GENESIS64 Standard Training Manual

© 2018 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, AnalytiX, MobileHMI, BizViz and their respective modules, OPC-To-The-Core, and Visualize Your Enterprise are trademarks of ICONICS, Inc.

Windows, Windows 10, Windows 8, Windows 7, Windows Server 2012, Windows Server 2008, Windows Server 2008 R2, Windows 2003 Server, Windows 2003-64, Windows Vista, Windows XP, Windows XP-64, Windows NT, Windows 2000, and Windows CE 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-STD-10952-EN

Visualize Your Enterprise™

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.

Introduction to GENESIS64 1

ICONICS provides a real-time panoramic view of your operations. Whether 3D graphics or detailed data, visualization enables you with actionable intelligence. ICONICS takes into account your industry, scale and unique needs, and delivers you the best in automation, visualization and operational management.

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. 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. 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

1.1. Objectives

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

1.2. 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.

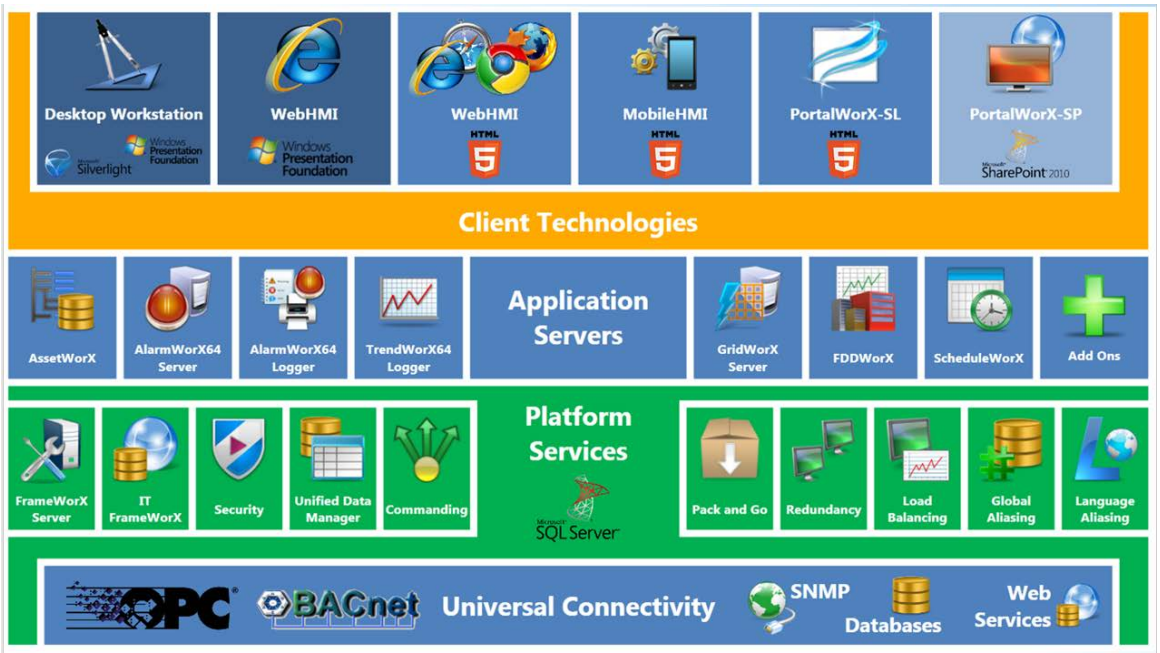


Figure 1.2 - Architecture

As you can 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.

1.2.1. 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, GraphWorX64 is shown in Runtime mode. You could also choose to open this display in a separate Configuration mode.

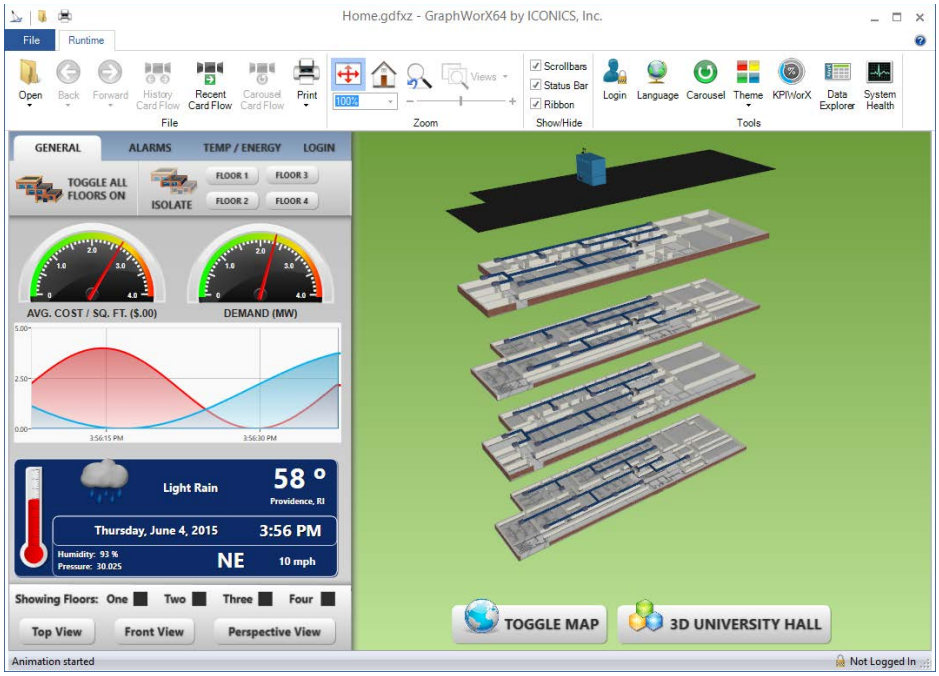


Figure 1.3 – GraphWorX64: Display in Runtime

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. An ESRI map is shown in Configuration mode within the work area in Figure 1.4.

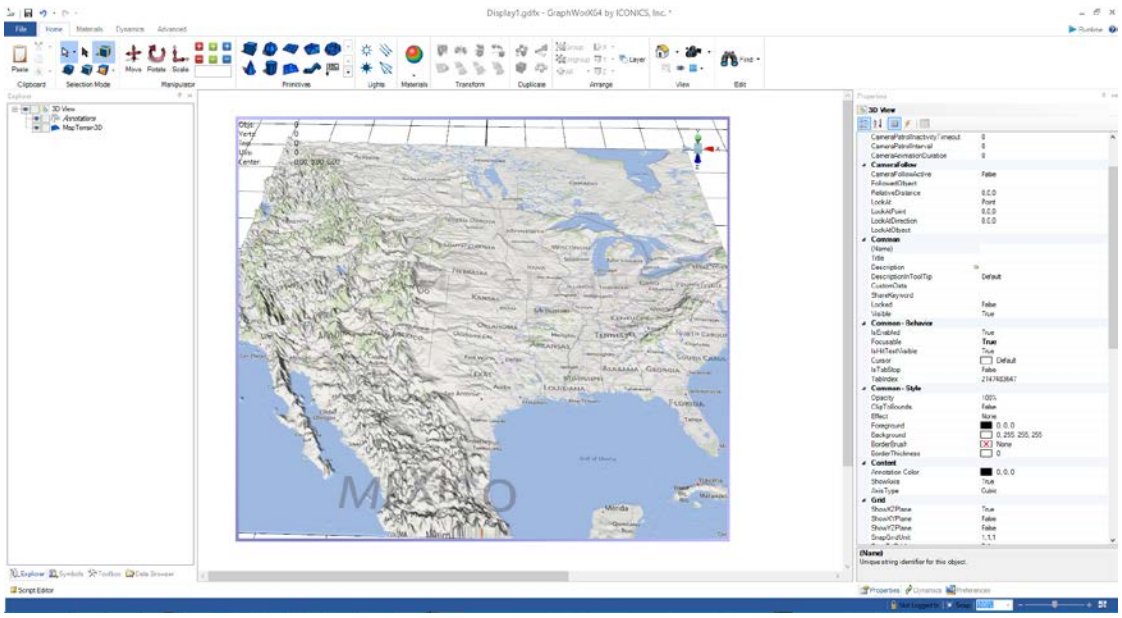


Figure 1.4 – GraphWorX64: ESRI Map

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.

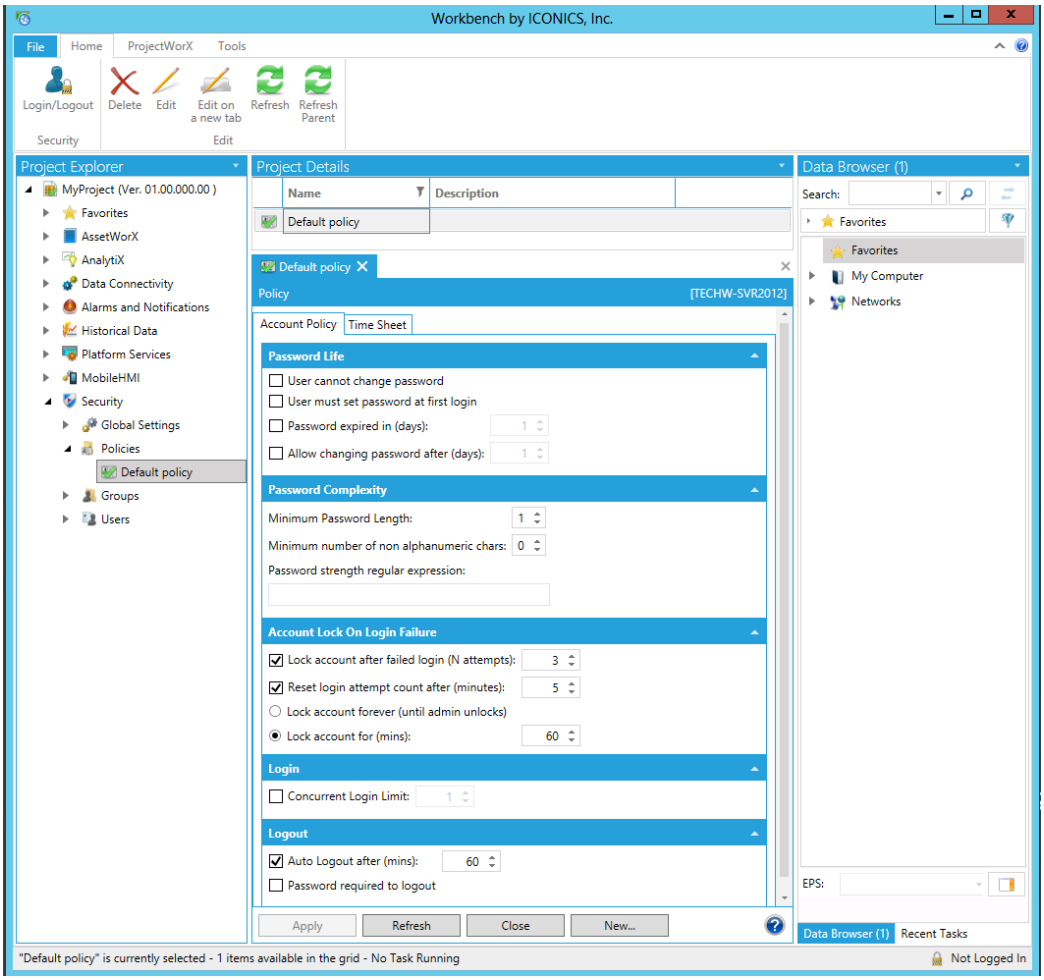


Figure 1.5 - The Security Configuration

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.

1.3. ICONICS System Options

Users have different options in installing a comprehensive ICONICS system solution, including Simple Visualization, Normalized Visualization, and Visualization with Calculations.

Simple Visualization

The Simple Visualization, as pictured in the following figure, option entails data connections from multiple sources passing through ICONICS FrameWorX Servers then, utilizing users' preferred communications technologies (such as BACnet or Modbus Ethernet, for example), providing data to visualization dashboards.

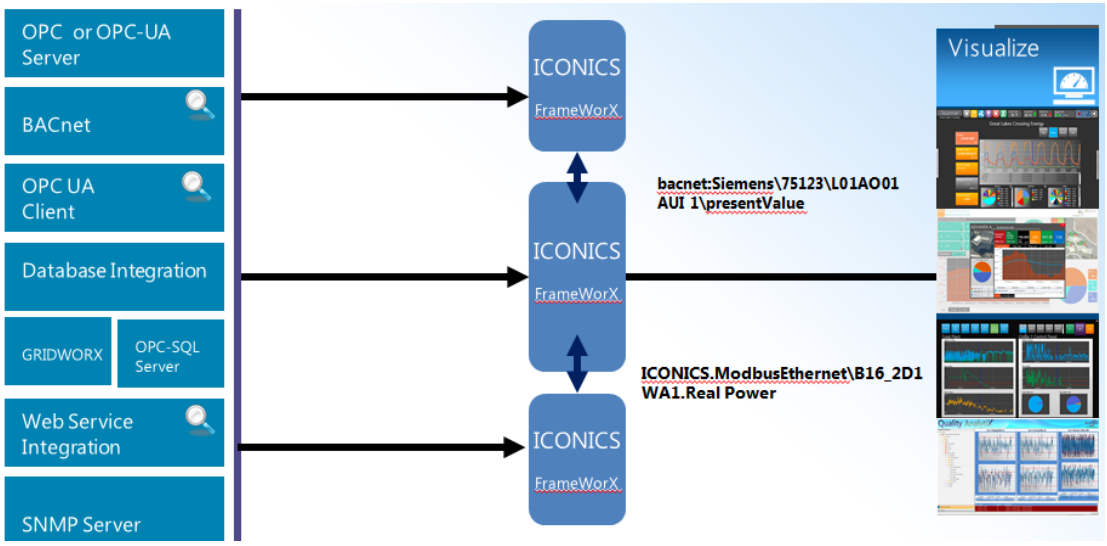


Figure 1.6 - Simple Visualization Option

Normalized Visualization

The Normalized Visualization option, as pictured in the following figure, entails data connections from multiple sources passing through ICONICS FrameWorX Servers, similar to the Simple Visualization option. However, this data can then be branched through an ISA-95-compliant, asset-based utility such as ICONICS AssetWorX, as well

as through a more traditional, Windows explorer-based utility such as ICONICS' Unified Data Manager tool. The benefits of both these approaches will be discussed in this training. The data from either branch then re-enters ICONICS FrameWorX Server to then proceed to visualization dashboards.

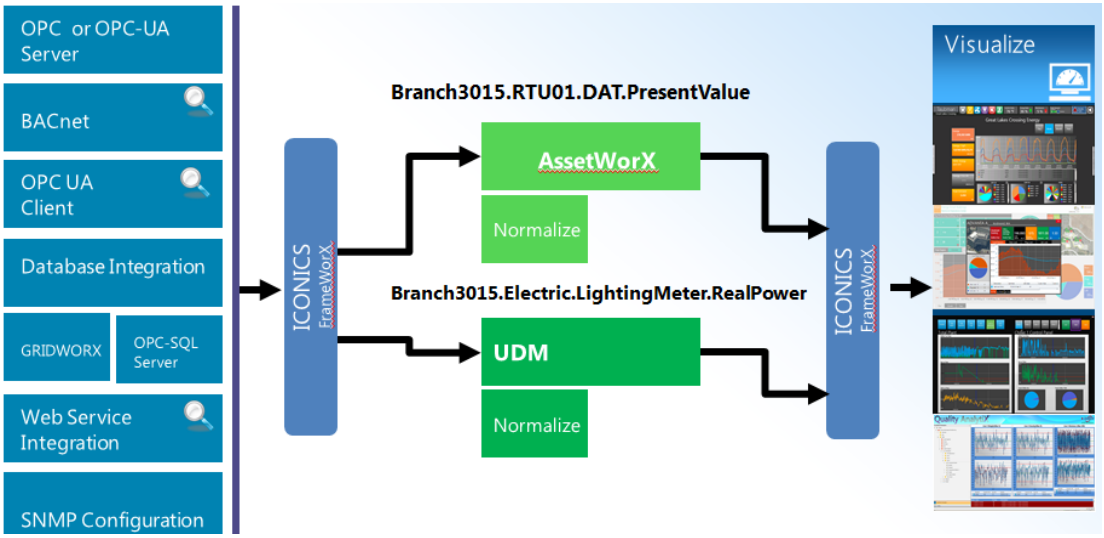


Figure 1.7 - Normalized Visualization Option

Visualization with Calculations

The Visualization with Calculations option, as pictured in the following figure, uses the same concept as the Normalized option, but includes the ability to perform calculations with the data using ICONICS Expression Engine. Of course, users can still opt to send the data directly, without calculations, as noted below.

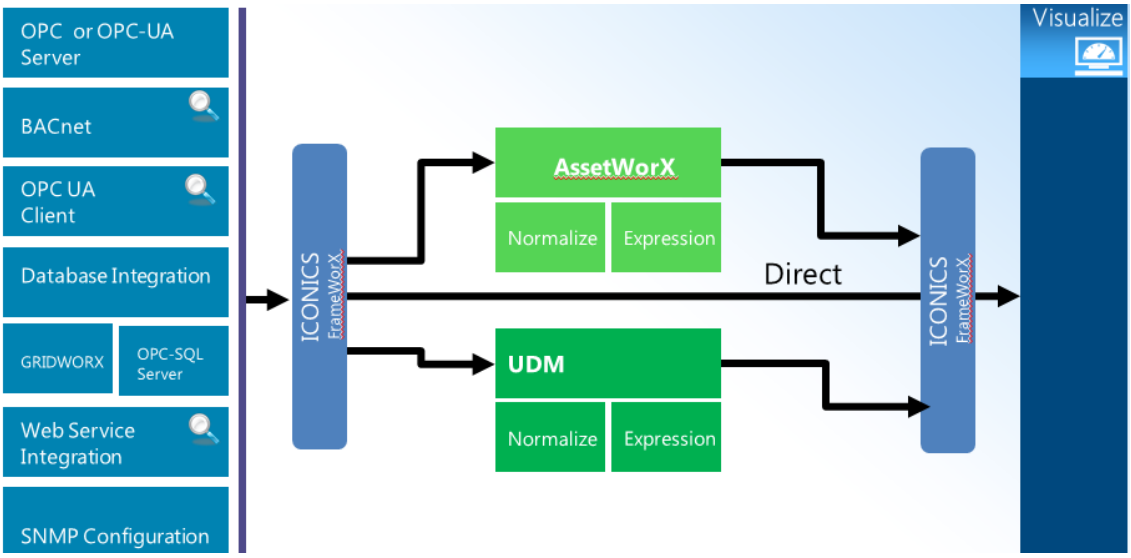


Figure 1.8 - Visualization with Calculations Option

Additional Choices and Applications

Additional choices are available depending on which approach you select. For instance, if you decide upon a Normalized Visualization or Visualization with Calculations option, you could add a Fault Detection and Diagnosis and/or Energy Analysis component to your asset-based (incorporating AssetWorX technology) approach, as shown below.

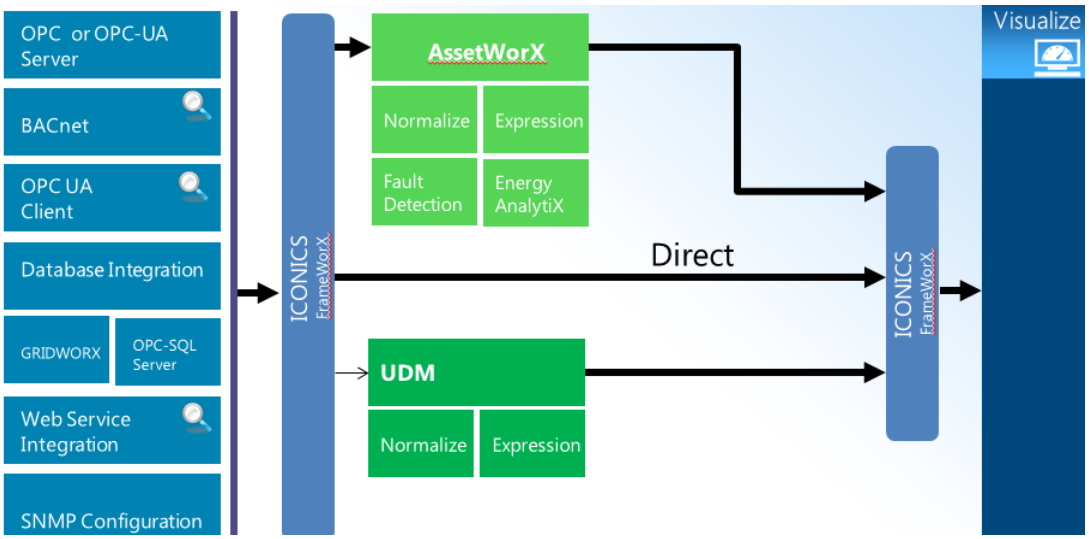


Figure 1.9 - Additional Choices for Normalized Visualization or Visualization with Calculations Options

ICONICS provides additional customization via add-on components such as an Alarm Server, Alarm Logger, Hyper Historian, Performance Calculations and SPC, SQL-integration, Productivity Analysis, User Security, Global Alias Server, Language Alias Server and more.

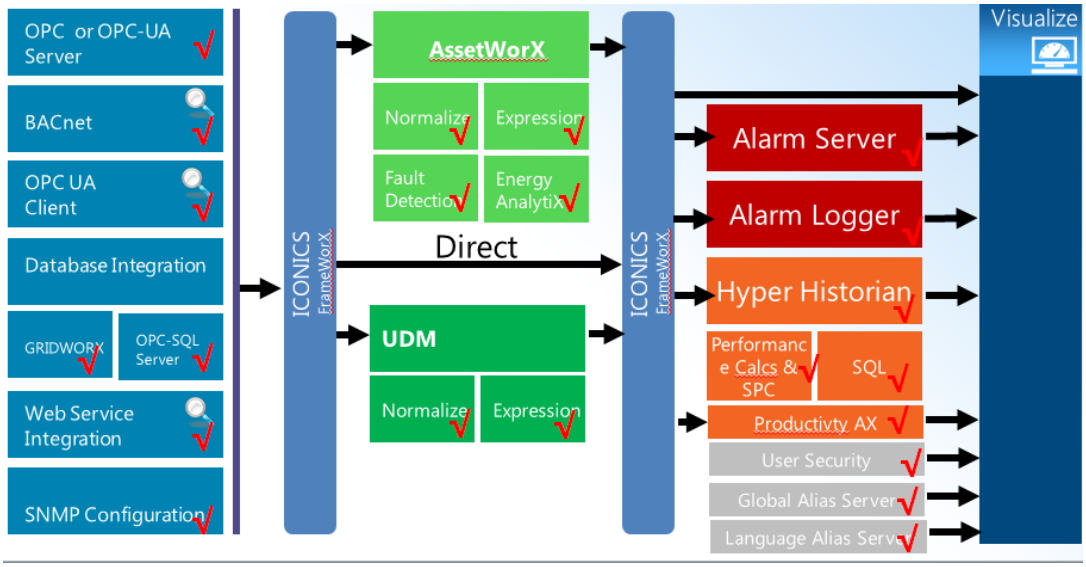


Figure 1.10 - Additional Applications

1.4. PLCs, OPC, BACnet, SNMP, Web Data 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 1.11 has the property that, at a constant voltage, the resistance measured changes linearly with temperature.

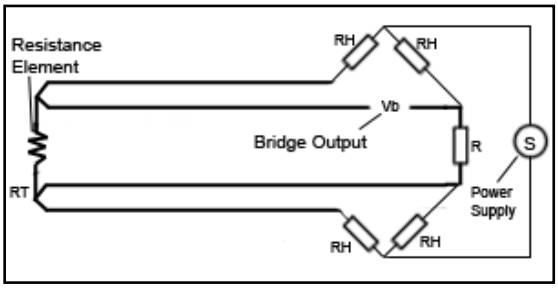


Figure 1.11 - A Four-wire Resistance Thermometer

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.

1.4.1. 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 1.12 shows an example of a Programmable Logic Controller (PLC).

Figure 1.12 - 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.

1.4.2. 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 HDA – 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
- Hyper Historian 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.

1.4.3. OPC UA

ICONICS GENESIS64 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.

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.

1.4.4. Data Displays

Figure 1.13 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.

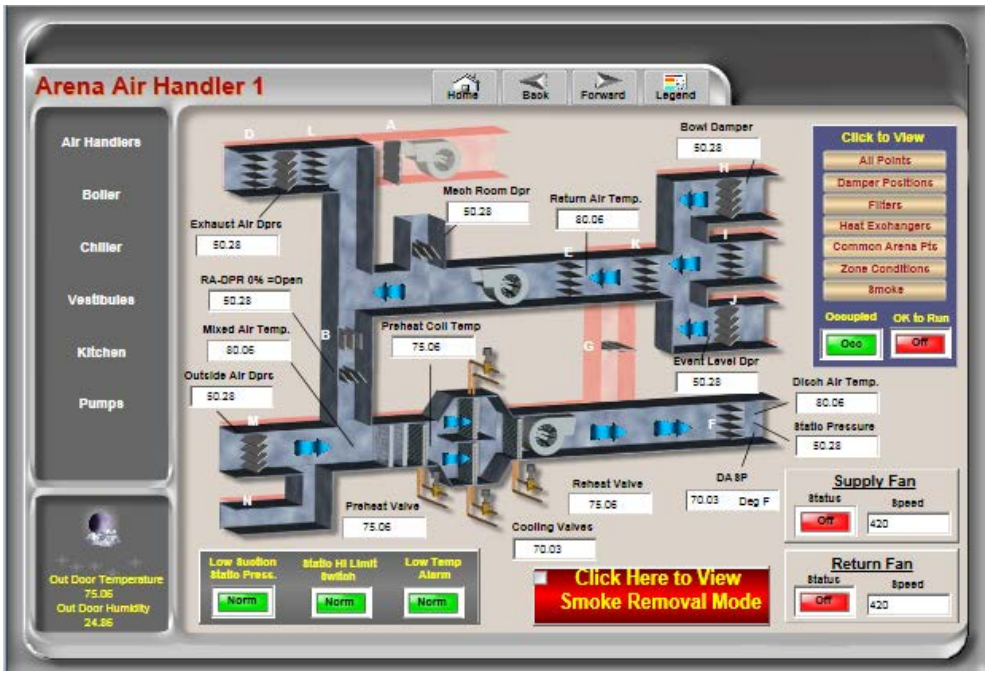


Figure 1.13 - An HVAC Display with Multiple Process Points

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*.

1.4.5. OPC Tags

Consider the schematic of a boiler shown in Figure 1.14 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
- 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 1.14 - 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).

1.4.6. BACnet

BACnet is a non-proprietary system, initially developed through the combined efforts of both building management system vendors and customers. According to the American Society of Heating, Refrigerating and Air-Conditioning Engineers (ASHRAE) Standing Standard Project Committee 135, BACnet is "a data communication protocol for building automation and control networks." Also according to ASHRAE, BACnet, "is an American national standard, a European standard, a national standard in more than 30 countries, and an ISO global standard." For more information about BACnet, please visit *www.bacnet.org*.

A data communication protocol is a set of rules governing the exchange of data over a computer network. The rules take the form of a written specification that spells out what is required to conform to the protocol; they govern the exchange of data over a computer network. These rules spell out what is required to conform to the BACnet protocol. What makes BACnet unique is that its rules relate specifically to the needs of building automation and control equipment.

They cover things like how to ask for the value of a temperature, define a fan operating schedule, or send a pump status alarm. Common BACnet applications include HVAC controls, fire detection and alarm, lighting control, security, "smart" elevators and utility company interfaces.

BACnet Client-Server Architecture

BACnet is based on a client-server model. BACnet messages are called "service requests." A client machine sends a service request to a server machine that then performs the service and reports the result to the client. BACnet currently defines 35 message types that are divided into 5 groups or classes.

One class contains ***messages*** for accessing and manipulating the properties of the objects described above. A common message is the "ReadProperty" service request. This message causes the server machine to locate the requested property of the requested object and send its value back to the client. Other classes of services deal with ***alarms and events***; ***file uploading and downloading***; managing the ***operation of remote devices***; and ***virtual terminal functions***.

ICONICS BACnet solutions integrate BACnet communications into the GENESIS64 and GENESIS32/Workbench32 product families. **ICONICS BACnet** solutions address customers' facility monitoring needs. Facility Engineers, facility managers and others involved with building management will appreciate its integration with ICONICS GENESIS32/Workbench32 and GENESIS64 HMI/SCADA suites.

Note that ICONICS is compliant with ANSI ASHRAE Standard 135-2008.

NOTE: Types which are part of the specification above are supported by ICONICS as well as other BACnet standards. ICONICS supports read/write property for all BACnet objects, but only the objects defined in ANSI ASHRAE Standard 135-2008 would be correctly shown in ICONICS clients (e.g. GraphWorX64).

1.4.7. SNMP

SNMP stands for Simple Network Management Protocol, and is a simple protocol that allows devices to expose useful information to other devices. This information can be the CPU fan speed of a computer or the routing table of the router. Almost every network device answers to SNMP requests. SNMP gives Network Managers access to information from nearly every device connected to the Network.

An SNMP network consists of a Network Management System (NMS), Managed Devices, and agents. NMS is software that collects data from the devices, organizes it, and shows it to the end user. The ICONICS SNMP connector functions as a manager in

an SNMP scenario. An NMS would monitor or control managed devices that provide SNMP information through the agents. An agent is software that runs on a device (such as a router, printer or PC) and answers to the messages from the NMS. These messages can either be ***read*** messages (NMS wants to retrieve data) or ***write*** messages (the manager wants to set data). The agent can also send a ***trap*** to the NMS. A trap is a notification similar to an alarm, such as "the CPU Temperature is too high!" For more information, refer to SNMP Basic Components, and SNMP Basic Commands.

Each variable that SNMP can read/write is identified in the form of an Object Identifier (OID). The OID identifies the information in a sequence of numbers. For example, 1.3.6.1.2.1.1.1 is an OID that describes the description (*sysdescr*) of the device. All of the OIDs for a device make up the MIB (Management Information Base), which represents all the data that the device can expose through SNMP. This information for a device is provided by the manufacturer in the form of a MIB text file. This MIB file acts as a dictionary for the device. The MIB files can be used by the NMS to translate numerical OID strings into human-readable text. For example, 1.3.6.1.2.1.1.1 can be translated as "iso.org.dod.internet.mgmt.mib-2.system.sysdescr", which describes the location for sysdescr (the description of the system) that is inside the folder "system" which is inside "mib-2", and so on.

ICONICS' SNMP provider allows you to create your own Network Management Systems. It allows you to manage the network through SNMP and perform Supervisory Control and Data Acquisition (SCADA through OPC) with the very same application. Moreover, you can monitor and manage the Ethernet-based PLC-control networks through SNMP for a complete and universal network management.

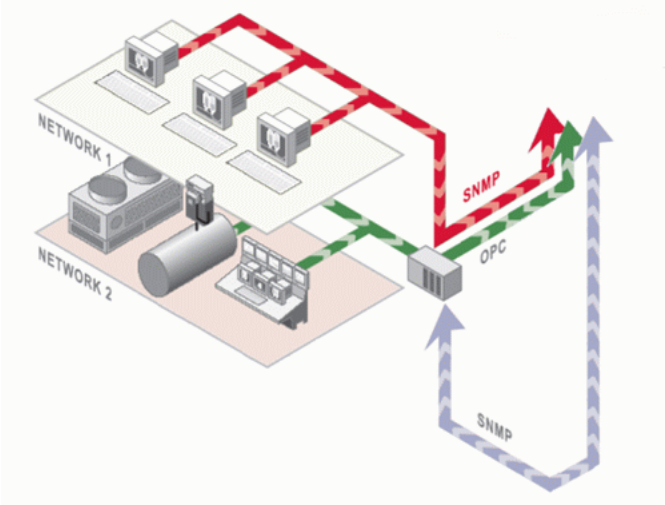


Figure 1.15 - Network Management Using SNMP Connector

With ICONICS' SNMP provider, you can quickly analyze a network, discover new devices, and create SNMP tags. However, this is just a little part of what is possible to

do. Searching through the MIBs of the devices, you can find multiple types of information. The only limit to what is possible to do with SNMP is just the imagination.

1.5. 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 1.16. 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.

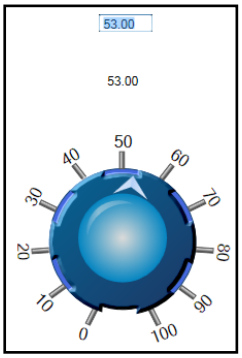


Figure 1.16- A Dial

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.

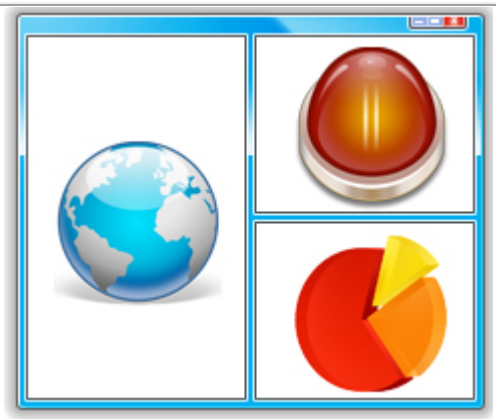
As long as the value you enter into the DataSource property is syntactically correct, it will be allowed; it is up to the developer to determine the correctness of the assignment. At runtime, the value of the data source refreshes at an interval that is another property of the display.

The knob and the process point in Figure 1.16 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.

1.6. Connectivity

ICONICS utilizes cutting edge technology in order to provide multiple Configuration and Runtime options that suit our customer and partner needs.

From Windows Presentation Foundation (WPF) to SharePoint to HTML5 technology, ICONICS software solutions cover the evolution of Flexible or Frame-based Canvas Technology for data visualization.

Flexible Canvas	**Frame-based Canvas**
Users: *SCADA, BAS, Plant Control, Machine Builder*	**Users: *IT Personnel, Enterprise-level, MES, Analytics/Reporting***
	

Product	Technology	Deployment		Product	Technology	Deployment
GraphWorX64	Windows Presentation Foundation	Desktop		PortalWorX	Microsoft SharePoint	Web-based
MobileHMI	HTML5	Mobile App		PortalWorX-SL	HTML5	Web-based
WebHMI	HTML5	Web-based		Workbench Classic Runtime	Windows Presentation Foundation	Web-based
Any Glass	HTML5	Web-based				

ICONICS provides both Flexible and Frame-based Canvas Data Visualization based on your organization's deployment needs (Desktop, Mobile or Web-based) and preferred technology standards (WPF, SharePoint or HTML5).

ICONICS 'Any Glass' technologies, integrated with HTML5, provide users with an additional mobile, Web-based option for data visualization using today's popular tablets and smartphones including iPad, iPhone and Android-based devices.

1.7. The Unified 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 1.17. An ellipsis button in GENESIS64 indicates that more choices are available through a utility, menu, or some other interface element.

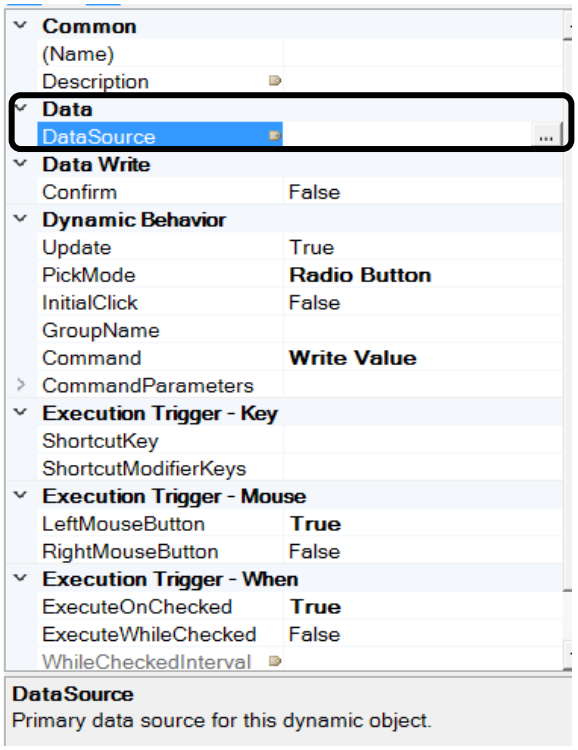


Figure 1.17 - The Data Source Property

For the DataSource property, the ellipsis button opens the ICONICS Unified Data Browser, shown in Figure 1.18. The Unified Data Browser is where you navigate to and select the data source. Because the data source can be specified or derived in a number of ways, there are several tabs in the Data Browser interview where you can navigate, view, and select a variety of data sources and functions.

In the Data Browser, you can create values, expressions, variables, provide name resolution, and other features required by many of the objects, dynamics, and their properties in the GENESIS64 application suite.

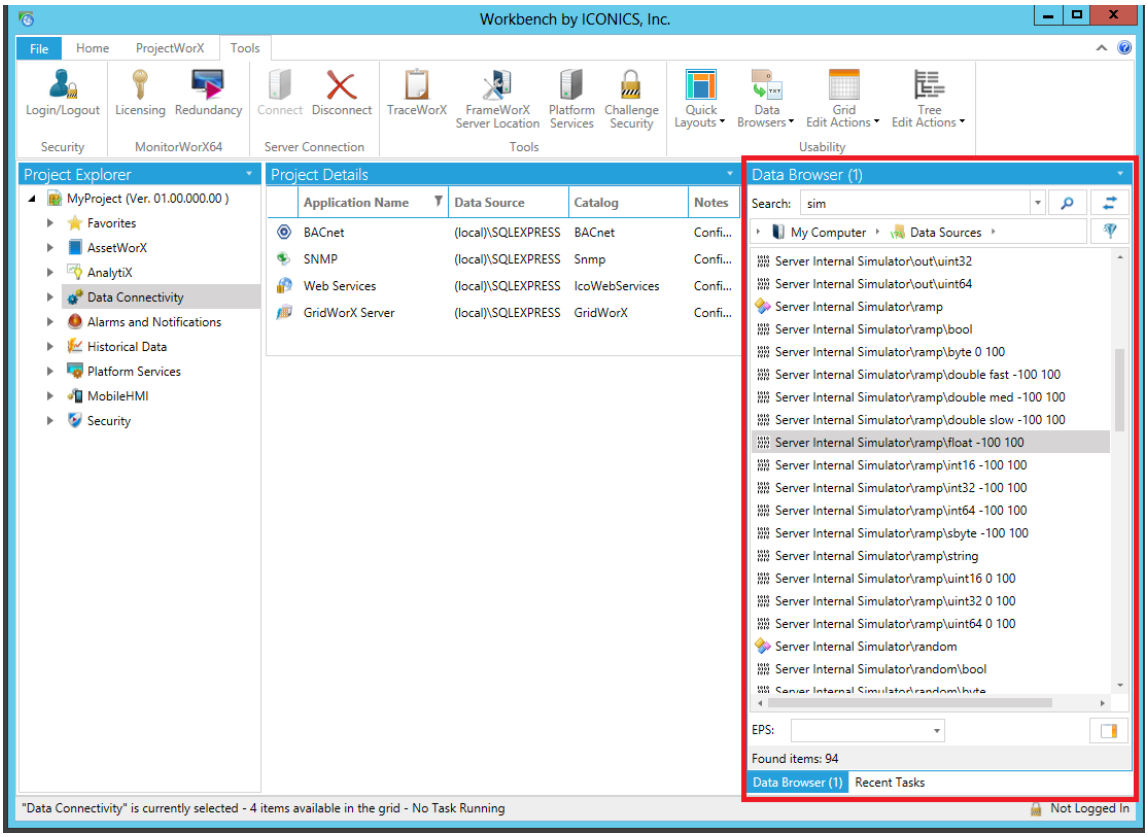


Figure 1.18 - Data Browser: Simulations

For example, the Simulations allow 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 is where you navigate to and select the data source. Because the data source can be specified or derived in a number of ways, there are several tabs in the Data Browser interview where you can navigate, view, and select a variety of data sources and functions. In the Data Browser, you can create values, expressions, variables, provide name resolution, and other features required by many of the objects, dynamics, and their properties in the GENESIS64 application suite.

The Unified 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 1.20, the OPC DA tab is shown with a tag already selected. Notice that the address to the device tag is displayed in the address bar. You can simply drag the tag from the Data Browser and drop it into the DataSource Property where the Data Browser enters the address in the DataSource property in the correct form using the appropriate syntax.

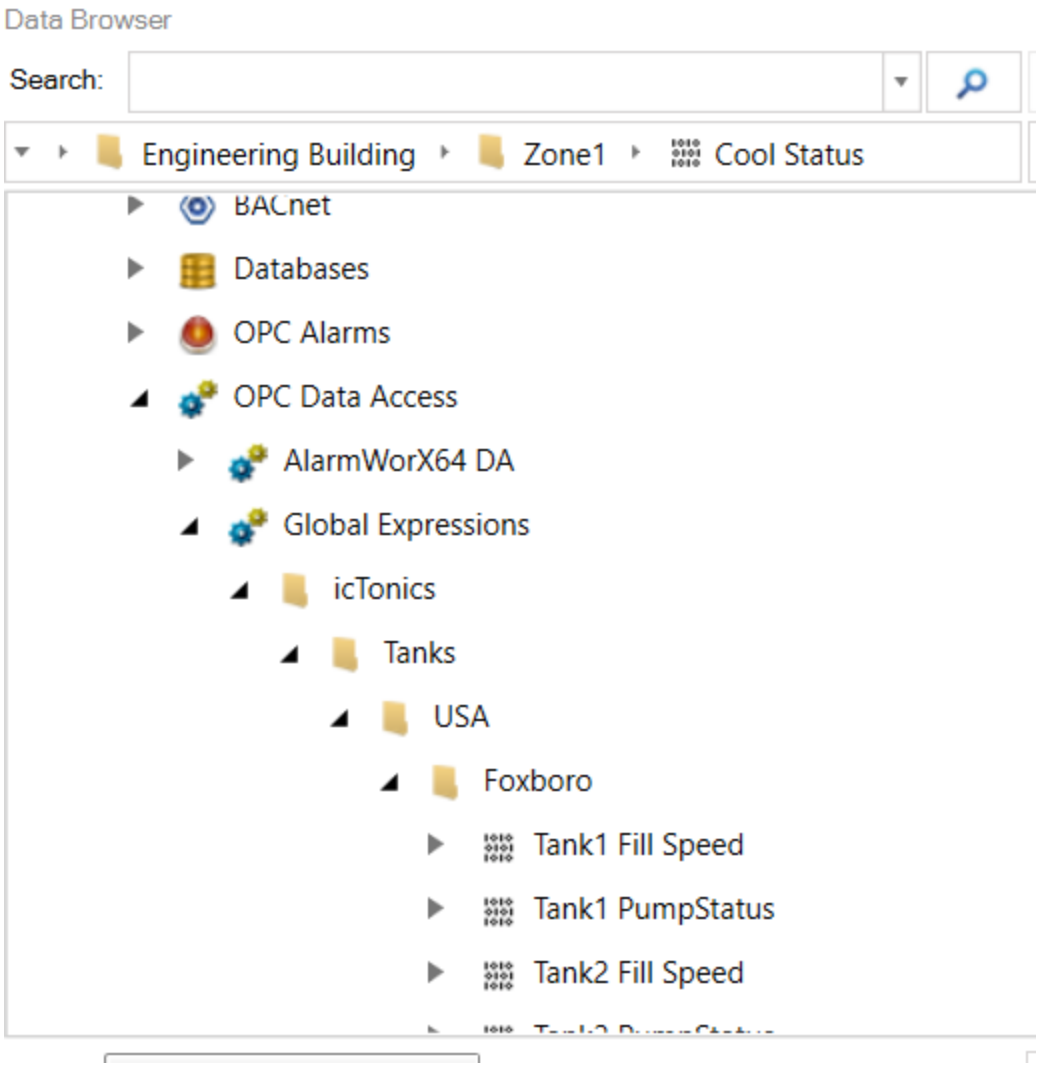


Figure 1.19 – Unified Data Browser: Tag View

The power of the Unified Data Browser is that it lets you know what data sources are actually available and correctly defines the addresses using the appropriate syntax and automatically builds a data connection expression that is correctly formatted for the source and purpose intended. Although you can enter tags directly into properties you are generally advise that it is safer to build expressions, enter tags, or create paths to tags using the Unified Data Browser.

The Unified 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 knob is connected to the tag you just assigned to it and it changes appropriately.

1.8. System Health Monitor

The System Health Monitor is a tool created to monitor the overall system, provide diagnostic tools and correct the situation. Multiple ICONICS services include performance-related information, via Performance Counter and Data Points (within Platform Services) and the System Health Monitor provides insight into this various information.

Figure 1.20 - System Health Monitor Dashboard

System Health Monitor settings can be accessed through the Health Monitor Settings properties in the FrameWorX node of the Project Explorer in Workbench for Desktop. See the “Health Monitor Settings” topic for more details.

System Health Monitor is a powerful tool that monitors, diagnoses, predicts and corrects. It runs as a standalone service, independent to other modules, without requiring a separate license. It integrates with ICONICS Platform Services, implementing a Point Manager API, with data accessible via the GENESIS64 HMI/SCADA and building management suite.

System Health Monitor can use inputs from Performance Counters and Data points that are available within:

- ICONICS Services
- Third-party Applications
- Operating Systems

This component can use aggregate functions. It can also generate alarms that can be visualized in AlarmWorX64 Viewer, logged using AlarmWorX64 Logger, and logged to the Windows Event Log. It also has the ability to recycle a Windows Service.

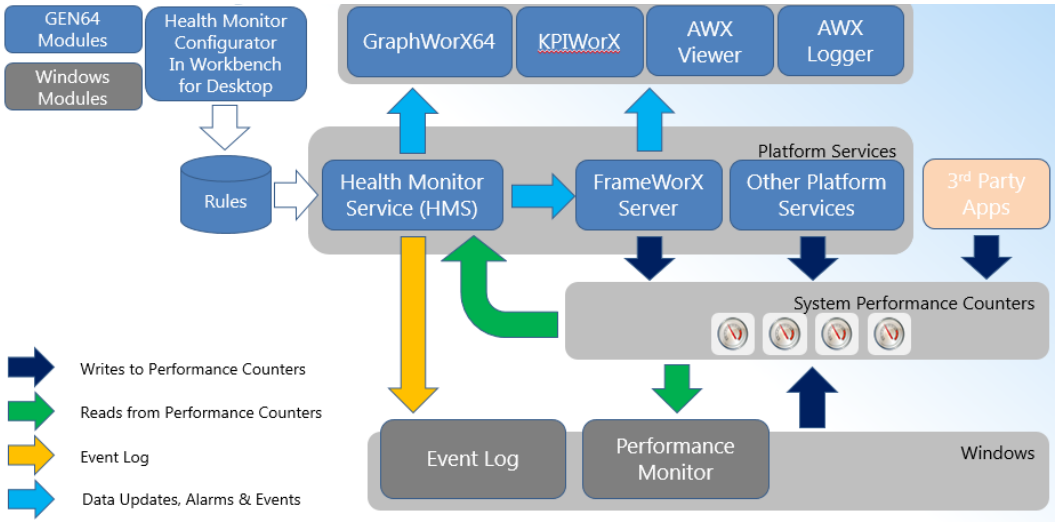


Figure 1.21 - System Health Monitor Architecture

Lab: Training Installation and Configuration

L1A

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 Global Training Centers and are using one of our provided machines, you may not need to perform this lab. Your instructor will inform you.

Estimated time to complete this lab: **60 minutes**

1A.1. Overview

Goals for this lab are to:

- Install GENESIS64 on the training machine.
- Understand the privileges inherent to local user "admin" (which you are automatically logged in as if you are using a Training Center computer) on each training machine.
- Review and/or adjust user permissions in SQL Server.

1A.2. Task Checklist

In this lab, you will complete the following tasks:

- Install GENESIS64.
- Create any necessary database files.

1A.3. Installation Software and Support Files

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

- GENESIS64 v10.95 Installation Disk
- GENESIS64 v10.95 Update 3 Disk
- icTonicsUnifiedDB_Assets_Config_(10.95.200.0).csv
- icTonicsUnifiedDB_Modbus_Config_(10.95.200.0).csv
- icTonicsUnifiedDB_UnifiedDataManager_Config_(10.95.200.0).csv

1A.4. 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.

1A.5. 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 that you already have installed.

- **Internet Information Services** (IIS) with **FTP** components
  NOTE: FTP is required when you intend to publish displays for use with WebHMI clients.
- **The latest Microsoft .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.
- **Microsoft SQL Server 2008 R2 SP1 Express or higher** – configuration databases in GENESIS64 can only be an MS SQL 2008 R2 SP1 or higher database. ICONICS Training Center machines have Microsoft SQL Express 2014 installed.
- **User Account Control (UAC)** – Although GENESIS64 works with UAC, it is 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 for the location on the operating system you will be using.

1A.6. Installing GENESIS64

First, insert the GENESIS64 installation disk.

When the splash screen opens, click on the Product Installation **ICONICS Suite** link. In case the splash screen does not open (you may have auto-run disabled), you can go to your DVD drive and double-click on the **Default.hta** file to open the splash screen.

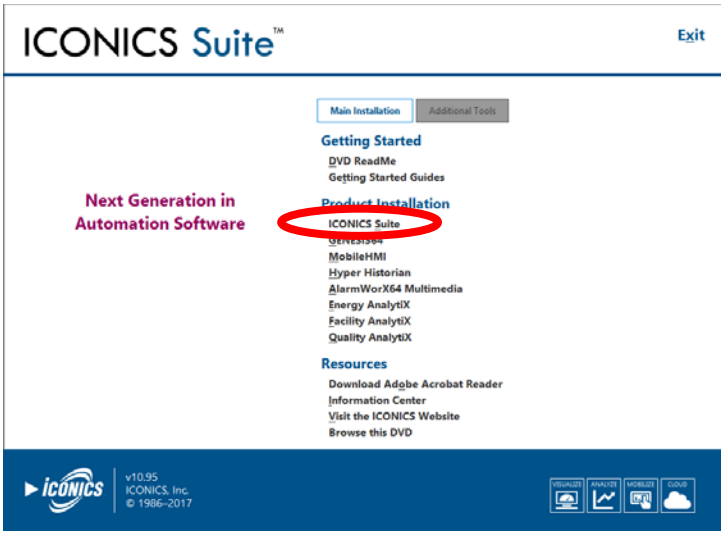


Figure L1A.1 - Installation Splash Screen

1. You should then see the **ICONICS Suite - InstallShield Wizard** as shown below. Select the **Install** button to continue.

   NOTE: Your list of prerequisites may not match the figure below.

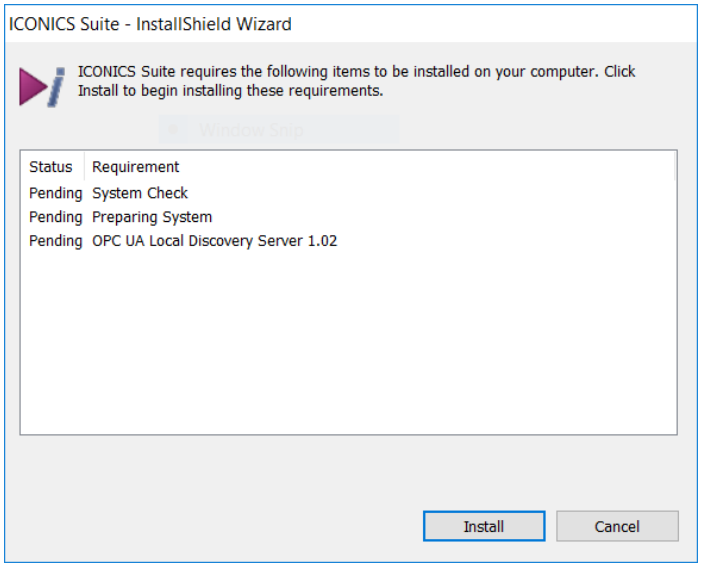


Figure L1A.2 - InstallShield Wizard

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

2. If you don't have SQL Server pre-installed on your machine, you will get a message. 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.

3. When the prerequisites installation is complete, you will be prompted to reboot your computer. Click **Yes** to reboot.

4. After the reboot is complete, the installation will continue. Click **Next**.

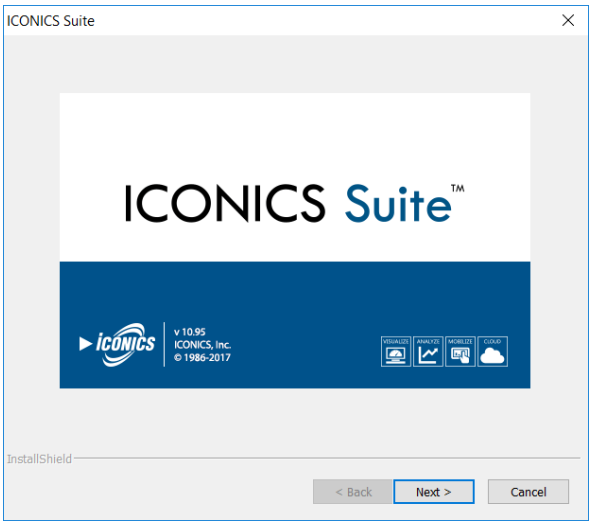


Figure L1A.3 – The InstallShield Wizard Reopens

5. Select the radio button "**I accept the terms of the license agreement**" and click **Next**.

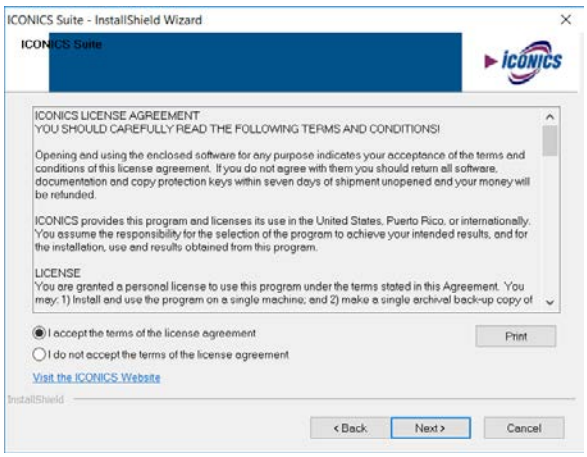


Figure L1A.4 – ICONICS License Agreement

6. Leave the radio button as **Suite Selection** with everything checked and click the **Next** button.

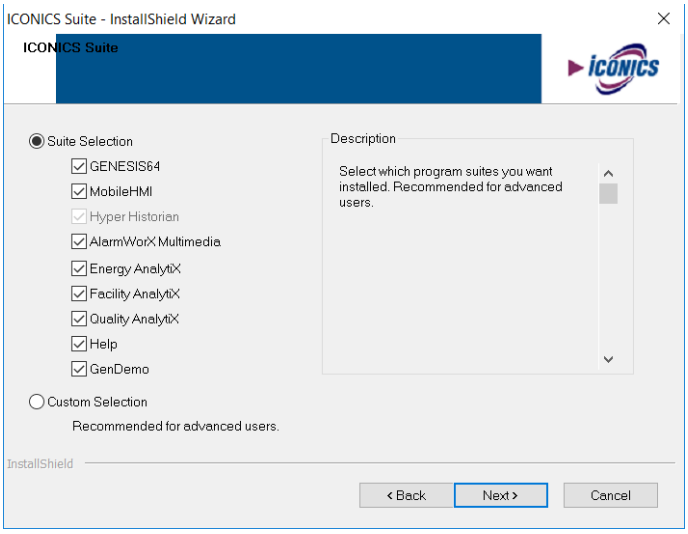


Figure L1A.5 – The Setup Type

7. Include the **GenDemo** samples and click **Next** to proceed.

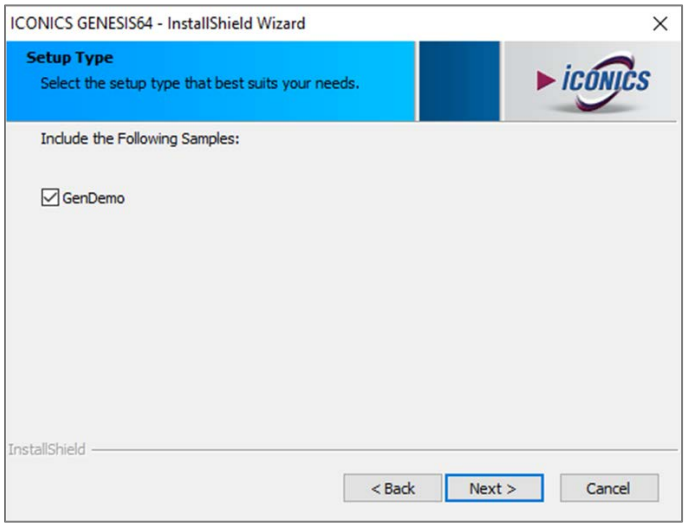


Figure L1A.6 – GenDemo Installation Selection

8. You can include the Facility AnalytiX Equipment Library if a Serial Number has been provided. Enter the **Serial Number** and click **Next**.

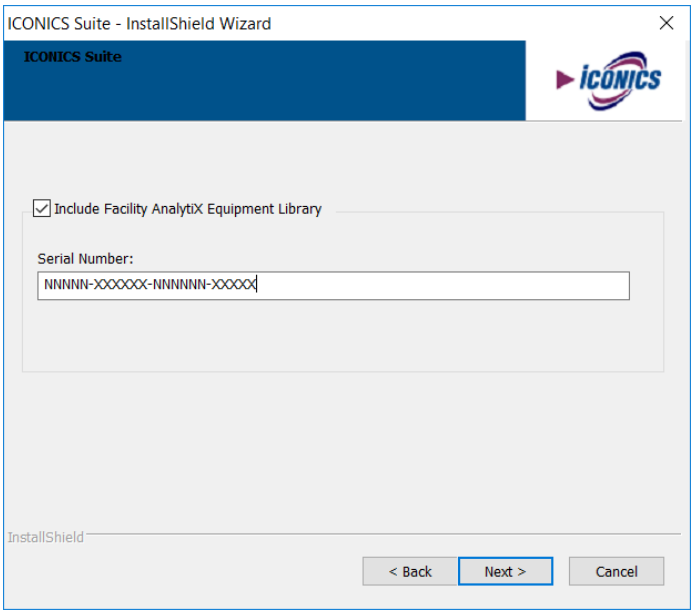


Figure L1A.7 – Facility AnalytiX Equipment Library

9. Leave the default selection, **For use with SMTP server**, and click on the **Next** button.

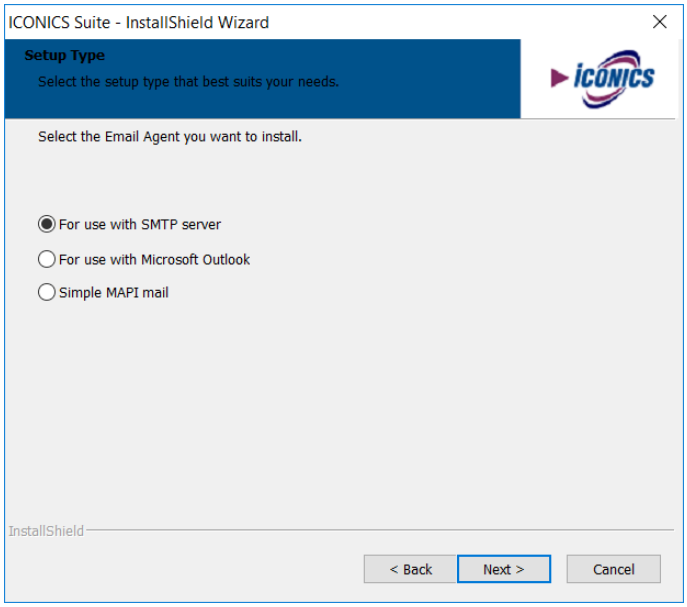


Figure L1A.8 – Login Information

10. Leave the default selection, **SNPP (Simple Network Paging Protocol)**, and click on the **Next** button.

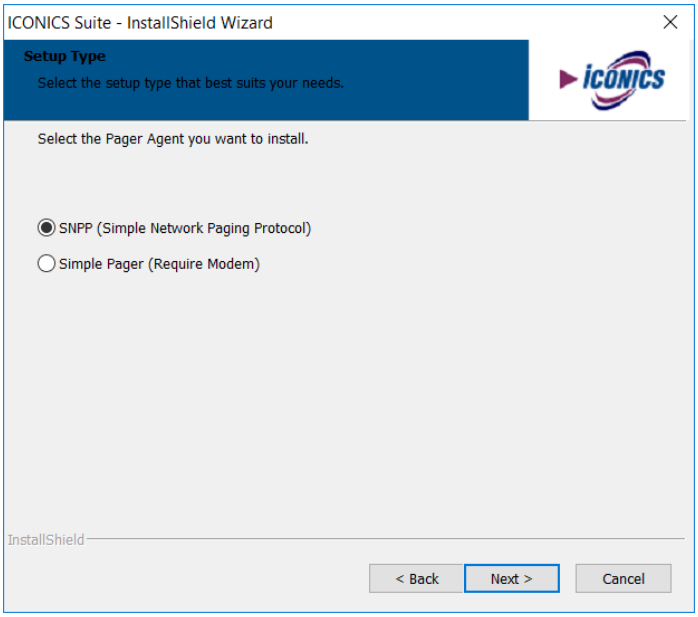


Figure L1A.9 – Select the Pager Agent

11. You will be prompted for login information. Enter the **User name** and **Password** of the user performing the installation. If you are attending this training course at one of our Global Training Centers and are using one of our provided machines, ask your instructor for the user name and password.

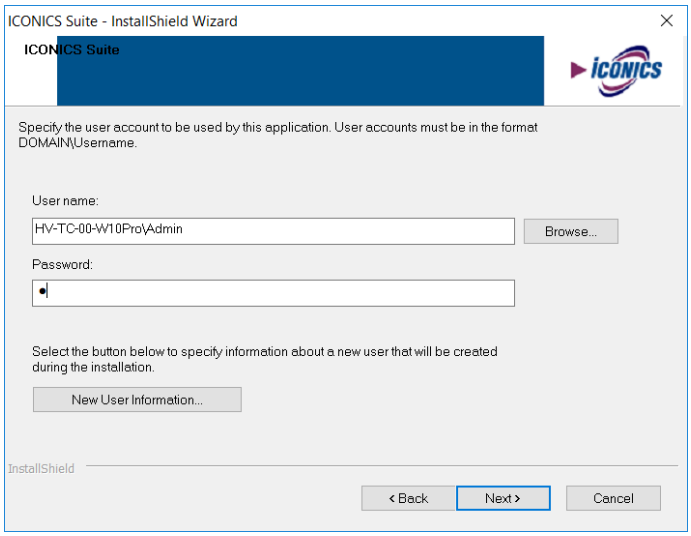


Figure L1A.10 – Login Information

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

12. As seen in the following figure, the **Database Server** dialog, 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. Since SQL Express is installed on the local hard disk, using the following syntax to connect **(local)\SQLEXPRESS** is also sufficient. If you are attending this training course at one of our Global Training Centers and are using one of our provided machines, use the instance name **(local)\SQLEXPRESS2014**.

13. Connect using **Windows authentication**. Click **Next** to proceed.

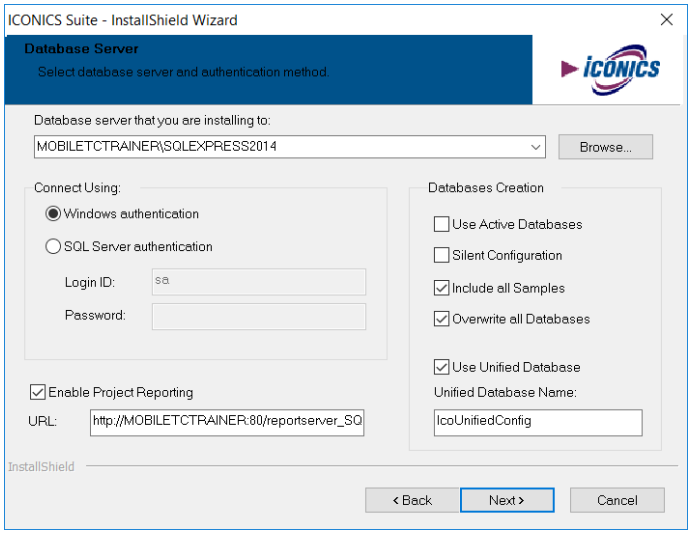


Figure L1A.11 – Database Server

14. At the **Ready to Install the Program** dialog, click **Install** to begin the installation.

15. At the **System Setup** dialog, the default selections should be fine. Click **Install**.

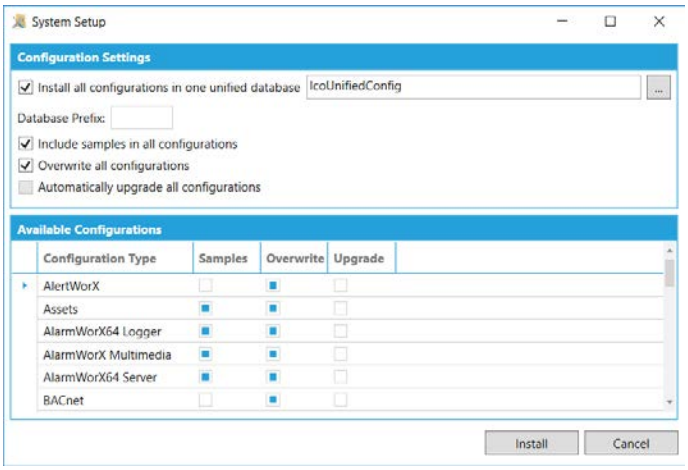


Figure L1A.12 – System Setup

16. System Setup should display a green check for each item [Completed] then opens the **InstallShield Wizard Setup Status**.

17. At the **InstallShield Wizard Complete** dialog, click **Finish** and allow the computer to restart.

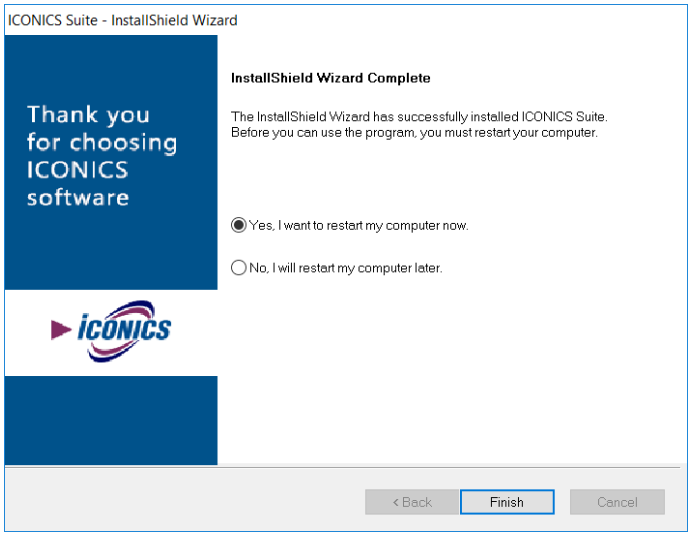


Figure L1A.13 – InstallShield Wizard Complete

1A.7. Training Support Files

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

1. Open Windows File Explorer and navigate to **C:** drive.

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

   Note: Ask your instructor for the support file if you do not have it.

3. Open **Workbench** and select the **Project** tab.

5. Select the **File Configuration** icon from the Project Tools ribbon.

6. Select the gray bar marked, **+ Click here to add new item**, and enter the Alias as **Training,** select the File Path to be **C:\Training**; to increase security you can change the **Server Allowed** from ***** to your computer name, and enable the **Include Sub Folders** box.

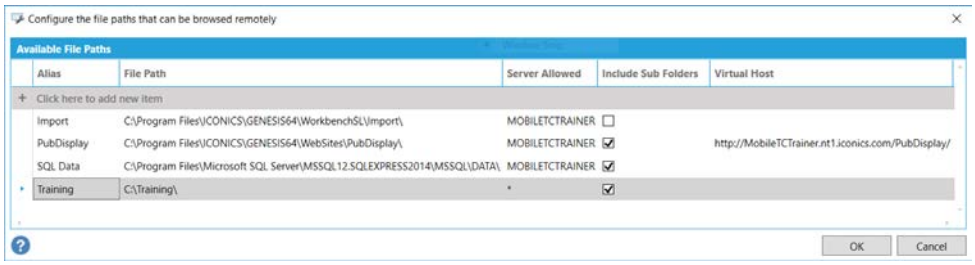


Figure L1A.22 – Configure the File Paths that can be Browsed Remotely

1A.8. SQL Server Permissions Review

1. Go to Start → Microsoft SQL Server 2014 → SQL Server Management Studio.

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

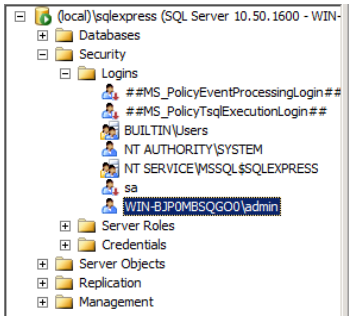


Figure L1A.30 – Admin User Account in SQL

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

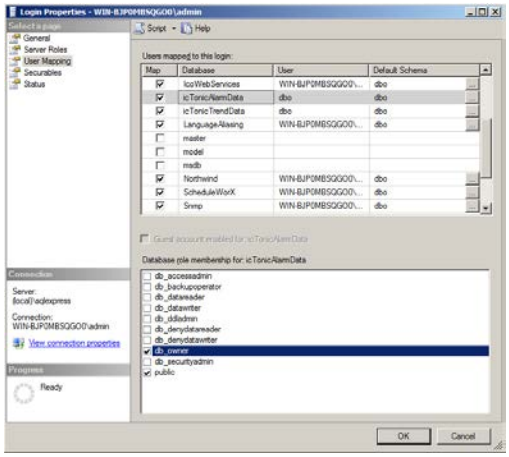


Figure L1A.31 – Authorize Admin User Permission

1A.9. Web Browser Configuration

1A.12.1. Enabling Internet Explorer File Download

For training at the Foxborough, MA facility, we use a Windows Server class operating system. Although GENESIS64 is certified for Windows 2012 x64 Server to Windows 10 Enterprise, in the Training Center, we are using Windows 10 Professional or above.

We will have to make an **Internet Explorer Security** change to **Enable File Download**. Here we go.

1. Select **Start → Control Panel → Internet Options** to launch the Internet Properties dialog.

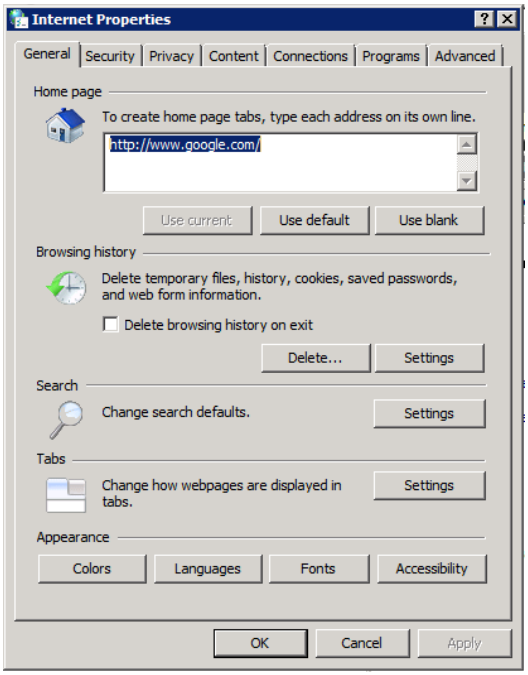


Figure L1A.32 – Configuring Internet Options

2. Select the **Security** tab.

3. Select the **Internet Icon**.

4. Click on the **Custom level** button.

5. Click on the empty section of the scroll bar on the right about 4 times to expose the **Downloads** property.

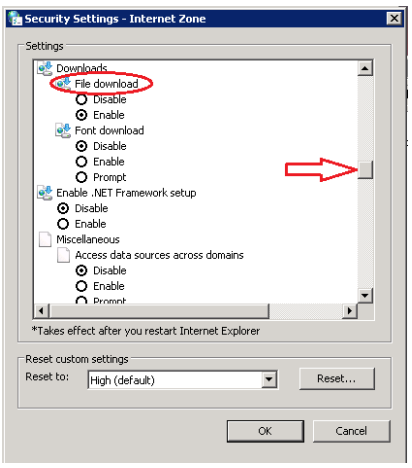


Figure L1A.33 – Configuring Internet Options

6. **Enable** the '**File Download**' property.

7. Click **OK** and keep the **Internet Properties** dialog **open**.

1A.12.2. Adding your Host to the List of Trusted Sites

1. From within the **Internet Options** dialog, select the **Security** tab.
2. Select the **Trusted Sites** icon.
3. Click on the **Sites** button.

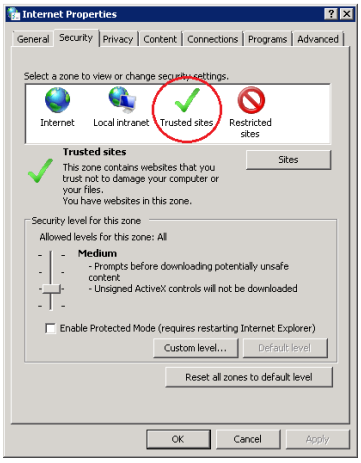


Figure L1A.34 – Internet Explorer Trusted Sites

4. **Uncheck** the **Require server verification (https:) for all sites in this zone** checkbox.
5. In the **Add this website to the zone:** field, enter: **http://localhost**.
6. Click **Add** button.
7. Click **Close** button.
8. Leave the Internet Properties dialog open.

1A.12.3. Configuring Internet Explorer Cache

The following setting is suggested during development. However, for performance reasons, you may want to leave this setting as **Automatic** at the time of deployment.

1. From within the **Internet Properties** dialog, select the **General** tab.
2. In the **Browsing history** section, click on the **Settings** button.
3. Under the header "**Check for newer versions of stored pages:**" select the "**Every time I visit the webpage"** radio button.

4. Click **OK** button.

5. Click **OK** again to close the **Internet Properties** dialog.

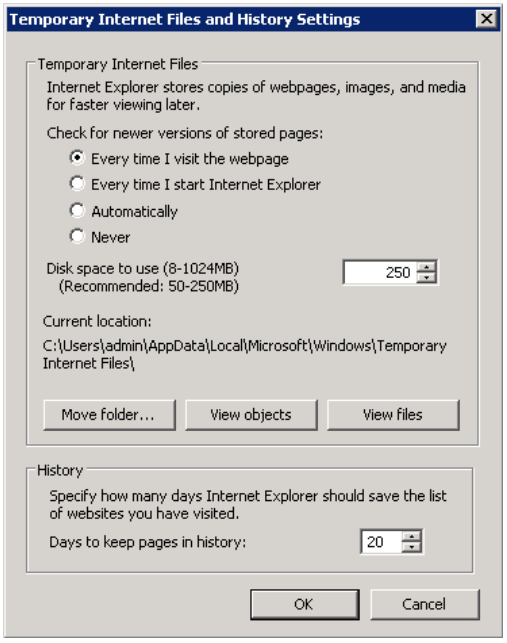


Figure L1A.35 – Internet Explorer Cache Setting

1A.10. The Symbol Libraries

When we unpacked our project we automatically added four symbol libraries ICONICS created specifically for this training. The following steps describe how to add a symbol library to GraphWorX64 separate from unpacking a project.

1. Workbench Desktop should still be open. Select the **File** tab from the top of the Workbench. In the File tab menu select **Configure Displays** from the dropdown list.

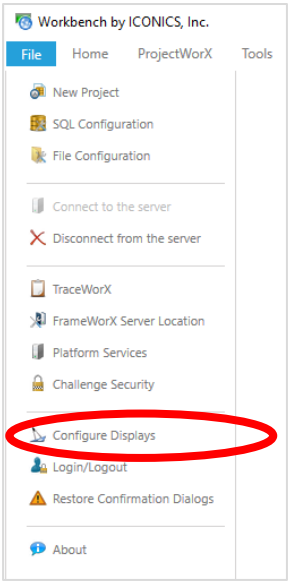


Figure L1A.36 – Workbench - Configure Displays

NOTE: You could also Launch GraphWorX64 by selecting **Start → ICONICS Product Suite → GraphWorX64.**

2. Once GraphWorX64 opens, you should see the Explorer on the left-hand side. At the bottom of the Explorer, you should see a tab called **Symbols**. Select the **Symbols** tab.

3. Click on the **Symbol Library Commands Menu** icon at the top of the explorer and select **Open Existing Category** from the popup menu.

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

5. If you click on the dropdown list of the Symbol library now, you should see the **SI_BA_Training** symbol library you added in the category list.

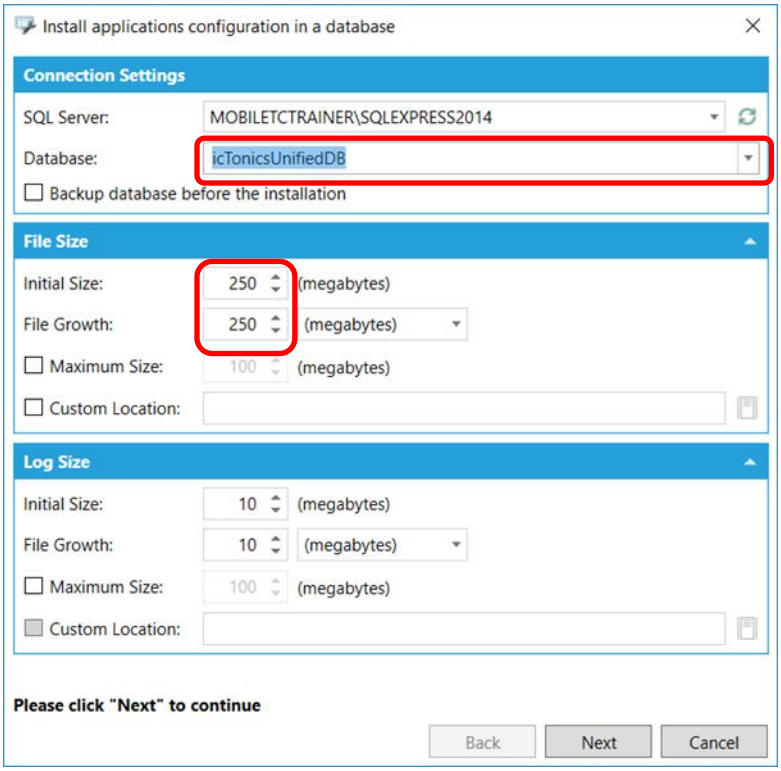
6. Leave the GraphWorX64 instance open.

1A.11. The Training Project

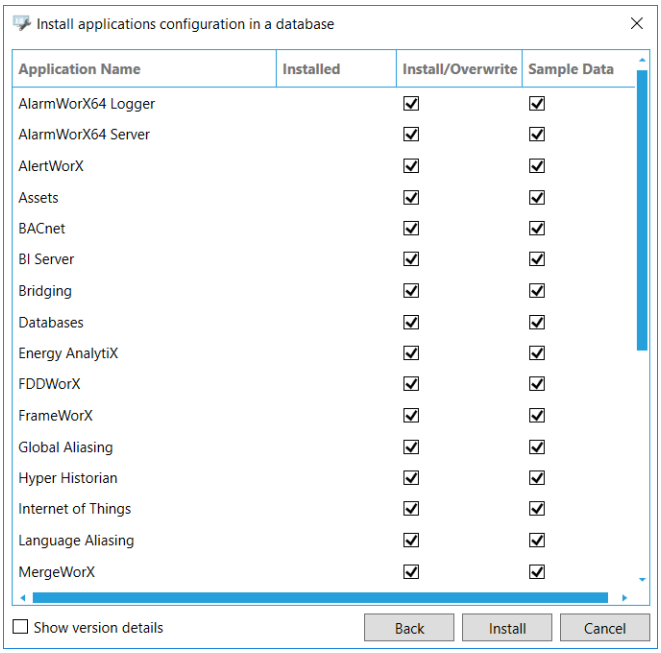
1A.14.1. Creating the icTonicsProject and Database

Certainly we could have unpacked the AssetWorX configuration like we did with the other providers, but you should understand that the Workbench also supports Import/Export of configurations.

1. From within the **Workbench**, you will first need to create an additional Project. Go to the **Project** tab in the top left and then click on **Configure Database** from the **Database Management** ribbon.



2. The Install applications configuration in a database popup should now be open. Type **icTonicsUnifiedDB** into the **Database** field of **Connection Settings**. Change the **Initial Size** and **File Growth** settings to **250** in the **File Size** section. Select **Next** button to continue.



3. Go to the **Project** tab in the top left and then click on **New Project**, as shown below.

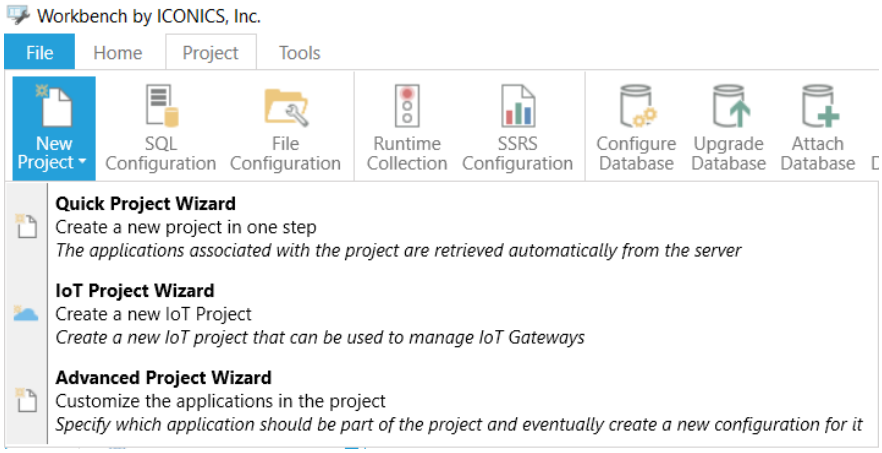


Figure L1A.37 – Creating a New Project

4. Select **Quick Project Wizard** to create a new project in one step.

5. This opens the **Create project Wizard** window, as shown below. Enter **icTonicsProject** as the Project Name. Type your name into the Author field. Enter, "**My GENESIS64 Standard Training Project**" without the quotes or an appropriate description of this project in the Description box. Click **OK** to proceed.

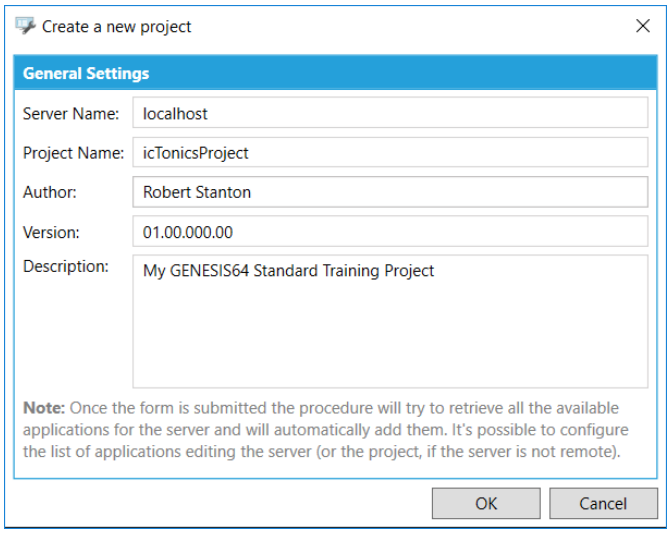


Figure L1A.38 – Create the icTonicsProject

6. The Project Details frame in the center of the Workbench will open with the icTonicsProject tab in the lower half. Type **icTonicsUnifiedDB** into the **Catalog:** field of the **Unified Database** section.

7. then enter an **Initial Size** of **250** (megabytes) and **File Growth** of **250** (megabytes). Click **Next** to continue.

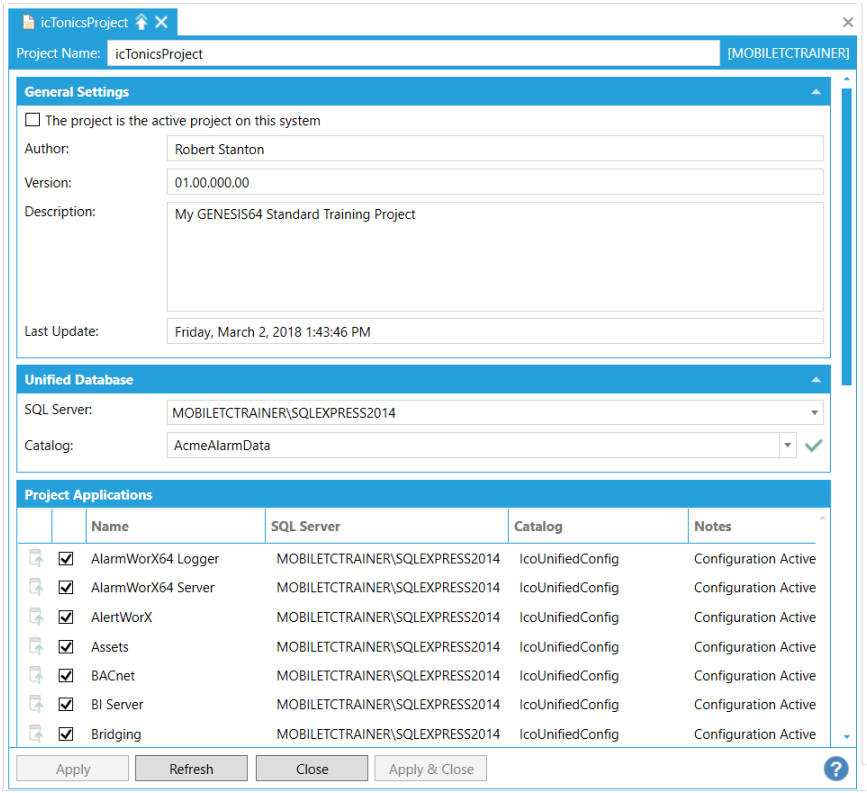


Figure L1A.39 – Create the icTonicsProject Database

8. On the Applications page of **Create project Wizard**, right-mouse click on the header **Install/Overwrite** to enable all applications. Click the **Ok** button to finish creating the new project.

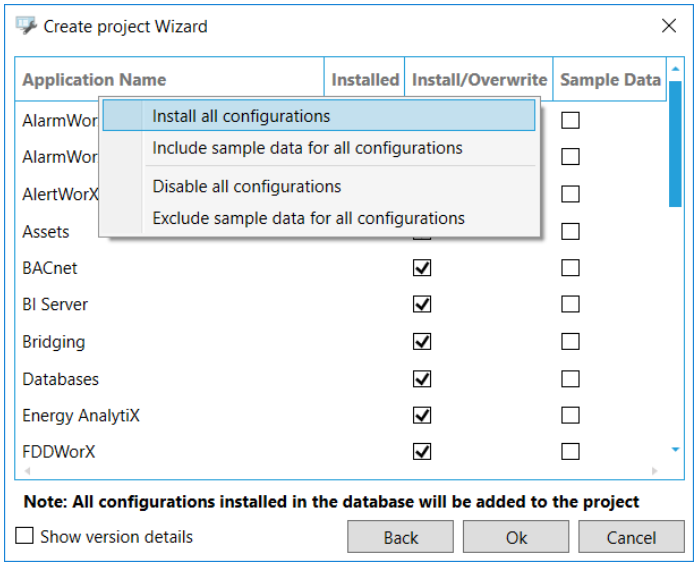


Figure L1A.40 –Install/Overwrite All Databases

1A.14.2. Importing icTonics Assets Data

You can import data into the **Assets** configuration or into the asset tree or equipment class node. It is assumed that the data you want to import has been exported from an Assets configuration and is stored in a Microsoft Excel 2007/2010 (*.xlsx) file. Be aware of the data being imported so that you import it to an appropriate location in the Project Explorer. For example, do not import Equipment Classes into the Assets Equipment branch.

1. Within your new **icTonicsProject**, right-click on the **Assets** application and select **Import** from the dropdown menu.

2. From the **Import Options** dialog, click on the **File** button to the right.

3. Locate and select the **C:icTonicsUnifiedDB_Assets_Config_(10.95.200.0).csv** file. Click the **Open** button.

4. Enable **Continue on errors**. Enable **Ignore GUID**. Click **OK** button. Continue through any Warning popups that appear. There should not be any errors. Errors will be obvious in the **Recent Tasks** window on the right.

5. Now import the **icTonicsUnifiedDB_Modbus_Config_(10.95.200.0).csv** and **icTonicsUnifiedDB_UnifiedDataManager_Config_(10.95.200.0).csv**.

5. Two popup windows should appear near the bottom right of the Workbench. The first should indicate the Import was added to the queue and the second should indicate the Import completed. Click on the windows to close them.

6. Right click on **Assets** and select **Refresh** the configuration.

7. You should now see **icoUniversity** under **Equipment** in the Assets tree.

1A.14.5. Copy the LibraryAHU.gdfxs Display to PubDisplay

In order for displays to be viewed in PortalWorX, the display has to reside in the PubDisplay folder (by default) and the display has to be saved as a Silverlight (.gdfxs) version. We will proceed to copy the LibraryAHU.gdfxs file to the PubDisplay folder.

1. Copy the **C:\Training\LibraryAHU.gdfxs** file to the **C:\ProgramFiles\ICONICS\GENESIS64\WebSites\PubDisplay** folder.

We will open the AHU display in a subsequent lab.

1A.14.6. Exploring GenDemo

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

2. Open **GenDemo** by going to **Start → ICONICS Tools → GenDemo**. Opening this version of GenDemo does not allow us to go into GraphWorX64 configuration mode.

3. Once the display opens, hover over the objects on the right and click on the **Overview** item you'd wish to view.

4. You can also select from the dropdown list on the left.

5. To start/stop Carousel Mode, click on **Advanced Settings → Run in Carousel** Mode on lower left.

6. Select **Hyper Historian** from the list on the left.

7. The Hyper Historian display shows an overall view of key feature, such as Continuous Storage of Mission Critical Data, Data Editing and Reporting Capabilities, and Flexible Open Standards Architecture.

8. Hover your mouse over the list on the left and select the **What's New in 10.95** item. We've included a New Products list to demonstrate the new product and modules available in 10.95 such as:

a. Products

i. IoTWorX – IoT software solution designed to work with affordable on-premises IoT Gateways.

ii. ReportWorX64 – Powerful reporting tool designed to turn volumes of data into visual intelligence.

iii. BridgeWorX64 – Next generation data bridging to move massive amounts of data efficiently and reliably.

b. New Modules

i. Workflows – All users to leverage customizable logic to execute complex actions and create triggered events that are continuously on scan.

ii. Analytix BI – Used to create predefined data flows and data modules to quickly analyze business intelligence (BI) within an operational context.

iii. Mixed Reality – Includes new Holographic Machine Interface (HMI) capabilities leveraging the Microsoft Hololens.

iv. Modbus – Native communications to Modbus Ethernet devices.

9. Click on a selection from the list on the left and then click on the items inside the display to see how they react. Consider using this as a future reference.

This concludes the lab.

Workbench

2

The Workbench (in some instances referred to as "Workbench for Desktop") is a centralized environment for GENESIS64 product configuration and runtime operations, providing a framework for security, project management, and multimedia technology to configure alarms.

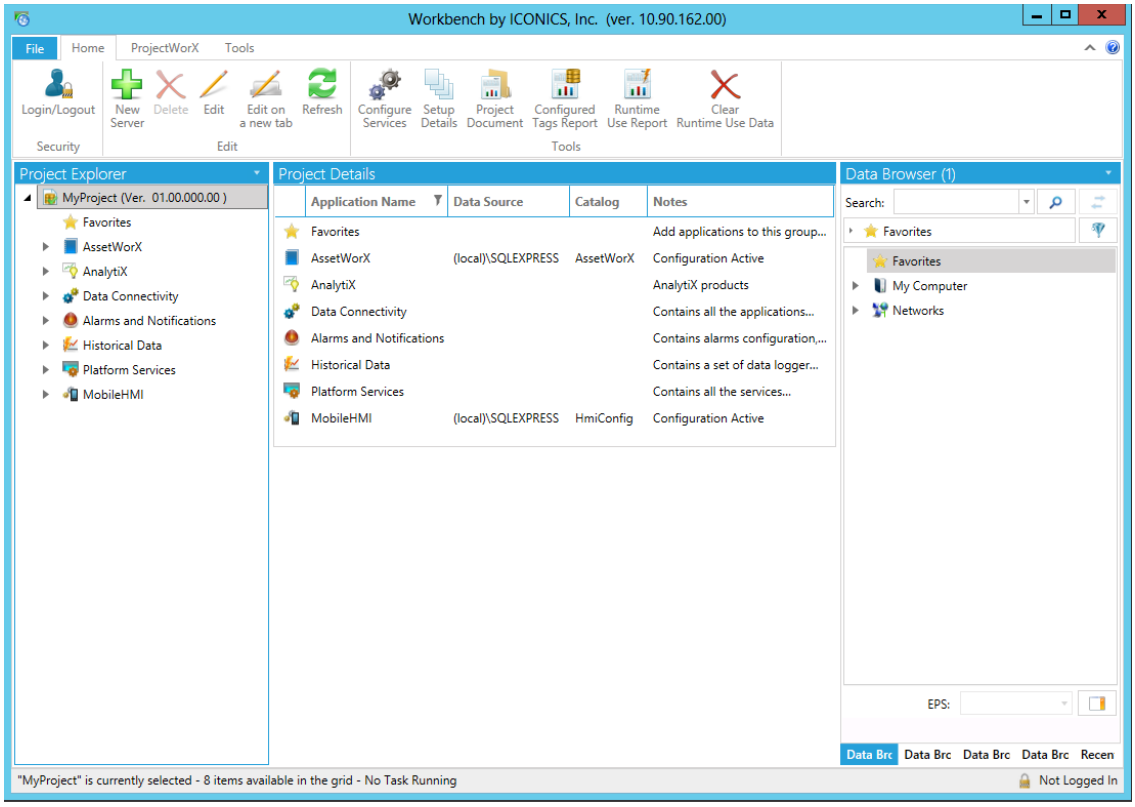


Figure 2.1 – ICONICS Workbench Project Configuration Tool

2.1. Objectives

After completing this chapter, you should know:

- How to Use the Workbench Configuration Tool
- How to Access Tools and Providers
- How to Use the Project Explorer

2.2. Introduction to the Workbench

2.2.1. Workbench

Multiple ICONICS products integrate with a centralized Web-based environment for product configuration and runtime operations. This environment is the **Workbench**. This is the latest iteration of the Workbench tool, which could also be referred to as "Workbench for Desktop". It integrates several new HMI/SCADA technologies, making project configuration and runtime as easy as possible.

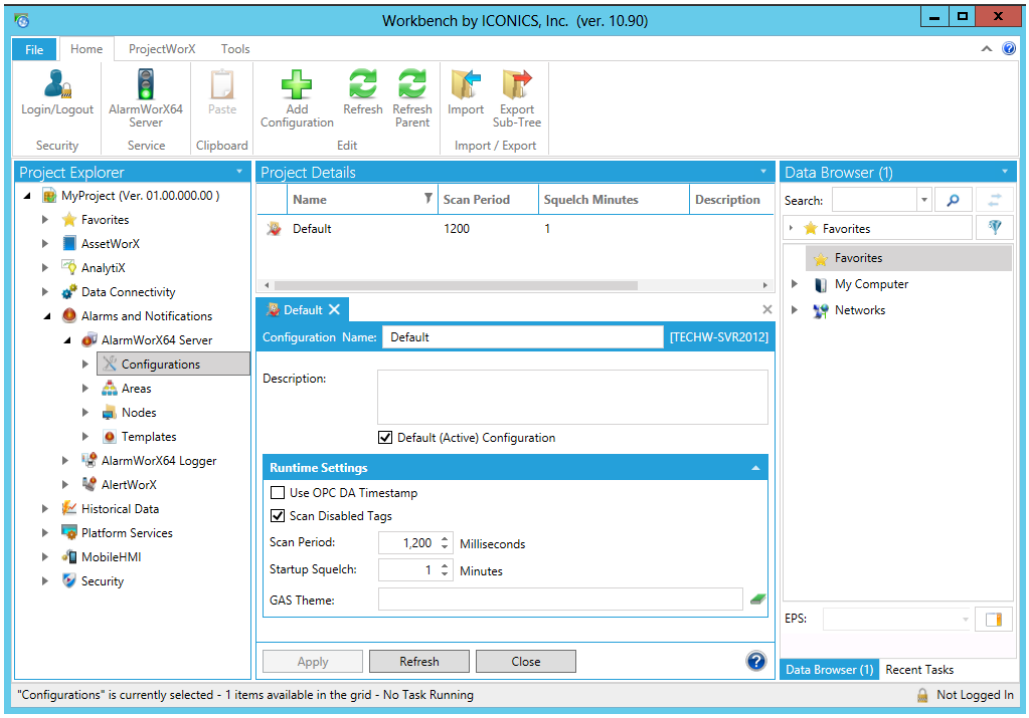


Figure 2.2 – Expanded AlarmWorx64 Example of a Workbench Environment

2.2.2. Opening the Workbench

Accessing the Workbench - Windows 10

In Microsoft Windows 10, users can find the Workbench under 'ICONICS Product Suite', as shown in the following image.

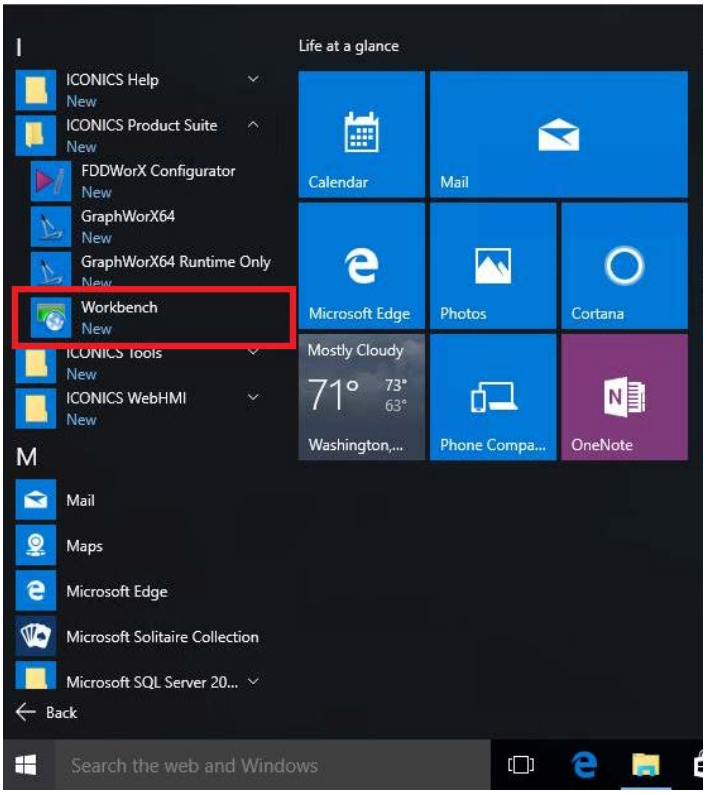


Figure 2.3 - Accessing the Workbench in Windows 10

Workbench Classic and Workbench-SL can be found under 'ICONICS Tools', as shown in the following image. Users can also enter the relevant product name in the search bar. The search results will display the product link.

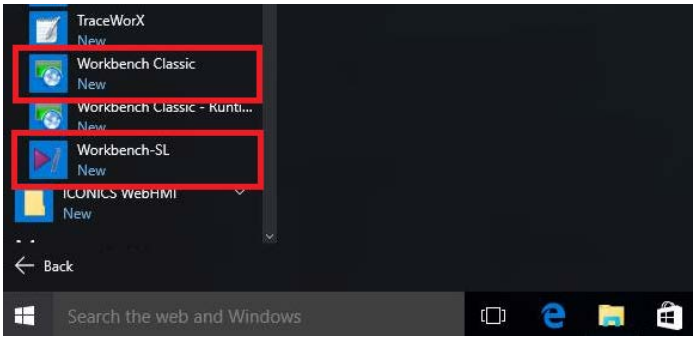


Figure 2.4 - Accessing Workbench Classic or Workbench-SL in Windows 10

Accessing the Workbench - Windows 8.1

In Microsoft Windows 8.1, users can look at all applications on the Start Screen to find ICONICS products. The Workbench is under 'ICONICS Product Suite', as shown below. Workbench Classic and Workbench-SL are under 'ICONICS Tools'. Users can also enter the relevant product name in the search bar. The search results will display the product link.

Figure 2.5 - Accessing the Workbench in Windows 8.1

2.3. Workbench User Interface

The Workbench configuration dashboard is divided into several main areas, as shown below:

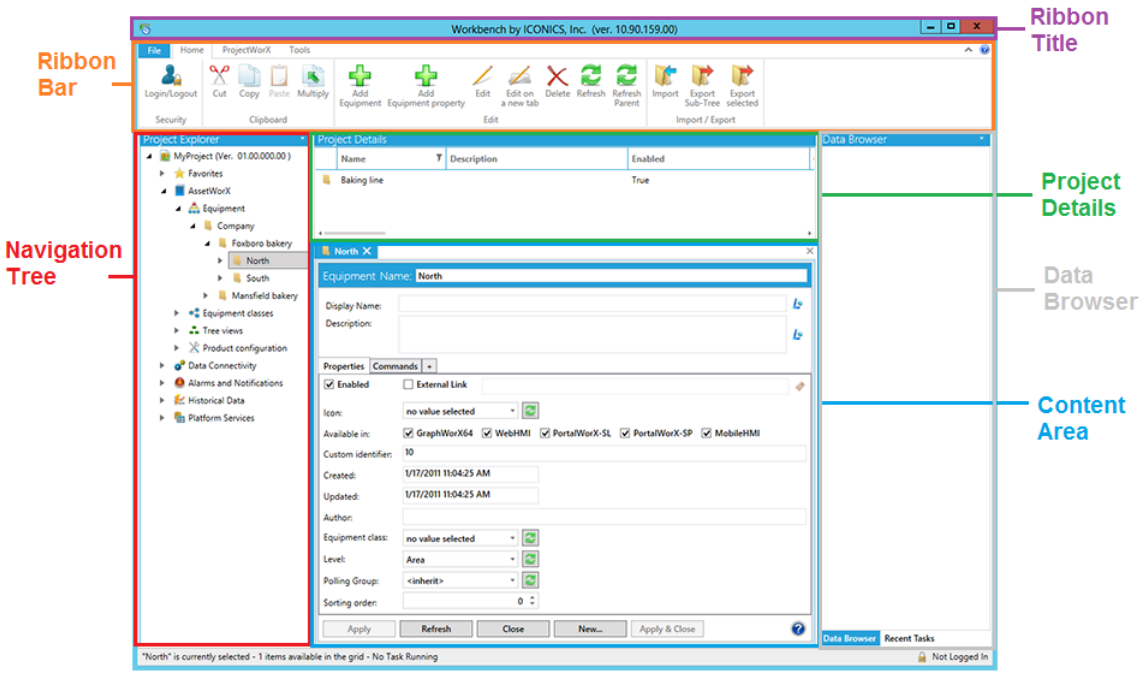


Figure 2.6 - Main Areas of the Workbench User Interface

The main areas include:

The **Ribbon Title**, which contains general information about the Workbench.

- A **Ribbon Bar** that includes several buttons for common actions for the different Workbench providers and general settings. This ribbon bar (in particular the **Home** ribbon) changes based on the element in the navigation tree that you have

selected. For more information about options that appear in the ribbon, see the Ribbon topic below.

- The **Navigation Tree**, which includes a hierarchy of folders and tags and lists the other providers within the Workbench.
- **Project Details** that allows you to search for folders and tags within the database and provides a list of the information within the next lowest level of the hierarchy (for example, if you have a folder selected in the navigation tree, that folder's tags will appear in list view).
- The **Content Area**, which displays options for configuring the element that you have selected in the navigation tree.
- The **Data Browser**, which provides an area to select from connected data sources for use within the providers. This section alternates with the **Recent Tasks** to show in progress tasks and their completion percentages.

2.3.1. The Ribbon

The ribbon for the Workbench is designed to help you quickly find the commands you need to complete a task. Commands are organized in logical groups. The ribbon has the following tabs:

- The **File** tab, which provides access to New Project, SQL Configuration, File Configuration, Connect to the Server, Disconnect from the Server, TraceWorX, FrameWorX Server Location, Platform Services, Challenge Security, Configure Displays, Login/Logout, and Restore Confirmation Dialogs.
- The **Home** tab, which changes depending on the item that is selected in the Project Explorer Navigation Tree.
- The **ProjectWorX** tab, which allows users to set connected databases (including a Unified Database), configure Deployment and Project Reporting options, update Project Properties, and Show/Hide Workbench items.
- The **Tools** tab, which provides access to Security (Login/Logout), MonitorWorX64 (Licensing, Redundancy), Server Connection (Connect, Disconnect), Tools (TraceWorX, FrameWorX Server Location, Platform Services, Challenge Security) and Usability (Quick Layouts, Data Browsers, Grid Edit Actions, Tree Edit Actions) set-up.

Examples of each are shown in the following images. You can minimize the ribbon by double-clicking on the ribbon tabs.

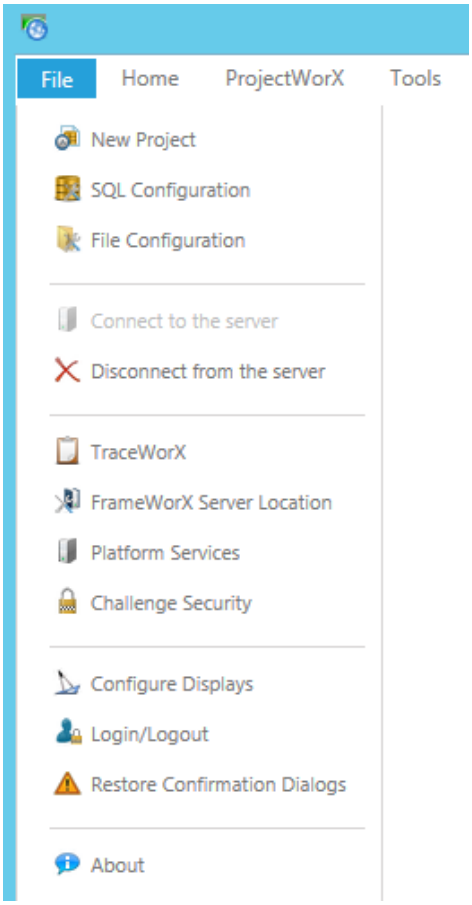


Figure 2.7 - Example of File Tab in the Workbench Ribbon

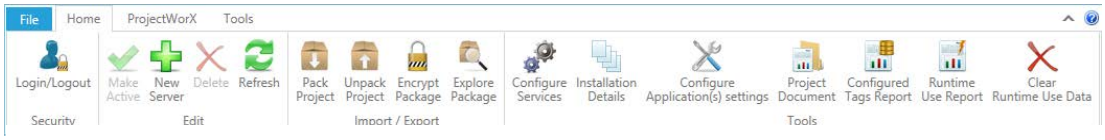


Figure 2.8 – Example of Home Tab in the Workbench Ribbon:
This Ribbon Changes with Your Project Explorer Selection

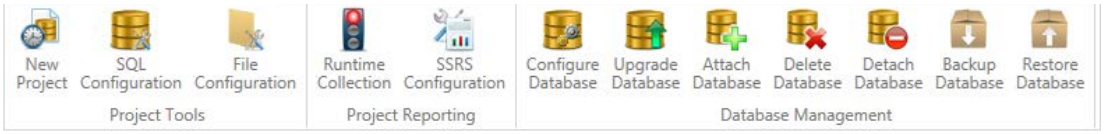


Figure 2.9 - Example ProjectWorX Tab in the Workbench Ribbon

Figure 2.10- Sample Tools Tab in the Workbench Ribbon

2.4. Tools and Providers

GENESIS64's Workbench can save all of your tools, application providers, and settings with your projects in their project files so that you can reopen and reuse the project easily. In each project are stored:

- Application providers
- Database configurations such as Global Aliasing, Language Aliasing, and other SQL databases
- Network services tools such as GenTray, SNMP, the FrameWorX Server, and the Security Server
- Workbench configuration settings that you apply to the Workbench window position, styles, modifications to the design surface, layouts, and templates

A project is typically stored where the servers are running but several servers may need to be accessed in order to run the different displays at an enterprise. As long as you have defined the servers' connections, those connections are stored with the project, too.

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-in, 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. For more information, refer to Security.

The Workbench is designed for concurrent configuration by multiple development team members over the internet. 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 also be used, to create plug-in from third-party and OEM content providers.

2.5. Providers in the Workbench

Development and runtime modes of all installed and licensed ICONICS GENESIS64 products can be accessed and used in the Workbench.

Even though the following providers are available in the Workbench, you can pick and choose which providers appear in the Workbench's Project Explorer, as shown in the following figure.

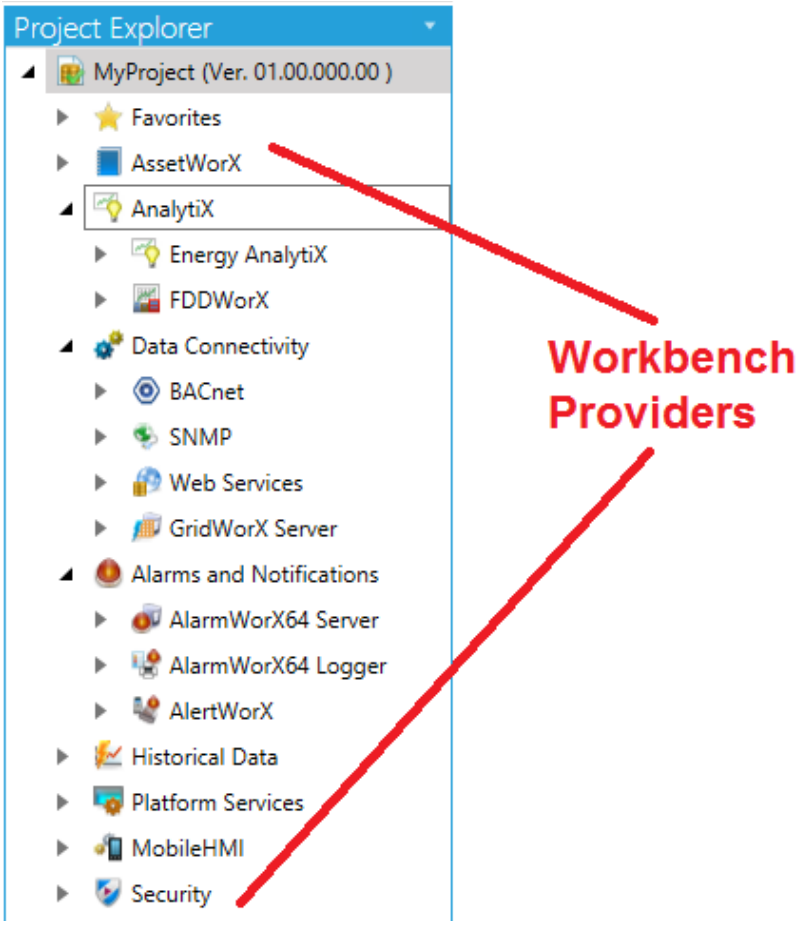


Figure 2.11 – Providers Listed Within the Workbench Project Explorer

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:

- **AlarmWorX Server** starts and stops the alarm server, as well as provides a configuration interface to configure alarms.
- **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.
- **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. The service can be started, stopped, and all of these data types and data instances can be created and modified within the UDM in the Workbench.

2.6. Data Browser in the Workbench

The Data Browser in the Workbench was created to provide an additional option for users' ease of use by:

- Simplifying Project Configuration
- Adding New Data Browser Controls to Browse Data Sources More Quickly and Easily
- Adding Drag and Drop Support

Users can configure projects easily and quickly, as the number of required clicks has been reduced consistently. The number of popup dialogs has also been considerably reduced.

2.6.1. Data Browser

Improvements to the Data Browser Control within the Workbench include:

- Tree-view-based Interaction
- Dockable Control
- Search Support
- Virtual Trees (Shown Under Assets)
- Multiple Views

Structural changes include:

- Reduced Number of Items Under Root
- Reduced Number of Levels
- Reorganization of Items

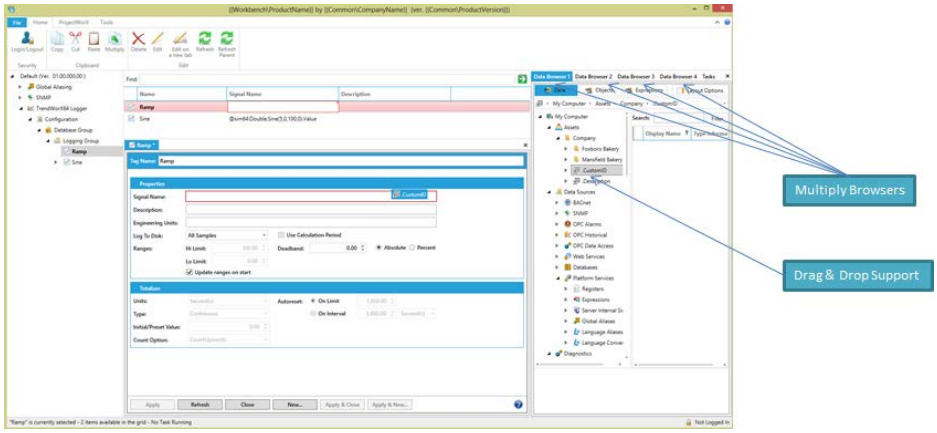


Figure 2.12 - Data Browser in the Workbench

2.6.2. Multiple Browsers and Drag and Drop Support

The Data Browser in the Workbench supports new Multiple Browser and Drag and Drop capabilities, featuring:

- Up to 4 Browsers Open as Modeless (Docked) Dialogs
- Possibility to Drag/Drop Sources from the Tag Browser Tree to Controls in Forms
- All Settings are Persistent and Restored Upon Next Start of the Application

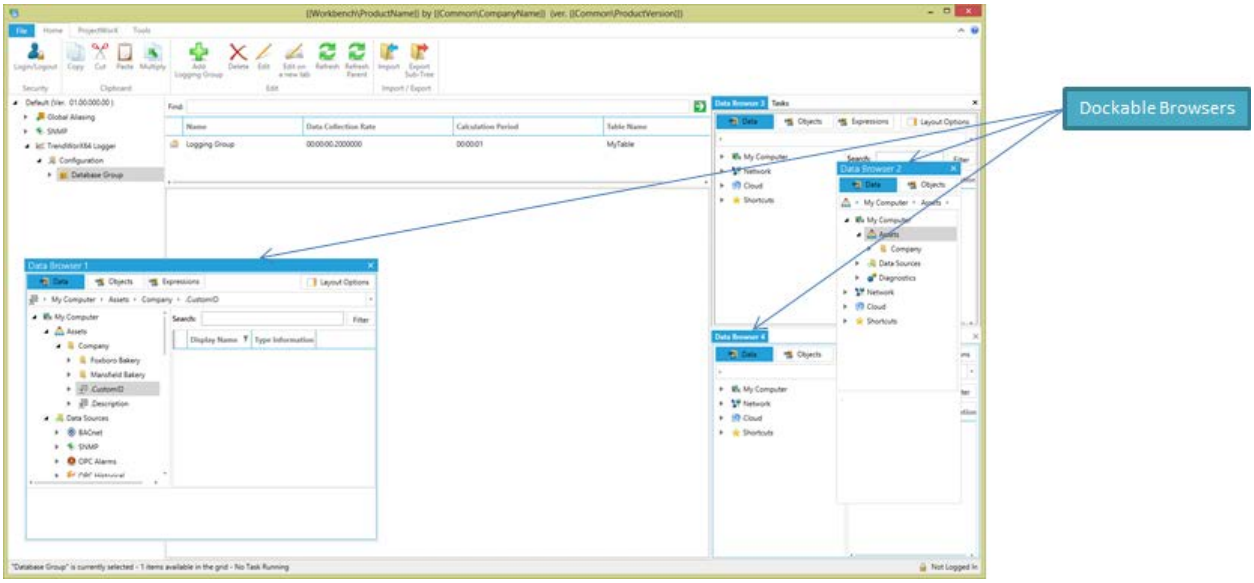


Figure 2.13 - Dockable Browsers in the Data Manager in the Workbench

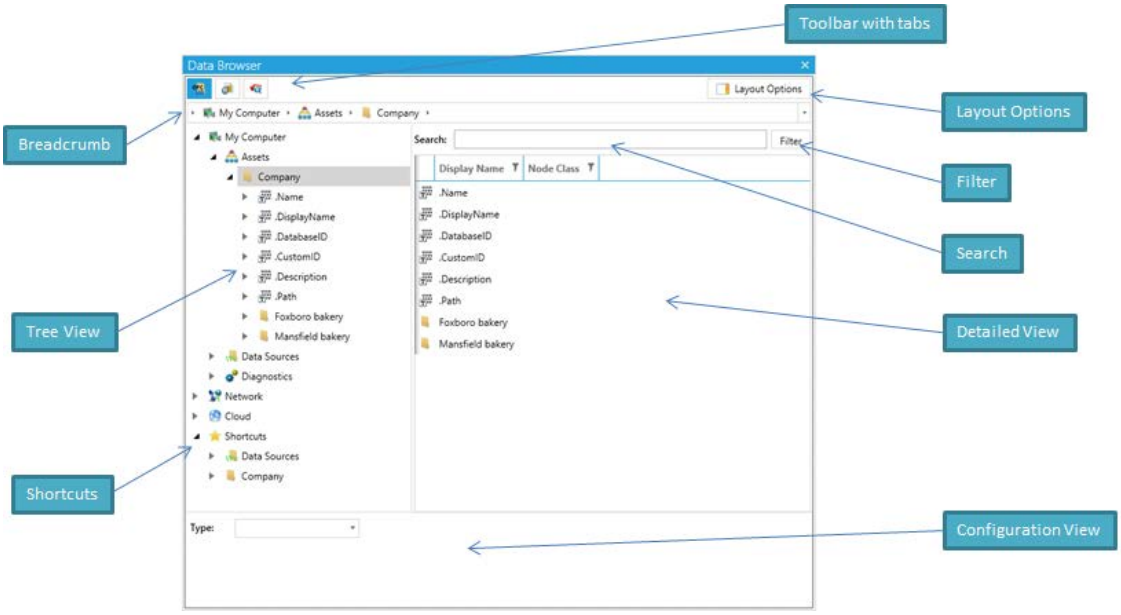


Figure 2.14 - Various Sections within the Data Browser in the Workbench

2.6.3. Additional Features

Additional features of the Data Browser within the Workbench include:

- Filtering by Type
- Search Using Wildcards - Starting from a Selected Node with the Possibility to Store the Search Expression
- Details View (As a Grid or Icons)
- Configuration View - To Set Additional Options for Global Aliases, Language Aliases, Language Conversions, Simulations
- Configurable Layout (Can Hide/Show Panels)

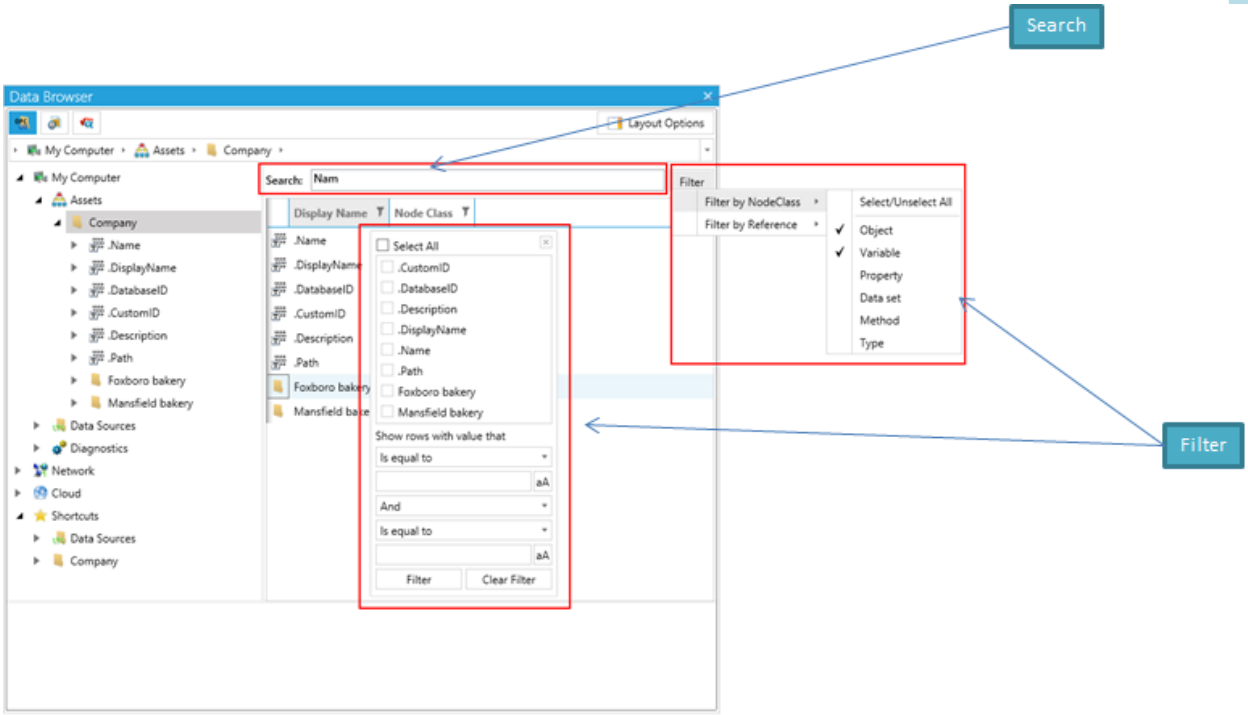


Figure 2.15 - Search & Filter in the Data Browser in the Workbench

2.6.4. Shortcuts

The Data Browser in the Workbench allows for shortcuts, which allow for the possibility to store a selected item and its subtree in the Shortcuts subtree. This can be accomplished:

- By Right-clicking the Item in the Tree or in Detailed View and Perform an Action in a Context Menu
- With Drag & Drop Support

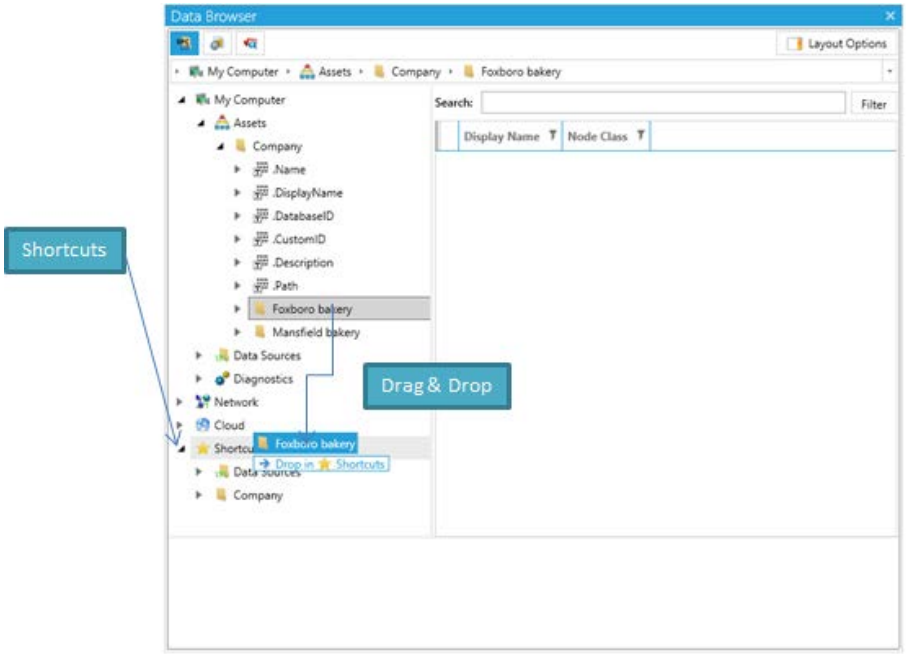


Figure 2.16 - Shortcuts within the Data Browser in the Workbench

2.6.5. Context Menu

The context menu within the Data Browser in the Workbench allows for the possibility to:

- Pin/Unpin a Selected item as Root and Browse Its Subtree Only
- Edit Values
- Add to, or Remove Items from, the List of Favorites

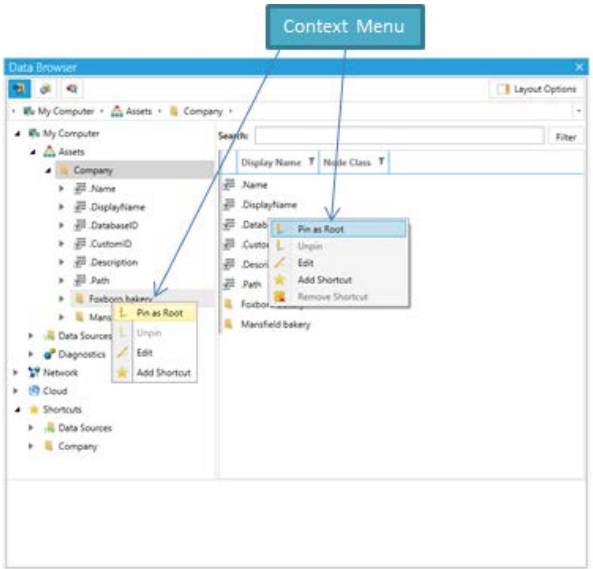


Figure 2.17 - Context Menu in the Data Browser in the Workbench

2.6.6. Configuration View

The configuration view within the Data Browser in the Workbench provides a section with specific options to configure:

- Language Aliases
- Global Aliases
- Language Conversions
- Simulations

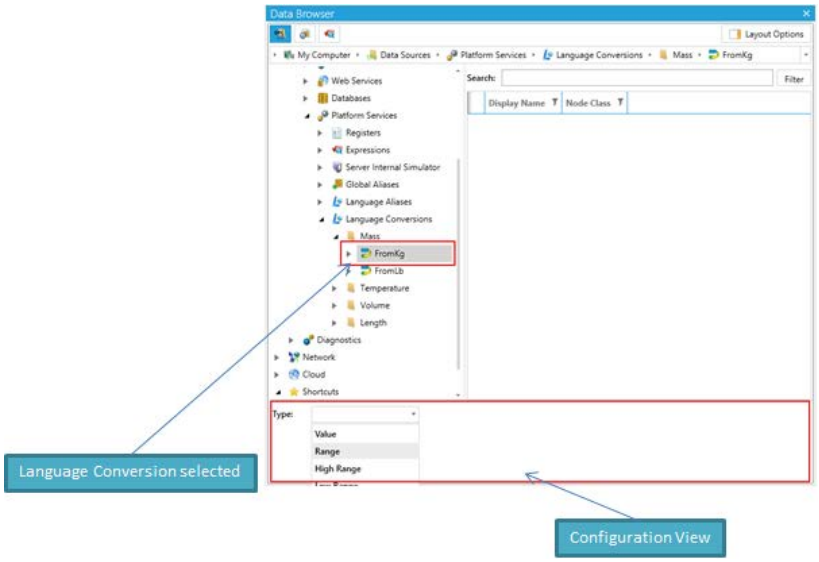


Figure 2.18 - Configuration View in the Data Browser in the Workbench

2.6.7. Additional Enhancements

Additional feature enhancements to the Data Browser in the Workbench include:

- The Use of New FrameWorX Classes
- Enhancement of the Point Manager to Expose Metadata for the Detailed View
- Modification of the Core Side to Show Tree Hierarchy As Expected

2.7. Filtering and Sorting Information in the Workbench

In the Workbench, you can filter and sort lists of objects in tabs. If there are a lot of calculations that are listed on the tab, you can filter out the ones you don't want to see by clicking any of the filter icons in the column heading on the tab. Clicking a filter icon pops up a small dialog where you can enter filter criteria. In the following example, we clicked the filter icon for the Name column, which popped up a filter window to filter table content to show rows for specific names.

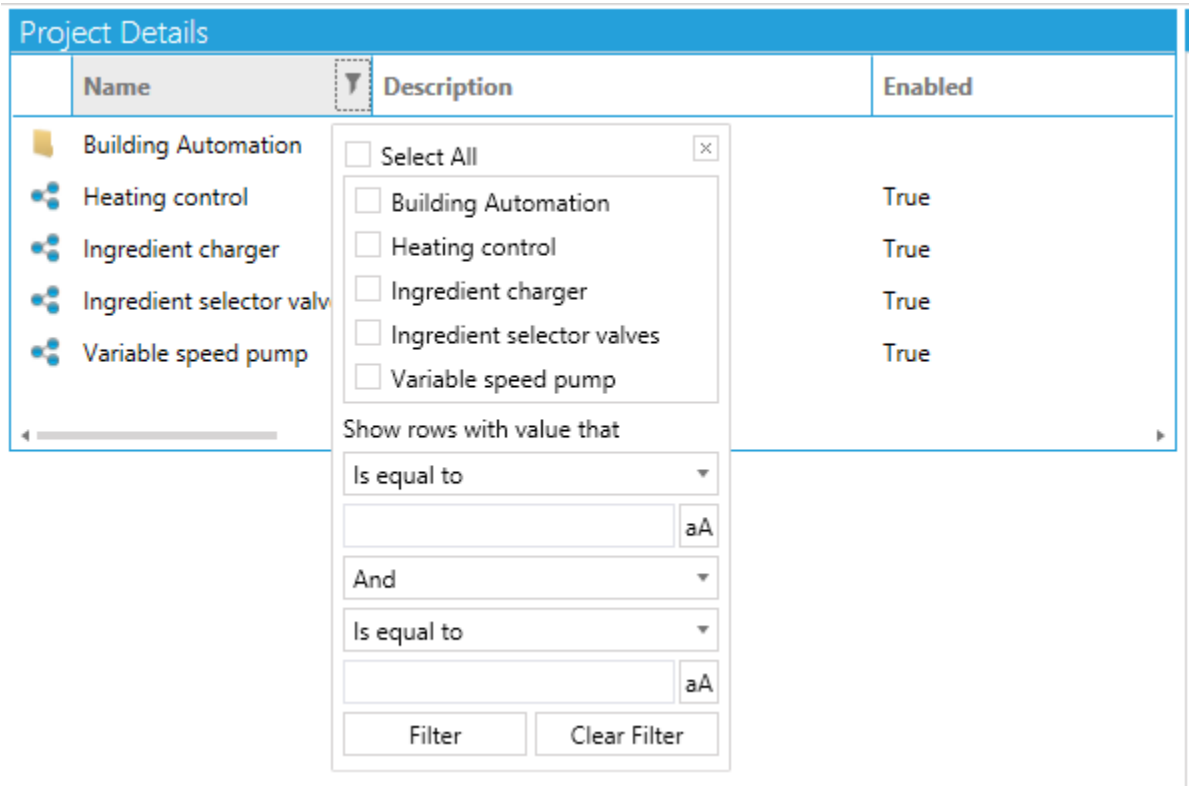


Figure 2.19 - Sorting and Filtering Options

Steps for filtering and sorting are provided in this topic:

- Sorting information
- Filtering information
- Clearing a filter
- Using Conditions in a Filter

2.7.1. Sorting Information

To sort the content of a table on the basis of values in one column, click on the column heading. This makes the sorting icon visible. Then click on the sorting icon to make the table sort in ascending or descending order, using the column as its basis.

2.7.2. Filtering Information

To filter information to display only what you want to see:

1. Start by thinking about what you want included in the listing and decide which column will let you choose that information.
2. Click on that column's filter icon in the column heading. This opens the column's popup filter.
3. Now specify your filter criteria. The data you select will be included in the table. There a few ways you can filter data:

   - You can select any or all of the items listed in the top half of the filter. Selecting no rows is the same as selecting all of them. But as soon as one item in the list is selected, only the selected items are included in the results. The following image is an example of where all items were selected using the Select All check box, and then individual items were deselected.

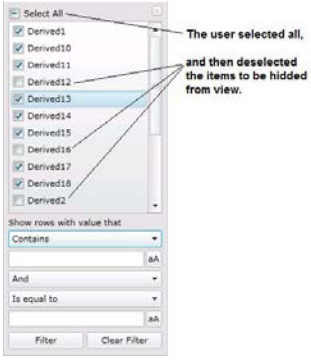


Figure 2.20 – Selection Example

   - You can also (or alternatively) enter one or two conditions. The following image shows and example of how to see the fields where you enter conditional information. For example, you can include only those rows that have "Foxboro" in the description, or exclude only those rows. Under the **Show rows with value that** heading, you can enter a condition (for example, **Starts with**) and then optionally choose a boolean value (**And** or **Or**), and then another condition. For help using this part of the popup filter, see the following “Using Conditions in a Filter" topic.

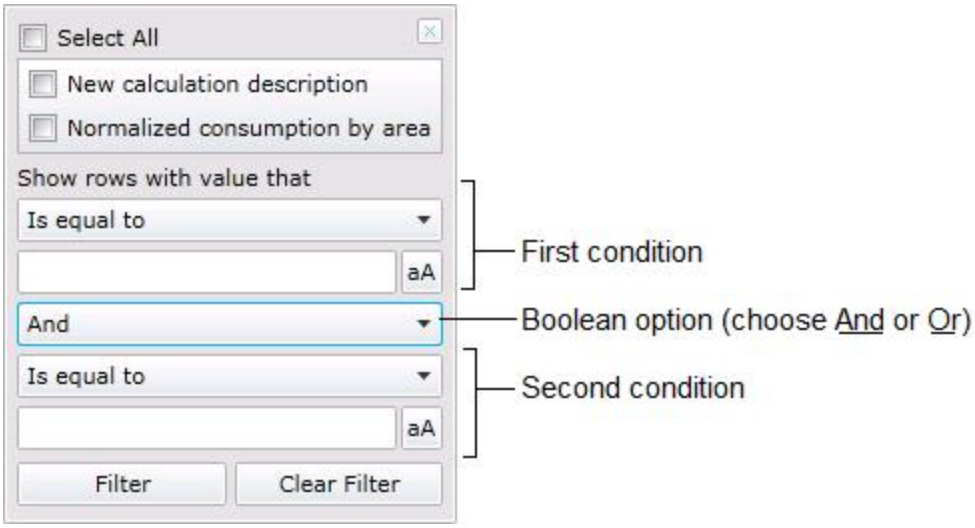


Figure 2.21 - Conditional Options for Filtering Rows in a Table

4. Click the **Filter** button to update the results in the table. In some cases, as you enter values, the data in the table is instantly filtered so you can see the result; in others you will have to click the Filter button to see updated results.

5. You can filter on more than one column. For example, you can filter the Name column to eliminate a few known rows, then filter on the Description column to display rows that rows that contain the word "West" in the description.

6. When you are done with a filter, simply click outside the filter box or click the popup box's **X** button to close it.

2.7.3. Clearing a Filter

You can tell which columns have a filter in place by the appearance of the column filters.

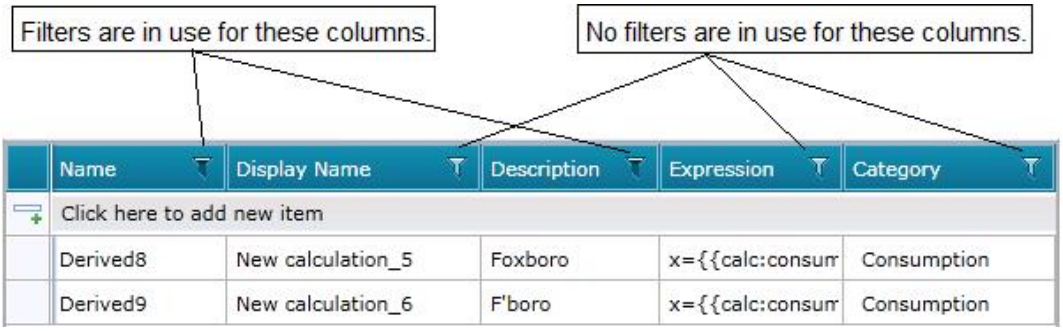


Figure 2.22 - Appearance of the Filter Icons in Column Headings

To clear a filter so that no rows are hidden based on that column:

1. Click on the filtered column's filter icon in the column heading. This opens the column's popup filter.
2. Click the Clear Filter button.
3. Click outside the popup filter or click the popup box's **X** button to close it.

2.7.4. Using Conditions in a Filter

In a column's filter popup window you can include up to two conditions for listing information, as shown in the sample below.

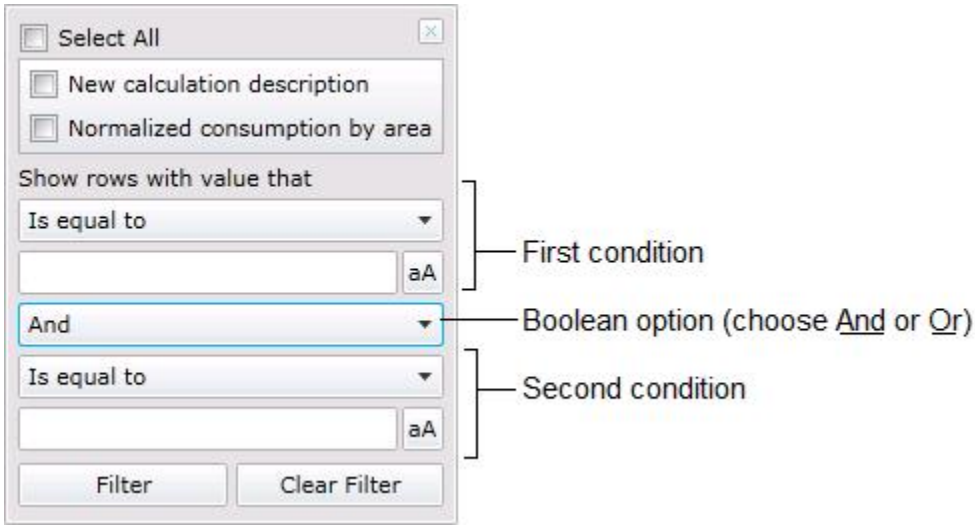


Figure 2.23 - Conditional Options for Filtering Rows in a Table

Each condition is made up the condition and the values for that condition. For the condition, you can select one of the following and enter an appropriate value in the field below your selection. If the value must be case-specific, you can click the **aA** button, also.

- **Is equal to**: Rows must contain the exact string that you specify as the value -- no more and no less.
- **Is not equal to**: Rows must not contain the exact string that you specify as the value.
- **Starts with**: Rows must begin with the string you specify as the value.
- **Contains**: Rows must have the string you specify as the value. For example, you might use this to list vendors with a phone number that contains "1-508".
- **Does not contain**: Rows must not have the string you specify as the value.

- **Ends with**: Rows must end with the string you specify as the value. For example, if you enter "USA" as the value, a row that has "Foxborough East USA" will be included but a row that has "Foxborough West U.S.A." will be filtered out.

Below are some examples.

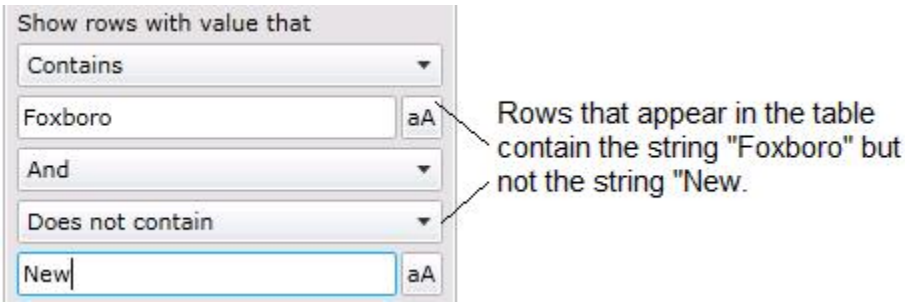


Figure 2.24 - Example 1

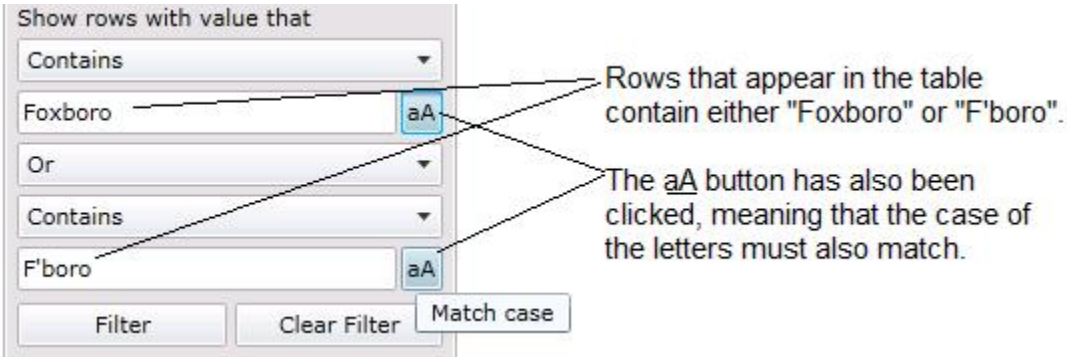


Figure 2.25 - Example 2

Lab: Workbench – Configuring Systems *L2*

2.1. Lab Overview

In this section, we will cover how to get started creating and configuring a new project and also give you a chance to practice inserting OPC tags into GraphWorX64 to view live data.

Estimated time to complete this lab: **25 minutes**

2.1.1. Task Checklist

In this lab you will complete the following tasks:

- 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. Workbench Navigation

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

1. Since 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 Product Suite → Workbench**.

Figure L2A.1 – The Workbench Ribbon Bar

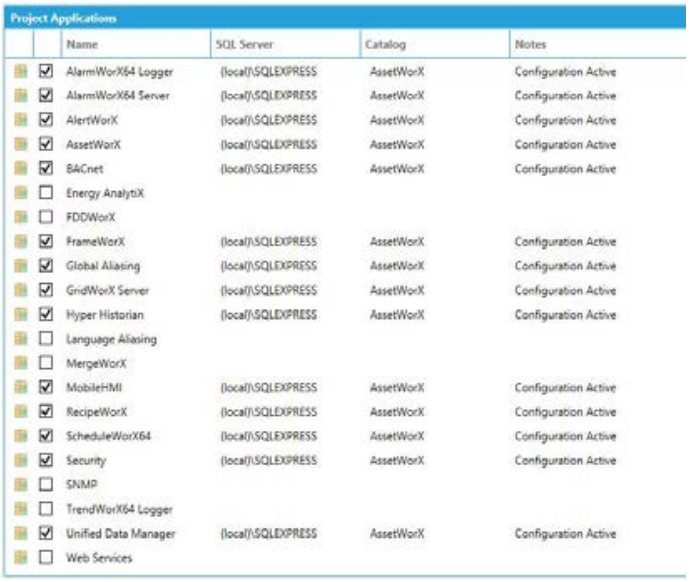
2. As explained in the lecture portion of this chapter, there are four sections within Workbench: The **Ribbon Bar** (shown in the previous image), the **Project Explorer** (which lists the providers), the **Project Details** (used for displaying and configuring the selected items), and the **Data Browser/Recent Task** (for viewing configured nodes and data sources as well as recently executed project tasks).

2.5. Workbench Provider Selection

In the following steps, we will edit the project and disable the providers that we will not use during training and link the remaining providers to the unified configuration database created in the last step.

1. Select **icTonicsProject** from the Project Explorer
2. Select **Configure Application Settings** from under the Home tab → Tools
3. Enter your name in the **Author:** field.

   Note: The **Last Update** field is maintained automatically.
4. Scroll down to the **Project Applications** and **uncheck** the following providers:
   - **FDDWorX**
   - **Energy Analytix (if it exists)**
   - **Language Aliasing**
   - **MergeWorX**
   - **SNMP**
   - **TrendWorX Logger**
   - **Web Services**
5. Click **Apply.** The Project Applications window should resemble the following figure.



Project Applications

	Name	SQL Server	Catalog	Notes
☑	AlarmWorX64 Logger	(local)\SQLEXPRESS	AssetWorX	Configuration Active
☑	AlarmWorX64 Server	(local)\SQLEXPRESS	AssetWorX	Configuration Active
☑	AlertWorX	(local)\SQLEXPRESS	AssetWorX	Configuration Active
☑	AssetWorX	(local)\SQLEXPRESS	AssetWorX	Configuration Active
☑	BACnet	(local)\SQLEXPRESS	AssetWorX	Configuration Active
☐	Energy AnalytiX			
☐	FDDWorX			
☑	FrameWorX	(local)\SQLEXPRESS	AssetWorX	Configuration Active
☑	Global Aliasing	(local)\SQLEXPRESS	AssetWorX	Configuration Active
☑	GridWorX Server	(local)\SQLEXPRESS	AssetWorX	Configuration Active
☑	Hyper Historian	(local)\SQLEXPRESS	AssetWorX	Configuration Active
☐	Language Aliasing			
☐	MergeWorX			
☑	MobileHMI	(local)\SQLEXPRESS	AssetWorX	Configuration Active
☑	RecipeWorX	(local)\SQLEXPRESS	AssetWorX	Configuration Active
☑	ScheduleWorX64	(local)\SQLEXPRESS	AssetWorX	Configuration Active
☑	Security	(local)\SQLEXPRESS	AssetWorX	Configuration Active
☐	SNMP			
☐	TrendWorX64 Logger			
☑	Unified Data Manager	(local)\SQLEXPRESS	AssetWorX	Configuration Active
☐	Web Services			

Figure L2A.2 – Disabled Providers

2.6. GraphWorX64 Preferences

In this section, we will change the default location where files are saved.

1. Open GraphWorX64 from the start menu.
2. On the lower right side of the window, click on the **Preferences** tab.
3. With the **General Settings** category selected, locate the DefaultSaveLocation property in the lower panel and click on the browse button.
4. From the data browser, locate and select the **C:\Training** folder.

   Note: When you close GraphWorX64, the preferences that have been changed become permanent.

2.7. OPC Connections

2.7.1. OPC DA

The OPC Data Access Specification is the first of a group of specifications known as the Classic OPC Specifications. OPC Data Access is a group of standards that provides specifications for communicating real time data from data acquisition devices such as PLC's to display and interface devices such as Human Machine Interfaces (HMI). The specifications focus on the continuous communication of data.

2.7.2. OPC UA

OPC Unified Architecture is the most recent Object Linking and Embedding for Process Control (OPC) specification from the OPC Foundation and differs significantly from its predecessors. The first version of Unified Architecture was released in 2006, after 3 years of specification work and another year of prototyping. The Foundation's goal for this project was to provide a path forward from the original OPC communications model (namely COM/DCOM) to a cross-platform service-oriented architecture (SOA) for process control, while enhancing security and providing an information model.

2.7.3. Configuring and Browsing OPC Tags

1. To confirm the Modbus OPC conector is available for browsing, open the **Modbus** configuration from the **Workbench**. Expand **Data Connectivity** in the Explorer and select **Modbus**.

2. In the **Home** ribbon confirm the **Point Manager** Service traffic light is **green**. If the light is not green, **Left-click** on the traffic light icon to start the service. Click OK in any popups that open. This launches the **Modbus Point Manager.**

3. Return to **GraphWorX64** and add a Process Point to the display. Select the **Dynamics** tab; select the **Process Point** (Text Object) and draw a box on the Canvas the appropriate size for your Process Point to be visible.

11. From the **Dynamics** tab on the right, click on the ellipsis […] button to the right of **Data → DataSource** to open the data browser

12. Browse from the **Simulation** tab to **Application Simulation** and select the localsim:ramp tag. Type the Name **Ramp1** into the (Name) field. Change the **UpdateRate** field to **500** and click **OK**. Notice that the syntax in the **DataSource** property will look like this:

localsim:ramp:Ramp1::::::500

13. If this were an OPC Tag the prefix **@** and suffix **.Value** would indicate that this is a tag from an OPC DA server and not an OPC UA server. The OPC DA tag might look like this:
@Kepware.KEPServerEX.V5\Channel_0_User_Defined.Ramp.Ramp1.Value

14. You can similarly browse for OPC UA tags from within the data browser.

15. When finished, save the display as **LocalSimRamp1_Display.gdfx** and enter **Runtime** to view tag(s) being populated.

This concludes this lab.

GraphWorX64 Basics 3

As you know by now, GraphWorX64's design surface is a canvas upon which you can display real-time or historical data from any OPC, BACnet, database, SNMP, or OPC-UA compliant data source in a meaningful way so that the state of your equipment, processes, conditions, or anything else you can quantify is instantly recognizable by an operator at a glance. Displays based on GraphWorX64 run water treatment plants, nuclear power plants, pharmaceutical production lines, and are found in dozens of industries where automation control is necessary for the overall product or service. 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.

3.1. Objectives

After completing this chapter, you should know how to do the following:

- Work with 2D shapes and objects.
- Work with text objects.
- Work with layers.
- 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.

3.2. 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
- 3D Viewport
- Embedded Image
- Referenced Image

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 3.1 shows an example of the different commands that you can use to create a complex shape.

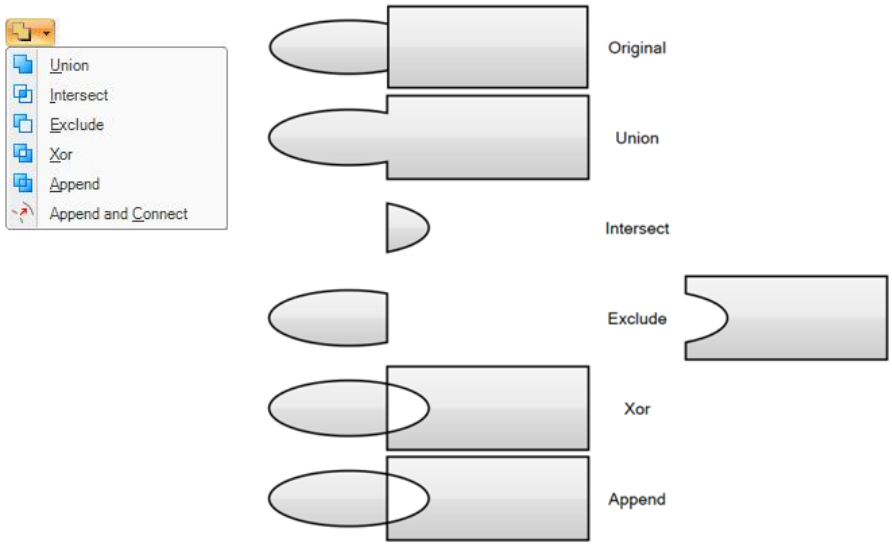


Figure 3.1 - Combine Basic Shapes into Complex Shapes

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 3.2.

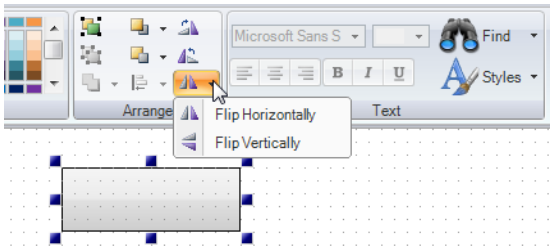


Figure 3.2 - Flip and Rotate

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 3.3. 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.

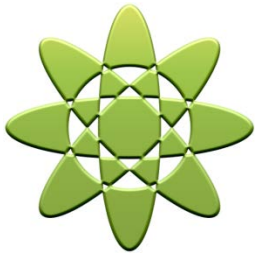


Figure 3.3 - Complex Shape Created Using Ellipses with a Predefined Style

Combining these shapes with smart symbols and controls will give you the ability to create easily dashboards (like the one shown in Figure 3.4), processing and electrical systems, facilities, campuses, and much, much more.

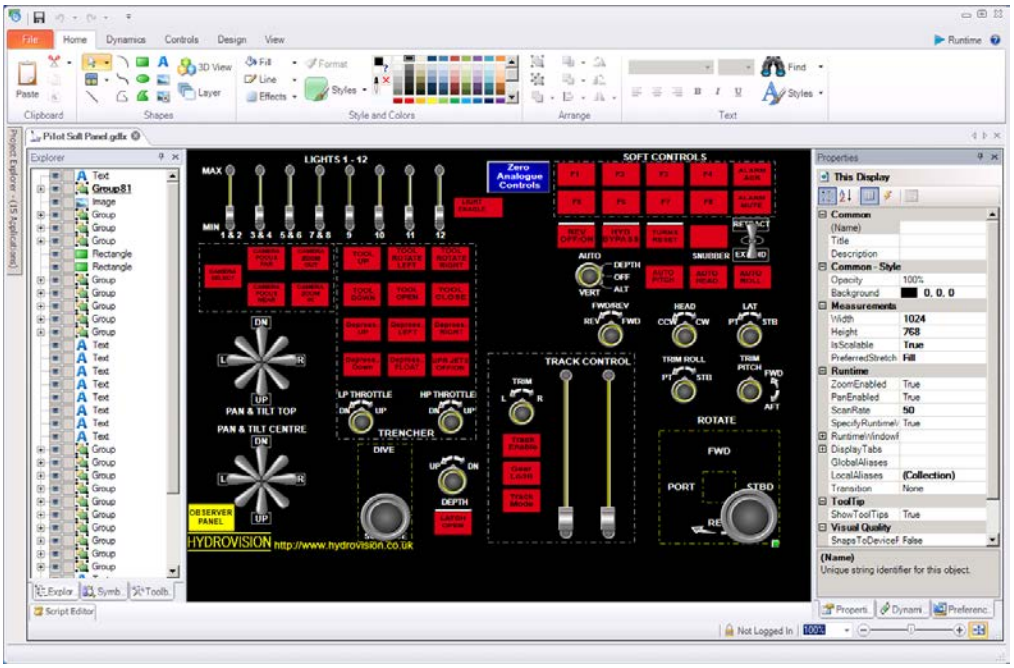


Figure 3.4 - Example of a Display in Configuration Mode

In Figure 3.4, 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.

3.3. 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.

3.3.1. 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 3.5.

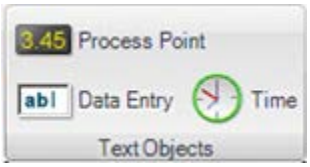


Figure 3.5 - Text Objects

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.

3.3.2. 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 3.6 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 3.6 - 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 3.7, 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.

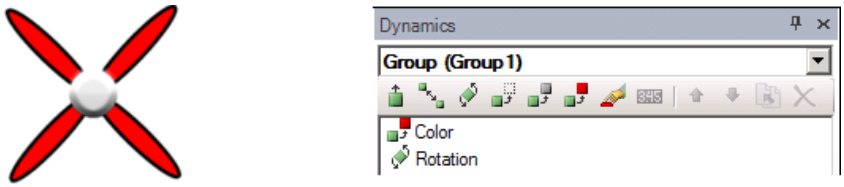


Figure 3.7 – A Ventilator (Left), with its Dynamics in the Dynamics Explorer (Right)

3.3.3. 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 3.8, 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.

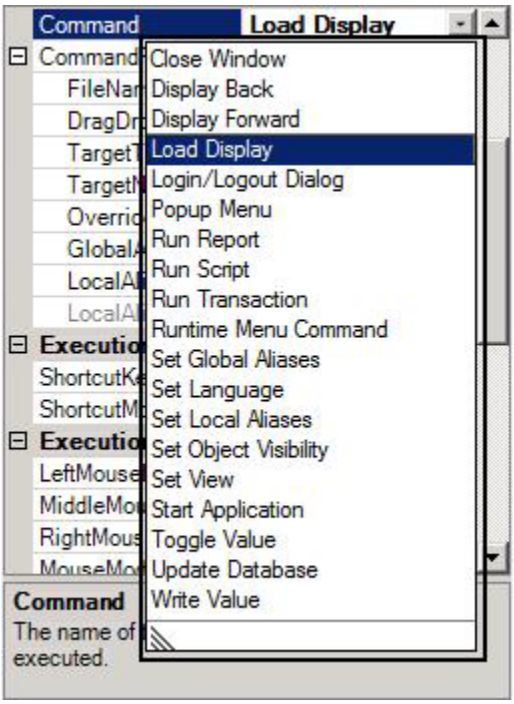


Figure 3.8 - Pick Action Dynamic Properties

Local Commands are: Close Window, Display Back, Display Forward, Load Display, Login/Logout Dialog, Popup Menu, Run Report, Run Script, Run Transaction, Runtime Menu Command, Set Global Aliases, Set Language, Set Local Aliases, Set Object Visibility, Set View, Start Application, Toggle Value, Update Database, and Write Value. Click a command above to see the command parameters that need to be set.

Global Commands are: Acknowledge, Call Method, Create Pen, Custom Command, Delete Pen, Expand Collapse, Export Image, Export Statistic, Go to Location, Group, Load Alarm Configuration, Load DataGrid Configuration, Load Energy Data, Load Executed Reports, Load Facility Data, Load Graphics Display, Load Report, Load Trend Configuration, Login Dialog, Navigate, Open URL, Print, Run Report, Run Transaction, Save Configuration, Select Asset, Select Element, Set Filter, Set Filter, Set Freeze Mode, Set Global Aliases, Set Language, Set Period, Set Time Range, Set Visibility, Sort, and Write Value.

3.4. Close Window

The **Close Window** command closes the GraphWorX64 display window when this pick action occurs during runtime. In a Silverlight display, this command can close only a GraphWorX popup window, not the main Internet Explorer (IE) window.

Command parameters are:

- **TargetType** is the type of window to be opened. Some of these selections are window types, some are windows hierarchy (relative paths), and some are absolute settings (such as the Root window). For more information about windows, refer to Windows Dynamics. Choices are:
  - **Current Display** opens the same GraphWorX64 display; this is a refresh of the currently open GraphWorX64 display. Not supported in a Silverlight display.
  - **Popup Window (Display Dependent)**
  - **Popup Window (Display Dependent - Modal)**
  - **Popup Window (Window Dependent)**
  - **Popup Window (Window Dependent – Modal)**
  - **Popup Window (Independent)**
  - **Embedded Window** is a window contained within another window. Not supported in a Silverlight display.
  - **Parent Window** is the window from which another window was created. Not supported in a Silverlight display.
  - **Root Window** is the first window shown in a display, from which all other windows are created. Not supported in a Silverlight display.
  - **New Instance (Separate Process)**
  - **Any Named Window**
- **TargetName** is the actual window to be opened. Click the ellipsis button to open the Data Browser so you can identify the target.

3.5. Display Back and Display Forward

GraphWorX64 maintains a history of the last 50 displays. You can navigate through the display history using the pick commands **Display Back** and **Display Forward**. These commands work similarly to Microsoft Internet Explorer's back/forward feature.

Info

Note: The display file history (forward/back commands) remembers the initial alias settings specified when a display is opened and will reset those aliases when you go back to that display in the file history.

- Display Back acts like the **Back** button in a browser, moving users through a browse sequence of GraphWorX64 displays. Using this pick action lets runtime users navigate backward through the display history.
- Display Forward acts like the **Forward** button in a browser, moving users through a browse sequence of GraphWorX64 displays. Using this pick action lets runtime users navigate forward through the display history. Command parameters are:

Command parameters for Display Back and Display Forward are:

- **TargetType** is the type of window to be opened. Some of these selections are window types, some are windows hierarchy (relative paths), and some are absolute settings (such as the Root window). For more information about windows, refer to Windows Dynamics. Choices are:
  - **Current Display** opens the same GraphWorX64 display; this is a refresh of the currently open GraphWorX64 display.
  - **Popup Window (Display Dependent)**
  - **Popup Window (Display Dependent - Modal)**
  - **Popup Window (Window Dependent)**
  - **Popup Window (Window Dependent – Modal)**
  - **Popup Window (Independent)**
  - **Embedded Window** is a window contained within another window.
  - **Parent Window** is the window from which another window was created.
  - **Root Window** is the first window shown in a display, from which all other windows are created.
  - **New Instance (Separate Process)** is a new instance of the same window.
  - **Any Named Window** is any window; specify the window name in the TargetName property.
- **TargetName** is the actual window to be opened. Click the ellipsis button to open the Data Browser so you can identify the target.

3.6. Load Display

The **Load Display** command opens a GraphWorX64 display window of the type that you specify with the properties that you define. The window can be dependent or independent, modal or non-modal, independent or embedded, or it can even refer to the current display (which is a refresh operation). You can specify other parameters such what loads in the display.

For help understanding how to manage windows through commands, refer to the topic Windows Dynamics. Aliases can be used with this command; for more information, refer to Aliasing and Pick Actions.

Command parameters are:

- **FileName** is the file name of the target window to be loaded during runtime; you can browse files or use the Data Browser.
- **DragDrop** — **True** lets users move the window to a new location by dragging it there; **False** makes the window stationery.
- **TargetType** is the type of window to be opened. Some of these selections are window types, some are windows hierarchy (relative paths), and some are absolute settings (such as the Root window). For more information about windows, refer to Windows Dynamics. Choices are:
  - **Current Display** opens the same GraphWorX64 display; this is a refresh of the currently open GraphWorX64 display.
  - **Popup Window (Display Dependent)**
  - **Popup Window (Display Dependent - Modal)**
  - **Popup Window (Window Dependent)**
  - **Popup Window (Window Dependent – Modal)**
  - **Popup Window (Independent)**
  - **Embedded Window** is a window contained within another window.
  - **Parent Window** is the window from which another window was created.
  - **Root Window** is the first window shown in a display, from which all other windows are created.
  - **New Instance (Separate Process)** is a new instance of the same window.
  - **Any Named Window** is any window; specify the window name in the TargetName property.
- **TargetName** is the actual window to be opened. Click the ellipsis button to open the Data Browser so you can identify the target.
- **OverrideWindowProperties** — **True** overrides the properties of the window into which the display is loaded; **False** leaves the window's properties in place.

- **GlobalAliases** — Click the ellipsis button to open the Set Global Themes and Aliases dialog box, where you can set any global aliases that will be needed for the window.
- **LocalAliasCommandType** — Specify how the local aliases will be set for the display to be loaded. Choose one of the following:
  - **None** - There is no local alias.
  - **From Data** - The local alias is provided by data. Specify the data in the **LocalAliasCommandData** property described below.
  - **From File** - The local aliases and their values are stored in a file. Specify the file in the **LocalAliasCommandData** property described below. For information about how and why this file is created, refer to "Saving a File of Local Alias Values for Use with Pick Actions".
- **LocalAliasCommandData** — Specify the file of local aliases and their values (if you chose From File in the LocalAliasCommandType property above), or open the Data Browser to select the data source from which the local alias values are obtained.

Note: You can browse for files under the "Load DataGrid Configuration" command as it is for Load Display pick action.

3.7. Login/Logout Dialog

The **Login/Logout Dialog** command opens a dialog in which the runtime user can log in or log out. For more information, refer to Logging in and out of GraphWorX64. There are no command parameters for this command.

3.8. Popup Menu

The **Popup Menu** command opens a popup menu. Popup menus are often used to allow runtime users to switch aliases or languages, or to open one display out of a set. They are a great way of providing user interaction so that your display can have situational logic. You can configure the size, font, and position of the popup.

The command parameter of **MenuPosition** determines the position of the popup menu in relation to the parent object. Select **MousePoint** to pop up the menu at the current mouse position at runtime. Specify **Left**, **Right**, **Top**, or **Bottom** to pop up the menu to the left or right, top or bottom of the parent graphical object.

3.9. Run Report

The **Run Report** command executes an ICONICS ReportWorX report during runtime. For more information about ReportWorX, refer to the ReportWorX documentation on the ICONICS BizViz product CD. This command is not supported in Silverlight displays.

Run Report's command parameters are:

- **ServiceURL**—Specify the URL address of the ReportWorX Web Service.
- **AccessKey**—Enter the password needed to access the ReportWorX Web Service.
- **ReportName**—Specify the report to be executed. Click the ellipsis button to select the report to be activated by the pick dynamic. In the Report Name dialog box that opens, refresh the list by the clicking the Get Reports button. Then select the report and click OK.
- **GlobalAliases**—Clicking within this field allows users to select aliases from either the "Set Global Themes and Aliases" tool or the Data Browser.
- **Parameters**—Specify the parameters, if any, to be used with the selected report. For each parameter, use the `"@parameter"` syntax (for example, `@Parameter1=34`). If you are using multiple parameters, separate each parameter using the pipe character "|" (for example, `@Parameter1=34|@Parameter2=128`). Clicking the ellipsis button […] opens the Data Browser.
- **Identity**—If a user name and password are required to access the ReportWorX server, specify the user login to be used for the report. Choose either **InteractiveUser** (to log in using the ID of the runtime user who is logged in to GraphWorX) or **ThisUser** then provide the ID and password to gain access.
- **User**—If the **Identity** is **ThisUser**, specify the ID for gaining access to the ReportWorX server. The ID is case sensitive.
- **Password**—If the **Identity** is **ThisUser**, specify the password for gaining access to the ReportWorX server. Passwords are case sensitive.
- **OutputTo**—Optionally, type the name of the text object the report output is to be sent to. If necessary, add the label object to the GraphWorX64 display, then return to this property and type its name here. For more information, refer to Labels.

3.10. Run Script

The **Run Script** command runs a JScript .NET script only. This command is not supported in Silverlight displays. There are no command parameters for this command

3.11. Run Transaction

The **Run Transaction** command executes an ICONICS BridgeWorX transaction during runtime. For more information about BridgeWorX, please refer to the BridgeWorX documentation on the ICONICS BizViz product CD. This command is not supported in Silverlight displays.

Command parameters for Run Transaction are:

- **ServiceURL**—Specify the URL address of the BridgeWorX Web Service.
- **AccessKey**—Enter the password needed to access the BridgeWorX Web Service.
- **Transaction**—Click the ellipsis button to select the transaction to be activated by the pick dynamic. In the Transaction Name dialog box that opens, refresh the list by the clicking the Get Transactions button. Then select the transaction and click OK.
- **GlobalAliases**—Clicking within this field allows users to select aliases from either the "Set Global Themes and Aliases" tool or the Data Browser.
- **Parameters**—Specify the parameters, if any, to be used with the selected transaction. For each parameter, use the `"@parameter"` syntax (for example, `@Parameter1=34`). If you are using multiple parameters, separate each parameter using the pipe character `"|"` (for example, `@Parameter1=34|@Parameter2=128`). Clicking the ellipsis button […] opens the Data Browser.
- **Identity**—If a user name and password are required to access the BridgeWorX server, specify the user login to be used for the report. Choose either **InteractiveUser** (to log in using the ID of the runtime user who is logged in to GraphWorX) or **ThisUser** then provide the ID and password to gain access.
- **User**—If the **Identity** is **ThisUser**, specify the ID for gaining access to the BridgeWorX server. The ID is case sensitive.
- **Password**—If the **Identity** is **ThisUser**, specify the password for gaining access to the BridgeWorX server. Passwords are case sensitive.
- **OutputTo—**Optionally, type the name of the text object the transaction output is to be sent to. If necessary, add the label object to the GraphWorX64 display, then return to this property and type its name here. For more information, refer to "Labels".

3.12. Runtime Menu Command

The Runtime Menu Command executes a command that is currently available on the GraphWorX64 runtime menu.

Command parameters are:

- **Command**—Choose the runtime menu command to be used. To gain a better understanding about how some of the runtime menu commands work, refer to the Runtime Menus in GraphWorX topic. Options you can choose from include the following:
  - **File** menu: File Open Dialog, Print, Print Dialog, Print Page Setup, Print Preview, Print Toggle White Background, Print Area Current View, Print Area Entire Display, Recent Files Card View, Exit.
  - **View** menu: Zoom In, Zoom Out, Unzoom, Fit to Window, Home View, Toggle Auto Scroll Bars, Toggle Navigation Bar, Toggle Status Bar, Toggle Mail Menu, Toggle Ribbon, History Card View, History Back, History Forward.
  - **Tools** menu: Login Dialog, Select Language Dialog, Carousel, Carousel Card View, Toggle Cache Displays.
  - **Help** menu: Help Topics, About
  - **Configuration Mode**
- **TargetType** is the type of window to be opened by the command. Some of these selections are window types, some are windows hierarchy (relative paths), and some are absolute settings (such as the Root window). For more information about windows, refer to Windows Dynamics. Choices are:
  - **Current Display** opens the same GraphWorX64 display; this is a refresh of the currently open GraphWorX64 display.
  - **Popup Window (Display Dependent)**
  - **Popup Window (Display Dependent - Modal)**
  - **Popup Window (Window Dependent)**
  - **Popup Window (Window Dependent – Modal)**
  - **Popup Window (Independent)**
  - **Embedded Window** is a window contained within another window.
  - **Parent Window** is the window from which another window was created.
  - **Root Window** is the first window shown in a display, from which all other windows are created.
  - **New Instance (Separate Process)** is a new instance of the same window.
  - **Any Named Window** is any window; specify the window name in the TargetName property.

- **TargetName** is the actual window to be opened. Click the ellipsis button ... to open the Data Browser so you can identify the target.

3.13. Set Global Aliases

The **Set Global Aliases** command sets a global alias theme at runtime for the This Display object when the pick dynamic is triggered. Global aliases are stored and maintained in a SQL Server database.

This setting has one command parameter:

- **GlobalAliases**—Click the ellipsis button ... then select the global alias theme from the Data Browser. Refer to Global Aliasing for more information. For additional information, refer to Aliasing and Pick Actions.

3.14. Set Language

The **Set Language** command sets the runtime language to a language of your choosing. At runtime, string substitutions (which the Language Alias System SQL database manages) translate items in the runtime display to the specified language. For example, use this command in a button that Spanish-speakers can click in order to see the runtime display in Spanish. Refer to the topic Language Aliases for more details. For additional information, refer to Aliasing and Pick Actions.

This setting has one command parameter:

- **Set Language**—Click the ellipsis button […] then select the language from the Language Selector dialog box.

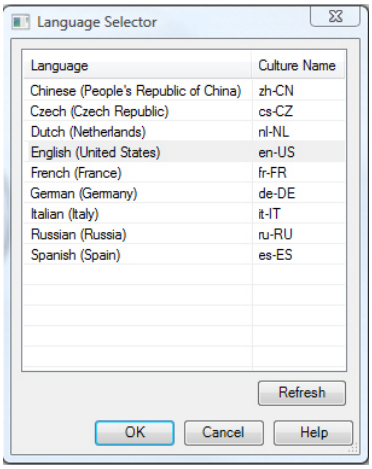


Figure 3.9 – Language Selector Dialog Box

3.15. Set Local Aliases

The **Set Local Aliases** command sets the value of one or more local aliases for the This Display object or for a specific object in the display. The local aliases are used for connecting to data in runtime mode. Local aliases are stored with the display in memory. "Aliasing and Pick Actions" describes the use of local aliases (including object-specific local aliases) with pick actions. In a Silverlight display, this command cannot set local aliases from an external .CSV or .TXT file.

Command parameters for Set Local Aliases are:

- **LocalAliasCommandType**—Specify how the alias value is to be set. The default is **None** (no alias values will be set). You can choose **From Data** (the local alias is entered directly into the LocalAliasCommandData property) or **From File** (the local alias is stored in a CSV or TSV delimited file). Selecting one of these options enables the **LocalAliasCommandData** parameter.
- **LocalAliasCommandData**—Choose the data that will set the alias value or the file that contains alias values.
  - If the **type** is From Data, type the local alias in the format **{{<<*localAliasName*>>}}** or click the ellipsis button to open the Data Browser, where you can go to the Local Alias Table tab to select or create local aliases and then assign their values.

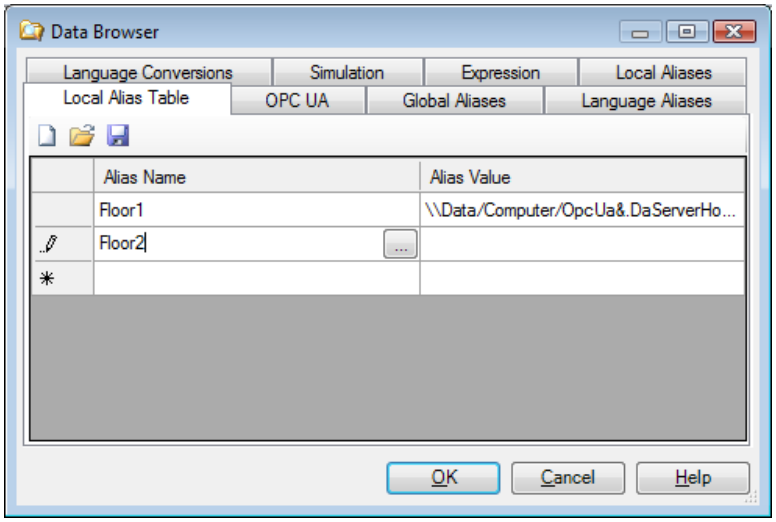


Figure 3.10 – Local Alias Table Tab

- If the **type** is From File, select **Browse Files** to select a delimited file or **Data Browser** to use CSV or TSV delimited files, text values, or other sources for local aliases. In a Silverlight display, this option is not supported.

- **TargetType** is the type of window to be target of the value changes. Some of these selections are window types, some are windows hierarchy (relative paths), and some are absolute settings (such as the Root window). For more information about windows, refer to Windows Dynamics. Choices are:
  - **Current Display** (the default) opens the same GraphWorX64 display; this is a refresh of the currently open GraphWorX64 display.
  - **Popup Window (Display Dependent)**
  - **Popup Window (Display Dependent - Modal)**
  - **Popup Window (Window Dependent)**
  - **Popup Window (Window Dependent – Modal)**
  - **Popup Window (Independent)**
  - **Embedded Window** is a window contained within another window.
  - **Parent Window** is the window from which another window was created.
  - **Root Window** is the first window shown in a display, from which all other windows are created.
  - **New Instance (Separate Process)** is a new instance of the same window.
  - **Any Named Window** is any window; specify the window name in the TargetName property.
- **TargetName** is the actual window to be target of the value changes. Click the ellipsis button […] to open the Data Browser so you can identify the target.
- **ObjectName** (optional) is the name of the visual object for which to set the local alias. Using this parameter sets values for local aliases for only the specified object. Type the name of the object as it appears in the Explorer. Object names are case sensitive. Also, be aware that object names can change; if they do, you must return to this dynamic to change its name here.

3.16. Set Object Visibility

The **Set Object Visibility** command applies a hide or show dynamic to an object based on a condition and using the trigger you specify. Command parameters are:

- **VisibilityCommand**—Specify the type of visibility command to be executed. Choose **Hide** to hide the object, **Show** to show the object, or **Toggle** to switch back and forth between hiding and showing the object.
- **ObjectName**—Specify the name of the object to be hidden or shown. Type the name of the object as it appears in the Explorer. Object names are case sensitive. Also, be aware that object names can change; if they do, you must return to this dynamic to change its name here.

3.17. Set View

The **Set View** command sets the view. This command is useful if the display has views defined for it; for more information refer to Creating 2D Views for Runtime Users and Controlling the Runtime Behavior of 3D Views. Command parameters for Set View are listed below.

- **TargetType** is the type of window to be target of the value changes. Some of these selections are window types, some are windows hierarchy (relative paths), and some are absolute settings (such as the Root window). For more information about windows, refer to Windows Dynamics. Choices are:
  - **Current Display** (the default) opens the same GraphWorX64 display; this is a refresh of the currently open GraphWorX64 display.
  - **Popup Window (Display Dependent)**
  - **Popup Window (Display Dependent - Modal)**
  - **Popup Window (Window Dependent)**
  - **Popup Window (Window Dependent – Modal)**
  - **Popup Window (Independent)**
  - **Embedded Window** is a window contained within another window.
  - **Parent Window** is the window from which another window was created.
  - **Root Window** is the first window shown in a display, from which all other windows are created.
  - **New Instance (Separate Process)** is a new instance of the same window.
  - **Any Named Window** is any window; specify the window name in the TargetName property.
- **TargetName** is the actual window to be target of the value changes. Click the ellipsis button to open the Data Browser so you can identify the target.
- **TargetName3D** is the destination name of the 3D viewport, if any. The name can include wild cards. Or click the ellipsis button to open the Data Browser and select the file.
- **ZoomType** specifies how you are identifying the view:
  - **ZoomToNamed** goes to the view specified in the ViewName property.
  - **ZoomToCustom** goes to the view specified in the ViewRectangle property.
  - **ZoomToObject** zooms to the object specified in the ObjectName property.
  - **ZoomByValue** zooms to the object specified in the ObjectName property and multiplies the current magnification by the percentage specified in the ZoomValue property.
- **ViewName** (ZoomToNamed only) is the name of the view as it is defined in the display.

- **ViewRectangle** (ZoomToCustom only) is the 2D coordinates of the rectangle that will display as the view. Coordinates are four numbers separated by commas (,) that delimit the left, top, right, and bottom of the view. Specify the number of pixels from the left (for the left and right) and from the top (for the top and bottom) of the display.
- **ZoomValue** (ZoomByValue only) specifies the percentage by which the view is to zoom to the object.
- **ObjectName** (ZoomToObject and ZoomByValue only) specifies the name of the visual object the button will zoom to. Type the name of the object as it appears in the Explorer. Note that object names are case sensitive. Also, be aware that object names can change; if they do, you must return to this dynamic to change its name here.
- **TransitionDuration** specifies, in milliseconds, how long you want it to take for the view to transition. The default, 0 (zero), provides no transition; the view changes abruptly. For the transition to take 2 seconds, enter 2000.

3.18. Start Application

The **Start Application** command launches a file or URL and opens the 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.

In a Silverlight display, this command's capability is limited; it can only open documents that can run in an Internet Explorer (IE) window. Such files may include an html page, text file, or office document.

This relatively simple command action requires that you specify as command parameters the **FileName** of the executable or document, as well as any command line **Arguments** to be passed when the application starts.

Tip

Tip: You could pass the command processor CMD.exe to the Start Application command, and by doing so provide a batch file as the input, or use a command as an argument. This opens up a powerful method for controlling devices, setting environmental variables, and performing any system or management function (WMI for example) that you can from a command line.

3.19. Toggle Value

The **Toggle Value** command toggles between two or more specified values or states during runtime using an action that you specify. Make sure you specify a **DataSource** property (the data source the toggled value is written to) and specify **True** for the **Data Write Confirm** property. Command parameters for the Toggle Value command are:

- **ValueOne**—The first value of the first state of the toggle.
- **ValueTwo**—The second value of the second state of the toggle.
- **MoreValues**—This property lets you define a list of values through which the toggle action can cycle. Click the ellipsis […] button to open the Edit Toggle Values dialog box where you can enter additional values. The first two lines are the properties in ValueOne and ValueTwo. The toggle button toggles sequentially through this list, from top to bottom.

3.20. Update Database

The **Update Database** command queries the database attached to the OPC-UA data source with the current values of tags that you specify based on the runtime condition or action that triggers the pick action. It updates the data mining server by writing any value to a data mining tag exposed by this server. This command is not supported in Silverlight displays. There are no command parameters for this command.

3.21. Write Value

The **Write Value** command writes a value (one that you specify or a value of a tag) to another tag. Make sure you specify a **DataSource** property (the data source the value is written to) and specify **True** for the **Data Write Confirm** property. Also, make sure the **Update** property is set to **True**. Command parameters for the Write Value command are listed below.

To enable a Command Parameter property, set its corresponding property in the **Execution Trigger - When** section to **True**. Click the property's ellipsis […] button to open the Data Browser to select a tag or file.

- **OnDownValue**—Specify the value(s) to be written when the mouse is down-clicked (MouseDown event). Normal button only. Available only if ExecuteOnDown is True.

- **WhileDownValue**—Specify the value(s) to be written while the mouse is down. Normal button only. Available only if ExecuteWhileDown is True.
- **OnUpValue**—Specify the value(s) to be written when mouse button is released (MouseUp). Normal button only. Available only if ExecuteOnUp is True.
- **OnDoubleClickValue**—Specify the value(s) to be written when the button is double-clicked. Available only if ExecuteOnDoubleClick is True.

3.21.1. 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.

Buttons are shapes that can have actions associated with click events, mouse overs, and other actions. GraphWorX64 provides four predefined button types, plus the Pick Action, which is a dynamic that you can apply to any object to turn that object into a button. A pick action identifies the command that a button performs; when you create a button, you select from the same commands no matter what type of button you are creating. The predefined buttons and pick action dynamic are available from the GraphWorX 2D **Dynamics** ribbon in the **Buttons** section; they are:

-

Normal Button. A text button is an animated labeled rectangle that performs an action when the runtime user interacts with it.
- **Checkbox**. A checkbox button belongs to a group that allows the runtime user to select any number of boxes in the group to set a program state. A checkbox button can stand alone or it can be grouped in with other checkbox buttons.
- **Radio Button**. A radio button belongs to a group that allows a runtime user to select one member of the group of buttons to specify a single program state or value. It is assumed that a radio button will be part of a collection of radio buttons that are grouped. A radio button is a state selector that is boolean; it is either **on** or **off**. When you group two or more radio buttons into a group object only one of the radio buttons in the group may be selected (on) at any given time.
- **Display Button**. A display button shows a thumbnail of a GraphWorX64 display or template and is used to perform the action Load Display to open the display that is illustrated on the button.
-

Pick Action. A pick action is a dynamic that you use to create a button of your own choosing if none of the buttons listed above meet your needs. (Note that each of the buttons described above has a Pick action already assigned to it.)

You apply a pick action to an object; the pick action dynamic identifies the trigger event that will occur when a runtime user interacts with the object. (The pick action dynamic is not described in this topic. Its range is so expansive that it has its own topic; refer to Pick Actions for more information.)

3.21.2. 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.

3.22. 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 3.11, the Object Count dialog counts the objects, as well as giving you other more information about your display.

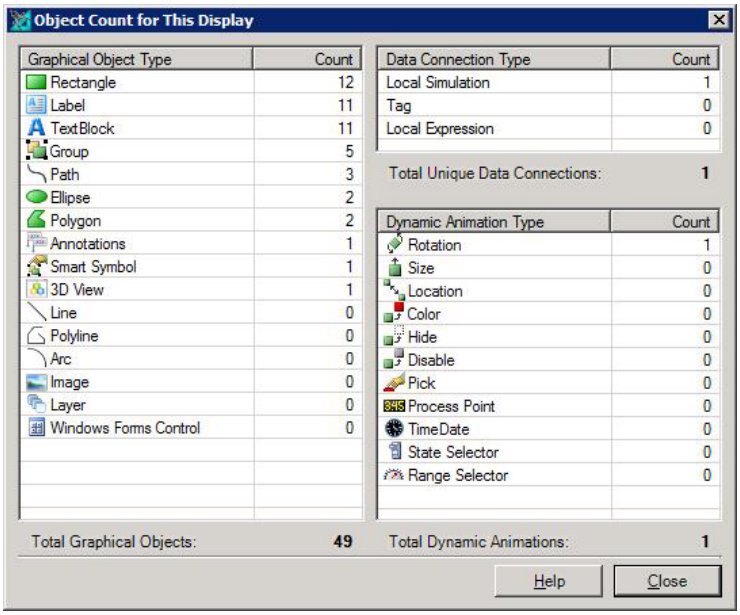


Figure 3.9 - Object Count Dialog

3.23. Layers

When you create a new GraphWorX64 display, any object that you draw or add to the work surface is placed on a primary layer of the display. You can think of that primary layer as providing the same capability as a background would in a paint program; the layer always exists — but is not shown in the object hierarchy of objects, groups, and layers that appear in the Explorer pane. As you add objects to the primary layer, those objects are drawn in an order from front to back such that the last drawn object is the top object. This allows you to work on any newly added object without interference from objects on the design surface from your previous work.

The following figure show four different objects and illustrates the concept of order in a layer. You can control the sequence of objects using the arrange commands for front and back. Note that the Layer object also shows up in the Explorer; if you only intend to use one layer, then there is no need to add a layer. Add layers only when you need more than one. (If you don't see the Explorer pane, bring it back into view by enabling the Explorer check box found in the Show/Hide tab group of the ribbon for the View menu.)

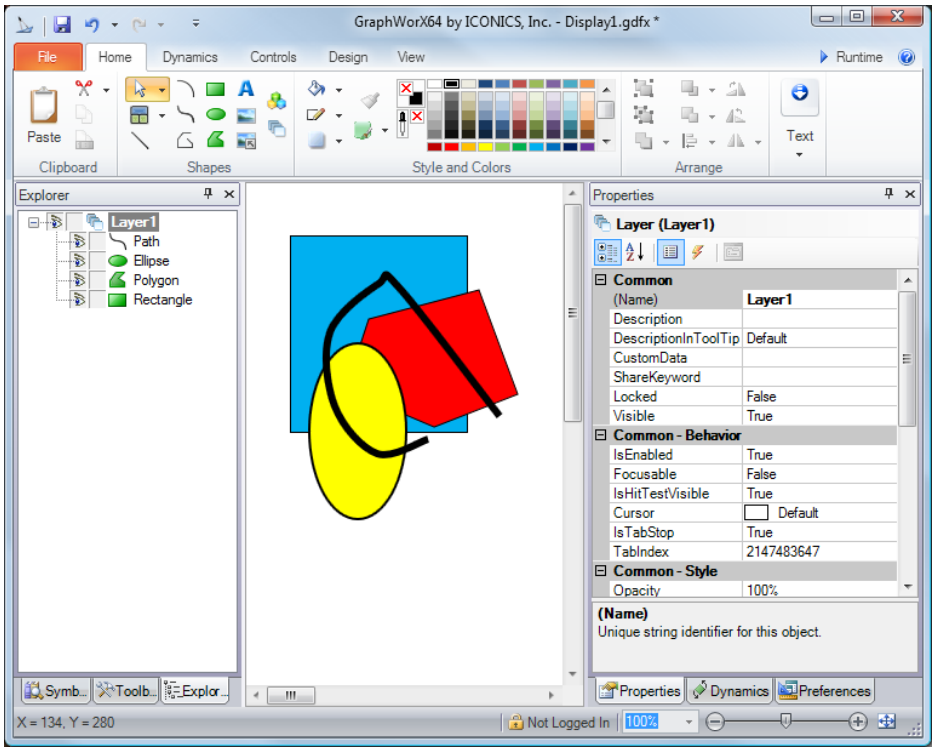


Figure 3.12 - Object Order is Shown in the Explorer Pane

Note that the layer has its own properties displayed in the Properties tab on the right. The Layer's properties are different than the properties of the This Display object because a display can have multiple layers. If you intend to use layers, you can set default values for layer properties on the Preferences tab. For more information, refer to New Layer Default Settings.

You can do the following things with layers:

- Layers can be **renamed** in the Explorer. However, names must be unique.
- Layers can be **selected** and multi-selected. When grey highlighted, they are active, and when blue highlighted, they are selected.
- Layers can be **hidden.**
- Layers can be **locked**.
- Objects and groups can be **drawn** on a layer in the work area
- Objects can be **moved into a layer** by dragging the object in the Explorer.
- If you use objects' ShareKeyword property to be able to maintain uniformity among identical objects in your GraphWorX64 displays, you can use the Update Shared Objects feature to update components in all layers in a display.

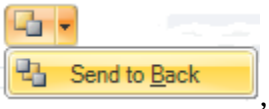
To **Show/Hide** an object:

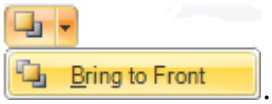
- Enable or disable the **Show object** check box (the eyeball) for the object in the Explorer pane

To **Lock/Unlock** an object:

- Enable or disable the **Lock object** check box (lock) for the object in the Explorer pane. A locked object will not display selection handles when you click on it.

To **Move an Object Order in a Layer** (the Z-order), do one of the following:

- Click the icon in the Arrange tab group of the ribbon for Send Object Backwards , Bring Object Forward , Send to Back 

, or Bring to Front 

.
- Cut an object and paste that object to move it from any position in the Z-order to the front.

- Click and drag the item in the Explorer pane up or down slowly; when the insertion bracket appears in the correct location release the mouse button. The figure below shows this operation.

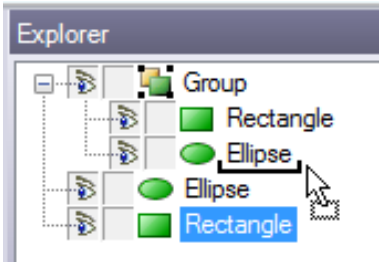


Figure 3.13 - Changing the Order of an Object in Explorer

Tip

Tip. Drag an object slowly as there is a slight delay in placement in order to allow GraphWorX64 to position the object correctly. If you want to move an object into a group, **pause under the group name** and wait for the group to open before you position your moving object in the group at the location desired.

Layers are useful when you want to organize collections of objects, but still return the ability to access each object in the layer individually. In a display that shows a collection of gauges you could use layers to hide a gauge when the condition is appropriate. Layers can also be used as an organizational unit when you have a complex display with a large number of objects.

To Add a Layer to a Display:

- Click on the **Home** menu, then click on the **Layer** button in the Shapes tab group on the ribbon.
- The new layer is added to the top of your drawing, no matter what the currently selected object is and numbered sequentially Layer n, where n=1, 2, 3 and so forth.

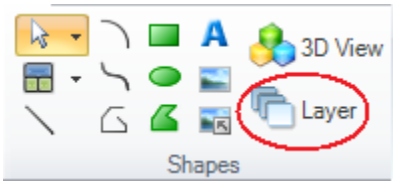


Figure 3.14 - The Add Layer Button

To Move a Layer Up or Down in the Z-order:

- Click and **drag** that layer object in the object Explorer to the new position, then release the mouse button.

To Create a Sublayer:

- Select the layer that will contain the sublayer, or an object contained inside that layer, then click the Add Layers button
- Or, drag a layer in the Explorer to the position in the hierarchy that you want the sublayer to occupy.

3.24. 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 Switch Between Application Modes:

- Click the Application Mode drop down menu in the Misc section of the View ribbon and select the mode you desire.

Figure 3.15 - Misc Section of the View Ribbon

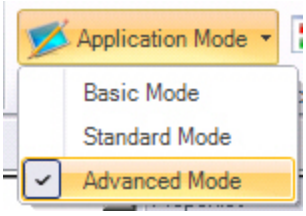


Figure 3.16 – Choose a Configuration Mode

To gain an appreciation of the differences between the three modes, consider Figure 3.15, 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.

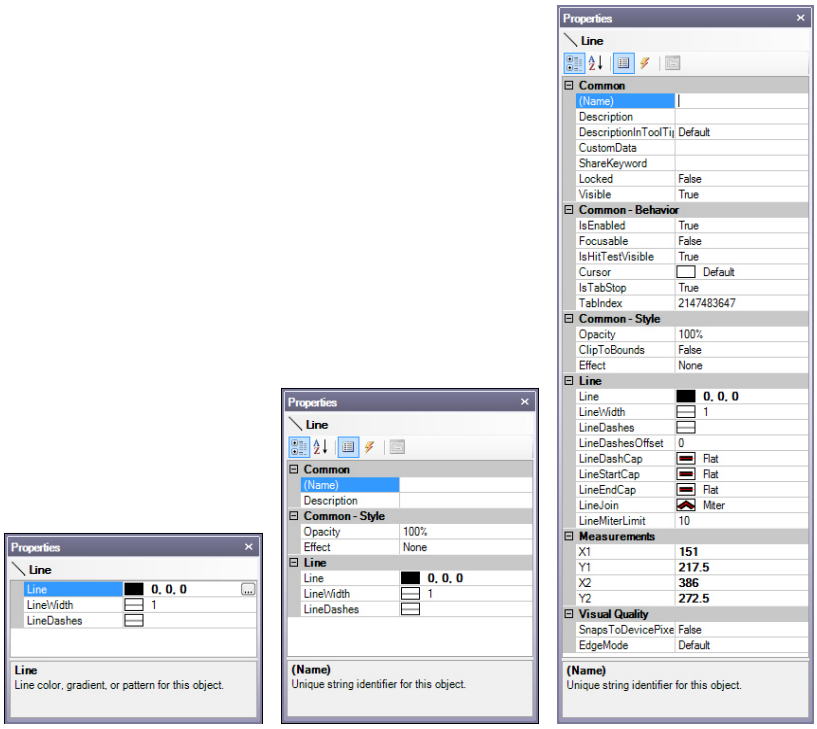


Figure 3.17 – Basic (Left), Standard (Middle), and Advanced (Right) Modes

3.25. 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.

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, as is shown in the following figure.

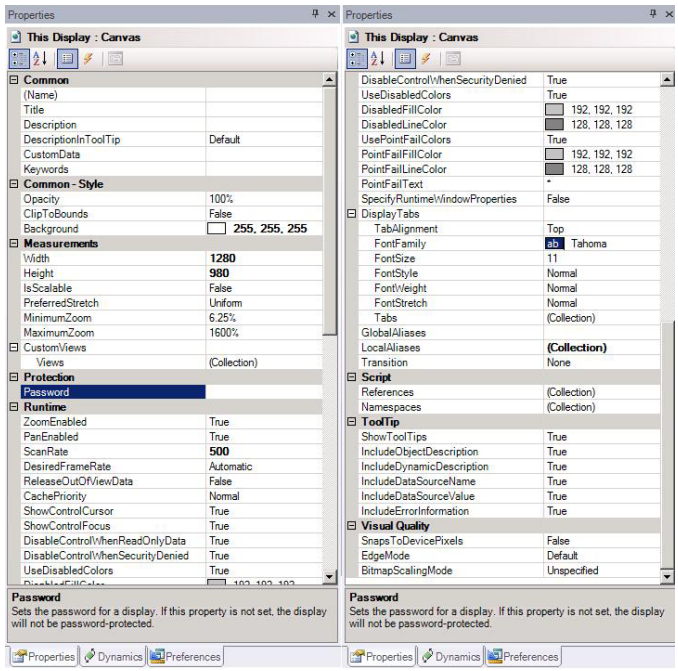


Figure 3.18 – Protection Property - Password

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.

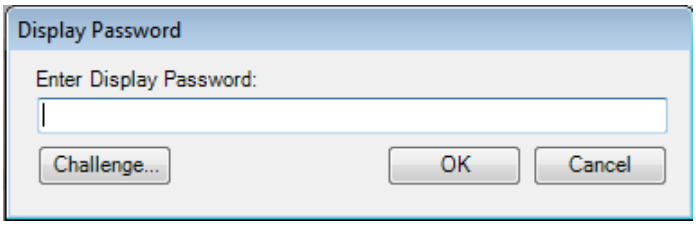


Figure 3.19 – Display Password

3.26. New Settings

In the File menu for GraphWorX64, you have the option of selecting the **New Settings** button, as shown below.

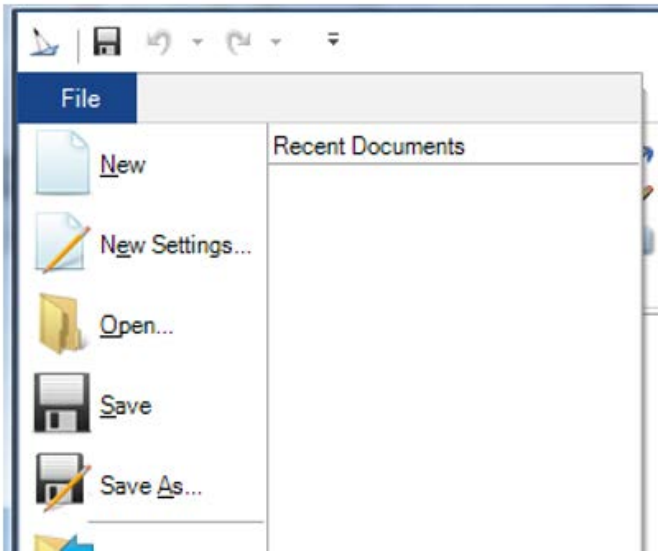


Figure 3.20 - New Settings Button in GraphWorX64's File Menu

This opens the New Display window, as shown below.

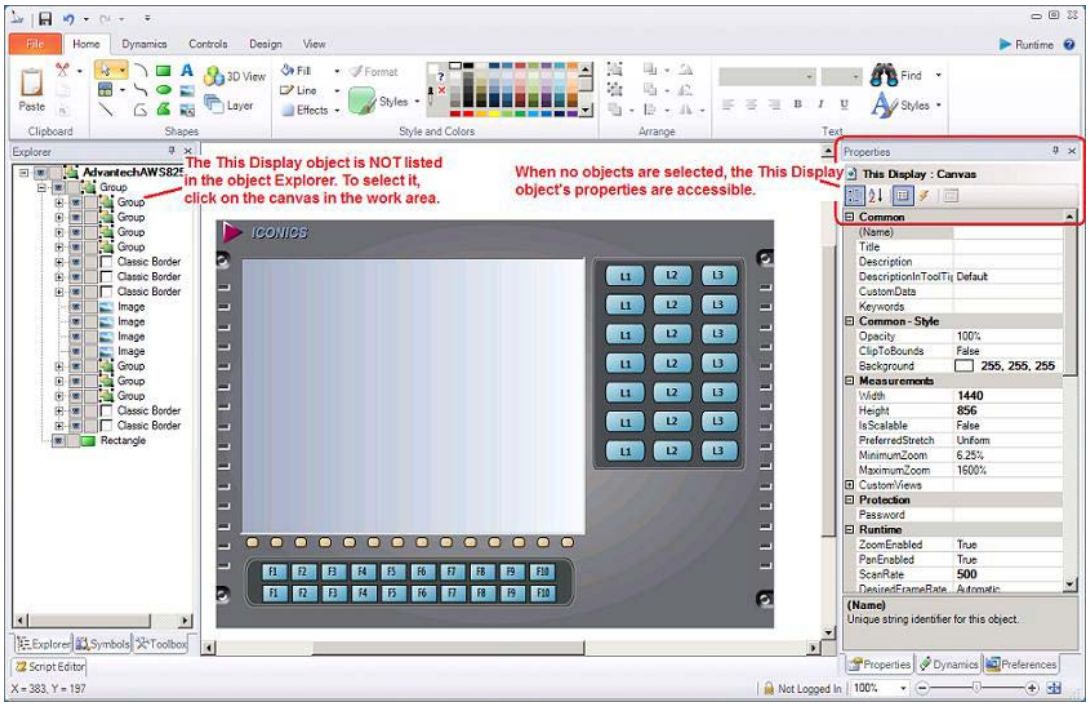


Figure 3.21 - New Display Window

The New Display window allows you to change the properties of the current GraphWorX64 display, including:

Display Type

The **File Extension** will change according to the Platform and Attributes you set.

- **Platform** - You can select from either **Desktop** (.gdfx), Compressed (**WebHMI)** (.gdfxz) or **MobileHMI** (.gdfxp) abd Compressed Phone Display (*.gdfxpz)
- **Attributes** - Depending on the Platform, you'll be able to select whether the display you are creating will be **Compressed** or a **Template**. Note that these selections also affect the File Extension.

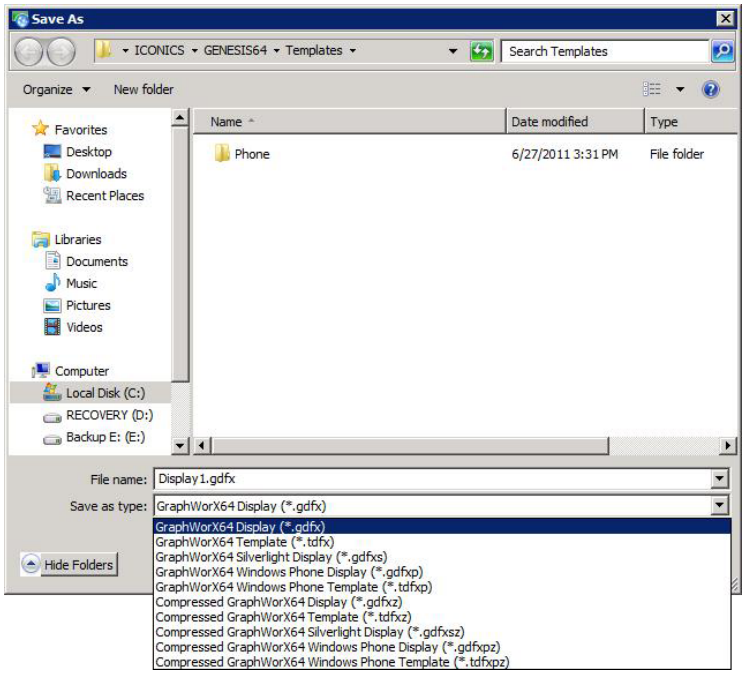


Figure 3.22 – Setting as Compressed or Template

The Display Type settings will then affect the Display Properties that are shown within the New Display window.

3.26.1. Grid Settings

The following is a list of all 2D grid settings properties that are available in Advanced application mode. All of these settings become the default grid properties for the GraphWorX64 work surface and can be overridden while working in displays. Setting these preferences is optional.

Grid Settings section:

- **Visible** - **True** makes the 2D grid visible; **False** hides it.
- **Snap** - To make it easier to align objects, select **True** to make the objects you add and move in a display snap to the grid so that objects align to the nearest grid point. To have more control and allow objects to move more freely, select **False**.
- **Width** - The number of horizontal pixels between grid points.
- **Height** - The number of vertical pixels between grid points.
- **Color** - Specify the default color for grid points.
- **AutoAdjustGridSize** - **True** adjusts the grid when you zoom in and out so that it is always relative to the objects in the display. **False** maintains the same grid marks regardless of the zoom level. This option cannot be overridden in a display.

Ruler Settings section:

- **HorizontalRulerVisible** - **False** (the default) hides the horizontal (X axis) ruler in 2D; **True** shows it. This option is also available on the 2D View ribbon.
- **HorizontalRulerMultiplier** - Adjust the scale of the tick marks that appear along the horizontal ruler.
- **VerticalRulerVisible** - **False** (the default) hides the vertical (Y axis) ruler in 2D; **True** shows it. This option is also available on the 2D View ribbon.
- **VerticalRulerMultiplier** - Adjust the scale of the tick marks that appear along the vertical ruler.

3.26.2. New This Display Default Settings

You can enter preferences for all ThisDisplay properties. The following is a list of all ThisDisplay property preferences that are available in Advanced application mode. All of these settings become the default for displays' properties and can be overridden for individual displays. Setting these preferences is optional. For more about This Display properties, refer to Displays.

Common section:

- **Name** - Unique string identifier for each individual object.
- **Title** - Title of document.
- **Description** - Description of object. (Oftentimes appears as a tooltip for the object).
- **DescriptionInToolTip** - Determines whether or not the object's description is included in the object's tooltip.
- **CustomData** - A custom data string.
- **Keywords** - Particular keywords that can be used when searching for the particular file.

Common - Style section:

- **Opacity** - Overall opacity percentage of object. (Range: 0% is fully transparent; 100% is fully opaque). [NOTE: You can't assign a tag directly to the Opacity property. However, you can use the Hide dynamic instead. When the Hide dynamic is set to "Analog" mode and is used with an analog tag, it will change the opacity of the object.]
- **ClipToBounds** - Specify True to clip the content of this object to the bounds of the object.
- **Background** - Background color, gradient, or pattern of object.

Measurements section:

- **Width** - Width of the object.
- **Height** - Height of the object.
- **IsScalable** - Select True to make the display scalable when loaded. Select False for a fixed-scale display.
- **PreferredStretch** - Stretch mode is for scalable displays and "Fit to Windows" operations. Options consist of: uniform (which preserves the display's aspect ratio while keeping the entire previous view visible), fill (which stretches the display to fit inside the new window), and UniformToFill (which preserves the display's aspect ratio and causes the display to fill the window).

- **MinimumZoom** - Set the smallest allowed zoom percentage.
- **MaximumZoom** - Set the largest allowed zoom percentage.
- **CustomViews** - Custom views that are used to change the current zoom and pan when selected by the user.
- Views – Edit the list of custom views by selecting the ellipse and selecting a view from the custom views window.

Phone section: These options are used only for displays saved as Windows Phone displays.

- **OrientationLock** - **None** is the default value. Choose **Landscape** or **Portrait** to lock the display in either a landscape or portrait orientation on a phone.
- **SwapDimensionsOnOrientation** - Select True to swap the display's Width and Height properties when the display's orientation is changed.

Protection section:

- **Password** - Use this preference to password-protect the display.

Runtime section:

- **ZoomEnabled** - **True** (the default) allows users to zoom in displays; **False** prevents users from zooming.
- **PanEnabled** - **True** (the default) allows users to pan in displays; **False** prevents users from panning.
- **ScanRate** - The rate in milliseconds in which displays request data values by default.
- **DesiredFrameRate** - Specifies the ideal frame rate for animations with a Duration (time-based). Select zero for the smoothest animation.
- **ReleaseOutOfViewData** - Runtime data is automatically released for objects that are not within the currently visible display area (due to panning/zooming).
- **CachePriority** - Displays are automatically removed from the cache if the cache is full. Specify your preference for handling caching when memory gets low during runtime. This property is relevant only if the **CacheDisplays** preference is set to **True** (set for the **Runtime Options** preference, above). For a detailed description of how to set up caching, refer to "Configuring the Caching of Displays". Options are:
  - **NotCached** – Displays with this setting are not added to the cache, regardless of the **CacheDisplays** preference setting. Select this option if you want to exclude most displays from being cached; then override this property for individual displays.

  - **Low** – Displays with this setting are most likely to be removed from the cache when memory is low.
  - **Below Normal** – Below the default value.
  - **Normal** – The default value.
  - **Above Normal** – Above the default value.
  - **High** – Displays with this setting are least likely to be removed from the cache when memory is low.
  - **NotRemovable** – Displays with this setting are not removed from the cache, even when memory is low.
- **ShowControlCursor** - Specify True to show an alternate cursor when hovering over a controllable dynamic object.
- **ShowControlFocus** - Specify True to show a highlighted indicator around selected controllable objects.
- **DisableControlWhenReadOnlyData** - Disables controllable dynamics that reference read-only DataSources.
- **DisableControlWhenSecurityDenied** - Disables controllable dynamics when security permissions are denied.
- **UseDisabledColors** - Specify True to use alternate colors to indicate that an object is disabled.
- **DisabledFillColor** - Displays alternate fill/background color to indicate that an object is disabled.
- **DisabledLineColor** - Displays alternate line/text color to indicate that an object is disabled.
- **UsePointFailColors** - Specify True to use alternate colors to indicate point failure.
- **PointFailFillColor** - Displays alternate fill/background color to indicate point failure.
- **PointFailLineColor** - Displays alternate line/text color to indicate point failure.
- **UsePointFailText** - Specify True to show the PointFailText to indicate that an object has a bad quality data source. When False, the last known good/uncertain value will continue to be shown (instead of the PointFailText) when there is a bad quality data source.
- **PointFailText** - The text displayed for process points with data errors. Specify a single character (repeated for length of PPT) or a multi-character string (not repeated).
- **SpecifyRuntimeWindowProperties** - Select **True** to specify non-default window properties for runtime mode.
- **RuntimeWindowProperties** - Open window runtime properties.
  - **SpecifyWindowLocation** - Set to **True** to specify the location of the window. When **False**, properties related to the window location will not be applied.

- **SpecifyWindowSize** - Set to **True** to specify the size of the window. When **False**, properties related to the window size will not be applied.
- **SpecifyWindowStyle** - Set to **True** to specify the style of elements within the window. When **False**, properties related to the window style will not be applied.
- **SpecifyCommandingOptions** - Set to **True** to specify the commanding options of the window. When **False**, properties related to the window commanding options will not be applied.
- **StartLocation** - Set the starting position of the window at runtime.
- **Left** - Specify the x-coordinate of the window's left edge in pixels or percentage value. Use "Unspecified" to skip setting the window's coordinate.
- **Top** - Specify the y-coordinate of the window's top edge in pixels or a percentage value. Use "Unspecified" to skip setting the window's coordinate.
- **Width** - Set the width of the window in pixels or a percentage value. Use "Unspecified" to skip setting the window's width. Use "Automatic" to set the window's width around the display size.
- **Height** - Set the height of the window in pixels or a percentage value. Use "Unspecified" to skip setting the window's height. Use "Automatic" to set the window's height around the display size.
- **WindowState** - Specify the window's state (Normal, Minimized, Maximized).
- **TitleBarVisible** - Set the visibility of the window's title bar.
- **Text** - Specify the text that appears in the title bar of the window.
- **ControlBox** - Set whether a control box is displayed in the caption bar of the window.
- **MinimizeBox** - Indicate whether a minimize button is displayed in the caption bar of the window.
- **MaximizeBox** - Indicate whether a maximize button is displayed in the caption bar of the window.
- **OuterBorder** - Set to true to give the window an outer border. Set to false for no outer border.
- **Resizable** - Set to true to allow the window to be resizable.
- **ToolWindow** - Set to true to specify the tool-window style for the window (small title bar).
- **InnerBorder** - Set to true to include an inner border on the window.
- **InnerBorderBrush** - Specify the color of the window's inner border.
- **InnerBorderThickness** - Set the thickness around all four sides of the window's inner border.

- **HorizontalScrollVisible** - Indicate whether the horizontal scroll bar is always visible (Visible), visible only when content exceeds the width of the window (Auto), invisible while maintaining the ability to scroll the window left and right (Hidden), or invisible while preventing the ability to scroll the window horizontally at all (Disabled).
- **VerticalScrollVisible** - Indicate whether the vertical scroll bar is always visible (Visible) visible only when content exceeds the height of the window (Auto), invisible while maintaining the ability to scroll the window up and down (Hidden), or invisible while preventing the ability to scroll the window vertically at all (Disabled).
- **SpecifyScrollBarColors** - Set to true to specify the colors of the scroll bar for the window. When set to false, the scroll bar colors will be determined by the operating system or platform.
- **RibbonVisible** - Set the visibility of the window's ribbon control.
- **MenuVisible** - Set the visibility of the window's main menu.
- **NavigationBarVisible** - Set the visibility of the window's navigation bar.
- **StatusBarVisible** - Set the visibility of the window's status bar.
- **ShowContextMenu** - Determine whether to show the runtime context menu after right-clicking within the window.
- **TopMost** - Set to true to have the window always appear above all other windows, even when it becomes inactive.
- **CommandingEnabled** - Determine whether commands are enabled for this display. Use the CommandingName setting to specify a registration name.
- **DragAndDropCommandingEnabled** - Set to true to enable receiving commands by drag and drop.
- **CommandingName** - Set the name used for registering commands within this display.
- **CommandingDisplayName** - Set the user friendly control name.
- **MaxCommandScope** - Indicate whether commands sent and received by this display affect this display only (Process) or all displays that are currently opened (Machine).
- **OverridePreviousDisplay** - Set to true to override the window properties of a previously loaded display. Set to false to retain the window properties that were set previously.

- **DisplayTabs** - Tabs used to load display files.
  - **TabAlignment** - Specify which edge of the window the tab strip is docked.
  - **FontFamily** - Specify the name of the font used to render the text.
  - **FontSize** - Specify the font size used to render the text.
  - **FontStyle** - Specify the font style used to render the text (italic).
  - **FontWeight** - Specify the font weight used to render the text (bold).

- **FontStretch** - Specify whether the tabs' rendered text is narrower than usual (Condensed), wider than usual (Expanded), or the same width as it usually is (Normal).
- **Tabs** - Select the ellipsis […] button to open the Display Tabs dialog, which allows you to edit the text on the display tabs as well as the display that opens when each tab is selected. These tabs appear when entering Runtime Mode and allow users to navigate among multiple displays inside of a full window. Each individual corresponding display can be accessed by selecting the tab. The default location of the tabs is at the top of the full window.

- **LocalAliases** - Local aliases used by most or all displays. For more information, refer to “Local Aliases”.
- **GlobalAliases** - Global aliases used by most or all displays. For more information, refer to Global Aliasing.
- **ResolveAliasesInDragDropDataSources** - Set to true to resolve all local and global aliases within a data source contained in a component as the component is dragged onto another component in a drag-and-drop operation. Set to false to delay resolving these aliases until after the component has been moved and dropped onto another component.
- **Transition** - The form of animation to be used by default when a runtime user switches from the current display to another one. For more about transitions and overriding this value for a display, refer to the Transition section of Design Ribbon in GraphWorX64 2D.
- **SafeTouchZoneWarningDuration** - This displays the amount of time the safe-touch warning will remain active before automatically resetting to inactive.

Runtime Compatibility section:

- **DisableControlWhenAnyRelatedDynamicIsDisabled** - Specify true so that if any dynamic on a given object is disabled, all other dynamics on that object will also be disabled. Specify false so that each dynamic on a given object can be disabled. The default value is false.
- **HideDynamicActiveWhenPointFailure** - When true, hide dynamics that have a bad quality data source will be set to the "active" state (i.e. hidden). When false, hide dynamics that have a bad quality data source will be drawn with the "PointFailColors" (default behavior).
- **StateFieldDropDownUsesSystemsColors** - When set to true, the statefield drop-down UI will use the current "System Colors" (defined by the operating system). When set to false, the statefield drop-down UI will use colors that match the Label associated with the containing ProcessPoint dynamic (default).

Script section:

- **References** - Assembly references are used by most or all displays for scripting.
- **Namespaces** - The namespaces are used by most or all displays for scripting.

ToolTip section:

- **ShowToolTips** - Displays available tooltips.
- **IncludeObjectDescription** - Provides a particular object's description in the tooltip.
- **IncludeDynamicDescription** - Provides descriptions for associated dynamic animations.
- **IncludeDataSourceName** - Provides datasource names used in associated dynamics.
- **IncludeDataSourceValue** - Provides datasource values used in associated dynamics.
- **IncludeErrorInformation** - Shows error information (i.e. the reason for the data failure).

Visual Quality section:

- **SnapsToDevicePixels** - Determines whether the rendering for this object should use device-specific pixel settings during rendering.
- **EdgeMode** - Determines the way the edges of objects are rendered.
- **BitmapScalingMode** - Determines the quality of scaling for bitmap rendering.

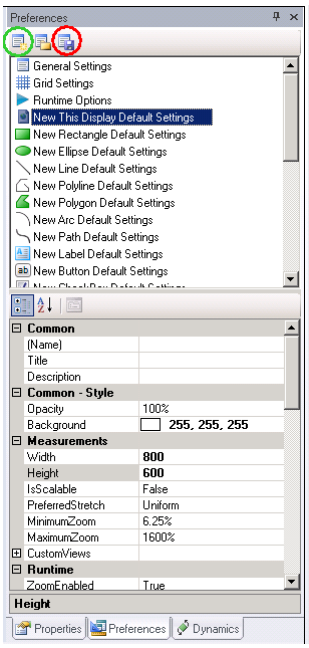


Figure 3.23 - Changing the Default New Root Object Type in GraphWorX64's Preferences (General Settings)

Lab: icTonics Monitor Display L3A

3A.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.

Your completed display should resemble the figure below.

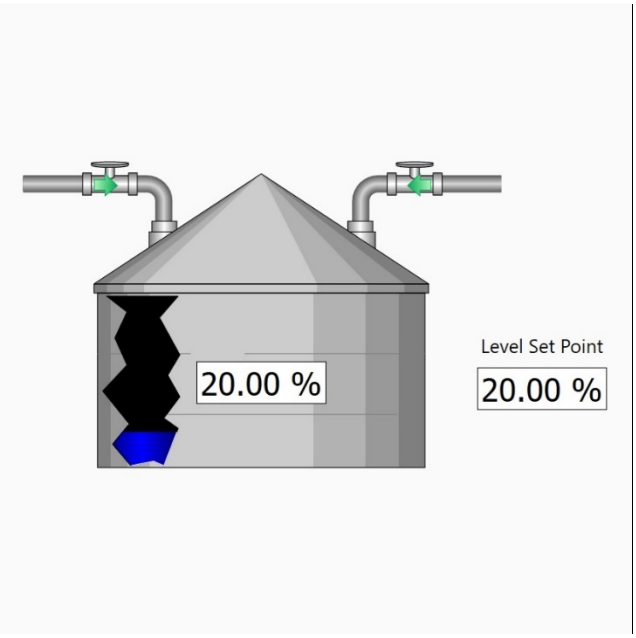


Figure L3A.1 – icTonics Tanks Monitor Display

Estimated time to complete this lab: **40 minutes**

3A.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.

3A.1.2. Support Files

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

- icTonicsHome.gdfx
- icTonicsMain.gdfx
- GENESIS64TrainingSymbols.sdfx

3A.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 GraphWorX64. We will build a display specific to monitoring the tank, and then incorporate it into a dashboard for access later.

2. Set the Background property of the canvas by selecting **Background** on the property grid at right, use the ellipsis and then select the Global Color tab. Finally, select "Background2" and select OK.

3. From the left-side navigation area, click on the Symbols tab in the bottom. Next, click on the Symbol Library commands menu button and then on "Open Existing Category. You can then navigate to the GENESIS64TrainingSymbols Symbol library file (.sdfx).

4. From the GENESIS64TrainingSymbols Symbol library, insert a "**BaseTank**" and two "**Pipes**". Also, from the **_General →Tanks → 2D_Tanks** library, insert the "**JaggedCutAway 2**". Arrange the objects to look like the following image.

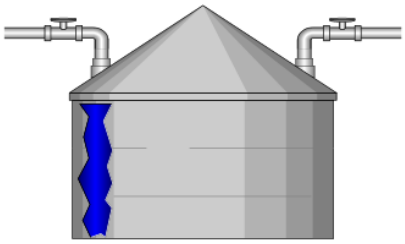


Figure L3A.2 - Mixer Tank

5. 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 the "**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.

6. 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.

7. In the Data Browser, select the Data Points tab if it is not already selected. Then go to **My Computer → Data Connectivity → Modbus → icTonics → Tanks → USA → Foxboro → Tank1 → TankLevel**. Click on "OK" to choose the TankLevel tag and close the Data Browser.

8. 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, or press the "Esc" key to de-select the item.

9. 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, as shown in the first figure. Link the DataSource to the same Tank Level tag as you selected previously.

10. 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.

11. Select everything that you have created thus far, then click on the **Group Objects** button in the Home ribbon's Arrange section. You can also press Ctrl+G to accomplish the same thing.

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.

12. In the Symbols library, browse to GENESIS64TrainingSymbols and drag the "**RightArrow**" into the display.

13. Resize the arrow so that it fits inside the pipe on the left. Hold Shift while dragging the corner control elements to resize the arrow uniformly.

14. Select the Dynamics tab. Note that there are two dynamics (hide and location) already associated with the arrow. We will edit these dynamics in the following steps.

15. In the Dynamics toolbar, select the Hide dynamic.

16. Click on the ellipsis [...] on the right side of the DataSource field and use the Data Browser to define the tag below as the data source:

modbus:icTonics.Tanks.USA.Foxboro.Tank1.SyrupValveStatus

17. 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: *modbus:icTonics.Tanks.USA.Foxboro.Tank1.SyrupValveStatus*

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

19. 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. You can hold Shift while placing the waypoints in order to snap the line to consistent angle intervals. Your path should look similar to Figure L5A..

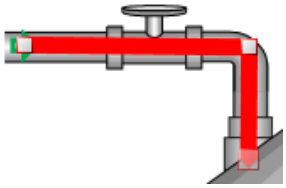


Figure L3A.3 - Location Dynamic

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

21. With only the arrow selected, use Ctrl-D to duplicate the arrow. Place the duplicate **Arrow** 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.

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

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

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

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

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

27. Set the **DataSource** to the **SyrupValveStatus** and change the **MenuText** property value to "**Open**".

28. **Add** a second Menu Item. This second one will be similar to the first, except the **MenuText** should be set to **Close**. Next, change the **OnUpValue** to **0**. Your finished configuration should look similar to the following image.

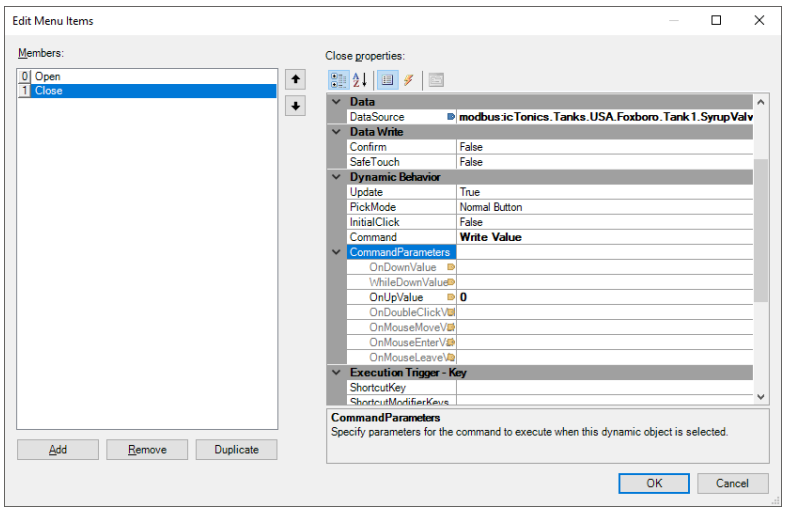


Figure L3A.4 - Popup Menu Configuration

29. Click on **OK** to save the configuration and close the dialog box.

30. Repeat steps 26-33 for the valve on the right-hand side of the tank. (Except, change the **DataSource** to **WaterValveStatus**.)

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

3A.3. Add a Level Set-Point Data Entry Field

1. For the steps in this section, refer to the first figure in this (Figure L3A.1) as a reference.

2. From the **Dynamics** tab in the **Text Objects** section, select the **Data Entry** Text Object. Place the data entry item toward the upper right corner of the display.

3. Set the **DataSource** to:

modbus:icTonics.Tanks.USA.Foxboro.Tank1.LevelSetPoint

4. From the **Home** tab **Shapes** section click on the **Draw Label** icon

5. Place a label above the **Data Entry** field and change the **Text** property (under the **Content** section) to **Level Set Point**.

6. Set the color of the label **Foreground** property to be "Foreground2" either by typing it in directly or finding it under the Global Colors tab found on the color picker.

7. **Save** the display in the C:\Training folder as "**icTonicsTankMonitor.gdfx**".and go into **Runtime**.

8. Enter a value of **55** into the **Level Set Point** data entry field and hit Enter.

9. If the **Tank Level** is **less than** the **Level Set Point**, opening the valves should fill the tank. We will add a way to drain the tank in a subsequent lab.

3A.4. Link the Tank Display into a Dashboard

In order to make this tank display accessible from other graphics, we will want to create a button and link to it from a dashboard for navigation.

1. Navigate to the C:\Training folder and open **icTonicsMain.gdfx**.

2. In the Explorer on the left, navigate to Menu→ ButtonsPanel→ HomeButton

3. Select the **HomeButton** and press Ctrl+D to duplicate it.

4. Name the duplicated item "TankButton", and then select the pick action for it on the Dynamics grid.

5. Change the **FileName** property to be "icTonicsTankMonitor.gdfx", either by typing it in directly or using the file browser from the drop-down at right.

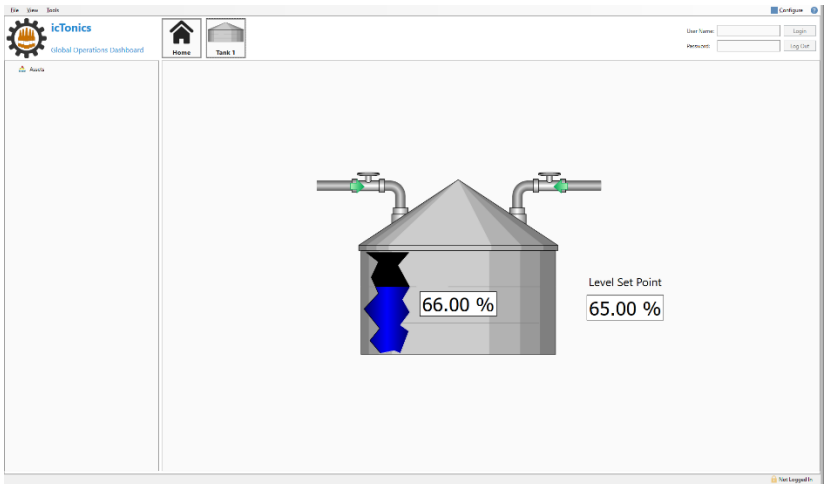
6. Expand **TankButton** in the explorer and find the **Label** object. Select it and in the property grid at right set the text to be "Tank 1".

7. Now expand the **Viewbox** object and navigate to find the Path inside for the symbol. Delete the old symbol, and drag and drop the tank from the GENESIS64TrainingSymbols library in instead.

8. The button should now resemble something like this:

9. Save the display, and put it into Runtime. You can see that by clicking on the Tank 1 button, it loads the display you created into the content area of this dashboard.

10. The end result should now resemble something like this:



This concludes this lab.

Lab: Layers and Dynamics *L3B*

3B.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 to 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**

3B.1.1 Task Checklist

In this lab, you will complete the following tasks:

- Create a Layer.
- Add a Scale Control.
- Add a Draining hide dynamic.

3B.1.2 Support Files

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

- icTonicsHome.gdfx
- icTonicsMain.gdfx
- icTonicsTankMonitor.gdfx

3B.2. Adding a Layer

We will add a 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 16.

1. Open **GraphWorX64**, if you closed it from the previous lab.

2. Open the **icTonicsHome.gdfx** display in a new tab for editing.

   Note: 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 in the **C:\Training** folder and select the file.

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**.

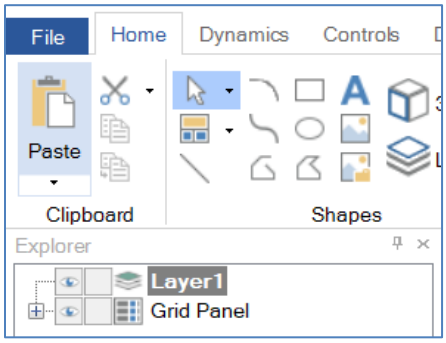


Figure L3B.1 – Explorer view of Layer1

5. Go to the **Dynamics** tab in the ribbon. In the **Buttons** section, click on the **Button** icon and draw a box in the bottom left of the GraphWorX64 Canvas to add a button to the display. You can use the arrow keys to micro-move your button.

6. Change the **Text** of the button to "**Tank Monitor**" by double left-mouse clicking on the button or by editing the **Context →Text** item of the button's Properties. Note that you can type a space before and after the text for readability.

7. Leave the button's **Dynamic Behavior → Command** property as "**Load Display**".

8. In the Dynamics explorer, find the **CommandParameters → 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 a relative path, delete the file path, **C:/Training/** in the **FileName** field so that it only reads "**icTonicsTankMonitor.gdfx**".

11. Confirm that the button is placed in the viewable area in the center part of the display. Your display should now look like Figure L3B.2 below.

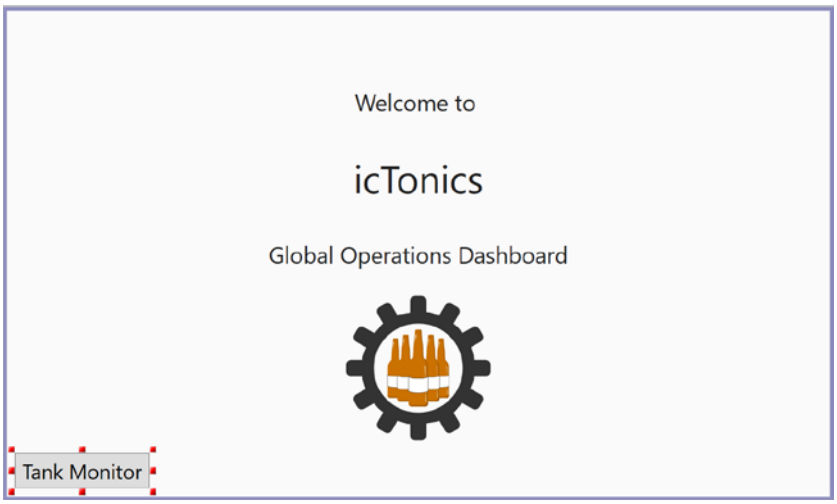


Figure L3B.2 – Completed Homepage Display

12. **Save** the display.

3B.3. Add a Scale Control

1. Open the **icTonicsTankMonitor.gdfx** 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, draw a box for 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 L3B.3.

   NOTE: You may first want to move the Level Set Point text and Process Point to make room for the Scale Control.

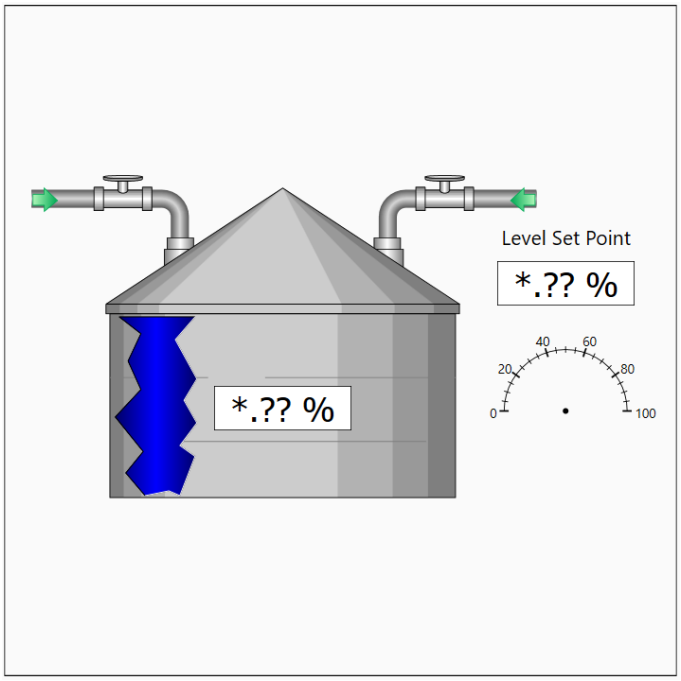


Figure L3B.3 - Scale Control

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.

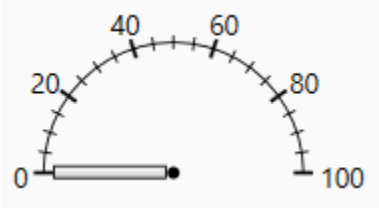


Figure L3B.4 – Gauge with Pointer

4. Add a **Rotation** dynamic to the pointer. Move the red crosshair, the red circle **Pivot Point**, and using the arrow keys center to the Pivot Point of the gauge.

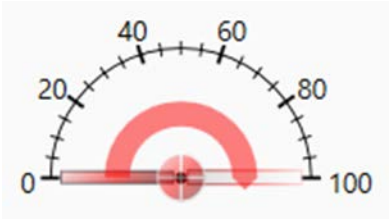


Figure L3B.5 – Rotation Path

5. Now click on the ellipse ... to the right of the **DataSource** field to launch the **Data Browser**. Define the **DataSource** as:

Modbus:icTonics.Tanks.USA.Foxboro.Tank1.TankLevel

6. 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**.

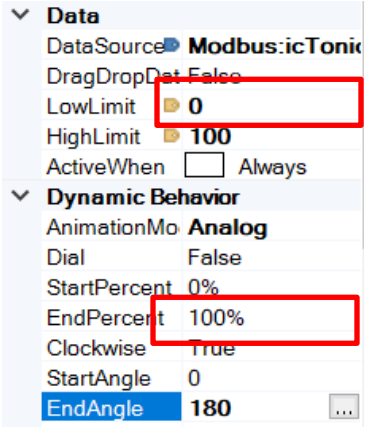


Figure L3B.6 – Rotation Dynamic Settings

NOTE: You can also edit the pointer by selecting it, click on the ***Edit Dynamic On-Screen*** link at the bottom of the **Dynamics** explorer.

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, click **Group Objects** (Ctrl+G).

3B.4. Add a Churning Blade Button

Now we'll add a button to toggle the churning blade on and off.

1. From the **Dynamic** tab in the **Buttons** section, select a **Button**.
2. Place the button above the tank and center it on the two pipes.
3. In the **Content** section of the **Properties** explorer, change the button **Text** to "**Churning Blade On / Off**".
4. Select the **Dynamics** tab and set the **Command** to **Toggle Value.**
5. Under **Command Parameters**, set **ValueOne** = **1** and **ValueTwo** = **0**.
6. In the **Dynamics** explorer, under the **Data** section, set the **DataSource** to:

Modbus:icTonics.Tanks.USA.Foxboro.Tank1.ChurningBladeStatus

3B.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. For the **DataSource** browse and select:

   Modbus:icTonics.Tanks.USA.Foxboro.Tank1.ChurningBladeStatus

3. In the **Measurements** section of the **Properties** explorer, set the **Width** to **320** and the **Height** to **40**.

4. In the **Format** section of the **Dynamic** explorer under **PrefixLabel**, type "Churning blade is". (Make sure you have enabled **Advanced Mode** in the **Application Mode** section of the **View** ribbon in GraphWorX64.)

5. In the **StateField** section set **StateField** to **True**. Click on the ellipsis ... to the right of **States** field. In the "**Edit Process Point States**" dialog box, click **Add**. On the right, under the **Data** section, set the **LowLimit** to **0** and the **StateText** to **Off**.

6. Click **Add** again, then, under the **Data** section, set the **LowLimit** to 1 and the **State Text** to **On**. Click **OK** to exit this window.

7. Your display should now be similar to the following Figure L3B.7.

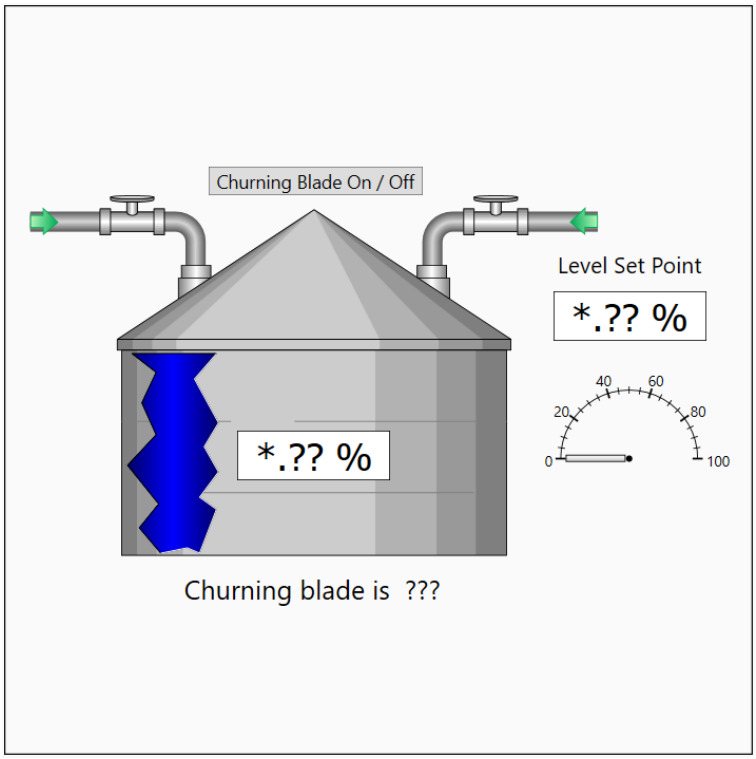


Figure L3B.7 – icTonicsTankMonitor Display

3B.6. Add a Text Object with a "Hide" Dynamic

We will add a label to our tank in this section so that a user knows when the tank is emptying. To facilitate this we will add a Text object with an associated Hide dynamic. We will be adding a Hide dynamic, so this text will only be visible when the tank is draining.

1. From the **Home** ribbon, in the **Shapes** section, select the **Text** (Draw Label) icon. This is represented by a blue capital A.

8. 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.

9. Change the **Text** to **Draining**.

10. Click on the **Dynamics** tab and select the **Hide** dynamic from the top of the **Dynamics** explorer.

11. Set the **DataSource** to:

Modbus:icTonics.Tanks.USA.Foxboro.Tank1.PumpStatus

12. In the **Data** section, set the **HideWhen** field to **data == 0**. The **Dynamics** explorer pane should now look like Figure L3B.8 below.

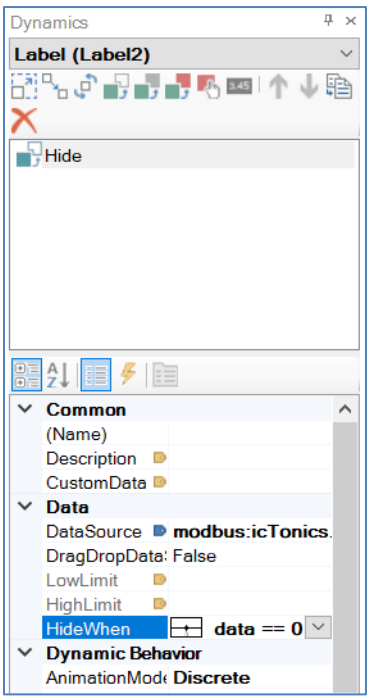


Figure L3B.8 – Dynamics Explorer Settings

13. Your final configuration should resemble the following Figure L3B.9.

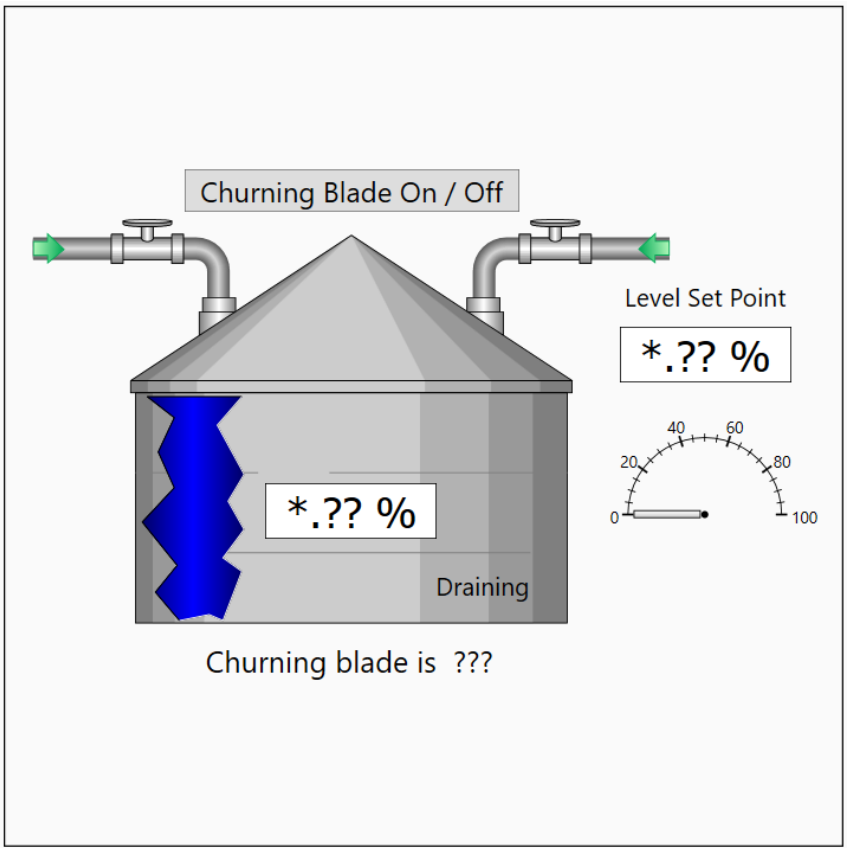


Figure L3B.9 – icTonicsTankMonitor with Level SetPoint, Scale, and Draining Dynamics

This concludes this lab.

Lab: Data Explorer *L3C*

3C.1. Lab Overview

This lab provides users with a familiarity with the Data Explorer control within GraphWorX64.

Estimated time to complete this lab: **10 minutes**

3C.1.1 Task Checklist

In this lab, you will complete the following tasks:

- Locate the Data Explorer control within GraphWorX64.
- Learn about the related properties, including Advanced options.
- See an example in Runtime.

3C.1.2 Support Files

You will not require any files in addition to GraphWorX64.

3C.2. About the Data Explorer

The Data Explorer is a control that allows the user to navigate all data sources with zero configuration. Using the data browser tree view, users can navigate between all data sources and view live values for real-time data, historical data, data sets, alarms, schedules, assets and BACnet.

Alternatively, the Data Explorer can be launched directly as a standalone application from the Start Screen or Start Menu (All apps > ICONICS Tools > Data Explorer) as well as the Runtime ribbon.

1. To insert a Data Explorer into a display, begin by selecting the **Controls** tab. This will expose all of the available controls, including the Data Explorer.

Figure L3C.1 – GraphWorX64 Controls Ribbon

2. Selecting the Data Explorer option will transform the cursor into a cross. Use the cross to draw the desired size of the Data Explorer on the canvas. This will insert a Data Explorer onto the canvas.

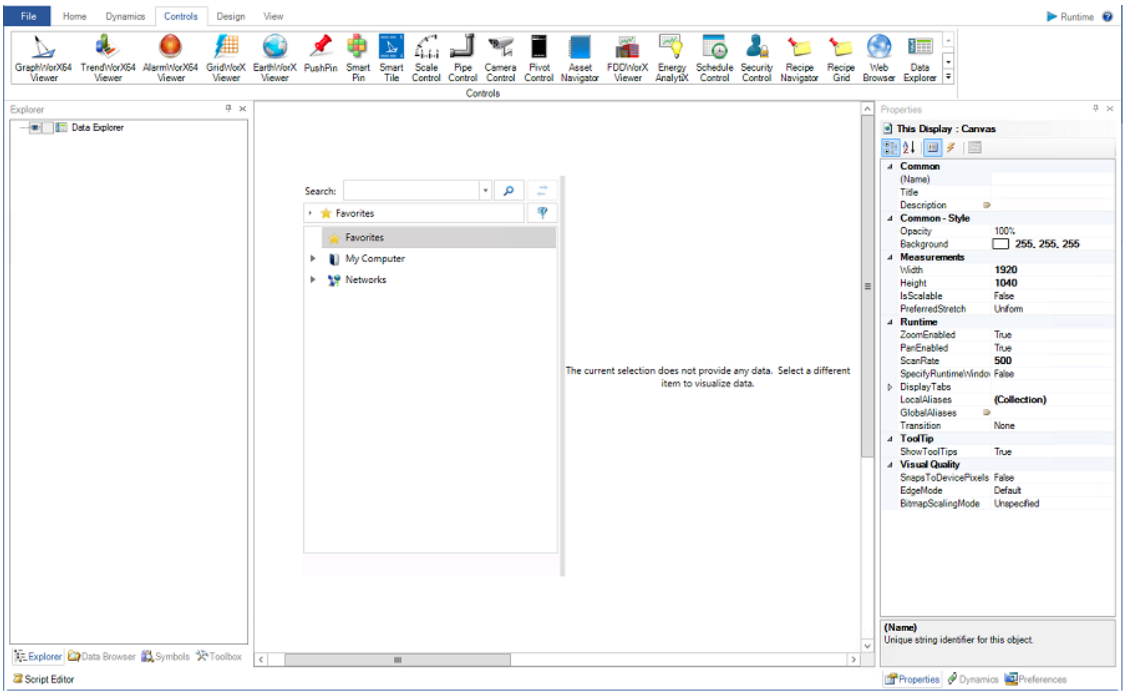


Figure L3C.2 - Data Explorer in a Display

3. The Data Explorer has a number of properties that can assist the user in customizing their control.

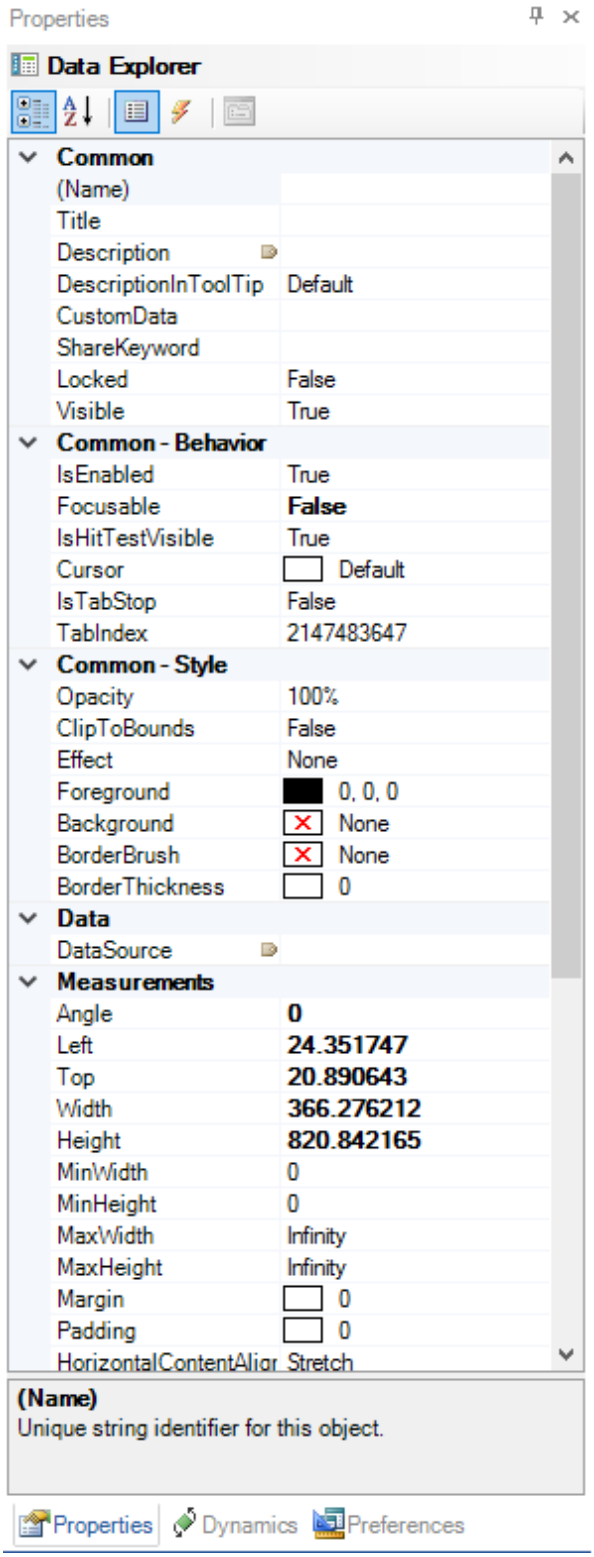


Figure L3C.3 - Data Explorer Properties

4. The following Properties can be set for the GraphWorX64 Viewer:

Common

- **Name** - Unique string identifier for this viewer.
- **Title** - Title for this viewer. Unlike the Name property, the Title does not have to be unique.
- **Description** - Description for this viewer. The Description typically appears as a tooltip for the viewer.

Common - Style

- **Opacity** - Overall opacity percent of this viewer. Range: 0% (fully transparent) - 100% (fully opaque).
- **Effect** - Visual effects (3D-edge, shadow, glow, blue) applied to this viewer.
- **Foreground** - Foreground color, gradient, or pattern of this viewer.
- **Background** - Background color, gradient, or pattern of this viewer.
- **BorderBrush** - Color, gradient, or pattern of the border surrounding this viewer.
- **BorderThickness** - Width of the border surrounding this viewer.

Data

- **DataSource** - Select the ellipse to browse for the data source.

Measurements

- **Angle** - The current angle of rotation for this viewer.
- **Left** - Horizontal position of top-left corner of the viewer.
- **Top** - Vertical position of top-left corner of the viewer.
- **Width** - Width of the viewer.
- **Height** - Height of the viewer.
- **Padding** - The amount of space surrounding the content of this viewer.

Style (unique to the Data Explorer)

- **DataBrowserVisible** - Set whether or not the Data Browser panel in this Data Explorer is visible.
- **GridViewStyle** - Specify a file defines the style used for Grid Views. If empty, a default style will be used. Select the ellipse to set a template file.
- **TrendViewStyle** - Specify a file defines the style used for Trend Views. If empty, a default style will be used. Select the ellipse to set a template file.
- **AlarmViewStyle** - Specify a file defines the style used for Alarm Views. If empty, a default style will be used. Select the ellipse to set a template file.

- **ScheduleViewStyle** - Specify a file defines the style used for Schedule Views. If empty, a default style will be used. Select the ellipse to set a template file.

Text

- **FontFamily** - Specify the name of the font used to render the text.
- **FontSize** - Specify the size of the font used to render the text.
- **FontStyle** - Specify the style of the font used to render the text (italics).
- **FontWeight** - Specify the weight of the font used to render the text.

5. Once set in Runtime mode, the user can leverage the built-in browser to find and automatically locate and render data and data sources. For example, locating and selecting a trend will automatically render a trend viewer while selecting a schedule will automatically render a schedule control.

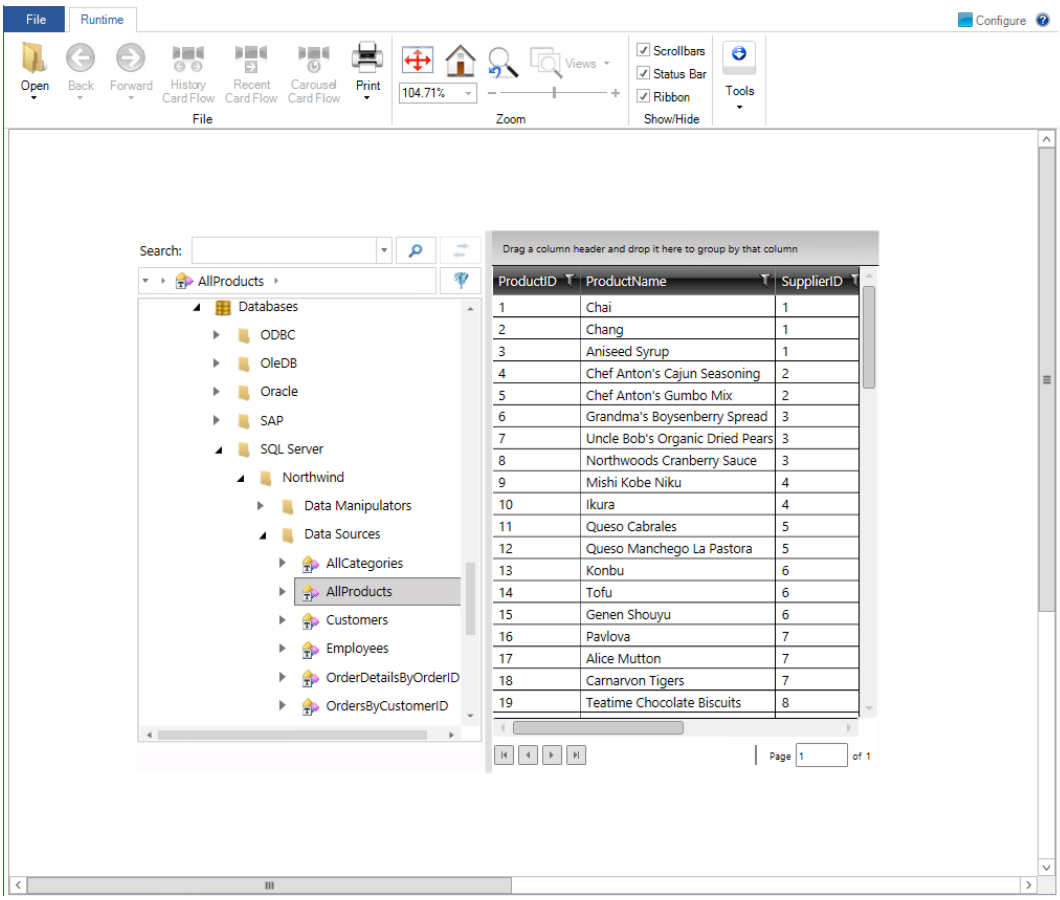


Figure L3C.4 - Grid Viewer in the Data Explorer in Runtime

6. In Runtime, users can take full advantage of all control and viewer features available to the specified display. Utilizing the properties, advanced users can further customize the Data Explorer for various other functions without using the default data browser.

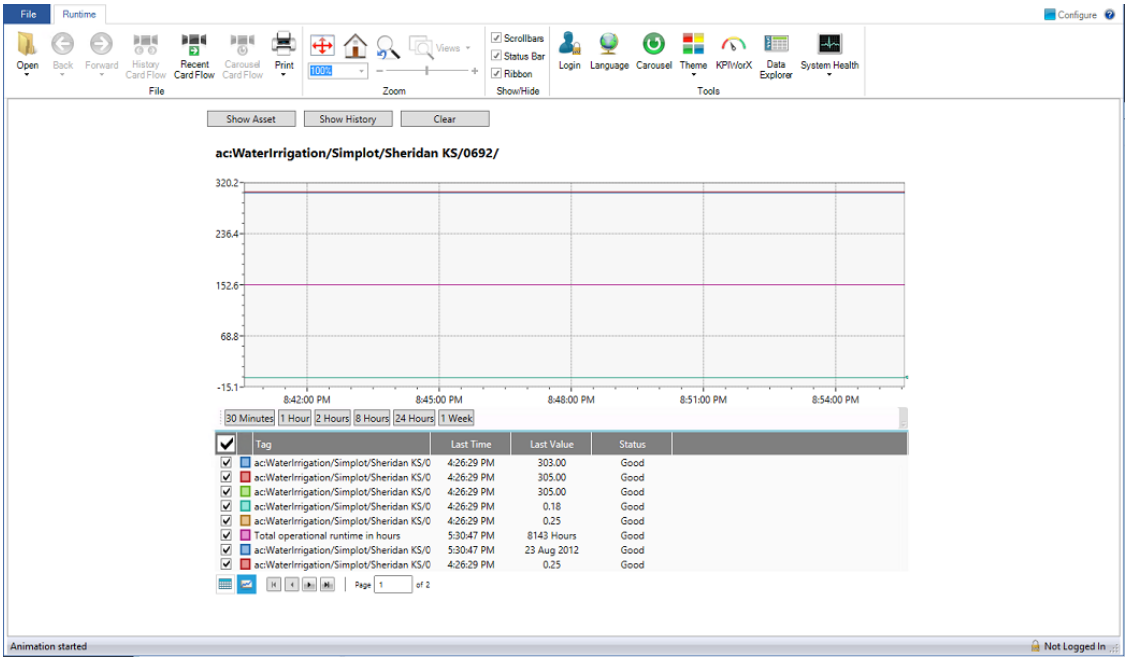


Figure L3C.5 - Advanced Use of the Data Explorer

This concludes this lab.

Unified Data Manager 4

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, global alarm subscriptions and registers by using the Unified Data Manager configuration interface, shown in Figure 4.1 below, under Platform Services node in the Workbench. All of the items and their settings are stored in SQL database based on your project. These items are exposed as OPC items and can be accessed from ICONICS applications via the Data Browser. UDM indulges traditional HMI development as well as deployment approach. It helps you reduce development time and eliminate redundant work.

You can access to UDM configuration within the Workbench by going under Platform Services and expanding Unified Data Manager.

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.

4.1 Objectives

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.

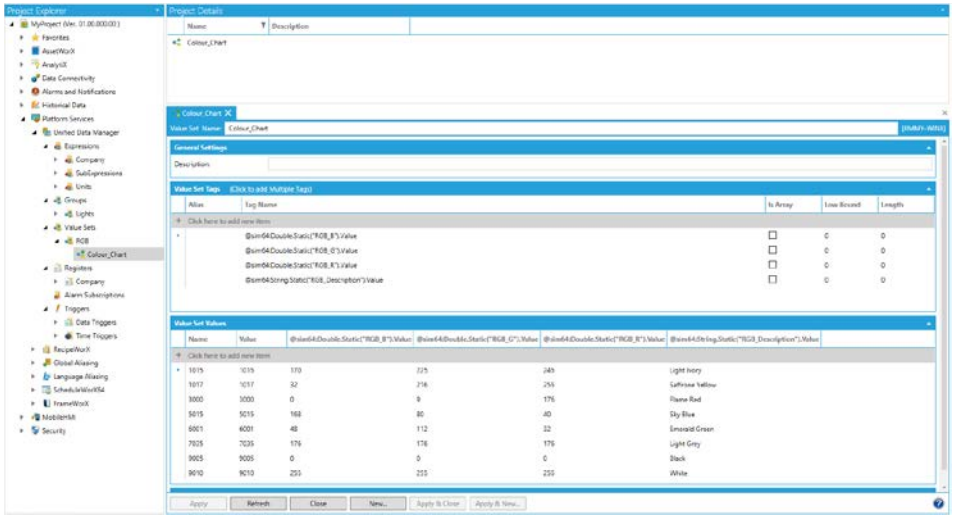


Figure 4.1 – Value Sets in Unified Data Manager Configuration

4.2 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 project, and then proceeds down from the categories to Unified Data Manager 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 **Configure Database** dialog under ProjectWorX tab on the ribbon of the Workbench, as shown in Figure 4.2. This dialog anticipants SQL connection setting as well as the applications you want to create new databases for. You do have the option to create database only for selective applications such as the UDM as shown in Figure 4.3. But keep in mind the Workbench is project oriented so it is recommended to have all application databases stored in one central database for easy management and deployment. Having standalone databases for each application will increase the difficulties of implementing and maintaining project.

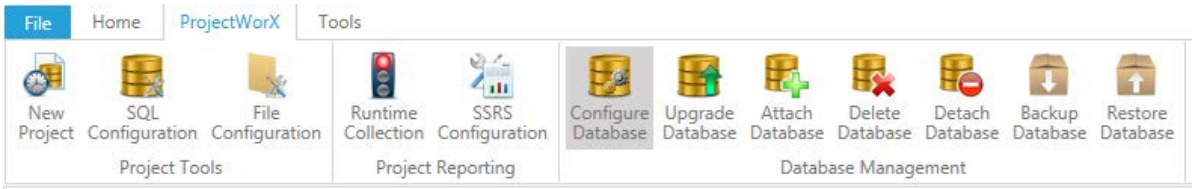


Figure 4.2 – Configure Database Button in the ProjectWorX Ribbon

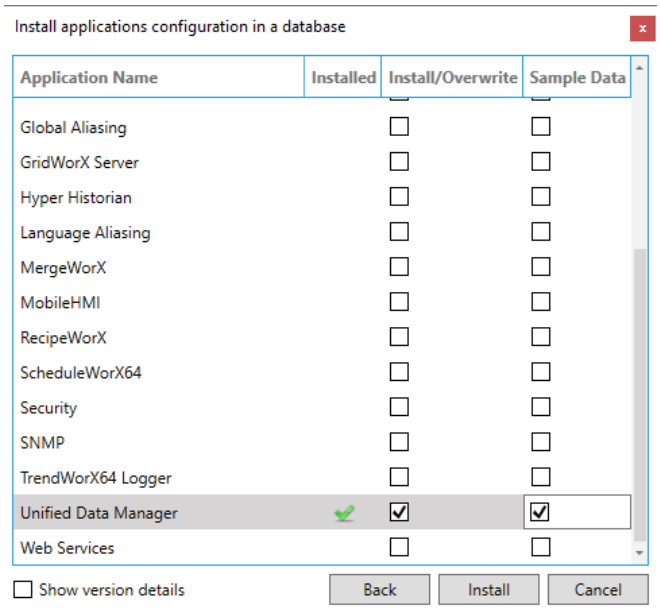


Figure 4.3 – Intall Applications Configuration in a Database

In the UDM navigation tree, the nodes **Expressions**, **Groups**, **Value Sets**, **Registers**, Alarm Subscriptions 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**, right-click on the item and select **Add Folder** from the dropdown list.
- To **create an item**, right-click on the item and select **Expression**, **Register**, **Value Set**, **Register**, **Alarm Subscriptions** or **Trigger**. Item choices vary depending upon 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 4.4 shows the popup menu on a folder and on a specific item.

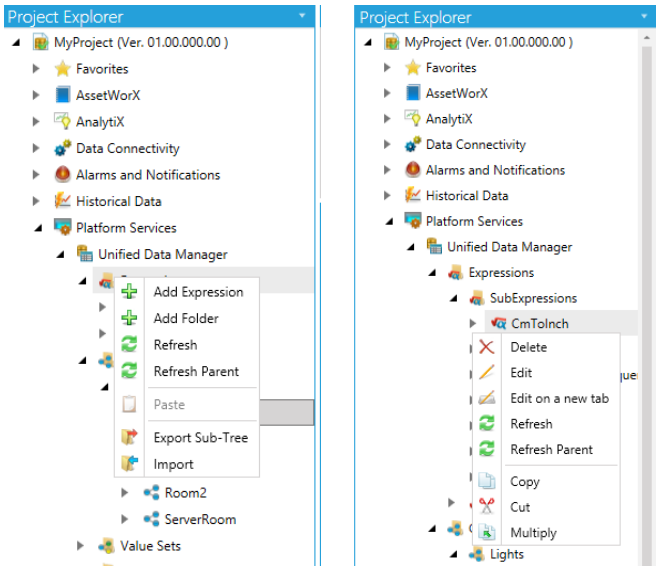


Figure 4.4 - Right-click Menus for Expression Folders (Left) and Items (Right)

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.

4.3 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 4.5. 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 *Celsius* defined, which stands for the current temperature reading from a given thermometer in degrees Celsius.

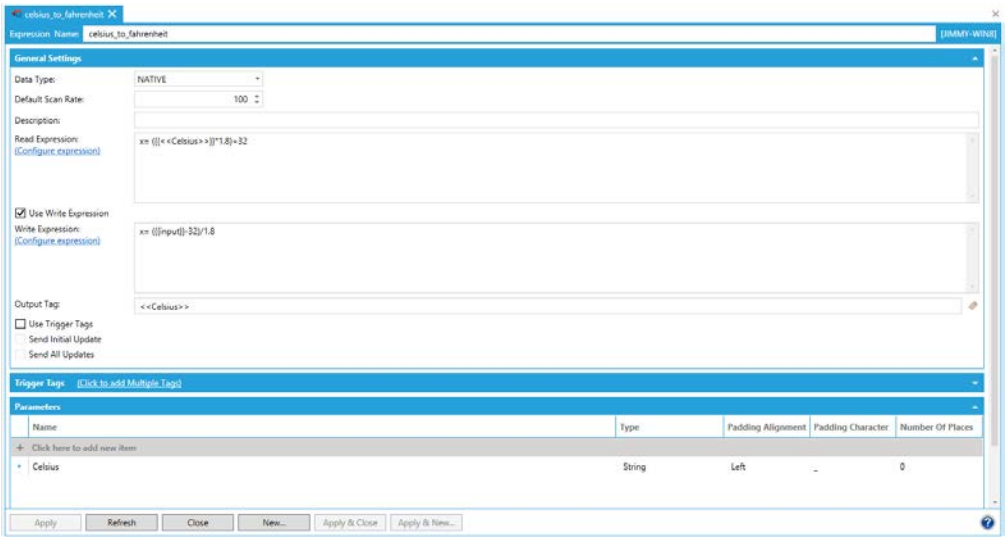


Figure 4.5 - Creating an Expression

The essential actions of an expression are contained in the **Read Expression** and **Write Expression** sections. There are some minor but important differences between the options that you see on these tabs, both of which are shown in Figure 4.6. In this figure, note the different content of the Read and Write expressions.

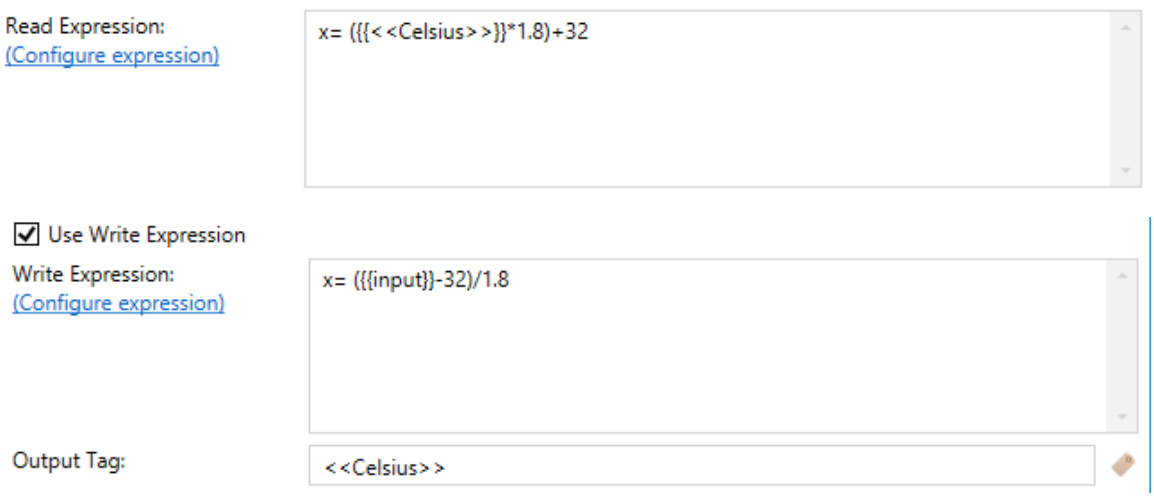


Figure 4.6 - The Read Expression (Above) and Write Expression (Below)

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= (<<Celsius>>*1.8) + 32

Double triangular brackets "<<" and ">>" surround the Celsius 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 link called **Configure expression**. The Configure Expression is a dialog that allows you to create expressions (and other functions), then checks for the validity of the expression.

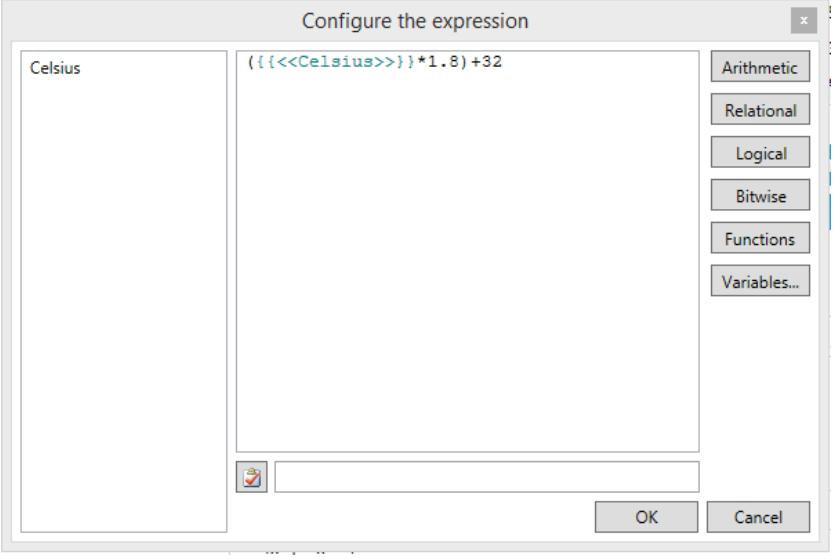


Figure 4.7 - Expression Editor

Each button in the dialog 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 Configure Expression is that although it checks your expression syntax, it does not check whether the expression is correctly applied.

You are not required to use Configure Expression. Instead, you can write your expression directly into the text box in the Read Expression section. 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, below the **Write Expression** section there is 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 My Computer→Data Sources→Platform Services→Expressions folder on the server that holds the Unified Data Manager configuration. In Figure 4.8, you can see the *celsius_to_fahrenheit* expression selected in Data Browser.

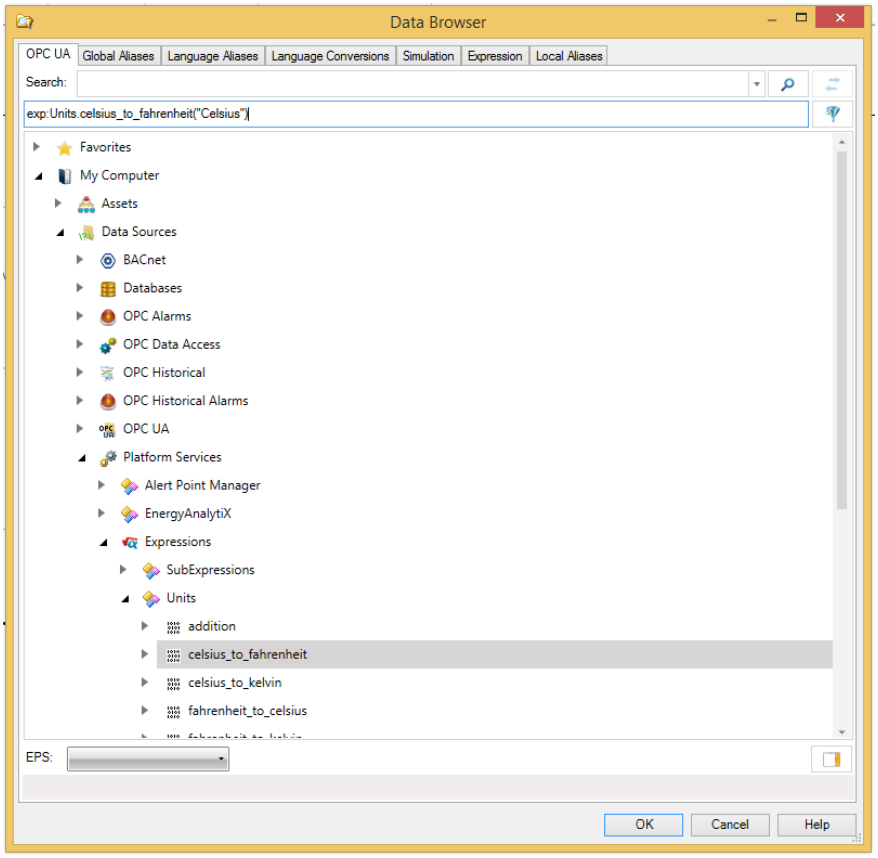


Figure 4.8 - Selecting Expression as a Data Source in the Data Browser

Note the string **"Celsius"** 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 (Celsius 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

4.4 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.

Under **General Settings**, you can change the data type of the tag, add its description. Under **Parameters**, you create parameters that can be used in the same way as parameters in expressions or other UDM items.

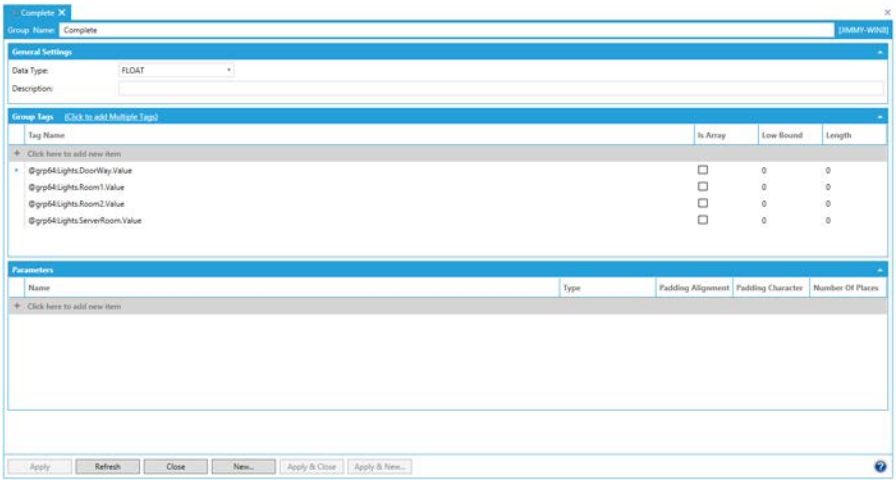


Figure 4.9 - Group Properties Dialog of Group Tag Item

Under **Group Tags**, 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.

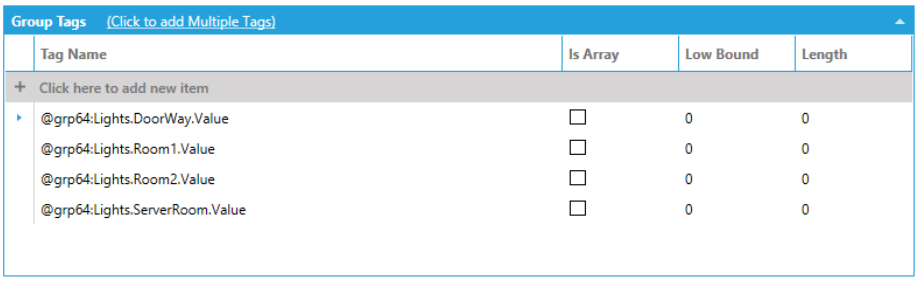


Figure 4.10 - Group Tags Dialog of Group Tag Item

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 Data Browser and browsing to My Computer→Data Sources→OPC Data Access→Groups (highlighted in Figure 4.11) on the server that holds the Unified Data Manager configuration.

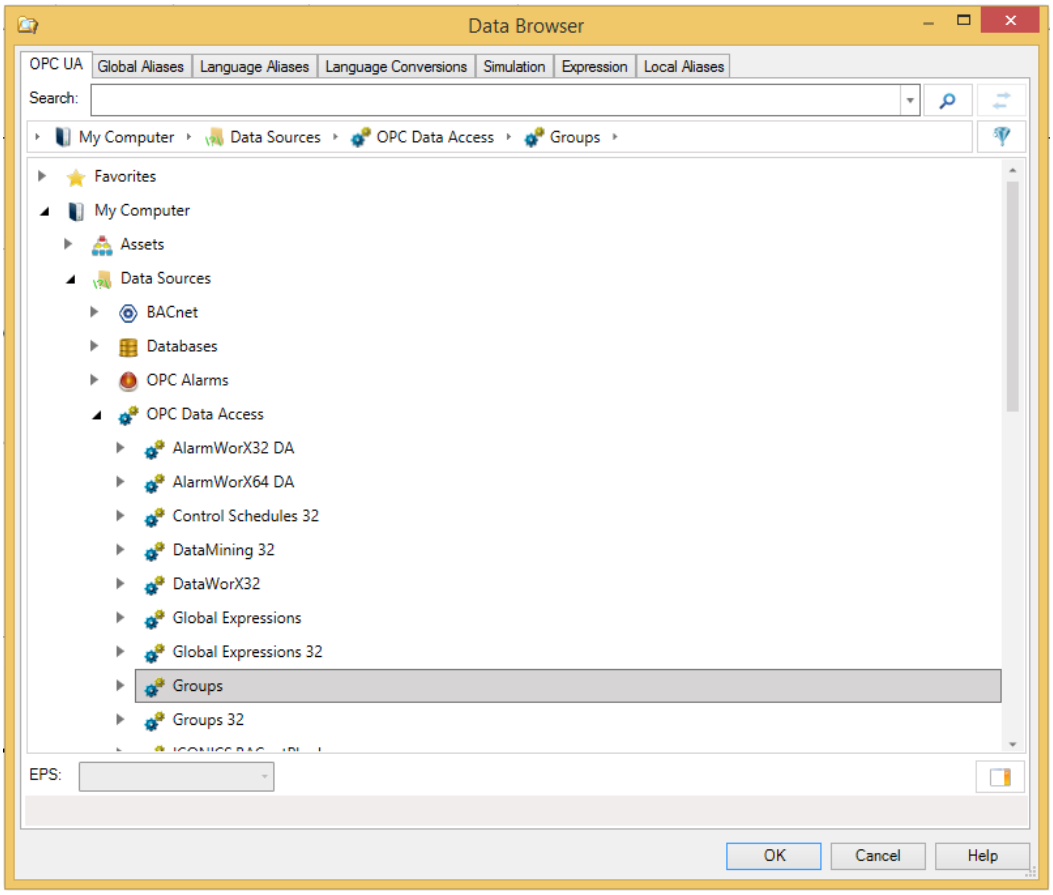


Figure 4.11 - Selecting Groups in Data Browser

4.5 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. Under the **General Settings** section, you can enter a description and under **Parameters** section, you can add parameters.

Add members to the value set under the **Value Set Tags** section. 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 Value Set Tags tab in the image below.

Value Set Tags (Click to add Multiple Tags)

Alias	Tag Name	Is Array	Low Bound	Length
+ Click here to add new item				
	@sim64:Double.Static("RGB_B").Value	☐	0	0
	@sim64:Double.Static("RGB_G").Value	☐	0	0
	@sim64:Double.Static("RGB_R").Value	☐	0	0
	@sim64:String.Static("RGB_Description").Value	☐	0	0

Figure 4.12 - 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 **Value Set Values** section as shown in the following image.

Value Set Values

Name	Value	@sim64:Double.Static("RGB_B").Value	@sim64:Double.Static("RGB_G").Value	@sim64:Double.Static("F
+ Click here to add new item				
1015	1015	170	225	245
1017	1017	32	216	255
3000	3000	0	9	176
5015	5015	168	80	40
6001	6001	48	112	32
7035	7035	176	176	176
9005	9005	0	0	0
9010	9010	255	255	255

Figure 4.13 - 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 a 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 Value Set Values tab.

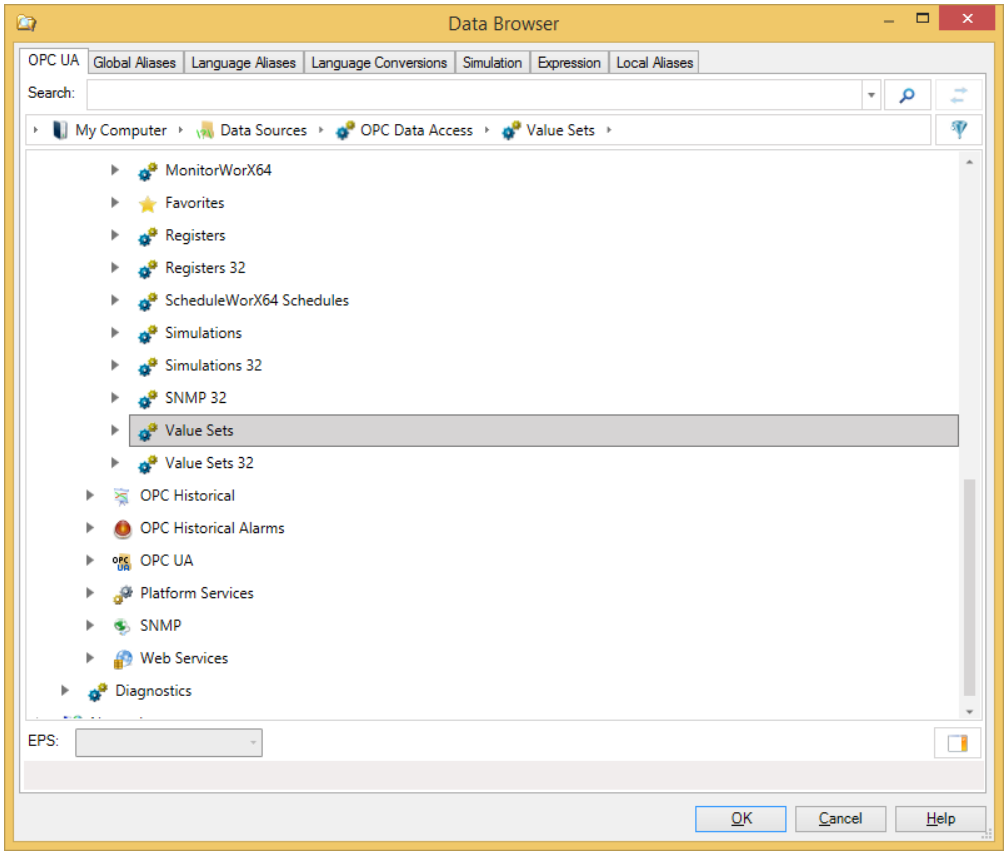


Figure 4.14 - Selecting the Value Sets Folder in the Data Browser

4.6 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.

Under the **General Settings** section, shown below, you define the **Data Type** and a **Description**. You can enable or disable "Input Updates Propagation", which indicates whether values changed on input tag will be sent to an output tag. You decide whether the register is read only or writable. And lastly you can choose whether or not you want to **Release tags when not in use,** 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.

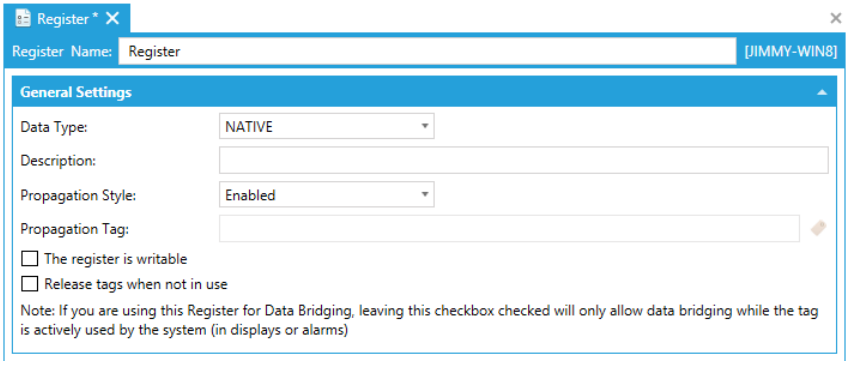


Figure 4.15 - Register Properties Configuration

How you configure the **Input and Output Tag** sections 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 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.

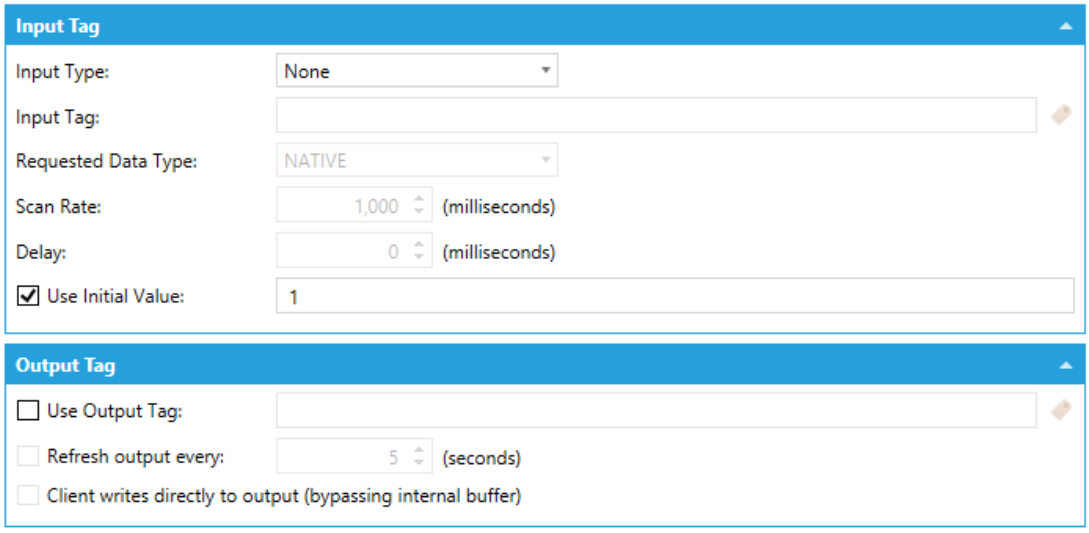


Figure 4.16 - Configuring Input/Output Properties of Register

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.

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.

4.6.1 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.

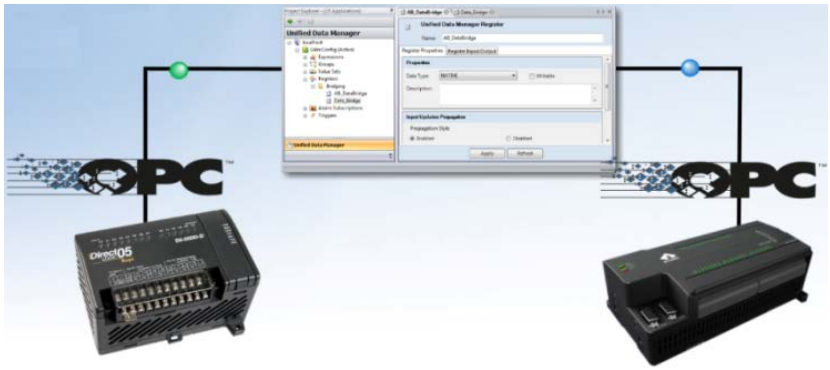


Figure 4.17 - Data Bridging

For bridging, you only need to specify an “Input OPC Point Tag” and an “Output Tag”. To select a tag, click the field’s tag button to open the Data Browser.

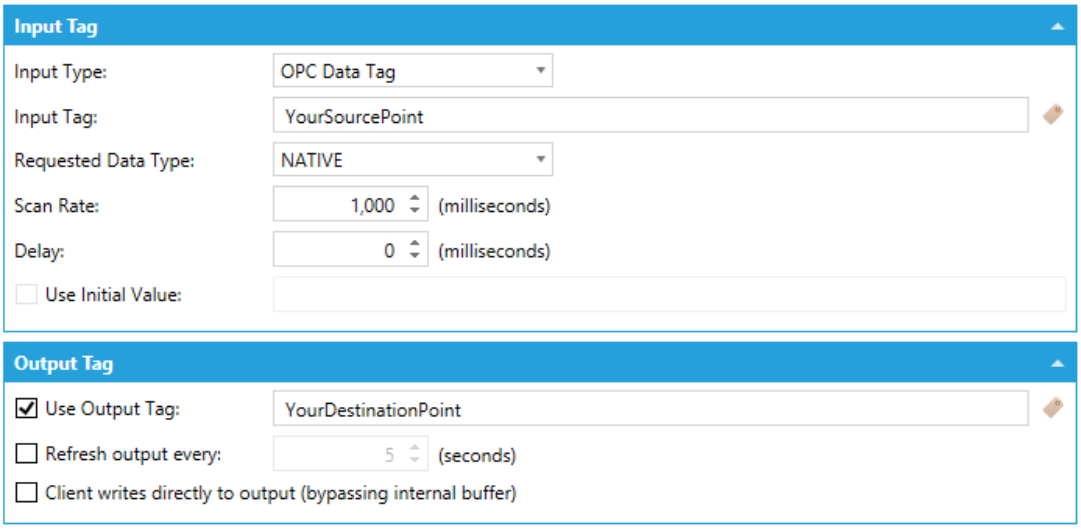


Figure 4.18 - Bridging configuration

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.

4.6.2 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 4.19, 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.

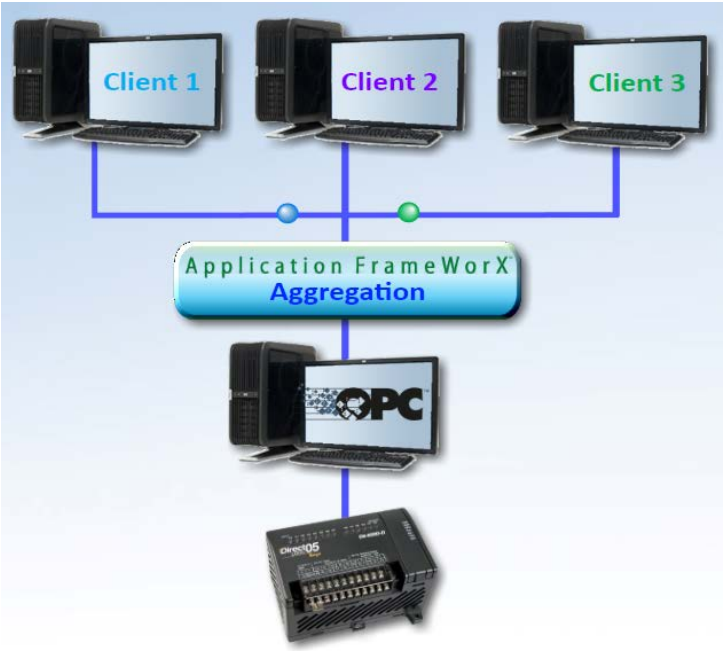


Figure 4.19 - Data Aggregation

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.

4.6.3 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*.

4.6.4 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 "Use Output Tag" checkbox in the Output section of the **Output Tag** section.

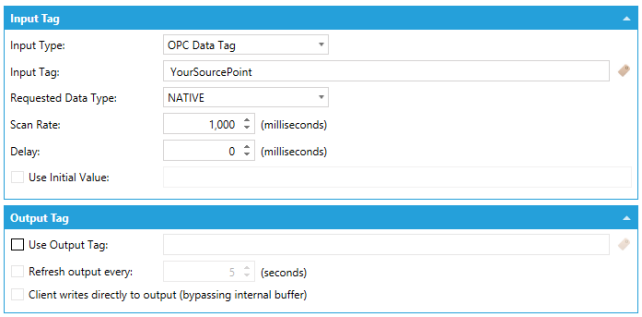


Figure 4.20 - Disabled Output Tag

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 **General Settings** 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 4.21.

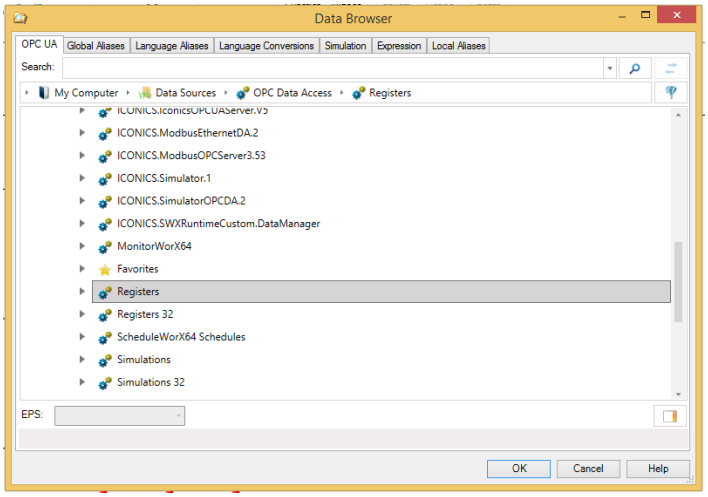


Figure 4.21 - Selecting Register in Data Browser

4.7 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.

In addition, either type of triggers can be scheduled to fire away by configuring the **General Settings** section, shown below.

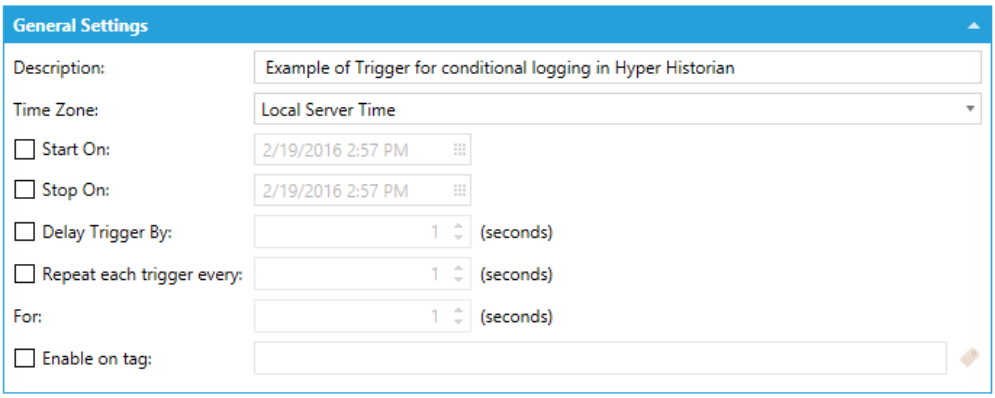


Figure 4.22 – General Settings for a Data Trigger

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.

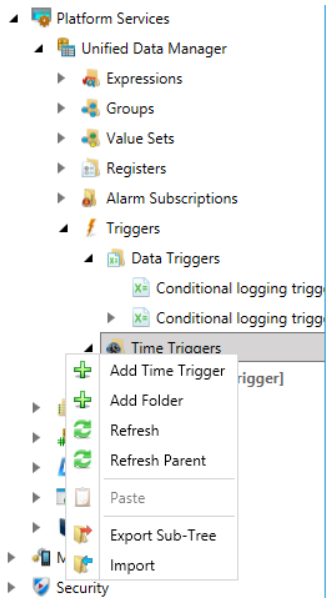


Figure 4.23 - Creating a New Trigger

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.

4.7.1 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 button next to the "Trigger Tag" field.

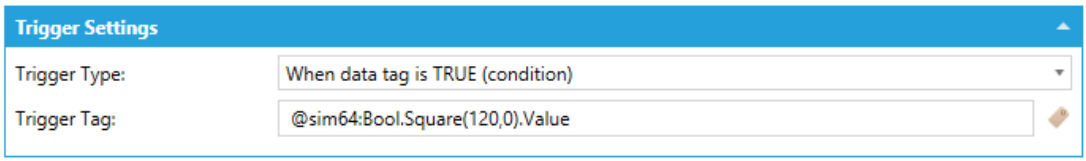


Figure 4.24 - Data Trigger Options with Selected Data Tag

4.7.2 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 4.25 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

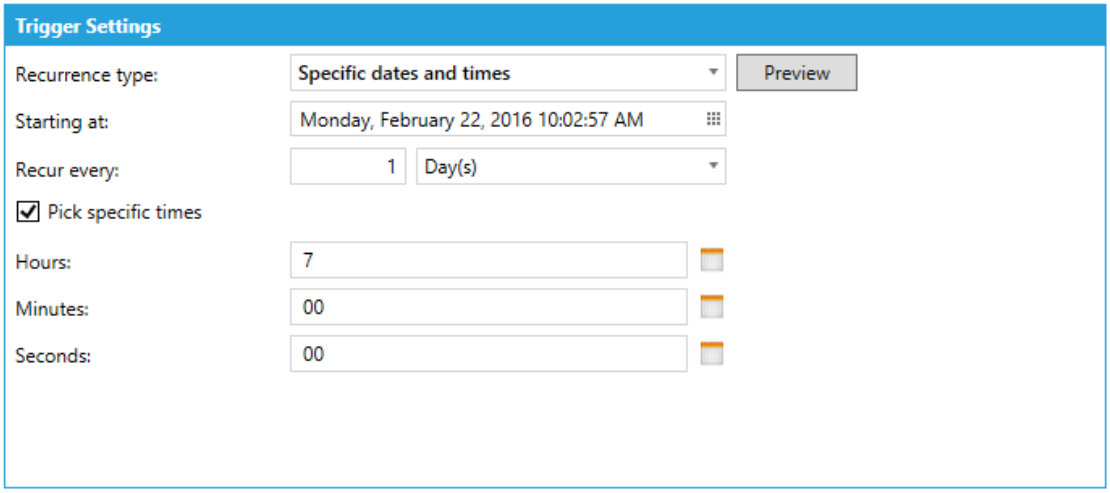


Figure 4.25 - Time Triggers Configuration

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.

4.7.2.1 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.

4.7.2.2 Time Interval Recurrence

To set a **time interval** at which an event fires, select the "Time interval" Recurrence type (as shown in Figure 4.26), 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).

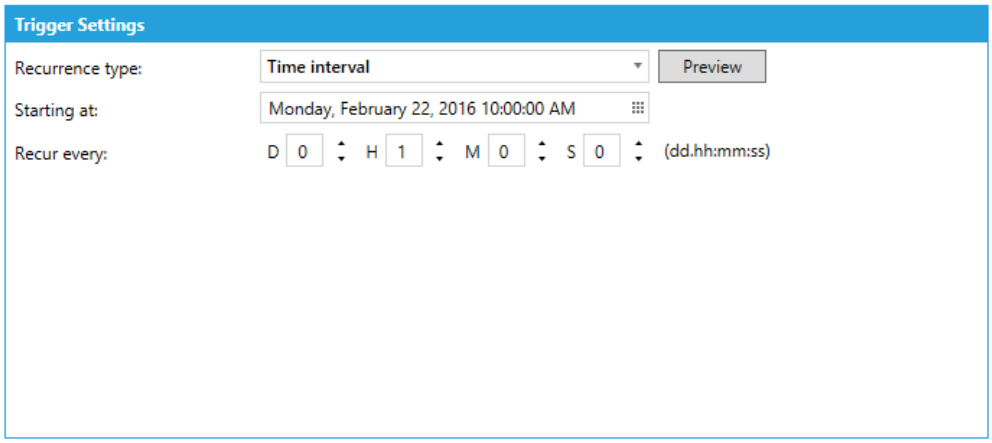


Figure 4.26 - Time Interval Selection

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.

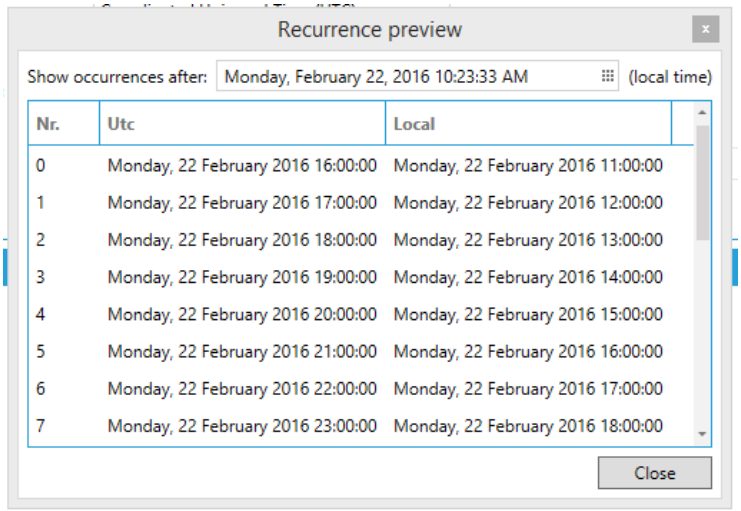


Figure 4.27 – Previewing a Trigger’s Starting Times

4.7.2.3 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.

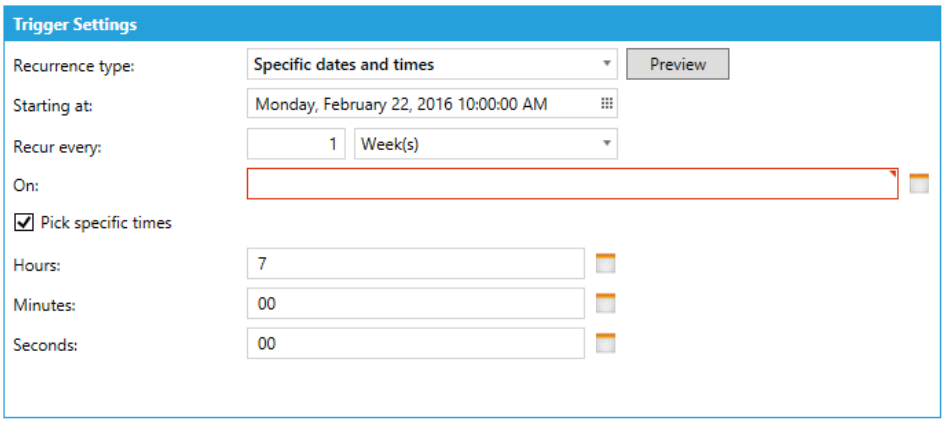


Figure 4.28 – Setting Trigger for Specific Dates and Times

The Recurrence Details section changes depending on the selected interval. (Shown in the following image is the *Recur Every # Week(s)* selection.) Select the different days, months or times in the Recurrence details section.

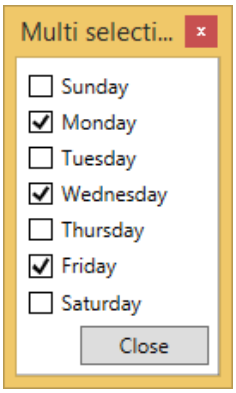


Figure 4.29 – Recurrence Details

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.

4.7.2.4 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.

4.7.3 Using Triggers

Data triggers and time triggers can be used in other ICONICS applications. For instance, when you browse for triggers in MergeWorX, the dialog box looks like below.

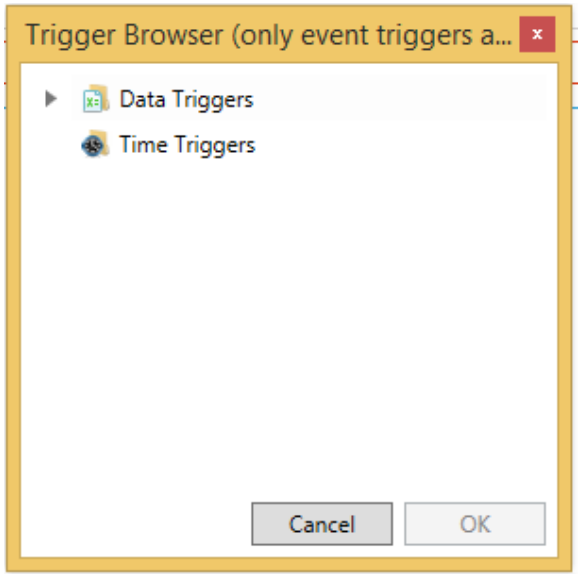


Figure 4.30 - Selecting a Trigger in Hyper Historian

Lab: Expressions and Registers L4

4.1. Lab Overview

This introduction to expressions and registers is intended to expose some of the available features and also demonstrate some rather simple expressions in an attempt to enhance your thoughts on how these very useful and simple to implement procedures can assist in performing some rather sophisticated mathematical exercises. There will also be some demonstration of how expressions can be interconnected with registers to produce some intended result.

Estimated time to complete this lab: **30 minutes**

4.1.1. Task Checklist

In this lab, you will complete the following tasks:

- Create folder for Registers and Expressions in the UDM.
- Create Holding Registers used for expression Inputs.
- Configure Expressions as a Counter and a Random Runtime Generator.

4.1.2. Support Files

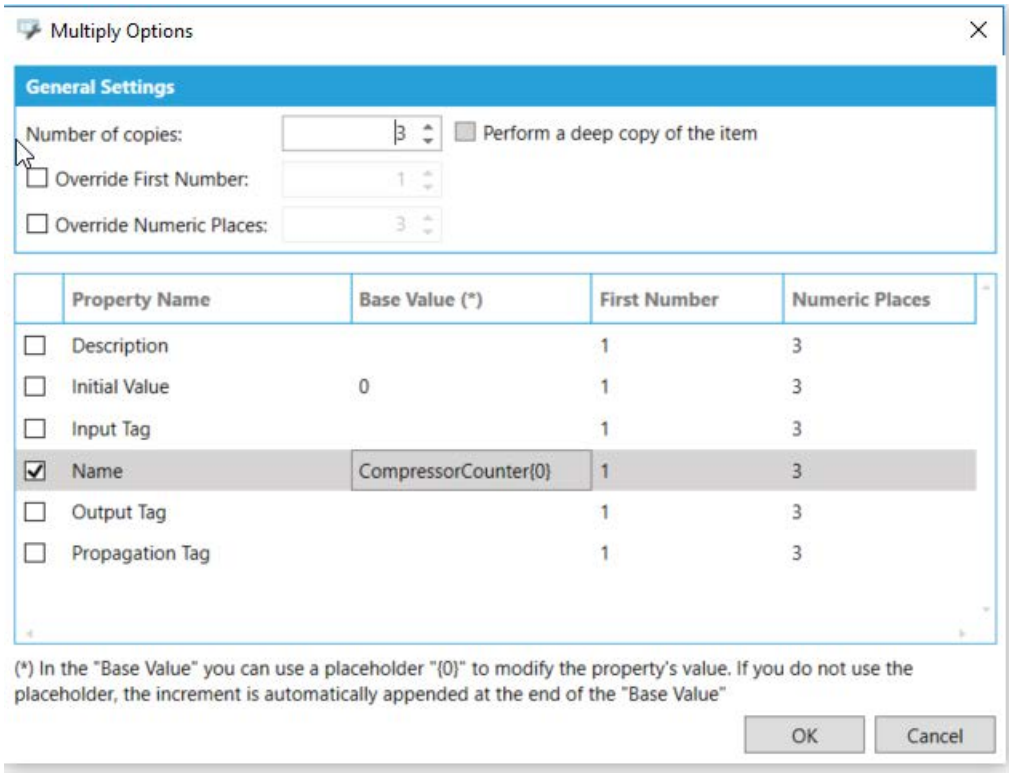
To complete this lab, you will need the following file(s):

- CompressorTest.gdfx

4.2. Registers

Registers are named variables that all GENESIS64 clients can use. If you have used registers inside DataWorX32, then the features of Unified Data Manager registers are a subset of the features available for registers in DataWorX32. Unified Data Manager doesn't support the use of Unified Data Manager registers through OPC, as an alias or redundancy register, or allow automation through Visual Basic. The most significant difference between Unified Data Manager and DataWorX32 registers is that Unified Data Manager registers are not real OPC tags, and Unified Data Manager registers can be accessed only with GenClient and the appropriate plug-in. The Unified Data Manager is also missing special registers for data aliasing, switching and redundancy. Refer to the GENESIS64 online help system for details on the various register types.

1. Open **Workbench,** go under **Platform Services** on the left side tree and expand the **Unified Data Manager** provider.
2. Add a folder called **Compressor** under **icTonics**
3. We will continue by adding four registers configured as **Holding Registers**.
4. Right click on the **Compressor** folder and select **Add Register**.
5. Name the register **CompressorCounterInput**.
6. **Under** the **Input Tag** section, set **Input Type** to "None" and make sure "Use Output Tag" is unchecked under **Output tag** section. This makes the register a **holding register**.
7. **Check** the **Use Initial Value** checkbox and enter a value of **0** (zero).
8. Click **Apply**.
9. Right Click on the newly created register called CompressorCounterInput and select "**Multiply**" from the dropdown menu that appears.
10. Configure the Multiply settings by changing the "**Name**" Row from CompressorCounterInput to **CompressorCounter{0}**. Change the number of copies to **3**. For reference see the screenshot below to match your values.



12. Once this matches make sure to hit the "**OK**" button on the dialog menu

13. We are going to rename each of the newly created Registers. Follow the naming convention as the table below:

Rename From	Rename To
CompressorCounterInput	CompressorCounterInput
CompressorCounterInput001	CompressorCounterOutput
CompressorCounterInput002	CompressorRandomInput
CompressorCounterInput003	CompressorRandomOutput

14. Please note that when changing the names of these newly created Registers that you go in and make them writable. This can be done by double clicking on each Register and making sure the "The Register is writable" checkbox is checked.

15. Apply & Close your changes.

16. Right-click on the Unified Data Manager provider in the tree and select Refresh.

4.3. Expressions

Expressions are combinations of data sources, numbers, operators and grouping symbols (such as brackets and parentheses) arranged in a meaningful way that can be evaluated. During Runtime mode, OPC servers resolve the data source for the expression and the expression is evaluated. There are two types of Expressions, local and global**.** Each of the two types are described in detail in the GENESIS64 online Help system.

We will begin by creating a one-second counter.

1. Open **Workbench** if it is not already open**;** expand **Platform Services** and expand the **Unified Data Manager** provider.
2. Add a folder called **icTonics** under **Expressions**.
3. Add a folder called **Compressor** under **icTonics**.
4. Right-click on the Compressor folder and select **Add Expression**.
5. Name the Expression **CompressorOnTimeCounter**.
6. Set the **Default Scan Rate** to **1000** (one second).
7. Under the **Read Expression** section. We will be building the following expression syntax.

   x=setvalue({{@rgs64:icTonics.Compressor.CompressorCounterOutput.Value}},{{ @rgs64:icTonics.Compressor.CompressorCounterOutput.Value}} + tonumber({{@rgs64:icTonics.Compressor.CompressorCounterInput.Value}}))

8. There's many ways you can create an expression. For instance, you can manually type in the syntax or you can use the expressions editor.
9. The following steps will describe how to select and insert a function as well as browse for and insert a tag into the expression. Beyond that, you will be on your own to complete the expression. Learning the expression editor may take some practice.
10. Click on the blue **(Configure expression)** link under **Read Expression** to launch the **Expression Editor** dialog.
11. From within the **Expression Editor** dialog, select everything in the window. This will be replaced as we proceed.
12. Click the **Functions** button and select the **setvalue** function.

    Note: Functions are case sensitive.

13. Select the word **variable**. This will be replaced when we select a tag.

Note: Be sure not to select any more characters than the ones you want to replace. For example, if you select the comma with the word 'variable', the comma would also be replaced and you would get a syntax error when you run the syntax checker.

14. Click the **Variables…** button and browse to select the tag**My Computer → Data Connectivity → OPC Data Access → Registers → icTonics → Compressor → CompressorCounterOutput** and click **OK**.

15. Proceed by selecting the word **value**.

16. When your expression is complete (the syntax should match step 7 above), click on the syntax checker button . If your syntax is okay, the box will indicate "**No Errors.**"

17. Click **OK**. Be sure to **Apply** your changes.

18. **Create** a second expression and leave all the settings in the expression at the default values.

19. Name this second expression **CompressorOnTimeRandom**.

20. Click in the **Read Expression** tab.

21. Click on the blue **(Configure expression)** link under **Read Expression** to launch the **Expression Editor** dialog.

22. From within the **Expression Editor** dialog, click on the **Logical** button and select the **IF THEN ELSE** function.

23. Within the IF statement replace the red "**BooleanCondition**" text with the float tag used in the expression below located at **My Computer → Data Connectivity → OPC Data Access → Simulations → Float → Random**.

Note: The random tag will look like this:

@sim64:Float.Random(Period[sec],Min,Max,Phase[deg]).Value

You will need to set the parameters manually:

x=IF {{@sim64:Float.Random(1,0.0,10.0,1).Value}} > {{@rgs64:icTonics.Compressor.CompressorRandomInput.Value}}THEN setvalue({{@rgs64:icTonics.Compressor.CompressorRandomOutput.Value}}, {{@rgs64:icTonics.Compressor.CompressorRandomOutput.Value}}+1)

24. The **ELSE** statement is not needed and can be removed.

4.4. Adding the 'KeepAlive' Register

Expressions won't execute if they are not actively being used by the system (in displays or alarms). This exercise will demonstrate this scenario since you may run into while developing expressions.

1. Open the **CompressorTest.gdfx** display in the GraphWorX64 editor.
2. Select the top process point under the Expression Output label.
3. Delete it using a **Shift+Delete**.
4. Enter **Runtime**. You should notice that the Random Output no longer increments. This is because nothing in the system is requesting updates from the expression. The following steps will correct this issue.
5. From within the Workbench, under the **Platform Services** → **Unified Data Manager** → **Registers** → **icTonics** → **Compressor** tree node, add another register named **KeepAlive**.
6. Under **Input Tag** section**, set** the **Input Tag** to:

   @exp64:icTonics.Compressor.CompressorOnTimeRandom.Value

7. Click **Apply** and restart the **UDM** service.
8. Check the **CompressorTest** display in **Runtime** and the updates should occur normally.

This concludes this lab.

AssetWorX 5

AssetWorX is an enterprise control system for integrating business and manufacturing control systems in Workbench. It provides a tree-like structure in which you can build your enterprise in the form of physical locations and business units, and organize equipment such as buildings and machinery in one centralized system for analysis by the ICONICS GENESIS64 suite of products. The tree-like structure provides a functional hierarchy for navigation and data roll-ups. Physical locations and areas of responsibility can be identified in this hierarchy.

5.1. Objectives

After completing this chapter, you should know:

- The Architecture of the AssetWorX provider in Workbench
- How to configure an AssetWorX database and Data Import or Export
- How to define Units of Measure, Temperature Sources/Instances and Polling Groups
- How to build Hierarchical Tiers within the Asset Tree
- How to define Tree View Rules
- How to visualize Assets in Runtime

5.2. Introduction to AssetWorX

When learning to use AssetWorX, one must first become acquainted with Nodes. Nodes are the expandable selection in the Asset Tree. For example, in the following figure, underneath "AssetWorX" you see the tree nodes “Equipment”, “Equipment Classes”, “Tree Views”, and “Product Configuration”. The individual Nodes are what composes an Asset Tree.

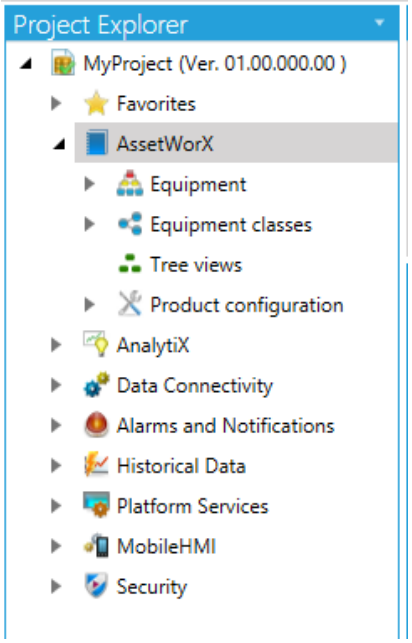


Figure 5.1 - Nodes in Asset Tree

Features of AssetWorX

- Designed for Massive Systems - able to handle millions of tags
- Object-oriented Database - to work with real world objects
- AssetWorX Runtime - store, cache and checkpoint tags
- Base for AnalytiX Products - integration with Energy AnalytiX/Facility AnalytiX

The asset tree provides a way to organize data sources (OPC, database, web services) and visualizations (such as human-machine interfaces (HMI), charts, and reports) in a logical hierarchical structure. For example, rather than OPC data sources being organized based on the address space of the server itself, these data sources can be organized by geographic/physical locations of the associated sensors (for example, by site, building, floor, and machine).

AssetWorX allows you to collect data from multiple applications and configures it into a single tree, organized in a logical structure that you define. AssetWorX comes with four predefined nodes (as shown in screenshot above) at the highest level of the structure. These nodes are:

- **Equipment**, which is where you configure your equipment and house your hierarchical asset tree.
- **Equipment Class,** which is where you add templates for your equipment so you can reuse later on.
- **Tree views,** which is where you define view rules to re-organize how the asset tree looks like when you visualize it on front end.

- **Product Configuration**, which is where you identify your preferences for AssetWorX settings.

It is in these four nodes that you configure your asset tree, define the structure of the asset tree, and set general configuration options for AssetWorX.

At any enterprise, equipment is interconnected with its physical location and in business units that are easily organized hierarchically. You define the interconnected relationships of equipment using a tree-like structure called the **asset tree**. Each piece of equipment gets its own node in the asset tree. Each piece of equipment in the asset tree can have its own associated properties. A property might be a variable data source, a reference to an HMI graphic, or a static value. It is at these nodes where you can analyze data to obtain an ongoing, current energy analysis for each piece of equipment, where both configuration and runtime features are available.

In configuration mode, for each node in the tree, you can add properties like dynamic data connections or associated runtime layouts. In runtime mode, the tree becomes the interface from which runtime users launch reports and HMI layouts, as well as visualize alarm summary data.

The asset tree conforms to a hierarchy of tiers that are predefined in AssetWorX. The hierarchical structure of the assets follows the ISA-95 standard for organization levels. Each tier in the hierarchy represents a type of equipment, such as cities, campuses, buildings, product lines, stockrooms, and machinery. Rules for each tier determine how equipment can be configured on each tier. A set of default asset tree tiers is installed for you as an example, but it is expected that you will customize these tiers later on.

AssetWorX is a Workbench tool for setting up the tools and applications that are used by other applicable Providers (for instance, Energy AnalytiX, FDDWorX, etc.).

5.2.1. Architectural Overview of AssetWorX

The AssetWorX provider found in GENESIS64's Workbench provides data sources and visualizations (HMI, charts, etc.) in a logical hierarchical structure. The Asset Tree itself provides intuitive, easy setup and configuration to any level of aggregation, from an entire campus or enterprise down to a specific meter or user. AssetWorX, following ISA-95 and S88 standards for organizational levels, allows users to import Autodesk® Building Information Modeling (BIM), and AutomationML models for streamlining projects, deliver them faster, and more economically.

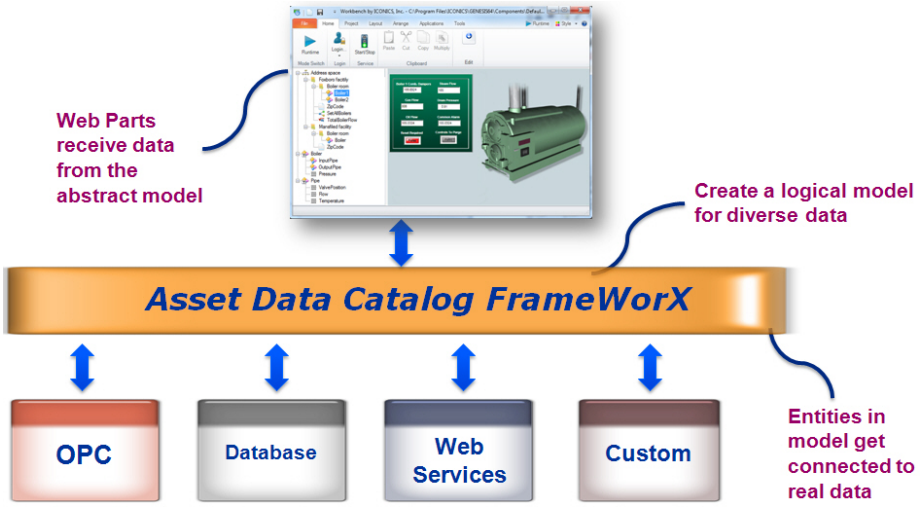


Figure 5.2 - AssetWorX Architecture

Below is an example of several asset tree characteristics.

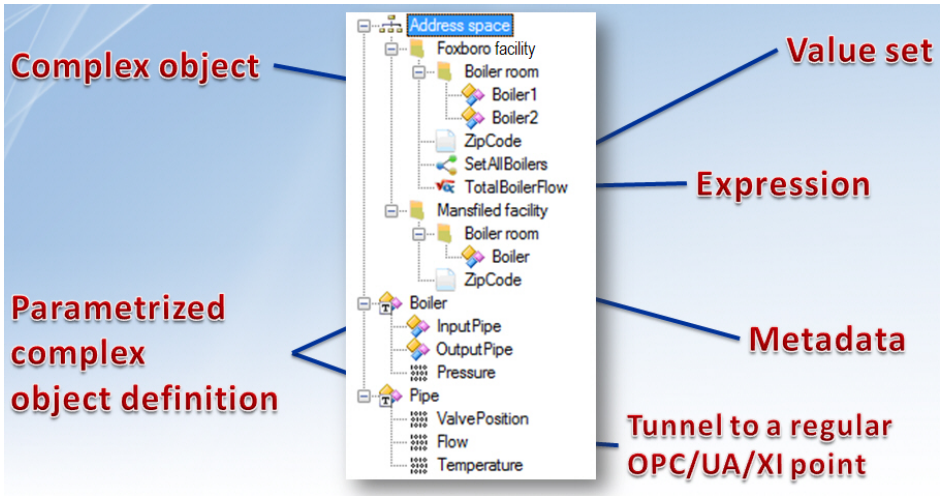


Figure 5.3 - Asset Tree Characteristics

5.2.2. Assets vs. Tags

Tags	**Assets**
Complex Path Structure	Tree Structure
Complex Naming ➤ @opc:…/…/…/… .Value	Simplified Naming ➤ Rapid Mixer 01
Varied Organization ➤ OPC Server Dependent ➤ Protocol Dependent	Hierarchical Organization ➤ Empty Asset Folders ➤ Parent/Child Relationships
BACnet/OPC-UA ➤ Object Oriented ➤ Non-Hierarchical Tags	Associated Properties ➤ Objects with Multiple Properties per Asset

5.2.3. Quick Start with AssetWorX

This section outlines the general steps you must take in order to use AssetWorX to build an asset tree. (Take note that "equipment nodes" are also known as "assets".)

1. Create an AssetWorX Configuration Database, and Connect to it.
2. Define basic information required on the Product Configuration node.
3. Follow the Steps to Building an Asset Tree, then add equipment to the Asset Tree.

5.3. AssetWorX Configuration Databases

In Workbench, any AssetWorX database that you create is shared among all AnalytiX providers (e.g. Energy AnalytiX and FDDWorX). Any changes made to an AssetWorX database will be propagated to all AnalytiX providers. You will see an example of an existing and added AssetWorX database during this chapter's lab. Activating an AssetWorX Configuration Database

Setting the active database in AssetWorX sets the active database for all participating servers. You can create multiple AssetWorX databases, but only one can be active and viewed at any time. This concept will be reviewed in the lab for this chapter.

5.3.1. Exporting and Importing

5.3.1.1. Exporting Data from AssetWorX

You can export data from an AssetWorX database, asset tree, or equipment class node to a .CSV (comma-separated values) file. The file can then be imported into AssetWorX for use elsewhere. Note that you can export from any equipment node in the asset tree, or from any equipment class. The name you assign to the .CSV file that is created should identify the exported content.

5.3.1.2. Importing Data into AssetWorX

You can import data into the AssetWorX database or into the asset tree or equipment class node. It is assumed that the data you want to import has been exported from an AssetWorX database and is stored in a .CSV (comma-separated values) file. Be aware of the data being imported so that you import it to an appropriate location in the Project Explorer. For example, do not import equipment classes into the asset tree.

5.4. General Setup in AssetWorX

5.4.1. Enabling the Available Solutions

When you create an asset, you will have the option of adding AnalytiX extension. If the solution is disabled, the option of adding the extension will also be disabled. This is done via the "Installed extensions" branch of the 'Product configuration' node in the AssetWorX provider, as shown below.

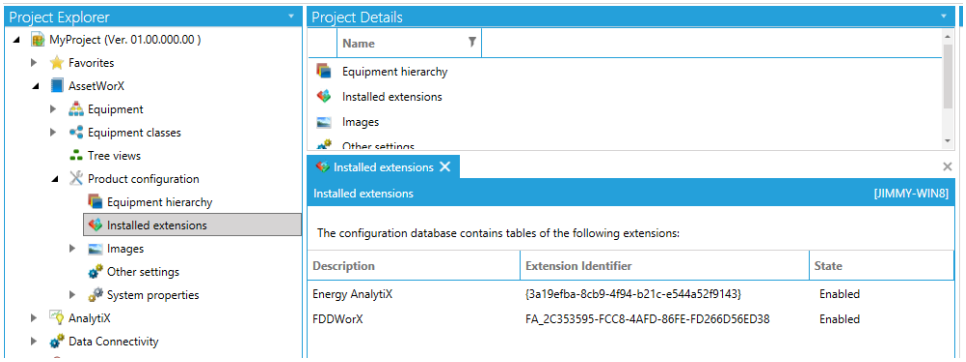


Figure 5.4 – Available Solutions: Energy AnalytiX Enabled

5.4.2. Other Settings Node

In AssetWorX, the Product Configuration node is where you set your system settings. In AssetWorX, under ***Product Configuration***, select **Other Settings**. This displays the Other Settings window, shown below. It has three sections.

General - Deals with Equipment Classes, which you will learn about later in this training.

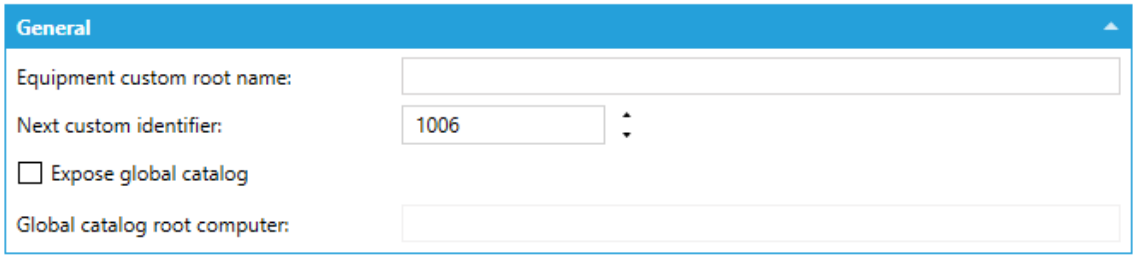


Figure 5.5 - Other Settings: General Tab

Cache Data Store – allows you to set a data storage location. Clicking on the 'Configure connection' link opens a setting window.

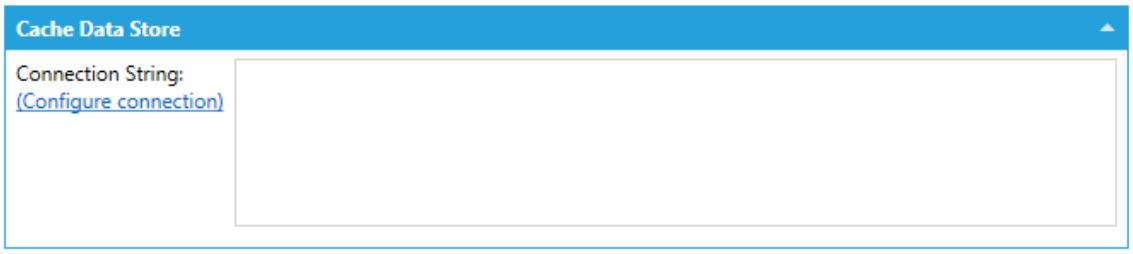


Figure 5.6 - Other Settings: Caching Tab

Use the pull-down menus to select a SQL server and Database name. Should you wish to make sure the connection will work, click on the 'Test Connection' button. Once you have made you selections, click the 'OK' button to return to the 'Other settings' page. Your selections will now appear in the 'Connection String' field.

Caching period – defines how frequent you want data to be cached in the location specified in Cache Data Store location.

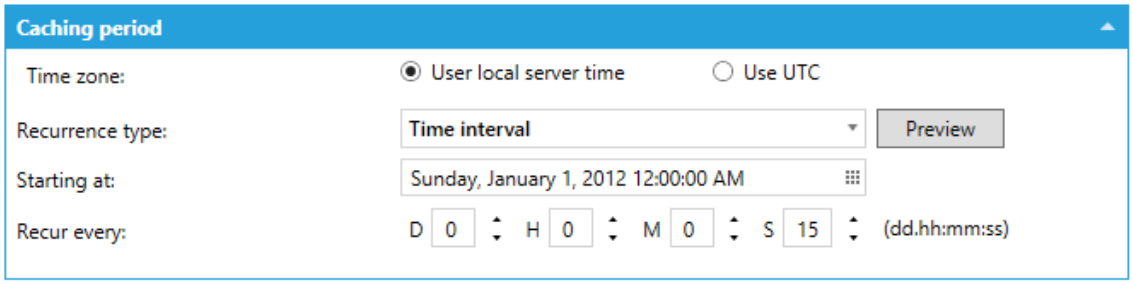


Figure 5.7 – Caching Period

Recurrence Type - Choose from One time only, Time interval or Specific dates and times.

Starting At – This function allows users to decide when caching should first occur (either once or repeatedly at a programmable time). You can enter a day, date and time in the text entry field OR click on the calendar button ( ). Clicking the calendar button opens a calendar selection tool. Select your date and time, then click 'Close'. Your selected day, date and time will appear in the text entry field.

Recur every - This function allows users to decide how often caching should occur. Set the amount of days/minutes/hours/seconds you wish the caching to recur. Highlight a day/minute/hour/second section within the text entry field and use the up and down arrows to change in set increments.

Click 'Apply' to save your settings or 'Apply & Close' to save your settings and return to the main Workbench/AssetWorX provider screen.

5.4.2.1. AssetWorX Server Caching

Benefits of AssetWorX Server Caching are:

- Values of properties can be stored persistently
- Stores values written from the client

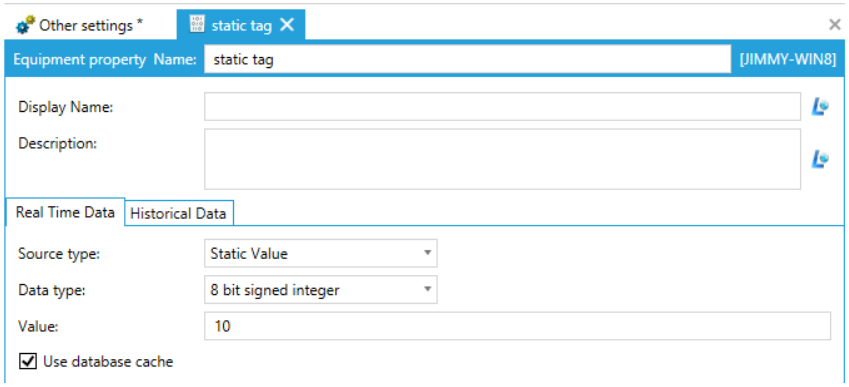


Figure 5.8 - Database Caching Selected within an Equipment Property in AssetWorX

5.5. How to Build the Asset Tree

In AssetWorX, the asset tree defines the physical distribution of equipment and assets for your corporation. In the Equipment node of the configuration database, you build the asset tree to be used by your enterprise for all aspects of reporting and rolling up costs for your business. If you will be installing additional providers, the providers place requirements on the asset tree that you must familiarize yourself with **before** adding equipment to the asset tree in AssetWorX. For example, if Energy AnalytiX will be installed, the asset tree is where you will be reporting and calculating energy consumption, energy-related costs, and carbon emissions. It is also where fault reporting and costing will be generated for FDDWorX.

5.5.1. What Is the Asset Tree?

The asset tree is a structure that represents the physical locations and/or business configuration of your enterprise. It contains buildings, equipment, product lines, and other components that affect costs at your company. Once built, it is a navigational tool for you to use for navigating your corporate structure. Runtime users will have access to the asset tree, and will be able to navigate it for running reports that are available to them through providers that are installed for use in Workbench.

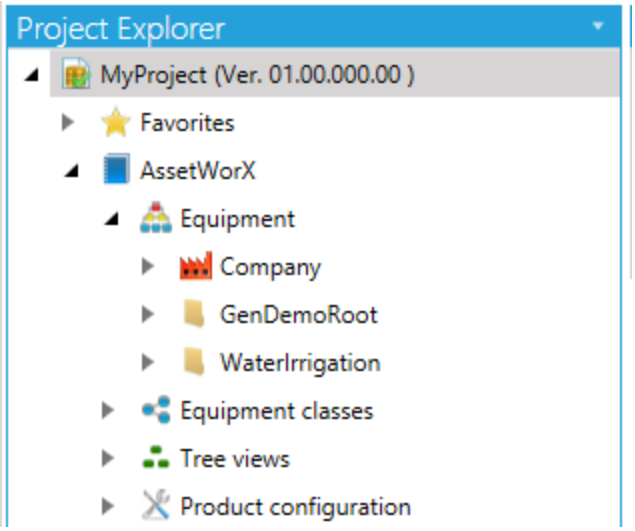


Figure 5.9 - The Asset Tree Resides on the Asset Node

The levels in the asset tree are determined by the equipment hierarchy, and cannot break from the hierarchy's rules. The following illustration shows a sample asset tree (on the left) with the equipment hierarchy that regulates it (on the right).

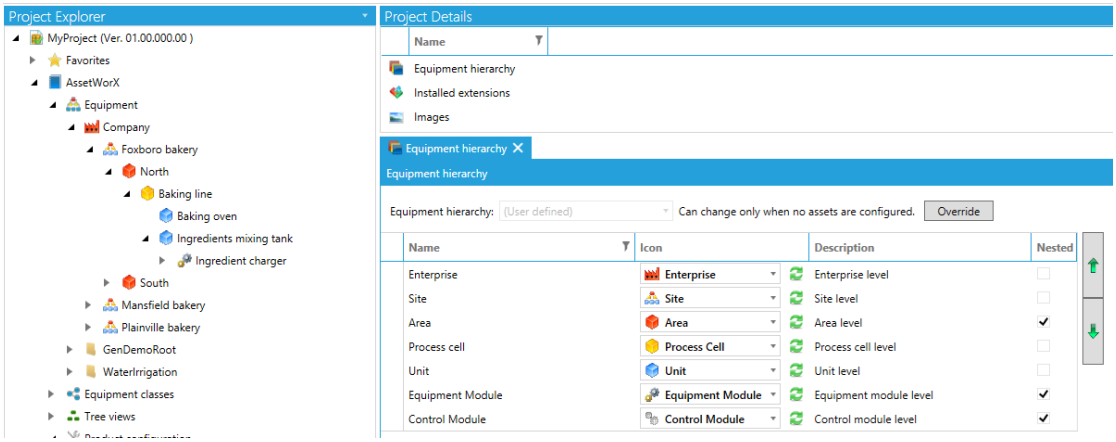


Figure 5.10 - An Asset Tree and its Regulating Equipment Hierarchy

5.5.2. Equipment in the Asset Tree

In order to be able to report on equipment, you must define all of the physical equipment you want to report on. In defining equipment, you need to consider the providers you will be using the asset tree for. If you will be installing the ICONICS Energy AnalytiX and/or FDDWorX providers, then these providers place requirements on the asset tree that you must familiarize yourself with **before** adding equipment to the asset tree. For an understanding of the requirements that any particular provider has, refer to the documentation for that provider before building the asset tree.

As you define equipment, you embed it into a hierarchical reporting structure called an asset tree. Each tier of the asset tree categorizes and defines the equipment for that level. In Workbench, the asset tree appears in the Project Explorer under the **Equipment** node. A default asset tree is installed for you but it is expected that you will override it with your own equipment structure.

AssetWorX displays in a tree-like structure in the Workbench's Project Explorer. An example of an asset tree is shown below. Note that the company has two bakeries (Foxboro bakery and Mansfield bakery), and the Foxboro bakery is divided between the North and South branches. The North branch's baking line is expanded so you can see some of its equipment.

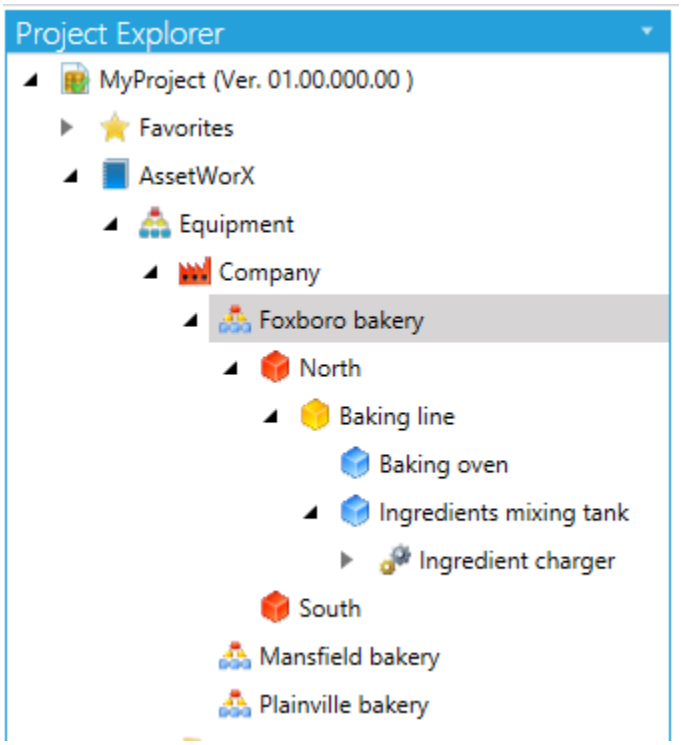


Figure 5.11 - Sample Asset Tree

This is just one example of an asset tree. Obviously, yours will be unique to your company's needs. For that reason, before you begin you must plan out your asset tree structure to make sure you have included all equipment and to make sure it is organized in a way that makes its use intuitive and useful.

Make sure you include all types of equipment you want to report against. Start at the highest level, such as a corporate entity or a geographic region, then break that level out into more detail about equipment that is lower in the asset tree. Continue with each level until you have worked your way down to individual robots, machinery, or other equipment such as meters. Once you have a plan that includes everything you want to report on and that accounts for future plans, too, you can create your asset tree.

Obviously, you have equipment that can be seen physically in front of you, such as robots, production lines, and plants. But you can also create *logical* structures such as business units that group your physical equipment into a hierarchy which you can use to review availability, performance and quality on a broader scale.

5.5.2.1. Equipment Placement in the Asset Tree

If you are using other providers with AssetWorX, those providers may require where in the asset tree some equipment is placed. For example, if Energy AnalytiX is installed, the placement of meters is critical to your ability to analyze energy consumption. It is critical that you pay attention to the requirements to each of the providers that are not installed today, but also the providers you plan to install in the future.

5.5.2.2. Icons in the Asset Tree

Note that equipment in the asset tree has icons. You can specify the icons that represent certain types of equipment tiers, equipment nodes, and equipment classes. Use these icons in a way that lets users easily identify the nodes in the tree. Providing well-planned icons gives your users a way to quickly and easily identify locations, buildings, types of equipment, types of meters, and so on. Make sure to use icons in a way that can provide this type of ease.

5.5.3. Strategies for Building the Asset Tree

Before building your asset tree in AssetWorX, it is important that you put together a strategy. The structure you ultimately create must reflect all of the needs for all of the providers you will be installing for use in the AnalytiX configuration database. You do not want to have to change its structure once it is built. Also, you should be aware of all of the tools available to you for building it so that you can exploit features that will help you. The tools namely are equipment classes and bulk asset configurator. We will go over them in the next chapter.

5.5.3.1. Plan Before You Start!

First of all, make sure you are familiar with what the asset tree is and what it provides for you. This means reviewing all of the documentation for setting it up **before** you actually build the asset tree.

Then, if you will be using Energy AnalytiX, FDDWorX, or other providers, refer to the online help and documentation for those providers before you construct the asset tree.

This is important to do up front because once your asset tree is created you can no longer take advantage of the template capabilities that are available with aliasing.

You can apply uniform information only while building the nodes of the asset tree; after that you will be maintaining each node individually. For that reason, set aside time to plan before you move ahead.

Of particular importance are the following considerations:

- Make sure you understand the Steps to Building an Asset Tree.
- **Some providers' requirements** may prompt you to change the structure of your *asset tree*. For this reason alone, it is important that you consider any future requirements you may have, also.
- Create an asset tree that is **easy to add to and modify**. For example, if you will be adding buildings or changing energy calculations, make sure the asset tree you create will be easy to change.
- Use **icons** in a way that makes the asset tree easy to work with. You can provide your own icons that represent certain types of equipment tiers and equipment nodes. Use these icons in a way that lets your users easily identify the nodes in the tree. Providing well-planned icons gives your users a way to quickly and easily identify locations, buildings, types of equipment, types of meters, and so on. Therefore, use icons in such a way, as to provide that type of ease.
- Come up with a strategy for applying **permissions** for runtime users to access equipment nodes in the asset tree, along with types of commands in each node's runtime menu. You can use the asset tree's structure to exploit the ability for equipment nodes to inherit the permissions set in parent nodes. The strategy for granting permissions is described in the Defining User and Group Permissions for Equipment topic as well as the Permissions Tab for Equipment topic. Permissions in the asset tree use the Security Server users, groups, and settings defined in Security provider in Workbench.
- Familiarize yourself with basic **tools** in AssetWorX, such as "Make Copies Using the Multiply Function", "Export Data", and "Import Data".

5.5.3.2. How to Proceed

Follow these general steps in order to build the asset tree:

1. Review the AssetWorX documentation as well as the documentation of all providers in Workbench, and then plan what the tiers and structure of the asset tree will be. (Establishing the Tiers of the Hierarchy as shown on the following page).

2. Create the equipment hierarchy (Defining Tiers for the Asset Tree as shown on page 5-15).

3. Start adding equipment nodes to the asset tree (Adding Equipment to the Asset Tree as shown on page 5-19).

5.5.4. Establishing the Tiers of the Hierarchy

Why do you need an Equipment Hierarchy with Tiers? When you configure your equipment in AssetWorX, you create an equipment hierarchy that specifies which types of equipment belongs at each tier of the asset tree. Doing this controls the structure of the asset tree by preventing you from accidentally assigning certain types of equipment to inappropriate tiers, and it also controls whether an asset level can have multiple tiers of equipment at each level. Therefore, the equipment hierarchy applies universally throughout the configuration database's asset tree.

When you first install AssetWorX, you will find that there are three different equipment hierarchies provided for you to choose from, each with their own predefined tiers. You can alternatively decide to define a hierarchy of your own choosing. Your options include the following:

- The **Batch Production** equipment hierarchy consists of tiers that structure the equipment typically used in batch processing.
- The **Building Automation** equipment hierarchy consists of tiers for buildings and campuses.
- The **Discrete Manufacturing** equipment hierarchy consists of tiers for the equipment that is frequently found in discrete manufacturing.
- The **User Defined** option lets you choose of the preexisting equipment hierarchies as a starting point, and lets you define the tiers that you need for your installation. You can add any number of levels, and can even define and add iconic images that uniquely identify each tier.

You can use these levels as they are, rename them, delete them, move them, and add more levels. Do what you need to do so that they suit your enterprise's purposes. The objective you want to keep in mind when doing so is that each tier defines a specific type of equipment that occupies that tier.

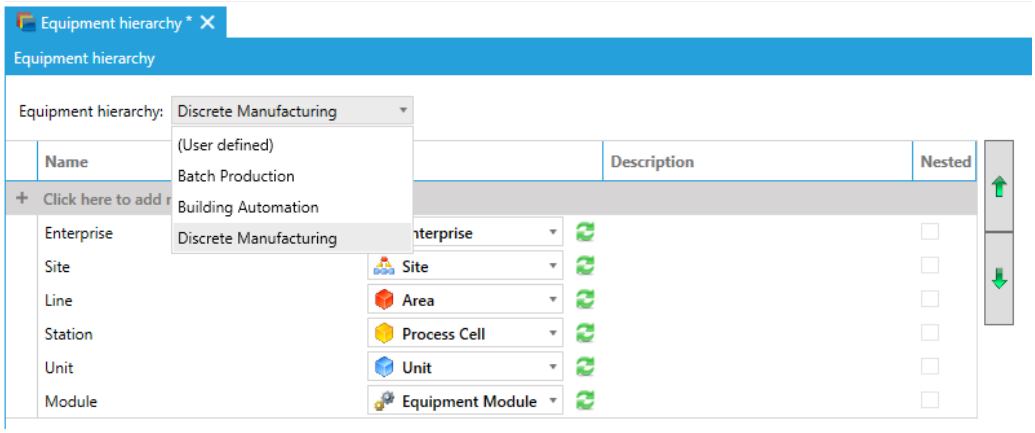


Figure 5.12 - Equipment Hierarchy Options

5.5.4.1. Understanding the Rules of the Equipment Hierarchy

Before you define the tiers of the equipment hierarchy, put together a plan for structuring your company's equipment in a tree-like structure. Then define the tiers of the equipment hierarchy in AssetWorX. You can add images that identify each level.

So that you can do this, it is important that you understand how the hierarchy works.

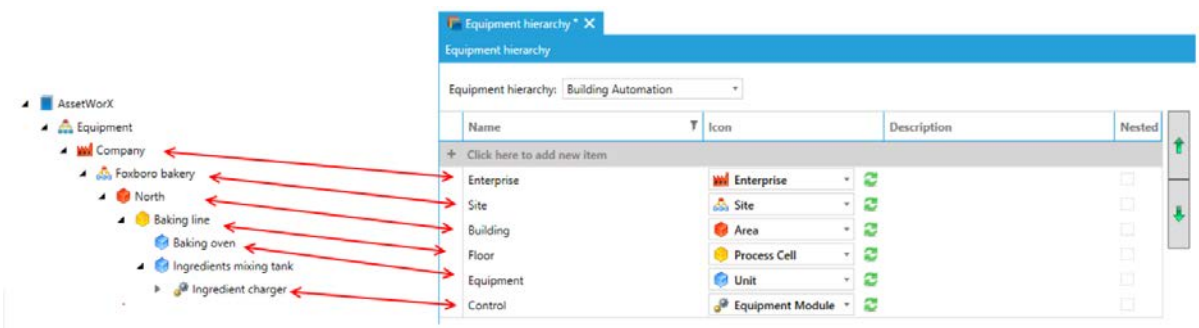


Figure 5.13 - Asset Tree (left) and the Equipment Hierarchy that Controls it (right)

Here are a few rules you need to keep in mind when deciding what you want the tiers of the equipment hierarchy to be:

- **Each tier identifies a type of logical or physical equipment.** Furthermore, each tier is intended to be a summarized version of the tier below it so that, as you work your way down the hierarchy you find that each tier is made up of the equipment at the next lower tier. For example, if the enterprise is made up of sites, each site is made up of areas, each area has process cells, and each process cell has units, and so on. The tiers in this example are Site, Area, Process Cell, Unit, and so on.

- **Equipment that you add to the asset tree cannot skip tiers.** For example, if you are using the tiers listed in the prior bullet item, the Site's child tier always contains equipment definitions for areas, and an area's child tiers are always equipment definitions for process cells. An area cannot have a child tier full of equipment definitions for units.
- **A tier can repeat itself for more than one level if you identify the tier as one that can be *nested*.** This means, for example, that facilities can contain other facilities.
- **The tiers display in the sequence, from broadest to most details, from top to bottom.** Once equipment is defined to a tier in the asset tree, you cannot raise or lower that tier in the equipment hierarchy.

5.5.4.2. Defining the Tiers of the Equipment Hierarchy

Use the Equipment Hierarchy tab to identify asset levels in your asset tree, by following these steps:

1. In AssetWorX, in the Project Explorer navigation tree expand ***localhost*** → **[the name of your AssetWorX database. In our examples, we've named it '*AssetWorX*']** → **Product Configuration**.

2. Click on **Equipment Hierarchy**. This opens the Equipment Hierarchy tab, which looks like this:

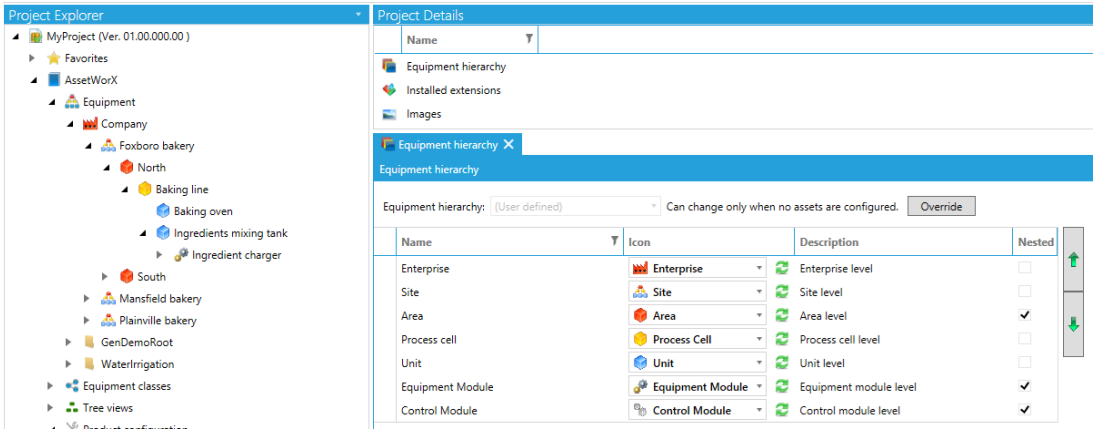


Figure 5.14 -Equipment Hierarchy Defines the Tiers of the Asset Tree

3. If there are assets in the asset tree already, an **Override** button appears on the tab. You cannot change the equipment hierarchy unless you click the Override button.

4. From the **Equipment Hierarchy** drop-down list, select the hierarchy you want to use for this configuration. If you are creating the hierarchy for the first time, choose the hierarchy that is closest to the one you want to use, then follow the steps below to modify it.

5. To add a new tier to the hierarchy, click on the gray row that says **Click here to add new item**. The row becomes enabled so that you can define a new tier. Enter the following:

   a. Enter a **Name** for the new tier.

   b. Choose the **Icon** to represent the tier in the equipment hierarchy. You can use any .JPG or .PNG as a placeholder if you like. For help creating and configuring a new icon, refer to the Configuring icons for the equipment hierarchy section, below in this topic.

   c. Enter a **Description**.

   d. If more than one level of equipment can be created in the asset tree for this tier, put a checkmark in the **Nested** column. If only one level of this tier can be created in the asset tree, leave the checkbox empty.

6. Click **Apply** to save the new tier.

7. Now, with the tier still selected, click the down arrow at the right of the table to move the tier to the position where it belongs in the hierarchy. Click the up and down arrows to position the new tier in the hierarchy. Note that the new tier cannot be inserted above any tier that has equipment in it in the asset tree.

8. Repeat these steps to define additional tiers.

9. To remove a level, select the level and press the **Delete** key on your keyboard.

10. Click **Apply** to save the change, or click **Apply & Close** to also close the tab.

5.5.4.3. Configuring Icons for the Equipment Hierarchy

Tiers in the equipment hierarchy and equipment in the asset tree are represented by small, iconic images that have been configured in the AnalytiX Catalog. If you are adding a new tier to the equipment hierarchy, you must choose an icon in AssetWorX that will represent the tier in the hierarchy. You can use any image as a placeholder until you have finalized the images you will be using.

Uses

Icons in the equipment hierarchy are used by default for each tier in the asset tree. You can, however, use icons for each individual type of equipment which you select at the time you add equipment to the asset tree. For example, you can use the icon of an office building for the Building tier of the hierarchy, but when you add a warehouse, select the warehouse icon instead. If you are using Energy AnalytiX, you can create a separate set of icons that identify meters and utility sources.

You can create icons that represent:

- **Tiers in the equipment hierarchy.** For steps to associate icons with tiers, refer to the Defining tiers for the asset tree topic. Icons that are defined for tiers are used by default for any equipment node that you add to the asset tree without the use of an equipment class.
- **Individual types of equipment in the asset tree.** For steps to associate an icon with any equipment node in the asset tree, refer to the Adding Equipment to the Asset Tree topic. Equipment that you add to the asset tree is given a default icon that you can override for that equipment node.

Strategies

Plan out the icons you intend to use. How you use them and set them up will be determined by your enterprise, the complexity of your asset tree, and the needs at your organization. Keep in mind that the icons in the asset tree will be used not only by the personnel who configures providers in the Workbench, but also by runtime user who are viewing charts and reports. For this reason, you might want to set strategies in place for colors and shapes that you use in the asset tree, to make equipment in the asset tree easy to identify.

AssetWorX comes to you with some default images in place. You can use these images or create your own. If you want to use your own, then you must configure them in AssetWorX by following the steps provided below. The images must be saved to a file in a .JPG or .PNG format.

Steps

To Configure an Icon for Use in a Tier of the Equipment Hierarchy:

1. Open **Workbench**, and drill into the **AssetWorX** provider in the Project Explorer.
2. Expand **AssetWorX → Product Configuration**.
3. Right-click on the **Images** node and select **Edit** or **Edit on New Tab**. This opens the **Images** tab, which looks like this:

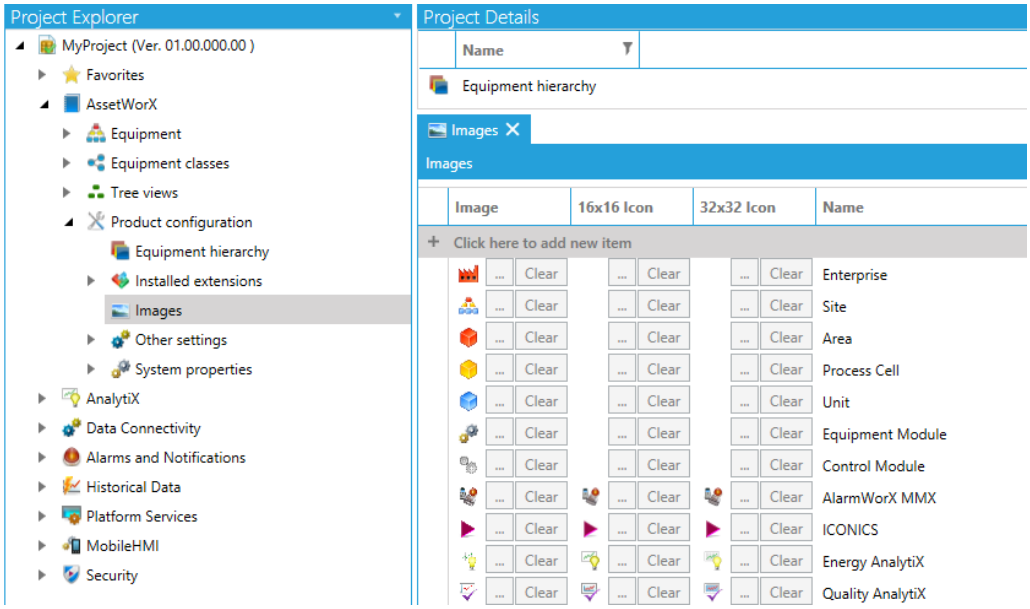


Figure 5.15 - Configure Icon Images for Use in the Equipment Hierarchy

4. Click on the gray row "**+ Click here to add new item**" to add a new image.

5. Click the ellipsis button [...] in the **Image** column to choose the image as it will appear in the asset tree. This opens an Open dialog where you must navigate to the .JPG or .PNG file created for this purpose. Select the file and click the **Open** button; the dialog box closes and the file image appears in the Image column.

6. Optionally, in the **16x16 Icon** column you can choose a 16-pixel by 16-pixel version of the icon. If you leave this column empty, the icon in the Image column is used by default.

7. Optionally, in the **32x32 Icon** column you can choose a 32-pixel by 32-pixel version of the icon that appears on the Equipment tab when you are modifying equipment values. If you leave this column empty, the icon in the Image column displays on the Equipment tab.

8. Enter a **Name** and **Description** for the new tier.

9. Move it to the correct location in the hierarchy by clicking the up and down arrow buttons to the right.

10. Click **Apply** to save the change, or click **Apply & Close** to also close the tab.

5.6. Assets in the Tree

5.6.1. Steps to Building an Asset Tree

In the "Strategies for Building the Asset Tree" topic, we looked at ways in which you can build the asset tree in AssetWorX. It is critical that you pre-plan and formulate a strategy for the creation of the asset tree.

Before you Begin

- Review the documentation of ***all other providers*** that are installed or will be installed.
- Create a strategy for building an asset tree that incorporates all of the other providers.
- Create the tiers of the asset tree as described in the “Establishing the Tiers of the Hierarchy” topic.

Top-Down Strategy

To build the asset tree, you must start at the top-most tier in the equipment hierarchy and work your way down to the bottom-most tier of the equipment hierarchy. The asset tree is always contained in the Assets node in AssetWorX.

5.6.2. Adding Equipment to the Asset Tree

In order to do production, energy, facility, or any other type of analysis on equipment at your enterprise, you must build your equipment into an asset tree. There are a few ways you can do this:

- You can add equipment to the asset tree piece by piece
- You can create equipment classes (which are templates) for individual types of equipment, and then use those equipment classes as a basis for adding equipment
- You can build equipment classes into a tree structure that can be copied to the asset tree and used as a basis for building equipment into your asset tree

The steps provided below describe how to add equipment one node at a time. On the Equipment form there are several tabs, including the Properties tab, Commands tab, Permissions tab, and + tab; each tab is shown below. Alternately, you can use equipment classes as templates to speed things up. In next chapter, we will dig more into the equipment class and its affiliated tools and see how they could maximize productivity.

To Add Equipment to the Asset Tree:

1. Expand **AssetWorX** and then right-click **Equipment**. Select **Add Equipment**. This displays the Equipment form, with the **Properties** tab on top as shown in the following image. For a description of the fields on this tab, refer to the Properties Tab for Equipment topic.

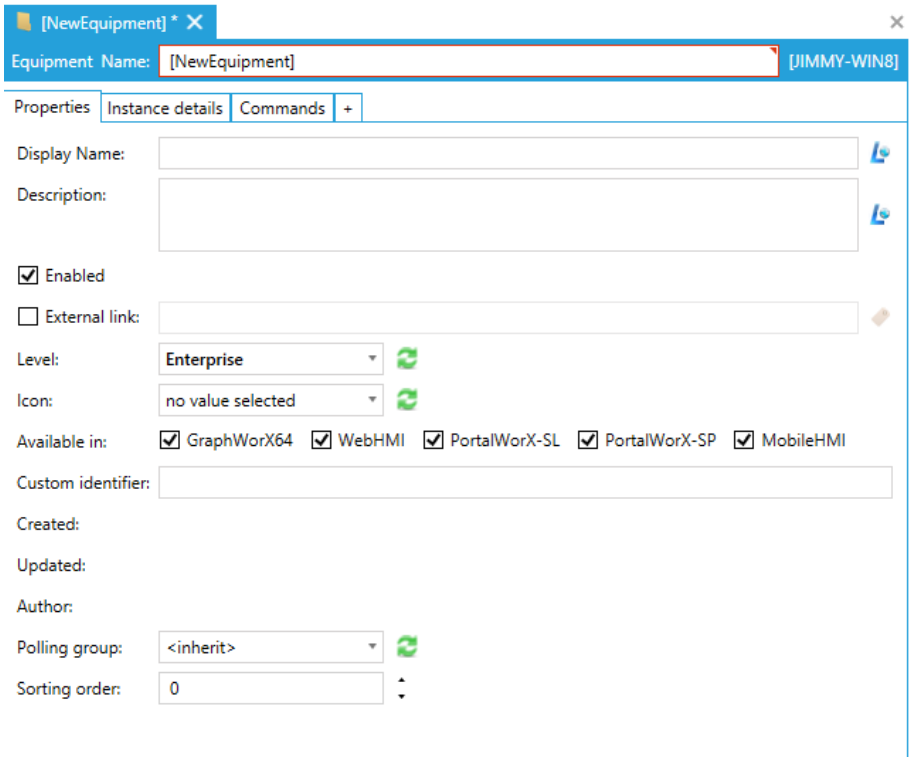


Figure 5.16 - Equipment Form - Showing the Properties Tab

2. In this tab, note your options in the **Level** drop-down list: you can only select the level or levels that are allowed by the rules defined in the equipment hierarchy. So when you define an asset directly under the Asset node, you can select only the top-most tier of your equipment hierarchy; in the example below, the top-most level is Enterprise. At this level, the equipment you are defining is most likely a business entity or location.

Figure 5.17 - Level Drop-down List

3. A tab called **Commands** is also available, as shown below. On this tab, you can set up Commands for Runtime Navigation. For more help using this tab, refer to the "Commands Tab for Equipment" topic.

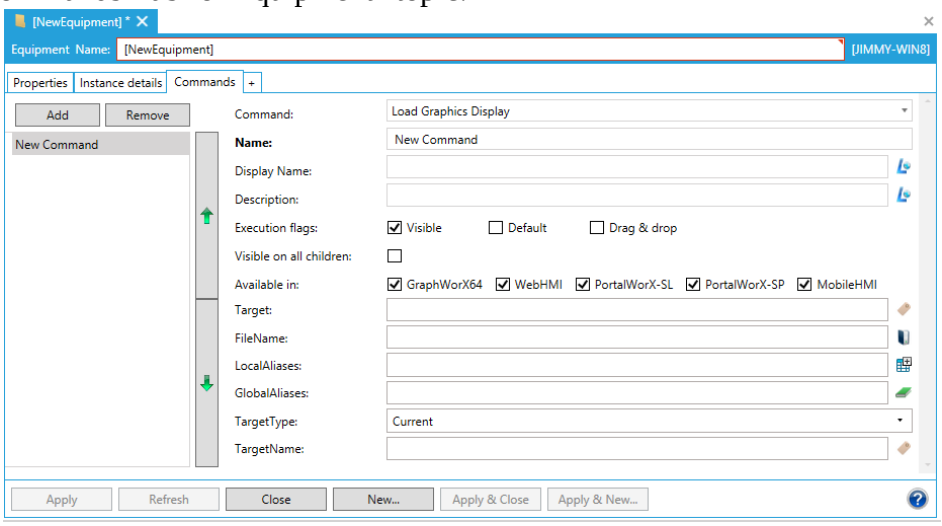


Figure 5.18 - Equipment Form - Showing the Commands Tab

4. When you save the new equipment record, an additional tab, the + tab, appears. Use this tab to add tabs to the form, as shown in the following image. In this example, the asset is an energy asset, so the Add New drop-down list lets the user configure the equipment for use in Energy AnalytiX. If FDDWorX is installed, the user could also configure the equipment for use with it, too. This tab is described in the + Tab for Equipment topic.

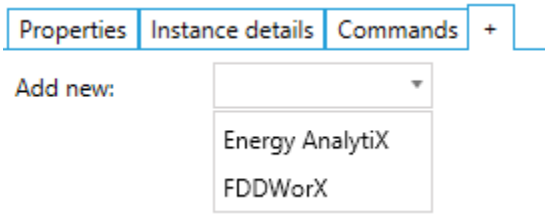


Figure 5.19 - Equipment Form - Showing the + Tab

5. For example, if the user elects to select Energy AnalytiX so that the equipment can be included in analysis in Energy AnalytiX, the tab shown below would be added to the form. For more about application-specific forms, refer to the help for the applications that appear in the + tab's **Add New** drop-down list.

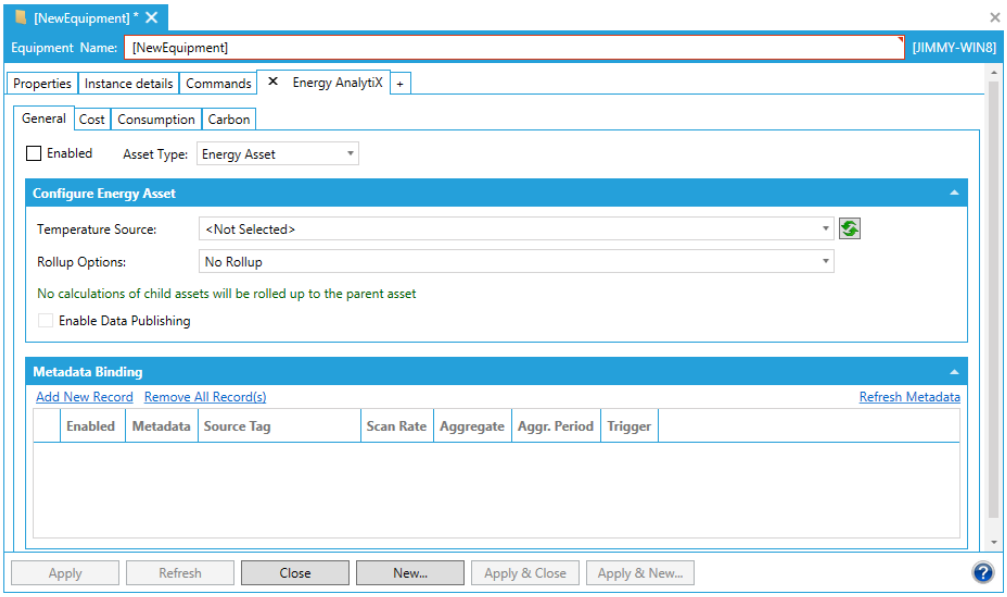


Figure 5.20 - Equipment Form - Showing an Added Tab for Energy AnalytiX

6. After defining the top-most level of your asset tree, now you can begin to define the next level of the asset tree. To do this, in the Project Explorer, right-click the equipment you just added to the Asset node and select Add Equipment again. Once again, this displays the Equipment form, with the **Properties** tab on top, except that now you are one level down in the equipment hierarchy. In the example below, this is the Site level.

Figure 5.21 - Site Level

7. You can continue to build the asset tree in this way, one equipment record at a time, or you can set up equipment classes that you can use a templates for creating similar pieces of equipment quickly and easily. Note that you can even set up structures of equipment classes that for quickly setting up multiple levels of the asset tree in one action.

5.7. Asset Security

GENESIS64 uses a security model that is both granular and additive. 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 based on time settings if you wish. The Security Server, the service that provides user and group policy-based security to the different GENESIS64 applications and features, offers a powerful means for configuring access to your applications. The security model evaluates the membership of users and groups and will deny access to a user when that feature is denied to them regardless of the group's access rights. However, if a user's access isn't specified as 'Denied' or 'Allowed', then the group membership sets the access policy.

The security system contains two components:

- **The Security Server** which manages user logins, and
- **Security system clients** contained within the applications in the GENESIS64 family (e.g., GraphWorX64 , TrendWorX64 , AlarmWorX64, etc.). Any stimulus that causes a change in security status (e.g., a user login or logout) will be immediately posted to the affected clients.

Security settings are configured in the Security Server using Workbench.

5.7.1. Asset Security Privileges for Users and Groups

The Security Server, as shown in the following figure, offers nearly identical security options for user accounts and for groups. You can apply security restrictions at the group level, the user level, or both. The Security Server uses the following rules for determining whether a privilege is extended or denied to a user based on his or her security:

1. If a user lacks a privilege and is added to a group that has that privilege, the privilege is extended to the user.

2. If a user or group has a privilege that is denied, then that privilege is denied to the user even if the user was allowed that privilege at the user account level. **Denials always take precedence**.

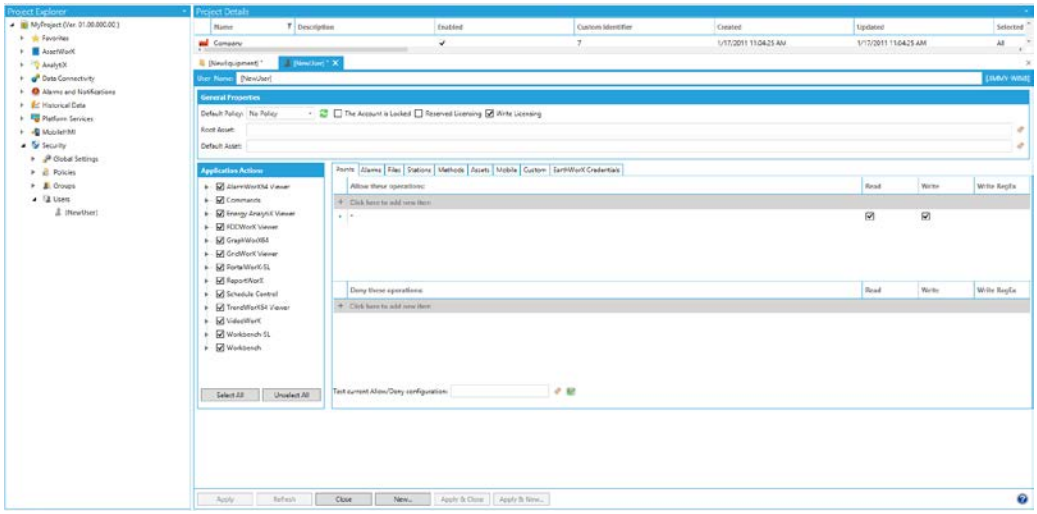


Figure 5.22 - New User Form within the Security Server Accessed via Workbench

The **General Properties** section of the User form (or of the Groups form) allows you to:

- Set a Default Policy (if already set through the Policies node in the same Security Server provider) as well as lock the account.
- Set a Root Asset (where the set security is carried through all its contained assets)
- Set a Default Asset.

Clicking the ellipsis [...] button opens the data browser, shown below, where you are able to select your specific asset.

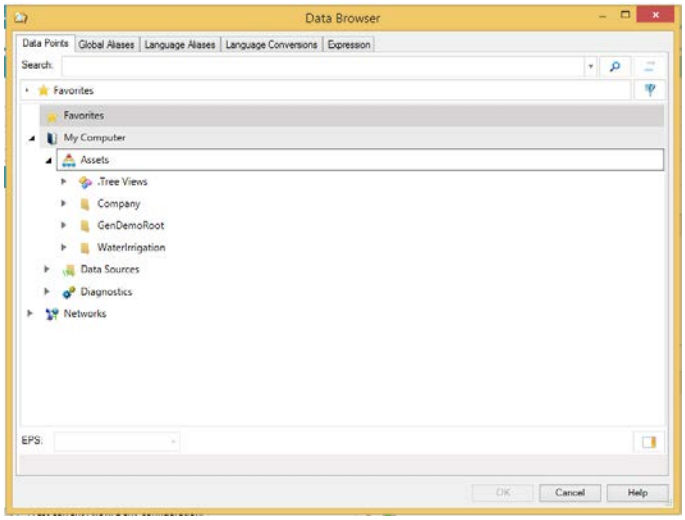


Figure 5.23 - Using the Data Browser to Select Either a Root or Default Asset for Security Privileges

Once you define a group or user account, you define its privileges. The **Applications Actions** section of the User (or Group) security settings provides the ability to Allow or Deny the use of certain features within GENESIS64 components.

Most relevant to this training is the 'Commands' section, shown below. All components are allowed by default, with a checkmark signifying the User (or Group) is currently able to use that feature. Removing the checkmark and saving the User's (or Group's) security setting means that feature will be restricted.

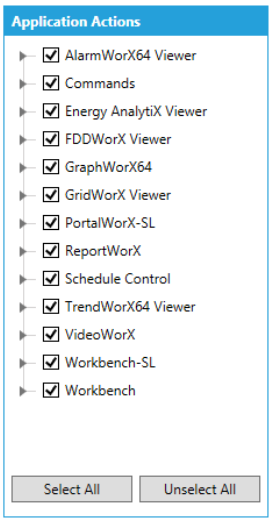


Figure 5.24 - Application Actions Section of the User Security Settings Form

Further security privileges (for a User or Group) can be defined by using the several listed tabs (Points, Alarms, Files, Stations, Methods, Assets and Custom). Note that the Application Actions section of the User or Group properties remains visible, regardless of which of these tabs is selected. This training will cover the 'Assets' tab. (More information on the Security Server, and the other tabs in the User or Group security settings, is available in other training sessions and materials.)

The Assets tab allows you to select specific assets (using the ellipsis button [...] to access the Data Browser) and allow visualization and/or execution operations for that asset or deny them, as shown in the following figure.

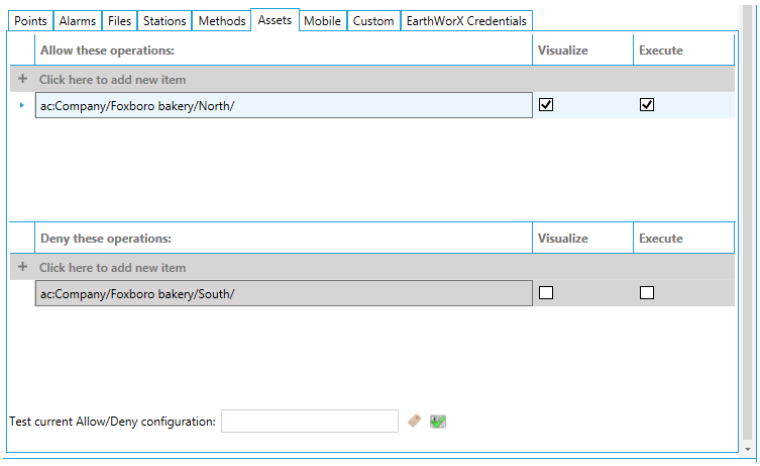


Figure 5.27 - Allowing/Denying Asset Operations for a User in the Security Server Provider in Workbench

In the previous figure, this specific user would be allowed to both visualize and execute operations for the /Foxboro bakery/North asset, but would be denied visualization or execution for the /Foxboro bakery/South asset.

Lab: Building the Asset Tree L5

5.1. Lab Overview

With Lab 1 complete, your training configuration is intact. With this lab, we will modify the **Assets** tree that corresponds to the **icTonics Project**.

Estimated time to complete this lab: **30 minutes**

5.1.1. Task Checklist

In this lab, you will complete the following tasks:

- Build an asset tree in Workbench that correlates to the icTonics theme.
- Implement expressions and static equipment properties.

Note: You should have completed Lab 1 and Lab 2 to configure the Modbus OPC server.

5.1.2. Support Files

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

5.2. Preparing the icTonics Asset Tree

Having completed Lab 2, the prerequisites have been installed and our systems are set up to accommodate **Workbench**. Let's get started!

5.2.1. Building an Asset Tree

At any enterprise, equipment is interconnected with its physical location and in business units that are easily organized hierarchically. You can define the interconnected relationships of equipment using a tree-like structure called the **Asset Tree**.

Now that we have the **icTonicsUnifiedDB** training database created, we will need to add our assets. Through the execution of the following steps, we will create an asset tree similar to the structure defined in the figure below.

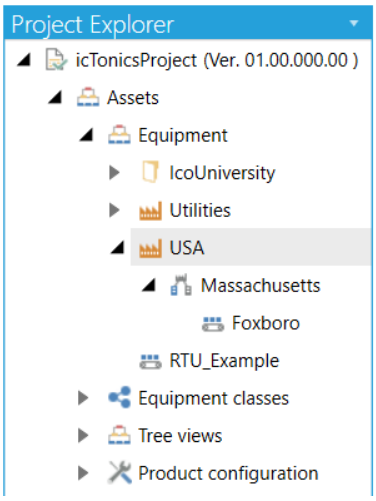


Figure L5.1 – Training Class Asset Structure

1. From within **Workbench**, expand the **Assets** provider.

2. Right-click on the **Equipment** node. Select **Add Equipment.**

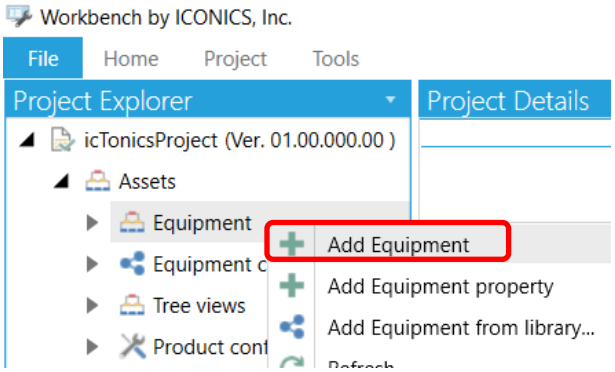


Figure L5.2– Adding the First Asset

3. Notice that this action will open the **NewEquipment**, **Properties** sub tab in the **Project Details** window. From the **[New Equipment]** tab on the right, with the **Properties** tab selected, enter the following entries into the appropriate fields:

   - **Name:** USA
   - **Level:** no value selected (select "Clear Value" from drop-down list)
   - **Icon:** Enterprise
   - **Description:** The USA Asset

Although each asset also has a **Commands** subtab, the configuration of the assets in this section will not require Command changes. However, we will be using commands later during this training.

4. Click **"Apply & Close"** after completing the property entries for each asset.

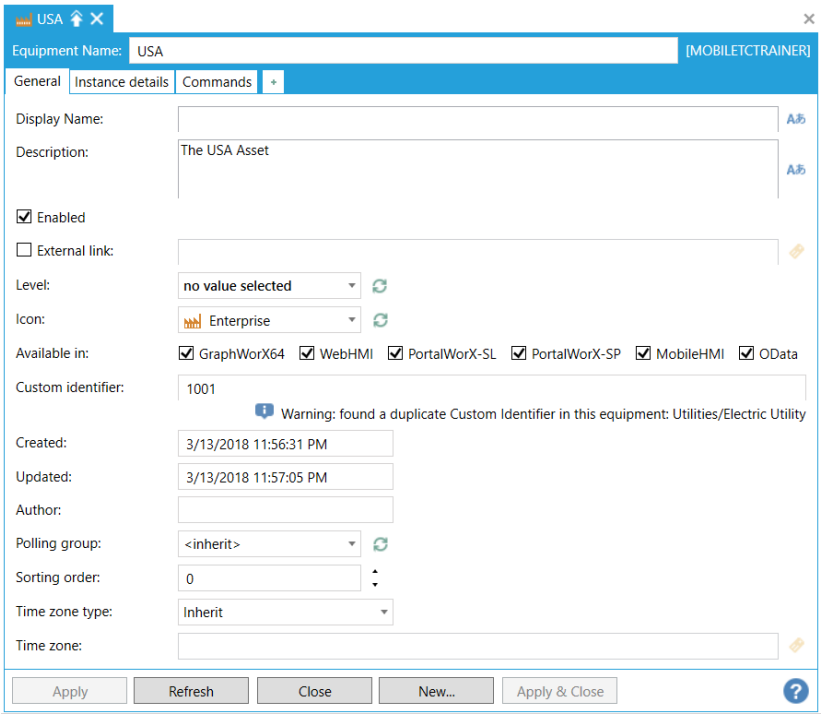


Figure L5.3 – Sample Asset Properties Dialog

5. At the **Equipment** level, add a new equipment called **RTU_EXAMPLE**:

   - **Name:** RTU_Example
   - **Level:** no value selected
   - **Icon:** Process Cell
   - **Description:** Roof Top Unit1

6. Proceed by **creating each new asset** with a right-click and select **Add Equipment** until you have completed the asset tree as depicted in the figure below. Remember to click **Apply & Close** to commit the property changes.

   Note: Reference the previous figure to check that your asset tree is being built as described. Be sure to add the **RTU_EXAMPLE** asset under the **Equipment** node (not under the Foxboro node).

7. Continue by adding the following assets. Click **Apply** after adding **Each** remaining asset has been added.

8. Starting with a right-click on the '**USA**' asset and select **Add Equipment**. In the **[New Equipment]** tab on the right and, with the **Properties** tab selected, enter the following entries into the appropriate fields:

   - **Name:** Massachusetts
   - **Icon:** Site
   - **Description:** Massachusetts Asset

9. Right-click on 'Massachusetts'. From the **[New Equipment]** tab on the right, and with the **Properties** tab selected, enter the following entries into the appropriate fields:

   - **Name:** Foxboro
   - **Icon:** Area
   - **Description:** Foxboro Asset

   After completing this section your assets tree should look like the figure below.

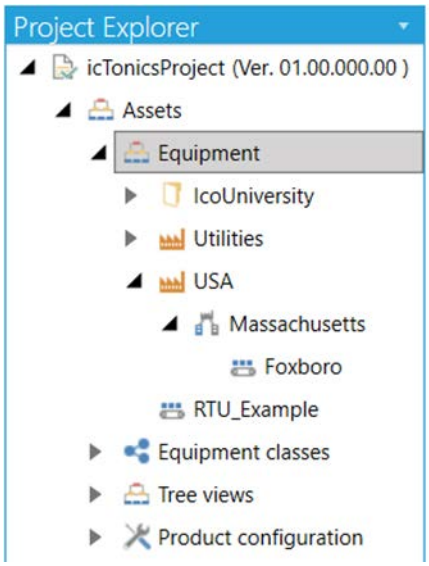


Figure L5.4 – Asset Tree Build Out Continued

5.2.2. Adding Equipment Properties

Equipment Properties allow you to connect the associated asset to real-time (utilizing multiple data source options) or historical data sources. **Note**: The concept of Equipment Properties will be covered in more detail in Chapter 4.

1. Right-click on the **RTU_Example** asset and select **Add Equipment property** with the following criteria:

Name:	CLG_STG1_ENA
Description:	Cooling Stage1 Enabled
Source Type:	Static Value
Use Database Cache	Checked
Data Type:	Boolean value
Value:	False

2. Click **Apply**.

3. In the Asset tree on the left side, right-click on the **CLG_STG1_ENA** equipment property and select **Multiply**.

4. In the **Multiply Options** dialog, enter the following criteria:

   Note: The checkbox allows (checked) or denies (unchecked) multiplying the various properties

   - Make sure the **Description** and **Name** checkboxes are **Checked**.
   - **Uncheck** the **Real time value** checkbox.

Number of Copies	3
Description	Cooling Stage{0} Enabled
Name	CLG_STG{0}_ENA
RealtimeValue	False
First Number	2
Numeric Places	1

5. The {0} is a placeholder that will be replaced by the number generated during the multiply process.

6. The completed dialog should resemble the figure below.

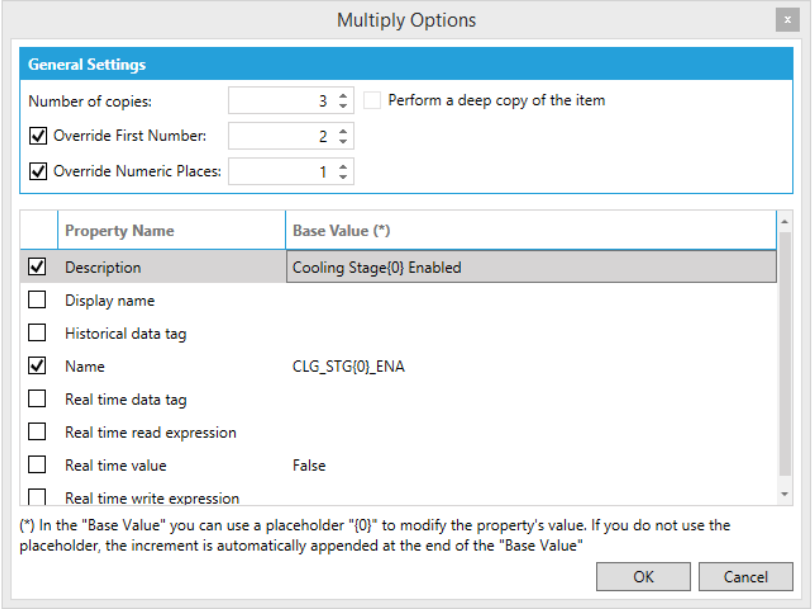


Figure L5.5 – Completed Multiply Options Dialog

7. Click **OK**.

8. Add a fifth equipment property with the following criteria.

Name:	CLG_STG_ACT_STS
Description:	Cooling Stage Active Status
Source Type:	Expression
Use Database Cache	Checked
Data Type:	32 bit signed integer
Read Expression:	x={{ac:RTU_EXAMPLE/CLG_STG1_ENA}}+ {{ac:RTU_EXAMPLE/CLG_STG2_ENA}}+ {{ac:RTU_EXAMPLE/CLG_STG3_ENA}}+ {{ac:RTU_EXAMPLE/CLG_STG4_ENA}}

9. You can enter this **{{ac:RTU_EXAMPLE/CLG_STG1_ENA}}+** and then copy, paste and edit it three times to help expedite the creation of the complete expression.

10. With the Read Expression complete, click on the **X=** button to launch the **Expression Editor**.

11. Click on the button that has a **Red Checkmark**. This executes a syntax check. If you see **No errors**, click OK. Otherwise, correct any reported issues and click OK.

12. We will be working more in depth with the Expression Editor later in this training.

13. Hit Apply to save the equipment property.

14. To review the result of the expression, you can use the **Data Explorer**.

15. Go to Start→All Programs→ICONICS Tools→Data Explorer (or just search for Data Explorer if you use Server 2012)

16. In Data Explorer, browse to My Computer→Assets and click on RTU_EXAMPLE

17. You will see a Grid Panel shown as below:

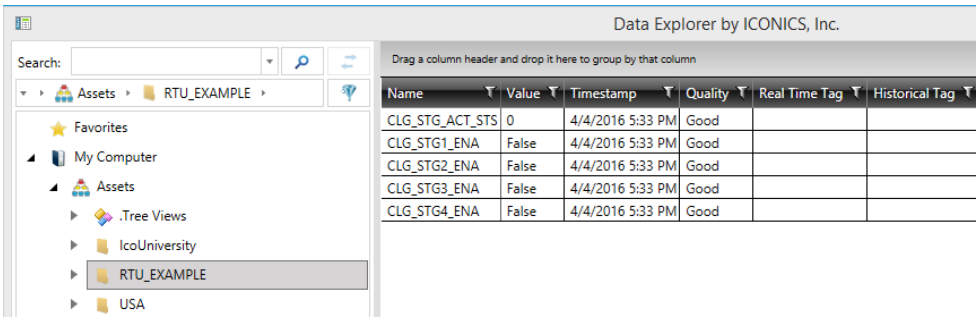


Figure L5.6 – Monitoring Equipment Properties

18. Double click on any of the tags ending in **_ENA** and you will be able to overwrite the default value.

19. Change False to **True** and click **OK**.

20. Since the expression is written to accumulate the number of ENA tags that are True, you can see the **CLG_STG_ACT_STS** tag **increment** each time you write True to an ENA tag.

21. You could have also used GraphWorX64 Runtime as a test if you were to add one process point (connected to **CLG_STG_ACT_STS**) and four Data Entry objects to the **_ENA** tags.

22. We will add Data Entry fields in the following labs.

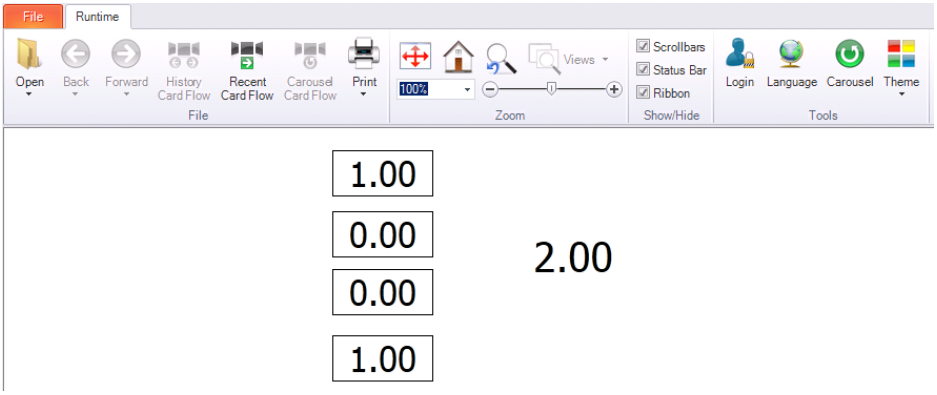


Figure L5.7 – RTU_Example Test Display in Runtime

This concludes this lab.

Equipment Classes, Aliasing and Bulk Instantiation 6

It is strongly recommended that you create equipment classes for use as templates when adding equipment nodes to the asset tree in AssetWorX. An equipment class looks a lot like an equipment node, except an equipment class is stored on the Equipment Classes node in AssetWorX, not in the asset tree itself. An equipment class is really just a template for creating an equipment node. As such, its properties are only anticipated values or default values or aliases. Using aliases in an equipment class lets you establish placeholders for properties and easily define these properties after the equipment node is added to the asset tree.

6.1. Objectives

After completing this chapter, you should know:

- How to Create Equipment Classes for Use as Templates
- How to Create a Multi-Tiered Structure of Equipment Classes
- How to Use Equipment Properties
- How to Use Aliases in an Equipment Class
- How to Use an Equipment Class to Add Equipment to the Asset Tree
- How to Use Bulk Asset Configurator to Massively Instantiate Equipment Classes

6.2. Creating Equipment Classes for Use as Templates

You can build equipment classes into multi-tiered structures so that they appear much like they would in the asset tree. Some equipment class configurations may require you to build a multi-tiered structure. For example, if you have installed the Energy AnalytiX provider, before you can configure a raw consumption calculation for an equipment class to be used for adding energy assets to the asset tree, you must create a second, "child" equipment class for a meter. Then you can use the multi-tiered equipment class as the basis for defining machines that have a meter.

Below are some possible uses you may have for creating single and multi-tiered equipment classes. You can create an equipment class to be used for defining the following:

- Buildings (the top tier) with many floors (as child tiers) in the asset tree.
- Machinery (the top tier) on a plant floor and the meters that monitor it (child tiers).
- A plant (the top tier) with workstations (child tiers).
- Campuses (one tier); or campuses with buildings (two tiers); or campuses with buildings and meters (three tiers).

By creating these structures in advance, you can preload all of their equipment information that they have in common, such as the right-click commands to be used for all sites. You can also provide aliases for equipment information, allowing you a way to easily define uniform information to be used for your equipment nodes.

The remainder of this topic describes how to work with equipment classes and use them as templates.

- Create an Equipment Class
- Create a Multi-Tiered Structure of Equipment Classes
- Use Aliases in an Equipment Class
- Use an Equipment Class to Add an Equipment Node to the Asset Tree
- Handle Instantiation Errors

6.3. Creating an Equipment Class

To Create an Equipment Class to be used as a Template:

1. In Workbench, expand AssetWorX → Equipment Classes.

2. Right-click Equipment Classes or right-click the equipment class folder you want to add the equipment class to. Then select Add Equipment Class from the pop-up menu. This opens the Equipment Class form with the Properties tab on top and the Commands tab underneath it.

3. In the Name field, type the name of the new equipment class. Use the name field to identify the type of equipment node that this equipment class is designed to create. Note that whatever you type here will be used as the name of any equipment that is created from this equipment class. For example, if you type "Template for Multi-Story Building" then every equipment node that you create from this equipment class will be given that same name. You will have to modify the name for each equipment node that you create.

Alternatively, you can enter an alias, instead. For example, if you type /?Multi Story Building NAME?/, when you create an equipment node, you will be prompted to replace the name as the equipment node is created. For more details, refer to the "Using Aliases in an Equipment Class" section, later on in this document.

4. Click **Apply**. This saves the equipment class in the Equipment Classes node and enables the + tab, which we will get to later.

5. Complete the fields on the **Properties** tab. These fields are described in the "Properties Tab for Equipment Classes" topic.

6. Go to the **Commands** tab to define any standard right-click menu items you want provided during runtime for equipment nodes created from this equipment class. For detailed information about how to use this tab to create runtime menus, refer to the "Setting up Commands for Runtime Navigation", as well as "Commands Tab for Equipment Classes" later in this manual.

7. If you have installed **other providers** that will be using the asset tree, or if you will be installing other providers that will use the asset tree, **now is the time to define information that those providers will be using**. Their information will be added as you add new equipment nodes to the asset tree. You cannot use equipment classes to add information once an equipment node has already been added.

8. If any provider that you will be using has not yet been installed, take the time now to install the provider so you can define information to be used by the provider. For more information, refer to "Installing Available Solutions" in the previous chapter.

9. If all of the providers you will be using are installed, click on the **+ tab**. On this tab you will see a drop-down list of all of the providers that are installed.

10. From the **Add New** list, select the provider whose information you want to configure for the equipment class. This adds a tab for the selected provider. For example, if you select FDDWorX from the drop-down list, then a tab called FDDWorX is added to the Equipment Class form.

11. Go to the provider's tab and define the information. For help, refer to the provider's help for that tab. For example, use the help for the FDDWorX tab or the Energy AnalytiX tab.

12. Return to the **+ tab** as many times as you need to configure information for additional providers.

13. Click **Apply** to save the equipment class.

14. At this point you can create additional equipment classes and add create a multi-tiered structure if you want to. For more information, refer to "Creating a Multi-Tiered Structure of Equipment Classes" below.

15. When you are done you can use the equipment class to add equipment nodes to your asset tree. For how to do this, refer to the "Using an Equipment Class to Add an Equipment Node to the Asset Tree" section later in this document.

6.4. Creating a Multi-Tiered Structure of Equipment Classes

Now that you know how to create an equipment class (as described above), let's take a look at how you can create a multi-tiered structure of equipment classes. As we work with multi-tiered equipment classes, keep in mind that a multi-tiered structure has a parent at the top tier with children under it in the next tier down. If there is a third tier, then the children have children too.

A multi-tiered structure equipment class has one parent at the top, but the parent can have as many child equipment classes as is necessary, and as many tiers as needed. You will want to take time to plan out your multi-tiered structures so that they are detailed enough to be of good use to you, but not so elaborate that they become cumbersome. You may need to experiment with them before putting them to use.

Example

The following illustration shows a multi-tiered structure of equipment classes for creating a building. The Multi-Floor Building equipment class is the parent, and the Floor Template equipment class is its child. The Electric Meter is the child of the floor template. Keep in mind that a child equipment class can also be used independently of its parent. Following the example shown in the image on the next page, you can use the Multi-Floor Building equipment class to create a building with one floor and one meter, and then use the Floor Template equipment class to add additional floors and meters, as shown in the sample asset tree (on the right) for the Atlanta equipment node.

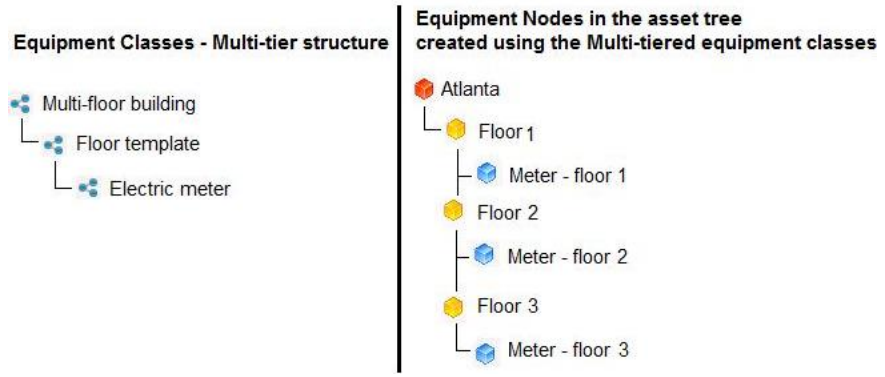


Figure 6.1 - Using a Multi-Tiered Equipment Class for Adding Equipment to the Asset Tree

6.4.1. Steps for Creating Multiple Tiers

The following steps assume that you are familiar with how to define an equipment class, described above in the "Creating an Equipment Class" section.

To Create a Multi-tiered Structure of Equipment Classes:

1. Create the ***parent*** equipment class first. The parent must be created in either the **Equipment Classes** node or the equipment class **folder** you want to add the parent to.

2. Add a new ***child*** equipment class by right-clicking the parent equipment class and selecting **Add Equipment Class** from the pop-up menu. This adds a new equipment class to the parent. Save the child equipment class.

3. At this point, you can add more child equipment classes to the parent, or can add a third tier by adding a new child equipment class to the child defined in step 2.

4. When you have finalized the structure that you want, you can return to each equipment class and refine its definitions so that as much information as possible is included in the equipment class before it is used as a template. For example, you can do the following:

   - You can use aliases as described in the following "Using Aliases in an Equipment Class" section.
   - Define any standard right-click menu items you want provided during runtime for equipment nodes created from this equipment class. For detailed information, refer to "Setting up Commands for Runtime Navigation", as well as "Commands Tab for Equipment Classes".

- If other providers will be using the asset tree, or if you will be installing other providers that will use the asset tree, **now is the time to define information that those providers will be using**. If any provider that you will be using has not yet been installed, take the time now to install the provider so you can define information to be used by the provider. For more information, refer to "Installing Available Solutions".

5. When you are done you can use the equipment class to add equipment nodes to your asset tree. For how to do this, refer to the "Using an Equipment Class to Add an Equipment Node to the Asset Tree" section later in this topic.

6.5. Using Aliases in an Equipment Class

When creating an equipment class, you must populate its fields with information that defines the type of equipment. For example, an equipment class for creating buildings might contain all of the pop-up menu commands to be associated with buildings. However, not all buildings share the same information. For that reason, after using the equipment class to add a building to the asset tree, you will need to visit every field that needs to be defined uniquely for the building. You do not, however, want important information to go undefined for the building. To prevent this from happening, and to easily manage the information that will need to be defined, you can use aliases in you equipment classes.

An alias is simply a placeholder for information that will need to be defined for any equipment node that is defined using the equipment class. Instead of entering values in every equipment class field, you can enter an alias surrounded by **/?** and **?/**, like this: **/?** ***Alias*** **?/**. The alias can be any text string that you want. The best practice is to describe the information to be replaced with later on, for example, /?BuildingName?/ or /?DescriptionOfMeter?/.

Then, when you use the equipment class to create an equipment node (as described in the following section, "Using an Equipment Class to Add an Equipment Node to the Asset Tree"), you will be prompted to replace each alias with the information needed for defining the equipment node. If you are unable to provide the information at that time, the aliases will remain on the equipment node until information can be provided. When information becomes available, you can easily enter it by instantiating (or replacing) the aliases, as described in "Instantiating an Equipment Node's Aliases".

It is important that you realize that not all equipment class fields can be aliased. If you are not sure, you can enter an alias to see if one is allowed, or you can refer to the online help topics for the equipment class tabs in AssetWorX.

6.6. How to Use Aliasing

The following tips and reminders will help you to use aliases:

- You can enter aliases only on equipment classes, as described above in "Creating an Equipment Class".
- To enter an alias in a field, type the alias surrounded by the characters **/? ?/** as in the following format: **/?*alias*?/**
- The alias is a string that can contain any alphanumeric characters as well as spaces and punctuation characters.
- You can use aliases as the names of equipment classes such as, **/?electric meter name?/** or **/?warehouse name?/**. As a result, the alias will appear as the name of the equipment class in the Equipment Class tree structure, as shown below. When an equipment node is created using the equipment class, the alias will be used as the node name until the alias is instantiated.

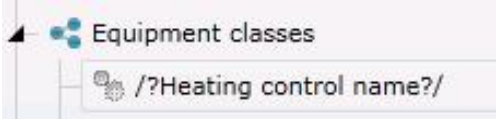


Figure 6.2 - Alias as Equipment Node Name

You can use an alias inside a string such as a URL or a description, as in the examples highlighted in red in the Description field in the following illustration:

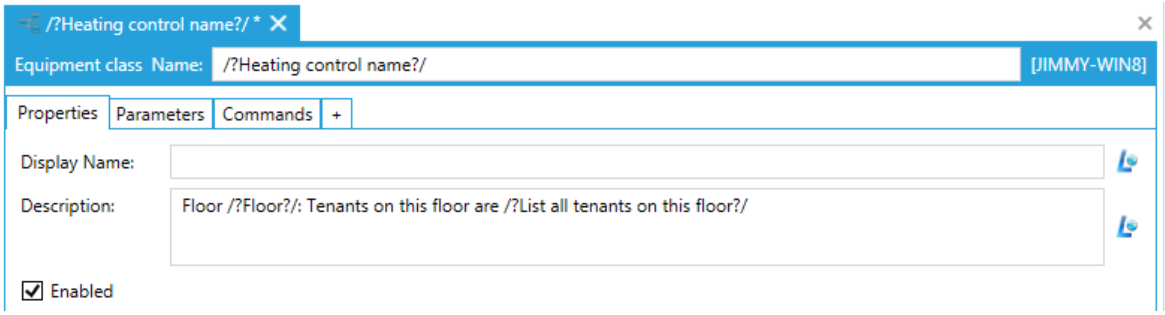


Figure 6.3 - Alias Used Inside String

Other examples:

- http://www.**/?office website?/**.com
- **/?Product-Line?/** manufacturing in **/?City, State of building?/**

6.7. Adding Equipment Properties

For each level of Assets within the tree control of the AssetWorX provider in Workbench SL, you can add Equipment Properties. These allow you to connect the associated asset to real-time (utilizing multiple data source options) or historical data sources.

To Add an Equipment Property to an Existing Equipment Asset in AssetWorX

1. First, you must have already created an asset. You could assign an equipment property to the top-level 'Assets' node if you wish, but in our example, we are connecting to a sub-branch in the tree, an equipment asset we've named 'Test Asset'.

   You can add the property by either right-clicking on the equipment in the asset tree and selecting 'Add Equipment property' OR you can click on the item and then, once the contextual ribbon changes to the items relevant to your item selection), click on the 'Add Equipment property' button in the Home ribbon at the top of the screen. Both methods are highlighted in the following image.

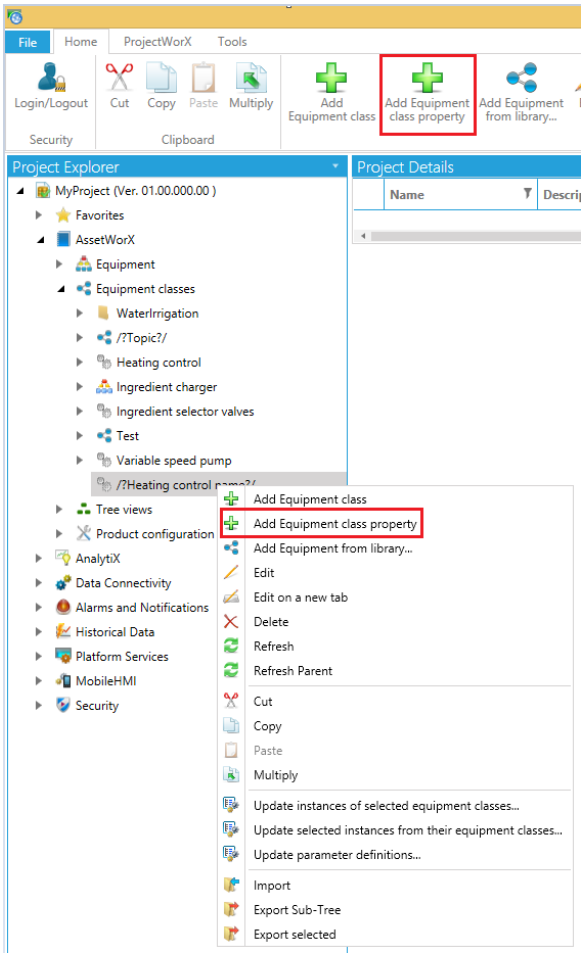


Figure 6.4 - 'Add Equipment Property' in AssetWorX

2. This will open a [NewEquipmentproperty] tab in the right-side window of Workbench SL, as well as a placeholder in the top right items list. This items list shows any sub-items connected to an asset, as well as any of its connected properties. Give your property a name and description and then decide whether your property will need to connect to Real Time Data or Historical Data.

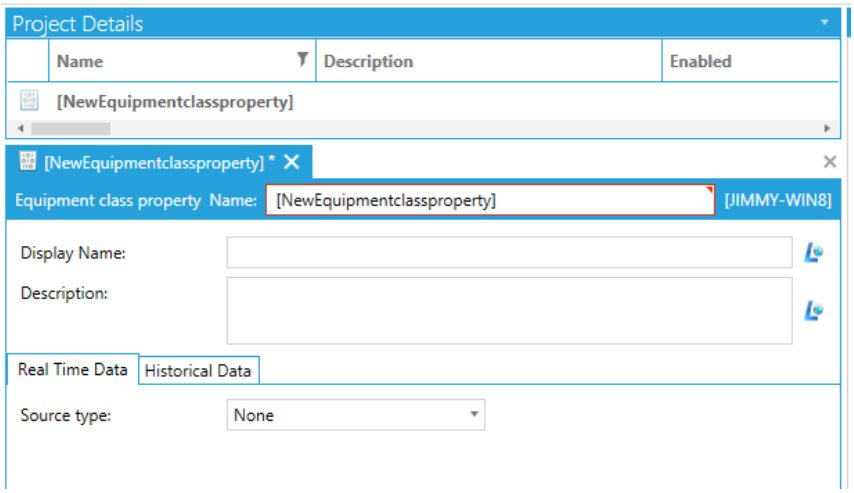


Figure 6.5 - Equipment Property Settings in Workbench SL

Next, see the following options for selecting either Real Time or Historical Data.

6.7.1. Real Time Data Tab

- **Source Type: None**

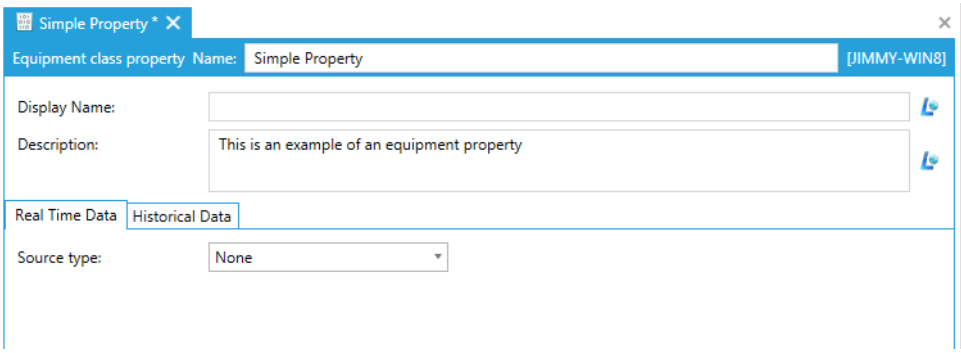


Figure 6.6 - Real Time Data - Source Type: None

- **Source Type: Static Value**

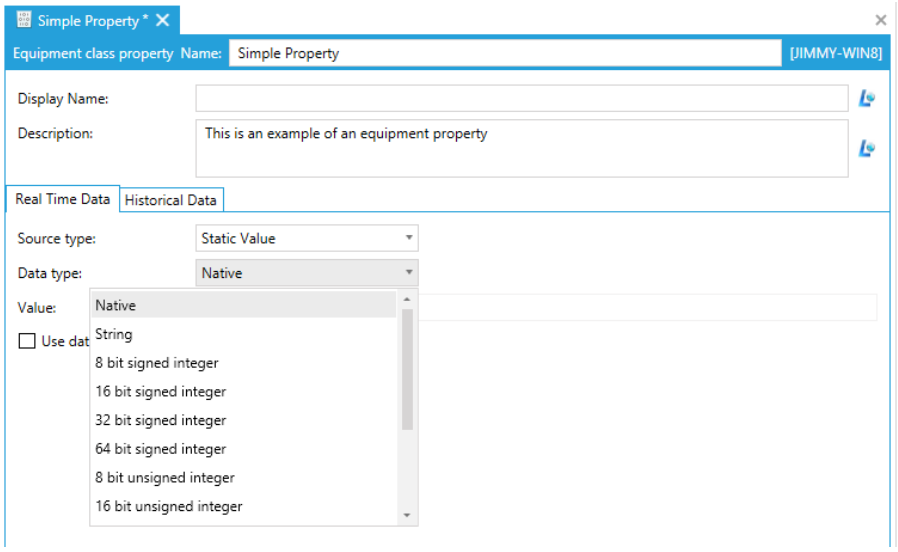


Figure 6.7 - Real Time Data - Source Type: Static Value

In this instance, once you select 'Static Value' from the pull-down menu, you'll see options to select a 'Data Type' and, depending on that type, be able to enter a Value (in the following text entry field). You can also decide to use any existing data cached in the database by clicking on the 'Use Database Cache' checkbox.

Data Types include:

- Native - will prevent you from entering anything in the Value text field.
- String
- 8 bit signed integer
- 16 bit signed integer
- 32 bit signed integer
- 64 bit signed integer
- 8 bit unsigned integer
- 16 bit unsigned integer
- 32 bit unsigned integer
- 64 bit unsigned integer
- 32 bit floating point number (single precision)
- 64 bit floating point number (double precision)
- Boolean value

- **Source Type: Dynamic Tag**

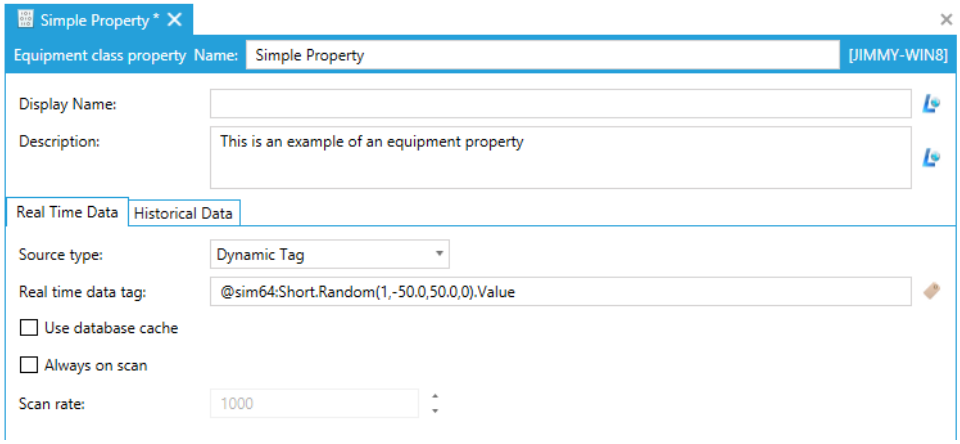


Figure 6.8 - Real Time Data - Source Type: Dynamic Tag

In this instance, once you select 'Dynamic Tag' from the pull-down menu, you'll be able to enter a **Real time data tag**. If you don't know the exact location, you can click on the tag button to bring up the Data Browser, where you will be able to select from all connected data sources. In the example above, we are using a simulated data value source. You can also decide to use any existing data cached in the database by clicking on the 'Use Database Cache' checkbox. The **Always on scan** checkbox lets you choose the rate at which the tag should be scanned (and can then be entered in the **Scan rate** text entry field).

- **Source Type: Polled Dynamic Tag**

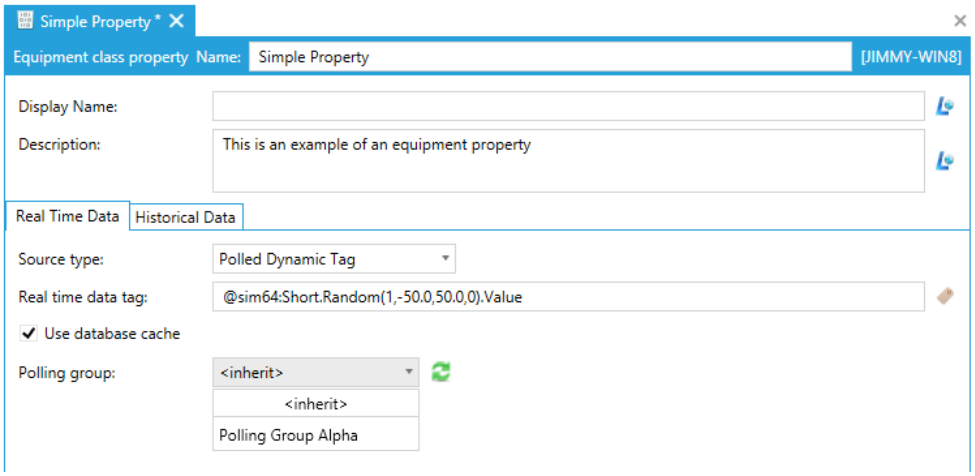


Figure 6.9 - Real Time Data - Source Type: Polled Dynamic Tag

In this instance, once you select 'Polled Dynamic Tag' from the pull-down menu, you'll be able to enter a **Real time data tag**. In the example above, we are using a simulated data value source.

In comparison to the regular 'Dynamic Tag' option explained above, the Polled Dynamic Tag allows you to set times/recurrences for data retrieval, using a **Polling Group**. A Polling Group, set through the Product configuration/System properties/Polling groups node in AssetWorX, contain polling criteria such as the scan rate, read timeout, group splitting, recurrence type, starting time/date and time of recurrence.

In the Equipment property, if you don't know the exact location, you can click on the tag button to bring up the Data Browser, where you will be able to select from all connected data sources. The **Polling Group** pull-down allows you to either <inherit> the polling settings from the higher branches of the asset tree in which this particular equipment asset has been placed OR you can select a Polling Group that has been created through the Polling Groups feature of the AssetWorX provider's System properties (in this example, we are selecting 'Polling Group Alpha').

You can also decide to use any existing data cached in the database by clicking on the 'Use Database Cache' checkbox.

Benefits of AssetWorX Server Polling include:

- Periodically gets values from occasionally connected OPC Servers
- Stores values into AssetWorX cache
- Reads data in groups of configurable size

- **Source Type: Expression**

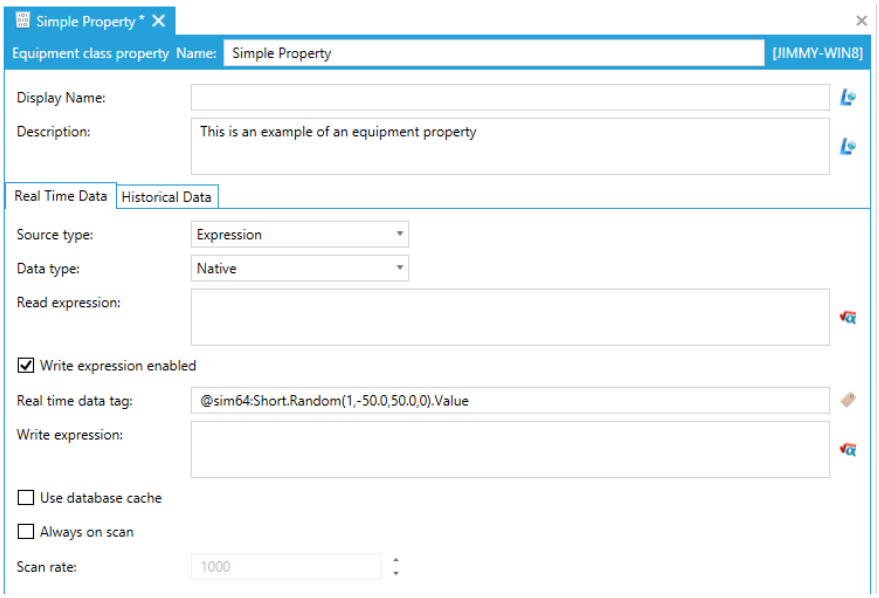


Figure 6.10 - Real Time Data - Source Type: Expression

In this instance, once you select 'Expression' from the pull-down menu, you'll see options to select a 'Data Type' and, depending on that type, be able to enter a Value (in the following text entry field). You can also decide to use any existing data cached in the database by clicking on the 'Use Database Cache' checkbox.

Data Types include:

- Native - will prevent you from entering anything in the Value text field.
- String
- 8 bit signed integer
- 16 bit signed integer
- 32 bit signed integer
- 64 bit signed integer
- 8 bit unsigned integer
- 16 bit unsigned integer
- 32 bit unsigned integer
- 64 bit unsigned integer
- 32 bit floating point number (single precision)
- 64 bit floating point number (double precision)
- Boolean value

Next, you'll be able to either enter a Read Expression directly into the text entry field or click on the [X=] button to open the Expression Editor.

Once you have entered a Read Expression, you have the option of also creating a Write Expression. Click on the 'Write expression enabled' checkbox in order to allow the remaining text entry fields to become active.

You'll be able to enter a **Real time data tag**. If you don't know the exact location, you can click on the tag button to bring up the Data Browser, where you will be able to select from all connected data sources. In the example above, we are using a simulated data value source.

Next, you'll be able to either enter a Write Expression directly into the text entry field or click on the [X=] button to open the Expression Editor.

6.7.2. Equipment Properties as Holding Registers

When adding an Equipment Property, and selecting the Real Time Data tab, you will be able to select a Source Type. By selecting “Static”, it means this Property has no further OPC input or output. If you also elect to "Use Database Cache" (by clicking the checkbox) on that property, it allows for Read/Write by clients. With these settings made, you can use GraphWorX64 (or any other client, or even a server) to write a value to this Property and AssetWorX will remember it, even during computer shutdown, as the value is being written to the Database Cache. In this way, the Property works as a sort of memory cell or holding register for ICONICS applications.

6.7.3. Implementing Equipment Properties in Other ICONICS Components

In real world applications, there are all sorts of OPC servers connected to different devices. When a user starts designing GraphWorX64 displays, in order to get to the OPC points, he or she has to browse all the different OPC servers, each possibly with a uniquely organized address space. In cases where the GraphWorX64 display designer is NOT the same person who configured the individual OPC servers, it might be difficult for the designer to immediately understand where to look for each required OPC point. For this reason, the person configuring the OPC servers could make things easier by creating an AssetWorX Property for each OPC point and organizing them into a logical structure.

Also, this decouples the GraphWorX64 displays from the OPC point names, so when an OPC point name changes (possibly due to someone starting to use a different OPC server or just due to a mistake), it is then possible to just update the OPC input of a single AssetWorX Property that maps this specific OPC point. There could be a dozen GraphWorX64 displays or other components (AlarmWorX64 Server, Hyper Historian, etc.) that subscribe to this AssetWorX Property, but they do not need to be re-configured to get data from the new OPC input, as they do not subscribe directly to it.

6.7.4. Historical Data Tab

The historical data tab offers an easy access for the client application which wants to retrieve historical data for the asset property from the historian database. In other words, it bridges the connection between asset property and backend historical tag (if it's logged in history data logger). What needs to be noted is it ONLY opens the connection between asset property and backend historical tag but does NOT automatically create the historical tag for you. You will need to create logging tag for the property in data logger either manually or through Bulk Asset Configurator. Please refer to "Using Bulk Asset Configurator to Massively Instantiate Equipment Classes" for more details about automating process.

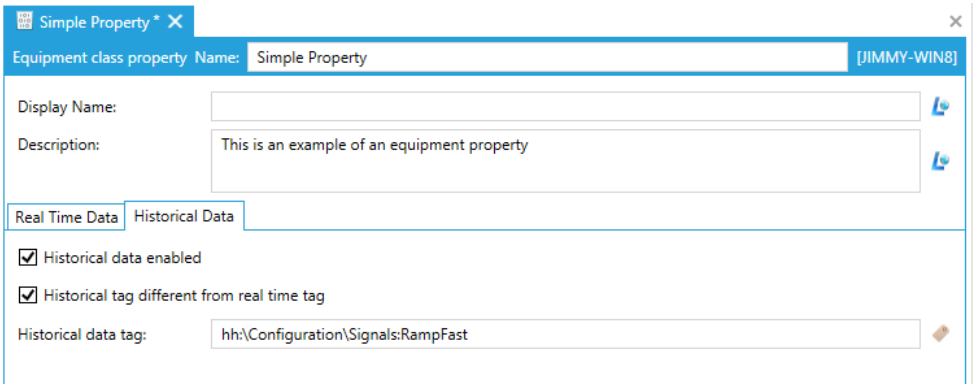


Figure 6.11 - Historical Data Tab in the Equipment Property Settings in AssetWorX

If you decide to link historical data with your equipment property, click on the checkbox for 'Historical data enabled'.

If you have <u>not</u> already changed settings in the Real Time Data tab, the 'Historical tag different from real time tag' checkbox will be inactive. If you did make changes to the Real Time Data tab, click the checkbox if you wish to use a different data tag. In the example above, we have used a simulated historical tag.

You'll then be able to enter a **Historical data tag**. If you don't know the exact location, you can click on the tag button to bring up the Data Browser, where you will be able to select from all connected data sources. In the example above, we are using a simulated data value source.

6.7.5. Activating Equipment

Equipment in the asset tree is not available for runtime processing until you activate it. For an equipment node to be active, it (as well as all parent nodes in the asset tree) must be active. If the equipment or any node above it in the tree is not activated, it will not be available for Runtime processing.

Info

Note: You can create an equipment class so that any equipment node it creates is automatically activated. Simply make sure the Enable check boxes for the equipment class are selected (that is, make sure they have a check mark in them). For more information, refer to the "Creating Equipment Classes for Use as Templates" topic.

To Activate an Equipment Node:

1. In the Workbench SL, expand **AssetWorX > Equipment**. This takes you to the asset tree. Expand the asset tree until you reach the equipment node to be activated

2. Select the node. If the Equipment form does not open, click the **Edit** button on the Home ribbon, or right-click the node and select **Edit** from the pop-up menu.

3. On the Properties Tab, click the **Enabled** check box to put a check mark in it as highlighted in the following image.

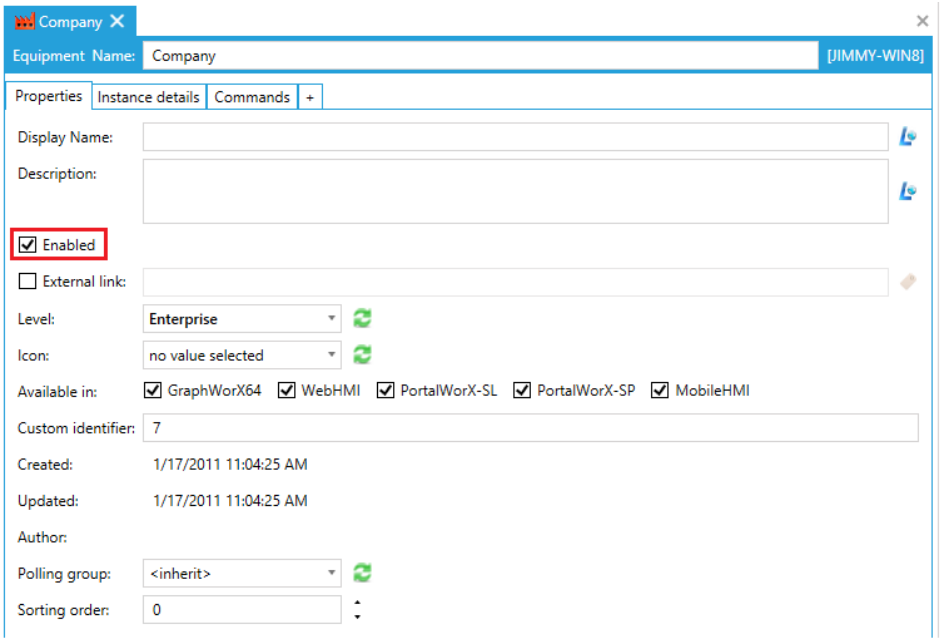


Figure 6.12 - Setting the Enabled Checkbox

4. If other providers' tabs are available for the equipment, make sure the equipment is enabled for the providers you want it to be used for. For example, if the Energy AnalytiX provider is installed and the Energy AnalytiX tab appears in the Equipment form, go to the Energy AnalytiX tab and make sure to click the **Enable Energy AnalytiX** checkbox to put a check mark in it as highlighted below.

5. Click **Apply**.

6. Repeat steps 3 through 6 for each equipment node above the equipment node in the asset tree. For example, following the example shown in these steps, you would repeat the steps for the Foxboro Bakery and Company nodes.

6.7.5.1. Setting up Runtime Menus for Equipment Nodes (Commanding)

At runtime, users can access reports, displays, trends, alarms, and other information about equipment from the AssetWorX's asset tree by right-clicking on any node in the tree. Right-clicking pops up a menu of options that you have defined for runtime users. Runtime menus can vary from node to node, and from platform to platform. For example, you can create menu options that are available only from a desktop monitor or only from a phone app.

All runtime menu options described below are configured on the **Commands** sub-tab of an equipment node's **Equipment** tab. Note however that the same commands can be configured for equipment classes using the **Commands** sub-tab of an equipment class's **Equipment Class** tab.

To Configure a Runtime Menu, Do the Following:

1. Open Workbench and, in the Project Explorer, click the **AssetWorX** provider button to open AssetWorX.

2. Then in the navigation tree do either of the following:

   - Expand **AssetWorX** > **Assets** to access the asset tree. Expand the asset tree and click on the equipment you are defining runtime menu commands for. This displays the Equipment tab, with the **Properties** tab on top.
   - Expand **AssetWorX** > **Equipment Classes** to access equipment classes. Navigate to the equipment class you are defining runtime menu commands for and select it. This displays the Equipment Class tab, with the **Properties** tab on top.

3. Click the **Commands** sub-tab.

4. On the Commands sub-tab, click the **Add** button to add a command to the menu. A form appears at the right.

5. From the **Command** drop-down menu, select the command you want to add. The remainder of this topic describes some common commands and options for configuring runtime menus for equipment nodes and equipment classes. Commands let users do the following in Runtime mode:

   - **Load a display** into the runtime view. You can load GraphWorX64 displays, trends created in TrendWorX64, and alarms created in AlarmWorX64.
     - For an **equipment node's menu**, refer to the "Load Display Command", "Load Alarm Display Command", or "Load Trend Display Command" topic for equipment nodes.
     - For an **equipment class**, refer to the "Load Display Command", "Load Alarm Display Command", or "Load Trend Display Command" topic for equipment classes.
   - **Write a value** to a field.
     - For an **equipment node's menu**, refer to the "Write Value Command" topic for equipment nodes.
     - For an **equipment class**, refer to the "Write Value Command" topic for equipment classes.
   - **Run a transaction** using ICONICS BridgeWorX.
     - For an **equipment node's menu**, refer to the Run Transaction Command topic for equipment nodes.
     - For an **equipment class**, refer to the Run Transaction Command topic for equipment classes.
   - **Set global aliases**.
     - For an **equipment node's menu**, refer to the "Set Global Aliases Command" topic for equipment nodes.
     - For an **equipment class**, refer to the "Set Global Aliases Command" topic for equipment classes.
   - **Open a URL** in the runtime view.
     - For an **equipment node's menu**, refer to the "Open URL Command" topic for equipment nodes.
     - For an **equipment class**, refer to the "Open URL Command" topic for equipment classes.

- **Run a report**.
  - For an **equipment node's menu**, refer to the "Run Report Command" topic for equipment nodes.
  - For an **equipment class**, refer to the "Run Report Command" topic for equipment classes.
- **Load a single report or multiple reports into the runtime view**.
  - For an **equipment node's menu**, refer to the "Load Report Command" or "Load Executed Reports Command" topic for equipment nodes.
  - For an **equipment class**, refer to the "Load Report Command" or "Load Executed Reports Command" topic for equipment classes.
- **Load data from ICONICS FDDWorX**.
  - For an **equipment node's menu**, refer to the "Load FDDWorX Data Command" topic for equipment nodes.
  - For an **equipment class**, refer to the "Load FDDWorX Data Command" topic for equipment classes.
- **Load data from ICONICS Energy AnalytiX**.
  - For an **equipment node's menu**, refer to the "Load Energy AnalytiX Data Command" topic for equipment nodes.
  - For an **equipment class**, refer to the "Load Energy AnalytiX Data Command" topic for equipment classes.

Commands appear on the left, and looks something like the following example.

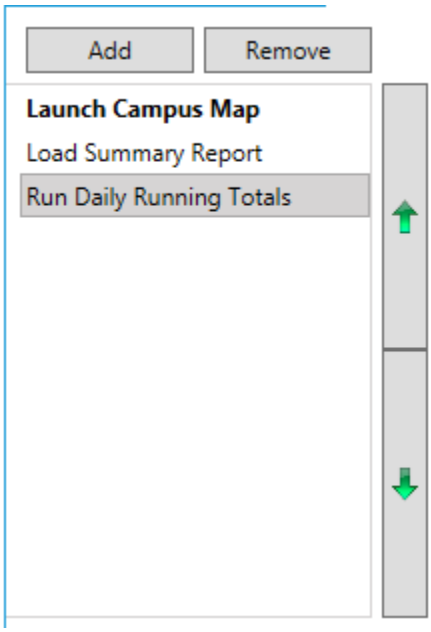


Figure 6.13 - Sample Menu Items on the Commands Sub-tab

6. If you want to modify the behavior of a menu item, select the item in the list so that its configuration information appears on the right. Then modify the menu item's configuration. Note that the type of change you want to make may require you to delete them existing menu item and create a new one.

7. If you want to delete a menu item, select the item in the list and click the **Remove** button.

8. If you want to change the sequence of menu items in the list, select a menu item you want to move, and then click the **up** or **down arrow** to move it up or down in the list.

9. If you want to change the text that appears in the menu, select the menu item with its configuration information on the right. Type the new name in the **Name** field.

10. Complete the fields for the command, and save the command by clicking **Apply** or **Apply & Close**.

6.7.5.2. Making Copies Using the Multiply Function

The multiply command is particularly convenient for creating a number of new items such as equipment nodes or equipment classes in AnalytiX Catalog. You can use it to make multiple copies of an item in the Project Explorer. Use it to make three copies or 50 copies or more. In making multiple copies you can enter the formula to be used for populating fields in the copies: you can leave those fields unchanged, or you can specify a number to be appended to those values.

If you are making copies of a node that has a multi-level structure, the entire structure is copied. Copies will appear above the original item in the same level of the asset tree as the original. The names of copies will have numbers appended to them in accordance with your numbering specifications.

To Use the Multiply Function:

1. In Workbench SL's **Project Explorer**, right-click the item you want to copy. From the pop-up menu, select **Multiply**, as shown below. (The Multiply function may not be available for all items.)

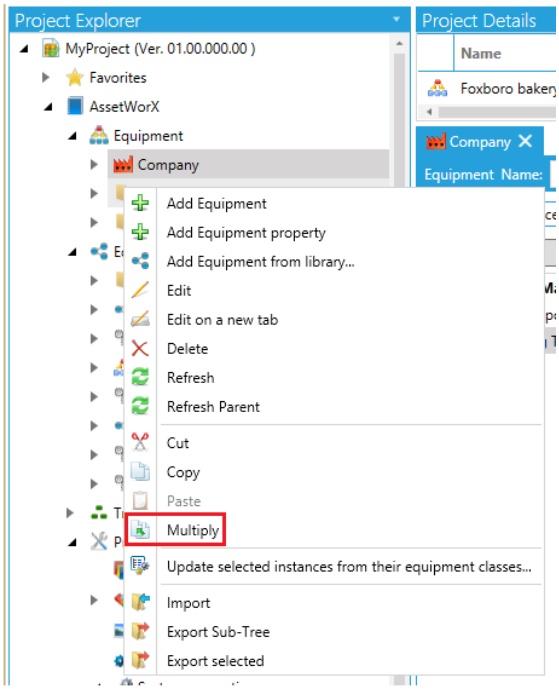


Figure 6.14 - Selecting Multiply from the Pop-up Menu

The Multiply Options dialog box appears, and will look something like the following:

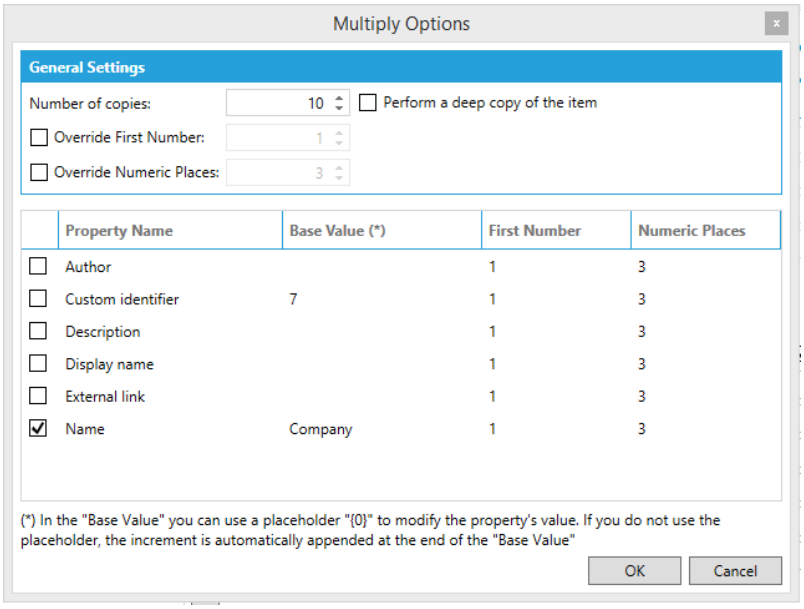


Figure 6.15 - Multiply Options Dialog Box

2. In the **Number of Copies** field, specify the number of copies you are creating.

3. This dialog box lists the properties that you can modify in the copies that are created. The item's properties are listed in the **Property Name** column, and the properties' values are listed in the **Base Value** column. If you want the base value to remain unchanged in all copies that are created, remove the check mark from the **Modify** column. (Note that some properties may be required to change; if they are, you will get an error message when you apply changes.)

4. Change the **Base Value** of any properties as you wish. If you wish the Base Value to be the incremented number that is generated when the multiples are created, enter {0} as the **Base Value**. Otherwise, the number will be added as a suffix to the base value.

5. For each property value that will change, in the **First Number** column specify the first number to be assigned to copies (for example, 100). In the **Numeric Places** column specify the number of numeric places to be reserved for the numbers that are appended. For example, if you enter 5, then five numeric places will be reserved; if the first number is 100, the first copy's name will be appended with 00100, the second with 00101, the third with 00103, and so on.

6. Alternatively, you can apply a uniform numbering scheme to all properties that are being modified by entering values in the **First Number** and **Numeric Places** fields at the bottom of the dialog box, then clicking the **Apply to All** button.

7. When you are ready, click the **OK** button. The copies will appear in the Project Explorer momentarily.

6.8. Using an Equipment Class to Add an Equipment Node to the Asset Tree

Once you have defined an equipment class for the type of equipment you want to add to the asset tree, you can use it to define equipment nodes in the asset tree. You do this by dragging the equipment class from the Equipment Class node and dropping it into the asset tree where it belongs and instantiate the aliases.

To Add an Equipment Node to the Asset Tree by Using an Equipment Class:

1. In Workbench, expand AssetWorX.

2. Under the **Assets** node, expand the asset tree to expose the equipment node you want to add equipment to.

3. Right click on the target equipment and select "Add Equipment From Library…" A dialog will pop up with the equipment classes that are fit for the asset hierarchy where the target asset is at and let you select one of them to instantiate

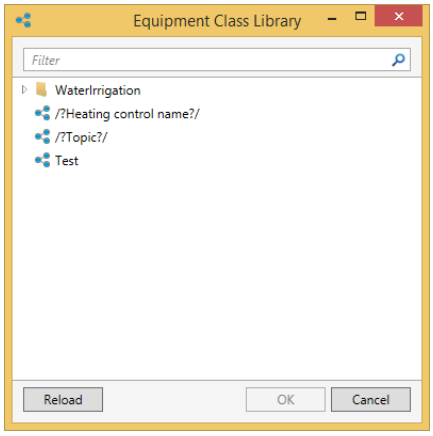


Figure 6.16 – Add Equipment from Library

4. After select equipment class and click OK, the **Instantiation Options** dialog box appears. This is where you will instantiate (or assign values to) the aliases in the equipment class(es). Before you continue, read these conditional items:

   - If you copied the equipment to the wrong equipment node, you can click the **Cancel** button. The dialog box will close and no change is made to the asset tree.
   - If the message "**No aliases found**" appears in the dialog box, click the Instantiate button. A second dialog box will appear; click **Close**, and you are done.

5. The **Instantiation Options** dialog box lists all of the aliases defined for all of the equipment classes you just used. The following example shows two aliases, but there may be many.

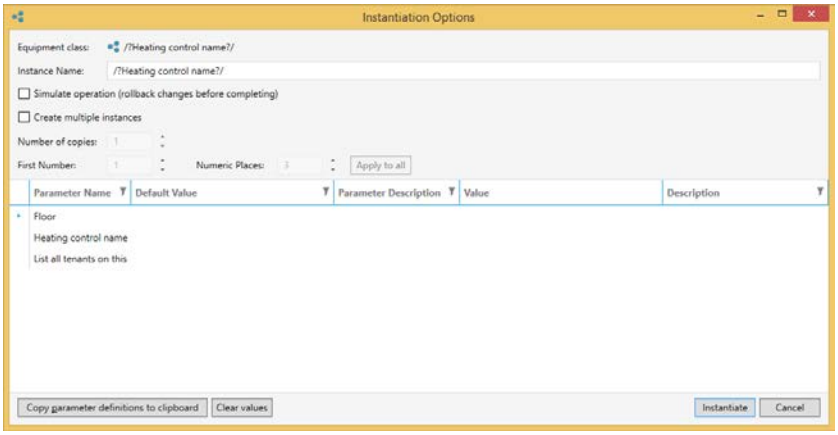


Figure 6.17 - Aliases in the Instantiation Options Dialog Box

At this point, you will be replacing aliases with values to be used for the equipment node you are creating -- or *instantiating* values for the equipment node. It is important to understand that you are not required to provide replacements for aliases at this point. In fact, it is possible that you do not yet know the values to use for the equipment node you are creating. You can skip step 8, or just instantiate a few aliases if you like, and go straight to step 9. You can instantiate aliases at any time later on by following the steps in "Instantiating an Equipment Node's Aliases".

6. For each alias in the dialog box, go to the **Replace With** column and replace the alias with the value to be used for the equipment in the asset tree. In the following example, both aliases have been replaced. The *charger_description* alias has been changed to "Forklift charger" and the *displayname* alias has been changed to "Load recharge times report". You also have the option to **multiply** the number of instances (setting first number and numeric places).

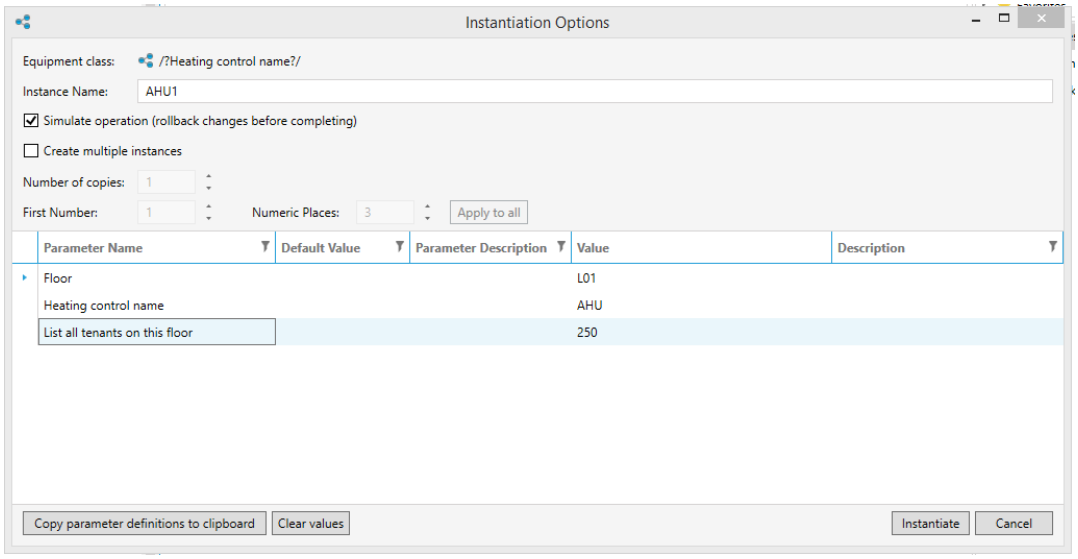


Figure 6.18 - Aliases Replaced with Real Values

7. It is recommended to check “Simulate operation (rollback changes before completing)” for the first time. That way, on clicking the “Instantiation” button, AssetWorX will simulate the process of creating asset without actually committing anything and produce a detailed transaction report toward the end. It helps you to testify your equipment class and alias values without spoiling your real asset tree.

8. Once on errors are found after simulate run, you can uncheck "simulate operation" option and run instantiation again to truly commit the change.

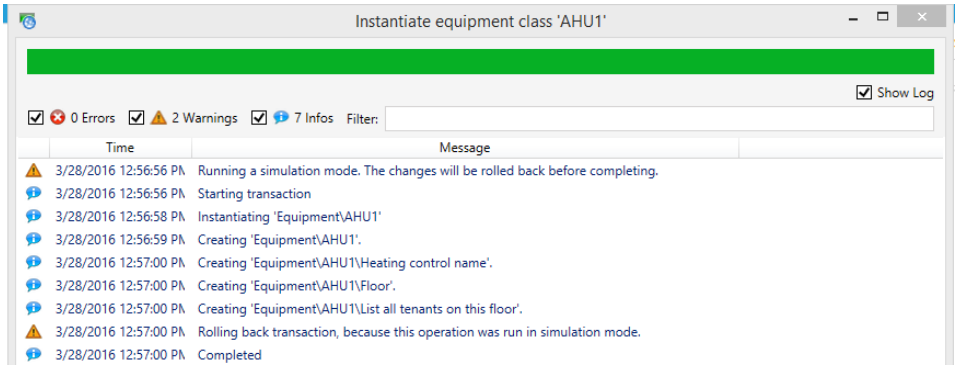


Figure 6.19 - A Successful Simulate Instantiation

6.9. Using Bulk Asset Configurator

Now that you know how to create equipment classes, use aliases to make the classes more dynamic and flexible, as well as use built-in feature to instantiate equipment classes in Workbench, you may have found that it's great to have so many features to ease the process of building asset tree but it's not that fast yet to massively produce assets on the tree. In this section, you will see how this standalone utility called "Bulk Asset Configurator" can max out your productivity through some simple steps.

6.9.1. Prepare an Instantiation Sheet

The bulk asset configurator instantiates assets based on existing equipment classes and an input file which contain the data entries that indicates what classes to use, what assets to create and what values to replace for aliases. The input file is expected to be a spread sheet in Excel 2007 format (*.xlsx). But before we begin defining the sheet, we always want to make sure we have the desired equipment classes created in AssetWorX provider in Workbench first. For more details about creating equipment class, please refer to previous sections in this chapter. For now, we will assume you have equipment classes created which are ready for use.

1. Create a blank new Excel sheet on desktop (or anywhere you prefer).
2. Make sure you DO NOT have the new sheet opened in Excel.
3. Go to Start→All Programs→ICONICS Tools and launch Bulk Asset Configurator (If you're using Windows 8 or Server 2012, in the search bar on the startup page, type in Bulk Asset Configurator).

4. In the configurator window, click on “Bulk Asset Configurator with a blank new input file” button as outlined below.

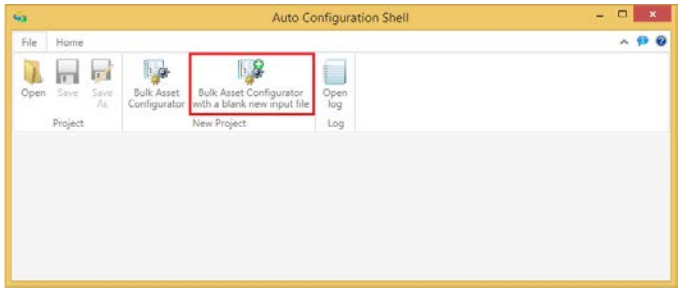


Figure 6.20 – Bulk Asset Configurator

5. Browse to the new spreadsheet just created and click save. You will be prompted to ask if you want to overwrite the existing file, click yes to acknowledge the change.

6. Open the spreadsheet now in Excel.

7. You will realize that the blank spreadsheet has now been reformatted and configured with multiple sheets.

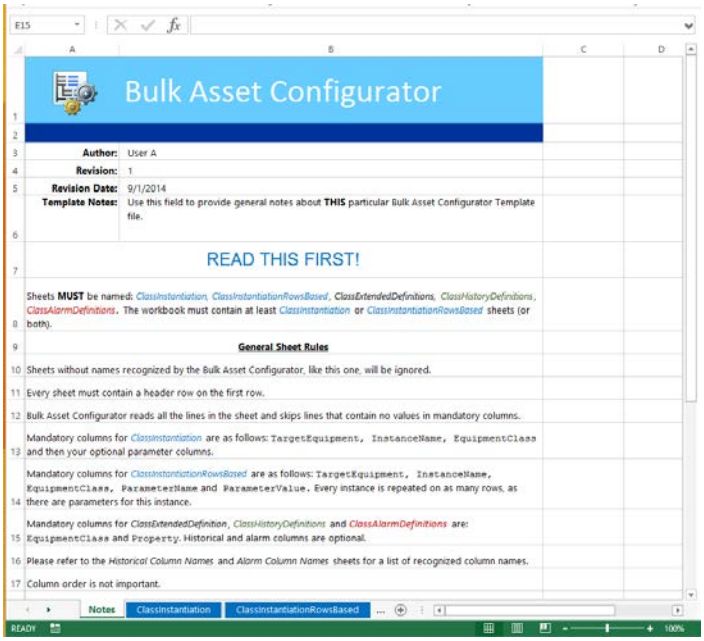


Figure 6.21 – Bulk Asset Instantiation Input File

6.9.2. Build Up the Sheet

Now we can start filling information into this input sheet with proper format which can later on be processed through Bulk Asset Configurator and "transformed" equipment classes into real assets. The reformatted input file has a "Notes" sheet to help you getting started with the sheets. It also contain two "column names" sheets to assist you in identifying what values to put for what columns for different type of object definitions.

1. The first sheet "Notes" provides you the general regulations you need to follow to create a valid input file to make sure it can successfully run through the bulk asset configurator.

READ THIS FIRST!

Sheets **MUST** be named: *ClassInstantiation*, *ClassInstantiationRowsBased*, *ClassExtendedDefinitions*, *ClassHistoryDefinitions*, *ClassAlarmDefinitions*. The workbook must contain at least *ClassInstantiation* or *ClassInstantiationRowsBased* sheets (or both).

General Sheet Rules

Sheets without names recognized by the Bulk Asset Configurator, like this one, will be ignored.

Every sheet must contain a header row on the first row.

Bulk Asset Configurator reads all the lines in the sheet and skips lines that contain no values in mandatory columns.

Mandatory columns for *ClassInstantiation* are as follows: `TargetEquipment, InstanceName, EquipmentClass` and then your optional parameter columns.

Mandatory columns for *ClassInstantiationRowsBased* are as follows: `TargetEquipment, InstanceName, EquipmentClass, ParameterName` and `ParameterValue`. Every instance is repeated on as many rows, as there are parameters for this instance.

Mandatory columns for *ClassExtendedDefinition*, *ClassHistoryDefinitions* and *ClassAlarmDefinitions* are: `EquipmentClass` and `Property`. Historical and alarm columns are optional.

Please refer to the *Historical Column Names* and *Alarm Column Names* sheets for a list of recognized column names.

Column order is not important.

All three class definition sheets (*ClassExtendedDefinition*, *ClassHistoryDefinitions*, *ClassAlarmDefinitions*) are merged together in the *ClassInstantiation* sheet. If there are duplicates, the Bulk Asset Configurator will generate an error. The *ClassAlarmDefinitions* should be used to define alarms only, The *ClassHistoryDefinitions* sheet is intended for historical definitions only. The `GenerateAlarm` and `GenerateHistory` columns are `TRUE/FALSE`. They are set to `FALSE` by default, but it is important to not forget to include this column.

All paths use the backslash (\) as a separator, not the forward slash (/).

Usage of the /??path?/ parameter: this parameter generates a generic path to the current equipment. It's main purpose is the use inside calculcated tag expressions for the historian. The syntax is: /??path(serparator, relativePath)?/. All arguments are optional. This parameter will concatenate the Equipment path with the 'relativePath'. 'relativePath' is forward slash separated and can contain '..' segments to go up a level. The resulting string from this parameter has no prefix and individual segments are separated by the 'separator' value. For example, if the path to the Equipment is 'ac;/Enterprise/Area/', using the syntax /??path(,..,/Area2)?/ will generate the result 'Enterprise.Area2'.

Figure 6.22 – Notes for Bulk Asset Instantiation Input File

2. The second sheet "Class Instantiation" is where you define the equipment class to use, the asset to create, the destination location and the value to replace for aliases. Between "Class Instantiation" and "Class Instantiation Rows Based", these are the two most important sheets that you have to fill up (either one) to get assets created.

3. On "Class Instantiation" sheet, there are five columns with headers entered by default, they are: TargetEquipment, InstanceName, EquipmentClass, Parameter 1 and Parameter 2:

- Target Equipment is the destination location where you want to instantiate your assets into. Each level in the location path has to be separated by back slash (\). You don't need to specify "Equipment" node in the path. If anyone of the level in the path does not exist in asset tree, it will be created on the fly when bulk asset configurator runs.
- Instance Name is the name of the asset you want to create.
- Equipment Class is the path to the equipment class (including the class's name) you want to use to create the instance. Each level should also be separated by back slash (\). You don't need to specify "Equipment Class" node in the path.
- Parameter 1 and 2 and so on are the aliases you want to replace with value/data/string. Parameter 1 and 2 should be replaced with the name of the aliases you have on the equipment class. For instance, if you have an alias called /?Device?/, you need to replace "Parameter 1" with "Device" and fill in some value into a line on that column which associate with the equipment class and instance you want to create. If there are multiple aliases (more than two), you can keep entering alias name into the header of the columns to the right and values you want to replace with.

A	B	C	D	E	F
TargetEquipment	InstanceName	EquipmentClass	Device	Floor	Temp
Campus\Site\Building\Roof	AHU1	AHU	AHU101	LR	bacnet:devic

Figure 6.23 – Class Instantiation Sheet

4. "Class Instantiation Row Based" sheet is just a pivot version of "Class Instantiation" sheet for certain users' convenience. As long as you have one of them configured, bulk asset configurator will be able to create assets for you.

5. "Class History Definitions" sheet is where you can convert equipment properties into Hyper Historian tags or calculations. If "Class Instantiation" configuration is instance based, then "Class History Definitions" is equipment class based. Unlike the "Class Instantiation" sheet where you specify instance and alias value, you list down the properties of the equipment classes you want to log. As long as the equipment class is going to be instantiated, the specified properties will be converted into logging tags in Hyper Historian.

6. "Class Alarm Definitions" sheet is very similar to "Class History Definitions" sheet except that it's going to convert properties into alarm tags in AlarmWorX64 server.

6.9.3. Understand Bulk Asset Configurator

Once we have a simple Bulk Asset Instantiation Input File configured, we can pass it through the Bulk Asset Configurator to start populating assets into Asset Tree. When the configurator is executing, it looks up equipment classes referred in the input file from AssetWorX database and duplicating them into Asset table with the aliases replaced by real values specified in the input file. Here is how it looks:

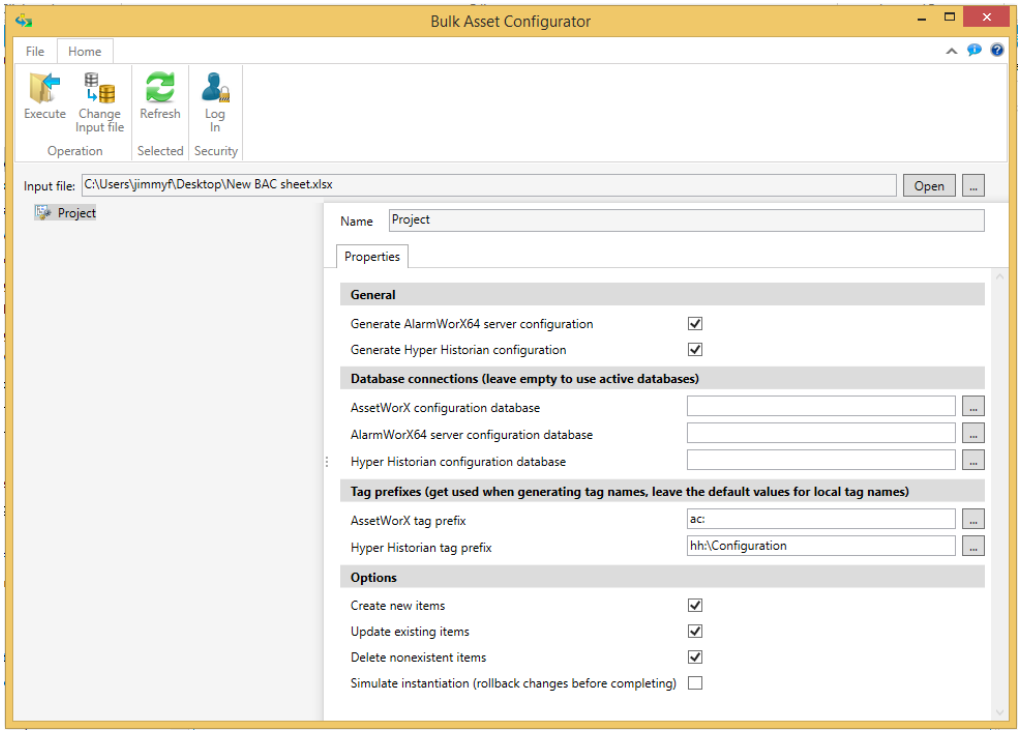


Figure 6.24 – Bulk Asset Configurator with Input File

Bulk Asset Configurator supports loading input file directly or loading from a previously saved project which contains both input file as well as detailed settings. To access to various loading options, click on file on the ribbon:

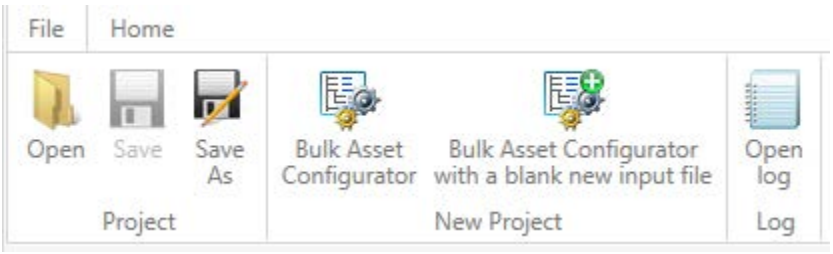


Figure 6.25 – File Ribbon in Bulk Asset Configurator

"Open" is to load a previously saved project file. The project file should end with .autoinst file extension. "Bulk Asset Configurator" is to load a saved Bulk Asset Instantiation Input File whereas "Bulk Asset Configurator with a blank new input files", as mentioned before, loads a new empty Excel file and reformats it to contain desired sheets.

The input file field shows the path of the input file. You can click on the "open" button to launch input file in Excel or click on […] button to pick another input file.

Move down to Properties tab, these are the main settings for Bulk Asset Configurator, as shown below. You can optionally change the name of the project file if you want to save the configuration and reuse it later on.

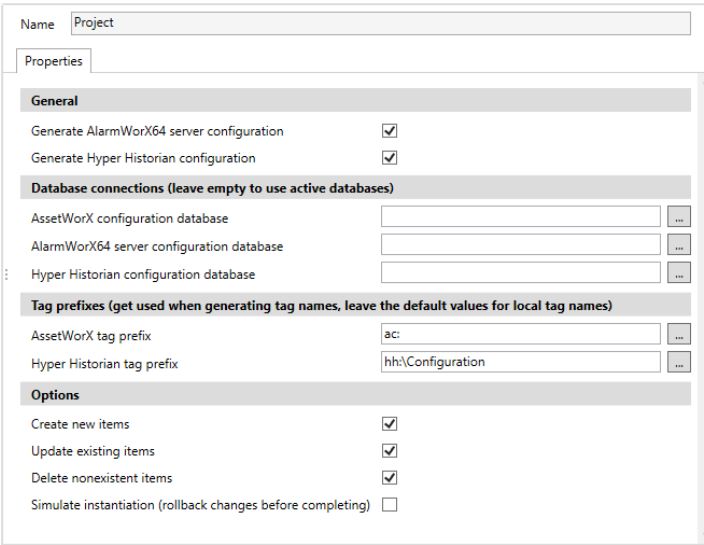


Figure 6.26 – Bulk Asset Configuration Properties

The General section is where you can decide if you want to generate History tags and/or Alarm tags. If you don't have anything added in "Class History Definitions" sheet and "Class Alarm Definitions" sheet, you are recommended to not to have them checked.

Under Database connection you are optional to insert connection string for the target database you want to add assets or history or alarm tags into. As the tooltip indicates, Bulk Asset Configurator will use current active databases (in the provider) as the destination database for new assets/tags. If there is a specific need that you want to create new items into offline databases, you can use the ellipsis […] button to pick up the desired SQL instance and database.

Tag prefixes need to be configured when you have a distributed setup where AssetWorX and Hyper Historian are not set up on the same machine. In this case, AssetWorX tag prefix is what Hyper Historian tags will use and Hyper Historian tag prefix is what AssetWorX Property (History Data tab) will use.

Lastly Options is where you can specify if you want to:

- Create new items
- Update existing items
- Deleting nonexistent items
- Simulate instantiation

Bulk Asset Configurator compares asset tree (and assets) with equipment classes referred in the input file. So if the instance name in the input file doesn't exist in asset tree and if "Create new items" is checked, it will create new items. If the instance name already exists but there are things that are different from the referred equipment class and "Update existing items" is checked, the discrepancies will be updated so they will follow what equipment class has. If there are things exist in instance but no longer exist in equipment class and "Deleting nonexistent items" is checked, these things will be removed from the instance.

Similar to Workbench, it is always recommended to simulate instantiation process every time and check out transaction log before you commit changes into AssetWorX.

6.9.4. Populate Assets through Bulk Asset Configurator

Now that you have the input file ready and understand what properties are for what purpose, you can set up Bulk Asset Configurator and ready to populate assets into AssetWorX.

Make sure you have input file loaded up and "Simulate instantiation" checked. To kick off Bulk Asset Configurator, go to "Home" tab on the ribbon and click on "Execute". You will see a transaction log populates as follows:

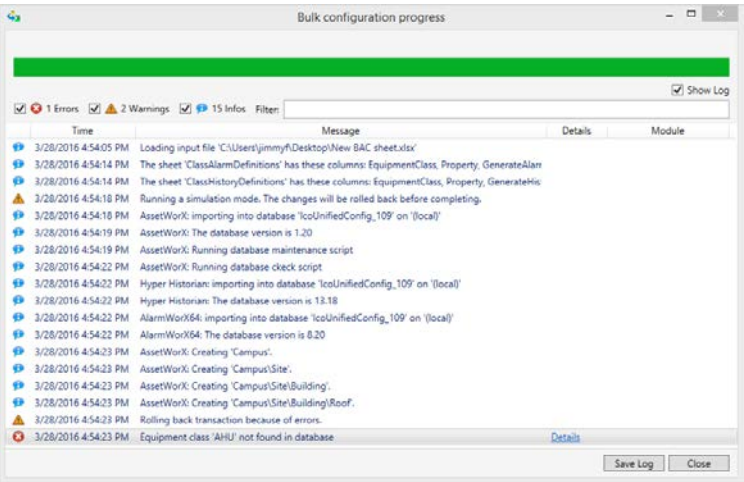


Figure 6.27 – Transaction Log

As you can see, the input file has an invalid entry of equipment class “AHU” which cannot be found under equipment class node in AssetWorX. With simulation run and the transaction log, you can easily pinpoint issue without endangering your AssetWorX configuration.

Lab: Equipment Classes L6A

6A.1. Lab Overview

In a previous lab, you created the foundation of the asset tree that will be used for the remainder of this training course. In this lab, we will proceed by creating an equipment class and instantiating equipment class into the asset tree.

Estimated time to complete this lab: **60 minutes**

6A.1.1. Task Checklist

In this lab, you will complete the following tasks:

- Create an equipment class with equipment properties.
- Instantiate the equipment class to resolve alias definitions.
- Update assets with shared equipment class.
- Use Bulk Asset Configurator to massively instantiate equipment classes

6A.1.2. Support Files

You do not need any support files to complete this lab.

6A.2. Adding Tree Nodes Using an Equipment Class

Let's get started.

6A.2.1. Configuring an Equipment Class Asset

We will proceed by creating a child node under the Equipment classes branch.

1. Launch Workbench.
2. From within the Project Explorer on the left, expand the **AssetWorX** provider.
3. Right-click on the **Equipment classes** node and select **Add Equipment class**.

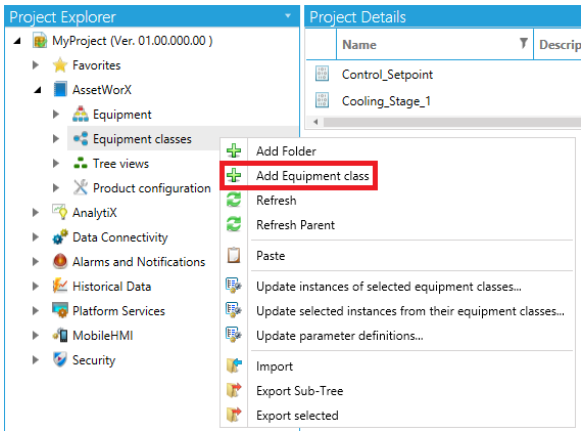


Figure L6A.1 – Adding an Equipment Class

4. Name the Equipment class **/?RTU_Name?/**.

   Note: The delimiters /? and ?/ are place holders for values that will be entered during the instantiation process.

5. Set Level and Icon to **Process Cell**. Click **Apply**.

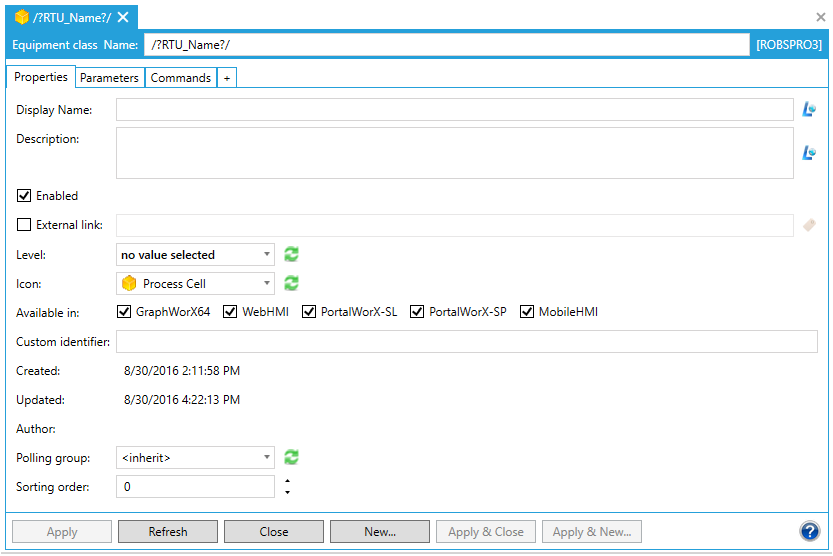


Figure L6A.2 – Define the Icon: as Process Cell

6. Select the **Commands** sub-tab.

7. Click on the **Add** button.

8. Configure the **Command** as described below:

   Note: Leave the default settings in any field that is not mentioned below.

   - **Command:** Load Graphics Display
   - **Name:** Load LibraryAHU Display
   - **Description:** Load Graphics Display LibraryAHU
   - **Execution flags:** Visible and Drag & drop
   - **Filename:** http://localhost/PubDisplay/LibraryAHU.gdfxs

   Note: Manually enter the **URL** associated with the **Filename:** (above).

   Note: localhost should be replaced with the hostname or IP address of the computer where IIS is running and the displays have been published.

   - **GlobalAliases:** #RTU=&/?RTU_Name?&//2;

   Note: It is recommended to set the above syntax through GUI. To do so: click on the book icon right to Global Aliases field. A form will pop up. Click on Add… under Aliases section. For Alias Name, enter: **RTU**. For Value, enter: **/?RTU_Name?/**. For Scope, check **Absolute** and leave it at **Document, Level0**. On clicking OK twice, the above syntax will be automatically filled up in the Global Aliases field.

9. Your configuration should resemble the figure below.

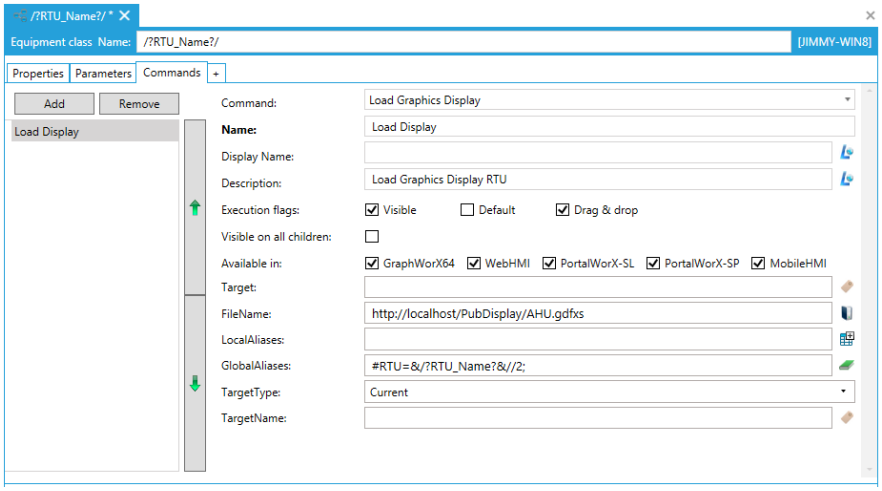


Figure L6A.3 – /?RTU_Name?/ Equipment Class Definition

10. Click **Apply & Close**.

6A.2.2. Adding Equipment Properties

For each level of Assets within the tree control of the AssetWorX provider in Workbench, you can add Equipment Properties. These allow you to connect the associated asset to real-time (utilizing multiple data source options) or historical data sources. Let's proceed to learn how this works.

1. We will continue by defining Equipment Properties within the /?RTU_Name?/ Equipment Class.

2. Let's proceed by right-clicking on the **/?RTU_Name?/** node under **Equipment classes**. Then select **Add Equipment class property.**

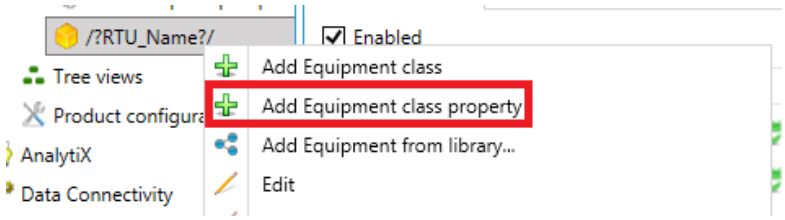


Figure L6A.4 – Adding an Equipment Property to the RTUs Equipment Class

3. Configure the first property:

   - **Name:** Room Temperature
   - **Source Type:** Dynamic Tag
   - **Real time data tag:**

   ac:IcoUniversity/North Campus/Engineering Building/HVAC 01/RoomTemperature_opc

   Note: You can either type in the tag manually or you can click on the ellipses to launch the Data Browser and browse for the tag Home → AssetWorX → IcoUniversity → North Campus → Engineering Building → HVAC 01 → RoomTemperature_opc

4. Click **Apply**.

5. We will keep adding the following properties.

6. Second property:

- **Name**: CLG_STG1_ENA
- **Source Type**: Static Value
- **Data Type**: Boolean
- **Value**: False
- **Use Database Cache**: check

NOTE: If you get a warning during this step, please consult with your instructor. The cache data connection string in **Other settings** may be empty.

7. Right-click on CLG_STG1_ENA under /?RTU_Name?/ on the left side tree and choose **Multiply**.

8. In Multiply Options dialog, enter:

- Number of copies: 3
- Override First Number: 2
- Override Numeric Places: 1
- Name: CLG_STG{0}_ENA

9. Click OK and you should see three more properties showing under /?RTU_Name?/

10. Add a last property:

- Name: CLG_STG_ACT_STS
- Source Type: Expression
- Data Type: 32 bit signed integer
- Read Expression:

```
x={{ac:USA/Massachusetts//?Location?///?RTU_Name?//CLG_STG1_ENA}
}+{{ac:USA/Massachusetts//?Location?///?RTU_Name?//CLG_STG2_ENA}
}+{{ac:USA/Massachusetts//?Location?///?RTU_Name?//CLG_STG3_ENA}
}+{{ac:USA/Massachusetts//?Location?///?RTU_Name?//CLG_STG4_ENA}}
```

11. Click **Apply** to save it.

12. Your final equipment class should resemble the figure below:

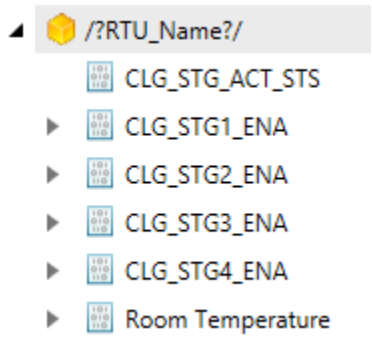


Figure L6A.5 – RTU Equipment Class with Properties

13. Double-click on /?RTU_Name?/ class again to open it up.

14. Click on **Parameters** tab and then click on **Detect**.

15. You will see **RTU_Name** and **Location** showing up in the list:

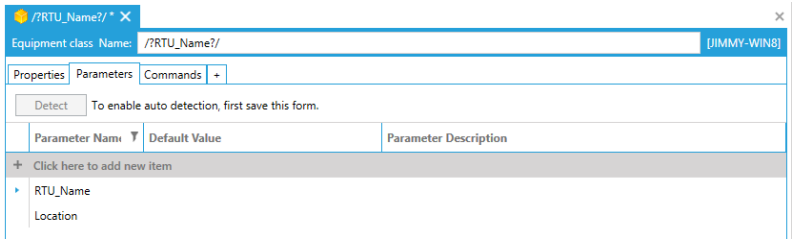


Figure L6A.6 – Equipment class parameters

6A.2.3. Adding the Equipment Class to the Asset Tree

The figure below depicts the current asset tree. We will proceed by instantiating /?RTU_Name?/ class onto the Foxboro node.

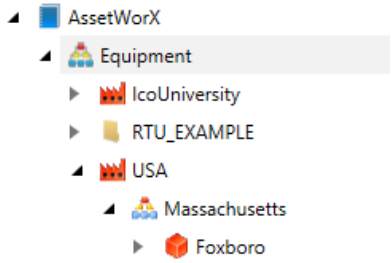


Figure L6A.7 – A View of the Current Asset Tree

1. Expand the **USA** tree until **Foxboro** is exposed.

2. Right-click on **Foxboro** and select **Add Equipment from library**.

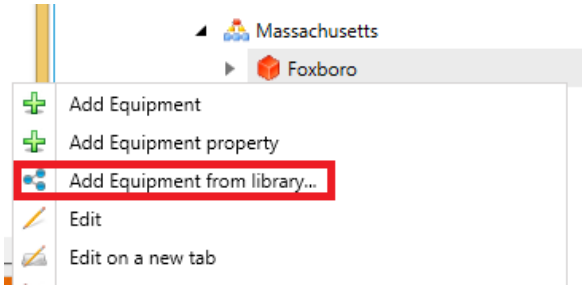


Figure L6.8 – Add Equipment from library

3. In the popup, select the **/?RTU_Name?/** equipment class and click **OK**.

4. A dialog called **Instantiation Options** will be populated as follows.

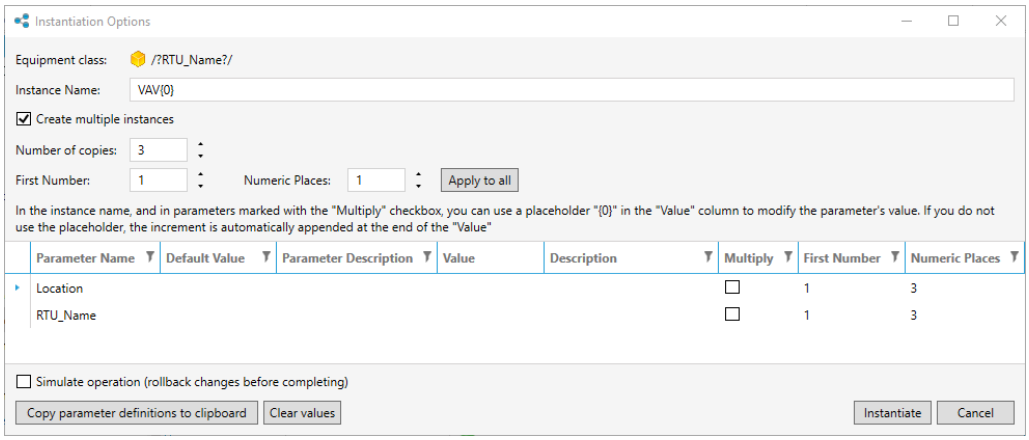


Figure L6A.9 – Instantiation Options

5. You will notice **Simulate operation** is unchecked. It is recommended to have it **checked** for the first time when you want to instantiate something. The merit of this is it will pretend to create assets with real transaction details but roll back afterwards. It gives you all the information happening behind the scene to help you validate if the equipment class is good or if the value you enter for parameter is good without spoiling your true asset tree. So we will check it for now.

6. Check "**Create multiple instances**" and enable (check) the "**Multiply**" option next to **RTU_Name** parameter.

7. Set the **Instance Name** to **RTU{0}**.

8. Enter **2** for "**Number of copies**", **2** for "**Numeric Places**" and hit "**Apply to all**".

9. Set the **Value** field to **RTU{0}** in the **RTU_Name** parameter.

10. Enter **Foxboro** into the **Value** field of the **Location** parameter.

11. Your complete form should now look like this:

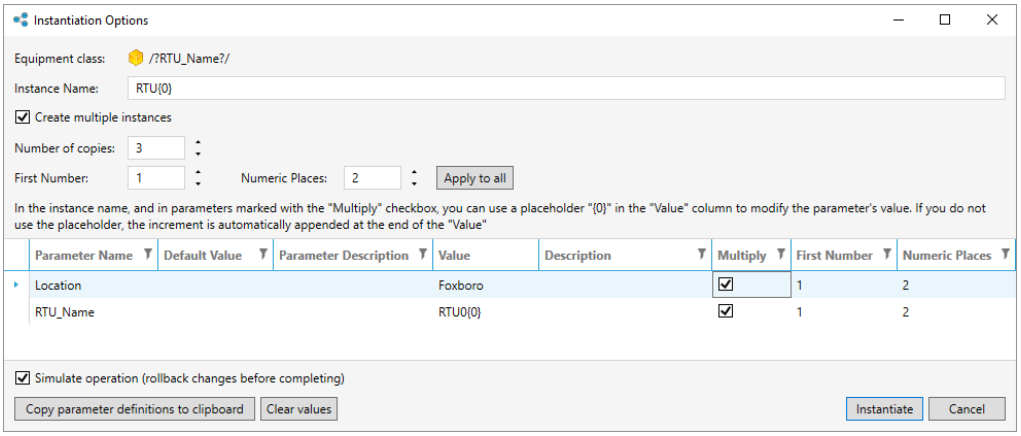


Figure L6A.10 – Completed Instantiation Options

12. Click on Instantiate button and you will see a transaction log coming up on screen.

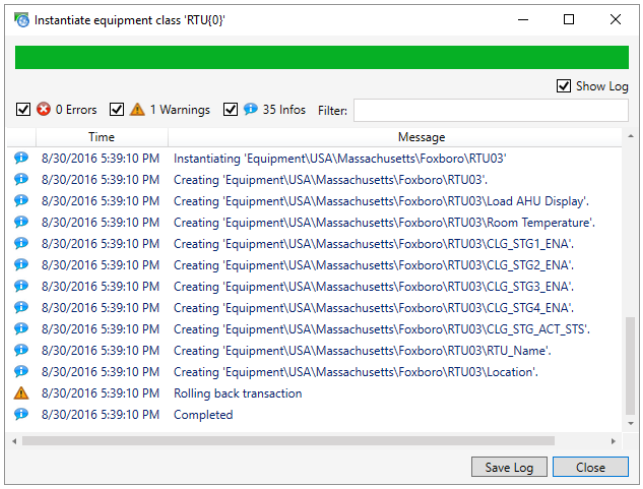


Figure L6A.11 – Transaction Log

13. Once it's completed and error free. Close the log and do "Add equipment from library" again but this time with **Simulate operation** unchecked.

6A.2.4. Update Assets with Shared Equipment Class

Now that you have two assets instantiated from an equipment class, you can maintain the assets with the equipment class they both share with. This is a new feature we introduce in 10.90 which is a true linkage between asset and class. If you double click on one of the asset we instantiated in the previous section, you will see under Instance details there is the equipment class from which this asset is created. Because of this true linkage, we can update items on the equipment class and push the updates to its linked assets.

In this section, we are going to add a new property into equipment class and push it to the two assets.

1. Right-click on **/?RTU_Name?/** equipment class and select **Add Equipment class property**.

2. Name it **Thermostat Set Point**.

3. Set **Source type** to **Dynamic Tag**.

4. For **Real time data tag**, enter:

   ac:IcoUniversity/North Campus/Engineering Building/HVAC 01/ThermostatSetPoint_opc

   Note: You can manually type in the tag, copy/paste, or click on the "tag" icon to launch the Data Browser and browse for the tag: **Home** → **AssetWorX** → **IcoUniversity** → **North Campus** → **Engineering Building** → **HVAC 01** → **ThermostatSetPoint_opc** and click on **OK**.

5. Click **Apply**.

6. Go to the previously instantiated **RTU01** and **RTU02** under USA → Massachusetts → Foxboro.

7. Right-click on **RTU01** and select "Update selected instances from their equipment classes"

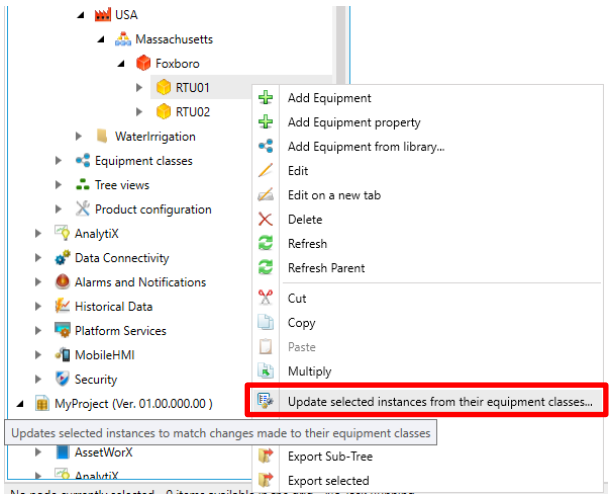


Figure L6A.12 – Update instance with equipment class

8. On the popup, you will see there is **Simulate operation** again. We will leave it uncheck for this time and proceed with update. Click on **OK**.

9. You will see the transaction log again and you should see the new property has been pushed to **RTU01**.

10. Close the log and expand **RTU01**. Your asset should resemble the figure below:

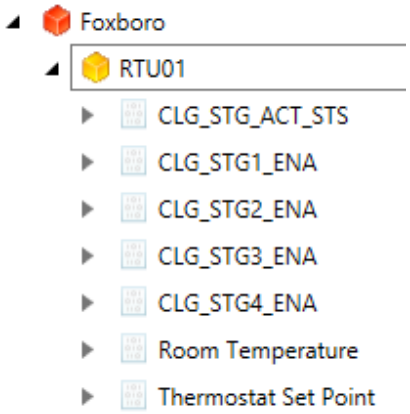


Figure L6A.13 – RTU01 with new property

11. Update **RTU02** with the same procedure.

12. Now both RTUs should have been updated.

6A.2.5. Using Bulk Asset Configurator to Instantiate Assets

Bulk Asset Configurator is a standalone application which gives you more flexibility in the process of asset instantiation. It utilizes Excel input where you can enter any assets which you want to create from any existing equipment classes and define values for their parameters. With a few clicks, you can instantiate assets and also convert properties into Hyper Historian tags or AlarmWorX64 server tags. In this section, we will create a few more RTUs with the existing equipment class.

1. Launch Excel and save the new workbook. Name it "**Bulk Asset Sheet.xlsx**".

2. Go to **Start → All Programs → ICONICS Tools → Bulk Asset Configurator** (or if you're using Server 2012, go to startup page and search for Bulk Asset Configurator).

3. On the ribbon, click on "**Bulk Asset Configurator with a blank new input file**".

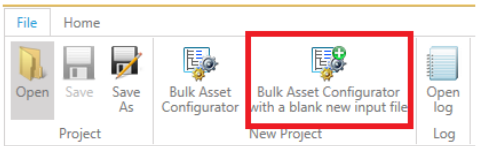


Figure L6A.14 – Button on the Ribbon

4. Browse to the "**Bulk Asset Sheet.xlsx**" we just created and click **Save**. Click on **Yes** on the popup.

5. Now you will see the interface has changed and more settings come up.

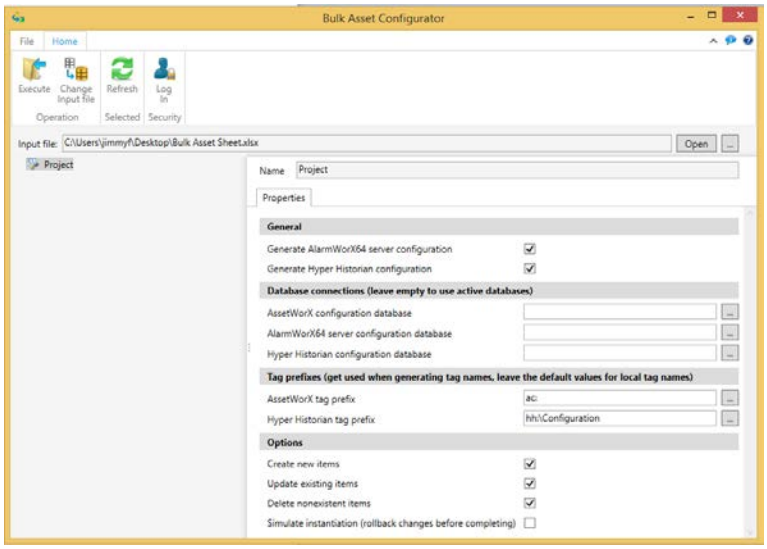


Figure L6A.15 – Button on the Ribbon

6. Let's leave the Bulk Asset Configurator open in the background for now and let's open the "**Bulk Asset Sheet.xlsx**" in Excel.

7. You will see the empty workbook now has multiple sheets.

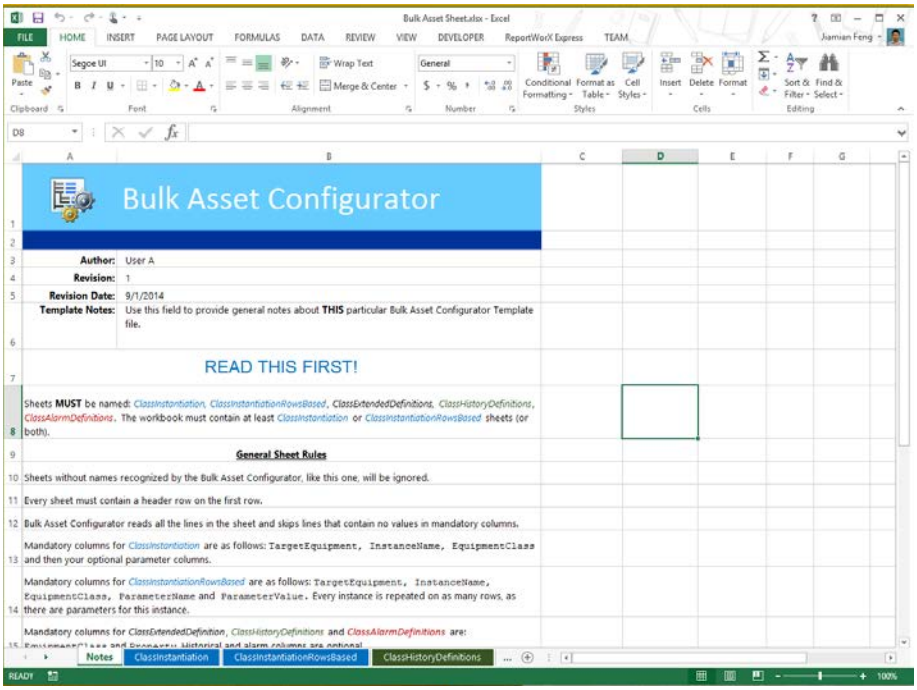


Figure L6A.16 – Button on the Ribbon

8. It is recommended to read the first sheet "**Notes**" carefully. It contains the regulations you need to follow to configure bulk asset sheet and make it work when run through the configurator.

9. Let's move to the "**ClassInstantiation**" sheet. You will see some entries in there. The first line is the header of the columns. The stuff in second and third lines are just samples. Let's remove those samples.

10. "**TargetEquipment**" is the destination where you want to create assets into. Let's key in: **USA\Massachusetts\Mansfield**

    Note: It has to be the complete path you want to create assets into. Each level should be separated by a back slash and the root node "Equipment" **should be excluded**.

11. "**InstanceName**" is the name of the asset you want to create. Let's type in: **RTU01**.

12. "**EquipmentClass**" is the name of the class you want to create assets from. Let's enter: **/?RTU_Name?/**

    Note: Similar to TargetEquipment. It has to be the complete path to pinpoint the equipment class, including the name of the class. Each level should be separated by aback slash and the root node "Equipment classes" **should be excluded**.

13. "**Parameter 1 Name**" and "**Parameter 2 Name**" are the parameters you want to replace value with. You need to replace "**Parameter 1 Name**" with the actual name of the parameter and then fill in values in that column for the target instance. You can keep replacing "**Parameter 2 Name**" and adding as many parameter names as you need for your equipment class to the right of the sheet (as far as Excel can go). In our case, you need to replace "**Parameter 1 Name**" with **RTU_Name** and "**Parameter 2 Name**" with **Location**

14. Enter **RTU01** under **RTU_Name** column, and **Mansfield** under **Location** column

15. Let's duplicate items in line 2 into line 3 and line 4. Update RTU numbers.

16. Here is what your final sheet should be looking like:

A	B	C	D	E
TargetEquipment	**InstanceName**	**EquipmentClass**	**RTU_Name**	**Location**
USA\Massachusetts\Mansfield	RTU01	/?RTU_Name?/	RTU01	Mansfield
USA\Massachusetts\Mansfield	RTU02	/?RTU_Name?/	RTU02	Mansfield
USA\Massachusetts\Mansfield	RTU03	/?RTU_Name?/	RTU03	Mansfield

Figure L6A.17 – Bulk Asset Sheet

17. Click through "**ClassInstantiationRowsBased**", "**ClassHistoryDefinitions**" and "**ClassAlarmDefinitions**" and remove all samples except headers.

18. We will not cover the rest of the sheets in this workbook. If you're interested, please ask the instructor or refer to the application notes for the Bulk Asset Configurator for more details.

19. **Save** the workbook and **Close**.

20. Let's go back to **Bulk Asset Configurator** and go over the settings on there.

21. Under **General**, you have options to generate Alarm tags and Hyper Historian tags. These are based on "**ClassAlarmDefinitions**" and "**ClassHistoryDefinitions**" sheet respectively. For now, we will uncheck them.

22. Under **Database connection**, these will be the target (configuration) databases you want the assets, alarms and history tags be created into. By default, they are empty and they are going to use the active databases for the respective provider. Let's leave them empty for now.

23. **Tag prefixes** takes into account the distributed architecture where AssetWorX and Hyper Historian are loaded on separated servers. For this lab, we will ignore them.

24. Under Options, you can choose to either "**Create new items**", "**Update Existing items**", "**Delete nonexistent items**" and "**Simulate instantiation**". Similar to what we have done with shared equipment class, we want to simulate the instanton first. For the sake of this lab, we can just uncheck "**Update**" and "**Delete**". Enable **Simulate Instantiation** if it is not already enabled.

25. Your complete settings should resemble the figure below:

26. Click on "**Execute**" on the ribbon.

27. You will see a familiar transaction log again.

28. If no errors, you can close the transaction log, uncheck "**Simulate instantiation**" and "**Execute**" again.

29. Once it's completed, you can go to **Asset** tree in **Workbench**, right click on **USA** level and select **Refresh**. Drill down and you will find the new RTUs under Mansfield. The USA asset tree should look like the figure below.

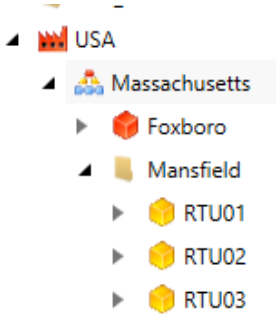


Figure L6A.18 – Completed Asset Tree

30. We should probably change the **Icon** for Mansfield to **Area**. Be sure to **Apply & Close** after the change is completed.

This concludes this lab.

Lab: Bulk Asset Configurator *L6B*

6B.1. Lab Overview

In this lab, we will generate multiple assets in bulk using the Bulk Asset Configurator. We will assign equipment properties, historical tags and alarms to the assets.

Estimated time to complete this lab: **30 minutes**

6B.1.1. Task Checklist

In this lab, you will complete the following tasks:

- Instantiate assets using the Bulk Asset Configurator
- Assign properties to those assets
- Make historical tags for those assets
- Make alarms for those assets

6B.1.2. Support Files

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

6B.2. Making the Excel Spreadsheet

1. Launch the **Bulk Asset Configurator** from the **ICONICS Tools** folder.
2. In the **New Project** section of the **File** ribbon, click the **Bulk Asset Configurator with a blank new input file** button. Create your Excel spreadsheet in the C:\Training folder. Name the file "**MyProjectBulkAssetConfig**".
3. Click the **Open** button to the right of the **Input File** name to open the spreadsheet in Excel.

6B.3. Creating Assets

1. At the bottom of the window, go to the sheet named **ClassInstantiation.**
2. Delete sample data leaving the column header row. In the **TargetEquipment** column, enter **Company\Foxboro bakery\North\Baking line** in the first cell. Then, grab the lower right hand corner of the cell (where the cursor changes to black crosshairs), and drag down three cells, so that all four rows have the same string.
3. In the **InstanceName** column, enter **Charger1**. Using the same click-and-drag trick, copy this down three more rows. The digit should automatically change going down the list; if it does not, manually change the cells, so that they read **Charger1, Charger2, Charger3** and **Charger4**.
   - The **InstanceName** cell provides the name for the individual asset. The **TargetEquipment** cell provides the path above that asset in the tree.
4. In the **EquipmentClass** column, enter **Ingredient charger** for all 4 rows.
5. **Save** the spreadsheet.

6B.4. Editing Properties

1. In the **Workbench,** drill down **AssetWorX > Equipment classes > Ingredient charger.**
   - If **Ingredient charger** is missing, **right-click** on Equipment classes and select **Add equipment class**. Name this new equipment class "**Ingredient charger**" and click **Apply.**
2. Right-click **Ingredient charger** and select **Add Equipment Class**.

3. Name this equipment class **Ingredient selector**. Set the **Level** to **Control Module**.

4. Right-click **Ingredient selector** and select **Add Equipment**.

5. Name this property **"MaterialTemperature"**. Change the **Source Type** to **Static Value**. Change the **Data Type** to **String**, and fill in the **Value** field with **/?Temperature?/** and then **Apply** the changes.
   NOTE: The /?string?/ syntax is used to create an alias. Aliasing is covered in-depth elsewhere in this course; here, we are using it to create a "fill-in-the-blank" within this Equipment class property. We will use the Bulk Asset Configurator to fill in that blank with a specific value for each piece of equipment.

6. In the **Excel** workbook → **ClassInstantiation** sheet, rename the column "**Parameter 1 Name**" to **Temperature** to match the value set for the MaterialTemperature Equipment class. Then input the following values for the respective pieces of equipment:

   - Charger1: 58
   - Charger2: 42
   - Charger3: 38
   - Charger4: 100

7. **Save** the spreadsheet.

6B.5. Adding Historical Tags

1. In the **Excel** workbook, at the bottom of the window, go to the sheet named **ClassHistoryDefinitions**. Delete the sample data leaving
2. In the first row, fill in the following columns with the following values (notice that not every column will be filled). Be sure to input these strings exactly, as they will match precisely the fields in the Hyper Historian tag configurator:

   - **EquipmentClass**: Ingredient charger
   - **Property**: Ingredient selector\MaterialTemperature
   - **GenerateHistory**: TRUE
   - **IsCollected**: TRUE
   - **LoggingGroup**: Sample Logging Group
   - **CollectorGroup**: Collector Group (1 second)

3. Delete any rows in this sheet which have values of "**C**" or "**CC**" in the **HistoryTagType** column.

    - This sample Bulk Asset Configurator sheet is also designed for use with the full Hyper Historian, which supports Conditional Tags. Because we are using Hyper Historian Express, as part of GENESIS64, Conditional Tags are not supported, which means we need to remove references to the unsupported tag types. Otherwise, the Bulk Asset Configurator will not execute.

4. **Save** the spreadsheet.

6B.6. Adding Alarms

1. In the **Excel** workbook, at the bottom of the window, go to the sheet named **ClassAlarmDefinitions.**

2. In the first row, fill in the following columns with the following values (notice that not every column will be filled). Be sure to input these strings exactly, as they will match precisely the fields in the AlarmWorX64 tag configurator:

    - **EquipmentClass**: Ingredient charger
    - **Property**: Ingredient selector\MaterialTemperature
    - **GenerateAlarm**: TRUE
    - **Enabled**: 1
    - **LimitOrDigital(L/D):** L
    - **HiLimit**: 50
    - **HiMsgText**: Temp High
    - **HiSeverity**: 500
    - **HiRequiresAck:** 1

3. **Save** the spreadsheet.

6B.7. Executing the Bulk Configuration

1. In the **Bulk Asset Configurator**, leave the configurations in the General, Database connections and Tag prefixes section set to their defaults.

2. Under **Options**, **check** the boxes for **Create new items** and **Update existing items** if they are not already checked.

3. Be sure to **uncheck** the box for **Delete nonexistent items**. In general, this is a good habit because it ensures that the entire asset configuration isn't lost when rolling out small bulk configurations.

4. For this first run, **check** the box for **Simulate instantiation (rollback changes before completing)**. This ensures that the bulk instantiation can be checked for errors before actually being implemented.

5. Select the **Home** tab of the **Bulk Asset Configurator**, click **Execute** in the top left.

6. As the configurator executes, monitor its progress in the log. Confirm that the configurator isn't doing anything it shouldn't. For example, if it reports deleting assets, be sure to **uncheck** the **Delete nonexistent items** box once it's done. If there are errors, ensure that there are no typos in the appropriate sections of the spreadsheets. The configurator will not successfully execute if there are errors. **Close** the log when finished.

7. If the execution ran without errors, **uncheck** the box for **Simulate instantiation (rollback changes before completing)** and click **Execute** again.

8. Drill into specific assets, such as **Company → Foxboro bakery → North → Baking line → Ingredient charger → Charger1**, and confirm the**Temperature** values you used in the Excel spreadsheet, **58** for Charger1, **42**, **38**, and **100** for Charger2, 3, and 4 respectively.

This concludes this lab.

GraphWorX64 Tools 7

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 uses controls to retrieve data.

7.1 Objectives

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.

7.2 GENESIS64 Controls

If you are familiar with GENESIS32, you should be familiar with ActiveX. GENESIS64 no longer uses ActiveX. Instead, there are 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.

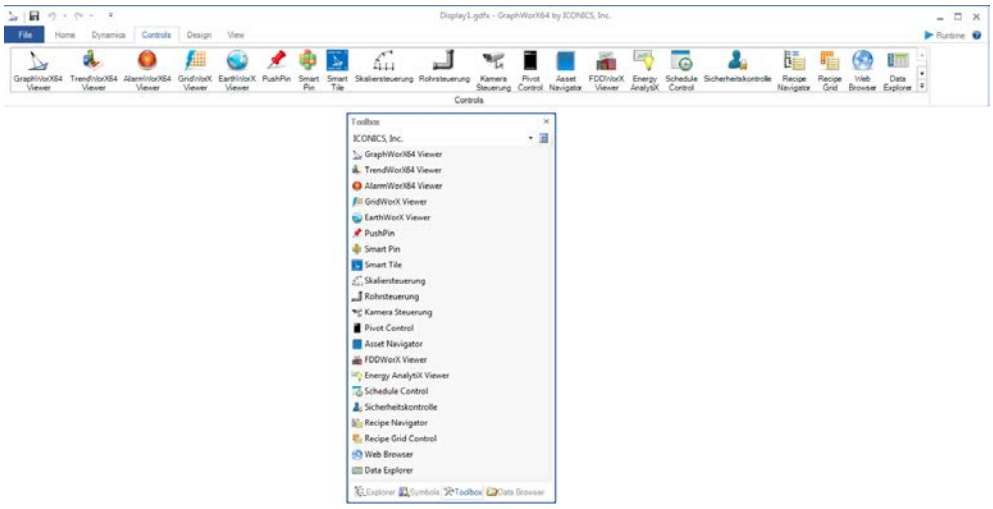


Figure 7.1 - Controls Ribbon (Above) and the Toolbox (Below)

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.

7.2.1 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 control, you can include a separate GraphWorX64 display inside the current display you are configuring. By adding a GridWorX64 Viewer, you can utilize GridWorX64 and create a grid inside the current display. By adding an EarthWorX64 Viewer, you can utilize EarthWorX64 and create a map inside the current 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.

7.2.2 EarthWorX

EarthWorX64 is a GENESIS64 provider that incorporates high-resolution maps into a GraphWorX64 display. An EarthWorX64 map can be added to any GraphWorX64 display as an added layer. You do this by adding an EarthWorX64 Viewer to the display from the GraphWorX64 Controls ribbon, which opens a map to which you can then add useful GraphWorX64 features. Using an EarthWorX64 map is helpful for monitoring geographically disbursed sites, not only on a global scale, but also on a local scale. Some of the features you can take advantage of include:

- Using a worldwide map or a local, regional map.
- Using SmartPins for reporting statuses and warning conditions at a glance.
- Using PushPins that mark significant points and their statuses on the map.
- Creating views that let runtime users "fly" over long distances.
- Displaying actual land features on your maps
- Confining or expanding the zoom level of the map.

- Configuring Pick actions that let runtime users act based on conditions that they see unfolding before them
- Importing GPS tracks and routes from a .GPX file.
- Monitoring logistical traffic using GPS input
- Using GraphWorX64 features as part of your interactive map, including shapes, 3D views, and layers
- Access to multiple Map Systems

7.2.2.1 Adding EarthWorX64 Control to a Display

To start using EarthWorX64, you will first need to launch GraphWorX64. Once the new display is ready to work with, go to the "Controls" ribbon and select the "EarthWorX64 Viewer" control to add it to the display.

Figure 7.2 - Adding the EarthWorX Viewer Control to a Display

Alternatively, you can also use the Toolbox to access the same functionality.

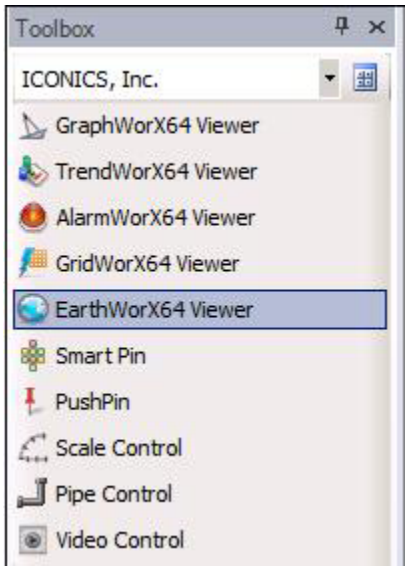


Figure 7.3 - Accessing EarthWorX64 from the Toolbox

Once the control has been added to the display, a warning message pops up to inform you that the display is not using the default EarthWorX64 size and will provide the choice to automatically resize the display. The reason for this choice is due to the EarthWorX64 world map having a default size of 512x512 pixels at its minimum zoom level. If your display is bigger than that, then you might end up having white areas around the map. However, it should be noted that if the final map will be zoomed in to a particular region or country, you can use another display size without any white space around your selected map.

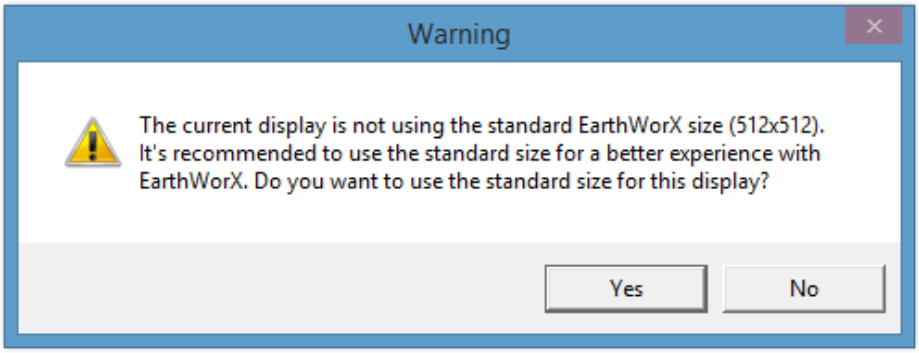


Figure 7.4 - Warning Message about Display Size

This example will use a default size display; therefore, select **Yes** to the above question. Once the display is resized, another Warning will come up.

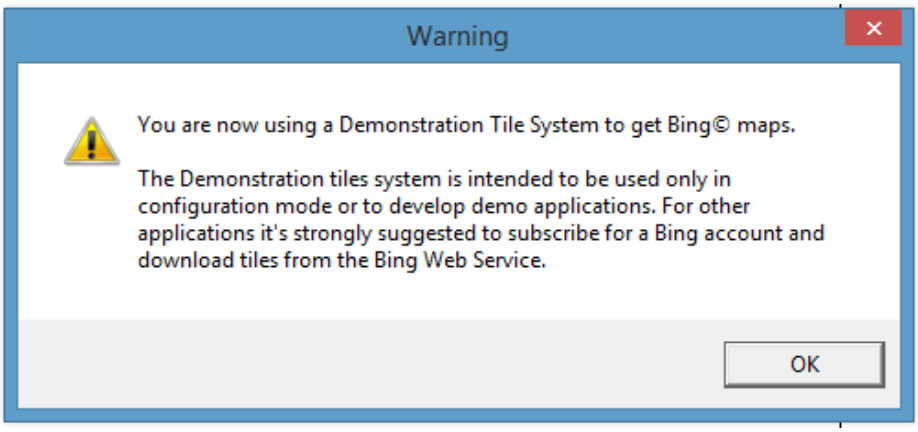


Figure 7.5 - Warning Informing about Bing Maps License Restriction

The above warning message informs you that you are now using the demo version of Bing Maps. Microsoft allows you to use the maps with some limitations. However, for real applications, you should obtain a valid license from Microsoft. For the purpose of this example, you can continue using the demo tiles to evaluate the functionality of the EarthWorX64 control. After you click **OK**, the display will contain the EarthWorX64 map and you are ready to start working with it.

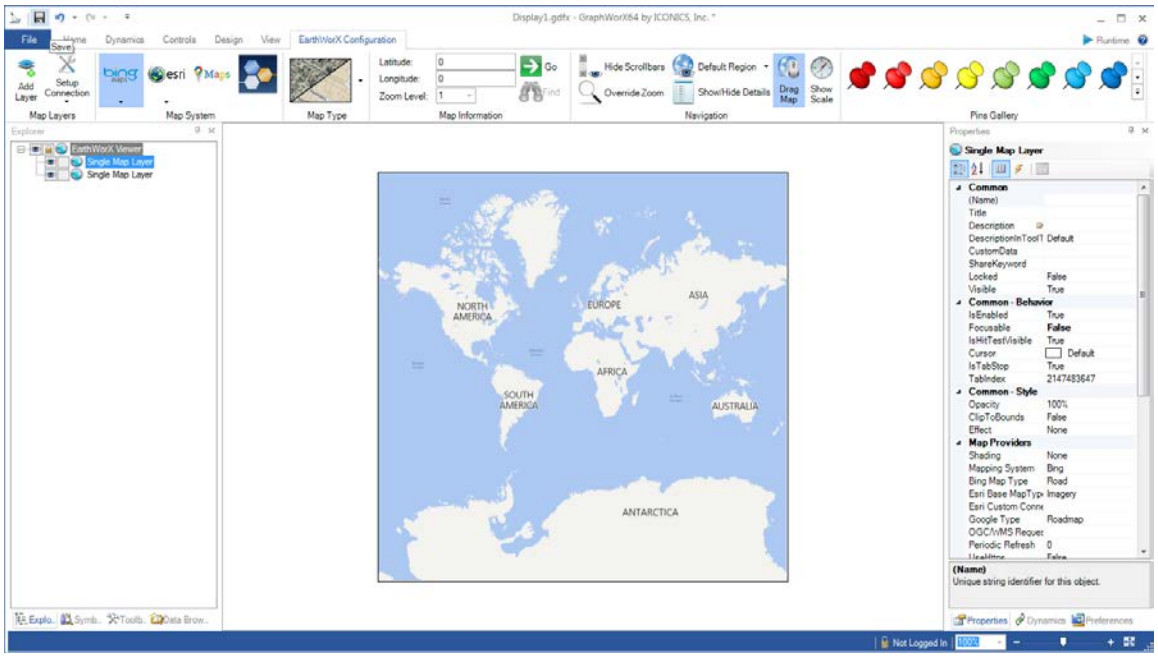


Figure 7.6 - EarthWorX64 Map in the Display

Start the map interaction by zooming in on it as you would do in a regular GraphWorX64 display. The map will load additional details as needed while zooming and panning over it. Alternatively, you can also use the commands provided in the EarthWorX64 Configuration ribbon to perform other operations, including changing the map type, jumping to a specific location, import additional data, adding pushpins and much more. To change the map type, click on the "Map Type" button in the "EarthWorX Configuration" ribbon, and select your preferred map type.

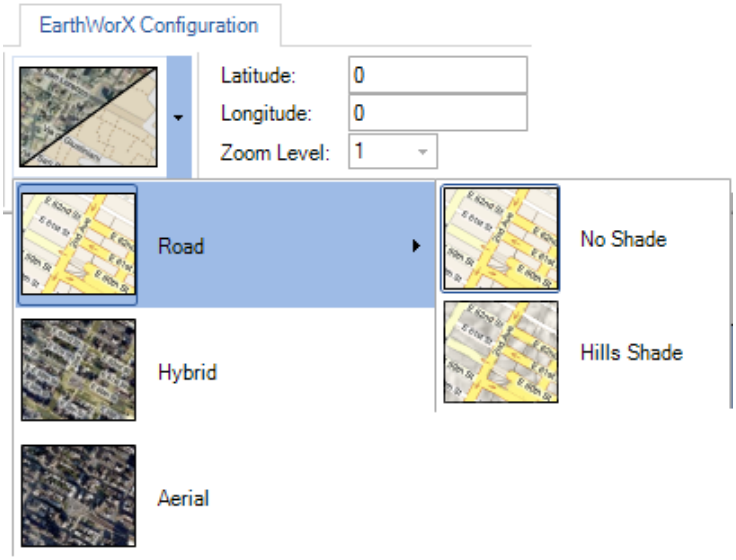


Figure 7.7 Available Map Types

7.2.2.2 Adding Objects to the Map

Generally speaking, you use EarthWorX64 to keep track of geographically dispersed assets, normally by adding objects on the map to represent those assets.

Adding a Pushpin

In this example, you will add a Pushpin to the map. [See the following section for more information on PushPins.] To do this, click on one of the Pushpin icons in the "EarthWorX64 Configuration" ribbon in the "Pins Gallery" section.

Figure 7.8 - Pins Gallery

After clicking on a Pushpin icons, click on a location on the map. In the example below, the user clicked on Rome, Italy, to position the Pushpin in that location; this adds a new object over the map. You can easily relocate the object by dragging it around, and you can resize it, as needed. For more help, refer to the PushPins topic.

Figure 7.9 - A Pushpin on the EarthWorX64 Map

The pushpin's Properties tab shows several properties for the object, which you can change as required.

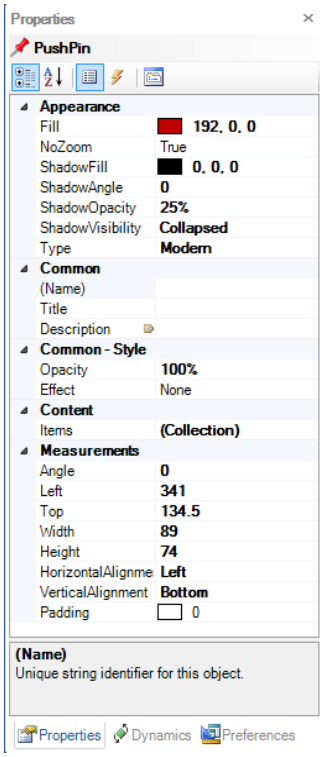


Figure 7.10 – Properties tab for a Pushpin

Adding Polygons

Pushpins are not the only objects you can put on a map. They are basic shapes that are easy to access, and for that reason may be suitable for many applications. But you may need to use other shapes to achieve what you're trying to do with EarthWorX64.

An alternative to Pushpins would be to highlight a specific geographic area and change its color depending on some values, for example to monitor your company sales activity in a region and to change the color if sales meet the expected target.

To do this, select the Polygon icon from the Shapes box in the “Home” ribbon, highlighted in the following figure.

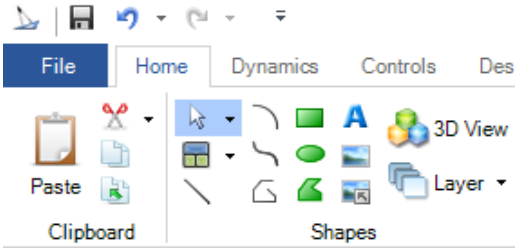


Figure 7.11 - Selected the Polygon Command

The mouse cursor will change to indicate the polygon drawing mode. Now, click on the points of the map that represent the border of the desired region. In this example, the border of the state of Massachusetts has been drawn.

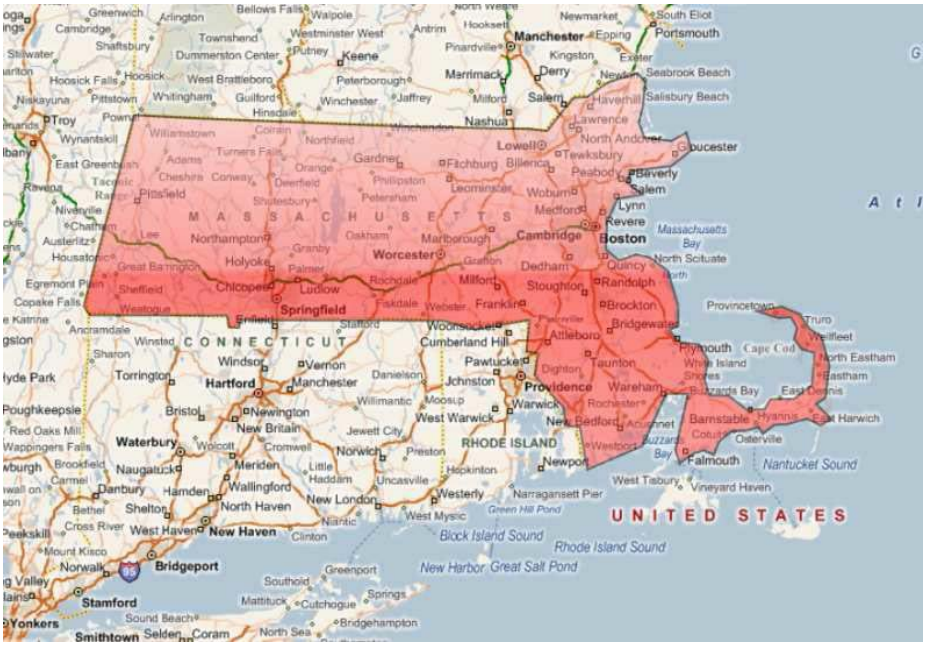


Figure 7.12 - A Polygon Drawn on the Map

The polygon object is a regular GraphWorX64 object. You can add dynamics or a pick action to it, for example to change its color depending on a data value or expression, or add a pick field to open a new display with that area's details when the polygon is clicked. Using the same method you can create additional polygons to enclose two other U.S. states as shown in the following example to complete this example. For additional information, refer to the Dynamics topic, as well as the Shapes topic.

Figure 7.13 - Some More Polygons Added

7.2.3 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 7.14), 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.

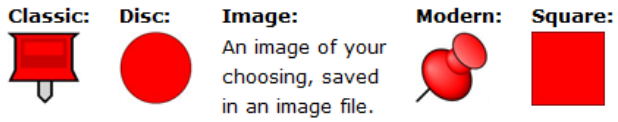


Figure 7.14 - A Few PushPin Shapes

7.2.4 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.

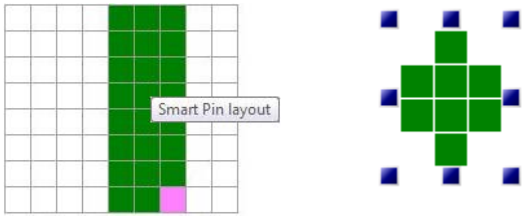


Figure 7.15 - Smart Pin Configuration Layout (Left) and a Selected Smart Pin in a Display

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.

7.2.5 Smart Tiles

Smart Tiles are useful, at-a-glance graphic visualization tools that provide real-time information similar to Live Tile technology found in Microsoft Windows 10, Windows 8, Windows RT, and Windows Phone operating systems. The difference is that ICONICS Smart Tiles, created through the Smart Tile control in GraphWorX64, are applicable within HMI screens, whether those are PC-based or for mobile devices (phones, tablets, etc.).

Figure 7.16 - Smart Tile Examples

Smart Tile Features Include:

- Shows Live Process Values
- Use in GraphWorX64 and the Mobile-based AppHub - In GraphWorX64, Smart Tiles use/share GraphWorX64's existing data management components to get runtime data and handle updates. In the AppHub, Smart Tiles instantiate their own shared data management components. The data management components are disposed when all instantiated smart tiles are unloaded or disposed.
- Easy Configuration and Flexibility
- Lightweight Tool - when needing to show lots of tiles simultaneously
- Can Run in Windows Presentation Foundation (WPF), Silverlight, Windows RT or in HTML5

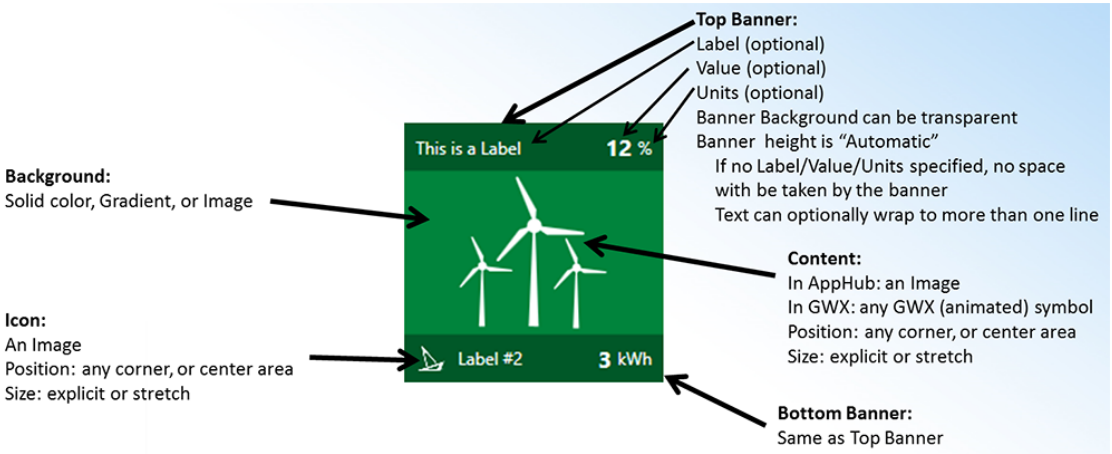


Figure 7.17 - Smart Tile Example with Area Descriptions

You can open a new display in GraphWorX64, then select the Smart Tile control from the 'Controls' ribbon, as shown below.

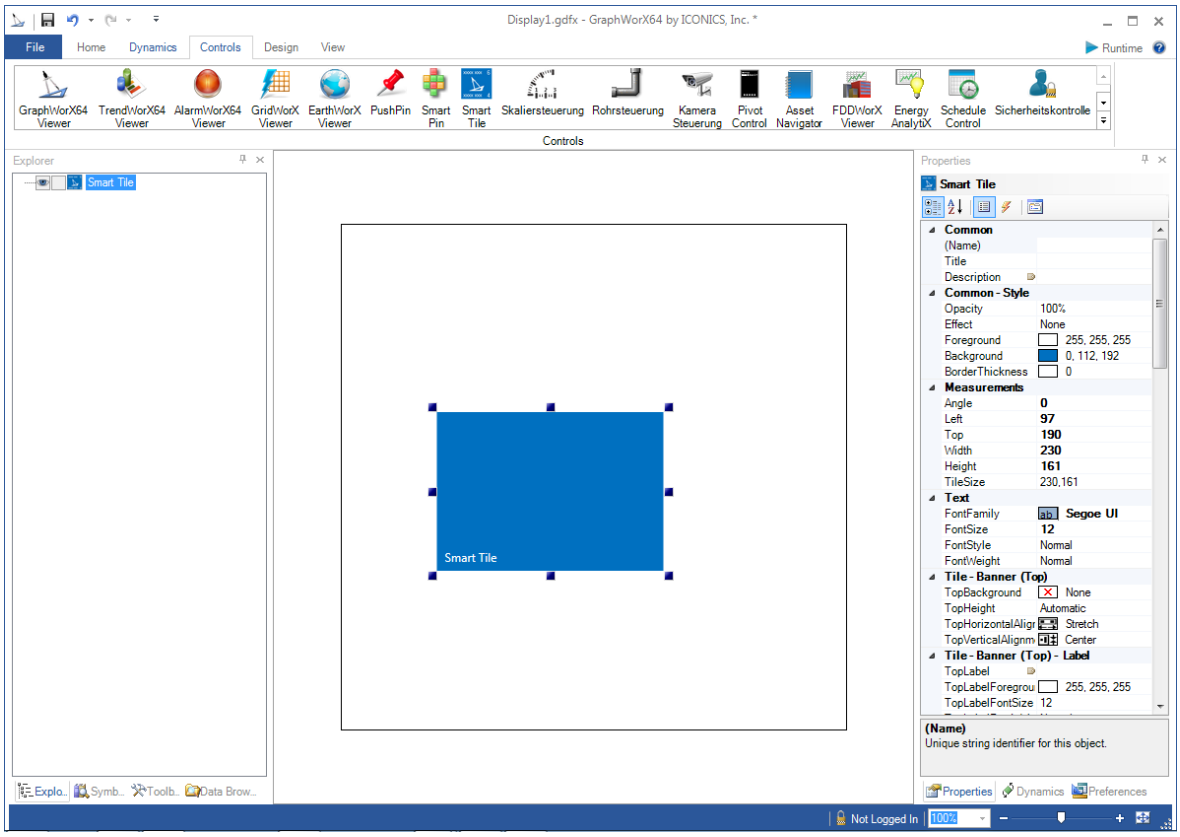


Figure 7.18 - Creating a New Smart Tile Within GraphWorX64

A default Smart Tile shape will appear in the GraphWorX64 canvas. Configuration can be done either through GraphWorX64's Properties grid or by double-clicking the Smart Tile to access the popup Smart Tile Configuration dialog, shown in the following figure. The Smart Tile Configuration dialog has several tabs, including:

- General
- Top Banner
- Bottom Banner
- Center Value
- Icon
- Content
- States

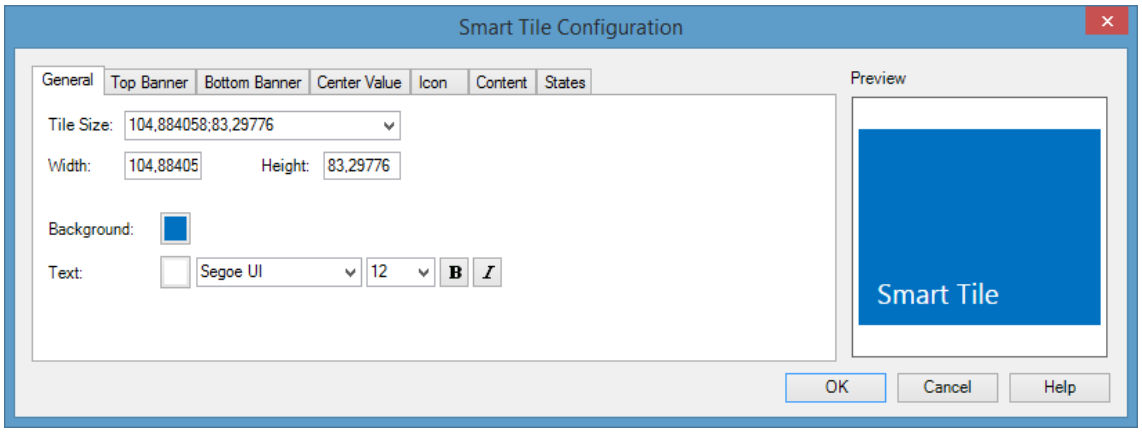


Figure 7.19 - Smart Tile Configuration Dialog

Once you have finished configuring the Smart Tile, it will be added to your GraphWorX64 display canvas.

7.2.6 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.

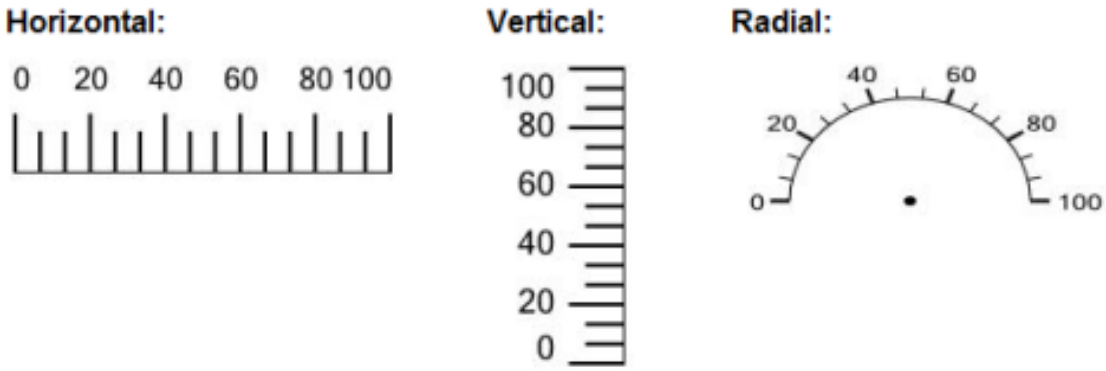


Figure 7.20 - The Scale Control

7.2.7 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.

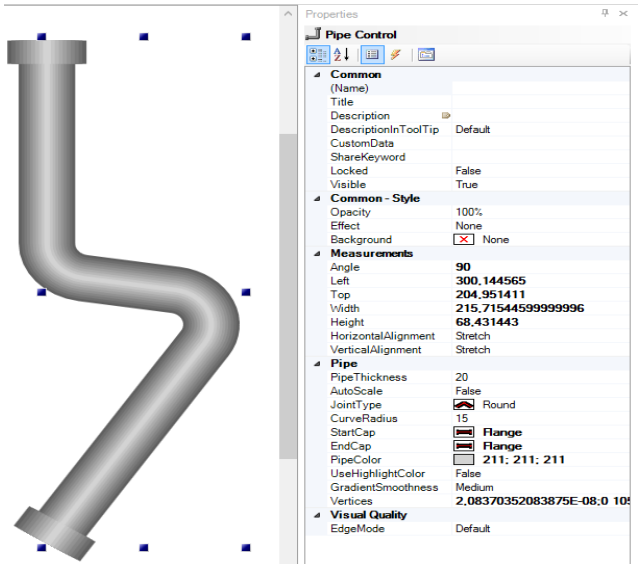


Figure 7.21 - The Pipe Control

7.2.8 Camera Control

The Camera Control in the GraphWorX64 viewer allows users to include output from a networked camera within their display. To use this item, select the **Camera Control** icon from the **Controls Ribbon**. Your cursor will now become a cross hair. Use the cross hair to draw the perimeter of the **Camera Control** on the display. Your Camera Control will now be visible.

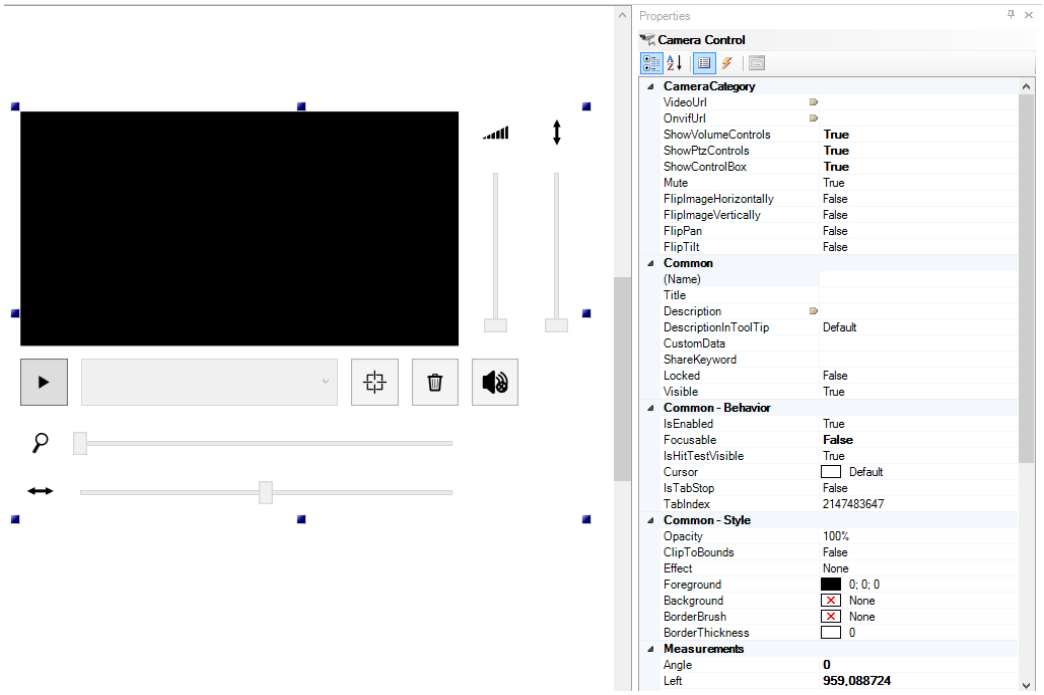


Figure 7.22 - Camera Control With All Options Enabled

The right side of the screen will now display the **Camera Control Properties**. Camera Controls have a number of standard properties such as Measurements and Text, but also have a number of unique properties listed under Camera Category, as shown in the previous figure.

CameraCategory

- **VideoURL** - The Video Source URL. This must point to a RTSP or MJPG stream.
- **OnvifUrl** - This must point to a URL where the ONVIF on the camera is listening.
- **ShowVolumeControls** - In Runtime mode, 'True' means the volume controls will be shown next to the video feed.
- **ShowPtzControls** - In Runtime mode, 'True' means the Pan/Tilt/Zoom controls will be visible with the video feed.
- **Mute** - In Runtime mode, 'True' means the audio from the control will be muted by default.
- **ShowControlBox** - In Runtime mode, 'True' means the playback controls will be shown under the video feed.

7.2.8.1 Camera Control in Runtime

Figure 7.23 - Camera Control in Runtime

In your GraphWorX64 display utilizing a Camera Control, once put into Runtime Mode, you will see the live feed from the VideoURL you entered in the control's properties in Configuration mode. Also, if you set the additional properties to 'True', you will see those specific settings around the video feed, as shown above. The ability to manipulate the various sliders, pulldown menus and buttons will correspond to the camera model/network at the location set via the VideoURL setting.

7.2.9 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.

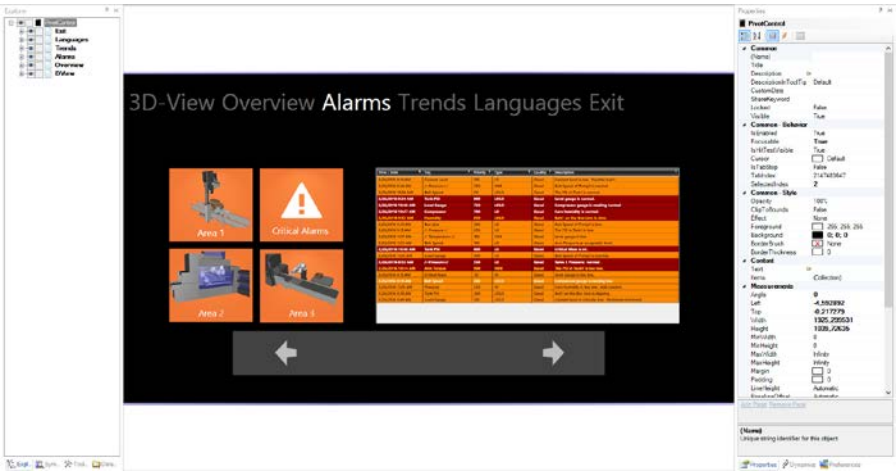


Figure 7.24 - Editing 'Page 1' within the Pivot Control

7.2.10 AssetWorX Navigator

The Asset Navigator Control in GraphWorX64 ties to the AssetWorX provider in the Workbench, which allows users to create a tree-based representation of an organization's assets in order to visualize data, perform commands and analysis and more. It can be used in conjunction with additional controls (including AlarmWorX64 Viewer, Energy AnalytiX Viewer, FDDWorX Viewer, GridWorX Viewer, GraphWorX64 Viewer, TrendWorX64 Viewer, etc.) within the same GraphWorX64 display to allow users to perform asset-based commands automatically.

To Add an Asset Navigator Control in GraphWorX64

1. In GraphWorX64, click on the Controls ribbon and select the Asset Navigator Control.

2. Draw a box that will hold the connected asset tree.

Tip: A vertical box will make things easier later on as compared to a more horizontal one.

The Asset Navigator control will show a default tree, but not from the default database within the AssetWorX provider in the Workbench. You'll notice the distinction in that the top level of the tree is labeled as "Root" with a sub-item as "Company" and a further sub-item as "Factory". You will need to connect to an active AssetWorX database in order to utilize any configured data in any additional control you wish to add to this same GraphWorX64 display.

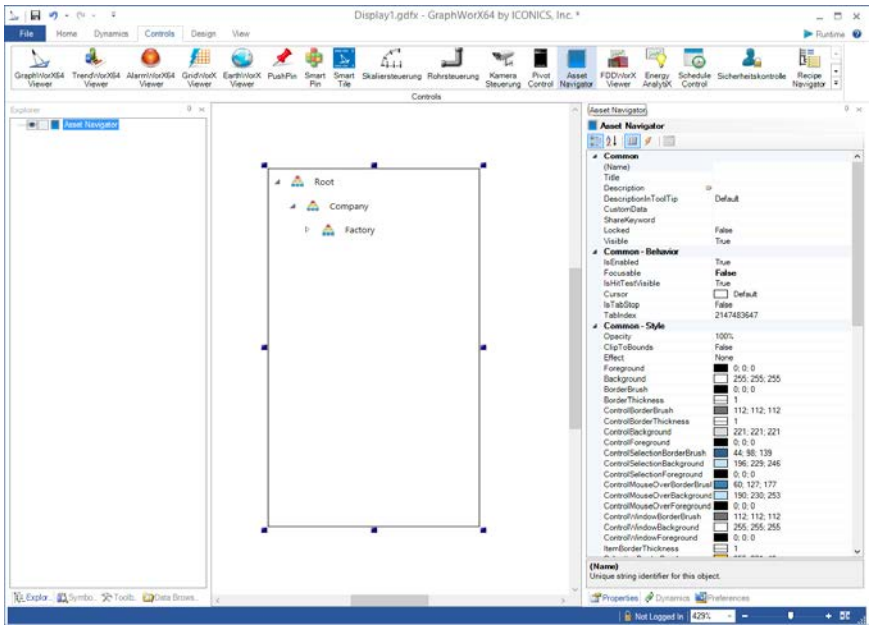


Figure 7.25 - Adding an Asset Navigator Control in GraphWorX64

7.2.11 Enabling Commands through the Asset Navigator Control in GraphWorX64

The latest version of GENESIS64/GraphWorX64 allows you to run commands from a tree navigation control into other controls in the display. This is done by utilizing the asset catalog structure set up through the AssetWorX provider in the Workbench.

The following is an example of how to utilize an Asset Navigator Control within a GraphWorX64 display to initiate a command for an additional control within the same display (in this example, a GraphWorX64 Viewer).

1. Open GraphWorX64. From the Start menu > All Programs > ICONICS > GENESIS64 > GraphWorX64 > GraphWorX64.

2. Select the Asset Navigator control from the Controls ribbon, as shown below.

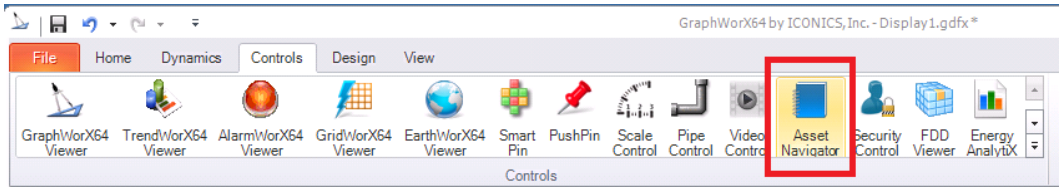


Figure 7.26 -Selecting the Asset Navigator Control in GraphWorX64

3. Draw a box in which to place the Asset Navigator control. Since a tree navigation control typically expands vertically, a more vertical box is appropriate. Once you've placed the control, you'll notice the asset tree has branch placeholders including "Root", "Company" and "Factory". These will be replaced during runtime if you have previously set up an asset database within the AssetWorX provider in the Workbench. For more information on how to accomplish this, see the GENESIS64 Silverlight Help included with this GENESIS64 installation.

4. Next, select the GraphWorX64 Viewer control and draw another box in which to place that control. Your GraphWorX64 display should closely resemble the following image.

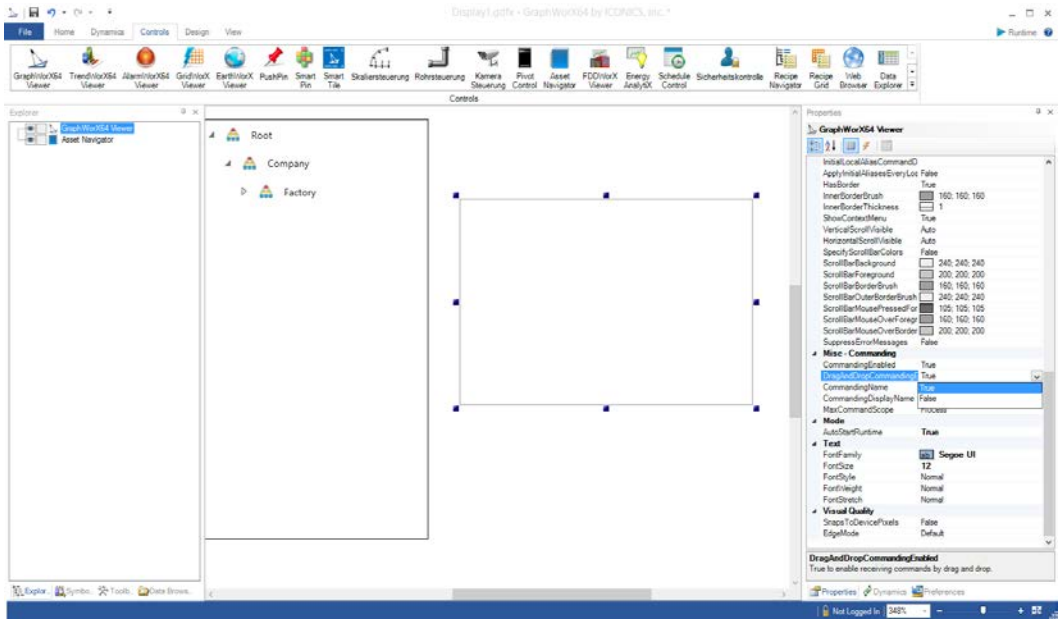


Figure 7.27 - Asset Navigator and GraphWorX64 Viewer Controls Added

5. At any point prior to including these two controls within a GraphWorX64 display, you should have already created an asset catalog database within the AssetWorX provider in the Workbench.

The image below shows the AssetWorX provider within Workbench. The host machine, 'localhost', contains our asset catalog database (also named 'AssetWorX'). For an example, we have created a sample asset that we've named 'Test Asset' (highlighted in the red box on the left).

When you double-click on the asset (in our example, 'Test Asset') its properties will appear in the form on the right. Click on the 'Commands' tab (highlighted in the red box in the top middle) to set a specific command.

In our specific example, we will select the 'Load Graphics Display' from the Command pulldown menu. We've also named the command as 'Load Display' so as to recognize this specific command selection during runtime later on. We've made sure its 'Visible' as well as enabled its 'Drag & drop' ability. We want to make sure that it can be used in GraphWorX64 (as well as the remaining selections). For this example, we will not use any aliases (Local or Global).

For the FileName, we will provide the location of a newly created GraphWorX64 display (in this instance, it is a MobileHMI display [.gdfxp file]).

For Target, we will enter what the GraphWorX64 Viewer control name will be once we return to the GraphWorX64 display we started creating (for our example, we've named it "Load BakingOven.gdfxp Display").

Click 'Apply' or 'Apply & Close' to save these settings.

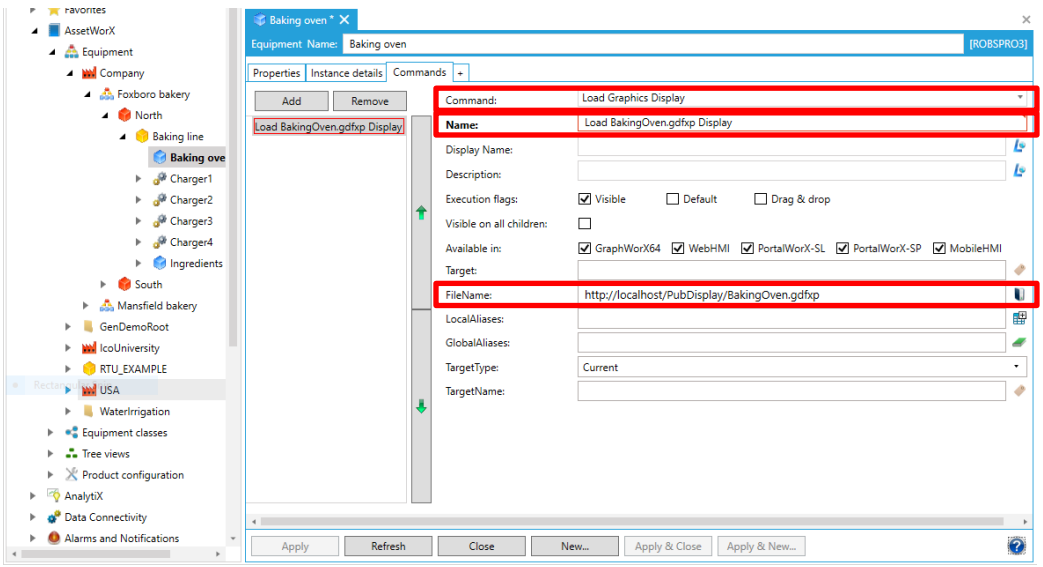


Figure 7.28 - Setting Commands Properties for an Asset in the AssetWorX Provider

6. Returning to our GraphWorX64 display, since we wanted to utilize Drag & Drop, we need to ensure each Control is set to use it. Click on the Asset Navigator control's box to select it, then edit the 'DragAndDropEnabled" field in the 'Misc' section of the Properties to the right of the screen, by setting the pulldown to 'True'.

7. Next, click on the GraphWorX64 Viewer control to select it, then edit the DragandDropCommandingEnabled field in the Misc - Commanding section of the Properties to the right of the screen, by setting the pulldown to 'True' (as shown below).

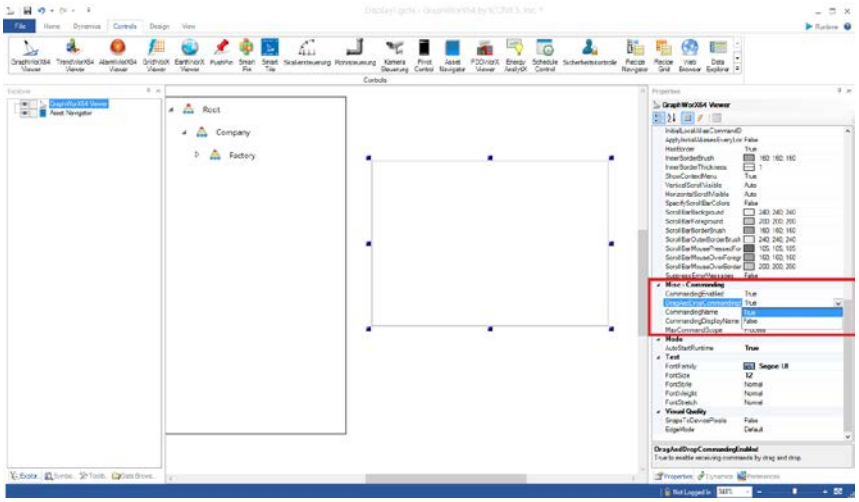


Figure 7.29 - Enabling Drag and Drop for the GraphWorX64 Viewer Control

8. At the same time, now that we have the GraphWorX64 Viewer control selected, we will give it the name we previously entered in the 'Target' field of the Commands tab of the properties for the 'Test Asset' in AssetWorX (in Step 5 above). We'll enter 'TestControlCommandTarget' in the Name field in the Common section of the Properties to the right of the screen, as shown below.

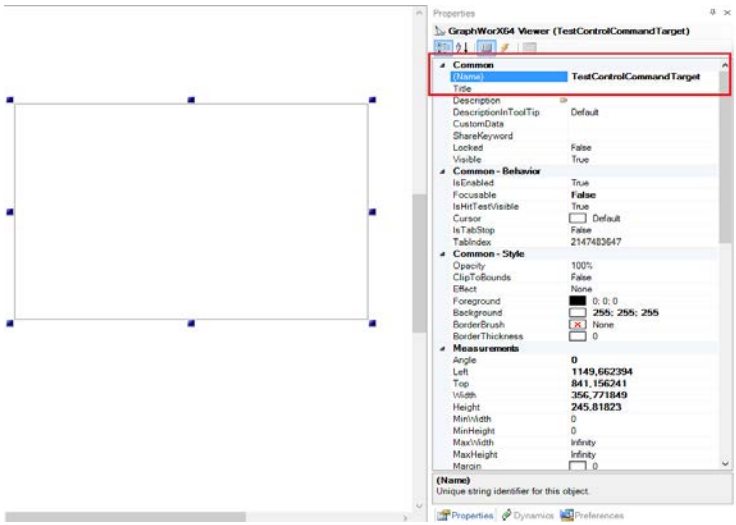


Figure 7.30 - Naming the GraphWorX64 Viewer Control

9. We can now put the display into Runtime (by either clicking on the Runtime button in the top right of the screen or using '**Ctrl+M**'). Our example display will resemble the following image. You'll notice that the Asset Navigator control no longer has the placeholder 'Root>Company>Factory' branches and that it now shows the top level of the asset catalog database you created in the AssetWorX provider in the Workbench or Workbench-SL (in our example, the top level is 'Assets'), as shown below. You'll also notice that the GraphWorX64 Viewer control is presently blank.

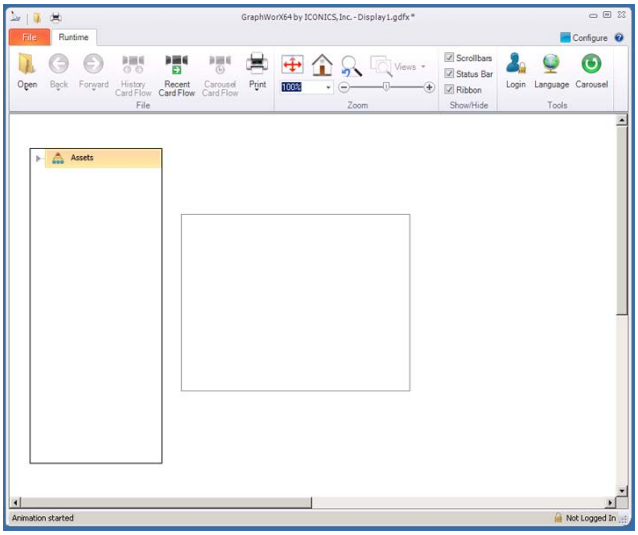


Figure 7.31 - Initial Runtime View of Asset Navigator and GraphWorX64 Viewer Controls

10. Expand the asset tree within the Asset Navigator control. You'll see the 'Test Asset' we created within the AssetWorX provider in the Workbench or Workbench-SL. Right-click on the 'Test Asset' branch and you'll see the 'Load Display' command we created and named, as well.

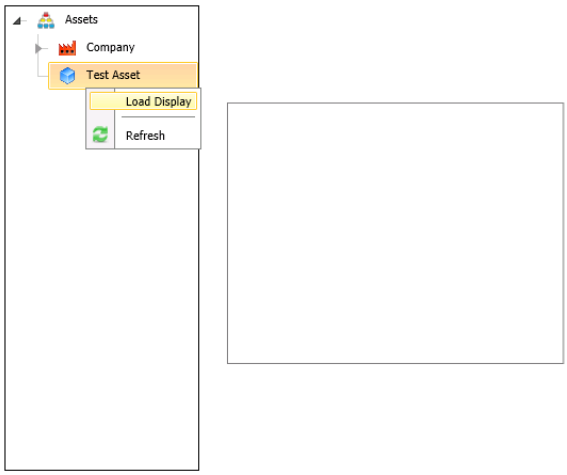


Figure 7.32 - Expanded Asset Tree in Asset Navigator with Associated Command

Since we also enabled 'Drag and Drop' for both controls, instead of right-clicking on the 'Test Asset', you could also left-click on its icon and drag it directly into the GraphWorX64 Viewer control. You'll see the cursor change to include a plus sign [+] to signify that the control will accept a dragged item.

11. The GraphWorX64 Viewer control will now show the successfully sent command from the Asset Navigator [which is the GraphWorX64 file you set in the 'FileName' field of the Commands tab of the properties for the 'Test Asset' in AssetWorX (in Step 5 above)] (see image below).

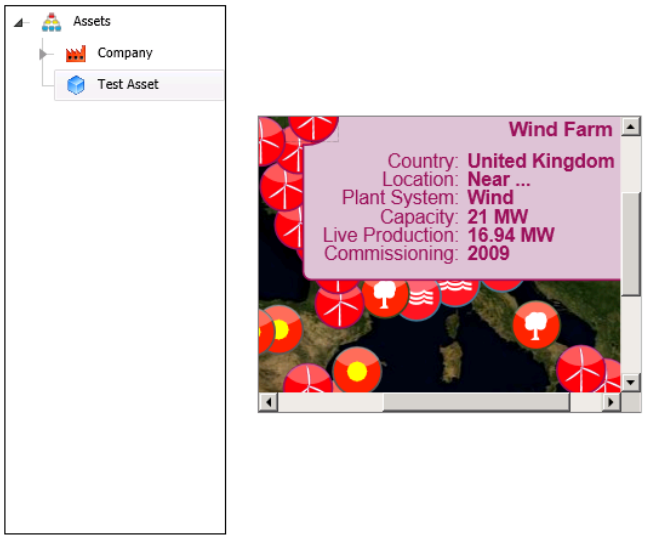


Figure 7.33 - GraphWorX64 Viewer Control with Successful Command Sent from Asset Navigator Control

7.2.12 FDD Viewer

The FDD Viewer allows you to configure and view a display containing information about faults that occur within the assets that you have configured in AssetWorX within the Workbench or Workbench-SL. With this display, you can detect and diagnose faults within your assets and identify strategies for reducing the chances of these faults occurring in the future.

The options on the FDD Configuration Ribbon allow you to customize the viewer. To customize how faults are detected and diagnosed within FDDWorX itself, however, you need to open the Workbench or Workbench-SL and navigate to the Facility AnalytiX control.

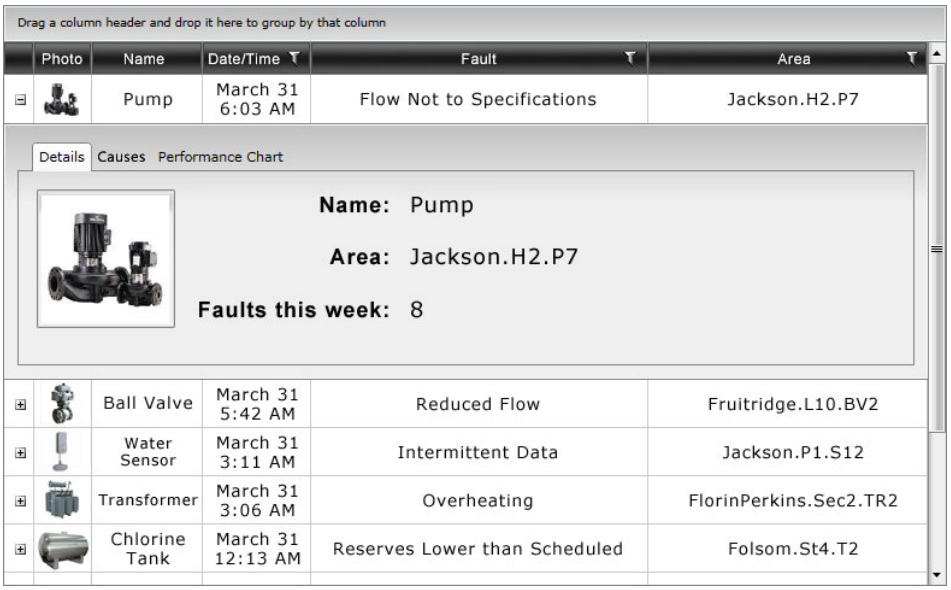


Figure 7.34 - FDD Viewer in Runtime in GraphWorX64

7.2.13 Energy AnalytiX Viewer

The Energy AnalytiX Viewer allows you to configure and view a display containing information about the energy consumption, energy-related costs, and carbon emissions of the assets that you have configured in AssetWorX within the Workbench or Workbench-SL. With this display, you can determine the energy performance of your assets and identify strategies for using more energy-efficient methods.

The options on the Energy AnalytiX Configuration Ribbon allow you to customize the viewer. To customize how energy consumption, costs, and carbon emissions are calculated within Energy AnalytiX itself, however, you need to open the Workbench or Workbench-SL and navigate to the Energy AnalytiX control.

7.2.14 Schedule Control

The Schedule Control allows users to insert a schedule into a GraphWorX64 2D display.

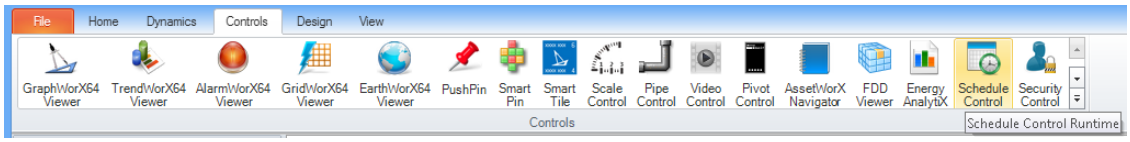


Figure 7.35 - Selecting the Schedule Control from the Controls Ribbon in GraphWorX64

After you have created a new display in GraphWorX64, you can insert a Schedule Control by selecting it from the Controls ribbon (as shown above) or from the Viewer Menu of the Toolbox.

Your cursor will turn into a '+', which allows you to set both the size of the control and its position within the display's canvas. The Schedule Control will then automatically size to the area you have created, as shown in the following image.

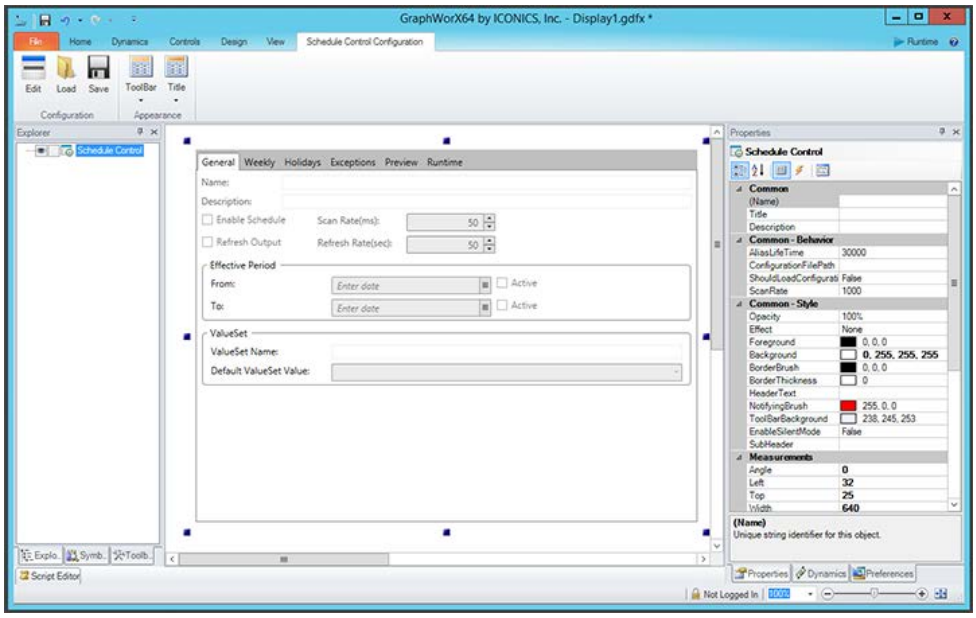


Figure 7.36 - A Newly Created Schedule Control in a GraphWorX64 Display Canvas

Note that the Schedule Control Configuration ribbon simultaneously appears at the top of the screen, as shown below.

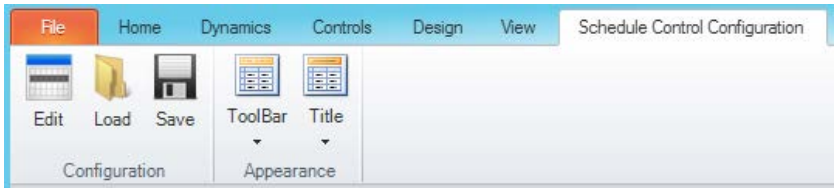


Figure 7.37 - Schedule Control Configuration Ribbon

The ribbon's buttons are used only in a Schedule Control that has been added to the GraphWorX64 display. Each button on the ribbon, as well as the configurator that the ribbon opens, is described below.

- The **Edit** button within the Configuration section of the Schedule Control Configuration ribbon opens the 'Configure your Control' dialog box, where you can add schedule views. (The Save button allows you to save an in-progress configuration and the Load button allows you to retrieve saved configurations.)
  In the configurator, you can add tabs to the control, and schedule views to each tab. This configuration dialog in the GraphWorX64 display configuration has the following components:

  - 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.

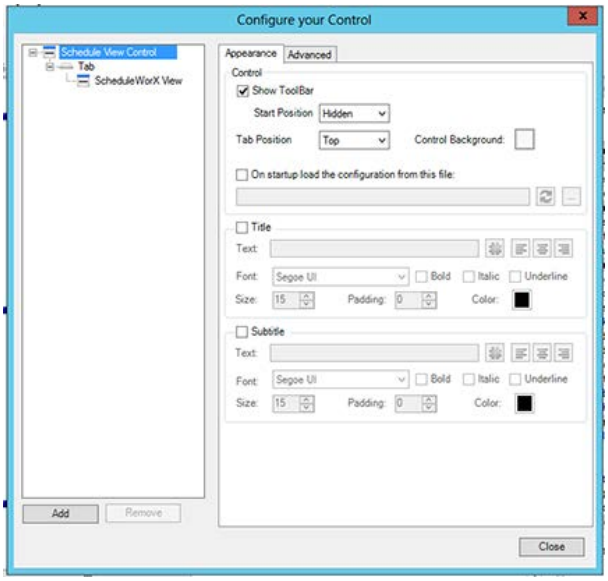


Figure 7.38 - Configure your Control Dialog Box - Appearance Tab

You can right-click on the 'Schedule View Control' item in the 'Configure your Control' dialog to add new Tabs - or you can select the 'Schedule View Control' and click on the 'Add' button in the bottom left.

- 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.

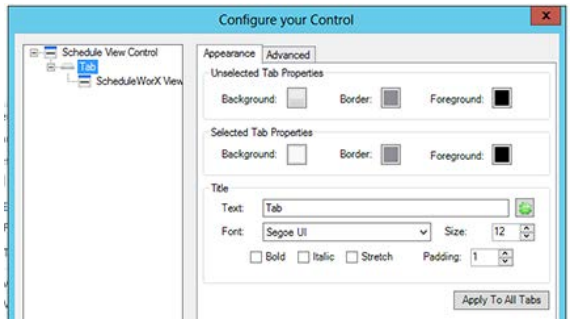


Figure 7.39 - Configuring Your Tab Settings

You can right-click on a Tab in the left-side navigation pane in the 'Configure your Control' dialog to add:

- a new **ScheduleWorX View**, or
- a new **BACnet Schedule View**, or
- a new **BACnet Calendar View**.

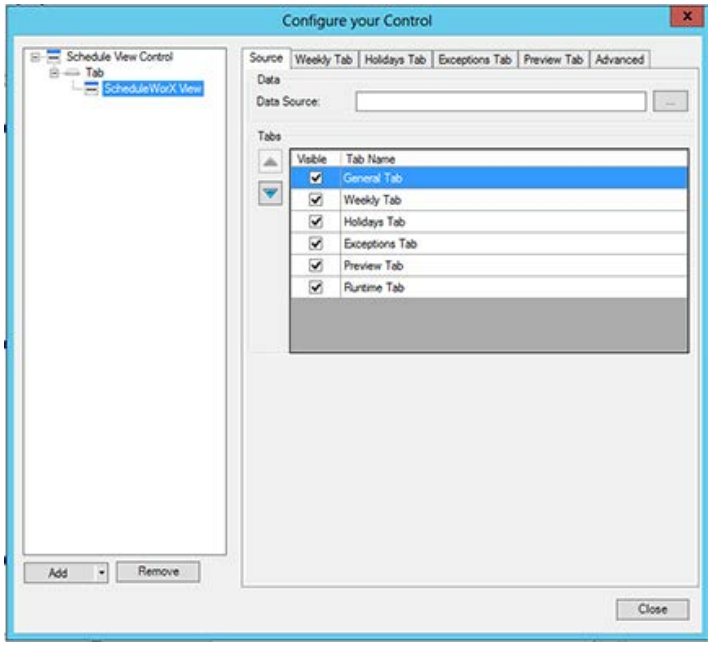


Figure 7.40 - Configure your Control Dialog Box - View Settings

When you have finished configuring your new **ScheduleWorX View**, **BACnet Schedule View** or **BACnet Calendar View**, you can see the results in Runtime, as shown in the following image.

Figure 7.41 - A Weekly Calendar View in Runtime

7.2.15 Security Control

The Security Control in the GraphWorX64 viewer is an embedded version of the Genesis64 Security Server. Creating a Security Control allows the user to password protect specific content in a viewer or control. A Security Control itself does not possess any secure properties without a **Security Server**.

To create a Security Control, select the **Security Control** icon from the **Controls Ribbon**.

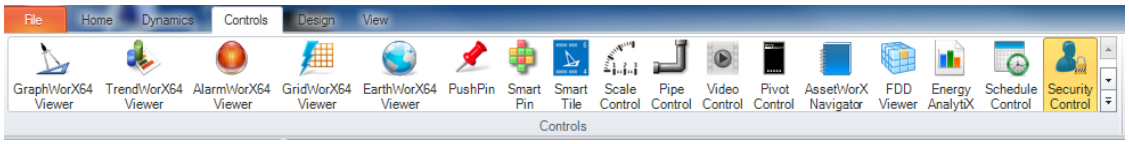


Figure 7.42 - GraphWorX64 Controls Ribbon

Your cursor will now become a cross hair. Use the cross hair to draw the perimeter of the **Security Control** on the display. Your Security Control will now be visible.

Figure 7.43 - Security Control

The right side of the screen will now display the **Security Control Properties**. Security Controls have a number of standard properties such as Measurements and Text, but also have a number of unique properties listed under Options.

Security Control Properties

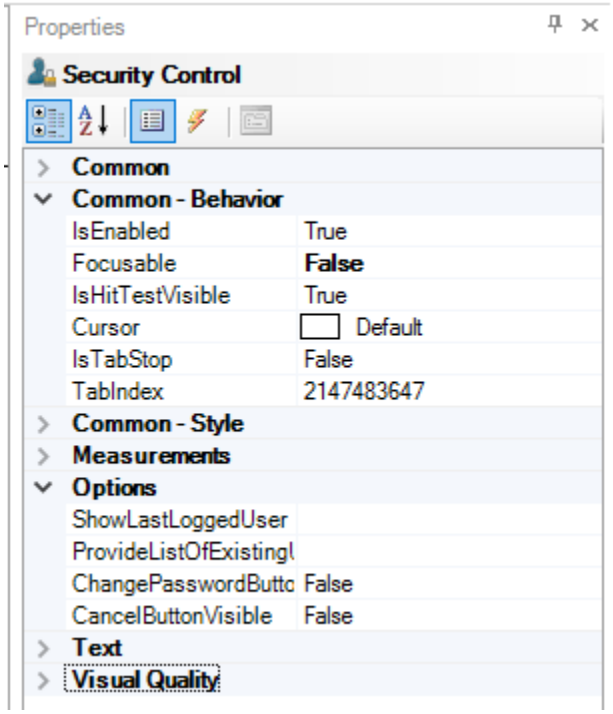


Figure 7.44 - Security Properties

Options

- **CancelButtonVisible** - Determines the visibility of the Cancel button. If visible, logging in or out will also automatically attempt to close the owner window.
- **ChangePasswordButtonVisible** - Determines the visibility of the Change Password button.
- **ProvideListOfExistingUsers** - Determines whether or not to show a list of all available user names in the drop-down field (no value to use the global settings).
- **ShowLastLoggedUser** - Determines whether or not to show the last logged in user in the User Name field (no value to use the global settings).

Figure 7.45 - Active Security Control within GraphWorX64 Display

7.2.16 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.

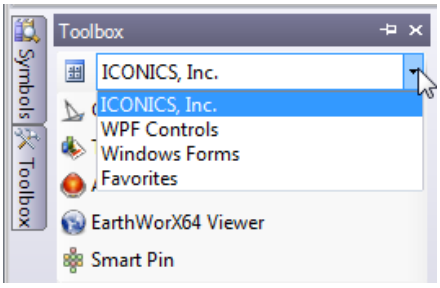


Figure 7.46 - Select Between Different Control Types

7.3 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.

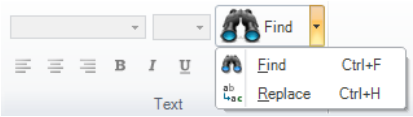


Figure 7.47 - Find/Replace

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.

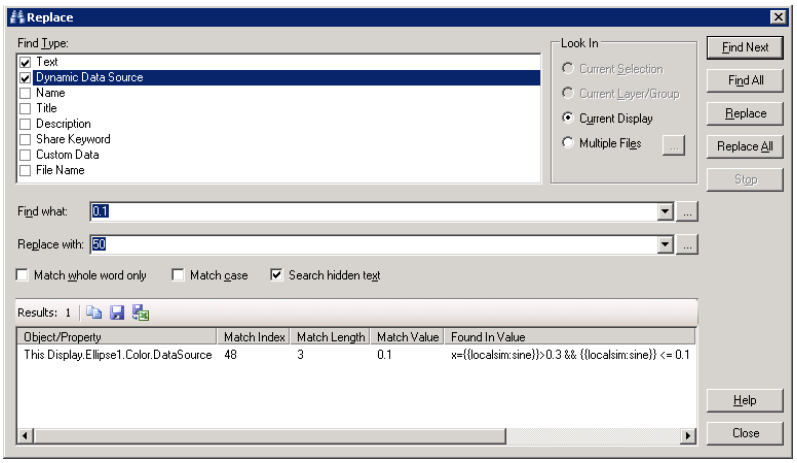
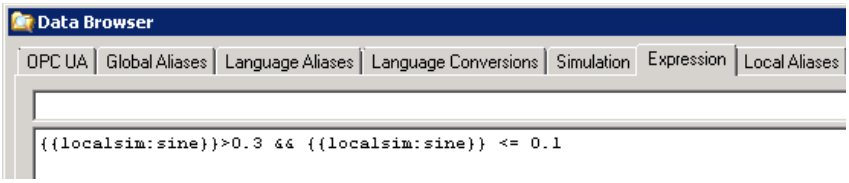


Figure 7.48 - Replace Dialog

Figure 7.49 shows an expression before and after the replacement illustrated in Figure 7.48. 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:

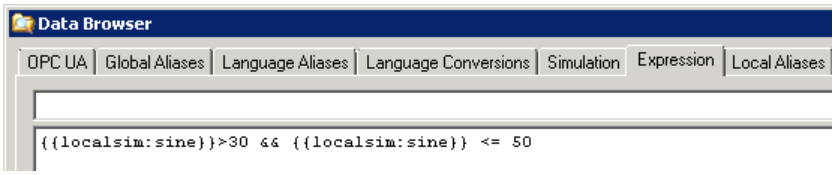


Figure 7.49 - Expression Before (Top) and After (Bottom) Replacement

7.4 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 7.50 are objects, too. They, too, can be arranged.

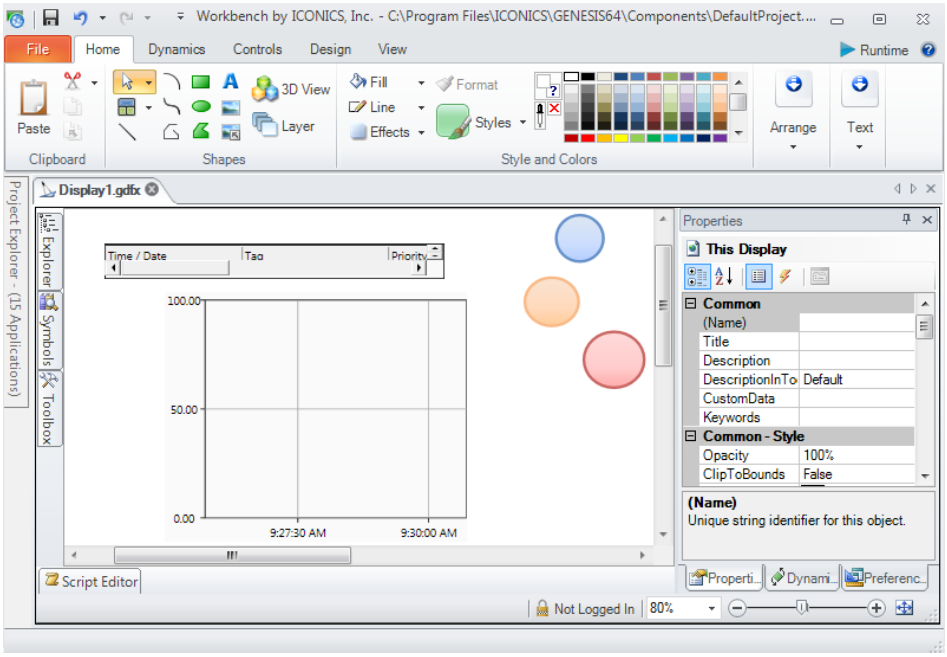


Figure 7.50 - A Display with Objects and Controls

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 7.51. All other objects are arranged in relation to the reference object.

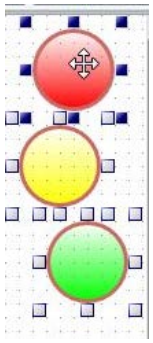


Figure 7.51 - The Top-Most Circle is the Reference Object

7.4.1 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.

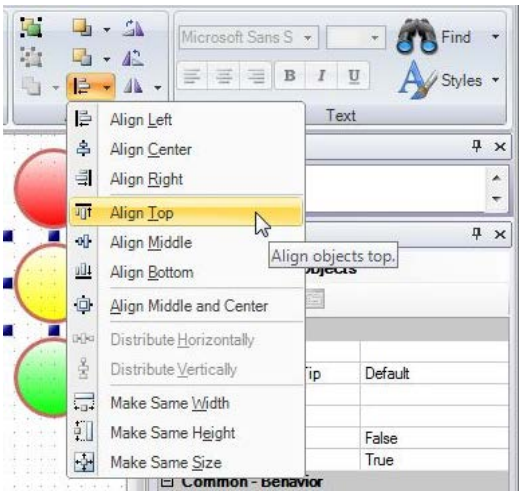


Figure 7.52 - Align Options

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

7.4.2 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 7.53 where the Rectangle object is on the bottom (or in back of the other objects).

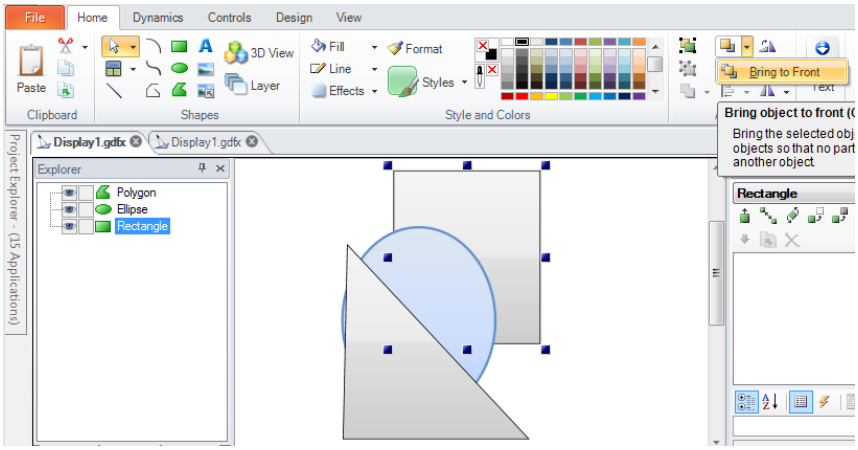


Figure 7.53 - Z-Order of Objects in a Layer

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 7.54.

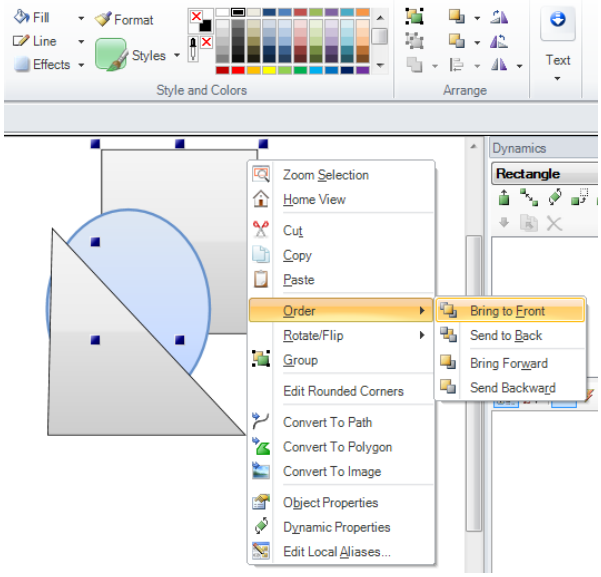


Figure 7.54 - Arrange Objects with Right-click Options

7.5 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 7.55. 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.

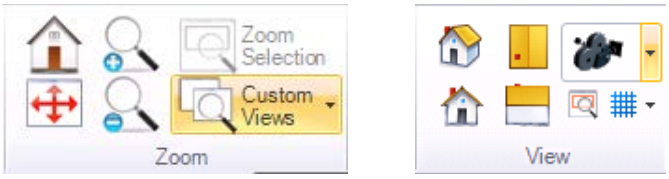


Figure 7.55 – Tools for Creating Views in 2D (Left) and 3D (Right)

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 7.56. 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.

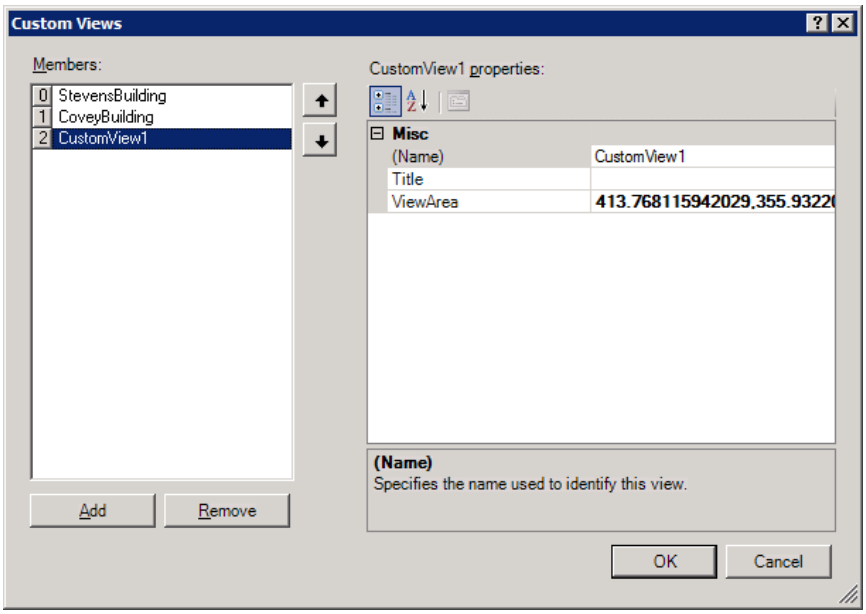


Figure 7.56 - Naming a View

7.6 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 7.57.

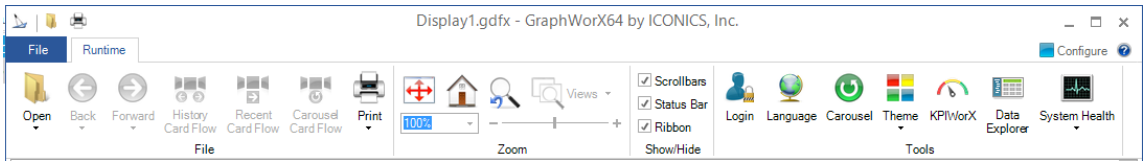


Figure 7.57 – GraphWorX64 Runtime Ribbon

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: Smart Tiles

L7A

7A.1. Lab Overview

In this lab, you will learn how to add and configure Smart Tiles. You will learn about adding process points, as well as adding symbols from a library.

Estimated time to complete this lab: **30 minutes**

7A.1.1. Task Checklist

In this lab, you will complete the following tasks:

- Adding and configuring Smart Tiles.
- Learning to use alignment.
- Learning to build ducts using symbols from a library.
- Learning to edit Dynamics in a Smart Symbol.

7A.1.2. Support Files

You will need the Symbol Library and AHU_Lab.gdfx to complete this lab.

7A.2. Creating an AHU Display

1. Open the **AHU_Lab.gdfx** file in the C:\Training folder.
2. Save the display as **AHU.gdfx**.
3. Select the **Controls** tab.
4. Select the **Smart Tile** control, then click and drag to draw a box into in the canvas. Any size will do, as we are going to resize it to match the other set points for the AHU.
5. **Double-click** on this new Smart Tile to enter the Smart Tile configurator.

6. From the **General** tab:
   - Change the **Width** and **Height** to **150** by **150**
   - Select the **Background** color, then enter the following numbers for:
     - **Red** = **192**
     - **Green** = **80**
     - **Blue** =**77**
7. From the **Top Banner** tab, set:
   - **Banner Height** = **46**
   - **Label** = **Discharge Cooling Setpt** (make it **Bold**)
   - **Font Size** = **14**

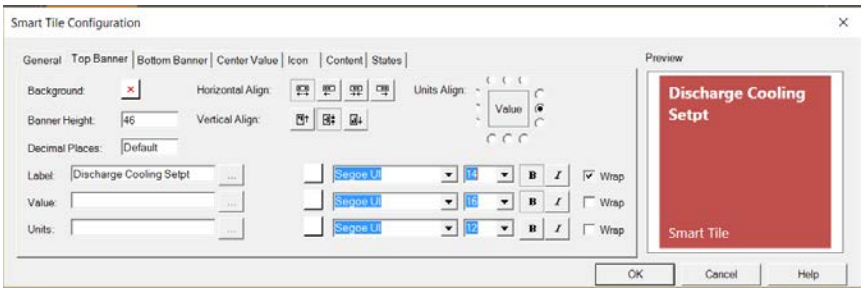


Figure L7A.1 - Smart Tile Top Banner Configuration

8. From the **Bottom Banner** tab:
   - Place the cursor in the **Units** field.
   - Hold the **Alt** key and enter **167** in the number pad, then release the **Alt** key.
   - You should see the **º** symbol. Type **F** after the symbol.
   - Set the **Units** font size to **13** and make it **Bold**.

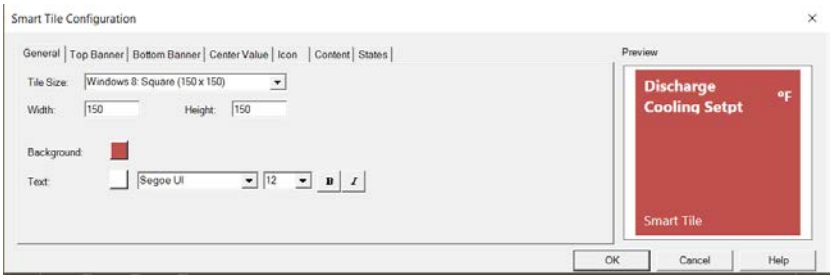


Figure L7A.2 - Smart Tile Bottom Banner Configuration

9. Click **OK**.

10. From the **Dynamics** tab, in the **Text Objects** section, select **Process Point**.

11. Place the **Process Point** on the screen. Any size is fine, as we will resize it later.

12. Click on the **Data Source** button in the **Dynamics** ribbon. Using the **Data Browser**, navigate to **AssetWorX → IcoUniversity → North Campus → Engineering Building → HVAC 01** then select the **ThermostatSetPoint_opc** tag.

13. Click **OK.**

14. In the **Dynamic** properties of the **Process Point,** set:

- **Data Entry** to **True**.
- **DecimalPlaces** = **1**

15. Then, under the **Properties** tab set;

- **Foreground** = **White** or **0,255,255,255**
- **Width** = **110**
- **Height** = **66**
- **Padding** = **0**
- **FontSize** = **48**

16. Select the **Explorer** tab to expose the object explorer.

17. We will move the **Label1** tree node beneath the **Smart Tile** tree node. This will add the **Process Point** into the **Smart Tile**.

18. Select **Label1**. While holding down the left mouse key, **drag** the **Label1** tree node **underneath** the **Smart Tile** and release the mouse key. The result should resemble the figure below.

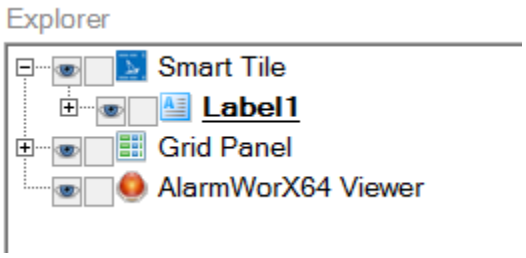


Figure L7A.3 - Process Point Added to the Smart Tile

Note: This feature works with anything in GraphWorX64. You can drop in a different control such as a Trend Viewer or Alarm Viewer. You could even create custom symbols and drop them into a Smart Tile.

19. Now, drag the Smart Tile into the **Control Point Group**.

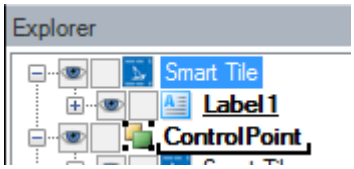


Figure L7A.4 - Smart Tile Dragged into the Control Point Group

20. Now we will use the **Arrange** tools to align the Smart Tile.

21. With the Smart Tile you just dragged into the **ControlPoint Group** selected, hold down the **Ctrl** key and select the **Discharge Heating Setpt** Smart Tile. It should look like the following image.

Note: The selected **Discharge Heating Setpt** smart tile should have blue handles. Blue handles indicate this is the focused object and the arrange function will order objects in relation to the focused object. If necessary, just left-click on another object to change the focused object.

Figure L7A.5 - Selected Smart Tiles

22. Then, in the **Home** tab, select the **Align** button in the **Arrange** section. Select **Align Middle**.

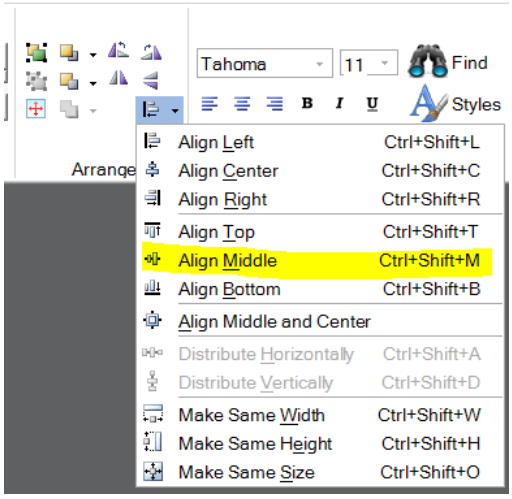


Figure L7A.6 - Alignment Options

23. Then, select only **Discharge Cooling Setpt** and **Enthalphy Changeover Setpt** with the blue squares (focus) over **Enthalphy Changeover Setpt** and select **Align Center**.

When you have properly aligned the tile with the others, you will have something that looks like the following figure.

Figure L7A.7 - Aligned Smart Tiles

24. Click in the gray area of the display with the left mouse key until you see **This Display : Canvas** at the top of the **Properties** section.

25. Now, from the **SI_BA_Training Symbol Library**, select the **AHU Diagram** drag and drop it in the upper frame (gray area).

26. Drop the **Duct – Smart Fan** and **Vent** and **Horizontal Duct Left End Cap** symbols onto the canvas.

27. Now, move them to complete the duct work, as depicted in the following image.

Note: With an object selected on the display canvas, the arrow keys will perform a fine adjust of the location.

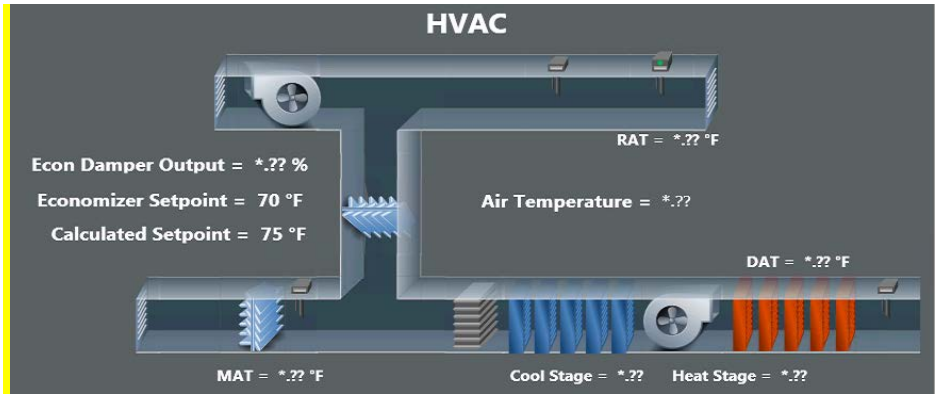


Figure L7A.8 - Duct Diagram

28. In the object **Explorer**, click on the + sign to expand the **Fan** tree.

29. Drill down and select **fan_blades1**.

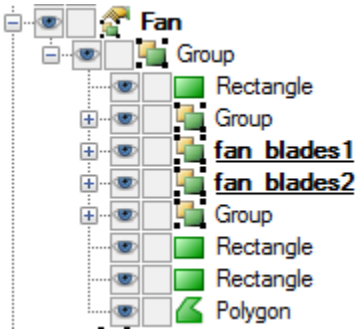


Figure L7A.9 - Selecting components of a Smart Symbol

30. Select the **Dynamics** tab. Select the **Rotation** dynamic.

31. Set the following properties:

- **AnimationMode** = **Discrete**
- **Duration** = **500**
- **RepeatCount** = **Infinite**

32. Again, from the object explorer, select the **Fan**. Now select the **Properties** tab and set the data sources as described below.

- **DataSource** = *ac:IcoUniversity/North Campus/Engineering Building/HVAC 01/CoolBlowerStatus_opc*
- **DataSource2** = *ac:IcoUniversity/North Campus/Engineering Building/HVAC 01/CoolBlowerStatus_opc*

33. **Save** the display.

34. Go into **Runtime**.

35. Select the **Data Entry** field you added to the **Discharge Cooling Setpt** Smart Tile.

Note: The **Discharge Cooling Setpt** is clipped between **55** and **90**.

36. Enter a value **+** or **– 2** degrees from the **Air Temperature** value.

37. If the system is cooling, the top fan will animate. If heating, the bottom fan will animate.

<u>This concludes this lab.</u>

Lab: Creating a Dashboard *L7B*

7B.1. Lab Overview

In this lab, you will learn how to create a dashboard utilizing both a floor plan and main "base" display. This lab will tie some previous labs' concepts together.

Estimated time to complete this lab: **30 minutes**

7B.1.1. Task Checklist

In this lab, you will complete the following tasks:

- Create a complete dashboard.
- Link previously created campus display for more detail.
- Learn to use Commanding in GraphWorX64.
- Add color dynamics to elements in your displays (e.g. "rooms").

7B.1.2. Support Files

You need the following support files created in earlier labs:

- LibraryEWX.gdfx
- AHU.gdfx
- LibraryOverview.gdfx
- LibraryMap.gdfx

7B.2. Creating a Dashboard

Floor Plan:

1. Launch GraphWorX64 and open **LibraryMap.gdfx**.

2. From the Home ribbon, select the solid **Polygon** and outline the perimeter of **StateRoom1** (M), as labeled in the following figure.

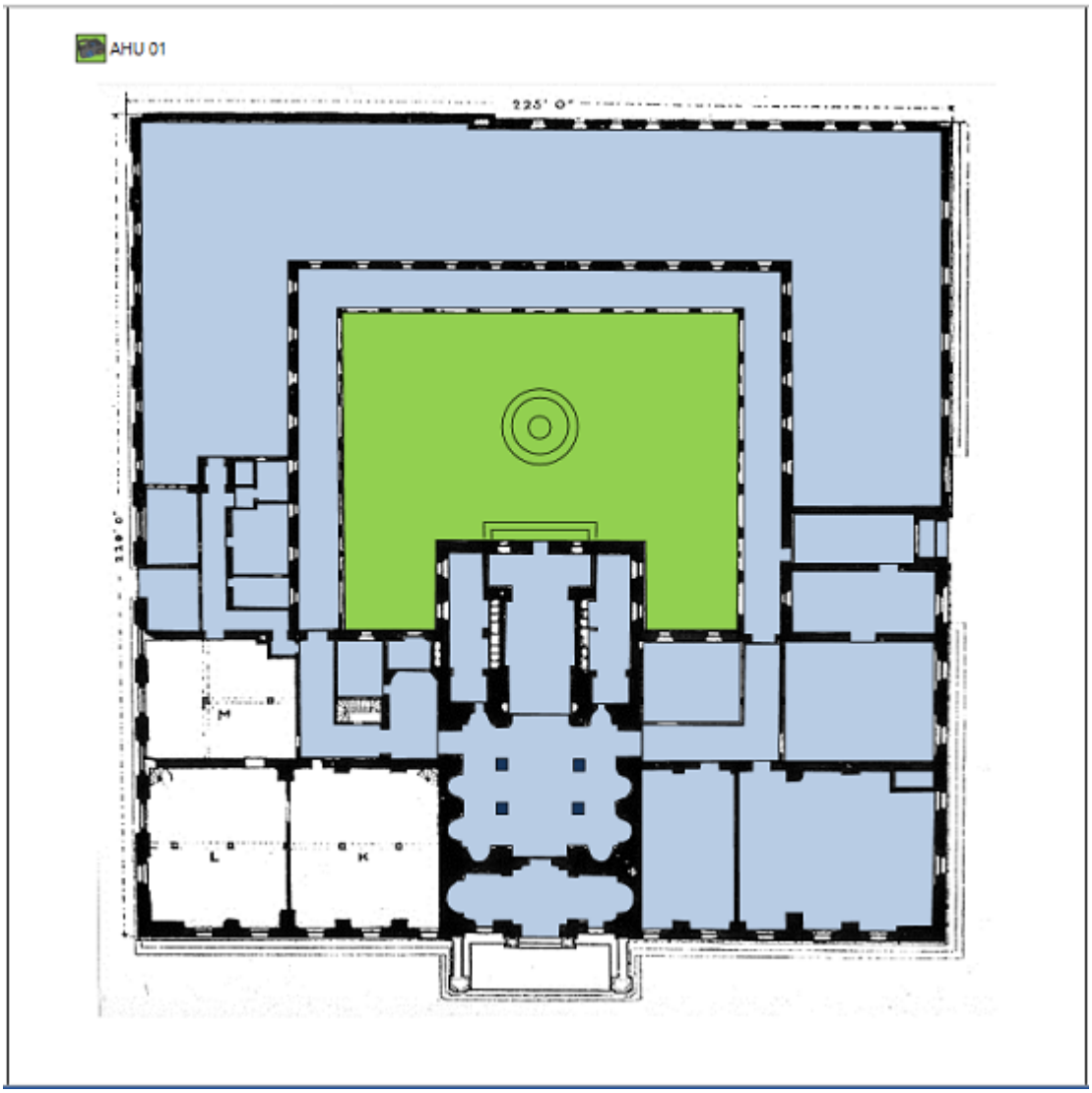


Figure L7B.1 - Adding StateRoom Ploygons

Note: After selecting the Polygon drawing tool, if you click on the upper left corner of StateRoom1, then click on the next location where you want the polygon to be drawn, then the next location and so on, the polygon will be drawn to the next location you click on.

3. Label the polygon in the object explorer, **StateRoom1**.

4. In the Properties tab set the Fill property to the solid blue color you see in the other rooms. **Fill = 184, 204, 228**.

5. From within the object explorer, drag and drop the **StateRooms** into the **Rooms** group.

6. Repeat steps **2** through **5** to create **StateRoom2** and **StateRoom3**.

7. Select the **Rooms** group and copy and paste it. You should see a new entry labeled **Rooms1** at the top of the explorer.

8. In the object explorer, select **Rooms1.** Hold the **CTRL** key and select **Rooms**. Both **Rooms1** and **Rooms** should be **selected**.

9. We will now align **Rooms1** and **Rooms** on top of each other. With the **Home** tab selected, in the **Arrange** section, select the **Align** dropdown list. Select **Align Center,** then **Align Middle.**

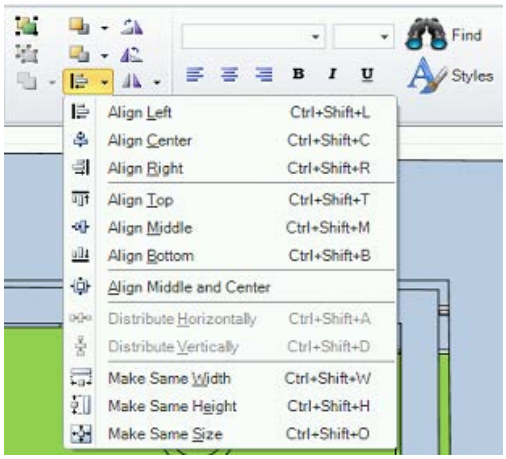


Figure L7B.2 - Alignment Options

10. In the object explorer, expand **Rooms1** and select **StateRoom1.**

11. Click on the **Dynamics** tab and add a **Color Dynamic** . Enter the following dynamic property values:

Common	**(Name)**	Red_Dynamic
Data	**DataSource**	ac:IcoUniversity/North Campus/Engineering Building/HVAC 01/RoomTemperature_opc
Data	**AnimationMode**	Analog
Data	**LowLimit**	68
Data	**HighLimit**	90
Dynamic Behavior	**EndColor**	Red 255,0,0

12. We will add a second color dynamic by duplicating the color dynamic we just added.

13. Select the **Red_Dynamic** color dynamic.

14. From the **Dynamic Explorer** ribbon, click on the **Green Arrow** icon as depicted with the circle in the figure below.

Note: The green arrow icon tooltip displays "**Duplicate Select Dynamics"**.

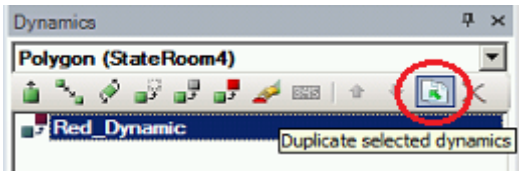


Figure L7B.3 – Duplicate Selected Dynamics

15. The result is a **Red_Dynamic1** color dynamic added to **StateRoom4**.

16. You will continue by editing the **Red_Dynamic1** dynamic properties.

17. Select the **Dynamics** tab and click on the **Red_Dynamic1** and make the following changes.

Common	**(Name)**	Blue_Dynamic
Data	**LowLimit**	50
Data	**HighLimit**	68
Dynamic Behavior	**UseStartColor**	True
Dynamic Behavior	**StartColor**	Dark Blue 0, 0, 255
Dynamic Behavior	**UseEndColor**	False

18. Click on the **Blue Up Arrow** in the dynamic properties ribbon (near the green arrow) to move the Blue_Dynamic to the top of the list.

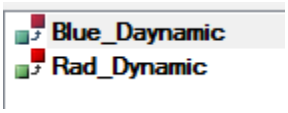


Figure L7B.4 - Reordered Dynamics

19. Double click out of the **Rooms1** group so that nothing is selected in the Explorer.

20. Select the **Home** tab. From within the **Shapes** section, click on **Layer**. This adds a **Layer1** to the display. Rename the layer to **HeatingLayer** in the Properties tab.

21. In the Explorer Tree, cut and paste **Rooms1** into the **HeatingLayer**.

22. Double click out of the **Rooms1** group again.

23. Select the **Home** tab. From within the **Shapes** section, click on **Layer**. This adds a **Layer1** to the display. Rename the layer to **EquipmentLayer**.

24. In the Explorer Tree, cut and paste the **AHU_01** object into the **EquipmentLayer**.

25. Then, on the **Display Canvas**, drag the **AHU_01** and lay it on top of **StateRoom3**.

26. Select the **Properties** tab for **AHU_01** and set the **FileName** to **AHU1.gdfx.**

Note: The Filename is under the Smart Properties section because AHU_01 is a Smart Symbol.

27. From within the object Explorer, click on the **Eye** icon associated with the **FloorPlan**.

28. Then, click on the square where the eye would be for **BackDrop**.

29. Click on the eye for the **HeatingLayer** in the Explorer. Your Explorer Tree should look like this when you are done:

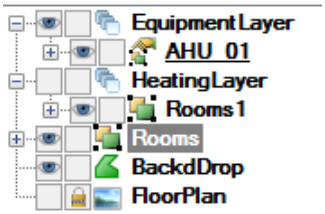


Figure L7B.5 - Explorer Tree Example

30. Select **File → Save**.

7B.1.1. Adding the Floor Plan

1. Open the **LibraryOverview.gdfx** display.
2. Select the **Floorview** in the Tree Explorer.
3. In **Properties**, under **Misc**, set the **Source** to **LibraryMap.gdfx**.
4. In the **Object Explorer Tree,** select the **Heatmap** Smart Tile under the **TopButtons** group**.**
5. Add a **Pick Action** Dynamic, then change its properties to match the following (choose the command from the dropdown and type the object name in manually):

Command	Set Object Visibility
ObjectName	HeatingLayer | Floorview

6. Select **File →Save**.

7B.1.2. Create the Base Dashboard Display

1. Open the display **LibraryMain.gdfx**.
2. To create the dashboard from scratch, launch another instance of GraphWorX64; click **File → Save As** name **LibraryMain.gdfx**.
3. When you complete the steps in this section and the LibraryMain.gdfx display is complete, it should resemble the figure below.

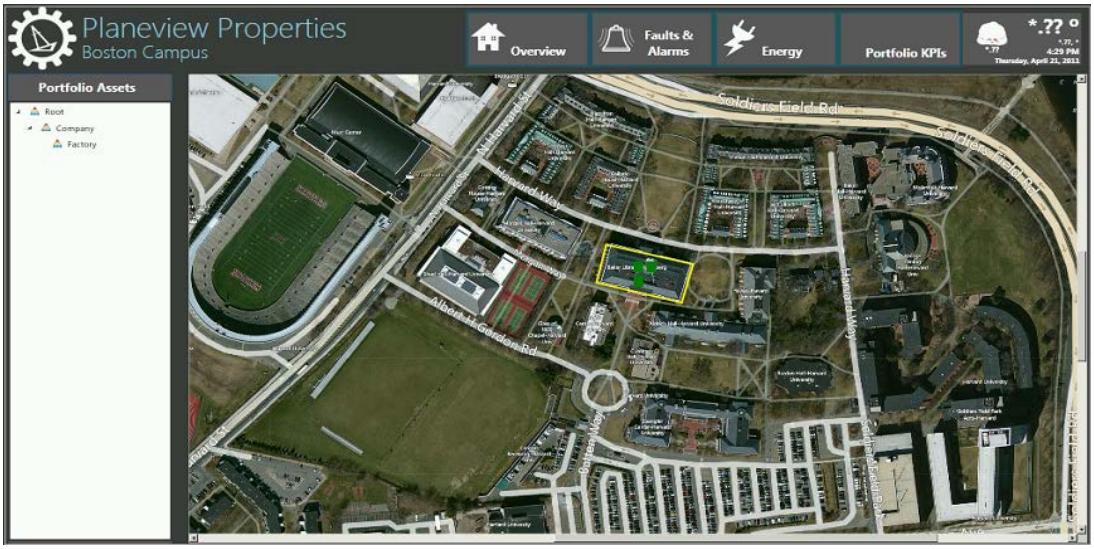


Figure L7B.6 - Complete Main.gdfx Display

4. Select the **Properties** tab in the lower right section of the GraphWorX64 application and enter the following properties.

Section	Property	Value
Common – Style	Background	63, 63, 63
Measurements	Width	1920
Measurements	Height	1040
Measurements	IsScaleable	True

Note: If the properties and symbol tabs are not available, select the View tab and verify that the Properties and Symbols checkboxes are selected under the Show/Hide section of the ribbon.

5. Zoom out until you see the white frame outlining the display area.

6. Click on the **Symbols** tab.

7. Click on the dropdown list of the Symbol library and select the **Dashboard Symbols** category listed. This was added in a previous lab. If you don't see this library, execute the following two steps.

   - Click on the **Symbol Library** commands menu icon at the top of the Explorer and select **Open Existing Category** from the popup menu.

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

8. Using the **CTRL** key, select the **Asset_Smart_Symbol, CompanyLogo** and **Navigation_Buttons**.

9. With the three symbols selected, use the **Right** mouse key to select then **Drag** and **Drop** the symbols into the display framed area.

10. Organize the symbols as seen in the figure below.

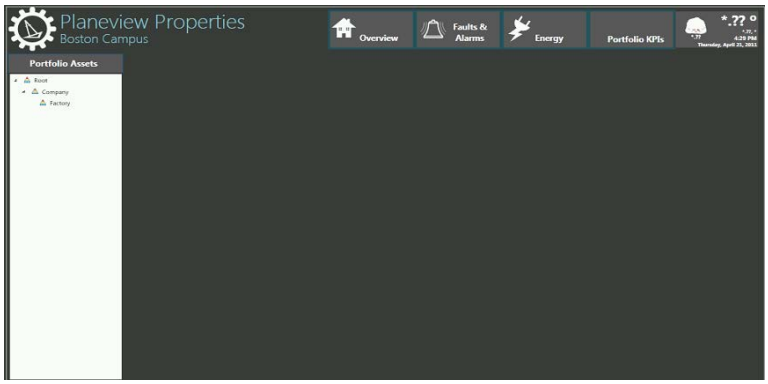


Figure L7B.7 - Symbols Added to the Main.gdfx Display

11. The **CompanyLogo** should be in the top left corner.

12. The **Asset_Smart_Symbol** should be along the left-hand side.

13. The **Navigation_Buttons** should be on the same line as the logo but on the right-hand side.

14. From the **Controls** tab, select the **GraphWorX64 Viewer**.

15. Click and drag to create a viewer to fill the white space to the right of the **Portfolio Assets** symbol.

16. With the **GraphWorX64 Viewer** selected, click on the **Explorer** tab in the lower left.

17. **Rename** the **GraphWorX64 Viewer** tree node to **Content**.

18. Now, in the properties of the **GraphWorX64 Viewer**, set the **Source** to **EWX.gdfx**.

19. **File → Save As Main.gdfx**.

20. Enter **Runtime** and click on the **Overview** selection in the dashboard ribbon. This is the only active button at the moment. The other button could be edited if needed.

    Note: You may have to zoom out by holding the Alt key and scrolling with the mouse wheel.

This concludes this lab.

AlarmWorX64 Server 8

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 management throughout your system. AlarmWorX64 is a family of modular alarming products, including the Alarm Server, Alarm Logger, Alarm Viewer Control, Multimedia Server and AlertWorX (the newest 64-bit multimedia alarming module).

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 primary source of alarm and event information.

7.1 Objectives

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.

8.1. 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.

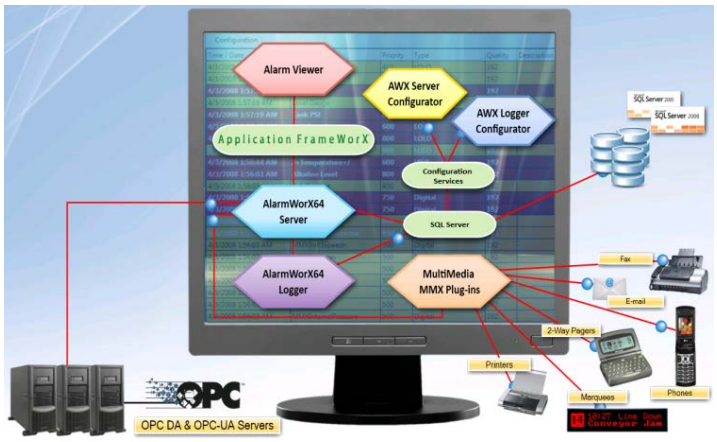


Figure 8.1 - Overview of AlarmWorX64

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 runs as a service. The server reads its configuration information from a Microsoft SQL Server database file. During runtime, the server polls the configuration database for changes, allowing on-the-fly configuration change without stopping the server.

AlarmWorX64 Server configuration is editable through the Workbench Desktop (Workbench) application. To start Workbench go to **Start → ICONICS Product Suite → Workbench**. In the Project Explorer of the Workbench browse to **MyProject → Alarms and Notifications → AlarmWorX64 Server**, as shown in Figure below.

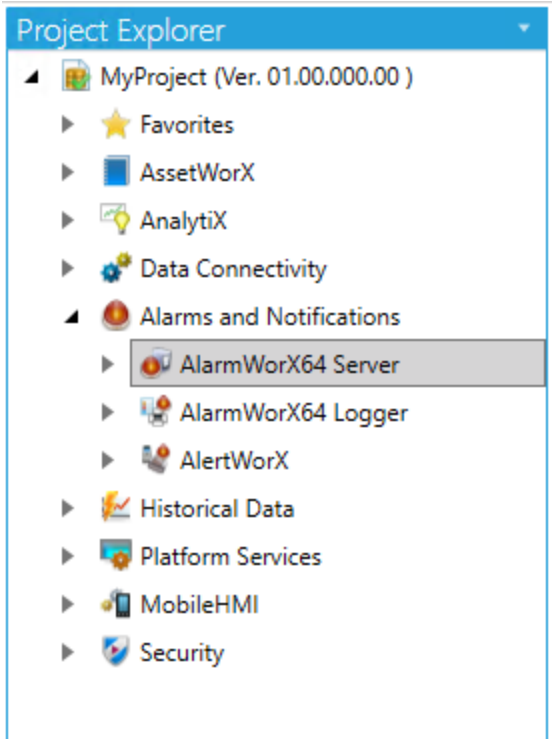


Figure 8.2 - AlarmWorX64 Server Configurator

8.2. 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(s) 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 the equipment or goods. 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.

8.3. 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.

8.3.1. 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 active configuration, then select **Add Tag** from the menu. This opens the alarm tag configuration form with the **General** tab on top, as shown in Figure 8.3.

Give the alarm a **Tag Name** and, in the **OPC Input** field, specify the OPC Server tag whose value you want to evaluate 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 "tag" icon to the right of the **OPC Input** field; this opens the **Data Browser** where you can navigate to and select the appropriate tag. Or you can use integrated Data Browser in Workbench to navigate and drag and drop the appropriate tag into the OPC Input field.

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 Tag Name, OPC Input, and Template fields always appear regardless the tab you are on in an alarm tag configuration. There are six different tabs in the alarm tag configuration: General, Alarms Limit / Digital, Alarms Deviation / Rate of Change, Alarms Rate Limit / Trigger Limit, Related Values 1-10, and Related Values 11-20. 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.

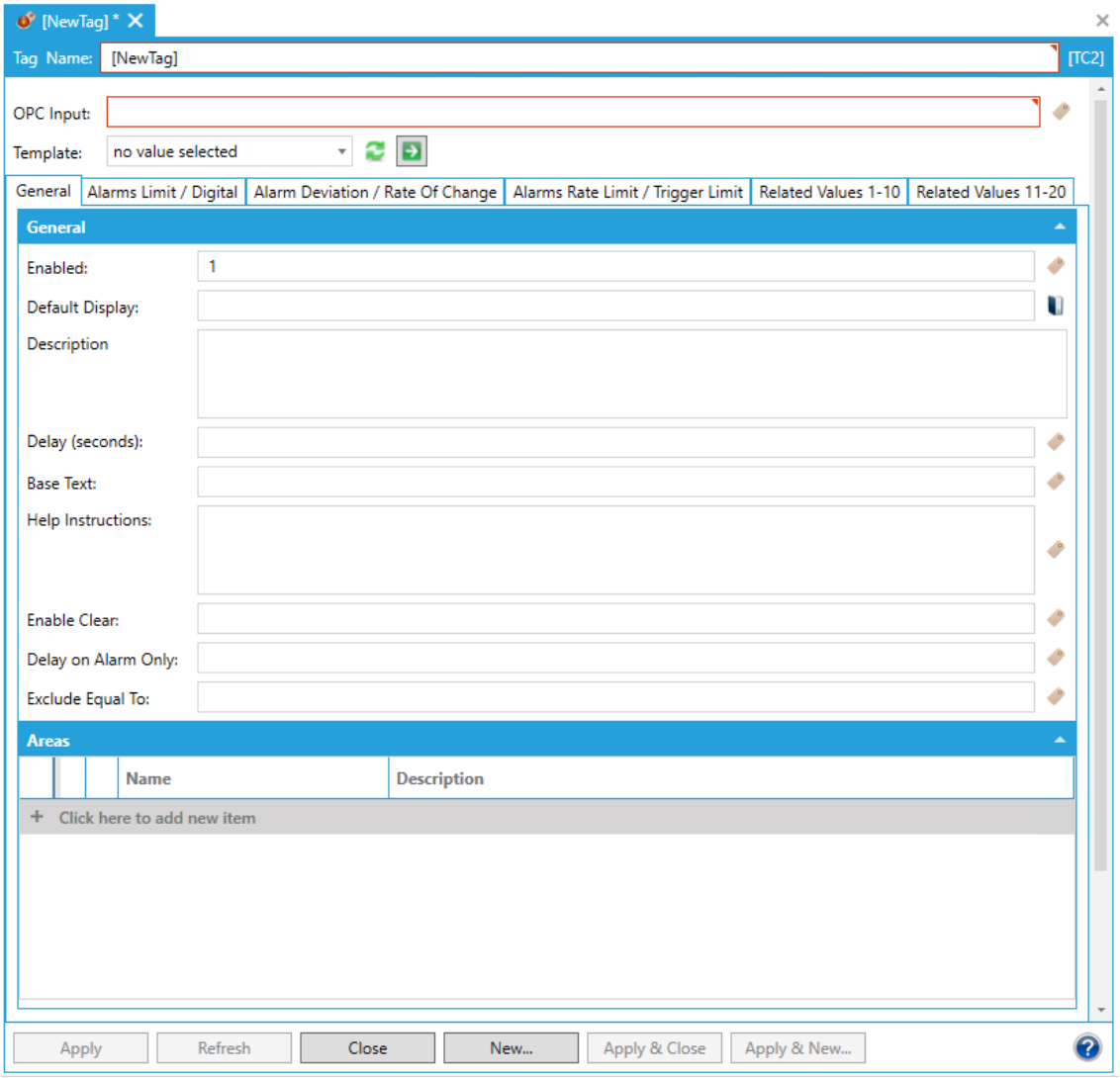


Figure 8.3 - Alarm Tag Configuration

8.3.1.1. General Tab

On the General tab, shown in Figure 8.3, 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/or mouses-over the alarm.
- **Enable Clear:** When checked, the alarm will contain an extra state. Normally, when an OPC tag value returns within the normal limit and an operator has acknowledged the alarm, the alarm is considered complete; therefore dropping out of the state server. With this option checked, the alarm will remain visible in the viewer instead of dropping out of viewer; it will then enter the **Clear State**, at which point, if someone acknowledges the alarm again, it will drop out.
- **Delay On Alarm Only**: Normally, when you enter a **Delay** for the alarm, it would delay both alarm messages and return to normal messages. When this option is enabled, it will only delay alarm messages.
- **Exclude Equal To:** When comparing present values to generate an alarm, the normal comparison would include the value for the alarm state. With this option enabled, the system would exclude the configured value for the state.

- **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.

8.3.1.2. Related Values Tabs

The **Related Values** tabs, shown in Figure 8.4, identify 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 now or 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.

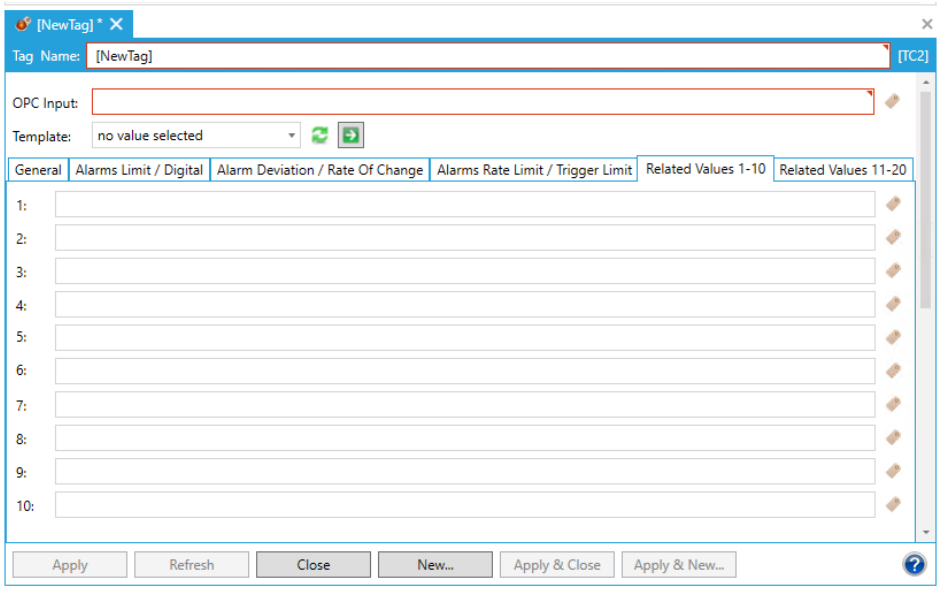


Figure 8.4 - Related Values tabs

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 you can configure in an alarm tag (Limit, Deviation, Digital, Rate of Change, Rate Limit, and Trigger Limit). Note that you can configure any of the alarms described here, and can even combine them into one alarm tag configuration.

8.3.2. Limit Alarm

If you go to the alarm tag's Alarms Limit / Digital tab, in the **Limit** pane, shown in Figure 8.5, you can configure an analog or **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 will override this tag by adding a 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 value. 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, unless left blank, otherwise the **OPC Input** field at the top of the configurator will be used.

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.

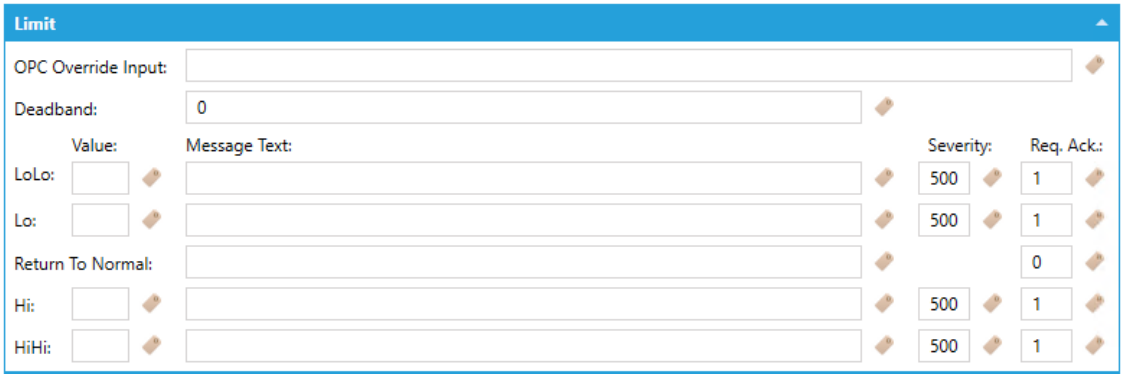


Figure 8.5 - Limit Alarm Configuration - Alarms Limit / Digital Tab

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 1 (lowest) to 1000 (highest) or blank (none).

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 and out of 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.

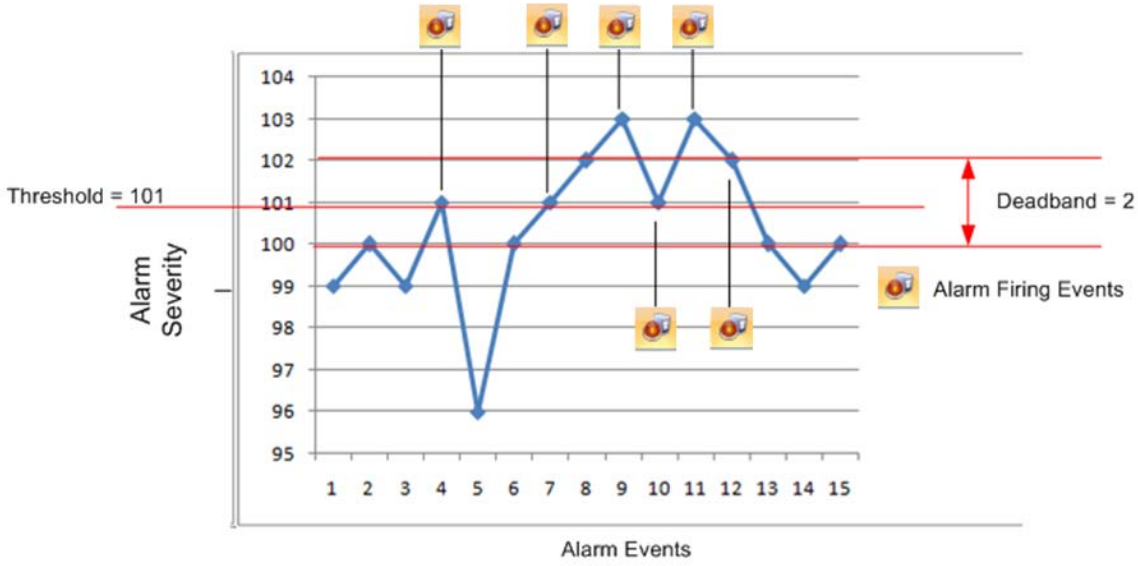


Figure 8.6 - Deadband for Alarms

8.3.3. 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 8.7. 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.

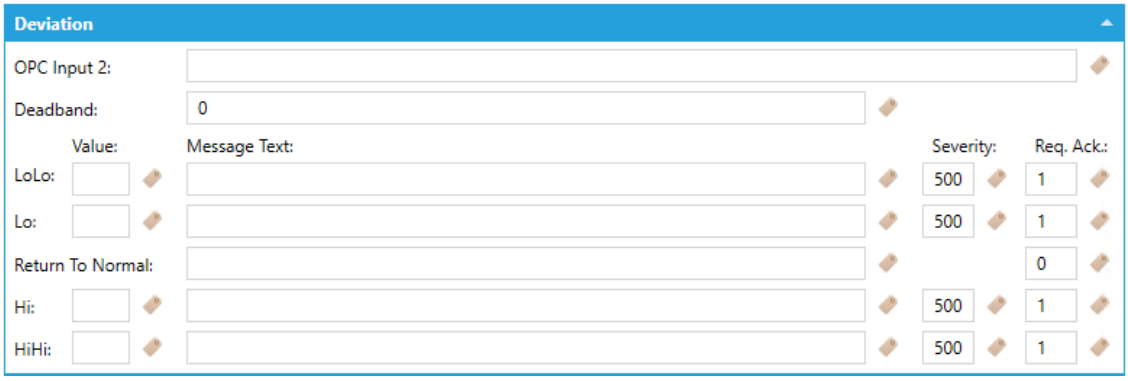


Figure 8.7 - Deviation Alarm Configuration - Alarms Deviation / Rate of Change Tab

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.

8.3.4. 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 8.8. 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, unless left blank, otherwise the **OPC Input** field at the top of the configuration will be used.

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.

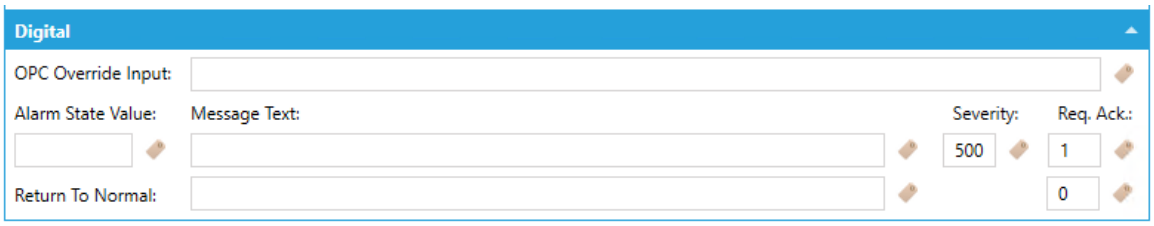


Figure 8.8 - Digital Alarm Configuration - Alarms Limit / Digital Tab

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.

8.3.5. 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 8.9. 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.

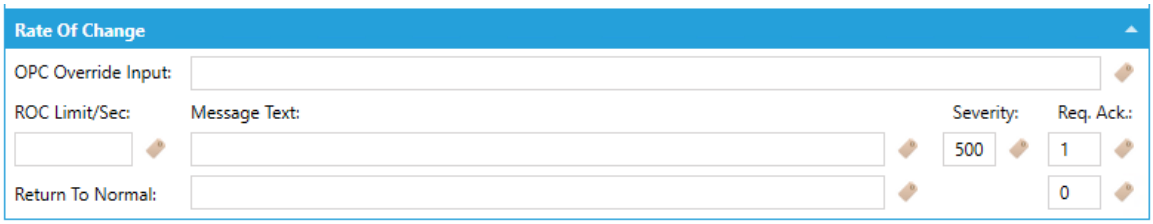


Figure 8.9 - Rate of Change Configuration - Alarms Deviation / Rate of Change Tab

This alarm reads the values of the OPC tag you specified in the **OPC Override Input** field in the Rate of Change pane (unless left blank), otherwise the **OPC Input** field at the top of the configuration will be used.

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.

8.3.6. 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** pane on the alarms Rate Limit / Trigger Limit tab, shown in Figure 8.10.

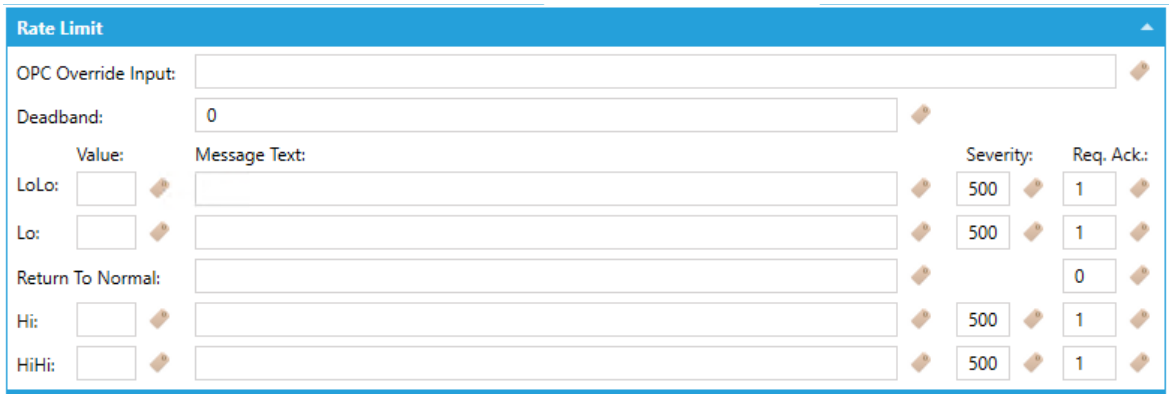


Figure 8.10 - Rate Limit Alarm Configuration - Alarms Rate Limit / Trigger Limit Tab

This alarm reads the values of the OPC tag you specified in the **OPC Override Input** field in the Rate Limit pane (unless left blank), otherwise the **OPC Input** field at the top of the configuration will be used.

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.

8.3.7. 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** pane on the Alarms Rate Limit/Trigger Limit tab, shown in Figure 8.11. By default, this alarm reads the values of the OPC tag you specified in the **OPC Input** field. You will override this tag by specifying a tag in the **OPC Override Input** field in the Trigger Limit pane (unless left blank).

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

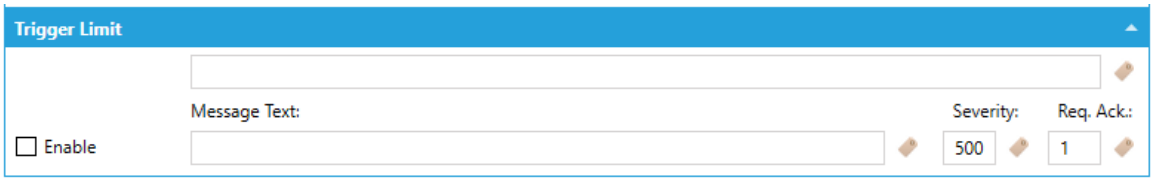


Figure 8.11 - Trigger Limit Alarm Configuration - Alarms Rate Limit / Trigger Limit Tab

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.)

8.3.8. 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.

8.3.8.1. 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 **Add 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.

8.3.8.2. 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.

8.3.8.3. 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 8.12 where you can see a list of all the alarm tags that reference the template.

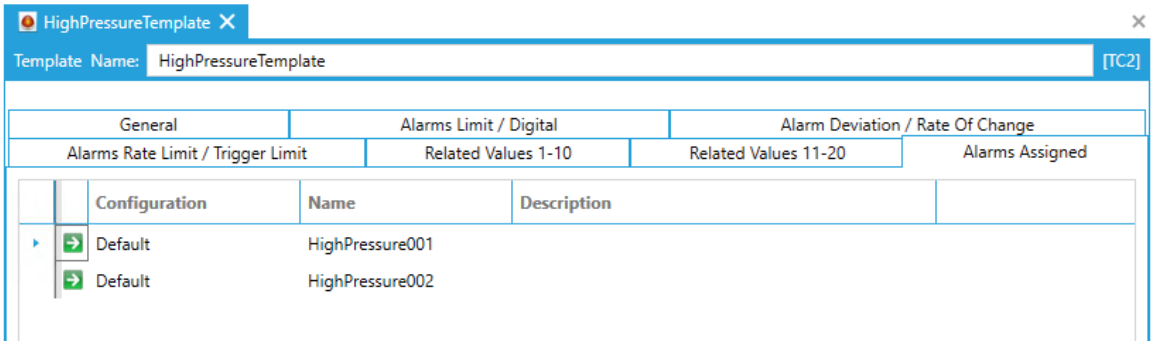


Figure 8.12 – Alarms Assigned Tab for a Template

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

- The tag's N**ame** 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

If you want maintain even these values from template, you must delete the original value in alarm. The background of the field change to ping and from that time even these fields are maintained automatically from template.

When maintaining a templates and alarms values, keep also in mind that once an alarm tag's field is changed manually, the field's link to the template is broken and that field is no longer be maintained from the template. You must delete the value to regain the link with template again as described above.

8.3.9. 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 8.13.

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.

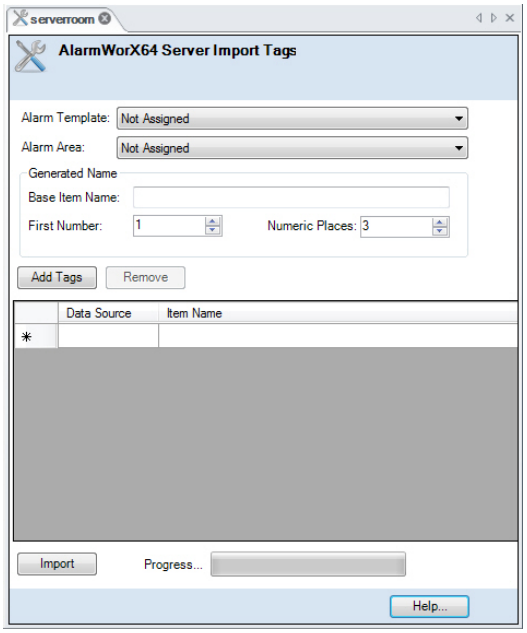


Figure 8.13 - Alarm Tag Import Wizard

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 created. 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.

8.4. 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.

A good example of filtering is adding the Multimedia Alarms area to any AlarmWorX64 Server alarm you intend to send to Multimedia alarming. The client can filter on the particular area in the Subscription.

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 8.14 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.

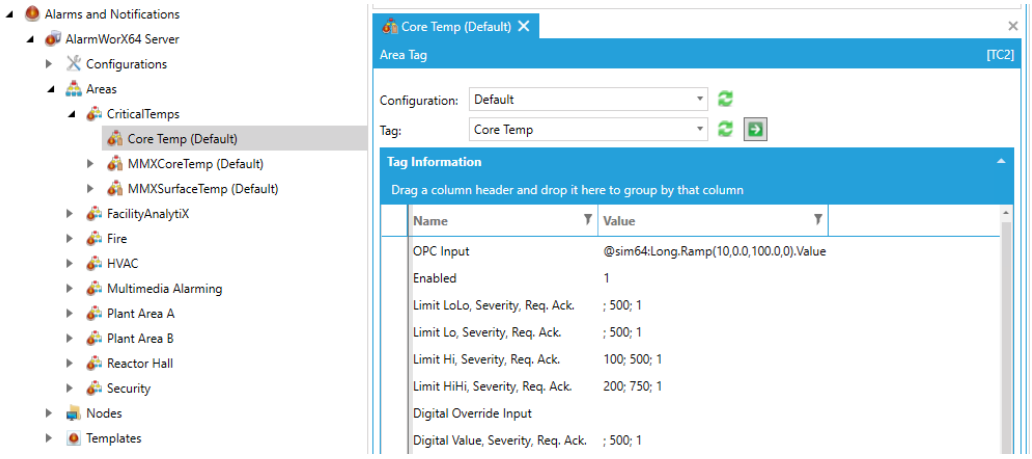


Figure 8.14 - Areas for Grouping Alarms

8.4.1. Adding an Area

To add an area, right-click on the **Areas** node, then select **Add 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 Add Area from the popup menu. You will get the same configuration interface.

8.4.2. 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 **Add Area Tag** from the popup menu, which displays the configurator illustrated in Figure 8.15. After you pick the tag's database configuration, you can select the alarm tag from the **Tag** dropdown list.

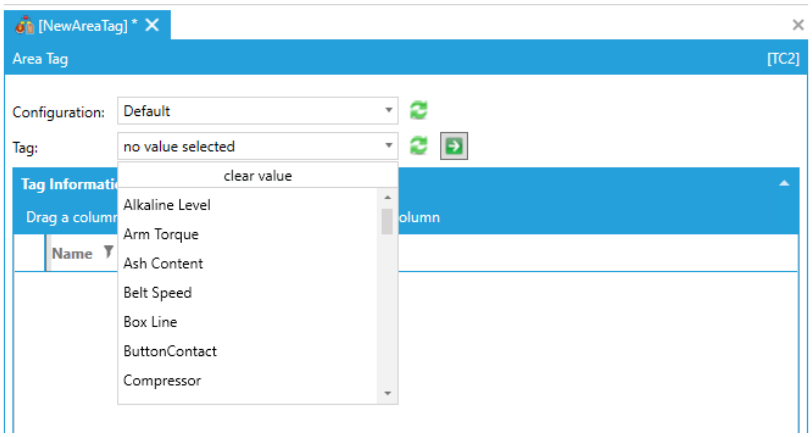


Figure 8.15 - Adding an Alarm to an Area

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 gray **"+ Click here to add new item**" or Remove by clicking on the red "X" icon in front of the area you want to remove.

Figure 8.16 - Areas Section in an Alarm Tag

If you **click here to add new item**, the **Please select an item** 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.

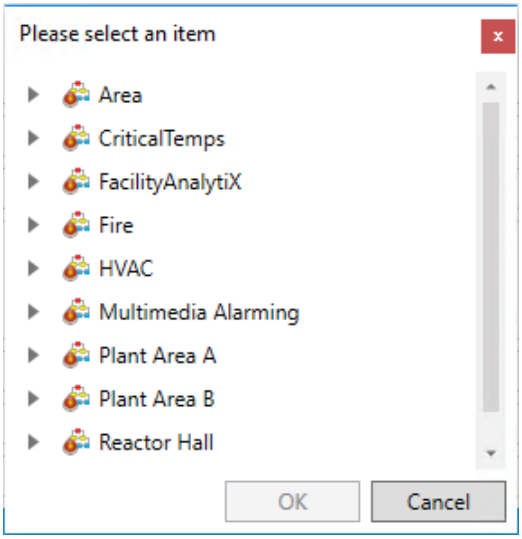


Figure 8.17 - Select or Enter Area Name Dialog

You can also remove an area from the alarm by selecting the red "X" in front of the area you want to remove.

8.5. 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.

8.5.1. 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 Sub-Tree** from the menu. In the dialog box that opens, select the format of export (e.g. CSV), choose whether Generate relative paths in export and click OK. The item(s) will be exported in background. A new task will appear in the Recent Tasks pane. You can download the exported file once the task is completed.

8.5.2. 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 old items*, too.

The import file types supported are: CSV file, TXT file, Microsoft Excel 2007/2010 file (XLXS), and eXtensible Markup Language file (XML).

To import a component from an exported (e.g., CSV) 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. From the dropdown menu specify if you want to **Only Create new items**, **Only Update old items**, or **Create and Update**. Click **OK**.

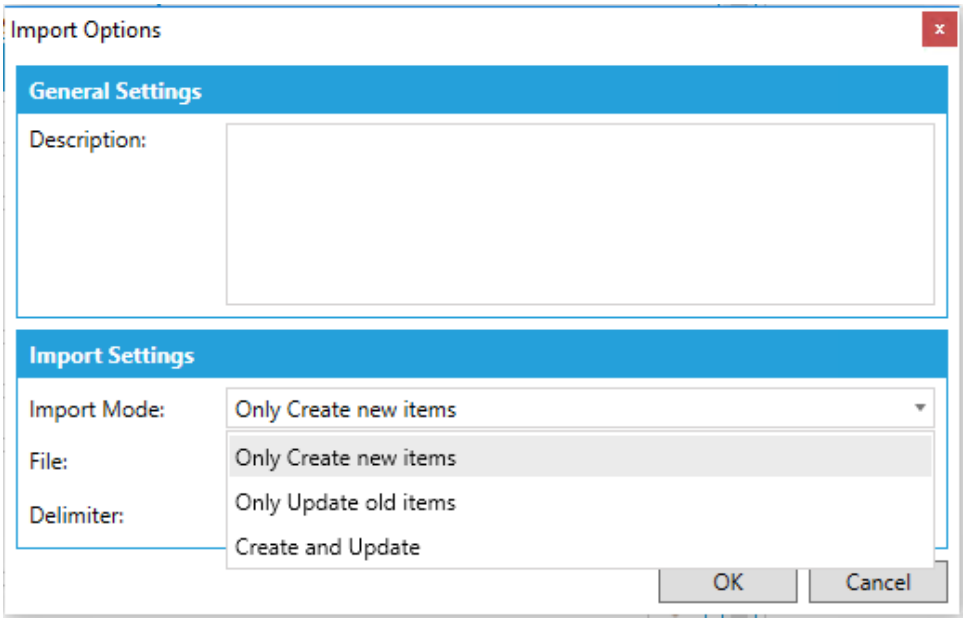


Figure 8.18 - Import Options

In the Recent Tasks pane you can see the progress of import. When the import is done, you can view the import log for the result. You can also 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 *L8*

8.1. Lab Overview

In this lab, we will create alarms that will alert us when the icTonics drink-mixer-tank level is too high or too low. We will also create an alarm that alerts us when the churning 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**

8.1.1. 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.

8.1.2. Support Files

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

8.2. 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. In the **Project Explorer** of the Workbench expand to **Alarms and Notifications** → **AlarmWorX64 Server**.

4. Expand the AlarmWorX64 Server tree. You will see four folders underneath it: **Configurations, Areas, Nodes, and Templates**.

5. 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.

8.2.1. Alarm Areas

1. Right-click on the **Areas** folder and select **Add Area**.

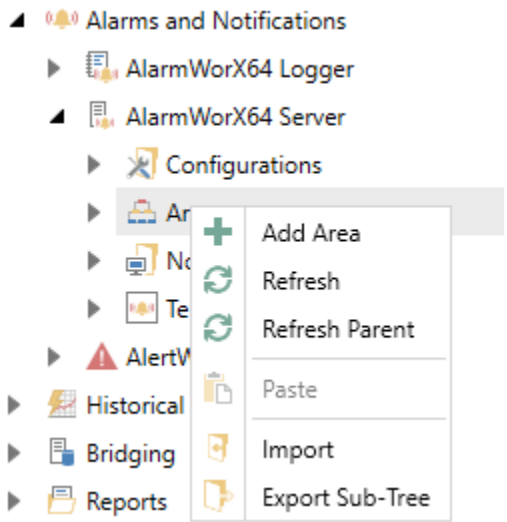


Figure L8.1 - Adding an Area

2. A new Workbench tab opens in Project Details that allows you to create a new Area. **Name** the Area, "**USA**". Click **Apply** and **Close**.

3. Create a sub-area by right-clicking on the area called USA and select **Add Area**. We'll name this area "**Foxboro**", Click on **Apply** and **Close**.

4. Create another sub-area under Foxboro called "**Tank Levels**". Click **Apply** and **Close**.

5. When you are done, your areas should look like the figure below.

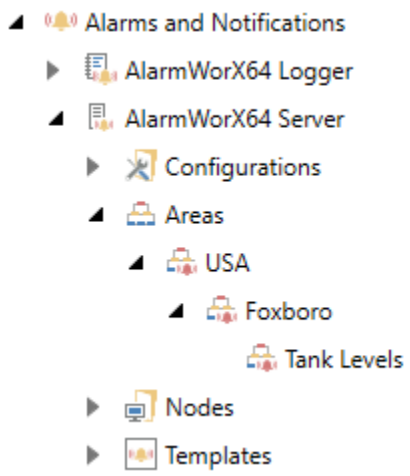


Figure L8.2 - Alarm Areas

8.2.2. Alarm Configurations

1. Right-click on Configurations and select **Add Configuration** to create a new alarm configuration.

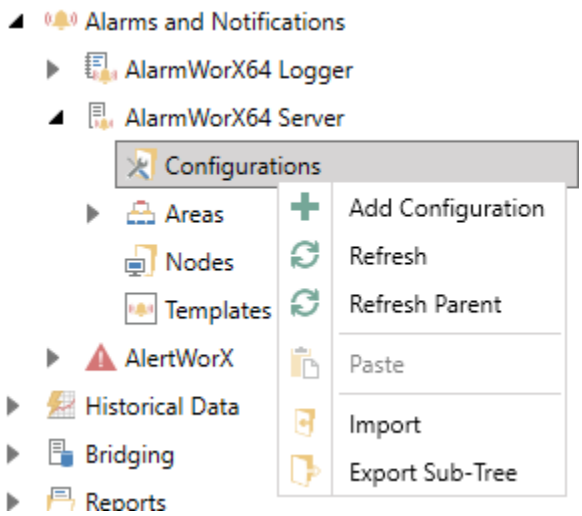


Figure L8.3 - Adding Alarm Configuration

2. **Name** this configuration "**icTonicsAlarms**" and put a checkmark in the "**Default (Active) Configuration**" checkbox to make this configuration active upon startup.

3. Click on **Apply** and **Close** to save changes.

4. Right-click on the **icTonicsAlarms**, under the Configuration tab, and select **Add Tag** to add an alarm tag.

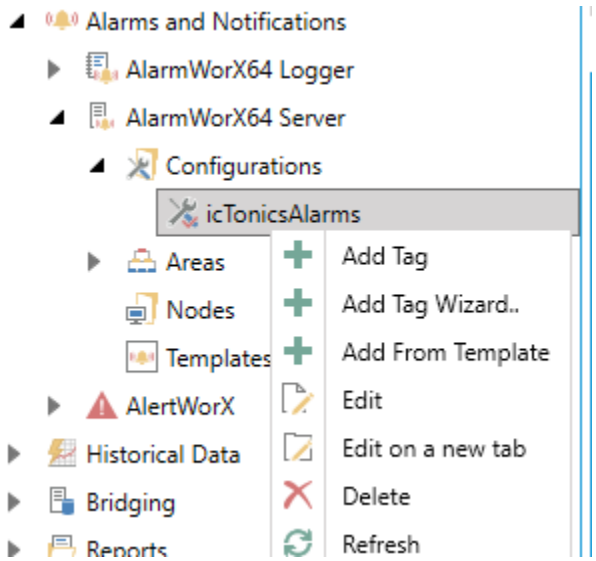


Figure L8.4 - Adding Alarm Tags

5. A new tab opens with alarm tag properties. Give this alarm tag the name "**Tank1 Level**".

6. In the OPC Input: field, click on the ellipse button to the right of the text box.

   In the **Data Browser**, locate the tag:
   modbus:icTonics.Tanks.USA.Foxboro.Tank1.TankLevel

7. Click **OK.**

8. In the **Base Text** field, with the **General** tab selected, enter the text "**Tank1 level is**".

9. Click "**Apply**" to save the tag.

10. On the **Areas** tab, in the blue ribbon, click on the white "**+ Click here to add new areas**" entry.

11. In the "Please select an item" dialog, expand the tree and select the **Tank Levels** area.

12. Click **OK.**

13. Select the **Alarms Limit** tab and configure the alarm properties as shown in the table below.

Limit Alarm Values

Condition	Value	Message	Severity
LoLo	10	very low	800
Lo	20	low	500
Hi	80	high	500
HiHi	90	very high	900

14. In the **Return to Normal**: message field, enter "**Normal**". Your alarm should look similar to Figure L8.5.

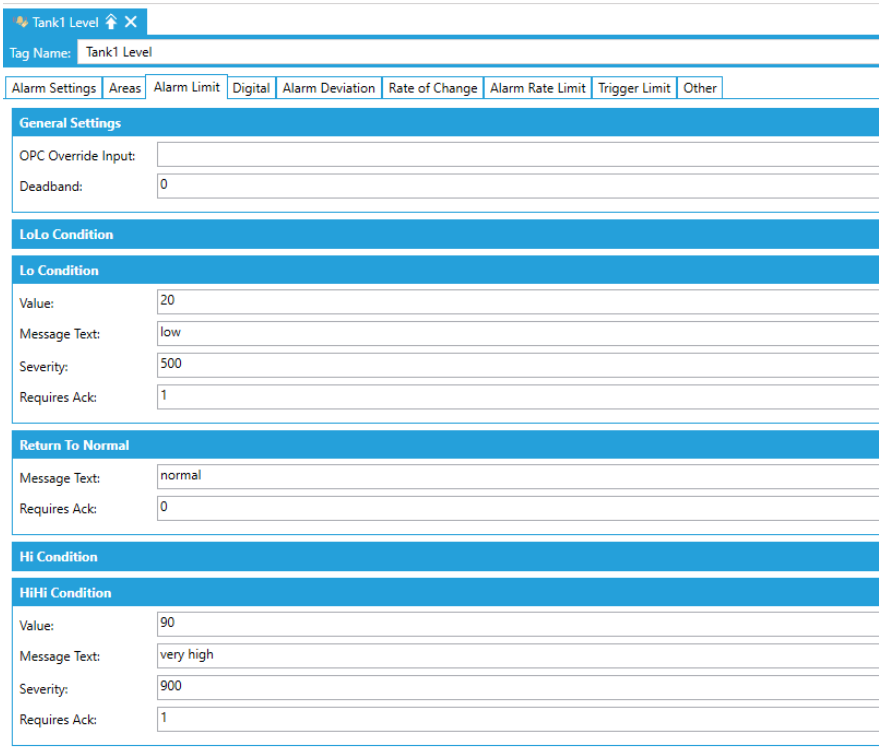


Figure L8.5 - Limit Alarm Configuration

15. Click on "**Apply**" to save changes.

16. Using a similar procedure, **add** another **Tag** under **icTonicsAlarms**. The configuration will differ slightly since this will be a Boolean tag. Under the **Digital** tab fill the entries according to the chart below. When configured completely, this tag will alarm if the churning blade1 has stopped moving.

Digital Alarm Details

Name:	Churning Blade1 Status
OPC Input:	modbus:icTonics.Tanks.USA.Foxboro.Tank1.ChurningBlade Status

Base Text:	The churning blade in Tank1
Alarm State Value:	0
Message Text:	has stopped moving
Return To Normal:	is moving

17. We are now done configuring the alarm server. However, there is one important step left. We need to confirm the AlarmWorX64 Server is started. Monitor the Traffic Light on the Home ribbon and ensure that the traffic light on the ribbon is **Green**.

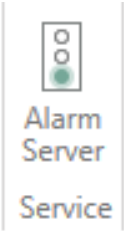


Figure L8.6 - AlarmWorX64 Server Service Traffic Light

8.3. Alarm Tag Multiply

1. Now we will add similar alarms for Tank2 and Tank3. For this, we can use the **Multiply** function, which makes it easier to create similar alarms.

2. In the upper section of the split Project Details window, right-click on the **Tank1 Level** alarm tag and select **Multiply**.

3. Set the **Number of copies** to **2**, **Override First number** to **2**, and **Override Numeric Places** to **1**.

4. In the first column, select the **Checkboxes** for the **Base Text**, **Name** and **OPC Input** fields.

5. Fill in the values of the selected fields according to the table below.

Note: The {0} is a place holder that will be replaced with the number generated by the **Multiply** feature.

Base Text	Tank{0} level is
Name	Tank{0} Level
OPC Input	modbus:icTonics.Tanks.USA.Foxboro.Tank{0}.TankLevel

6. When your dialog resembles the figure below, Click on "**OK**".

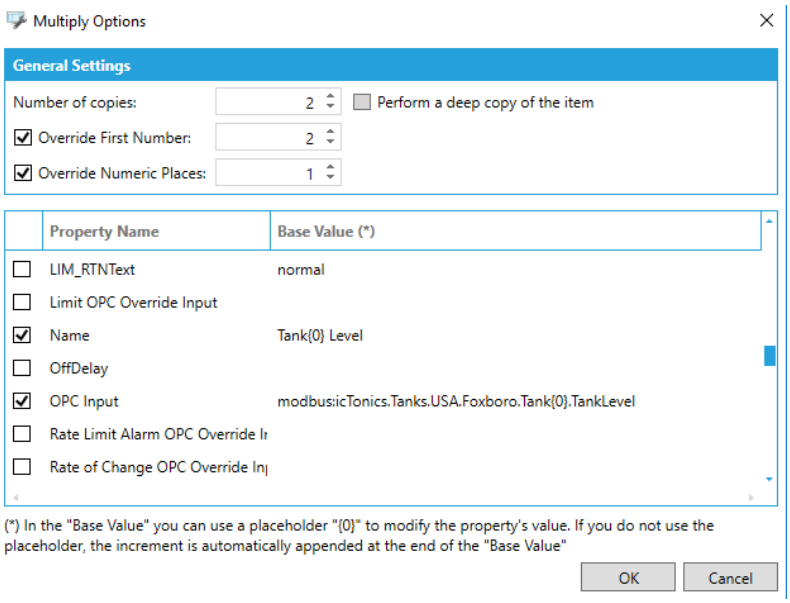


Figure L8.7 - Multiply Options Dialog Box

Note: As an alternative method, you could **Copy / Paste** to create multiple alarm tag entries.

7. From within the upper section of the split window, left-click on the **Churning Blade1 Status** alarm tag and select **Copy**.

8. Right-click on the **icTonicsAlarms** configuration in the tree control and select **Paste**.

9. Again, right-click on the **icTonicsAlarms** configuration in the tree control and select **Paste**.

10. Manually update the two new entries as follows.

Name	Churning Blade2 Status and Churning Blade3 Status
OPC Input	Change 1 to 2 and 3 respectively
Base Text	Change 1 to 2 and 3 respectively

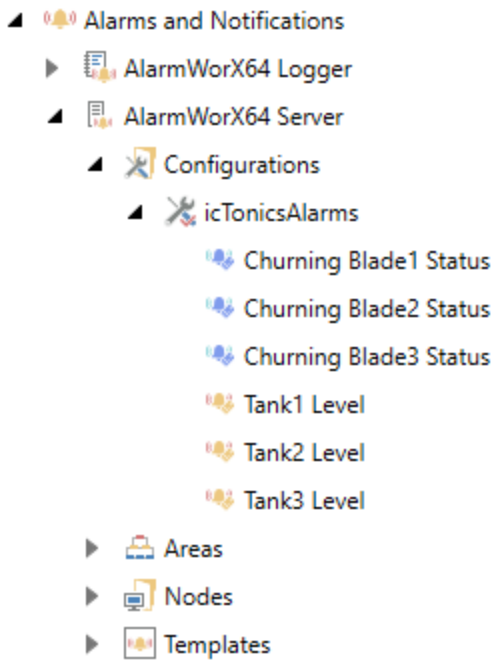


Figure L8.8 - IcTonicAlarms after Multiply and Copy / Paste

<u>This concludes this lab.</u>

AlarmWorX64 Viewer 9

The AlarmWorX64 Viewer is a 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 the polling rate of raw data to be evaluated for alarm states. The AlarmWorX64 Viewer subscribes to the AlarmWorX64 Server. The Viewer allows you to determine:

- Which alarms display on the screen (Condition).
- What the alarm display looks like (Appearance).
- How the alarms are grouped and sorted (Grouping).

9.1 Objectives

By the end of this chapter, you will become 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.

9.2 AlarmWorX64 Viewer

The AlarmWorX64 viewer, as indicated by its name, can be used to view alarms like those shown in Figure 9.1. You can also use it to acknowledge alarms and do simple alarm analysis. You can add an AlarmWorX64 Viewer onto a GraphWorX64 display. The AlarmWorX64 Viewer Control can also be inserted into any .NET Framework form.

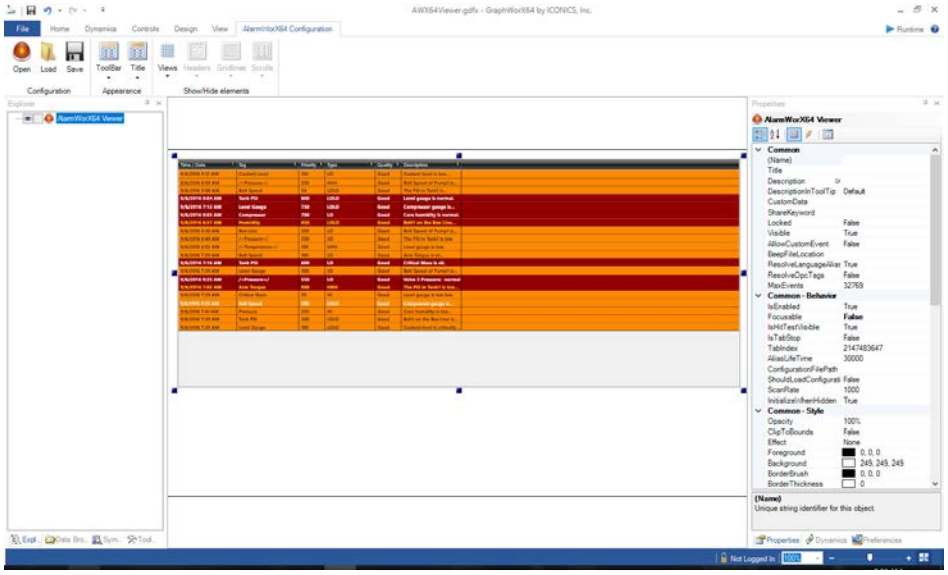


Figure 9.1 - AlarmWorX64Viewer Grid Inside GraphWorX64 (in Configure Mode)

If you like, you can display alarms in a grid or chart, like the one shown in Figure 9.2, 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.

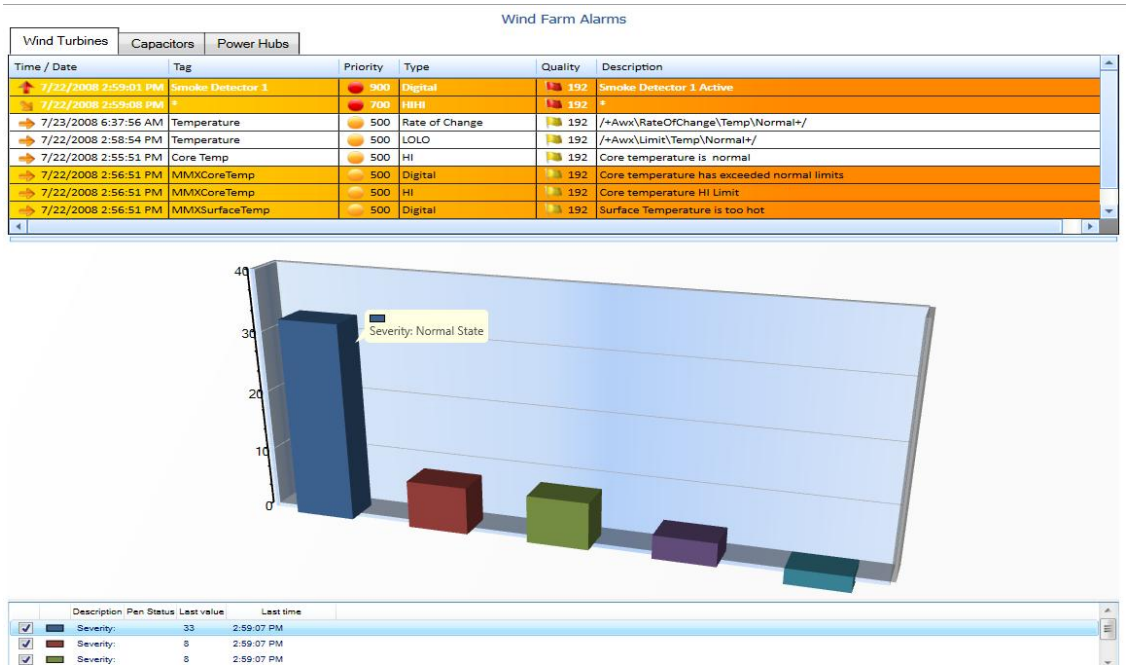


Figure 9.2 - AlarmWorX64 Viewer Grid with a Summary Chart

9.3 Configuring the Viewer

AlarmWorX64 Viewer can be configured within GraphWorX64. To open the configurator, double-click on the AlarmWorX64 Viewer object in a GraphWorX64 display.

The AlarmWorX64 Viewer configurator appears as shown in Figure 9.3 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. The AlarmWorX64 Viewer Control automatically adds at least one tab and one grid to the display which is automatically subscribed to the local Alarm Server. You can alter the display of the viewer, tab, or grid objects; but only the grid object and chart object can connect (*subscribe*) to an OPC UA or legacy OPC A&E alarm event source.

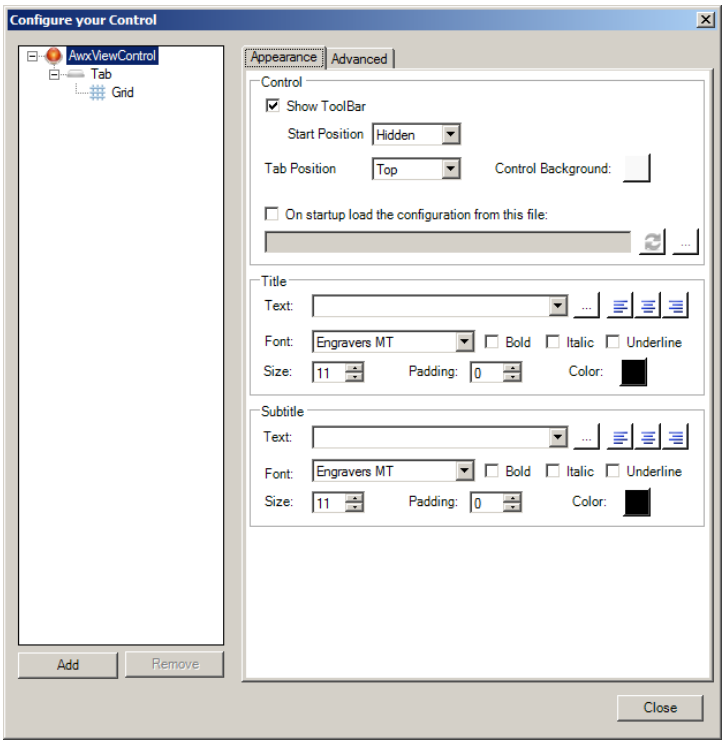


Figure 9.3 - Configuring the Alarm Viewer's Appearance Tab

In Figure 9.3, 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, or 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 all properties for the viewer as well. Note that all items in the navigation tree have an **Advanced** tab that lists all properties for that item.

9.3.1 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 9.4. Then you can edit the tab properties in the tab's Appearance and Advanced tabs on the right.

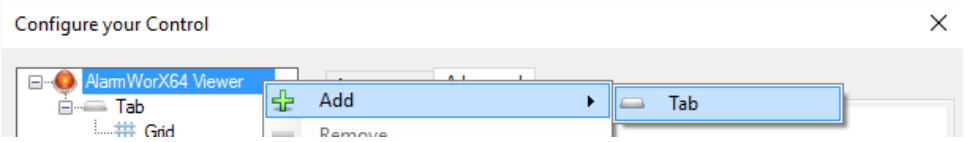


Figure 9.4 - Configuring the AlarmWorX64 Viewer

Using the Tab's Appearance tab, shown in Figure 9.5, 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 button next to the title's Text field. Clicking the button opens the Data Browser where you can assign a Data Point or Expression, Global Alias, language Alias (letting you translate tabs for a multi-lingual user base) or Simulation. You can also apply these property values to all of the tabs in the AlarmWorX64 Viewer.

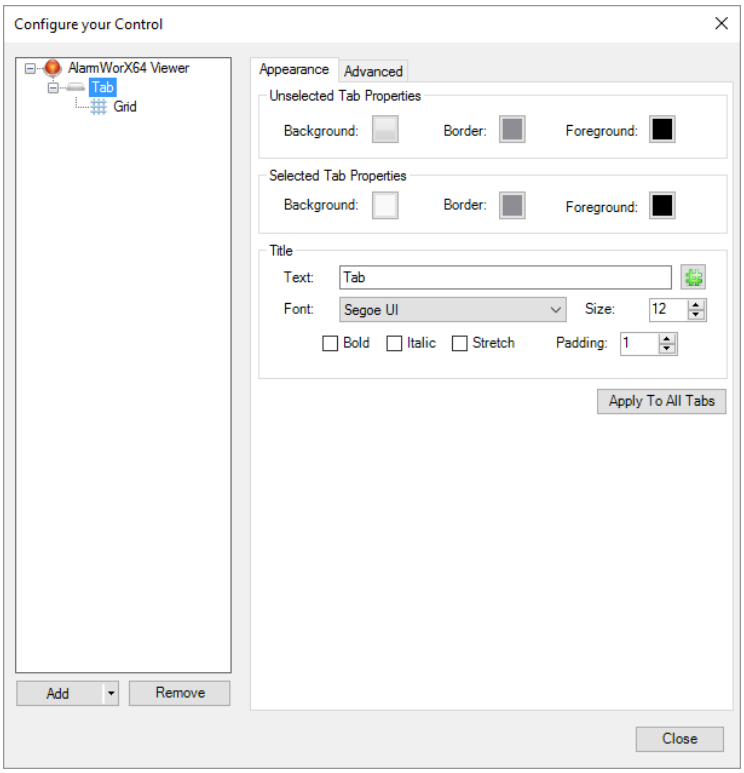


Figure 9.5 - Configuring AlarmWorX64 Viewer

9.3.2 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 9.6.

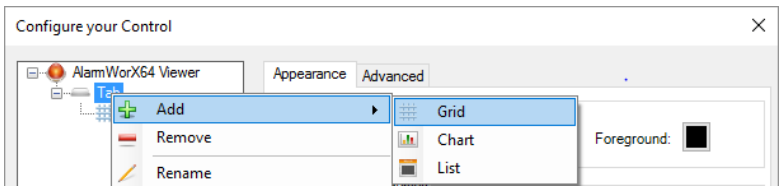


Figure 9.6 - Adding a Grid or Chart to a Tab

Adding a chart is like adding a grid, but with fewer options. The some chart properties are shown in Figure 9.7. 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.

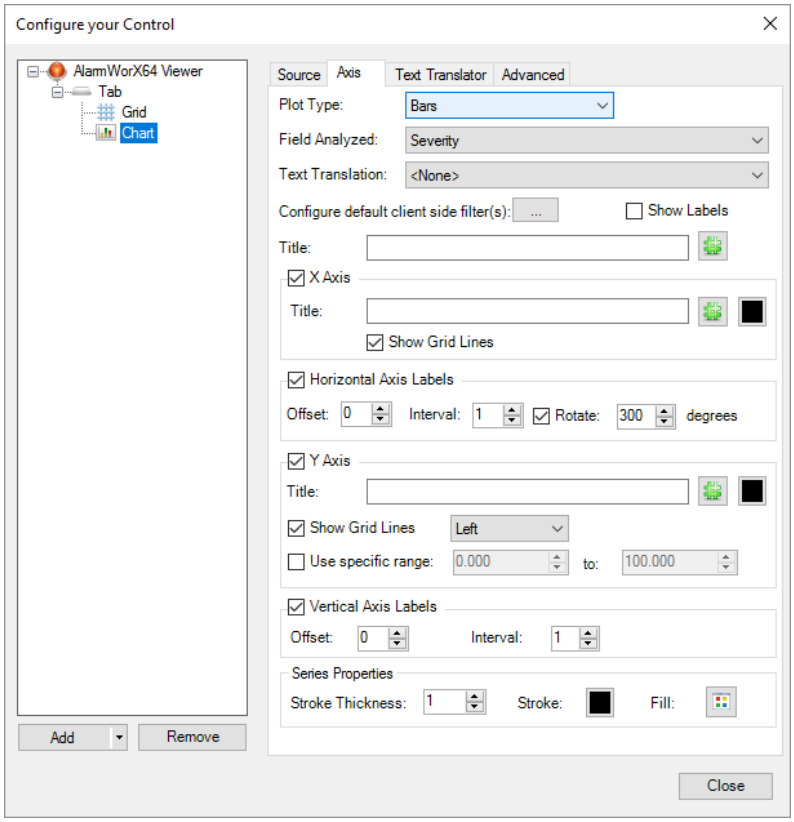


Figure 9.7 - Chart Added to a Tab in AlarmWorX64 Viewer

9.3.3 Adding an Alarm Subscription to a Grid or Chart

Grids and charts are the AlarmWorX64 Viewer component where you add the OPC tag alarms that you want the viewer to monitor. When you add an alarm, the grid or chart 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 and/or event server, which is the source of our alarms and/or events.

To make this connection, you must specify the alarm source. To subscribe to the alarm source, click on the grid or chart object in the tree control. As a result, the **Source** tab, shown in Figure 9.8, displays in the subscription properties where you define the alarm or event source.

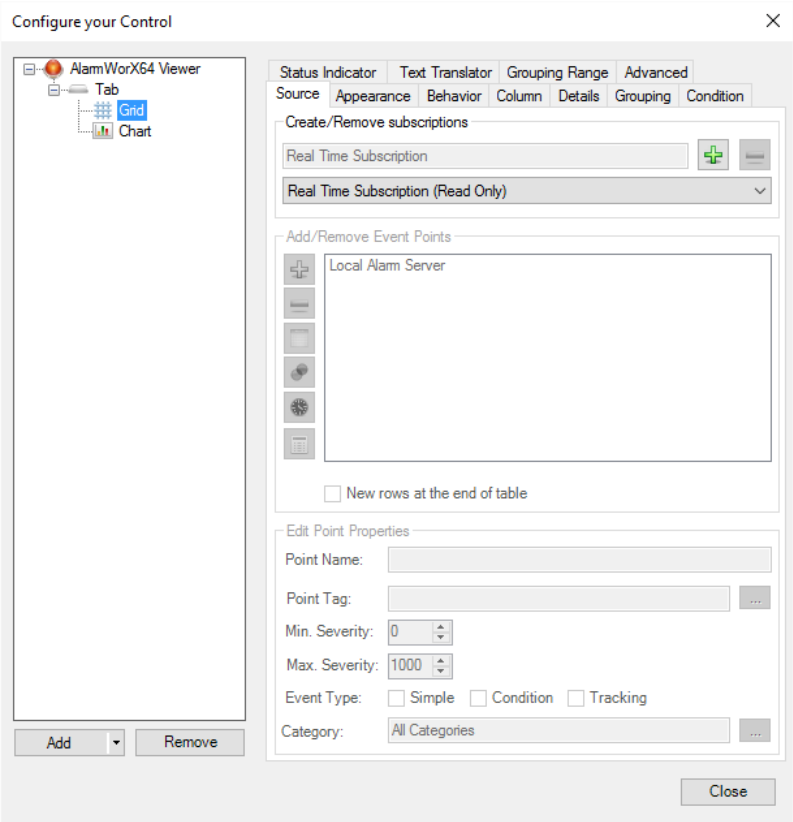


Figure 9.8 - Real Time Alarm Subscription

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 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.

Now 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 9.9.

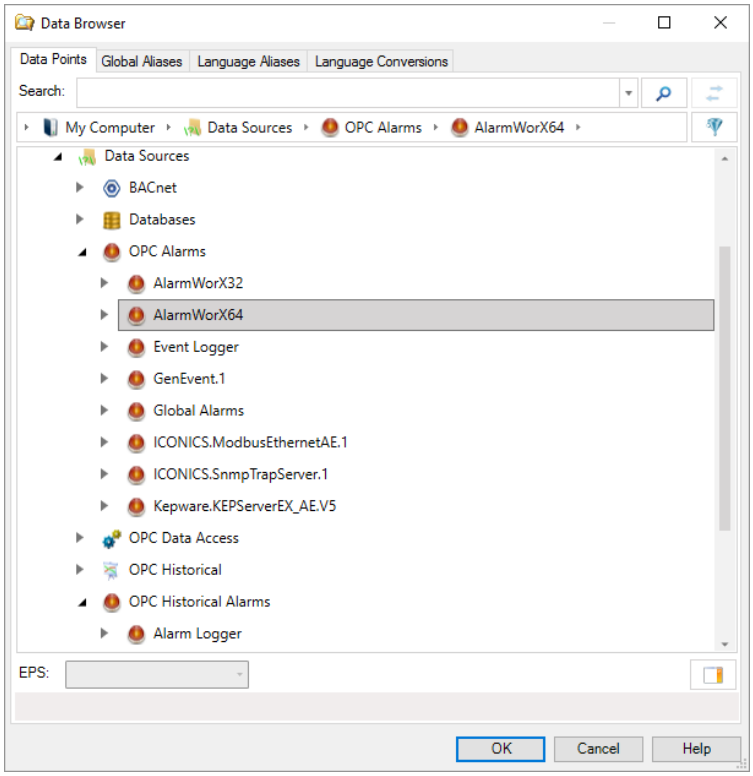


Figure 9.9 - Browsing for an Alarm Subscription

As you can see, there are many OPC servers in the list. Make sure you select the **AlarmWorX64** under **OPC Alarms** (not the one that reads AlarmWorX64 DA under OPC Data Access). You will also be able to subscribe to legacy alarm servers like AlarmWorX32 Server, if you have it installed, and the Alarm Logger under OPC Historical Alarms.

> **Note:** Historical alarm servers will replay a timeframe of your choosing. If you select the Historical Subscription 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 9.10 on the next page.

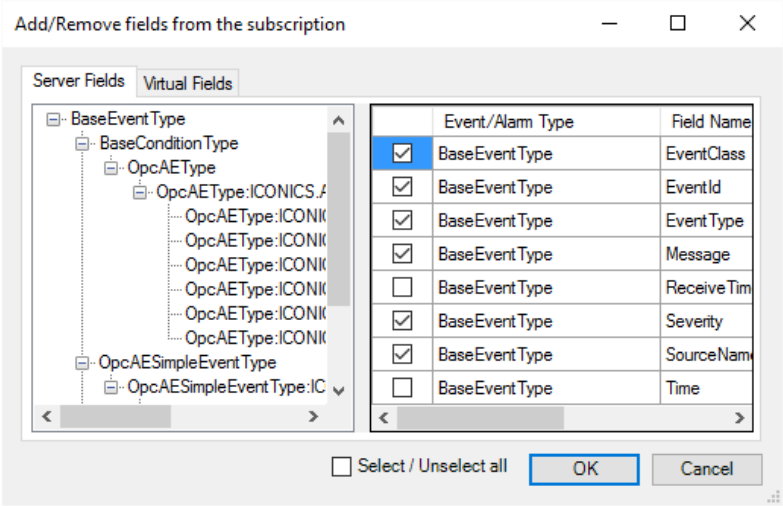


Figure 9.10 - Add or Remove Fields in an Alarm Subscription

9.3.4 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.

9.3.4.1 Appearance Tab

Use the grid's **Appearance** tab, shown in Figure 9.11, 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.

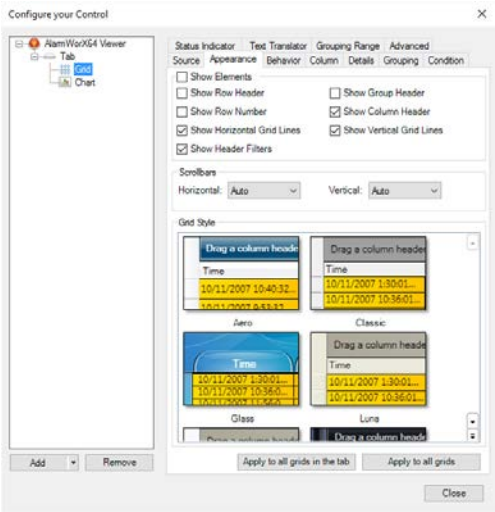


Figure 9.11 - Appearance tab for a Grid

9.3.4.2 Behavior Tab

The grid's **Behavior** tab is shown in Figure 9.12. On this tab are options for runtime features.

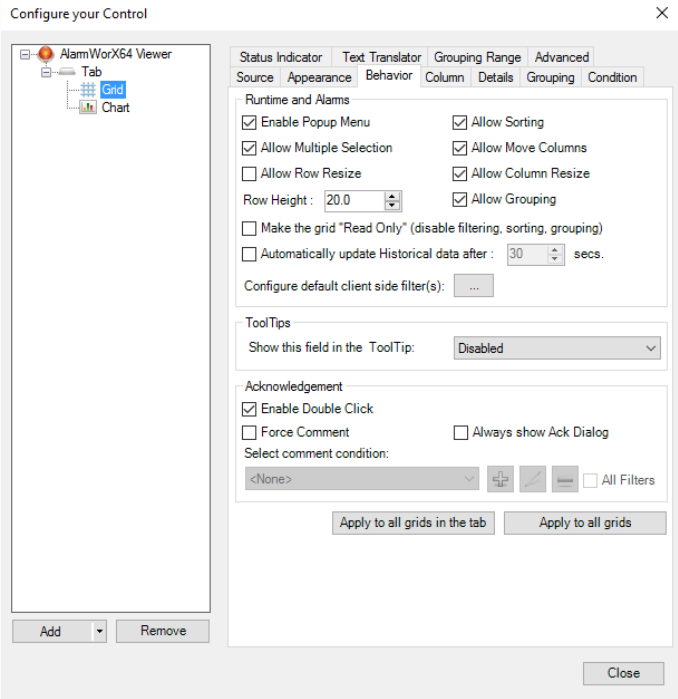


Figure 9.12 - Behavior Tab for a Grid

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 ellipsis […] button next to the "Configure default client side filter(s):" label to bring up the **Client Side Filter** dialog. Click on the edit button to edit the existing filter. Click the remove button to delete a selected filter. When creating your own Client Side Filter you will find 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 check boxes.

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 9.13 shows an example of a filter. Any filters you create within this dialog are usable throughout this instance of the control.

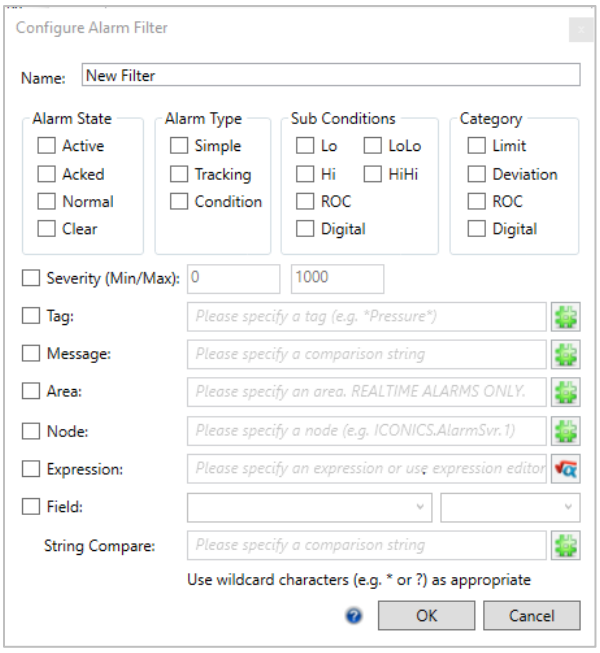


Figure 9.13 - Filter in the Edit Shared Filter Dialog

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.

9.3.4.3 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 9.14, 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.

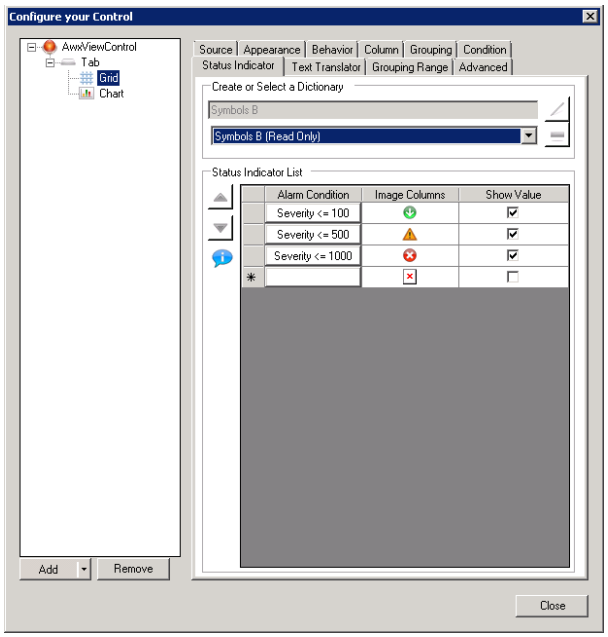


Figure 9.14 - Status Indicator Tab for a Grid

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 9.13. 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 9.15, to apply the status indicator to any column.

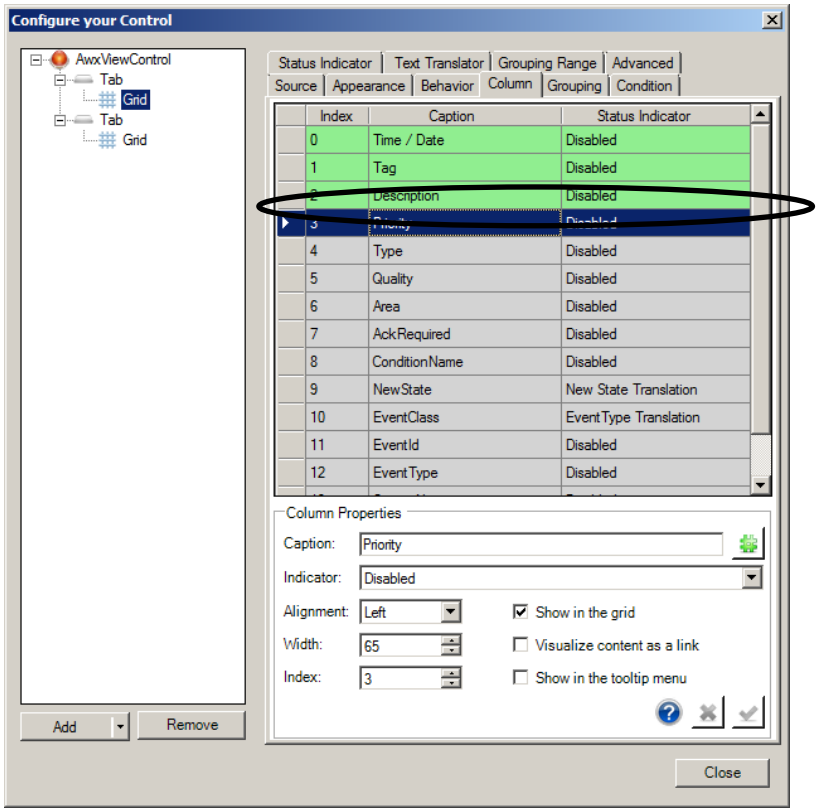
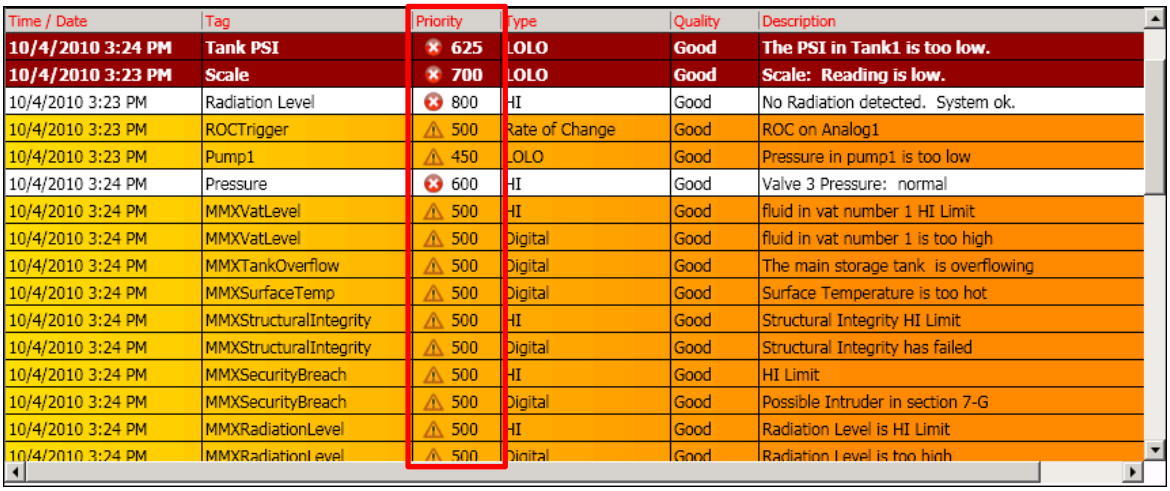


Figure 9.15 - Column Tab Showing a Column's Status Indicator

The result is shown in Figure 9.16. 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 9.16 - Status Indicator in Runtime

9.3.4.4 Column Tab

The grid's **Column** tab, shown in Figure 9.17, 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 type the up or down arrow on the keyboard to move the row 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.

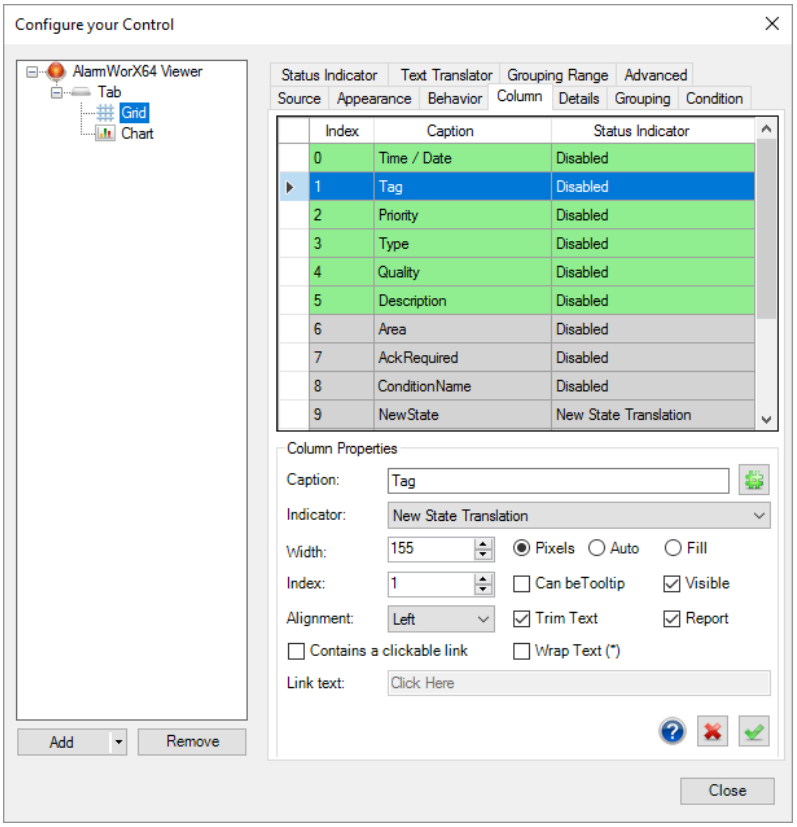


Figure 9.17 - Column Tab for a Grid

9.3.4.5 Grouping Range Tab

When there is considerable 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 9.18 below) allows you to set up a set of filters that you can apply to a column later, for proper grouping.

As an example there is one preconfigured grouping range in the grid called "Severity Range (Read Only)" – it is a read-only range that you cannot modify. It is visible by clicking the down arrow on the right below the minus sign. The items in this grouping range are filters that determine the severity of the alarms. To accommodate different severity range needs, there are five different filters for alarms.

To create your own group range, you must select the Add button on the right, then type the name your new range replacing "New Grouping Range" in the textbox. In the new **Criteria List** section, click on the down arrow in the new Alarm Condition column to select a predefined filter. Type a custom caption for this filter if you wish.

You can also create a new filter by opening the Filter dialog. Select the Add button on the right in the **Edit Filter** section. You can use this dialog to create your filter; it is the same filter dialog used for status indicators, as shown back in Figure 9.13.

Once you create each filter, you can give a descriptive name for that filter in the **Caption** column. 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.

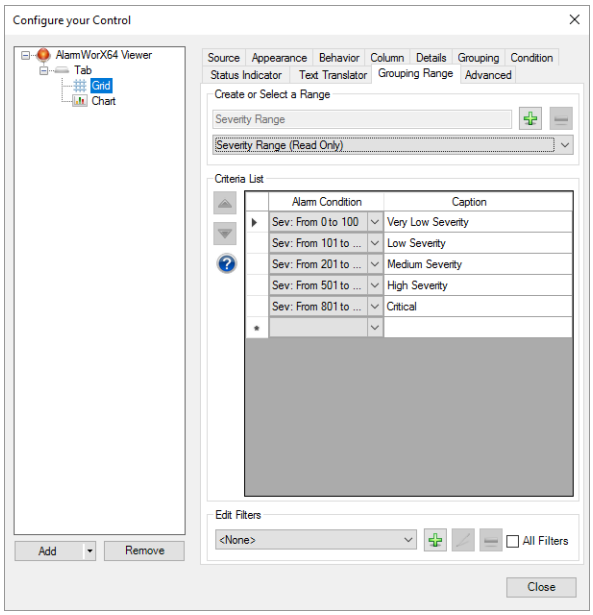


Figure 9.18 - Grouping Range Tab

9.3.4.6 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 9.19, the grid will first group by the **Source Name** of the alarm, then by the **Time/Date**, and lastly by the **Event ID**.

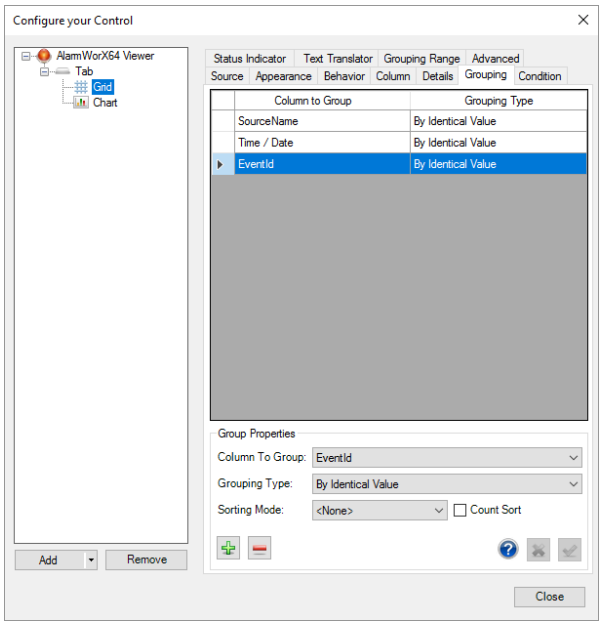


Figure 9.19 - Grouping Tab

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 9.19, we are grouping the SourceName column by **Identical Value** (which is the **default** grouping type).

Since Severity Range was preconfigured, by you have two grouping types: **By Identical Value** and **Severity Range**. You can also select other grouping types that were defined using the 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.

9.3.4.7 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 Severity (which is OPC-defined to have a value from 1 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.

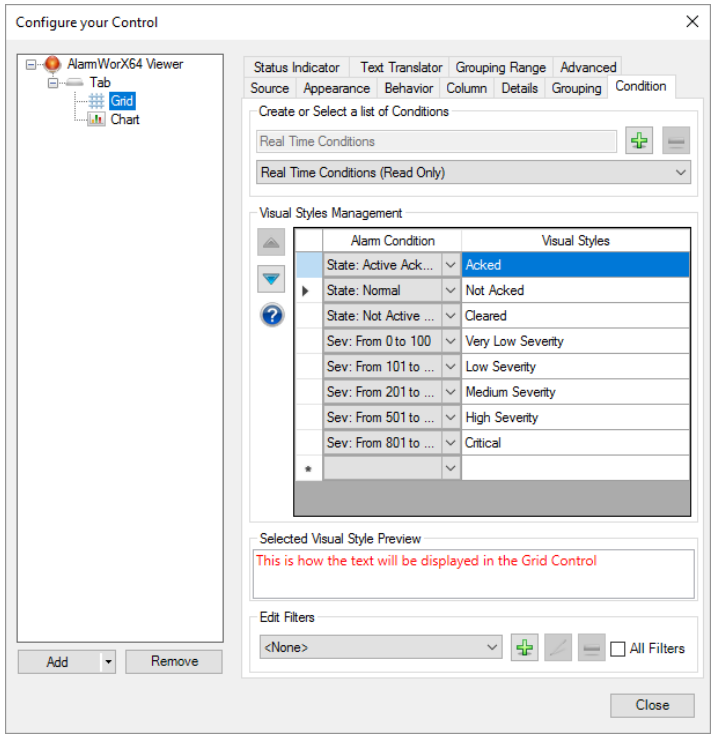


Figure 9.20 - Condition Tab

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, since it looks exactly like Figure 9.13. The Visual Style tab, shown in Figure 9.21, provides options for color, blinking, and beeping – all ways to get the runtime operator's attention when an alarm condition occurs.

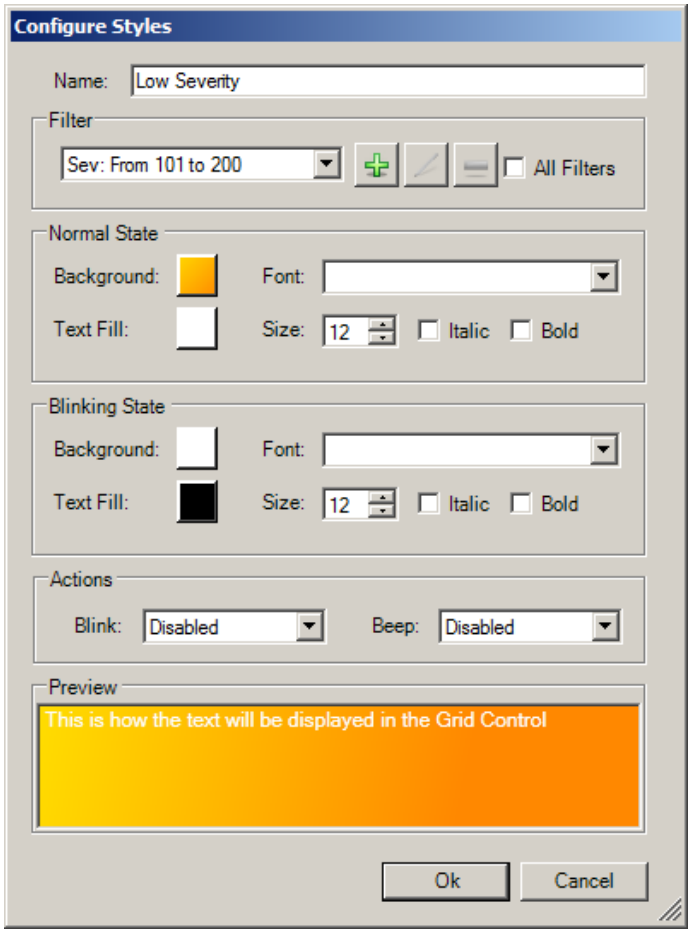


Figure 9.21 - Configure Styles Dialog for Conditions

9.3.4.8 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 9.22.

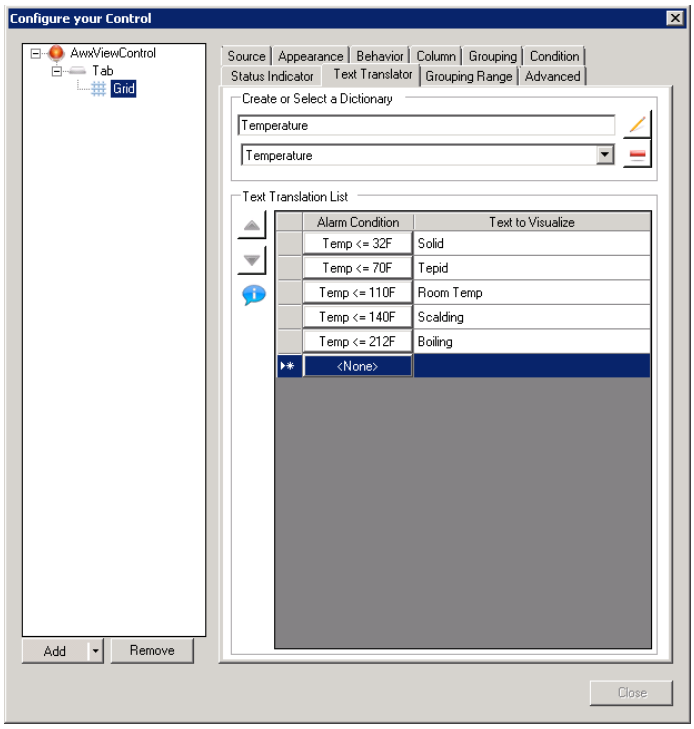


Figure 9.22 - Text Translator Tab

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, click the empty box under Alarm Condition. This opens the Filter dialog box that you are now so familiar with (the one shown in Figure 9.13). When you return to the Text Translator tab, next to the filter you can enter text that the runtime operator will see.

9.4 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 9.23. 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.

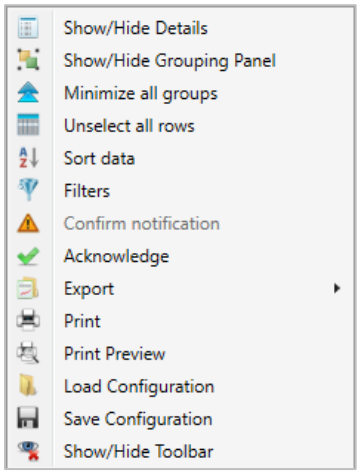


Figure 9.23 - Alarm Viewer's Right-click Runtime Menu

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 9.24, 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 9.24 - 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 L9

9.1 Lab Overview

In the previous lab, we 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. The company, 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**

9.1.1 Task Checklist

In this lab, you will complete the following tasks:

- Set up an AlarmWorX64 Viewer Control in GraphWorX64.
- Configure an alarm subscription.
- Create and apply a client-side filter.
- Add flash to alarms with a condition.

9.1.2 Support Files

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

- icTonicsMain.gdfx
- icTonicsHome.gdfx
- icTonicsTankMonitor.gdfx

9.2 AlarmWorX64 Viewer Control

The AlarmWorX64 Viewer Control is what you use to view alarms in GraphWorX64. Let's configure the AlarmWorX64 Viewer for use with the icTonicsMain display.

1. Open a new **GraphWorX64** display**.**
2. Open the **File** tab and select **New Settings**.

3. In the **New Display** window under **Advanced**, set **Layout** = **Grid**. The **Properties** pane should now indicate **This Display : Grid Panel** at the top.

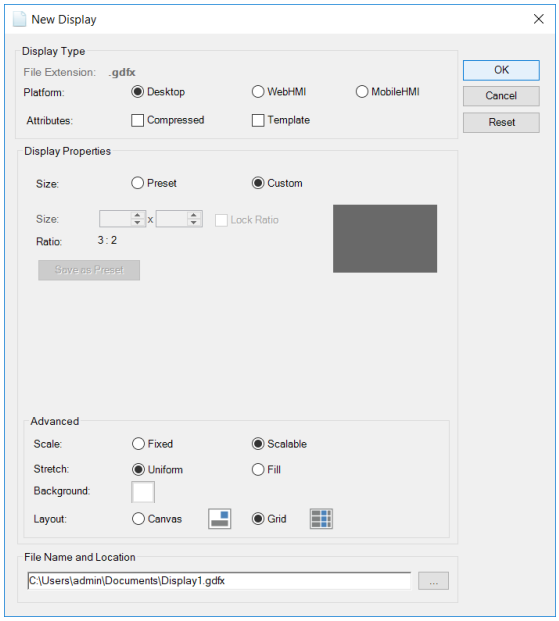


Figure L9.1 – New Display, Scalable, Grid Settings

4. Click on OK to set this display.

5. Select the **Controls** tab.

6. Select the **AlarmWorX64 Viewer** and draw the control to fill most of the display.

7. While the AlarmWorX64 Viewer is selected, the ribbon will reflect the **AlarmWorX64 Configuration** tab and ribbon.

   Note: You may have to de-select and re-select the viewer control in order for the AlarmWorX64 Configuration ribbon to appear.

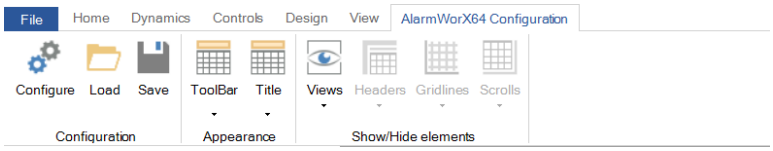


Figure L9.2 – AlarmWorX64 Configuration Tab and Ribbon

8. In the **Configuration** section of the AlarmWorX64 Configuration ribbon, select **Configure** or you could double-click in the viewer to launch the **Configure your Control** dialog.

9. In the **Configure your Control** dialog box, enable the **Title** and set the text to **ALARMS** in the Appearance tab.

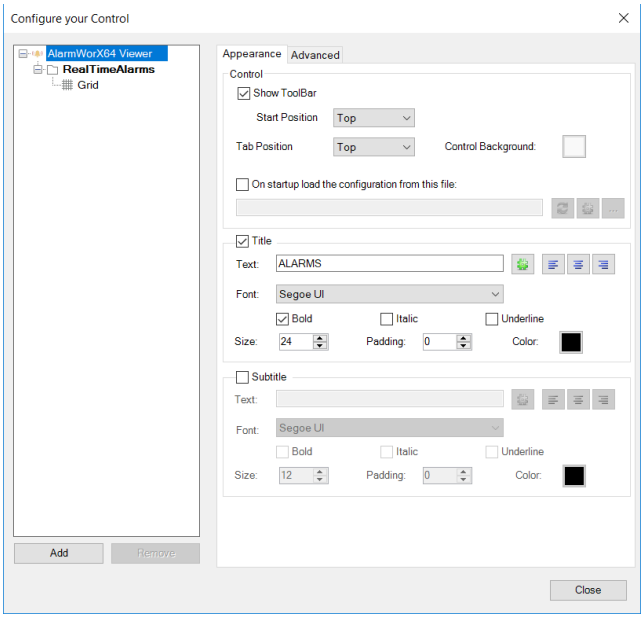


Figure L9.3 – Alarm Viewer Control Configure Dialog Box

10. Select the **Grid** selection in the left-hand navigation pane.

11. The right-hand pane should now show the **Source** tab. Under the **Create/Remove subscriptions** section, click on the green plus sign (add button) to create a new **Real Time Subscription**.

12. In the field just below the header, we can leave the name **New Subscription** for this exercise. Click on **Close**.

13. Click on **Runtime** in the upper right-hand corner. You can easily go back to Configuration mode by clicking the **Configuration** button in the same place.

Note: Ctrl+M is the shortcut to toggle between Runtime and Configuration.

14. With the icTonicsAlarms display in Runtime, we will generate alarms by opening the **icTonicsTankMonitor** display in another instance of GraphWorX64. Toggle the **Churning Blade On / Off** button in Runtime. This will turn the Churning Blade On (no alarm) and Off (in alarm).

15. Right-click on the alarms in the viewer and select **Show/Hide Toolbar**. Feel free to experiment by selecting various options (Acknowledge for instance) in the dropdown.

16. Select **Ctrl+M** to go back into **Configuration** mode.

17. Select the **AlarmWorX64 Viewer Control**.

18. Click on the disk icon in the Configuration section of the **AlarmWorX64 Configuration** ribbon to Save the alarm viewer configuration. Enter the name **icTonicsAlarmViewer** and the **Save as type**: .awxx in the **C:\Training** folder.

19. Try other **Save as types:** by clicking on the down arrow. Save as the **Mobile** file type (.awxxp) for example.

9.3 Dashboard and Extended Alarm Configuration

9.3.1 Add the Alarms Button

In this section, you will modify our dashboard to view alarms, as well as customize the control to suit the project.

1. From within **GraphWorX64**, select **File → Open** from the menu. You should be able to open the **icTonicsMain.gdfx** display located in the C:\Training folder.

   Note: You should save your work periodically as you progress through the labs.

2. We will proceed to add the Alarms button by duplicating another button and modifying the Dynamics. Select the **Tanks** button then type **Ctrl + D**.

3. The duplicate button will be selected and appear to the right of the original. Modify the **Properties** tab → **Common** → **(Name)** = **AlarmsButton**.

4. Modify the **Dynamics tab** → **Dynamic Behavior** → **CommandParameters** → **FileNam**e = **icTonicsAlarms.gdfx**.

5. Expand the **AlarmsButton** → **Group** → **Grid Panel** in the **Explorer** pane on the left and select the **Rectangle**.

6. Open the **Symbols** tab in the **Explorer** and search for **Alarm**. Choose the **Alarm Bell** and drag and drop it into the **Rectangle** of the **AlarmsButton**.

7. The symbol will likely fill the entire Rectangle. Resize the symbol by selecting one of the corner handles and while holding the **Shift** key, move the handle to make the symbol small enough to fit into the upper space of the Rectangle. The ButtonsPanel should look similar to Figure L9.4 below.

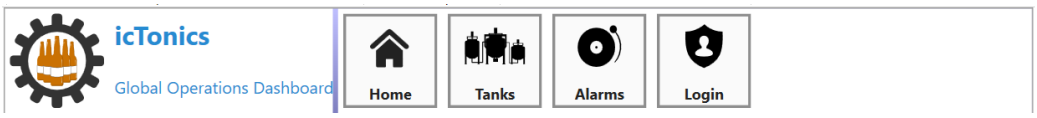


Figure L9.4 – Adding the Alarms Button

9.3.2 Configure the AlarmWorX64 Viewer

1. 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 below the template.

   Note: The zoom control is in the lower right section of the GraphWorX64 configurator. You may want to zoom out to see the entire frame of the display.

2. Double-click on the AlarmWorX64 viewer control to bring up the **Configure Your Control** dialog box.

3. In the **Configure your Control** dialog box, click on the **Grid** selection in the left-hand pane.

4. 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**.

5. In the field just below the header, replace the **New Subscription** text with **icTonicsAlarms**.

6. 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.

7. Select the **Appearance** tab, and then in the **Grid Style** section, select the "**Glass style**".

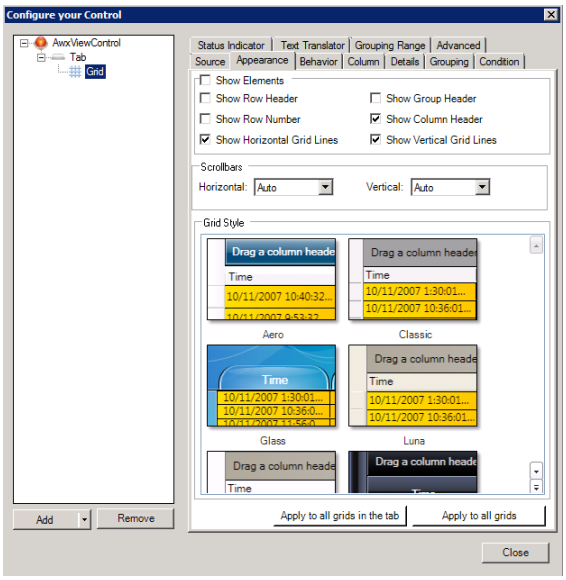


Figure L9.4 – Selecting a Style for the Grid

8. 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.

9. For the **Type** and **Quality** fields, uncheck the property **Is Visible** and apply these changes. These two fields will change to a grey color. Your Column tab should resemble the following figure.

   Note: Until you apply the changes, the Close button at the bottom of the dialog box may be grayed out.

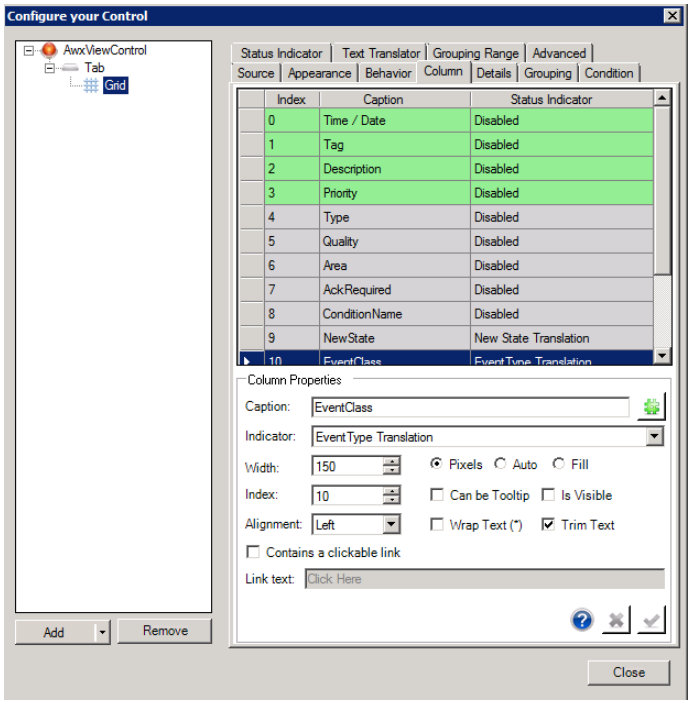


Figure L9.5 – Alarm Viewer Configuration: Column

10. Click on the **Condition** tab. Notice that there are preconfigured conditions here. We cannot modify these because they are read-only.

11. We will create a new **Alarm Condition** based on a filter for alarms with a **severity of 900** or higher. The conditions are applied from the top down.

12. In the right-hand pane, under the **Create or Select a list of Conditions** section, click on the green plus sign (add button) to create a **New Alarm Condition**.

13. At the bottom of the list (the row with the * in the first column), double click in the **New Style** field and enter the name **icTonics Condition**.

14. Click on the **ellipses** […] to bring up the **Configure the style** dialog box.

15. 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.)

16. In the **Actions** section, change **Blink** to **Slow**.

17. Notice the **Preview** section blinks.

18. When you're done making changes, click **OK**.

19. In the bottom of the dialog, click on the **Add** (green plus sign) button in the **Edit Filters** section. The **Configure Alarm Filter** dialog box will now be displayed.

20. Name the filter **SeverityBelow900Filter**, select the **Severity** checkbox and in the second data entry, then enter a maximum value of **900**.

21. Your **Configure Alarm Filter** dialog should look similar to the figure below.

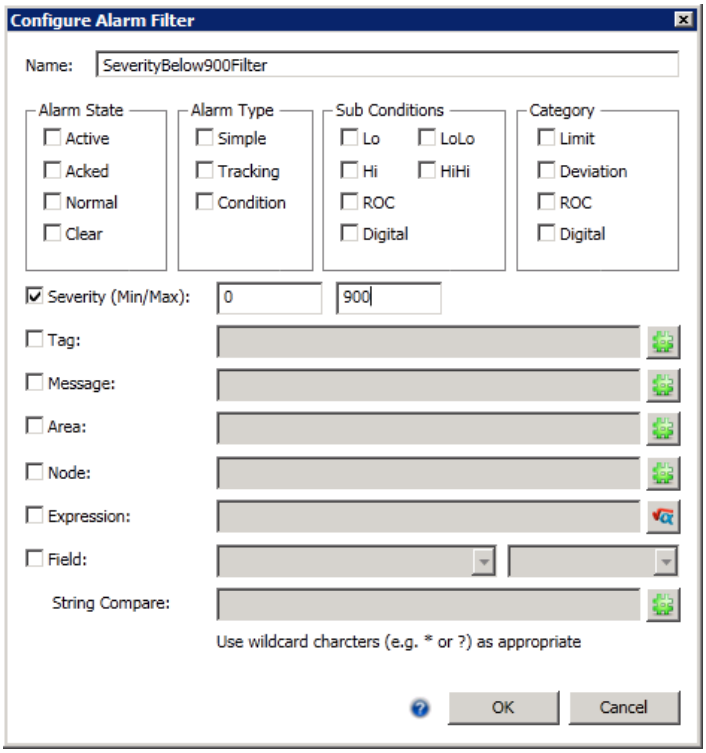


Figure L9.6 – Configure Alarm Filter

22. Click **OK** to commit this filter.

23. In the Alarm Condition column, expand the drop down list and select **SeverityBelow900Filter**.

24. The completed Condition dialog should resemble the following figure.

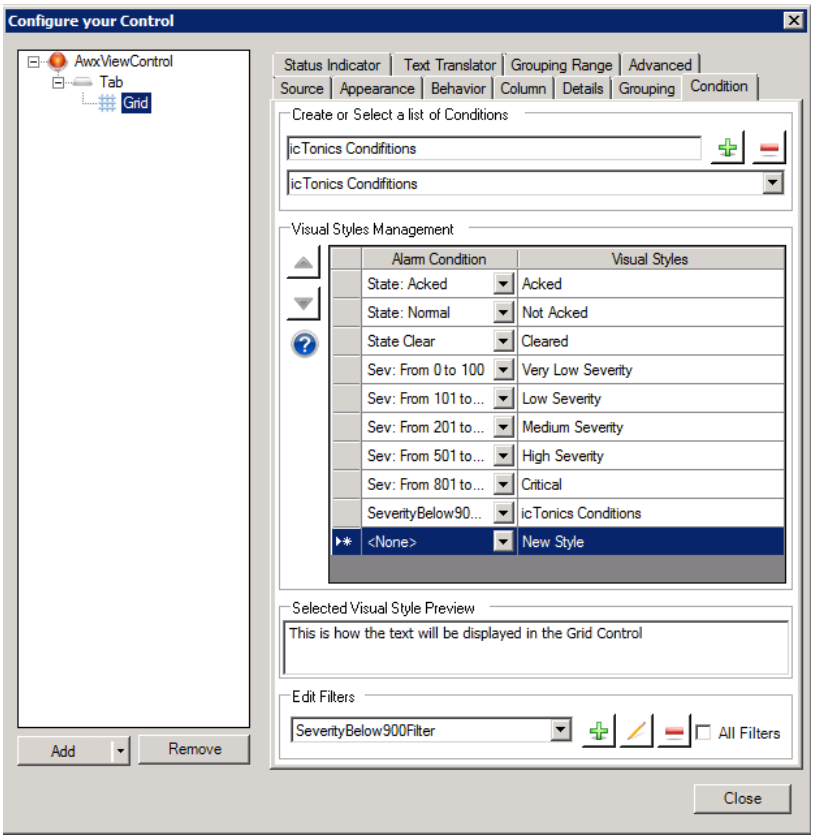


Figure L9.7 – Completed Condition Dialog

25. Using the blue up arrow button on the left side, move the **icTonics Condition** that you have just created to the one below the High Severity condition.

26. Click **Close** to close out of the configuration dialog box.

27. **Save** the display and go into **Runtime**. You may notice that you already have alarms.

28. Right-click on the Alarm Viewer and select **Show/Hide Toolbar** to bring up the toolbar.

29. Click on the **Filter Data** icon (blue funnel). We will now apply a filter to view the Tank level alarms only.

30. Click on the **Add** button (green plus sign) to create a new filter. In the **Configure Alarm Filter** dialog box, change the Name to **Tank Levels**.

31. Check the **Tag** checkbox and write inside ***Level*** (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.

32. Your filter should look similar to Figure below. Click on **OK** when you are done.

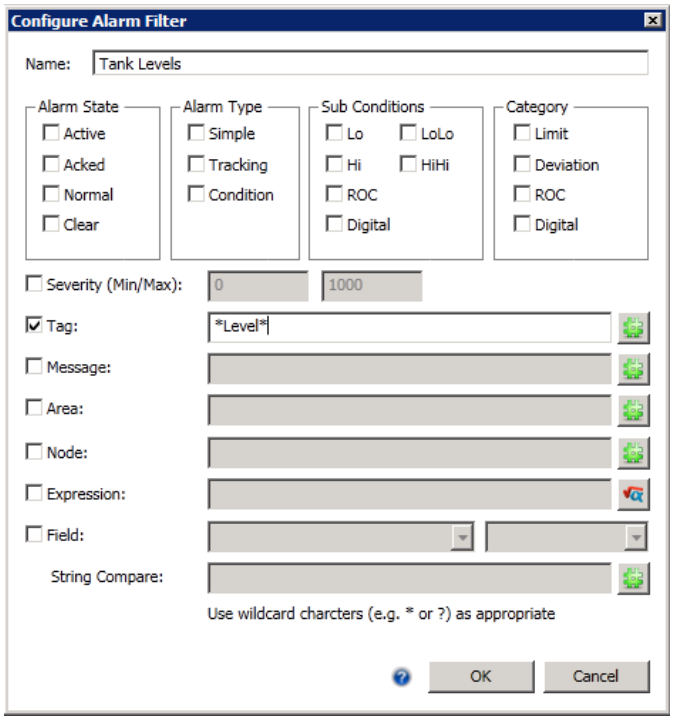


Figure L9.8 – Alarm Filter

33. Select (check) the **Tank Levels** filter and click on **OK**.

Note: To open a second instance of GraphWorX64, you can right-click on the GraphWorX64 icon in the Windows taskbar and select GraphWorX64.

34. In a second instance of GraphWorX64, open the **icTonicsTankMonitor.gdfx** display.

35. Make sure at least one valve is open.

36. Set the tank **Level Setpoint** to **99** and hit enter. This will generate a HiHi alarm.

37. Take a minute to experiment and familiarize yourself with the Alarm Viewer control in runtime.

9.3.3 icTonics Alarm Viewer

In this section, we will configure filtering in the **icTonicsAlarms** display.

1. Open the **icTonicsAlarms.gdfx** display in GraphWorX64.
2. Double-click on the control to open the **Configure your Control** dialog.
3. Select **Grid** in the left-hand pane and click on the **Behavior** tab.
4. Click on the button to the right of the **Configure default client side filter(s)** field.
5. From the **Select a filter from the list** dialog, click on the Plus sign to open the **Configure Alarm Filter** dialog box.
6. Name this filter **Tanks**.
7. 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.
8. Click **OK**.
9. Check the **Tanks** checkbox and Click **OK**.
10. Click on **Close** to close the **Configure Your Control** dialog box.
11. **Save** the display. If you go into runtime now only Tank1 alarms will show up. This is because the default **<#AlarmFilter#>** global alias value is **Tank1**. We will work with Global Aliasing in detail later in this training.

This concludes this lab.

AlarmWorX64 Logger 10

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 a SQL Server database.

10.1. Objectives

After completing this chapter, you should know how to:

- Startup 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.

10.2. Architecture

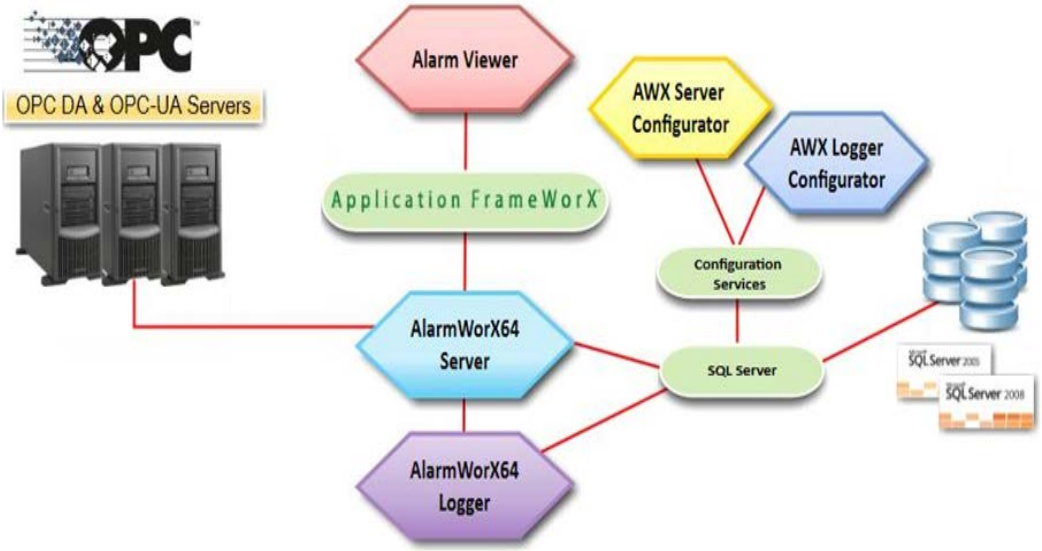


Figure 10.1 – Alarm Logger Architecture

10.3. Starting the AlarmWorX64 Logger Configurator

You will find the AlarmWorX64 Logger configurator in the Workbench under **Alarms and Notifications**.

- From the Windows desktop, click on Start → All Apps → ICONICS Product Suite → Workbench.
- In the Project Explorer of the Workbench, expand Alarms and Notifications → AlarmWorX64 Logger → Configurations. You will see a screen like Figure 10.2.

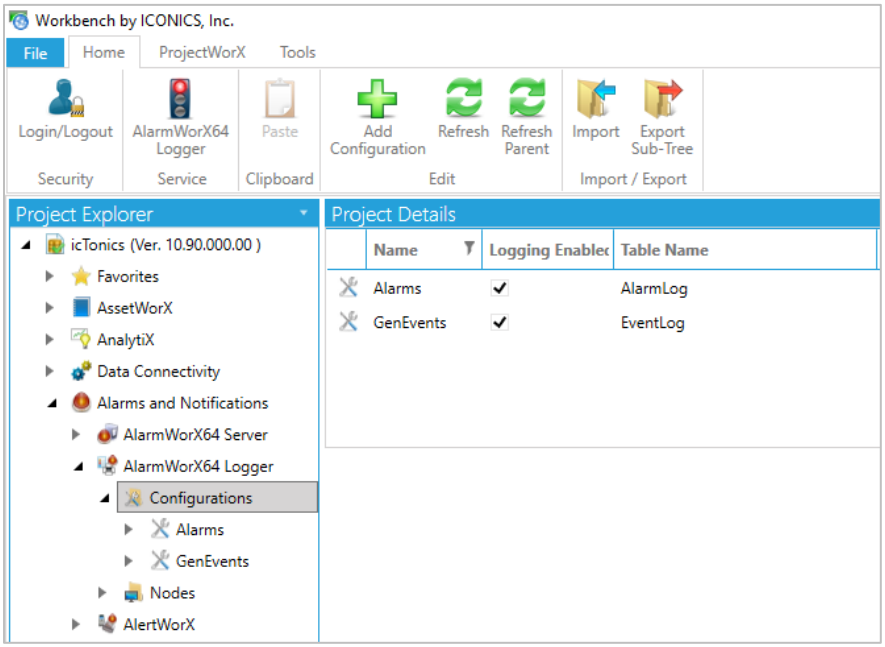


Figure 10.2 - AlarmWorX64 Logger Configurations

10.4. 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 selecting **Add Configuration**. Alternatively, you can right-click on **Configurations** in the AlarmWorX64 Logger tree of the Project Explorer and select **Add Configuration**.

The now-familiar **[New Configuration]** will open in the Project Details panel for 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.

10.4.1. 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 **Add Configuration** from the dropdown menu. Alternatively, on the **Home** tab, you can click on the **Add Configuration** item on the ribbon.

When you do this, Project Details opens the Alarm Log Configuration tab, which displays the configuration properties, as shown in Figure 10.3.

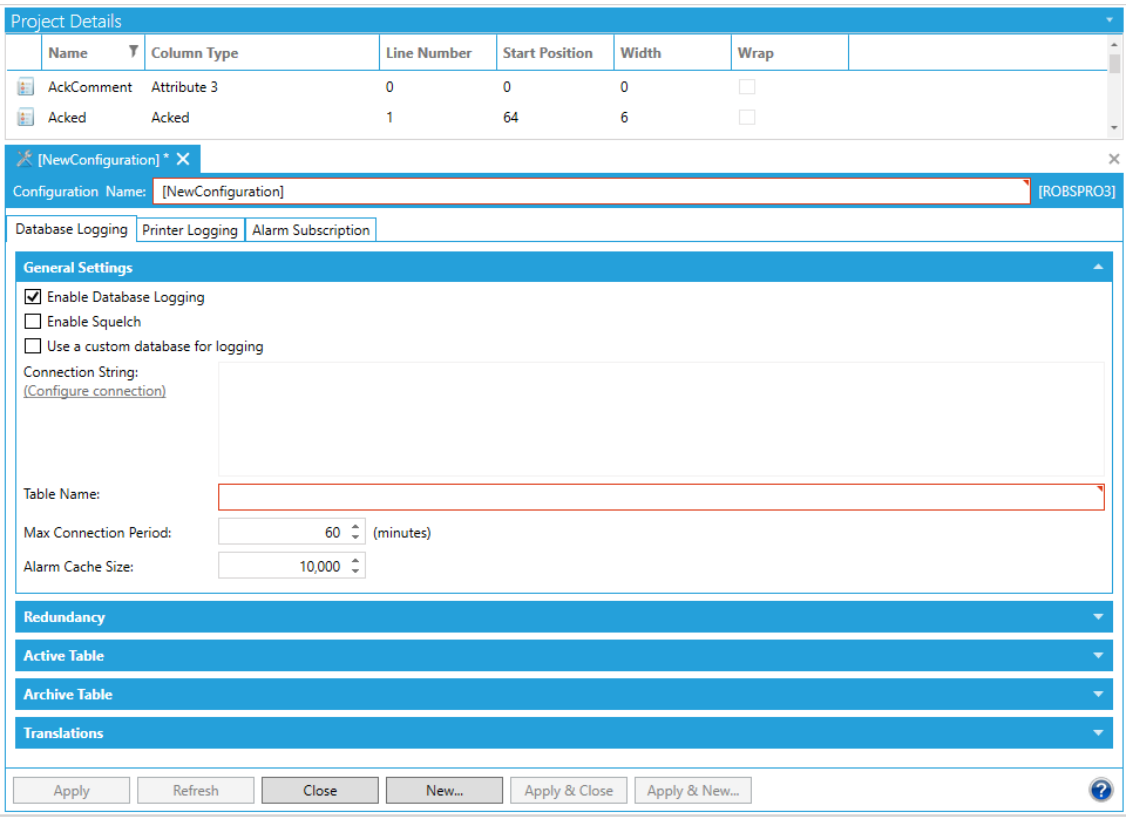


Figure 10.3 - New Configuration in AlarmWorX64 Logger

10.4.2. AlarmWorX64 Logger Subscriptions

Once you create a configuration, by default it is automatically subscribed to the local Alarm Server, as shown in Figure 11.4. If an additional subscription is needed, simply click on the gray bar [**+ Click here to add new item**] to create a new alarm subscription. The **Logged Alarm Tag** field will highlight and the Data Browser icon, , will appear allowing you to select another alarm server and/or area.

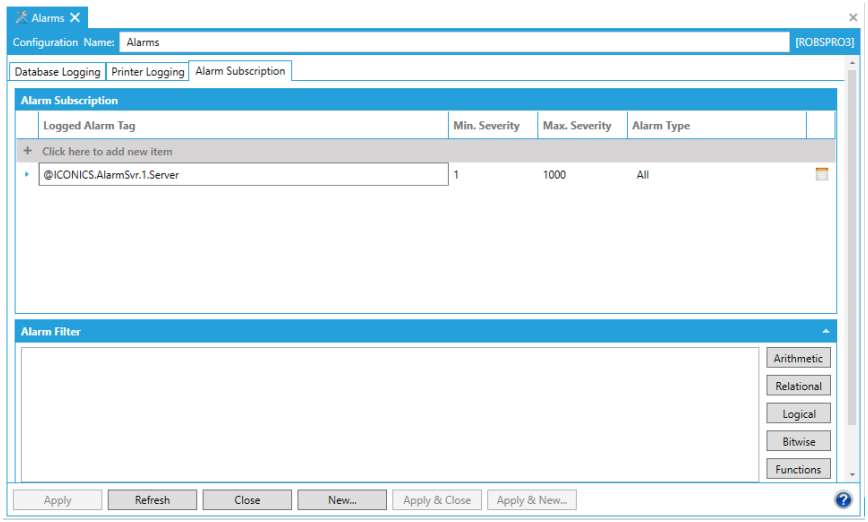


Figure 10.4 – Default Alarm Subscription

You can delete a subscription by highlighting it and using the **Delete** key.

10.4.3. 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 and Attributes** button to open the subscription's properties dialog, which is shown in Figure 10.5.

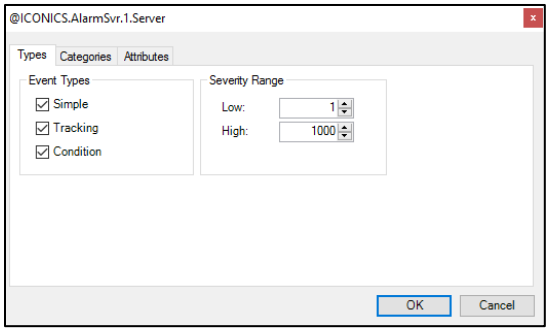


Figure 10.5 - Server-side Alarm Subscription Filter Window

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 later tabs is that if nothing is in the subscribed section, you will be subscribed to all available categories and attributes. However, if you add any categories or attributes to the subscribed side, then you will only receive alarms and event in those subscribed categories or attributes.

10.4.4. 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 below the Alarm Subscription, as shown in Figure 10.6.

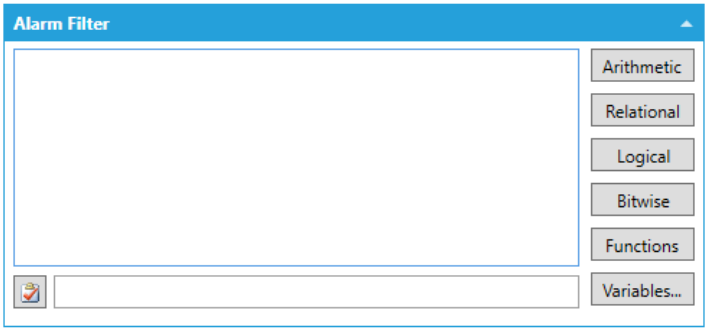


Figure 10.6 - AlarmWorX64 Logger Filter

You can start typing in the **Alarm Filter** box to use the Advanced Expression engine. Or you can click on the buttons to build your Expression.

Why two different types of filters? Use the *server-side filter* to conserve 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.

10.4.5. 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 10.7 shows this tab with the General Settings sub-tab on top. Notice that there are four additional sub-tabs: Redundancy, Active Table, Archive Table and Translations.

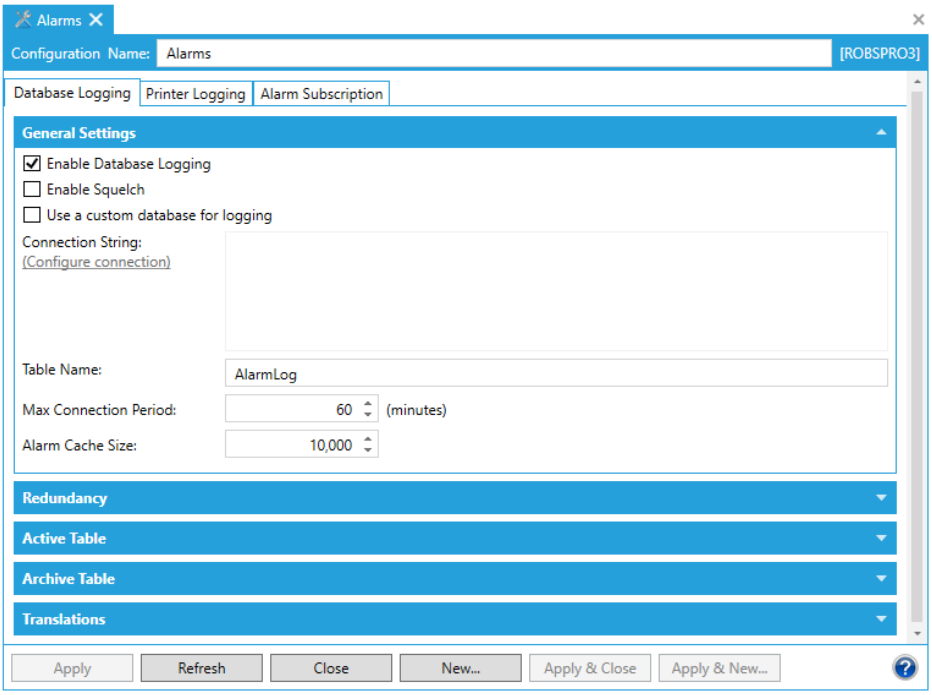


Figure 10.7 - AlarmWorX64 Logger Database Connection

If you want data logged to a different database, you will first need to check the **Use a custom database for Logging** checkbox and configure the database connection by clicking on the **(Configure connection)** text. This opens the **Select a logging database** dialog box as shown in Figure 10.8, where you simply select the **SQL Server** you want to use and the **Database** to which the alarms will be logged. Click on the **Test Connection** 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 Connection String field in General Settings.

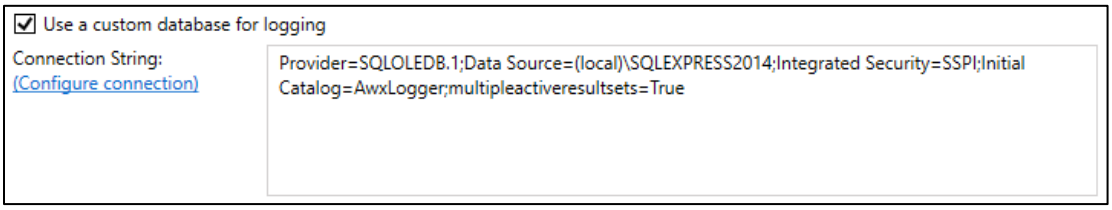


Figure 10.8 – Connection String

Although the connection string is automatically created when you configure the connection, 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 edit it in the Connection String field.

- **Table Name:** Enter the name of the active table.

10.4.6. Table Management

Alarms or Events are logged to a master, active table in the database. If there are no restrictions placed on the size of the active table, alarms 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 **Active Table** tab, shown in Figure 10.9. The active table can be restricted to a maximum size determined by number of records, **Max Records**, or a time interval, **Max Interval**, and/or database size, **Max Records Size**.

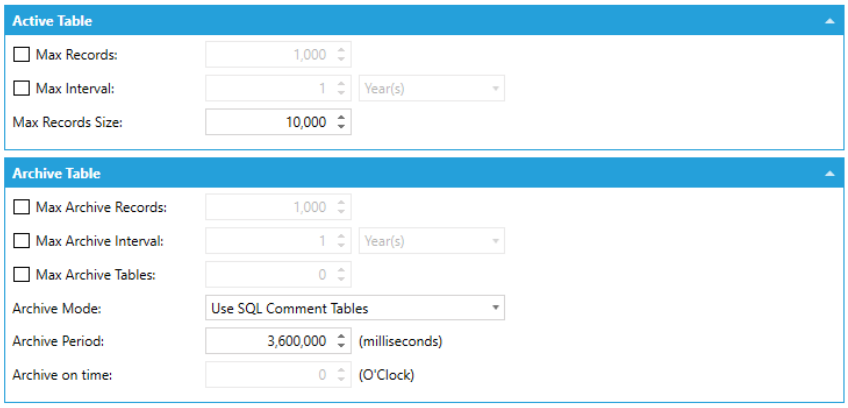


Figure 10.9 - AlarmWorX64 Logger Table Management

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 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:

- **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.

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.

10.4.7. 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.

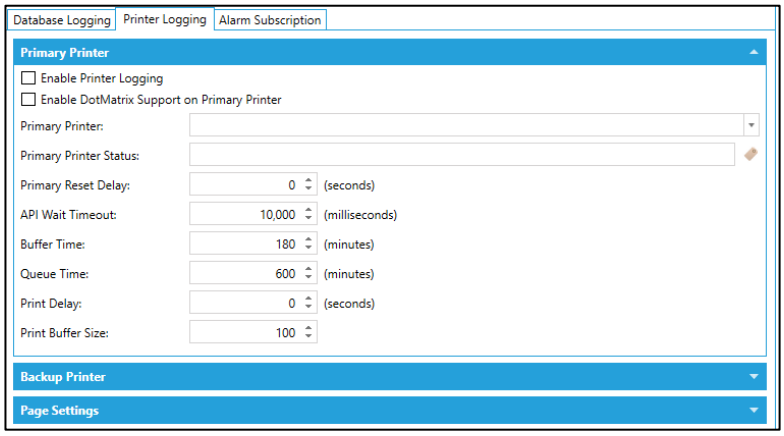


Figure 10.10 - AlarmWorX64 Logger: Printing Logger Events

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.

On the **Page Settings** tab, you can specify a header for the top and/or footer for the bottom of each printed page. You can also 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.

10.5. 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 **Add Column** option. This opens the Column Definition configuration as shown in Figure 10.11.

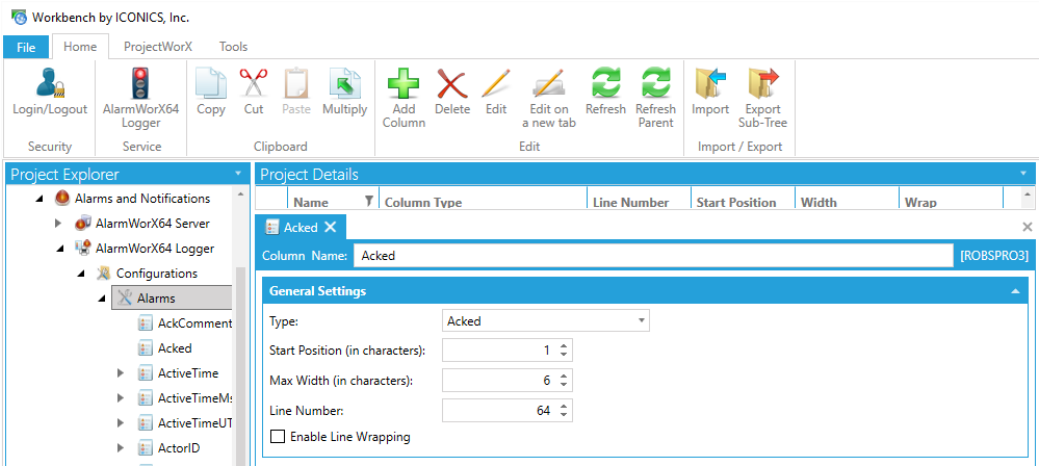


Figure 10.11 - Column Definition

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.

10.6. 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. 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 that node.

You can add a node, by right-clicking on **Nodes** and then selecting **Add Node** from the dropdown menu, as shown in Figure 10.12.

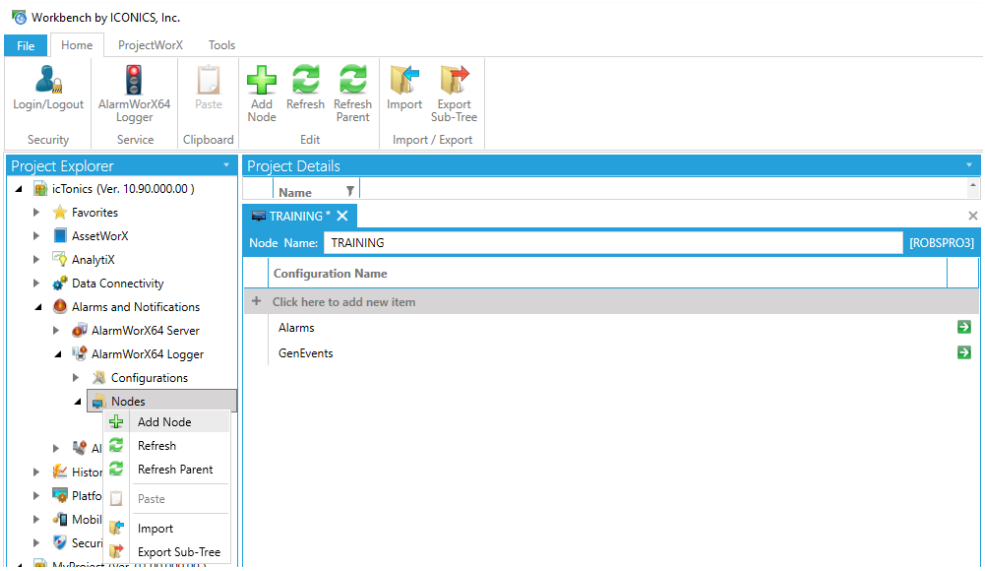


Figure 10.12 - Node Configuration

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), then click on **Apply**.

To associate a configuration with the node, right-click the node and select thwe gray bar labeled [+ **Click here to add a new item**]. 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 "**(local)**" or "**localhost**" to run one or more alarm logger configurations on the local GENESIS64 node.

Lab: Alarm Logger L10

10.1. Lab Overview

Logging generated alarms can help pinpoint problem areas in production operation. Therefore, icTonics has decided to log the alarms for future evaluation and analysis. Because the AlarmWorX64 Server can track both alarms and events, logging operator events such as set points and alarm acknowledgement is also required.

Estimated time to complete this lab: **20 minutes**

10.1.1 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.

10.1.2 Support Files

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

- icTonicsAlarms.gdfx

10.2. Alarm Logger Configuration

In order to log the alarms and/or events generated by an alarm server or event server, you will need to subscribe to an OPC Alarms or Events Server and configure the attributes that need to be saved in the logging database.

10.3. Alarm Database Configuration

Open the **Workbench** if you closed it from the previous lab and expand **Alarms and Notifications**. Select **AlarmWorX64 Logger** from the provider list. Ensure that the ribbon is visible.

1. Expand the **Configurations** tree. You should see two tree configurations (nodes). This is because, when we initially created the database, we left the "Include Sample Data" checkbox checked.

2. Notice that there is already an **Alarms** configuration that is subscribed to the local alarm server "**@ICONICS.AlarmSvr.1.Server**" under the **Alarm Subscription** tab.

3. Select the **Alarms** configuration and rename it **icTonicsAlarms**. Click **Apply**.

4. Select the **Storage** tab and enable the **Use a custom database for logging** checkbox.

5. Create the **Connection String** by clicking the blue **(Configure connection)** text.

6. Since we are using a local SQL Server, the **Select the SQL Server** field should contain **(local)\SQLEXPRESS** (this syntax is not case sensitive).
   Note: Depending on how your machine is configured, the (local) string may need to be replaced with either a period "." or the computer name (i.e.: .\SQLEXPRESS or *hostname*\SQLEXPRESS).

7. From the **Database:** drop-down list, select the **icToncisAlarmData** database you created in Lab 1. You can test the connection if you'd like. Click **OK**.

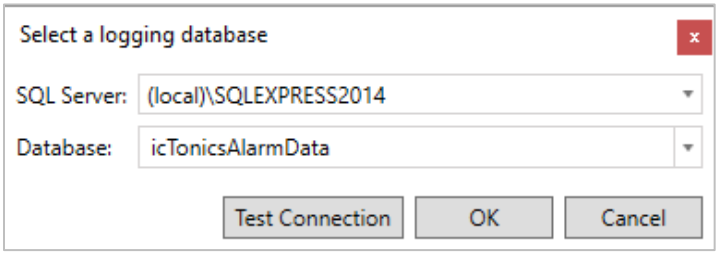


Figure L10.1 – SQL Connection Settings

8. In the **Archiving and Redundancy** tab, expand the **Archiving** section and set **Archive Mode** to **Archive Daily**.

9. Expand the **Archiving and Redundancy** section and put a check-mark in the **Max Archive Interval** checkbox and change the interval to **5 Day(s)**.

Note: This configuration will allow us to keep only one day's records into one active logging table, but archive five days of alarm records into one archive table.

10. Click on **Apply** to save your changes.

10.4. Alarm Node Configuration

1. Expand **Nodes**. Depending on how your computer is set up, you may see your own computer name in the list instead of localhost.
2. Right-click on localhost (or computer name) and select "**Edit on New Tab**".
3. By either selecting from the drop down list or entering the text manually, change the localhost so that it reflects your **Computer Name**. Click on **Apply** to save your changes and **Close** the tab or click on the **Apply & Close** button.

   Note: From a Windows command prompt, you can type Hostname <enter> to determine your Computer Name.
4. Now **Start** the **AlarmWorX64 Logger** service by clicking on the traffic light in the **Service** section of the **Home** ribbon. Wait for the light to turn green.

10.5. Viewing Historical Alarms

Once you have logged your alarms into a database, you can view these logged records. Of course, you can view them through a database tool, or you can use the alarm viewer control. We will be using the AlarmWorX64 Viewer control in a GraphWorX64 display.

10.6. AlarmWorX64 Viewer Control

1. Open the **icTonicsAlarms.gdfx** display that you created in the previous lab.
2. From within the GraphWorX64 Configuration mode, double-click on the **Alarm Viewer control** to open the **Configure your Control** dialog box.

   Note: As we proceed, you may notice the changes you make during Configuration are immediately reflected in the control.
3. Select **Tab** in the navigation tree on the left. In the **Title** section on the right, change the **Text:** field to **Real Time Alarms**.

Note: Although this is not required, you may want to rename the Tab names in the tree also. Tab names cannot contain spaces.

4. Right-click on **AlarmWorX64 Viewer** in the navigation tree and select **Add → Tab**.

5. Select the new tab and change the **Title →Text:** field 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 similar to the one that you set up in the previous lab.

7. In the **Create/Remove Subscriptions** section, click the green plus button to create a new subscription.

8. Select **Historical Subscription**. In the top text box, replace the **New Subscription** text with **icTonicsHistoricalAlarms**.

9. Notice that the Local Alarm Server is already in the **Add/Remove Event Points** section.

10. Click on **Local Alarm Server** to select it. The detailed information of this Event point will be displayed in the Point Tag field.

11. Click on the **ellipsis** [...] button to the right of **Point Tag** to open the **Data browser**.

12. Select the **Data Points** tab and expand **My Computer → Alarms and Notifications → Alarm Logger → Alarms → @ICONICS.AlarmSvr.1.Server**.

13. Click on "**OK**" to exit the Data Browser and add the Event Point(s).

14. In the **Edit Point Properties** section, change the **Point Name** to **awx:Alarms/ @ICONICS.AlarmSvr.1.Server**. Note that this also changes the name under the Add/Remove Event Points Section.

15. Within the **Add/Remove Event Points** section, click on the **Clock** button on the left to open the **Edit Time Range** dialog box. Select a **Historical Data** range of **Today** from the drop-down list. This will limit the alarms to show only today's records.

16. Your Source tab should look similar to the following figure when you are done.

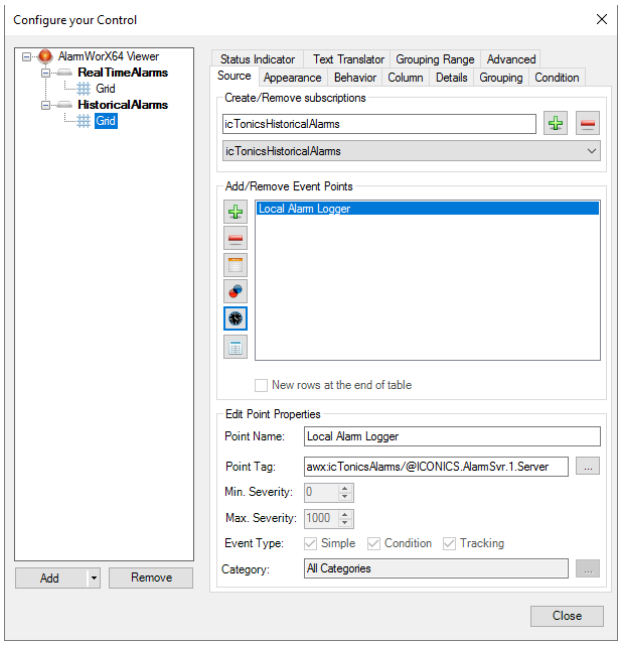


Figure L10.2 - Historical Alarm Sources

17. Select the **Column** tab and uncheck the **Visible** option for the **Priority** and **Quality** columns clicking the green checkmark **Apply** ✔ (button) to save your changes.

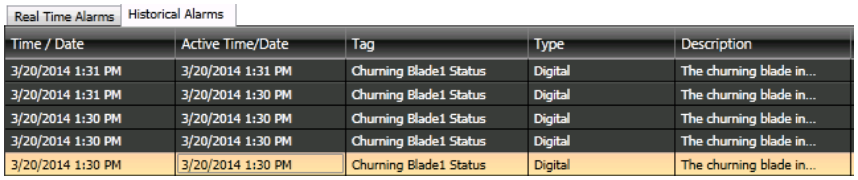
18. Click **Close** to exit the configuration dialog box and **Save** your display.

19. Go into **Runtime**.

20. You may notice a message like **Operation completed with errors.** This is OK. There may not have been any alarms triggered since we started logging alarms and there is no data to report. Hover over the icons in the lower right section of the viewer. Make a selection as you see fit.

21. From within **Workbench**, confirm the services are started by verifying the **Traffic Light** is green for the AlarmWorX64 Server and the AlarmWorX64 Logger.

22. Trigger a new alarm by changing a tag value (e.g. turn the churning blade on and off) in the **icTonicsTankMonitor** display. You should see alarms in the Historical Alarms tab of the viewer, like the example in the figure below.



Time / Date	Active Time/Date	Tag	Type	Description
3/20/2014 1:31 PM	3/20/2014 1:31 PM	Churning Blade1 Status	Digital	The churning blade in...
3/20/2014 1:31 PM	3/20/2014 1:30 PM	Churning Blade1 Status	Digital	The churning blade in...
3/20/2014 1:30 PM	3/20/2014 1:30 PM	Churning Blade1 Status	Digital	The churning blade in...
3/20/2014 1:30 PM	3/20/2014 1:30 PM	Churning Blade1 Status	Digital	The churning blade in...
3/20/2014 1:30 PM	3/20/2014 1:30 PM	Churning Blade1 Status	Digital	The churning blade in...

Figure L10.3 - Historical Data

This concludes this lab.

Hyper Historian Express 11

Hyper Historian Express is a server application that is installed as part of the GENESIS64 suite. It handles data logging and data retrieval replacing TrendWorX64 Logger in version 10.90 and above. The Hyper Historian uses advanced data integration, providing unsurpassed connectivity to any device via OPC-UA, OPC-DA, OPC XML, SNMP, BACnet, or database values, as shown in Figure 11.1.

Hyper Historian sends that data to a historian database. Its basic functions are to collect and log data for analysis and to provide optional data history redundancy. Users can easily collect information from multiple plants, facilities, or throughout the enterprise. Data sources such as PLCs, I/O devices, HMI applications, IIoT devices, and network devices can be collected and stored for charting, reporting, and analysis.

The Hyper Historian HDA server will connect to a Hyper Historian database configuration and serve the data contained in that database to OPC HDA clients such as the TrendWorX64 Viewer. You can deploy Hyper Historian Express using the same computer on which the database resides or connect to a database on a different computer on the network. You can configure Hyper Historian Express using configuration forms in the Workbench under the Historical Data → Hyper Historian provider.

11.1. Objectives

After completing this chapter, you should be familiar with:

- Hyper Historian Express and how to launch it
- Database configurations
- Database Loggers and Groups
- Tags that collect historical data

11.2. Architecture

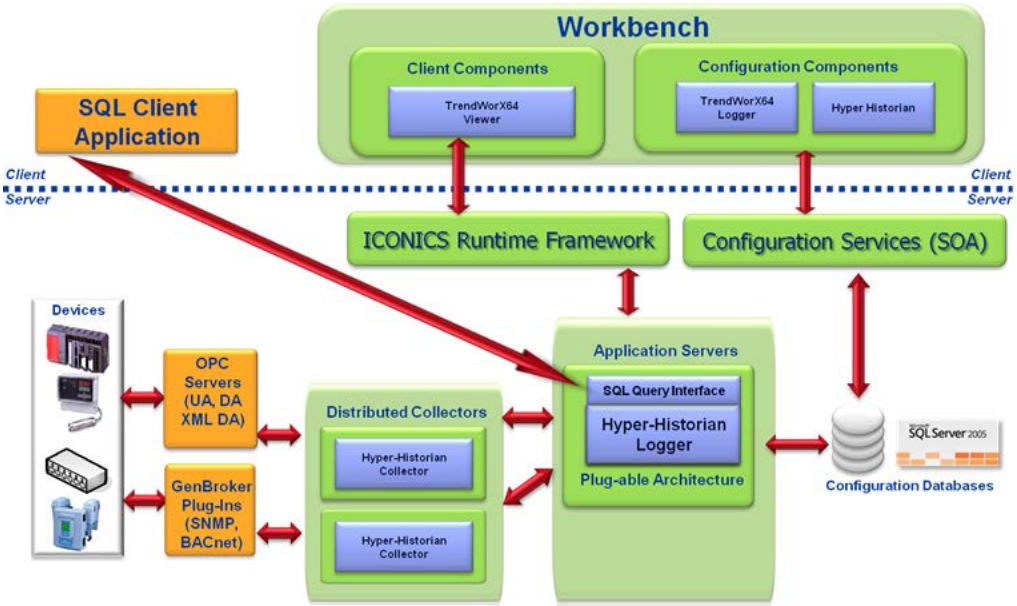


Figure 11.1 - Hyper Historian Express Overview

11.3. Hyper Historian Express in the Workbench

The Workbench houses the Hyper Historian Configurator, which features a thin client, optimistic concurrent design, acting as the central configuration environment and operator interface. The Workbench's advanced configuration console performs complete service management and has integrated layout, project management, and remote pack-and-go deployment capabilities.

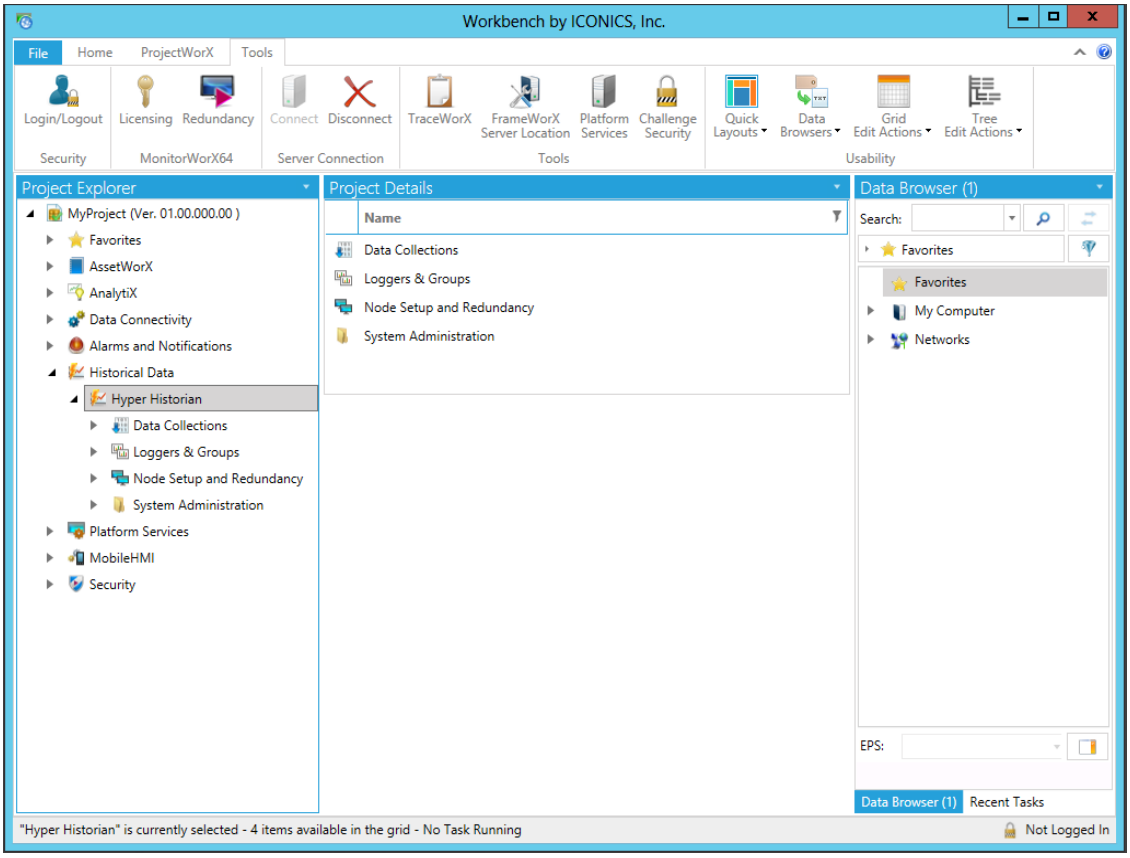


Figure 111.2 - Hyper Historian Express in the Workbench

11.4. Logger Configurations

The Hyper Historian uses Loggers for grouping tags that collect similar types of data. It allows you to create multiple **Loggers** based on the data you are collecting. Once created, these loggers can be organized into **Groups** (Logging Groups). In addition, multiple **Data Collections** can be used for collecting values from multiple tags and locations.

For each Logger that is created, the Hyper Historian creates a series of binary files to hold the Plant Historical data. These files can be archived on the basis of configuration options that you select. Let's take a look at what types of data we are talking about.

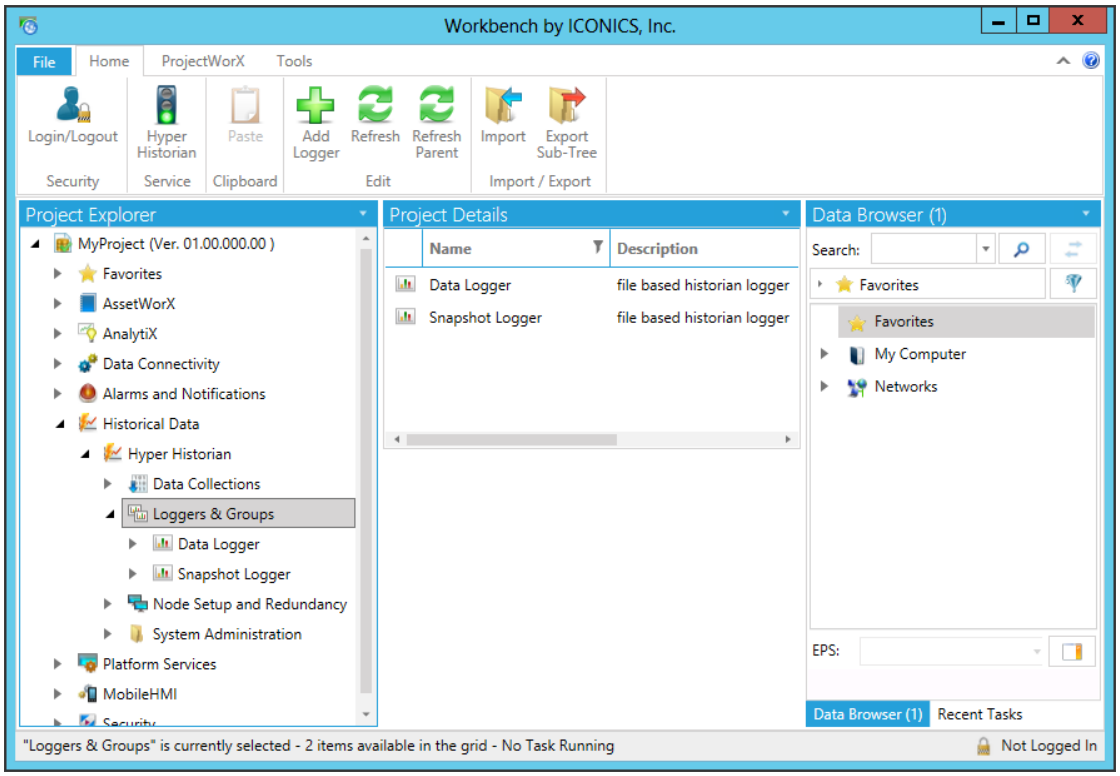


Figure 111.3 – Hyper Historian Configuration

11.4.1. Two Types of Loggers

In Hyper Historian Express, there are **two types of** L**ogger configurations** that govern how data gets logged to the Logger. They are, ***Data Logger*** configurations and ***Snapshot Logger*** configurations. You can create any number of each type of Logger. Data stored with each type of Logger is available for replay and reporting for a timeframe of your choosing. Stored data is automatically archived at a point predetermined by you, and can be easily restored from archives.

A **Data Logger**, as illustrated in Figure 111., collects plant data and saves it to a hard disk; this is the type of Logger that is used most often. This type of logging can be used to collect:

- Plant data for use in historic replay with TrendWorX64
- Plant data you need to collect for reports created using BizViz products
- Plant data you need to archive for regulatory compliance
- Plant data you want to save for future reference and analysis

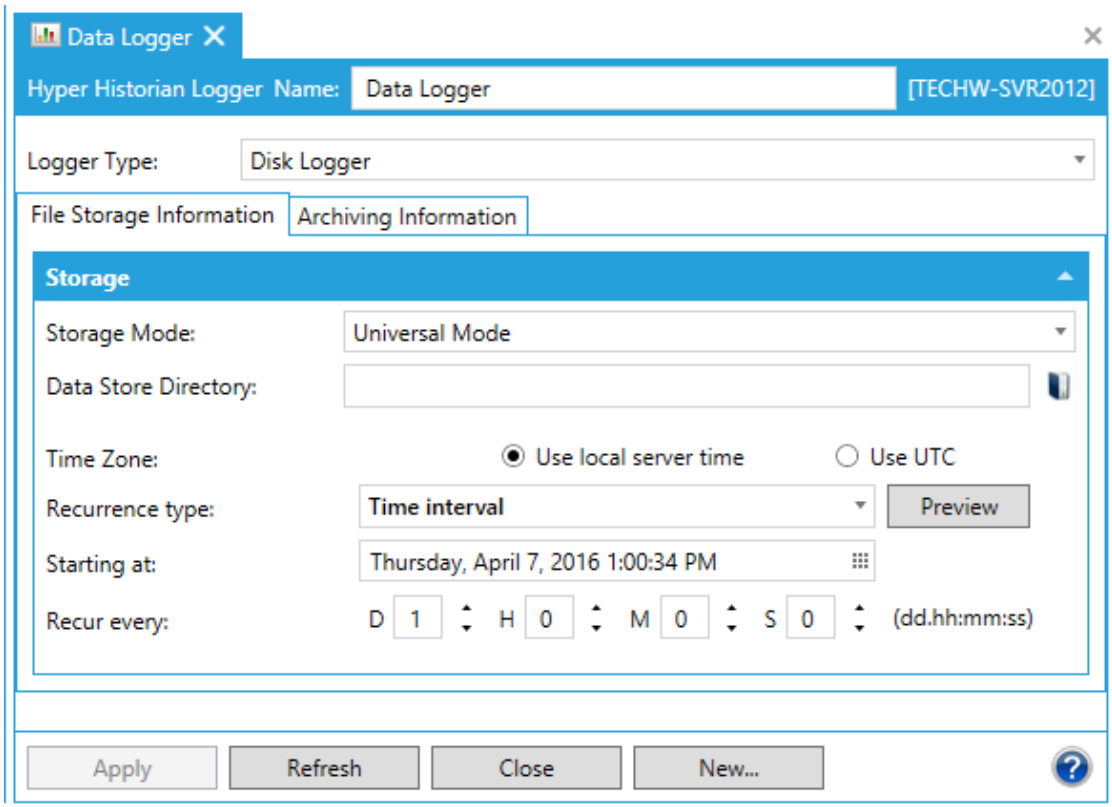


Figure 11.4 - Disk Logger Configuration

11.4.2. File Storage Information

It is within the **File Storage Information** section that you control the length of time (amount of plant data) should be logged to each file and the folder where these files are stored. The storage folder must be local but could be a mapped drive. In the **Data Store Directory** field you specify the path where data files will be stored. You can click the button to navigate to the location.

Do not enter a file name because files are named automatically using the following format:

loggerConfigName_startDate_startTime_endDate_endTime

NOTE: Dates are in *yyyymmdd* (year, month, day) format, and times are in *hhmmss* (hour, minute, second) format.

Select the conditions for closing the data file based on a recurring **Time Interval**, Days, Hour, Mins, Secs, or **Specific Dates and Times**, recurring every X year(s), month(s), week(s), day(s), hour(s), minute(s), or second(s). As soon as the configured condition is

reached, the current file is closed but remains online for data retrieval and a new file is created.

11.4.3. Logging Groups

After you create a Disk Logger configuration, you will look at how the Logger's tags are organized into **Groups** (Logging Groups). To set up a **Logging Group** you must:

- Create and configure **Logging Groups** that identify the server where the Logger resides, and the Logger configurations to be used for logging data.
- Create **Data Collections** to be used for collecting values from tags.
- And create the **Hyper Historian tags** for each Data Collection. Each Hyper Historian tag identifies the OPC tag whose values are to be collected, along with an assortment of configuration information for collecting the tag's values.

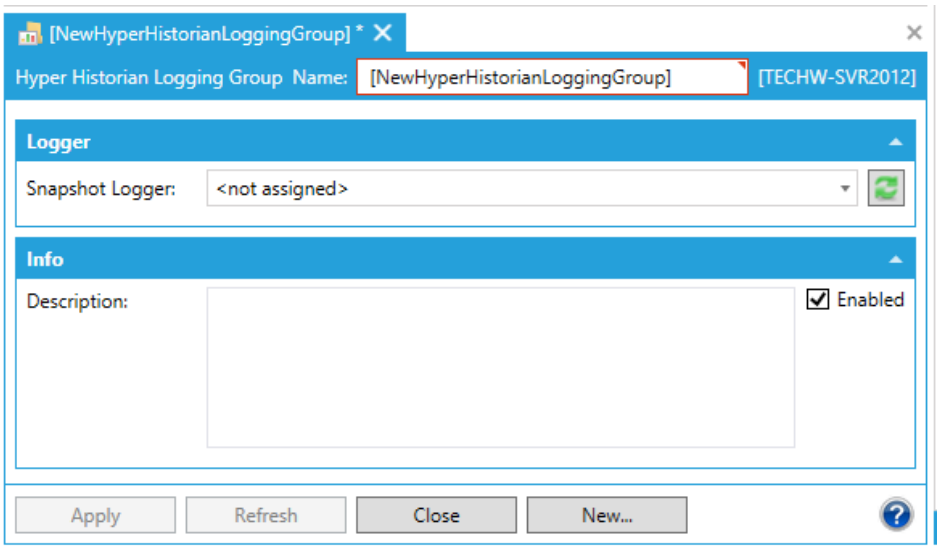


Figure 111.5 – Hyper Historian Logging Group

11.5. Collector Groups in a Logging Group

Once you've defined a Logging Group, you can add a **Collector Group**. A Collector Group defines the collection rate and period of calculation for the OPC tags you will be attaching to it. If you want to use more than one Data Collection Rate or Calculation Period, simply create an additional Collector Group, and then add the tags to be collected at that rate. A logging group can have any number of Collector Groups. Once you have created the Collector Group, then you can specify the tags to collect samples for the Collector Group.

11.5.1. Create a Collector Group

To create a Collector Group, right-click on the Logging Group in the navigation true in the Project Explorer, then select **Add Collector Group**. The Hyper Historian **Collector Group** configuration appears as show in **Error! Reference source not found.**11.6.

Enter a name in the **Hyper Historian Collector Group Name** box for the Collector Group. In the **Info** section at the bottom of the form, specify what you are using the collector group for. You should make the description clear enough so that anyone who maintains the group understands its purpose.

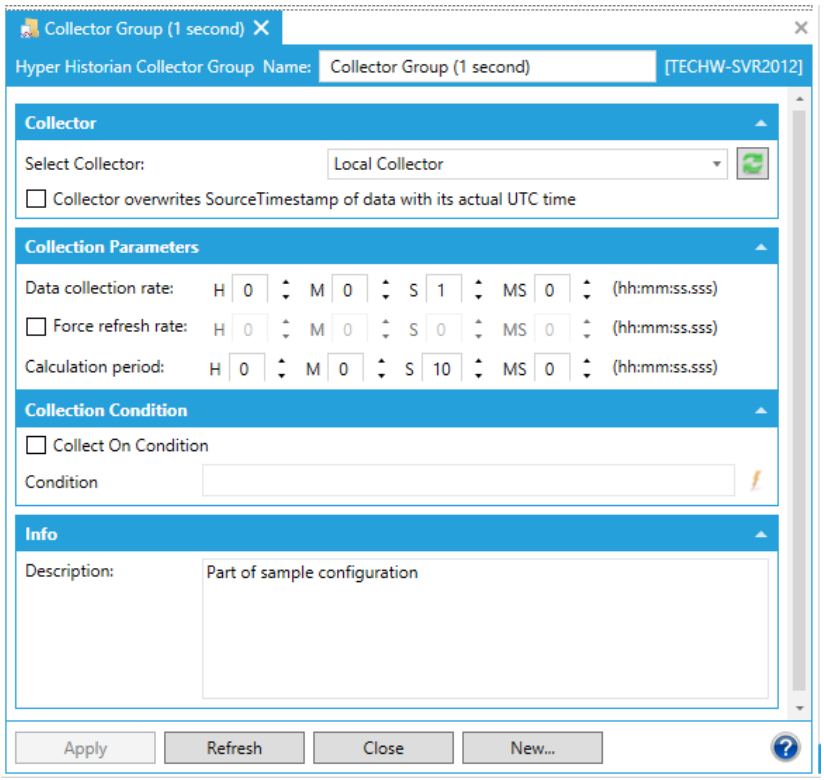


Figure 11.6 - Hyper Historian Collector Group

11.6. Hyper Historian Tags in a Collector Group

Each Hyper Historian tag collects data from an OPC tag or an expression that calculates a value from multiple OPC tags. As you work with Hyper Historian tags, keep in mind that it is the ***Collector Group*** that determines:

- The Data Collection Rate at which values are collected for the tags, and

- The Calculation Period, at which calculations are performed on the collected samples, and then forwarded to the Logger

11.6.1. Adding a Hyper Historian Tag

To add a tag, right-click within the Data Collections node or on a folder within the node, then select **Add Tag** to proceed. The Hyper Historian tag configurator appears as shown in **Error! Reference source not found.**7.

NOTE: You can create a folder, then right-click the new folder and select **Add Tag**.

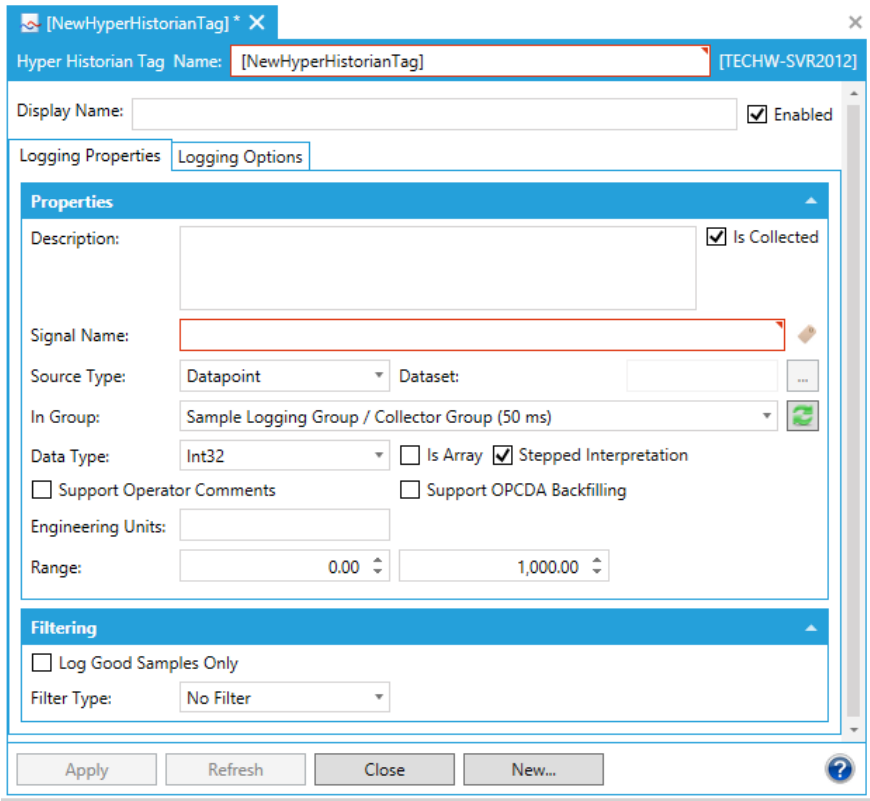


Figure 11.7 - Hyper Historian Tag Configuration

The Hyper Historian Tag properties has two tabs. In this chapter, we will be looking at the **Logging Properties** and **Logging Options** tabs.

Start by entering a **Hyper Historian Tag Name**. Make sure the name is descriptive and unique but accurate, so that others can recognize it. You may want to use the name of the OPC tag, itself. If you are configuring more than one Hyper Historian tag for an OPC tag, make sure you use a distinct name of each tag; for example, add “Minimum” to the name. Each tag name must be unique for the Collector Group. You can provide a separate name for the tag in the **Display Name** field – this name will appear in the navigation tree.

11.6.2. Logging Properties Tab

1. The **Description** field can contain a detailed description of several lines for the Hyper Historian tag. Use the description to provide a complete definition of the tag.

2. The **Is Collected** checkbox is selected by default. This means that the Hyper Historian tag's data is collected by the collector from a signal. However, if the tag's data is not collected using a signal, but is collected and written directly to the logger using some other means, such as a TrendWorX64 Viewer or a .CSV file via MergeWorX, de-select the Is Collected checkbox. As a result, the tag's Signal Name field becomes disabled, as does the Filtering field and (on the Tag Logging Options tab, described later in these steps) the Log to Disk options.

3. For the **Signal Name** field, select the signal (that is, the tag or point) to be collected by clicking the Tag button and browsing for the OPC tag whose data is to be collected. If you leave this field empty, an exclamation point icon will remind you that you have to enter one. (Note that this field is not used if the **Is Collected** checkbox is empty.)

4. Select a **Source Type** for the data signal associated with the tag using the provided pulldown menu. The signal can come from a **Datapoint** [e.g. from GenClient (recommended for processing OPC DA and UDM data) or a FrameWorX Client (recommended for processing OPC UA and GridWorX data)] or a **Dataset** (recommended for processing BACNet data) or **Dataset Generic**.

5. If you select ***Dataset***, leave the **Dataset** field blank if the data set contains only one value or a default set of column names, or enter the user-defined name that appears at the end of each column name in the dataset (for example, if your data set includes columns named *Value_test* and *StatusCode_test*, enter "test" in the **Dataset** field). You can click on the ellipsis [...] button to open the **Dataset Configurator** or **Dataset Generic Configurator**.

6. For the **In Group** field, select the group in which the tag should appear within the configuration database. Click on the refresh button in order to load the current list of groups.

7. Select the **Data Type** from the Data Type drop-down list. Options include: Native, Boolean, Int8, UInt8, Int16, UInt16, Int32, UInt32, Int64, UInt64, Float32, Float64, String, Duration and DateTime.

8. The **Is Array** checkbox allows you to set if the data type is an array.

9. Put a check mark next to **Stepped Interpretation** to enable this feature if it is desired. Stepped Interpretation is used primarily during playback, for interpreting the tag's data values. This feature is applied only if you request a value for a certain point of time (in the Log to Disk option on the Tag Logging Options tab). The returned value will be the nearest older value available. In general, if the data comes from a discrete signal such as an on/off switch (or a Boolean data type), put a check mark in the Stepped Interpretation check box; but if data comes from a sine wave or analog signal, leave the box empty.

10. You can optionally click a check mark in the **Support Operator Comments** check box to allow operators to comment on the signal in displays. Later, as the tag's data values play or replay in the Trend Viewer, you may want to be able annotate the tag's data values as you see them trending before you. For example, maybe you want to note an operator action that altered the data slightly, or highlight a piece of data for analysis later. Putting a check mark in the Support Operator Annotations check box lets you do that. Such annotations are saved with the tag's data in its logger file, which makes the annotations available in a replay, accessible to the Hyper Historian SQL Server Interface, and archived with the tag's data.

11. Enter the **Engineering Units**.

12. Enter the **Ranges** for the High Limit and Low Limits in the provided text entry fields. The High and Low Limits are used by the filter algorithms (deadband filter and swinging door filter).

13. Optionally, you can select the type of **Filtering** to be used and enter the filter setting via the **Filter Type** pulldown menu. By default, No Filter is used.

14. If you are using ***Deadband*** filtering, set the threshold as either an **Absolute** or **Percent** and enter the value in the text entry field.

15. If you are using ***Swinging Door*** filtering, set the threshold as either an **Absolute** or **Percent** and enter the value in the text entry field. Then, set both the **Maximum Period** and **Minimum Period** in the nearby text entry fields (in milliseconds).

Note: Filtering options are not available if the Is Collected check box is empty. A tag's Filtering option is applied to data in the collector. If the tag's data is written directly to the logger, no filtering can be applied because the data never passes through the collector.

16. Click the **Apply** button to enforce your selections for the Tag Properties tab.

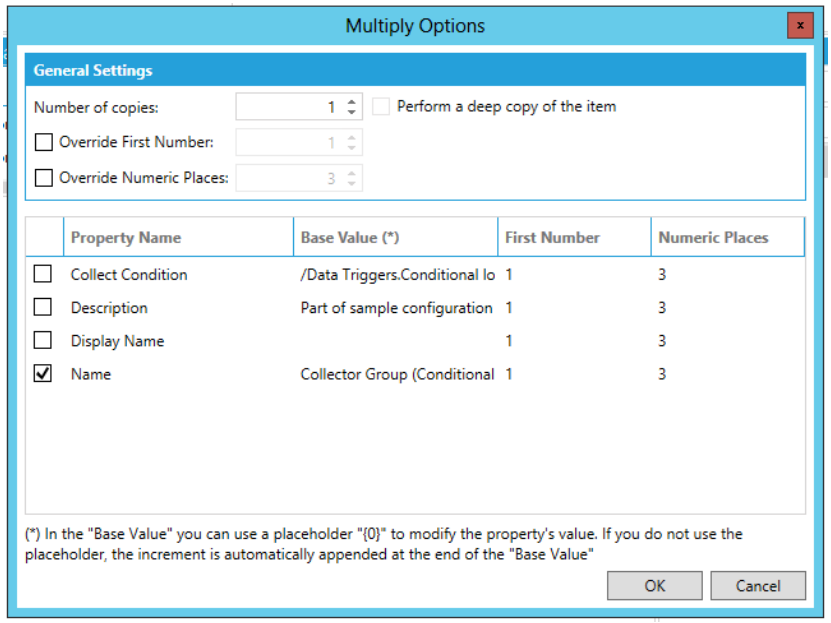
11.6.3. Logging Options Tab

1. Go the **Tag Logging Options tab**. In the **Log to Disk** section, click on the **Aggregate Type** pulldown menu to select from All Samples, Max, Min, Avg, Std. Dev., Running Max, Running Min, Running Avg, Running Avg, Moving Avg, Moving Max, Moving Min, Totalizer or Most Recent On Time.

11.6.4. Making Copies Quickly and Easily

You have the option, at any time, of quickly making any number of copies of a Logging Group, Collector Group, Hyper Historian Tag or Tag Folder, or Aggregate Group. Once copies are made, you can then tailor them to specific uses. This allows you to quickly create items that share characteristics, or to easily propagate characteristics throughout your system.

To make multiple copies of one of the items listed above, navigate to the item in the Project Explorer. Then right-click and select **Multiply**. This opens the Multiply Options dialog box, shown in Figure 11.8.



In the **Number of Items** box, specify the number of copies you want. Put a checkmark in the **Perform a Deep Copy of the Item** checkbox to copy *all* of the values defined for the item. This option is helpful if most of the object's properties are relevant for the copies you are making. If they aren't relevant, leave the checkbox empty.

Each copy's name will contain a number incremented by 1, starting with the value in the **First Number** box. The incremented number will have the number of digits specified in

the **Numeric Places** box. You can override these increments for individual fields (the item's name, its display name, and its description) by leaving a checkmark in the field's **Modify** checkbox and entering new values for the line you want modified from the original.

To prevent a field from being modified, remove the checkmark from the field's **Modify** checkbox. When you are ready, click **OK** to make the copies; the copies are added to the navigation tree, where you can access them and modify them.

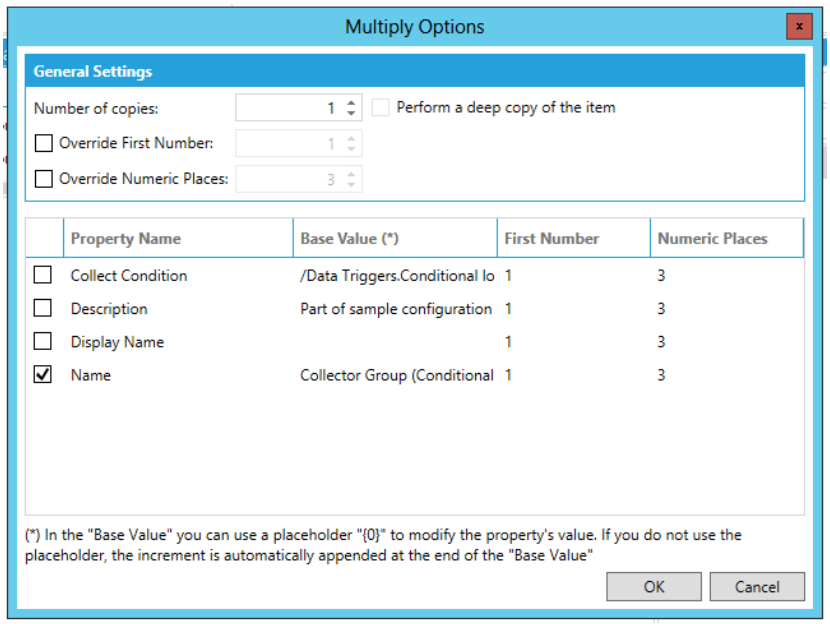


Figure 11.8 – Making Multiple Copies

11.7. Organizing Tags into Folders

If you have several tags that the collector group is collecting data from, you can organize the tags into folders if you like. To create a folder, right-click on **Data Collections**, then select **Add Folder**. Enter a name for the folder and a description of the tags in the folder. Click on **Apply** to save the folder. Then add tags to it by right-clicking on the folder and selecting **Add Tag**.

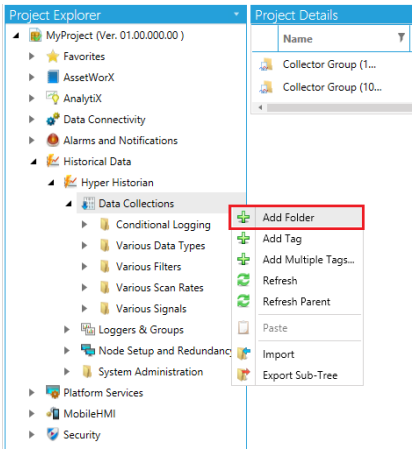


Figure 11.9 - Hyper Historian Folder Tag Folder

Lab: Hyper Historian Express L11

11.1 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. With this procedure, we will be logging to Binary files. Hyper Historian is well suited for applications requiring very fast logging for thousands of tags. Hyper Historian's performance is far superior to SQL Server.

Estimated time to complete this lab: **20 minutes**

11.2 Task Checklist

In this lab, you will complete the following tasks:

- Create a new Hyper Historian configuration database.
- Create a new Data Logger configuration.
- Create a new Logging Group.
- Create a new Collector Group.
- Add Tags to Data Collections.

11.3 Support Files

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

11.4 Configure Hyper Historian Express

The Hyper Historian configuration allows you to set up OPC tag values you want to log. We will create a configuration that specifies how fast to collect data, where to log data, and how often to write data into the binary files.

11.4.1 Configure Hyper Historian Express

1. Open **Workbench** if you have closed it since the previous lab and select **Hyper Historian** in the Project Explorer window under **Historical Data**.

2. Expand **Data Collections** to expose any configured items.

3. Right-click on **Loggers & Groups** to select **Add Logger**, as shown in Figure L11.3.

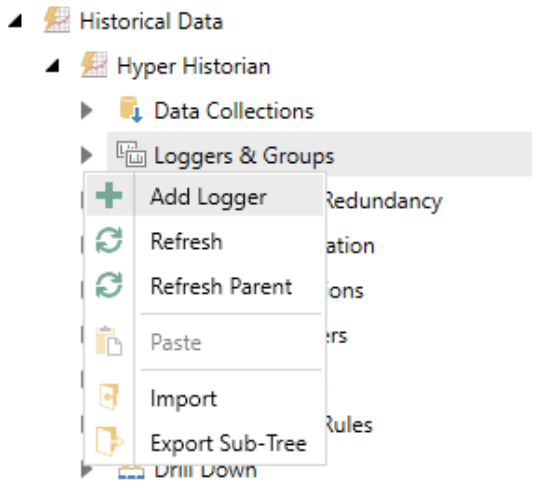


Figure L11.3 – Adding a Logger

4. A new tab will open in the bottom of the Project Details window showing configuration properties for the new **Logger**. Name this Logger **icTonicsLogger**. In the **Logger Type** field, use the pulldown menu to select **Disk Logger**.

5. In the **File Storage Information** tab, set the **Recurrence type** to **Time interval**, and set it to recur every day.

6. In the **Archiving Information** tab, set the condition to **Minimum Time Extent** and set the value to 7 days, so that the logged data gets archived once every 7 days.

7. Click **Apply & Close**.

8. Right-click on the newly created Logger to select **Add Logging Group**, as shown in Figure L11.4.

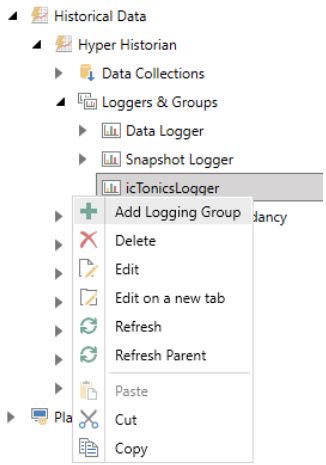


Figure L11.4 – Adding a Logging Group

9. Name the Logging Group **TanksLoggingGroup**. Click **Apply & Close**.

10. Right-click on **TanksLoggingGroup** to select **Add Collector Group**. Name this Collector Group **LevelsCollectorGroup**.

11. In the **Collection Parameters** section of the Collector Group properties, change the **Data collection rate** and **Calculation period** to **5 seconds**. Click **Apply & Close**.

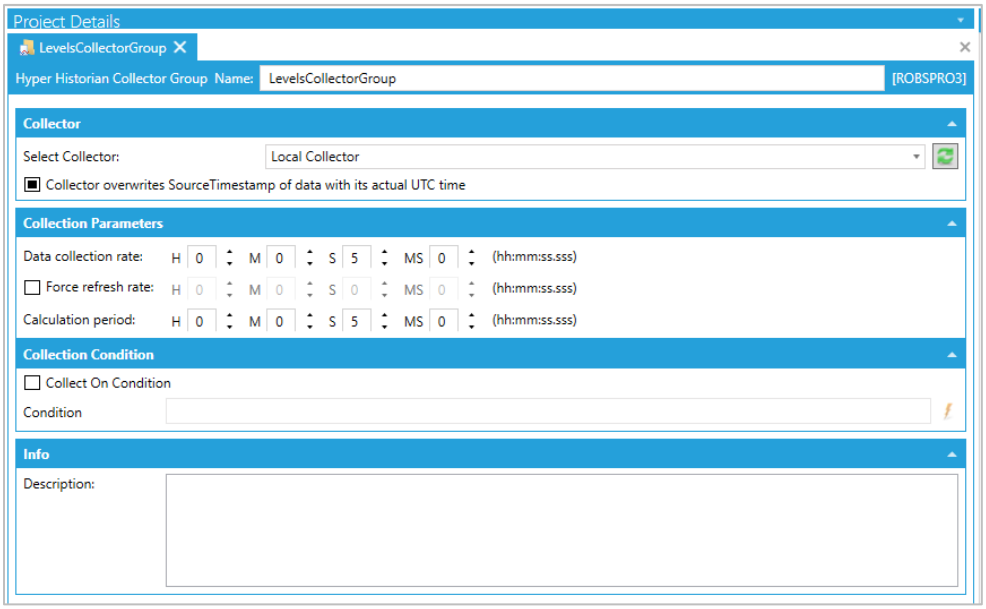


Figure L11.5 – Collection Parameters Section of the Collector Group Properties

12. From the Project Explorer, Right-click on the **Data Collections** node to select **Add Folder**. A new tab should open called [NewHyperHistorianFolder]. Rename this Data Collections folder **TankLevels**. Click **Apply & Close**.

11.4.2 Adding a Tag

1. In the Project Explorer, right-click the **TankLevels** node and select **Add Tag**. A new tab will open with tag properties.

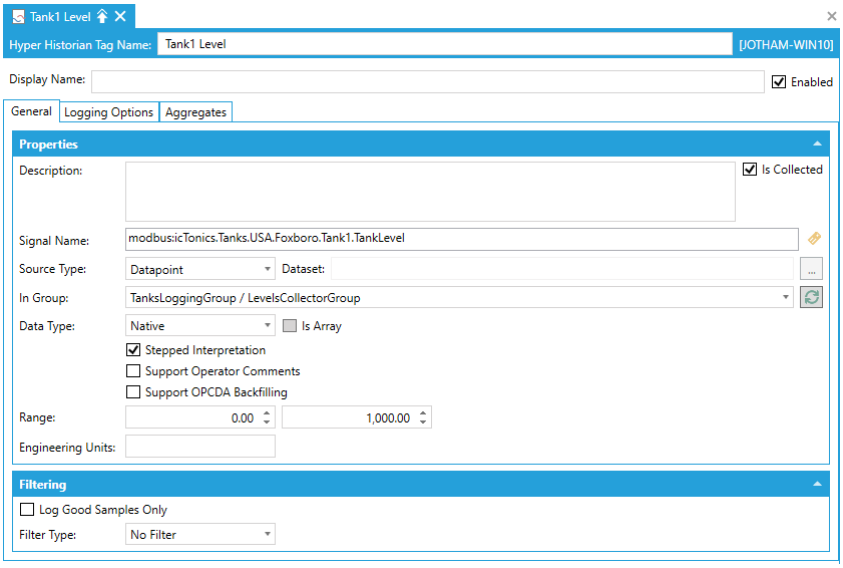


Figure L11.6 – Adding a Historian Tag

2. Name this tag **Tank1 Level**. Click on the Tag icon next to the **Signal Name** field to open the **Data Browser**.

3. In the Data Browser, we will look for the following tag:

modbus:icTonics.Tanks.USA.Foxboro.Tank1.TankLevel

4. Select the **In Group** dropdown menu and select: **TanksLoggingGroup/LevelsCollectorGroup.**

5. We can leave everything else as the default for the tag configuration. Click on **Apply & Close** to save the changes. Your finished tag configuration should look similar to Figure L11.6.

6. Right-click on **TankLevels** again and select **Add Tag**. A new tab should open with Hyper Historian Tag configuration properties.

7. Name this tag **Tank1 LevelSetpoint**. Click on the Tag icon next to the **Signal Name** field to bring up the **Data Browser**.

8. In the Data Browser, we will look for the following tag:

modbus:icTonics.Tanks.USA.Foxboro.Tank1.LevelSetPoint

9. Select it and then click on **OK** to go back to the **Tag Configuration** screen.

10. We can leave everything else as the default for the tag configuration. Click on **Apply & Close** to save the changes. Your finished tag configuration should look similar to the figure below.

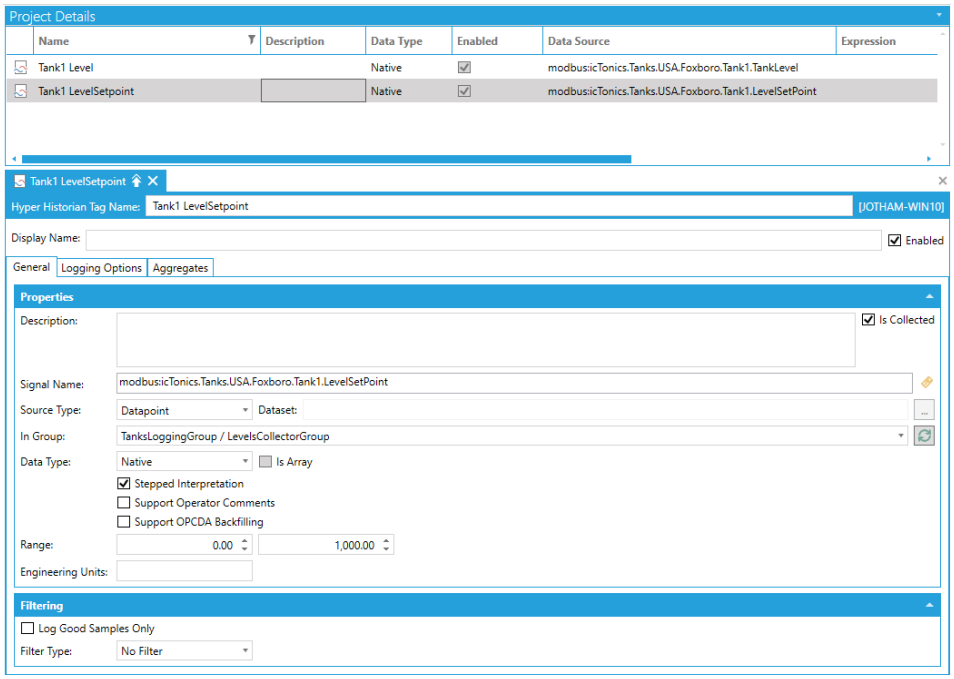


Figure L11.6 – Tank1 Level Setpoint Tag Configuration

11. In the **Home** ribbon, if the Hyper Historian Service traffic light is green, click on the traffic light icon to **Stop** the service. The traffic light should turn **red**.

12. Click on traffic light in the ribbon again and **Start** the service. The light should turn **green**. While it is not necessary to Stop/Start the service for your changes to take effect we want to become familiar with the traffic light.

11.5 Tag Multiply

Now we will add similar logging tags for **Tank2** and **Tank3**. For this, we can use the **Multiply** function, which makes it easier to create similar alarms.

1. Right-click on the **Tank1 Level** data collector tag and select **Multiply** to launch the **Multiply Options** dialog box. Alternatively you could have selected **Tank1 Level** and clicked on the **Multiply** icon in the **Clipboard** section of the ribbon.

2. Let's use the **placeholder {0}** to auto-generate the **Name** and **Signal Name** field's proper 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.

3. Set the **Multiply Options** as follows:

Modify Check Box	Field Name	Value
	Number of copies	2
Yes	First Number:	2
Yes	Numeric Places	1
Yes	Data Source	modbus:icTonics.Tanks.USA.Foxboro.Tank1.Tank {0}Level
Yes	Name	Tank{0} Level

4. When your configuration resembles the following figure, select **OK**.

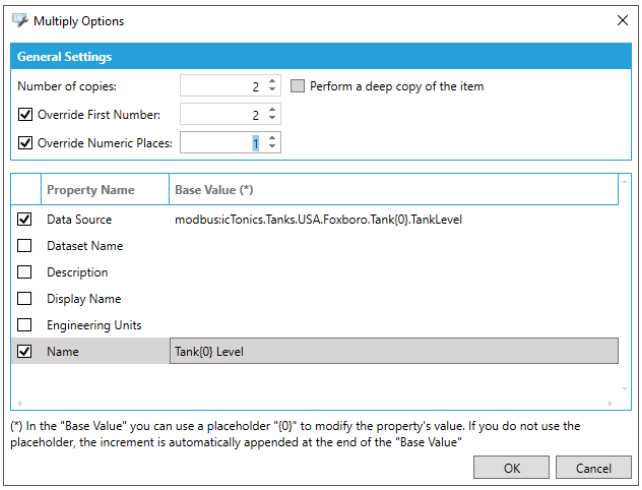


Figure L11.7 – Multiply Options Dialog Box

This step is very important as it will affect your data logging 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:

modbus:icTonics.Tanks.USA.Foxboro.Tank1.TankLevel
modbus:icTonics.Tanks.USA.Foxboro.Tank2.TankLevel
modbus:icTonics.Tanks.USA.Foxboro.Tank3.TankLevel

5. Repeat the steps that you've learned to add the following two tags to the **Tank Levels** logging group.

modbus:icTonics.Tanks.USA.Foxboro.Tank2.LevelSetPoint
modbus:icTonics.Tanks.USA.Foxboro.Tank3.LevelSetPoint

Your finished configuration should look similar to the following figure:

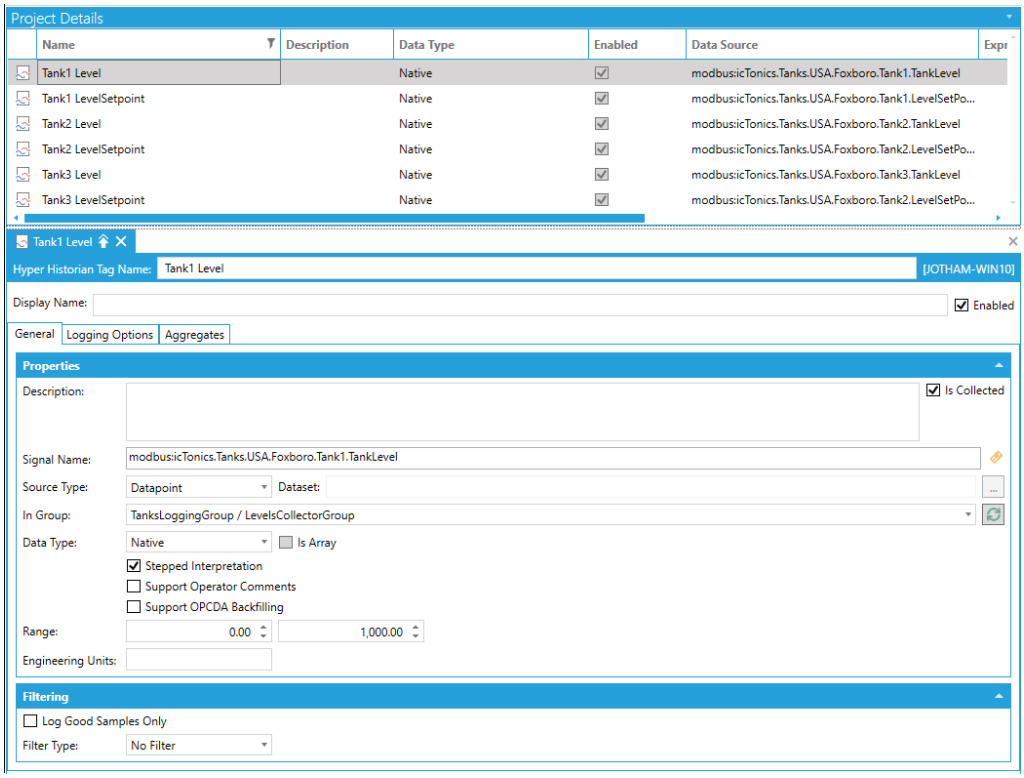


Figure L11.8 – Completed Hyper Historian Configuration

In the next chapter, we will add a TrendWorX64 viewer to a GraphWorX64 display, allowing us to view both real-time and historical data.

This concludes this lab.

TrendWorX64 Viewer Control 12

The ICONICS TrendWorX64 Viewer, shown in Figure 12.1, 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.

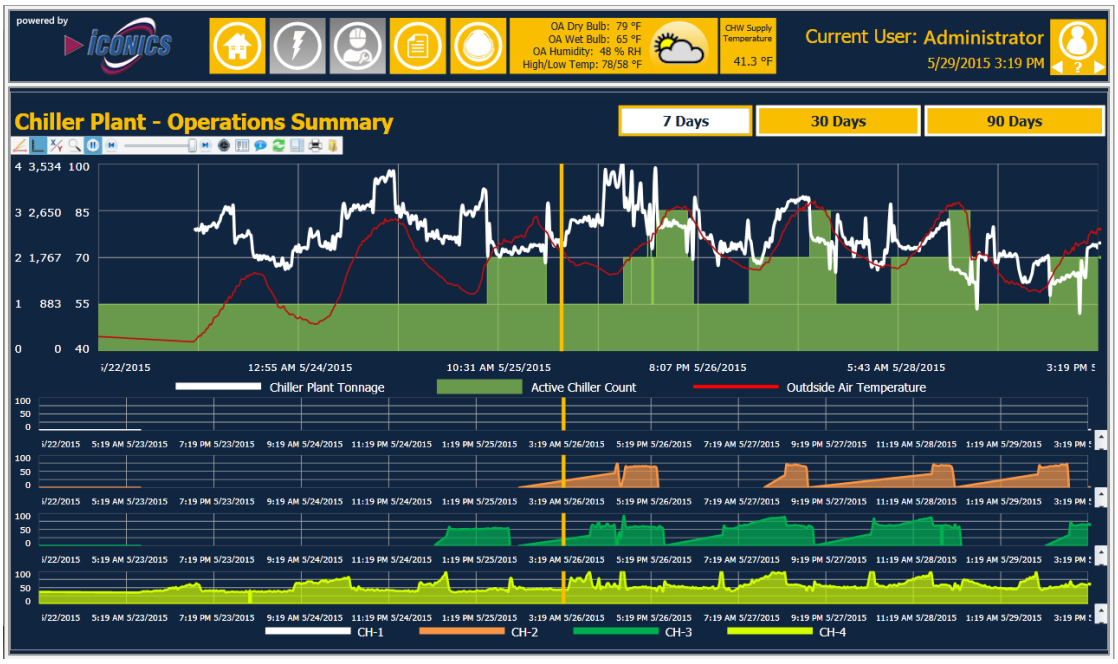


Figure 12.1 - TrendWorX64 Viewer in a Display

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

12.1. Objectives

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

12.2. TrendWorX64 Viewer Modes

The TrendWorX64 Viewer has three different modes: **Configuration** mode, **Runtime** mode, and **Freeze** mode.

Configuration mode is designed 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. The data still collects, but the display stops updating with current data until you exit freeze mode.

12.3. TrendWorX64 Configuration Menu

The TrendWorX64 Configure your Control dialog, shown in Figure 12.3, is the trend configuration interface, which is accessible from 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, double click on the TrendWorX64 Viewer in the Explorer window of GraphWorX64. Alternatively, certain aspects of this menu become available in the Chart and Pens tabs that will appear in the ribbon once a TrendWorX64 Viewer is selected.

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. The Tab has a chart in it, with a Plot, and a Pen. It looks something like the tree in Figure 12.2.

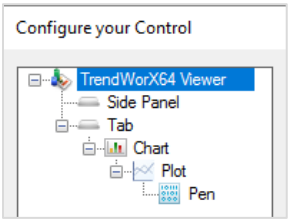


Figure 12.2 - Default Structure of Objects in the TrendWorX64 Viewer

Clicking on TrendWorX64Viewer in the tree offers general options for Appearance such as whether Runtime Toolbar appears, startup file to load (configuration), as well as Title and Subtitle options. Figure 12.3 shows the Appearance tab (with the Advanced tab behind it) for the TrendWorX64Viewer tree item.

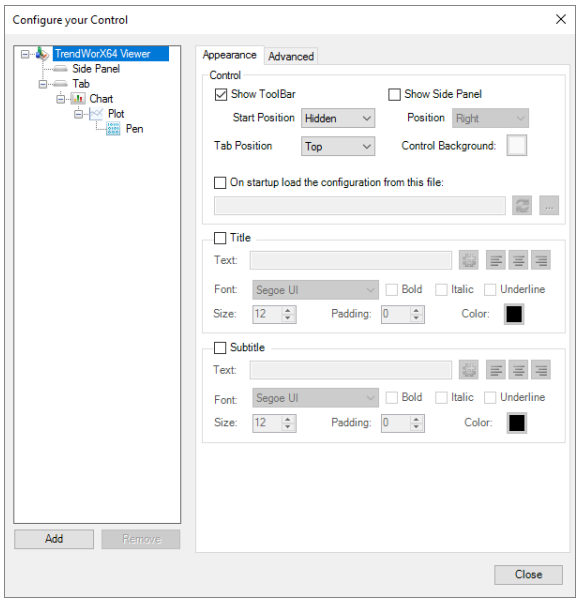


Figure 12.3 - TrendWorX64 Viewer Appearance Properties

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.

12.3.1. Appearance Settings

The **Appearance** tab, shown previously in Figure 12.3, appears by default when you select the viewer object. This tab sets the appearance properties for the TrendWorX64 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, let it float or leave it Hidden (default). The toolbar appears only in Runtime or Freeze mode. If the toolbar is hidden, runtime operators can double-click in the viewer to show the toolbar.

The "Show Side Panel" checkbox and "Position" option dictate 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 the 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.

12.3.2. 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.

12.4. 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 12.4 illustrates 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.

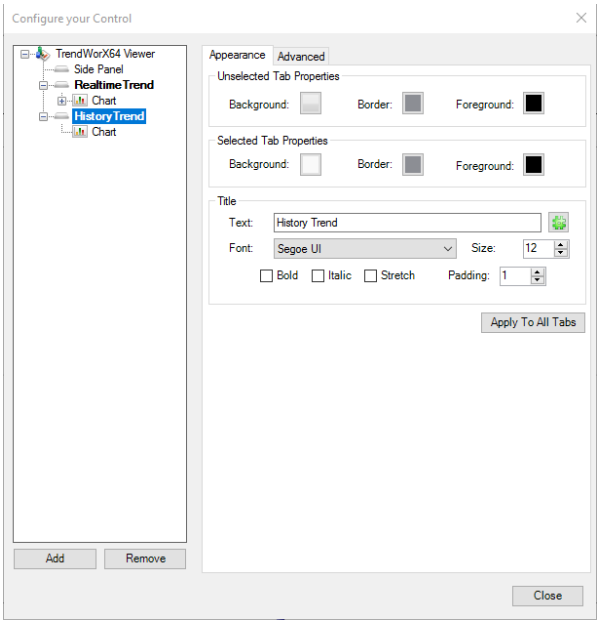


Figure 12.4 - Tab Options

> **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 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.

12.5. 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 12.5. You will also notice in the figure that charts can also be added to the side panel.

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 the options, which are spread over seven tabs: the Chart tab, the Data tab, the X Axis tab, the Y Axis tab, the Summary tab, the Legend tab, and the Advanced tab. We will look at the first six tabs in this course.

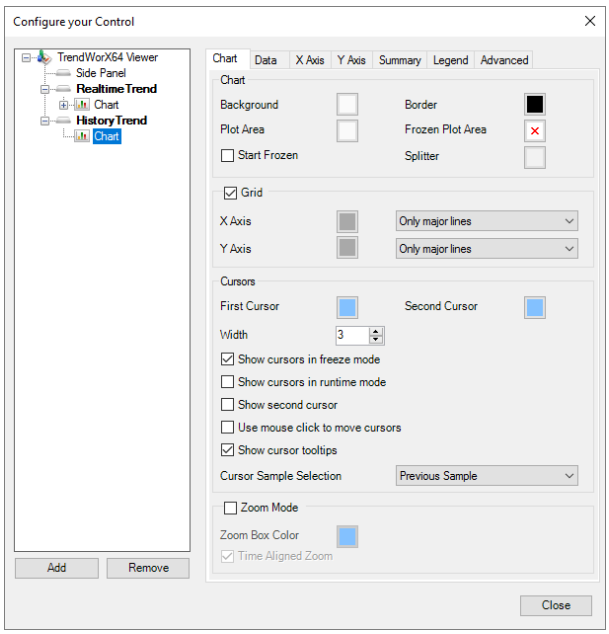


Figure 12.5 - Chart Options

12.5.1. Chart Tab

On the **Chart** tab shown in Figure 12.5, you can set the following:

- **Chart appearance**
- **Grid**
- **Cursor**
- **Zoom Mode**

Additionally, you can specify the Background, Border, Plot Area, Frozen Plot Area, and Splitter color. You can set the chart to Start Frozen and set Major and Minor Ticks for the Grid. You can also enable 1 or 2 cursor lines or have no cursor if you wish while in Freeze mode.

Within the Grid section, you can set the following options:

- **Grid** - Set the presence of the grid on the TrendWorX64 Viewer
- **X Axis** - Set the color and level of visualization for the X Axis Grid
- **Y Axis** - Set the color and level of visualization for the Y Axis Grid

Within the Cursor section, you can set the following options:

- **First Cursor** - Set the color of the first color
- **Second Cursor** - Set the color of the second cursor
- **Width** - Set the cursor width

Additionally, you can select to show cursor in freeze mode, show cursor in runtime, show second cursor, use mouse click to move cursors, show cursor tooltips, or select the cursor sample size.

Within the Zoom Mode section, you can set the following options:

- **Zoom Mode** - Specify whether or not charts can be zoomed in or out
- **Zoom Box Color** - Set the fill color of the zoom box
- **Time Aligned Zoom** - If selected, all zoom operations will result in a time aligned period.

Alternatively, select options are available within the Chart section of the ribbon.

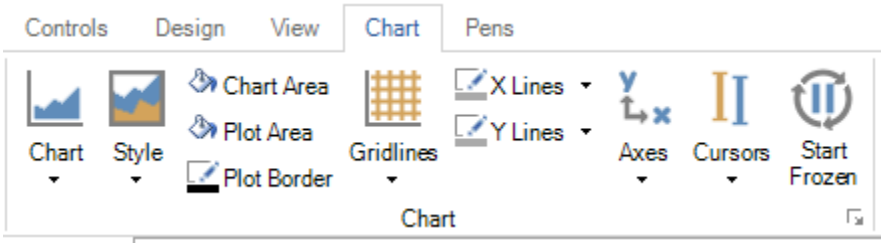


Figure 12.6 – The Chart Section of the Chart Ribbon

12.5.2. Data Tab

On the chart's **Data** tab, shown in Figure 12.7, you can set the following:

- **Sampling Method**
- **Rates**
- **General Date/Time Settings**
- **Global Pen Settings**

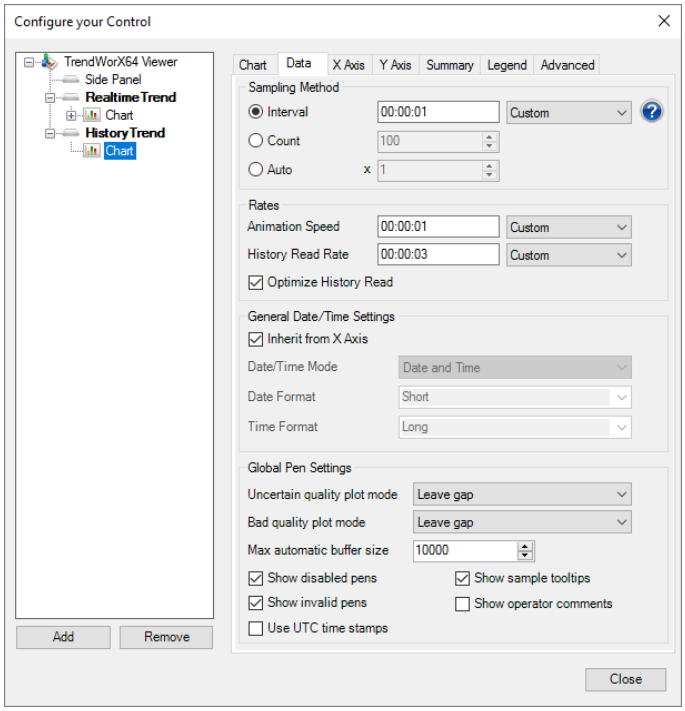


Figure 12.7 - Data Tab of Configure your Control window

Within the Sampling Method section, you can select between three types of sampling:

- **Interval** - The data sampling in interval method is determined by the interval between each sample. The interval and unit can be set in this method.
- **Count** - The data sampling in count method is automatically adjusted to fit the desired count of samples. The desired number of samples can then be set.
- **Auto** - The data sampling in auto method is dynamically set based on the time axis settings. The number of samples per major tick mark can then be set.

Within the Rates section, you can set the following options:

- **Animation Speed** - Set animation speed and units
- **History Read Rate** - Set the desired historical data request interval and units

- **Optimize History Read Rate** - Selecting this will automatically adjust the frequency of reading historical data to improve performance

Within the General Date/Time Settings section, you can set the following options:

- **Inherit from X Axis** - Selecting this box will synchronize date and time format settings from the time axis
- **Date/Time Mode** - Set what components of date and time are available
- **Date Format** - Set the date format in tooltips, legends, and reports
- **Time Format** - Set the time format in tooltips, legends, and reports

Within the Global Pen Settings section, you can set the following options:

- **Uncertain quality plot mode** - Specify if and how samples with uncertain data quality are plotted
- **Bad quality plot mode** - Specify if and how samples with bad data quality are plotted

Additionally, you can set the maximum automatic buffer size as well as choose to show disabled pens, show invalid pens, use UTC time stamps, show sample tooltips, and show operator comments.

Alternatively, select options are available within the Data section of the ribbon.

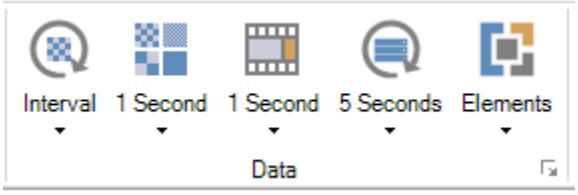


Figure 12.8 – The Data Section of the Chart Ribbon

12.5.3. X Axis Tab

On the chart's **X Axis** tab, shown in Figure 13.9 below, you can set the following:

- **Trend Period**
- **X Axis**
- **Trend Period Toolbar**

Within the Trend Period section, you can set the period of time shown on the trend's X Axis.

Within the X Axis section, you can set the following options:

- **X Axis** - Set whether or not the X Axis is visible
- **Automatic Time Alignment** - Allow the control to automatically set tick marks based on the control's size and actual trend period
- **Date/Time Mode** - Selecting this will automatically adjust the frequency of reading historical data to improve performance
- **Date Format** - Set the date format of the tick mark labels
- **Time Format** - Set the time format of the tick mark labels
- **Major Ticks** - Set the number of major ticks
- **Minor Ticks** - Set the number of minor ticks
- **Time Alignment** - Select this option to ensure time alignment of tick marks and gridlines
- **Base Alignment** - Select this option to ensure base alignment of tick marks and gridlines, which will factor in constant tick marks as well as time-aligned tick marks

Additionally, users can select the **Style…** button to further configure the appearance of the X Axis.

Within the Trend Period Toolbar, you can select to include a toolbar that will readjust the Trend Viewer to show a specific period of time based on selected increments or manually set increments.

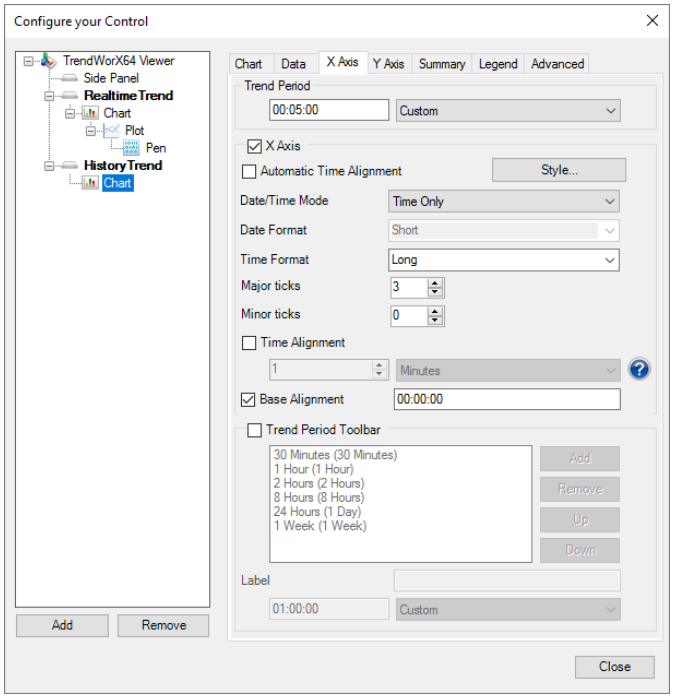


Figure 12.9 - X Axis Tab of Configure your Control window

Alternatively, select options are available within the X Axis section of the ribbon.

Figure 12.10 – The X Axis Section of the Chart Ribbon

12.5.4. Y Axis Tab

On the chart's **Y Axis** tab, shown in Figure 12.11, you can set the following:

- **Y Axis options**

Within the Y Axis section, you can set the following options:

- **Y Axis** - Set whether or not the Y Axis is visible
- **Zoom Ruler** - Set whether or not a zoom ruler is present on the Trend Viewer
- **Pen Axis** - When selected, the range of the pen selected is automatically moved to the next chart area

Additionally, you can choose to set a Global Range for the Y Axis. If a Global Range is set, all pens will share the same value axis for its Y coordinates. If not selected, each pen will have its own value axis. The Global Range can then be modified with a minimum and maximum value, set automatically or manually. The Y Axis tick marks can also be set with an Auto Scale margin percent while also having the total number of Major and Minor ticks set. Select the **Style…** button to further configure the appearance of the Y Axis.

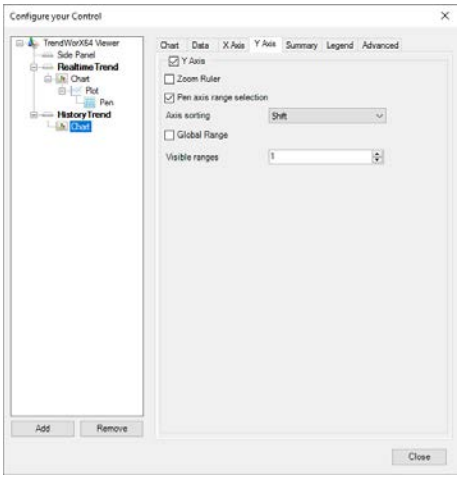


Figure 12.11 - Y Axis Tab of Configure your Control Window

Alternatively, select options are available within the Data section of the ribbon.

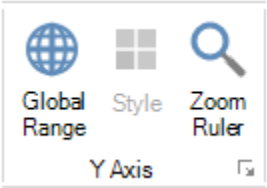


Figure 12.12 – The Y Axis Section of the Chart Ribbon

12.5.5. Summary Tab

On the chart's **Summary** tab, shown in Figure 12.13, you can set the following:

- Summary View
- Summary X Axis

Within the Summary View section, you can set the following options:

- **Summary View** - Show/hide the view used to display data history for the tags in the chart control
- **Position** - Set the positioning of the summary view within the chart control
- **Size** - Set the size of the summary view within the chart control
- **Show Plot Area** - Show/hide the plot area of the summary view
- **Show Scrollbar** - Show/hide the scrollbar of the summary view
- **Show Expand Button** - Show/hide the expand button of the summary view
- **Snap Mode** - Enable to snap summary cursor and scrollbar to the closest tick marks
- **Cursor Fill** - Set the fill color of the summary view cursor
- **Cursor Handles** - Set the color of the cursor handles of the summary view
- **Time Aligned Cursors** - If time alignment is enabled on the X axis, dragging summary cursors results in a time aligned period
- **Summary Period** - Set the period in which the summary will aggregate

Within the Summary X Axis section, you can set the following options:

- **Summary X Axis** - Show/hide trend period axis in summary chart
- **Custom Summary Axis** - Set whether or not to use custom settings for trend period axis in summary chart
- **Date/Time Mode** - Selecting this will automatically adjust the frequency of reading historical data to improve performance
- **Date Format** - Set the date format of the tick mark labels

- **Time Format** - Set the time format of the tick mark labels
- **Major Ticks** - Set the number of major ticks
- **Minor Ticks** - Set the number of minor ticks
- **Time Alignment** - Select this option to ensure time alignment of tick marks and gridlines
- **Base Alignment** - Select this option to ensure base alignment of tick marks and gridlines, which will factor in constant tick marks as well as time-aligned tick marks

Select the **Style…** button to further configure the appearance of the Y Axis.

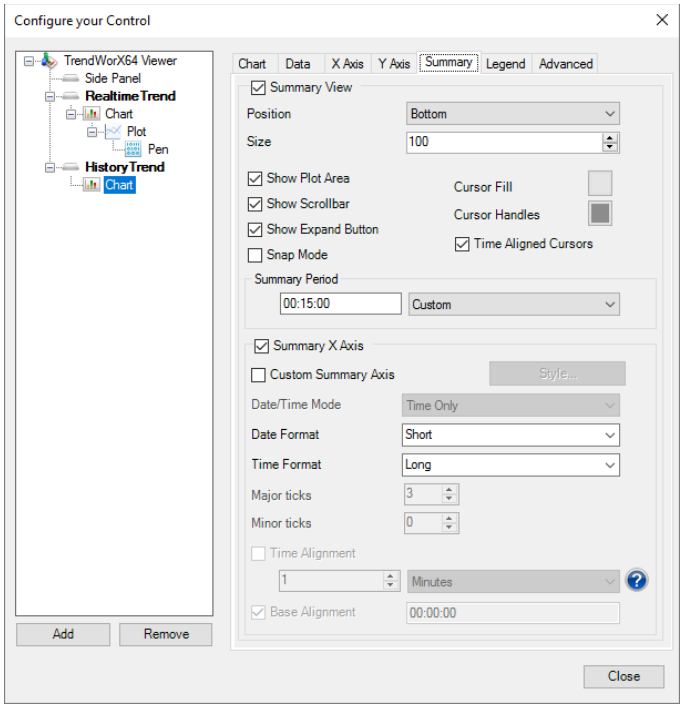


Figure 12.13 - Summary Tab of Configure your Control window

Alternatively, select options are available within the Summary section of the ribbon.

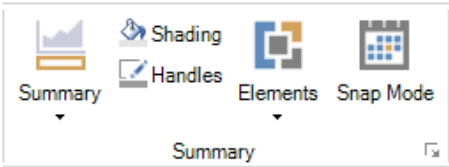


Figure 12.14 – The Summary Section of the Chart Ribbon

12.5.6. Legend Tab

The **Legend** shows the assignment of pens. On the chart's **Legend** tab, shown in Figure 12.14 below, you can set the following:

- Legend Options

Within the Legend section, you can set the following options:

- Legend - Set whether or not a legend is visible
- Position - Set the position of the legend on the chart control
- Size - Set the size of the legend
- Font Family - Set the font for the text in the legend
- Font Size - Set the font size for the text in the legend
- Colored Details - Show/hide text elements and other details in the legend by the color of their associated pens
- Show Header - Show/hide the legend header

Within the Legend section, you can also modify the appearance of the legend by setting the Background, Header color, Row color, Selected Row color, Inactive Row color, Header Border, Header Text color, Row Text color, Selected Row Text color and Inactive Row Text color. Below the appearance settings, you can also select which criteria are displayed as headings as well as what title will be displayed.

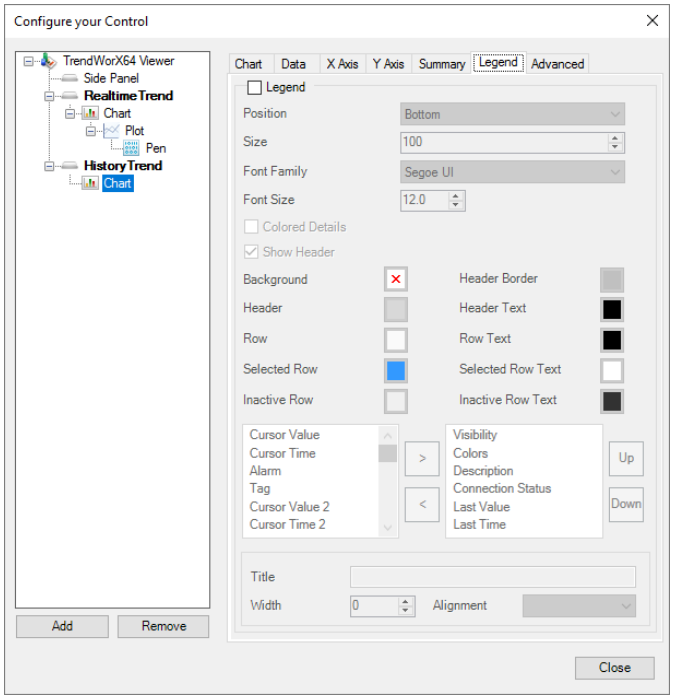


Figure 12.15 - Legend Tab of Configure your Control window

Alternatively, select options are available within the Legend section of the ribbon.

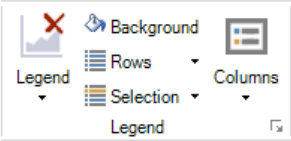


Figure 12.16 – The Legend of the Chart Ribbon

12.6. 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 12.17. 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.

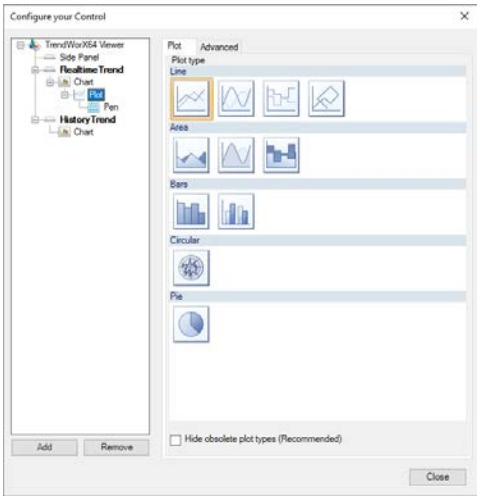


Figure 12.17 – Different Plot Types in TrendWorX64 Viewer

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.

12.7. 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 Ranges tab, the Calculations tab, the Ideal Pen tab, and the Advanced tab.

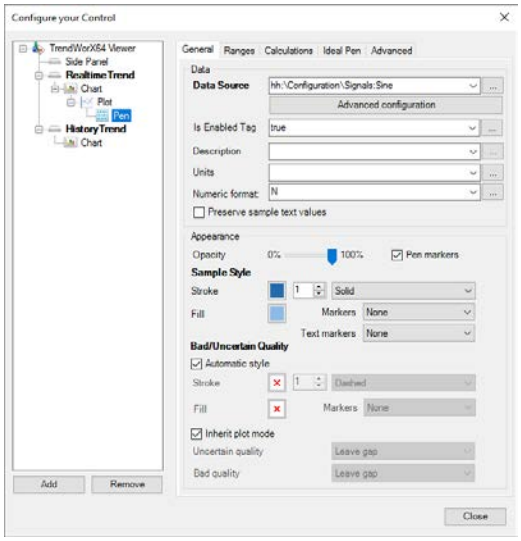


Figure 12.18 – Pen Configuration in TrendWorX64 Viewer

12.7.1. 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 12.19.

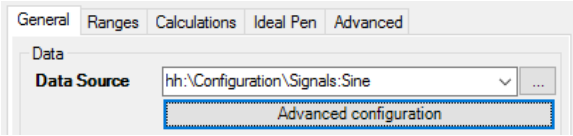


Figure 12.19 – Pen Data Source Advanced Configuration Button

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.

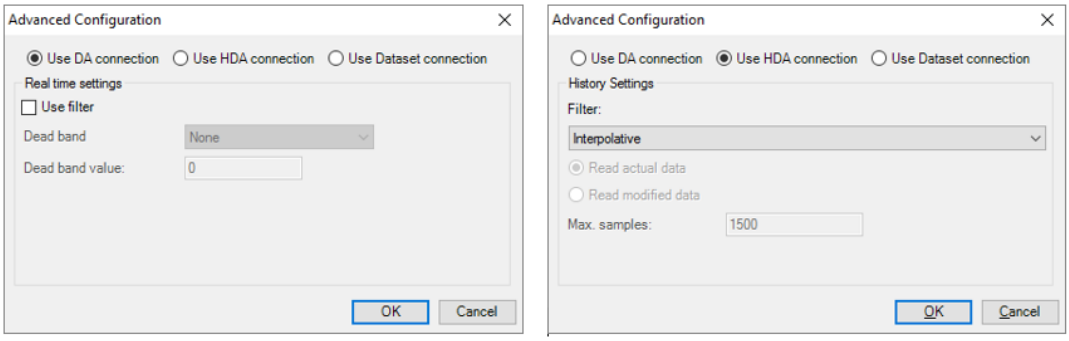


Figure 12.20 – Data Source Advanced Configuration, (left) Setting Real-Time or (right) Historical

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.

12.7.2. Ranges Tab

The settings of the Ranges tab are dictated by the Global Range. If you prefer to set the range manually, select **Turn Global Range off**. The scale for the pen can be set to **Auto**, **AutoScale** or **Manual**. 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).

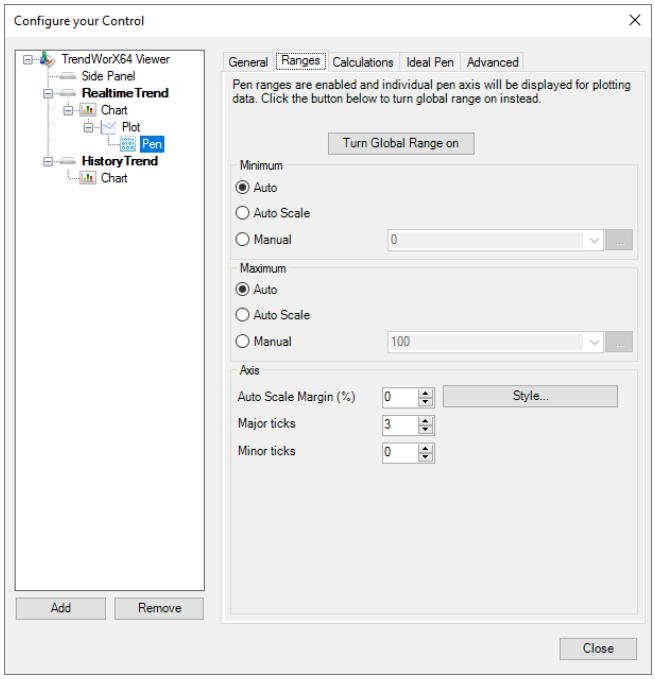


Figure 12.21 – Pen Ranges Tab

12.7.3. Calculations 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.

This is also where you would enable and configure the **Totalizer** of a pen. Think of the Totalizer as a counter. The Totalizer gives you a count for a particular period, expressed in the units being collected. When a value is collected, the Totalizer updates the count then outputs the count to the viewer.

The following examples illustrate uses for the Totalizer option:

- For a signal that comes from a flow rate gauge that measures liters, the Totalizer gives you the total number of liters that pass through during an interval that you specify, such as liters per minute.
- For an electrical power gauge that measures kilowatts, the Totalizer gives you the total energy consumed during a measured interval, such as kilowatts-hours.

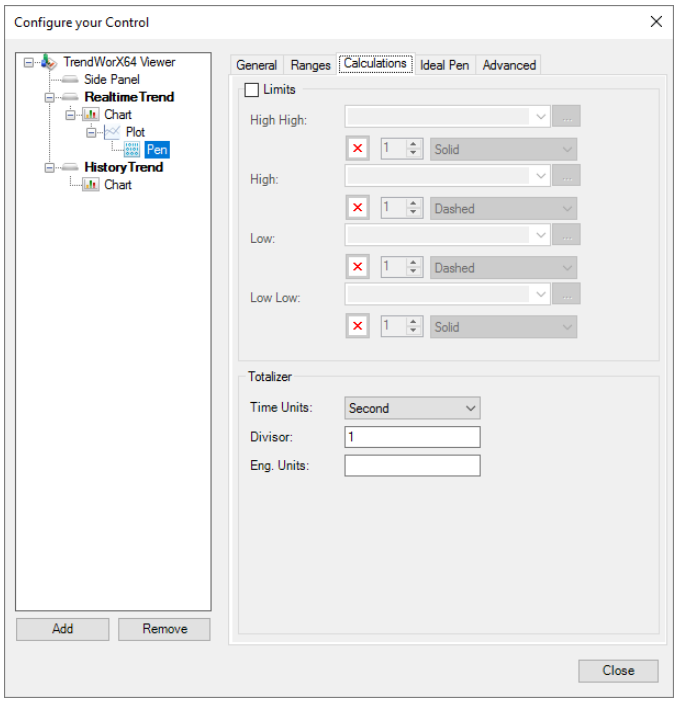


Figure 12.22 - Pen Calculation Tab

If you choose to enable this option, you must go to the **Time Units** field, specify the time measure for capturing the throughput rate per **Second**, **Minute**, **Hour**, or **Day**. For example, to get liters per hour select **Hour**, but to get liters per minute select **Minute**.

To choose the unit that will give you an accurate totalized count, you need to *anticipate how fast your raw input signal is going to change*. For example, if the raw input signal takes more than an hour to change, then a sampling rate of once per *hour* is reasonable. However, if the raw signal changes significantly over the period of an hour, you should choose a faster collection rate of one per *minute*.

12.7.4. 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 12.23, you can use the Ideal Pen tab to make a pen an ideal pen and set its starting time and date.

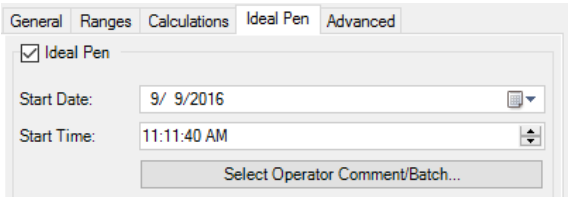


Figure 12.23 - The Ideal Pen Tab

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.

12.8. 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 ribbon will allow you to access additional Data Analysis options. Figure 12.24 shows the runtime ribbon.

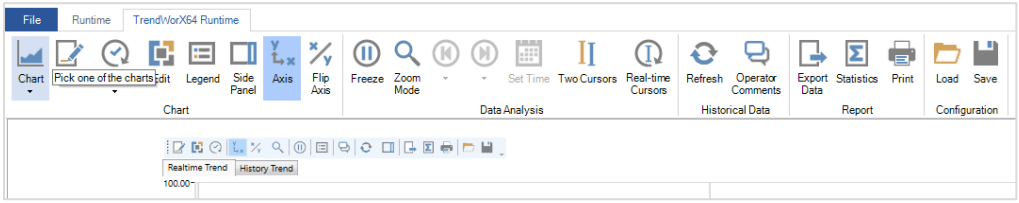


Figure 12.24 – Viewer Runtime Ribbon (top) and Toolbar (middle) in Unfrozen Mode

Many of the buttons on the Runtime Ribbon and Toolbar provide access to some properties configurable in both Configure and Runtime modes, such as **Chart**, **Plot** and **Pens**.

Some options, such as setting the trend period back/forward and Set Time are only available when the viewer is in Freeze mode.

Selecting “Chart” allows you to select between indexed charts.

Selecting “Pens” opens the chart’s **Configure your Control** dialog, allowing you to modify the chart, plot, and pen display. Modifications appear immediately in the runtime display, as soon as you close the dialog.

Selecting visible range will allow you to reset the visible range in various increments.

The “Edit” button opens the **Edit Trend** dialog box shown in Figure 12.25. 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 during runtime.

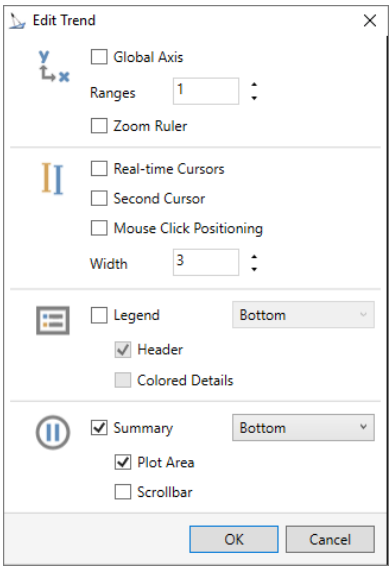


Figure 12.25 - Edit Trend Dialog Box

Selecting “Legend” will show/hide the legend.

Selecting “Side Panel” will show/hide the side panel.

The “Axis” button will show/hide the X and Y axes, giving your chart more room to display.

The “Flip Axis” button will switch the X-axis with the Y-axis, which can make visualization of the chart easier in some instances.

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 backward are forward arrows to move forward and backward in time. Additionally, you can set the specified time using the “Set Time” button. Clicking the Freeze button a second time exits Freeze mode and returns you to real-time mode.

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.

Clicking the “Two Cursors” button will show/hide a second cursor on the trend.

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 12.26 shows how the summary view can be manipulated to show a specific timeframe of data.

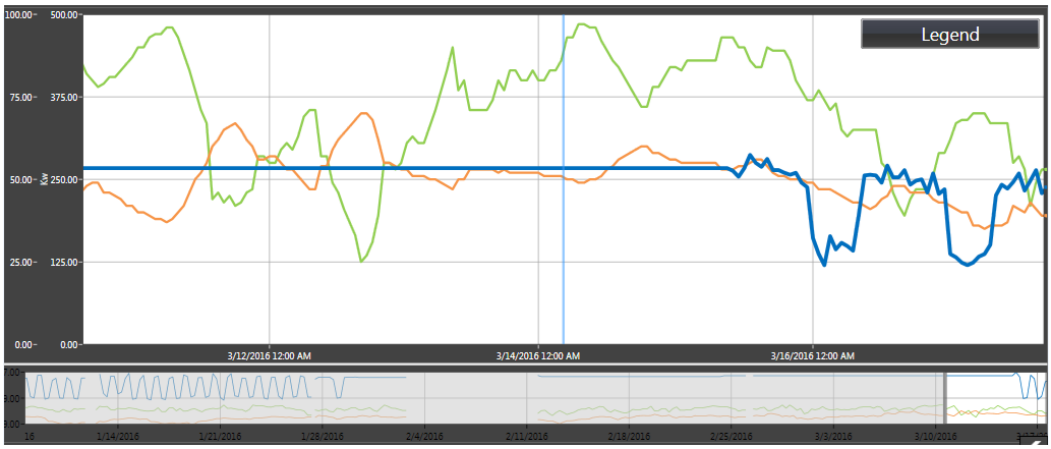


Figure 12.26 - Summary View

> **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.

Lab: Trend Viewer L12

12.1. Lab Overview

icTonics wants to view logged history data that they deem important. They also want to view real time data 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**

12.1.1. 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

12.1.2. Support Files

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

- icTonicsMain.gdfx

12.2. TrendWorX64 Viewer Control

You can use the TrendWorX64 Viewer control to view real-time or historical data in GraphWorX64. In this lab, we will create a GraphWorX64 display that will display both types of data.

12.2.1. TrendWorX64 Viewer Control Configuration

1. Open **GraphWorX64** if you closed it after the previous lab.
2. Click on **File → New Settings**, and at the bottom change the **Layout** parameter from Canvas to Grid. Then click OK.
3. Save this new display as **icTonicsTrends.gdfx** in our project folder, **C:\Training**. Ensure you save your work periodically as you progress through the lab.
4. Select the background so that the Properties pane on the right indicates **This Display** : **Grid Panel**. Set **Measurements → IsScalable** = **True**.
5. 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 and drawing a box for the viewer.
6. In the **Properties** pane right-click on the **Width** and **Height** properties and select **Reset** to set them to automatic.
7. Double-click on the control to bring up the "**Configure your Control**" dialog box.

   Note: You can also click on the "Edit" button in the Configuration portion of the Chart Tab that will automatically open when a Trend Viewer is selected.

8. Click on the **Tab** entry in the tree on the left and change the **Title Text** to **icTonicsTrends**.
9. Rename the Tab entry in the tree **icTonicsTrends**.
10. Confirm that beneath the **Chart** in the Explorer Tree there is a **Plot**. New plots can be added by right-clicking on **Chart** and selecting **Add → Plot**.
11. Confirm that beneath the **Plot** in the Explorer Tree there is a **Pen**. New pens can be added by right-clicking on **Plot** and selecting **Add → Pen**.

12. Click on the **Pen** and confirm the **General** tab is selected.

13. Click on the ellipsis […] button to the right of the **Data Source** textbox.

14. In the **Data Browser**, browse to **My Computer → Data Connectivity → Modbus → icTonics → Tanks → USA → Foxboro → Tank1 → TankLevel** or, if you have added icTonics to **Favorites**, browse to **icTonics → Tanks → USA → Foxboro**
→ Tank1 → TankLevel.

15. Click on "**OK**" to exit out of the Data Browser and return to your TrendWorX64 Viewer configuration dialog box.

16. Still on the pen's **General** tab, click the ellipsis […] button to the right of the **Connection** textbox, then click **Configure**. In the popup dialog box, make sure "**Use DA connection**" is selected. Click **OK**.

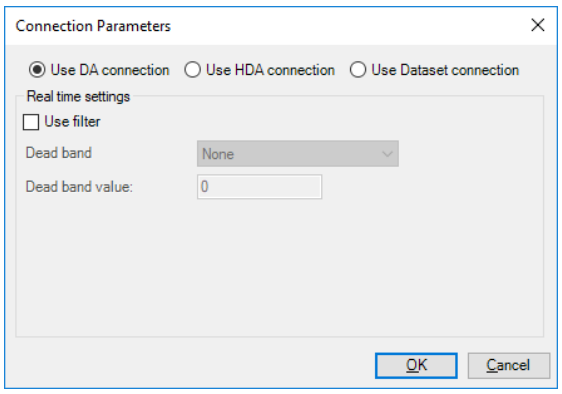


Figure L12.1 – Connection Parameters

17. Remaining on the pen's **General** tab, change the description of the pen to **Tank1 Level**.

18. In the Appearance section, change the **Stroke Thickness** to **3** and the color to **Purple** (R:112, G:48, B:160).

Figure L12.2 – Solid Color Setting for Purple

19. Your finished pen configuration should look similar to the figure below.

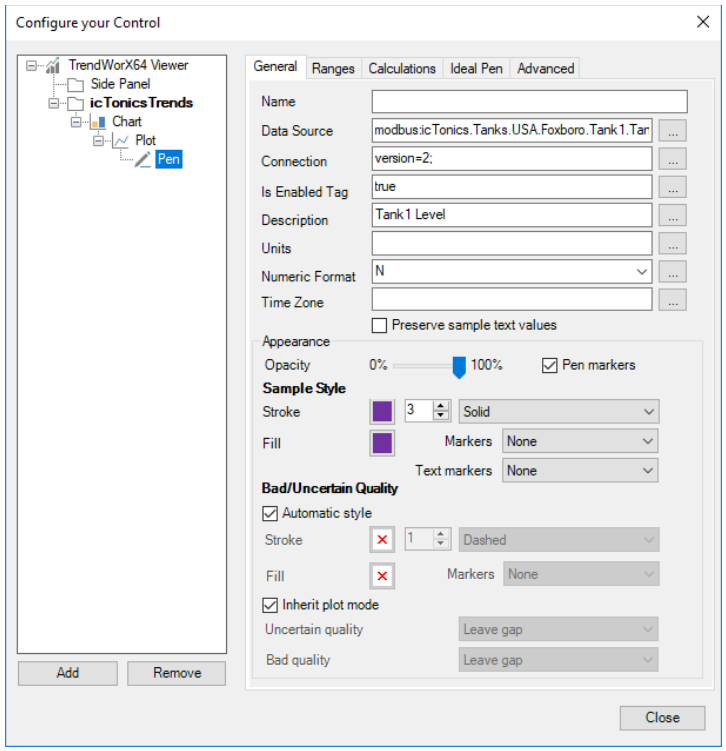


Figure L12.3 – Real-Time Pen Configuration

20. Right-click on **IcTonicsTrends** on the left and select **Add → Chart**.

21. Select the **Data** tab on the new chart.

22. In the **Sampling Method** section, change the **Interval** to **5** seconds. As you may recall, we set the Trend Logger Data collection rate to 5 seconds also.

23. Select the **Range** tab.

24. In the **Realtime Mode** section, change the **History Read Rate** to **30** seconds.

25. Now we are going to add a historical pen to the new chart. We will repeat the same steps that you executed previously.

26. Add a **Plot** and a **Pen**.

27. Set the **Data Source** to **My Computer → Historical Data → Hyper Historian → Configuration → TankLevels → Tank1 Level**. The Data Source should now read:

hh:\Configuration\TankLevels:Tank1 Level

28. From within the **General** tab, type the new Pen the **Description** = **Tank1 Level History**.

29. Change the **Stroke** to **3** and set **Fill** and **Stroke** color to **Dark Blue** (R:0, G:112, B:192).

30. Select **Circle** as the **Markers** type.

31. Click the ellipsis […] button to the right of the **Connection** textbox, then click **Configure**. In the popup dialog box, make sure "**Use HDA connection**" is selected. Click **OK**.

32. On the **Ranges** tab select **Manual** for both the **Minimum**, and **Maximum** and leave the default values of 0 and 100. **Note:** If you leave the Ranges set to **Auto** they will pull from the range fields of Hyper Historian in the Workbench.

33. Your finished historical pen configuration should resemble the following image:

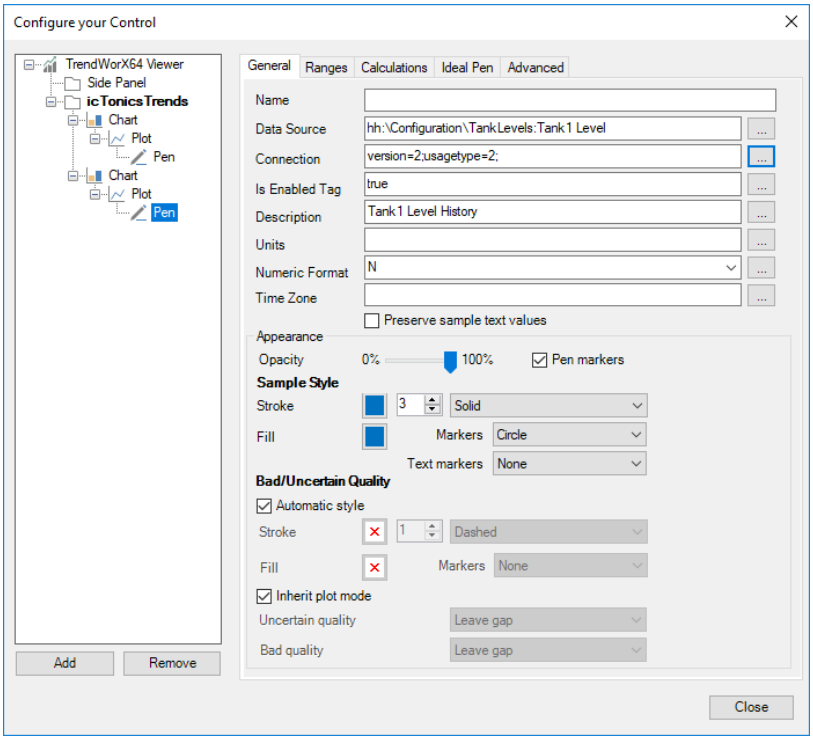


Figure L12.5 – Historical Pen Configuration

34. Click on **Close** to close the configuration dialog box.

35. Save this (**icTonicsTrends.gdfx**) display and go into **Runtime**.

12.2.2. The TrendWorX64 Viewer Display

1. Launch a second instance of GraphWorX64 (you can right-click on the GraphWorX64 icon in the task bar and select GraphWorX64).
2. Load the **icTonicsTankMonitor** display into **Runtime**.
3. Change the **Tank1 Level Setpoint** 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.
4. Notice that your historical trend has circles indicating data collection points, as shown in the figure below.

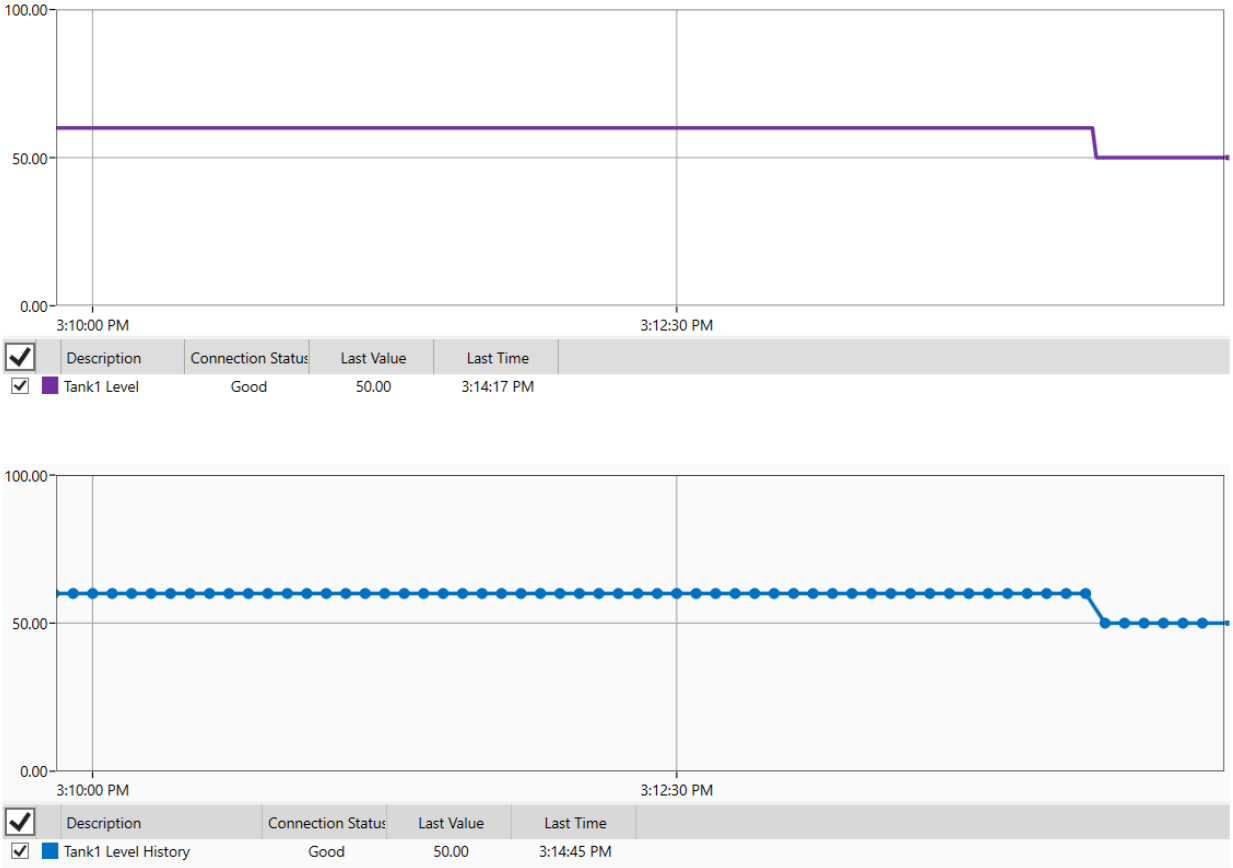


Figure L12.5 – TrendWorX64 Viewer

12.2.3. Trend Embedded Window

We will now add a button to the **icTonicsMain.gdfx** display to load the trend display we created earlier.

1. In the **icTonicsMain.gdfx** display, in the **ButtonsPanel** at the top select the **HomeButton** button and press **Ctrl+D** to duplicate the button. Note: We are duplicating this button to act as a basis for our new button.
2. Rename our duplicated button to **TrendButton.**
3. Select the **Dynamics** tab and click on the **Pick Action** dynamic.
4. Expand **CommandParameters** and change the **FileName** to "**icTonicsTrends.gdfx**".
5. Expand **TrendButton** in the explorer and find the **Label** object. Select it and in the property grid at right set the text to be "Trend".
6. Open the **Symbols** pane and type trend into the search bar. Choose a symbol right-click on it and select copy.
7. In the Explorer tree expand out the elements of the button until you get to the group with the house icon. Right-click on the path for the house icon and choose paste.

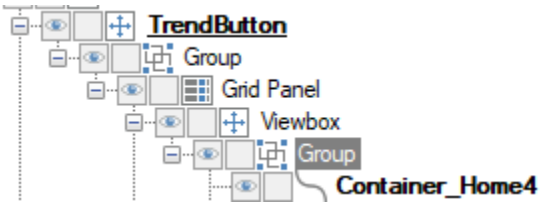


Figure L12.6 – TrendButton – in this image "Container_Home4" is the house icon

8. Select your new icon and then the house icon, then in the **Arrange** tab of the Home ribbon select the align button and choose **Align Middle and Center**.
9. You can now delete the old house icon.
10. Save the **icTonicsMain** display.
11. Open the **icTonicsTrends**.gdfx display if you closed it.
12. In the Properties Pane under Measurements – Grid click on the ellipses next to **RowDefinitions.**

13. Add a new row and change the **Height** of Row 1 to 10 Star, and click OK.

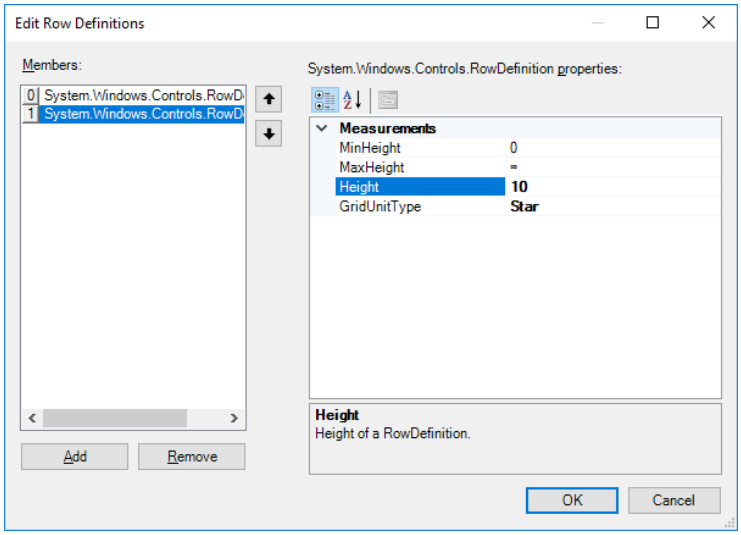


Figure L12.7 – Edit Row Definitions

14. Click on the **TrendWorX64 Viewer** and in the **Measurements – Grid** section of the Properties Pane change the Row from 0 to 1.

15. Add a **Text Label** (Home tab → Shapes) to the new top row, double-click on the label and change the text to "**Tank 1**".

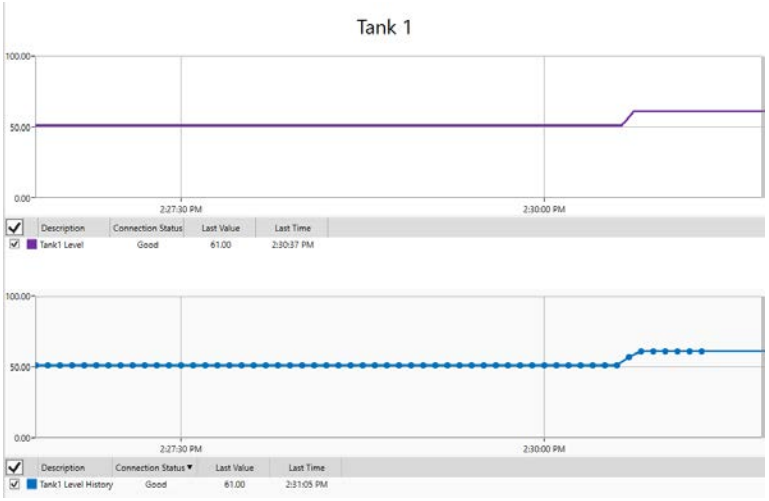


Figure L12.8 – Trend Display

16. Your finished configuration should resemble the previous figure.

17. Save the display.

18. Go back to your **icTonicsMain** display in **Runtime**.

19. Click on the Trend button to load the new trend display into the content window.

This concludes this lab.

Global Aliasing 13

The Global Aliasing System (GAS) is a powerful referencing feature that can be used to quickly switch between multiple data sources in GENESIS64 applications like, GraphWorX64, TrendWorX64 Viewer, and AlarmWorX64 Viewer 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 alias references. 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 display. 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.

13.1 Objectives

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

The Global Aliasing System follows a certain structure of values that groups common objects into categories. Consider Table 13.1, which demonstrates the global aliasing hierarchy in the left column on an example of building control in the right column.

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

Table 13.1 - Global Aliasing Mapping

13.2 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 13.1 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.

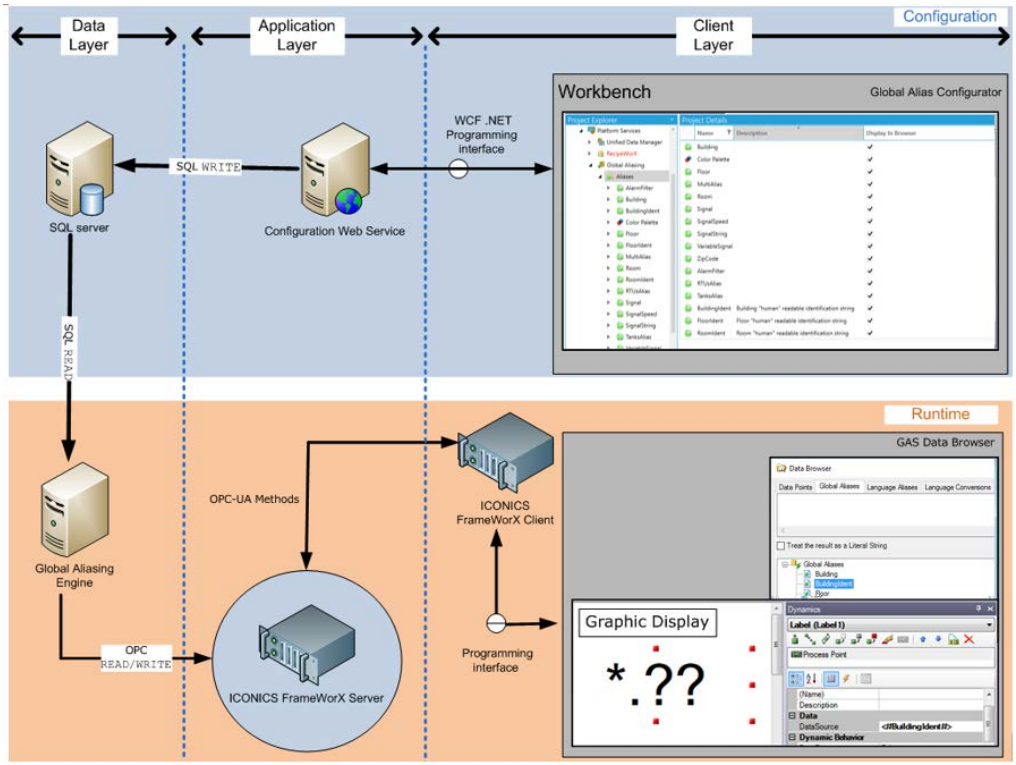


Figure 13.1 - Global Aliasing System (GAS) Architecture: Configuration (Top) and Runtime (Bottom)

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.

13.3 Defining a 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 13.2 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.

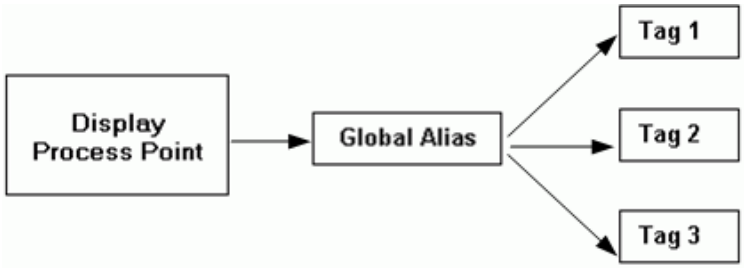


Figure 13.2 - A Single Global Alias for Multiple Data Sources

A collection of aliases with similar properties is called a theme. Aliases can be nested, eventually resolving to the value or property you wish.

13.3.1.1 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!

13.3.1.2 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.

13.4 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 13.3 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).

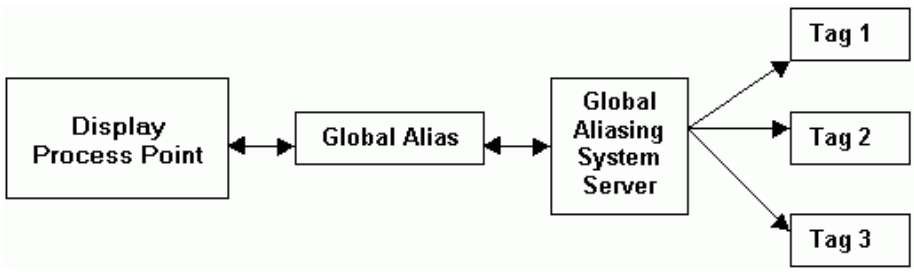


Figure 13.3 - Alias Resolution

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 13.5. 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 13.4. 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 13.4. You can also associate multiple aliases with a single theme.

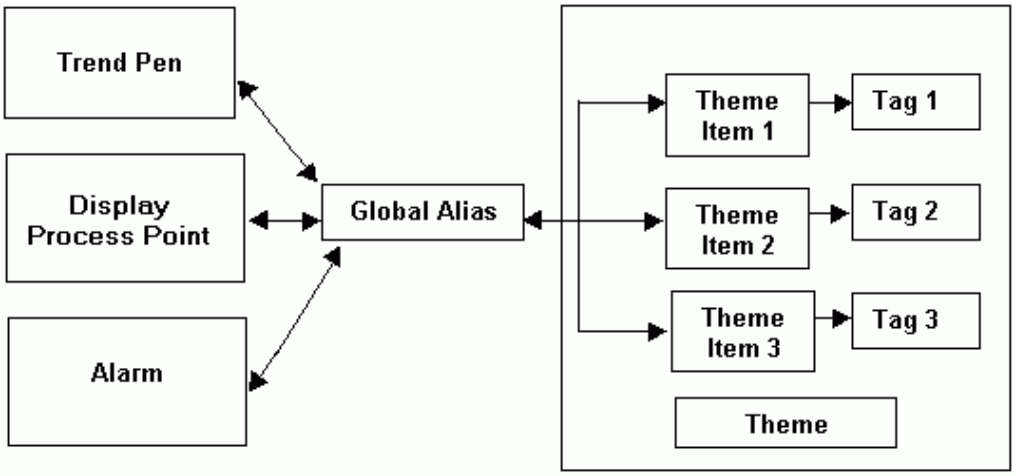


Figure 13.4 - Using Theme Values to Switch between Data Values

13.5 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 13.5.

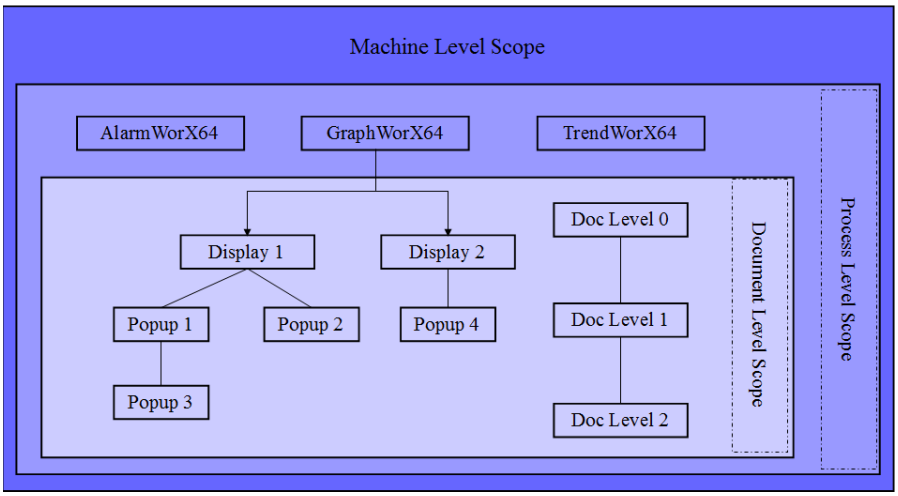


Figure 13.5 - Theme Scope

13.6 Configuring Themes and Global Aliases

To define themes and global aliases, open the Global Aliasing configuration interface in the Workbench by clicking **Platform Services** → **Global Aliasing** in the Project Explorer. After installation of GENESIS64, you will find a sample Global Aliasing configuration database, which demonstrates an example of themes and global aliases.

13.6.1 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 "**Refresh**".

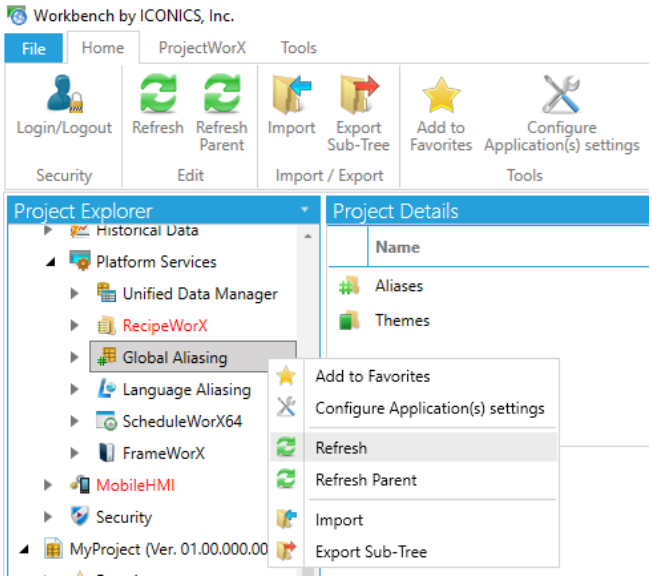


Figure 13.6 - Reload Configuration when Modified

13.6.2 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 13.7.

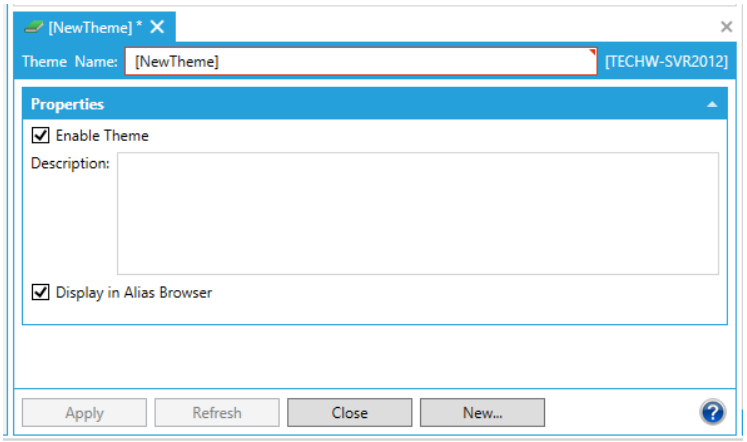


Figure 13.7 - Adding a New Theme

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 13.8, a theme value of *Building A* 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 13.8, the theme value is enabled and will be listed under the Theme tab that appears in the Data Browser.

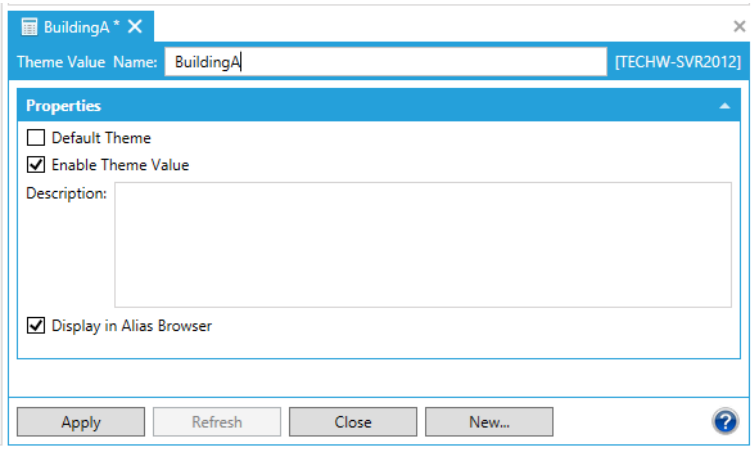


Figure 13.8 - Theme Values for the Theme Called Buildings

13.6.3 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 13.9, appears.

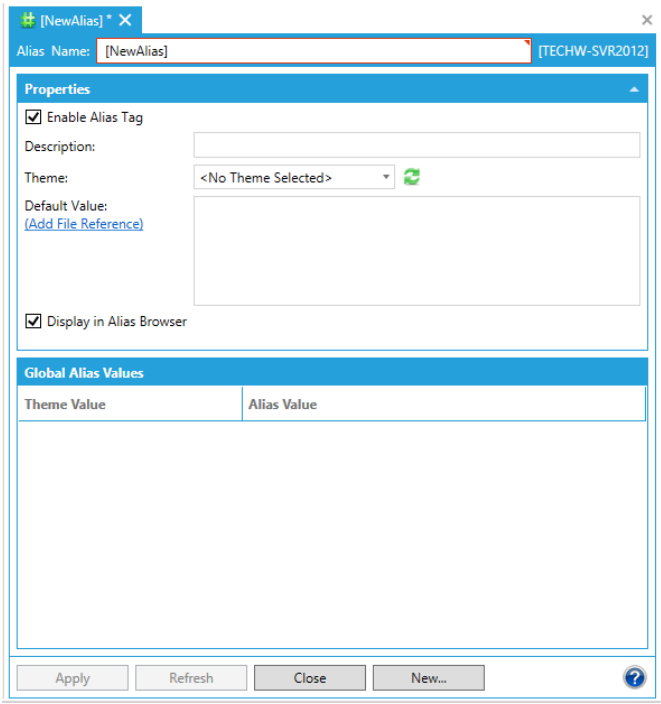


Figure 13.9 - Adding a Global Alias Definition

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 13.10. When adding a global alias to the alias value, use the following syntax:

  <#GlobalAlias#>

Alias Values

Theme Value	Alias Value
Doors	Doors are open in <#Floor#> <#Room#>
Lights	Lights are off in <#Building#> <#Floor#> <#Room#>
Temperature	Temperature is high in <#Room#>

Figure 13.10 - 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.

13.7 Using Global Aliases

13.7.1 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.

13.7.2 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 13.11.

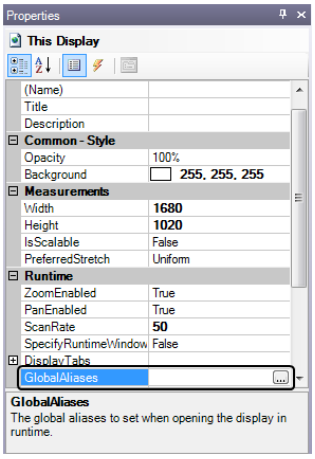


Figure 13.11 - Global Aliases Property for This Display

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 13.12.

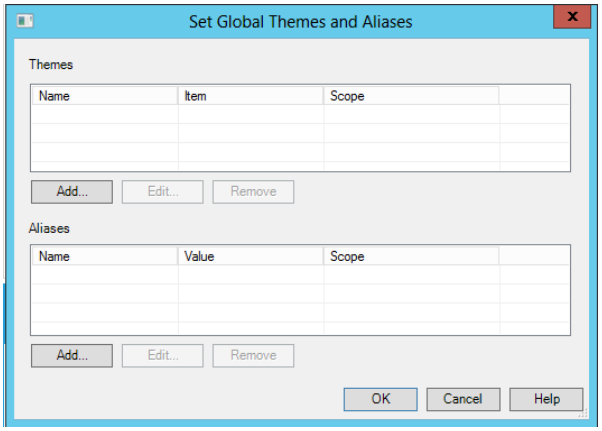


Figure 13.12 - Set Global Theme and Aliases Dialog

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 13.13 where you select a theme and its scope for the display.

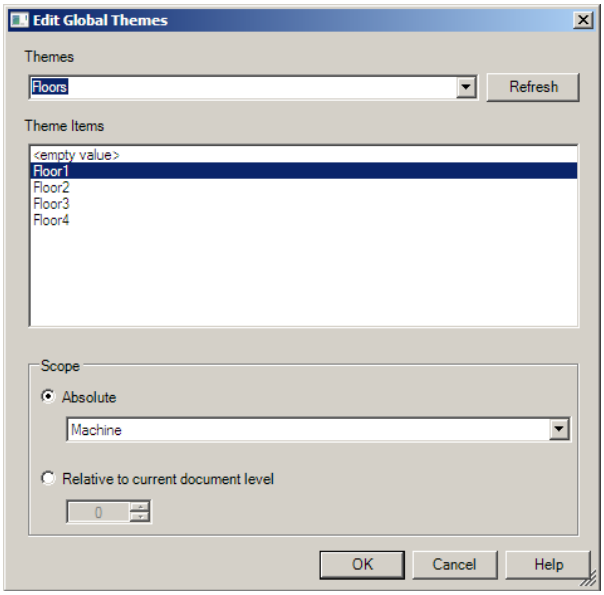


Figure 13.13 - Selecting the Display’s Themes

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.

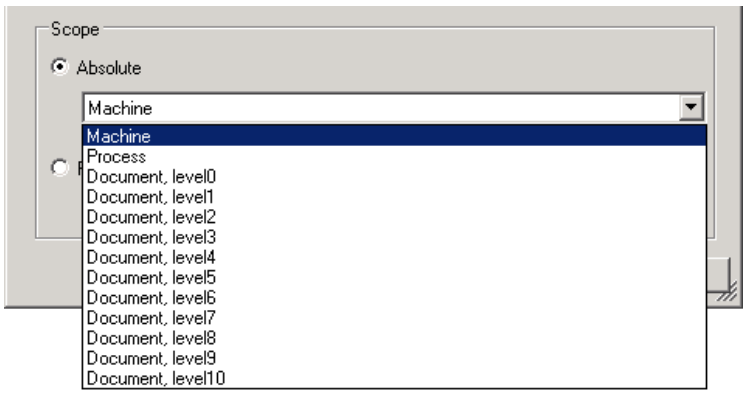


Figure 13.14 - Absolute Scope Options

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 13.15 where you select a global alias, optionally a value, and the scope.

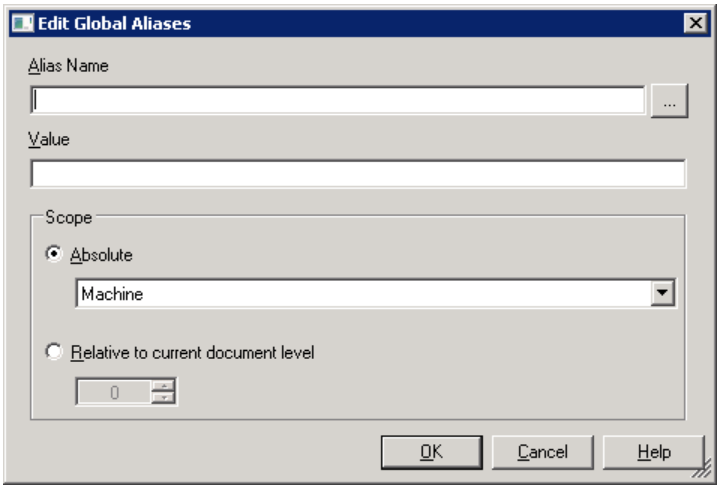


Figure 13.15 - Edit a Display's Global Aliases

13.7.3 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.

13.7.4 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.

13.7.5 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 13.16 to specify the theme using the "GAS Theme" field. Clicking on the button opens the **Set Global Themes and Aliases** dialog box, as shown in Figure 13.16, where you set Global alias themes and aliases the same way you did for GraphWorX64.

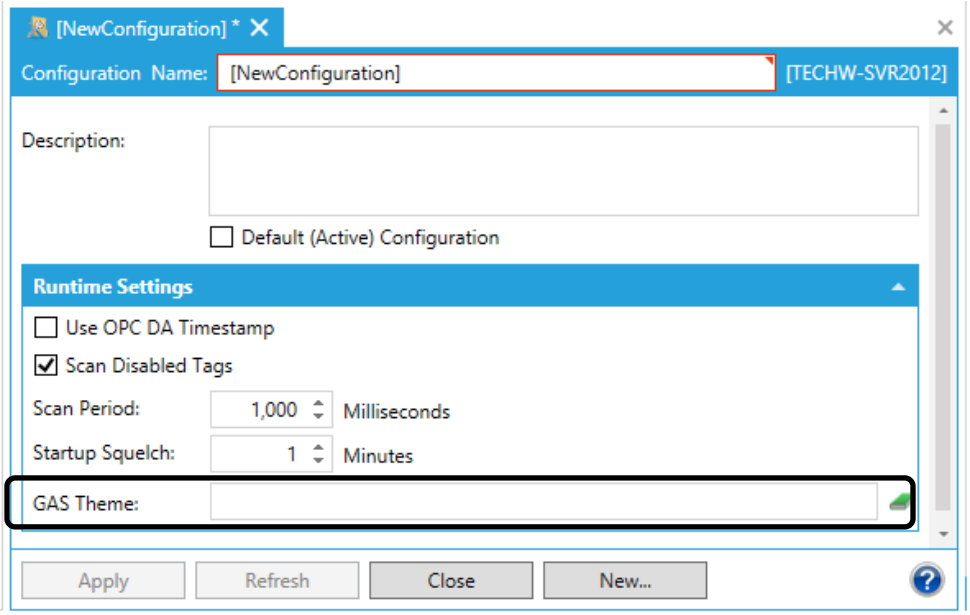


Figure 13.16 - AlarmWorX64 Server Configurator Sets the Theme

13.8 Global Color Palettes

13.8.1 About Global Color Palettes

GENESIS64 provides the ability to create Global Color Palettes that provide the following benefits:

Swappable Palettes of Colors

One possible use for Global Color Palettes is for applications that require separate distinct Day/Night modes, such as for navigation purposes, marine use, etc.

For example, one defined color could be named “Background” and could be set to be white for “Day Mode” and to black for “Night Mode”. Then, the color setting of “Background” could be configured as (assigned to) the Background property for a Display or Control. Switching the active palette from 'Day' to 'Night 'mode would cause the background of the Display/Control to switch from White to Black.

Named Colors

Another possible use for Global Color Palettes is 'color standardization' within a project. For example, a defined color named “HighAlarm” could be set to red.

Later on, that color could be redefined as yellow, but the change would only need to be made in one place (the Global Color Palette). All references to that color in HMI displays will automatically use the updated color.

Dynamic Colors

This use for Global Color Palettes is useful for colors that get their values from tags. For example, one single tag could provide one 32-bit color value. Three tags could provide Red, Green, and Blue (RGB) values. Four tags could provide Alpha (for transparency/opacity), Red, Green, and Blue values.

13.8.2 Configuring Global Color Palettes

The configuration of Global Color Palettes is done through the Global Aliasing provider in the Workbench, as show below.

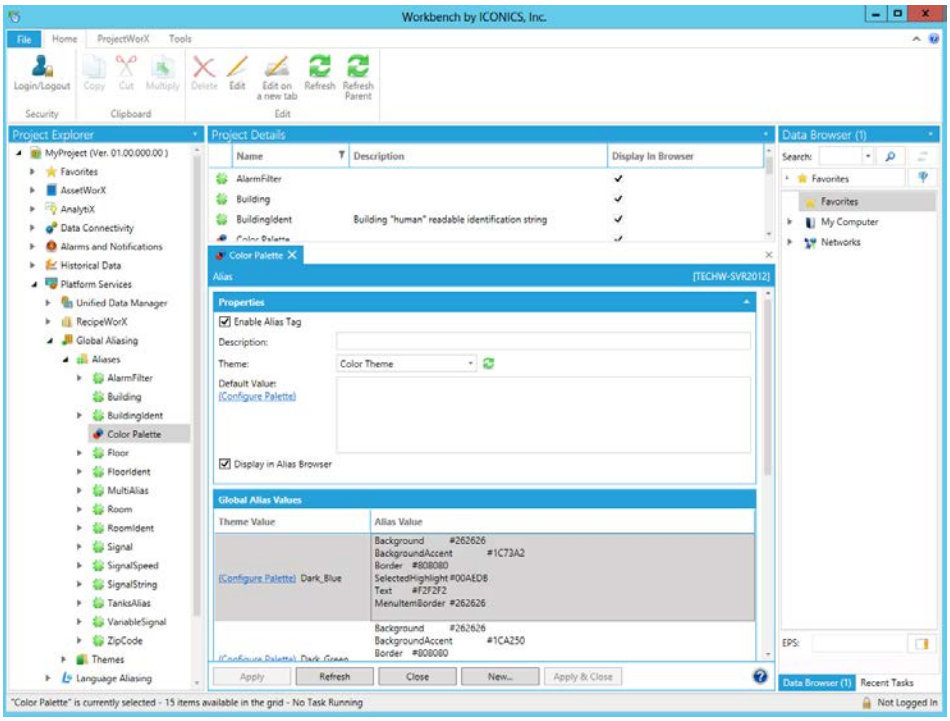


Figure 13.17 - Configuring a Global Color Palette in the Global Aliasing Provider in the Workbench

13.8.3 Create a Color Palette Theme

Expand the Themes node within the Global Aliasing provider and click on the **Color Palette** theme. You'll see the Color Palette properties in right-side window within the Workbench, as shown below.

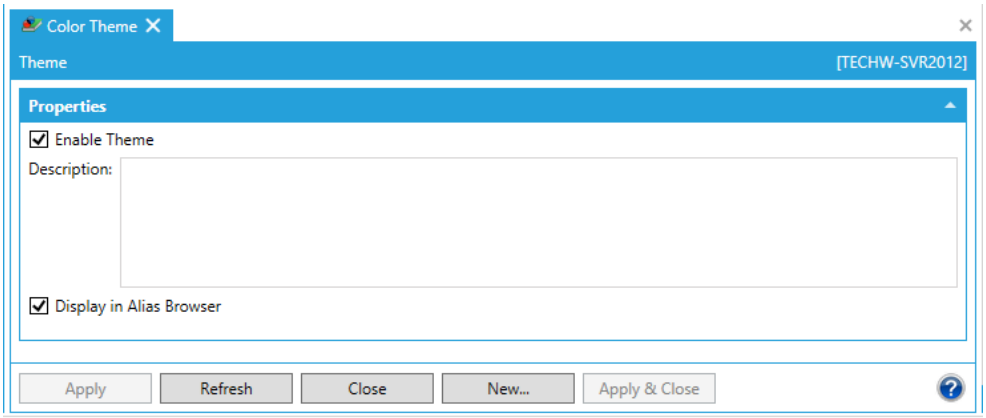


Figure 13.18 - Color Palette Theme in the Global Aliasing Provider in the Workbench

You can change the description, decide if this theme should be displayed in the Alias Browser and if you wish to use a separate Display Name for this theme within the Alias Browser (if you click this checkbox, you can enter a different Display Name in the text entry field or click on the ellipsis [...] to select one through the Data Browser). Click 'Apply & Close' to proceed.

13.8.4 Add Theme Values

Right-click on the Color Palette theme and select 'Add Theme Value'. This opens the Theme Value properties in the right-side window of the Workbench, as shown below.

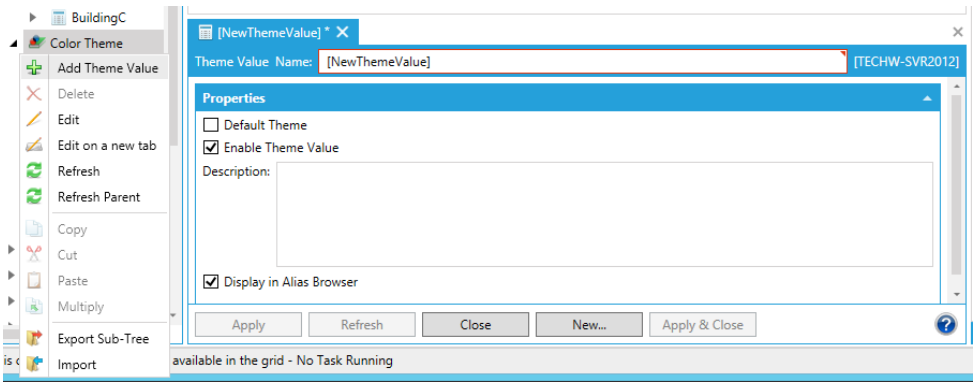


Figure 13.19 - 'Dark Theme' Theme Value Properties

You can give your Theme Value a Name. In this example, we'll be setting up a 'Dark Theme'.

You can also change the description, decide if the connected theme should be the Default Theme, decide if this theme value should be displayed in the Alias Browser and if you wish to use a separate Display Name for this theme value within the Alias Browser (if you click this checkbox, you can enter a different Display Name in the text entry field or click on the ellipsis [...] to select one through the Data Browser). Click 'Apply & Close' to proceed.

In this example, we'll also create a second Theme Value following the same steps and call that one 'Light Theme'.

13.8.5 Edit a Color Palette Alias

Click on the **Aliases** branch and you'll see the Color Palette theme in the right-side window, as shown below.

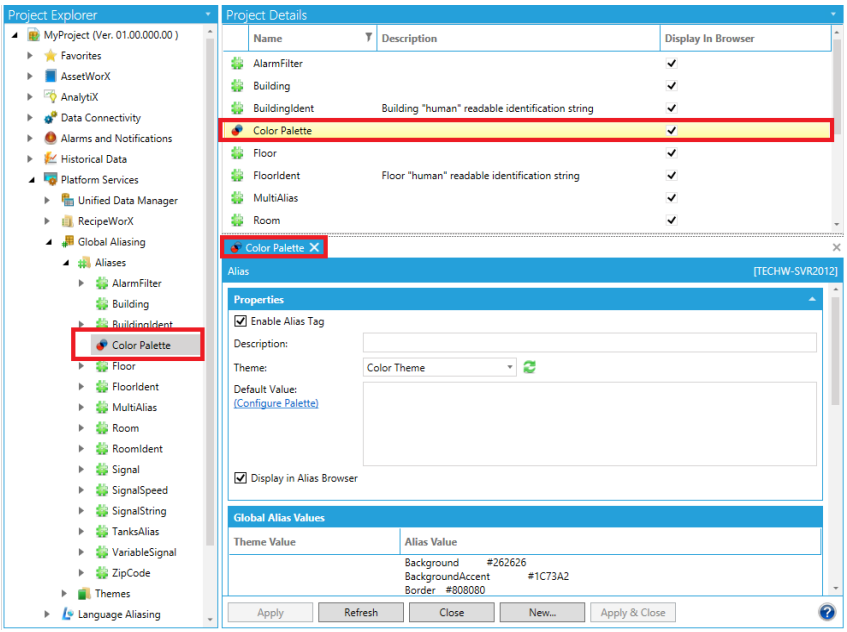


Figure 13.20 - Color Palette Theme in Global Aliasing Provider

Double-click on the Color Palette alias. This opens the Color Palette alias properties in the right-side window of the Workbench, as shown below.

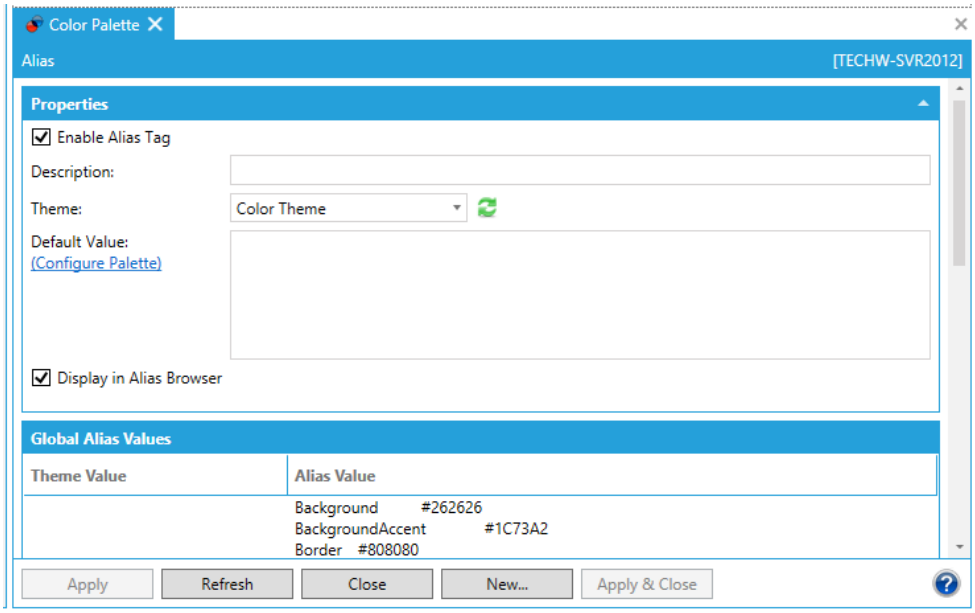


Figure 13.21 - Color Palette Alias

Within the Color Palette Alias properties, you can provide a Description, select the associated Theme (from the pulldown menu) and provide a Default Value (by entering one within the text entry field or by clicking on the ellipsis button [...] and selecting one via the Data Browser.

You can also decide if this alias should be displayed in the Alias Browser and if you wish to use a separate Display Name for this alias within the Alias Browser (if you click this checkbox, you can enter a different Display Name in the text entry field or click on the ellipsis [...] to select one through the Data Browser).

Notice that our 'Dark Theme' and 'Light Theme' Theme Values are listed in the Alias Values section of the Alias properties. In the next step, you'll be able to add Palette Values to these themes.

13.8.6 Adding Palette Values

Within the **Alias Values** section of the Color Palette alias properties, you can click within the **Palette Value** column's cell for your selected Theme Value. This enables an ellipsis button [...] to the right side of the column's cell. Click on this ellipsis to open the **Configure Color Palette** window, as shown below.

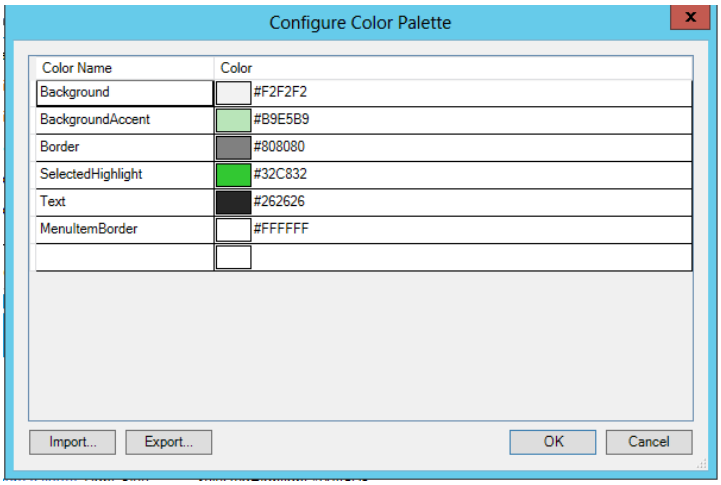


Figure 13.22 – Configure Color Palette Window

In the **Color Palette Configuration** window, you can add a **Color Name** and then select the **Color** through multiple ways. One way is to enter characters into the text entry field that correspond with the following encodings:

- **Eight digit hexadecimal** [e.g. #FF00800]
- **Six digit hexadecimal** [e.g. #008000] - where alpha is assumed to be opaque
- **RGB** [e.g. rgb(0,128,0)

- **ARGB** [e.g. argb(255, 0, 128, 0)]
- **{{tagname}}** [expected to be a 32-bit integer]
- **rgb({{tagnameRed}}, {{tagnameGreen}}, {{tagnameBlue}})**
- **argb({{tagnameAlpha}}, {{tagnameRed}}, {{tagnameGreen}}, {{tagnameBlue}})**
- **Gradient Encoded as XAML**

Another way to select a **Color** is to click within the Color column cell for a specific Color Name. This activates the Color menu window, as shown below.

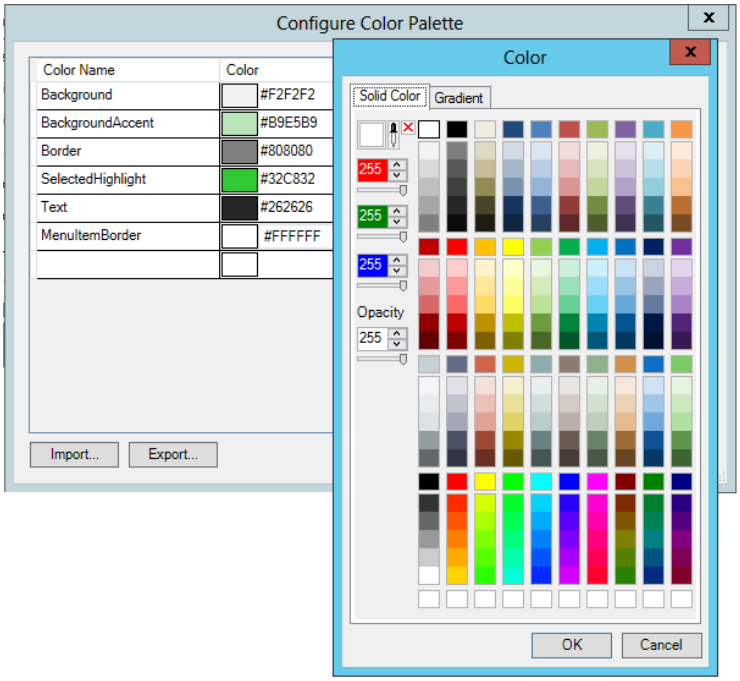


Figure 13.23 - Color Menu Window Activated in Color Palette Configuration Window

Once you have selected a color by entering text or via the Color Picker, you can add another color by hitting **Enter**. You can remove any color by highlighting the row within the window and hitting **Delete**.

Click 'OK' to save the current Color Palette Configuration for your selected Theme Value. The Palette Value column cell for your selected Theme Value will now show your provided values, as shown in the following image.

Global Alias Values

Theme Value	Alias Value
(Configure Palette) Dark_Blue	Background #262626 BackgroundAccent #1C73A2 Border #808080 SelectedHighlight #00AEDB Text #F2F2F2 MenuItemBorder #262626
(Configure Palette) Dark_Green	Background #262626 BackgroundAccent #1CA250 Border #808080 SelectedHighlight #32C832 Text #F2F2F2 MenuItemBorder #262626 Background #262626 BackgroundAccent #C81C1C Border #808080

Figure 13.24 - Filled Palette Value for Selected Theme Value

Click '**Apply & Close**' to enable your settings for your selected Color Palette Alias.

13.8.7 Import/Export

You can also choose to Import or Export these colors. In the Color Palette Configuration window, clicking the **Export** button allows you to save the current configuration within the Color Palette Configuration as a text (.txt) file. Clicking the **Import** button allows you to retrieve any previously saved exported Color Palette Configuration text (.txt) file.

13.8.8 Using a Global Color Palette in GENESIS64 Applications

Once you have configured a Global Color Palette within the Global Aliasing provider in the Workbench, you'll be able to make use of the configured Global Colors and Themes within integrated GENESIS64 applications, such as GraphWorX64.

The example below shows the new Global Color tag added to the Fill dialog, an option within GraphWorX64 while creating/editing a new display. Selecting colors from the Global Color tab is performed in the same way as selecting from the existing Solid Color or Gradient tabs.

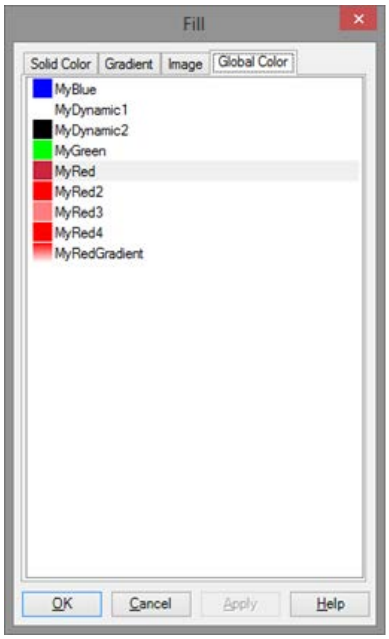


Figure 13.25 - Global Color Tab in the Fill Dialog in GraphWorX64

As previously mentioned in 'About Global Color Palettes', one of the benefits of Global Color Palettes is fast-switchable or 'swappable' palettes of colors. The following example shows how to make a quick color theme change (from Dark Theme to Light Theme) while in GraphWorX64's Configuration Mode.

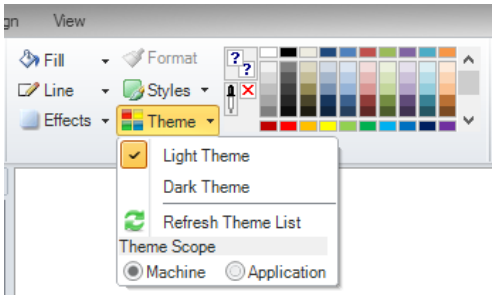


Figure 13.26 - Quick Switch of Color Theme in Configuration Mode

And, of course, the same fast color theme switching can be done in Runtime Mode, as well, as shown below.

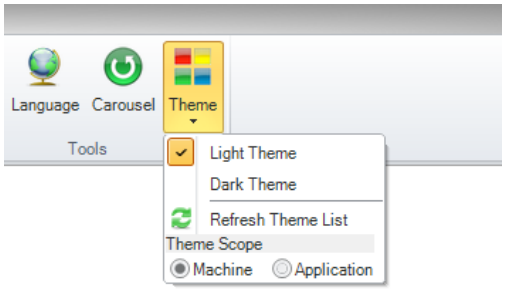


Figure 13.28 - Quick Switch of Color Theme in Runtime Mode

Lab: Global Aliasing L13

13.1. 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 single dynamic display that can view each of the tanks.

Estimated time to complete this lab: **20 minutes**

13.1.1. 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.

13.1.2. Support Files

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

- icTonicsMain.gdfx
- icTonicsTankMonitor.gdfx
- icTonicsTrends.gdfx

13.2. Global Alias Configuration

We will need to define the global alias and the corresponding value that the alias will 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.

13.2.1. Theme Configuration

1. Open the **Workbench** if you closed it after finishing the previous lab.
2. Expand the **Platform Services** node in the Project Explorer and expand the **Global Aliasing** provider.
3. Right-click on **Themes** and select **Add Theme** to create a new theme.
4. In the **New Theme** tab in the lower half of the Project Details window, name the theme **TanksTheme** and then click on **Apply & Close**.
5. Next, right-click on **TanksTheme** and select **Add Theme Value**.
6. Name this theme value **Tank1Theme**.
7. **Check** the box **Default Theme**, then click **Apply & Close**.
8. Right-click on **Tank1Theme** and select **Multiply**.
9. Within the **Multiply Options** dialog box, **enable** the check box of the **Name** field and change the Base Value from Tank1Theme to **Tank{0}Theme**.
10. Change the **Number of copies** field to **2**.
11. Check **Override First Number** and change the value to **2**.
12. And check **Override Numeric Places** and change the value to **1**.
13. The **Multipy Options** dialog box should look like the following figure:

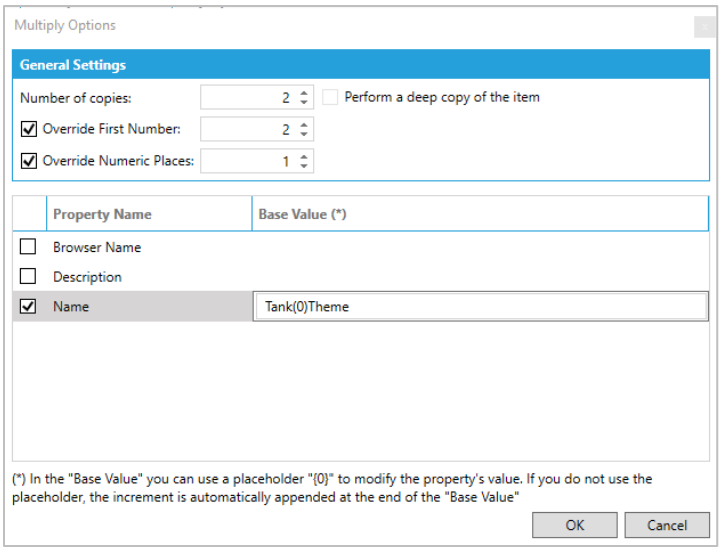


Figure L13.1 – Multiplying a Theme Value

14. Click **OK** to start the multiplication. The theme values named **Tank1Theme**, **Tank2Theme**, and **Tank3Theme** will now exist.

13.2.2. Alias Configuration

1. Right-click on **Aliases** and select **Add Alias** to create a new global alias.

Note: Beware of typos or extra spaces in the alias definition, as any wrong character will prohibit your alias from working.

2. Name this Alias **TanksAlias** and choose **TanksTheme** from the Theme: dropdown list.

3. Notice that once you choose the **Theme**, the **Global Alias Values** section expands to display the TankAlias theme values.

4. Set the values in the Alias Value column as shown in the table below.

Note: Each of the entries in the table below are single words and do not contain spaces.

Theme Value	Alias Value
Tank1Theme	Tank1
Tank2Theme	Tank2
Tank3Theme	Tank3

Table L13.1 – Alias Values for Tanks

5. Click **Apply & Close** to save the configurations.

6. Repeat the steps you've learned to create another alias called **AlarmFilter** with values as indicated in the table below.

Theme Value	Alias Value
Tank1Theme	Tank1 Level
Tank2Theme	Tank2 Level
Tank3Theme	Tank3 Level

Table L13.2 – Alias Values for AlarmFilter

13.3. 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. As done previously, you may need to unlock the template.
2. In the **Properties** tab on the lower right corner go to **Content** → **Text** and click on the ellipsis […] button.
3. When the **Data Browser** opens, select the **Global Aliases** tab.
4. Expand **Global Aliases** tree in the bottom pane and double-click on **TanksAlias**.

   Note: <# and #> are the delimiters for Global Aliases. When used in text boxes and labels, they need to be enclosed in double quotation marks **"<#LikeThis#>"**, because they are strings.
5. Check the box labeled **Treat the result as a Literal String** and select the **Copy Selection to Textbox** button.
6. The Data Browser should resemble the following figure. Click on "**OK**".

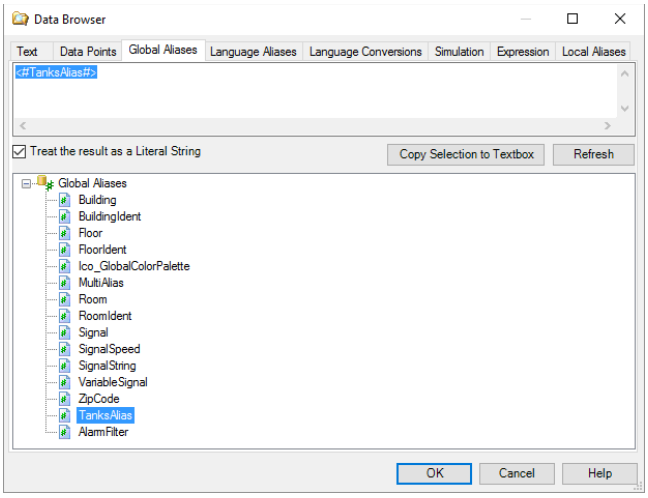


Figure L13.2– Browsing for Global Alias

7. Double-click on the **Trend Viewer Control** to bring up the **Configure your Control** dialog box.

8. Click on your real-time pen under Plot. Find **Tank1** in the **Data Source**, and replace **Tank1** with **<#TanksAlias#>**. Your resulting syntax should be:

   modbus:icTonics.Tanks.USA.Foxboro.<#TanksAlias#>.TankLevel

9. Change the Description to read **"<#TanksAlias#> Level"** (include the double quotes around the entire string).

10. Do the same thing for the other **historical tags** configured in the control.

11. Click on **Close** to close the control. Then save the display.

12. Open the **icTonicsTankMonitor** display in a second instance of GraphWorX64.

    Note: You should always **make a backup** of your work (displays) before using the **Replace** feature.

13. In the **Home** tab of the ribbon, locate the **Text** section and click on the chevron next to the **Find** action and select **Replace**.

14. When the **Replace** dialog box comes up, in the **Find Type** section, enable the **Dynamic Data Source** checkbox.

15. Select the **Current Display** radio button.

16. In the "**Find what:**" field, type in **Tank1** without quotes and in the **Replace With** field type in **<#TanksAlias#>.** Your Replace dialog box should resemble the following figure.

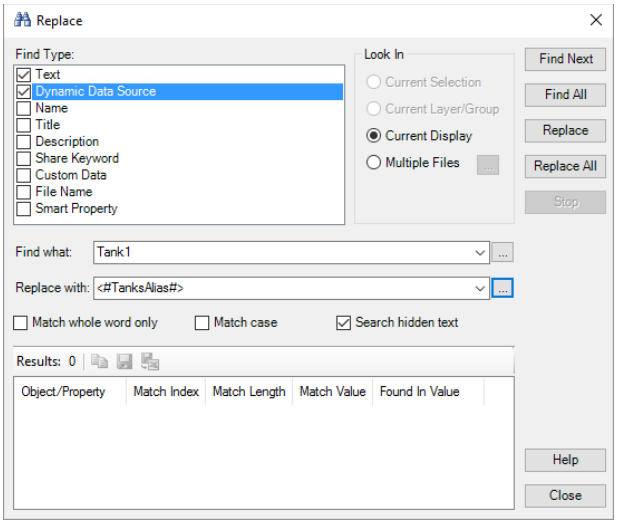


Figure L13.5 – Search and Replace

17. 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.

18. **Close** the dialog box and **Save** the display.

19. Create a button called **Mixer Tanks** the left hand side of the display.

Note: You can label a button by double-clicking on it.

20. In the Dynamic properties of the button, change the Command to **Popup Menu**.

21. In the Dynamic properties of the button, open the **Menu → MenuItems (Collection)** by clicking on the ellipsis […] button.

22. Click **Add** in the **Edit Menu Items** dialog box.

23. In the Dynamic properties of newly-created Menu Item, change the **Dynamic Behavior → Command** to **Set Global Aliases**.

24. Expand the **Command Parameters** section; expand the **GlobalAliases** drop down list and select **Global Aliases**.

Note: Make sure GraphWorX64 Application Mode is in **Advanced Mode** or you won't be able to see all your options.

25. In the **Select Global Themes and Aliases** dialog box, click on the **Themes → Add** button.

26. In the **Themes** dropdown list, select **TanksTheme**.

27. Select **Tank1Theme** in the **Theme Item**s pane.

28. In the **Edit Global Themes** dialog box, under **Scope**, select the **Absolute** radio button and in the dropdown list select **Process.**

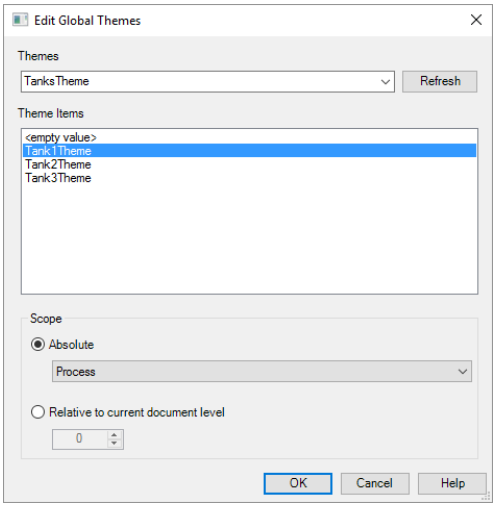


Figure L13.6 – Edit Global Themes Dialog Box

29. Click on **OK** all the way back to the list of menu items.

30. Change the **Menu** → **MenuText** to **Tank 1**.

31. Click **Duplicate** twice to create two new menu items.

32. In the second menu item, expand the **Command Parameters** section and click into the **GlobalAliases** field and edit the text to read **TanksTheme=Tank2Theme/1;** (including the semi-colon at the end).

33. Update the **MenuText** for the second menu item to read **Tank 2**.

34. Repeat the previous two steps for the third menu item, but instead of updating the values to read Tank2, set them to **Tank3.**

35. Add a process point above the tank; set the Foreground color to **Foreground2** and set the Data Source to **"<#TanksAlias#>"** (with the double quotes).

36. Be sure to **Save** your display and enter to **Runtime** mode.

37. Observe how the process point label changes as you click on each menu button item. With Tank3 selected, change the **Level Set Point** to **75%**.

38. Return to **Configuration** mode and select the button you created earlier. In the dynamics pane on the right, right-click on the Pick Action (that genereates the popup menu) and select **Copy**.

39. Switch back to your **icTonicsTrends** display and select the aliased **{{“<#TanksAlias#>”}}** label. Open the dynamics pane and, in the top box, right-click and select **Paste.** This will add an identical dropdown menu, accessed by clicking the label, to choose the desired tank.

40. **Save** your display and enter **Runtime.**

41. Use the pop-up menu to select a tank. Notice your plotted trendlines change depending on which tank is selected. Notice that when you select Tank3, the real-time trend is at the 75% level you set in the other screen.

13.4. Clientside Aliasing

Global aliases can also be configured “on-the-fly” in GraphWorX64, with the same functionality as themes and aliases configured in Workbench. We will configure a simple proof-of-concept screen using simulation tags.

1. **Open** a new instance of GraphWorX64**.**

2. Add a process point. Use the Data Browser to set the DataSource: **My Computer** > **Data Connectivity** > **OPC Data Access** > **Simulations** > **Float** > **Ramp** > **FastL000.** Notice that there are several other types of simulation data under Float, including Sine, Random, Square, Step and Triangle.

3. Change the first value from 1 to **5**; this will increase the duration of the simulation’s cycle, slowing it down enough for us to observe it change.

4. Your DataSource should look like this:

   @sim64:Float.Ramp(5,-50.0,50.0,0).Value

5. Switch into **Runtime** to observe how the simulated data behaves. Then return to **Configuration** mode.

6. In the tag, replace Ramp with the alias syntax: **<#Type#>** (without quote marks). This will have the same effect as creating the corresponding alias and theme in Workbench.

   Note: Quote marks are not necessary here because the entire string, from the @ symbol at the start to the *.Value* at the end, will be treated as a single OPC tag.

7. If you switch into **Runtime**, you will observe that the tag is now bad quality, because the alias is not resolved to create a valid data path. Return to **Configuration** mode.

8. Add a **button** and set the label to read **Ramp**. Set the command to **Set Global Aliases** (under *Local Commands*)**.**

9. Expand the **Command Parameters** and type into the **GlobalAliases** field: **#Type=Ramp/1;**

   - **#Type=** specifies the *name* of the alias to be resolved
   - **Ramp** specifies the *value* that the **#Type** alias will be resolved to
   - **/1** specifies the *scope* of the alias (/0 is machine-level, /1 is process-level, /2 is document-level, and so on)
   - **;** ends the definition

10. Switch into **Runtime;** notice that the tag is still bad quality.

11. Click the **Ramp** button you created and observe how the tag now resolves to a valid path, showing the same live data you saw when the DataSource used an absolute data path instead of an aliased one. Return to **Configuration** mode.

12. Hold down the **CTRL** key and click and drag the **Ramp** button to create a copy of it. Repeat this process until you have 4 buttons.

13. On the second button, change the label to *Sine* and update the GlobalAliases parameter to read **#Type=Sine/1;**

14. Do the same on the third button (changing Ramp to *Random* in both places) and the fourth button (changing Ramp to *Square* in both places).

15. Finally, create a new process point and set the DataSource as **"<#Type#>"** (with the quote marks).

    Note: Quote marks are necessary here because the value will resolve as a string, and not as the live value of a data tag.

16. Switch into **Runtime** mode and click the various buttons to observe the changing behavior of the data value. Your second process point will indicate the current value of the alias. You can also hover over the first process point to view the current tag path of the process point, which will depend on which value the alias is currently set to.

    Note: The *Random* and *Square* tags will change value infrequently (every 8 seconds for the *Random* tag, every 4 for the *Square*).

All of the above clientside alias configuration can also be done using the Set Global Themes and Aliases GUI in GraphWorX64. In the next steps, we will create a second set of aliases to adjust the period of our simulated tags in Runtime.

17. Select the process point that contains that simulated data tag. Change the *DataSource*, replacing the *8* inside the parentheses with **<#Period#>.**

18. Your DataSource should look like this:

```
@sim64:Float.<#Type#>(<#Period#>,-50.0,50.0,0).Value
```

19. Add a new **button** to your display. **Label** it *8 seconds* and set the **Command** to **Set Global Aliases** (under *Local Commands*).

20. Expand **Command Parameters.** In the *GlobalAliases* field, select the dropdown menu and choose *Global Aliases…*

21. In the pop-up, select **Add…** under **Aliases.**

22. In the *Alias Name* field, input *Period* (with no spaces or punctuation marks).

23. In the *Value* field, input *8* (with no spaces or punctuation marks).

24. Set the scope as *Process* by clicking **Absolute** and choosing **Process** from the drop-down menu.

25. Click **OK** and then **OK** again to apply your changes and close out the pop-up GUI.

Note: You can use the pop-up GUI to set up multiple aliases to be set by this single command. For example, we could use a single button to set the aliases for data type (Sine, Random, Ramp, etc.) *and* period with a single click.

26. Notice how the *GlobalAliases* field now reads **#Period=8/1;** using the same syntax as you input manually earlier.

27. Hold down the **CTRL** key and click and drag the **8 seconds** button to create a copy of it. Repeat this process until you have 3 buttons.

28. On the second button, change the label to *16 seconds* and update the GlobalAliases parameter to read **#Period=16/1;** either by manually editing the text or by using the drop-down menu to open the pop-up GUI, selecting the row with the Period alias and clicking **Edit…**

29. On the third button, change the label to *32 seconds* and update the GlobalAliases parameter to read **#Period=32/1;** either by manually editing the text or by using the drop-down menu to open the pop-up GUI, selecting the row with the Period alias and clicking **Edit…**

30. Finally, create a new process point and set the DataSource as **"<#Period#> seconds"** (with the quote marks).

31. Switch into **Runtime** mode.

32. Click the **Sine** button and the **8 seconds** button to resolve each alias, creating a valid data path for the Process Point. Until you resolve both, the tag will likely have bad data quality.

33. Click the various buttons to observe the changing behavior of the data value. Your second process point will indicate the current value of the *Type* alias, while your third will indicate the current value of the *Period* alias. You can also hover over the first process point to view the current tag path of the process point, which will depend on which value the aliases are currently set to.

This concludes this lab.

Security 14

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.

14.1 Objectives

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.

14.2 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.

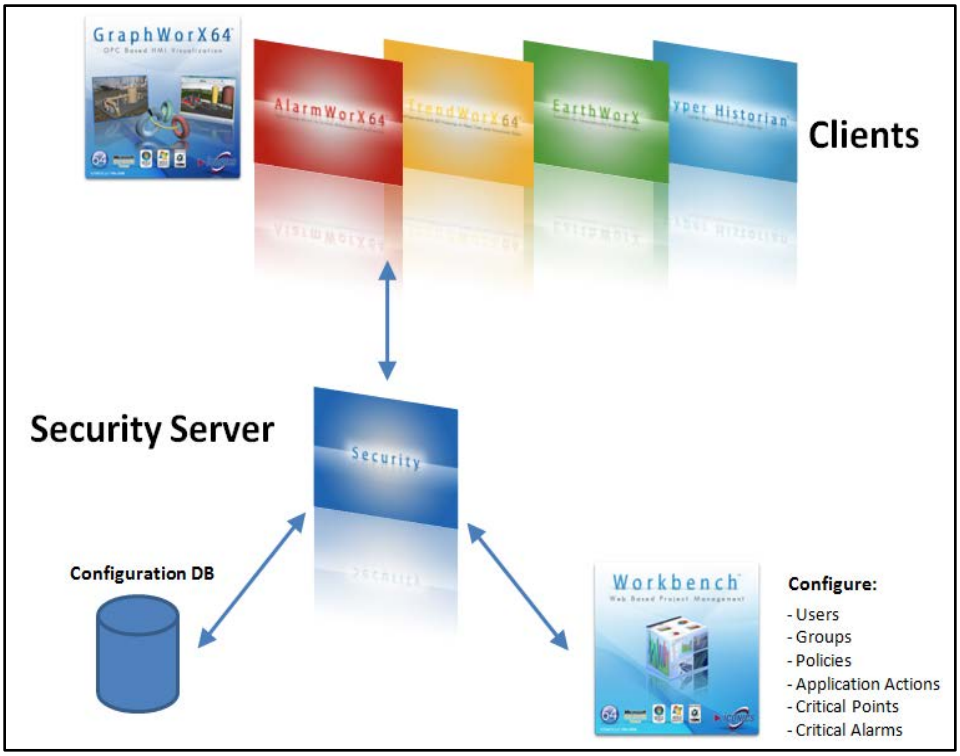


Figure 14.1 - Security Server and Clients Architecture

NOTE: Security assumes the least restrictive permissions (security is disabled out of the box).

Security settings for the server are set from the **Security provider** in Workbench. You can click on the Security Server icon in the Project Explorer of the GENESIS64 Workbench, as shown in Figure 14.2.

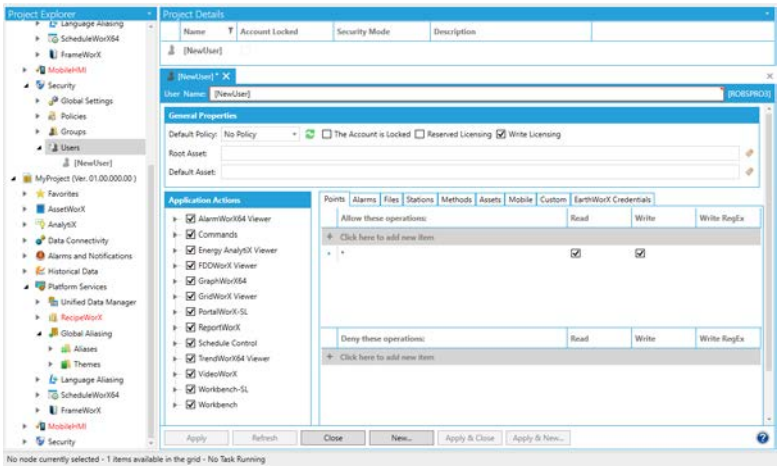


Figure 14.2 - The Security Server Configurator

14.3 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 14.3, 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.

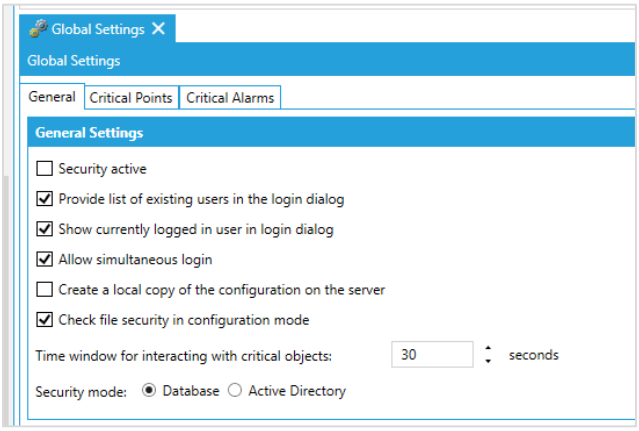


Figure 14.3 - General Settings

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 14.4.

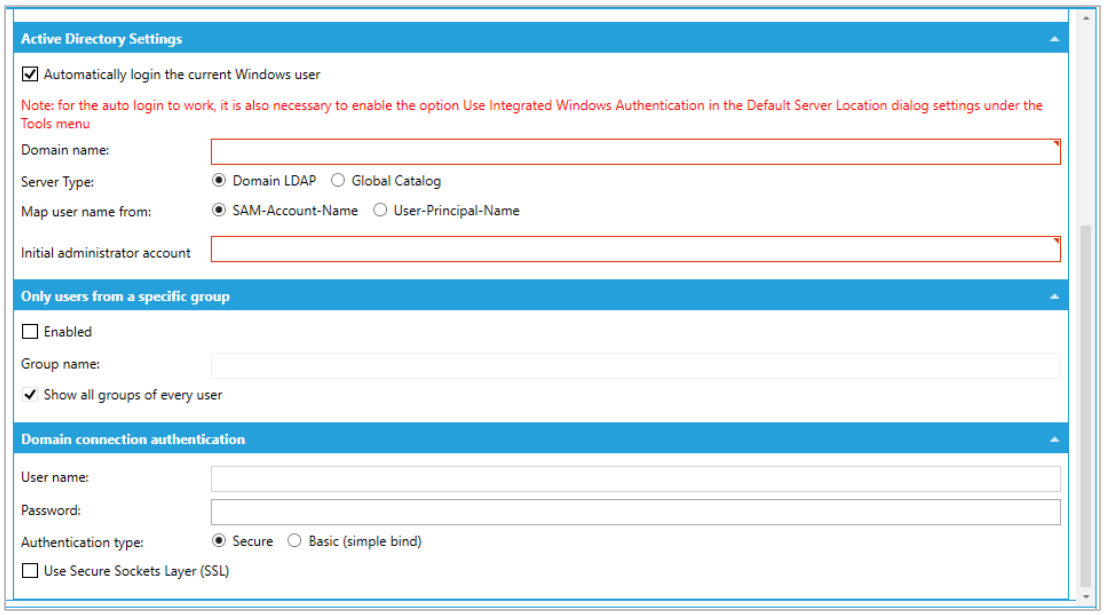


Figure 14.4 - Active Directory Settings Dialog box

- **Domain name**: the name of the domain that you wish to synchronize with
- **Server Type**: Domain LDAP or Global Catalog, as shown in Figure 14.4.
- **Map user name from**:
  - SAM-Account-Name: is the name of the Windows account in the domain.
  - User-Principle-Name: 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.
- **Initial administrator account:** Login credentials for the domain. You should enter credentials that can query the Active Directory here.

14.3.1 *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 15.6.2) in the user and group properties dialog boxes. This lets you specify multiple tags without listing them individually.

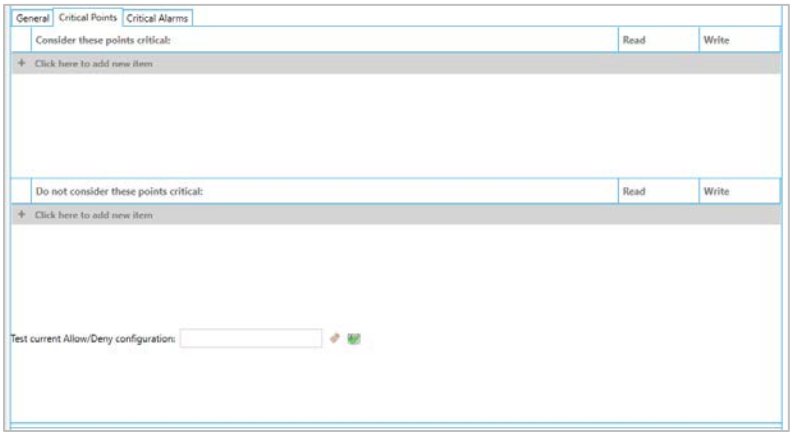


Figure 14.5 - Critical Points Settings

Note: If neither the 'Read' nor the 'Write' checkboxes are checked, then the request will be denied.

14.3.2 *Critical Alarms*

In the **Critical Alarms** tab of Global Settings, shown in Figure 14.6, 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 15.6.3) in the user and group properties dialog boxes. This allows you to specify multiple alarms without listing them individually.

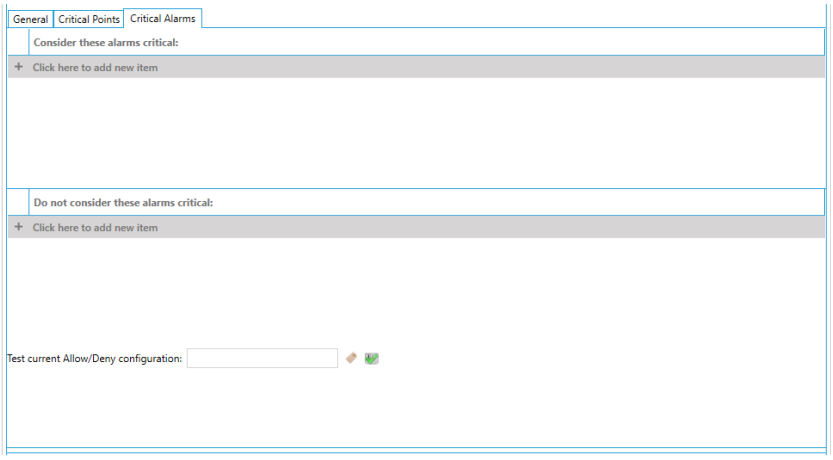


Figure 14.6 - Critical Alarms Settings

14.4 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.

14.4.1 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.

14.4.2 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 in the following image. You can use the Properties tab to change account policies or, by checking the **Account locked** checkbox, quickly disable a user account.

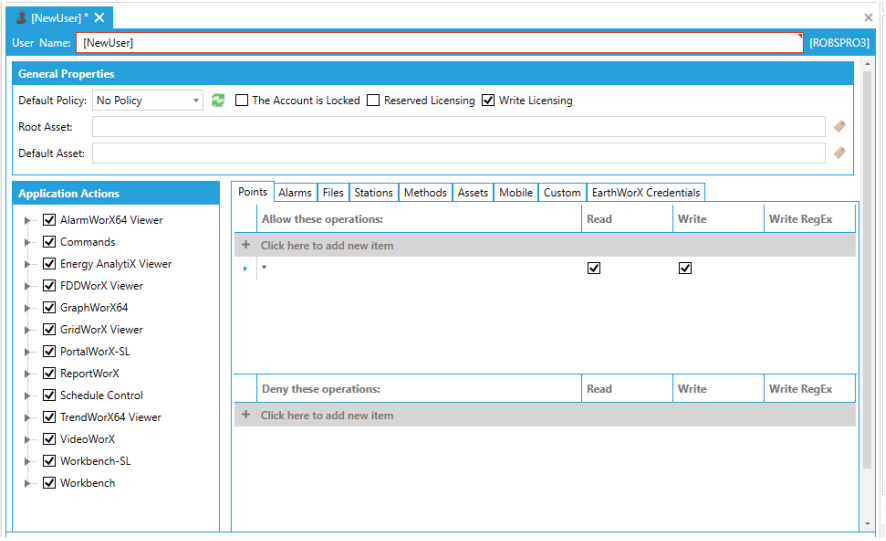


Figure 14.7 - User Properties Tab

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.

14.4.3 *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 14.8 is the Membership menu item for a group; a similar menu appears for a user.

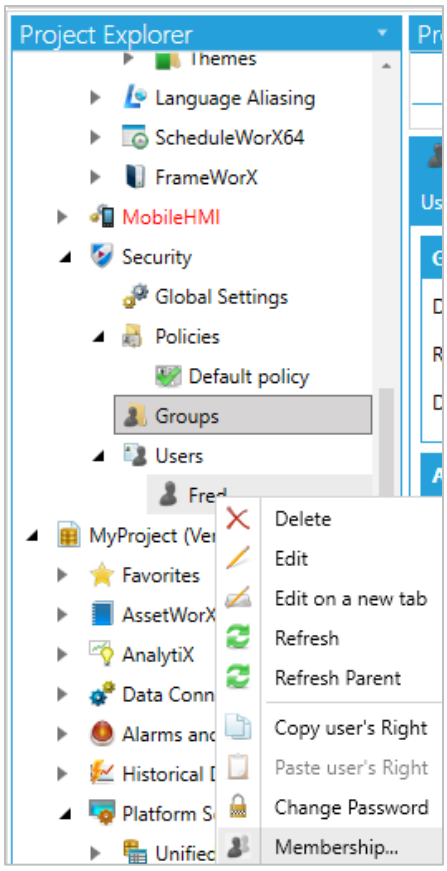


Figure 14.8 - Membership Command

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 14.9.

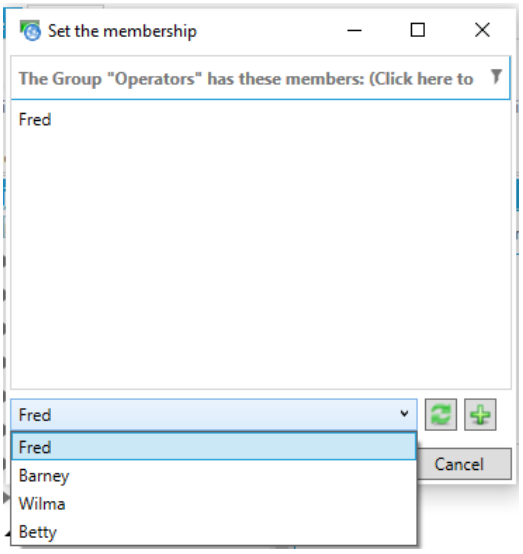


Figure 14.9 - Adding Users to a Group

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.

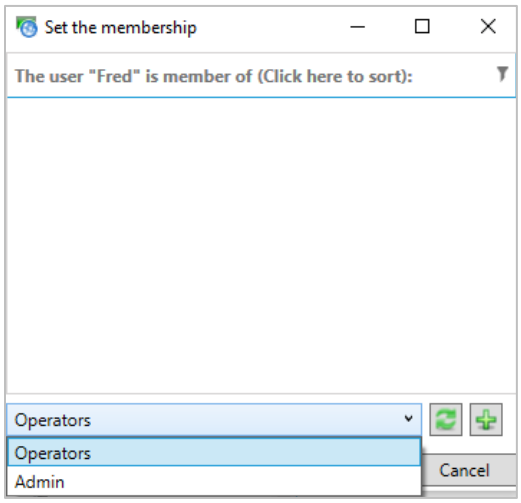


Figure 14.10 - Selecting Groups for a User

14.4.3.1 *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.

14.4.4 *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.

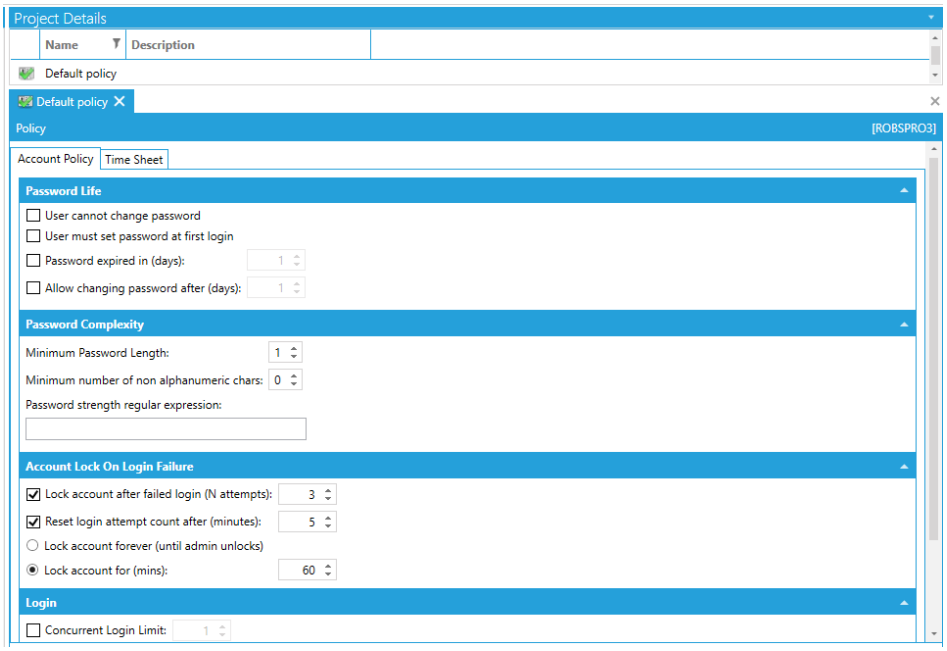


Figure 14.11 - Account Policy Settings

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.

Security Option	Purpose
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.
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 one digit and one alphabetic characters, and must not contain a special character.

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.

14.4.4.1 *Time-Based Restrictions*

The Time Sheet tab for account policies, shown in Figure 14.12, 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 14.12 shows a configuration that allows access from 6AM to 1PM 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.

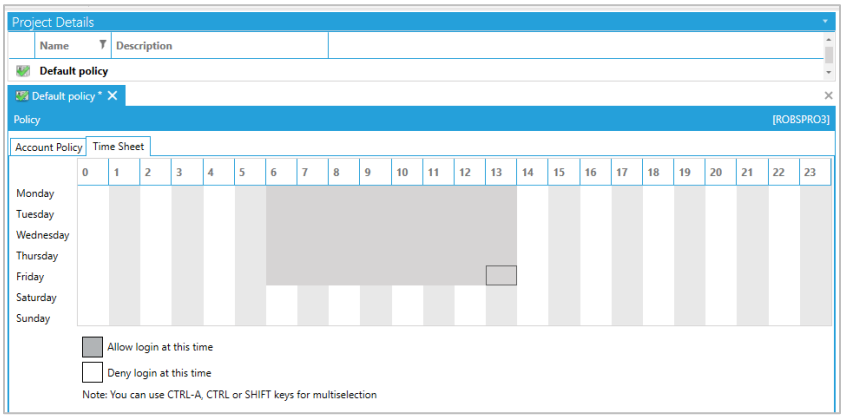


Figure 14.12 - Time Sheet Settings

14.5 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 14.13, 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.

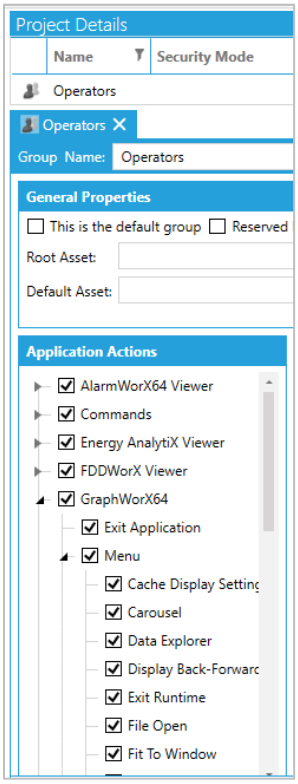


Figure 14.13 - Security Role Application Actions

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.

14.5.1 *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 14.14, to configure which OPC output points are allowed to be read from and written to by the user or group.

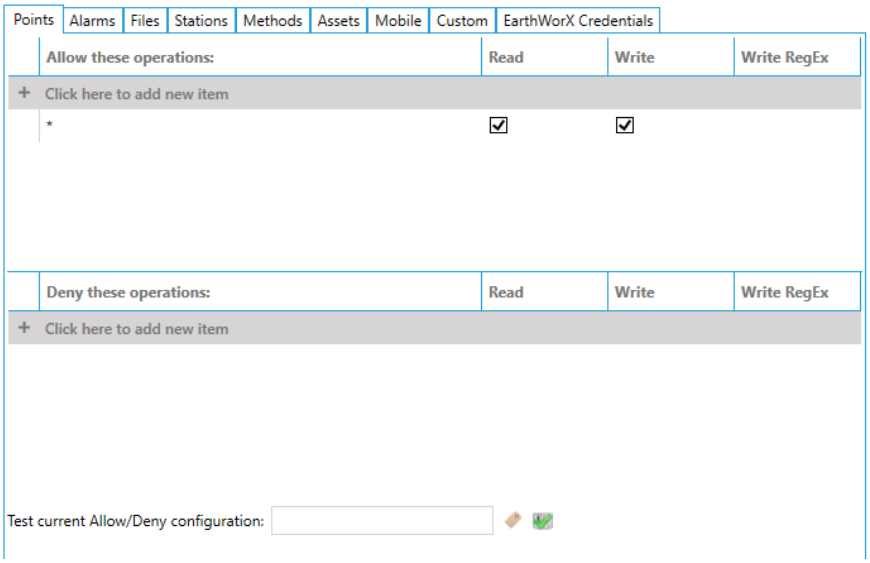


Figure 14.14 - Security Role Points Tab

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 the sections covering "Permissions" and "Wildcards and Performance Optimization". Information in these sections will help you form a strategy for setting up the Allow and Deny lists.

14.5.2 *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 14.15, is used to allow or deny user acknowledgement of alarms during runtime.

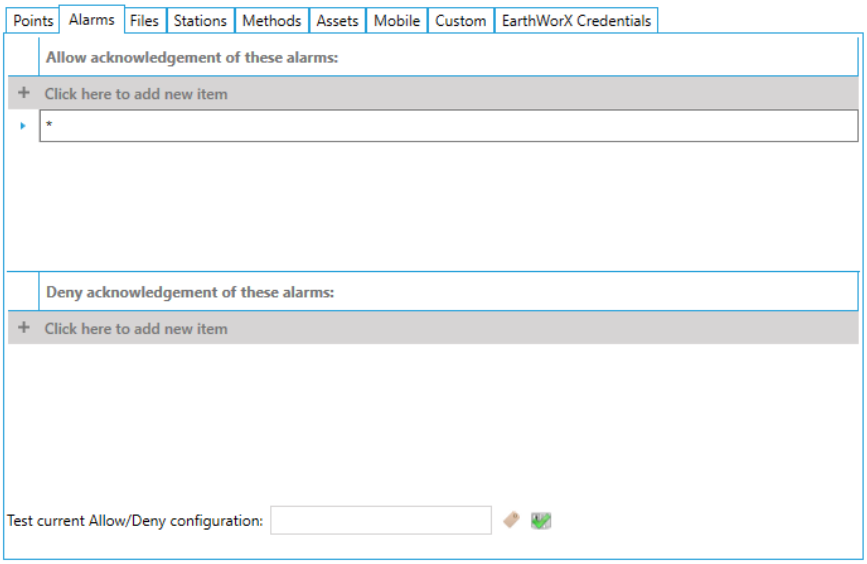


Figure 14.15 - Security Role Alarms Tab

The runtime processing and wildcard pattern matching for the Points tab apply here as well.

14.5.3 *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 14.16, 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.

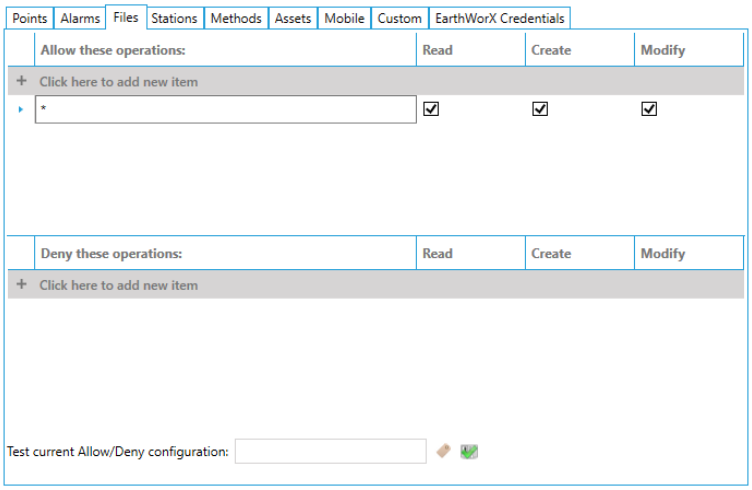


Figure 14.16 - Security Role Files Tab

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.

.

14.5.4 *Stations*

Use the **Stations** tab, shown in Figure 14.17, 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.

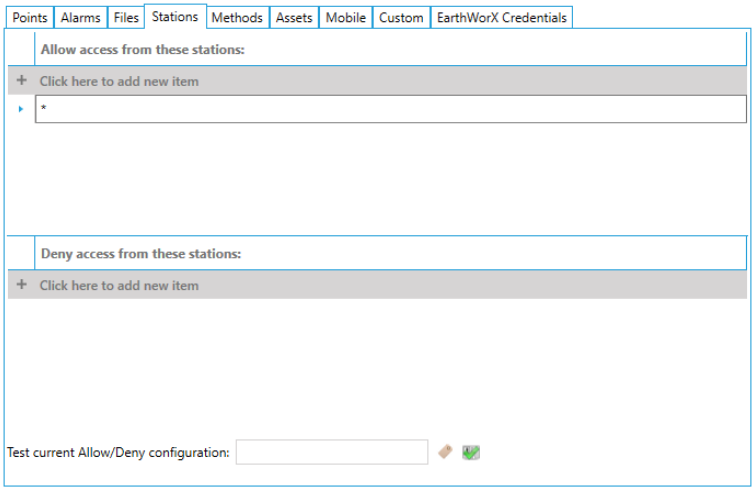


Figure 14.17 - Security Roles Stations Tab

To add a station to either list, click on the "**+ Click here to add new item**." This opens the field and allows selection of the Data Browser dialog box, as shown in Figure 14.18. 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".

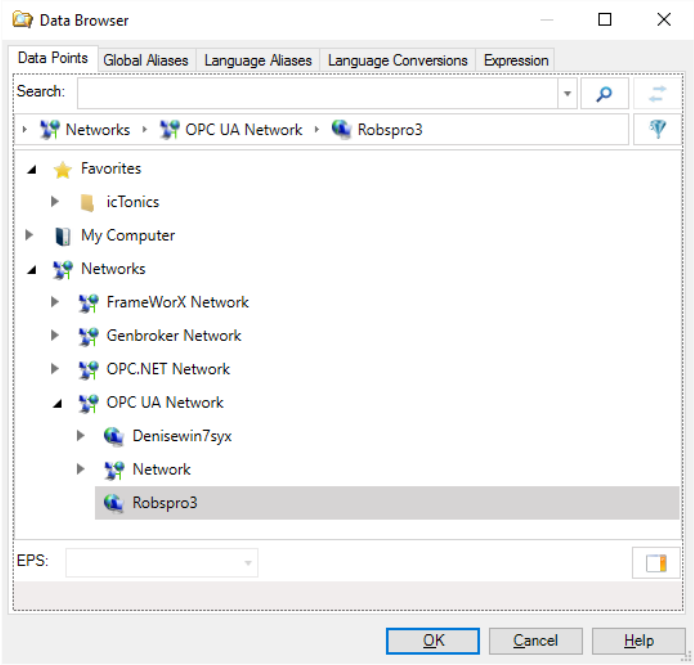


Figure 14.18 - Select a Station Dialog Box

14.5.5 *Methods*

Use the **Methods** tab to grant or restrict access to executing specific methods from being run by specific users or groups.

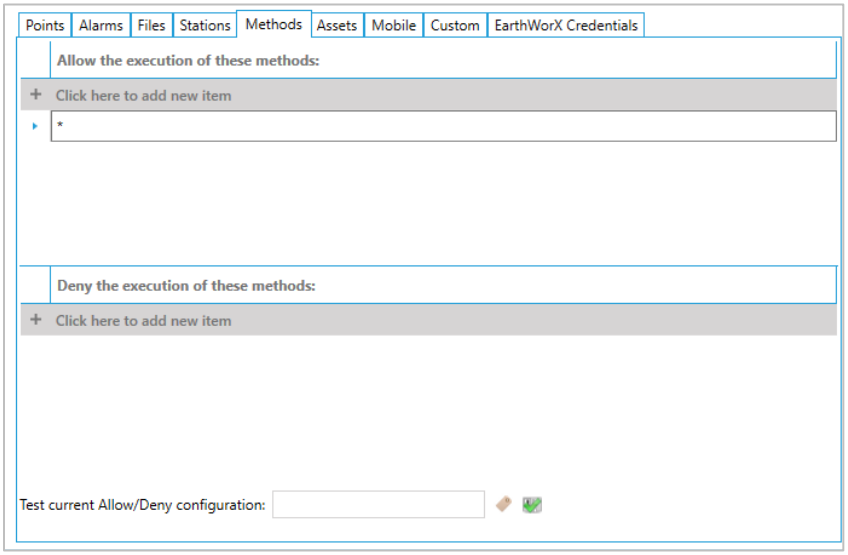


Figure 14.19 - Method Security

14.5.6 *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.

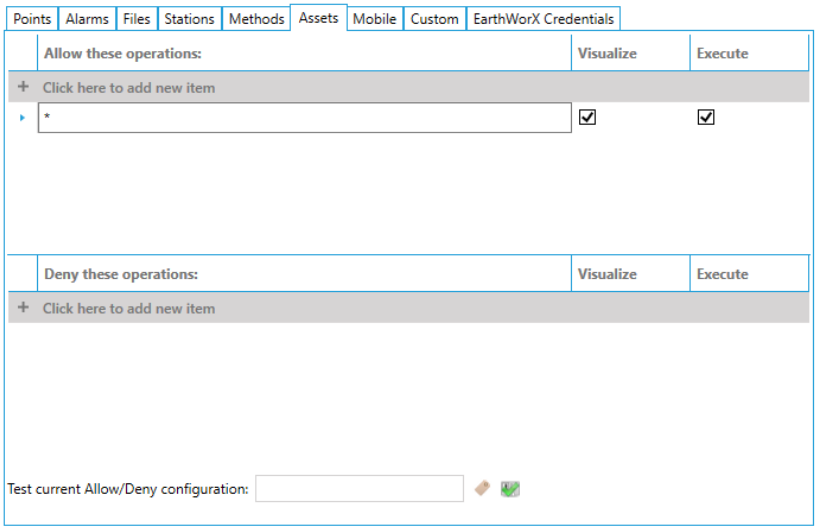


Figure 14.20 - Asset Security

14.5.7 *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.

14.5.8 *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

14.6 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.

14.6.1 *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 14.21. 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.

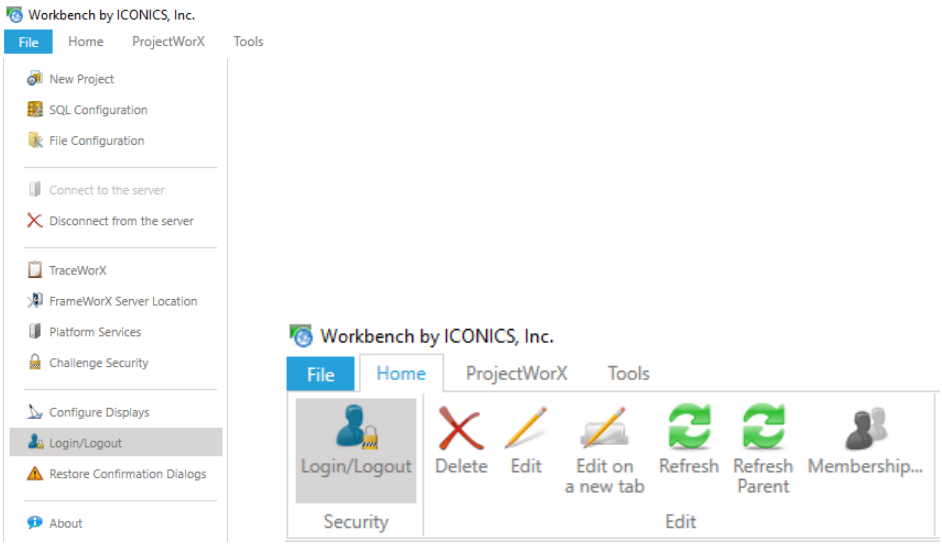


Figure 14.21 - Log In from the File Menu (Left) or the Home Ribbon (Right) when Available

Once the Security Login is open, as shown in Figure 14.22, 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.

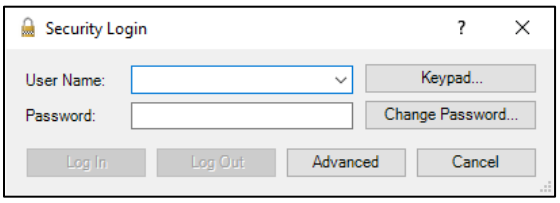


Figure 14.22 - The Security Login Dialog

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.

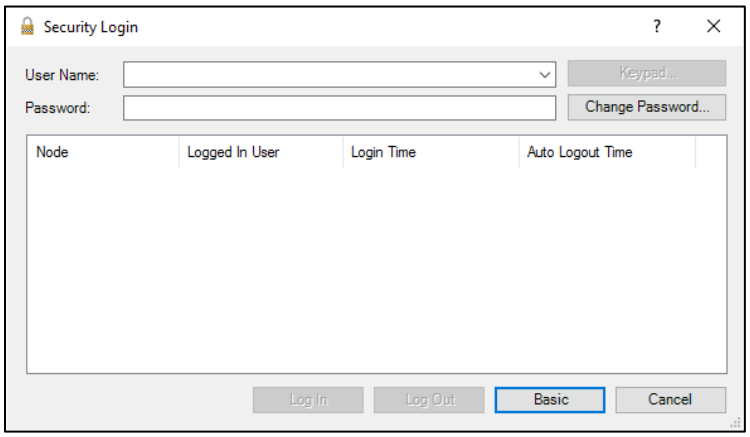


Figure 14.23 - The Security Login dialog in Advanced Mode

14.6.2 *Multiple Logins*

As you may have noticed in Figure 14.23, 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.

14.7 Advanced Settings

The GENESIS64 Security System also includes some advanced features that you may find useful when working with GraphWorX64 and other client applications.

14.7.1 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 14.24, a layer called SecuredLayer contains critical information that needs to be secured. Non-critical information resides in a separate layer.

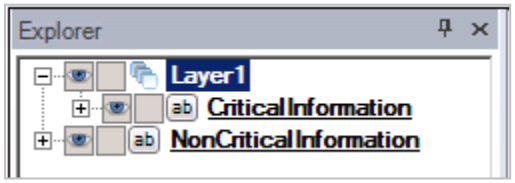


Figure 14.24 - Creating a secured layer

For the SecuredLayer to follow security rules, its **HideWhenSecurityDenied** property must be set to True in its Properties tab, shown in Figure 14.25.

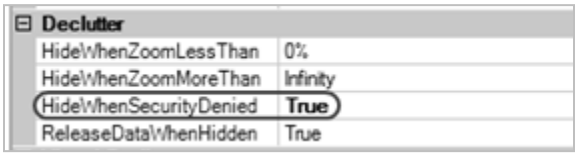


Figure 14.25 - Setting HideWhenSecurityDenied Property

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.

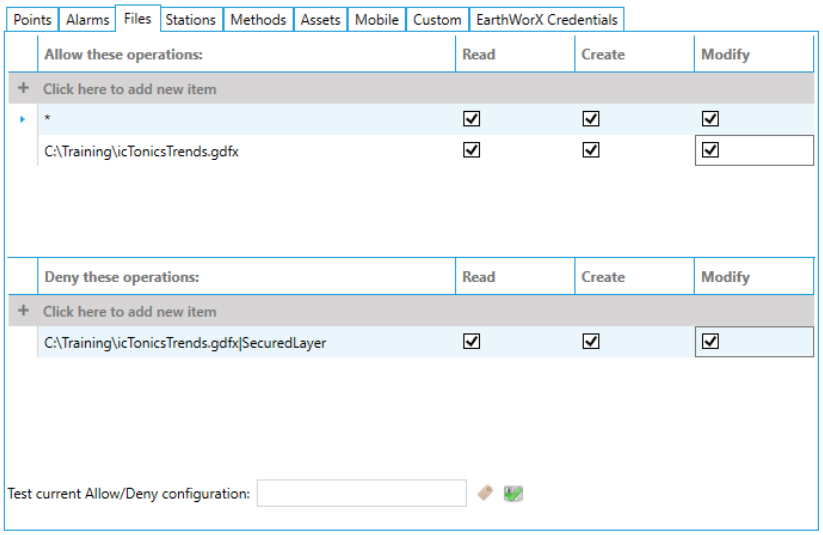


Figure 14.26 - Denying a User Access to a Secured Layer

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.

14.7.2 *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 14.27, which shows the point's value in the top line and the security value (that is, the user's permissions) in the bottom line.

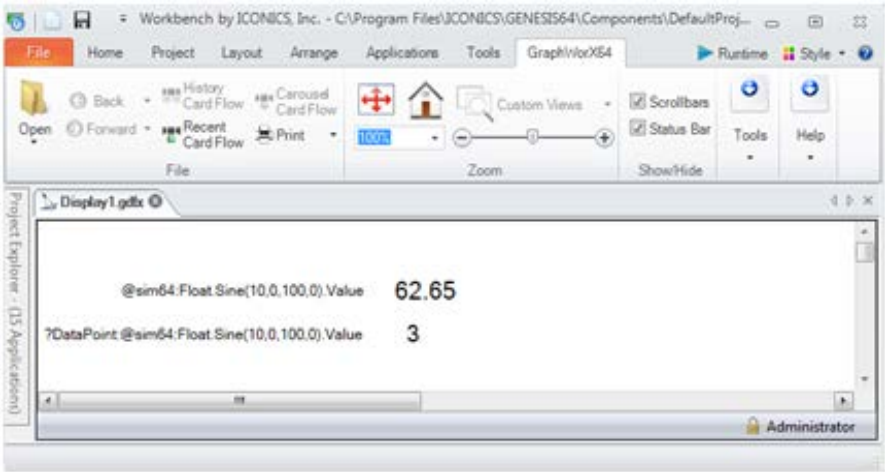


Figure 14.27 - Accessing Security Information

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.

14.7.3 *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 14.28, 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 14-26, you will see this means that this sets the user's security permissions value for the CriticalObject point to **0** (zero).

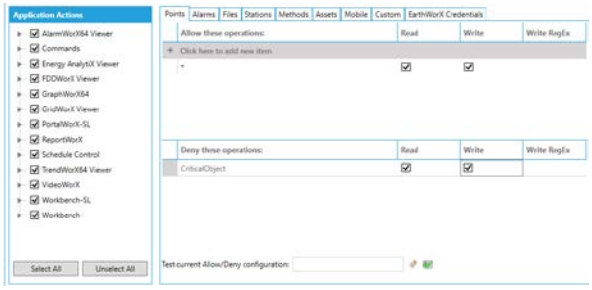


Figure 14.28 - Preventing the User from Reading and Writing to CriticalObject

Next, configure a GraphWorX64 display that contains the 3D objects you will show or hide depending on the user's permissions.

In Figure 14.29, 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.

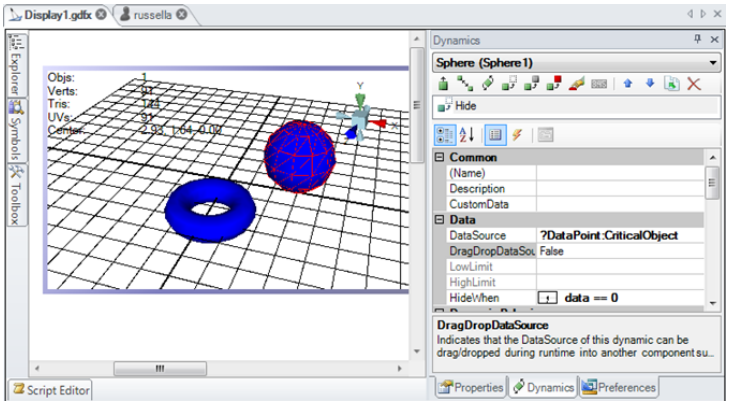


Figure 14.29 - Using a Hide Dynamic Combined with Security Settings

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 14.30.

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.

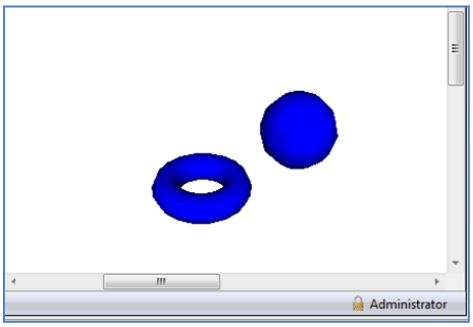


Figure 14.30 - 3D View Seen Using an Administrator Account

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 14.31.

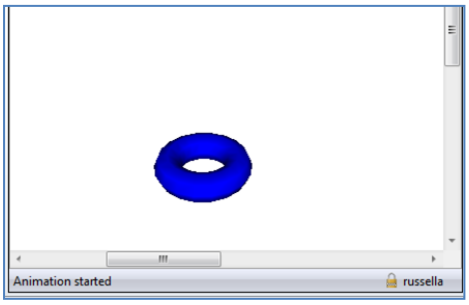


Figure 14.31 - 3D View Seen Using User Account russella

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.

14.7.4 *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 14.32. Notice that this is a 2D feature that applies to 2D displays.

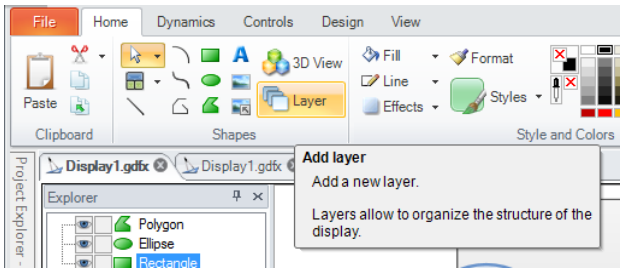


Figure 14.32 - How to Add a Layer

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 14.33 shows Layer4 (which is selected) being moved into a new position below Layer2 in the Explorer.

Figure 14.33 - How to Move a Layer Before (Left) and After (Right)

You can also nest layers using sub-layers, as shown in Figure 14.34. 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.

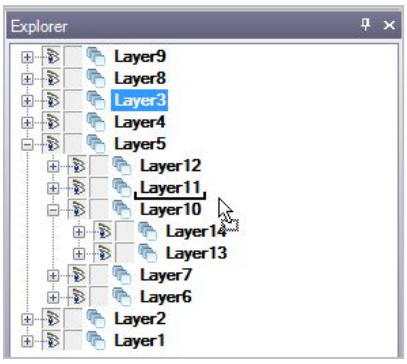


Figure 14.34 - Layers and Sub-layers in the Object Explorer

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 14.33, 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.

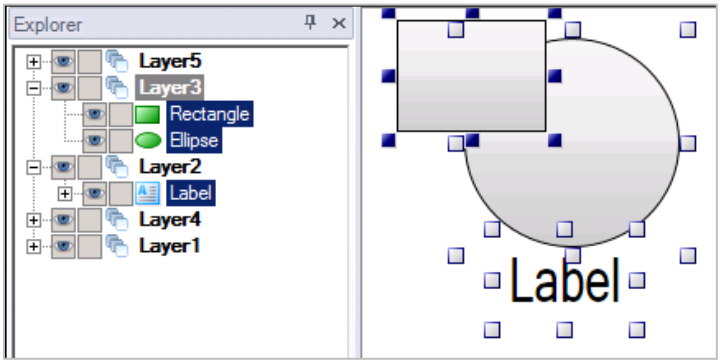


Figure 14.35 – Objects in Layers

Figure 14.36 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.

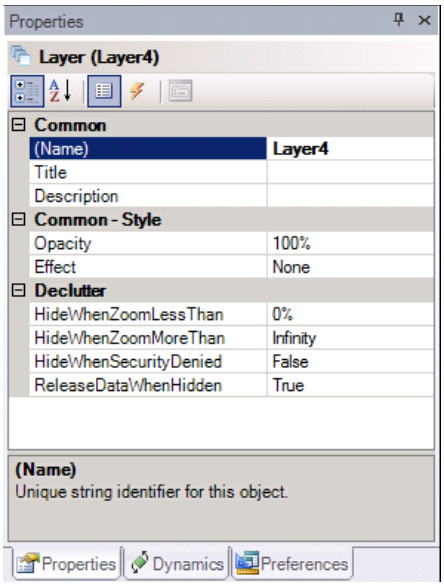


Figure 14.36 - Properties of a Layer

14.8 FrameWorX64 Server

FrameWorX64 is the core GENESIS64 product that, among other things, handles aliasing, security, and data exchange between GENESIS64 applications. There is just one communication means between the client applications and FrameWorX Server. By default, the FrameWorX Server is local on each GENESIS64 node. You must configure the default FrameWorX Server using the **Platform Services Configuration** option.

The **Passwords** tab, **Allowed Clients** tab and **Unsecured Clients** tab within the **Platform Services Configuration** tool are relevant to security considerations.

Passwords Tab

The **Passwords** tab allows users to specify credentials for different modules to connect to various servers. Currently, FrameWorX Servers and OPC XML DA servers are supported. Use * for the credentials to be used for all target addresses. These credentials are used for the modules running on the local computer.

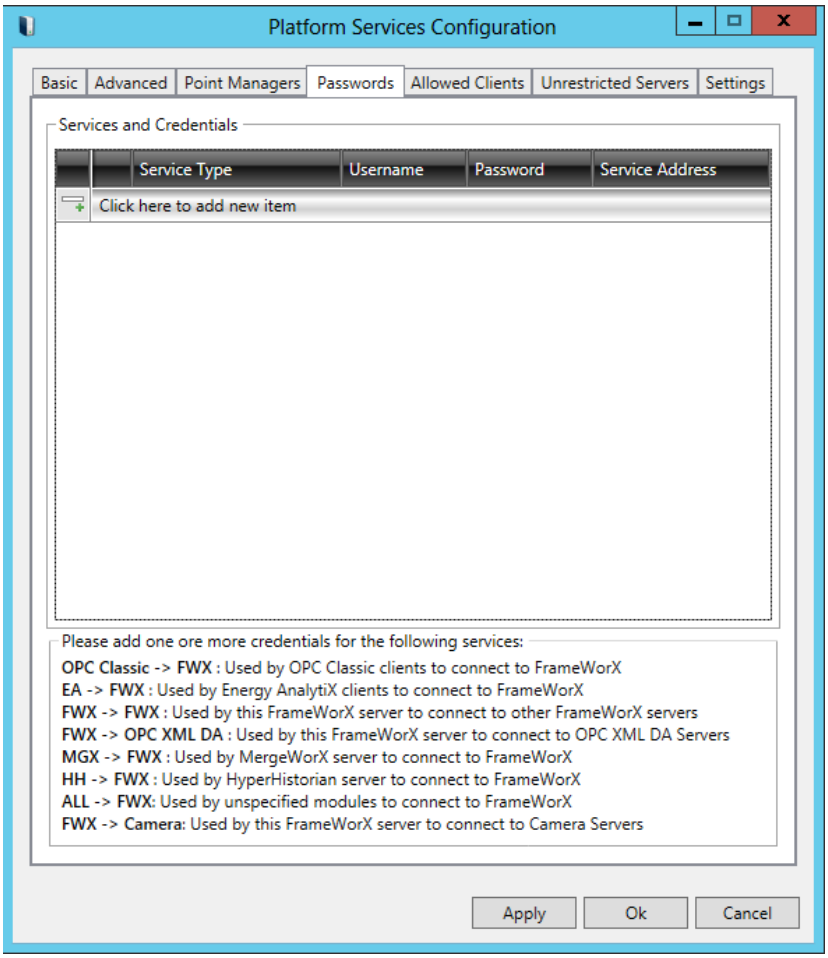


Figure 14.37 - Passwords Tab in the Platform Services Configuration Window

Allowed Clients Tab

The **Allowed Clients** tab allows users to define which client computers are allowed to connect to the FrameWorX Server. Clients from other nodes will not be allowed. The client computer must match both lists: the Allowed IP Address Ranges (IPv4 and IPv6) and the Allowed Computer Names. Use * for any name. Use IPv4 and IPv6 addresses (Windows uses IPv6 internally). All clients are allowed by default.

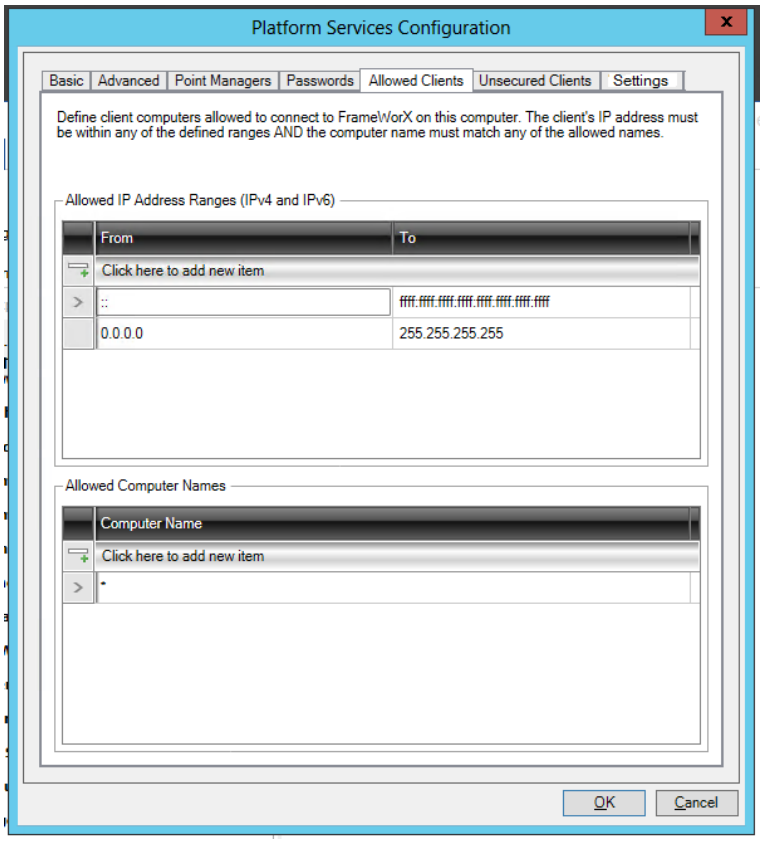


Figure 14.38 - Allowed Clients Tab in the Platform Services Configuration Window

Unsecured Clients

The **Unsecured Clients** tab allows users to define which FrameWorX Servers (acting as clients) can connect to the FrameWorX Server with no security restrictions. The client computer must match both lists: the Unsecured IP Address Ranges (IPv4 and IPv6) and the Unsecured Computer Names. Use * for any name. Use IPv4 and IPv6 addresses (Windows uses IPv6 internally). Unsecured access is only allowed for other FrameWorX Servers acting as clients. There are no unsecured clients by default.

This setting is related to setting a user name/password for FrameWorX Server to FrameWorX Server communication. When a FrameWorX Server connects to another FrameWorX Server, it is treated like other client (unless configured), i.e. it gets the permissions of the default user.

There are two ways to get better permissions:

1. Set up a user name/password for FrameWorX Server to FrameWorX Server communication on the ‘client’ FrameWorX Server. The ‘client’ FrameWorX Server then logs in using that user name.

2. Set up the ‘client’ FrameWorX Server as an unsecured client on the ‘server’ FrameWorX Server. The ‘server’ FrameWorX Server then will not check the security for the ‘client’ FrameWorX Server.

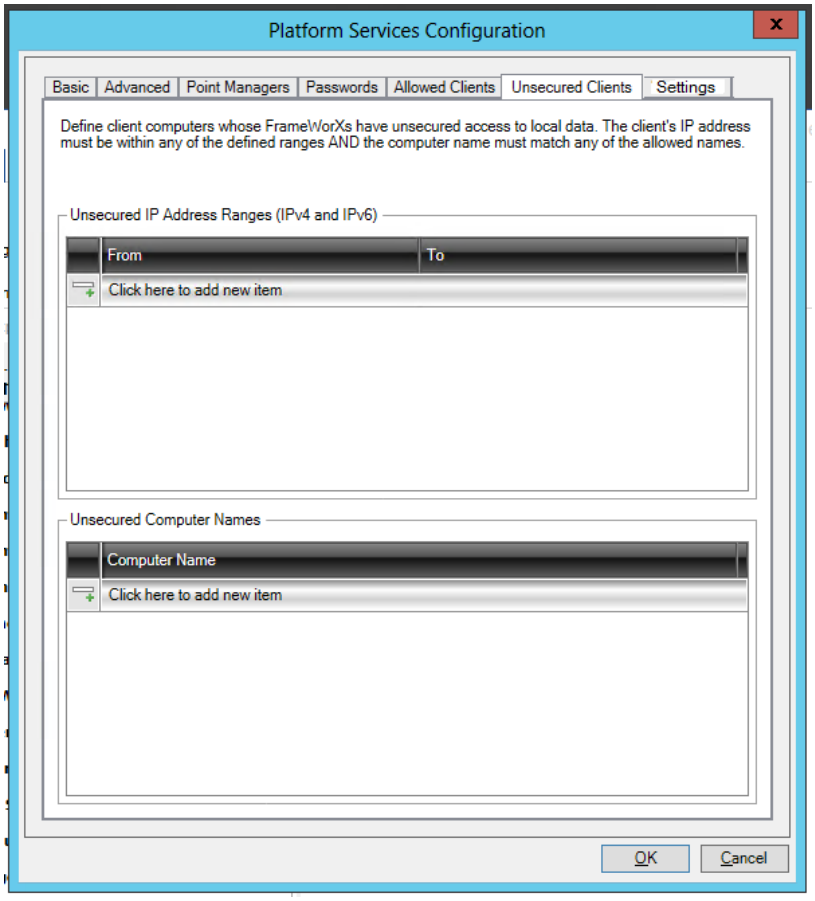


Figure 14.39 - Unsecured Clients Tab in the Platform Services Configuration Window

Lab: Security L14

14.1 Lab Overview

Many companies require security to prevent outsiders from looking at and controlling their displays as well as to 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. Note that users will assume the least restrictive permissions. Also note that Deny overrides Allow.

Estimated time to complete this lab: **30 minutes**

14.1.1 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.
- FrameWorX64
    - Communication Outline
    - Node Discovery
    - Testing
- Platform Services Configuration

14.1.2 Support Files

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

- **icTonicsHome.gdfx**
- **icTonicsTemplate.tdfx**

14.2 Security Users

On the icTonics production floor, there are operators and administrators. The administrators have full permissions. 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, expand **Security**.
3. Next, we will add a User. Right-click on **Users** and select **Add User**.
4. Name this user **Administrator** and click **Apply**.
5. Set the password to **admin**.

   NOTE: When you apply these changes and the user is added, you will see that your configuration is now grayed out. This is because security immediately applies once you add a user.

6. Click **OK**.
7. Confirm that **Security** > **Global Settings** > **Security Active** is enabled.
8. Now you are going to **log in** as **Administrator**. If the login dialog does not appear, you can access it from Workbench using the **File** tab and select **Login\Logout** to open **Security Login** popup window.
9. From within the **Security Login** dialog, enter the user name and password that you just created for the Administrator.

   NOTE: Passwords are case sensitive. User names are not.

10. Now, add another user named **Operator** and set **operator** as the password.
11. We will continue by setting some Application Actions for the user Operator.
12. In the **Application Actions** frame, uncheck **Workbench**. This will restrict the operator from entering the Workbench to change configuration settings.
13. Expand the **GraphWorX64** section, and uncheck **Exit Application** so that the operator cannot close a Runtime GraphWorX64 display.
14. Expand the **Menu** subtree and uncheck the **Exit Runtime** option as well.

15. Select the **Files** tab. In the **Deny These Operations** section, enter ***.*|SecuredLayer**. Remember to include the pipe character, "|", before the layer's name. Placing checkmarks in the Read, Create, and Modify checkboxes results in the user called Operator unable to see the layer called SecuredLayer.

16. **Check** the **Read**, **Create** and **Modify** checkboxes. This will disable (deny) these attributes.

17. Click on **Apply** to save the configuration.

18. Confirm that **Security > Global Settings > Security Active** is enabled.

19. Open your **icTonicsHome** display. Verify that the **Tank Monitor** button is under the **SecuredLayer**. Make sure that you do not add the Login button to the secured layer, though. **Save** the display.

20. If you are not already logged out, log out now by clicking on the Login/Logout Security button in the Workbench ribbon or by going to **File → Login/Logout**, and in the **Server Login** dialog box that appears click on the **Logout** button.

21. You can confirm that you are logged out by looking at the status in the bottom-right corner of Workbench or GraphWorX64. (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 you see a user ID there, log out again.

22. While you are not logged in, return to GraphWorX64 and put the **icTonicsHome** display into **Runtime**. You should see an **Access Denied** screen as in Figure L14.1.

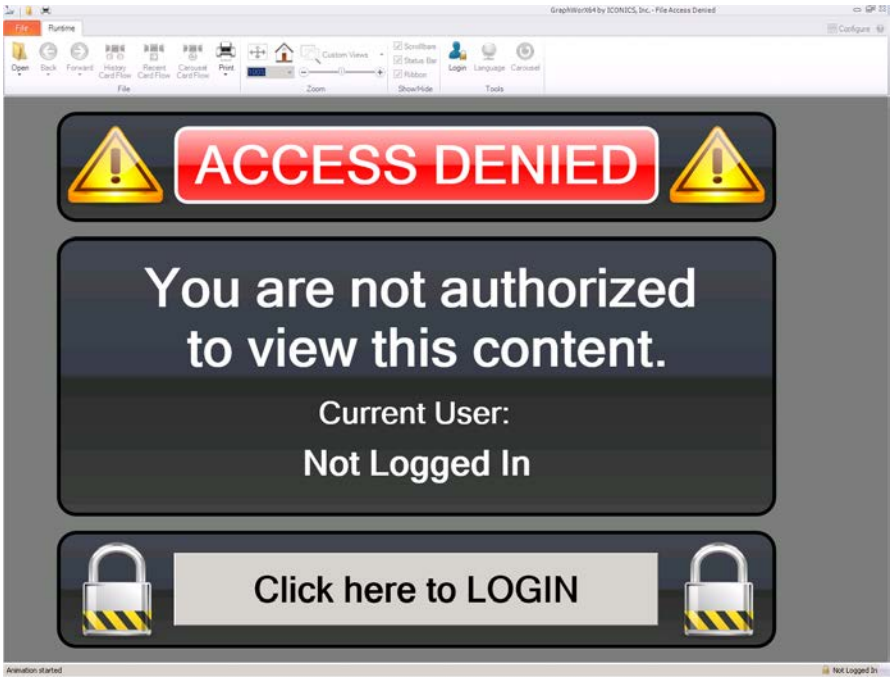


Figure L14.1 - Access Denied

23. Click on this screen's **Click here to LOGON** button and log in as **Operator**. You should not see the **Tank Monitor** button.

24. Notice that when you are logged in as Operator, you cannot exit the application because the Configure button is grayed-out.

25. Launch the **Security Login** dialog and login as **Administrator**. You should now see the Tank Monitor button and you should also be able to go back into Configuration mode.

14.3 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 **Security** Server. From within the expanded Project Explorer tree, right-click on **Groups** and select **Add Group**.

2. **Name** this group **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 Read, Create and Modify permissions that you denied to user Operator.)

4. Click on **Apply** to save the configuration.

5. Create a new user call **Operator2**.

6. Go to the **Application Actions** section and click on the **Unselect All** button so that there are no application permissions defined.

7. Click **Apply** and set **operator** as the password again.

   Note: Depending on the GENESIS version, you may 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** and select **Membership**.

9. From within the **Set the membership** dialog, expand the user dropdown list and select Operator.

10. Click on the green plus sign to **add** the **Operator** user to the **Operators Group**.

    NOTE: To remove a user from the group, select the user name and hit the delete key.

11. **Add Operator2** to the group also and click **OK** to close the dialog.

12. You may see a dialog informing you the **Operation was Successful**. Click **OK**.

13. You can confirm that the two users are in the Operators group by right-clicking on either a **User** or the **Group** and selecting **Membership** from the dropdown list.

14. To test the defined functionality, you will need to logout. You can go ahead and **Logout** now.

    NOTE: If you are still logged in as both users, you will have to logout twice.

15. Open the **icTonicsHome** display and go into **Runtime**. You should be blocked by the **Access Denied** display because you are not logged in.

16. Click on the button to login. **LOGON** as the user **Operator2**.

17. You should then see the **Tank Monitor** button as you would if you were logged in as Administrator. But you cannot exit Runtime.

18. Login as **Operator** now. 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 can still see the layer. That is because we specifically denied this access for the Operator but not for the group. Remember, that users will assume the least restrictive permissions.

14.4 FrameWorX64 Communication Outline

Once security is enabled on a FrameWorx64 server, clients expecting to retrieve data from that FrameWorX64 sever are required to be authorized through the GENESIS64 Security system. The following procedures are intended to outline key elements in the FrameWorX64 configuration that should be considered when security is enabled. This section focuses on three of the commonly used FrameWorx64 connect types.

- **OPC Classic → FWX** (FrameWorX)
- **FWX** (FrameWorX) → **FWX** (FrameWorX)
- **HH** (Hyper Historian) → **FWX** (FrameWorX)

Each of the entries in this section will be discussed in more detail as we proceed.

NOTE: On site, you may need Windows Administrator privileges or assistance from a Network, Firewall or SQL Server administrator to perform some of the actions listed below.

ICONICS software has been 'Network Aware' nearly since the company started in 1986 and our engineers have done a great job making GENESIS network implementation simple. However, computer networking by nature is complex. Cabling, routers, firewalls and security, just to name a few, are all components that comprise a network today. Coordination with a variety of specialized administrators, including those handling Windows and SQL Server, may be necessary in order to successfully facilitate what you intend to accomplish.

Considerations:

- GENESIS64 Security is not involved when browsing for tags using the Data Browser.
- It is recommended that each GENESIS64 node that will be included in a pool of nodes that will be communicating to remote FrameworX64 services share the same GENESIS64 Security database. Sharing the same security database can be accomplished in a variety of ways.

- Using the GENESIS64 database Export and Import utilities. This **Does Not** require authorization from the remote FrameworX64 server.
- Sharing the GENESIS64 Security SQL Server database, allowing remote connections to this security database and utilizing the GENESIS64 Connect Database feature. This **Does** require authorization from the remote FrameworX64 server.
- Using SQL Server backup and restore utilities. This **Does Not** require authorization from the remote FrameworX64 server.

Network Discovery Prerequisites:

- Any network node that is intended to be a FrameworX64 Server or Client must be "Discoverable" on the local area network. In order to accomplish this, confirm the following:
  - All the FrameworX64 nodes need to belong to the same Windows Domain or Workgroup.
  - Windows networking can discover each GENESIS node on the LAN.

Utilities (expected to be executed by a network administrator):

- Windows Network Discovery.
- DOS Commands such as NSLookup, IPConfig and Ping. Network administrators may have additional tools at their disposal.

GENESIS64 Utilities:

- TraceWorX64
- MonitorWorX Viewer
- Dataspy64
- License Utility

14.5 FrameWorX64 Node Discovery

GENESIS64 leverages the Windows networking platform for FrameWorX64 discovering computers on the network. Essentially, if Windows does not discover nodes on the network, FrameWorX64 will not discover the nodes either. Communications may, or may not, be accomplished using IP addresses. However, this is not the preferred method.

1. We will start by attempting to discover the GENESIS64 nodes in the 'Network Neighborhood'.
2. Open Windows **File Explorer** and navigate to **C:\Program Files\Common Files\ICONICS\FWX64\BIN**.
3. Double-click on **Dataspy64.exe**
4. On the right-hand frame Double-click on **Network Neighborhood** and expand the **Network** Neighborhood.
5. The following Figure L14.2 displays in Dataspy: the nodes on the Foxboro, MA, USA training facility network.

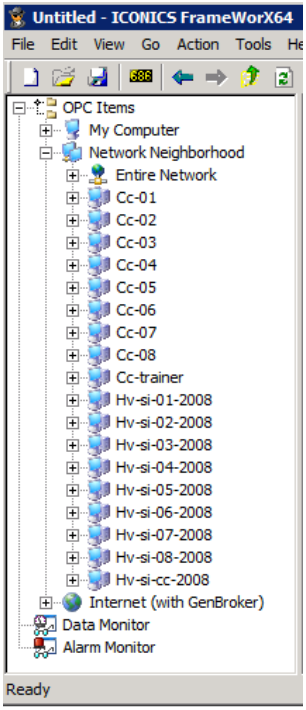


Figure L14.2 – ICONICS Training Facility Network Neighborhood

6. If you did not discover all the GENESIS64 nodes, consider this outline for troubleshooting.

7. Operating System-level Troubleshooting:

a. Does network discovery in Windows File Explorer find all GENESIS nodes?
   i. Yes.
      1. Go to FrameWorX Testing.
   ii. No.
      1. Check your **Domain** controller or **Workgroup** configuration.
         a. Are all the GENESIS64 nodes in the same Domain or Workgroup? You may consider creating a new Workgroup for GENESIS64 nodes.
      2. Is Windows **Network Discovery enabled** under Network and Sharing Center → Advanced sharing setting?
      3. Is there a **Firewall** that may be blocking this functionality?
      4. When network discovery is functional, proceed to FrameWorX Testing.

14.6 FrameWorX64 Testing

Following the steps in this lab will demonstrate the effects GENESIS64 Security has on remote FrameWorX64 communications. By now, your instructor should have assigned at least two nodes for testing. You will have to substitute the Server and Client references in this lab with actual computer names assigned by your instructor. Let's get started.

NOTE: At least two nodes are required to perform the following steps. As a matter of reference in this procedure, the two nodes will be referred to as **Server** (FrameWorX64 server) and **Client** (FrameWorX64 client). For documentation purposes:

- Server = Hv-si-08-2008
- Client = Hv-si-03-2008

Firewall:

Depending on your Firewall Rule Base, there could be constraints on the ports that GENESIS will be using for communications (ports **8778** and **38080**, for example). In order to avoid unnecessary complication, we will **Turn Off** the **Firewalls** for both the Server and the Client.

1. On each computer, select Control Panel → Windows Firewall → Turn Windows Firewall on or off.

2. Select the **Turn off Windows Firewall** for both the **Private** and **Public** networks and click **OK**.

   NOTE: On site, a Firewall Administrator could open the ports that FrameWorX64 uses for communications, rather than shutting the firewall off completely. Contact ICONICS Technical Support if this is necessary.

Client GraphWorX64 Display:

1. On the **Client** node, launch **GraphWorX64** and save the display as C:\Training**FWXTest.gdfx.**

2. Add a **Process Point** to this blank display. This launches the Data Browser.

   Note: When selecting the following tag, select the **Sine Folder** (not the Sine tag) and the **VerySlowXL359** tag is at the bottom of the tag list.

3. From within the **Data Browser**, browse to **Networks → FrameWorX Network → Hv-si-08-2008 → Data Sources → OPC Data Access → Simulations → Float → Sine → VerySlowXL359**.

4. Click **OK**.

5. Go into **Runtime**. The tag will not update and there will be an asterisk on the screen indicating a communication failure.

   Note: In the following step, if you do not see a tooltip in Runtime, go back into the GraphWorX64 configurator and, from the Display Canvas, set the **ShowToolTips** property to **True**.

6. Hover the mouse over the process point. As seen in the figure below, notice the **BadUserAccessDenied** entry in this message. This indicates the Server is not allowing the Client access to read the tag value.

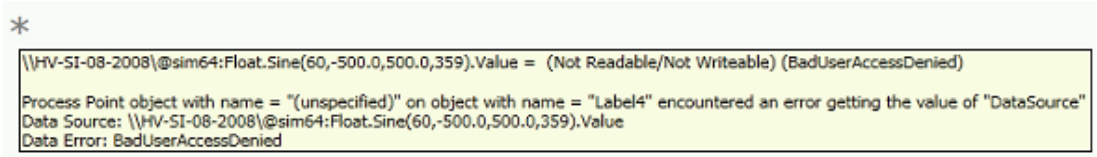


Figure L14.3 – Server Denies Client Access

Server Security Database Export (FYI Only - Not needed in the training).

You only have to perform this section if the Security databases on the Client and Server differ.

If this lab was executed from beginning to end on each client and server computer being used for FrameWorX64 communication, the security database configuration should be identical on both computers. If a student decided to use (create) user names and/or password different than what the lab called for, this will need to be accounted for as we proceed. Having said this, and in order to be sure the security databases are identical, we will use the GENESIS64 export and import utility.

1. On the **Server**, open **Workbench**.
2. Select the **Security** Server.
3. Right-click on the **Security** Server and select **Export Sub-Tree**.
4. Leave the **default setting** in the **Export Options** dialog and click **OK**.
5. Click **Close.**
6. In the Export Sub-Tree dialog, click on the file name.
7. Select **Save As** and save the file as **C:\Training\MyComputerSEC.csv**
8. Close the **Sub-Tree** dialog and **Click** the **Open Folder** button.
9. Right-click on the **MyComputerSEC.csv** file and select **Copy**.

   NOTE: If you are attending class in Foxboro, the instructor can assist in mapping the C$ drive of the Client computer. Otherwise, you can use a flash drive to move the file.

10. **Paste** the **MyComputerSEC.csv** file into the Client's **C:\Training** folder.

Server Security Database Import (FYI Only - Not needed in the training).

You only have to perform this section if the Security databases if the Client and Server differ.

1. On the **Client**, open **Workbench**.
2. Select the **Security** Server.
3. Right-click on the **Security** Server and select **Import**.
4. Leave the **Default** settings in the **Import Options** dialog and click on the File **Browse** button.
5. Select the **C:\Training\ MyComputerSEC.csv** file and click **Open**.
6. Click **Upload**.

NOTE: There are other techniques available, other than export/import, which will allow for security database replication. You may consider researching a technique that works best for your application.

7. Click **Close**.

14.7 Platform Services Configuration

Now that we have ensured that the records (User and Group info) in the Server Security database also exist in the Client Security database, we can continue. Be sure that the user name (Administrator) and password (admin) exist. We will use this credential pair for the Client FrameWorX64 configuration.

FrameWorX64 Configurator:

In this section, we will configure the Client user name and password to gain access and authorization to retrieve data through **FrameWorX → FrameWorX** from the Server to the client. Let's check it out.

1. On the **Client** (HV-SI-03-2008), select **Tools** tab on the Workbench and open **Platform Services** from the Tools section of the ribbon.
2. In the **Platform Services Configuration** dialog, select the **Passwords** tab.
3. Select the gray bar under Servers and Credentials labeled + Click here to add new item.
4. Figure L14.4 shows the **Platform Services Configuration → Passwords** dialog.

NOTE: From within the Basic tab, you could configure a Remote License Server. See the GENESIS64 online Help for more information.

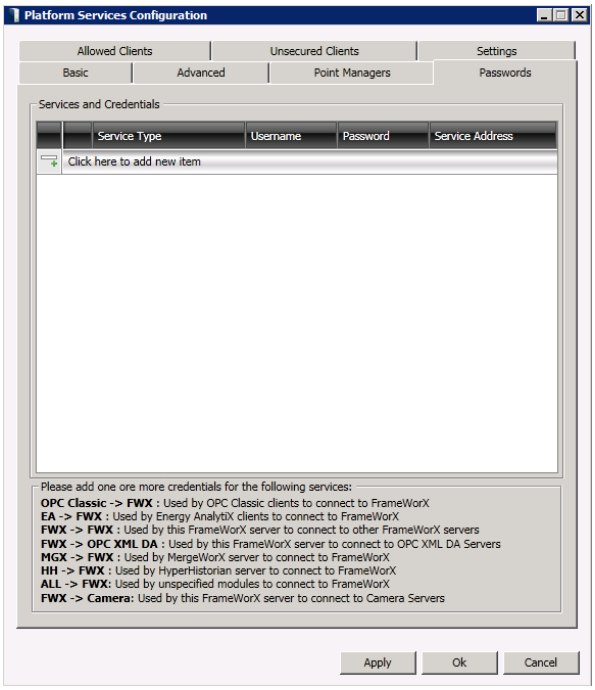


Figure L14.4 – FrameWorX64 Authentication Configuration

5. Add a new item and specify the **Service Type** as **FWX → FWX**.

6. Enter the Username **Administrator** and the Password **admin**.

   NOTE: To remove an entry from the list in this dialog, select the entry and hit the delete key.

7. Leave the **Service Address** as ***.**

8. Click **Apply** and **OK**.

   NOTE: FrameWorX64 service-related changes may require a **Restart** of the **FrameWorX64** service for the changes to take effect.

9. We will proceed to restart the **FrameWorX64** service on the **Client.**

10. Type **Services.msc** into the **Search the web and Windows** box. Hit **Enter**. This launches the Windows Services dialog.

11. From the **Client**, within the **Services** dialog, right-click on the **ICONICS FrameWorX** service and select **Restart**.

12. Once the restart is complete, on the **Client**, return to the **GraphWorX64** display and **Log in** as Administrator.

13. You should see the data updating in the display.

This concludes this lab.

RecipeWorX Basics 15

A **Recipe** is a list of ingredients that is used to create a product. A recipe specifies the amount of each ingredient and is categorized by product variation. Recipes simplify work with a large set of data tags (recipe parameters). Each recipe consists of any number of recipe parameters.

The **Recipes** branch of the RecipeWorX provider in the Workbench enables you to specify a list of ingredients that are used to create a product, such as cookies.

15.1. Objectives

After completing this chapter, you should know how to:

- Create and edit recipes (products).
- Create folders for recipes.
- Create and edit recipe items (product variations).
- List types of ingredients (e.g. flour, sugar, oil) used to create a product (e.g. cookies).
- Specify the amount of each ingredient needed.
- Categorize recipes (e.g. version, flavor, or color) by product variation (e.g. chocolate chip, oatmeal, or peanut butter).
- Easy runtime integration to select recipes within applications.
- Display and edit recipe data within applications.
- Import/export recipe configurations from/to XML and CSV files.

15.2. Architecture

RecipeWorX consists of the following key components:

- **Workbench** – Allows configuration of recipes using the RecipeWorX provider.
- **GraphWorX64** and **PortalWorX-SL** controls:
  - **Recipe Navigator** (tree) – Used to browse for recipes and recipe items in Runtime.
  - **Recipe Grid** control – Displays the recipe parameters in Runtime.
- **Commanding** – Provides support for important actions related to recipes.
- **Recipe Point Manager** (FwxRecipeService) – the engine that handles recipe processing.

15.3. Overview of Recipe Configuration

The following section provides an overview of how recipes are configured in the RecipeWorX provider in the Workbench. Recipes are organized in a tree structure of folders, subfolders, recipes, and recipe tags (items).

- **Folders (product category):** Folders allow organization of recipes into a hierarchical tree structure of different related recipe categories. As an example of this, we will create recipes contained in a folder named **icTonics** in the next lab.
- **Subfolder (product type):** Subfolders enable you to further categorize recipes by product type. Using our previous example, the **icTonics** folder could contain a subfolder called **Chocobeer, CocoFizz,** and **Frappe,** which contain different product recipes for pills.
- **Recipe (product variations):** A recipe represents a variation of a specific product type. For each product type, you can have multiple recipes. Using our previous example, the **icTonics** folder could contain recipes for **Chocobeer, CocoFizz,** and **Frappe.**
- **Recipe items (recipe variations):** Each recipe consists of any number of recipe items. Using our previous example, the **Choocbeer** recipe could contain recipe items for both **Extra Strength** and **Regular** varieties.

15.4. Using Recipes in Applications

The recipes and recipe items that you configure in the RecipeWorX provider in the Workbench are available as data tags that can be used in several different GENESIS64 client applications.

15.4.1. Creating a New Recipe Folder

Recipes can be organized in a tree structure of folders and subfolders. Folders allow organization of recipes into different recipe categories. Subfolders enable you to further categorize recipes by product type.

To Create a New Recipe Folder:

1. Right-click on the **Recipes** branch of the RecipeWorX provider in the Workbench and select **Add Folder**, as shown in the following figure.

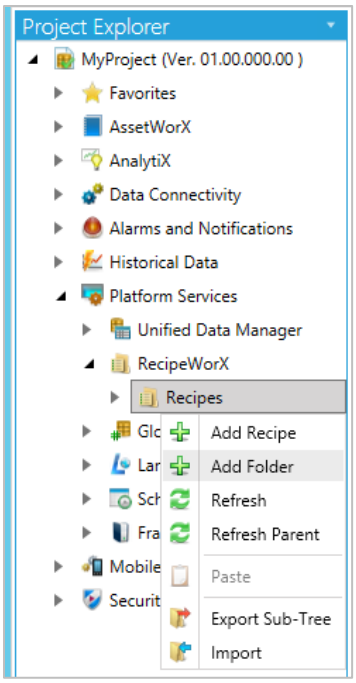


Figure 15.1 - Creating a New Recipe Folder in the RecipeWorX Provider in the Workbench

-OR-

Select the Recipes branch of the RecipeWorX configurator provider in the Workbench and click on the **Add Folder** button in the Edit section of the Home ribbon.

Figure 15.2 - Creating a New Folder in the Edit Section of the Home Ribbon in the Workbench

2. This opens the Recipe Folder properties window in the right-pane of the Workbench, as shown in the following figure. In the **Name** field, type a name for the new folder.

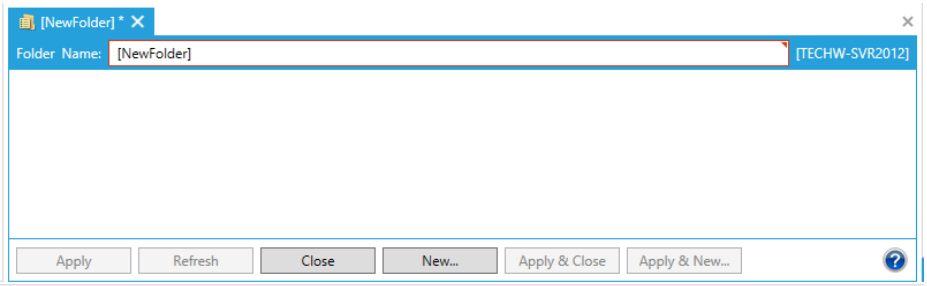


Figure 15.3 - Setting the Recipe Folder Properties

3. When you have finished configuring the folder properties, click the **Apply** button to save your settings. The new folder appears under the **Recipes** branch of the tree explorer.

15.5. Creating a New Recipe

To Create a New Recipe:

1. Right-click on the Recipes branch of the RecipeWorX configurator provider tree explorer in the Workbench and click on **Add Recipe**, as shown in the figure below.

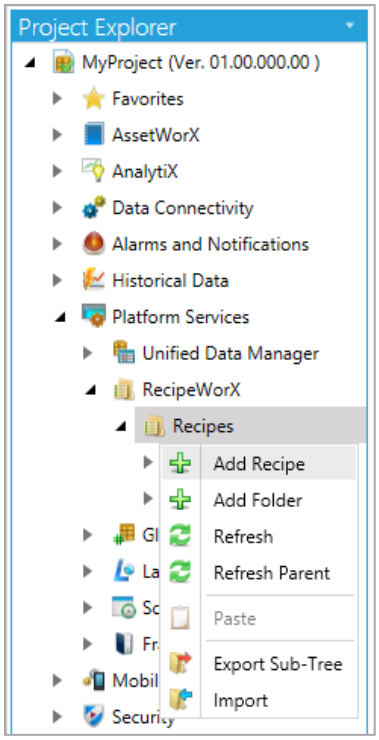


Figure 15.4 - Creating a New Recipe in the RecipeWorX Provider in the Workbench

-OR-

Select the Recipes branch of the RecipeWorX configurator provider tree explorer in the Tree Explorer in the Workbench and click on the **Add Recipe** button in the Edit section of the Home ribbon.

Figure 15.5 - Creating a New Recipe in the Edit Section of the Home Ribbon in the Workbench

2. This opens the Recipe properties window in the right-pane of Workbench, as shown in the figure below. In the **Recipe Name** field, type a name for the new recipe.

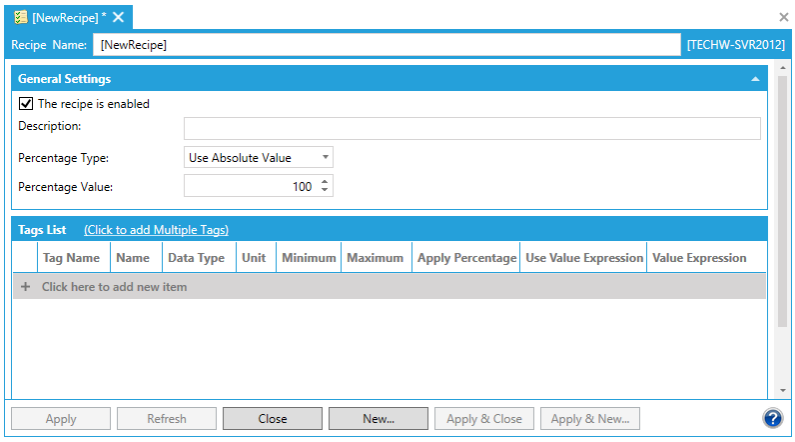


Figure 15.6 - Setting Recipe Properties in the Workbench

3. When you have finished configuring recipe properties, click the **Apply** button to save your settings. The new recipe appears under the **Recipes** branch of the tree explorer.

15.6. Recipe Properties

A **recipe** represents a variation of a specific product type. For each product type, you can have multiple recipes.

In the **Recipe properties**, you can configure the following settings:

General Settings

- **The recipe is enabled:** Activates the recipe in the recipe server configuration. Enabled recipes are available to client applications.
- **Description:** Enter a description for the recipe (optional).
- **Percentage Type:** The Percentage allows you to create a smaller or larger variation of a recipe without needing to change the individual values in each of the Recipe items. When the values are downloaded to the data tags, the percentage will then be applied values downloaded. You have the option of Use Absolute Value, where you can then type in the **Percentage Value** or you can Use Tag, where you can then select the **Percentage Tag** via the button.

Tags List

The next step is to define a set of ingredients (parameters) for the recipe. Each ingredient (parameter) corresponds to a data tag.

- Click on "+ Click here to add new item" to add individual tags. You can then edit a tag row that will include the Tag Name, Name, Data Type, Unit, Minimum and Maximum settings described below. You can also opt to **Apply Percentage** (see "Percentage Type" in the General Settings description on the previous page) and/or **Use Value Expression** and then enter an expression directly into the text entry field or click on the button, which opens to the **Configure the expression** window, allowing you to construct your expression.

- **Tag Name -** Enter a tag name in the text entry field or click on the ellipsis button to open the Unified Data Browser to select a tag.
- **Name -** Provide a name for the tag.
- **Data Type -** Use the pulldown menu to select from NATIVE, FLOAT, DOUBLE, BOOL, BYTE, WORD, DWORD, CHAR, SHORT, LONG or STRING.
- **Unit -** Enter your selected unit in the text entry field.
- **Minimum** - Enter a minimum in the text entry field or use the up and down buttons to set a number.
- **Maximum** - Enter a maximum in the text entry field or use the up and down buttons to set a number.

You can also click on the "(Click to add Multiple Tags)" link, which will open the Data Browser directly and allow you to select multiple tags.

15.8. Creating a New Recipe Item

To Create a New Recipe Item:

1. Right-click on a **Recipe** in the **Recipes** branch of the RecipeWorX configurator provider in the Workbench and click on **Add Recipe Item**, as shown in the figure below.

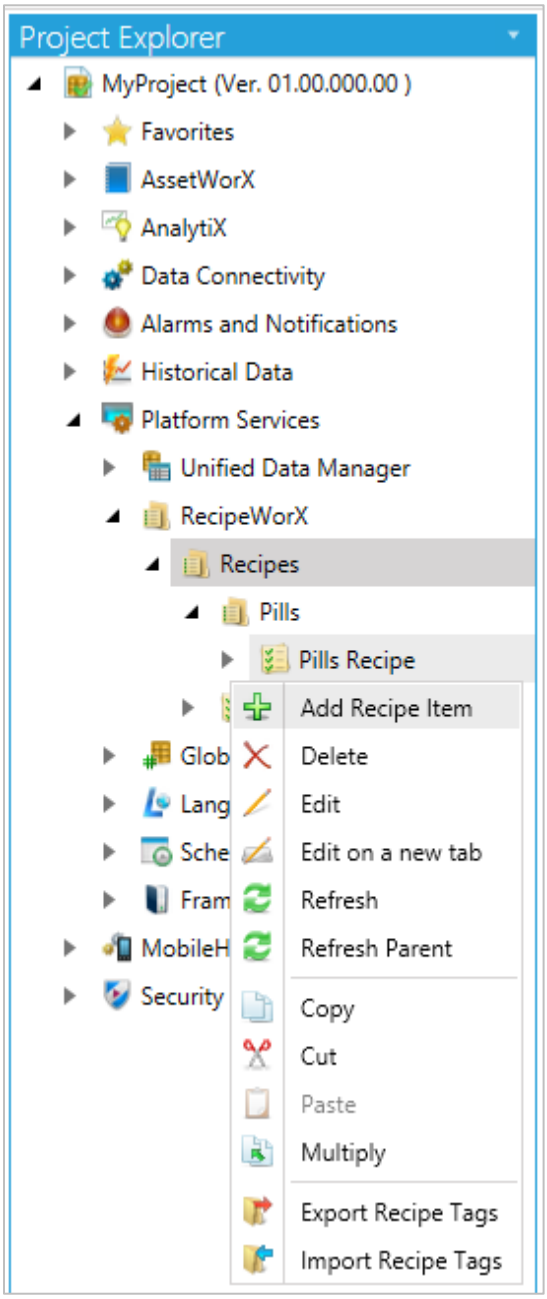


Figure 15.7 - Creating a New Recipe Item in the RecipeWorX Provider in the Workbench

-OR-

Select a Recipe in the Recipes branch of the RecipeWorX configurator provider in the Workbench and click on the **Add Recipe Item** button in the Edit section of the Home ribbon at the top of the screen.

Figure 15.8 - Creating a New Recipe Item in the Workbench Toolbars

2. This opens the Recipe Item properties window in the right-pane of the Workbench, as shown in the figure below. In the **Name** field of the **Recipe Item Properties** tab, type a name for the new recipe item.

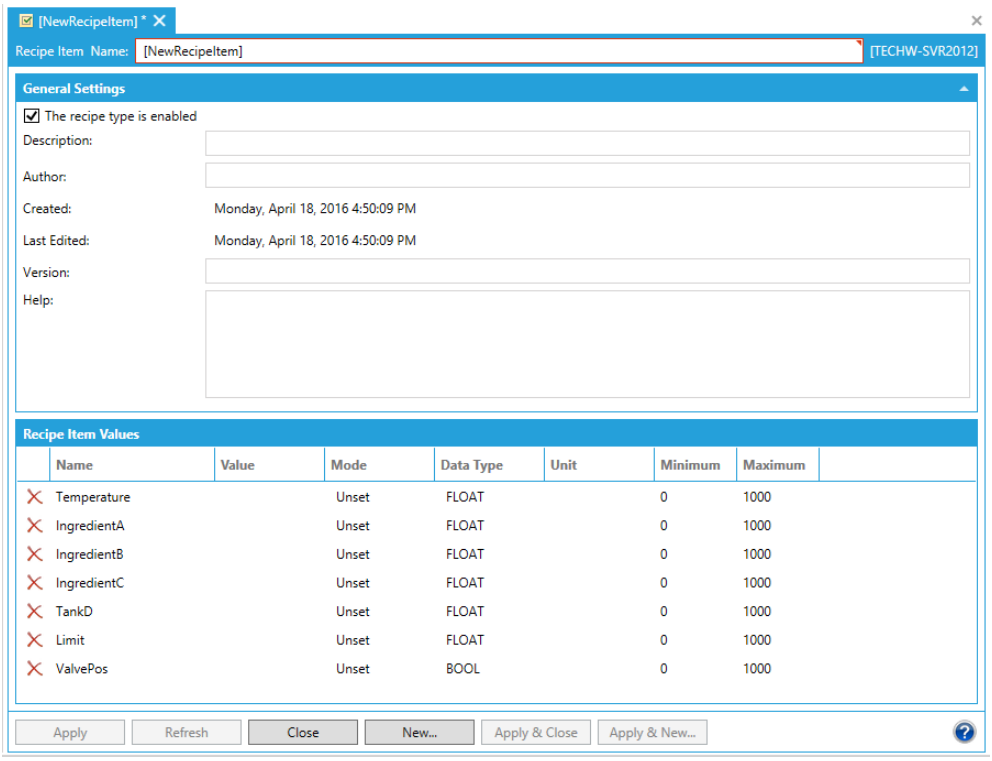


Figure 15.9 - Setting Recipe Item Properties in the Unified Data Manager Provider in the Workbench

3. When you have finished configuring recipe item properties, click the **Apply** button to save your settings. The new recipe item appears under the selected recipe, under the **Recipes** branch of the tree explorer.

15.9. Recipe Item Properties

Each recipe consists of any number of recipe items. In the **Recipe Item properties**, you can configure the following settings:

General Settings

- **The recipe type is enabled:** Activates the recipe item in the recipe.
- **Description:** Enter a description for the recipe item (optional).
- **Author:** Specify the name of the recipe author (optional).
- **Created:** Displays the date and time when the recipe item was created.
- **Last Edited:** Displays the most recent date and time when the recipe item was modified.
- **Version:** Specify a version number for the recipe item (optional).
- **Help:** Enter help instructions for the recipe item (optional).

Recipe Item Values

The **Recipe Item Values** lists the human-readable names for each of the ingredient index (parameter) numbers that are defined for the recipe ingredients, including the data type, unit of measurement (e.g. milligrams), and the range of minimum and maximum values. In the **Value** column, you must manually type in a value (quantity) for each recipe ingredient listed within the specified range of minimum and maximum values.

NOTE: Recipe values that are undefined will not be downloaded.

15.10. Recipe Commands

GENESIS64 features new Recipe-based Commands that provide a variety of functions.

The following parameters are available for all commands:

- **Command:** Use the pulldown menu to select your desired command. The remaining parameters will change automatically once you have made your selection.
- **Name**: The text to be listed in the equipment's right-click menu at runtime.
- **Description***: A description that explains what happens when a runtime user clicks this command. This description is used for informational purposes.
- **Execution Flags**: Multiple commands on a single asset can be default. All the default commands get executed when the user left-clicks the asset and show in bold in the popup menu. Select from Visible, Default or Drag & Drop.

- **Visible on all children**: Select this option if the command is to be made available in the right-click popup menu of all child equipment nodes. This option applies only if the equipment node has child nodes.
- **Available in**: The ICONICS products on which this command will be available to runtime users, such as GraphWorX64, WebHMI, PortalWorX-SL, PortalWorX for SharePoint and Mobile MI. Select all that apply. Note that options may vary depending on the command.

***** If the equipment node was created using an equipment class, and this or other fields contain aliases (that is, variables that are surrounded by **/?** and **?/**), you can replace the aliases with values that are appropriate for this equipment node by instantiating the aliases. For more information, refer to the Instantiating an Equipment Node's Aliases topic.

The remainder of this topic describes each command and its parameters.

15.11. Recipe: Copy Command

This command can be invoked by right-clicking on a recipe item in the Recipe Navigator and choosing “Copy Recipe Item” from the context menu. It can also be configured as a Pick action command in GraphWorX64. The command copies a selected recipe item from the tree explorer.

Parameters for the Recipe: Copy command are:

- The Command, Name, Description*, Execution Flags, Visible on all children, and 'Available in' prompts are described in section 15.9.
- Target - Registered name of the target Recipe Navigator control.

15.12. Recipe: Delete Command

This command can be invoked by right-clicking on a recipe item in the Recipe Navigator and choosing “Delete Recipe Item” from the context menu. It can also be configured as a Pick action command in GraphWorX64. It deletes the selected recipe item from the tree explorer. Parameters for the Recipe: Delete command are:

- The Command, Name, Description*, Execution Flags, Visible on all children, and 'Available in' prompts are described in section 15.9.
- Target - Registered name of the target Recipe Navigator control.

15.13. Recipe: Download Command

This command can be invoked by right-clicking on a recipe item in the Recipe Navigator and choosing “Download Recipe” from the context menu. It can also be configured as a Pick action command in GraphWorX64. It downloads the values of the selected recipe item to the OPC tags. Parameters for the Recipe: Download command are:

- The Command, Name, Description*, Execution Flags, Visible on all children, and 'Available in' prompts are described in section 15.9.
- Target - Registered name of the target Recipe Navigator control.

15.14. Recipe: Load Command

This command can be invoked by right-clicking on a recipe item in the Recipe Navigator and choosing “Load Recipe” from the context menu. It can also be configured as a Pick action command in GraphWorX64. It loads the recipe item selected in the Recipe Navigator into the Recipe Grid control. Parameters for the Recipe: Load command are:

- The Command, Name, Description*, Execution Flags, Visible on all children, and 'Available in' prompts are described in section 15.9.
- **Target** - Registered name of the target Recipe Navigator control.

15.15. Recipe: New Command

This command can be invoked by right-clicking on a recipe in the Recipe Navigator and choosing “New Recipe Item” from the context menu. It can also be configured as a Pick action command in GraphWorX64. It creates a new child recipe item under the selected recipe in the tree explorer. Parameters for the Recipe: New command are:

- The Command, Name, Description*, Execution Flags, Visible on all children, and 'Available in' prompts are described in section 15.9.
- Target - Registered name of the target Recipe Navigator control.

15.16. Recipe: Save Command

This command can be invoked by right-clicking on a recipe item in the Recipe Navigator and choosing “Save Recipe” from the context menu. It can also be configured as a Pick action command in GraphWorX64. It applies changes to the dataset that is loaded into the Recipe Grid control. Parameters for the Recipe: Save command are:

- The Command, Name, Description*, Execution Flags, Visible on all children, and 'Available in' prompts are described in section 15.9.
- Target - Registered name of the target Recipe Grid control.

15.17. Recipe: Upload Command

This command can be invoked by right-clicking on a recipe item in the Recipe Navigator and choosing “Upload Recipe” from the context menu. It can also be configured as a Pick action command in GraphWorX64. It uploads the values from the OPC tags and updates those values in the database for the selected recipe item. Parameters for the Recipe: Upload command are:

- The Command, Name, Description*, Execution Flags, Visible on all children, and 'Available in' prompts are described in section 15.9.
- Target - Registered name of the target Recipe Navigator control.

15.18. Recipe-based Commanding in Equipment Properties in AssetWorX

- Recipe: Copy
- Recipe: Delete
- Recipe: Download
- Recipe: Load
- Recipe: New
- Recipe: Save
- Recipe: Upload

15.19. Commanding for Recipes

In addition to issuing recipe commands in the Recipe Navigator, it is also possible to use GraphWorX64 pick actions to issue recipe commands.

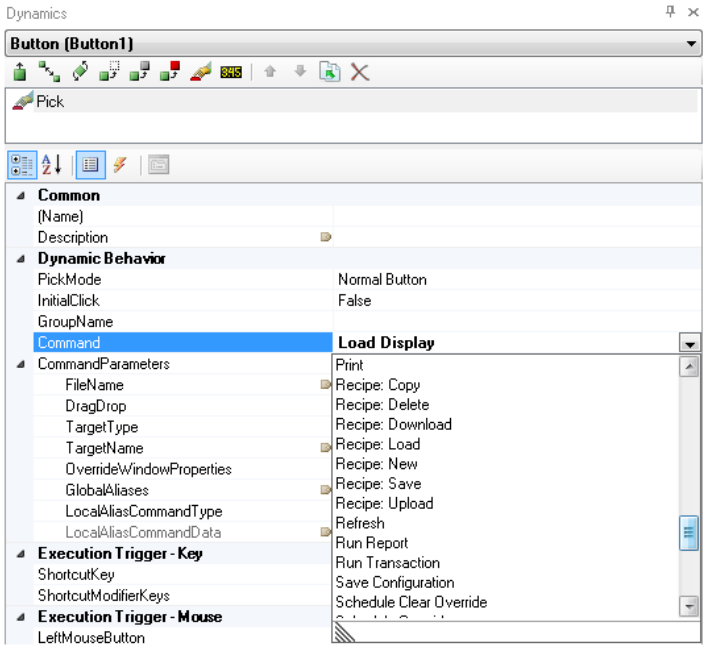


Figure 15.10 - Recipe Commands

You can preconfigure your display with all of the command buttons that user will need to control recipes.

Figure 15.11 - Sample GraphWorX64 Buttons Showing Recipe Commands

15.20. Runtime

RecipeWorX Runtime support is primarily exposed through the Recipe Navigator and Recipe Grid controls. The **Recipe Navigator** is a tree control that allows you to browse the configured recipes and issue commands related to recipes. The **Recipe Grid** control provides a grid view to display the parameters of a recipe item. Using the Recipe Grid, you are able to modify parameter values and apply changes to the recipes items. These controls are available in the ribbon in GraphWorX64.

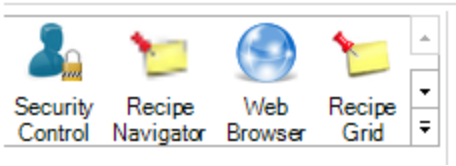


Figure 15.12 - Recipe Controls in the GraphWorX64 Ribbon

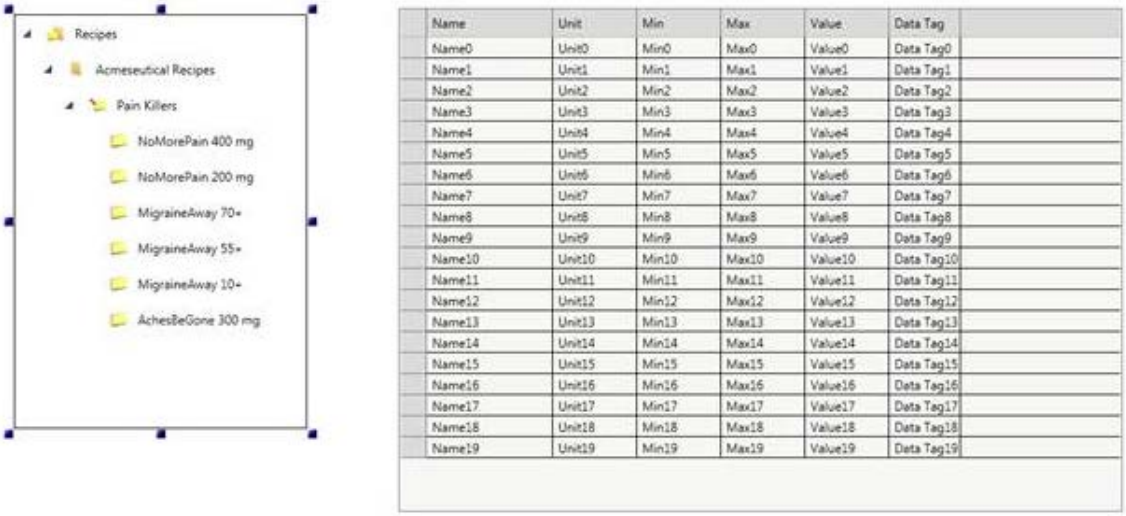


Figure 15.13 - Recipe Controls in GraphWorX64 (Configuration Mode)

15.21. Important Properties

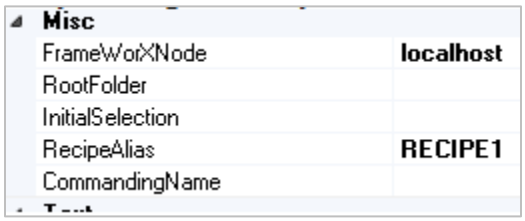


Figure 15.14 - Recipe Navigator Properties

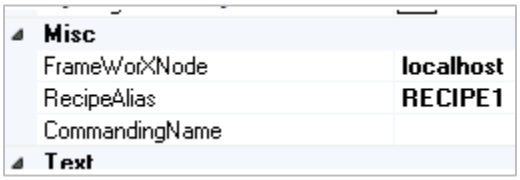


Figure 15.15 - Recipe Grid Properties

- **FrameWorXNode**: Enter the location of the computer where the Recipe Point Manager (FwxRecipeService) is running. The default value is “localhost”. The Recipe Navigator and Recipe Grid must connect to the same computer.

- **RootFolder**: This can be used to show a subtree of the recipes in the Recipe Navigator, which can be useful for restricting access to a subset of recipes and presenting only those. This value should be the full path of the desired folder, separated by periods. For example, "Folder1.SubFolder1" or "Beverages.Soda". If the RootFolder is not specified, then all recipe folders, recipes, and recipe items will be shown.
- **InitialSelection**: This property can be used to automatically expand the tree to a particular item and select it. The value should contain the full path, separated by periods. For example: "Beverages.Soda.Cola". If the RootFolder is specified, then you only need to specify the full path starting at the RootFolder. So, if the RootFolder is "Beverages.Soda", then the InitialSelection can be "Cola" to select the Cola item.
- **RecipeAlias**: This field is used to bind the Recipe Navigator to the Recipe Grid, so the values must match in the two controls. The default value in both controls is "RECIPE1".
- **CommandingName**: Specify the name which should be used as a Target value for commands that are directed at this control.

15.22. Using Recipes in PortalWorX-SL

PortalWorX-SL has a built-in Recipe Navigator that can be enabled. To enable the Recipe Navigator, go to the Project tab, and enable the Recipe Navigator under the Navigator selection options.

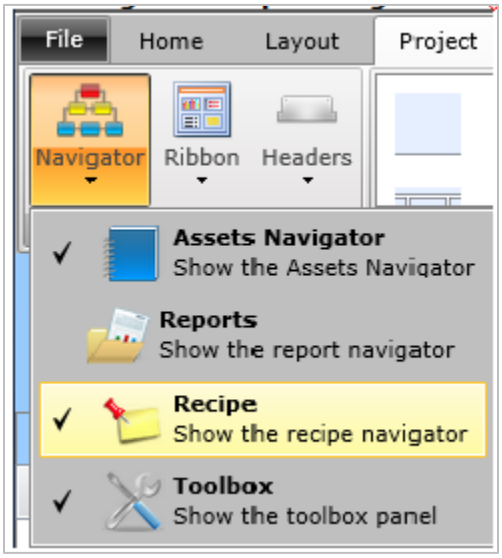


Figure 15.16 - Enabling the Recipe Navigator in PortalWorX-SL

To show recipes in a grid control in PortalWorX-SL, you need to save a recipe grid control configuration file in Silverlight format.

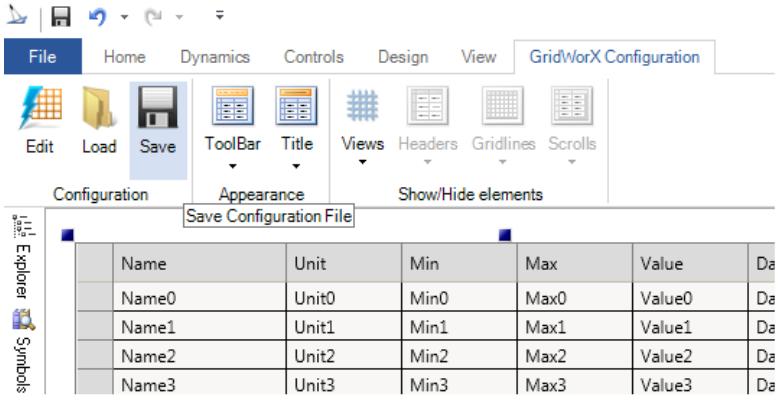


Figure 15.17 - Save the GridWorX Configuration

This file should be saved to the PubDisplay folder, or another Web-accessible location.

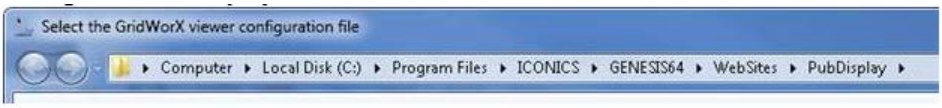


Figure 15.18 - Saving to the PubDisplay folder

In PortalWorX-SL, you need to insert a GridWorX Viewer and then load the saved configuration file.

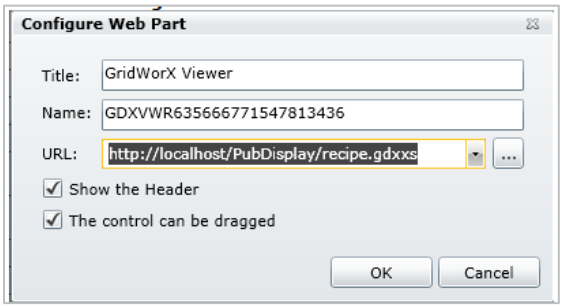


Figure 15.19 - Load the Configuration File

Now you are ready to use the Recipe Navigator in PortalWorX-SL to control your recipes.

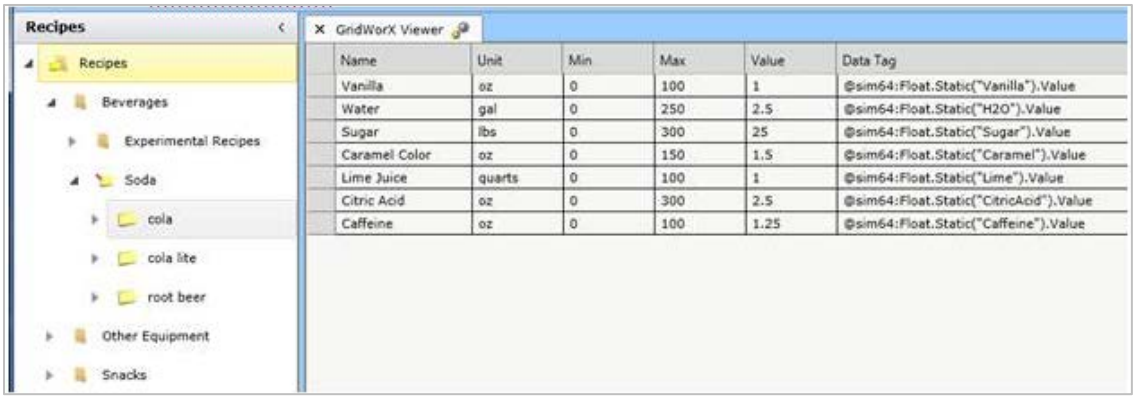


Figure 15.20 - Recipes in PortalWorX-SL

15.23. Recipe Management Data

Additional information about the currently loaded recipe is exposed through data access tags. These tags can be found in the Data Browser under "Data Sources/Platform Services/Recipes/.Recipe Management".

There are two categories of recipe management tags:

1. **Recipe Data** – provides detailed information about the currently loaded recipe, such as Author and RecipeName. The tags that provide information about a particular row in the recipe are: DataTag, MaxValue, MinValue, Unit, Value, and ValueName.

2. **Runtime Info** – provides Runtime-related information about the results for uploading or downloading recipes.

   a. **DownloadedErr** – number of tags that failed to download for the last download operation.
   b. **DownloadedOk** – number of tags that were downloaded successfully for the last download operation.
   c. **DownloadedPending** – number of tag downloads pending for the last download operation.
   d. **LastRecipeDownloaded** - the name of the last recipe that was downloaded.
   e. **LastRecipeUploaded** – the name of the last recipe that was uploaded.
   f. **RequestedErr** – number of tags requested in error for the last upload or download.
   g. **RequestedOk** – number of tags requested successfully for the last upload or download.
   h. **RequestedPending** – number of tags requests pending for the last upload or download.
   i. **UploadedErr** – number of tags that failed to upload for the last upload operation.

j. **UploadedOk** – number of tags that were uploaded successfully for the last upload operation.
k. **UploadedPending** – number of tag uploads pending for the last upload operation.

Each of these tags contains an alias parameter, which by default matches the default Recipe Alias for the Recipe Navigator and Recipe Grid. For example, this tag would show the author of the currently loaded recipe item:

rcpm:RecipeData.Author("RECIPE1")

Some tags also contain a row parameter, to specify a particular row in the grid (a parameter of the recipe item). These row numbers start at zero for the first row.

For example, this tag will display the name of the parameter (ingredient) for the first row:

rcpm:RecipeData.ValueName("RECIPE1",0)

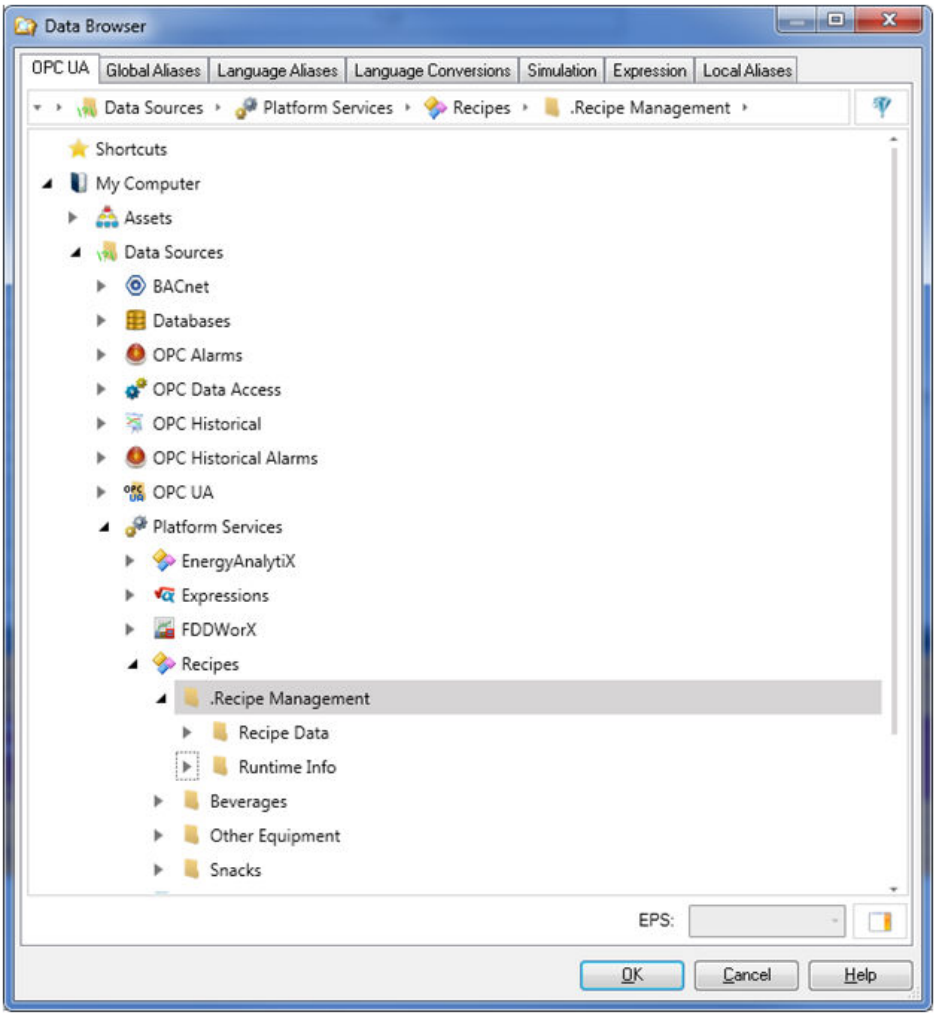


Figure 15.21 – Data Browser Showing Recipes

Lab: Recipes (Value Sets) *L15*

15.1. 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**

15.1.1. Task Checklist

In this lab, you will complete the following tasks:

- Create a Recipe folder.
- Add Recipes.
- Edit the displays to allow for setting Recipes.

15.1.2. Support Files

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

- **icTonicsTankMonitor.gdfx**

15.2. RecipeWorX

Recipes allow you to write different preset values to different OPC tags with a click of a button. In this section, we will add recipes and allow for changes in recipes during Runtime.

1. Open the Workbench if you closed it from the previous lab.

2. In the Project Explorer, click **Platform Services** to expand the menu. Select RecipeWorX in the Project Explorer. Select the Recipes folder and right-click to add a folder.

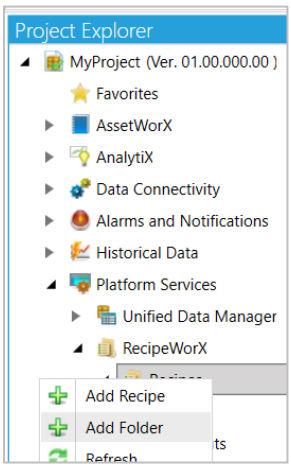


Figure L15.1 – RecipeWorX in the Project Explorer

3. Name the folder **icTonicsRecipes** then click **Apply**.

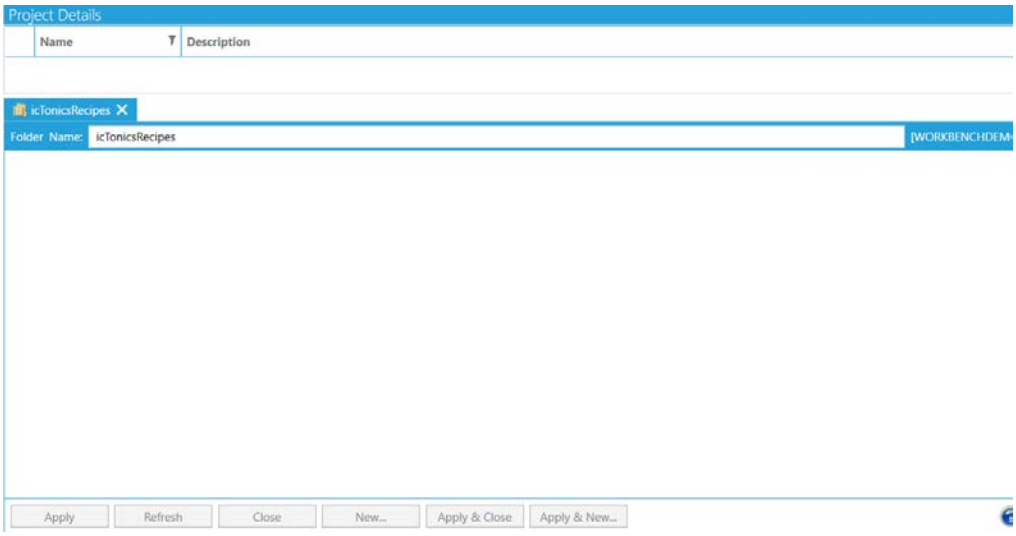


Figure L15.2 – Naming the Folder

4. Right Click on the **icTonicsRecipes** folder icon and Select Add Recipe Item.

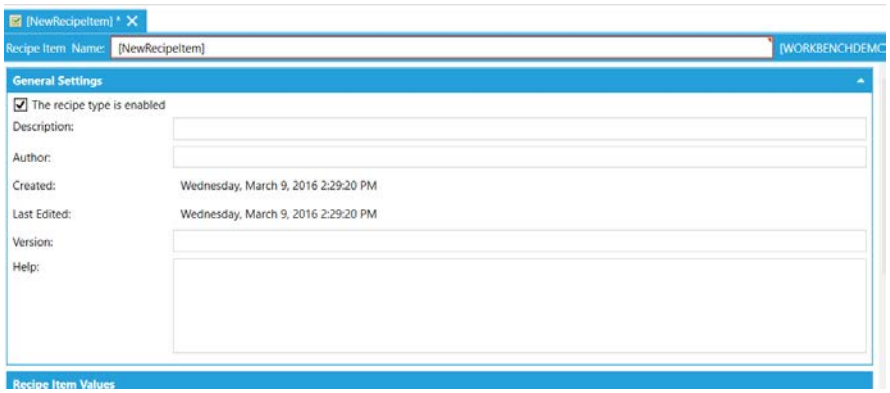


Figure L15.3 – General Settings in a New Recipe

5. Add a **Recipe** and name it **icTonics** then click on **Apply** to save it.

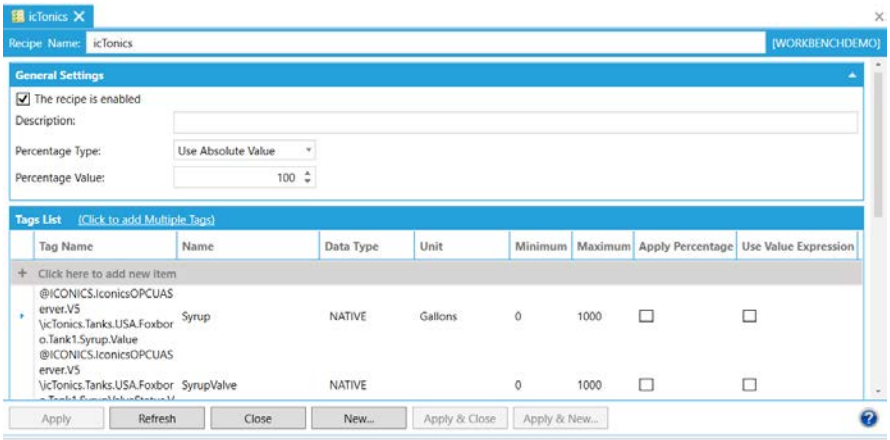


Figure L15.4 – Named Recipe

6. Scroll Down to the icTonicsRecipes recipe in the Data Browser and add a **Tags List**. Right click on the tag icon under Tag Name and browse to your Data Browser and select the tag Syrup.

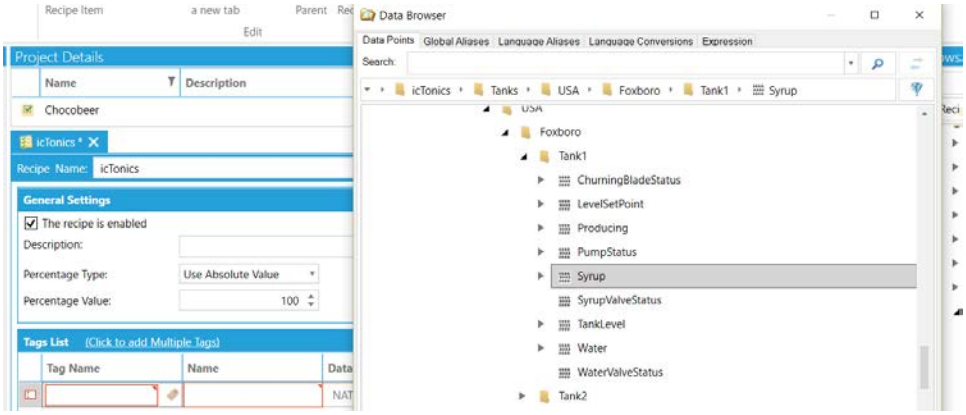


Figure L15.5 – Adding a Tags List

7. Click on the **+ Click Here to Add Tags** tab. click on the tag icon to bring up the Data Browser and select the **Syrup** tag.

8. Repeat step 6 for the **Producing**, **SyrupValve, LevelSetPoint, Water, WaterValve and ChurningBlade** tags. Your finished configuration should look similar to the figure below.

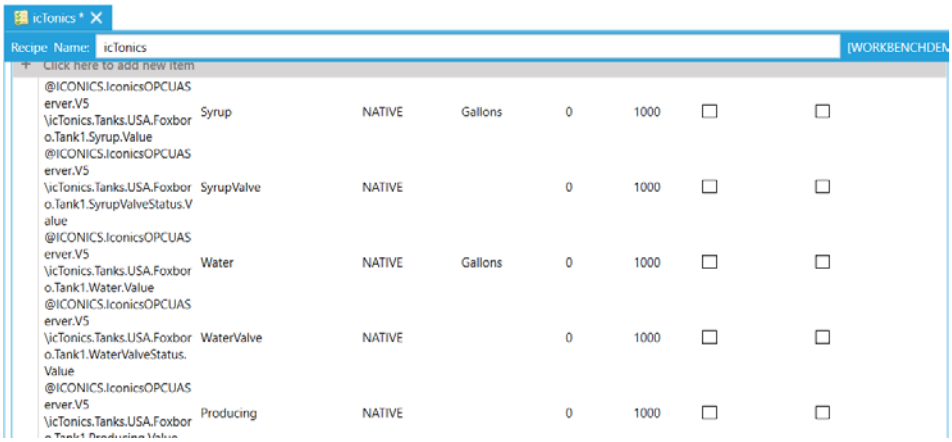


Figure L15.6 – Finished Configuration

9. Right-click the icTonics folder in the Project Explorer and click the + to add a Recipe.

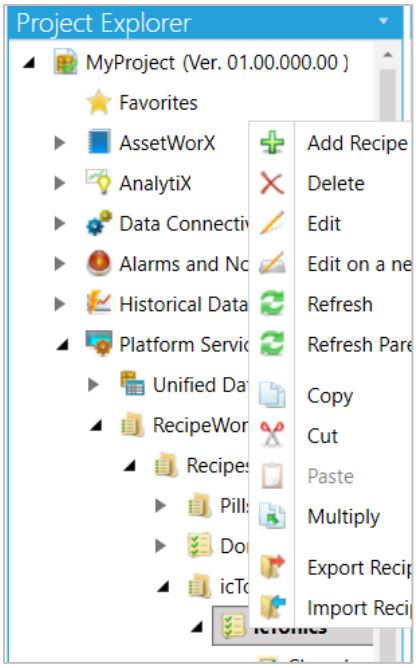


Figure L15.7 – Adding an Additional Recipe Item

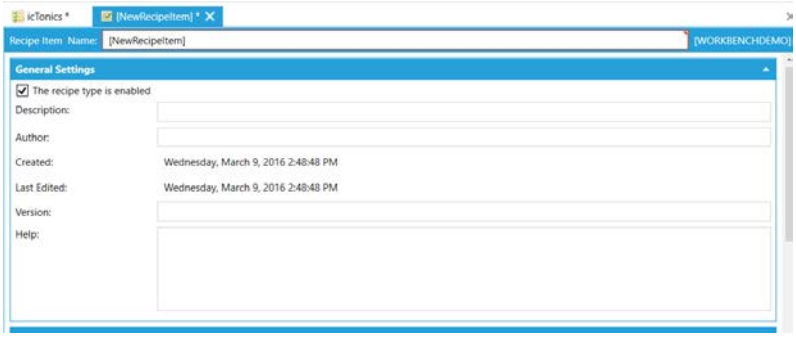
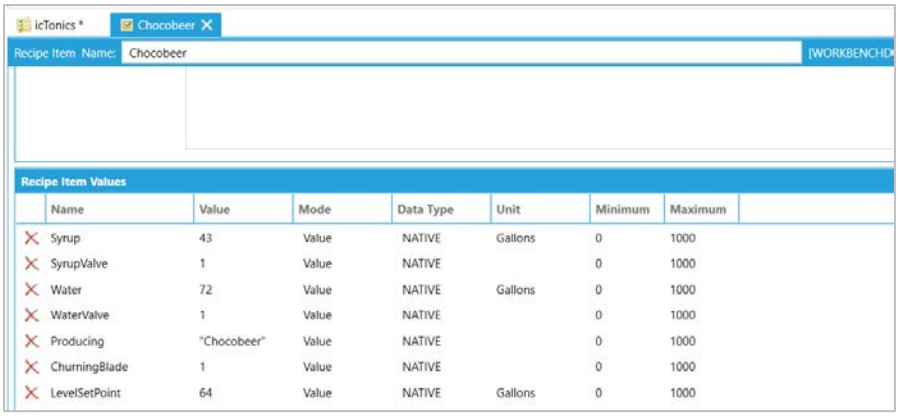


Figure L15.8 – Completing the Settings for an Additional Recipe Item

10. Click on the Recipe Item Name and type in **Chocobeer**, and it will inherit all the tags defined in the icTonics folder. You can now click on the value area and enter in the default values for Recipe Items and also the units for the items. Click on "**Apply**" to save the recipe item value set.

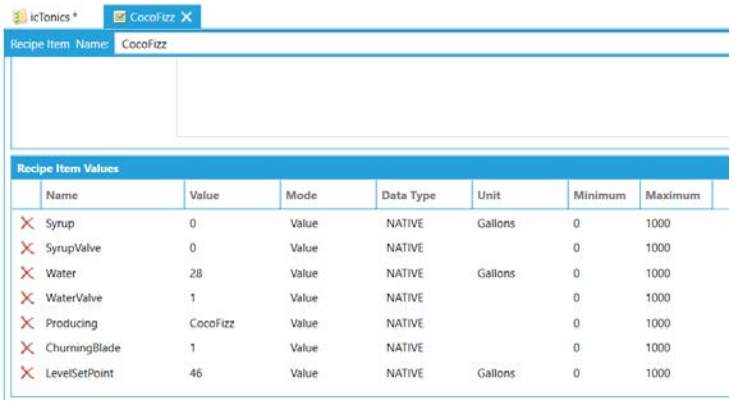


Name	Value	Mode	Data Type	Unit	Minimum	Maximum
Syrup	43	Value	NATIVE	Gallons	0	1000
SyrupValve	1	Value	NATIVE		0	1000
Water	72	Value	NATIVE	Gallons	0	1000
WaterValve	1	Value	NATIVE		0	1000
Producing	"Chocobeer"	Value	NATIVE		0	1000
ChurningBlade	1	Value	NATIVE		0	1000
LevelSetPoint	64	Value	NATIVE	Gallons	0	1000

Figure L15.9 – Naming an Additional Recipe Item

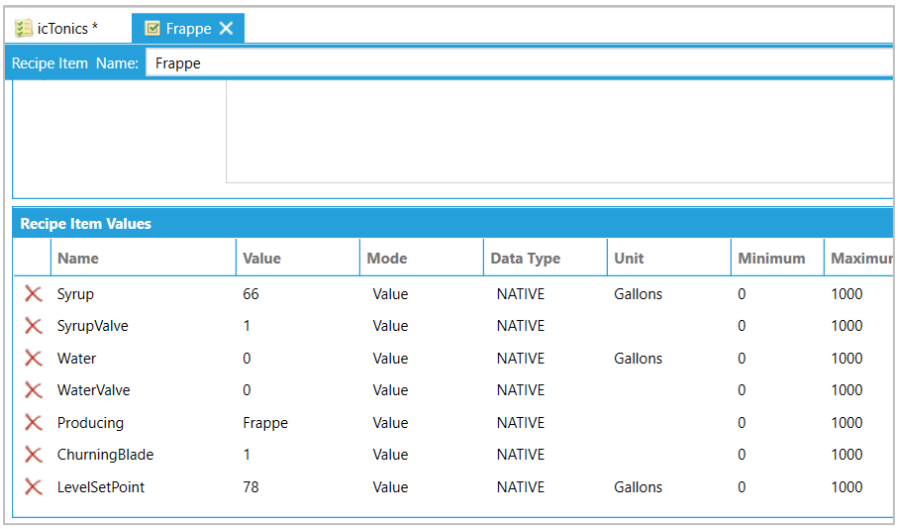
12. Repeat the above steps to insert two more value sets, **CocoFizz** and **Frappe**.

13. Enter the corresponding values according to the chart below so that your finished configuration looks like the figure below.



Name	Value	Mode	Data Type	Unit	Minimum	Maximum
Syrup	0	Value	NATIVE	Gallons	0	1000
SyrupValve	0	Value	NATIVE		0	1000
Water	28	Value	NATIVE	Gallons	0	1000
WaterValve	1	Value	NATIVE		0	1000
Producing	CocoFizz	Value	NATIVE		0	1000
ChurningBlade	1	Value	NATIVE		0	1000
LevelSetPoint	46	Value	NATIVE	Gallons	0	1000

Figure L15.10 – Values for CocoFizz Recipe Item



Name	Value	Mode	Data Type	Unit	Minimum	Maximur
Syrup	66	Value	NATIVE	Gallons	0	1000
SyrupValve	1	Value	NATIVE		0	1000
Water	0	Value	NATIVE	Gallons	0	1000
WaterValve	0	Value	NATIVE		0	1000
Producing	Frappe	Value	NATIVE		0	1000
ChurningBlade	1	Value	NATIVE		0	1000
LevelSetPoint	78	Value	NATIVE	Gallons	0	1000

Figure L15.11 – Values for Frappe Recipe Item

14. Click on "**Apply**" to save the recipe item value set.

15.3. Changing Recipes in GraphWorX64

1. Open the **icTonicsTankMonitor.gdfx** and add a Process Point near the top of the display. In the Data Browser by navigating to Data Connectivity > Modbus > Address Space > icTonics > Tanks > USA > Foxboro > Tank 1

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 **@rcp:icTonicsRecipes.icTonics**

   NOTE: From within the **DataSource** field, you can click on the ellipsis [...] to launch the **Data Browser** and select **Home → Recipes → icTonicsRecipes → IcTonics** and the recipe 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.

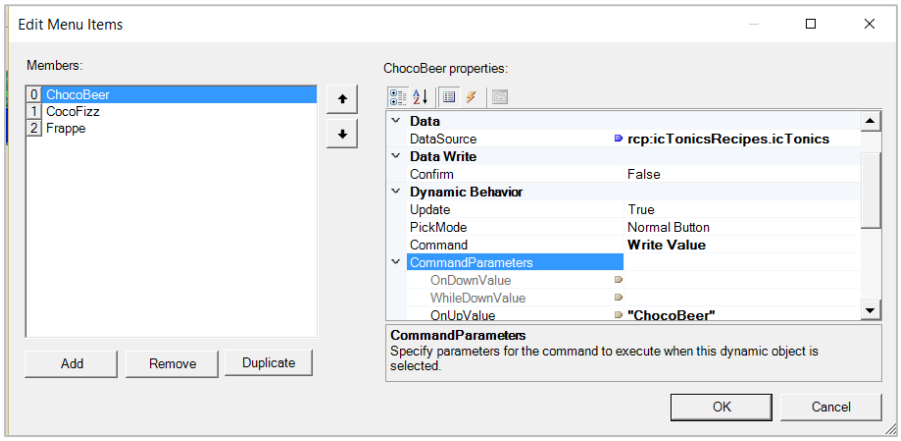


Figure L15.12 - Value Set: Write Value and Menu Text

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 L15.13, and save it.

11. 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.

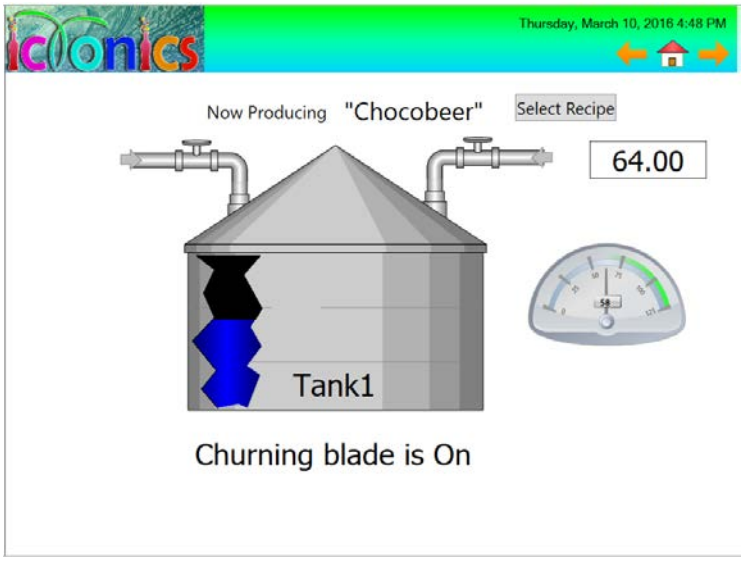


Figure L15.13 - icTonicsTankMonitor Display

15.4. Using Parameters

If you have some time, you can add a parameter to the Recipe. 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 Workbench Project Explorer, go to **Global Aliasing** and right-click on **Themes** and add a theme named "TanksTheme". Once the theme has been added, right-click on TanksTheme and add "Tank1Theme", "Tank2Theme" and "Tank3Theme", as shown below.

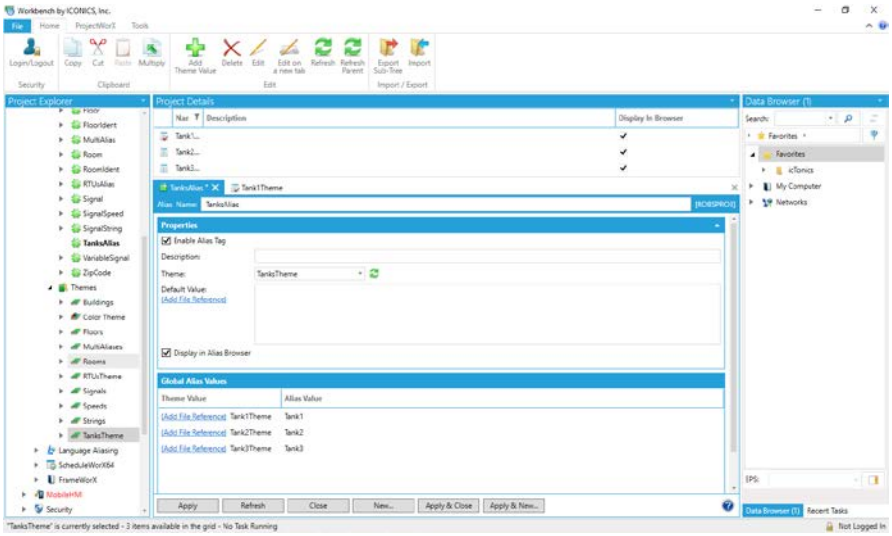


Figure L15.14 – Adding Themes

2. Now go to **Global Aliasing** and right click on **Aliases** and add the new "**TanksAlias**", as shown below.

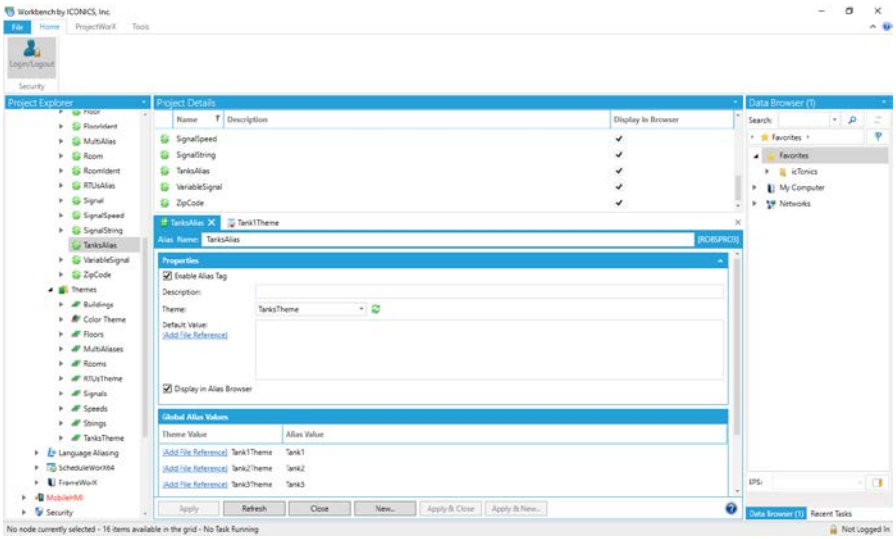


Figure L15.15 – Adding TanksAlias

3. In the **icTonics** recipe value set you previously created, from the **icTonicsRecipe** folder, scroll down to the Tag List, select each OPC tag and change the **Tank1** value to "**<#TanksAlias>**" (without the quotes) for each tag as shown below.

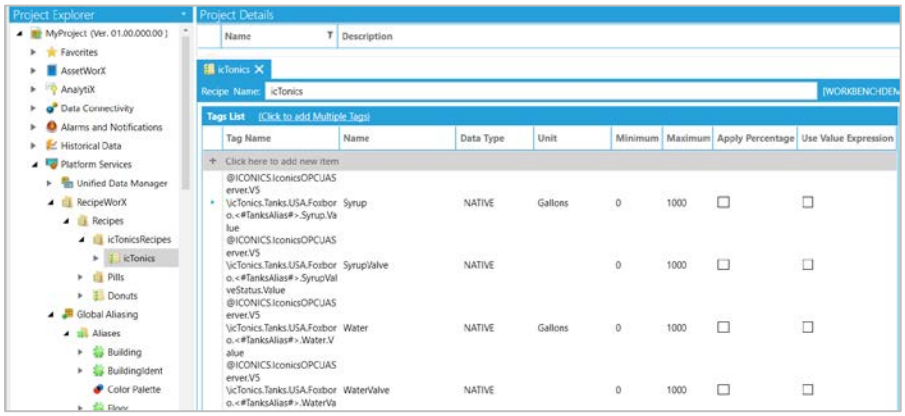


Figure L15.16 - Parameters

4. We will need to change the tags in the GraphWorX64 display to give a parameter value.

5. 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.

6. Change the data source in each of the menu items to include a parameter value. You can either reselect the tag, or simply add "**("TanksAlias")**" (without outside quotes)" in the **DataSource** field. The syntax should look like this:

@\icTonics.Tanks.USA.Foxboro."TanksAlias".Water.Value

7. Try running the display again, it should work just the same as if you did not have parameters.

This concludes this lab.

ScheduleWorX64 16

Schedules are a collection of named events that are linked to the instance of a resource. As an example, a light or group of lights can have a schedule assigned to it (them). A schedule can be a single event or set of events. Events are schedule items and can occur periodically. Schedules are collections of events, which can also be defined to reoccur.

Schedules are linked to a set of tags via valuesets. When a scheduled event begins it reads the current value of the tag, and if necessary writes a new value to it. Depending upon the asset being controlled, valuesets can be a variety of datatypes: booleans, integers, and so forth. Since a schedule can define a set of events, you have fine control over the resource you have scheduled. That is, you can have lights turn on at the beginning of a schedule in the morning, dim during the middle of the day, turn up at dusk, and return to low light conditions when evening turns to night. You could also have the level of the light react to a sensor that measures the light that enters through your skylights and adjusts the lighting appropriately.

You can define holidays that are observed in your country or by your organization. Schedules that are assigned to those holidays override regular weekly schedules you created that would run on that day. However, you can create schedules that are exceptions, and exceptions are imposed in place of any other schedule.

When two or more schedules have events that overlap, a conflict exists. The ScheduleWorX64 Configurator provider in GENESIS64's Workbench allows you to set priorities for each event in a schedule and uses these priorities to determine which schedule's event applies at the time of conflict. A monitor view allows you to see which events are currently in force.

The monitor view also allows you to manually override a scheduled event. This can be a temporary override, which expires after a given amount of time, or it can be manually cleared.

16.1. Objectives

After completing this chapter, you should know:

- How to Use the Workbench Configuration Tool
- How to Access Tools and Providers
- How to Use the Project Explorer

16.2. ScheduleWorX64 Configurator

In ScheduleWorX64, nearly all of the work in creating schedules, attaching those schedules to resources, defining holidays, and setting up exceptions is done within the ScheduleWorX64 Configurator provider user interface (shown below). The notable features are the Main toolbar, the Project Explorer, the Project Details, and main panel where you enter properties for items or view the Calendar.

The General, Weekly, Holidays, Exceptions, Preview and Runtime tabs that you see in the schedule properties window below are shown when a schedule item is selected. Additional tabs may appear depending upon the tab you select.

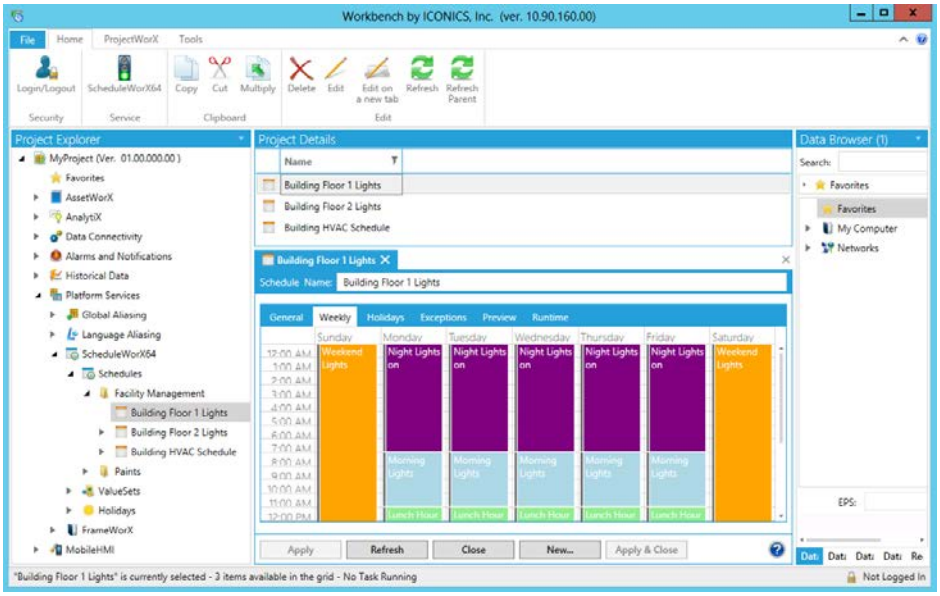


Figure 16.1 - ScheduleWorX64 View

16.3. Date Picker

In the ScheduleWorX64 configurator, the Date Picker control allows you to move quickly through the calendar months and years and select a date of interest to view in the Calendar control.

To View the Date Picker:

1. Click on the Date Picker button; the single month Date Picker appears as shown below.

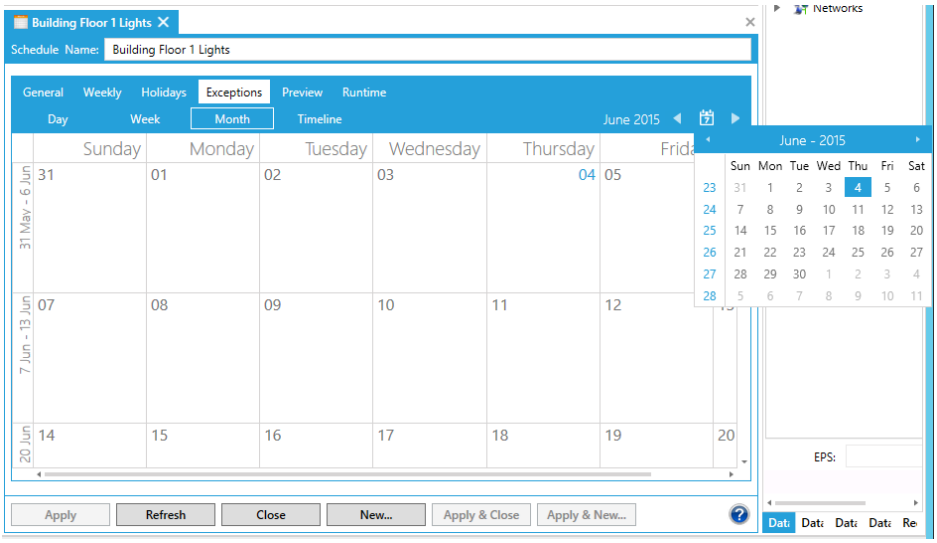


Figure 16.2 - Date Picker Go To Control

2. The Date Picker can be navigated using the following actions:

- To directly **select a date** in view, click on that date.
- To **move back one month**, click on the left facing arrow key.
- To **move forward one month**, click on the right facing arrow key.
- To move quickly through the **several previous or next months**, click on the month name and select your preferred month name from the popup menu. When the data picker title text turns into years, click the year to show groupings of years in order to more quickly move through the calendar.
- To jump to the current date, select “Today” by right-clicking in the date/timescale on the left, as shown in the following image.

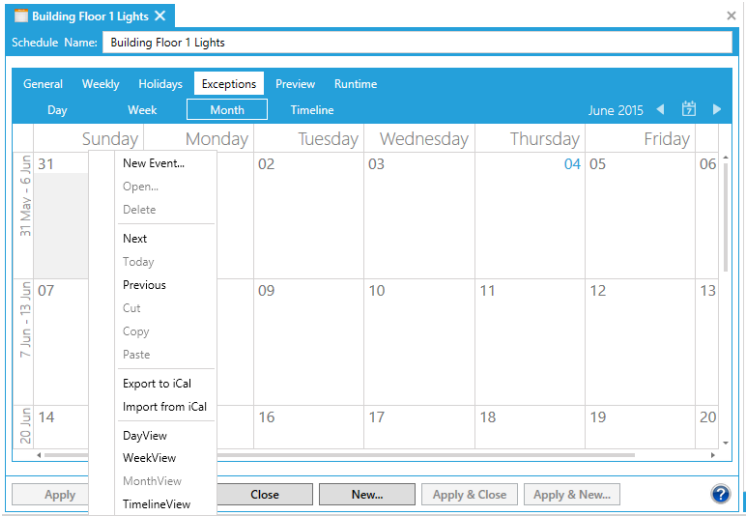


Figure 16.3 - Jumping to Current Date

16.4. Day, Week, Month, and Timeline Views

In the ScheduleWorX64 configurator, the Day, Week, Month, and Timeline Views alter the time period intervals that display in the Calendar control. These tabs (indicated) only appear when you select either the Exceptions or Preview tab.

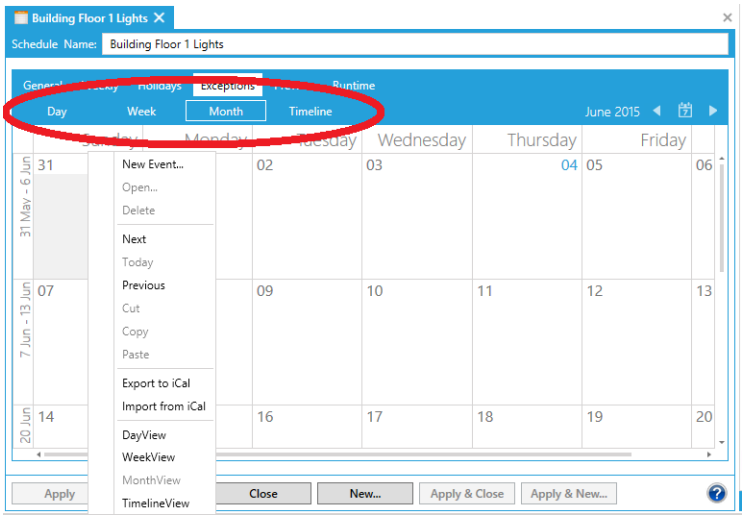


Figure 16.4 - Day, Week, Month, and Timeline View Subtabs

The following items help you select the time period you want to display.

Use the right or left arrows to scroll through the calendar. The calendar increment depends upon the view you select. For example, in Month View you will scroll through months at the rate of one month at a time. A full six weeks will display.

Date Picker. Use the Date Picker to click through the months until you find the date you want to display. That date will appear in the interval (Day, Week. Month, or Timeline) that you selected on your tab settings.

To jump to the current date, select "Today" by right-clicking in the date/timescale on the left, as shown in the following image.

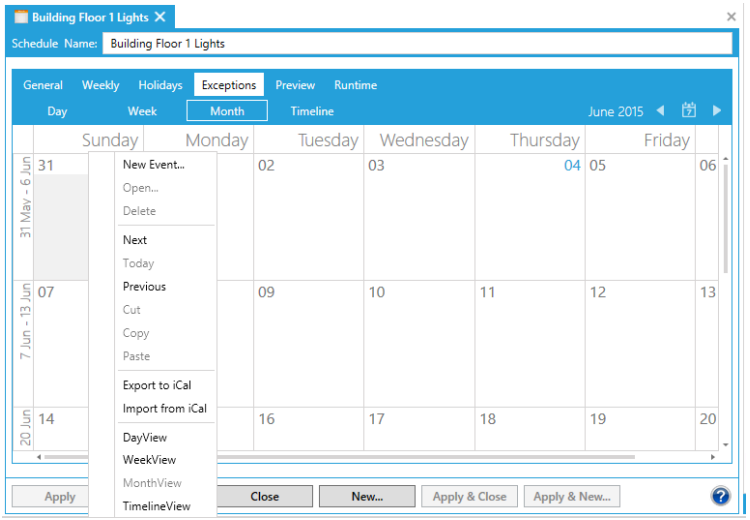


Figure 16.5 - Jumping to Current Date

16.5. Calendar Views

16.5.1. Calendar Views in ScheduleWorX64

You view your calendar by selecting it in the Workbench. The calendar view has the following six tabs:

- **General tab**. Use this tab to name a schedule, set the effective period, and assign ValueSets.
- **Weekly tab**. Use this tab to create, configure and delete weekly recurring events.
- **Holidays**. You can pick a holiday schedule and assign different holiday events for each holiday date. Holiday schedules are defined with the Holidays node of the Tree Explorer. You can see all of your holidays on the Preview tab.
- **Exceptions tab**. Exceptions are scheduled events that override both weekly and holiday events. Exceptions can recur or be defined as one-time events. You can use exceptions to create events that recur on a daily, weekly, monthly, or annual basis.
- **Preview tab.** This tab shows all weekly, holiday, and exception events that are defined for the date(s) in view.

- **Runtime tab**. This tab allows users to view a schedule's Current Status and Next Event, as well as Override and Clear Override.

16.5.2. General Tab for Schedules (ScheduleWorX64)

The General tab of the ScheduleWorX64 Schedule properties lets you set initial schedule information. You can edit information in the following tab sections:

Properties

- Enter a **Description** by filling in the text entry field.
- Click the **Enable Schedule** checkbox to activate or deactivate the schedule from controlling the underlying resources. If enabled, you may optionally set up the Effective Period section of the screen.
- Enter a **Scan Rate** for this schedule (in milliseconds).
- Click the **Refresh Output** checkbox if you would like to refresh the output over a certain amount of time. You can then enter the **Refresh Rate** in seconds into the text entry field (or by using the up and down buttons).

Effective Period

- If you selected Enable Schedule in the Properties section, click the Active checkbox next to the **From:** and/or **To:** pulldown menus to activate your desired time period. If active, the ScheduleWorX64 Runtime Service will stop processing the schedule before the **From** date and after the **To** date.

ValueSet

- Select a **ValueSet Name** from the pulldown menu. The pulldown menu is populated by items currently listed under the Value Sets branch in the ScheduleWorX64 Configurator Tree Control.
- The **Default ValueSet Value** is "Not Assigned". Other values are determined by the ValueSet name.
- Once you have completed your edits to the ScheduleWorX64 Schedule properties, click **Apply** to save your changes.

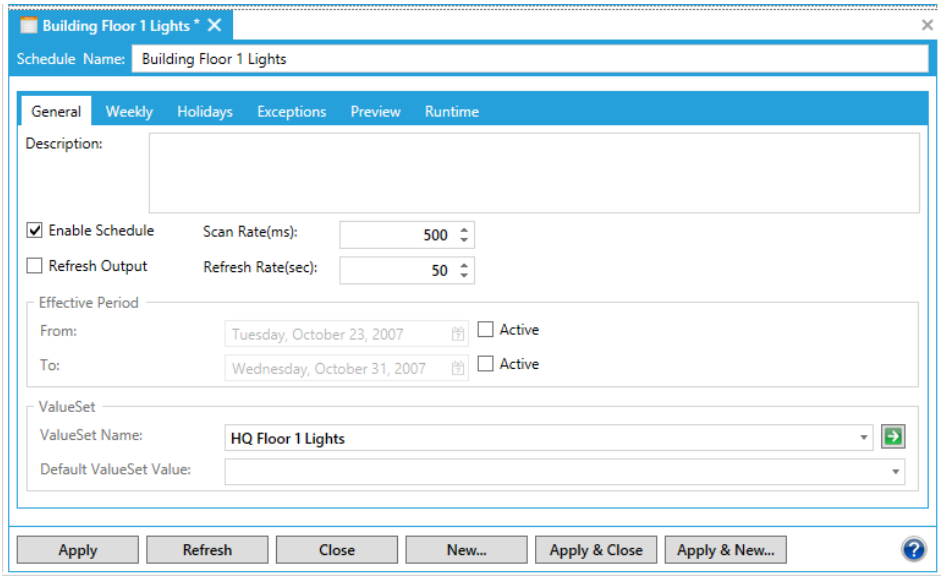


Figure 16.6 - General Tab in ScheduleWorX64 Schedule Properties

16.5.3. Weekly Tab for Schedules (ScheduleWorX64)

The Weekly Tab is used to configure any weekly recurring events. The recurrence pattern will be applied once the event is added to the calendar.

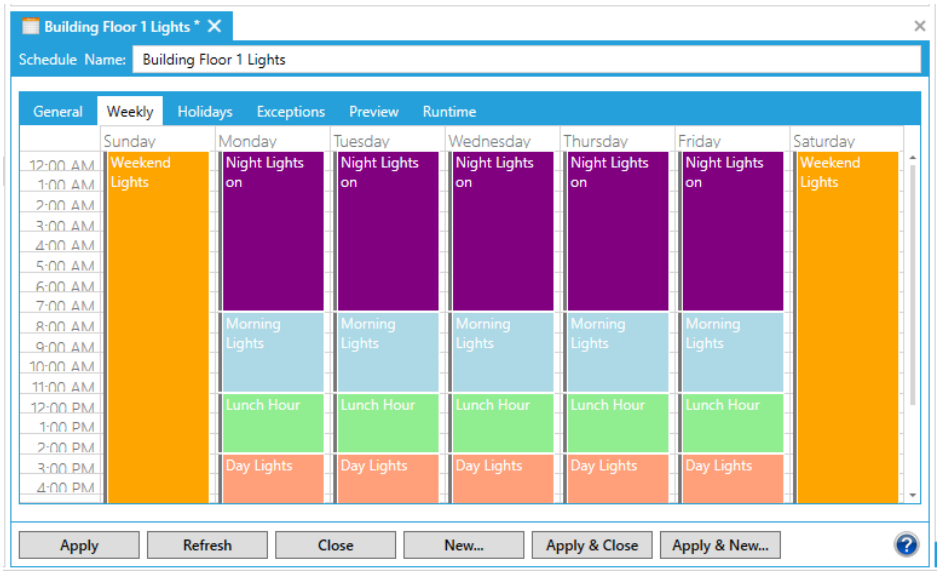


Figure 16.7 - Weekly View

Double-click within your calendar to open the Edit Event dialog box. You can modify an existing event or create a new event.

- **Show As**. Select the background color that will be used for this event on the calendar.

- **Subject**. If a new event, enter a subject. For an existing event, the subject will appear here.
- **Location**. If known, enter the location for the event.
- **Start Time** and **End Time**. Select the starting and ending times for this event. For an all day event, check the **All day event** checkbox.
- **Priority**. Set a priority number for this event. The default is 1.
- **ValueSet Value**. The ValueSet Value is determined by the ValueSet available for this schedule. In the following example, the ValueSet Value "occupied" corresponds to the ValueSet "corporate building lights" (not shown).

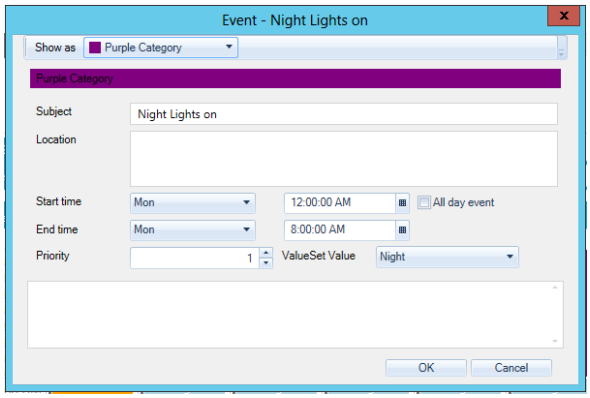


Figure 16.8 - Edit Event Dialog Box

16.5.4. Holidays Tab for Schedules (ScheduleWorX64)

On this tab, you can pick a holiday schedule and assign different holiday events to the different schedules. Holiday schedules are defined with the Holidays node of the ScheduleWorX64 Tree Explorer. After following the instructions for configuring the Holiday Time Settings, you will be able to see configured holidays on the preview tab. Only the events belonging to a Holiday Time Setting will be displayed.

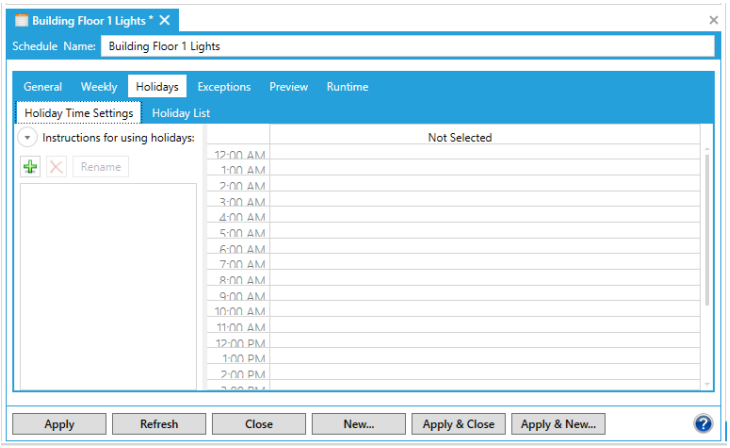


Figure 16.9 - Holiday Time Settings Tab

To Configure a Holiday, Follow These Instructions:

1. Holiday Time Settings Tab Create a new Holiday Time Setting by pressing the Add button, and name it "Day Off". Click **OK**.

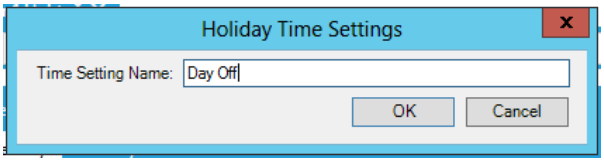


Figure 16.10 - Holiday Time Settings

This setting will be used later on specific holidays.

2. On the Holiday Time Settings tab, highlight those times corresponding to the Holiday Time Setting you just created. For example, for a Day Off setting you might highlight the entire workday from 8:00AM to 6:00PM.

3. Right click to display the New Event box and select it.

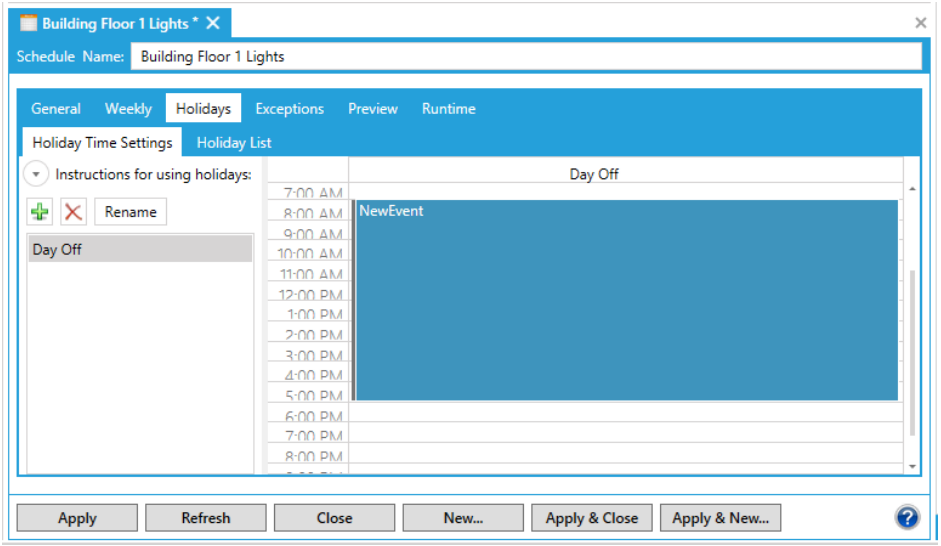


Figure 16.11 - Day Off Time Setting with New Event Selected

4. The Edit Event dialog box is displayed. The Start Time and End Time are pre-filled with the times you selected. Complete the rest of the fields as appropriate.

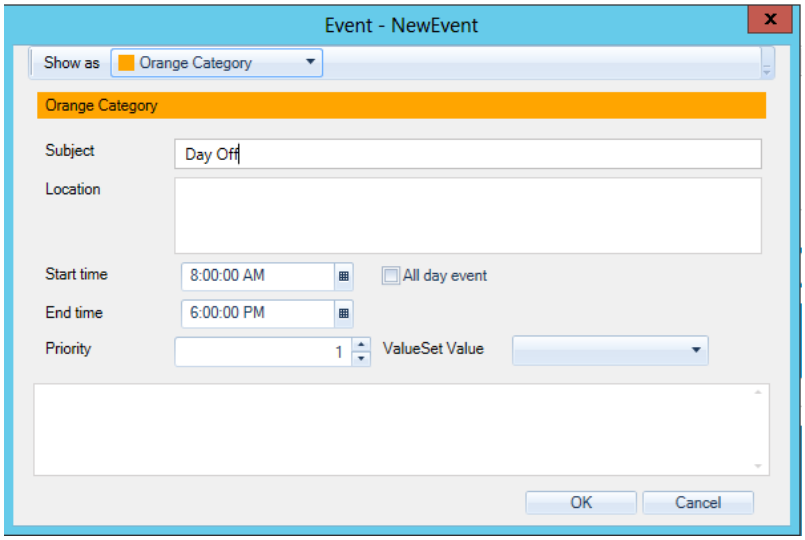


Figure 16.12 - Edit Event Dialog Box

5. Click **OK** to redisplay the Holiday Time Settings Tab. The time settings now contain the information you entered.

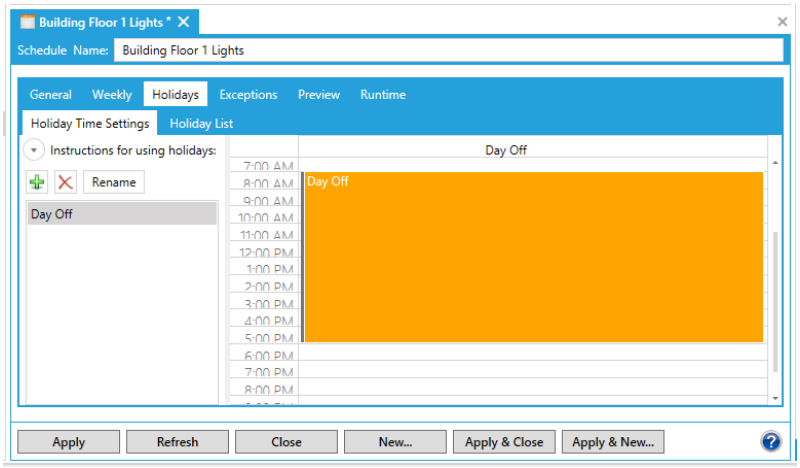


Figure 16.13 - Day Off Defined on Holiday Time Settings

6. Select the Holiday List tab and assign the time setting you just created to the appropriate holidays. In the following example, the Day Off time setting is applied to the Labor Day holiday.

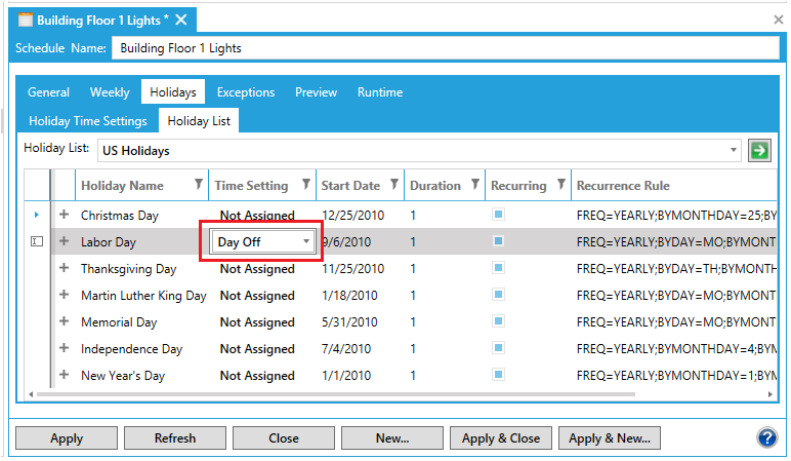


Figure 16.14 - Assignment of Day Off Time Setting

7. Press **Apply** to save your settings.

8. Go to the Preview tab for this day to view the change in your calendar.

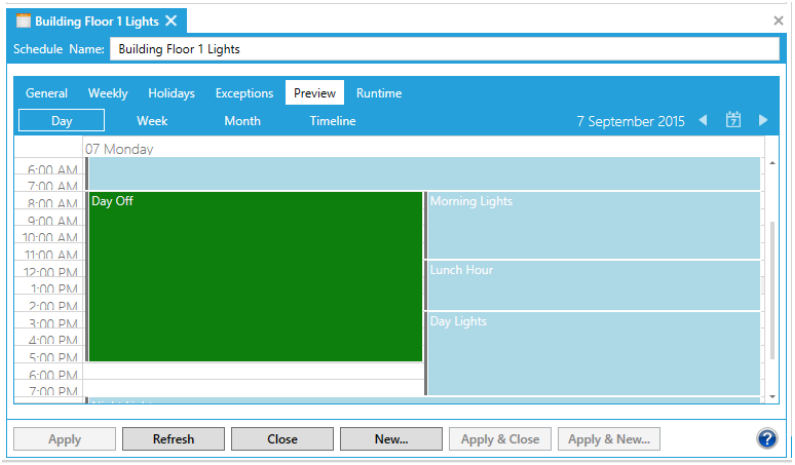


Figure 16.15 - Preview Tab with Calendar Change

16.5.5. Exceptions Tab for Schedules (ScheduleWorX64)

The Exceptions tab can display a daily, weekly, monthly or timeline view calendar. The primary use of the Exceptions tab is to create events that take precedence over previously defined scheduled events and to create recurring events that have periods other than weekly. When an exception takes place at the same time as a scheduled weekly or holiday event, the exception takes precedence. Only an override takes precedence over an exception.

To Create a Recurring Exception, use the Following Instructions:

1. Select the Exceptions tab and select a date for the exception.

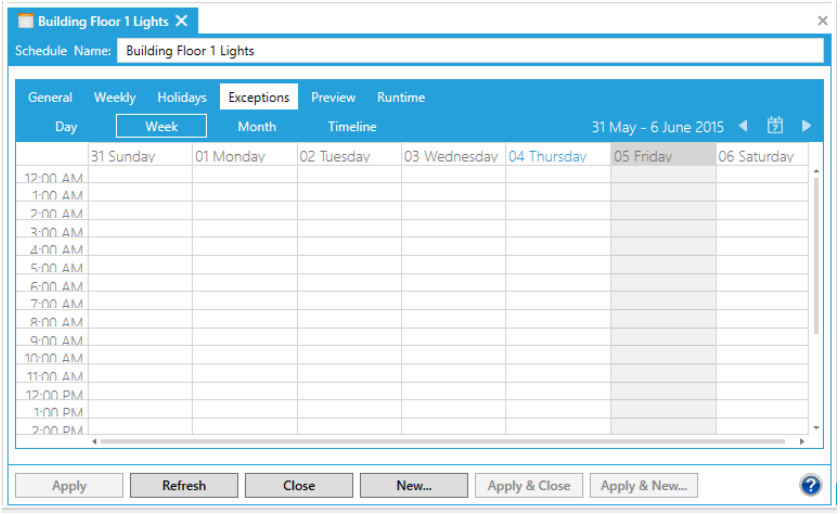


Figure 16.16 - Exceptions Tab with Date Selected in Week View

2. Double-click within your calendar to open the Edit Event dialog box. You can modify an existing event or create a new event.

- **Show As.** Select the background color that will be used for this event on the calendar.
- **Subject**. If a new event, enter a subject. For an existing event, the subject will appear here.
- **Location**. If known, enter the location for the event.
- **Start Time** and **End Time**. Select the starting and ending times for this event. For an all day event, check the **All day** event checkbox.
- **Priority**. Set a Priority number for this event. The default is 1.
- **ValueSet Value.** The ValueSet Value is determined by the ValueSet available for this schedule.

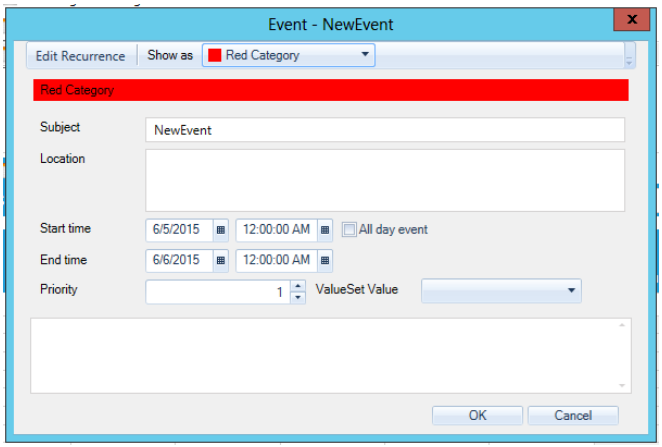


Figure 16.17 - Edit Event Dialog Box

3. To repeat this event, click **Edit Recurrence**. The Edit Recurrence dialog box appears.

4. Complete the fields as described below.

- **Appointment Time**. The Start, End, and Duration are pre-filled with the times from the original event that you selected for the recurrence.
- **Recurrence Pattern**. Select the frequency and the days of the week on which this event should recur.
- **Range of Recurrence**. Select the length for which this event should occur.

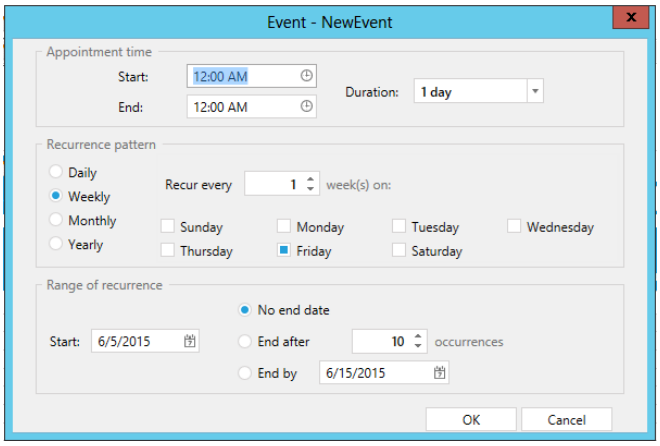


Figure 16.18 - Edit Recurrence Dialog Box

- Click **OK** to accept the information and return to the Edit Event dialog box.
- Next. click **OK** to accept the information and return to the schedule view. The Exception information you entered is now contained in your calendar.

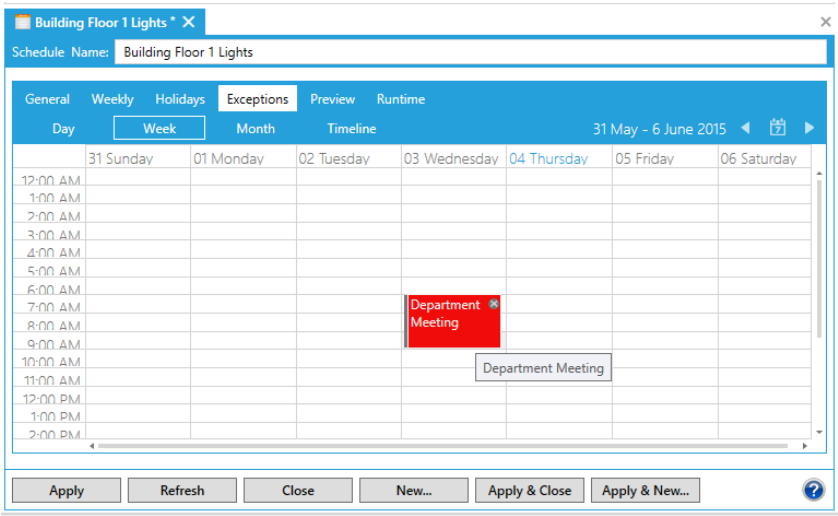


Figure 16.19 - Exception in ScheduleWorX64 Calendar

16.5.6. Preview Tab for Schedules (ScheduleWorX64)

Use the Preview View as the primary calendar display for your users.

The Preview tab displays the Calendar for all possible time periods. You can't configure or create new events in this view, which makes it safe to display for any user viewing the calendar on their monitor. Scheduled events, exceptions, and holidays all appear on this view. The Preview view is shown below in the weekly view.

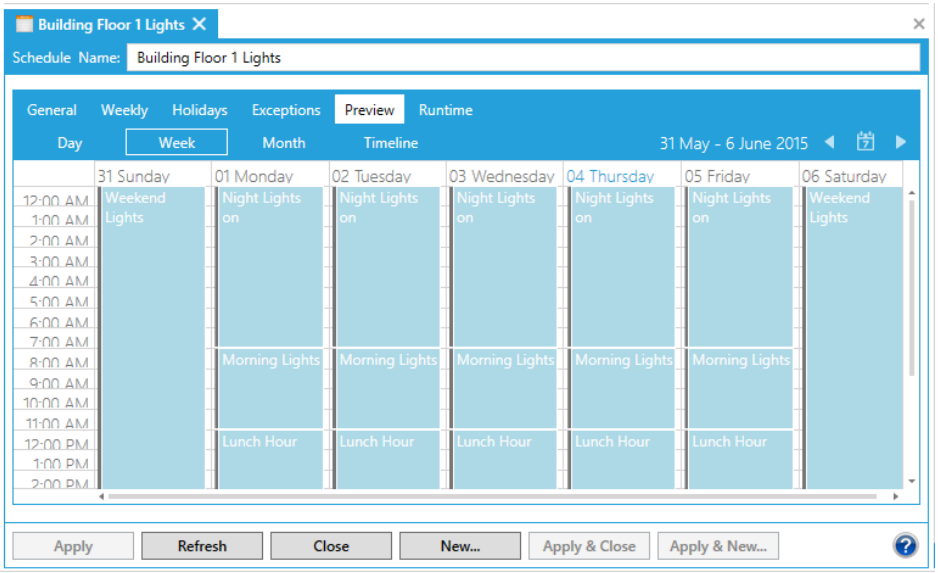


Figure 16.20 - Preview View shown in Weekly Format

16.5.7. Runtime Tab for Schedules (ScheduleWorX64)

The Runtime Tab allows you to view a schedule's Current Status and Next Event. You can also Override and Clear Override.

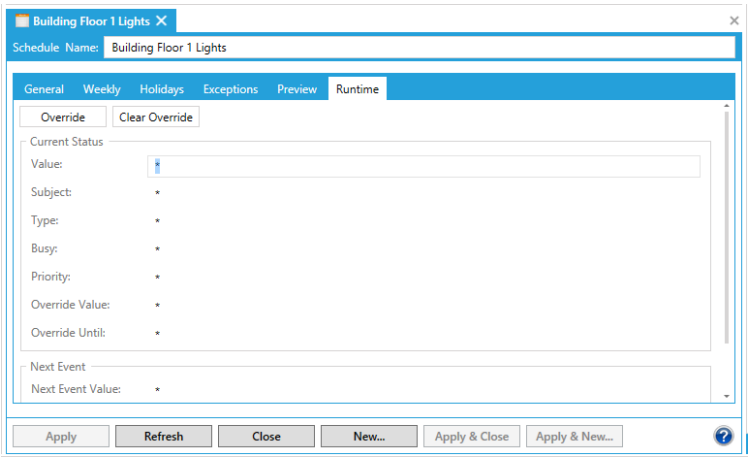


Figure 16.21 - Preview View shown in Weekly Format

The text entry fields in Current Status and Next Event cannot be edited directly within the Runtime tab, but you can use the Override and Clear Override buttons within this tab.

- **Override** - Clicking this button opens the 'Select Value to Override' dialog, shown below. The Value section is populated by the ValueSet Values attributed to the Data source you selected when configuring the schedule. Clicking the 'Temporary' checkbox sets a limited time for the override to take effect (in Seconds/Minutes/Hours/Days/Weeks). Click 'OK' to proceed.

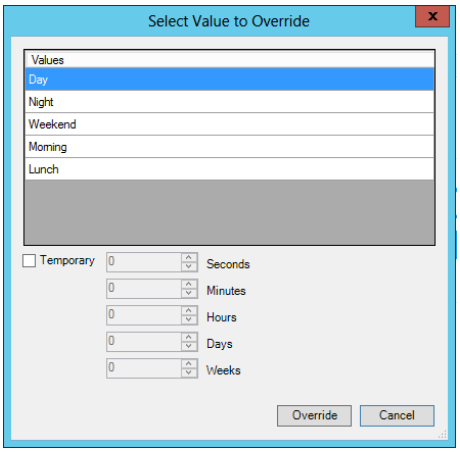


Figure 16.22 - Select Value to Override Window

- **Clear Override** - Clicking this button removes any previously set existing Override data.

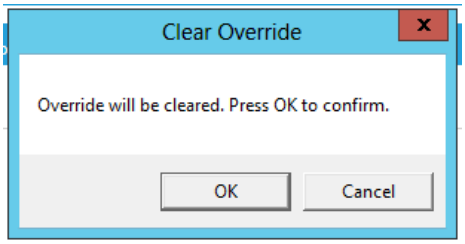


Figure 16.23 - Clear Override Window

16.6. Schedules

16.6.1. Schedules in ScheduleWorX64

In ScheduleWorX64, schedules are container objects for events. A schedule can contain one event or it can contain many events. Schedules exist to control resources, and resources are anything that is controlled by an OPC tag, BACnet tag, or other data source.

A schedule must have a valueset assigned to it, and the valueset in turn must be associated with one or more tags. The values in the valueset control the state or condition that is passed by the event to the data source.

Schedules can be defined so that they have a start time and a stop time; or they can be open ended. Schedules can be duplicated and modified, but the name for a schedule must be unique in the folder it is in. There are no program limits to the number of schedules you can define, or the number of events in a schedule.

You can import and export schedules.

16.6.2. Creating a Schedule

The ScheduleWorX64 Configurator in the Workbench lets you create new schedules and folders, either through the Project Explorer or top toolbar. The best way to manage schedules is to create a folder for each set of resources you wish to manage. Then, you can place all of the schedules that you create for that resource in that folder. This organizational scheme is the best way to view your schedules as a group or to determine which of the schedules you have defined is running and which are not.

To Create a Schedule

1. Right-click on the **Schedules** node and click on **Add Schedule**, as shown below.

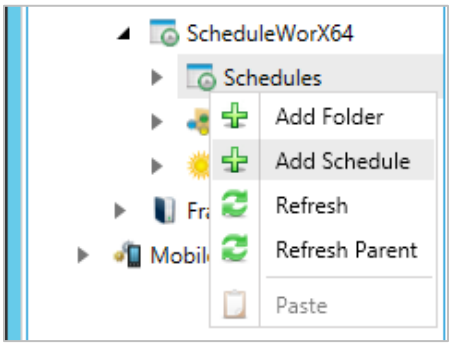


Figure 16.24 - New Schedule in ScheduleWorX64 Tree Explorer

-OR-

Select the **Schedules** node then click on the **Add Schedule** button, shown below, in the Edit section of the Home ribbon in the Workbench.

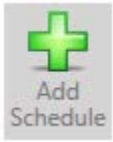


Figure 16.25 - Add Schedule Button

2. The **Schedule** properties displays, showing the default General tab.

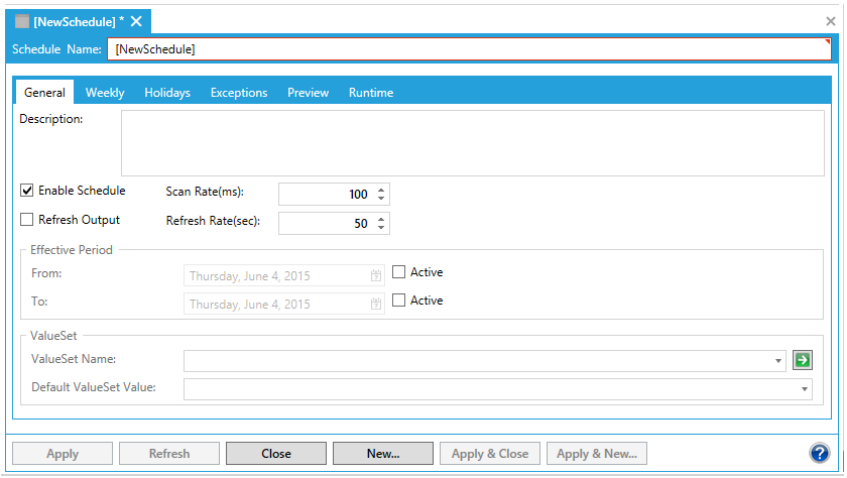


Figure 16.26 - Schedule Properties Window in Workbench

The ScheduleWorX64 Schedule properties are split over the following six (6) tabs:

- General Tab
- Weekly Tab
- Holidays Tab
- Exceptions Tab
- Preview Tab
- Runtime Tab

16.6.3. Creating a Schedule Folder in ScheduleWorX64

In ScheduleWorX64, schedules are organized into folders, so that a set of schedules for similar resources can be grouped together and more easily found. You can create a folder at any level in the Schedule tree. You can also drag and drop a folder in the tree to move a folder within the hierarchy. Folder names are editable in the Name text box in the Details pane of the Schedule window (lower right pane).

To Create a Schedule Folder

1. Right-click on the **Schedules** node and click on **Add Folder**.

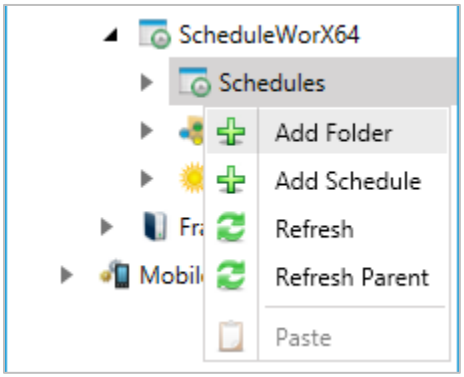


Figure 16.27 - New Schedule Folder in ScheduleWorX64 Project Explorer

-OR-

Select the **Schedules** node then click on the **Add Folder** button, shown below, in the Edit section of the Home ribbon in the Workbench.

Figure 16.28 - Add Folder Button

2. The Folder properties window appears.

3. Enter a name for the Schedule Folder and click Apply to save your changes.

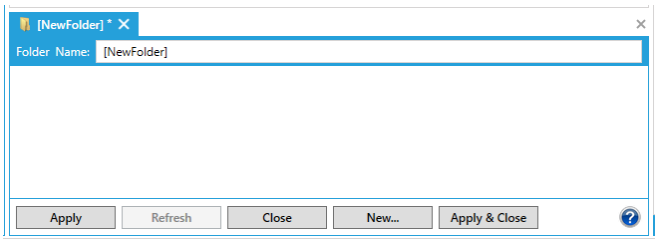


Figure 16.29 - ScheduleWorX64 Schedule Properties Window in the Workbench

16.6.4. Schedule Configuration (ScheduleWorX64)

Schedule configuration is the main control used to create and assign properties to schedules in ScheduleWorX64. When you select a different type of item from the ScheduleWorX64 Tree Explorer, such as a ValueSet or Holiday, you see a different set of property tabs.

For schedules, the configurator displays the following tabs:

- **General Tab** - Use the General tab to name a schedule, set the effective period, assign a ValueSet, and set a default value. The General tab is the only tab that is required to save a schedule to the database.

The next tabs allow you to create events that will be part of your schedule.

- **Weekly Tab** - Create, configure and delete weekly events on this tab. Events defined on this tab repeat on a weekly basis.
- **Holidays Tab** - You can pick a holiday schedule and assign different events to the different schedules. Holiday lists and the holidays they contain are defined with the Holiday node of the Tree Explorer.
- **Exceptions Tab** - Exceptions are scheduled events that override both weekly and holiday events. Exceptions can recur or be defined as a onetime event.

The next tab is a read-only view of all events:

- **Preview Tab** - This tab shows all weekly events, holidays, and exclusions that were defined for the date(s) in view. You can print your full schedule from this tab, but you cannot create new events on this tab.

The remaining tab provides additional information about the schedules as well as other runtime-based actions.

- **Runtime Tab** - This tab allows users to view a schedule's Current Status and Next Event, as well as Override and Clear Override.

16.6.5. Enabling and Disabling Schedules (ScheduleWorX64)

ScheduleWorX64 schedules can be set so that they run during their effective time and are thus enabled, or they can be disabled and the scheduled events do not run.

To Enable or Disable a Schedule and Set an Effective Period

1. Click on a **Schedule Name**. (Associated schedules can be found in the right-side window of the Workbench once a containing folder is clicked.)

2. In the **Schedule** detail pane, click on the **General** tab if it is not already selected.

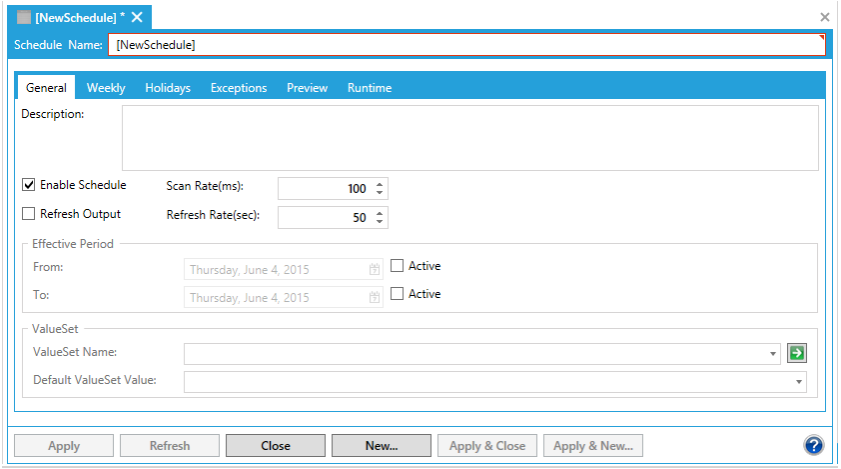


Figure 16.30 - General Tab for Selected Schedule

3. Click the **Enable Schedule** checkbox if you want the schedule to be active and events to be applied. Uncheck the Enable Schedule checkbox if you want to disable the schedule. In the following example, the "Building Floor 1 Lights" schedule is enabled, as the 'Enable Schedule' checkbox has been ticked.

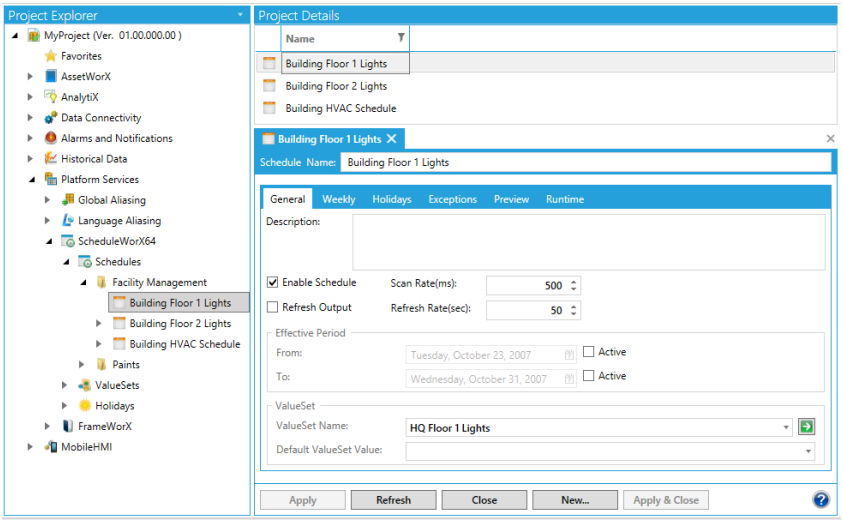


Figure 16.31 - Enabled and Disabled Schedules in the Project Explorer

4. You can set an **Effective Period** bounding condition so that an enabled schedule becomes active on a certain date or so that a schedule ceases to be active after a certain date. If these options aren't set, then there is no bounding condition and an enabled schedule will be applied to start immediately or not to end.

To Set a Schedule to Start on a Particular Day

1. Click the checkbox next to **Active** for the **From** date. Select the pulldown box to set a date. Once the checkbox has been selected, click the **From** pulldown box and select a date or use the up and down arrows to do so.

2. Click **Apply** to activate the new Effective Period.

Figure 16.32 - Setting a Schedule to Start on a Set Day

To Set a Schedule to End on a Particular Day

1. Click the checkbox next to **Active** for the **To** date. Select the pulldown box to set a date. Once the checkbox has been selected, click the **To** pulldown box and select a date or click on the up and down arrows to do so.

2. Click **Apply** to activate the new Effective Period.

Figure 16.33 - Setting a Schedule to End on a Set Day

When both Active checkboxes are enabled, the **From:** and **To:** dates define a range during which time the schedule runs (at runtime).

When both Active checkboxes are disabled, the **From:** and **To:** dates are ignored and the schedule runs whenever ScheduleWorX64 is in runtime.

16.6.6. Editing a Schedule (ScheduleWorX64)

When you create a ScheduleWorX64 schedule and assign various properties to it (such as a holiday list, specific and recurring events, etc.), you create the schedule when you click on the **Apply** button. The Apply button acts as a Save command. Therefore, editing a schedule is an almost identical procedure.

To Edit a Schedule:

1. Click on the schedule in the ScheduleWorX64 Configurator provider that you wish to edit. (Associated schedules can be found in the right-side window of the Workbench once a containing folder is clicked.)

2. On the General, Weekly, Holidays, and Exceptions tabs, alter the properties and events that you want to be part of your schedule.

3. Click Apply to save your changes.

In the following example, building lights are shut off at a headquarters building for a holiday weekend.

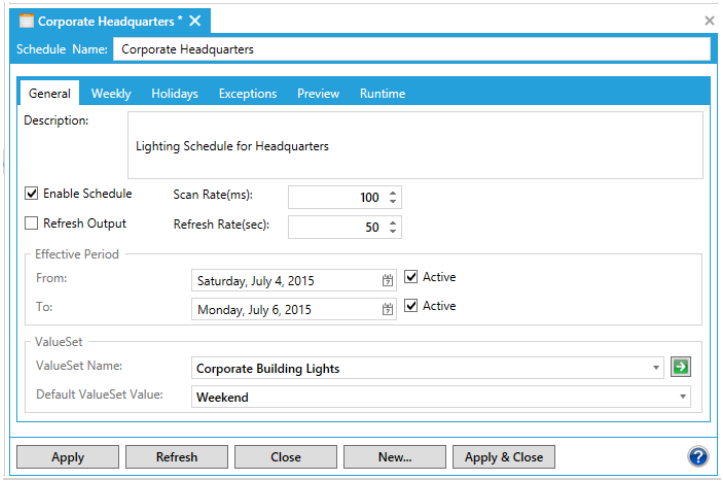


Figure 16.34 - Schedule Properties

16.6.7. Editing the Override List in ScheduleWorX64

In ScheduleWorX64 schedules, overrides take precedence over all other types of events. The Override List displays all of your schedules, gives you the current status of existing overrides, and the next event for that schedule. The list also lets you edit existing override information.

To Open and Change Information in the Override List:

1. Click on a Schedule in the ScheduleWorkX64 Configurator Provider. (Associated schedules can be found in the right-side window of the Workbench once a containing folder is clicked.)

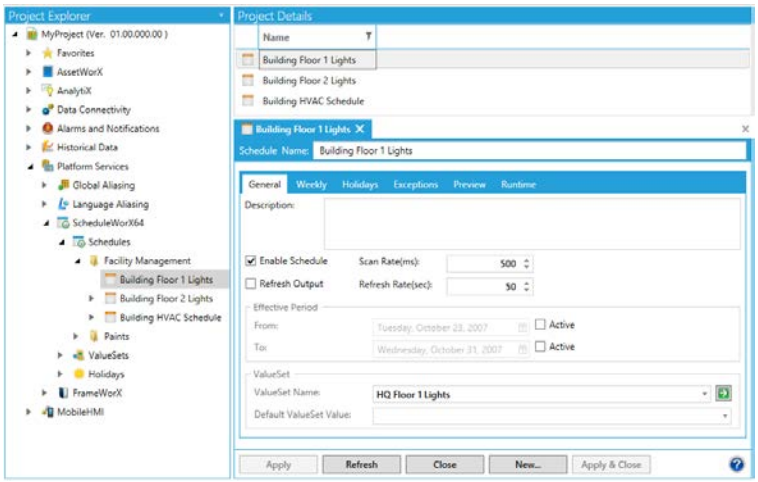


Figure 16.35 - Schedule Selected

2. Click on the **Runtime** tab. Notice the Override and Clear Override buttons at the top of the window. You will also see the Current Status and Next Event sections.

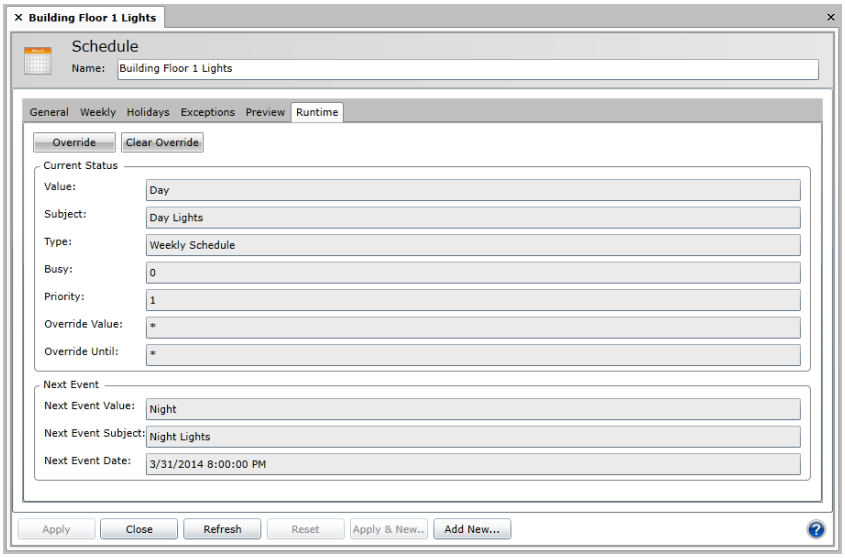


Figure 16.36 - ScheduleWorX64 Schedules Override Office List

3. This step describes the two options available to you:

- **Override - Click** on the **Override** button to open the **Select Value to Override** dialog box. Set the Override Value. This value corresponds to one of your Valueset values. If the override is temporary, click the 'Temporary' checkbox, then schedule a length of time that the override will take place (in Seconds/Minutes/Hours/Days). Click **OK** to accept the override. You will then notice that the Current Status information in the Runtime tab of the Schedule updates accordingly.

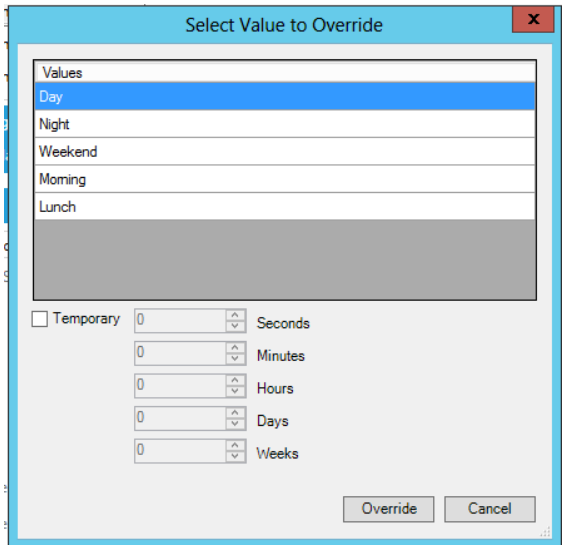


Figure 16.37 - Override Schedule Dialog Box

- **Clear Override** - Click this button to clear the existing Override. You will see the following confirmation window. Click 'OK' to proceed. You will then notice that the Current Status information in the Runtime tab of the Schedule updates accordingly.

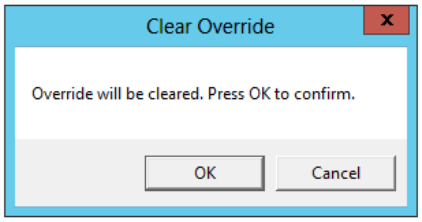


Figure 16.38 - Clear Override Confirmation

16.7. Holidays

16.7.1. Holidays and Special Events in ScheduleWorX64

In ScheduleWorX64, holidays and other special events that are handled in the same manner are events that take precedence over regularly scheduled events. Holiday events can occur once or be recurring. You can control the time period for the holiday. A holiday can be a full day, half day or just an hour, as you require.

Holidays can recur. They can be set to recur on a daily, weekly, monthly or yearly basis. They can recur on any day of a month (e.g., the fourth Thursday in November) or any date in a month (July 4th).

Although holidays can recur on a daily or weekly basis, such events are better defined as weekly events, since that is the lowest priority.

16.7.2. Holiday Lists

Holidays are organized into lists. Lists are usually country specific, but they can contain whatever you want. You can create a holiday list or import holidays into a holiday list. Once the holiday list is created, you can apply that list to a schedule. A schedule can only have one holiday list associated with it.

To Create a New Holiday List:

1. Right-click on the **Holidays** node of the ScheduleWorX64 Configurator provider, then on **Add Holiday**.

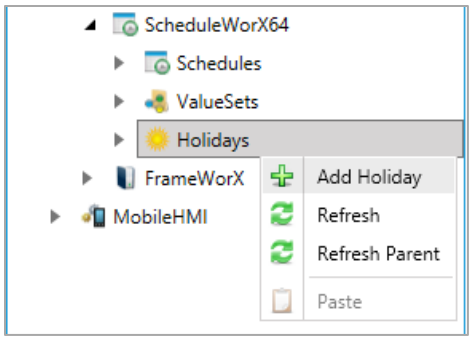


Figure 16.39 - Creating a Holiday from the ScheduleWorX64 Configurator Tree Explorer

-OR-

Click on the **Holidays** branch of the ScheduleWorX64 Configurator Provider and then on the **Add Holiday** button, shown below in the Edit section of the Home Ribbon in the Workbench.

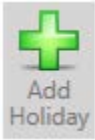


Figure 16.40 - Add Holiday Button

2. The ScheduleWorX64 Holiday properties dialog box appears.

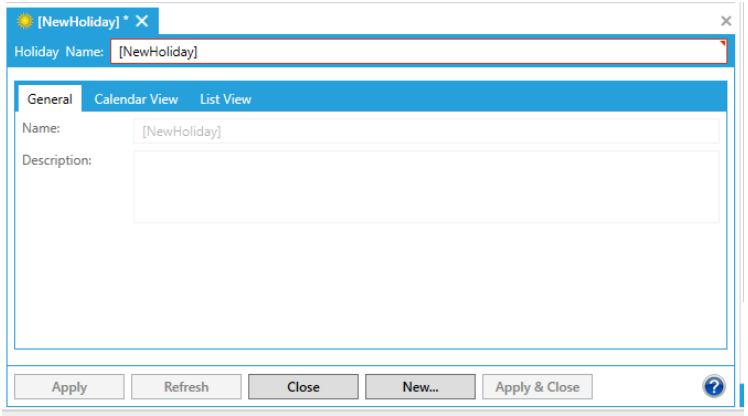


Figure 16.41 - ScheduleWorX64 Holiday Properties

The ScheduleWorX64 Holiday Properties consists of three separate tabs.

- General Tab
- Calendar View Tab
- List View Tab

General Tab for Holidays in ScheduleWorX64

The General tab of the ScheduleWorX64 Configurator Holiday properties lets you enter the Holiday List Name and Description.

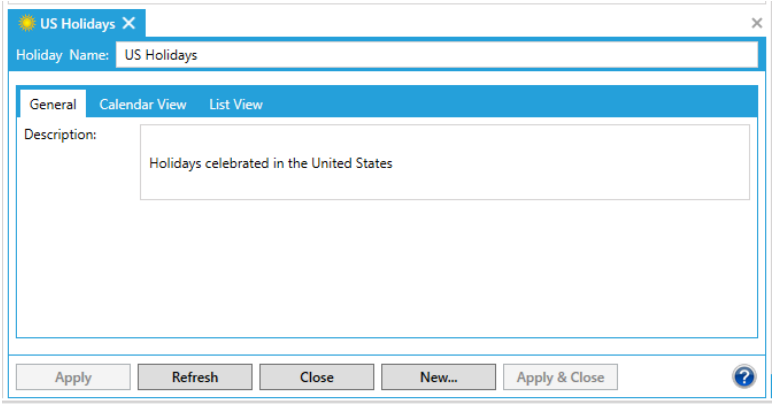


Figure 16.42 - ScheduleWorX64 Holiday Properties - General Tab

Click on **Apply** to save your changes. Click on another tab to continue editing the holiday properties.

16.7.3. Calendar View Tab for Holidays in ScheduleWorX64

The Calendar View tab of the ScheduleWorX64 Holiday properties allows you to set holidays by adding separate events within the calendar.

To add a new event, right-click on a selected date and select New Event.

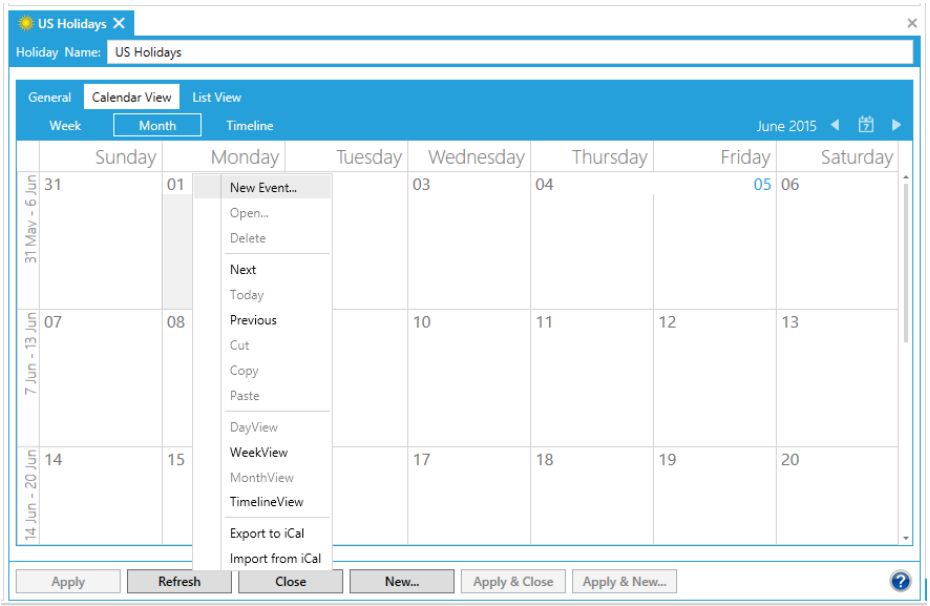


Figure 16.43 - Add New Event Function in the Calendar View Tab for a Holiday

The Edit Event dialog box appears. Enter the Subject and Location in the text entry fields. Use the 'Show As' pulldown menu to select a color (for instance, use one color for a shared type of events or different colors to distinguish individual event types). Enter a Start and End time by entering text or using the date/time picker buttons [ ]. The All day event checkbox will be initially pre-checked and grayed out.

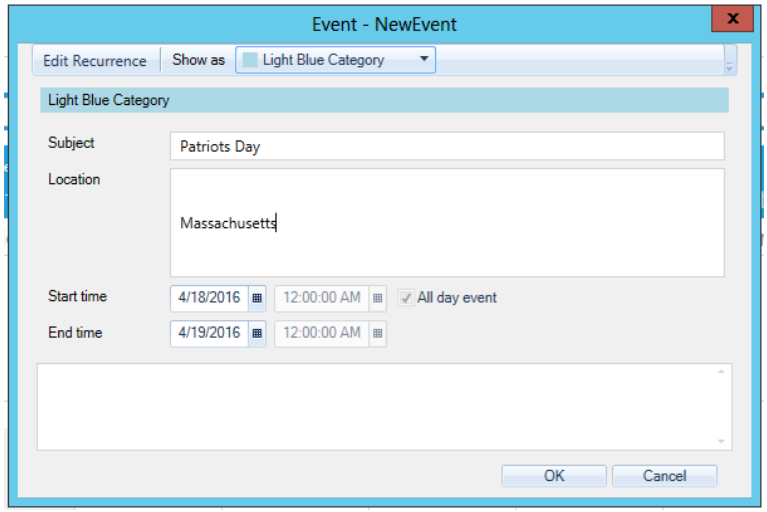


Figure 16.44 - Edit Event Dialog Box

If the event type is recurring, click 'Edit Recurrence'. If the event is not recurring, click OK to return to the main Calendar View tab settings in the ScheduleWorX64 Holiday Properties. If you click 'Edit Recurrence', the Event Recurrence dialog box appears. The Event Time will be initially grayed out for a full-day holiday event.

To Enter a Recurring Holiday:

1. From the Edit Event dialog box, click on **Edit Recurrence**. The dialog box shown below appears.
2. Enter the Recurrence Pattern (Daily, Weekly, Monthly, Yearly and fill in the appropriate "Every" or "The Day of Month").
3. Enter the Range of recurrence by entering a Start date in the text entry field (or using the up and down arrows) and the end date by number of occurrences or actual date (or select No end date).
4. Click **OK** to save your changes and return to the Edit Event dialog box, then click **OK** again to return to the main Calendar View tab settings in the ScheduleWorX64 Holiday Properties.

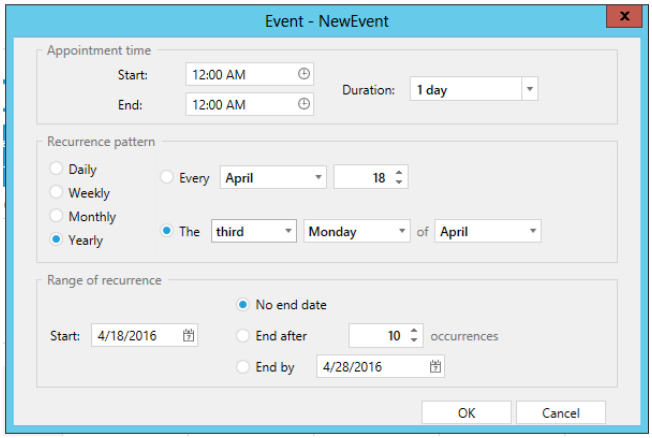


Figure 16.45 - Edit Recurrence Dialog Box

5. The new holiday entry appears in your calendar. Click **Apply** to save your changes.

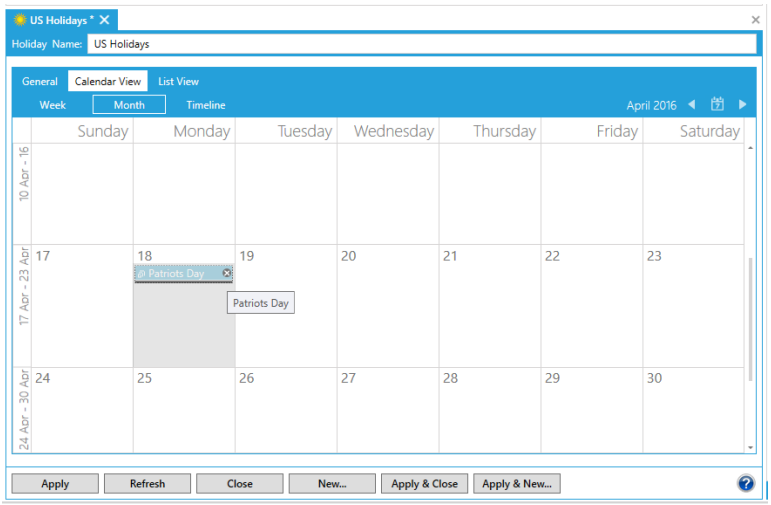


Figure 16.46 - Monthly Holiday Calendar with Holiday Added

16.7.4. List View Tab for Holidays in ScheduleWorX64

The List View tab for holidays in ScheduleWorX64 provides a list of all the items associated with a holiday list. You cannot add, change or delete any holiday information on the List View tab.

You can change the name of the Holiday List by editing the Name field and clicking **Apply**.

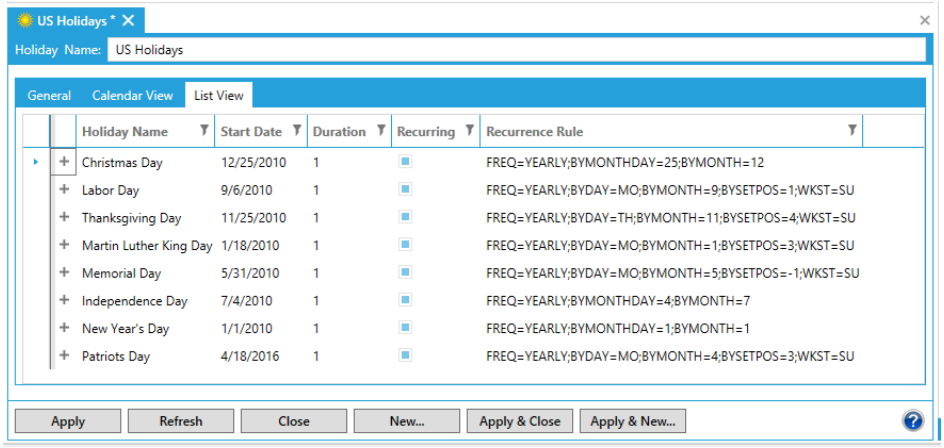


Figure 16.47 - List View Tab with US Holidays

16.7.5. Creating a Holiday in ScheduleWorX64

You can create holidays within existing Holiday Lists in the ScheduleWorX64 Configurator provider. With a holiday list selected, the Calendar control allows you to define holidays that occur once or are recurring.

Holidays can be for one day or half a day (morning or afternoon), or can take place during some other part of the day. The ScheduleWorX64 Configurator in the Workbench gives you granular control over the time that a holiday event occurs.

Info

NOTE: After creating a holiday, you must apply the holiday to a schedule in order for users to see it through the Preview tab in a ScheduleWorX64 schedule. For further information, see the "Applying a Holiday List to a Schedule" topic.

To Create a Holiday Event:

1. Click on the **Holidays** node in the ScheduleWorX64 provider. You will see a list of pre-filled holidays in the Project Details, as shown below. Click on one of the holiday lists or you can create your own holiday list.

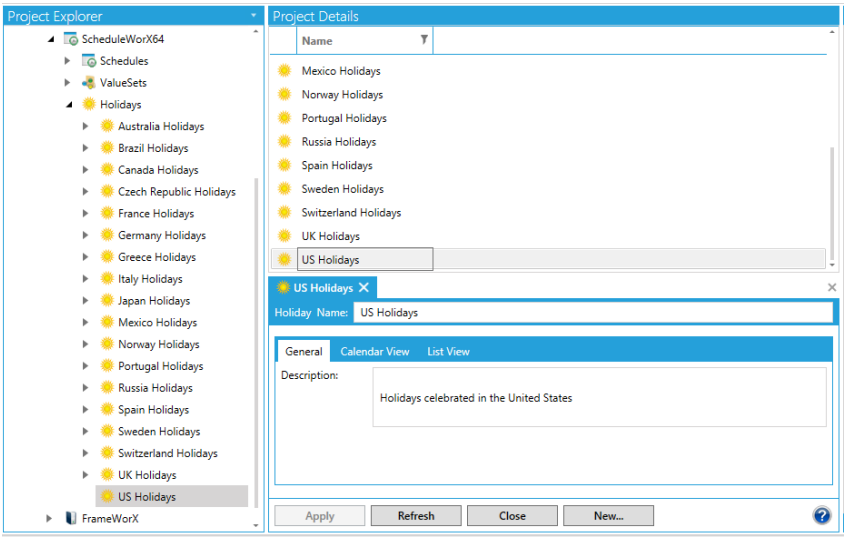


Figure 16.48 - US Holidays List Selected in ScheduleWorX64 Configurator

2. Click on the **Calendar View** tab.

3. Using the calendar tools, select the **date** that will become a holiday on this holiday list. You can use the Date Picker icon on the toolbar to move quickly through the calendar.

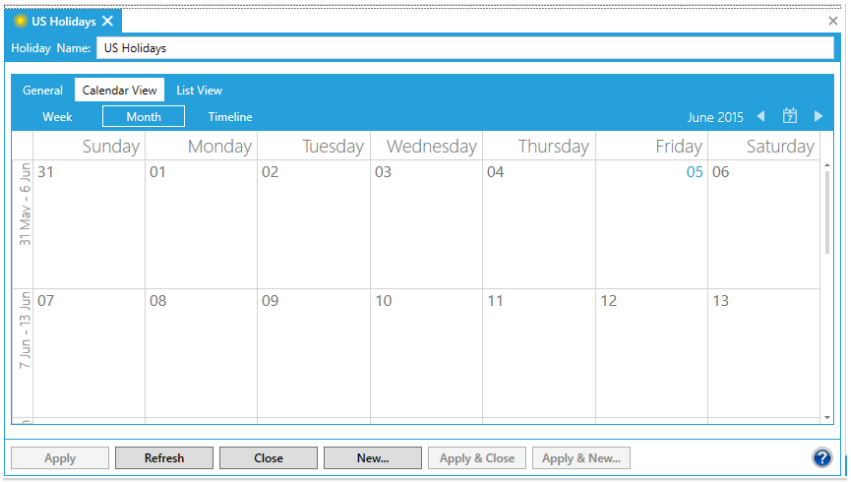


Figure 16.49 - Selection of Holiday Date

4. Double-click on the calendar view of your selected date (or right-click, then click on **New Event**. The 'Event - New Event' dialog box appears.

5. Complete the fields in the **Edit Event** dialog box.

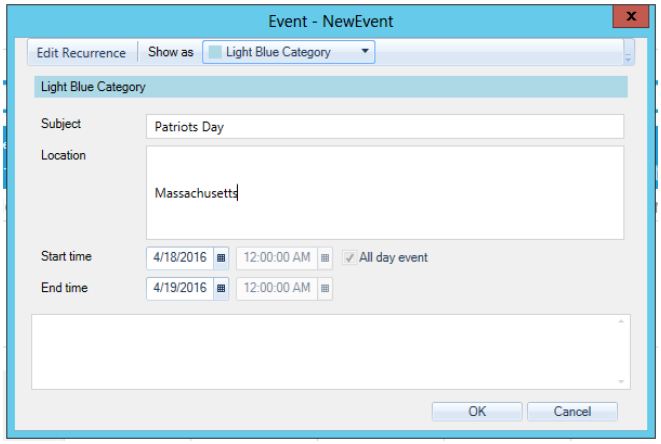


Figure 16.50 - Edit Event Dialog Box with Holiday Added

6. Click **OK** to add the holiday to your calendar.

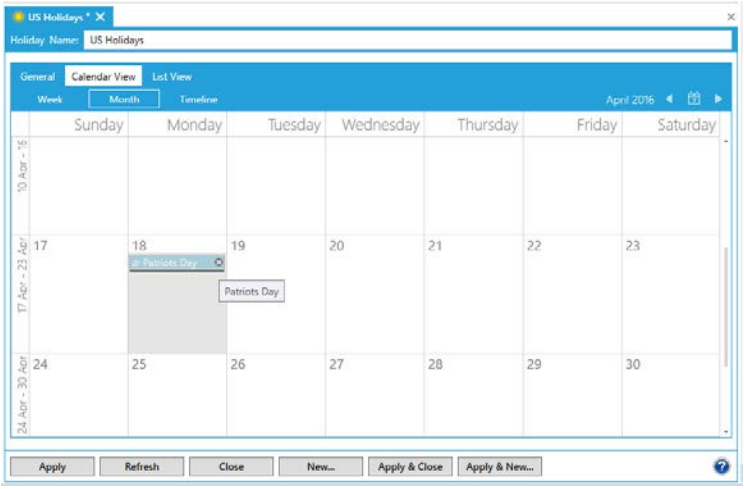


Figure 16.51 - Patriots Day Holiday Added to Calendar

When you create a holiday and apply it to a holiday list, that holiday is added to the schedule of every schedule that is associated with this holiday list. Should you wish to suppress a holiday, define an exception that occurs on the same date.

16.7.6. Applying a Holiday List to a ScheduleWorX64 Schedule

Schedules can have a holiday list applied to them. When you assign a holiday list to a schedule, the holidays on that list will take precedence over any schedule events that are already defined. However, exceptions and overrides still take precedence over holidays.

Info

NOTE: Before continuing with this step, you must have already created the holiday(s) you want to assign to your schedule. For further information see Creating a Holiday.

To Assign a Holiday List to a Schedule:

1. In the ScheduleWorX64 Configurator, double-click on the schedule to which you want to attach a holiday list. This brings up the ScheduleWorX64 Schedule settings.

2. Click on the Holidays tab and then on the Holiday List sub-tab and then on the **Holiday List** pulldown menu.

3. Click on the Holiday List you desire.

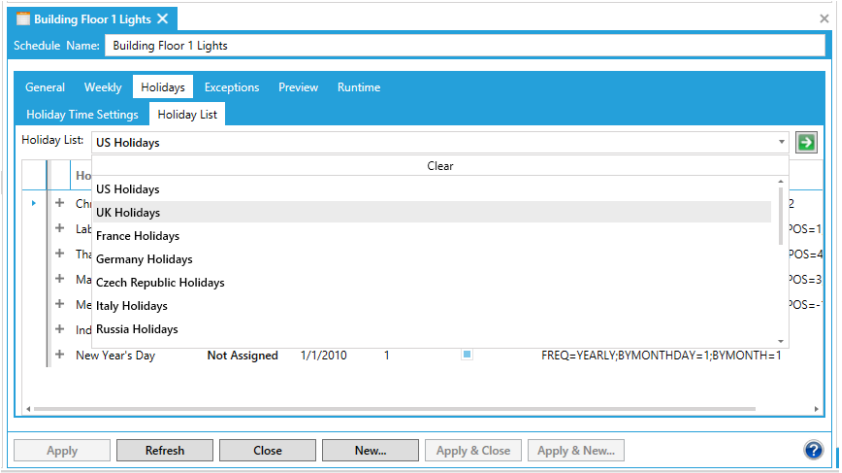


Figure 16.52 - Selecting a Holiday List in the Holidays Tab/Holiday List Sub-tab

4. To assign a Holiday Time Setting to a holiday in the list, select your preferred Holiday Time Setting (in this case, 'Closed for Holiday') from the Time Settings drop down list.

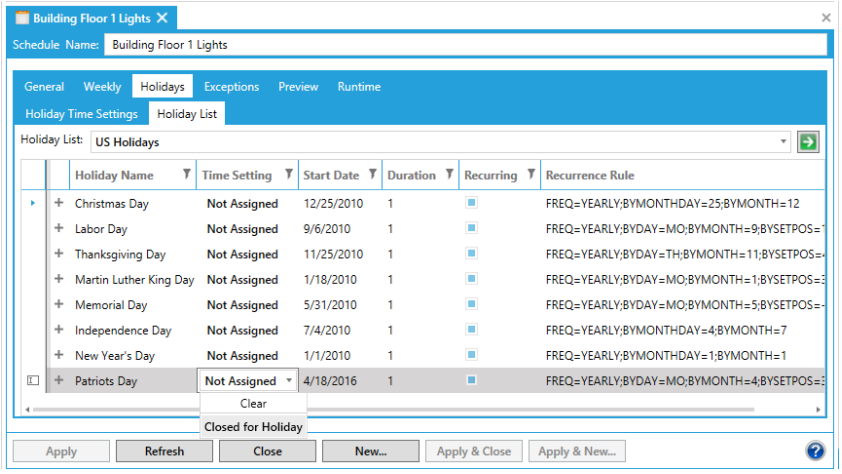


Figure 16.53 - Assigning a Closed for Holiday Time Setting to Patriots Day

5. Click **Apply** to save your selection. You can continue to make changes within the other tabs of the ScheduleWorX64 settings.

16.7.7. Holiday Time Settings in ScheduleWorX64

In ScheduleWorX64, when you define a holiday that is part of a holiday list on the calendar, the holiday is defined as a full-day event. You can create other types of holidays on the Holidays Time Settings tab below the Holidays tab.

To Specify a Specific Time of Day that a Holiday Event Begins and Ends

1. Double-click on a schedule in the ScheduleWorX64 Configurator provider that contains the holiday(s) you wish to modify.

2. Click on the **Holidays** tab and then click on the **Holiday Time Settings** sub-tab.

3. Click the button. The **Holiday Time Settings** dialog box will appear. Enter a name for the holiday time setting that you want to create.

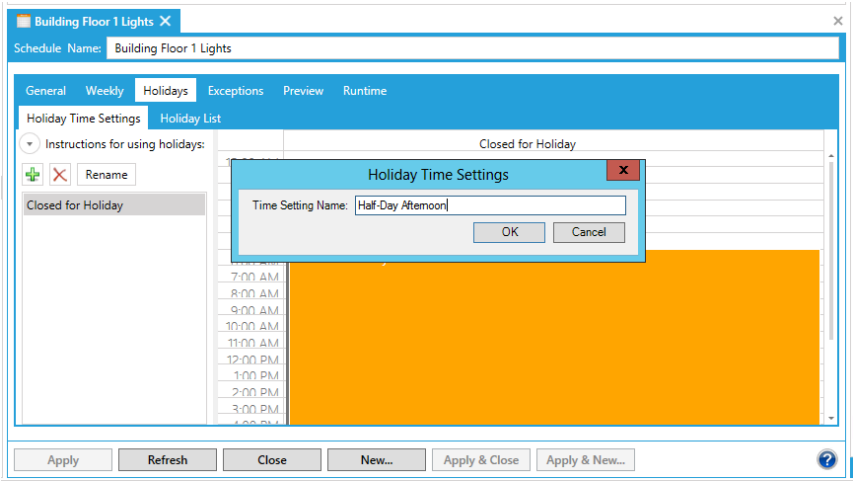


Figure 16.54 - Naming a Holiday Time Setting

4. Click **OK**. With that Holiday Time Setting selected in the left-side menu, you can select a time range for it in the right-side time listing, as shown below. For example, for the 'Half-day Afternoon' holiday time setting, we've selected between 1:00 PM and 5:00 PM, then right-clicked to activate the 'New Event' selection.

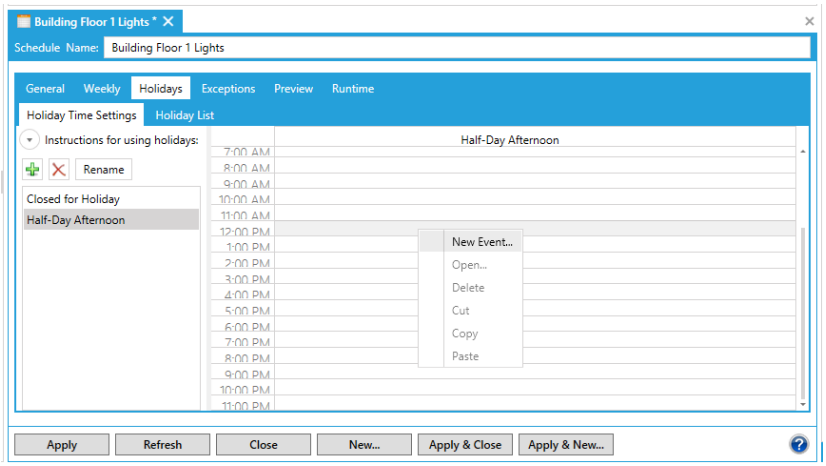


Figure 16.55 - Select Time Period and Add a New Event

5. When you click on 'New Event', the 'Event - New Event' dialog box is displayed, as shown below. Enter the subject of the event and its location. You can set a Start time and End time or make it an 'All day event'. You can also set a numerical Priority or a ValueSet Value. The text entry field allows for further personalization of the event.

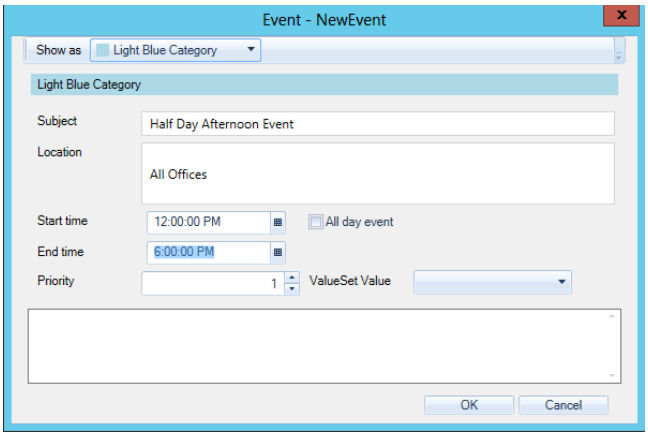


Figure 16.56 - Event Properties with Start/End time

6. Click the **OK** button to return to the Holiday Settings tab. The new holiday time setting's 'event' now appears in the right-side time listing.

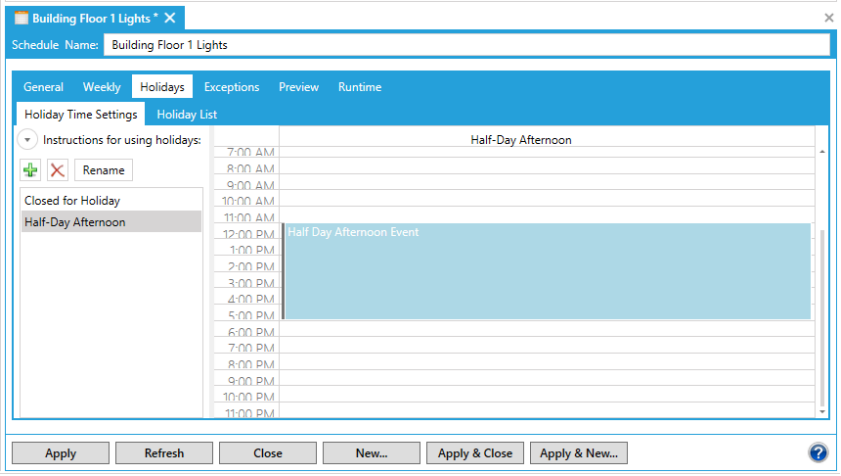


Figure 16.57 - Holiday Time Setting after Event Properties are Set

At this point, you now have a Holiday Time Setting that you can apply to selected holidays within the Holiday List tab.

16.8. ValueSets, Tags, and Resources

16.8.1. ValueSets, Tags, and Resources in ScheduleWorX64

ValueSets are collections of values or states that can be assigned to a resource by a tag. The utility of ValueSets is that they let you create a group of values that you can apply to multiple data tags. You can think of ValueSets as the group of values that populate a list box. When you create a schedule in the ScheduleWorX64 Configurator provider in Workbench for Desktop, you are defining the schedule for a resource that is controlled by a device. The tag that actually controls the device is essentially a device driver. Tags abstract the details of controlling the device from the settings or states that you assign to that device.

You could have a ValueSet with values that are:

- Static values
- Expression
- Alias
- Another OPC tag

When you create a schedule, you associate a ValueSet with that schedule on the General tab as shown in the figure below.

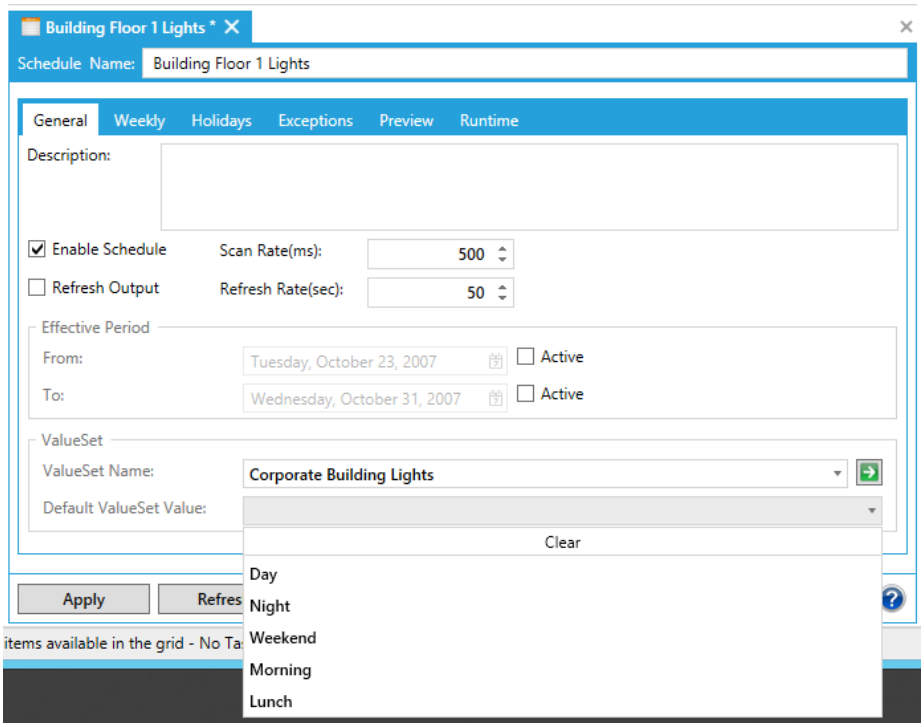


Figure 16.58 - Assigned ValueSet within the General Tab of a ScheduleWorX64 Schedule

When you assign a ValueSet Name to a schedule, you are indicating that this ValueSet is the range of values or conditions that can be set by the tag at the device that controls the resource. You can use a ValueSet in any number of schedules, but you can only assign one ValueSet to a schedule.

Also notice that you can specify a Default ValueSet Value. If there are no events occurring that would alter the value passed by the schedule to the tag, then the default value is the value given to the tag and set on the device that controls the resource.

In order to be assigned to a schedule, ValueSets must be defined in the ValueSet branch of the ScheduleWorX64 Configurator provider in the Workbench.

16.8.2. Creating a ValueSet in ScheduleWorX64

You can create a new ValueSet and assign values to it using the ValueSet branch in the ScheduleWorX64 Configurator provider in the Workbench. When you create a new ValueSet, several tabs appear that let you set the properties of the ValueSet.

To Create a Folder to Contain your Value Sets

1. Right-click on the ValueSets branch (or an existing folder) of the ScheduleWorX64 Configurator provider and click on **Add Folder**.

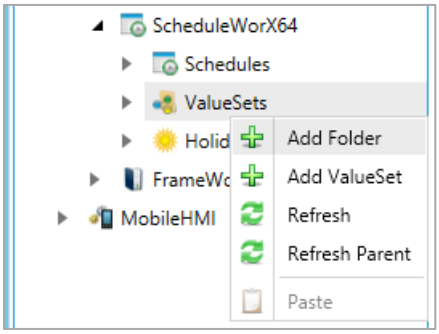


Figure 16.59 - New Folder Selected for Value Sets

-OR-

Select the ValueSet branch (or an existing folder) of the ScheduleWorX64 Configurator and click on the **Add Folder** button, shown below, in the Edit section of the Home ribbon in the Workbench.

Figure 16.60 - Add Folder Button

2. The ScheduleWorX64 ValueSet Folder appears. Enter a name for the folder into the **Folder Name** text entry field.

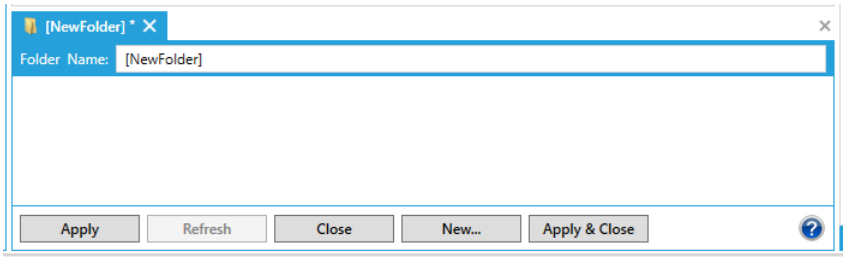


Figure 16.61 - New ScheduleWorX64 ValueSet Folder Name

3. Click **Apply** to save your changes. Your new ValueSet Folder will now appear under the ValueSet branch of the ScheduleWorX64 Configurator project explorer.

To Create a New Value Set:

1. Right-click on the ValueSet branch (or an existing folder) of the ScheduleWorX64 Configurator provider tree explorer and click on **Add ValueSet**.

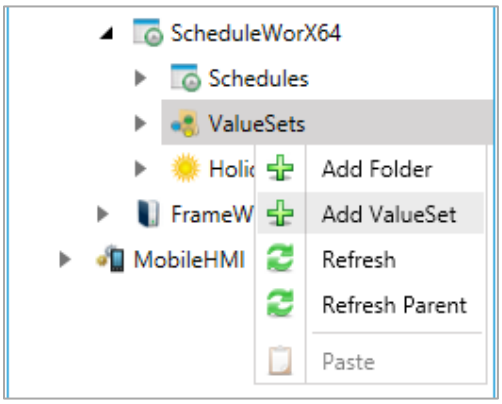


Figure 16.62 - Creating a New Value Set within the ScheduleWorX64 Configurator Tree Explorer

-OR-

Select the ValueSet branch (or an existing folder) of the ScheduleWorX64 Configurator provider and click on the **Add ValueSet** button, shown below, in Edit section of the Home ribbon in the Workbench.

Figure 16.63 - Add ValueSet Button

The ScheduleWorX64 ValueSet properties panel appears in the right-pane window.

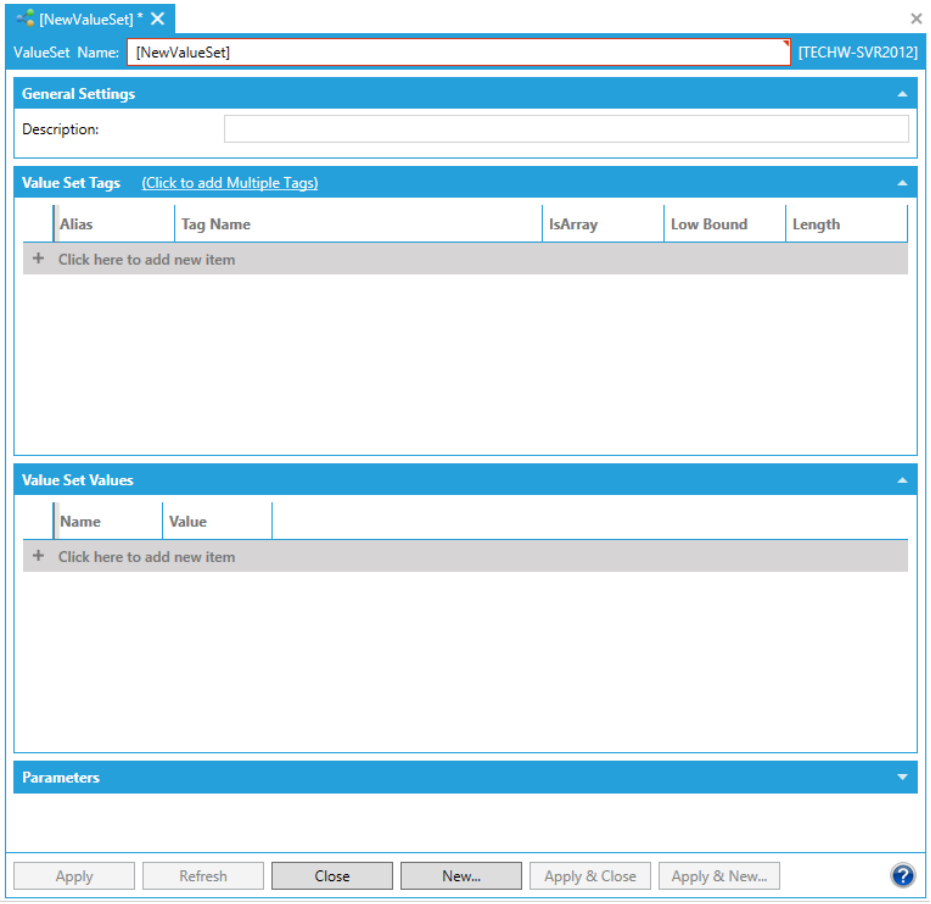


Figure 16.64 - ScheduleWorX64 ValueSet Properties

The ScheduleWorX64 ValueSet Properties are split between three tabs. See the individual tab topic for additional information.

16.8.3. Value Set Properties

The Value Set Properties of the ScheduleWorX64 ValueSet settings allows you to name the ValueSet, provide a description of its properties and set multiple parameters.

Value Set Tags

The Value Set Tags section of the ScheduleWorX64 ValueSet lets you specify the software that will send your event values to the device you are interested in controlling. Different tags can talk to different OPC devices.

1. You can edit a single tag listing by clicking on the plus sign [+] or you can click on the "(Click to add Multiple Tags)" link, which opens the Data Browser.
2. In the Data Browser, you can select the tag(s) to which your valueset values will be applied.
3. After you select your tags, click **OK**. The tags you selected appear in the **Tag Name** column of the ValueSet Tags tab. If you want to create a descriptive alias name, as shown in the following figure, click the corresponding field in the **Alias** column.

Value Set Values

The ValueSet Values section allows you to provide value types for the Tags set in the ValueSet Tags section.

1. If you have entered tags (such as from the Data Browser) into the Value Set Tags section, you will see the tag names used as column headers in the Value Set Values section. If you entered Alias names, those alias names will appear instead.
2. Enter Value Set values and names. When you get to the column containing a Value Set Tag Name or Alias Name (from the Value Set Tab tab), right-click on that field to bring up the following choices: Set as Undefined, Set as Value, and Set as Tag.
3. Enter values beneath the Value Set Tag columns.
4. Click **Apply** to save your settings on this tab.

Parameters

The Parameters section allows you to enter a Name, Type, Padding Alignment, Padding Character and Number of Places. Parameters are a way of using the same Value Set in multiple schedules. You pass the parameter to each schedule using the parameter name as part of the Value Set reference. Parameters let you set many instances of the same Value Set. For example, use parameters to assign a Value Set to the lighting on many floors, with one parameter assigned to each floor.

16.9. Calendars

16.9.1. Calendars in ScheduleWorX64

In ScheduleWorX64, calendars represent the view of a schedule over time. You can alter the schedule you see on your screen by moving between the different tabs of the ScheduleWorX64 Schedule properties (See the “Creating a Schedule” topic).

- The **Weekly tab** shows weekly recurring events. You can set the number of days in the work week, and which days are part of the work week (or schedule). Only weekly recurring events show on this tab. You can print this calendar showing your regularly scheduled events.

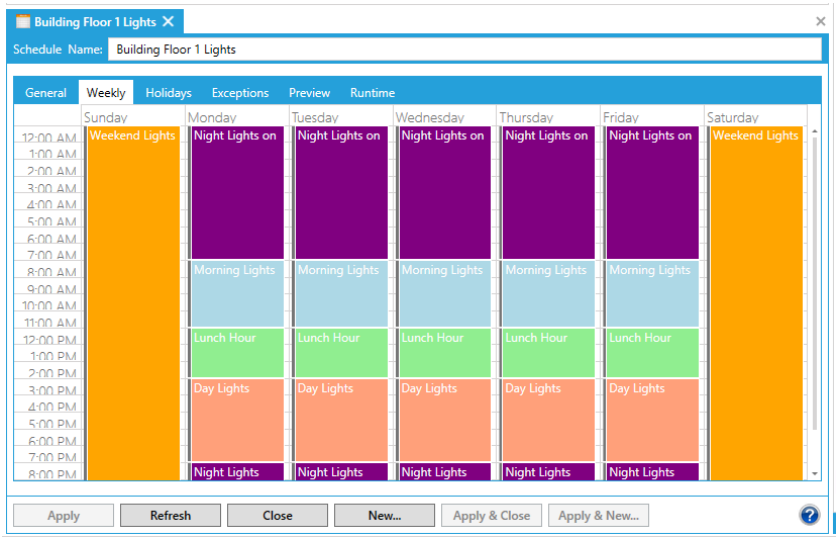


Figure 16.65 - Weekly Calendar in ScheduleWorX64 Schedule Properties

The **Holidays** tab does not display a calendar.

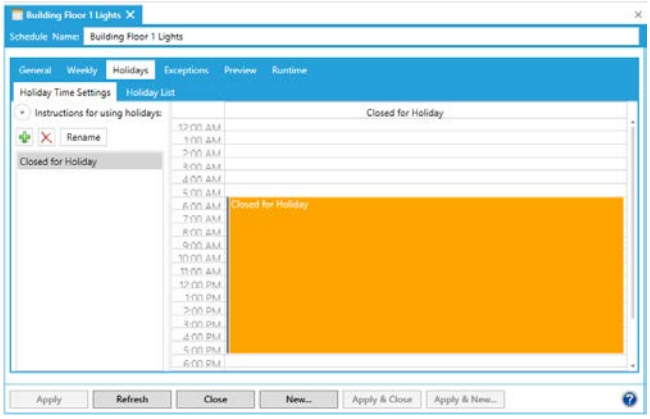


Figure 16.66 - Holidays Tab in ScheduleWorX64 Schedule Properties

- The **Exceptions tab** shows a calendar with only exception events displayed. You can set the number of days displayed in the calendar to Day View, Week View, Month View, or Timeline View. In the Week View, you can display or hide weekends. The Timeline view lets you select any seven day period. You can print the Exception calendar and it will print your current calendar view but will only display exception events in the printout.

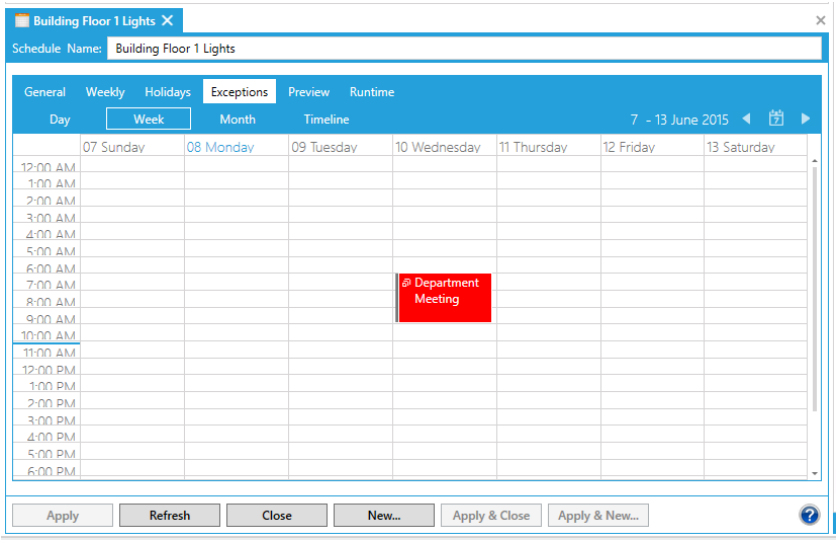


Figure 16.67 - Exceptions Tab in ScheduleWorX64 Schedule Properties

- The **Preview tab** shows all defined events in a read-only display. You cannot create events on this tab, but you have the greatest flexibility in how you set up the calendar on this view. You can set daily, weekly, monthly, or timeline views on this tab. You can also print this calendar to see all defined events of all types.

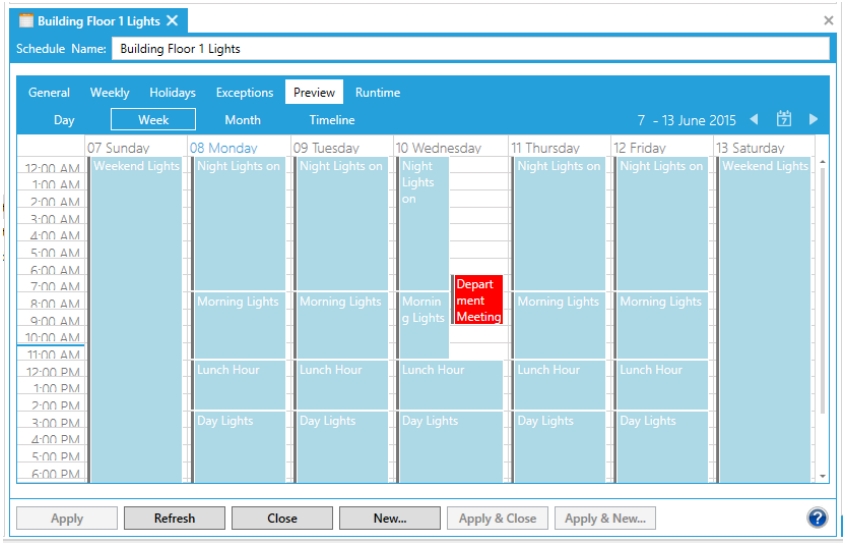


Figure 16.68 - Preview Tab in ScheduleWorX64 Schedule Properties

16.9.2. Events in ScheduleWorX64

In ScheduleWorX64, events are time periods, with associated actions, that are applied to resources. You can create a single event (or a recurring event) and assign an action, controlled by a ValueSet, to the resource for that time period.

For example, you can have the lights turn on and then off every Monday at 8 AM and 5 PM (respectively) on your main floor for restocking. Each event is given a name and the action is triggered by the start and stop time assigned to the event. When the resource is, for instance, "lights," the action can be the values contained in the OPC tag that controls the lights setting.

In order to create an event, you need to assign a ValueSet to that event. You can view a ValueSet by simply double-clicking on its name in the ScheduleWorX64 Configurator provider tree explorer.

Events can be created and assigned on the Weekly, Holiday, or Exceptions tab. When there is no event active, the default condition on the General tab applies. The Preview tab lets you view all of your events, but doesn't let you create events.

An event that you create on the Weekly tab is a recurring event with a period of one week.

To Create a Weekly Event:

1. Double-click on an existing schedule (or create a new schedule; see the “Creating a Schedule” topic) in the ScheduleWorX64 Configurator provider tree explorer. Click on the Week View tab in the ScheduleWorX64 Schedule properties.

2. Double click on a **time period**, or click and drag the time period desired. The Event Properties dialog box appears.

3. Enter information in the following fields:

   - **Subject** - Enter the name of the event. Unlike schedule names, event names do not need to be unique.
   - **Location -** Optional setting.
   - **All day event** - If you select this checkbox, the **Start Time** and **End Time** fields become inactive.
   - **Start time** and **End time** - Enter the start time and end time into those two boxes, respectively.

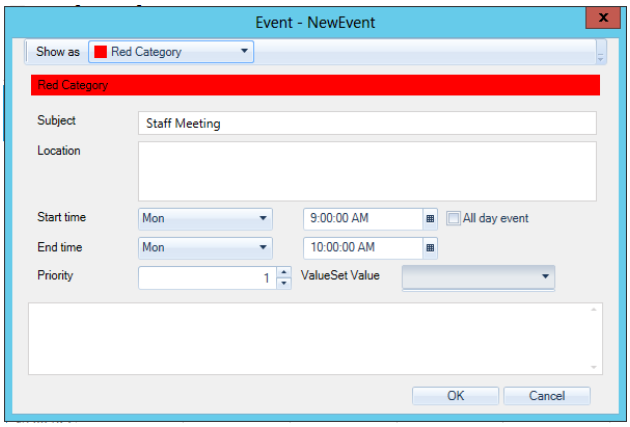


Figure 16.69 - Edit Event Dialog Box

Info

NOTE: Weekly events, by definition, reoccur on a weekly basis. This dialog box does not contain the recurrence button that is visible for Exception events.

4. A schedule has a ValueSet assigned to it, but you need to select the **ValueSet Value** from that field. That value is the value given to the OPC tag when this event begins, and represents the action to be taken by the resource. In the preceding screen shot, the 'Weekend Lights' is set to the ValueSet Value of 'Weekend'.

5. Set the **Priority** by either entering a value from 1 to 1000 or by selecting the value from the pulldown menu. Event priorities are used to determine which event takes priority and runs when there is a conflict. The Event priority is used to compare two weekly events.

In order of priority, the following is the order of precedence in ScheduleWorX64:

- General Events
- Weekly events take precedence over general events
- Holidays take precedence over weekly events and general events
- Exceptions take precedence over holidays, weekly, and general events
- Overrides take precedence over all other events. You apply an override directly to your schedule during runtime, specifying the override condition as well as its starting and stopping time. An override is similar to taking control over your schedule manually and takes precedence over all other events.

For a fuller discussion of priorities and precedence, refer to the "Priorities" topic.

6. Click the **OK** button to return to the Weekly tab.

7. Click the **Apply** to save the event.

Weekly events are central in ScheduleWorX64, as this tool is meant to provide an organizational structure for the work week. The Exceptions tab creates one-time or recurring events that take precedence over weekly events. In the ScheduleWorX64 Configurator provider in Workbench, if you want to create an event that is neither a weekly recurring event nor a holiday, then you should define that event as an Exception event.

Holidays are defined as a day of the year. Some holidays always fall on the same day or date of the year. For example, Christmas is always December 25th. When December 25th falls on a weekend, you can schedule the following or preceding day as the holiday for employees. Other holidays can fall on a certain day within a certain week of a month, such as the Thanksgiving holiday in the United States, which falls on the fourth Thursday of November. Thanksgiving always impact a work week. ScheduleWorX64 gives you the flexibility to define both types of holidays.

You can define a collection of holidays as a holiday list, define holidays as days that contain events, and assign an event or collection of events on that holiday's date. If you do not create a time for a holiday, then the holiday is defined by default as an All Day event in the Edit Event dialog box.

16.9.3. Priorities in ScheduleWorX64

ScheduleWorX64 prioritizes scheduled events using a strict order of precedence that first evaluates the event type and then evaluates the event priority. This order determines which event runs whenever there is a conflict between two or more events.

The order in which event types are processed is as follows:

If there are no scheduled events, ScheduleWorX64 runs the default value defined on the General tab shown below.

A default value runs whenever no other event is scheduled.

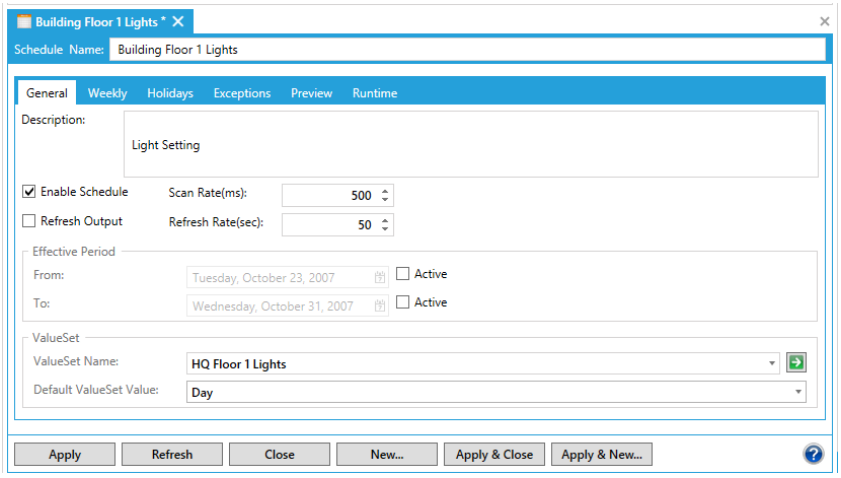


Figure 16.70 - A Default Value Runs Whenever No Other Event is Scheduled

- A weekly event run takes precedence over a regularly scheduled event. A weekly scheduled event with a higher priority will run before a weekly scheduled event with a lower priority. Priorities can be set at a number from 1 (lowest) to 1000 (highest).

Info

Note: One way to remember the order of precedence for event types is to think of the Event Properties multi-tabbed pane as a wizard. The lowest precedent event you define is on the General tab. Moving from left to right, weekly is higher, and holidays higher still and exceptions run in precedence to all of these other events. You can also set an override to any event and that override value will be applied for the duration of the override you specified.

- A holiday event takes precedence over either a weekly or regularly scheduled event. A holiday event with a higher priority will run before a holiday event with a lower priority. Priorities can be set at a number from 1 (lowest) to 1000 (highest).
- An exception event takes precedence over a holiday, weekly, or regularly scheduled event. An exception event with a higher priority will run before an exception event with a lower priority. Priorities can be set at a number from 1 (lowest) to 1000 (highest).
- Overrides take precedence over all other events. You apply an override directly to your schedule during runtime, specifying the override condition as well as its starting and stopping time. An override is similar to taking control over your schedule manually and takes precedence over all other events.

NOTE: If you want to schedule an event with only one instance, then you need to schedule it as an exception event.

16.9.4. Recurring Events

Recurring Events in ScheduleWorX64

Recurring events in ScheduleWorX64 are events that take place over a specified period of time. You can have recurring events that occur:

- Daily at the same time of day.
- Weekly at the same time and day of the week
- Monthly on the same date of the same week of the month (e.g., the first day of each month).
- Monthly at the same time and day of the month (e.g., the third Thursday of a month).
- Yearly on the same date of the year (e.g., December 25th - Christmas).

There are two places where you can set recurring events:

- **Weekly tab -** All events that you create on the Weekly tab are weekly recurring events.
- **Exception tab -** All recurring events that you create on the Exception tab are exception events and take precedence over the weekly recurring events defined on the Weekly tab, or any other regularly scheduled event.

16.9.5. Regularly Scheduled Weekly Events in ScheduleWorX64

In ScheduleWorX64, to create a weekly event at the same day and time of the week:

1. Click on the Weekly tab in the ScheduleWorX64 Schedule properties.

2. Double-click on the time period you want to define as a weekly event. Or, if the event will span more than one time period, click and drag to create an event for that length of time then right-click on the event and select the New Event command from the context menu. In either case, the Event - NewEvent dialog box appears.

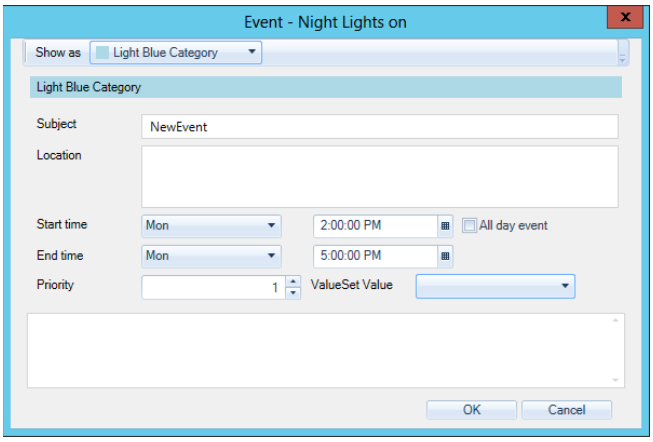


Figure 16.71 - Event - NewEvent Dialog Box in the Weekly View Defines Weekly Recurring Events

3. Select a label (color) from the "**Show As**" drop down list, if desired (optional).

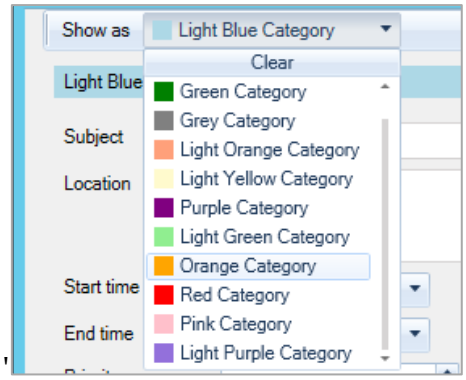


Figure 16.72 – "Show As" Menu in the Event - NewEvent Window

4. Enter the **Subject**, which is the name of the event that appears in the calendar. Subject names must be unique.

5. Enter the location into the **Location** text box (optional).

6. Enter the **Start time** and **End time** directly into the text entry fields, or select the dates from the Date Picker by clicking the down arrow.

7. If the event is to be an all day event, then enable the **All day event** check box.

8. Should a conflict including regularly scheduled events occur, the event with the higher priority controls the resource. Set the Event Priority in the pulldown menu to any value between 1 (the lowest) and 1000 (the highest).

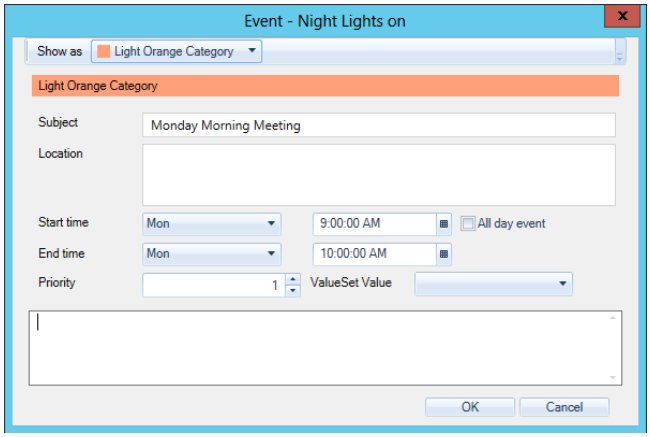


Figure 16.73 - Edit Event Dialog Box

9. Click the **OK** button to create the regularly scheduled weekly event with the properties you entered.

16.9.6. Recurring Exceptions in ScheduleWorX64

In ScheduleWorX64, all recurring events, other than the regularly scheduled weekly events described in the previous section, are scheduled as recurring exceptions. These recurring events take precedence over any regular event that occurs at the same time.

To Set a Recurring Event as an Exception Event

1. Click on the **Exceptions** tab of the ScheduleWorX64 Schedule properties.

2. Double-click on the **time period** you want to define as a recurring event. Or, if the event will span more than one time period, **click and drag to create an event** for that length of time, then right-click on the event and select the **Create New Event** command from the context menu.

   In either case the Exception Event Properties dialog box appears.

   Info

   NOTE: The Exception Event Properties dialog box is identical to the Event Properties dialog box for the Weekly event, but now has an extra button for Recurrence.

3. Fill out the remaining properties, as desired, then click the **Edit Recurrence** button in the top left of the Edit Event Dialog Box.

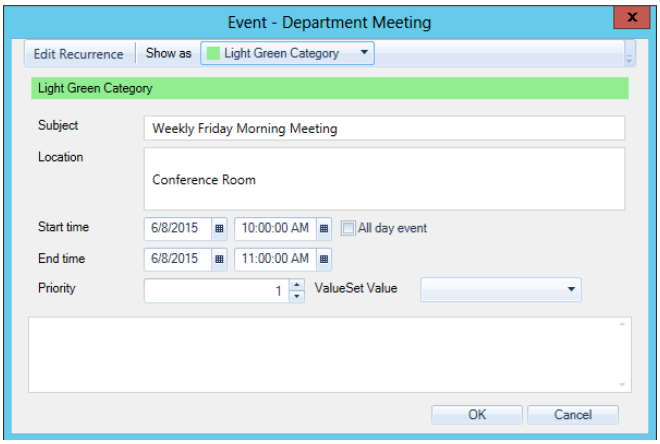


Figure 16.74 - Edit Event Dialog Box

4. The Appointment (Edit Recurrence) dialog box opens. Complete the recurrence fields. In the example shown, a recurrence pattern is set for a weekly Friday morning with no end dates.

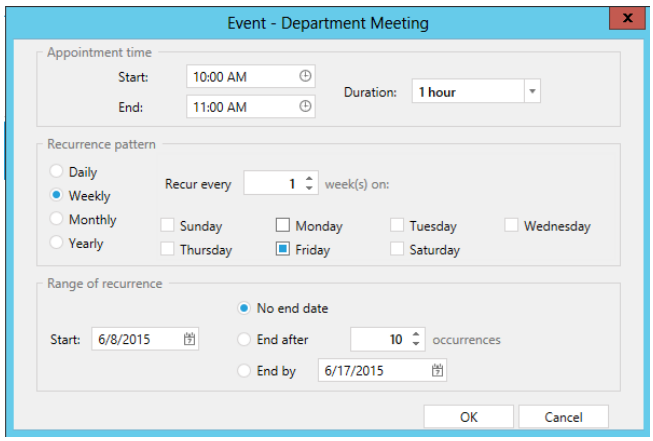


Figure 16.75 - Appointment (Edit Recurrence) Dialog Box

5. Click **OK** on the Edit Recurrence dialog box to save this recurrence.

6. Click **OK** on the Edit Event dialog box to save this event.

16.9.7. Editing a Recurring Event in ScheduleWorX64

If you want to edit an event that is recurring in ScheduleWorX64, you can do so by working with an instance of the recurrence or with all instances.

To Edit Recurring Events

1. On the **Exceptions** tab, view the recurring event in either the Day, Week, or Month view.

2. Select the event, right-click and select **Open...** .

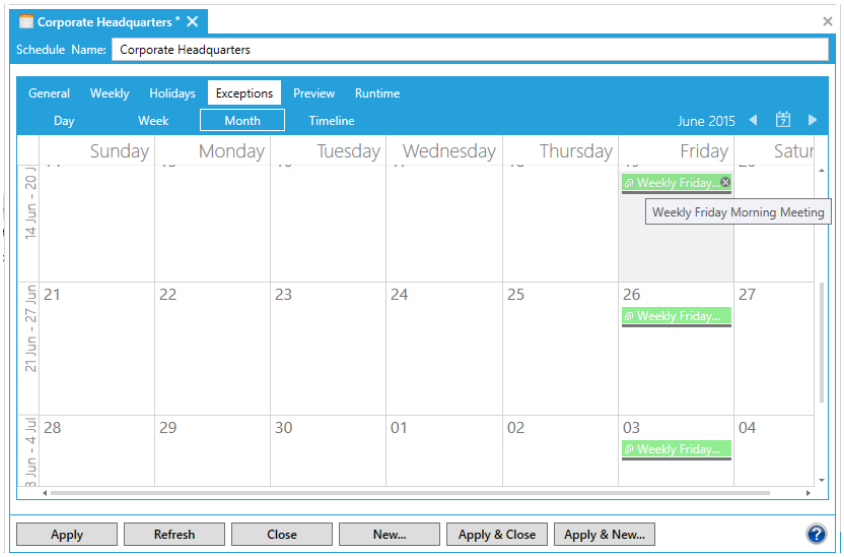


Figure 16.76 - Exception Event Shown in Monthly View

The Appointment Recurrence Window will open, providing you with the option either open an individual occurrence or the entire series.

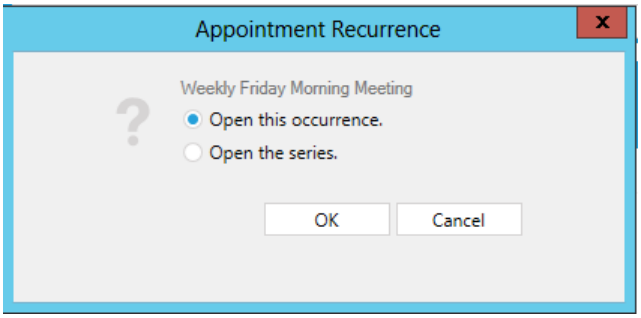


Figure 16.77 - Appointment Recurrence Window

3. Either selection will display the Edit Event Dialog Box.

- If you select **Open this Occurrence**, the Start Time and End Time are available to you to edit. You also have the option to click on the 'Edit Parent Event' button to edit the series instead.

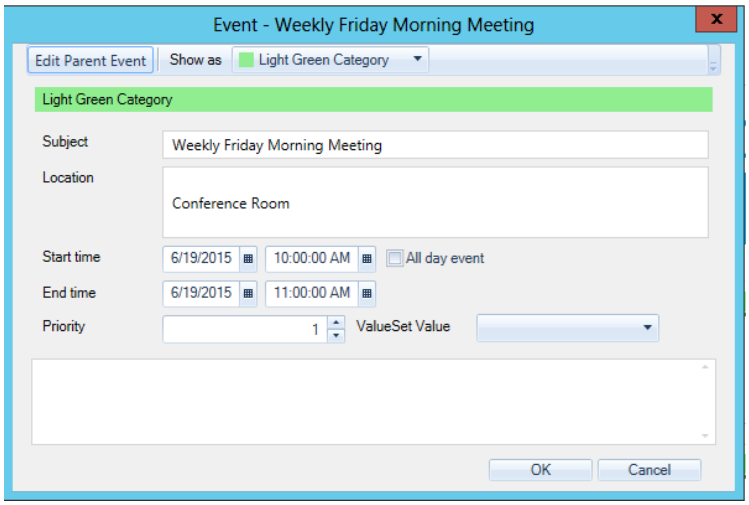


Figure 16.78 - Edit Event Dialog Box with One Time Event

If you select **Open this Series**, the Start Time and End Time are disabled, but the **Edit Recurrence** button is available to you. To edit the properties of the recurring period, click the **Edit Recurrence** button and make any changes desired in the Appointment (Event Recurrence) dialog box, described in the "Recurring Exceptions" topic.

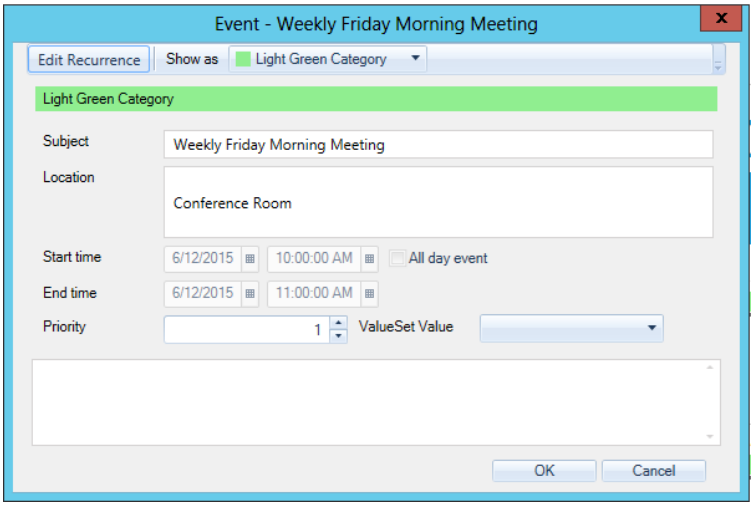


Figure 16.79 - Edit Event Dialog Box with Recurring Event

16.9.8. Deleting a Recurring Event in ScheduleWorX64

This topic describes how to delete a recurring event in ScheduleWorX64.

To Delete a Recurring Exception Event

1. Select the **Exceptions** tab and display the event on your Day, Week, or Month calendar.

2. Select the event and right-click.

3. Select **Delete**.

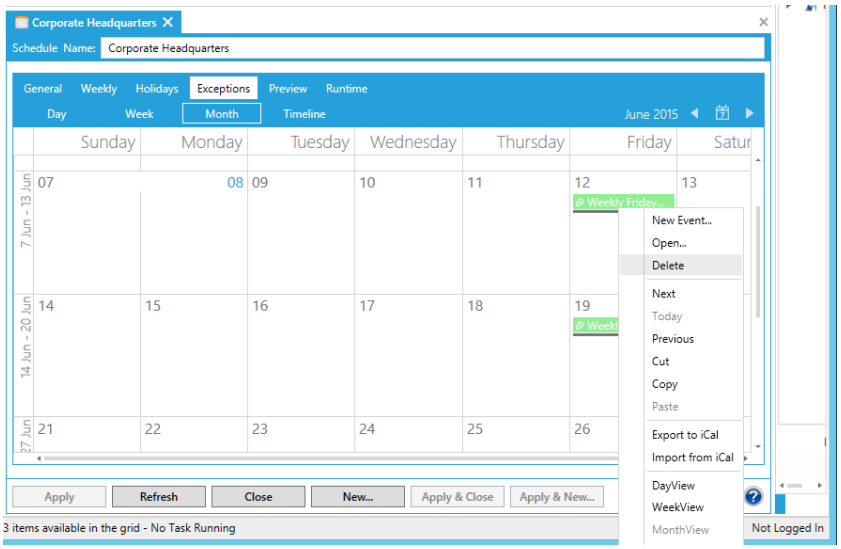


Figure 16.80 - Deleting a Recurring Event

4. The Appointment Recurrence window appears. You will have the option of deleting either the individual occurrence or the entire series.

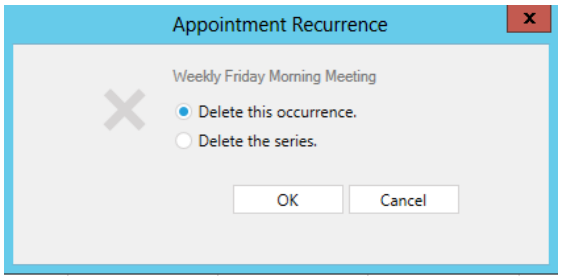


Figure 16.81 - Appointment Recurrence Window

Depending on your decision, you will see either that specific event removed from the calendar or the entire recurring series.

Lab: ScheduleWorX64 L16

16.1. Lab Overview

In this lab, we will create a schedule that can be used by an operator to manage the heating, ventilation, and air conditioning (HVAC) systems for a facility. Also, we will create a GraphWorX64 display that the operator can use to view, modify, and override the schedule.

Estimated time to complete this lab: **15 minutes**

16.1.1. Task Checklist

In this lab, you will complete the following tasks:

- Create a Valueset to define the resources to be scheduled
- Create a Holiday list that can be reused in multiple schedules
- Create a Schedule
- View the Schedule in a GraphWorX64 display
- Manually override the Schedule

16.1.2. Support Files

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

16.2. Creating a ValueSet

First, you will create a ValueSet to define the resources that will be controlled by the schedule.

1. Open **Workbench**. In the Project Explorer tree view, go to **Platform Services** → **ScheduleWorX64**.
2. Expand ScheduleWorX64 and then expand **ValueSets** in the tree.
3. Right-click on **ValueSets** and choose **Add ValueSet**.

4. In the **ValueSet Name** field, enter the name **Training Center Heating**. In the **General Settings** section **Description** field, enter **This ValueSet identifies the tags that control the heating system for the training center.**

5. Under the **Value Set Values** section, enter the following two tags in the **Tag Name** column:

   1) **Alias:** Room
      Tag Name: @sim64:Long.Static("TrainingTemperatureSetPoint").Value
   2) **Alias:** Hallway
      Tag Name: @sim64:Long.Static("CorridorTemperatureSetPoint").Value

6. Under the Value Set **Values** section, fill-in 2 entries:

Name	Value	Room	Hallway
Occupied	Occupied	72	70
Unoccupied	Unoccupied	62	60

7. Press Apply. When you are finished, the ValueSet configuration should look like this:

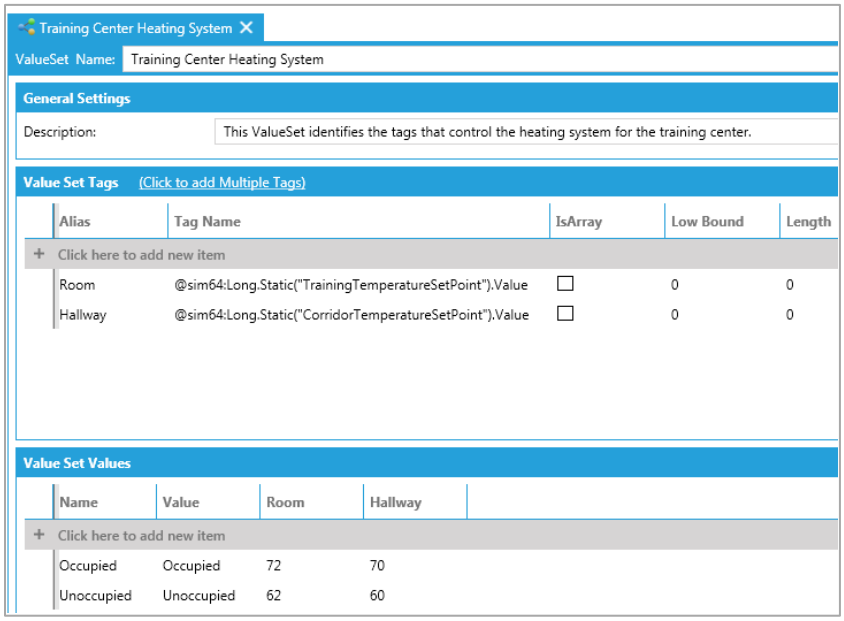


Figure L16.1 – Completed ValueSet Configuration

16.3. Creating a Holiday List

Now you will create a holiday list. The holiday list defines any number of dates that will be used later in one or more schedules to identify dates that require special behavior in the schedules. For example, if the building is closed on a holiday then the lighting and heating systems can be adjusted to save energy.

1. Open **Workbench**. In the Project Explorer tree view, go to **Actions → ScheduleWorX64**.
2. Expand ScheduleWorX64 and then expand **Holidays** in the tree.
3. Right-click on **Holidays** and choose **Add Holiday**.
4. In the **Holiday Name** field, enter **Training Center Holidays**.
5. In the **Description** field, enter "**This holiday list defines the list of holidays for the training center. It can be shared between multiple schedules.**"
6. Next, click on the **Calendar View** tab as shown here:

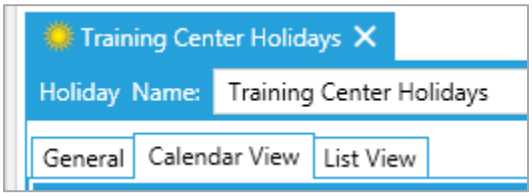


Figure L16.2 – Calendar View Tab

7. In the **Calendar View** tab, we will enter a recurring holiday for New Year's Day:
   a. Navigate to the past **January 1st** by using the date navigation tools in the upper right corner:

Figure L16.3 – Date Navigation Tools

   b. Double-click on **January 1** and you will see the new event dialog. Enter the **Subject** as **New Year's Day**. In the **Show As** field, change the **Color** to **Light Green Category**. The color is only used to give an event a different visual representation, so using it is optional.

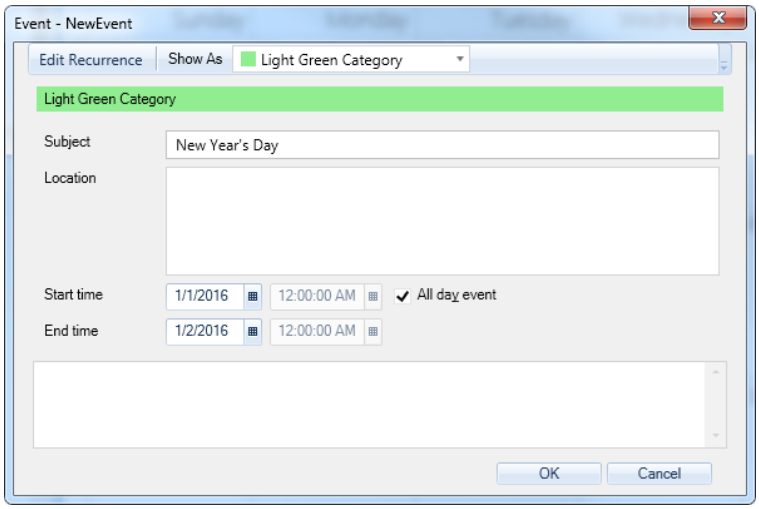


Figure L16.4 – Setting the Recurrence Color Category

c. Next, click on the **Edit Recurrence** button. Set the event to have a yearly recurrence pattern for every **January 1**, as shown here:

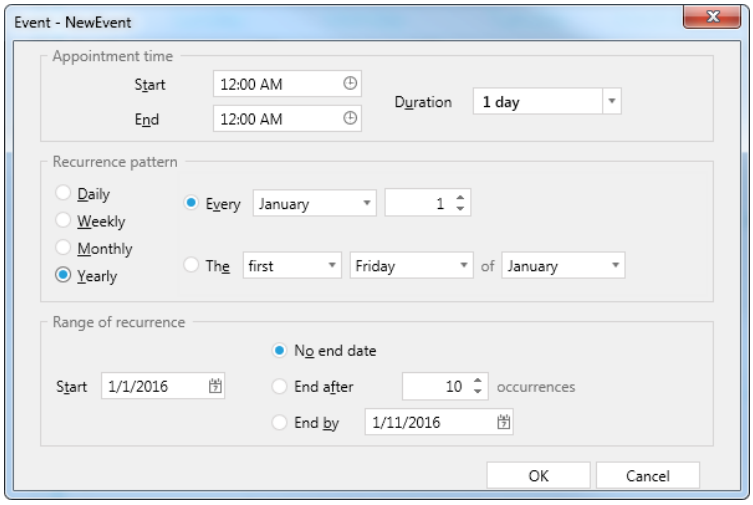


Figure L16.5 – Setting the Recurrence Pattern

d. Press **OK** to close the popup windows.

e. Press **Apply** to commit the changes to the holiday list.

16.4. Creating a Schedule

In this section you will create a schedule that controls the resources defined in the **Training Center Heating System** Value Set. This schedule will also use the **Training Center Holidays** holiday list.

1. Open **Workbench**. In the Project Explorer tree view, go to **Platform Services → ScheduleWorX64**.
2. Expand ScheduleWorX64 and then expand **Schedules** in the tree.
3. Right-click on **Schedules** and choose **Add Schedule**.
4. In the **Schedule Name** field, enter **Training Center Heating Schedule**.
5. In the **Description** field, enter "**This schedule defines the heating settings for the training center.**"
6. In the **ValueSet** section, click on the **ValueSet Name** combobox to view the configured ValueSets. Click on **Training Center Heating System** to associate the **ValueSet** with this schedule.
7. Click on the **Default ValueSet Value** combobox to view the available values. Select **Unoccupied** to set it as the default value. The **General** tab should look like this:

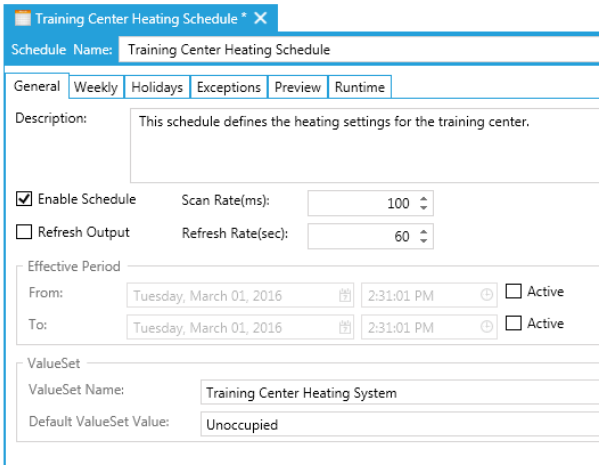


Figure L16.6 – The Schedule's General Tab

8. Next, you will configure weekly events. A weekly event takes precedence over a default ValueSet Value.
9. Click on the **Weekly** tab.

10. **Highlight** the region from **8:00 AM – 5:00 PM** on **Monday** by clicking on **8:00 AM** and holding down the mouse to drag the region to **5:00 PM**. Right-click on this region and choose **New Event…** as shown here:

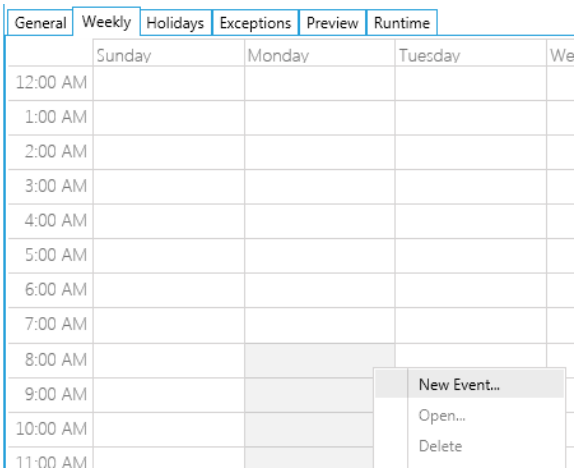


Figure L16.7 – Creating a New Event

11. Configure the new event as follows:

    a. For the **Subject** field enter **GENESIS64 Training Class**.

    b. In the **Location** field enter **Training Center**.

    c. On the ValueSet **Value** field choose **Occupied** from the list.

    NOTE: You can increase the Priority field for any weekly event if it should take precedence over other weekly events that occur at the same time and have a lower priority. This priority setting can also be used for holiday events or exception events in the following exercises.

    d. The configured event dialog should look like the following figure.

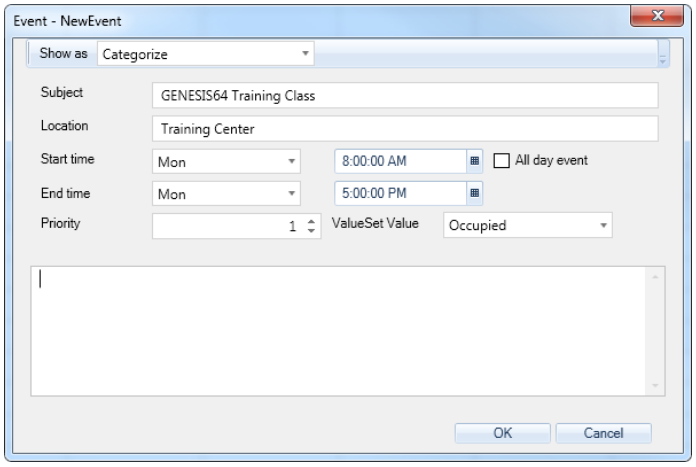


Figure L16.8 – Configured Event Dialog

12. Press **OK** to close the dialog and create the event

13. Right-click on the new event and choose **Copy**:

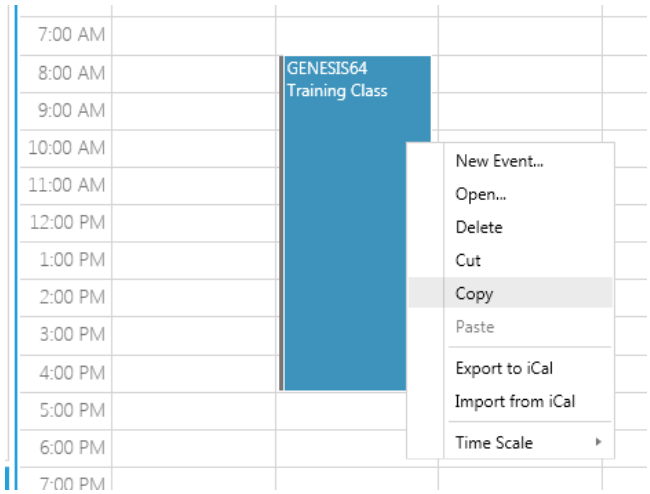


Figure L16.9 – Copying the Event

14. Right-click on **Tuesday** at **8:00 AM** and choose **Paste** to insert the copied event.

15. Repeat the Paste operation for **Wednesday**, **Thursday**, and **Friday**.

16. Press **Apply** to commit the changes. The result should look like this:

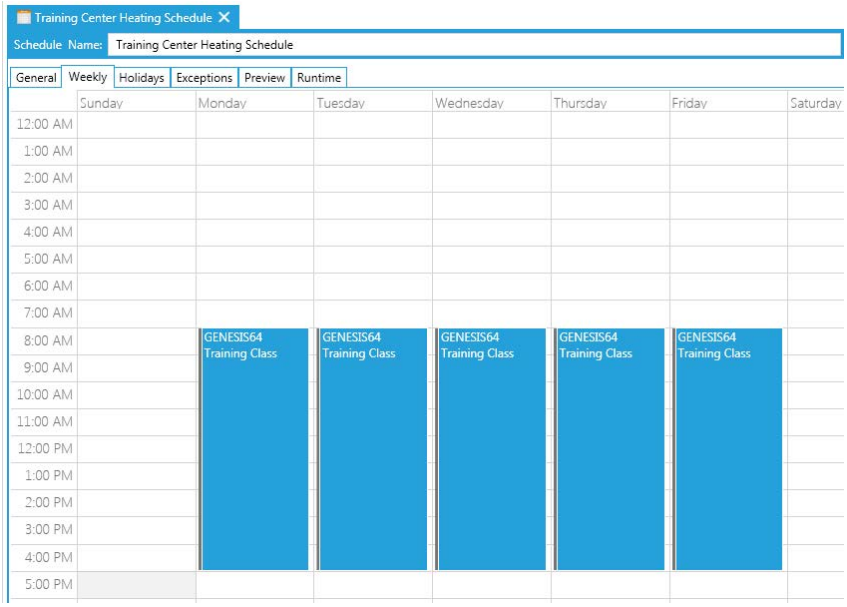


Figure L16.10 – Resulting Copied and Pasted Events

17. Click on the **Holidays** tab. You will now associate this schedule with the previously created Holiday list. A holiday event takes precedence over a weekly event or a default ValueSet Value.

18. On the **Holiday Time Settings** tab, press the '+' button to add a new Holiday Time Setting. This will be used to instruct the schedule what actions to take on a holiday date.

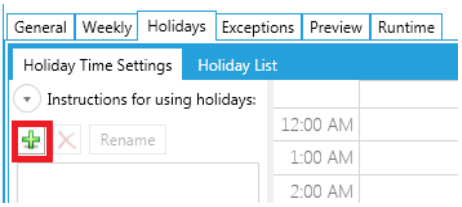


Figure L16.11 – Holidays Tab

19. For the **Time Setting Name**, enter **Office Closed**. Press **OK**.

20. Double-click on the right side of the screen (anywhere between 12:00AM and the end of the day). This will pop up a New Event dialog. Fill in this dialog as shown below. Be sure to check the **All day event** checkbox, and set the **ValueSet Value** to **Unoccupied**:

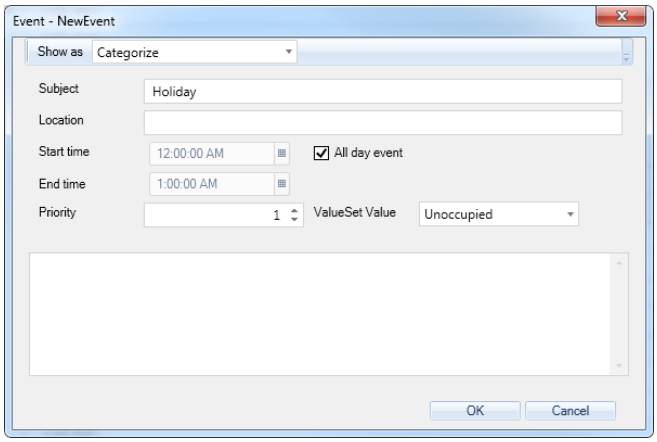


Figure L16.12 – New Event Dialog

21. Press **OK**.

22. You should see the **Holiday** event on the right side of the screen lasting all day.

23. Now, click on the **Holiday List** tab.

24. Click on the **Holiday List** combobox, scroll down and select **Training Center Holidays**.

25. You should see all of the holidays that are configured in the holiday list:

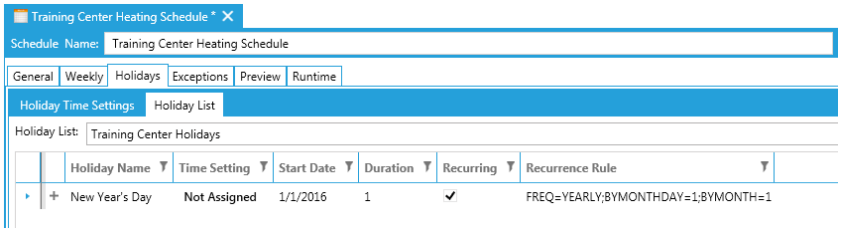


Figure L16.13 – Configured Holiday in Holiday List

26. Under the **Time Settings** columns, click on **Not Assigned** and a drop down list will appear. Click again to select **Office Closed** from the list:

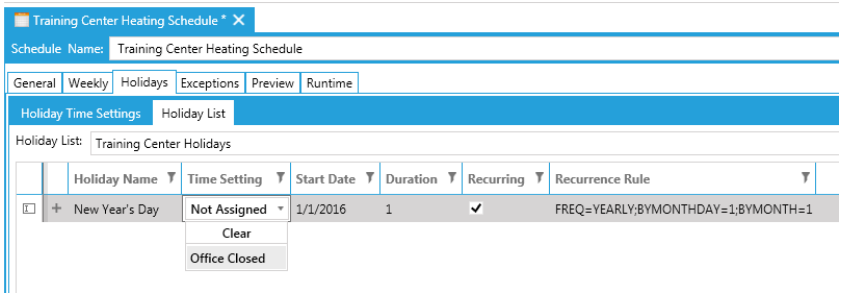


Figure L16.14 – Using the Time Settings Column Pulldown

27. The **New Year's Day** holiday is now properly configured to apply the **Unoccupied** setting for this schedule.

28. Next, click on the "+" sign in the left-most column to see a preview that shows multiple occurrences of the holiday. This feature is useful to determine when the holiday will occur in the future.

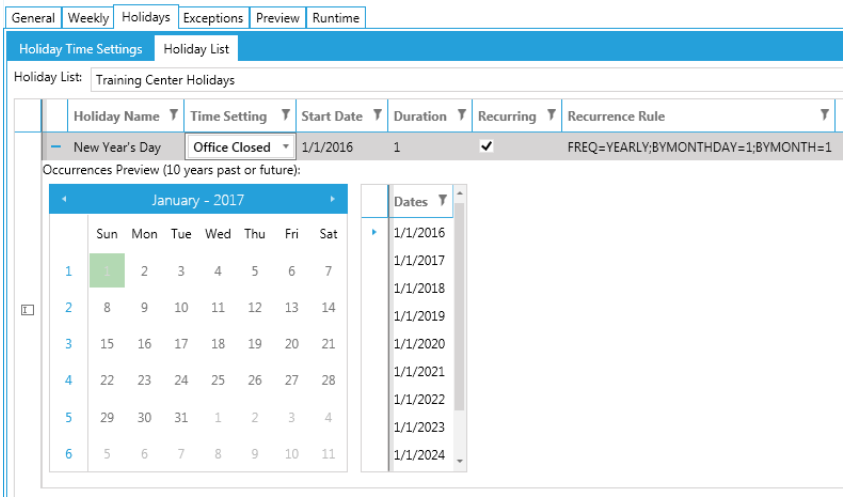


Figure L16.15 – Using the Preview for Holiday Occurences

29. Press **Apply** to commit the changes.

30. Next, you will configure an Exception event. An exception event takes precedence over a holiday event, weekly event, or a default ValueSet Value.

31. Click on the **Exceptions** tab to configure an exception. An Exception represents an event that may not be a regular Weekly event, nor a Holiday that would apply to multiple schedules. An Exception can be configured to occur once, or on a recurring interval.

32. Click on **Month** to view a month layout of the calendar.

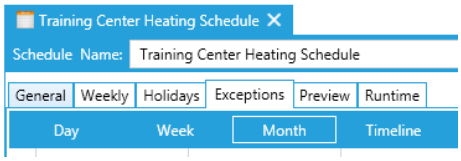


Figure L16.16 – Selecting Month Layout

33. Double-click on the **first Saturday** of the current month, and configure an event as shown below:

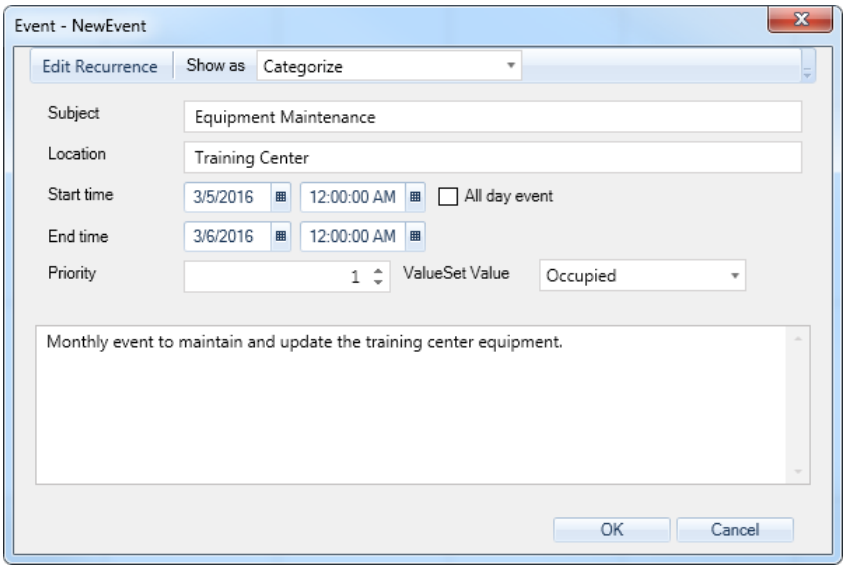


Figure L16.17 – Event Configuration

34. Now, click on the **Edit Recurrence** button.

35. Set up the event to occur on the first Saturday of every month as shown here:

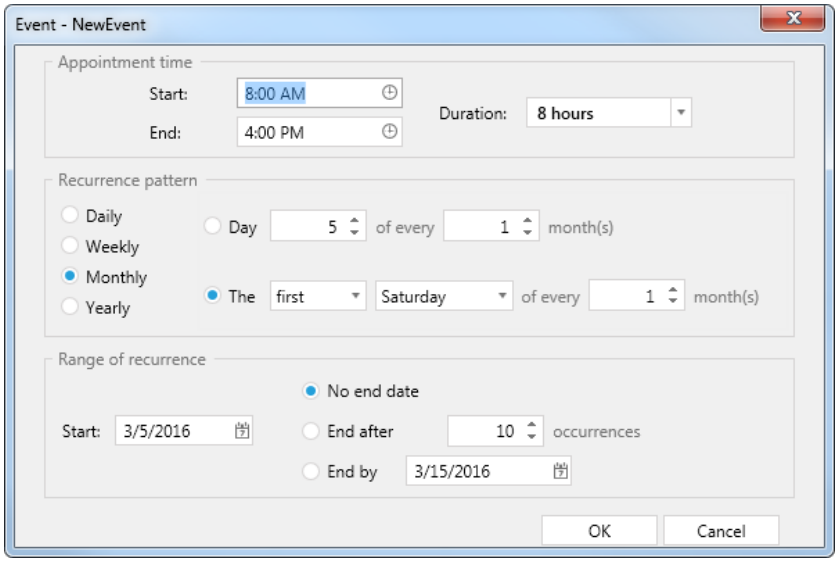


Figure L16.18 – Configuring a Recurring Event

36. Press **OK** on both of the open dialogs. You should then see the monthly recurring exception event:

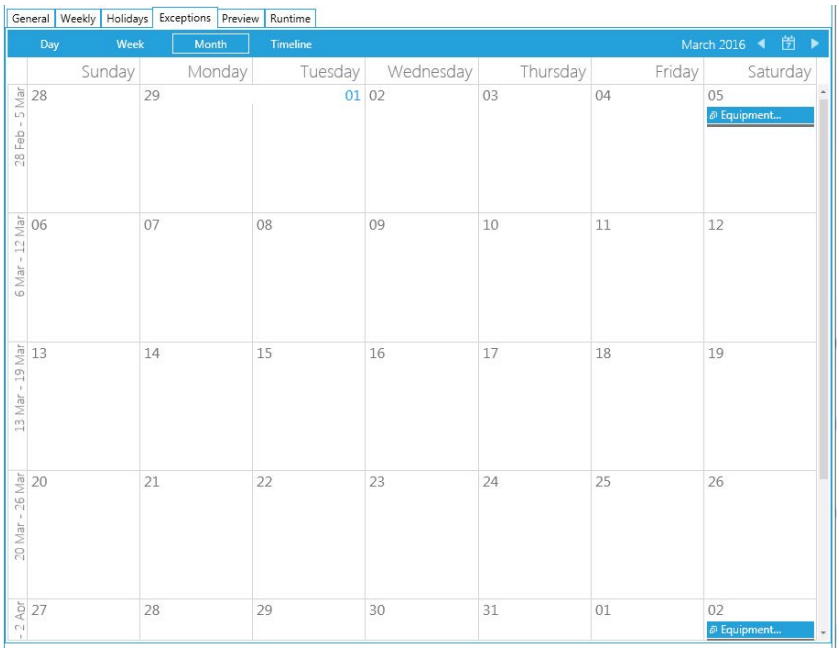


Figure L16.19 – Monthly Recurring Exception Event

37. Press **Apply**.

38. Click on the **Preview** tab to see a preview of all of the configured events.

39. If you click on the **Month** view, and then Navigate to the following January 1st, then you will see the **New Year's Day** holiday event, the weekly events, and the monthly **Equipment Maintenance** events:

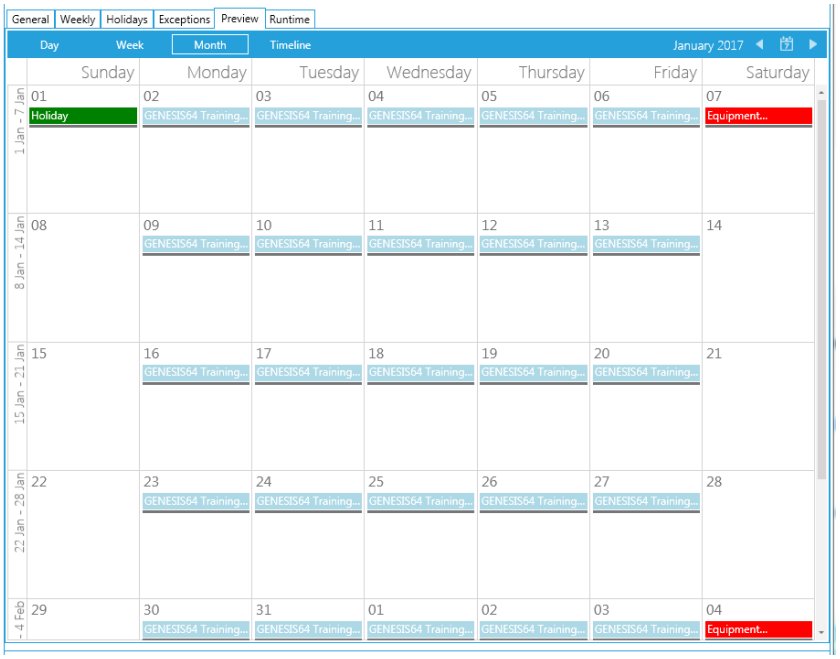


Figure L16.20 – Review of Configured Events

40. Click on the **Runtime** tab.

41. Press the **Refresh** button to force the dialog to load the Runtime values.

42. The **Runtime** tab provides information about the status of the running schedule. If the ScheduleWorX64 service is not running, then you will just see "*****" for all of the values on this tab. To start the ScheduleWorX64 service, go to the Home tab at the top of the Workbench. If the traffic light is red then the ScheduleWorX64 service is not running. In this case, click on the traffic light to start the service.

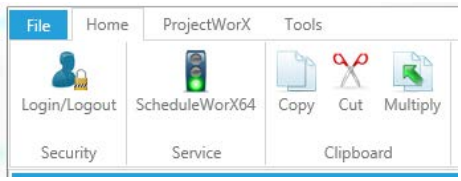


Figure L16.21 – Starting the ScheduleWorX64 Service in the Workbench Ribbon

43. Now you should see the Runtime values of the schedule. Notice that this tab provides information about the current event as well as the next event that is scheduled to occur:

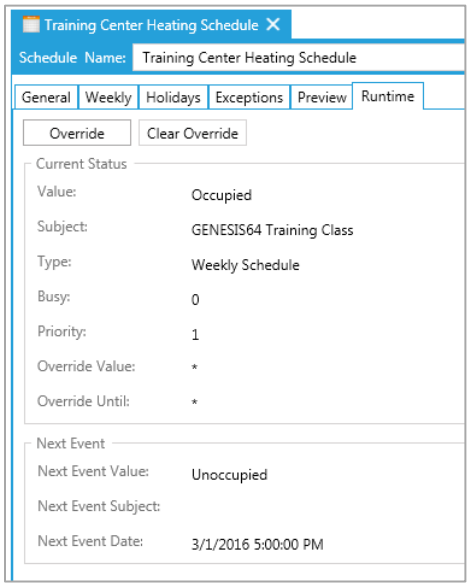


Figure L16.22 – Runtime Values of the Schedule

16.5. Viewing the Schedule in GraphWorX64

In this exercise, you will create a GraphWorX64 display that allows operators to view and modify the **Training Center Schedule**.

1. Open **GraphWorX64**.
2. Click on the **Controls** tab, and click on the **Schedule Control**.
3. Left-click and drag in the **GraphWorX64** canvas to insert the control and customize the size of the control. Or, simply click in the **GraphWorX64** canvas.
4. Once inserted, select the control and the **Schedule Control Configuration** section will appear in the GraphWorX64 ribbon. Click on the **Edit** button to configure the connection to the schedule.

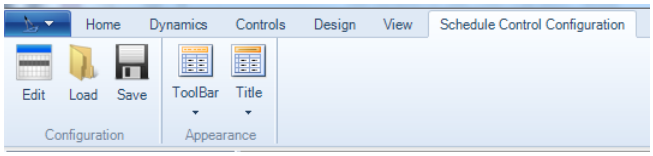


Figure L16.23 – Configuring a Schedule Control in GraphWorX64

5. You will see the **Configure your Control** dialog. Click on the **ScheduleWorX64 Viewer** in the tree.

6. On the **Source** tab, fill in the **Data Source** by clicking on the button on the right side:

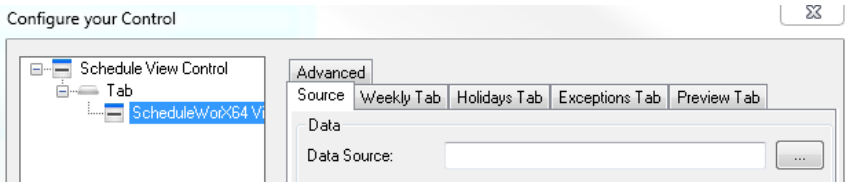


Figure L16.24 – Schedule View Control Source Tab

7. The **Data Browser** dialog will open.

8. Expand **My Computer → Data Sources → OPC Data Access → ScheduleWorX64 Schedules**.

9. Select **Training Center Heating Schedule** and Press **OK**:

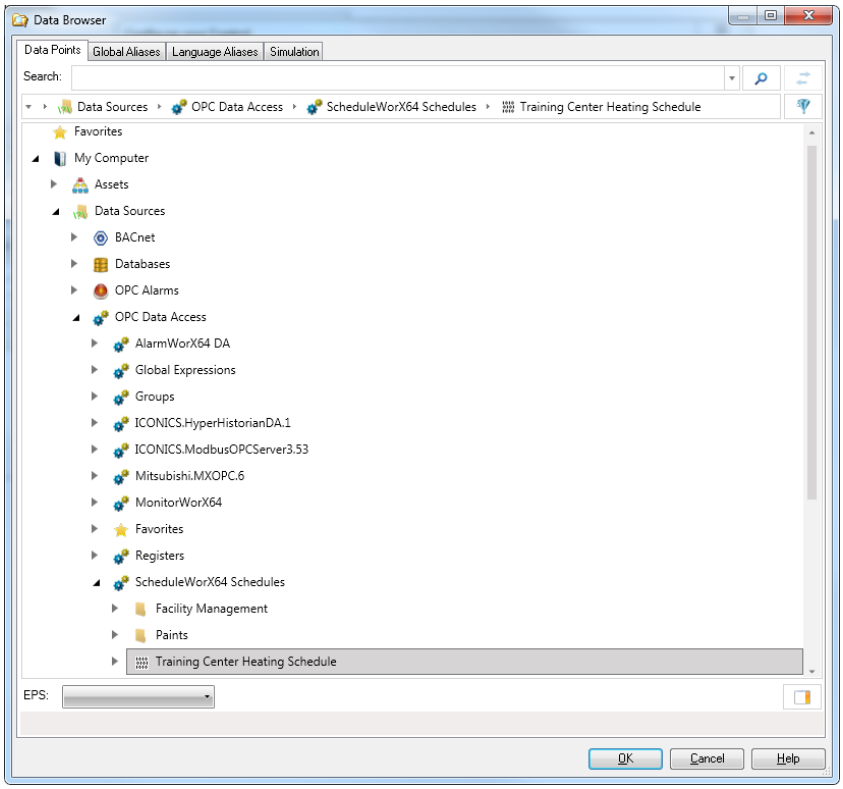


Figure L16.25 – Selecting a Schedule Via the Data Browser

10. The **Data Source** field should now be filled in:

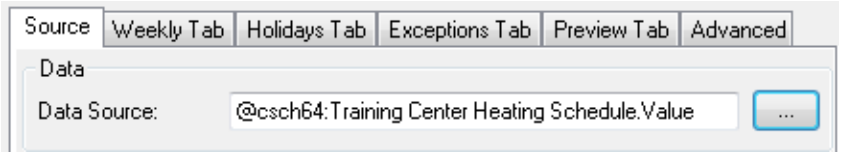


Figure L16.26 – Selected Data Source

11. Press the **Close** button.

12. In GraphWorX64, press the **Save** button in the top left and save this display as **TrainingScheduleView.gdfx**.

13. In GraphWorX64, press the **Runtime** button to go to Runtime mode. The schedule control will now connect to the schedule to display to the operator:

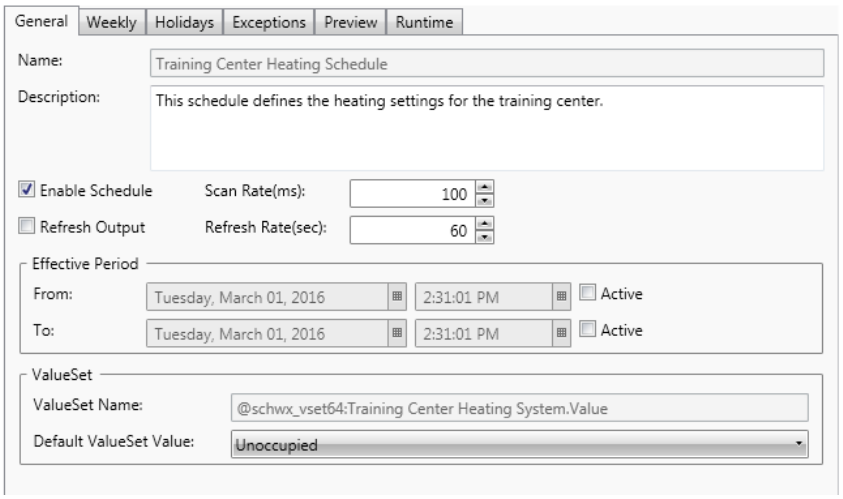


Figure L16.27 – Schedule Control in Runtime Mode

14. Leave this GraphWorX64 display and continue to the next lab section.

16.6. Manual Override

The ability to define weekly schedule of events, holidays, and planned exceptions can be done in the schedule as described previously. Often times, however, the operator needs the ability to manually stop the schedule and force a different state for the resources that are controlled by the schedule. For example, a schedule for the lighting system may be configured to turn off the lights in the office at 6:00PM, however, the staff unexpectedly needs to stay for an additional hour to complete an important assignment. By using the Manual Override feature in ScheduleWorX64 you are able to change the current value of the schedule, and hold that overridden state for a specific length of time, or indefinitely.

The Manual Override takes precedence over any Exception events, Holiday events, Weekly events, or a default ValueSet Value.

1. Start with the GraphWorX64 display still open from the previous section.
2. Click on the **Runtime** tab.
3. Click on the **Override** button.

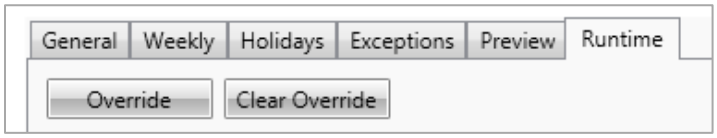


Figure L16.28 – Override Button

4. Configure the override to set the room as **Unoccupied** for **30 minutes**.

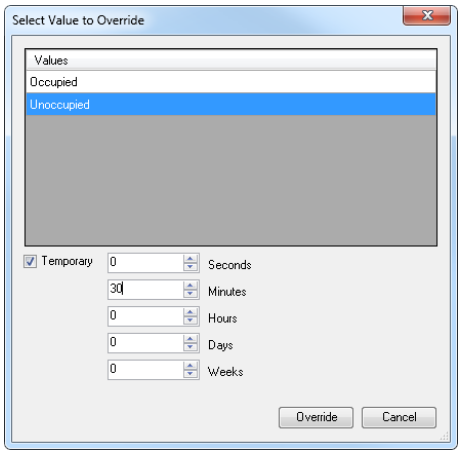


Figure L16.29 – Configuring the Override Properties

5. In the **Runtime** tab, you will see that the state of the schedule is updated.

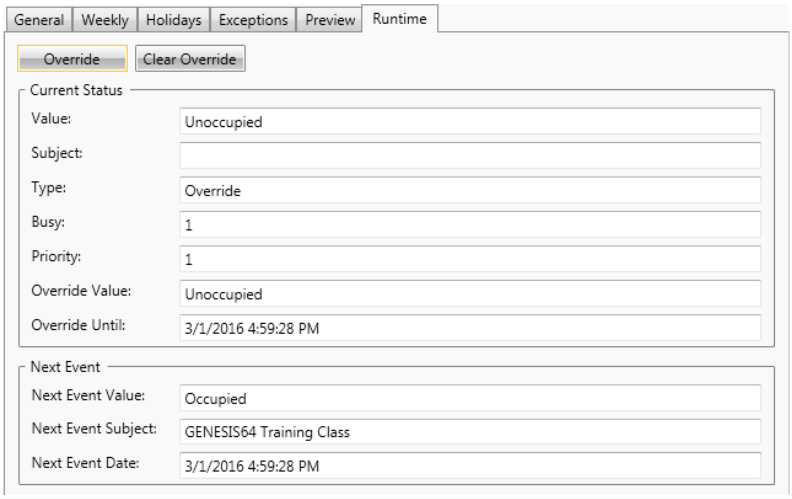


Figure L16.30 – Overridden Schedule in Runtime Mode

6. The set Override will remain in place for 30 minutes, and then it will expire and return to the normally scheduled event. An Override can be manually cleared to return the schedule to the normal running state. Press the **Clear Override** button to manually clear the Override.

7. Press **OK** on the confirmation dialog:

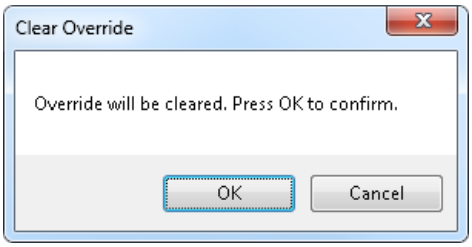


Figure L16.31 – Clear Override Window

8. You will see that the state of the schedule has returned to the regularly scheduled Weekly event:

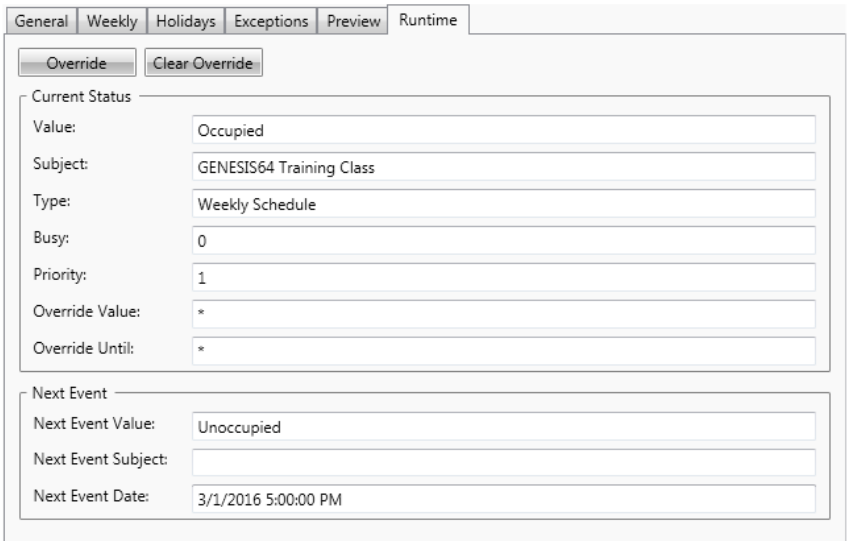


Figure L16.32 – Schedule Returned to Normal in Runtime Mode

<u>This concludes this lab.</u>

GraphWorX64 Shortcuts

A.1. Keyboard Shortcuts

A.1.1. File Functions

Keystroke	Function	Description
Ctrl+N	New	Creates a new display file.
Ctrl +O	Open	Opens an existing display file.
Ctrl+S	Save	Saves the current display file.
Ctrl+P	Print	Prints out the current display.

A.1.2. Edit Functions

Keystroke	Function	Description
Ctrl+Z	Undo	Reverses the last editing action.
Ctrl+Y	Redo	Restores the last editing action that was undone.
Ctrl+X	Cut	Moves one or more selected objects to the Windows clipboard, removing them from the work area. Cutting data/objects to the clipboard replaces the contents previously stored there.
Ctrl+C	Copy	Copies selected data/objects to the clipboard. This command is unavailable if no data/objects are currently selected. Copying data to the clipboard replaces the contents previously stored there.
Ctrl+V	Paste	Places a copy of the object currently in the clipboard into the work area. The objects remain on the clipboard until you copy or cut another object to the clipboard. Use this command to insert a copy of the clipboard contents at the insertion point. This command is unavailable if the clipboard is empty.

Keystroke	Function	Description
Ctrl+D	Duplicate	Duplicates a selected object. When an object is duplicated, all of its attending properties accompany the duplicated object.
Delete	Delete	Deletes the selected object from the work area.
Ctrl+F	Find	Opens the Find dialog box. Enter a string to search for in the Find What field.
Ctrl+H	Replace	The Replace operation works in a similar way and replaces the Find What string(s) with the Replace With string(s). All dialog options are the same, except that the Replace dialog box has a field called Replace With.
Ctrl+A	Select All	Selects all objects in the display.

A.1.3. View Functions

Keystroke	Function	Description
Ctrl+Numpad+	Zoom In	Zooms in by increments of 50 percent.
Ctrl+Numpad-	Zoom Out	Zooms out by increments of 50 percent.
Ctrl+Shift+S	Zoom Selection	Zooms in on the selected object.
Ctrl+W	Fit to Window	Fits all objects inside the work area; toggles back to original view.
Ctrl+Shift+H	Default Zoom/Pan	Restores the default zoom setting.
Ctrl+F11	Main Menu	Hides/shows the menu bar.
Ctrl+R	Runtime Window Preview Mode	Enables you to see what a display will look like in runtime mode without actually entering runtime. This helps you set the proper display dimensions and zoom factors.

A.1.4. Runtime Functions

Keystroke	Function	Description
Ctrl+M	Runtime	Toggles configuration and runtime modes.

A.1.5. Arrange Functions

Keystroke	Function	Description
Ctrl+Shift+G	Snap > Toggle Grid	Hides/shows the grid lines in the work area.

A.1.6. Draw Functions

Keystroke	Function	Description
Ctrl+G	Group	Groups two or more selected objects in the work area into one item. The resulting object can then be moved as a single object.
Ctrl+U	Ungroup	Undoes the grouping of two or more grouped objects in the work area into individual objects.

A.1.7. Help Functions

Keystroke	Function	Description
F1	Help Topics	Launches the online Help for the application.
Alt+?	Help	Opens online Help for GENESIS64, its tools, and its applications.

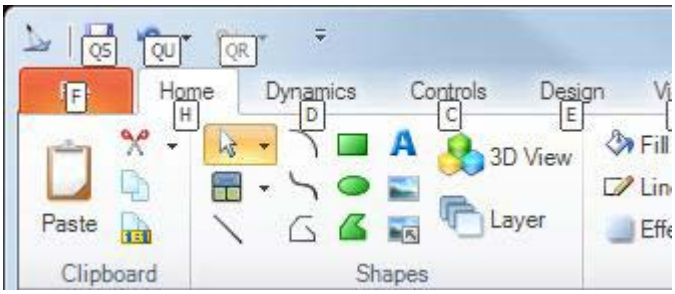
A.1.8. Move Object

Keystroke	Function	Description
Arrow Left	Move Left	Moves the object 1 pixel to the left.
Arrow Right	Move Right	Moves the object 1 pixel to the right.
Arrow Up	Move Up	Moves the object 1 pixel up.
Arrow Down	Move Down	Moves the object 1 pixel down.
Arrow keys	More vertices	When editing the vertices of a polyline, polygon, or path, pressing the arrow keys moves the selected vertices.

A.1.9. Resize Object

Keystroke	Function	Description
Ctrl+Arrow Left	Upsize horizontally	Decreases the size of the object horizontally.
Ctrl+Arrow Right	Downsize horizontally	Increases the size of the object horizontally.
Ctrl+Arrow Up	Upsize vertically	Increases the size of the object vertically.
Ctrl+Arrow Down	Downsize vertically	Decreases the size of the object vertically.
Ctrl+Shift+Arrow Left	Downsize object	Decreases the size of the object (proportionately).
Ctrl+Shift+Arrow Right	Upsize object	Increases the size of the object (proportionately).

A.1.10. Menu Commands

Keystroke	Function	Description
Alt	Menu key equivalents	Displays the values that will activate the menus when you press Alt+*key*.. 
Alt+Spacebar	Windows command	Activates the Windows command menu's Move, Restore, Resize, Minimize, and Maximize commands.
Alt+Q+S	Save	Performs the Save command.
Alt+Q+U	Undo	Performs the Undo command.
Alt+Q+R	Redo	Performs the Redo command.
Alt+F	GraphWorX64 menu	Opens the application menu with New, Open, Save, Save As, Import, Publish, Print, Tools, Login/Logout, and Exit commands.
Alt+H	Home ribbon	Activates the Home ribbon's Clipboard, Shapes, Styles and Colors, Arrange, and Text commands.
Alt+D	Dynamics ribbon	Activates the Dynamics ribbon's Data, Text Objects, Buttons, Dimensions, Color, Selectors , This Display, and Smart Symbols commands.
Alt+C	Controls ribbon	Activates the Controls ribbon and gives you access to GENESIS64 Viewers, SmartPins, PushPins, and the Scale Control.
Alt+E	Design ribbon	Activates the Design ribbon's Display Settings, Templates, and Transitions.

Keystroke	Function	Description
Alt+V	View ribbon	Activates the View ribbon's Navigation, Zoom, Grid, Window Layout, Show/Hide, and Misc commands.
Alt+R	Runtime	Takes you into Runtime mode.

A.2. Mouse Functions

Function	Description
Left mouse click	Selects static objects, such as rectangles and ellipses, tools, and menus.
Ctrl+Left mouse click	Brings an object to the front
Double-click	Action is dependent on object type: Double-click on a polyline, polygon, or path to edit vertices. Double-click on group of objects to edit inside the group. Double-click on an arc object to edit the start/end angles of the arc. Double-click on a text object to in-place edit the text. Double-click on 3D viewport to activate 3D-edit-mode.
Shift+Left mouse click	Selects multiple objects one by one.
Right-click	Opens the context pop-up menu, which varies depending on which item is selected. Right-clicking on the empty display also shows some of the menu commands. Right-clicking is also used to dock, float, and hide toolbars.
Drag (while pressing left mouse button) an object or an object's handles	Moves, stretches, and resizes objects

Function	Description
Drag (while pressing left mouse button) on the background of the display	Rubber-band select: selects all objects in the dragged area.
Shift+Drag (while pressing left mouse button) on the background of the display	Rubber-band select: adds all objects in the dragged area to the current selection.
Hold down **Alt** while the grid is on	Turns off snap-to-grid temporarily.
Drag and drop	Objects can be dragged and dropped to other instances of GraphWorX64 and the Symbol Library .
Ctrl+Drag (on a line)	While editing a polyline, polygon, or path, click and drag on the object to add a new vertex.
Right-click (on a vertex)	Right-click on a vertex to delete the vertex.
Shift+Click (on a vertex)	Hold down the Shift key when clicking on existing vertices to select multiple vertices (or mouse-drag-select multiple vertices)
Alt+LeftClick (on the background)	Zooms in by a factor of two.
Alt+RightClick (on the background)	Zooms out by a factor of one half.
Alt+LeftDrag (on the background)	Zooms in to the dragged area.
Alt+MiddleDrag (on the background)	Pans/scrolls the view.

Alt+mousewheel	Zooms in/out.
Function	**Description**
Tab	Select Next Object Selects the next graphical object.
Shift+Tab	Selects the previous graphical object.
Shift while drawing	While creating a rectangle or ellipse, holding down the **Shift** key makes the width and height equal, producing a perfect square or circle.
Shift while drawing	While creating a line, polyline, polygon, or path, hold down the **Shift** key to restrict line segments to increments of an angle. The default of 15 degrees can be changed using the Preferences > General Options > RestrictAngle property.
Shift while dragging	While dragging the handles of an arc to change its size, hold down the **Shift** key to restrict angles to increments of an angle. The default of 15 degrees can be changed using the Preferences > General Options > RestrictAngle property.
Ctrl while creating a curve	While creating a path (curve), hold down the **Ctrl** key to add a straight line segment instead of a curved segment.

Delimiters and Syntaxes

Historical Datasource Prefixes	Module
@ICONICS.TWXSQLSvr64.1\	TrendWorX64 Historical
awx:	AlarmWorX64 Historical
@ICONICS.HyperHistorian.1\	Hyper Historian64 Historical

Alarm Datasource Prefixes	Module
@ICONICS.AlarmSvr.1.Server	AlarmWorX64 Server
@GenEvent.1.Server	GenEvent Server
awxgs:	Global Alarms

Datasource Prefixes	Provider
ac:	Equipment Property from AssetWorX
lic:	License FrameWorX Plug-in tags
sim64:	Simulation FrameWorX Plug-in tags
localism:	Simulation tags from GraphWorX64
rdcy64:	Redundancy FrameWorX Plug-in tags
bacnet:	ICONICS Bacnet OPC Server FrameWorX Plug-in tags
snmp:	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
csch64	ScheduleWorX64 Schedules Plug-in tags
rcp:	RecipeWorX Plug-in tags
mobile:	Mobile Layouts Plug-in tags
alert:	Alert Point Manager Plug-in tags
svrsim:	Server Internal Simulator Plug-in tags
ws:	Web Services Plug-in tags
ua:	OPC UA Plug-in tags
db:	DataBase SQL Server Plug-in tags
hms:	Health Monitor Plug-in tags
$info:	License Information Plug-in tags
mwx:	Redundancy Information Plug-in tags
@ICONICS.HyperHistorianDA.1\	Hyper Historian DA Plug-in tags
@ICONICS.AlarmSvr_.1\	AlarmWorX Server DA Plug-in tags

Beginning Delimiter	End Delimiter	Function
/?	?/	Equipment Class Alias definition from AssetWorX
{{	}}	OPC Tag delimiters in an expression
<#	#>	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*.

GridWorX64 *D*

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 provides an interface to configure the desired queries and stores the configuration~~information~~ in its configuration ~~a~~ database. ~~that is queried~~ The configuration information is then used by the GridWorX64 Runtime Server ~~as~~ during runtime operation. GridWorX64 Viewer is a front end client component that~~and~~ allows you to visualize~~see~~ data from the GridWorX64 Server data sources or any dataset based data sources~~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

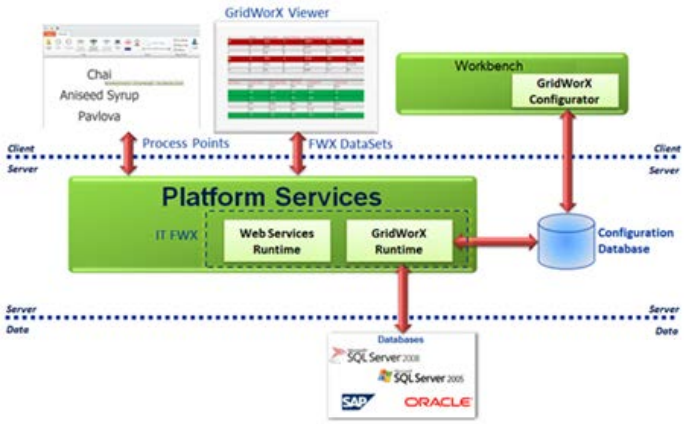


Figure D.1 – GridWorX64 Architecture

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, Microsoft Access and 32-bit Microsoft Excel are not supported. However, 64-bit is supported.

D.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 Desktop. ~~Expand~~From the provider list in the Explorer tree on the left, expand to MyProject → Data Connectivity → Databases~~ and click on the GridWorX64 Server icon to open the provider~~. Expand the ~~GridWorX64 Server~~Databases ~~localhost~~ branch and ~~on the left. Y~~you should see your GridWorX64 Server as shown in Figure D.2.

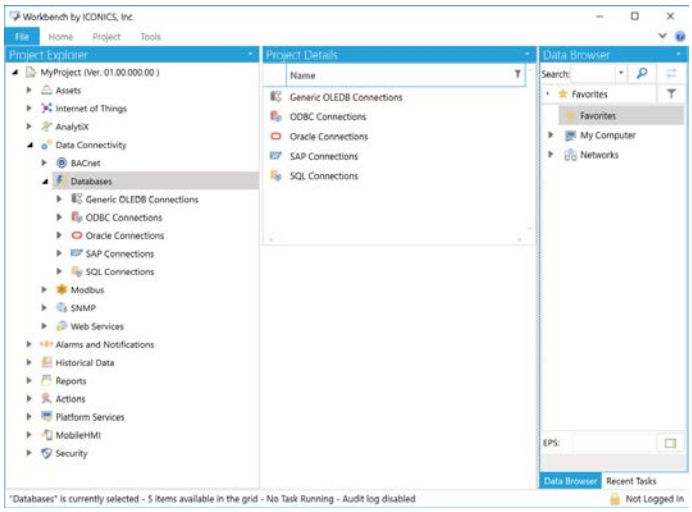


Figure D.2 – GridWorX64 Server

Expand the SQL ~~Server~~ Connections → Northwind. Inside the configuration, you can find ~~four types of GridWorX64 objects~~data sources and data manipulators.~~:~~

- ~~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.

D.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~~Then 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. Right click on SQL Connections and choose ~~Select~~ "~~New Database Connection~~Add SQL Server Native Connection" from the dropdown list. You can alternatively select the Add button from the Ribbon Menu, as is shown in the figure below.

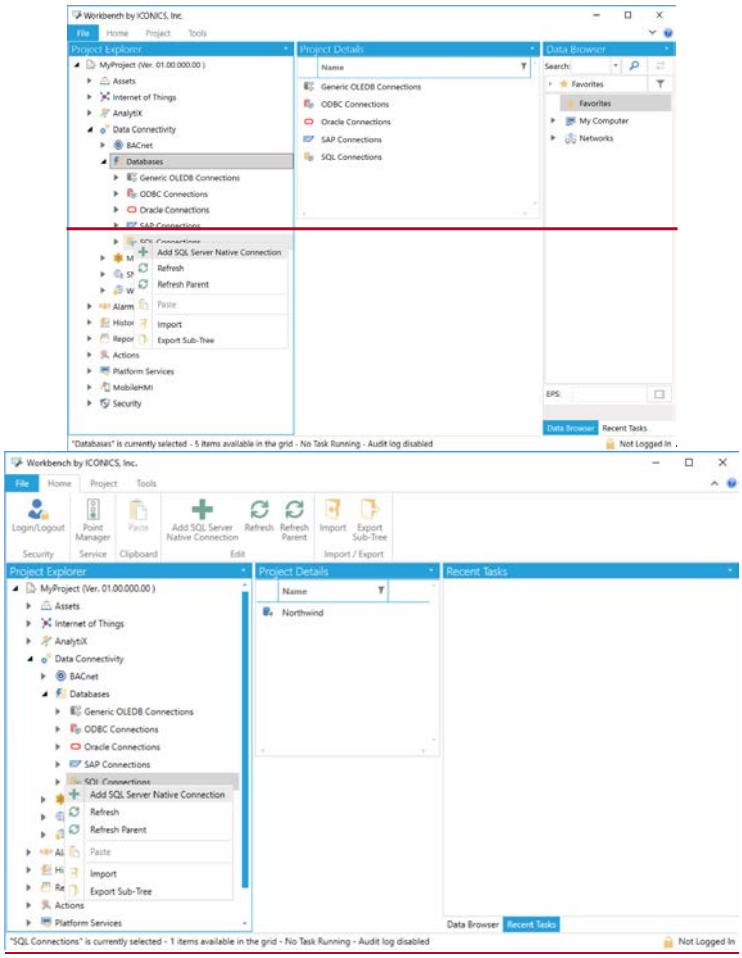


Figure D.3 – Adding New Connection

To build a Connection String, click on the ~~"Change~~ Add SQL Server Native Connection~~Connection~~" ~~button~~ item. ~~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~~Settings" dialog, ~~double~~ click on "~~SQL Server Connections~~Configure Connection" link ~~. Click on the "Add Connection String" + button~~ to open up the "SQL Server 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.

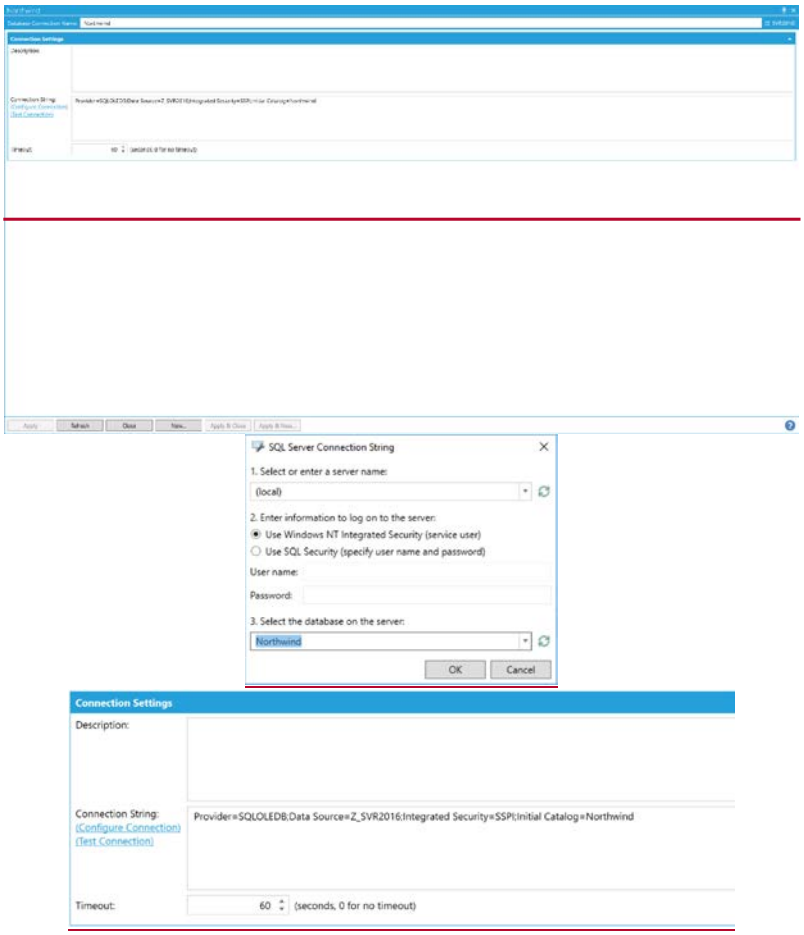


Figure D.4 – Connection Strings

Verify you have a successful connection by clicking on the "Test Connection" ~~button~~link.

D.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~~ Add Data Source"~~'~~ from the drop down list. You have the 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 in the Data Refresh tab:

- **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**~~ **By Trigger**~~**s**~~ - 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.

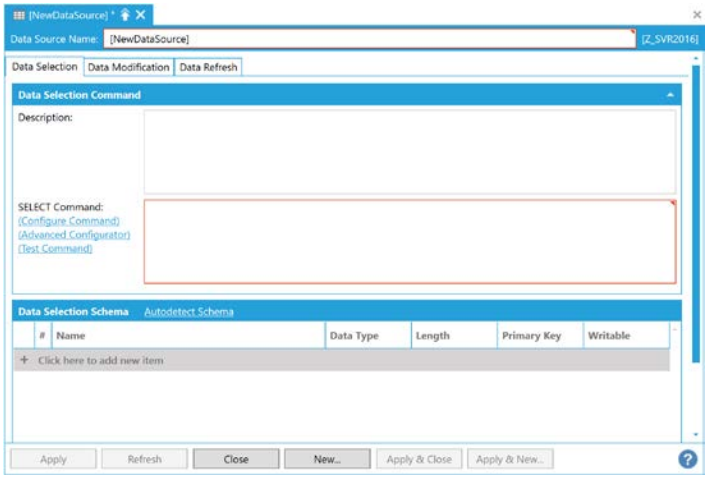


Figure D.5 – New Data Source Dialog

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.

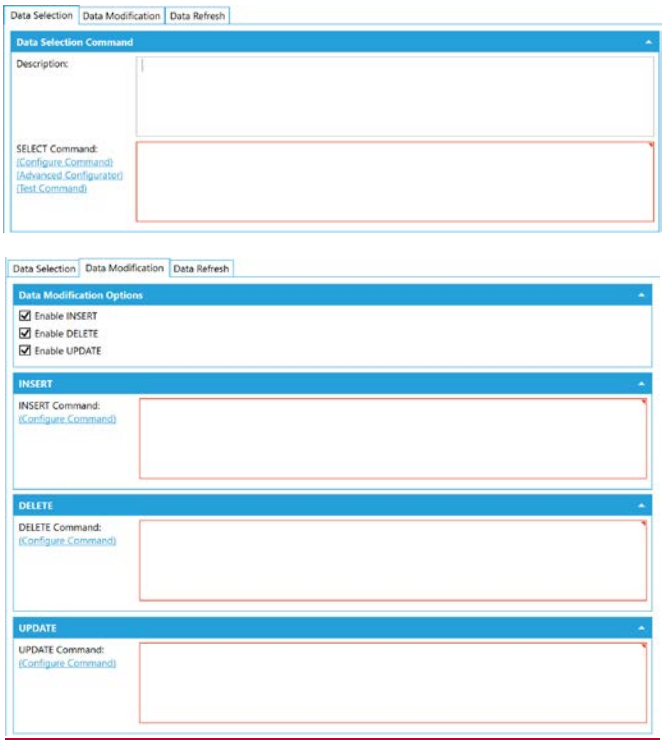


Figure D.6 - All Database Commands Selected

There are four command text editors available: Custom Query, ~~Simple SQL Designer~~Table Query, ~~Advanced SQL Designer,~~ Stored Procedure~~ Call and Manual SQL Editor~~. For the scope of this class, we'll focus on the ~~Simple SQL Designer~~Table Query. Please refer to the GENESIS64 Help documentation for information on the additional text editors.

Figure D.7 – Simple SQL Designer Dialog

Inside the ~~Simple SQL Designer~~Configure SELECT Command dialog, ~~from the Select Table dialog, clicking on the~~ expand dbo → Tables ~~Name ellipse button~~ allows you to browse for the table you would like to work with. After specifying a table and clicking on "~~"OK~~Next~~"~~", ~~through a checkbox selection interface you can choose table columns~~the resulting query with all columns selected is created for you.

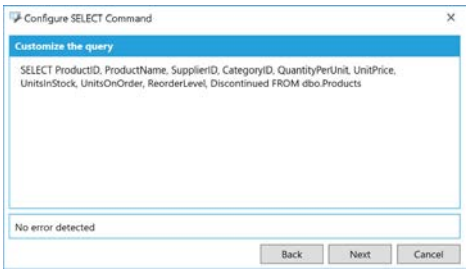
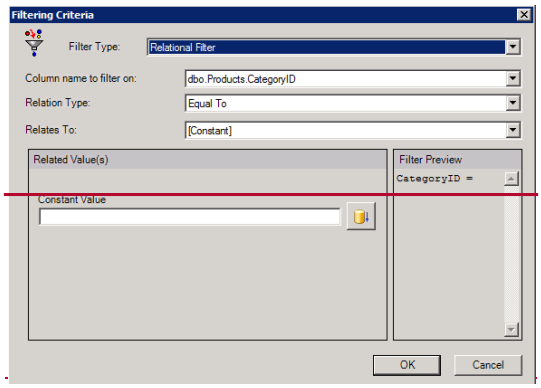


Figure D.8 ~~–~~ Select ~~query fing Columns f~~rom the Products Table

~~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 D.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~~"Test Command" link 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 "Add ~~a new~~ D~~d~~ata I~~i~~tem". 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

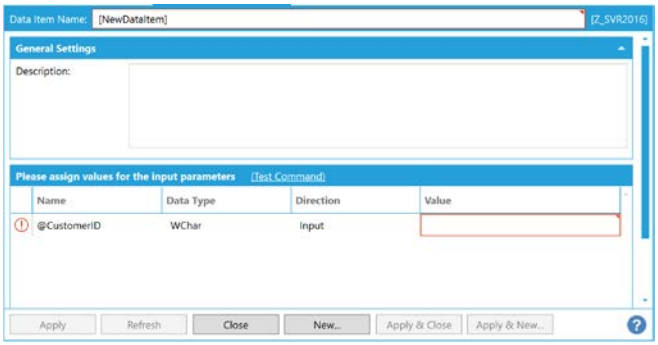


Figure D.10 - GridWorX64 Data Item Dialog Box

3. Click **Apply**. Click the "Test ~~Data Source~~Command" link ~~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~~ Add 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 "Configure Commandbutton [...]" link to open the Configure Manipulator Command dialog box.and chose Browse Table for Delete to open the Select Table dialog box.

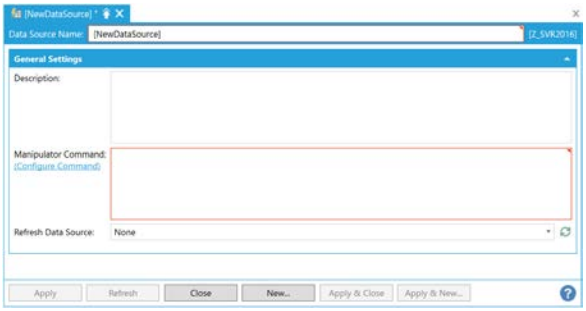


Figure D.11 – Creating a Data Manipulator

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, and choose "For DELETE"..

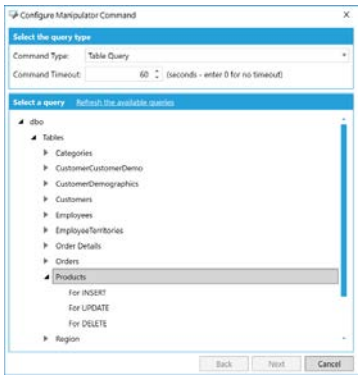


Figure D.12 – Select Table Screen

5. Click on ""OKNext"" to create the Data Manipulator.

Formatted: Font: Not Bold

Formatted: Font: Not Bold

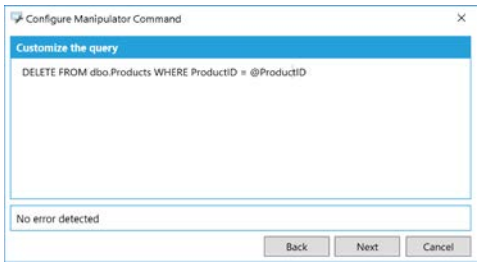


Figure D.13 - Data Manipulator Screen

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.

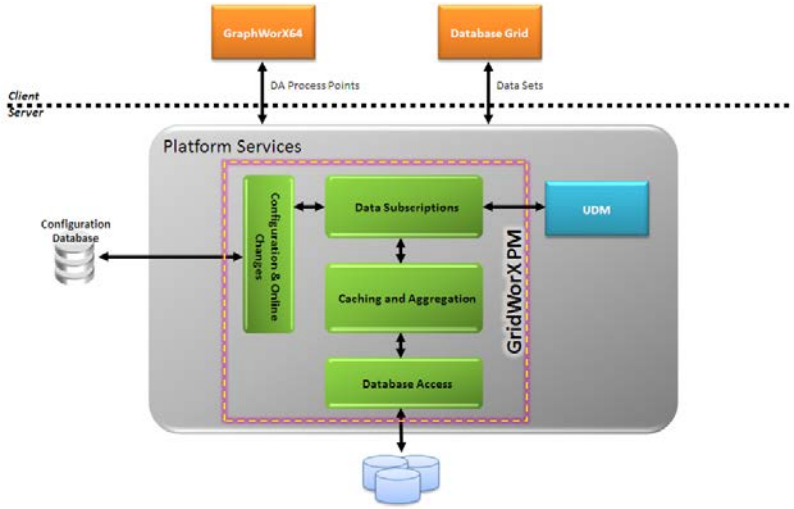


Figure D.14 - GridWorX64 Runtime Architecture

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.

~~*D.2 Create a Grid Using GridWorX64*~~

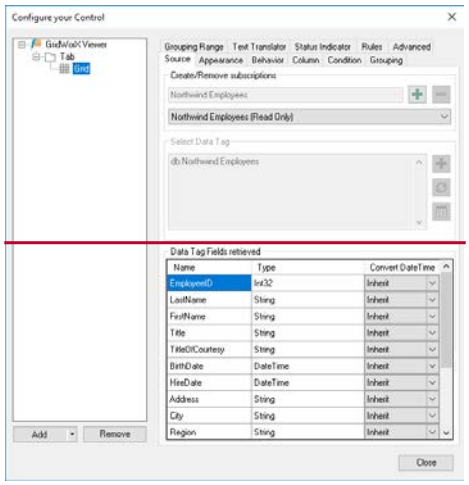
~~To create a grid using GridWorX64 in the **GENESIS64 Workbench**:~~

~~**4.** From the Windows desktop, open the Workbench by selecting Windows Start Menu →All Programs → ICONICS →Workbench.~~
~~**4.** In the Project Explorer, click on the GridWorX64 provider button bar to bring up GridWorX64.~~
~~**4.** 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 D.15 – New Grid Button on Home Tab**~~

Formatted: Indent: Left: 0"

~~**4.** Go to the **GridWorX64 Configuration** ribbon and click the **Configure Viewer** button. This opens the configurator where you can configure the grid.~~



~~**Figure D.16 - GridWorX64 Configuration**~~

~~**4.** Close the configurator when you are done.~~
~~**4.** Use the File menu to save the project with the grid in it.~~

~~D.23~~D.2 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 D.17 - GridWorX64 Viewer Control in GraphWorX64

4. Click the **GridWorX~~64~~ 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 **GridWorX~~64~~ Configuration** ribbon and click the **Configure ~~Viewer~~** button. This opens the configurator where you can configure the grid.

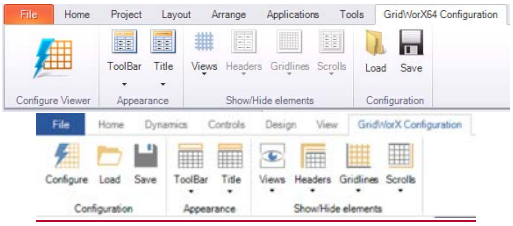


Figure D.18 - GridWorX64 Configuration Ribbon

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:

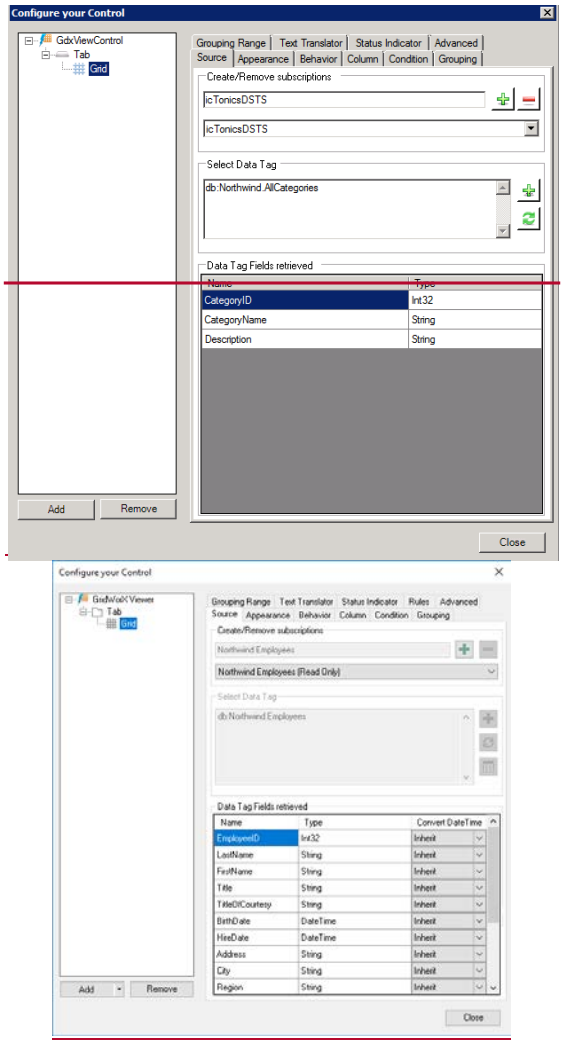


Figure D.19 - GridWorX64 Viewer Control Configuration Dialog

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 Dach 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 the Appendix D Lab in this manual for an example of how this interaction is performed.

Lab: GridWorX64 Configuration LD

D.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

D.2. Task Checklist

In this lab, you will complete the following tasks:

- You will create a New Database Connection, Data Source, Data Item and Data Manipulator
- View your configured data in GraphWorX64 using the GridWorX64 Viewer Control or a Process Point

D.3. Support Files

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

- Microsoft Northwind database that was installed with GENESIS64

D.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 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 a data source, data item and data manipulator.

D.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 to **MyProject → Data Connectivity → Databases**. Expand **Databases**.

3. Under **Databases**, select **SQL Connections**. Right click and select **Add SQL Server Native Connection**.

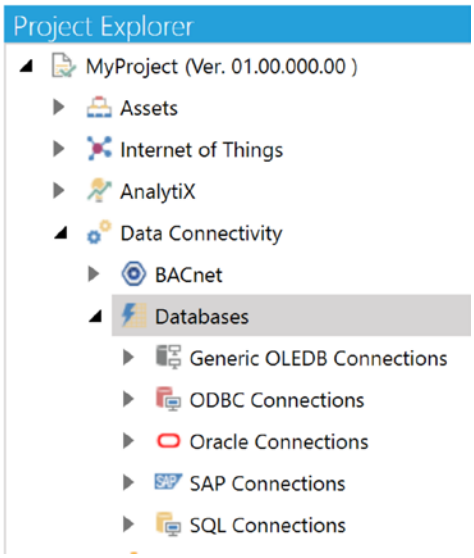


Figure LD.1 - GridWorX64 Server Explorer Tree

4. Name the Database Connection **icTonicsNorthWindDC** and click on **Apply**.

5. Click on the **Configure Connection** link.

6. Enter your SQL Server name, log in information and choose the Northwind database to connect to.

7. Your finished configuration should look like the figure below. Click on **OK**.

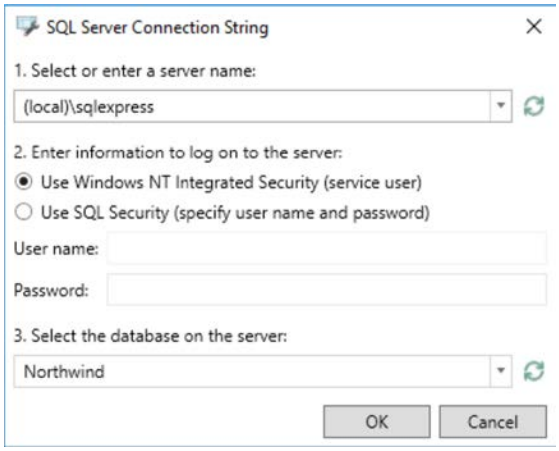


Figure LD.2 - SQL Connection String Syntax

8. ***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

9. Click on the **Test Connection** link. If you do not see the **Connection test succeeded** message, then review your configuration. Then click on **Apply**.

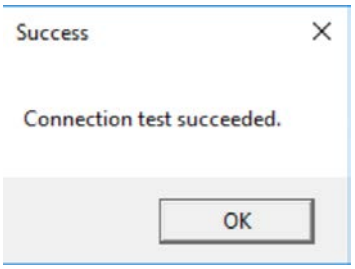


Figure LD.3 – Successful Connection Test Dialog box

D.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 **Add Data Source**.
2. Name this data source **icTonicsProductsDS**.
3. From the **Data Selection** tab, click on the **Configure Command** link on the left and select **Table Query** from the drop down.
4. From the **Select a query** section, expand **dbo → Tables.**
5. Select **Products**
6. Click **Next**
7. Paste the following query into the dialog:

   SELECT SupplierID, QuantityPerUnit, UnitsInStock, UnitsOnOrder, ReorderLevel, Discontinued

   FROM dbo.Products

   WHERE CategoryID=@CategoryID

8. Click on **Next** to continue.
9. The Configure Parameters dialog shows the **@CategoryID** parameter that we inserted.
10. Click Finish.
11. The "Schema Auto Detection" dialog should now show up. Since our query has a parameter, we need to provide a value that will be used to query the database to obtain the result schema.
12. Enter "**1**" as the default value for the **@CategoryID** parameter and click Ok.
13. The result schema should now be detected and viewable at the bottom of the Data Source dialog.
14. In the Data Source dialog, select the Data Modification tab.
15. Check "**Enable UPDATE**".
16. In the UPDATE query dialog that shows up, click on the "**Configure Command**" link.
17. In the query dialog, paste the following query:

UPDATE dbo.Products SET ProductName = @p1, CategoryID = @p2, UnitPrice = @p3 WHERE ProductID = @pKey1

18. Click **Next**

19. The four command parameters should be automatically populated as in the following figure.

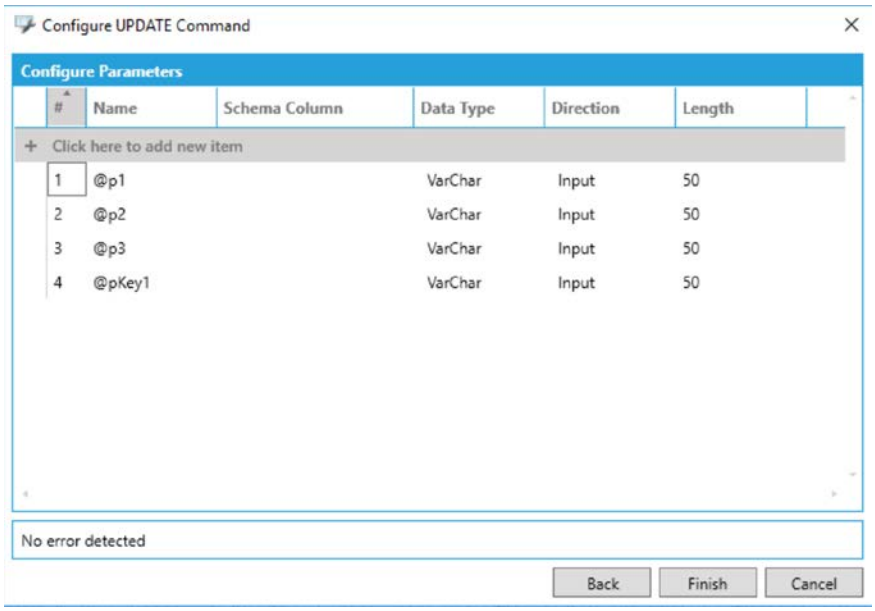


Figure LD.5 – Database Command

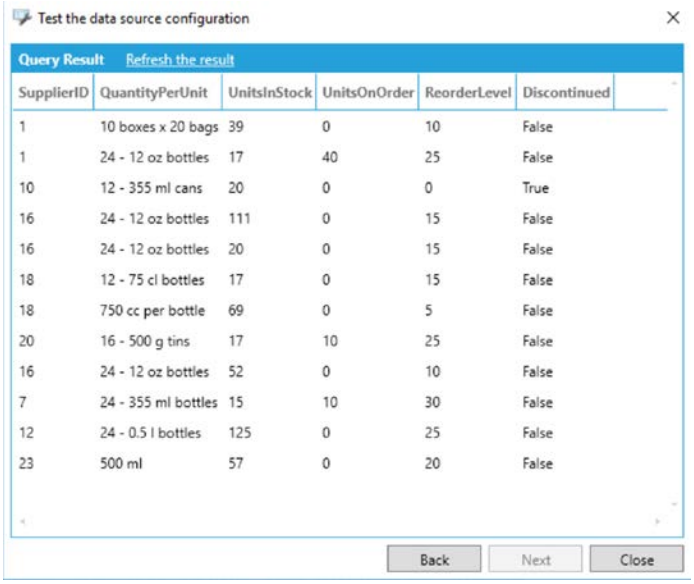
20. Click on **Finish** to finalize your configuration.

21. Let's execute the query to confirm proper operation.

22. Click on the **Data Selection** tab. Click on "**Test Command**" to execute this query.

23. Enter **1** in the Value field and click on **Next**.

24. Your **Query Results** should look similar to the figure below. Note the values in the CategoryID column.



SupplierID	QuantityPerUnit	UnitsInStock	UnitsOnOrder	ReorderLevel	Discontinued
1	10 boxes x 20 bags	39	0	10	False
1	24 - 12 oz bottles	17	40	25	False
10	12 - 355 ml cans	20	0	0	True
16	24 - 12 oz bottles	111	0	15	False
16	24 - 12 oz bottles	20	0	15	False
18	12 - 75 cl bottles	17	0	15	False
18	750 cc per bottle	69	0	5	False
20	16 - 500 g tins	17	10	25	False
16	24 - 12 oz bottles	52	0	10	False
7	24 - 355 ml bottles	15	10	30	False
12	24 - 0.5 l bottles	125	0	25	False
23	500 ml	57	0	20	False

Figure LD.6 - Data Source Query Results

D.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 **Add Data Item**.
2. Name the data item **icTonicsDI**.
3. Assign a **Parameter Value** of **2** and click on **Apply**.

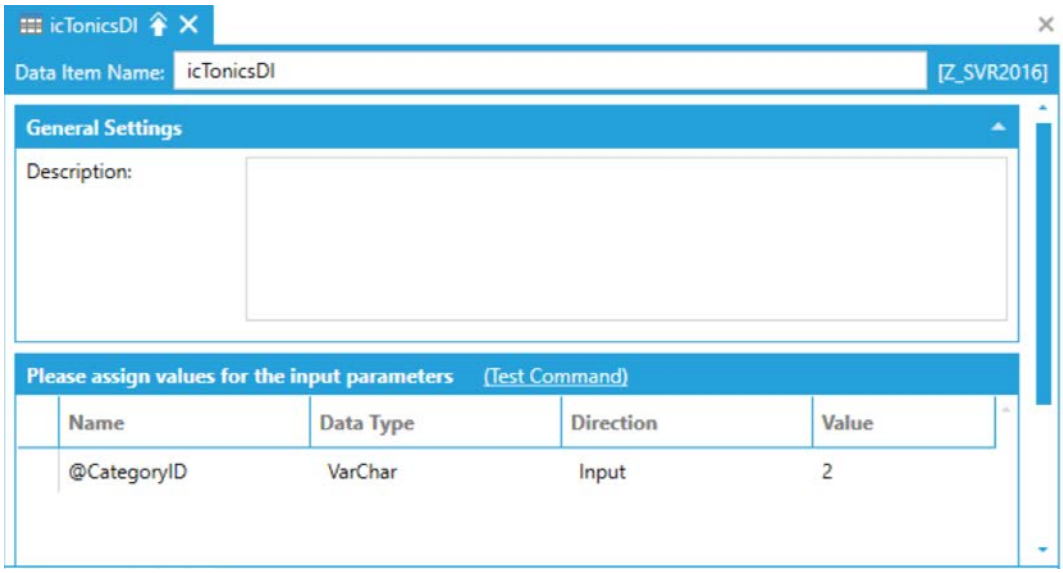


Figure LD.7 – Assigning a Parameter Value

4. You can click on "**Test Command**" to preview the Query Results.

D.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 are 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. Click on "**Configure Command**" link
4. In the "**Configure Manipulator Command**" dialog, under the Select a query section, expand **dbo → Tables → Categories**
5. Expand the **Categories** table and choose "**For INSERT**"
6. Click Next.

7. 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.

8. From the query dialog, remove the **Picture** and **@Picture** references. Remember to also remove the leading comma.

9. The following figure depicts the correct query.

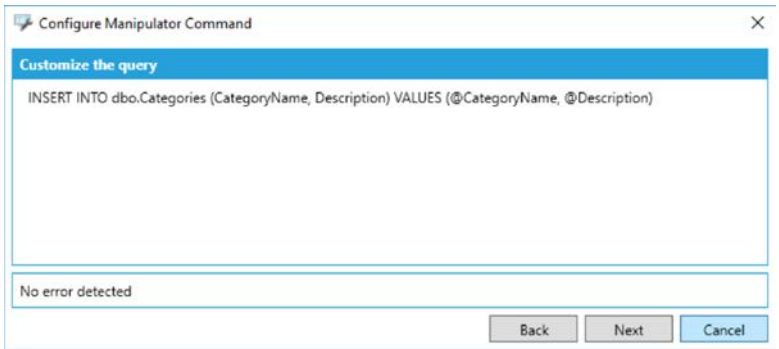


Figure LD.9 – Correct Insert Query

10. Click Next and Finish.

11. In the **Refresh Data Source** dropdown, select **icTonicsProductsDS**.

D.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 thusfar 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.

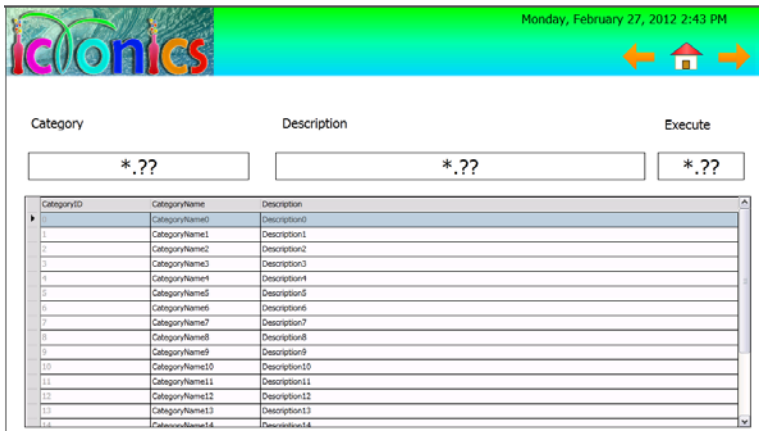


Figure LD.10 - Example icTonics Gridview Display

D.5.1. Adding GridWorX64 Viewer

1. Open up GraphWorX by going to Start → All Programs → ICONICS Product Suite → GraphWorX64.
2. 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.

D.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 dropdown list.
3. Click on **My Computer** → **Data Connectivity** → **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.

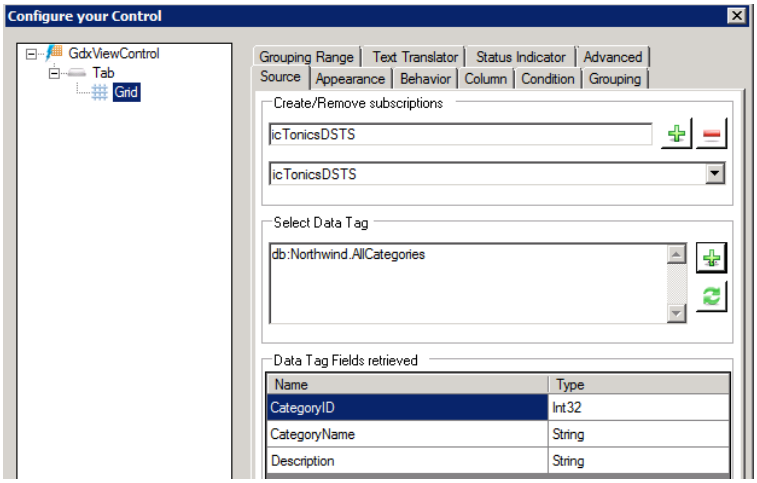


Figure LD.11 - GridWorX64 Viewer Source Configuration

6. Click Close to save this configuration. Save this display with the name **icTonicsGridView.gdfx**

7. Go into **Runtime** and you should see 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 LD.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.

D.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 **My Computer → Data Connectivity → 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 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.

D.5.4. Using Filters in Runtime

1. Let's exit Runtime and return to the GraphWorX64 Configuration 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. Select the **Grid** in the tree on the left and click on the **Behavior** tab.

3. 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.

4. To the right of the **Select Default Filter** field, click on the to create a new filter. Name the filter **CategoryIDLessThan5**.

5. 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**.

6. Your completed configuration should look similar to the follow figure. Click on **OK**, then close the configurator and enter **Runtime**.

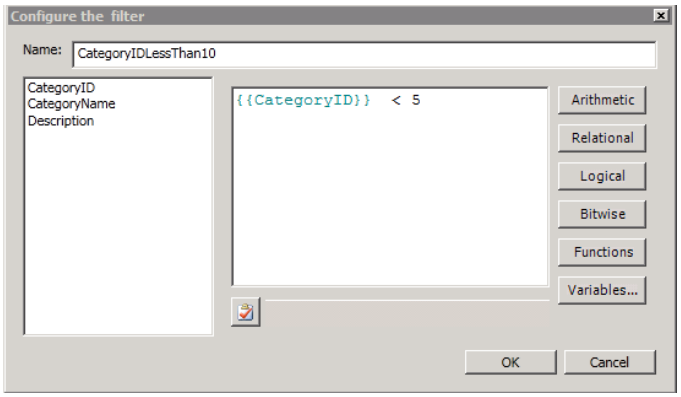


Figure LD.13 - Setting a Behavior Filter for a GridWorX64 Viewer

7. Notice that, when the filter is applied, you will only see the **CategoryID** records 1 through 4.

8. 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.

Lab: KPIWorX

LG

G.1. Lab Overview

Management at icTonics wants to create an on-demand, self-service dashboard to view facility KPIs (Key Performance Indicators) from any location.

Estimated time to complete this lab: **20 minutes**

G.1.1. Task Checklist

In this lab, you will complete the following tasks:

- Build a simple KPIWorX display.
- Connect to data.
- Customize the appearance of KPIs.
- View and configure KPIs in runtime.
- Save a KPIWorX display.
- Share a KPIWorX display.
- Import a symbol from GraphWorX64 into KPIWorX.

G.1.2. Support Files

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

G.2. KPIWorX Runtime Configuration

You can use the KPIWorX runtime option to view real-time or historical data in GraphWorX64 runtime. In this lab, we will create a KPIWorX display with several tags and appearance modifications.

G.2.1. Creating a KPIWorX Display

1. Open **GraphWorX64** if you closed it after the previous lab.

2. Begin with a **New** display and select **Runtime** in the top right corner causing the display to enter runtime mode.

3. Select the **KPIWorX**

icon within the **Tools** section of the **Runtime** ribbon.

4. This will launch a new KPIWorX display. The current cell should encompass the entire canvas.

5. Selecting any symbol or control will automatically populate the cell. Begin by selecting **KPI Gauge 1** from the General section of the Symbol Library.

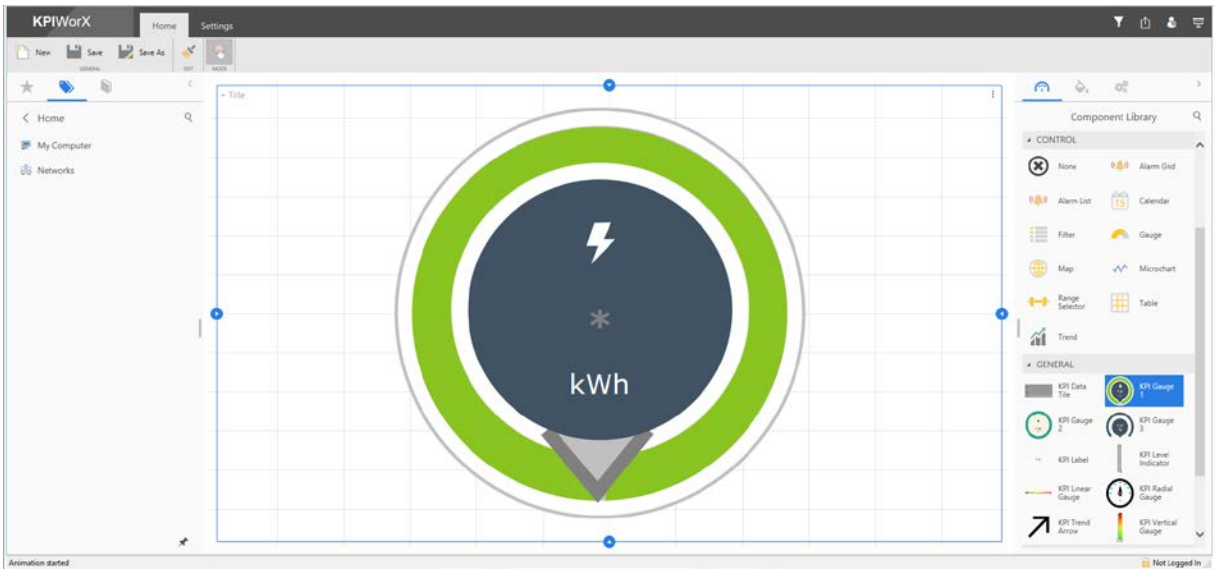


Figure LH.1 – Selecting a Symbol

6. This gauge will instantly populate the cell. To add data to the cell, select the **Tag** icon on the left. Navigate to **My Computer** → **Data Connectivity** → **OPC Data Access** → **Simulations** → **Double** → **Random**. Drag this data source to the gauge.

7. Select the **Container Settings** tab at the upper right of the KPIWorX Configuration Menu. From this menu, you can customize every aspect of the cell. Set the **DisplayTitle** = **Random**, **FontSize** = **24**, and leave Title **Foreground** and **Background** = blank. Set the **Alignment** = **Center**. The Appearance **Background** should be left blank.

8. Select the **Widget Settings** tab within the KPIWorX Configuration Menu. The **DataSource** should reflect the source of data selected earlier: **My Computer → Data Connectivity → OPC Data Access → Simulations → Double → Random.**

9. Under the **General** section, set **ForeColor** to **Black**. Leave **BackColor** on **Dark Blue**, **NeedleColor** on **Green**, and **Icon** set to **1**.

10. Under the **Data** section, set **Units** to **kWh** and leave **DecimalPlaces** at **2**.

11. Under the **Limits** section, leave **LowLimit** at **0** and **HighLimit** at **100**.

12. Now that this cell is fully configured, you can begin creating a second cell.

13. Select one of the **Arrow** from the side and drag the gauge over to make room for your second cell.

14. Click into this new section to highlight it.

15. Select the **Component Library** from the KPIWorX Configuration Menu. From the **Controls** section, select **Microchart**.

16. To add live data to the Microchart, select the **Tag** icon. Navigate to **My Computer → Data Connectivity → OPC Data Access → Simulations → Double → Sine**. Drag this data source to the Microchart. The Microchart should become active right away.

17. Select the **Container Settings** tab within the KPIWorX Configuration Menu. Set the **DisplayTitle** = **Sine**, **FontSize** = **24** and leave Title **Foreground** and **Background** = blank. Set the **Alignment** = **Center**. The cell **Background** should be left blank.

18. Select the **Widget Settings** tab within the KPIWorX Configuration Menu. The **Series** should reflect the source of data selected earlier: **My Computer → Data Connectivity → OPC Data Access → Simulations → Double → Sine**. Under the **Chart** section, leave the **Refresh Rate** = **3** and the **Interval** = **60**.

19. Under the **Series** section, click on the Data Source. This should open a new **Series options** tab.

20. Under **General**, change **Type** = **SplineArea**.

21. While still under the **Series options** tab, go to the **Appearance** section. Change **Thickness** = **5**.

22. Now that this second cell is fully configured, you can begin creating a third and final cell.

23. Once again grab one of the arrows and drag to create a third cell. Click this cell to highlight it.

24. From the **KPIWorX Settings ribbon** select **Symbols**. This will give you access to all of the symbol libraries.

25. Select **Building Automation** and click **Ok**.

26. Select the **Component Library** in the KPIWorX Configuration Menu. The **Building Automation** library should now be available.

27. Select the **KPI Boiler** for the final cell.

28. To add data, select the **Tag** icon . Navigate to **My Computer → Data Connectivity → OPC Data Access → Simulations → Double → Ramp**. Drag this data source to the boiler.

29. Select the **General Settings** tab within the KPIWorX Configuration Menu. Set the **DisplayTitle** = **Boiler Temperature**, **FontSize** = **24** and leave Title **Foreground** and **Background** = blank. Set the **Alignment** = **Center**. The cell **Background** should be left blank.

30. Select the **Widget Settings** tab within the KPIWorX Configuration Menu. The **DataSource** should reflect the source of data selected earlier: **My Computer → Data Connectivity → OPC Data Access → Simulations → Double → Ramp**.

31. Under the **General** section, set **Subtitle** = **Boiler 1**. Leave **Color** = **black**.

32. Under the **Data** section, set **Units** = **F** and leave **DecimalPlaces** = **1**.

33. Select the **KPIWorX** menu and select **Save**. Save to the **Root** folder with **Visibility** set to **Public**, and title the lab **icTonics KPIWorX Display**.

G.2.2. Share a KPIWorX Display

1. Select the **Share** button from the top right opening the **Share** menu. Choose **Link** and then **Clipboard**.

2. This will copy a link to your dashboard to the Clipboard which you can then send to someone.

G.2.3. Importing a Symbol into KPIWorX from GraphWorX64

1. **Open** a new instance of **GraphWorX64**.
2. On the bottom of the Explorer, select the **Symbols** tab.
3. Within the Symbol **Search** bar, type **Dehumidifier 4** and press **Enter** key.
4. **Dehumidifier 4** should appear in the search results. **Drag** this symbol on to the canvas.
5. Select the **Dynamics** tab from the GraphWorX64 ribbon and select **Process Point**.
6. **Draw** a rectangular process point directly below the dehumidifier symbol.
7. **Select** both the symbol and process point. Within the **Arrange** section, select **Align → Align Center**.
8. **Group** the two items. Check the display **Explorer** on the left to ensure that the process point is grouped with the symbol correctly.

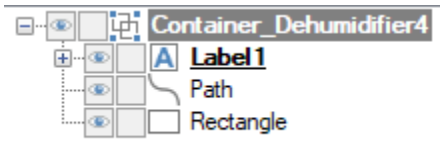


Figure LH.2 – Dehumidifier 4 Project Explorer

9. **Right-click** on the **Dehumidifier4** grouped symbol and select **Convert to smart symbo**l.
10. The **Automatic Smart Binding Options** window will appear. Ensure that the **Expose data sources as Smart Properties** box is **checked**. Select **OK**.
11. The **Edit Smart Properties** window will open. Select **OK**.
12. Select the **label** within the symbol and select the **Dynamics** tab. Right-click on **DataSource** and select **Add to Smart Properties…** to open the **Add Smart Property** window. Ensure that the **Property Name** is listed as **DataSource**. Select **OK**.

13. Right-click on **PostfixLabel** and select **Add to Smart Properties…** (Note that any property can be added as a **Smart Property** when configuring a smart symbol).

14. The **Add Smart Property** window should appear. Set the **Property Name** = **Units**. Select **OK**.

15. **Select** the whole grouped symbol then select the **Properties** tab.

16. Set the **Name** = **KPI_Dehumidifier**.

17. On the **Explorer**, select the **Symbols** tab. Close out of the latest search results.

18. Right-click within the window and select **Add New Category…** to open the **Category Properties**. Select **Browse…**

19. A new **Save As** window will open. Navigate to **Local Disk → Program Files → ICONICS → Genesis64 → WebSites → AnyGlass → Projects → KPIWorX**. Set the file name as **Training Library** and **Save**.

20. **Category Properties** should now reflect the set **Category File** location and the **Category Title**. Select **OK** to close this window.

21. **Drag** or Copy the symbol into the new library.

22. **Right-click** within the library and select **Save Selected Category**.

23. Select **Runtime** from the GraphWorX64 ribbon and select the **KPIWorX** icon from the **Tools** section of the **Runtime** ribbon. KPIWorX should launch. You will be prompted to save the display. Select **No**, as you will not need this display anymore.

24. Select the **KPIWorX Settings ribbon** and select **Symbols**. This will give you access to all symbol libraries allowing you to select **Training Library** and close the KPIWorX menu by selecting **Back**.

25. Select the **Symbol Library** in the **KPIWorX Configuration Menu**. **Training Library** should now be available.

26. **KPI Dehumidifier** should now be available like any other KPI symbol with a customizable Datasource and Units.

This concludes this lab.

Lab: Remote Configuration LH

H.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**

H.1.1. Task Checklist

In this lab, you will complete the following tasks:

- Publish displays
- View displays directly in a web browser
- View displays in a web browser, accessed through the AppHub

H.1.2. Support Files

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

- **icTonicsMain.gdfx**
- **icTonicsTankMonitor.gdfx**
- **icTonicsTrends.gdfx**

H.2. Viewing Displays using HTML5

The easiest way to publish most displays to view in a web browser is to use the HTML5 AnyGlass engine, an under-the-hood component of GENESIS64 that renders GraphWorX64 files into HTML5.

1. Copy your .gdfx file into **C:\Program Files > ICONICS > GENESIS64 > WebSites > PubDisplay**. This directory is configured during installation to be accessible over the internet. Only place files in this directory when you are ready for them to be accessed over the web.

2. Open a modern web browser, such as Google Chrome or Microsoft Edge.

3. Navigate to http://localhost/anyglass/PubDisplay/<YourFileName>.gdfx. If viewing from a remote machine, replace <localhost> with the name or IP address of your server.

   Note: If you do not know know your machine name, go to **Start → Command Prompt,** type *hostname* in the command prompt window and hit **Enter.** The machine name will populate on the next line.

A small number of features are not supported by the AnyGlass engine; these include scripting. A small number of additional features currently have partial support in the AnyGlass engine; these include Smart Symbols and 3D graphics.

If a display does not render in the way you expect in HTML5, one option is to save your display in the .gdfxp format, which will strip out unsupported components. The next steps are optional practice for doing so.

4. In GraphWorX64, return to **Configuration** mode for the file you wish to view in a web browser.

5. From the **File** menu, choose **Save As…**

6. From the **Save as type** dropdown menu, select *GraphWorX64 MobileHMI Display (*.gdfxp).*

7. After saving, you may be offered a log of significant changes that were made to your display; click **Yes** to view the log.

8. In your web browser, navigate to http://localhost/anyglass/PubDisplay/<YourFileName>.gdfxp (make sure to add the *p* at the end of the extension).

H.3. Viewing Displays using the AppHub

The AppHub is a landing page generated by the ICONICS platform, which can be used to launch directly into web-published displays. The AppHub is also used as the landing page in the MobileHMI app. Configuration for the AppHub is identical, no matter whether it will be accessed in a web browser or in the MobileHMI app.

The AppHub is useful for creating a unified portal for a large number of dashboards or applications. In order to ensure compatability with the MobileHMI app, only files saved in the .gdfxp format are accessible through the AppHub.

1. If you have not already done so, save your display in .gdfxp format by going **File > Save As** and choosing *GraphWorX64 MobileHMI Display (*.gdfxp)* from the **Save as type** dropdown menu.
2. If you have not already done so, move or copy your .gdfxp file into **C:\Program Files\ICONICS\GENESIS64\WebSites\PubDisplay**.
3. Open **Workbench,** and expand the **MobileHMI** provider.
4. Right-click on **Layouts** and select *Add Layout.* Layouts are used to segregate different views in the App Hub (for example, creating separate views for operators versus managers).
5. Name the layout *Training.* Click **Apply & Close.**
6. Right-click on **Training** and select *Add Group.* Groups are used to organize large numbers of tiles within the App Hub-- for example, into separate columns.
7. Name the group *Displays.*
8. Right-click on **Displays** and select *Add Tile.*
9. Name the tile *icTonics Main.*
10. Set the tile background to **green**; notice the preview tile on the right change color to match.
11. Under **Bottom Banner,** set the **label** to *icTonics Main.* Again, notice the preview tile change to match.
12. At the bottom of the tile configuration, under **Command,** choose from the dropdown list the **Load Graphics Display** command.
13. Click on the icon to the right of the *FileName* field to open the File Browser. Double-click on *PubDisplay* and choose your .gdfxp file. Click **OK.**

14. Click **Apply & Close.**

15. In the Project Explorer tree, right-click on the *icTonics Main* tile and select **Copy.**

16. Right-click on the *Displays* object in the tree and select **Paste.** This will create a duplicate *icTonics Main* tile. Double-click on this tile, and change its background color and change the label in the bottom banner. (It does not matter what you set either to.)

Next we will access the AppHub in a web browser to view our display. If loaded in the MobileHMI app, the AppHub would appear identically.

17. Open a modern web browser, such as Google Chrome or Microsoft Edge.

18. Navigate to http://localhost/anyglass/projects/apphub.ag. If viewing from a remote machine, replace <localhost> with the name or IP address of your server.

    Note: If you do not know know your machine name, go to **Start → Command Prompt,** type *hostname* in the command prompt window and hit **Enter.** The machine name will populate on the next line.

19. If you have multiple layouts configured, the AppHub will first load the *Layout Selection*; click on the **Training** tile if this is the case.

20. See the two tiles you configured earlier as they load into the App Hub.

    Note: If you only configure one tile in a layout, loading that layout will cause that tile's display to load directly. For example, if we had not duplicated our tile earlier, clicking on *Training* now would immediately load our dashboard, effectively skipping this step and the next.

21. Click on **icTonics Main** to load your dashboard.

H.4. Viewing Displays using a XAML Browser Application (XBAP) in Internet Explorer

Older applications, or applications with more limited scope, may be deployed using a XAML Browser Application solution. An XBAP temporarily enables Internet Explorer to use the resources of the client machine's CPU to render graphics using the same WPF technology that is used in GraphWorX64. As such, XBAP-based deployments do support the more advanced features not yet fully supported in HTML5, including scripting and 3D graphics.

XAML Browser Applications are only available in Internet Explorer, and so cannot be accessed from other browsers such as Chrome or Safari.

What follows are instructions for an optional exercise in publishing graphics using an XBAP solution. These include some extra steps for configuring IIS and your machine's FTP server. Publishing using HTML5 avoids the need for these extra steps.

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** and hit enter.)

3. Type **hostname** and the **Enter** key. The hostname below is HV-01. Leave the command prompt open for future testing.

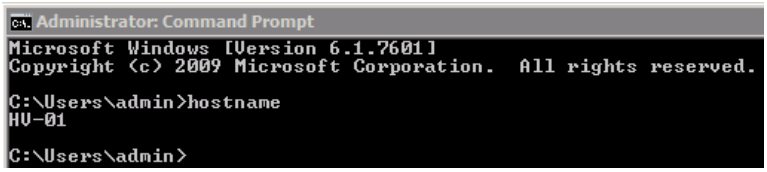


Figure LE.1 – Windows Command Prompt with Hostname Command

4. Launch **Internet Explorer** and type **http://<hostname>** (e.g. http://HV-01) into the address field. If you see the following picture after you launch the URL, it means your Internet Information Services (IIS) is working.

Figure LE.2 – Internet Information Services (IIS)

5. **Close Internet Explorer**.

Add FTP Publishing to the Default Web Site

1. From **Start** → **Control Panel** → **Administrative Tools** → **Internet Information Services (IIS) Manager** to open the IIS Manager. If there are two selections for IIS Manager, choose the one **WITHOUT** 6.0 in the name.

2. Expand the **Connections** navigation tree on the left until you can right-click on the **Default Web Site**.

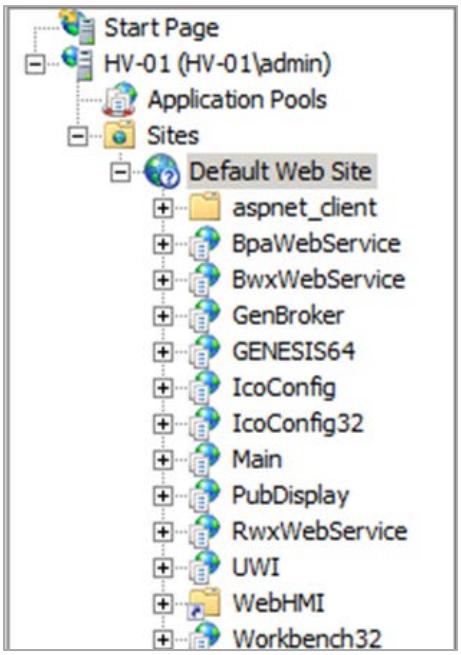


Figure LE.3 – Default Web Site

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** to save your changes.

   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 LE.4 – 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. Log in with the administrator level user account that you use to log in to your Operating System. If the command prompt returns this message "**230 User logged in**" after you enter your username and password, it means that your ftp connectivity is working. If any of the above steps do not work properly, ask your instructor for assistance. Close the Command Prompt window.

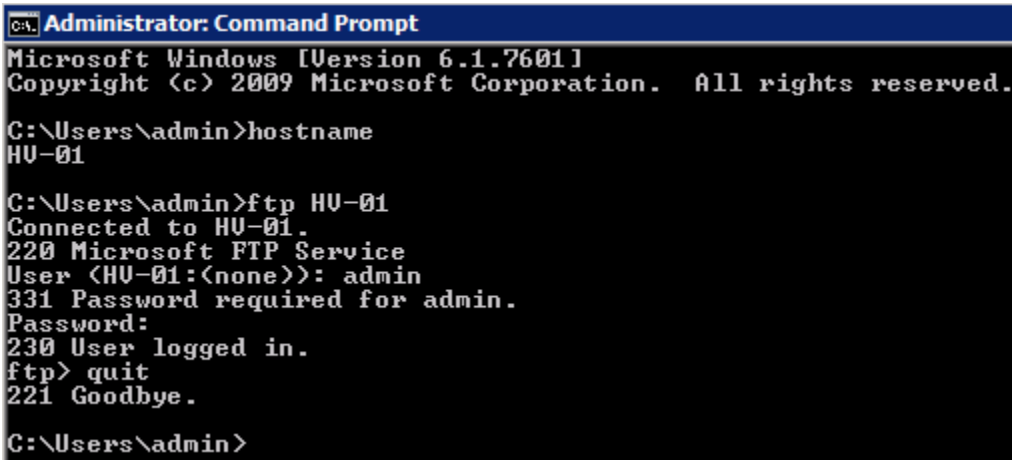


Figure LE.5 – Ftp Connectivity Test

Publish a Display

1. Open **GraphWorX64** 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**.

4. Make sure **No** is selected for "**Do you want to publish linked files?**"

   NOTE: Under **Add&Remove Files…**, you can add more Displays in the Wizard to the Publish Selection.

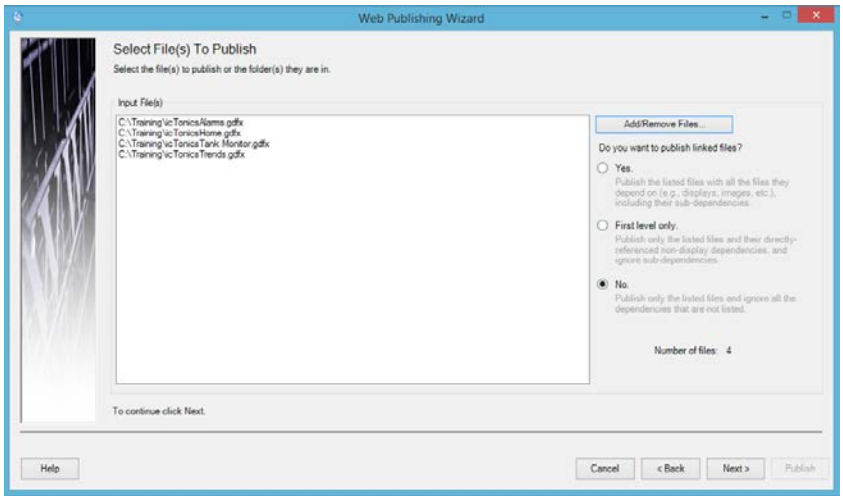


Figure LE.6 – Web Publishing Wizard: Select a File

5. Click on **Add/Remove Files…** A popup window will open.

6. Select **Add Files** and select the following files: **icTonicsAlarms.gdfx**, **icTonicsTank Monitor.gdfx**, **icTonicsTrends.gdfx**

7. Click on **OK** to add these displays to the Wizard Configuration.

8. Click on **Next** to select the **Web Server** to publish the four files.

9. Click on **Next** in the **Publishing Options** dialog. We will leave the configuration as default.

10. The next popup will show the files that will be published. Click **Next**.

11. In the **Select a Publishing Location** dialog box, you should see something like **ftp://HV-01/Pubdisplay**. Go back to Step 8 if the computer name is incorrect.

12. Alternatively, you could select **New**.

- 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.

- In the **Virtual Directory** section, point it to **/PubDisplay**. The PubDisplay FTP site was set up during the GENESIS64 installation.

- 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 LE.7.

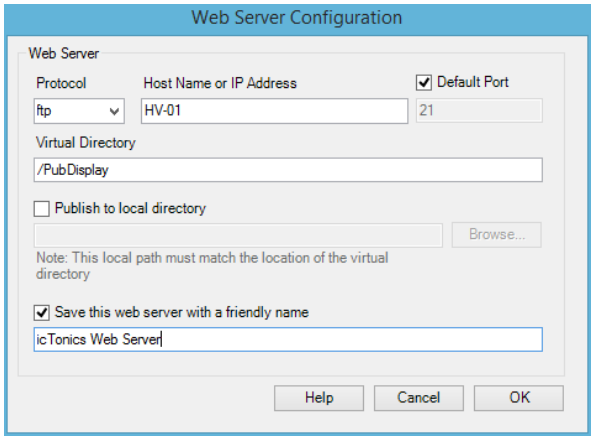


Figure LE.7 – Web Server Configuration

- Click on **OK** to finish the server configuration and return to the **Select a Publishing Location** dialog box.

13. Click on **Next** to proceed.

14. In the **User Name and Password** dialog box, enter a local administrator user name and password, and then click on the **Publish** button.

15. It may take a while to publish but, once it finishes, you should see a dialog box indicating the status of the publishing as shown in the Figure below.

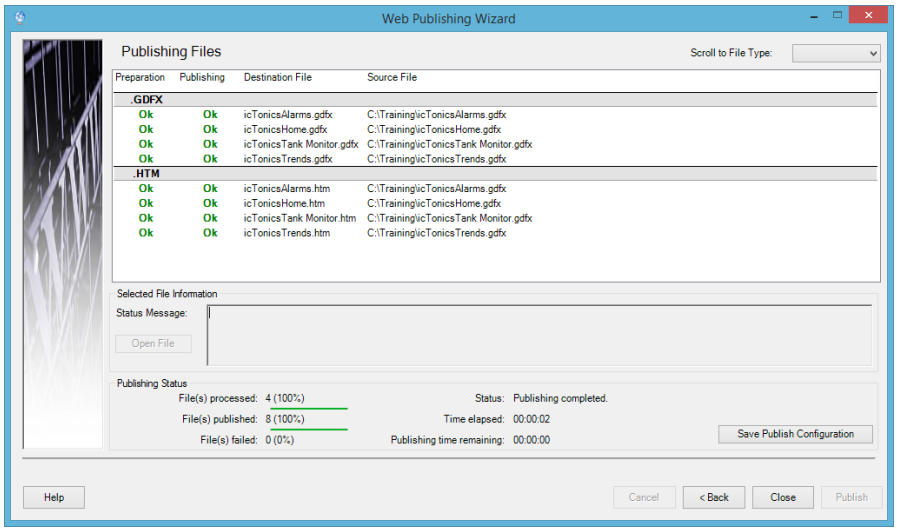


Figure LE.8 – Publishing Wizard

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.

16. Launch an **Internet Explorer** session.

17. Under **Tools → Internet Options → Security → Trusted Sites**, add your Server Name to the list of trusted sites.

18. You can view your displays in **Internet Explorer** by going to **http://*hostname*/PubDisplay/icTonicsHome.htm**. Feel free to navigate to this display as you would any other.

H.5. 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 computer and operating system. This section will walk you through the necessary setup to configure a project on a remote 32-bit computer.

1. Check that you have .NET Framework 4 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. If you are using a 64-bit machine, you can open Internet Explorer and navigate to **http://*servername*/GENESIS64/websites/PubDisplay/Workbench.htm** (where ***servername*** is the hostname of your server computer) to bring open the Workbench and begin configuration.

LG. Lab: System Health Monitor

LG

LG.1 Lab Overview

In this lab, we will demonstrate how to use the System Health Monitor, a tool used to monitor the overall system, utilize diagnostic tools and correct any existing situation.

Estimated time to complete this lab: **20 minutes**

LG.1.1 Task Checklist

In this lab, you will complete the following tasks:

- Configure Health Monitor Settings
- Generate Related Alarms
- Set Event Logging

LG.1.1 Support Files

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

LG.1 Configure Health Monitor Settings

The Health Monitor Settings node in the FrameWorX provider tree explorer in the Workbench allows users to configure the inputs and rules for System Health Monitoring.

To Configure Health Monitor Settings:

1. In the Workbench, expand the **Platform Services** node and then the **FrameWorX** node.

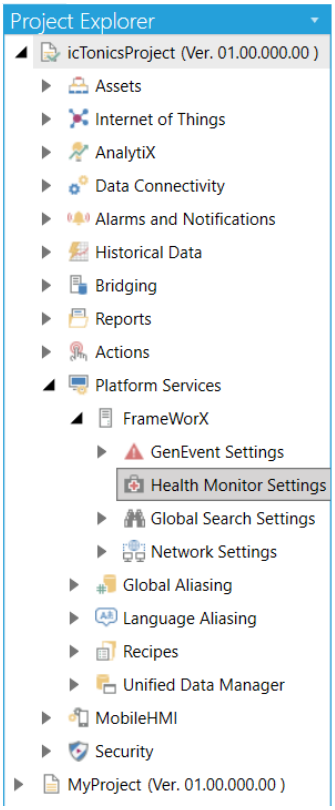


Figure L1B.1 - Selecting the Health Monitor Settings Node in the FrameWorX Provider in Workbench

2. Double-click the **Health Monitor Settings** node.

-OR-

Select **Health Monitor Settings** in the Workbench Project Explorer. Then click on the **Edit** button, shown below, in the Edit section of the Home ribbon.

Figure L1B.2 - Edit Button

This opens the Health Monitor Settings properties window in the lower half of the Project Details frame in the center of the Workbench, as shown in the following figure.

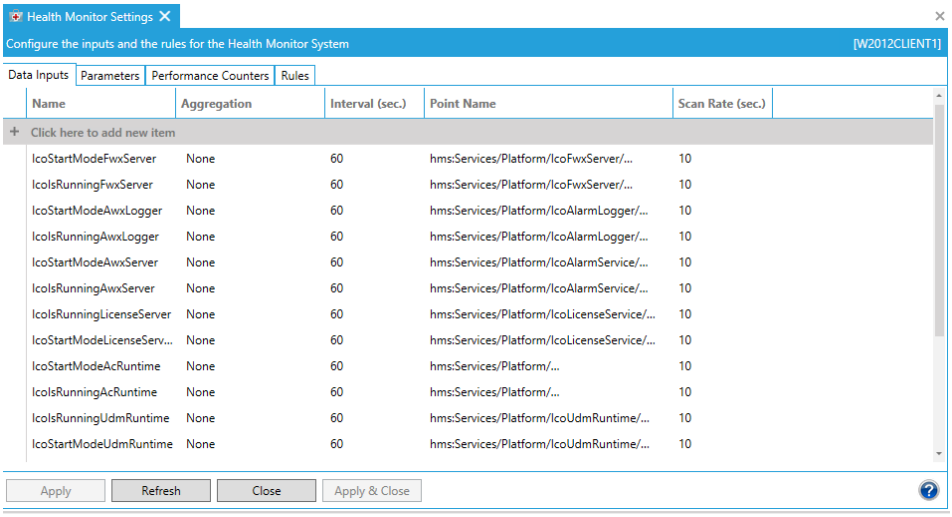


Figure L1B.3 - Health Monitor Settings - Data Inputs Tab

3. The Health Monitor Settings properties contains four tabs: **Data Inputs**, **Parameters**, **Performance Counters**, and **Rules**.

4. In the **Data Inputs** tab, click on "**+ Click here to add new item**". Enter **ModbusActiveTagCount** in the **Name** text entry field. Leave the default value, **None**, in the **Aggregation** field. Next, enter an **Interval** in the following text entry field. Click within the **Point Name** field. You can enter a point name directly in the text entry field, or click on the button to open the Data Browser and navigate to **Diagnostics** → **License** → **Category** → **Modbus_OPC** → **InUse**. The field should now show the tag **$Info:Category_System._ActiveTagCount.Value**. Finally, enter a **Scan Rate** like **60** seconds in the last text entry field. Select **Apply** to save your Data Input item.

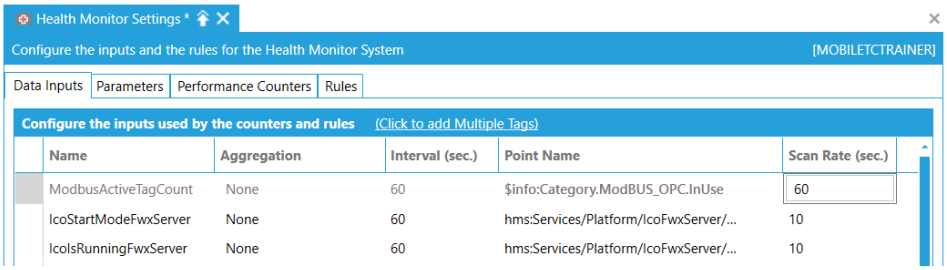


Figure L1B.4 – Health Monitor Settings – Data Inputs Tab

5. In the **Parameters** tab, you can enter a parameter **Name** and **Value**. Skip this item for now.

   a. In the **Performance Counters** tab, click on "**Click here to add new ~item**". Enter a name for the Performance Counter in the **Name** text

entry field. Click in the **Aggregation** field and you will be able to use the pulldown menu to select from either None, Minimum, Maximum or Time Average. Next, enter an **Interval** in the following text entry field. Click in the **Category** field and you will be able to use the pulldown menu to select from the list of categories. Depending on what you selected in the Category field, the Counter field will provide an additional pulldown menu for you to select from the list of counters. The last field is **Instance Name**, where you can enter this info into the text entry field or, depending on the Category/Counter, use the pulldown menu to select from the list of instance names.

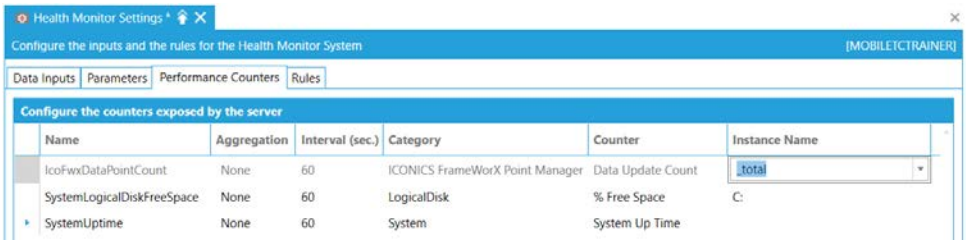


Figure L1B.5 – Health Monitor Settings – Performance Counters Tab

6. In the **Rules** tab, in the **General Settings** section, you can enter an **Update Rate** (in seconds) in the text entry field. Compare the Expression of several Rules. We are not going to configure a new Rule at this time.

   If you are interesting in creating a new Rule, you would click on "**Click here to add new item**". Enter a name for the Rule in the **Name** text entry field. The **Enabled** field will allow you to set, via checkbox, whether the rule is currently enabled (default setting is **Enabled**). In the **Expression** field, you can enter an expression directly into the text entry field or click on the button to open the **Configure the expression** window. Once you have set an expression, you would click OK to return to the Health Monitor Settings.

   In the **Alarm Condition** field, you can use the pulldown menu to select from: Is False, Is True, Is Greater Than Threshold, Is Greater or Equal Than Threshold, Is Less Than Threshold, or Is Less of Equal Than Threshold. You can then set a **Threshold** in the following text entry field. Next, you can set the **Severity** in the next text entry field.

   The **Log System Events** field allows you to set, via checkbox, whether to log the system events associated with the rule (this is set as enabled by default). The **Restart Service** field allows you to set, via checkbox, whether to restart a specific service (this is set as disabled by default). If the Restart

Service checkbox is checked, this activates the **Service Name** field, where you can select the service to be restarted from the pulldown menu. Also, depending on if the Restart Service checkbox is checked and a service has been selected from the Service Name field, you can decide whether to **Restart Immediately** (via checkmark) or enter a **Restart Time** in the text entry field or click on the button to open the time picker window. The buttons allow you to move the rule within the list of rules.

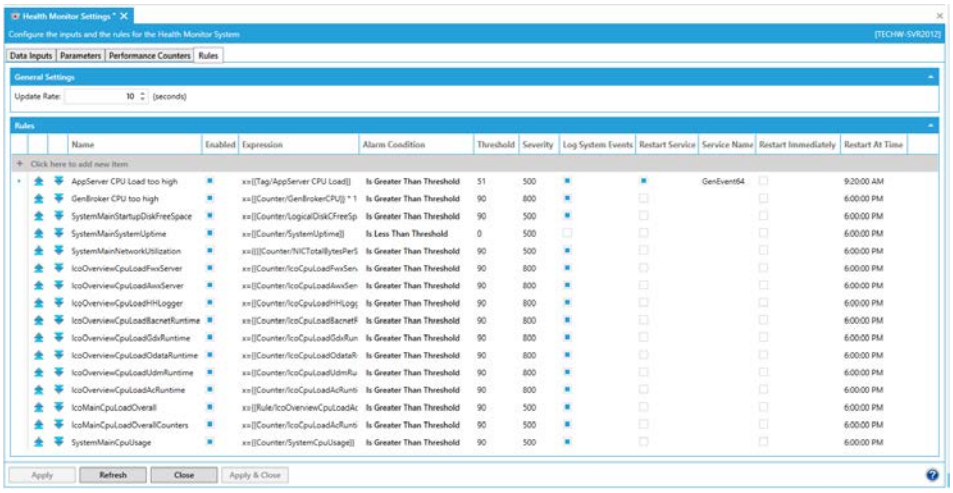


Figure L1B.6 - Health Monitor Settings - Rules Tab

7. Once you have made your changes to the Health Monitor Settings properties, click **Apply** to save your settings, then **Close** to close the window.

LG.1 Generating Alarms, Event Logging

Diagnostic Counters may be configured to generate alarms when a certain limit is reached. The alarm can be viewed with AlarmWorX64 Viewer. When an alarm is generated by a Diagnostic Counter, an event may be logged in Windows Event Log.

1. To enable this feature, use the **Platform Services Configuration – Diagnostics** tab. From the Workbench, click on **File** > **Platform Services**.

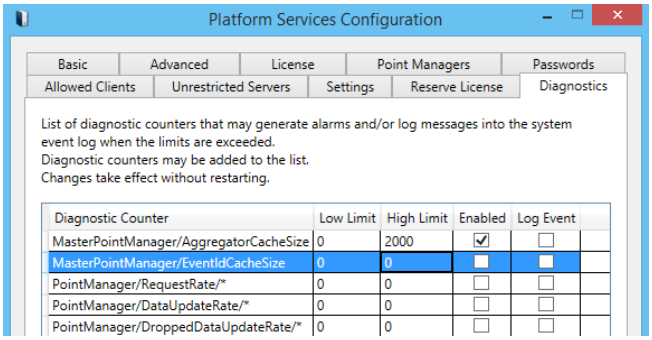


Figure L1B.7 – Platform Services Configuration – Diagnostics Tab

2. Users may add other Diagnostics Counters for generating alarms.

   For multi-instance Diagnostic Counters, you may use * to indicate the alarm generation is enabled for all instances. The alarm then contains the information for which instance generated it.

3. You can access **AlarmWorX64 Viewer** properties in **GraphWorX64** via the **Controls** ribbon. You can see more information on GraphWorX64 later in this manual. In AlarmWorX64 Viewer, you can subscribe to the Diagnostics Counters folder or any of its subfolders.

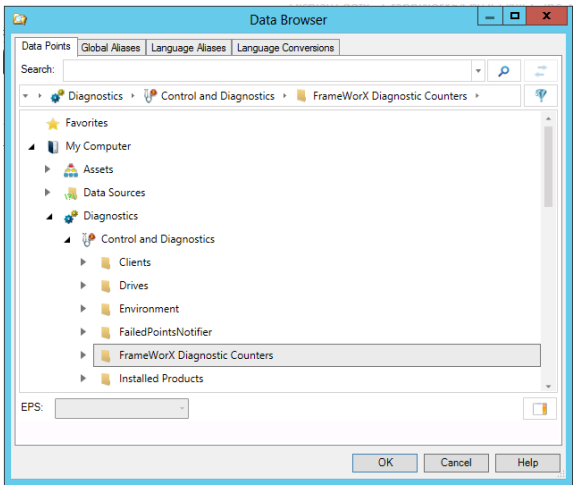


Figure L1B.8 – Subscribe to Diagnostics Counters Folder

4. The events may be logged in Windows Event Log:

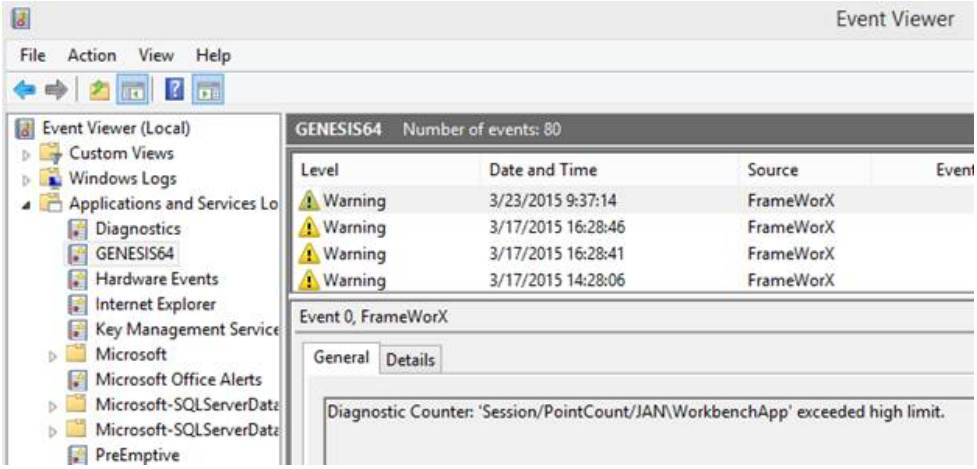


Figure L1B.9 – Logging in Windows Event Log

<u>This concludes this lab.</u>

Visualize Your Enterprise™

100 Foxborough Blvd., Foxborough, MA 02035
508.543.8600 • FAX: 508.543.1503
www.iconics.com • E-mail: info@iconics.com

© 2017 ICONICS, Inc. All Rights Reserved.
Specifications subject to change without notice.