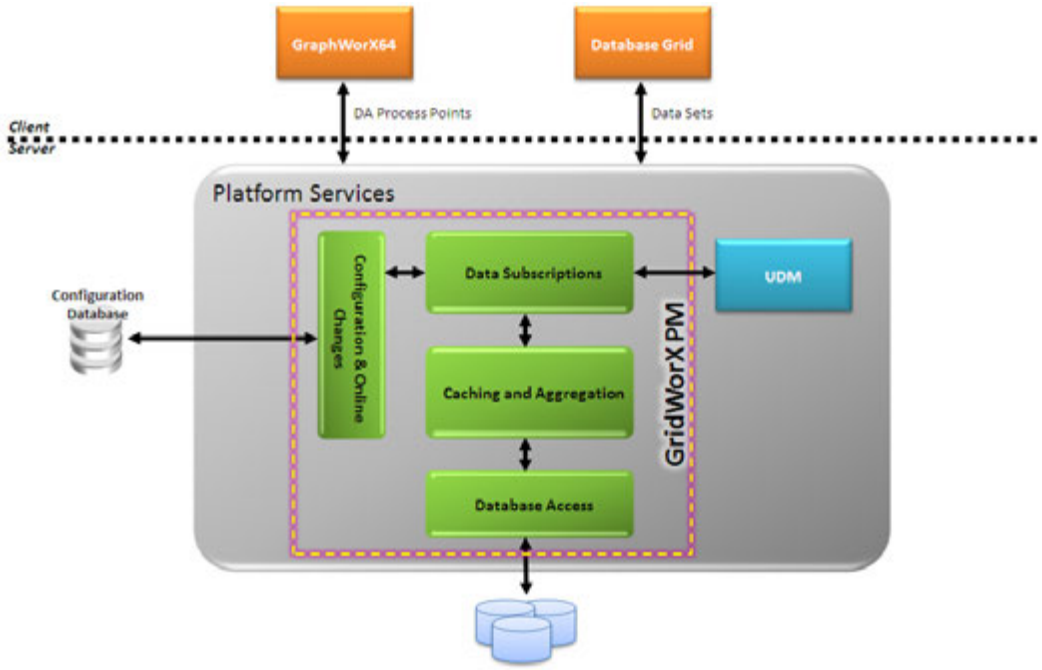


Figure 15.14 - GridWorX64 Runtime Architecture

With GridWorX64, you can visualize database data in the form of OPC-DA or through the GridWorX64 Viewer Control. You can perform basic read and write actions along with stored procedures for connected databases. This application works similar to Data Mining OPC Server in GENESIS32.

You can create a GridWorX64 viewer by using the GridWorX64 provider in the GENESIS64 Workbench. You can also create a GridWorX64 Viewer inside GraphWorX64 by using the GridWorX64 Viewer Control available in GraphWorX64. Steps for creating each are described as follows.

15.2. Create a Grid Using GridWorX64

To create a grid using GridWorX64 in the **GENESIS64 Workbench**:

1. From the Windows desktop, open the Workbench by selecting Windows Start Menu →All Programs → ICONICS →Workbench.
2. In the Project Explorer, click on the GridWorX64 provider button bar to bring up GridWorX64.
3. On the ribbon's **Home** tab, click the **New Grid** button. This opens a new tab called New Grid with a grid on it.

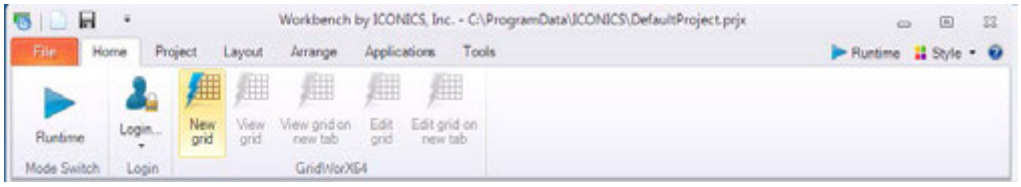


Figure 15.15 – New Grid Button on Home Tab

4. Go to the **GridWorX64 Configuration** ribbon and click the **Configure Viewer** button. This opens the configurator where you can configure the grid.

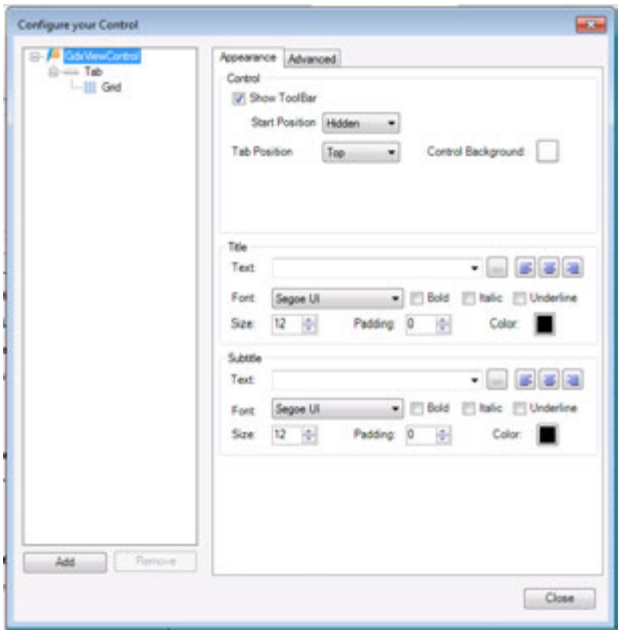


Figure 15.16 - GridWorX64 Configuration

5. Close the configurator when you are done.
6. Use the File menu to save the project with the grid in it.

15.3. Create a Grid in a GraphWorX64 Display

To Create a GridWorX64 Grid **inside a GraphWorX64 display**:

1. From the Windows desktop, open GraphWorX64 by selecting Windows Start Menu →All Programs → ICONICS →Workbench → GENESIS64 → GraphWorX64 → GraphWorX64.
2. Create a new display or open an existing one.
3. On the ribbon, go to the **Controls** tab.

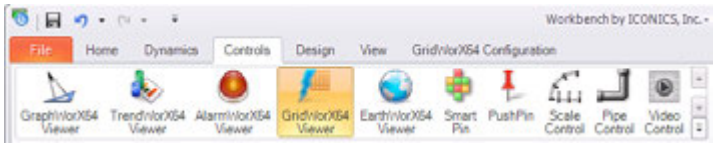


Figure 15.17 - GridWorX64 Viewer Control in GraphWorX64

4. Click the **GridWorX64 Viewer icon**. This changes the cursor pointer to a crosshairs. Using the crosshairs, draw a rectangle on the display canvas where you want the grid to be added.

5. Go to the **GridWorX64 Configuration** ribbon and click the **Configure Viewer** button. This opens the configurator where you can configure the grid.

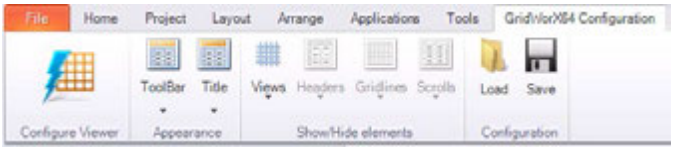


Figure 15.18 - GridWorX64 Configuration Ribbon

The ribbon's buttons are used only in a GridWorX64 Viewer that has been added to the GraphWorX display. Each button on the ribbon, as well as the configurator that the ribbon opens, is described as follows.

- **Configure Viewer** opens the GridWorX64 Configurator in the form of the Configure Your Control dialog box, where you can configure basic read and write actions for the grid. In the configurator, you can add tabs to the control, and grids to each tab.

  The GridWorX64 configurator in GraphWorX64 display configuration has the following components:

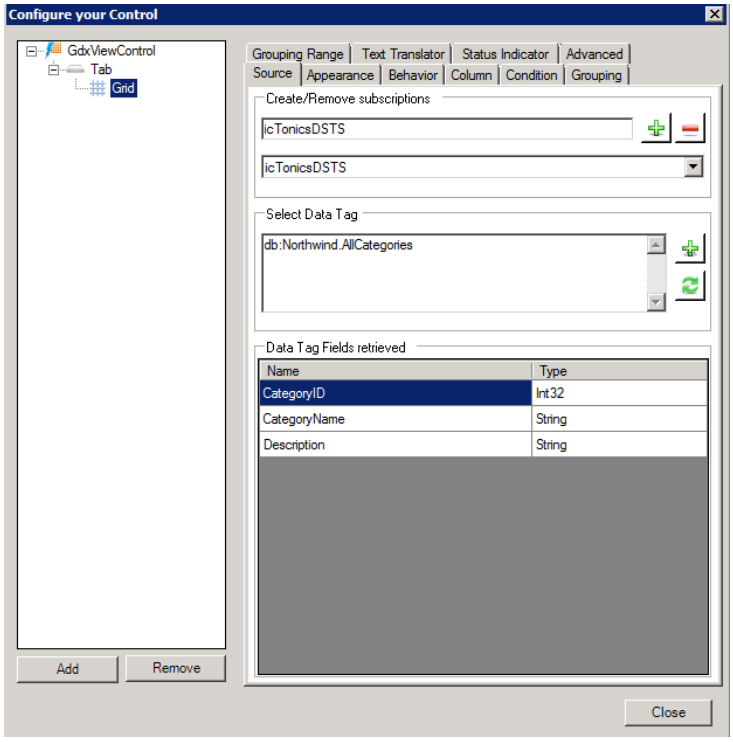


Figure 15.19 - GridWorX64 Viewer Control Configuration Dialog

For the **Control**:

- The Appearance tab lets you configure the control's overall appearance.
- The Advanced tab gives you access to the control's properties. Helpful tool tips that appear on the tab describe selected properties.

For each **Tab**:

- The Appearance tab lets you configure the tab's appearance.
- The Advanced tab gives you access to the tab's properties. Helpful tool tips that appear on the tab describe selected properties.

The following tabs appear for each **Grid**:

- **Source** is where you select a subscription and Data Tag.
- **Appearance** is where you configure the grid's general appearance.
- **Behavior** is where you configure how the grid works under specified conditions.
- **Column** is where you configure each column to appear in the grid.
- **Condition** is where you configure the visual cues that display for all values in the grid; to do this, you specify the conditions under which a value appears normally and all levels under which the data escalates to warning and critical stages.
- **Grouping** is where you select the criteria on which to group columns in your grid.
- **Grouping Range** is where you group large volumes of information so that runtime users can find it more easily.
- **Text Translator** is where you can translate data values into words so that text, not the values themselves, appears in the grid.
- **Status Indicator** is where you can define sets of icons to be used in the grid.
- **Advanced** gives you access to the grid's properties. Helpful tool tips that appear on the tab describe selected properties.
- **Toolbar** lets you hide or specify the location of the GridWorX64 toolbar during runtime.
- **Title** lets you show and hide the title and subtitle of the data grid during runtime.
- **Views** lists the tabs (views) in the grid along with information about each, and lets you select one.
- **Headers** lets you choose the types of headers to display in the grid.
- **Gridlines** lets you hide and show gridlines in the grid.
- **Scrolls** lets you control when and whether the scroll bars are shown or hidden for the grid.
- **Load** lets you open a saved grid configuration.
- **Save** lets you save the current grid configuration.

For information on **Interacting with GridWorX64 –** Reading/Writing to a Cell in GraphWorX64, refer to Lab 15 of this manual for an example of how this interaction is performed.

Lab: GridWorX64 Configuration L15

15.1. Lab Overview

GridWorX64 was developed in order to visualize database data in the form of OPC-DA or through the GridWorX64 Viewer Control. You can configure basic read and write actions along with stored procedures for connected databases. How it works is similar to how Data Mining for the OPC Server works in GENESIS32.

Estimated time to complete this lab: 30 minutes

15.2. Task Checklist

In this lab, you will complete the following tasks:

- You will create a New Database Connection, Data Source, and Data Item
- View your configured data in GraphWorX64 using the GridWorX64 Viewer Control or a Process Point

15.3. Support Files

To complete this lab, you will need the following file(s):

- Microsoft Northwind database that was installed with GENESIS64

15.4. GridWorX64 Configuration

In this lab, you will learn how to create a data source, data item and data manipulator in GridWorX64.

In this example, we will connect to Microsoft's Northwind example database in SQL Server. GridWorX64 also supports Generic OLE DB Connections, ODBC Connections, SAP Connections and Oracle Connections. You can choose the appropriate option depending on what type of database you want to interact with. The configuration will differ when creating a database connection, but will be similar in creating data source, data item and data manipulator.

15.4.1. Connecting to a Database

1. Open up Workbench by going to Start → All Programs → ICONICS GENESIS64 → Workbench.

2. In Project Explorer, expand the provider list in the explorer tree on the left and click on the GridWorX64 Server icon. Expand the GridWorX64 Server localhost branch.

   Under "GridWorX (Active)", select SQL Server Connections. Right click and select "New Database Connection."

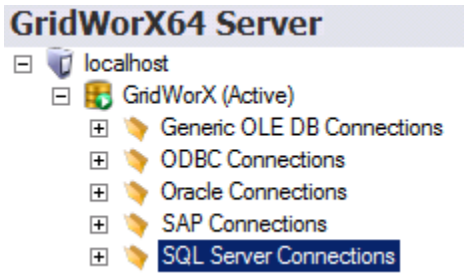


Figure L15.1 - GridWorX64 Server Explorer Tree

3. Name the Database Connection **icTonicsNorthWindDC** and click on "**Apply**".

4. Click on the "**Change Connection**" button.

5. Next, we will create a connection string. In the **Connection String** dialog box, double click on "SQL Server connections". Click on the "Add connection string" button to open up the "SQL Connection String" dialog box.

6. Enter your SQL Server name, log in information and choose the Northwind database to connect to. Your finished configuration should look like the figure below. Click on "OK".

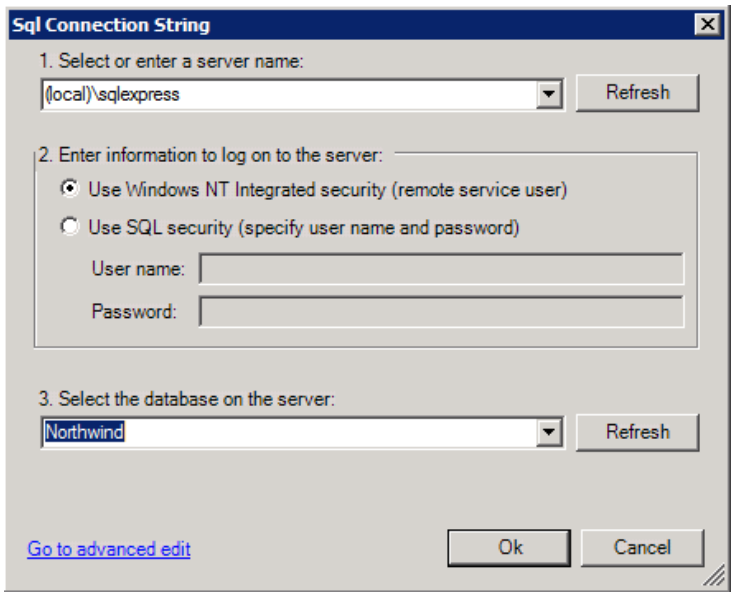


Figure L15.2 - SQL Connection String Syntax

7. Leave the default name “Northwind_1” and click on "OK".

8. Click on the “**Change Connection**” button again. Double-click on the “SQL Server connection” name and select **Northwind_1**. Click on "OK".

9. ***Important***! Your “Database Connection String” must resemble the string below. If it does not, return to step 4 and then proceed.

Provider=SQLOLEDB;Data Source=(local)\sqlexpress;Integrated Security=SSPI;Initial Catalog=Northwind

10. Click on the “Test Connection” button. If you do not see the "**Connection test succeeded**" message, then review your configuration. Click on "**Apply**".

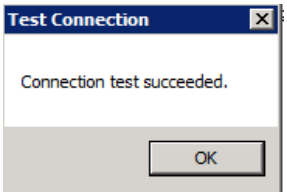


Figure L15.3 – Successful Connection Test Dialog box

15.4.2. Create a Data Source

A data source is a handle provided to the Unified Data Browser and is used to connect to a database table. It stores SQL command information to select, insert, update and delete records in a database table. "Select" configuration information is required. "Insert", "Update" and "Delete" configurations are optional.

To Create a Data Source:

1. In the **Project Explorer**, right-click on the database connection you created and choose "**New Data Source**".
2. Name this data source **icTonicsProductsDS**.
3. From the **SELECT Command** tab, click on the "**Edit Command Text**"  icon on the right and select **Simple SQL Designer** from the drop down. This will launch the "**Simple SQL Designer**" dialog box.
4. From the "**Selection Fields**" tab, click on the **Table Name**: ellipsis button. From the Select Table dialog box, click on the + sign and select **Table → dbo → Products**. Click on "**OK**".
5. Check the following columns: **ProductID, ProductName, CategoryID, UnitPrice**. Notice the SQL syntax in the Query Preview pane.
6. Select the "**Record Filtering**" tab and click on  to add a filter. Change the "**Relates To:**" field to **[Parameter]**. In the "**Related Value(s)**" section click on the icon  to the right of the "Query Parameter" field. With the "**Parameter Name:**" defined as **@CategoryID**, click on "**OK**".

7. Your "**Filtering Criteria**" configuration should resemble the following figure. Click on "**OK**" to proceed.

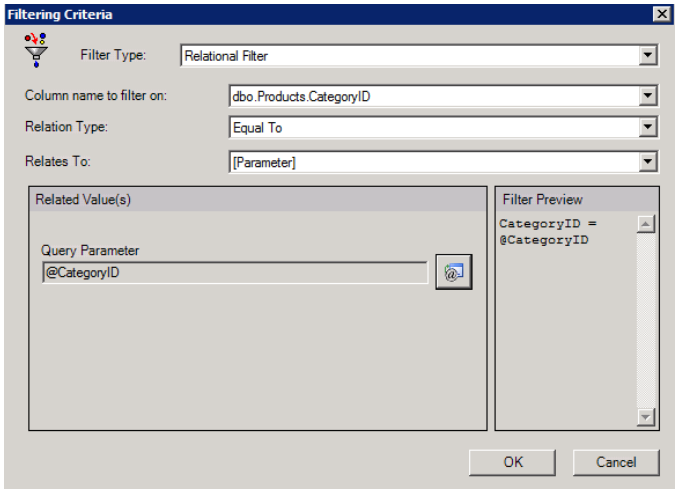


Figure L15.4 – Filtering Criteria

8. Notice there is also a "Sorting" tab. We'll be working with sorting later in this lab. Click on "**OK**" to continue. At any time, if you receive a request to "**Refresh Data Modification Commands**" click on "Yes", Proceed with Auto-Refresh.

9. Check the **Updates** checkbox and click on the "**UPDATE Command**" tab.

10. Choose **SQL Text** as the command Type.

11. Click on the ellipsis button and choose **Browse Table** from the drop down list. Expand Table → dbo and select the **Products** table. Click on "**OK**". The four command parameters should be automatically populated as in the following Figure L15.5.

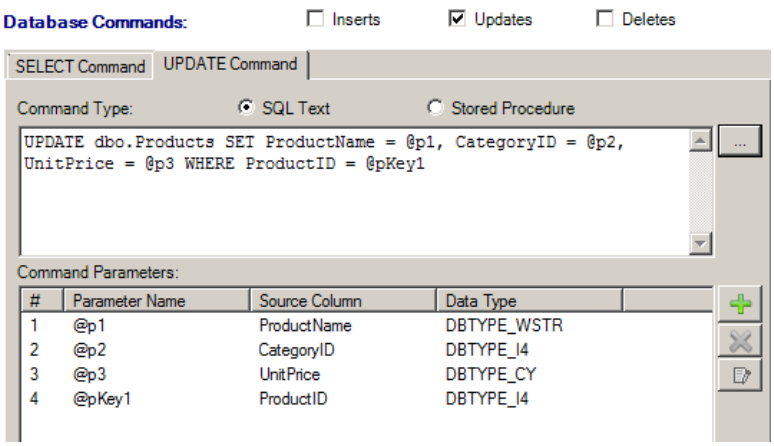


Figure L15.5 – Database Command

12. Click on "**Apply**" to finalize your configuration.

13. Let's execute the query to confirm proper operation. Click on the **Select Command** tab. Click on the magnifying glass icon  to execute this query. Replace the 0 with 1 in the Value field and click on "OK".

14. Your Query Results should look similar to the figure below. Note the values in the CategoryID column.

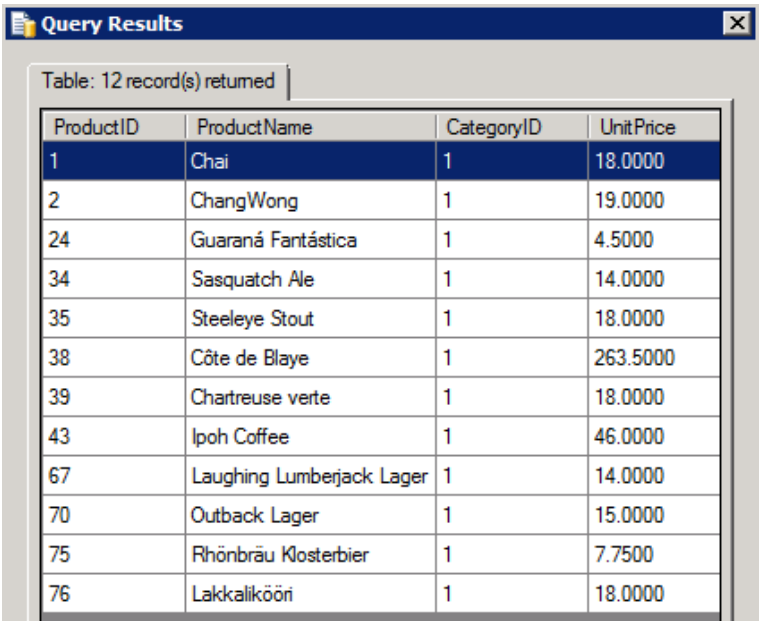


Figure L15.6 - Data Source Query Results

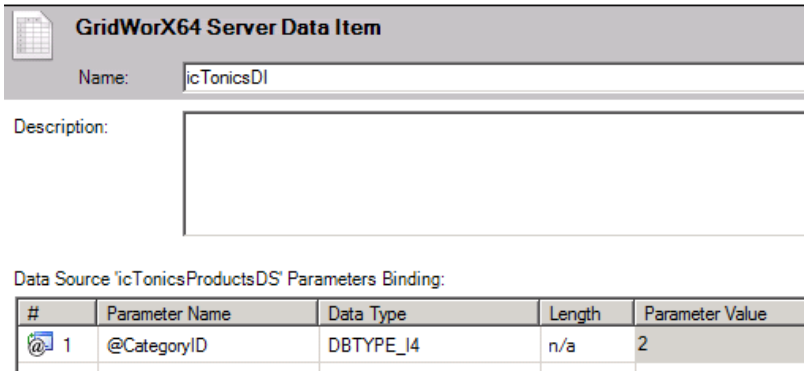
15.4.3. Create a Data Item

A data item is an instance of a data source with parameters already set. This gives users an easy handle to choose a customized data source without having to remember the parameter values to choose. Please note, data items cannot be created unless your data source has a parameter defined. To add a data item:

1. Right click on the **icTonicsProductsDS** data source, click on "New Data Item".

2. Name the data item **icTonicsDI**.

3. Assign a "**Parameter Value**" of 2 and click on "Apply".



L15.7 – Assigning a Parameter Value

4. You can click on the magnifying glass icon to preview the Query Results.

15.4.4. Create a Data Manipulator

Data Manipulator objects were designed to manipulate a database's data and/or structure using stored procedures or simple SQL commands when a database server or engine does not support stored procedures. Data manipulation actions are often executed manually using the @@Execute, read/write process point and typically initiated on a per-request basis. Let's create a data manipulator that will insert a record into the Categories table:

1. Right click on the **icTonicsNorthwindDC** database connection you created and click "**Add Data Manipulator**".
2. Name the data manipulator **IcTonicsInsertDM**.
3. Choose **SQL Text** as the Command Type and click on ellipsis button on the right to choose "**Browse Table for INSERT"**.
4. Expand the **Tables** tree and select the "**Categories**" table. Click on "**OK**".
5. Pictures are not currently supported by GridWorX64 so we'll have to modify the SQL statement and parameter list to accommodate. During the next steps you will remove all references to the picture column.
6. From the SQL statement in the Command Type, remove the **Picture** and **@p3** references. From the Command Parameters remove the **@p3** reference by selecting line # 3 and clicking on the red X icon to the right.

7. In the "**Data Source to refresh upon execution:**" drop down, select **icTonicsProductsDS**.

8. The following two screen captures depict the starting and finished configurations. Click on "**Apply**".

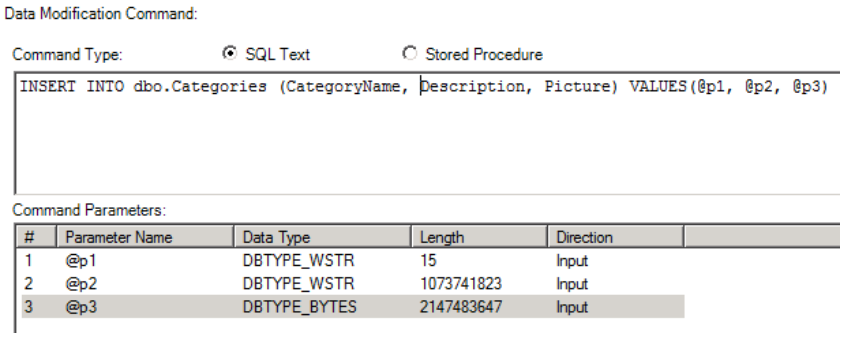


Figure L15.8 - Starting Data Manipulator Configuration

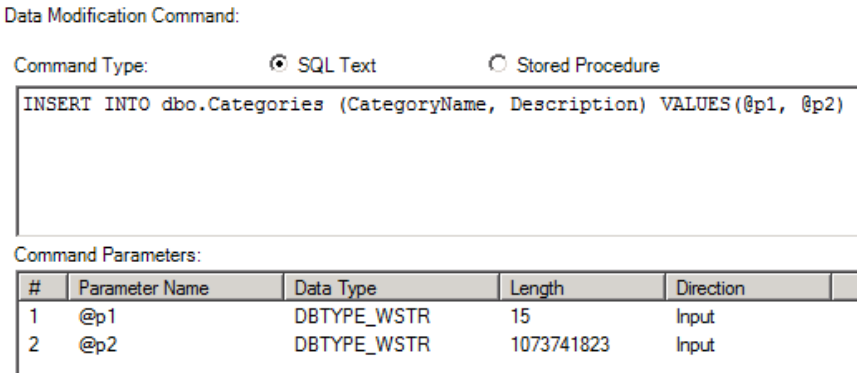


Figure L15.9 - Ending Data Manipulator Configuration

15.5. Interacting with Data from GraphWorX64

The GridWorX64 Viewer control allows you to view database information in a GraphWorX64 display. We'll continue by using the elements you've created thus far to create a user interface that can read and write information to and from the Categories database. The figure below is an example of what your icTonicsGridView display might look like when you're done.



Figure L15.10 - Example icTonics Gridview Display

15.5.1. Adding GridWorX64 Viewer

1. Open up Workbench by going to Start → All Programs → ICONICS Workbench.
2. In Project Explorer, choose **GraphWorX64** from the list of providers on the left. Click on "**New Display**" from the ribbon bar. As we did previously, add the icTonics template by clicking on the Design tab and selecting the **icTonics template**.
3. Select the **Controls** tab in the ribbon bar and click on the **GridWorX64 Viewer**.
4. Click anywhere on the empty display to size and drop the control into it.

15.5.2. Configure a GridWorX64 Viewer for Database Data.

1. Double-click on the GridWorX64 Viewer control to bring up the "**Configure your Control**" dialog box.

2. From the tree on the left, click on "**Grid**" and select the **Source** tab. Under the "**Create/Remove Subscriptions**" section, click on the icon to create a new subscription. Choose **Data Set Tag Subscription** from the drop down list.

3. Click on **Home → Databases → SQL Server → Northwind → Data Sources → AllCategories**. Note that the AllCategories data source is associated with the Northwind samples created by the GENESIS64 installation. Then click on "**OK**".

4. Change **New Subscription** to the name subscription name **icTonicsDSTS**.

5. Click on the icon to refresh the data tag fields retrieved. Your finished configuration should resemble the figure below.

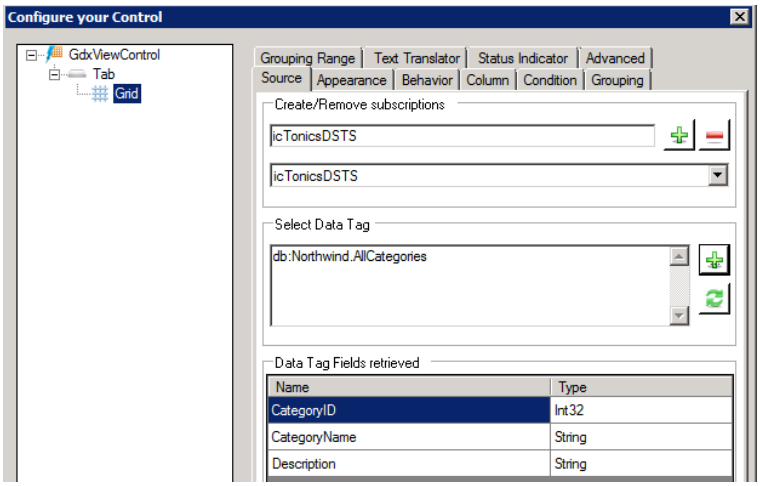


Figure L15.11 - GridWorX64 Viewer Source Configuration

6. Click Close to save this configuration. Save this display with the name **icTonicsGridView.gdfx**

7. Go into **Runtime** and you should see the query results similar to the figure below.

CategoryID	CategoryName	Description
1	Beverages	Soft drinks, coffees, teas, beers, and ales
2	Condiments	Sweet and savory sauces, relishes, spreads, and seas...
3	Confections	Desserts, candies, and sweet breads
4	Dairy Products	Cheeses
5	Grains/Cereals	Breads, crackers, pasta, and cereal
6	Meat/Poultry	Prepared meats
7	Produce	Dried fruit and bean curd
8	Seafood	Seaweed and fish

Figure L15.12 - GridWorX64 Control Query Results

8. Since we included the update command in the data source, if you double click on any cell (other than the CategoryID which is the key) you should be able to edit the values. After editing, pressing the enter key will write the changed value to the database.

15.5.3. Configure Data Entry Fields

We can connect the data source, data item and data manipulators that we created to GraphWorX64 Process Points or Data Entry fields. Let's examine how to use data entry dynamic objects to read and write to and from a database.

1. Let's add three data entry fields to the display to write information to the Categories table.

2. With the **icTonicsGridView** display open in configuration mode, click on the **Dynamics** tab and select the **Data Entry** object from the "Text Objects" section.

3. Place the object in the display above the GridWorX control. From the Data Browser select **Home → Databases → SQL Server → icTonicsNorthwindDC → Data Manipulators → IcTonicsInsertDM → Parameters → @p1** and click on "**OK**". Your DataSource will look like this: **db:icTonicsNorthwindDC.icTonicsInsertDM.@p1**

4. Place a **Text Label "Category"** above the data entry field you just added.

5. Add a **second** data entry to the right of the first. From the **Data Browser** click on "**OK**".

6. From within the **Dynamics** tab, edit the **DataSource** string to this: **db:icTonicsNorthwindDC.icTonicsInsertDM.@p2**

7. Place a **Text Label "Description"** above the data entry field you just added.

8. Add a **third** data entry to the right of the first two. From within the **Data Browser**, double click on the **icTonicsInsertDM**, and then click on the **@@Execute** element. This is a Boolean tag that will trigger writing the data to the database. Click on "**OK**".

9. From within the Dynamics tab, confirm the **DataSource** string looks like this: **db:icTonicsNorthwindDC.icTonicsInsertDM<@p1=, @p2=>.@@Execute**

10. Place a **Text Label "Execute"** above the data entry field you just added.

    Note: You can **Execute** using a **Button** with a pick action **Write Value** and an **OnUpValue** of **True**.

11. You can use other GraphWorX64 tools such as alignment, text labeling, sizing, etc… and GridWorX64 Viewer features such as you'll find on the Configure your Control, Appearance tab to spruce up the display if you like. When you're done, save the display and go into runtime.

12. Type in the word ICONICS into the Category field and hit enter. Type in GENESIS64 into the Description field and hit enter. Type a value of 1 into the Execute field and hit enter. Note the "**Successfully wrote**" message in the lower left corner of the viewer.

13. Your update to the table should be apparent after the next table refresh.

15.5.4. Using Filters in Runtime

1. Let's exit runtime and return to the GraphWorX64 configure mode. In this section, we will add a **filter** to show products whose **CategoryID** is less than **5** as below.

2. Double click on the GridWorX64 Viewer to open up the Configurator again. Notice that when the **Grid** is selected in the tree on the left, and you click on the **Behavior** tab. Checking the "**Automatically update data after**:" checkbox will allow you to alter the refresh rate of the grid in runtime. Since the data grid refresh is fairly resource intensive, you should configure this parameter wisely.

3. To the right of the "**Select Default Filter**" field, click on the  to create a new filter. Name the filter **CategoryIDLessThan5**.

4. In the "**Configure the filter**" dialog box, double click on **CategoryID** and click on the **Relational** button. Select the **< Less than** sign. Place the cursor to the right of the < sign and type in the value **5**. Your completed configuration should look similar to the follow figure. Click on "**OK**", then close the configurator and enter Runtime.

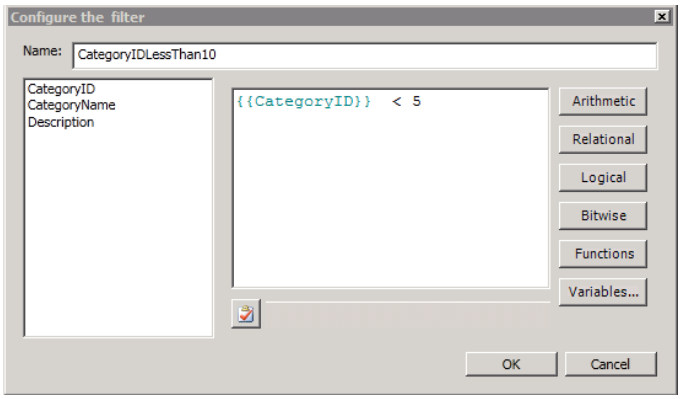


Figure L15.13 - Setting a Behavior Filter for a GridWorX64 Viewer

5. Notice that when the filter is applied you only see the CategoryID records 1 through 4.

6. Right click on the GridWorX64 control (except for the title bar) and select "Filters". Uncheck "**CategoryIDLessThan5**" and click on "**OK**". The Viewer will show all the records again.

   Note: This is client-side filtering and does not alter the *Select* statement issued to the database by the GridWorX64 Server.

Review: GridWorX64 R15

15.1 Objectives Review

You should now know and understand the following topics:

- Understand the functionality of GridWorX64 components
- How to create Database Connectors
- Uses of Data Sources, Items and Manipulators

15.2 Summary

GridWorX64 allows GENESIS64 clients to natively access database information through the FrameWorX Server. GridWorX64 can visualize database data in the form of OPC-DA or through the GridWorX64 Viewer Control in GraphWorX64. You can create basic read and write actions along with stored procedures against the connected databases. How it works is similar to how Data Mining OPC Server works in GENESIS32.

GridWorX64 consists of a GridWorX64 Configurator, GridWorX64 Runtime Server and a GridWorX64 Viewer. The GridWorX64 Configurator stores information in a database that is queried by the GridWorX64 Runtime Server as runtime is invoked. GridWorX64 Viewer is a client and allows you to see data from different databases.

If you have experience with GENESIS32 you can compare the following products:

- GridWorX64 Server Configurator – v9 Data Mining Configurator
- GridWorX64 Runtime – v9 DBOPC Server
- GridWorX64 Viewer – v9 DB Mining ActiveX

15.3 Terms

Data Connector	Data Source	Data Item
Data Manipulator	ODBC	OLE DB

15.4 Test Your Knowledge

In this chapter, you learned about GridWorX64. Let's review some of the key features by answering the following question.

1. True or False: The GridWorX64 reacts to information contained in the GridWorX64 configuration database and delivers data to the client(s)?

2. Which of the following database interfaces and database types are supported by GridWorX64?

   a. MS SQL Server
   b. Oracle
   c. SAP
   d. 64-bit OLEDB
   e. 64-bit ODBC
   f. All of the above

3. True or False: A Data Source is a handle provided to the Unified Data Browser that is used to connect to a database table?

4. True or False: Data Items are selectable and configurable independent of the Data Source configuration?

5. Which of the SQL commands are supported by GridWorX64?

   a. Select
   b. Insert
   c. Update
   d. Truncate
   e. Delete
   f. All of the above
   g. a, b, c and e
   h. a, b, d, and e

6. In a GridWorX64 viewer control, you can configure the viewer's…?

   a. Appearance
   b. Behavior
   c. Grouping
   d. Sorting
   e. All of the above
   f. None of the above

7. True or False: During runtime, GridWorX64 supports custom stored procedures?

8. True or False: In order to be accessible for configuration, data items require its parent data source to have at least one parameter defined?

9. True or False: Operators cannot trigger GridWorX64 commands through objects in a graphic display.

10. Data manipulators can be configured to perform which of the following operations to database tables?

    a. Update
    b. Distinct
    c. Insert
    d. Delete
    e. Select
    f. a, b, c and e
    g. a, c, d, and e
    h. All the above

Security 16

GENESIS64 uses a security model that is both granular and additive. The system is installed as part of the GENESIS64 installation, but it is not activated. Before creating an Administrative user, you do not need to log in to run the Security Configurator.

You can set individual rights and access based on users and groups, create named policy settings that can be applied to users and groups, and control access to individual applications, features, systems, and more. These settings can also be time based.

A security system administrator configures the system by adding users and assigning them specific privileges. Administrators may also associate a user with groups that have pre-assigned privileges so that the user has the effective rights of all the groups to which the user belongs to, in addition to the user's own privileges.

The Security Server offers a powerful way to configure access to your applications. All actions that can be performed by a user in any GENESIS64 application can be controlled via the Security Server.

After completing this chapter, you should know how to:

- Create an Administrative user in GENESIS64 Security Server
- Create standard user accounts with GENESIS64 Security
- Set up security groups and users
- Create account policies
- Set security privileges to define what a user can do
- Log in and log out from the system
- Use Active Directory accounts

16.1 Security System Components

The security system contains two components: the Security Server and the Security Server configurator that manages user logins and the various clients contained within GENESIS64 (GraphWorX64, TrendWorX64, AlarmWorX64, and so on). Any user login or logout that causes a change in security status is immediately posted to the affected clients.

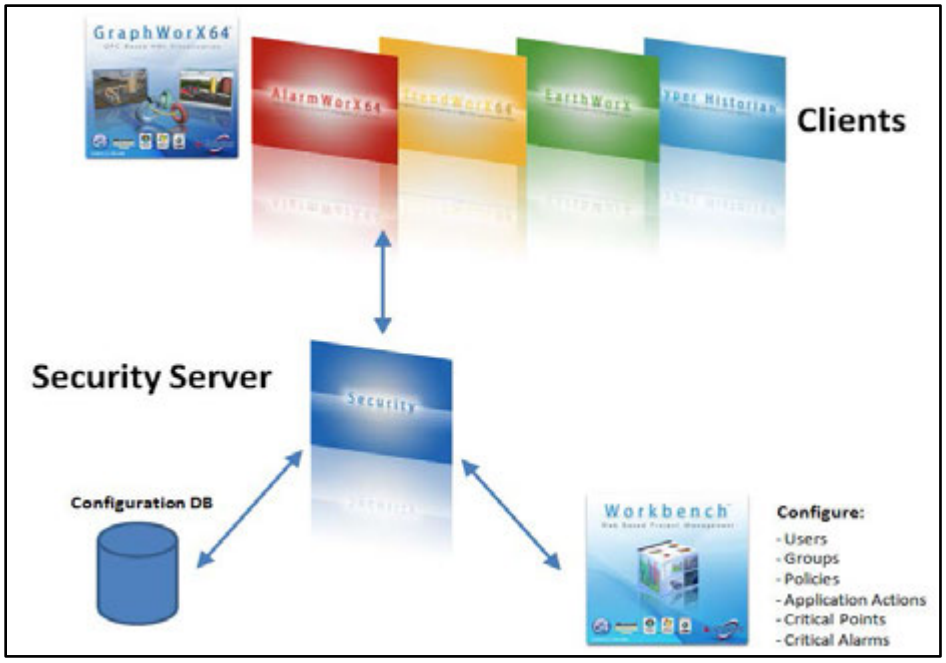


Figure 16.1 - Security Server and Clients Architecture

Note: Security assumes the least restrictive permissions. (Out of the box there is no security).

Security settings for the server are set from the **Security Server configuration forms**, an example of which is shown in Figure 16.2. To launch the Security Server from the Windows desktop, click **Start → All Programs → ICONICS → Tools**, then select the **Security** command. Alternatively, you can click on the "Security Server" icon in the Project Explorer of the GENESIS64 Workbench, as shown in Figure 16.2.

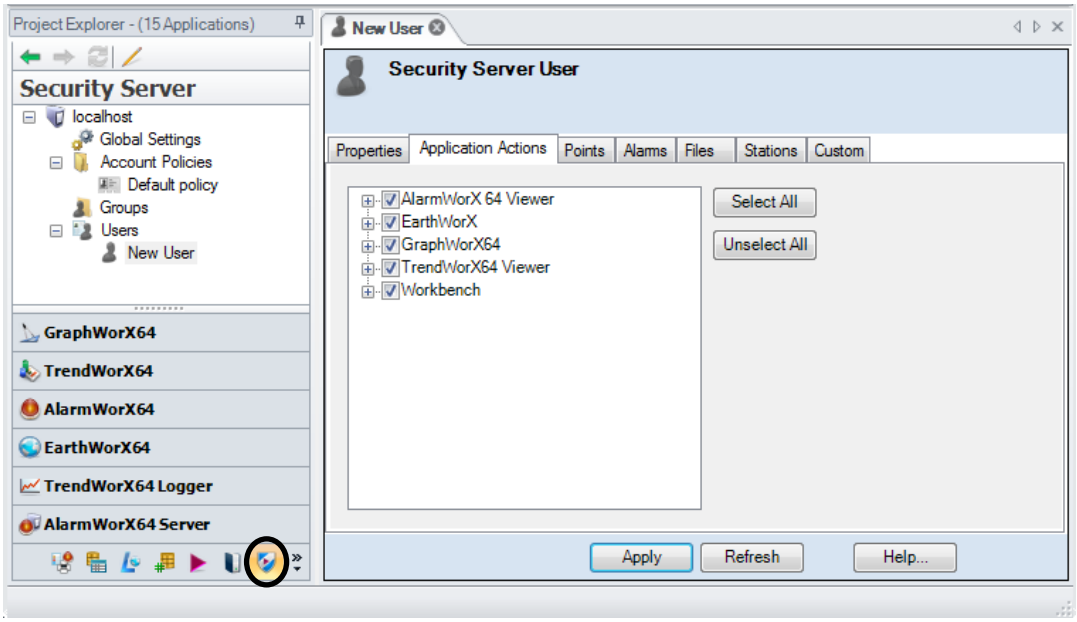


Figure 16.2 - The Security Server Configurator

16.2 Global Settings

Global Settings apply to the entire security server regardless of the security configuration. In the Global Settings, you can configure general settings of how the security server behaves as well as critical alarms and points.

You can access global security settings by double-clicking on **Global Settings** in the Security Server's navigation pane. The Global Settings **General** tab, shown in Figure 16.3, provides controls for how you want to display user information when you log in, how the user logs in (for example, with Windows), and where the security system obtains the list of security users.

You can choose to take advantage of already GENESIS64 users already setup in Windows' Active Directory instead of creating users in the security system by providing the necessary domain credentials, and then simply give these domain users or groups the proper permissions.



Figure 16.3 - General Settings

The following properties may be set on the General tab:

- **Security Active**: A checkmark here enables the use of the Windows domain services for account login. When this setting is enabled users with matching user names and domain names will automatically be logged into the Security Server. This eliminates the need for users who have already authenticated with their Windows domain to enter a user name and password a second time to gain access to the Security Server.
- **Provide list of existing users in the login dialog box**: When this box is checked, the Security Login dialog box displays all users in the User Name dropdown list. This option is useful for touch-screen systems.
- **Show last logged user in login dialog box**: When this box is checked, the Security Login dialog box displays the name of the last user that successfully logged-in in the User Name field.
- **Allow Simultaneous Login**: When this box is checked, different users can log into the same computer to use GENESIS64. When one user logs out, the other user can continue using GENESIS64.
- **Time window for interacting with critical objects**: This is the number of seconds after logging in that a user will be allowed to manipulate a critical point before being required to log in again. (Critical points are described in the next section.)
- **Security Mode**: You can choose to authenticate using either the ICONICS Security Server database or the Active Directory.

While using the Active Directory mode, it is possible to use groups and users defined in the domain to create your security settings. The Security Server is able to read data coming from Active Directory. Since we are connected to the Active Directory constantly, whenever there is a change in users and groups in the domain, the Security Server automatically gets the update. To preserve domain security, you will not be able to add active directory users or groups within the Workbench from a domain node. The Domain Administrator will need to access the domain controller in order to alter domain accounts.

Clicking on the "Active Directory Settings" button next to the Security Mode, allows you to configure domain connection settings and credentials, as shown in Figure 16.4.

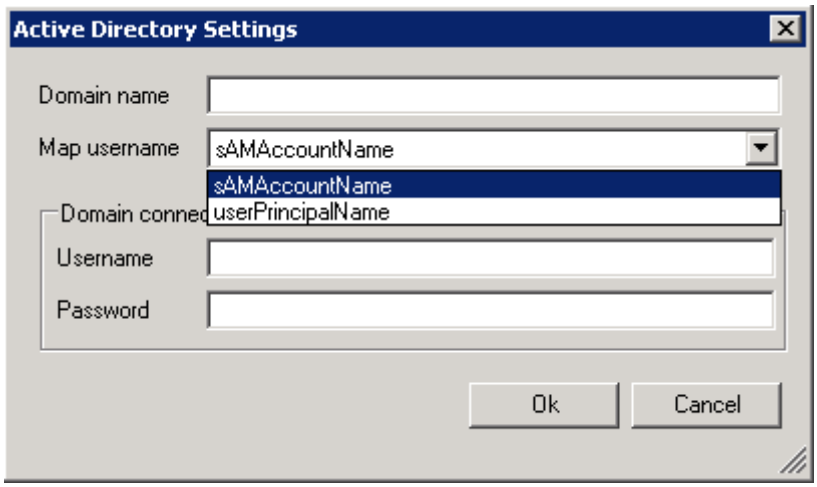


Figure 16.4 - Active Directory Settings Dialog box

- **Domain name**: the name of the domain that you wish to synchronize with
- **Map username**: sAMAccountName or userPrincipleName, as shown in Figure 14.4. The
  - **sAMAccountName:** is the name of the Windows account in the domain.
  - **userPrincipleName:** is the Internet login style based on the RFC 822 standard, and is typically the e-mail address in the form of username@domain.com.
- **Domain connection credentials**: Login credentials for the domain. You should enter credentials that can query the Active Directory here.

16.2.1 Critical Points

On the **Critical Points** tab of Global Settings, shown in Figure 14.5, you can designate some subset of writable points (OPC data items) as Critical Points. In order for a user to write a new value to a critical point, the following two conditions must be met:

- The user must be granted rights to the critical point via user or group configuration or via one of the explicit groups to which the user belongs.
- The user must have logged in within the **Time window for interacting with critical events** configured on the **General** tab of the Global Settings dialog box.

If the first condition is met, but not the second, the client application (for example, GraphWorX64) will launch the **Security Login** dialog box, requiring the user to login again and the second condition. This identifies and authenticates the user who is writing the value.

Critical points use the same *include/exclude lists with wildcards* concept as the Points configuration (which we will discuss in detail in section 16.6.2) in the user and group properties dialog boxes. This lets you specify multiple tags without listing them individually.

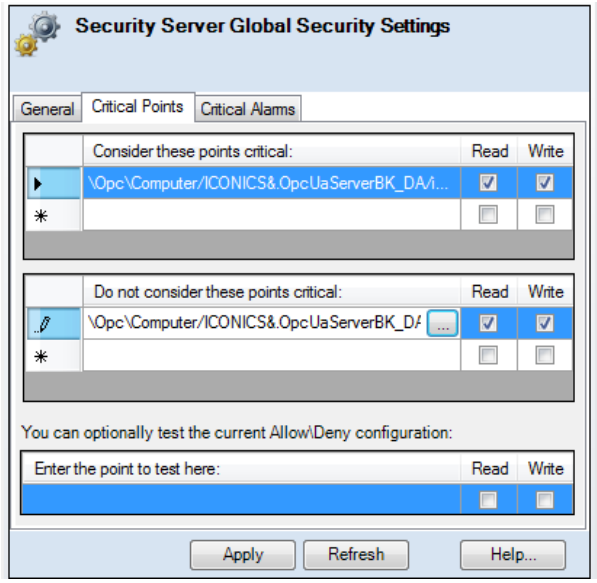


Figure 16.5 - Critical Points Settings

Note: If neither the 'Read' nor the 'Write' checkboxes are checked, then the request will be denied.

16.2.2 Critical Alarms

In the **Critical Alarms** tab of Global Settings, shown in Figure 16.6, you can designate a subset of alarms as Critical Alarms. When acknowledging a critical alarm, a user must login again immediately before acknowledging it, thus identifying and authenticating the user who acknowledges the alarm.

Critical alarms use the same *include (allow ack)/exclude (deny ack) lists with wildcards* concept as the Alarms configuration (which we will also discuss later in section 16.6.3) in the user and group properties dialog boxes. This allows you to specify multiple alarms without listing them individually.

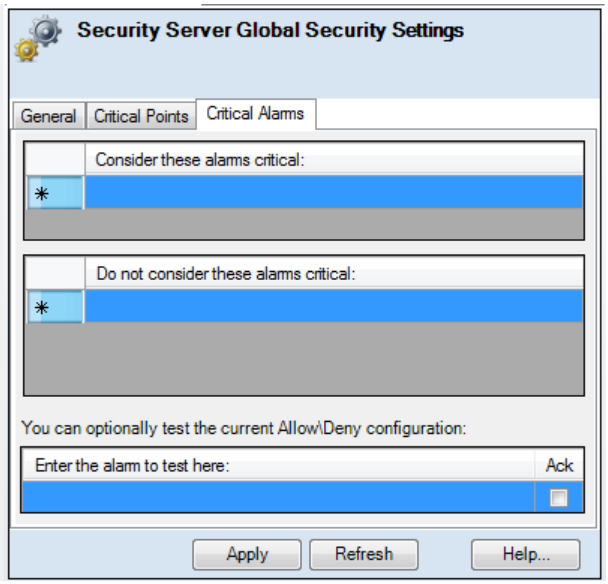


Figure 16.6 - Critical Alarms Settings

16.3 Security Objects and Their Relations

Security is created by adding security objects to the system. There are three types of security objects: Account Policies, Groups, and Users, as reflected on the navigation tree. There is an intricate relationship between these objects. Let us first understand the difference between a group and a user and how they relate before we talk about account policies.

16.3.1 Group

A group is a collection of users that share the same privileges. By default, there are no groups and any newly created group has the least-restrictive permissions possible, meaning the group is allowed to do everything. You can edit these security settings.

After a group is created, you can add users as members of the group. You can add any user to any number of groups, but only users can belong to groups. That means you cannot have a group belonging to another group. However, you can duplicate a group and then make changes to the duplicate if you want to create similar groups.

16.3.2 User

A user is an account that has a name, password, and specific privileges. When you first open the default Security Server, there are no users defined. The first account you create is the Security System Administrator account. After you apply the configuration for this account, the Security Server requires you to login to it to continue, which you should do in order to create additional user accounts.

Like groups, by default, a new user has the least-restrictive permissions possible, meaning the user is allowed to do everything. You can edit these security settings. Just as in a group, you can duplicate a user and make changes to it accordingly for users with similar settings.

When you define a user account, first you must fill in the Name and Password fields on the **Properties** tab, as shown below. You can use the Properties tab to change account policies or, by checking the **Account locked** checkbox, quickly disable a user account.

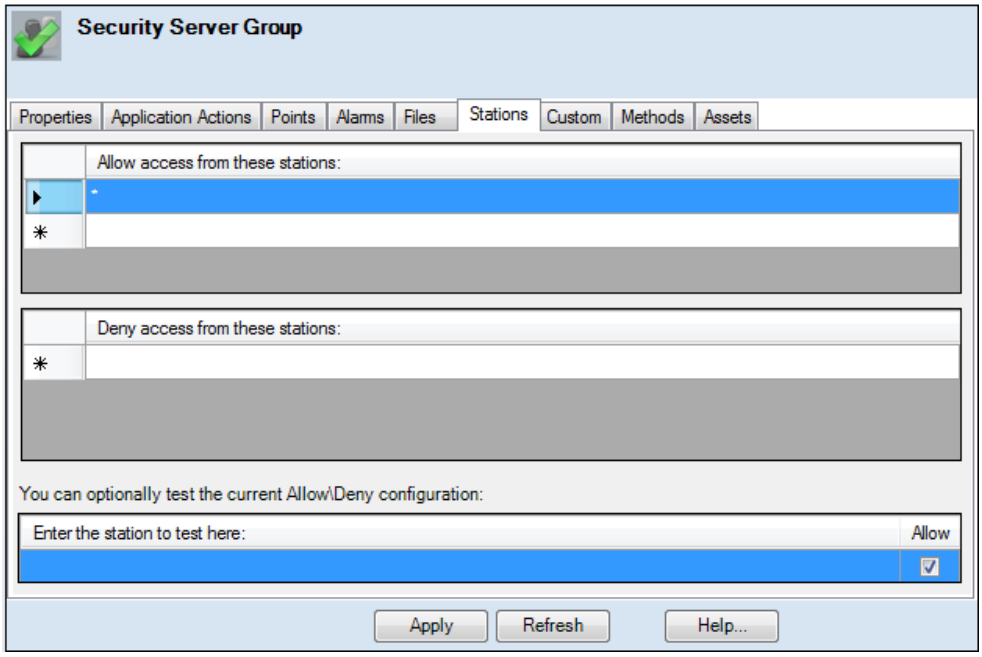


Figure 16.7 - User Properties Tab

This Figure shows the Properties tab for a user account. A group account would have a **Properties** tab with only the **Name** field for the account showing. Groups do not have passwords or an account policy, and they cannot be locked. The Root and Default Assets are referring to AssetWorX and what should appear for specific users.

16.3.3 Groups and Users

When it comes to the hierarchy of users versus groups, users always belong to groups. Or in other words, you can add users to any group. When you assign a user to a group, you are *extending membership* to the user. You extend membership either by altering the user account or by altering the group account. This is done in the Security Server using the **Membership** menu item from the object's right-click menu in the navigation tree control. Shown in Figure 16.8 is the Membership menu item for a group; a similar menu appears for a user.

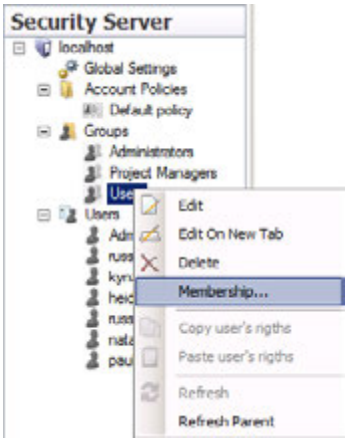


Figure 16.8 - Membership Command

Selecting **Membership** for a group opens the Group Properties dialog box, shown in background. You can then add users to a group by clicking on the "Add" button and selecting the users to be added, also shown in Figure 16.9.

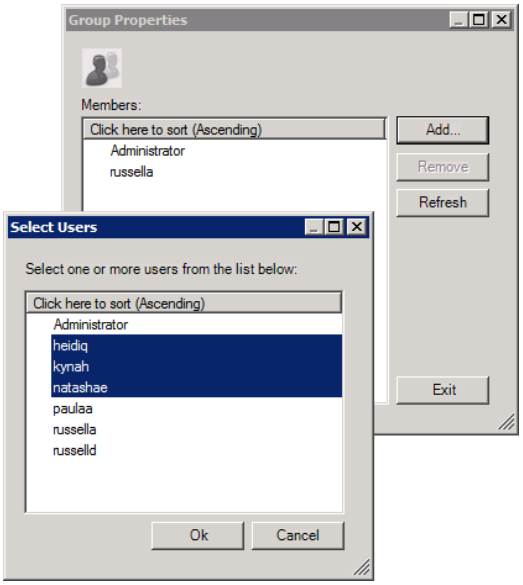


Figure 16.9 - Adding Users to a Group

The steps for associating groups to a user account are nearly identical. When you select the **Membership** command from the user account's right-click menu, you will see a dialog box for User Properties that is nearly identical to the Group Properties dialog box. Associate groups to a user by clicking Add and select the groups, as seen below.

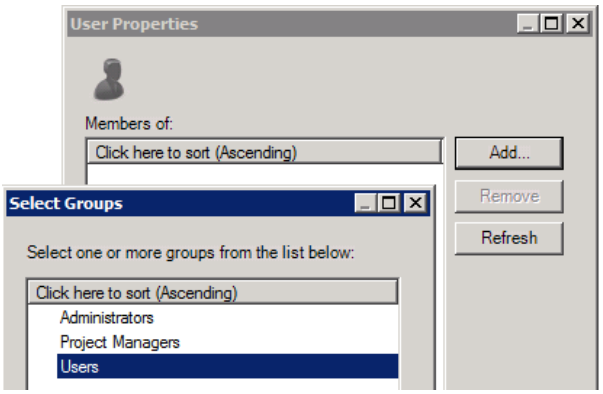


Figure 16.10 - Selecting Groups for a User

16.3.3.1 Permissions

When you add a user to a group, the user inherits the privileges of the group. For example, if the group has an *allow* permission to write to a certain process point, the users belonging to this group have permission to write to the point as well, without being specifically given the permission again in their user permissions.

A user's permissions are combined with the group permissions. This is important to remember because if you deny permission to the group, and then grant that same permission to a user of the group, the effect is that the user will be denied permission. A denied permission is always at the top of the permissions hierarchy.

16.3.4 Account Policies

Now that we understand groups and users and how they relate, let us look at Account Policies. An account policy is a set of rules that you can apply to any user. A default policy exists, but you can create additional ones. The Default Policy sets the base policy and is the one applied by default to any newly created user. You can create multiple account policies, but you can only apply one policy to any one user at a time.

Settings on the Account Policy form, as shown the Figure below, determine how passwords are used and whether user accounts automatically gets locked out after a series of incorrect login attempts.

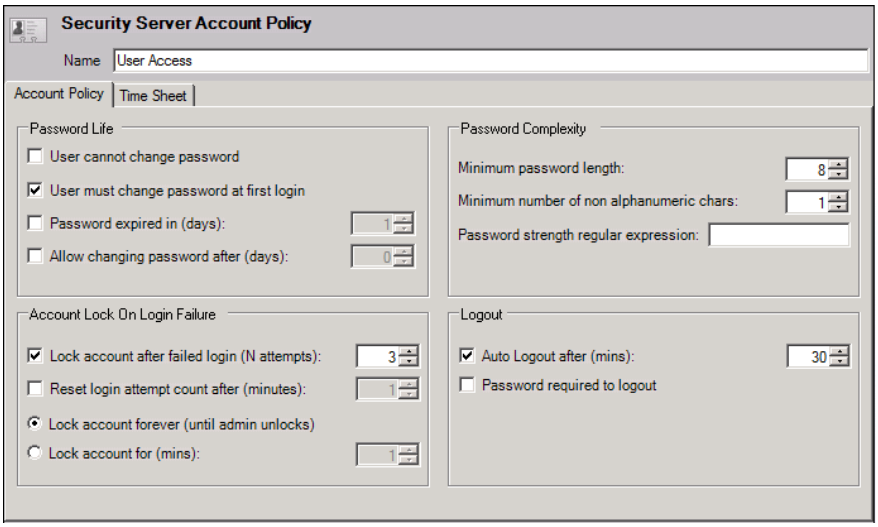


Figure 16.11 - Account Policy Settings

The four sections on the Account Policy tab are similar to the ones you would see in a Windows domain-based policy. The table below lists the fields and descriptions for the Account Policy tab in detail.

Table 14.1 - Account Policies

Security Option	Purpose
User cannot change password	If checked, the password can be changed only by an Administrator. If unchecked, the user can change it.
User must change password at first login	If checked, the user must change the password that the account starts with. If unchecked, the user does not.
Password expires in (days)	If checked, the user must create a new password after the current password has been used for the number of days specified. If unchecked, the user can use a password for as long as he or she wishes.
Allow changing password after (days)	If checked, the user cannot change the password again inside the number of days specified. If unchecked, the user can change the password again at any time.
Lock account after failed login (N attempts)	If checked, the account locks (and cannot be used) after anyone using the account to log in enters the wrong password the number of times specified. If unchecked, there is no limit.
Reset login attempt after (minutes)	If checked, the number of minutes (1 to 999) that must pass between any two login attempts to ensure that a lockout will not occur. If unchecked, the login does not reset after failed attempts.
Lock account forever (until admin unlocks)	If selected, when the number of failed logins is reached, the account is locked and requires a Security Server administrator to unlock the account.
Lock account for (mins)	If selected, when the number of failed logins is reached, the account is locked and cannot be logged in until this number of minutes has passed.
Minimum password length	The minimum number of characters required in the account's password.

Security Option	Purpose
Minimum number of non-alphanumeric chars	The minimum number of numeric characters or symbols (in addition to letters) required in the account's password. Setting this option to at least 1 is a good idea.
Password strength regular expression	A regular expression, which is the pattern describing a certain amount of required text of the account's password. (Its name comes from the mathematical theory on which it is based.) A few expressions are provided in the **Tip** at the end of this table.
Auto Logout after (mins)	If checked, the number of minutes (1 to 999) from the time of user login to the time the system automatically logs the user off. If unchecked, the user is never automatically logged out. By default, this feature is enabled and logout occurs after 60 minutes.
Password required to logout	If checked, a password is required to log out. If not checked, no password is required for logout.

Tip! You can use regular expressions to enforce password strength. Providing detailed information on regular expressions is out of the scope of this chapter, but you can find more details about regular expressions online. To get you started, you can try the strings below.

`^[a-zA-Z''-'\s]{1,40}$`
Sets a name up to 40 characters with some extra symbols such as apostrophe or comma. An example would be John Smith.

`^\d{3}-\d{2}-\d{4}$`
Useful for a social security number. An example would be 123-45-6789.

`(?!^[0-9]*$)(?!^[a-zA-Z]*$)^([a-zA-Z0-9]{8,10})$`
Sets a strong password between 8 and 10 characters which must contain at least

Each user is tied to one account policy. You may not apply an account policy to a group. As you now know, the account policy dictates how often a password needs to be changed, how complex it needs to be, as well as how to handle multiple login failures. You may need different account policies for different users but each user can use only one account policy.

Users with access to sensitive data should be subjected to more stringent policies than users who access less sensitive data. The following policies are recommended for users with access to sensitive data:

- Enforce a change of passwords after no more than 30 days; fewer for more sensitive data access.
- Lock the account after three or four failed login attempts, and force the locked out user to contact an administrator if access is not time-sensitive. For time-sensitive access, consider an appropriate timeout period in the **Lock account for** option.

- For the **Auto Logout after** option, set a reasonable logout time so that, if a user leaves a console unattended, the system cannot be compromised.
- Enforce password length and complexity. A password of six characters or more with mixed characters (alphabetic and numeric) is considered the minimum required to defeat brute-force attacks.

16.3.4.1 Time-Based Restrictions

The Time Sheet tab for account policies, shown in Figure 16.12, allows time-of-day restrictions on an hourly basis for users and groups. For hours that are selected (highlighted) in the lists, access is allowed. For hours that are not selected, access is denied. Figure 16.12 shows a configuration that allows access from 8AM to 2PM Monday through Friday.

You can use the **Shift** and **Ctrl** keys to make your selections. When you want to create a range or extend a range of selected time for *inclusion*, hold down the **Shift** key and click the start and end of the range. If you already have a range selected, hold down the **Shift** key then click on the point to which you would like to *extend* the selected range. Holding down the **Ctrl** key allows you to click and select *noncontiguous* areas of the time sheet.

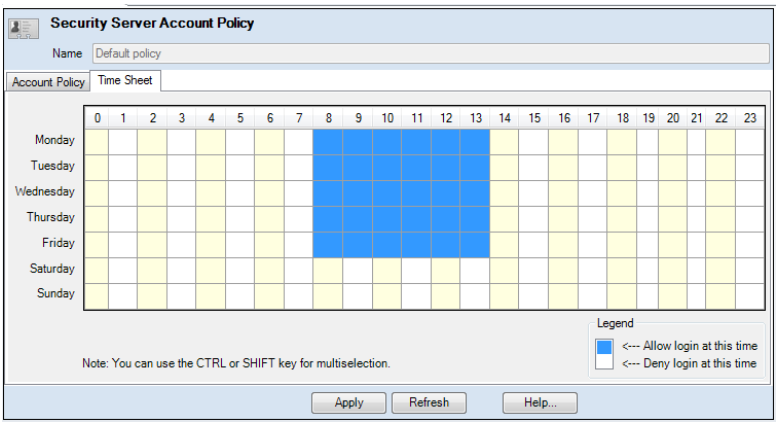


Figure 16.12 - Time Sheet Settings

16.4 Security Privileges

In Security Server, you configure users and groups using two separate forms. We have already looked at the Properties tab, which is similar for users and groups. These settings let you apply security privileges at either the group level or the user level. The Security Server uses the following rules for determining whether a privilege is extended to a user or denied:

- If a user lacks a privilege and is added to a group that has the privilege, the privilege is extended to the user.
- If a user has a specific privilege, but is added to a group where that privilege is denied, the user still has that privilege. *The **least** restrictive privileges apply.*

Application Action privileges may be extended or denied to users and groups based on the client application. The **Application Actions** tab, shown in Figure 16.13, is where these privileges are configured. Each GENESIS64 application exposes its own set of actions and you have fine control over each particular action. When you create a new account, all application actions are enabled. You should begin the configuration of an account by visiting the Application Actions tab and setting the proper application restrictions.

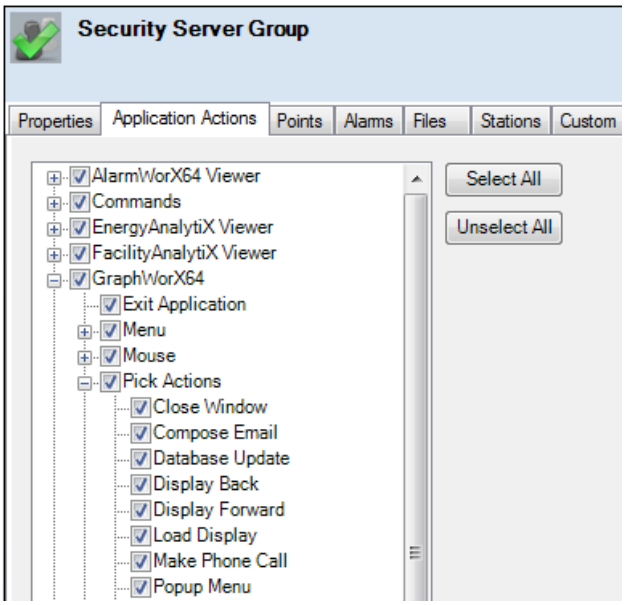


Figure 16.13 - Security Role Application Actions

For example, you can expand the **TrendWorX64 Viewer** node and uncheck the **Export Data** item to prevent the user from being able to export data. This lets you set up security so that some users can perform actions that others cannot.

16.4.1 Process Points

A user or group can be denied access to certain OPC points. You do this by placing OPC point addresses with or without wildcards in the *include* or *exclude* list of a user or group. Then, before a GENESIS64 client reads a value from or writes a value to an OPC server, the OPC point address is sent to the Security Server to determine if the operation should be allowed based on the currently logged-in user and or the groups to which the user belongs.

Use the **Points** tab, shown in Figure 16.14, to configure which OPC output points are allowed to be read from and written to by the user or group.

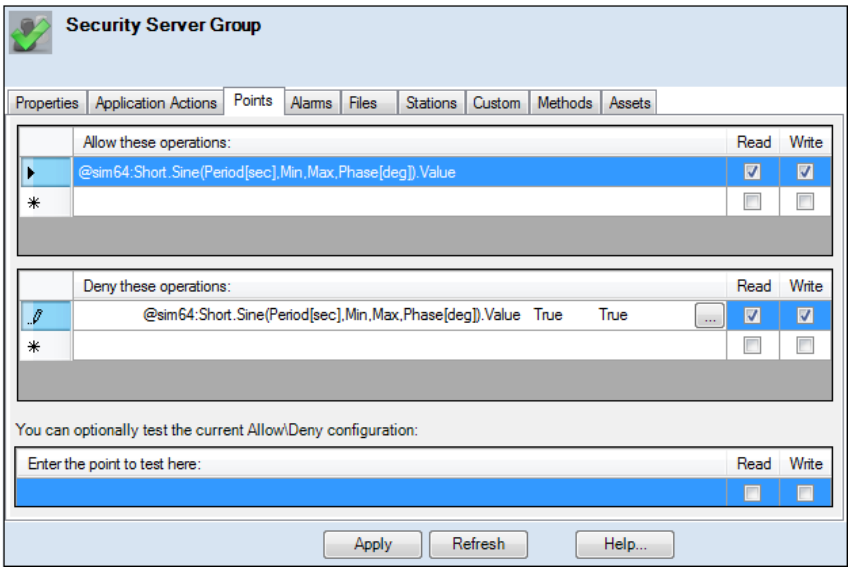


Figure 16.14 - Security Role Points Tab

The Points tab is divided into two sections: Include (**Allow these operations**) and Exclude (**Deny these operations**). To add additional criteria, click on the next blank line and enter the required string. During runtime, when a GENESIS64 client sends an OPC point address to the Security Server for access testing (to see if it is granted or denied), the Security Server compares the OPC point address to the include and exclude lists as described below for each active user and group until access is granted:

- Security Server compares the OPC point address with each string in the *include* list until a match is found. If no match is found, access is denied.
- If a match is found in the *include* list, Security Server compares the OPC point string with each string in the *exclude* list. If no match is found in the exclude list, access to the point is granted, and no further testing of active groups and users is performed. But if a match is found, access is denied.

Please see section 16.6.6 for more information about “Permissions” and section 16.6.7 for more information about “Wildcards and Performance Optimization”. Information in these sections will help you form a strategy for setting up the Allow and Deny lists.

16.4.2 Alarms

You can also protect individual alarms and groups of alarms by placing alarm names, with or without wildcards, into the include list (to **allow acknowledgement**) or the exclude list (to **deny acknowledgement**) for a user or group. GENESIS64 applications will query the Security Server for alarm access before acknowledging an alarm. The **Alarms** tab, shown in Figure 16.15, is used to allow or deny user acknowledgement of alarms during runtime.

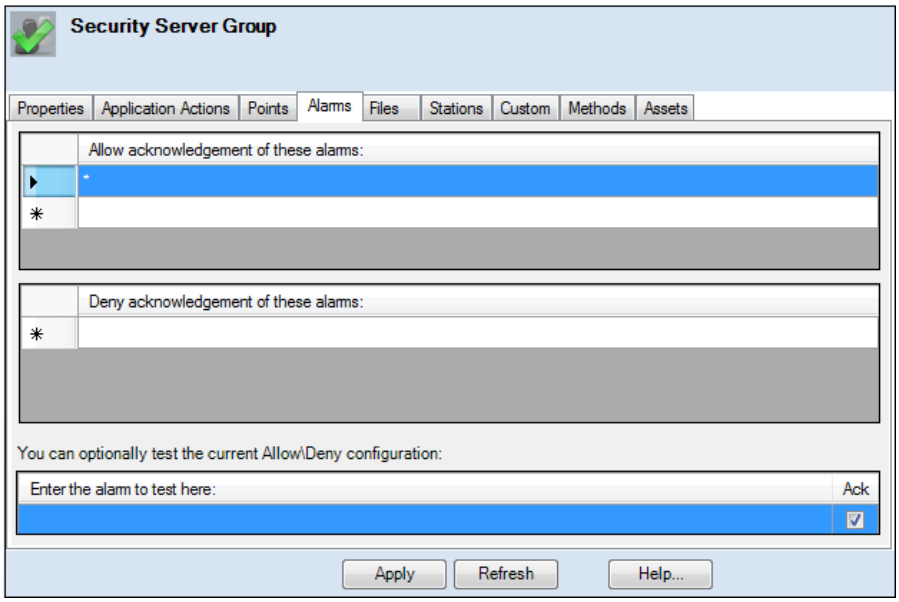


Figure 16.15 - Security Role Alarms Tab

The runtime processing and wildcard pattern matching for the Points tab apply here as well. Please see section 16.4.7 for more information about "Wildcards and Performance Optimization".

16.4.3 Files

You can also protect individual files and groups of files by placing file names, with or without wildcards, into the include list (to **allow** access) or into the exclude list (to **deny** access) for a user or group. When opening a file, Security Server checks for GENESIS64 permission to see if the user has permission to read, write, and/or modify the file. Files that are typically secured in this way are GraphWorX64 display files.

You can grant file permissions for the following actions:

- **Read** lets a user open existing files in runtime and configuration mode.
- **Write** lets a user create and save new files.
- **Modify** lets a user modify and save existing files.

The **Files** tab, shown in Figure 16.16, controls access to files that GENESIS64 applications may open at runtime. Entries here would be used to restrict certain users or groups from viewing, for example, specific GraphWorX64 displays.

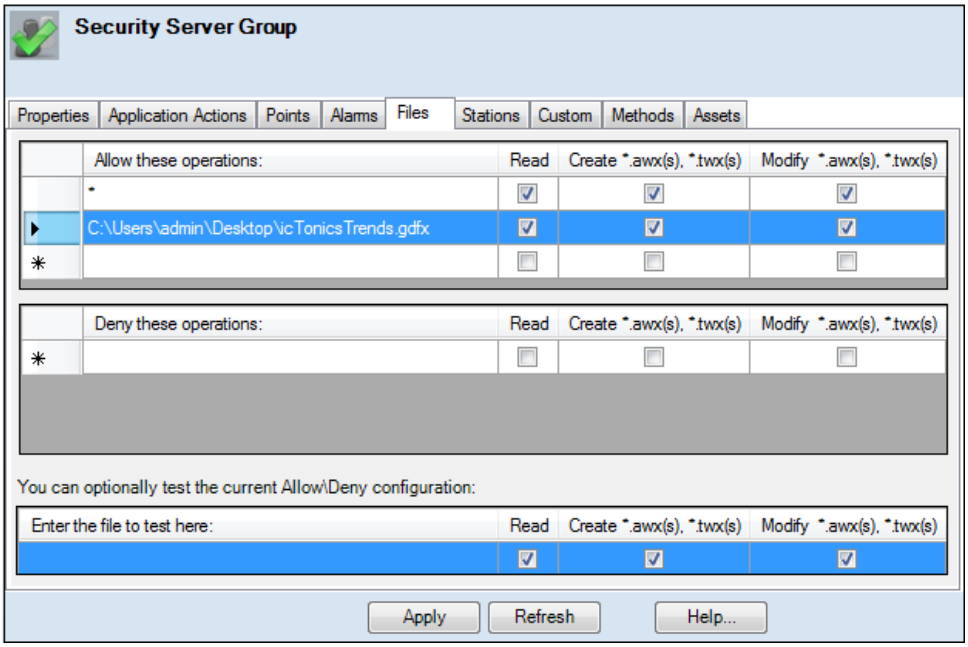


Figure 16.16 - Security Role Files Tab

The runtime processing and wildcard pattern matching for the Points tab apply here as well with the following differences:

- The pattern matching is done on the file extension, separate from the file name, to match the DOS wildcard semantics. For example, the wildcard string to indicate *all* files is *.*
- File names entered without a path are considered a match no matter what directory they are in.

Please see section 16.6.7 for more information about “Wildcards and Performance Optimization”.

16.4.4 Stations

Use the **Stations** tab, shown in Figure 16.17, to grant access to (allow login) or restrict access from (deny login) specific nodes on the network. Each node on a Microsoft network is identified by a unique computer name.

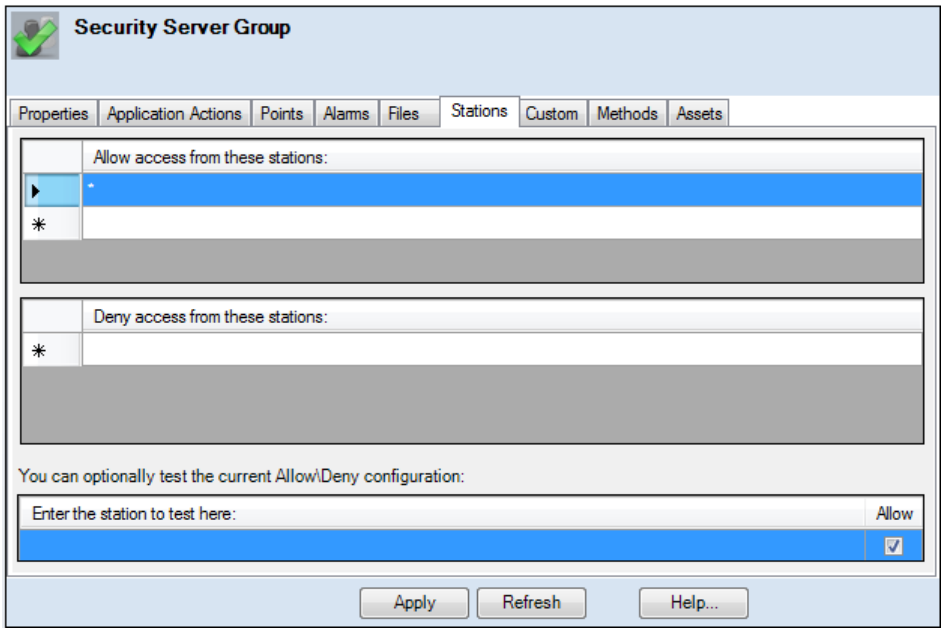


Figure 16.17 - Security Roles Stations Tab

To add a station to either list, click on the Ellipsis button. This displays the Select a Station dialog box, as shown in Figure 16.18. Select the Station Name from the dropdown list, or enter the name or system IP address to define the station criteria. Then click on "OK".

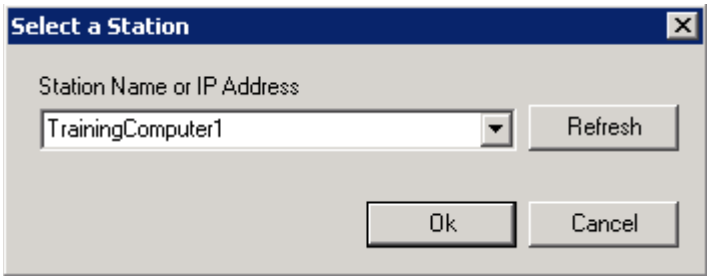


Figure 16.18 - Select a Station Dialog Box

16.4.5 Methods

Use the **Methods** tab to grant or restrict access to executing specific methods from being run by specific users or groups.

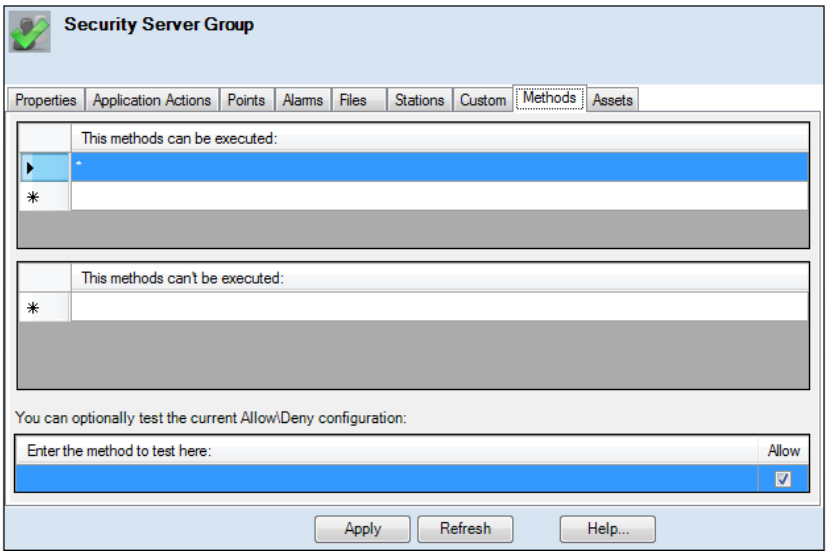


Figure 16.19 - Method Security

16.4.6 Assets

Use the **Assets** tab to grant or restrict access to seeing (Visualize) or utilizing (Execute) Assets from AssetWorX and any associated commands connected to the asset.

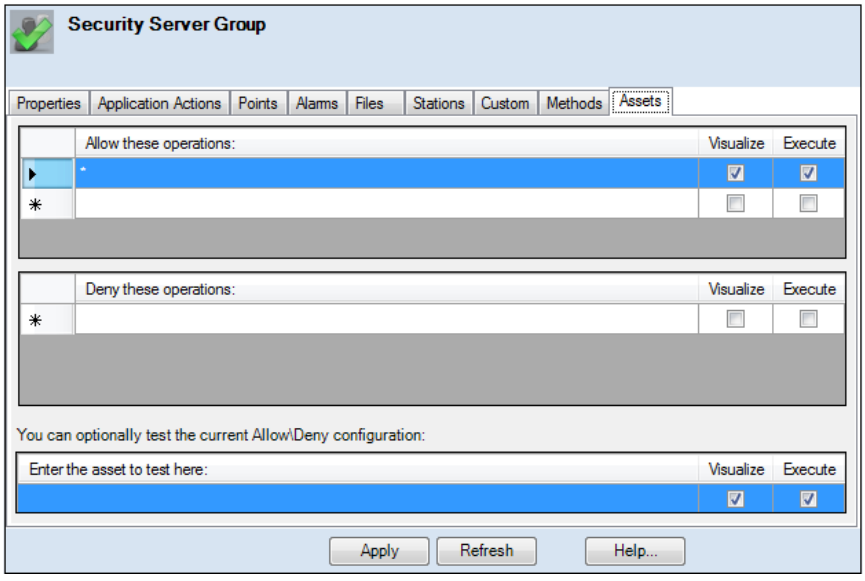


Figure 16.20 - Asset Security

16.4.7 Permissions

Permission to access a certain resource is granted to a user only after a security check is performed. The security check verifies the user's permission for the resource, as well as his or her groups' permissions on that resource. It is important that you understand how the system evaluates the permissions so that you set them up correctly. You should keep in mind the following security behaviors when setting up permissions.

A group identifies permissions for its user members. When a user, who is a member of multiple groups, tries to access a resource, the least restrictive permission in place for that resource is used. For example, group G1 *allows* write access to Tag1, while group G2 *denies* write access to Tag1. If User1 is a member of *only* G1, he will be able to write values to Tag1. If he is member of *both* G1 and G2, he will still be able to write values to Tag1 because the he was allowed access from G1 (that is, the least restrictive permission) is used.

Explicit permissions set for the user are combined with the user's group permissions. Following the example above, if User1 has explicit access that allows him to write to Tag1 and he is a member of G2, he will be allowed to write a value to Tag1, because the user setting overrules the group one.

Restrictions posed by the Allow list overrule permissions granted in the Deny list. This means that if a resource is in both lists, the user will be allowed to access the resource. Although it may seem strange that a resource would be in both lists, this is likely to happen when you use wildcards. For example, if you want to deny a user access to resource R1 only, you might put 'R1' in the Deny list and the wildcard character '*' in the Allow list (which also matches with R1). As a result, the system will find R1 in both lists, and will deny access to R1 because the Deny list overrules the Allow list.

16.4.8 Wildcards and Performance Optimization

The previous sections described the use of wildcards in place of tag names, file names, and other strings. Using wildcards lets you represent a swath of resources with just one string, letting you save time when configuring security. However, saving time is not the only advantage to using wildcards.

It is important to note that the system will always perform a security check when a user tries to access a resource. For example, when a user performs an action to read a tag value, the system uses the information in the user's Allow and Deny lists, as well as the Allow and Deny lists for the user's group. A comparison with all the strings in these lists is needed to check for access permissions. If there are many strings in those lists, the whole system slows down. Additionally, because access to a tag is granted *now* may be denied a second *later* (because the security privileges may have changed), a security check is performed each time the user accesses the tag, further slowing the system.

For this reason, having fewer strings to compare would improve overall performance. Therefore, it is always a good idea to try to use wildcards and to set up your system with these limitations in mind. Here are some examples of wildcard characters:

- **?** Any single character
- * Zero or more characters
- # Any single digit (0..9)
- **[charlist]** Any single character in *charlist*
- **[!charlist]** Any single character not in *charlist*

16.5 Logging-in and Logging-out

After the Security System has been set up, it will be necessary for users to log into the system to do their work and log out when they have completed their work so that their access can be authenticated based on the permissions granted to them. Authentication is handled by the Security Login dialog.

In GENESIS64, it is possible to log in and log out in both GraphWorX64 and the Workbench. It does not matter where users log in, they will be able to use all the GENESIS64 applications they have permission to access. The same is true for logging out; a logout takes effect throughout all GENESIS64 applications.

16.5.1 Security Login Dialog

To authenticate yourself, you must invoke the **Security Login** dialog, which you can launch from within the Workbench or GraphWorX64.

Open the Security Login dialog from the Workbench's File menu. Go to the Login\Logout button, as shown in Figure 16.21. There are times when the Login/Logout button has a submenu where you can select Client Login to log in to the client. And before you open any files, some providers' Home ribbons have a Login button, too.

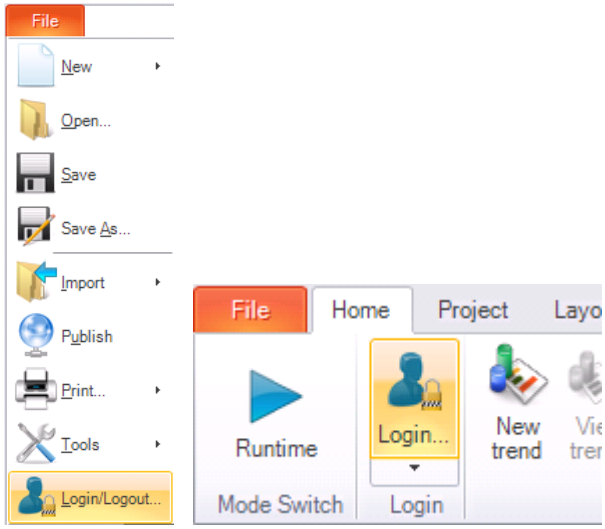


Figure 16.21 - Log In from the File Menu (Left) or the Home Ribbon (Right) when Available

Once the Security Login is open, as shown in Figure 16.22, users can enter their login information, and then click on **Log In** to authenticate. If the information they provide is correct, they are logged in to the system. Otherwise, they will be notified of that the user name and password are not correct.

The Security Login dialog stays open so users can try again. Remember, though, that if user security may be set up to lock out users who don't gain entry after a prescribed number of failed attempts.

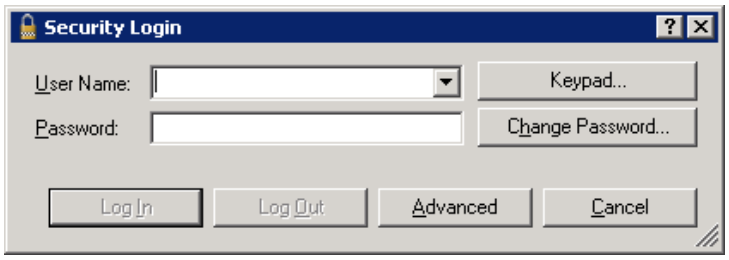


Figure 16.22 - The Security Login Dialog

On this dialog, it is also possible to show a **Keypad** window for touch screen systems. This can be helpful if you do not have a keyboard available.

You can also use the Security Login dialog to change your password. You will be asked to type your current password, and then to type a new password twice for confirmation.

To log out, use the Security Login dialog by clicking on the "Log Out" button. Depending on how your security profile is set up, you might be asked to enter your password to log out as well.

Clicking on the "Advanced" button lists the users who are currently logged in, the time at which each logged in, and the time at which each will be automatically logged out if the account remains inactive for a prescribed period of time, as shown in below.

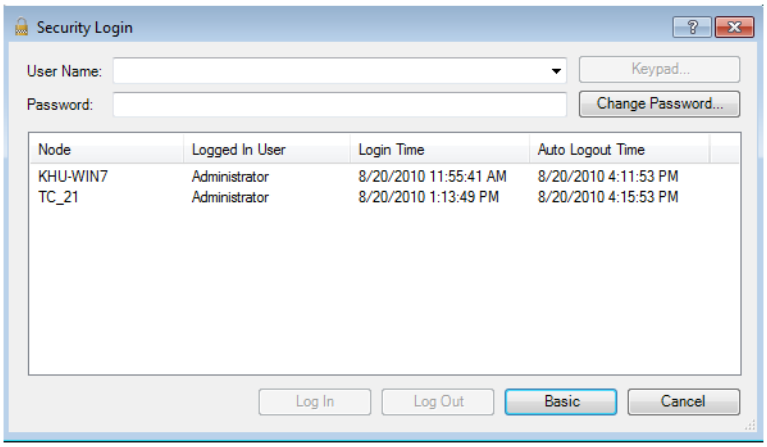


Figure 16.23 - The Security Login dialog in Advanced Mode

16.5.2 Multiple Logins

As you may have noticed in Figure 16.23, it is possible for more than one user to log in from the same node. This happens when a user logs in while another user has already logged into the system. It is important to note however, that only the last user that logged in is the active user, leaving other logged in users in a stand-by mode.

Using multiple logins is helpful if you need someone with additional permissions to perform a task you do not have rights for, or when you are getting help from an Administrative user. For example, someone with administrative rights may come to your machine to perform a task that requires administrative permissions. He would log in with his credentials, do what he needs to do, and then log out, returning control to you.

16.6 Advanced Settings

The GENESIS64 Security System also includes some advanced features that you may find useful when working with GraphWorX64 and other client applications.

16.6.1 Hiding GraphWorX64 Layers Based on Security

GraphWorX64 displays typically contain data that is accessible to one employee but not another. Creating different displays for each user or group would be time-consuming and it would be difficult to maintain solutions created in this way.

To help you in this task, GraphWorX64 allows you to show and hide layers based on users' security permissions. In this way, you can create a single display with information separated into layers that shows or hides depending on the logged-in user's credentials.

In the example in Figure 16.24, a layer called SecuredLayer contains critical information that needs to be secured. Non-critical information resides in a separate layer.

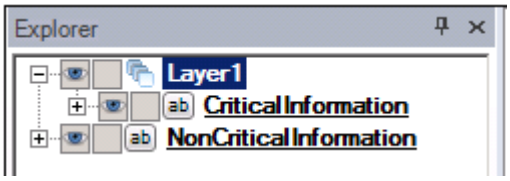


Figure 16.24 - Creating a secured layer

For the SecuredLayer to follow security rules, its **HideWhenSecurityDenied** property must be set to True in its Properties tab, shown in Figure 16.25.

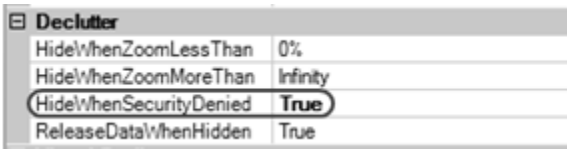


Figure 16.25 - Setting HideWhenSecurityDenied Property

What this achieves is that the critical information will be hidden to a user whose security denies it from viewing the layer. However, you still need to configure the Security Server to allow or deny a user access to the SecuredLayer.

This can be done in the Files tab of a user or group. By default, one layer is always accessible to everyone, but you can easily deny a user access to it by adding the layer to the list of denied points, using custom syntax.

The custom syntax you use in the list of denied files is ***FileName|LayerName*** – consisting of the file name and the layer name separated by a pipe (|) symbol. In the security example below, the GraphWorX64 display file is called MyDisplay.gdfx, and the layer is called SecuredLayer. Access to this critical layer is denied to this user.

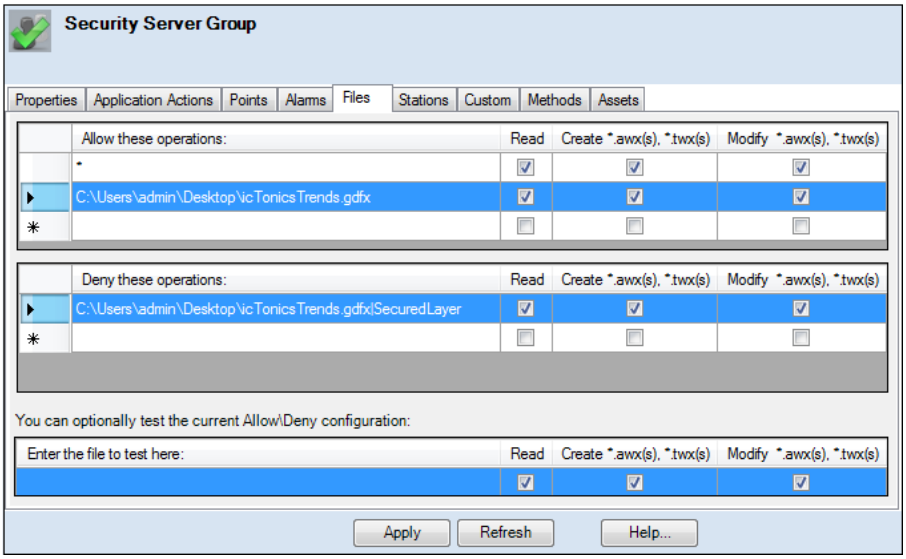


Figure 16.26 - Denying a User Access to a Secured Layer

With these settings, when this user accesses MyDisplay.gdfx, the layer called SecuredLayer will not be displayed. Restoring access to this user is as easy as modifying this Security form.

16.6.2 The ?DataPoint:TAGNAME Feature

We have just seen how to show and hide a layer depending on the Security permissions for a specific user. This can be a useful feature, but there are details it does not address. For example, how you do the following:

- Hide an individual object that cannot be part of a layer, such as a 3D object in a 3D View control.
- Have a specific dynamic be based on security permissions. For example, how to change the background color to red for a data source that the user can only read, and to green for a data source that the user can also write values to.
- Use scripting to return information about the user's security permissions for certain data sources.

To address these needs, GENESIS64 Security System provides an advanced feature that lets you query the user's security status of a single point by reading another point.

Suppose you want to know the current user's permissions for the following point: **@sim64:Float.Sine(10,0,100,0).Value**. If you create a Process Point using that tag name as the point's data source, you can see the point's value in runtime mode. If you modify the tag name by preceding it with **?DataPoint:** you will get a different value: a bitmask of the user's permissions for that point. Consider the following example in Figure 16.27, which shows the point's value in the top line and the security value (that is, the user's permissions) in the bottom line.

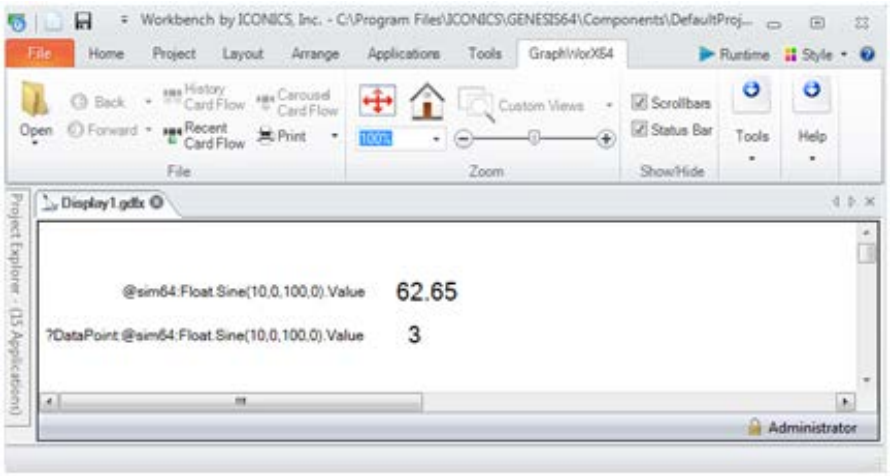


Figure 16.27 - Accessing Security Information

As you can see, the security value is 3. The possible values are:

- **3**: the user can both read and write to the point (the write bit is set to 1, the read is bit set to 1, which is '11' in binary, or 3 in decimal).
- **2:** the user can only write to the point (the write bit is set to 1, the read is bit set to 0, which is '10' in binary, or 2 in decimal).
- **1**: the user can only read the point (the write bit is set to 0, the read bit is set to 1, which is '01' in binary, or 1 in decimal).
- **0:** the user has no access to the point (both bits are set to 0).

In this example, the value is 3, therefore we know the user has both read and write permissions to the point.

> ***Tip!*** If you would like the process point to show up with meaningful text instead of numbers that represent the different states, try using state fields.

16.6.3 Hiding or Showing Objects Based on User Permissions

Now that we have seen how to read the security permissions of a user for a specific point, we will use this information to show and hide a 3D object.

> **Note:** You cannot use GraphWorX64 layers to show or hide 3D objects based on security; instead, you must show and hide 3D objects individually. The same is true of all other GraphWorX64 objects.

Before we begin, it is important to note that a point does not need to exist in any OPC server in order to set security permissions on it. However, the Security Server, when asked, will tell us if a certain user has permissions to read or write to the point.

With this in mind, there are two scenarios to consider:

- Scenario 1: Prevent an object from showing when the user lacks permissions to access the object's data sources. For example, you do not want to show a boiler if the user has no access to its temperature data source.
- Scenario 2: Set security permissions based on the confidentiality of the data, and set it independently from other factors.

In the first scenario, you would use an existing data source. However, in the second scenario, you would use a non-existent point, giving it a custom name that is meaningful for your application but does not match the name of any existing data source. For example, you might want to be able to set permissions easily for highly-secured objects, one type of object is called CriticalObject. You can use CriticalObject as the custom name of a non-existent data source that you use only for the purpose of evaluating user permissions to access the object.

Let us take a look at an example of how you might set this up.

First, you set up russella's permissions to deny him read/write access to the non-existent point, which in this example is called CriticalObject. The non-existent point acts as a real data source, but if you try to read its value, you will not get a valid value because the point does not exist.

In Figure 16.28, you can see the non-existent CriticalObject point added to the list of denied data sources for russella, and the user is denied both read and write access. If you refer back to the user permissions values on page 16-27, you will see this means that this sets the user's security permissions value for the CriticalObject point to **0** (zero).

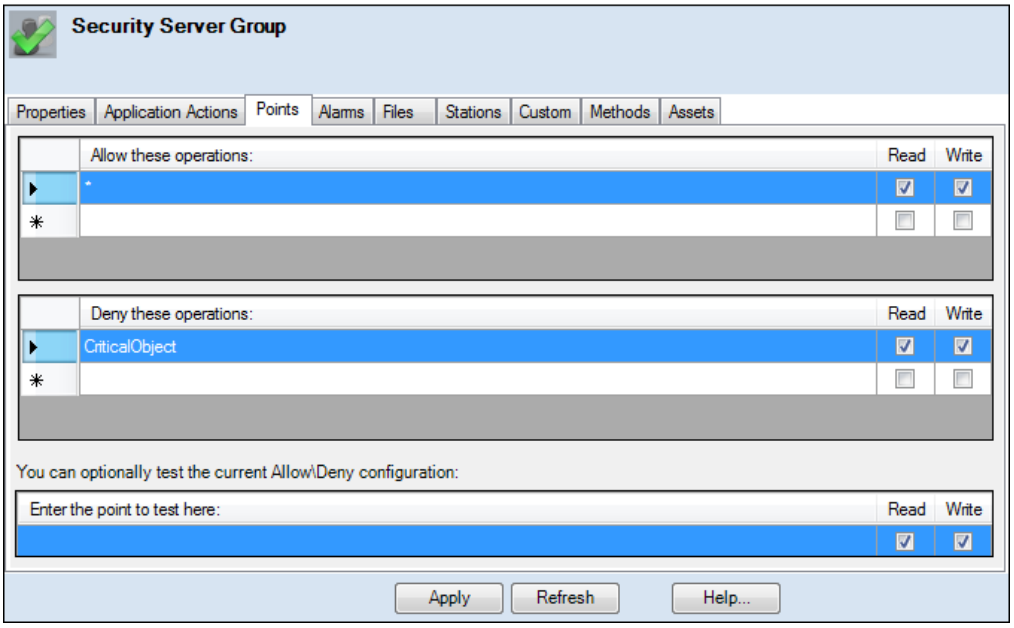


Figure 16.28 - Preventing the User from Reading and Writing to CriticalObject

Next, configure a GraphWorX64 display that contains the 3D objects you will show or hide depending on the user's permissions.

In Figure 16.29, the 3D objects are a donut and a sphere. The sphere is selected. Note the following settings for the sphere, which are shown on the right:

- Its data source is **?DataPoint:CriticalObject**. This means it will retrieve the user's security permissions for CriticalObject.

- Its **HideWhen** dynamic is set to hide the sphere when the data source value (which in this case is the user's security permissions) is equal to 0, which means the user has no read and write permissions.

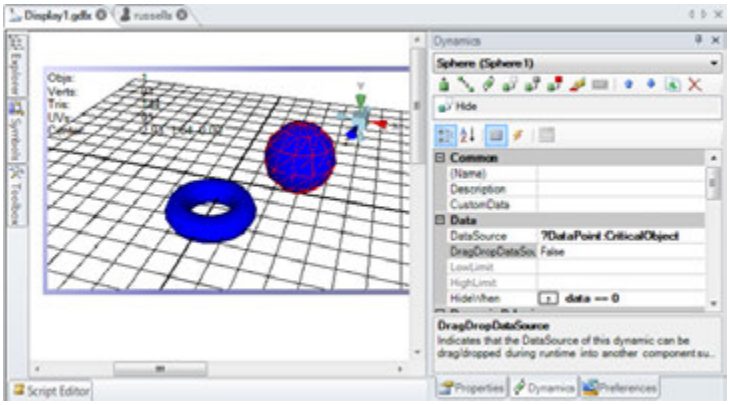


Figure 16.29 - Using a Hide Dynamic Combined with Security Settings

If you login with Administrator privileges, you will see that for administrators, the display shows all of the objects in the 3D View control, as shown in Figure 16.30.

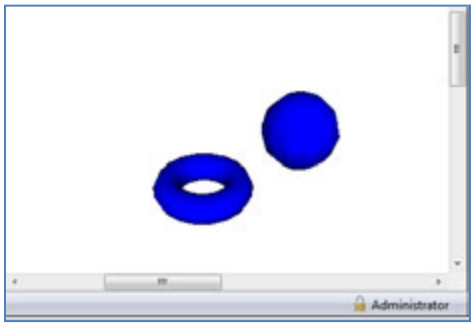


Figure 16.30 - 3D View Seen Using an Administrator Account

Now, if you log in as russella, you can see that the content has changed to reflect the permissions granted to russella; the sphere is no longer visible, as shown in Figure 16.31.

> ***Tip!*** When you are in GraphWorX64, you can easily tell who is currently logged into the system. The lower right corner of the display shows the logged in user.

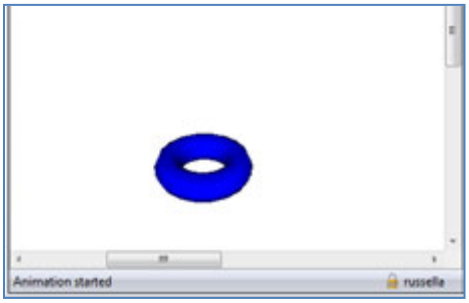


Figure 16.31 - 3D View Seen Using User Account russella

This advanced feature gives you very granular control over what users can see in a GraphWorX64 display. You can use **layers** for an easy-to-configure and powerful security integration, and you can use the **?DataPoint:** feature when layers are not sufficient.

16.6.4 Layers

Layers can be used to display information selectively based on selections made by the runtime user, or based on the security levels that are currently active in runtime. Additionally, layers can be used to de-clutter the screen content based on zoom level.

When you create a new GraphWorX64 2D display, any object that you draw or add to the work surface is placed on a *primary layer* of the display. The primary layer provides the same capability as a background in a paint program. This primary layer always exists and is visible, but is not shown in the hierarchy of objects, groups, and layers that appear in the Object Explorer.

But you can add other layers to a display. To add another layer to a display, click on the "Layer" button in the Shapes section of the Home tab, as shown in Figure 16.32. Notice that this is a 2D feature that applies to 2D displays.

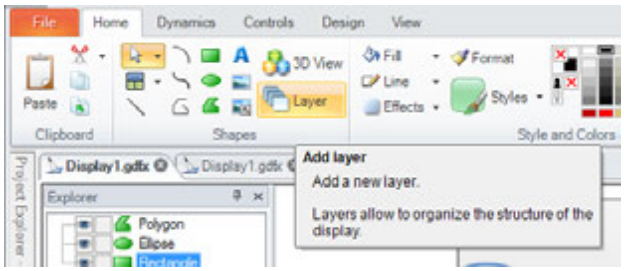


Figure 16.32 - How to Add a Layer

The new layer is added to the top of your drawing, no matter what the currently selected object is, and is numbered sequentially as "Layer *n*", where *n* is numbered numerically starting at one.

You can select any number of layers at a time. When highlighted in *gray*, a layer is active; when highlighted in *blue*, a layer is selected.

To move a layer up or down in the Z-order, click the layer until its highlighting turns from grey to dark blue, then drag the layer object in the object Explorer to the new position and release the mouse button. Figure 16.33 shows Layer4 (which is selected) being moved into a new position below Layer2 in the Explorer.



Figure 16.33 - How to Move a Layer Before (Left) and After (Right)

You can also nest layers using sub-layers, as shown in Figure 16.34. To create a sub-layer, select the layer that will contain the sub-layer (or select an object contained inside that layer), then click the "Add Layers" button. You can also drag a layer in the Explorer to the position in the hierarchy that you want the sub-layer to occupy.

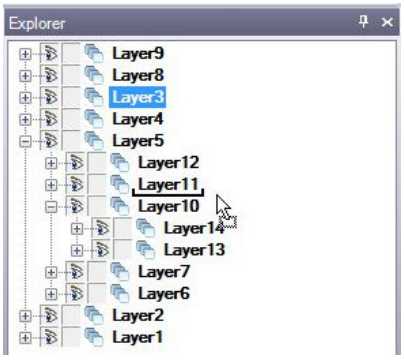


Figure 16.34 - Layers and Sub-layers in the Object Explorer

Layers can be renamed in the Object Explorer. Each layer's name must be unique. Layers can be hidden and locked just like any other object.

To add objects and groups to a layer, add them to or draw them in the contents area. Then move them into a layer by dragging the object in the Explorer, or by cutting it from one layer and pasting it into another.

Objects appear in the content area regardless of the layer they belong to. In Figure 16.33, all three objects in the content area are selected; the Rectangle and Ellipse belong to Layer 3, but the Label object belongs to Layer 2.

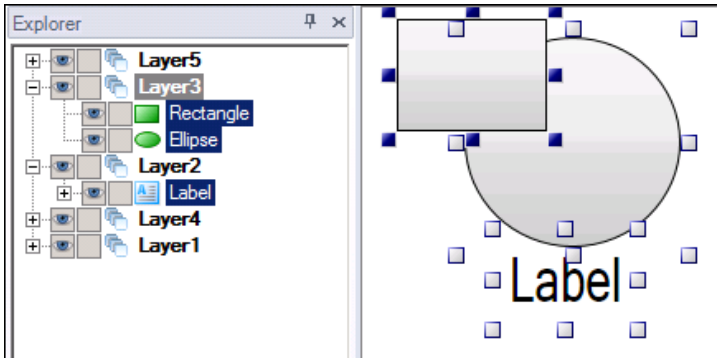


Figure 16.35 – Objects in Layers

Figure 16.36 shows a layer's properties in the Properties explorer. Notice the properties in the **Declutter** section. This set of options allows you to hide or show the selected layer in runtime depending on the zoom level being used by the operator.

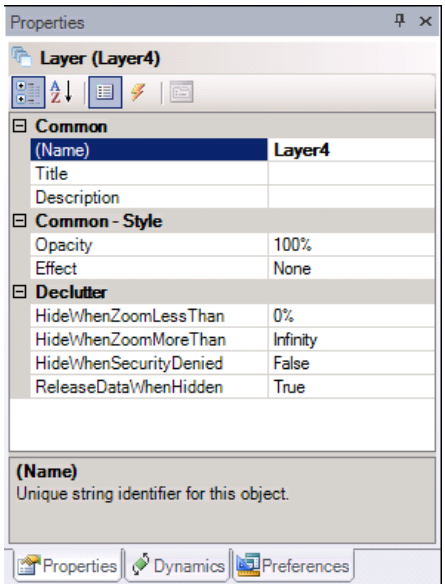


Figure 16.36 - Properties of a Layer

Lab: Security L16

16.1 Lab Overview

Many companies require security to prevent outsiders from looking at and controlling their displays as well as give different permissions to different people within the company. IcTonics is no exception! They are looking for additional security features as well. In this lab, we will secure the icTonics project using secured layers.

Estimated time to complete this lab: 20 minutes

16.1.1 Task Checklist

In this lab, you will complete the following tasks:

- Create a secured layer
- Create a popup display to view trends
- Define users
- Create a security group
- Add users to a security group
- Configure application actions
- Configure a secured layer

16.1.2 Support Files

To complete this lab, you will need the following file(s):

- icTonicsHome.gdfx
- icTonicsTemplate.tdfx

16.2 Security Users

On the icTonics production floor, there are operators and administrators. The administrators can do anything they want. However, the operators are not allowed to configure applications or close and exit runtime in the GraphWorX64 display. In this section, we will add an administrator that can do everything as well as an operator who has limited permission on the system.

1. Open the Workbench if you have closed it from the previous lab.
2. In the Project Explorer, click on the "Security Server"  icon from the tray.
3. Expand the localhost, right-click on Users and select "User" to add one.
4. In the Properties tab, give the user a name of "Administrator" and set "admin" as its password and click on "Apply".
5. You will see that your configuration is now grayed-out. This is because security immediately applies once you add a user.
6. Now you are going to log into the system as the Administrator. Go to the File menu and select Log In\Log Out → Server Login.
7. This brings up the **Security Login** dialog box. To log in, enter the user name and password that you just created for the administrator.
8. Now add another user and call this user "Operator" and use "operator" as the password.
9. Select the **Application Actions** tab and uncheck "Workbench" to limit the operator from coming into the Workbench to change configurations.
10. Expand the GraphWorX64 section, and uncheck "Exit Application" so that the operator cannot close a runtime GraphWorX64 display.
11. Expand the "Menu" subtree and uncheck the "Exit Runtime" option as well.
12. Go to the Files tab. In the **Deny These Operations** section, type "*.*|SecuredLayer" – remember to include the pipe character (|) before the layer's name. Leave the checkmarks in the Read, Create, and Modify checkboxes. As a result, when you log in as the user called Operator, you will not see the layer called SecuredLayer.
13. Click on "Apply" to save the configuration.

14. Open your icTonicsHome display. In the display, make sure the "Tank Monitor" button is on the SecuredLayer. Make sure you do not add the Login button, though. Save the display.

15. Now log out by going to File → Log In/Log Out → Server Login, and in the dialog box that appears click on the "Logout" button.

16. Confirm that you are logged out by looking at the status in the bottom-right corner of the Workbench. (When a GraphWorX64 display is open, look in the GraphWorX64 status at the bottom of the display.) It should look like this: Not Logged In. If instead you see a user ID there, log out again.

17. Go into runtime while you are not logged in. You should see an Access Denied screen.

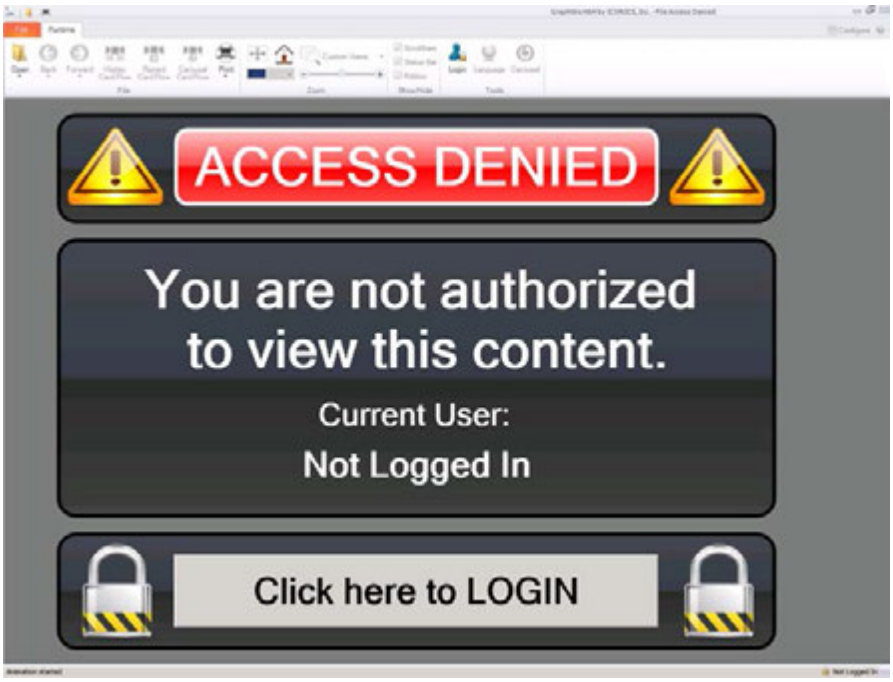


Figure L16.1 - Access Denied

18. Click on this screen's Login button and log in as Operator. You should see only the Welcome message, the language buttons, and the Login button.

19. Notice that when you are logged in as Operator, you cannot exit the application because the configured button is grayed-out.

20. Log in again by clicking on the Login button, and login as Administrator. You should now see your other navigation buttons on the bottom of the screen. You should also be able to go back into configuration mode.

16.3 Security Groups

It is possible that you will have more than one operator, so you will need logins for each user. However, configuring security for each user individually would be very inefficient. In this section, we will create a user group, configure security settings for the group, and associate users with this group. The users will then inherit the group's permissions.

1. Return to the Security Server. In the Project Explorer, right-click on Groups and select "+ Group" to add a new group.
2. In the New Group configuration, give the group the name "Operators".
3. In the **Application Actions** tab, give this group the same action permissions that you gave to the user called Operator. (Do not configure the file permissions that you gave to Operator.)
4. Click on "Apply" to save the configuration.
5. Create a new user call "Operator2" and use "operator" as the password again.
6. Go to the **Application Actions** tab and click on the "Unselect All" button so that there are no application permissions here.
7. Click on "Apply". You will get a message giving you a warning that the user will not have any application permissions. Click on "Yes" to acknowledge the message.
8. Right-click on the Operators group in the Project Explorer and select "Membership"; this will open the **Group Properties** dialog box.
9. Click on the "Add" button, select Operator and Operator2 in the **Select Users** dialog box and click on "OK".
10. Exit the Group Properties membership dialog box. You can confirm that the two users are in the Operators group by right-clicking on each of the users in the Project Explorer and selecting "Membership". The dialog box that comes up should show you that the user belongs to the Operators group.
11. To test this functionality, you will need to log out. You can go ahead and do this now; (if you are still logged in as both users, you will have to log out twice).
12. Open the icTonicsHome display and go into runtime where you should be blocked by the Access Denied display because you are not logged in. Click on the button to log in. In the Security Login dialog box, choose Operator2 and type in the password, and click on "Log In".

13. You should then see the display as you would if you were logged in as Administrator, except you cannot exit the display.

14. Log in as Operator again. Even though we did not deny access to the Operators group for the secured layer, the user inherits the permissions from the group. Therefore, the user called Operator still cannot see the layer. That is because we specifically denied this access for the Operator.

Review: Security *R16*

16.1 Objectives Review

You should now know and understand the following topics:

- Create an Administrative user in GENESIS64 Security Server
- Create standard user accounts for use with GENESIS64 Security
- Set up security groups and users
- Create account policies
- Set security privileges to define what a user can do
- Log in and log out from the system
- Use Active Directory accounts

16.2 Summary

GENESIS64 Security System allows you to define users, groups, and policies in order to determine which users are allowed to use the system, and what they can do in it.

To set up an efficient Security System you first need to create an Administrator account that you can use for further security configuration of the system. Once you create it, you can define additional users and groups for additional employees who need to access the system.

Security is created by adding account policies, user accounts, and groups of user accounts to the system.

An account policy is a set of rules that you can apply to users and groups. This affects aspects like password expiration, account locking, password strength, and so on. You can also apply restrictions based on different parameters (for example, time-based parameters) to enforce access to some specific resources. You can create multiple account policies that are used with various users and group, but each user's and group's security is enforced by only one account policy at a time.

You also define permissions for users and groups. Permissions allow or deny the user or group access to actions in individual applications, points, alarms, files, stations, methods, assets and operations.

The GENESIS64 Security System also includes advanced features that you may find useful when working with GraphWorX64 and the other client applications. You can, for example, prevent a user from visualizing a layer. You can also use a special syntax to query the security status of a resource like an OPC point. Using this feature, you can hide individual objects, like a 3D element in a 3D View control.

16.3 Terms

Account Policy	Critical Alarm	Critical Point
Membership	Regular Expression	Security Group
Security Permission	Security Server	Security User
Time-based Restriction		

16.4 Test Your Knowledge

In this chapter, we discussed how to configure and use GENESIS64 Security Server. Before continuing to the next chapter, make sure you can answer the questions below:

1. In which application can GENESIS64 Security be configured?

   a. None; you must use SQL Server
   b. Workbench
   c. GraphWorX64
   d. A command box

2. What special account must you create before you can configure your Security System?

   a. A highly-secured group account
   b. The start-up account
   c. An Administrator account
   d. No special account is needed

3. What do you need to edit if you want to enforce a specific password format for a user?

   a. The account policy used for that user
   b. The group account used for that user
   c. The user's account
   d. The account policy used for the user's group

4. How do you alter the number of failed login attempts that are allowed?

   a. On the user account, set the number in the Lock account after failed login (N attempts) field
   b. On the global settings, set the number in the Lock account after failed login (N attempts) field and check the checkbox
   c. On the account policy, set the number in the Lock account after failed login (N attempts) field and check the checkbox
   d. You cannot change it; the number of login attempts allowed is always 3

5. What can be used to enforce a hard-to-crack password?

   a. You must use scripting
   b. Global Security settings provide a random password mask
   c. You must use third-party password tooling
   d. Use Regular Expressions to define a password format

6. Is it possible to log in with the current Windows user, rather than having to manually log in?

   a. No, all users must log in manually
   b. Yes, use the Automatically login the current Windows user option in the General tab of the Global Settings
   c. Yes, users can click the Automatic login button in the Workbench
   d. Both B and C are available

7. Which of the following statements about Application Actions are true?

   a. Application actions are organized hierarchically.
   b. There is an application action for each GENESIS64 product
   c. Each GENESIS64 product has a hierarchy of actions that you can control
   d. Application actions can be set for groups and for individual users
   e. All of the above
   f. None of the above

8. What is the name of the property that can be used to secure a layer inside a GraphWorX64 display?

   a. Hide When Security Denied
   b. Hide Data
   c. Show Data Layer
   d. Hide Layer

9. What does the ?DataPoint:*TAGNAME* feature do?

   a. Queries the account policy for user access
   b. Redirects data to a user who has access to an OPC point
   c. Temporarily allows write permissions
   d. Queries the user's security permissions for an OPC point

GraphWorX64 Shortcuts

Shortcut Key	Function	Config Mode	Runtime Mode
F1	Help	✓	
F4	Properties window	✓	
Alt + F4	Close a display	✓	
Alt + F11	Script editor	✓	
Alt + HA	Arrange menu	✓	
Alt + C (Runtime)	Configure mode		✓
Alt + C (Config)	Controls Menu		
Alt + D	Dynamics menu	✓	
Alt + E	Design menu	✓	
Alt + F	File menu	✓	✓
Alt + H	Home menu	✓	✓
Alt + O	Open menu		✓
Alt + P	Print menu		✓
Alt + R	Runtime mode	✓	
Alt + T	Tools menu	✓	✓
Alt + V	View menu	✓	✓
Alt + Y	Dynamics menu	✓	
Ctrl + A	Select all objects	✓	
Ctrl + C	Copy	✓	
Ctrl + D	Duplicate selected object(s)	✓	
Ctrl + F	Find	✓	
Ctrl + G	Group into symbol	✓	
Ctrl + H	Replace	✓	
Ctrl + N	New file	✓	
Ctrl + M	Toggle between runtime mode and configuration mode	✓	✓
Ctrl + O	Open file	✓	✓
Ctrl + P	Print file	✓	✓
Ctrl + R	Runtime Window Properties Mode	✓	
Ctrl + S	Save file	✓	
Ctrl + T	Toggle toolbars	✓	
Ctrl + U	Ungroup symbol	✓	
Ctrl + V	Paste object	✓	
Ctrl + W	Show whole display	✓	✓
Ctrl + X	Cut object	✓	
Ctrl + Z	Undo	✓	
Ctrl +]	Bring forward	✓	
Ctrl + [	Send backward	✓	

Shortcut Key	Function	Config Mode	Runtime Mode
Ctrl + Numpad +	Zoom in	✓	✓
Ctrl + Numpad –	Zoom out	✓	✓
Ctrl + Alt + C	Set current active layer	✓	
Ctrl + Shift + A	Space evenly across	✓	
Ctrl + Shift + B	Align bottoms	✓	
Ctrl + Shift + C	Align centers	✓	
Ctrl + Shift + D	Space evenly down	✓	
Ctrl + Shift + H	Make same height	✓	
Ctrl + Shift + L	Align lefts	✓	
Ctrl + Shift + M	Align middles	✓	
Ctrl + Shift + O	Make same height and width	✓	
Ctrl + Shift + G	Snap to grid	✓	
Ctrl + Shift + R	Align rights	✓	
Ctrl + Shift + T	Align tops	✓	
Ctrl + Shift + U	Undo zoom	✓	✓
Ctrl + Shift + W	Make same width	✓	
Ctrl + Shift +]	Bring to front	✓	
Ctrl + Shift + [	Bring to back	✓	
Shift + resize	Preserve aspect ratio while resizing an object	✓	
Shift + object creation	Create an object with equal width and height	✓	

Delimiters and Syntaxes

Beginning Delimiter	**End Delimiter**	**Function**
{{	}}	OPC Tag delimiters in an expression
~~	~~	Local Variable delimiters
lic:		License FrameWorX Plug-in tags
sim64:		Simulation FrameWorX Plug-in tags
localism:		Simulation tags from GraphWorX64
rdcy64:		Redundancy FrameWorX Plug-in tags
bacnet64:		ICONICS Bacnet OPC Server FrameWorX Plug-in tags
snmp64:		ICONICS SNMP Server FrameWorX Plug-in tags
exp64:		Global Expression FrameWorX Plug-in tags
grp64:		Group FrameWorX Plug-in tags
vset64:		Value Set FrameWorX Plug-in tags
rgs64:		UDM Register FrameWorX Plug-in tags
<#	#>	Global Aliasing delimiters
/+	+/	Language Aliasing delimiters
“	”	String literal in a process point
<<	>>	Local aliasing delimiters
A:AliasName		Allows dynamic global aliasing definition. The Complete syntax is *A:AliasName=“AliasValue”/ThemeScope;*. ThemeScope can be, but not limited to the following: Not defined – Relative to current document level 0 0 – Machine Level 1 – Process Level 2 – Document Level 0 3 – Document Level 1 +1 – Relative to current document level 1 +2 – Relative to current document level 2

Glossary

3D Visualization is a three-dimensional rendering of a GraphWorX64 display or TrendWorX64 trend.

Account Policy is a set of rules that define login and logout policies and *Time-based Restrictions* for set of users and/or groups.

Address Space is an organizational scheme for objects. An address space is used in OPC UA for objects, properties, and relationships.

Alarm Area, or Area for short, is a logical folder grouping option that allows you to categorize and organize your alarms. You can have any number of areas and subareas. You can also associate alarms to any number of areas as you wish. Areas can be used for filtering alarms.

Alarm Filter is a rule that controls which alarms display in the viewer in runtime and how they display.

Alarm Indication Dictionary contains a set of conditions along with symbols or icon to display when the condition for the alarm becomes true.

Alarm Logger Columns are defined within an *Alarm Logger Configuration*. A column represents an alarm attribute that you would like the log into the database. The name of the column is the name of the table field.

Alarm Logger Configuration stores information such as the subscription the alarm and/or event server and the database to which you want to log the alarm information. You can configure as many alarm logger configuration as you please, but you must add it under an *Alarm Logger Node* to make the configuration active. This should not be confused with the Alarm Logger Configuration Database. You can only have one *active* alarm logger configuration database at a time.

Alarm Logger Node refers to a computer where you want to use the *Alarm Logger Configurations*. The node is typically the local computer name or IP address where the AlarmWorX64 Logger is installed.

Alarm State Value is used by digital alarms to identify the state that will trigger an alarm. Since the alarm is digital, the two values that this state can have is 1 or 0.

Alarm Subscription is the connection between the alarm client and the alarm server. This determines which alarms the server will send to the client.

Alarm Tag is an OPC AE tag that defines an alarm. You can associate an OPC DA or OPC UA tag to monitor the values. Inside this tag, you can set up the different alarm types.

Alarm Template is optionally used as a basis for creating and maintainin alarm tags. Instead of modifying each OPC alarm tag individually, you can maintain the alarms uniformly from the alarm template. In the configuration form of an alarm tag created from a template, values that the alarm template populates are highlighted in pink. An alarm template must be used when using the *Tag Import Wizard* to create multiple alarm tags at a time.

AlarmWorX64 is a set of applications that can be used to monitor device conditions and events. The applications are the Alarm Server, Alarm Logger, and Alarm Viewer. Once alarms are configured in the server, you can log the data using the Alarm Logger or view it through the AlarmWorX64 viewer. The AlarmWorX64 viewer can run within the Workbench, or as a viewer within GraphWorX64.

AlarmWorX64 Logger subscribes to an OPC Alarm and Events server and stores data in a database, providing a permanent copy of alarm and event notifications.

AlarmWorX64 Server is the application used to generate alarms notifications. It receives OPC complaint DA or UA data and generates alarms based on configured information. It sends generated alarms to any subscribed clients.

AlarmWorX64 Viewer is a client application that subscribes to OPC UA or OPC AE serves. The viewer allows you to view alarms and events based on the *Alarm Subscription* and properties set in the viewer.

Alias Theme is a type of classification that links a global alias to one or more data values and bundle Global Aliases together.

Alias Value is a predefined value for a *Global Alias* that can resolve to during runtime when the correct theme is set.

Alignment is a way to organize display objects along a grid line in GraphWorX64.

Application Mode is the choice of mode in which GraphWorX64 configuration is done. The three modes work in tandem with an engineer's familiarity with GraphWorX64. They are: Basic mode (beginner level), Standard mode (appropriate for development of most applications), and Advanced mode (all available options are exposed to the developer).

AssetWorX is an application providing asset organization and logical hierarchy to GENESIS64. Along with GraphWorX64 it is a primary source for executing commands. AssetWorX is an ISA-95 compliant asset structure offered in conjunction or independent of standard tag based designs.

Chart is a container that holds one or more plots. Each tab in the TrendWorX64 Viewer can have any number of charts in it.

Client-Side Filtering is a type of alarm filter. You can create client-side filtering from OPC AE clients to limit the amount of alarm data going to the destination. Client-side filtering does not limit the server from sending the alarms to the alarm client.

Configuration Mode is a design mode for GENESIS64 in which displays are built. This mode can display sample data, but not live or historical data.

Content Area is the location where you can create, edit, and display different files in the Workbench. This area can span a configuration of multiple monitors.

Conversion Alias is a type of language alias. This type of alias converts one type of measurement unit to another. The final real-time value depends on *Unit Bindings* and *Unit Conversions* defined in the *Language Alias* configuration.

Critical Alarm is defined in the *Security Server*. When acknowledging *Critical Alarms* you must log in again and acknowledge the alarms within a defined period.

Critical Point is defined in the *Security Server*. When you try to write to this point, you must log in again and write to the point within a defined period.

Database Group is a hierarchical grouping in the TrendWorX64 configuration that allows you to connect to a database for storing data. You can also optionally connect to a secondary database for redundancy.

Data Aggregation is a feature of *UDM Registers*, which pools together OPC tag requests from clients to reduce I/O traffic to a hardware device.

Data Bridging is a feature of *UDM Registers*, which sends a device' s output to another device.

Data Collection Rate is the rate at which samples are collected (read) from the OPC server.

Data Entry Object is a *Process Point Object* that allows operators to enter data that can be written to the process point during runtime.

Data Prefetch is a feature of *UDM Registers*, which polls and collects output from a hardware device in anticipation of incoming client requests.

De-cluttering is a method of showing or hiding details of a display based on the zoom level using *Layers*.

Deviation Alarm is one of the alarm types. It uses two inputs and takes the absolute value of the difference between these two inputs. Alarms are generated when this difference exceeds the Hi and HiHi limit or falls below the Lo and LoLo limit.

Digital Alarm is one of the alarm types. It tags a Boolean tag or expression that evaluates to either 1 or 0 and triggers an alarm based on the *Alarm State Value* you entered.

Display Refresh Rate is the rate at which the TrendWorX64 Viewer updates the viewer with new data. This rate along with the *Data Collection Rate* determines how much data is in the viewer buffer.

Distributed Control System (DCS) is a control system whereby the controllers are geographically distributed and independent. Data from these remote devices are usually collected by a central server.

Dynamic is an action property of an object that makes the object actionable or changeable. Dynamic actions include dimension, color, selectors, and other actions available in GraphWorX64.

Dynamics Explorer is the section of the Workbench where the properties of a dynamic can be modified.

EarthWorX64 is a geographical asset management application based on Bing technology. EarthWorX64 can run within the Workbench, or as a viewer within GraphWorX64.

FrameWorX64 is the ICONICS GENESIS64 OPC UA server. All communications between client applications and OPC data sources go through the FrameWorX64 server.

Freeze Mode is a runtime mode in which an operator has paused the TrendWorX64 Viewer to stop the scroll of the display's incoming data. When the operator ends Freeze mode, the data that has accumulated appears in the graph. Some Viewer options are available only in Freeze mode.

Function Key Action is a *Pick action* that applies only to the This Display object. At runtime, the action is triggered by the runtime operator pressing a function key.

GENESIS64 Controls is a set of controls in GraphWorX64 that allows you to create and view other data. The viewer controls include: AlarmWorX64, TrendWorX64, GraphWorX64, and EarthWorX64. Tool controls include: the Smart Pin, PushPin, Scale Control, and Pipe Control.

GenBroker is a service that uses COM/DCOM to communicate with GENESIS32 clients for backward compatibility.

GenEvent is an event generated by ICONICS. The event can be, but is not limited to, a write to OPC tag or a startup of a service.

GenTray is an application used to configure the *GenBroker* service that communicates with legacy OPC data sources.

Global Alias is a reusable variable that can have one or more values. A global alias is configured and stored in the Global Aliasing System (GAS) database is available for use with all GENESIS64 applications and can be assigned to a theme. A global alias created dynamically in a GENESIS64 file is available only in that file and does not have a theme.

GraphWorX64 is the GENESIS64 display technology that allows you to build *HMI* or *SCADA* GUIs (Graphical User Interfaces). GraphWorX64 is the lone GENESIS64 application that can run as a standalone application outside of the Workbench.

GraphWorX64 Content Area is the defined work surface where you can add GraphWorX64 objects. In this area, you can build your *HMI* display.

GraphWorX64 Template is a means for creating a uniform appearance among the displays that you create. A GraphWorX64 template has a file extension of .tdfx.

HMI stands for Human Machine Interface, a GUI display for automation and control software.

History Refresh Rate is similar to the *Data Refresh Rate*. However, instead of refreshing and updating the viewer with real-time data, the *History Refresh Rate* determines how often the viewer is updated with historic data.

Home Page is a dedicated space of GENESIS64 Workbench's content area reserved for a web-based display of your choosing. You can incorporate contents such as RSS feed or a web site into this page. It can be launched at Workbench startup or configured into a layout.

Hyper Historian is a high speed data logger that can store data to disk for later analysis.

Ideal Pen is a pen that allows for comparisons of past historical data to current historical data for the same historical pen.

Key Performance Indicator (KPI) is a set of variables used to determine the success of an operation. For example, managers may use KPIs to indicate if the resources in a plant floor are used to maximum productivity.

Language Alias is used for displays that have a trans-lingual requirement for users. At runtime, the language alias substitutes a string in one language for a string in another using the Language Aliasing System (LAS). You must predefine the languages, the strings you want available, and the language-specific translation of each string.

Layers are levels of a display that you can arrange objects into. You can hide or show layers under specified security or zoom conditions.

Legend is the key for a trend display. The legend includes data points and their marker colors in the display, and can be shown or hidden from view.

Limit Alarm is a type of alarm. When configured, it can generate alarms based on thresholds that you configure. The thresholds for this alarm type are LoLo, Lo, Hi, and HiHi. Values at LoLo or below generates a LoLo alarm; values between LoLo and Lo, including Lo generates a Lo alarm; values between lo and Hi (exclusive) are considered normal; values at Hi or above, but below HiHi generates a Hi alarm; values at HiHi or above generates a HiHi alarm.

Logging Group defines the table structure for the logged data. You can define the data collection rate, logging rate, and manage the database tables.

Markers are optional symbols displayed at each data point in a trend display.

Membership refers to the relationship between security groups and users. You extend *Security Group* membership to *Security Users* by adding users to the group.

Node Pairs refers to a primary server and its stand-by server when paired for the purpose of *Redundancy*.

Object Explorer, also known as the "Explorer" or "Project Explorer", is the panel inside GraphWorX64 that lists all objects, including hidden ones, in the currently open display. It is easier to select objects in a group if you select them in the Object Explorer rather than in the content area.

OPC AE stands for OPC Alarms and Events, which is used to monitor devices for conditions and provide a reporting function.

OPC Client can receive information from OPC Server if both use the OPC Foundation standards. GraphWorX64, AlarmWorX64, TrendWorX64, etc., are all examples of OPC Client.

OPC DA is OPC Data Access, which is used for reading real-time data.

OPC HDA is OPC Historical Data Access, which is a format that allows you to log data into data files either held in memory or written to disk as database files so that the data can be replayed or analyzed at a later time.

OPC Server can communicate with OPC devices through PLC or other I/O channels and can format the data into something that an OPC client can work with.

OPC stands for OLE for Process Control, which is a standard for connection and communicating with industrial automation devices based on the specification form OPC Foundation (OPC.org).

Pack and Go Wizard copies a project's resources into a file, making the project portable. The wizard can compress the project and encrypt the project.

Pens are the points that put values on a trend chart or graph. A pen is the connection to the trend display from the OPC UA, OPC DA, or OPC HDA data source.

Plot and Plot Type are a set of pens and the type of chart they will draw in a trend display.

Pick Action is a type of *Dynamic* used to assign a command to a shape or object that can open a display or a popup, download a value to an OPC Tag, run a script, or perform one of many other actions as the result of runtime user interaction with the object.

Pipe Control makes it easy to create pipes and elaborate plumbing schemes in a GraphWorX 2D display. You can easily animate an object so that, during runtime, it moves along the path of a 2D pipe.

Pivot Control allows the creation of simple, elegant menus in the ModernUI style. Each pivot page is a GraphWorX64 canvas allowing any control, object or dynamic to be placed inside. The Pivot Control supports click, touch, or swipe to navigate.

PLC stands for Programmable Logic Controllers, which are programmable devices used to provide industrial automation controls. OPC servers can connect to these devices to obtain human readable data.

Preferences Explorer is where you configure default settings and behaviors for display objects in GraphWorX64.

Printer Logging is an optional way of sending alarm events to a printer so that a written record is made of the log files.

Process Point Object or generally called a Process Point, is a dynamic that connects to an OPC DA or OPC UA data source to display the value inside a GraphWorX64 display.

Project is the founding concept of the GENESIS64 Workbench. A project is the set of applications and related resources (such as database configurations, services, and project files) that belong to and should be available to a single project inside the Workbench.

Project Explorer is the navigational component of the GENESIS64 Workbench. You can use the Project Explorer to navigation to different applications within the GENESIS64 Suite.

Project File is the wrapper for a set of displays and settings in GENESIS64 that can be opened and worked with as a set.

Properties Explorer allows you to configure properties of each object on your GraphWorX64 display. If there is no object selected, the properties shown are the properties for the This Display object.

PushPin Control is a simple shape with complex features, which you can add to an EarthWorX map or a GraphWorX display for identifying locations or assets.

Rate Limit Alarm is much like the Rate of Change alarm except that the change limit is not an absolute number – instead, it is expressed in terms of a positive or negative value.

Rate of Change Alarm is one of the alarm types. This type of alarm is generated when the OPC value changes faster than the rate you configure. The rate is calculated by taking the absolute value of the difference between the last two readings of an OPC tag, divided by the change in time (timestamps) from the last two reading. The rate you enter is the Limit per Second rate.

Reciprocal Conversion is a unit conversion that expressions an additional way of converting a *Unit Conversion* that you just defined. For example, after you define a unit conversion from Celsius to Fahrenheit, you must define its reciprocal converstions which are: Fahrenheit to Celsius, Celsius to Celsius, and Fahrenheit to Fahrenheit.

Redundancy is a means for establishing a parallel backup database to act as a stand-by for the primary database where data is actively being logged. A redundant stand-by can become active if the primary database becomes unavailable, reducing the risk of losing data due to loss of database availability. Databases, loggers, clients, licenses, and security can all be safe-guarded using redundancy.

Reference Object is the last object selected among many. When you arrange the selected objects, all objects are arranged based on the properties of the reference object.

Regular Expression is a syntax that can be used to define the required format of a secured password string. This is a standard syntax whose details can be easily found on the internet.

Ribbon is synonymous to the old style application menu at the top of an application. The GENESIS64 Workbench has a ribbon at its top with different tabs that change depending on the application you are working in and the object you have selected.

Runtime Preview Mode allows you to view how the GraphWorX64 display would look like in runtime. However, since this is only a preview, you can still edit the display because you are still in *Configuration Mode*.

Runtime Mode, as opposed to Configuration Mode, allows for viewing live or historical data. You may not edit data or make configuration changes in this mode in viewers. Some ICONICS servers, such as the AlarmWorX64 server, allow for updates to the configuration while the service is running.

Runtime Ribbon lets runtime users zoom in and out, access views, browse through display history, hide objects in the display, log in or out, open a Carousel of displays, and more. A GraphWorX64 tab appears on the ribbon, but if the display has Viewer controls in it, additional tabs can provide additional EarthWorX64, AlarmWorX64, or TrendWorX64 tools, depending on the Viewer that the runtime user clicks on.

SCADA, which stands for Supervisory Control and Data Access, is a system that provides the means to observe and control automated systems.

Scale Control provides a shortcut for creating a GraphWorX64 object that has a dial or ruler, such as a gauge, a thermometer, a scale, a clock or watch, a meter, a speedometer, or a visual counter.

Security Group is any group defined in the *Security Server* configuration. A security group has its own permissions, and it exists solely to provide those permissions to the *Security Users* in its *Membership*.

Security Permission is the application access and data access granted to a *Security User* or *Security Group*.

Security Server defines the security configuration for and prevents unauthorized user access in GENESIS64 resources.

Security User is any user defined in the *Security Server* configuration. A security user has its own *Security Permissions*, and can have *Membership* in a *Security Group* that has overriding Security Permissions.

Server-Side Filtering is a type of alarm filter. You can use this type of filtering to limit the number of alarms that the alarm server sends to the client. This is most useful for reducing network traffic when the alarm server and the alarm clients are on different machines.

Service Oriented Architecture (SOA) is a software architecture whereby each user does not start the software application individually. Instead, the software is started and managed by the operating system as services and can serve multiple users concurrently.

Side Panel is a section of the TrendWorX64 Viewer that gives runtime users access to multiple tabs in the Viewer.

Signal Name is the OPC tag path that you choose to log trend data.

Smart Pin Control is a collection of squares called SmartIcons that form a set of indicators for use in a high-level GraphWorX64 overview display or EarthWorX64 map. Each square's location in the Smart Pin grid is meaningful. SmartIcons are strategically arranged and attached to data sources to create a shape and color scheme that allows runtime users to easily monitor the status of multiple data sources.

Summary View can be made visible at the bottom of the TrendWorX64 Viewer, only in Freeze mode. It provides access to all collected data, and has features that give the runtime user detailed, time-specific access to collected data.

Symbol is a saved grouping of GraphWorX64 objects.

Symbols Library is a collection of symbols within GraphWorX64 designed to help you quickly build a display.

Tabs can be used for displaying multiple TrendWorX64 charts in one view. Using tabs with the *Side Panel* in a viewer gives runtime users easy access to the trend graphs on each viewer tab.

Table Management is a means for limiting the size of the table that is actively logging alarms, for managing the creation of new logging tables, and for archiving older logged events.

Tag Import Wizard is a tool for the mass-creation of multiple alarm tags in AlarmWorX64 or trend tags in TrendWorX64. This tool is helpful when you have hundreds or even thousands of OPC tags that you need alarm or trend tags defined for, and defining them one at a time will be too time-consuming. The AlarmWorX64 wizard uses a predefined *Alarm Template* for defining the alarm tags it creates; the TrendWorX64 wizard does not.

Tags, also known as Process Points, are a named data source formatted into an OPC communications standard.

Text Object allows you to enter static text into the text area in a GraphWorX64 display. This is similar to a text box or label in MS Word.

Time-based Restriction is that part of an *Account Policy* that limits user access to applications and data to specified days and times.

Time Date Object is a dynamic text object that displays the current date and time in a GraphWorX64 display during runtime.

Transitions are special effects (such as a fade out) applied to a GraphWorX64 display as it closes or loses focus when another display is opened in the same window. It is similar to transitions in MS Power Point.

Translation Alias is a type of language alias that is replaced by the equivalent string defined in the Language Alias System (LAS) configuration.

TrendWorX64 is a set of applications used to log numerical data into memory or database for historical viewing or data analysis. Included in TrendWorX64 is a historical server for logging data and a graphing and charting module for viewing data. The TrendWorX64 Viewer can run within the Workbench, or as a control within GraphWorX64.

TrendWorX64 Configuration houses the database groups, logging groups, and data tags for TrendWorX64. You can have only one active configuration at a time.

TrendWorX64 Logger is the application that collects OPC data and stores the information into a database. The *TrendWorX64 Configuration* dictates how the data collection and logging occurs.

TrendWorX64 Viewer is the control that allows you to view real-time or historical trend data in different chart types.

Trigger Limit Alarm is a type of alarm. It reports when the tag's value changes, and does not reset until after a runtime operator acknowledges it. With this alarm, there is no "normal" condition; any change in the value triggers the alarm. This type of alarm is useful for a tag whose value rarely changes.

UDM Expression is a function that takes an input, performs a specified calculation, and returns an output.

UDM Group is a collection of data tags that are manipulated as a single item, which are represented by a single group tag.

UDM Register is a named address into which you can store values.

UDM Trigger is a rule that defines conditions under which an event occurs. There are two types of triggers: data triggers and time triggers.

UDM Value Set allows you to write pre-configured sets of values into a group of data tags.

Unified Data Browser is a utility that can browse OPC data sources, select them, and return their addresses. It is also known as the Data Browser.

Unified Data Manager (UDM) is a front end to a set of SQL databases that store information used in GENESIS64 applications, including expressions and local aliases.

Unit Binding is the connection between a unit conversion and the languages it is valid for. Unit binding is maintained in the Language Alias System (LAS) configuration.

Unit Conversion is the definition of an equation for converting one unit of measure to another. It uses expressions when converting measurement units. It is maintained in Language Alias System (LAS) configuration.

UTC Time stands for Universal Coordinated Time. When using this time, you can be worry free about time zones and daylight saving time.

Views are pre-set scenes or snapshots inside a complex or oversized GraphWorX64 display or EarthWorX64 map that help runtime users navigate the display with ease and convenience. Views can also be useful to engineers in configuration mode for quickly moving to otherwise difficult-to-find objects in the display.

Web Publishing Wizard is the ICONICS GraphWorX64 tool that publishes displays as standard HTML pages.

Window Layout stores specific arranged files for the content area in the Workbench. Opening a layout opens the designated files in a designated arrangement. You can look at it as a *Workbench Template* with specific files assigned in the different areas.

Workbench is a centralized location where you can manage, configure, and run all of GENESIS64's applications, and most of its tools.

Workbench Template stores arrangement information for use with any files in the content area of the Workbench. Opening a template arranges any open files to fit the template.

WPF stands for Windows Presentation Foundation. You can insert *WPF* objects into any GraphWorX64 display.

X-axis and Y-axis are the vertical and horizontal bases for charts in TrendWorX64 Viewer. The axes can be made 3-dimensional, and can be swapped in runtime.

Z-order, in GraphWorX64, is the sequence of objects, from front to back, on the display surface. The Z-order of objects can be changed with the help of the *Object Explorer*.

Test Your Knowledge - Answers

Questions: / Chapters:	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**
Chapter1	D	B	T	D	D									
Chapter2	T	T	C	B	A	F	T							
Chapter3	A	B	E	G	T	G	C	B						
Chapter4	T	T	T	T	T	T	F	T	F					
Chapter5	T	F	F	T	F	T	T	F	F	F	C	T	F	A
Chapter6	E	H	T	T	F	B	D	F	D	A	B	D	E	
Chapter7	E	F	F	T	T	F								
Chapter8	A	C	D	T	F									
Chapter9	B	C	E	F	E	T								
Chapter10	D	E	E	T	T	E	T	T	C	T	F	A		
Chapter11	B	D	T	T	E	D								
Chapter12	F	T	F	T	F									
Chapter13	A	A	T	A	A	C								
Chapter14	A	T	F	A	A	A	D	F	B	B	D	C		
Chapter15	T	F	T	F	G	E	T	T	F	G				
Chapter16	B	C	A	C	D	B	E	A	D					

T=True, F=False (except where multiple choice item).



100 Foxborough Blvd., Foxborough, MA 02035
508.543.8600 • FAX: 508.543.1503
www.iconics.com • E-mail: info@iconics.com

© 2013 ICONICS, Inc. All Rights Reserved.
Specifications subject to change without notice.