



Make the Invisible Visible™

Getting Started with the Sample Project

Version 2
February 2024

Contents

1	Purpose	3
1.1	Scope.....	3
2	Architecture of the Sample Project.....	4
3	Setting up the Sample Project	7
3.1	Accessing the Graphics in Runtime	7
3.2	Using the Bulk Asset Configurator	7

Copyright and Confidentiality

By accessing and using the installation instructions (the “instructions”) you acknowledge and agree, on your behalf and on behalf of the person, entity or other organization on whose behalf you are accessing the instructions, that neither ICONICS nor any of its service providers, including, without limitation, any system integrator or independent software vendor: (1) makes any representations or warranties of any kind, either express, implied, statutory or otherwise with respect to the instructions, including the accuracy, completeness or usefulness thereof; and (2) shall be liable for damages of any kind, under any legal theory, arising out of or in connection with your election to follow or use, or inability to follow or use, the instructions, including any direct, indirect, incidental, special, punitive or consequential damages, or for loss of use, loss of profits, loss of data, loss of business, or loss of privacy or security, even if foreseeable, arising out of or in connection with your election to follow or use, or inability to follow or use, the instructions. You further acknowledge and agree that your use of the instructions, whether directly or indirectly, is at your own risk and that you expressly assume all risk in connection with your use of the instructions. If you do not agree to the foregoing, you may not access or use the instructions.

Copyright © 2024, ICONICS, Inc. All rights reserved.

Authors

- Jotham Kildea, Solution Sales Engineering Supervisor

1 Purpose

ICONICS has endeavored to make a collection of standard Workbench configurations and GraphWorX display files available to all interested users which we are referring to as the “Simple Sample Project”.

These files are intended to be a realistic starting point for your real-world projects if you wish, but at the very least be a solid reference for best practices in designing dashboards and user interfaces.

This Getting Started document will outline how to deploy and use the Sample Project files, as well as help you to understand the recommended order of configuration steps when building this or a similar project with ICONICS.

Supplemental to this document is a more extensive set of documentation for the specific configurations included in the Project, both for understanding how they are built as well as how they may be extended. We recommend referring to this once you have the project deployed as it will allow you to interrogate and compare the configurations against the documentation in real time.

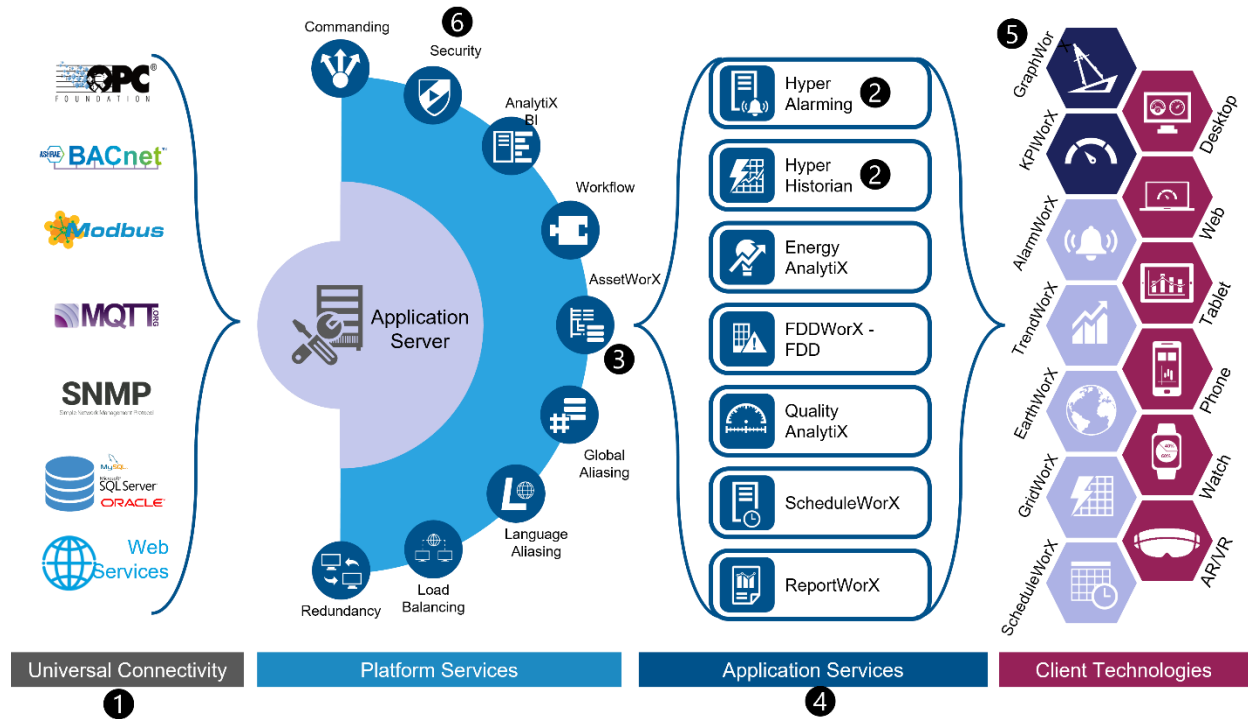
1.1 Scope

The Sample Project will cover some of the most commonly used modules of the ICONICS platform, and in a fashion that is well-suited to scalability of any future solution. Currently the Sample Project makes use of the following:

- GraphWorX64
- WebHMI (for HTML5)
- AssetWorX
- Hyper Alarm Server
- Hyper Alarm Logger
- Hyper Historian
- ICONICS Simulator

2 Architecture of the Sample Project

The Sample Project uses an architecture which is compatible with a wide range of projects. The way that data gets into the system is as follows:



If you were to be creating a new project from scratch, we would typically follow the following steps:

1. We leverage our various Data Connectivity protocols to subscribe to the data sources as appropriate. Each protocol supported by ICONICS will have some different settings that you may need to configure in order to enable subscribing to data. Certain projects may use just one data protocol, while others may use several.
2. Configuration of Hyper Alarm Server and Hyper Historian. As you see in the architecture, the configurations from the Hyper Alarm Server and Hyper Historian are important as the foundation for configuring alarms and data logging on Assets later. We don't need to worry about configuring alarmed or logged tags just yet, but it is still important to first set up some basics in the respective modules.

In the Hyper Alarm Server (Alarms and Notifications -> Hyper Alarm Server) you will want to make sure you have the appropriate alarm types defined that you will need later. It may be the case that the base types in the default configuration are sufficient for you in which case no changes are necessary.

In the Hyper Historian (Historical Data -> Hyper Historian) you will want to make sure that the logging groups and collector groups are configured to suit your needs. Again, we will configure the tags for logging within Assets, but those configurations rely upon general settings in Hyper Historian and so it is good to evaluate them first.

3. Once we have one or more of our data protocols configured, we are ready to move on to configuration of the assets. We use AssetWorX as a central configuration because as you can see from the architecture, it provides us with a number of benefits:
 - By routing all data subscriptions through Assets, it becomes much easier to update if we should have to change point subscriptions later. It also makes it more efficient for managing large numbers of point subscriptions simultaneously.
 - Assets can be used to “normalize” data to a consistent format. A single asset could conceivably have data supplied through multiple different protocols, but Assets make it consistent and easy to find the data when you need it.
 - It will act as a unified place for our configuration of alarms as well as data logging. This will also become easier for us to go and subscribe to this data in runtime
 - Assets can be used in runtime applications for navigation, making use of the commanding associated with them.
 - Assets are very scalable, especially if you have configured them first as Equipment Classes. These Equipment Classes can be paired with the Bulk Asset Configurator to allow for larger deployments to happen quickly.
4. Configure other Application Modules as appropriate. Not all Application Modules are appropriate or necessary for all applications, but as applicable you may configure these at this time. They are configured at this stage as typically they may rely upon the real-time data, historical data, or alarms from prior steps.
5. Configure your visualizations based upon the data connections configured earlier within the GraphWorX64 editor. This part of the process can be very open-ended, but the concept is that now that you have your data available and well-organized into your asset hierarchy, you can begin to configure and design the dashboard and graphics which you intend to use for the runtime interface. Doing this after the data model is already in place makes it easier to leverage power user features in the graphics designs such as aliasing, cloning, and asset-based navigation. You will also have a good sense of which additional application level modules you might want to incorporate into the user interface, as appropriate.
6. Configure your security model. At this point, you should have the majority of information available about what data you have, what assets you have, and what graphics you have. From this you should be able to be informed as to how to assign the permissions for the necessary users and groups in your application. You can configure the security model in the Workbench under Security, and can use either locally-defined users and groups or import them from Active Directory. Import of users may allow for easier management long term, and also can facilitate multi-factor authentication (MFA) and single-sign-on (SSO) as needed.

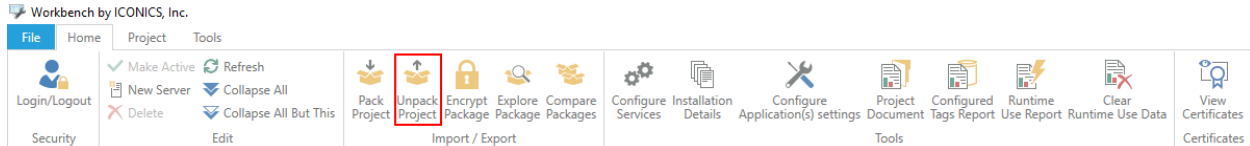
If you are using the Sample Project as a foundation for your project, the steps of configuration can be amended as follows:

1. Same as above. You will want to configure your data connectivity as needed for your project. The Sample Project cannot predict which connections you will use in all cases, but a reference example can be found in the Equipment Class configurations in Assets for how this data might be used
2. Same as above. The default configurations for Hyper Alarm Server and Hyper Historian which are included in the Sample Project may be sufficient as-is, or you may wish to amend them.
3. Same as above. You of course will want to configure Equipment Classes and Assets as are appropriate for your projects. It is recommended however to follow the same practices as used in the Sample Project when configuring those Assets.
4. Same as above. The Sample Project comes equipped with example configurations for Assets, Hyper Alarming, and Hyper Historian, but of course there may be others of interest to use for your project.
5. The Sample Project includes the basics for developing a functional dashboard out of the box, so these you can reuse as-is. You may additionally create new graphics as needed however if you have specific visuals to create, have assets with differing data sources, or wish to amend the graphics to leverage other Application Modules.
6. Same as above. The Sample Project does not include any security configurations out of the box.

3 Setting up the Sample Project

You will find that the Sample Project is provided as a .pkgx file. This is a compressed file format which is used by the ICONICS Pack and Go configuration in the Workbench as an compact means of migrating configurations.

Using the Pack and Go file is easy. In the Workbench, you will find under the Home Ribbon the option for “Unpack Project”.



Select this and follow the dialogs that appear, selecting the Sample Project file as the .pkgx to unpack. Once it completes, you should find that it has imported the relevant configurations for:

- Assets
- Hyper Alarm Server
- Hyper Historian
- Project Files

During the unpacking process, the Workbench will prompt you for where you should unpack the Project Files contents. These files were designed so as to be compatible with AnyGlass publishing to clients, and so we would recommend to select the following root directory to unpack into:

C:\Program Files\ICONICS\GENESIS64\WebSites\PubDisplay

This will create a folder called “SampleProject” that includes the GraphWorX64 displays and additional configuration files as needed.

3.1 Accessing the Graphics in Runtime

By default the Sample Project when unpacked will create a directory in your PubDisplay folder called “Sample Project”. In this folder there is a “Main.gdxf” file which acts as the primary entry point to access the project in runtime.

The Sample Project is designed for use with HTML5 graphics rendered in a web browser using AnyGlass. To access this, once you have the project unpacked you can use a web browser and navigate to: <http://localhost/AnyGlass/PubDisplay/SampleProject/Main.gdxf>

3.2 Using the Bulk Asset Configurator

The Sample Project includes a spreadsheet which is compatible with the Bulk Asset Configurator. While the Pack and Go project already includes the assets which are the result of executing this BAC file, it can be useful to use the BAC as the “master” copy for further configuration.