WebHMI
Training Manual

Version 9.20

© 2011 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.

GENESIS32, GENESIS64, Pocket GENESIS, BizViz and their respective modules, OPC-To-The-Core, and Visualize Your Enterprise are trademarks of ICONICS, Inc.

Windows, Windows 7, Windows Vista, Windows Server 2008, Windows Server 2003, Windows XP, Windows 2000, Windows NT, Windows XP-64, Windows 2003-64 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-WEB-920

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 archival back-up copies 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 which 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.

Contents

1. Introduction 1-1
1.1 Objectives 1-1
1.2 Overview of WebHMI Training 1-1
1.3 What Is OLE for Process Control? 1-2
1.4 What Is WebHMI? 1-4

Quick Lab 1: Course Project Overview L1-1

2. Web Server and WebHMI Installation 2-1
2.1 Objectives 2-1
2.2 Overview 2-1
2.3 Web Server Installation 2-1
3.3.1 Installing IIS in Windows Server 2003 2-2
3.3.2 Installing IIS in Windows XP 2-4
3.3.3 Installing IIS in Windows 2000 2-6
2.4 WebHMI Installation 2-8
2.4.1 System Requirement 2-8
2.4.2 Installation Steps 2-8

Quick Lab 2: Getting Familiar with WebHMI L2-1

3. GenBroker Communications Overview 3-1
3.1 Objectives 3-1
3.2 Overview of GenBroker Communications 3-1
3.2.1 GenBroker Configuration for OPC Server and Clients 3-2
3.2.2 GenBroker Configuration for WebHMI Clients 3-3
3.3 GenBroker Communication Options 3-3
3.3.1 Using OPC Direct 3-4
3.3.2 Using a Direct Channel Over DCOM 3-4
3.3.3 Using a Direct Channel Over TCP/IP 3-5
3.3.4 Using a Direct Channel Over SOAP/XML 3-5
3.3.5 Using an Indirect Channel Via a Mediator Node 3-5
3.4 GenBroker Confiuguration 3-6
3.4.1 WebHMI Client Configuration 3-6
3.4.2 Running the GenBroker Server 3-7

Quick Lab 3: GenBroker Configuration L3-1

4. Testing and Troubleshooting the Web Server..... 4-1
4.1 Objectives..... 4-1
4.2 Testing the Web Server 4-1
4.2.1 WebHMI Sample Files 4-1
4.2.2 Troubleshooting Common Problems 4-3

Quick Lab 4: Testing the Web Server L4-1

5. Creating Network – Aware Displays..... 5-1
5.1 Objectives..... 5-1
5.2 Making Data Items Network – Aware..... 5-1
5.3 Preparing Displays for Network Use..... 5-4
5.4 Making Existing Displays Network – Aware..... 5-4
5.4.1 Edit Connections Tool 5-4
5.4.2 Replace Tool..... 5-5

Quick Lab 5: Creating Network – Aware Displays L5-1

6. Publishing GENESIS32 Displays 6-1
6.1 Objectives..... 6-1
6.2 Publishing Overview 6-1
6.3 Publishing from GraphWorX32 6-2
6.3.1 General Settings..... 6-2
6.3.2 Advanced Settings 6-3
6.5 Publishing from ProjectWorX32..... 6-5

Quick Lab 6: Publishing GraphWorX32 Displays L6-1

7. Viewing the Project Remotely..... 7-1
7.1 Objectives..... 7-1
7.2 Viewing the Project..... 7-1
7.3 Troubleshooting Common Problems..... 7-2

Quick Lab 7: Viewing the Project Remotely.....L7-1

Practice Lab 8: Advanced Networking Example.....L8-1

Appendix A. Using the Change Host Name Utility A-1
A.1 Introduction to the Change Host Utility A-1
A.2 Changing the Directory Name A-1
A.3 Finding and Replacing Files A-2

Appendix B. Using the Versions Utility B-1
B.1 Introduction to Version Support and the Versions Utility B-1
B.2 Running the Versions Utility B-2

Appendix C. Securing Your Web Server C-1
C.1 Introduction to Web Server Security C-1
C.2 Level 1: Base Line Security C-1
C.2.1 Configuring Internet Information Services (IIS) C-2
C.2.2 Windows Files Security C-5
C.3 Level 2: Enhanced Security C-6
C.3.1 Setting up ICONICS Applications on the WebHMI Server C-7
C.3.2 Setting up Secure Socket Layer in IIS C-9
C.4 Level 3: Security for a Demilitarized Zone (DMZ) C-10

Appendix D. WebHMI Security and Licensing D-1
D.1 Using Security and Licensing with WebHMI D-1
D.2 WebHMI Security ActiveX D-1
D.3 Logging Into the Security Server D-3
D.4 Changing the Security Server Password via WebHMI D-4
D.5 Viewing the Logged User List D-4
D.6 Logging out of the Security Server D-5
D.7 Security OLE Automation D-5
D.8 Viewing Licensing Information Using WebHMI D-6
D.9 License for the WebHMI Server D-8
D.10 Obtaining Additional Connection Units D-8

Introduction 1

1.1 Objectives

In this chapter, you will be introduced to OPC architecture and WebHMI. After completing this chapter, you should:

- Understand what OPC is
- Understand what WebHMI is
- Be familiar with the project to completed in this course

1.2 Overview of WebHMI Training

Welcome to ICONICS' WebHMI Training. This training course will take you through the development process for creating Web-based applications using GENESIS32™. This training manual includes hands-on exercises that cover WebHMI™, which is an add-on component of GENESIS32.

1.2.1 Before You Begin

The material in this course assumes that you are familiar with one or more Microsoft® Windows® operating systems, including Windows 2000, XP, as well as Windows Server 2003®. You should be able to manipulate windows and launch and close applications. In addition, this course assumes that you have experience using standard Web browsers, such as Microsoft Internet Explorer®.

This course also assumes that you are familiar with or have had experience using the ICONICS GENESIS32 suite of software applications, including GraphWorX™32, AlarmWorX™32, and TrendWorX™32. It also assumes that you are familiar with or have been introduced to industrial automation and HMI (Human Machine Interface) and industrial devices such as PLCs, controllers, and I/O hardware.

1.2.2 Course Goals

After completing the WebHMI training course, you will have the basic skills you need to develop GENESIS32 applications using ICONICS WebHMI. This course will introduce you to the major features and functionality of WebHMI. You will

install and configure these applications, and explore the configuration interfaces by creating actual projects. Please see the *WebHMI Reference Guide* for complete information about using and configuring WebHMI.

1.2.3 Course Outline

This course is divided into the following chapters or labs:

Chapter 1: Introduction

Chapter 2: Web Server and WebHMI Installation

Chapter 3: GenBroker Communications Overview

Chapter 4: Testing and Troubleshooting the Web Server

Chapter 5: Creating Network – Aware Displays

Chapter 6: Publishing GENESIS32 Displays

Chapter 7: Viewing Displays from Remote Clients

Practice Lab 8: WebHMI with Advanced Networks

NOTE: This training manual also includes several helpful appendixes that contain additional information about WebHMI and Web Server setup.

1.3 What is OPC?

OPC™ is a standards-based approach for connecting data sources (e.g., PLCs, controllers, I/O devices, databases, etc.) with HMI client applications (graphics, trending, alarming, etc.). It enhances the interface between client and server applications by providing a universally supported and well-documented mechanism to communicate data from a data source to any client application. Included is not only a detailed guide on how to pass the data, but also specific information on other attributes to supplement those data, such as range information, data type, quality flags, and date and time information.

To understand better the difference between the old/traditional client-server architecture and the OPC based client-server architecture, consider the illustrations shown in **Figure 1.1** and **Figure 1.2.**

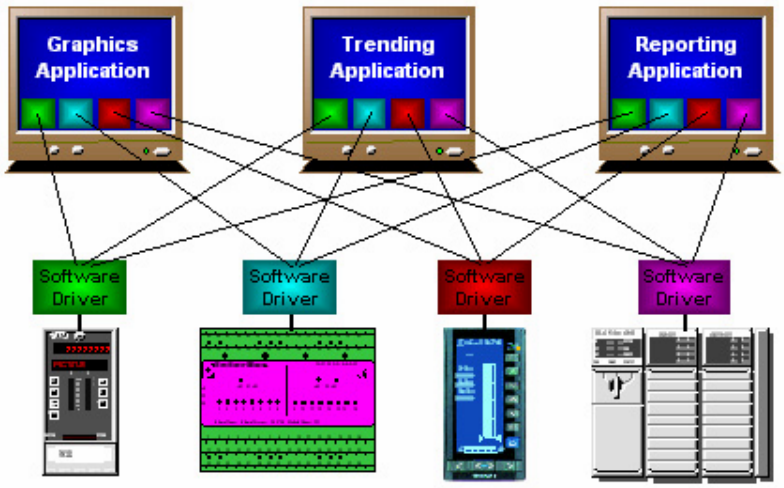

Figure 1.1: Old/Traditional Client-Server Architecture

In the old/traditional case, as shown in **Figure 1.1**, each client is required to write its own set of servers to the devices (each line above represents a software interface driver). This is due to the fact that although the protocol of a device or a server may not change, the architectures of the different client applications (from different vendors) are different. This leads to increased load on the machine, work in setting it up, costs for doing the project, and time needed to implement the solution. If the device or server's protocol is changed or updated, then each of the client applications must also change.

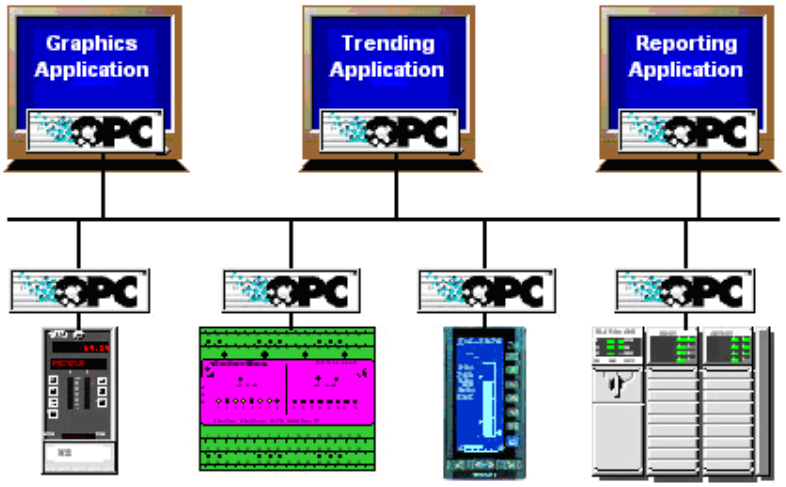

Figure 1.2: OPC-Based Client-Server Architecture

Figure 1.2 shows the OPC Architecture, which was introduced by the OPC Foundation mainly to overcome this issue. By following the OPC Architecture, a device needs only one standard driver, which is an OPC-compliant server. All OPC-compliant client applications can then be connected to that device, either locally or over a network. Furthermore, connections can be made to more than one OPC server at the same time.

Any OPC client application can connect to any OPC server. In other words, OPC offers true Plug-and-Play capability in the fields of HMI and industrial automation. OPC server types include OPC Data Access (DA), OPC Alarm and Events (AE), and OPC Historical Data Access.

1.4 What Is WebHMI?

ICONICS WebHMI is a thin-client Web solution that enables standard Web browsers, such as Microsoft Internet Explorer, to be used as real-time operator interfaces to manufacturing and factory floor applications. Based on ActiveX® technology, WebHMI provides you with a powerful and versatile approach to using the same standard HMI (Human Machine Interface) components included in GENESIS32 Enterprise Edition™.

WebHMI delivers industry-standard, real-time OPC information. WebHMI likewise delivers fast, worldwide operator graphical visualization, trending and alarming information—both real-time and historical—and HTML-based reports.

Since WebHMI Web components are packaged in standard Microsoft .cab files, your server and clients can be located anywhere. Similarly, you can store .cab files anywhere on your network. Installed and resident on one or more WebHMI servers, these components (e.g., GraphWorX32, TrendWorX32, or AlarmWorX32) are delivered automatically, quickly, and "in the background" to a browser on the client-side machine. Since WebHMI delivers the necessary components required for performing HMI and SCADA functions (e.g., building control, manufacturing, and process monitoring), it is not necessary to have any ICONICS products installed on the client machines.

Ultimately, WebHMI turns a Web browser into an OPC client when the browser views Web pages located on any WebHMI server. As mentioned above, Microsoft Internet Explorer is used to view web pages from the WebHMI server. Once a WebHMI site is up and running, you can instantly create multiple browser stations by having your clients launch the Web browser and visit the appropriate page.

Interaction between clients and the WebHMI server is made possible by ICONICS GenBroker™, which uses TCP/IP communication over the Internet. The GenBroker Configurator allows you to customize your client/server architecture based on your network configuration.

1.4.1 WebHMI Architecture

The architecture of WebHMI can best be explained in an illustration. **Figure 1.3** shows WebHMI, Security, Licensing, and OPC servers on the same node. The WebHMI clients are used to view Web pages downloaded from the WebHMI server

through a Web browser. GenBroker acts as a bridge that links WebHMI clients to the WebHMI server over the Internet using TCP/IP communications.

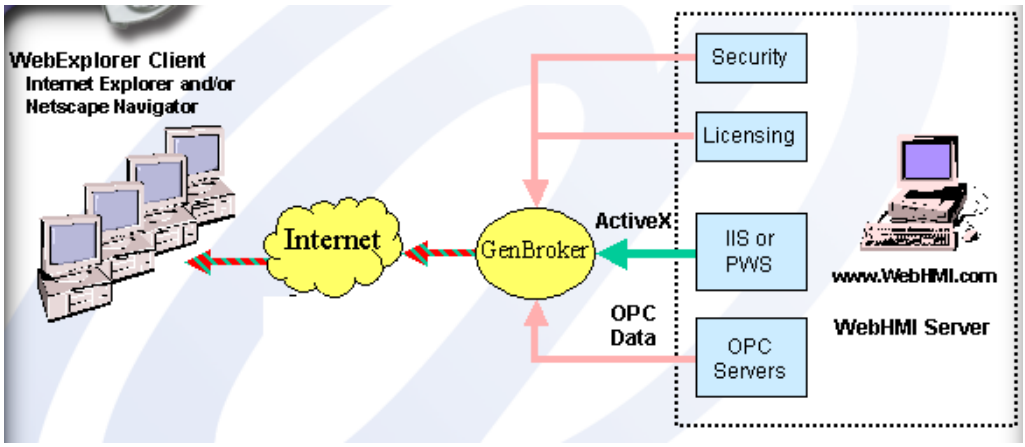

Figure 1.3: WebHMI Client/Server Architecture

Of course, you are not limited to this setup. Since GENESIS32 is modular, various components can be spread out over PCs with multiple networks. If you wish, you can have a single node devoted only to storing and supplying all necessary data files to be transported over HTTP; or you can have a setup in which any other node can be responsible only for supplying security information and serving ICONICS' licensing. You can have as many nodes as you wish supply OPC data.

Quick Lab: Course Project Overview 1

1.1 Quick Lab Overview

Throughout this manual you will be asked to complete several exercises or labs, in order to build your own WebHMI project. The labs will build upon each other and a central theme so that in the end you will have a complete Web enabled HMI project. This first lab will introduce you to the project you will build in these labs throughout the class.

1.1.1 Task Checklist

In this quick lab, you will complete the following tasks:

- Get an overview of the project to be built in this class

1.1.2 Support Files

To complete this quick lab, you will need the following support files:

- Configuration Files
  - TankLevelSimulation.mdb
  - AlarmServer.mdb
  - AWXLog32.mdb
  - DataLogger.mdb
  - TrainingCourse.sec
- GraphWorX32 Displays
  - Operator3.gdf
  - ValvePopup.gdf
  - LevelPopup.gdf
  - AlarmViewer.gdf
  - AlarmReporter.gdf

- HistTrendViewer.gdf

1.2 Basic WebHMI Example Overview

In order to expose your process data on a Web page, you need to perform a few steps in preparation first. First configure the **OPC Servers,** then create a set of GraphWorX32 **process displays** (*.gdf). Then configure **GenBroker** in order to determine how the Web browser can reach the remote OPC Servers over the intranet/Internet. Finally, publish the files to a Web server. Incremental troubleshooting throughout completion of these steps is always a good idea as well.

The example application in this class is a standard WebHMI application with a simple networking setup, assuming all GENESIS32 and OPC data resides on the same machine as the WebHMI server. This example can be used on an intranet or local area network (LAN) with the use of the node name, or over the World Wide Web with the use of the IP address.

We can break the process of this basic WebHMI application design down into the following steps, which (with the exception of Step 1) you will have accomplished by the end of this class:

1. Configure OPC servers, the ICONICS Security Server, and your GENESIS32 system.

2. Configure GenBroker Communications appropriately for the network the system resides on.

3. Test and troubleshoot the Web server.

4. Create network-aware displays for simple networking setups, or alter existing displays to prepare them for use over a network or the Internet.

5. Publish the process displays to the Web Server.

6. View the process displays from Web browsers, and troubleshoot the WebHMI application if necessary.

The WebHMI application to be deployed in this class assumes the simple network configuration shown in **Figure L1.1.**

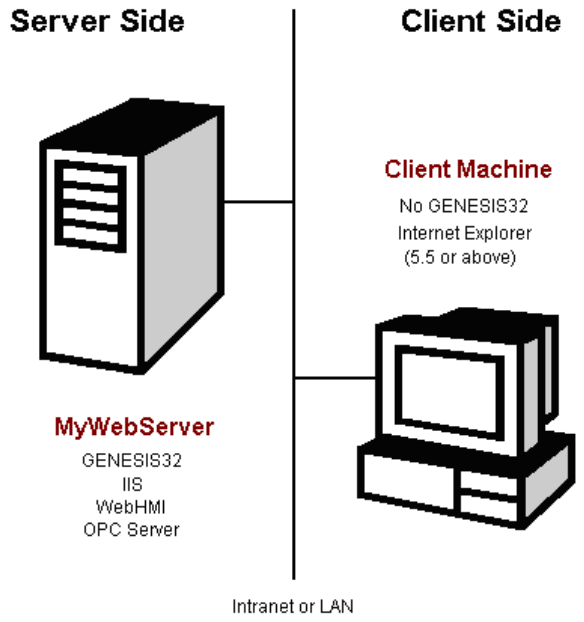

Figure L1.1: Network Setup for Basic WebHMI Application Example

All OPC data access (DA) servers, alarm and event (A&E) servers, and historical data access (HDA) servers reside and run on **MyWebServer**, which runs as the WebHMI server.

NOTE: The server name **MyWebServer** will be used throughout the labs in this course, and should be replaced with your actual server name as you complete the labs.

1.3 Acme Pharmaceutical Co. Overview

The Acme Pharmaceutical Co. is a global manufacturer for pain medicine, which has production facilities all over the world.

The Acme Headquarters is in Foxborough, Massachusetts in the United States. Other continents, however, have their own regional headquarters. There are several Acme factories located in the United States and Europe. Acme also has a factory in China, as well as one in Australia. These factories report directly to the US Headquarters.

Every Acme factory includes its own water treatment, pill processing, and bottling plants for which GENESIS32 and BizViz systems are in place to control, monitor, and report on their processes.

1.4 The Acme Water Treatment GENESIS32 System

This course will teach you to Web enable your GENESIS32 HMI project, using WebHMI. You will be provided with the existing HMI project in place in the Acme Water Treatment plants, which you will work with throughout this class. This section will familiarize you with this system.

As mentioned before, all OPC DA, A&E, and HDA servers will run on one machine for this standard example; the machine to be the WebHMI server.

⇒ **OPC Data Access Server**

DataWorX32 will act as the OPC Server for this project. The DataWorX32 configuration database **TankLevelSimulation.mdb** has logic built into it, similar to logic that would be programmed in a PLC controlling water tanks, valves, and pumps.

The configuration, shown in **Figure L1.2,** includes status and control tags for all pumps and valves in the system, as well as level monitoring tags for all tanks in the system.

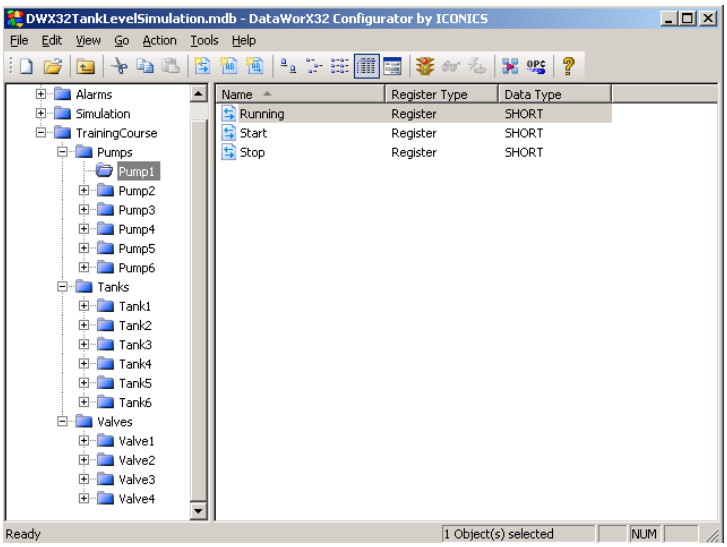

Figure L1.2: DataWorX32 OPC Server Configuration

The logic built into this system includes safety shut-offs on the pumps and valves. When a tank empties, the open valve emptying the tank will close. If a final holding tank fills, the pump supplying that final holding tank will turn off.

⇒ OPC Alarm and Event Server

AlarmWorX32 is configured to monitor the levels of the tanks. The AlarmWorX32 Server configuration, **AlarmServer.mdb** shown in **Figure L1.3,** includes a Limit alarm for each tank to indicate when the water levels are higher or lower than they should be.

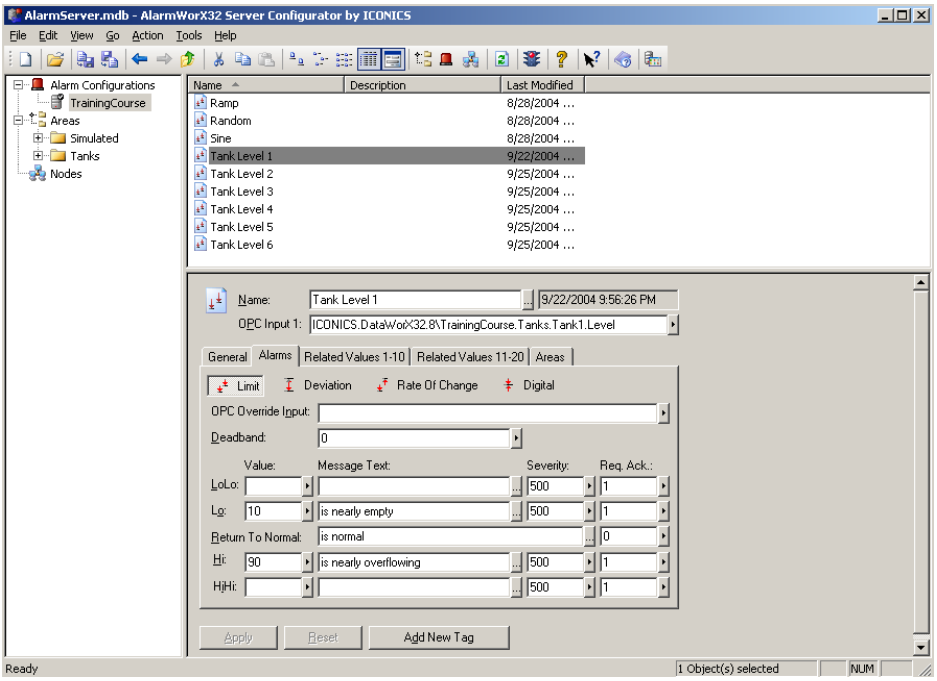

Figure L1.3: AlarmWorX32 Server Configuration

⇒ OPC Historical Data Access Logger/Server

The TrendWorX32 SQL Logger is configured for the ability to include historical trend replay in the system. The configuration, **DataLogger.mdb** shown in **Figure L1.4,** logs all tank levels to a SQL Server database. Samples are collected every 10 seconds.

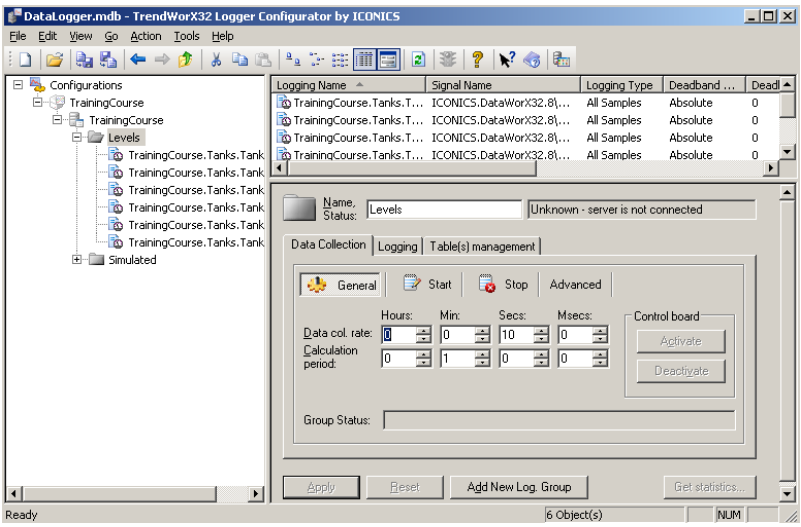

Figure L1.4: TrendWorX32 Logger Configuration

⇒ **Historical Alarm Logger**

The AlarmWorX32 Logger is configured so that alarm and event information can be logged to a database for historical viewing at a later time. The configuration database Acme has in place, **AWXLog32.mdb** shown in **Figure L1.5,** has two identical configurations enabled. One that logs the level alarms coming from AlarmWorX32 Server to a SQL Server database, and another that logs events posted by the GenEvent Server to a separate table in the same database.

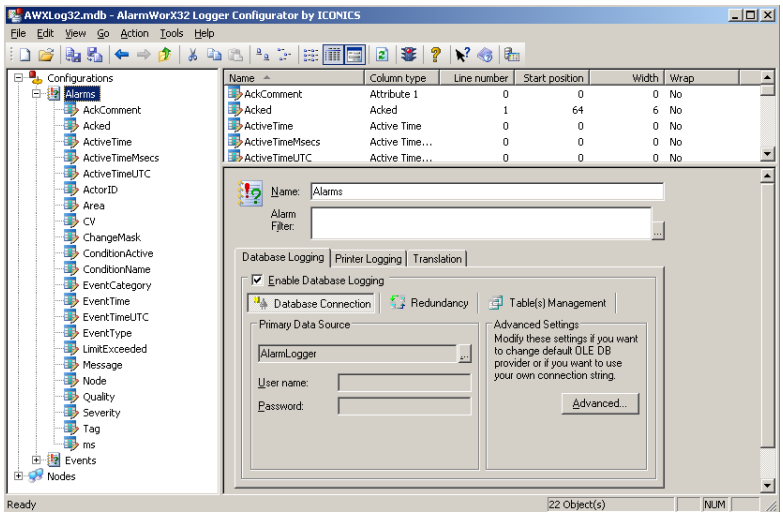

Figure L1.5: AlarmWorX32 Logger Configuration

⇒ **Security Server**

Visualize Your Enterprise™

The security server allows you to protect your process data from unauthorized access. The same security server used in GENESIS32 is also used for securing WebHMI browsers. Acme has a security system in place, shown in **Figure L1.6.**

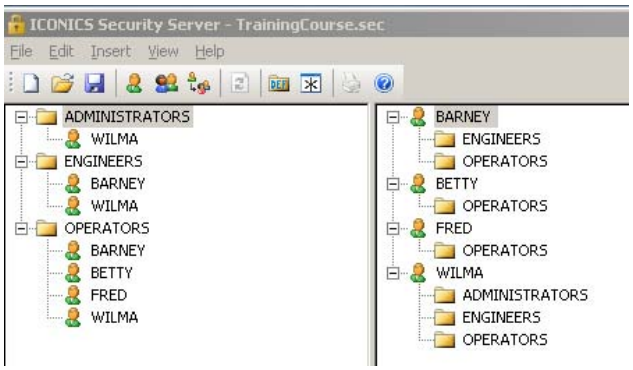

Figure L1.6: Security Server Configuration

This configuration contains three different levels of security for four different users. One Administrative user, Wilma, has administrative access to the GENESIS32 system in place.

⇒ **Graphical User Interface**

A set of GraphWorX32 displays are used by the operators to control the flow of water through the tank farm. Using the **Operator3.gdf** display, the operators can turn on and off the pumps, as well as open and close the valves to control the filling and emptying of tanks.

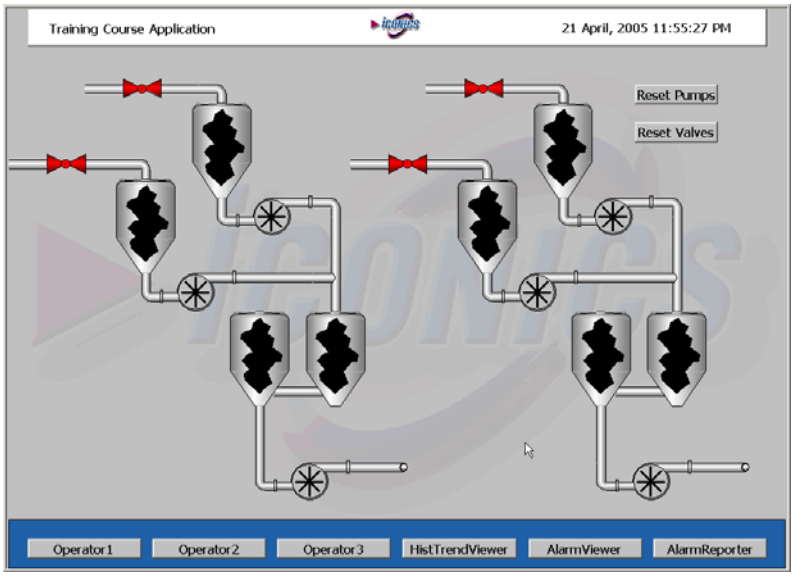

Figure L1.7: Operator3 GraphWorX32 Display

By clicking on each tank, a real-time alarming and historical data access monitoring window opens, with information specific to the tank that was clicked on, as shown in **Figure L1.8.** This tank specific information is made available by using Global Aliasing.

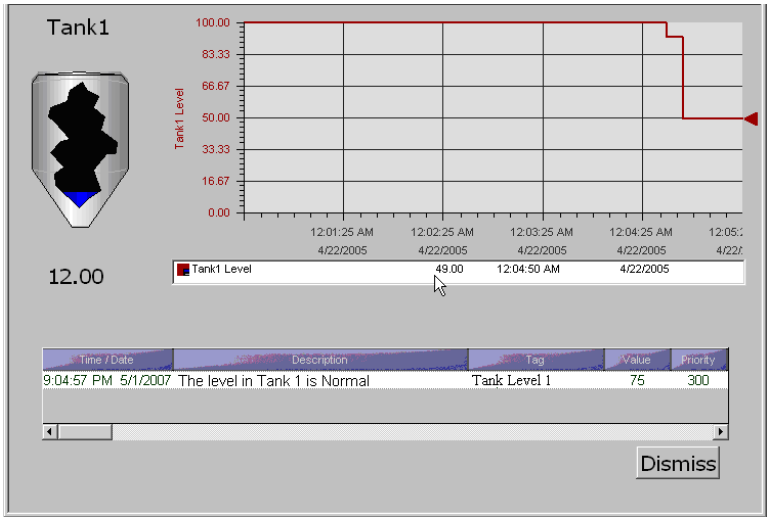

Figure L1.8: LevelPopup GraphWorX32 Display

Full system real-time alarm, historical alarm, and historical trend monitors are all part of the system as well, as shown in **Figure L1.9, Figure L1.10,** and **Figure L1.11** respectively.

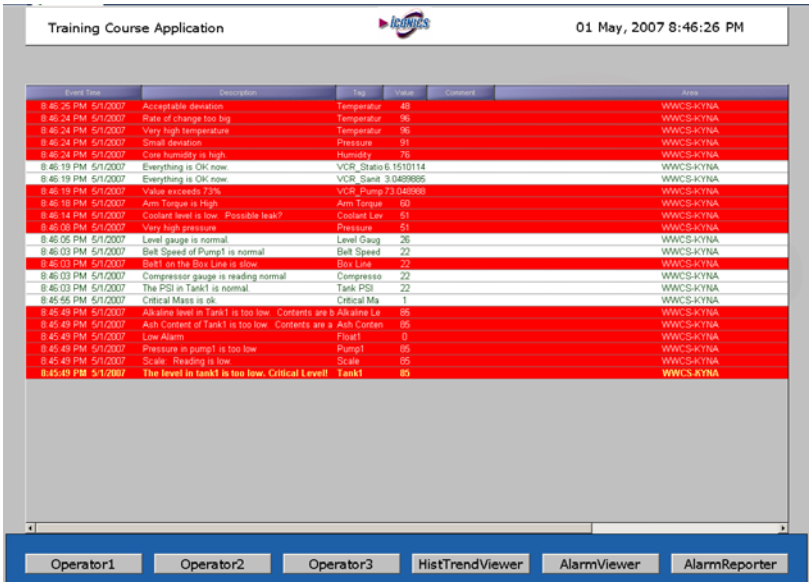

Figure L1.9: Alarm Viewer GraphWorX32 Display

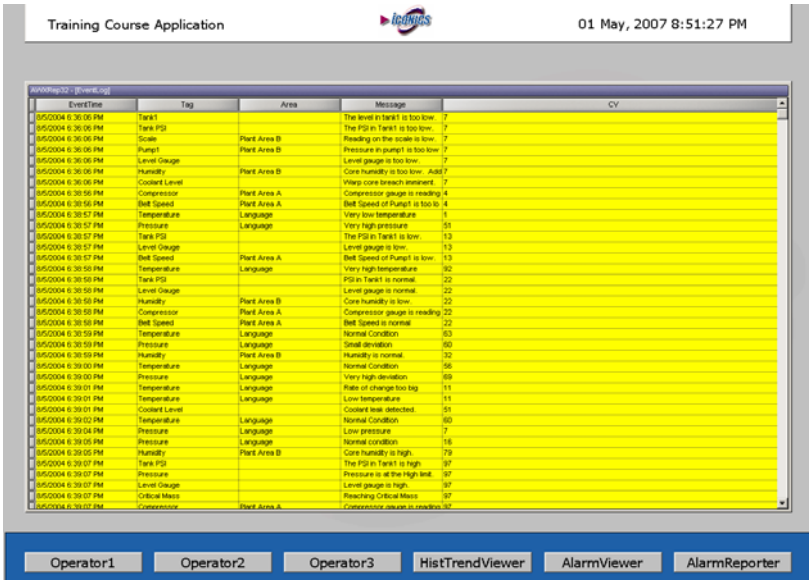

Figure L1.10: AlarmReporter GraphWorX32 Display

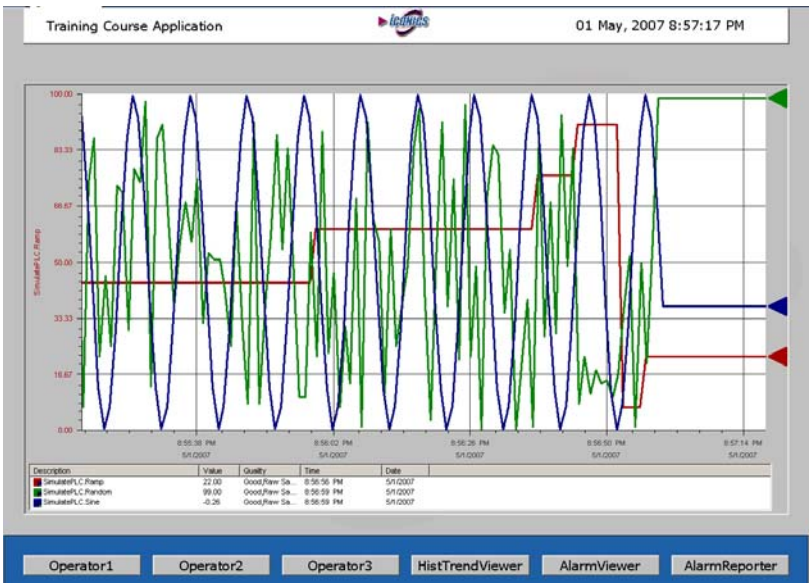

Figure L1.11: HistTrendViewer GraphWorX32 Display

1.5 Project Guidelines

Acme has hired you to web enable their existing GENESIS32 project using WebHMI. The following are the project guidelines they have provided you:

⇒ **Phase 1**

1. Prepare the existing system for communications via their plant LAN, by making sure that all OPC calls include the server node name or IP address.

2. Test and troubleshoot their Web Server to be sure that it is set up properly for GenBroker communications to send OPC and GENESIS32 specific information to remote machines using TCP/IP protocol.

3. Publish the GraphWorX32 display system to their Web Server.

4. Test and troubleshoot the system to be sure that all OPC DA, A&E, HDA, and Historical A&E data is received on the remote clients properly.

⇒ **Phase 2**

1. Increase security on their WebHMI system by separating the OPC DA Server to a separate server machine, and placing both the WebHMI and OPC DA Server machines behind a firewall.

2. Adjusting the GraphWorX32 and WebHMI configurations appropriately to allow communications through the firewall, as well as to and from the remote OPC DA Server machine.

Web Server and WebHMI Installation 2

2.1 Objectives

In this chapter, you will learn how to set up a Web server on different platforms. After completing this chapter, you should know how to:

- Install a Web Server:
  - Install Internet Information Server (IIS) in Windows Server 2003
  - Install IIS in Windows 2000
  - Install IIS in Windows XP
- Install WebHMI

2.2 Overview

Before installing WebHMI, you must first install some sort of Web server, such as Internet Information Server (IIS) or a third party Web Server such as Apache HTTP Server. These Web servers will control the security and permissions of your Web server, and the data available on it.

Once the Web Server is successfully installed, you can install WebHMI. The WebHMI installation will configure the installed Web Server appropriately for use with WebHMI.

2.3 Web Server Installation

WebHMI supports Windows Web Servers, such as IIS as well as Apache Web Servers. It is important to note, however, that if WebHMI is installed on an Apache Web Server, SOAP/XML communications are not supported.

This chapter will not go into detail about installing an Apache Web Server, but will go through the installation of IIS.

2.3.1 Installing IIS in Windows Server 2003

If your operating system is Windows XP, use the following instructions to install Internet Information Server (IIS) on the system:

1. Open your Windows Control Panel from **Start → Settings → Control Panel.**

2. In the **Control Panel,** open the **Add/Remove Programs** dialog, shown in **Figure 2.1.** Click on the **Add/Remove Windows Components** icon in the left-hand column of the window.

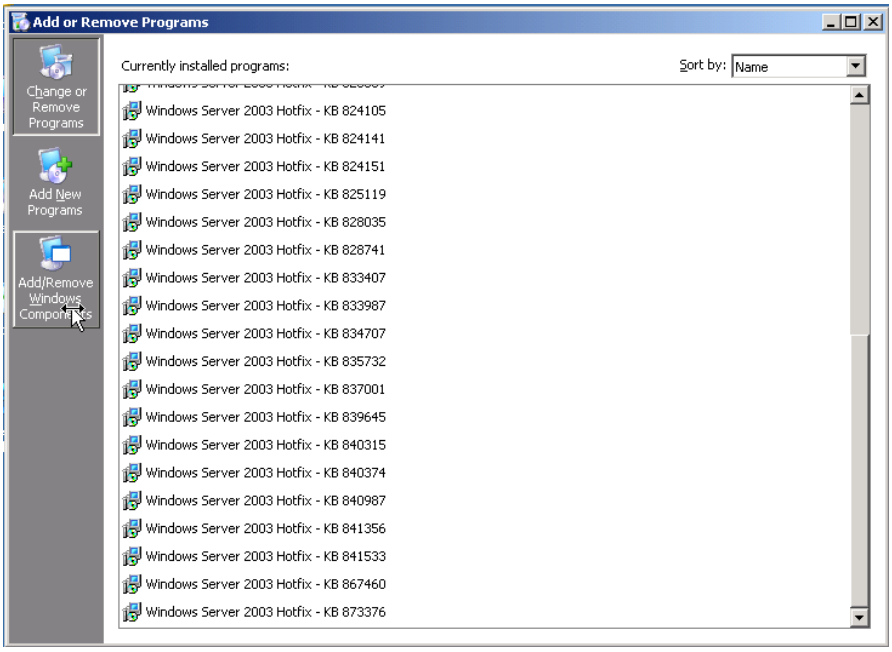

Figure 2.1: Microsoft Windows XP Add or Remove Programs

1. The **Windows Components Wizard** dialog box will appear, as shown in **Figure 2.5.** Select **Application Server** and click **Details** to view the contents of that component.

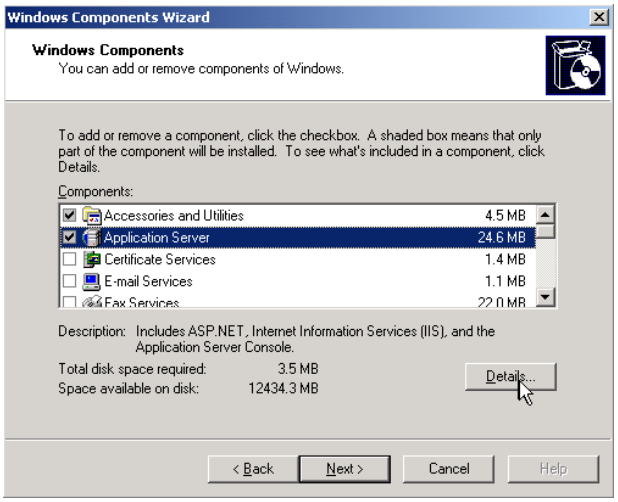

Figure 2.2: Windows Components Wizard

2. To add IIS to the system, select **Internet Information Services (IIS)** listed in the **Application Server** dialog, and enable the check box. If the box is already checked, IIS is already installed on the system.

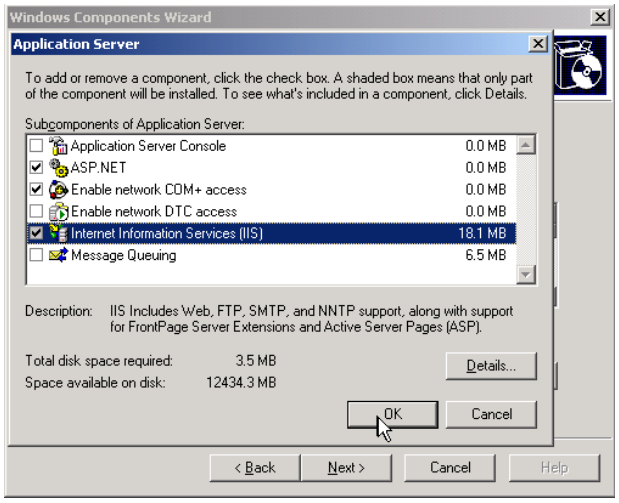

Figure 2.3: Application Server Details

3. With **Internet Information Services (IIS)** selected, click **OK** to return to the **Windows Components Wizard** dialog. Click the **Next** button to begin installation and configuration.

4. You will be informed that additional files are necessary, and prompted to insert your Windows Server 2003 installation CD, as shown in **Figure 2.4.**

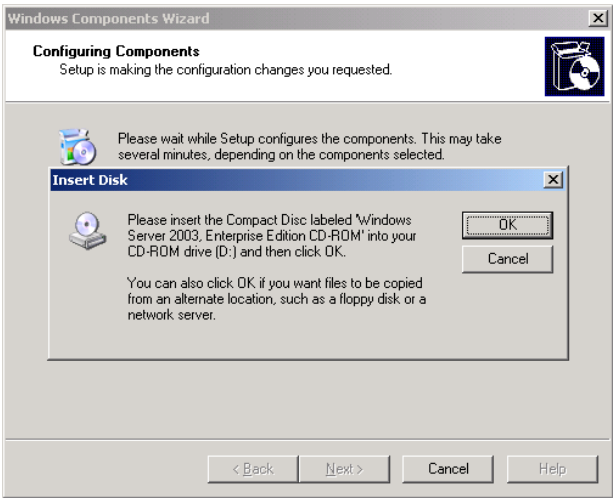

Figure 2.4: Files Needed

5. Do not click OK, and insert the installation CD as prompted. If the CD has auto run capability the dialog box will disappear and installation will continue automatically.

6. The **Windows Components Wizard** dialog box will reappear. The progress bar will track the file copying.

7. After all of the necessary components have been installed, the **Finish** dialog box will appear. To close the **Windows Components Wizard** dialog box, click the **Finish** button.

You have successfully configured Internet Information Services software for Windows XP

2.3.2 Installing IIS in Windows XP

If your operating system is Windows XP, use the following instructions to install Internet Information Server (IIS) on the system:

8. Open your Windows Control Panel from **Start → Settings → Control Panel.**

9. In the **Control Panel,** open the **Add/Remove Programs** dialog, shown in **Figure 2.5.** Click on the **Add/Remove Windows Components** icon in the left-hand column of the window.

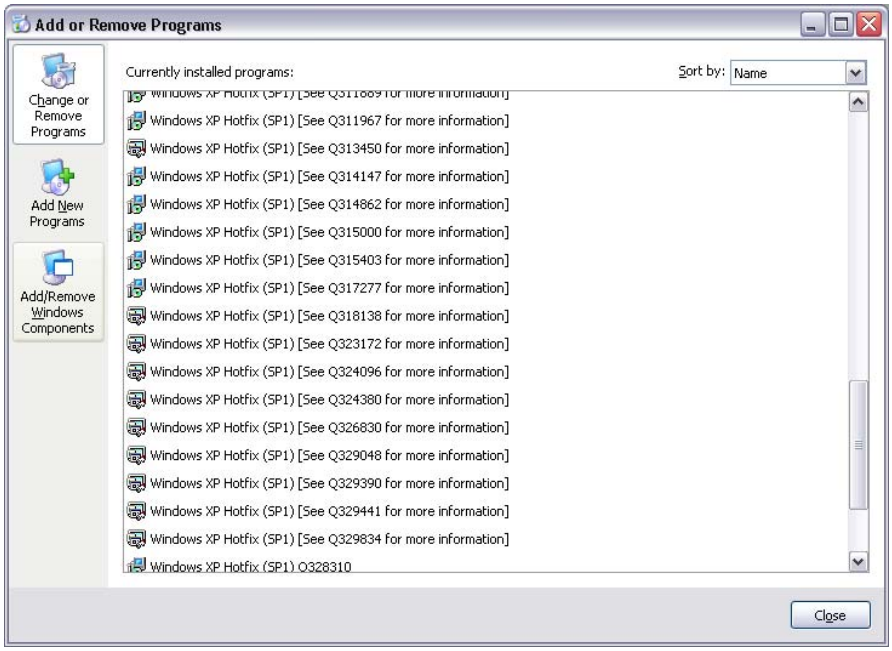

Figure 2.5: Microsoft Windows XP Add or Remove Programs

10. The **Windows Components Wizard** dialog box will appear, as shown in **Figure 2.6.** To add IIS to the system, select **Internet Information Services (IIS)** and enable the check box. If the box is already checked, IIS is already installed on the system.

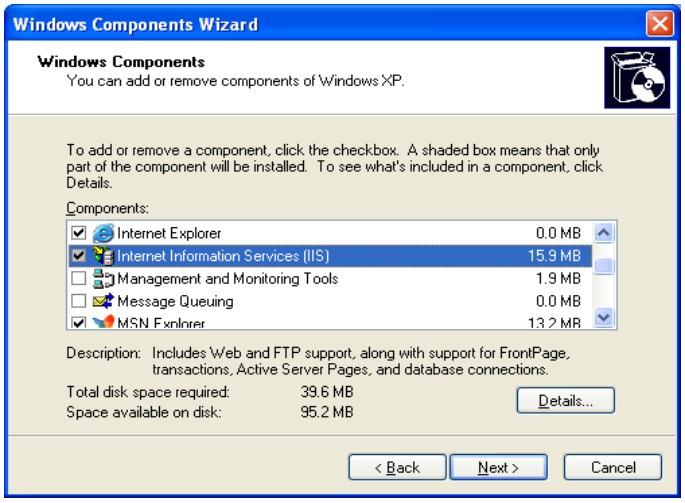

Figure 2.6: Windows Components Wizard

11. With **Internet Information Services (IIS)** seleced, click the **Next** button to begin installation and configuration.

12. You will be informed that additional files are necessary, and prompted to insert your Windows XP installation CD, as shown in **Figure 2.7.**

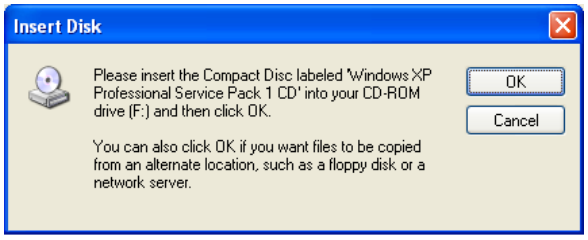

Figure 2.7: Files Needed

13. Do not click OK, and insert the installation CD as prompted. If the CD has auto run capability the dialog box will disappear and installation will continue automatically.

14. The **Windows Components Wizard** dialog box will reappear. The progress bar will track the file copying.

15. After all of the necessary components have been installed, the **Finish** dialog box will appear. To close the **Windows Components Wizard** dialog box, click the **Finish** button.

You have successfully configured Internet Information Services software for Windows XP

2.3.3 Installing IIS in Windows 2000

If you have Windows 2000 with Internet Information Server (IIS), you must configure it on your computer.

1. To configure IIS in Windows 2000, select **Start → Settings → Control Panel.**

2. In the **Control Panel,** click on **Add/Remove Programs.** The **Add/Remove Programs** dialog box will appear, as shown in **Figure 2.8.** Click on the **Add/Remove Windows Components** icon in the left-hand column of the dialog box, as shown in **Figure 2.8.**

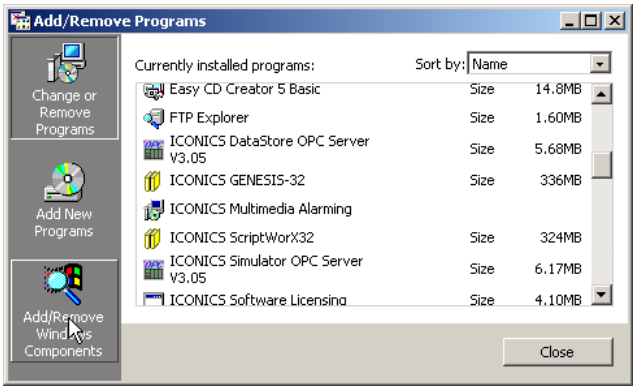

Figure 2.8: Add/Remove Programs Dialog Box in Windows 2000

3. The **Windows Components Wizard** dialog box will appear, as shown in **Figure 2.9.** To add IIS, select **Internet Information Services (IIS)** and click on the check box. After you have clicked on the **IIS** check box, the **Windows Components Wizard** dialog box will change its appearance and begin configuring components. Shortly after the configuring begins, you may be notified that additional files are needed.

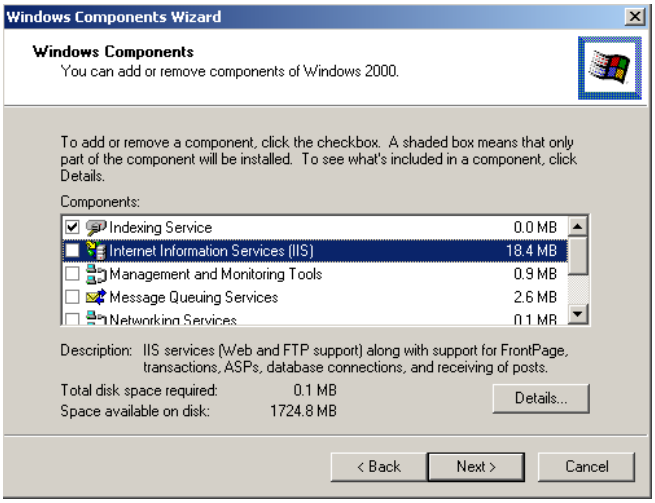

Figure 2.9: Windows Components Wizard Dialog Box

4. If additional files are necessary, the **Files Needed** dialog box will appear, as shown in **Figure 2.10.** In the **Copy files from** field, type in the drive on which your computer runs CD-ROMs, plus ":\i386"; or click the **Browse** button to locate the **i386** folder on your CD-ROM drive. After locating the **i386** folder, click on it to have it appear in the **Copy files from** field. Insert the Windows 2000 Professional CD-ROM and click on the **OK** button on the **Files Needed** dialog box.

5. The **Windows Components Wizard** dialog box will reappear. The copying of files is tracked on the progress bar of the window.

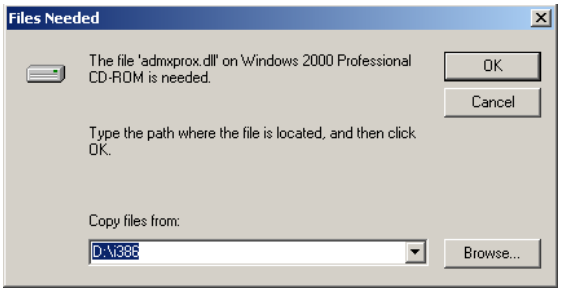

Figure 2.10: Files Needed Dialog Box

6. After all of the necessary components have been installed, the **Finish** dialog box will appear. To close the **Windows Components Wizard** dialog box, click the **Finish** button.

You have successfully configured Internet Information Services software for Windows 2000.

2.4 WebHMI Installation

This section of the document will describe the steps necessary to install WebHMI. It is assumed that all system requirements are established before completing these steps, including GENESIS32 and Web Server installation.

2.4.1 System Requirements

To use the WebHMI software, you must have the following *minimum* system requirements:

- 8X speed CD-ROM
- VGA video card, 256 or more colors for best results.
- Microsoft Internet Explorer 5.5 and above.
- Microsoft Data Access Components (MDAC) 3.0
- Microsoft Windows Server 2003 Service Pack 2, Microsoft Windows 2000 with Service Pack 3, or XP with Service Pack 2**.**
- A Web Server installed, such as Internet Information Services
- GENESIS32 installed (same version as WebHMI being installed)

2.4.2 Installation Steps

Follow the steps below to install WebHMI. Be sure to close any other applications before installing WebHMI.

7. WebHMI is installed from the ICONICS Options CD. Select WebHMI from the software installation menu, as shown in **Figure 2.11.**

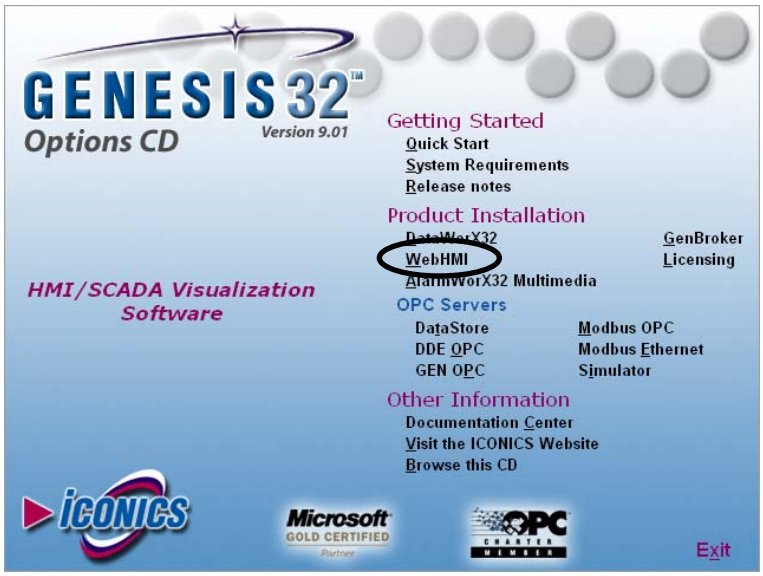

Figure 2.11: Software Installation Menu

8. The **Welcome** dialog box will appear. Click **Next** to continue.

9. The **ICONICS WebHMI License Agreement** dialog box will appear. Carefully read the ICONICS License Agreement. If you agree with the terms and conditions, click **Yes** to continue. Otherwise, click on No to abort the installation.

10. The **ICONICS WebHMI Copyright** dialog box will appear. After reviewing the copyright information, click **Next** to continue.

11. The **User Information** dialog box will appear, as shown in **Figure 2.12.** Type your (the user's) name in the **Name** field; your company's name in the **Company** field. Click the **Next** button to continue.

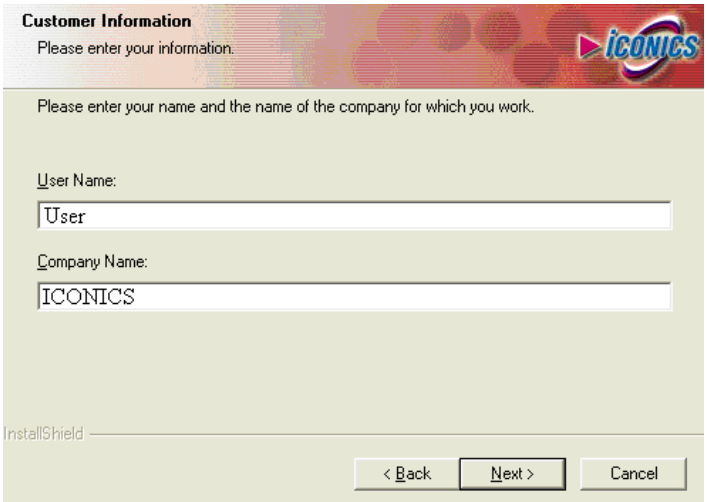

Figure 2.12: Entering User Information

12. The **Setup Type** dialog box will appear, as shown in **Figure 2.13.** Choose the **Typical** installation.

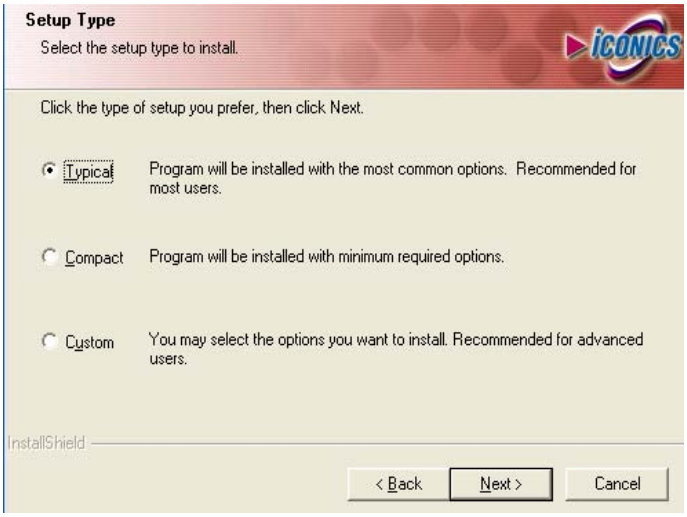

Figure 2.13: Choosing the Installation Type

13. The **Choose Communication Type** dialog box will appear, as shown in **Figure 2.13.** Choose **OPC over TCP/IP, OPC over SOAP/XML,** or **OPC over DCOM** for installation. Since the default communication type is **OPC over TCP/IP,** it is recommended that you use it to install WebHMI server. If desired, you will then be able to change the settings at any time. **OPC over TCP/IP** is easy to use on a local area network (LAN), over an intranet or the Internet.

- **OPC over DCOM** is recommended for use on a local area network where you know that all nodes have been properly set up. It is also recommended for use if you wish to avoid connecting over the Internet because you are uncertain of how all nodes in a network have been set up.

NOTE: OPC over DCOM works only in an environment where Microsoft Network has been installed.

- **OPC over SOAP/XML** SOAP uses XML to communicate over HTTP. The main advantage of using SOAP is that, because SOAP operates on the HTTP level (over TCP/IP), data can pass through a firewall without having to change the firewall settings. The disadvantage of SOAP is that it is slightly slower than TCP/IP. **SOAP** uses **port 80** by default.

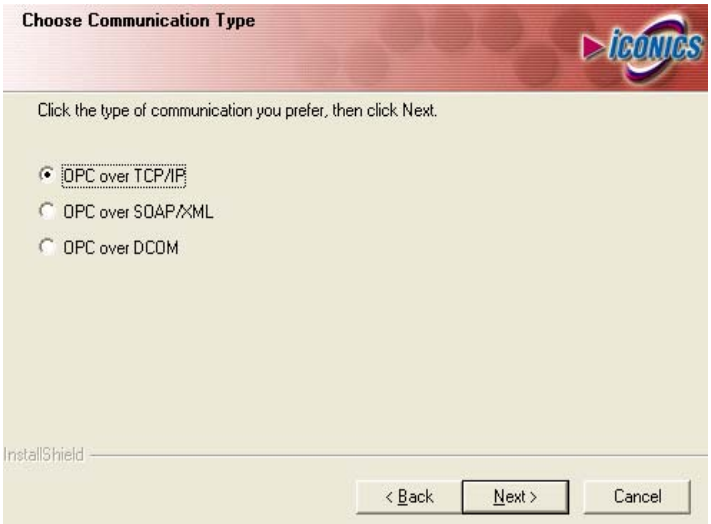

Figure 2.13: Choosing the Communication Type

14. The **Copying System Files** window will appear. While the system files are being automatically copied, the percentage of the files already copied will appear in the window's progress bar.

15. After all of the necessary system files have been copied, a dialog box will appear, as shown in **Figure 2.14.** Enter the name of your WebHMI server or select a local machine name. Customize your application by selecting one of three choices under **Name Convention:**

- **MS Network.** This is the name of the WebHMI server machine located on your local network. This is recommended with DHCP.
- **DNS (Domain Network Service).** This is your registered domain name, such as www.myserver.com.
- **IP Address.** The numeric IP address will automatically appear in the **DNS Name** field.

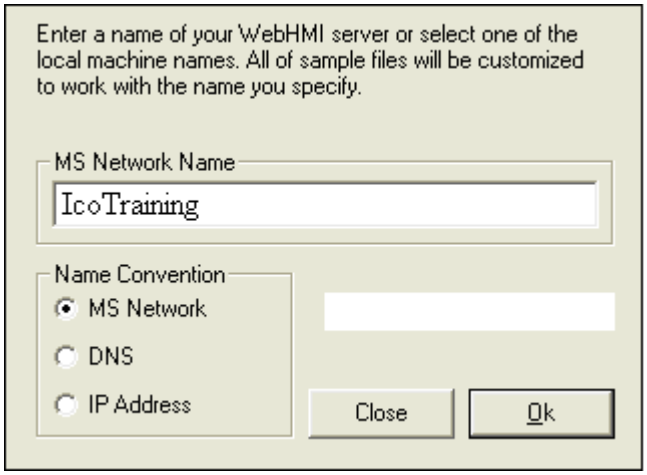

Figure 2.15: Specifying the WebHMI Server Host Name

16. After you have customized your application, click the **OK** button. The **Setup Complete** dialog box will appear. Click the **Finish** button to complete the setup.

You have successfully completed installation of WebHMI on your computer.

Quick Lab: Getting Familiar with WebHMI 2

2.1 Quick Lab Overview

This lab will familiarize you a little with your Web Server, Internet Information Services (IIS). Understanding the role of IIS in your WebHMI system will provide you with more troubleshooting skills and capabilities when deploying the system. You will also become familiar with the WebHMI installation and where files and applications can be found. This lab is purely exploratory.

2.1.1 Task Checklist

In this quick lab, you will complete the following tasks:

- Learn how to launch the Internet Information Services Console
- Understand how the Web server makes WebHMI files available across the Internet
- Explore the files installed by WebHMI

2.1.2 Support Files

To complete this quick lab, you will not need any support files.

2.2 The Internet Information Services Console

The following steps will help you launch the Internet Information Services (IIS) console, and explore the settings necessary and automatically put in place for WebHMI. IIS controls all directories and files on your Web Server, and how they can be accessed from remote clients. For this reason, the WebHMI directory which holds all files necessary for the WebHMI client to download, run, and view needs to be defined inside of IIS. This section will not go in depth about IIS configurations and security, and no changes in IIS are necessary.

1. Open the IIS Console from the **Administrative Tools** folder of your Windows **Control Panel.** The console will look similar to **Figure L2.1.**

NOTE: **MYWEBSERVER** will be replaced by the actual node name of your machine.

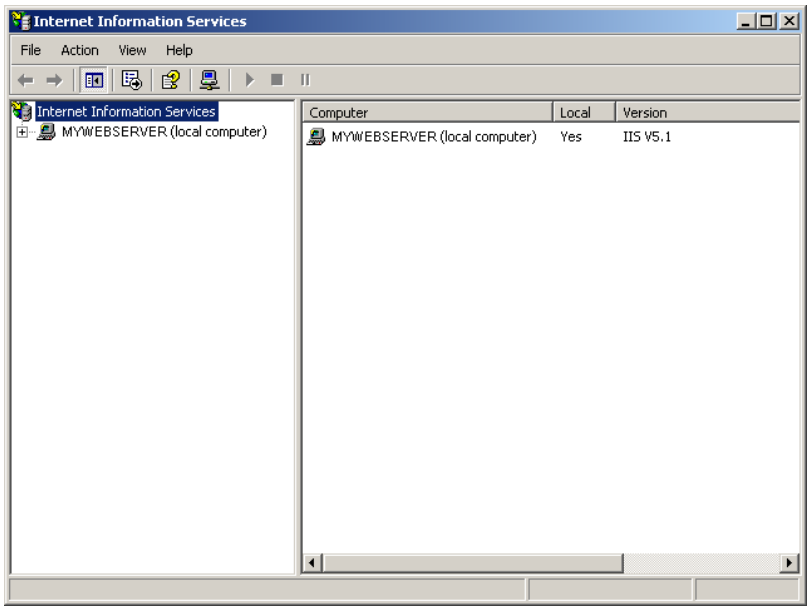

Figure L2.1: Internet Information Services Console

2. In the tree-control on the left-hand side, browse into the local computer into **Web Sites → Default Web Sites,** Note the existence of a **WebHMI** folder listed there, as shown in **Figure L2.2.**

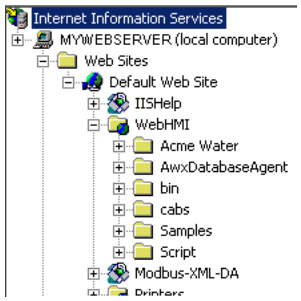

Figure L2.2: WebHMI Virtual Directory

3. The WebHMI folder is listed under the Default Web Site, so that the files inside that folder are available to Web clients for download and use. However, this folder is known as a **virtual directory,** because it doesn't actually exist on the in the Web Server root directory itself (which by default is C:\Inetpub\wwwroot), but elsewhere on the machine.

4. The icon indicates that it is a virtual directory, where just a regular folder (without a globe) indicates that the folder actually exists in the **C:\Inetpub\wwwroot** directory. The actual directory where the WebHMI folder

is defined in the properties of this virtual directory for redirection by the Web Server. All the folders that exist inside the real WebHMI folder are shown in IIS as well, as indicated in **Figure L2.2.**

5. To see the real directory path of the WebHMI folder, right-click on the **WebHMI** virtual directory in the IIS console, and choose **Properties.** As shown in **Figure L2.3,** the **Local Path** of the directory is **C:\Program Files\Common Files\ICONICS\WebHMI.**

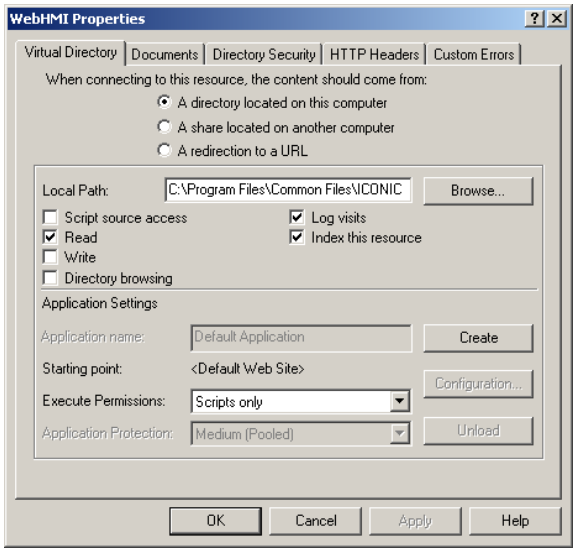

Figure L2.3: WebHMI Virtual Directory Properties

6. Cancel out of the **Properties** window, and close the IIS console.

2.3 The WebHMI Installation

Now that we know that the WebHMI directory is defined in IIS for file distribution to the all Web clients, let us understand what is in that directory and what it is used for.

1. Using a Windows Explorer, browse to the **C:\Program Files\Common Files\ICONICS\WebHMI** directory on your machine. You will see a number of folders and a few files, as shown in **Figure L2.4.**

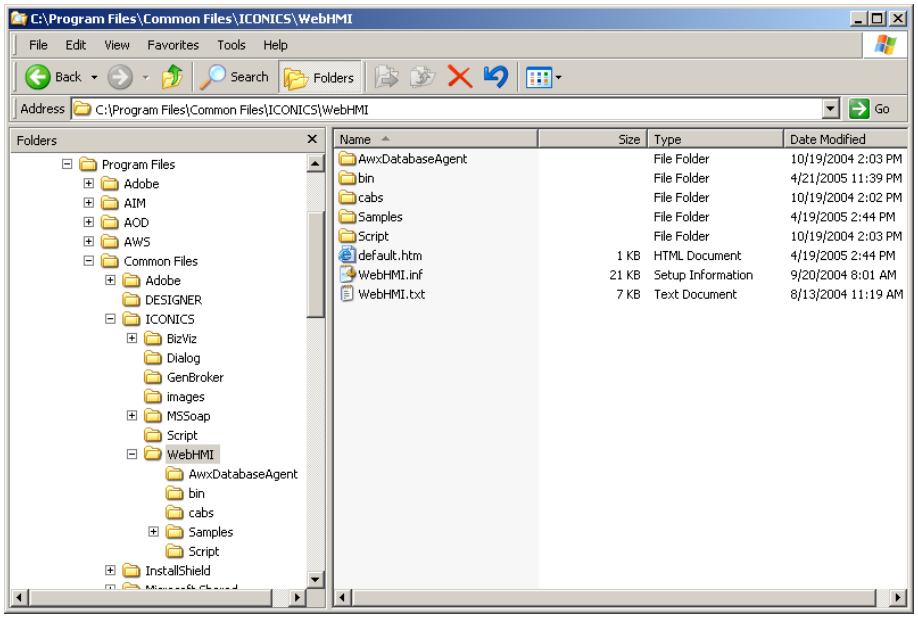

Figure L2.4: WebHMI Directory

2. The **bin** folder holds all your WebHMI tools and executables, necessary for your WebHMI Server machine.

3. The **cabs** folder holds all the ICONICS ActiveX controls, packed up into **cabinet** files (a type of compressed file) for download by the client. The client downloads these files when necessary and installs it on itself in order to run ActiveX components.

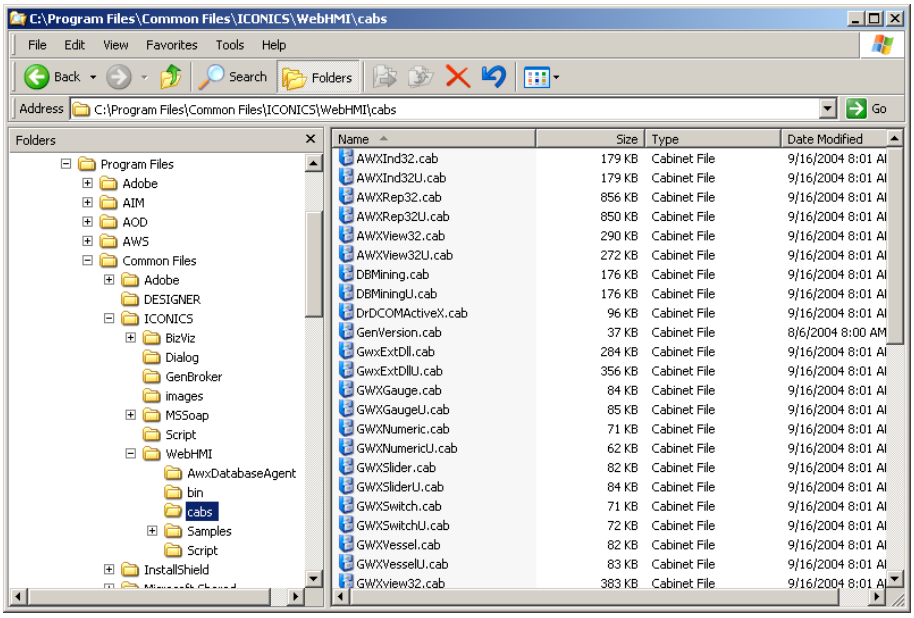

Figure L2.5: WebHMI Cabs Directory

Visualize Your Enterprise™

4. The **Samples** directory holds a number of sample GraphWorX32 displays, published to the WebHMI server, meant for testing and demonstration of the WebHMI installation.

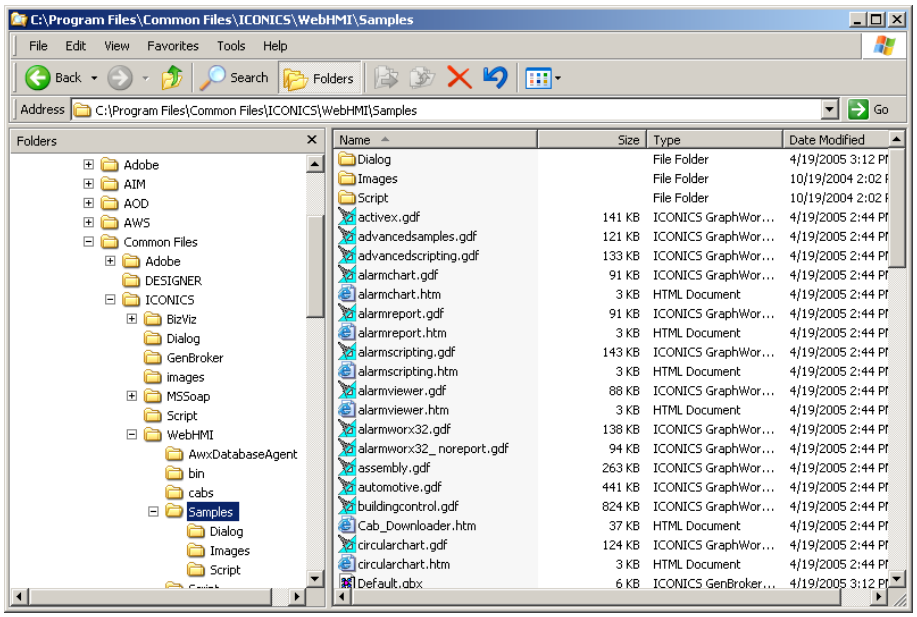

Figure L2.6: WebHMI Samples Directory

5. Also located in this directory is the **Default.gbx** file. This is the default GenBroker Configuration file, used to control the communications between the WebHMI server and client machines. You will learn more about this file, and the will use the sample published files later in this course.

GenBroker Communications Overview 3

3.1 Objectives

In this chapter, you will learn how to use GenBroker Communications, which is used in WebHMI. After completing this chapter, you should be able to:

- Understand the two different uses for GenBroker communications
- Distinguish between five different types of GenBroker communications
- Know the basic setup requirements on server and client nodes

3.2 Overview of GenBroker Communications

GenBroker uses five basic methods of networked communications:

- OPC direct channel
- Direct channel over DCOM
- Direct channel over TCP/IP
- Direct channel over SOAP/XML
- Indirect channel via a mediator node

Most channels, either direct or indirect, translate the OPC COM calls into one of two protocols (TCP/IP or SOAP/XML) and transmit them to the remote node, where GenBroker can execute them. The OPC Direct channel is the only exception; it uses OPC calls with DCOM protocol.

- **DCOM:** Suitable for local networks enabled with DCOM. Not recommended for Internet use.
- **TCP/IP:** Communicates over intranet/Internet through routers and firewalls.
- **SOAP/XML:** Communicates over intranet/Internet through routers and firewalls; does not require port openings.

It is very important to note that there are two places to configure GenBroker communications, and only one of them is necessary for running WebHMI, as explained in the next two sections.

3.2.1 GenBroker Configuration for OPC Servers and Clients

The main **GenBroker Configurator** found in **Start → Programs → ICONICS Tools** is shown in **Figure 3.1.** This configuration tool is used to configure the GenBroker settings for OPC data communications. When the configurator is first opened, it will prompt to input a name for the file and then display the configuration screen seen below.

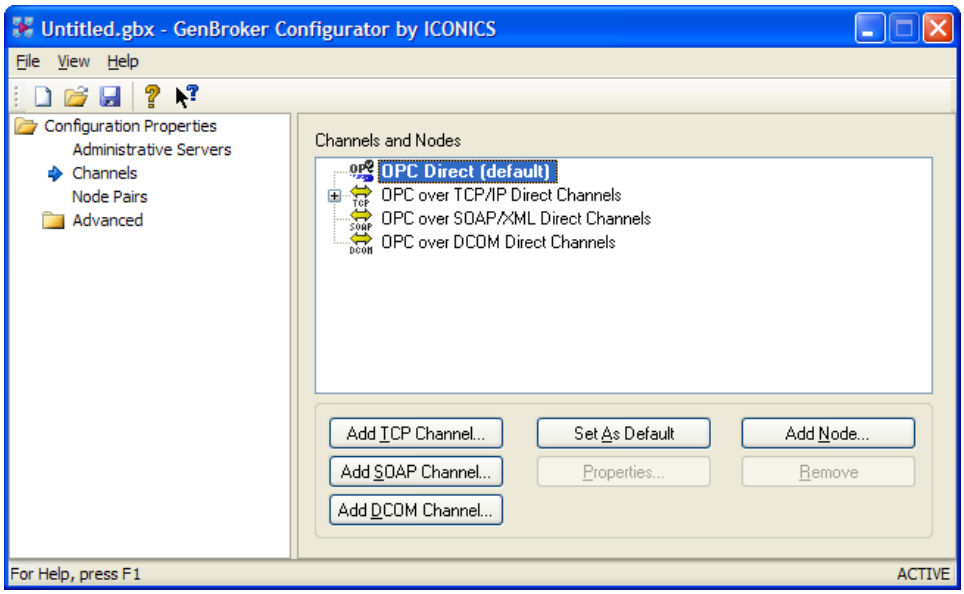

Figure 3.1: GenBroker Configurator for OPC Data

GenBroker configurations done with this tool define how OPC data is communicated between OPC servers and clients either locally or remotely. When configuring a computer to be a WebHMI server, if all OPC data is local (i.e. all OPC servers are located on the WebHMI server or its local area network) enabling GenBroker communications is not necessary (no changes need to be made). The configuration shown in **Figure 3.1** is the default configuration, and will allow OPC communication between the OPC servers and clients through DCOM. This is the case where the current GENESIS32 machine is a client to itself and can use DCOM to talk OPC.

If OPC data will be received remotely or through a firewall or router, GenBroker needs to be configured a little differently. This is the case where the current GENESIS32 machine is a client to remote OPC data, and DCOM communications is not sufficient. The details of the protocol options available when creating a configuration file to be used by GenBroker Communications in this case are explained in Section 3.3.

3.2.2 GenBroker Configuration for WebHMI Clients

The WebHMI **Client Configuration** tool found in **Start → Programs → ICONICS WebHMI** is the second place GenBroker communications can be configured. This is where a GenBroker Configuration file (.gbx) is created for use in WebHMI, and is shown in **Figure 3.2.** We will concentrate on the use of the Client Configuration GenBroker tool in this class, rather than the GenBroker Configurator tool shown in **Figure 3.1.**

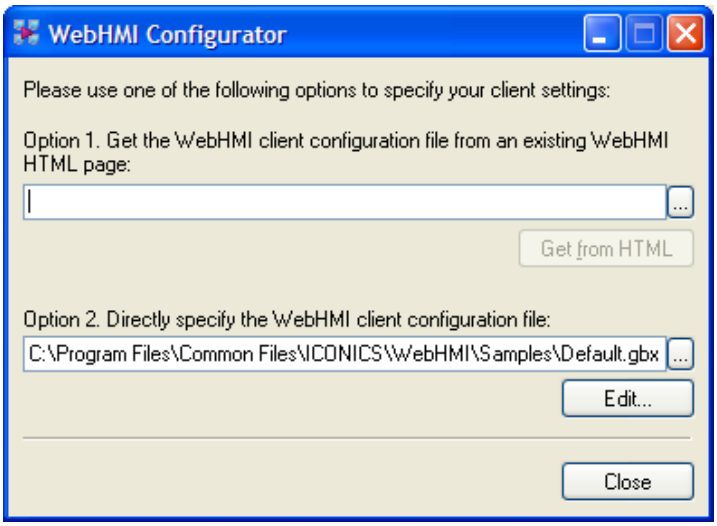

Figure 3.2: GenBroker Configurator for Client Configuration

NOTE: This tool is ONLY a file creation tool, and in no way selects a configuration file to be used in WebHMI. Configuration File selection must be done during file publication. By default, a GenBroker Configuration file (for WebHMI client configuration) **default.gbx** is used when publishing files to WebHMI, and is located in the C:\Program Files\Common Files\ICONICS\WebHMI\Samples directory.

During installation of ICONICS WebHMI, the **default.gbx** file is modified to work with your system based on the server name you enter. For this reason, all that should be necessary on a basic network setup to get WebHMI up and running for the first time is to START GenBroker Communications. However, for more complicated networks, a different configuration file may need to be created.

3.3 GenBroker Communication Options

Note that the configuration files created in the Client Configurations tool shown in **Figure 3.2** and the GenBroker Configurator for OPC data shown in **Figure 3.1** are of exactly the same format. However, it is very important to remember that they are for two completely different uses and should be configured differently. This section explains the protocol options available in both of these configuration files, and their uses.

3.3.1 Using OPC Direct

In an **OPC direct channel**, the client applications, such as GraphWorX32, TrendWorX32, and AlarmWorX32, communicate directly with OPC servers. As shown in **Figure 3.3,** this method uses standard fast DCOM over an intranet or local area network (LAN) but does not allow you to set a connection retry period or timeout settings. This method of communication is similar to that of OLExpress.

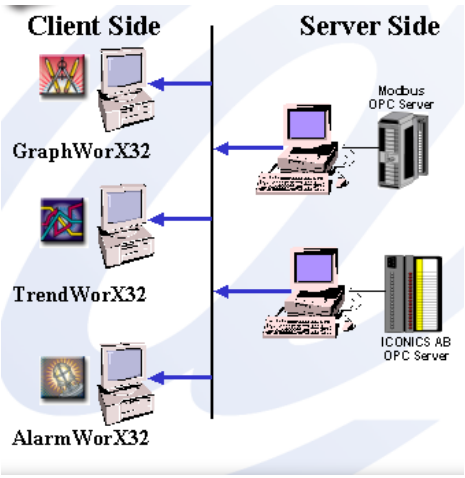

Figure 3.3: OPC Direct (Intranet Only)

3.3.2 Using a Direct Channel Over DCOM

In **direct channel over DCOM**, client applications communicate with GenBroker software located on remote nodes on an intranet or LAN, as shown in **Figure 3.4.** This method allows you to set a connection retry period and timeout settings. GenBroker then communicates (locally) with the target OPC servers on behalf of the client application. This is a predefined direct channel that users cannot delete from a GenBroker configuration.

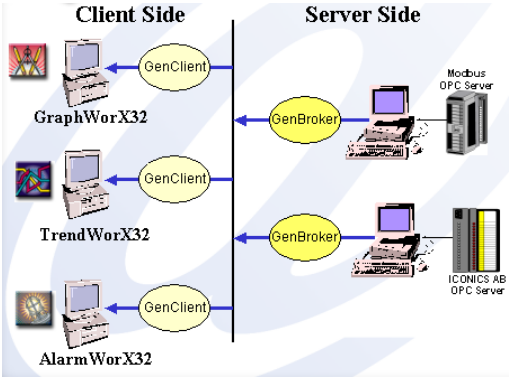

Figure 3.4: OPC Direct Channel Over DCOM (Intranet Only)

3.3.3 Using a Direct Channel Over TCP/IP

In a **direct channel over TCP/IP**, the client applications communicate with GenBroker software (GenClient) located on remote nodes on an intranet and over the Internet, as shown in **Figure 3.5.** This allows you to set a connection retry period and timeout settings. GenBroker then communicates (locally) with the target OPC servers on behalf of the client application. This is a predefined direct channel that users cannot delete from a GenBroker configuration.

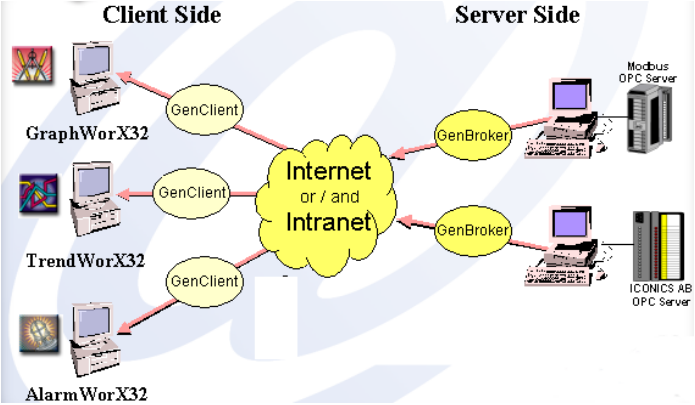

Figure 3.5: OPC Direct Channel Over TCP/IP (Intranet and/or Internet)

3.3.4 Using a Direct Channel Over SOAP/XML

In a **direct channel over SOAP/XML**, GenBroker responds to requests from clients that use the SOAP/XML protocol. **S**imple **O**bject **A**ccess **P**rotocol (SOAP) uses e**X**tensible **M**arkup **L**anguage (XML) to communicate over HTTP. The main advantage of using SOAP is that because SOAP operates on the HTTP level (through Port 80), data can pass through a firewall without having to change the firewall settings to open ports. The disadvantage of SOAP, however, is that it is slightly slower than TCP/IP.

NOTE: In order to use SOAP/XML channels, you must first install Microsoft Internet Information Server (IIS).

3.3.5 Using an Indirect Channel Via a Mediator Node

In an **indirect channel**, the client applications do not communicate directly with the final node, but instead communicate over the Internet with a **mediator node.** As shown in **Figure 3.6,** the mediator node communicates with the final node on behalf of the client. The mediator can communicate with OPC servers over an intranet or a LAN. The client applications do not need to have any direct connection with the node where the GenBroker utility in use is located in this case.

This is typically the type of channel used in a WebHMI client configuration file, allowing computers anywhere on the Internet to communicate with the WebHMI server although they are not directly connected.

Indirect channels are not included in a default NEW configuration (not to be confused with **default.gbx**, which has an indirect channel configured in it). However, you can create, configure, and delete indirect channels using either of the two GenBroker Configurator tools. The GenBroker Configurator allows you to specify a mediator node for the channel. If you want to route all requests for non-specified nodes through this mediator, then set the channel as default.

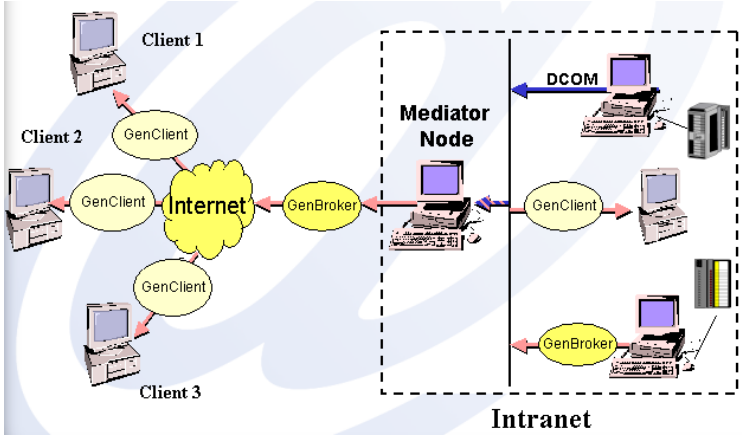

Figure 3.6: OPC Indirect Channel via a Mediator Node

NOTE: For more details about GenBroker communications, please see the *WebHMI Reference Guide.*

3.4 GenBroker Configuration

For WebHMI clients to be able to communicate with the WebHMI server, GenBroker communications need to be set up for both the server and the clients.

3.4.1 WebHMI Client Configuration

Configuring GenBroker Communications for a WebHMI client consists of creating a **.gbx** file and defining the protocols that will be used in communication between the WebHMI client and the WebHMI server. This file is referred to when publishing your displays to your Web server and downloaded by the WebHMI client before viewing displays.

Each WebHMI installation includes a default.gbx file, shown in **Figure 3.7,** which is used in WebHMI GenBroker Communications unless otherwise configured.

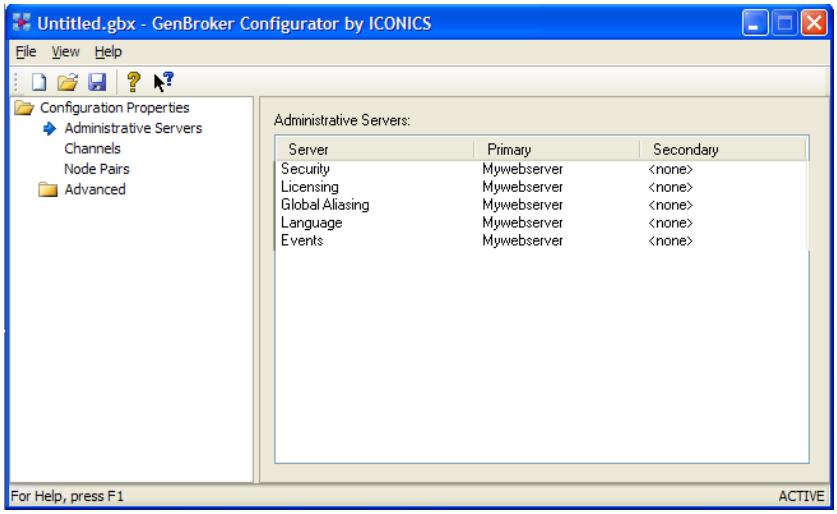

Figure 3.7: Default.gbx GenBroker Configuration File

3.4.2 Running the GenBroker Server

Once GenBroker Communications are configured for both the WebHMI server (optional) and clients, the GenBroker Server needs to be started. This needs to be **running on the server** for the WebHMI server and clients to communicate in any way other than DCOM. This is also true when GenBroker communications are being used to communicate OPC data between GENESIS32 machines. The GenBroker server should always be running on the server when two machines are using GenBroker communications.

The easiest way to start and stop the GenBroker Server is from GenTray, as shown in **Figure 3.8.**

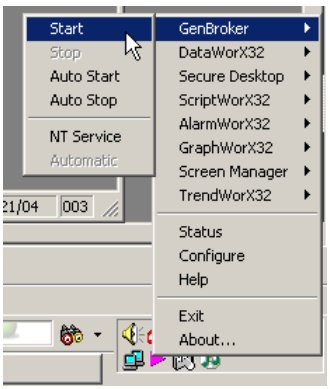

Figure 3.8: Starting the GenBroker Server

Quick Lab: GenBroker Configuration 3

3.1 Quick Lab Overview

Recall that there are two types of GenBroker configurations: for a GENESIS32 system's OPC communications with other GENESIS32 systems, and the WebHMI client's communication with the WebHMI server. For systems where all the OPC data is local to the WebHMI server, no special configuration is needed on the server side. This lab will help you to make the necessary configurations on both the server and client side of the simple networking WebHMI project (Phase 1) requested by Acme.

3.1.1 Task Checklist

In this quick lab, you will complete the following tasks:

- Understand the settings in the **Default.gbx** GenBroker Configuration file for client-side configuration
- Learn how to start the GenBroker Server for server-side configuration

3.1.2 Support Files

To complete this quick lab, you will need the following support files:

- Default.gbx

3.2 WebHMI Client Configuration

The node name is used in this example, assuming the application will be used on an intranet or local area network. This application could easily be converted to be fully World Wide Web enabled by replacing the node name with the IP address of the Web server.

1. Recall that the **Client Configuration** tool is a means of creating and editing .gbx files for WebHMI. Open this tool from **Start → Programs → ICONICS**

WebHMI → Client Configuration. The configurator will look like **Figure L3.1.**

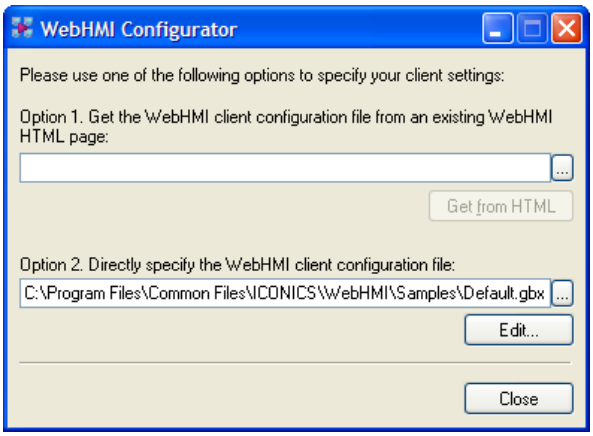

Figure L3.1: GenBroker Client Configuration Tool

2. For this basic example, and for most WebHMI applications, the **Default.gbx** file is configured appropriately. Open the **Default.gbx** file from the **C:\Program Files\Common Files\ICONICS\WebHMI\Samples** directory by pressing the ... button.

3. Click the **Edit** button to view the file so that we can understand how it is configured. The configuration file is shown in **Figure L3.2.**

<u>NOTE:</u> In the following analysis of the Default.gbx file, you will see the node name of the server you are working on rather than **MyWebServer.**

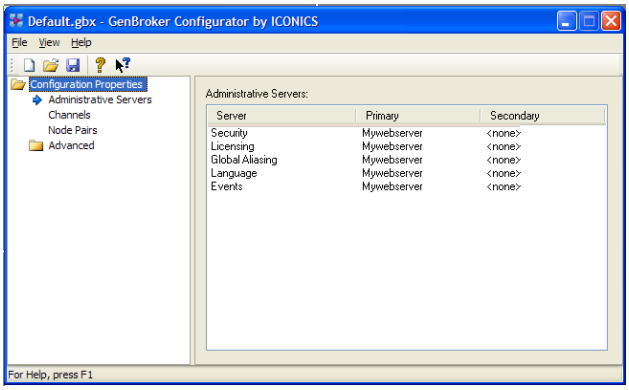

Figure L3.2: Configuration of Default.gbx

4. Note that the WebHMI server is configured as the **Primary** server for all of the servers listed (Security, Licensing, Global Aliasing, Language, and GenEvent). This means that the WebHMI clients will get all of their Security, Licensing, Global Aliasing, Language, and GenEvent information from the WebHMI server.

Visualize Your Enterprise™

5. Double clicking on the server itself in this list allows you to configure the primary and backup locations for these servers. A node listed under a channel in the configuration can be set as the primary or backup for all servers by right clicking on the node and selecting **Primary Servers** or **Backup Servers**.

6. Notice the use of an **indirect channel** with a **mediator node.** During WebHMI installation you are asked to enter the name of your WebHMI server. This configuration file is automatically created so that the mediator node of the indirect channel is your WebHMI server. In the example shown in **Figure L3.3** the WebHMI server is named **Mywebserver**.

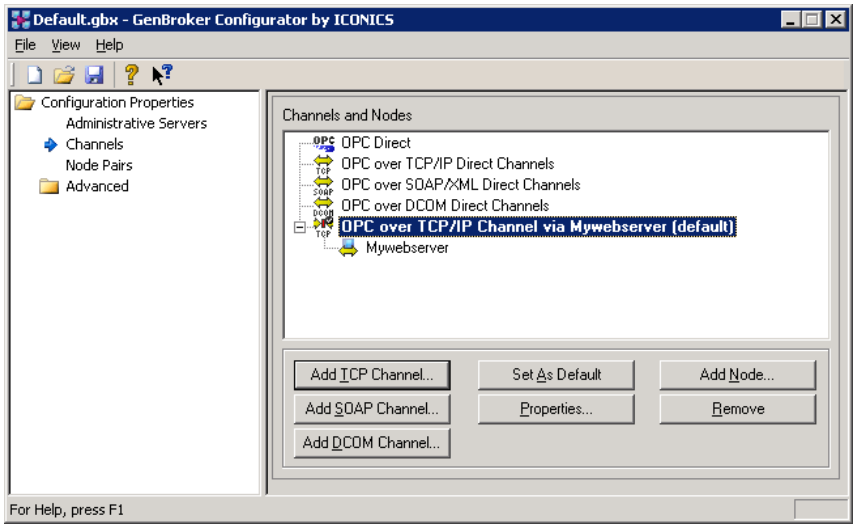

Figure L3.3: Configuration of Default.gbx

7. If you need to create an indirect channel, it can be cone by using the three buttons shown in **Figure L3.3** in any of the following three protocols: TCP/IP, SOAP/XML, or DCOM. You will then be asked to enter the name of the **Mediator Node**.

8. When data is requested from a certain node on a WebHMI client, the .gbx client configuration file is referred to. If the node is listed explicitly underneath one of the channels, that type of communication is the only protocol that will be used in communicating with that node. If the node is not listed explicitly in the configuration file, then the protocol selected as default will be used. In default.gbx, the OPC Direct protocol is the default protocol. A channel can be set as default by right clicking the channel and selecting **Set as Default**.

9. In the example shown in **Figure L3.3** (default.gbx), all the nodes listed underneath the indirect channel will automatically be routed through the mediator node using the specified protocol. The WebHMI server is the only one listed, although others can easily be added. This is sufficient since we know that all OPC, DA, A/E, and HDA data is located on the server itself (explained in

Chapter 1). If data were requested from any node other than the WebHMI server (Mywebserver), it would be communicated through OPC Direct protocol.

NOTE: If a node PC1 were added under the indirect channel configured in default.gbx, any time a WebHMI client requested data from PC1, communication would be begin with Mywebserver (the WebHMI server) using TCP/IP protocol, and then Mywebserver would initiate communication with PC1 using whatever type of protocol is configured for that node in the GenBroker Configuration active on the WebHMI server. This way, the WebHMI client would not need to be directly connected to the node data is requested from in any way. A node can be added underneath any channel by right clicking on the channel and selecting **Add Node**.

10. Now that you understand what the default.gbx file is configured to do, click **Cancel** to exit the configurator leaving the file in its original state.

NOTE: Remember that opening or creating a .gbx file using the Client Configuration tool in no way chooses that file as the file to be used by the WebHMI clients. This will be set during publication of a display.

3.3 Running the GenBroker Server

Once GenBroker Communications are configured for both the WebHMI server and clients, the GenBroker Server needs to be started. This needs to be **running on the server** for the WebHMI server and clients to communicate in any way other than DCOM. This is also true when GenBroker communications are being used to communicate OPC data between GENESIS32 machines. The GenBroker server should always be running on the server when two machines are using GenBroker communications.

1. The best way to start GenBroker is through **GenTray** (or a GenTray configuration in ProjectWorX32). Open and start GenTray from **Start → Programs → ICONICS Tools → GenTray**, or through ProjectWorX32 if you use ProjectWorX32 to organize and manage your projects.

2. When GenTray starts up, you will see the GenTray icon in your system tray, as shown in **Figure L3.4.**

Figure L3.4: GenTray Shown in the System Tray

3. To start GenBroker from GenTray, click on the icon and select **GenBroker→ Start**, as shown in **Figure L3.5.**

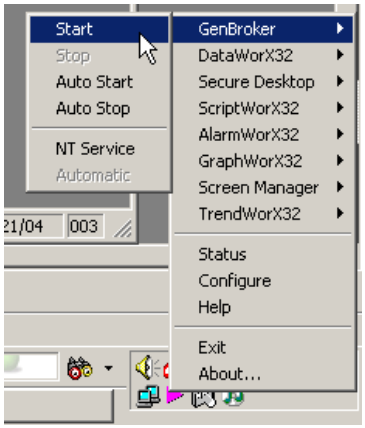

Figure L3.5: Starting GenBroker from GenTray

Testing and Troubleshooting the Web Server 4

4.1 Objectives

In this chapter, you will learn how to test and troubleshoot your Web server before your WebHMI project is in place. After completing this chapter, you should be able to:

- Use sample WebHMI files to test the communication between your WebHMI server and clients
- Be able to troubleshoot common problems in WebHMI communications

4.2 Testing the Web Server

Remember that proper development of any project should include incremental testing along the way. Before moving on to creating and viewing your WebHMI display, the Web Server should be tested to be sure that the communication is correctly configured and operating correctly.

The client machine used for testing should have the following characteristics:

- On the same local area network as the WebHMI server (the Default.gbx file is setup for a client on the same network since the node name is used and not the IP address)
- Does NOT have GENESIS32 installed
- Has Microsoft Internet Explorer 5.5 or greater installed

4.2.1 WebHMI Sample Files

As part of the WebHMI installation, several sample displays are installed on your machine and tailored to work on your Web server for viewing on WebHMI clients. As long as the name or address of the Web server was entered correctly during installation, and GenBroker is running on the WebHMI server machine, these sample pages should be viewable. All of these sample files use the **Default.gbx** GenBroker configuration file for communication.

The sample files are located in your **C:\Program Files\Common Files\ICONICS\WebHMI\Samples** directory, as shown in **Figure 4.1.** As you can see, many sample displays exist, enabling you to test every ICONICS ActiveX component, and many of the GraphWorX32 features that are supported over WebHMI without having to develop your own project first.

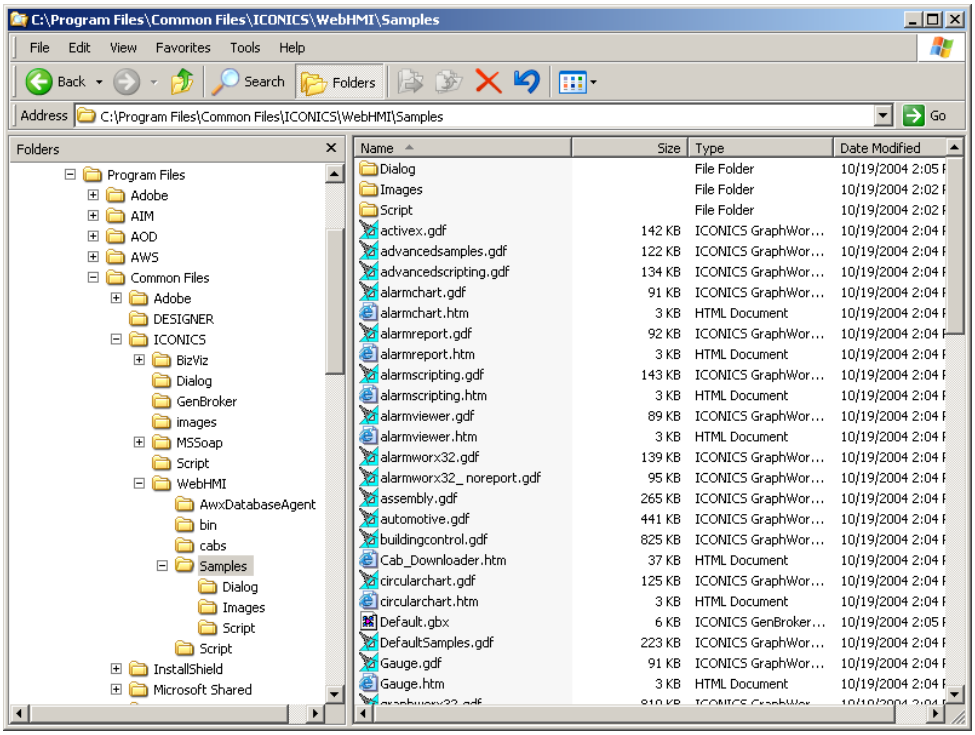

Figure 4.1: WebHMI Samples Directory

You can view the WebHMI Sample files over WebHMI by pointing the Web Brower on the client machine to <**http://ServerName/webhmi>.** This will redirect the browser to the index page for the sample project, <**http://ServerName/webhmi/samples/index.htm>,** as shown in **Figure 4.2.**

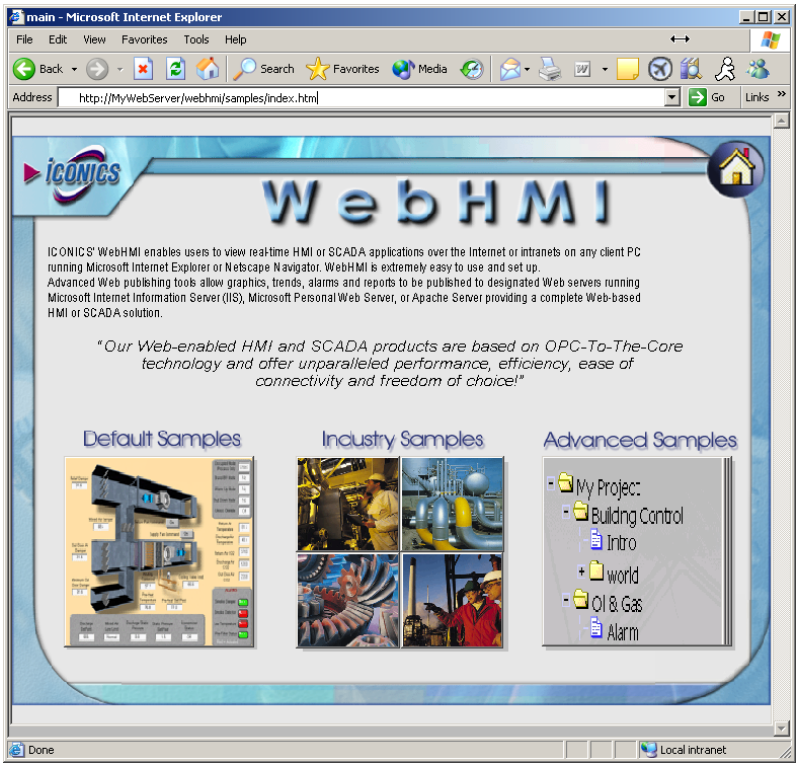

Figure 4.2: WebHMI Samples on a Client Machine

NOTE: **ServerName** in the URL examples should be replaced with the actual name or IP address of your server for the URL to work. The name of the server in the example shown in **Figure 4.2** is **MyWebServer.**

4.2.2 Troubleshooting Common Problems

Error! Reference source not found. summarizes the many common problems encountered while setting up and testing a WebHMI server. The table holds error messages or symptoms that you may encounter in the following lab. Possible solutions are proposed in the table as well.

Table 4.1 - Common Problems and Possible Solutions

Symptom	Possible Solutions
You are redirected to a search page	The .htm file does not exist, or is not at the address you are using. This could also mean that your Web server is not installed, or is installed incorrectly.
Unable to download .gbx file to cache	The .htm file refers to an incorrect path for the .gbx file, or the file does not exist. This could also mean that your Web server is not allowing the client permissions to download files. If your Web server is Win2003 Server, extra IIS configurations may be necessary. See the appropriate application notes.
License Failure	Either GenBroker is not running on the server, or the client cannot communicate with the server using the protocol chosen in the .gbx file (firewalls, etc. could block the communication).
Unable to determine security status	The remote servers (specifically the security server node) are not configured correctly in the .gbx file (GenBroker configuration).
*** is seen instead of OPC data in process points	The tag paths may not be correctly set for remote communication. The wrong node name/IP address may have been used during installation of WebHMI, which would be used in these sample displays. The Host Name Change utility can be used to fix this problem in existing displays.
The alarm viewer ActiveX control is not on the page	The ActiveX control was not downloaded. Make sure that you are logged into the computer with permissions to download and install such objects. If your Web server is Win2003 Server, extra IIS configurations may be necessary. See the appropriate application notes.
No data is shown in the alarm viewer (just a gray screen)	The alarm server may not have an active configuration on the WebHMI server, or the subscription in the alarm viewer may not be correctly set for remote communication. The wrong node name/IP address may have been used during installation of WebHMI, which would be used in these sample displays. The Host Name Change utility can be used to fix this problem in existing displays.
ActiveX controls are not shown where they should be	The ActiveX control was not downloaded. Make sure that you are logged into the computer with permissions to download and install such objects. If your Web server is Win2003 Server, extra IIS configurations may be necessary. See the appropriate application notes.
There is no data displayed in your Alarm Report ActiveX	Make sure that you have enabled Web Access in the ActiveX properties. If you have Access Connection, then it could be that someone else has the page open. Access has a very limited number of concurrent connections to a database. If you are connecting to an SQL database, make sure you are using a specific database user to log-in instead of Windows credentials.

Quick Lab: Testing the Web Server 4

4.1 Quick Lab Overview

In this lab, you will use the WebHMI Sample project to test that the Web Server and WebHMI communications are working properly. By the end of this lab, you should be able to view OPC data on a remote client. You should also be able to view all ICONICS ActiveX controls properly on the WebHMI client.

4.1.1 Task Checklist

In this quick lab, you will complete the following tasks:

- Learn how to browse a WebHMI project from a remote client machine
- Use the WebHMI Sample project to test and troubleshoot the communications between the WebHMI server and client

4.1.2 Support Files

To complete this quick lab, you will need the following support files:

- All sample files installed by WebHMI in the **C:\Program Files\Common Files\ICONICS\WebHMI\Samples** directory

4.2 Testing the Web Server

To be sure that the WebHMI server is configured and communicating correctly (with GenBroker running), go through the following steps. If at any point things are not working correctly, refer to the troubleshooting information in Chapter 4 to find a list of common problems and their solutions.

1. Find a machine with the following characteristics as the test client:

- On the same local area network as the WebHMI server (the Default.gbx file is set up for a client on the same network since the node name is used and not the IP address)

- Does NOT have GENESIS32 installed
- Has Microsoft Internet Explorer 5.5 or greater installed

2. Open Internet Explorer on the client and point the browser to <http://ServerName/WebHMI>. In the example in this lab the Web server name is **Mywebserver**, in which case the address would be **<http://Mywebserver/WebHMI>**. You should use your actual server name.

NOTE: During installation of WebHMI, some settings in the Internet Information Services (IIS) are configured so that the WebHMI sample index page is the default web page for that server. For this reason, simply pointing to the WebHMI folder on the server will bring you directly to the sample displays on the server.

3. You will be asked to download objects while the page is the loading. These are the things that must be loaded onto the WebHMI client such as the .gbx GenBroker configuration file, and ActiveX controls like the GraphWorX32 viewer, AWXView32, TWXView32, and others. Authorize all necessary downloads.

NOTE: You will only be prompted to download what is necessary for the display (or set of displays) you are trying to view, and only if that object is not already present on the machine (from a previous display viewing).

4. After downloading the necessary objects you should see the main display (or index page) for the WebHMI samples, as shown in **Figure L4.1.**

NOTE: Notice that the browser is automatically redirected to **<http://Mywebserver/WebHMI/ Samples/Index.htm>**.

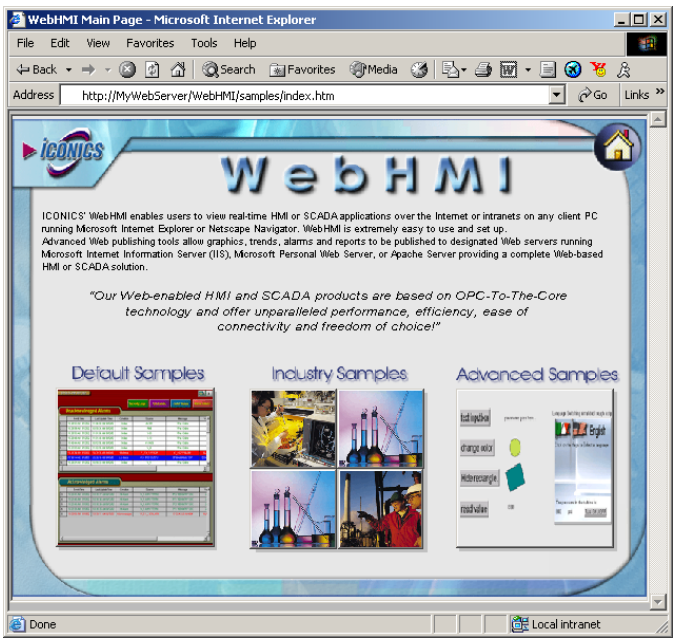

Figure L4.1: Main WebHMI Sample Page

5. To test each of the ICONICS ActiveX controls, click on the **Default Samples** section. This will bring you to another page, shown in **Figure L4.2,** containing links to sample pages containing each of the main ICONICS ActiveX controls used.

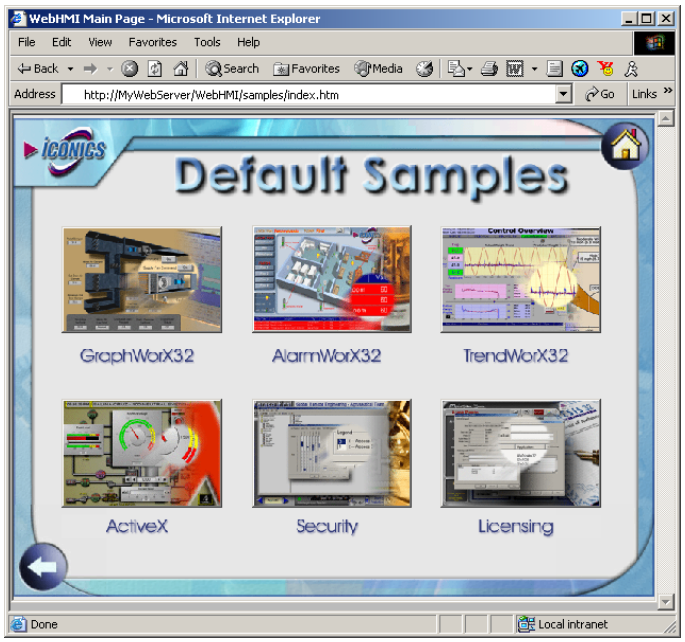

Figure L4.2: WebHMI Default Samples Page

6. To test the OPC communication between the client and the server, click on the **GraphWorX32** link to see the display shown in **Figure 5.15**.

NOTE: You should not be prompted to download any ActiveX controls for this display because it only uses the GraphWorX32 Viewer ActiveX control, which has already been downloaded for the previous displays.

7. You should see simulated OPC data updating on the display and animation based on that data.

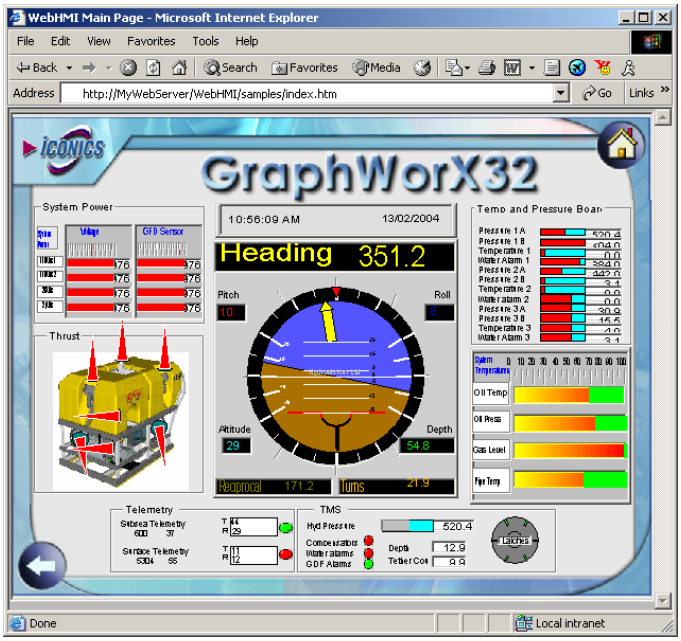

Figure L4.3: WebHMI GraphWorX32 Sample Page

8. Hit the back arrow button at the bottom left-hand corner of the screen to return to the **Default Samples** page. This time click the **AlarmWorX32** link to view the display shown in **Figure L4.4.**

NOTE: If you click the **Back** button of the Web Browser page, you will be returned to the index page, because from that point new html pages have not been loaded. Load display pick actions have been executed within the GraphWorX32 Viewer ActiveX control.

Visualize Your Enterprise™

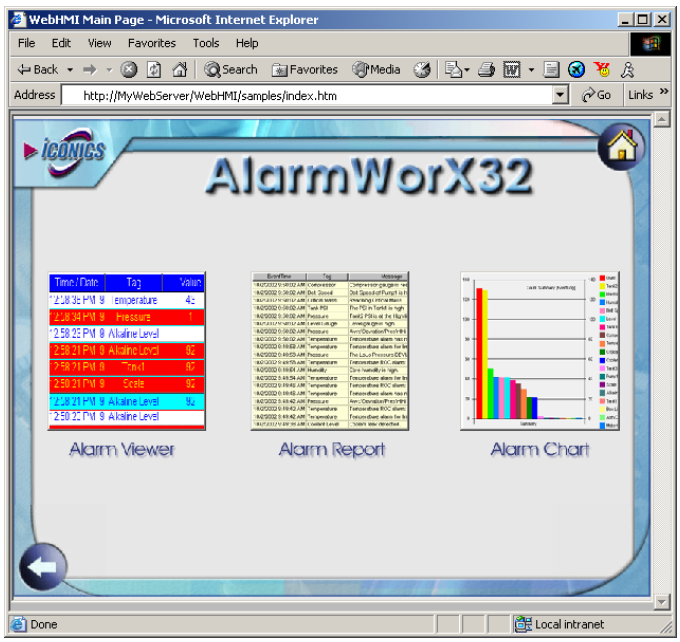

Figure L4.4: WebHMI AlarmWorX32 Sample Page

9. Click on the **Alarm Viewer** link to open a display with an AWXView32 ActiveX control, which will connect to live alarm and event data on the WebHMI server.

10. You should be prompted to download the AWXView32 ActiveX control before the display loads. After the download, the display should look similar to **Figure L4.5.**

NOTE: You are prompted for a download here because a new URL is being loaded, rather than just a Load Display pick action being executed. Notice that the URL changes at this point. This new HTML page has a reference to the ActiveX control embedded in this display, where the main HTML page did not.

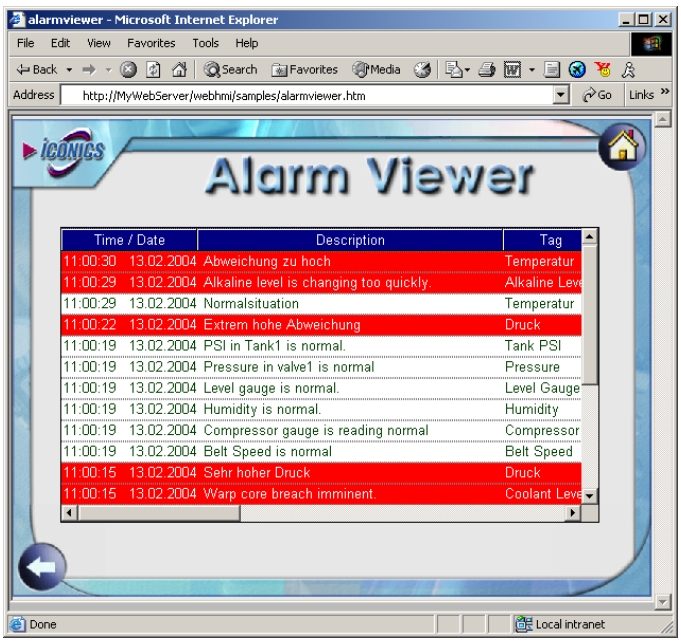

Figure L4.5: WebHMI Alarm Viewer Sample Page

11. Click the back arrow on the bottom of the page to return to the AlarmWorX32 sample page, and continue to view the other AlarmWorX32 pages to test those ActiveX controls.

12. Continue to browse through all of the other Default Sample pages (TrendWorX32, ActiveX, Security, and Licensing) to test the ActiveX controls used in those displays.

13. After browsing through all the **Default Samples,** return to the samples index page by pointing your browser to **<http://MyWebServer/WebHMI>** (replacing MyWebServer with your actual server name).

14. Click on the **Advanced Samples** section on the right to open the page shown in **Figure L4.6.**

Visualize Your Enterprise™

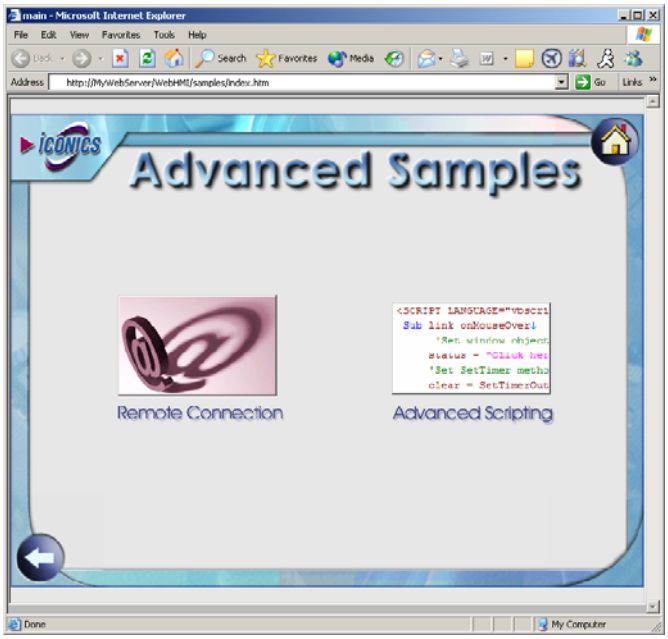

Figure L4.6: Advanced Samples Page

15. Click the **Remote Connection** button to open the page shown in **Figure L4.7.** This will allow us to test the OPC connection to the Web server by viewing OPC data located on the server (all data seen in the previous displays were simulated locally).

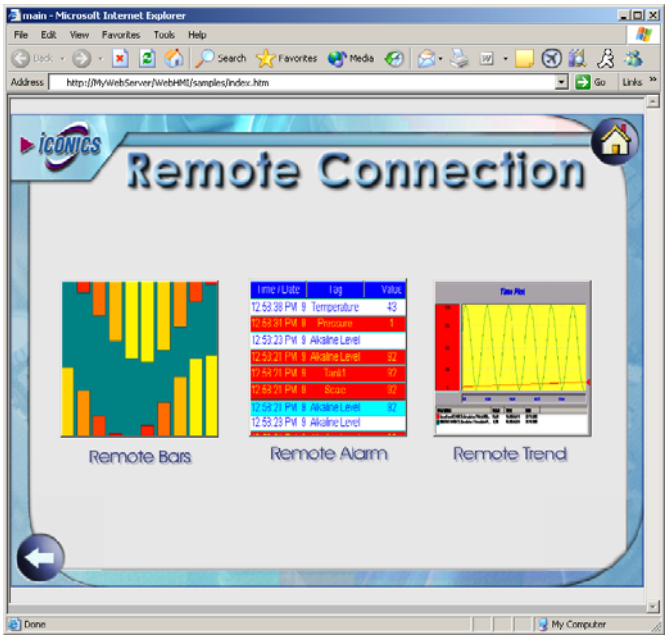

Figure L4.7: Remote Connection Sample Page

16. Test the OPC DA data by connecting to the ICONICS Simulator OPC Server located on the WebHMI server. Do this by clicking the **Remote Bars** button to load the display shown in **Figure L4.8.**

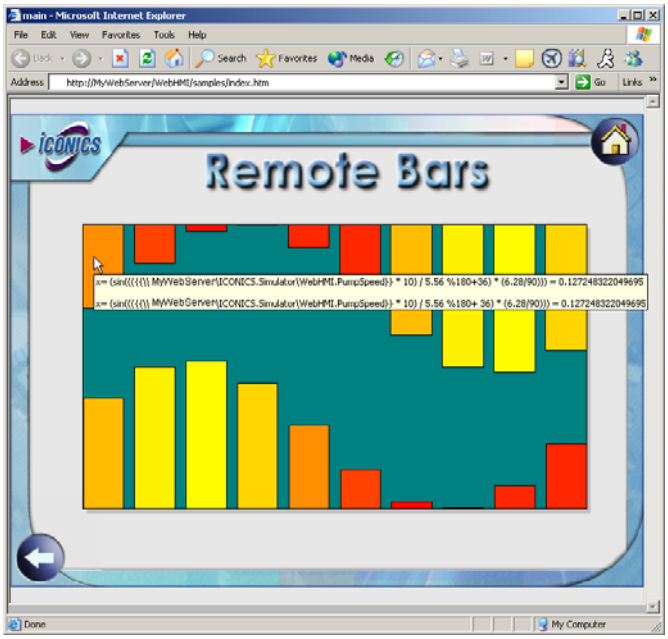

Figure L4.8: Remote Bars Sample Page

17. Rest your mouse over one of the bars to see the Tool Tip, which reads the remote tag the object is connected to and its current value. This is shown in **Figure L4.8.** If you are not successfully connected, all bars will be grayed out, and the Tool Tip will show **(bad quality)** after the tag path.

Visualize Your Enterprise™

Creating Network – Aware Displays 5

5.1 Objectives

In this chapter, you will learn the preparations required for GraphworX32 projects that will be published to WebHMI. By the end of this chapter, you will be able to:

- Know how to make real-time and historical data items network-aware
- Know how to reference other displays for use over WebHMI
- Alter existing displays for use over WebHMI

5.2 Making Data Items Network-Aware

Your displays that will be used by WebHMI need to be "network-aware". This means that when the displays are viewed on remote computers via WebHMI, the data and file requests in that display need to include enough information to find that data on the server across the network or Internet.

In this class, we are designing a WebHMI application to be used on a Local Area Network (LAN) or Intranet, so node names can be used. If the application were to be used on the World Wide Web, the IP or Web address would be needed to find the server. For now we will concentrate on preparing the displays to be used across a LAN, and the other case will be addressed in Practice Lab 8.

⇒ **Tag Definitions**

OPC tags in GraphWorX32 displays need to include the node name of the computer where the OPC Server is located, or in this case the WebHMI server name.

\\NodeName\OPCDAServerID\OPCItem

For example:

\\Mywebserver\ICONICS.Simulator.1\PLC1.Boiler.Temperature

You can add this information to the tag definition manually, or you can browse to the tag through the Network Neighborhood branch instead of the My Computer branch of the Unified Data Browser.

⇒ **TrendWorX32 History Pens**

Just like OPC tags, Historical Trend Pens in the TWXView32 ActiveX control (placed in a GraphWorX32 display) need to include the node name of the computer where the TrendWorX32 SQL Data Logger is located, or in this case the WebHMI server name.

\\NodeName\OPCHDAServerID\TrendTag

For example:

\\Mywebserver\ICONICS.TWXSQLSvr.1\msde.Boiler.Temperature

You can add this information to the tag definition manually, or you can browse to the tag through the Network Neighborhood branch instead of the My Computer branch of the Unified Data Browser.

⇒ **AlarmWorX32 Viewer Subscription**

In the alarm subscription of the AWXView32 ActiveX control the node name of the computer where the **OPC Alarm Server** is located, or in this case the WebHMI server, needs to be specified as shown in **Figure 5.1.**

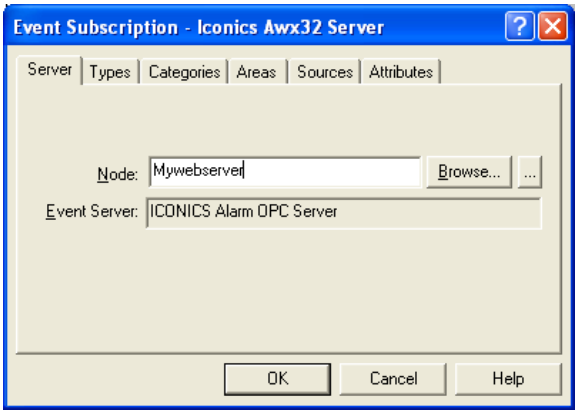

Figure 5.1: Remote Alarm Server Subscription

Just as with pen and tag definition, the alarm server can be browsed for through the Network Neighborhood of the Unified Data Browser to include the node name, or the node name can be entered manually.

⇒ AlarmWorX32 Report Configuration

The AlarmWorX32 Report ActiveX control pulls all of its information from a database. When using the alarm report over WebHMI, that database is located on the server machine potentially across the Internet from the client running the alarm report. Communication of this type is made possible in the ICONICS AlarmWorX32 Report ActiveX control via a Remote Data Agent (RDA).

The RDS, which is hosted by an Internet Information Server (IIS), enables the downloading of report data from the logger database to a client over the Internet. This feature must be enabled in the ActiveX properties before the display is published, as shown in **Figure 5.2.**

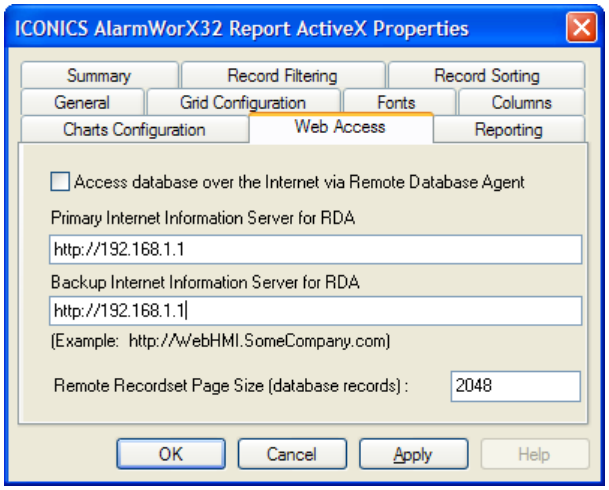

Figure 5.2: Enabling Web Access for the Alarm Report AcitveX

When Internet Access to the logger database is enabled, the client makes a request over the Internet to the IIS Remote Data Agent. The RDA relays the request to the local logger database on the server. The RDA retrieves the relevant information for the alarm report from the logger database so that the client can download the data from the server over the Internet. The client machine then displays the data in the Report ActiveX per the client's report configuration settings.

The alarm report ActiveX must also have the necessary authentication and permissions to access the database remotely. Windows NT Authentication will not work over the Internet (or outside the local domain), so a user's information with appropriate rights must be entered when establishing the connection to the database before publishing the display.

5.3 Preparing Displays for Network Use

In addition to data items, certain precautions should be taken with your GraphWorX32 displays to be sure that they will work properly with WebHMI and over a network or the Internet.

⇒ **References to Other Displays**

When references to other displays are made, for instance in a **Load Display** pick action, relative paths should be used. If all the displays are in the same folder during development, just the filename (i.e. filename.gdf) should be used when referring to the file. For more information about the use of relative path names, see the application note entitled **WebHMI Publishing – Maintaining Directory Structures.pdf** found on the installation CD.

⇒ **Saving the Displays Without VBA**

It is important to remember that the use of VBA is not supported over WebHMI. GraphWorX32 displays inherently contain support for VBA, so displays that are to be published to HTML for use with WebHMI must be saved as a format without VBA: **GraphWorX32 Displays without VBA (*.gdf)**.

5.4 Making Existing Displays Network-Aware

The tools discussed in this section can be helpful when altering existing displays to be network-friendly.

5.4.1 Edit Connections Tool

The **Edit Connections** tool in GraphWorX32 allows you to view every data connection in the display in list form, as well as export/import that information. These data connections are linked to the objects they reside in by the object name. An example is shown in

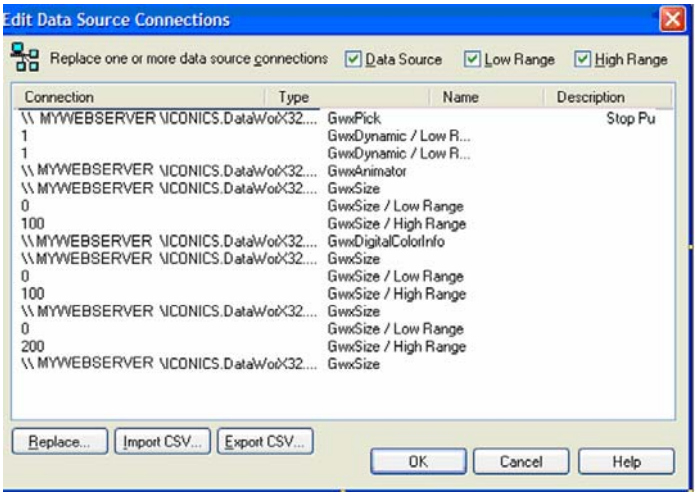

Figure 5.3: Edit Connections Tool

Using this tool, you can view every data source in the display to see whether or not the server path is included in the tags paths. If you find tags that do not have the server paths included, use the Replace tool to add the server path, as discussed in the next section.

5.4.2 Replace Tool

GraphWorX32 has a **Replace** feature, which can aid in updating displays and data items within them so that they are network-aware.

If you have IP addresses or node names in data items already, but they need to changed to that of your WebHMI server, you can search for the old IP address or node name, as shown in **Figure 5.4.**

You can use this tool to replace node names or IP addresses in **Dynamic Tags** in the current display or in multiple files at once. You can also use this tool to replace strings in **File Names** if necessary, among other things.

NOTE: Performing a replace on Dynamic Tags will not affect alarm subscription paths. This will need to be replaced manually or by using the **Change Host Name utility,** discussed in **Appendix A.**

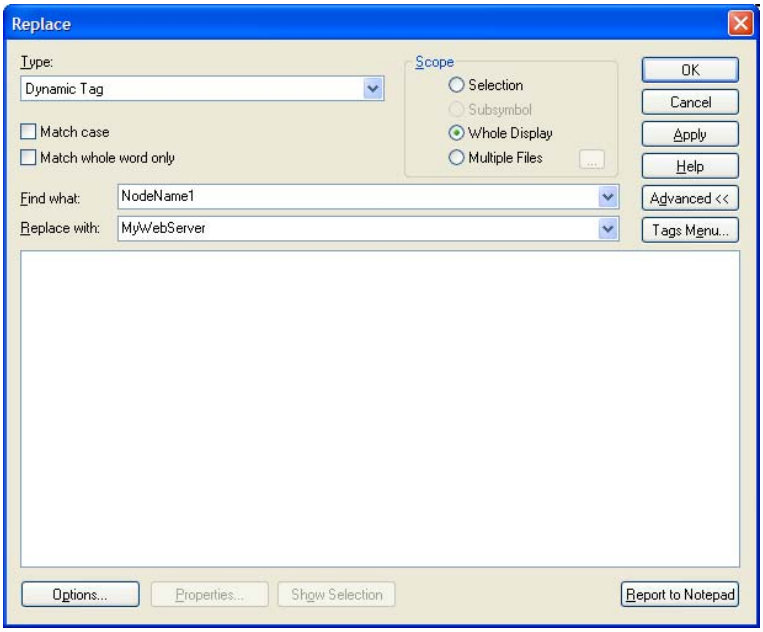

Figure 5.4: Replacing Node Names in GraphWorX32

If you do not have node names or IP addresses defined in your data items, you can still use the **Replace** tool to insert node names or IP addresses. However, you will need to be creative. As an example, consider the following the tag path:

ICONICS.Simulator.1\SimulatePLC.Ramp

You can append the server path (MyWebServer) with proper syntax to the beginning of all tags from the ICONICS.Simulator.1 OPC Server by searching for "ICONICS.Simulator.1" and replacing it with "\\MyWebServer\ICONICS. Simulator.1". The result will be:

\\MyWebServer\ICONICS. Simulator.1\SimulatePLC.Ramp

Quick Lab: Creating Network – Aware Displays 5

5.1 Quick Lab Overview

In this lab, you will take the GraphWorX32 displays that Acme currently has in place in their system (discussed in Quick Lab 1) and make sure that all data items are network-aware. You will also check that file paths are relative, and all file types are without VBA. After this lab, these displays will be prepared for use over WebHMI.

5.1.1 Task Checklist

In this quick lab, you will complete the following tasks:

- Use the Edit Connections tool in GraphWorX32 to view all data connections in a display
- Use the Replace tool in GraphWorX32 to insert server paths into all data items, or update server paths that are incorrect
- Insert node information into an alarm subscription
- Enable remote connection for an AlarmWorX32 Report ActiveX

5.1.2 Support Files

To complete this quick lab, you will need the following support files:

- All server configuration files (outlined in Lab 1)
- All GraphWorX32 displays (outlined in Lab 1)

5.2 Viewing Connections in GraphWorX32

The following steps will help you determine what changes need to be made in your Acme GraphWorX32 displays in order to use them over WebHMI.

1. Open **Operator3.gdf** just as a starting point. In configure mode, choose **Dynamics → Edit Connections** to open the **Edit Data Source Connections** window shown in **Figure L5.1.**

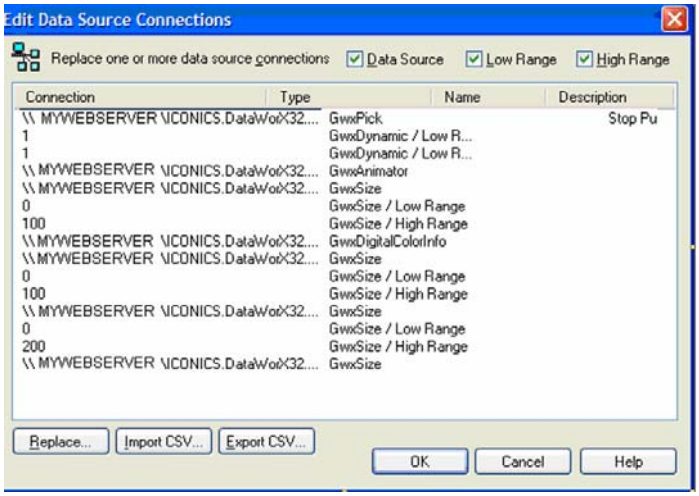

Figure L5.1: Edit Data Source Connections

2. You should notice that all data sources in this display already have the server's node name in them, so they do not need to be altered before using them over WebHMI.

3. Use the Edit Connections tool in each one of your Acme displays to determine which of them have data items in them that should be edited. You should find that all OPC DA data items in your displays are already prepared.

NOTE: If you find data connections in the previous three steps that do not have the server name in the tag path already, use the Replace tool as discussed in the end of Section 5.4.2.

5.3 Using the Replace Tool

The current Acme project has HDA data items in two displays: **HistTrendViewer.gdf** and **LevelPopup.gdf.** To illustrate the point, the historical pens in the LevelPopup display are already network aware, but those in the HistTrendViewer display are not. In this section, you will use the GraphWorX32 Replace tool to make sure that all OPC HDA data items in your displays are network-ready.

1. Open the **HistTrendViewer.gdf** display, and choose **Replace** from the **Edit** menu.

2. The **Replace** dialog, shown in **Figure L5.2,** will open. In the **Find what** field, enter "ICONICS.TWXSQLSvr.1". In the **Replace with** field, enter "\\MyWebServer\ICONICS.TWXSQLSvr.1", where **MyWebServer** is the name of your WebHMI server machine.

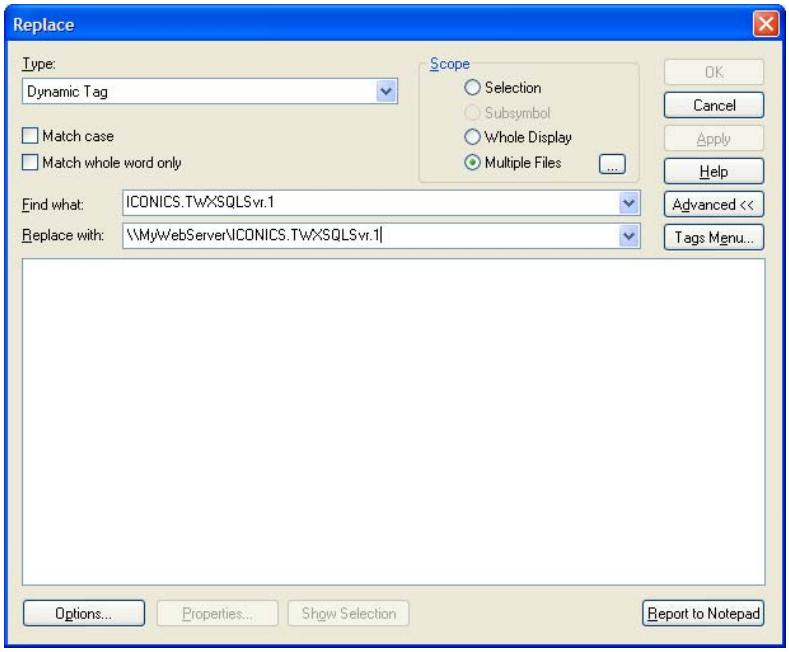

Figure L5.2: Replace Dialog

3. Choose **Multiple Files,** click the button, and add all your Acme displays to the list, as shown in **Figure L5.3,** and click **OK.**

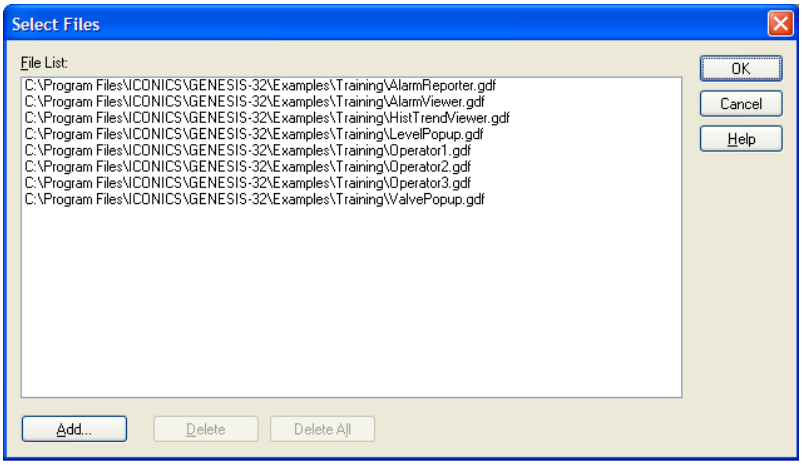

Figure L5.3: Select Multiple Files

4. In the **Replace** dialog, click **Apply** to start replacing items. You may get the warning shown in **Figure L5.4,** asking you to save your changes. Click **Yes** to continue.

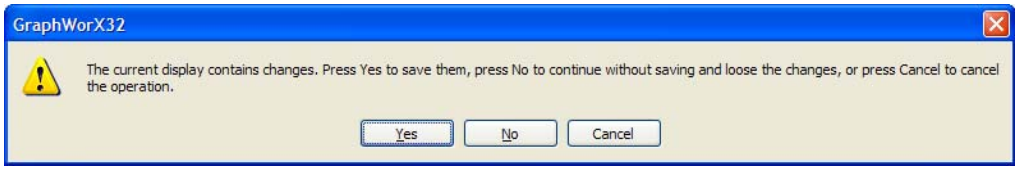

Figure L5.4: Save Changes Warning

5. You will be asked item by item if you would like to replace things. If you choose to expedite the process, choose **Yes to All.**

6. When replacements are finished, you will see a summary in the Replace dialog of all the items replaced, grouped by display. This information can be exported to a file if you desire.

7. Click **Close** to return to your display.

8. To confirm that your historical pens now have your server name in their tag path, double-click on the trend viewer ActiveX to open its **Properties** dialog.

9. Choose the **Pens** tab, and double-click on of the tags top open the **Edit Trend Pen** dialog. You should see the server path in tag path, as shown in **Figure L5.5.**

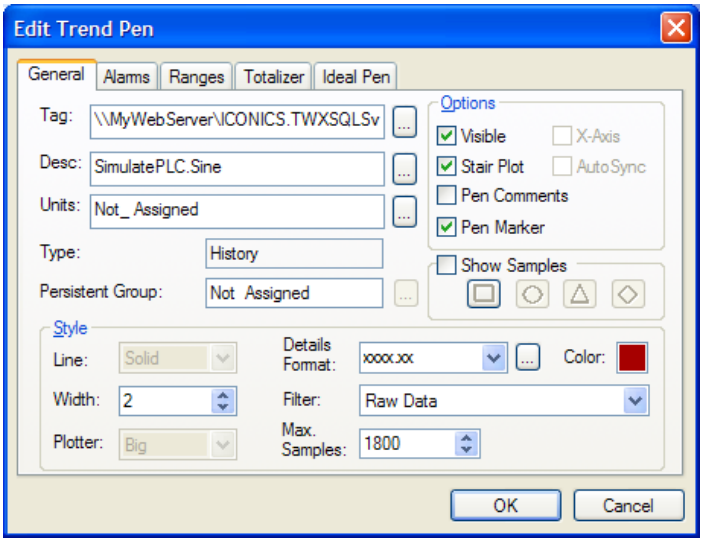

Figure L5.5: Confirming Change in Pen

10. All HDA data items in your displays should now be ready for remote communications.

5.4 Preparing Alarm Subscriptions

The replace tool does not have the capability of drilling into an OPC alarm and event subscription, so we will need to edit that manually.

1. Open the **AlarmViewer.gdf** file, and double click on the viewer to bring up the **Properties** dialog.

2. Choose the **Subscription** tab, and double-click on **Iconics Awx32 Server** to open the subscription properties for the ICONICS Alarm Server subscription.

3. Enter the node name of your server (in this example MyWebServer) into the **Node** field, as shown in **Figure L5.6.**

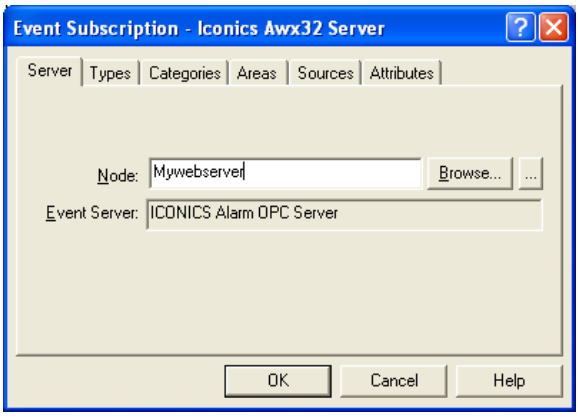

Figure L5.6: Subscription Node Settings

4. Click **OK** to save the changes and return to the **Subscription** tab of the Properties dialog.

5. Save the display and close it. It is now prepared for use remotely.

6. Follow these same steps, if necessary, to prepare the alarm viewer in the **LevelPopup.gdf** display.

5.5 Preparing Alarm Reports

The alarm report ActiveX control needs to be configured to allow remote clients to connect to the historical alarm database via Internet Information Services (IIS).

1. Open the **AlarmReport.gdf** display and double-click on the ActiveX control to open the **Properties** dialog.

2. Click the **Connection Parameters** button on the **General** tab, and then **Build Connection String** to edit they way you are connecting to the database. Up to this point, the connection has used NT Authentication, which will not use over the Web without adding permissions to the SQL Server database.

3. Connect to the **AlarmLogger** SQL database using the system administrative SQL login (sa with blank password), as shown in **Figure L5.7.**

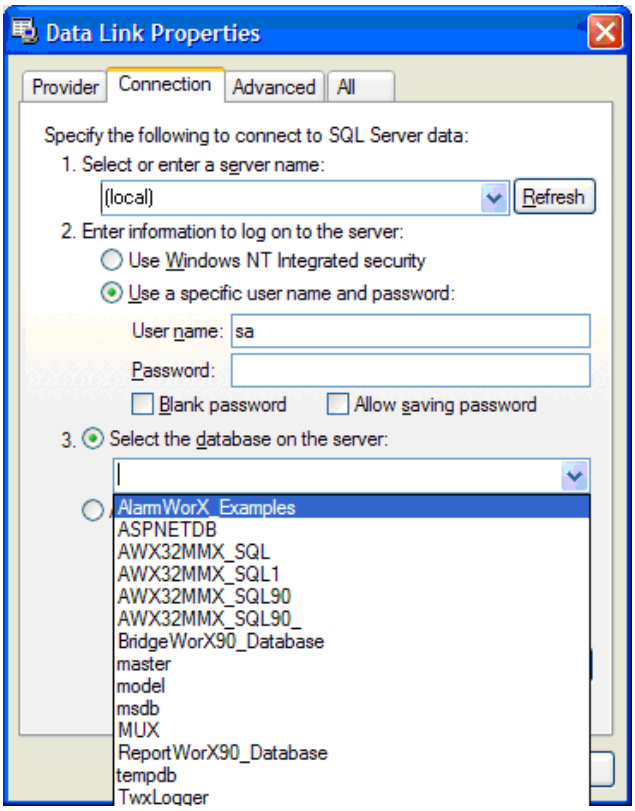

Figure L5.7: Network-Aware Authentication

4. Click **OK** and connect to the **AlarmLog** table in the database, as shown in **Figure L5.8.**

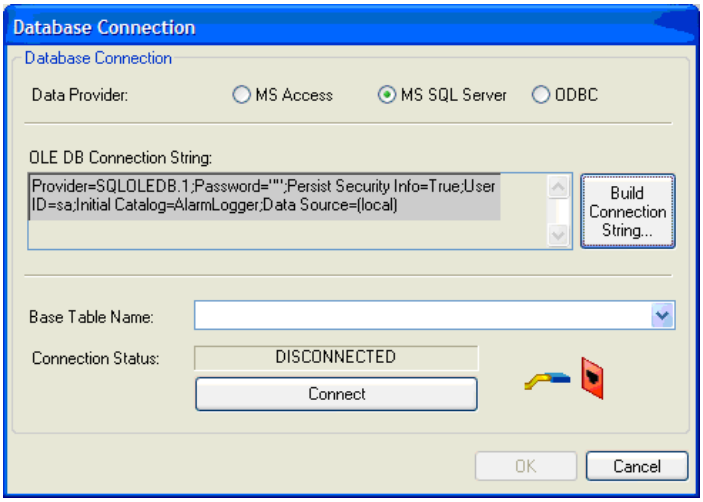

Figure L5.8: Database Connection

5. When the light turns green to indicate the connection is complete, click **OK** to return to the **Properties** dialog.

6. Select the **Web Access** tab, as shown in **Figure L5.9,** and check the **Access database over the Internet via Remote Database Agent.** Your IP address should automatically be filled in, which you can leave.

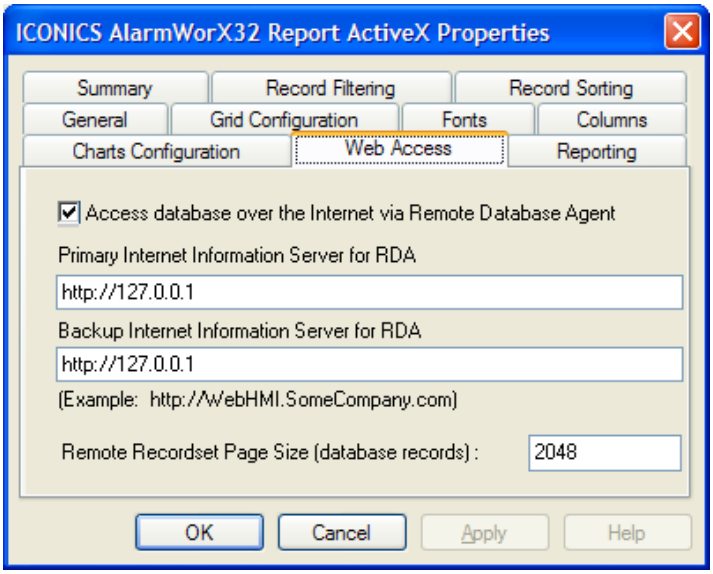

Figure L5.9: Enabling Web Access for the Alarm Report ActiveX

7. Click **OK** to save the settings, save the display, and close it.

Publishing GENESIS32 Displays 6

6.1 Objectives

In this chapter, you will learn how to make your process displays available to remote Web clients. By the end of this chapter you should be able to:

- Publish GraphWorX32 displays using the Web Publishing Wizard in GraphWorX32
- Understand how the Web Publishing Wizard can be used in ProjectWorX32
- Understand the different settings available in the Web Publishing Wizard

6.2 Publishing Overview

Once network-aware displays have been created, they need to be published to the Web Server so that they are available to your WebHMI clients. You can publish GraphWorX32 Container displays (*.gdf), AlarmWorX32 Container displays (*.a32), or TrendWorX32 Container displays (*.t32) using the same tool. Making a display available on an established WebHMI server involves the following basic steps:

1. Create the display file on a developer workstation with the container application (GraphWorX32, AlarmWorX32, or TrendWorX32) installed.

2. Use the Web Publishing Wizard to generate an HTML file and publish the file to the WebHMI Web server (URL) address.

3. On the Web server PC (with GENESIS32 and WebHMI installed), start GenBroker Server from GenTray.

4. On a client PC, open a Web browser, such as Microsoft Internet Explorer, and browse to the URL address of the HTML file. The client PC downloads the HTML file from the Web server. The Web server delivers all the necessary components (e.g. ActiveX controls) to the client PC's Web browser, which then runs the display directly within the browser. This way you can view the display in real time as a Web page.

NOTE: In this chapter, we will concentrate on the use of GraphWorX32 and the container, so we will discuss how to publish GraphWorX32 displays. Publishing the other types of displays is done in very similar ways.

Publishing of GraphWorX32 displays can be done from GraphWorX32 directly, or from within ProjectWorX32.

6.3 Publishing from GraphWorX32

To publish a GraphWorX32 file to a WebHMI server from within in GraphWorX32, you can run the **Web Publishing Wizard** from **Tools → Publish to HTML,** or click the button on the **Main** toolbar. The Web Publishing Wizard dialog is sown in **Figure 6.1.**

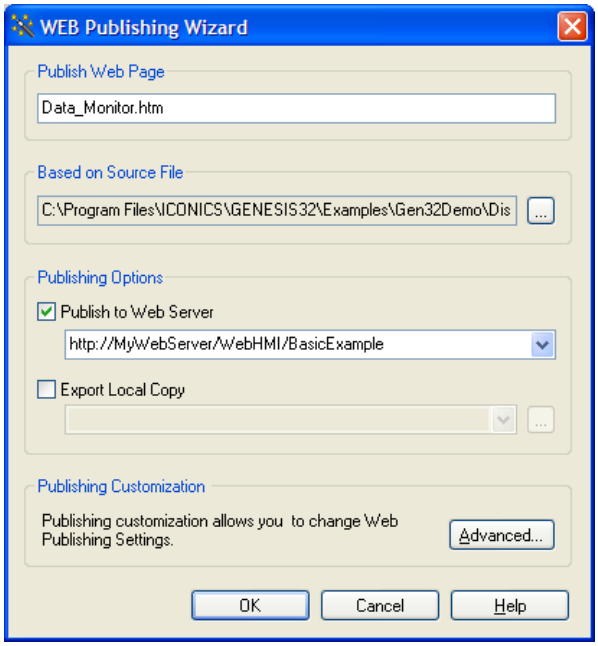

Figure 6.1: Web Publishing Wizard

6.3.1 General Settings

In the **Based on Source File** field of the **Web Publishing Wizard**, you must specify the name of the GraphWorX32 display (.gdf) file to be published. To choose a different display, click the button to browse for a file. Once a file is selected, the directory path and the file name are shown in the text field.

The **Publish Web Page** field of the **Web Publishing Wizard** specifies the name of the HTML file that will be created. The .gdf file name is used by default, but you can give the HTML file a different name as desired.

In the **Publishing Options** field, check the **Publish to Web Server** check box and type in the URL address of your Web server with the complete path indicating where you want to publish the HTML file to. In the example shown in **Figure 6.1,** the display is published to a folder called "BasicExample" inside the WebHMI directory on the Web server.

NOTE: Both the **Publish to Web Server** and **Export Local Copy** actions can be performed at the same time. For more information on exporting a local copy, see the **Web Publishing Wizard** section of the **GraphWorX32 Help File**.

The basic form of the web address to publish the display to is <http://Mywebserver/WebHMI/MyProject>, of which the various components are:

- **Mywebserver**: IP address (or server name) of the WebHMI Server
- **WebHMI**: Name of the WebHMI server root directory
- **MyProject**: Name of the directory within the WebHMI root directory on the server to which the HTML file will be saved (optional)

6.3.2 Advanced Settings

Click the **Advanced** button to view the **Web Publishing Wizard Properties** window shown in **Figure 6.2.**

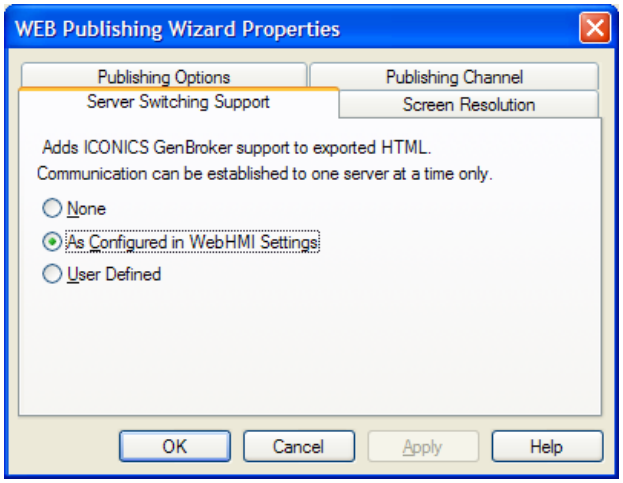

Figure 6.2: Server Switching Support Properties

The **Server Switching Support** tab allows the user to define the GenBroker client side configuration file (.gbx) to be used with this display. Leave **As Configured in WebHMI Settings** selected to use the **default.gbx** file.

The **Screen Resolution** tab, shown in **Figure 6.3**, determines the screen resolution and size of the GraphWorX32 Viewer ActiveX control, which is referenced in the

generated HTML file and then downloaded to a client PC when the HTML file is viewed in the client's Web browser. You can specify the screen size in the Width and Height fields in terms of pixels (px) or percentage (%). Click the **Resolutions** button to select from a pop-up menu of standard screen resolutions.

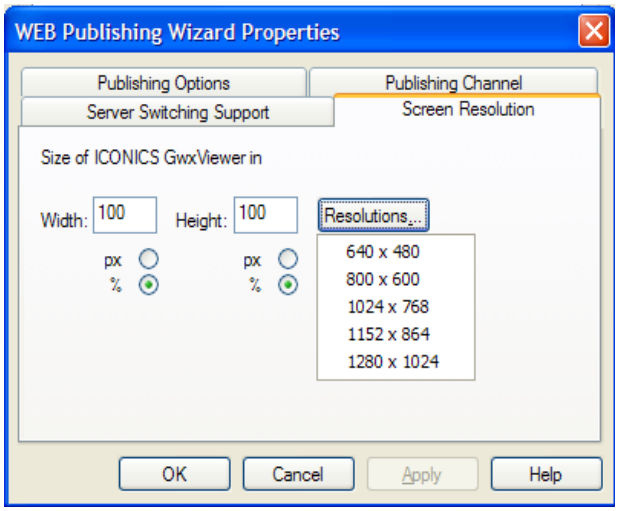

Figure 6.3: Screen Resolution Properties

The **Publishing Options** tab, shown in **Figure 6.4**, enhances the publishing process. By default, the Web Publishing Wizard provides **Multiple Display Support**. Suppose that you have a main GraphWorX32 display (.gdf) file that is linked to other display files through pick actions or Load Tabs. When you publish the main display to an HTML file, you want all the links and references to the other dependent display files to be functional when the HTML file is downloaded to a client Web browser. The Web Publishing Wizard makes this possible by detecting all mutually linked display files, looping through all dynamic actions, and checking for pick actions in which a file name is specified as one of its parameters.

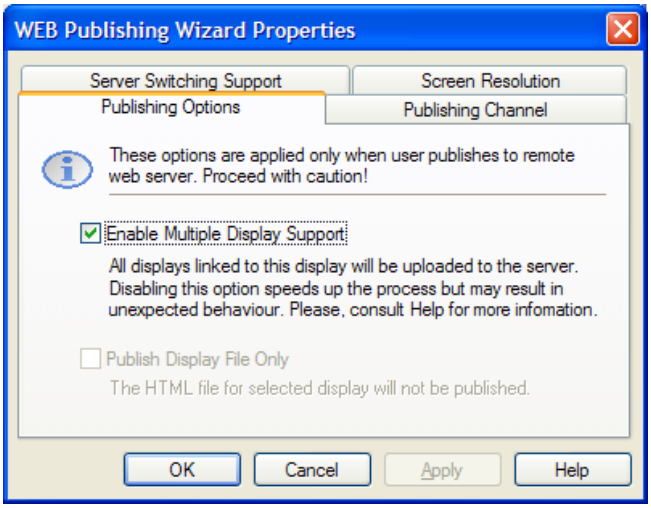

Figure 6.4: Publishing Options Properties

The multiple display detection could be a rather lengthy process. You can speed up the process by disabling the **Enable Multiple Display** Support option. This is especially useful, if you have already published your project files to the Web server, and now you want to update a display you have changed.

The multiple display detection mechanism ensures that the Web page will be published properly for the source file. Disabling the Multiple Display Support may cause publishing of an incomplete web page, which may result in unexpected behavior. Therefore, it is suggested to disallow publishing of the Web page (or HTML file) when disabling multiple display detection. You can do so by enabling the **Publish Display File Only** option.

The **Publishing Channel** tab allows you to choose between HTTP and FTP protocol for posting to your Web Server. Some security configurations and operating systems (Windows Server 2003) do not allow HTTP posting to the Web Server. For this reason, the Web Publishing Wizard supports posting via FTP. Additional setup on your Web Server may be necessary to use this feature.

NOTE: Refer to the WebHMI publishing Application Notes on the installation CD for more information on publishing via FTP.

The file analysis that occurs during publishing, shown in **Figure 6.5,** determines all the linked displays and ActiveX controls associated with the display being published (when multiple display detection is enabled), so that this information can be included in the HTML file.

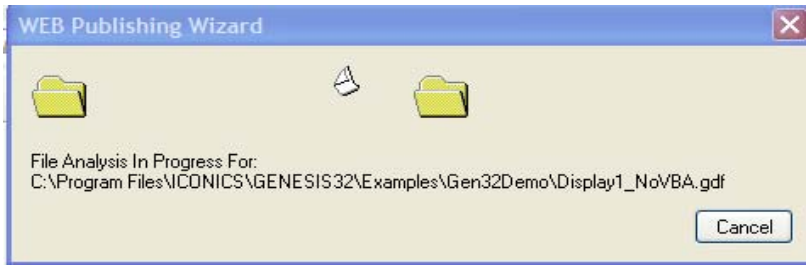

Figure 6.5: Publishing File Analysis in Progress

When complete, the new HTML file is uploaded to the Web server and then saved to the URL address path specified in the **Publish to Web Server** field. A copy of the .gdf file (GraphWorX32 display) that was published and all those linked to it (if multiple display support is enabled) are also sent to the specified directory, in order to be downloaded by the WebHMI client.

6.4 Publishing from ProjectWorX32

The Web Publishing Wizard can be accessed through the ProjectWorX32 console, allowing you to easily export/publish one or more GraphWorX32 display files within

a project to a Web server. Each project includes options to publish a single display or multiple displays in the project. This section of the document explains the different ways to publish displays using ProjectWorX32.

The **Applications/WebHMI** tree for each project in the ProjectWorX32 console, shown in **Figure 6.6**, manages Web publishing for the entire project. From here you can right-click and publish HTML files to a Web server. The WebHMI tree includes the following three categories:

- GraphWorX32-Based HTML
- Local HTML Files
- Published Documents

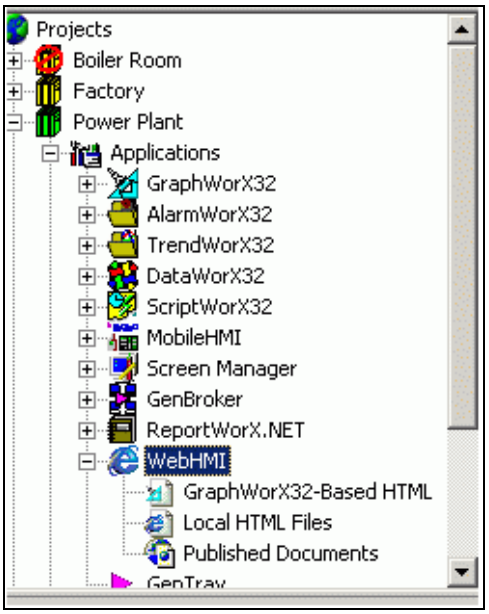

Figure 6.6: WebHMI Tree in ProjectWorX32

When publishing displays from ProjectWorX32, the same Web Publishing Wizard is used as in GraphWorX32 and the other ICONICS containers. The only difference is the means by which the wizard is opened, and what it does in each case.

You can open the Web Publishing Wizard dialog from the ProjectWorX32 console in the following ways:

- Opening the Publishing Wizard from the **Tools** menu
- Publishing all GraphWorX32 displays in a project
- Publishing a single GraphWorX32 display file
- Creating a new HTML file in a project

Opening the Publishing Wizard from the Tools Menu:

One way to launch the Publishing Wizard from ProjectWorX32 is to select the project root and then choose **Publish Wizard** from the **Tools** menu, as shown in **Figure 6.7**.

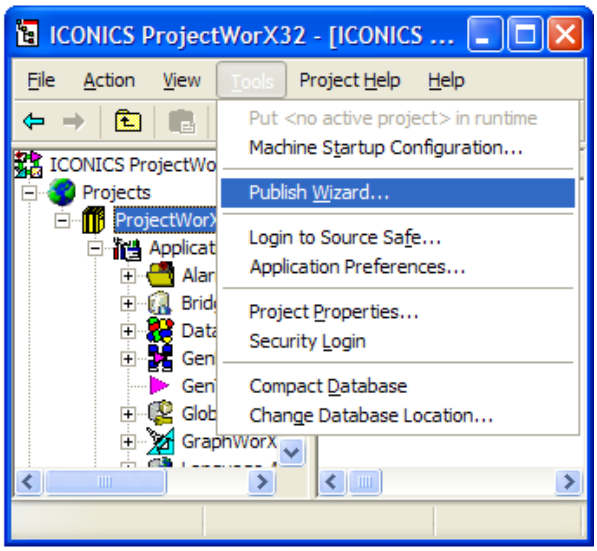

Figure 6.7: Opening the Publishing Wizard From the Tools Menu

In this case, both the **Publish Web Page** field and the **Based on Source File** field in the Web Publishing Wizard dialog are pre-configured and read-only. The button is also disabled, as shown in **Figure 6.8.**

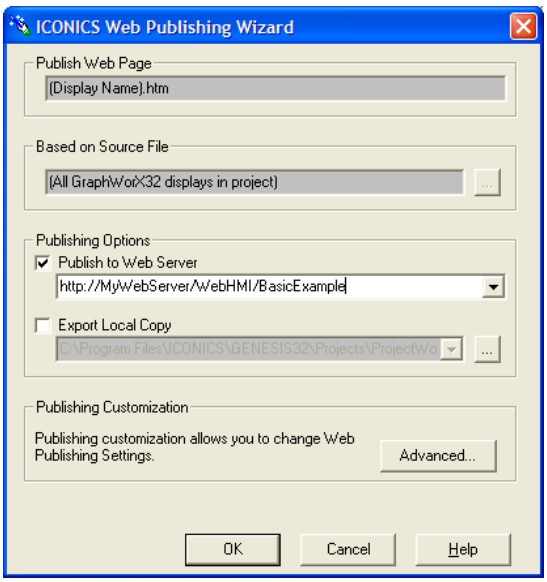

Figure 6.8: Web Publishing Wizard Opened from the Tools Menu

Publishing all GraphWorX32 Displays in a Project:

You can also launch the Publishing Wizard if you right-click on the **Applications/GraphWorX32/Displays** tree in ProjectWorX32, and select **Publish All Displays,** as shown in **Figure 6.9.**

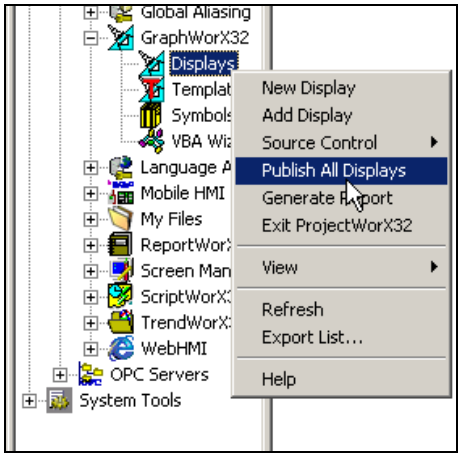

Figure 6.9: Publishing All GraphWorX32 Displays in a Project

In this case, the wizard looks exactly as it does when opened from the **Tools** menu, shown in **Figure 6.8.**

Publishing a Single GraphWorX32 Display File:

To publish displays individually, right-click on a GraphWorX32 display and select **Publish to HTML,** as shown in **Figure 6.10.**

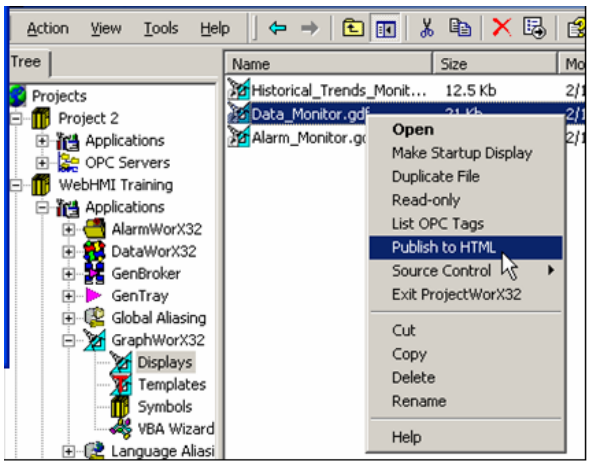

Figure 6.10: Publishing a Single GraphWorX32 Display File

In this case, the **Based on Source** field in the Web Publishing Wizard dialog is read-only and contains the name of the selected GraphWorX32 display. The button is also disabled, as shown in **Figure 6.11.**

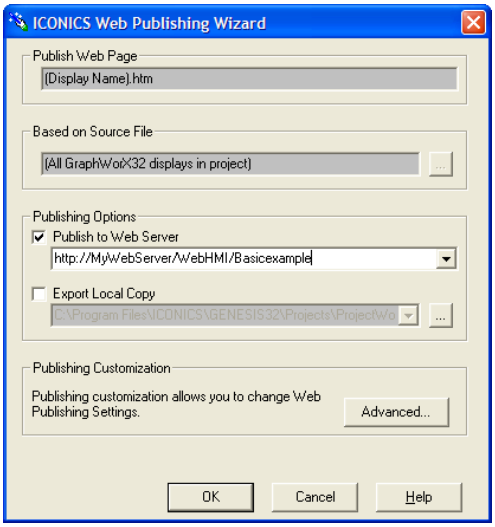

Figure 6.11: Web Publishing Wizard Publishing a Single Display

Creating a New HTML File in a Project:

The final way to publish a display from within ProjectWorX32 is to right-click on the **Applications/WebHMI/GraphWorX32-Based HTML** tree and select **New HTML Document**, as shown in **Figure 6.12.**

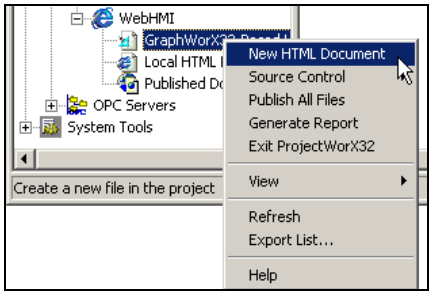

Figure 6.12: Creating a New HTML File in a Project

In this case, you need to specify a GraphWorX32 display file in the **Based on Source File** field by browsing for the file or by typing in the file name. You also need to specify the name of the new HTML file in the **Publish Web Page** field. The wizard in this case is shown in **Figure 6.13.**

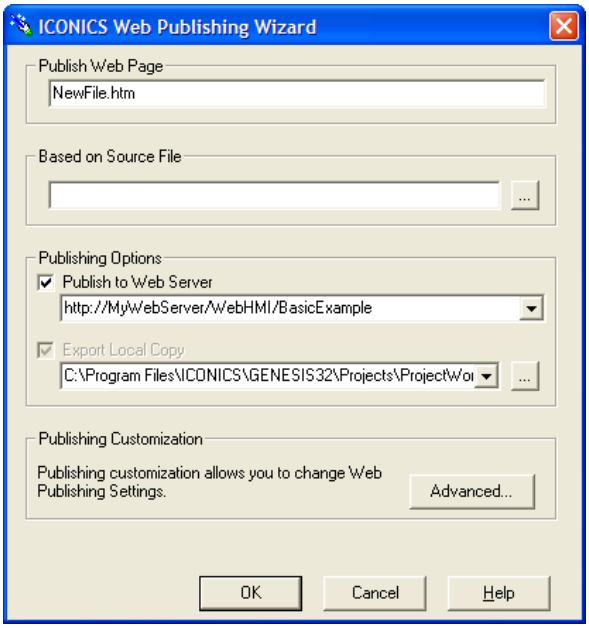

Figure 6.13: Web Publishing Wizard Creating a New HTML File

Quick Lab: Publishing GraphWorX32 Displays 6

6.1 Quick Lab Overview

In this lab, you will publish all the Acme GraphWorX32 displays to the WebHMI server using the Web Publishing Wizard built into GraphWorX32. By the end of this lab, you will have a copy of all the displays on the Web Server inside the WebHMI directory, along with the associated HTML file. This will satisfy Phase 1, Step 3 of the Project Guidelines given to us by Acme.

6.1.1 Task Checklist

In this quick lab, you will complete the following tasks:

- Publish all of the displays in the Acme water treatment GENESIS32 project
- Confirm that all displays and associated HTML files are posted to the Web Server

6.1.2 Support Files

To complete this quick lab, you will need the following support files:

- All GraphWorX32 displays outlined in Lab 1

6.2 Publishing the Displays in GraphWorX32

Use the following steps to publish the Acme displays (which you prepared in the last chapter) to the WebHMI directory on your Web server:

1. Open the **Operator3.gdf** file, and launch the **Web Publishing Wizard** using the button on the **Main** toolbar. The Web Publishing Wizard will look similar to **Figure L6.1.**

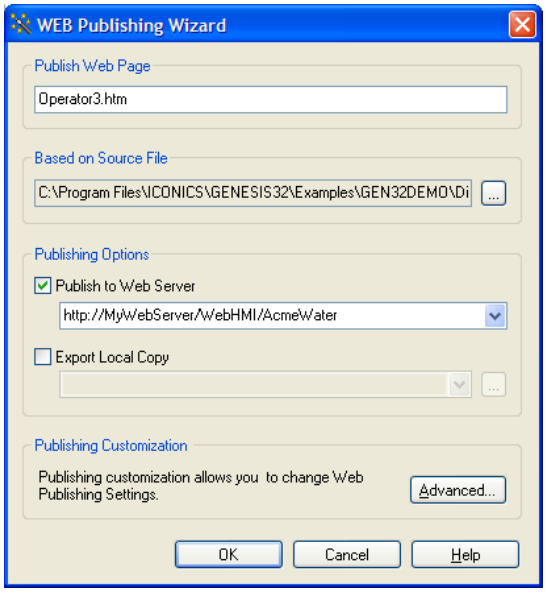

Figure L6.1: Web Publishing Wizard for Operator3.gdf

2. Make sure that the appropriate URL is in the **Publish to Web Server** field, as shown in **Figure L6.1.** This example shows that we are publishing the files to a sub-folder called **Acme Water** in the **WebHMI** directory on the Web server named **MyWebServer.**

   NOTE: You will replace **MyWebServer** in this URL with the actual name of your server machine.

3. Click the **Advanced** button and view the **Server Switching Support** tab, as shown in **Figure L6.2.** We will leave this set to **As Configured in WebHMI Settings.** This will create a reference in the resulting HTML file to the **Default.gbx** file as the GenBroker Configuration file to be used on the WebHMI clients.

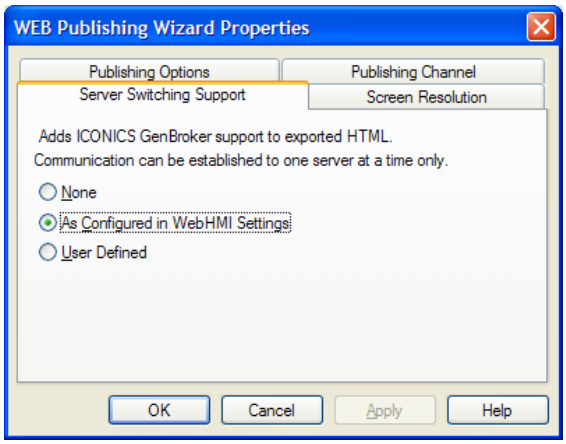

Figure L6.2: Server Switching Support

4. View the **Publishing Options** tab of the Advanced properties, as shown in **Figure L6.3.** We will leave the **Multiple Display Support** enabled, so that all of the Acme displays will be posted to the Web server, and analyzed together for the one HTML page to be created.

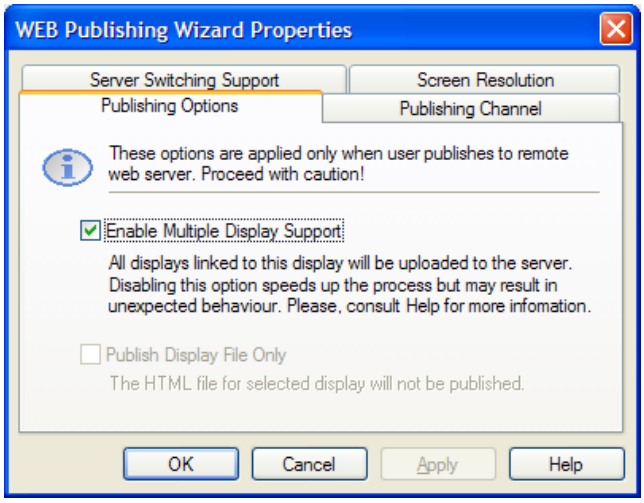

Figure L6.3: Publishing Options

5. Click **OK** to return to the **Web Publishing Wizard,** and then **OK** again to begin publishing the displays.

6. A dialog will open indicating that analysis is in progress for the files. When finished, you will be indicated that publishing was successful.

6.3 Viewing the Files That Were Published

View the files that were created during publishing, to make sure they were published properly:

1. Browse to the **C:\Program Files\Common Files\ICONICS\WebHMI\Acme Water** directory on your WebHMI server machine, which was created during the publishing. Because there is a virtual directory defined in IIS that points to this directory, it is a directory on the Web server.

2. The following files should now exist in this directory:

   - AlarmReporter.gdf
   - AlarmViewer.gdf
   - HistTrendViewer.gdf
   - LevelPopup.gdf

- Operator1.gdf
- Operator2.gdf
- Operator3.gdf
- Operator3.htm
- ValvePopup.gdf

3. The one HTML file is all that is needed to be able to view all of these displays over WebHMI.

Viewing the Project Remotely 7

7.1 Objectives

In this chapter, you will learn how to see completed WebHMI projects come to life remotely. After completing this chapter, you should be able to:

- View a completed WebHMI project from a remote Web client
- Be able to troubleshoot common problems in a completed WebHMI system

7.2 Viewing the Project

Now that you have prepared the Acme project for use over an Intranet, and published the displays to your Web server, it is time to try viewing your WebHMI displays. The same type of approach should be taken when testing your actual project as was taken in Chapter 4 when testing the Web server communications.

The client machine used for testing should have the following characteristics:

- On the same local area network as the WebHMI server (the Default.gbx file is set up for a client on the same network since the node name is used and not the IP address)
- Does NOT have GENESIS32 installed
- Has Microsoft Internet Explorer 5.5 or greater installed

On the client machine, point the Web browser to the URL the displays were published to. The structure of the URL is typically **<http://ServerName/webhmi/subfolder/filename.htm>.** In this URL, **ServerName** should be the name or IP address of the Web server where WebHMI resides, and **subfolder** (optional) should be any folder or directory structure inside the WebHMI directory on the Web server in which **filename.htm** resides.

Make sure to thoroughly test each part of the project. View each page with ActiveX components on them to be sure that they download properly. Also make sure you view all types of data included in your project (real-time DA, historical DA, A&E, etc.) to be sure the data is coming across the network properly. It is also a good idea to view each display once to be sure that the client can download every display from the server without any problems.

Quick Lab: Viewing the Project Remotely 7

7.1 Quick Lab Overview

You have finished configuring the Web server and WebHMI project for Acme. It is time to complete the last step of Phase 1 by testing and troubleshooting the final product. Use the table of common problems in the content of this chapter to help in troubleshooting.

7.1.1 Task Checklist

In this quick lab, you will complete the following tasks:

- View the Acme displays on a remote WebHMI client
- Troubleshoot the system if necessary

7.1.2 Support Files

To complete this quick lab, you will need the following support files:

- All GraphWorX32 displays outlined in Lab 1
- Default.gbx
- All server configuration files outlined in Lab 1

7.2 Viewing the WebHMI Displays

Using the same WebHMI client as used during testing of the WebHMI server in Chapter 4 (or one with the same characteristics) go through the following steps to view your new Acme WebHMI application.

1. Make sure that GenBroker is running on the WebHMI server. If it is not, start it from GenTray.

2. On the client, point the browser to the Web address for the HTML page that the main display was published to:

<<http://MyWebServer/webhmi/Acme Water/Operator3.htm>>

NOTE: Replace **MyWebServer** in this path with the actual name of your WebHMI server rmachine.

3. Just as when you viewed the sample displays during the testing of the WebHMI server, you should be prompted to download any ActiveX components or other objects that are not already on the PC. If you used this same PC to test the WebHMI server, and viewed all of the ActiveX pages, you should not be prompted unless the temporary Internet files and objects on the client have been cleared since.

4. When all necessary objects are downloaded, the first display **Operator3.gdf** should be loaded into the Web browser, as shown in **Figure L7.1.**

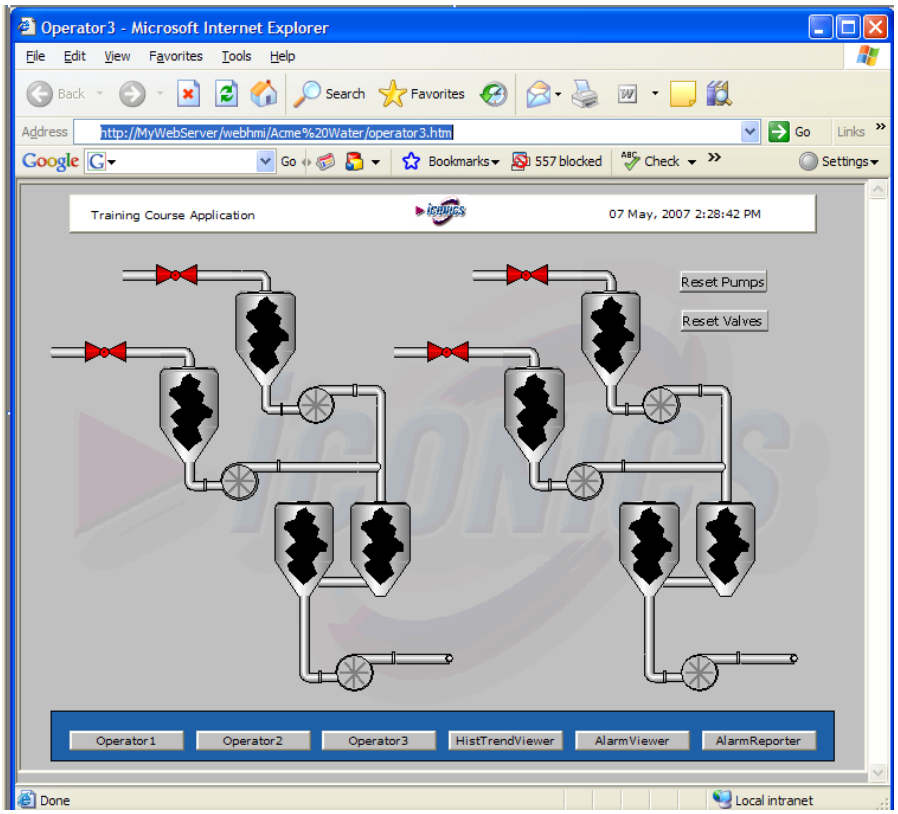

Figure L7.1: Operator3 on the WebHMI Client

5. Test the display (fill and empty the tanks) to be sure that all data items are updating, and you can interact with them and the popup windows properly.

6. Now switch to **HistTrendViewer.gdf** by clicking on the appropriate button. The Web browser should look like **Figure L7.2.**

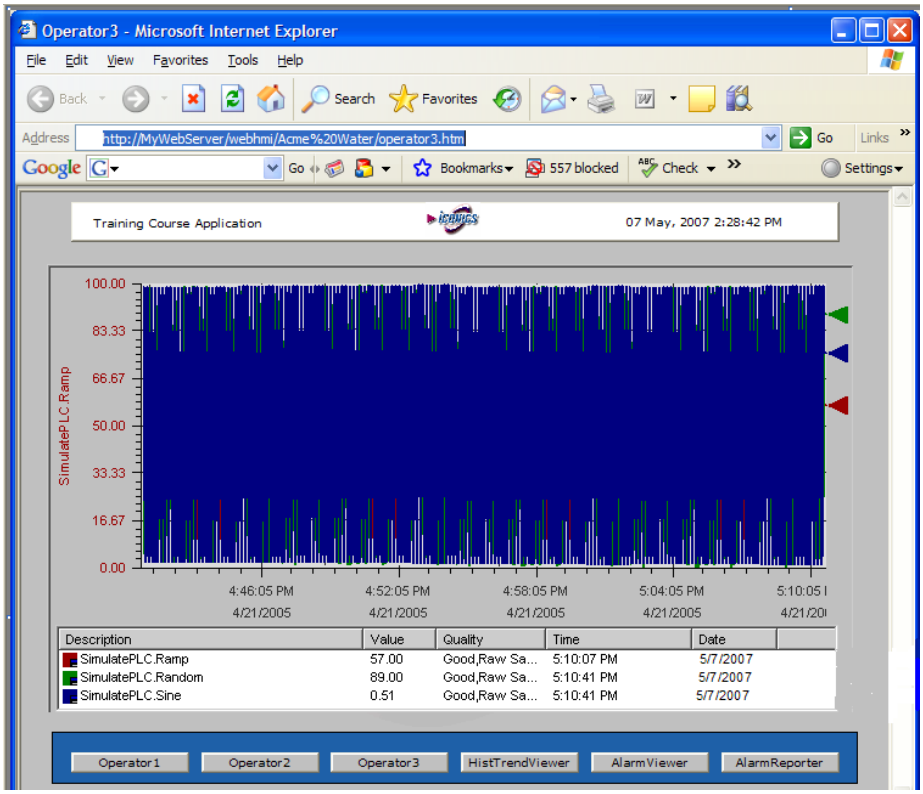

Figure L7.2: HistTrendViewer.gdf on the WebHMI Client

NOTE: The URL for both displays is the same, because a new web page is not being loaded. The Load Display pick action loads a new display within the GWXView32 ActiveX control that is running within the Web browser. If you were to press **REFRESH** or **BACK** on the Web browser itself, these display loads will not be remembered.

7. Double click on the trend viewer, and notice that the **Runtime Toolbar** opens and is usable just as in runtime locally on the server machine. Experiment with these options, to see that you can browse back into the historical data, change pen and view properties, view statistics (in freeze mode), etc.

8. Finally, switch to **AlarmViewer.gdf** by clicking the appropriate button. The Web browser should look like **Figure L7.3.**

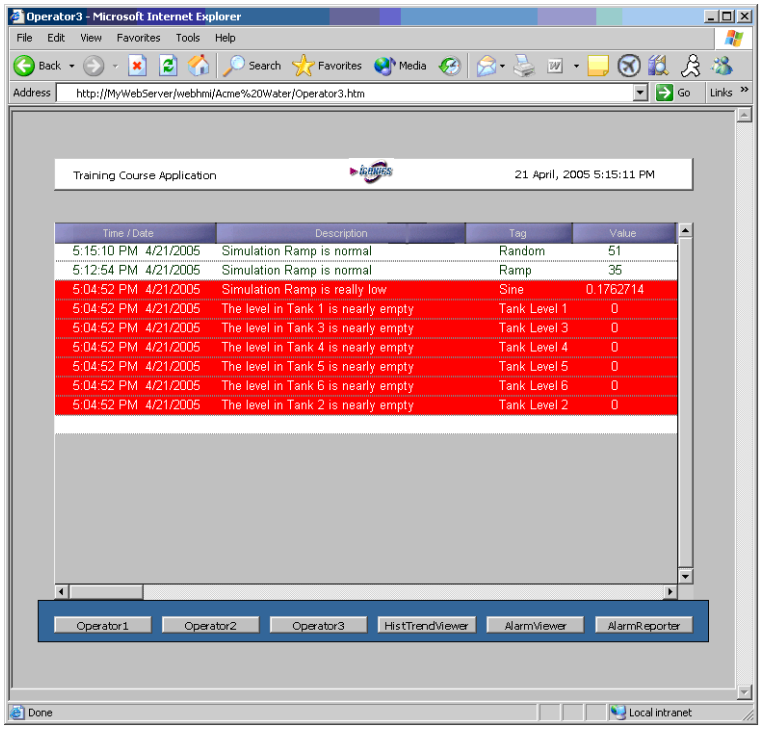

Figure L7.3: AlarmViewer.gdf on the WebHMI Client

9. Just as before, the URL has not changed, but the new display is loaded. Just as with the trend viewer, the runtime options are available for the alarm ActiveX control as well.

10. Double click on an alarm in the viewer to acknowledge the alarm, or right-click and select **Ack** for all the acknowledgment options. Notice that the alarm is acknowledged just as if it was acknowledged on the server. You can also sort, filter, etc. WebHMI truly is OPC interactive!

11. Click on the **AlarmReporter** button to view your historical alarms display. It should look similar to **Figure L7.4.**

Visualize Your Enterprise™

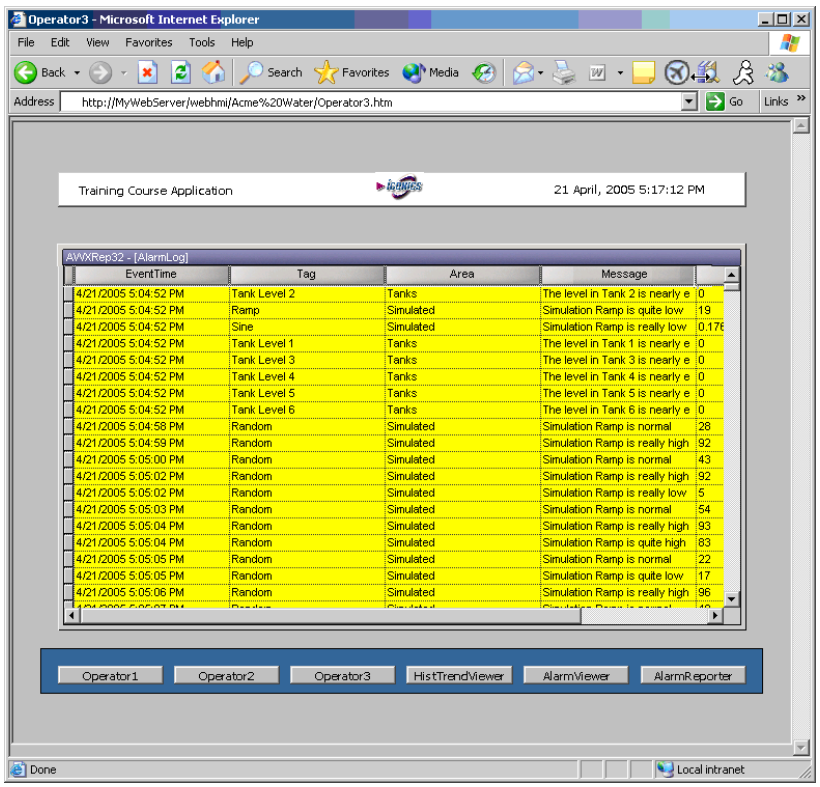

Figure L7.4: AlarmReporter.gdf on the WebHMI Client

12. Double click on the historical alarm report to open the **Runtime Toolbar**. The only runtime options that are not available for the alarm report over WebHMI (that are available on the server) are the **Crystal Report** options. This is because at least the runtime portion of Crystal Reports needs to be installed on the machine to use these options. If Crystal Reports is installed on the client machine, these options can be made available.

13. You can still sort the alarms in the report, view charts or the summary, filter the entries, etc. just as you can on the server.

14. These GraphWorX32 displays run virtually the same on the WebHMI client machine as they would on the server. The only difference is that they are embedded within a Web browser, and GENESIS32 does not need to be installed on the machine!

Practice Lab: Advanced Networking Example 8

8.1 Quick Lab Overview

Throughout this course, you have gone through the design steps of a basic WebHMI application example. There are more advanced configuration options available, which allow WebHMI to work with more complicated and secure networking setups. For instance, OPC data is not always on the same machine as the WebHMI server, and often firewalls or routers are used between the WebHMI server and the Internet for security reasons. The example in this chapter will show you the changes in the WebHMI design necessary to accommodate for these advanced configurations.

This lab assumes the network configuration shown in **Figure L8.1.**

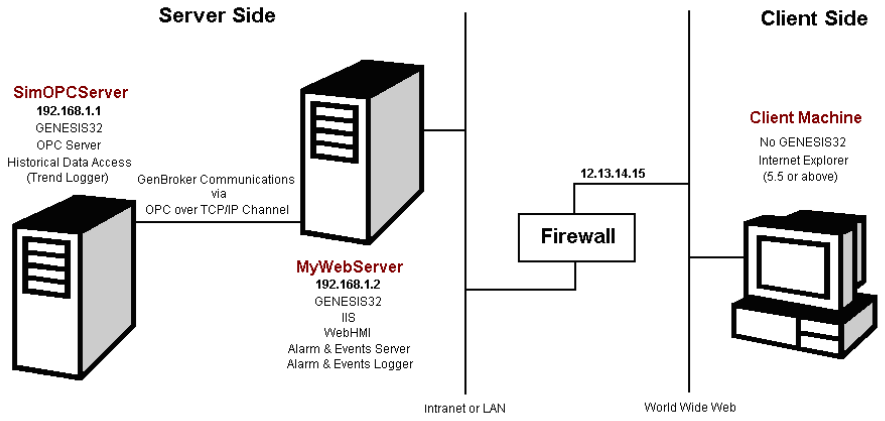

Figure L8.1: Network Setup for Advanced WebHMI Application Example

The OPC DA server (ICONICS DataWorX32 for the Acme example) should reside and run on the **Simopcserver** machine, as well as the Historical DA server (TrendWorX32 Logger). All A&E (AlarmWorX32) servers and loggers should reside and run on the **Mywebserver** machine, acting as the WebHMI server.

This network setup has a firewall in between the World Wide Web and the servers. The firewall has its own IP address, which becomes the external IP address. Any data

requested by the WebHMI client will need to be requested through the firewall's external address. The firewall re-routes all requests to the WebHMI server's internal IP address.

8.1.1 Task Checklist

In this quick lab, you will complete the following tasks:

- Configure GenBroker communications for the network setup shown in **Figure L8.1**
- Test and troubleshoot the Web server
- Adjust existing GraphWorX32 displays to work over the advanced network
- Re-publish the GraphWorX32 displays
- View the reconfigured displays on Web browsers across the advanced network

8.1.2 Support Files

To complete this quick lab, you will need the following support files:

- All server configuration files outlined in Lab 1
- All GraphWorX32 displays outlined in Lab 1

8.2 GenBroker Configuration

Recall from Chapter 3 that there are two types of GenBroker configuration: one for OPC communications between GENESIS32 systems, and one for communications the WebHMI server and Web client. For systems where all the OPC data is local to the WebHMI server, no special configuration is needed on the server side. Recall that this was the case in the basic WebHMI example created throughout this class.

This advanced WebHMI example application will consider the case where the alarm server and alarm history logger are both on the WebHMI server. The OPC data access server (i.e. DataWorX32) and the historical data access server are located on a separate GENESIS32 machine. This machine will communicate data to/from the WebHMI server machine via GenBroker communications.

8.2.1 GenBroker Configuration for the WebHMI Server

The node name of the GENESIS32/WebHMI machine in this example will be **Simopcserver** with IP address **192.168.1.1**. The Alarm & Event servers and loggers are running on the WebHMI server itself.

Set-up of the Machine Receiving the OPC Data (Mywebserver):

1. The WebHMI server machine **Mywebserver** will be acting as the OPC client machine in this case. Open the **GenBroker Configurator** on the WebHMI server machine from **Start → Programs → ICONICS Tools**.

2. Since this computer is acting as a client to the **Simopcserver** machine, we will create a new configuration (.gbx) file by clicking **Edit,** as shown in **Figure L8.2.**

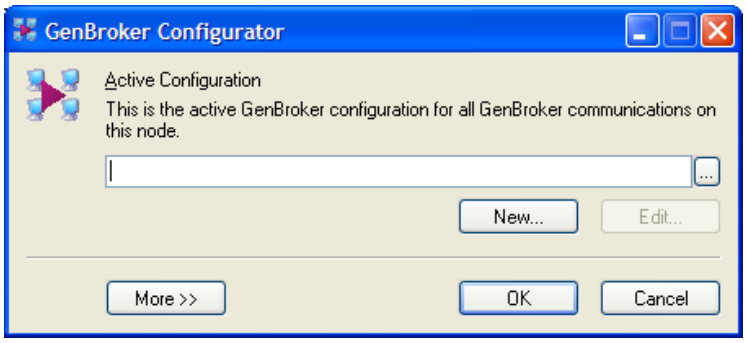

Figure L8.2: Client Tab Configurations on Mywebserver

3. A new configuration file (.gbx) will open. Click on **Channels,** as shown in **Figure L8.3.**

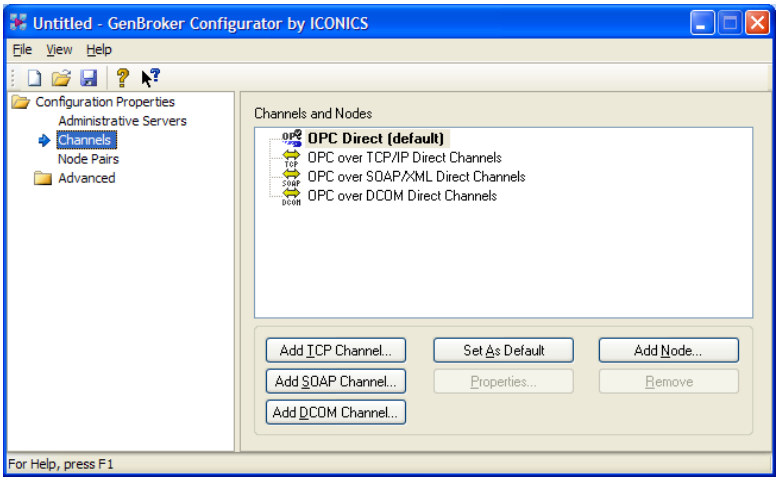

Figure L8.3: New GenBroker Configuration File

4. Configure this file so that all communications between this client and the **Simopcserver** machine will be on the **OPC over TCP/IP Direct Channel**. Do this by right-clicking on the channel and selecting **Add Node** as shown in **Figure L8.4.**

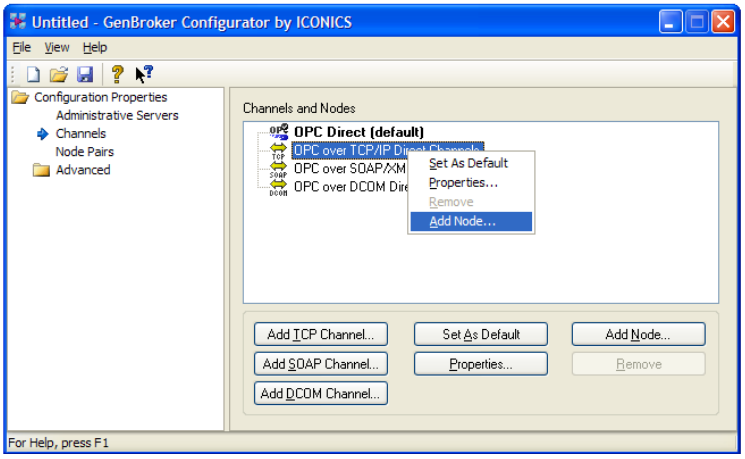

Figure L8.4: Adding a Node to a Channel

5. In the **Select a Node** box, enter the IP address of **Simopcserver** (in this example **192.168.1.1**) as shown in **Figure L8.5.** Click **Ok** when finished.

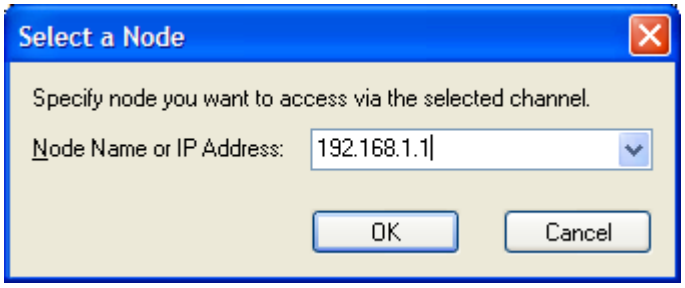

Figure L8.5: Select a Node Dialog

6. Click the plus sign next to the **OPC over TCP/IP Direct Channel** to expand it, and you should see the node listed underneath it as shown in **Figure L8.6.**

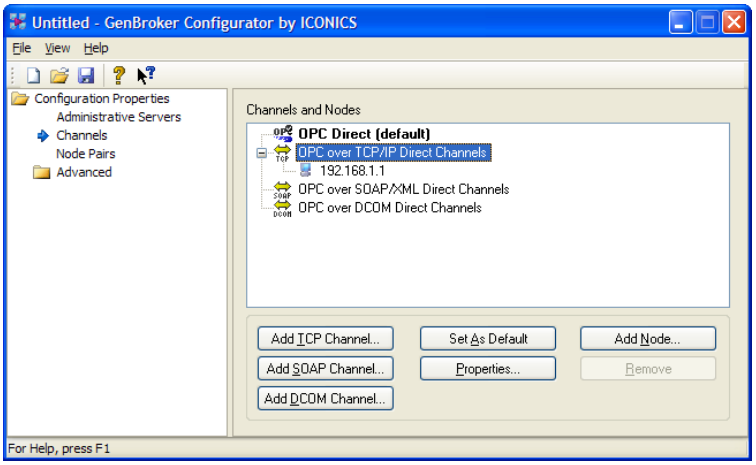

Figure L8.6: GenBroker Configuration for the Client (Mywebserver)

7. With this configuration, any time **Mywebserver** makes a request for data from **Simopcserver**, the **OPC over TCP/IP Direct Channels** will be used. In all other cases, **OPC Direct** (DCOM) will be used because it is default.

8. Highlight **Administrative Servers** on the left-hand pane to view the remote server settings on the right. The license, security, etc. are all local on this machine, so all remote servers should be left as local, as shown in **Figure L8.7.**

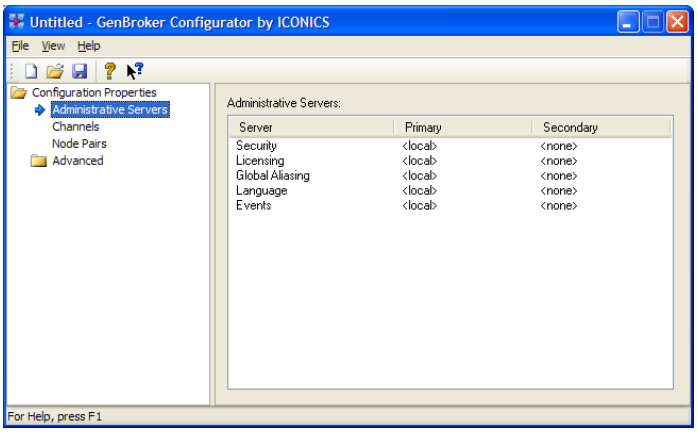

Figure L8.7: Remote Servers Pointing to the Local Node

9. Save and close the configuration, choosing a name and location appropriate to your project and the configuration.

NOTE: It is important to remember that whatever is configured in this GenBroker Configurator has nothing to do with WebHMI, and is necessary for remote communications via GenBroker even when WebHMI is not being implemented. These communication settings define the protocols used during communication between OPC, DA, A&E, and HDA servers and OPC clients such as GraphWorX32.

8.2.2 WebHMI Client Configuration

Configuring GenBroker Communications for a WebHMI client consists of creating a **.gbx** file and defining the protocols to be used in communication between the Web client and the WebHMI server. This file is referred to when publishing your displays to your Web server and downloaded by the WebHMI client before viewing displays.

For this advanced application example, the **default.gbx** configuration file will not be used. Recall that this example includes a firewall in the configuration, and OPC data is coming from more than one computer. This more advanced configuration will require a more advanced GenBroker configuration (.gbx) file for the WebHMI client.

1. As mentioned in Chapter 3, the **Client Configuration** tool is a means of creating and editing .gbx files. Open this tool from **Start → Programs → ICONICS**

WebHMI → Client Configuration. The Configurator will look like **Figure L8.8.**

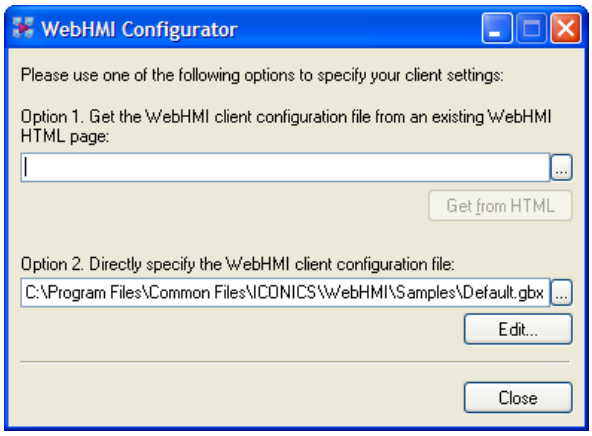

Figure L8.8: GenBroker Client Configuration Tool

2. Click **Edit** to create a new configuration file, shown in **Figure L8.9.**

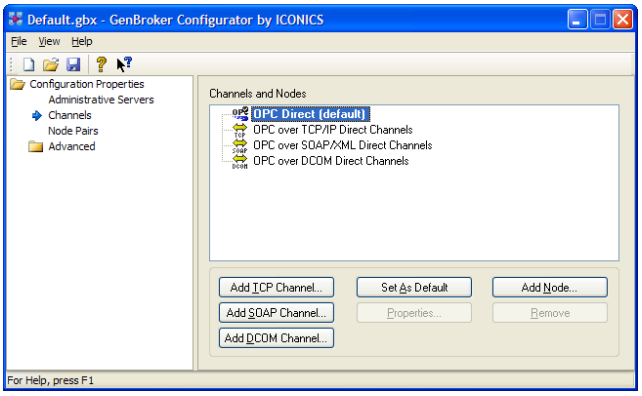

Figure L8.9: New GenBroker WebHMI Client Configuration (.gbx) File

3. Because the WebHMI client will need to request all data through the firewall, we will create an **Indirect Channel** with the firewall external IP address as the **Mediator Node**.

4. Click the **Add TCP Channel** button in the **GenBroker Configurator** to create an **OPC over TCP/IP Channel**. You will be asked to define a **Mediator Node**, as shown in **Figure L8.10.** Enter the external IP address of the firewall, in this example **12.13.14.15**.

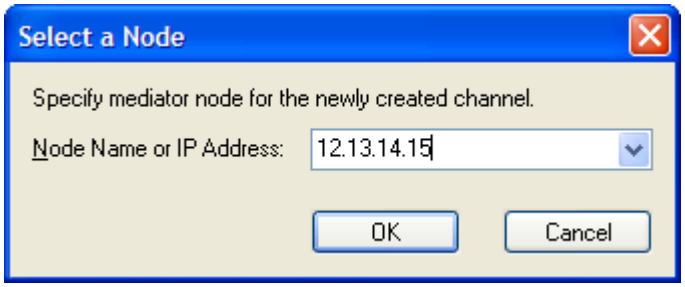

Figure L8.10: Selecting a Mediator Node for an Indirect Channel

5. Click **Ok** to return to the **GenBroker Configurator**, which should now look like **Figure L8.11.**

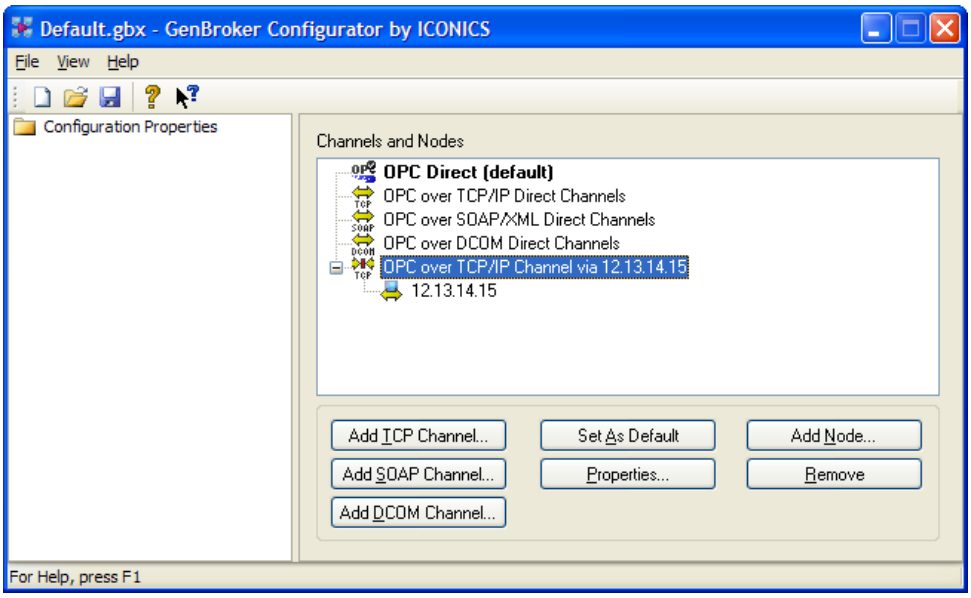

Figure L8.11: Indirect Channel Added

6. If you expand the plus sign of the channel, you will see that **12.13.14.15** is automatically added to the channel.

7. Recall that data requested from any nodes listed under an indirect channel will be requested through the mediator node. Since our two GENESIS32 machines are both on the inside of the firewall, we will need to request data from them through the external firewall IP address.

8. Add the two IP addresses of our two computers **Simopcserver (192.168.1.1)** and **Mywebserver (192.168.1.2)** to the **OPC over TCP/IP Channel via 12.13.14.15** by right clicking on the channel and selecting **Add Node**. This is shown in **Figure L8.12.**

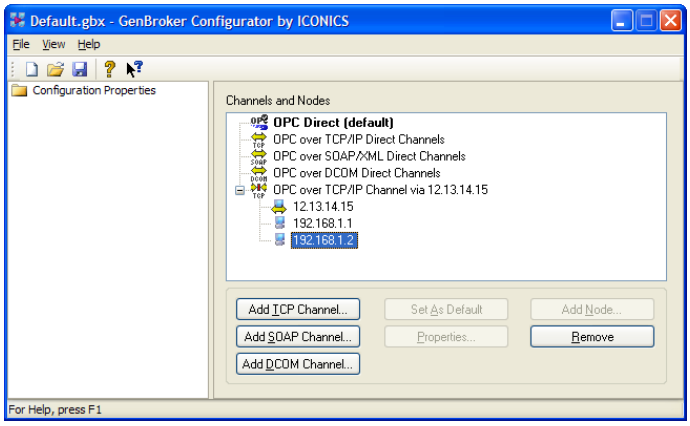

Figure L8.12: Adding a Node to an Indirect Channel

9. Finally, click on **Administrative Servers** and set Primary to **Mywebserver**, which has IP address **192.168.1.2** as shown in **Figure L8.13.**

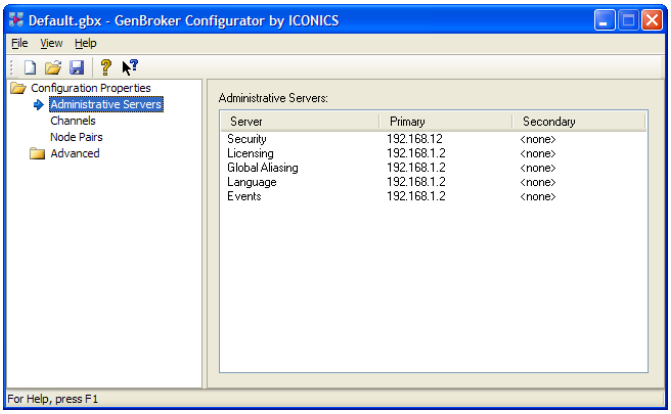

Figure L8.13: Setting the Primary Servers

10. When complete, save the file and close. Save the file as **AdvancedCFG.gbx** in the **C:\Program Files\Common Files\ICONICS\WebHMI\AdvExample** directory (a new folder in the WebHMI directory).

NOTE: Remember that opening or creating a .gbx file using the Client Configuration tool in no way chooses that file as the file to be used by the WebHMI clients. This will be set during publication of a display.

Visualize Your Enterprise™

8.2.3 Running the GenBroker Server

Once GenBroker Communications are configured for both the WebHMI server and clients, the GenBroker Server needs to be started. This needs to be **running on the server** in a server-client network. This means that GenBroker needs to be running on **Simopcserver** for **Mywebserver** to get data from it (since Mywebserver is acting as an OPC client to Simopcserver). GenBroker also needs to be running on **Mywebserver** for the WebHMI server and clients to communicate.

1. The best way to start GenBroker is through GenTray (or a GenTray configuration in ProjectWorX32). Open and start GenTray from **Start → Programs → ICONICS Tools → GenTray.**

2. When GenTray starts up, you will see the GenTray icon in your system tray, as shown in **Figure L8.14.**

Figure L8.14: GenTray Shown in the System Tray

3. To start GenBroker from GenTray, click on the icon and select **GenBroker → Start**, as shown in **Figure L8.15.**

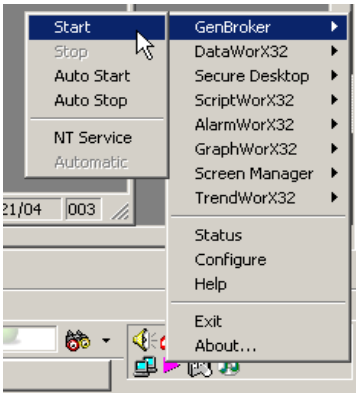

Figure L8.15: Starting GenBroker from GenTray

4. Follow these steps to start GenBroker on both the **Mywebserver** and **Simopcserver** machines.

8.3 Testing the Web Server

Remember that proper development of any project should include incremental testing along the way. Before moving on to creating and viewing your WebHMI display the Web Server should be tested to be sure that the communication is configured and operating correctly.

We will use the WebHMI samples that come with the installation to test the Web server, just as we did in the basic example. However, because the sample HTML files refer to **default.gbx** we will have to edit them slightly to use our new GenBroker configuration file **AdvancedCFG.gbx** for them to work on our more advanced network setup.

Changing the Host Name in the Sample HTML Files:

1. Since we developed the GenBroker configuration files with IP addresses for use over the World Wide Web, we first need to change the host name used in the sample HTML pages we will be testing with from **Mywebserver** to **192.168.1.2** (the IP address of Mywebserver).

2. Open the **Change Host Name Utility** from **Start → Programs → ICONICS WebHMI**. The utility will look like **Figure L8.16.**

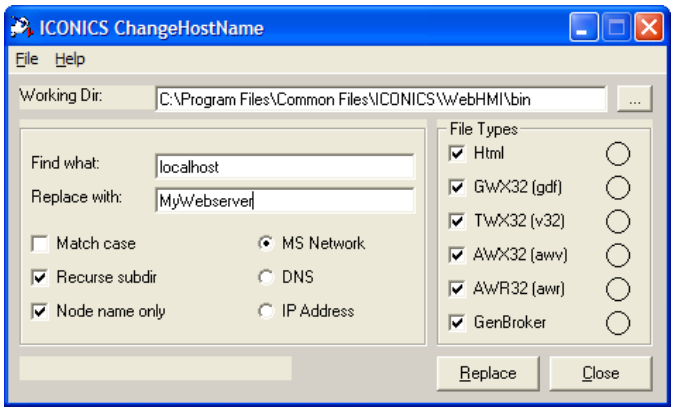

Figure L8.16: Change Host Name Utility

3. We want to find **Mywebserver** and replace it with **192.168.1.2** in **HTML Files** in the **C:\Program Files\Common Files\ICONICS\WebHMI\Samples** directory.

4. Clicking the **IP Address** option automatically enters the local machine's IP address into the **Replace with** text box. We do not need to recurse subdirectories, so that option can be unchecked.

5. When finished configuring the **Change Host Name Utility**, it should look like **Figure L8.17.**

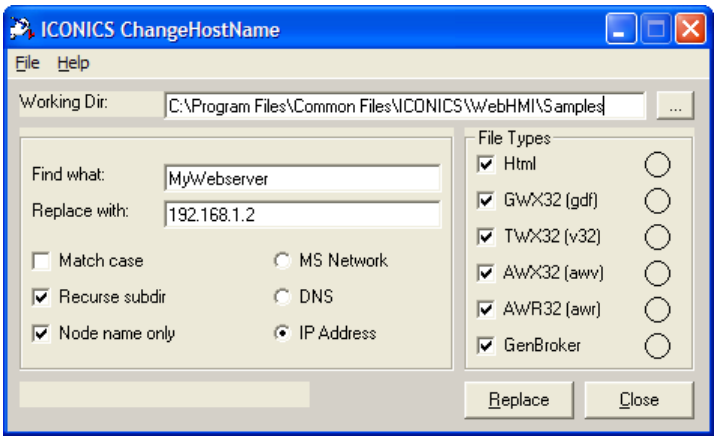

Figure L8.17: Changing Node Name to IP Address

6. Click **Replace** to begin the process. When finished, click **Close**.

<u>Editing the Sample HTML Files:</u>

7. Browse to the **C:\Program Files\Common Files\ICONICS\WebHMI\Samples** directory and find the **Index.htm** file. Recall that this is the main page of the sample WebHMI displays.

8. Right-click on the file and select **Open With → Notepad** to edit the HTML.

9. In **Notepad**, key **CTRL+F** to open the **Find** dialog. Type in **gbx** to be found and click **Find Next**.

10. The cursor should be placed on the following line of code:

```
'<PARAM NAME="CfgName" VALUE="http://192.168.1.2/WebHMI/Samples/Default.gbx"> ' +
```

This is the line of code that points the client machine to the correct GenBroker configuration file (.gbx) to download.

11. Edit this line of code so that the path points to the **AdvancedCFG.gbx** configuration file that was just created. The path should read:

http://192.168.1.2/WebHMI/Samples/AdvExample/AdvancedCFG.gbx

12. With the **Find** dialog still up, click **Find Next** to find the second place the configuration file is referred to. Edit this line of code in the same way.

13. Save the file over the original, and close it.

14. Edit each of the following sample HTML files in the same way, as we will use them to test just the three main ActiveX sample pages. They are all found in the **C:\Program Files\Common Files\ICONICS\WebHMI\Samples** directory.

- alarmviewer.htm
- alarmreport.htm
- timeplot.htm

NOTE: Republishing these files with the new .gbx file chosen will accomplish the same thing, but rather than going through the publishing steps for each of the sample displays associated with the HTML pages, we can edit the single line of code in these files.

Viewing the Edited Sample Pages via WebHMI:

15. Find a machine with the following characteristics as the test client:

- Is either on the same local area network as the WebHMI server, or somewhere else across the World Wide Web (you may want to test it for both cases)
- Does NOT have GENESIS32 installed
- Has Microsoft Internet Explorer 5.5 or greater installed

16. Open Internet Explorer on the client and point the browser to <http://ServerName/WebHMI>. In this advanced example, the server path is the IP address of the WebHMI server machine **192.168.1.2**. The Web address would be **<http://192.168.1.2/WebHMI>**.

17. Recall from the basic example that you should automatically be redirected to the **Index.htm** page, and you will be prompted to download any ActiveX components necessary and the .gbx file.

NOTE: If this client machine has been used as a WebHMI client in the past, you should clear all temporary Internet files and objects using the **Tools → Internet Options** menu of Internet Explorer.

18. After downloading the necessary objects you should see the main display (or index page) for the WebHMI samples, as shown in **Figure L8.18.** Notice that the browser was automatically redirected to **<http://192.168.1.2/WebHMI/Samples/Index.htm>**.

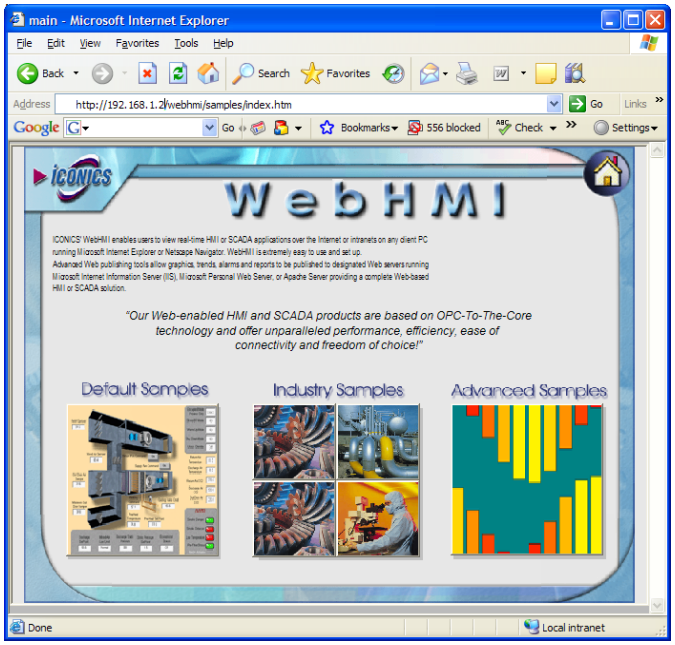

Figure L8.18: Main WebHMI Sample Page

19. Click the **Default Samples** link so that we can test OPC communication and the major ActiveX functionality. The **Default Samples** display will open, as shown in **Figure L8.19.**

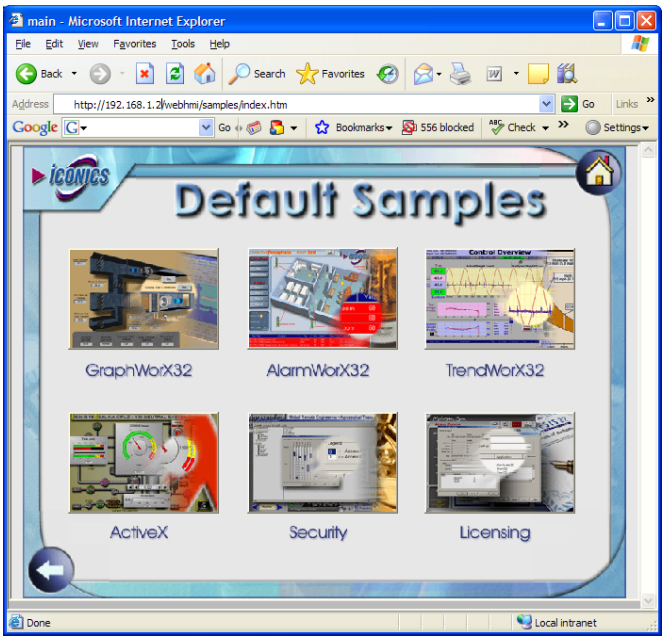

Figure L8.19: WebHMI Default Samples Page

20. To test the OPC communication between the client and the server, click on the **GraphWorX32** link to see the display shown in **Figure L8.20.** You should not

be prompted to download any ActiveX controls for this display because it only uses the GraphWorX32 Viewer ActiveX control, which has already been downloaded for the previous displays.

21. You should see OPC data updating on the display and animation based on that data.

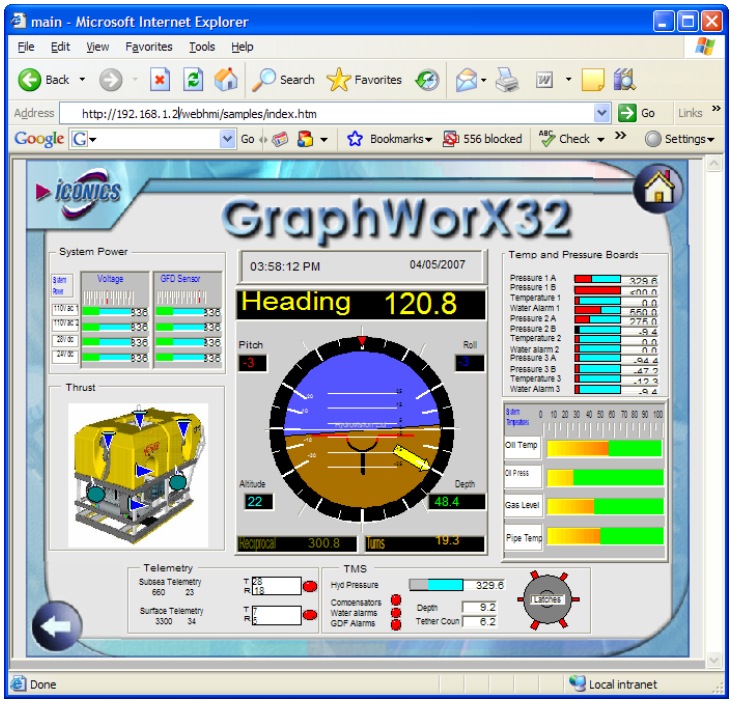

Figure L8.20: WebHMI GraphWorX32 Sample Page

22. Hit the back arrow button at the bottom left-hand corner of the screen to return to the **Default Samples** page. This time click the **AlarmWorX32** link to view the display shown in **Figure L8.21.**

<u>NOTE:</u> If you click the **Back** button of the Web Browser page, you will be returned to the index page, because from that point new html pages have not been loaded, but simply "load display" pick actions have been executed within the GraphWorX32 Viewer ActiveX control.

Visualize Your Enterprise™

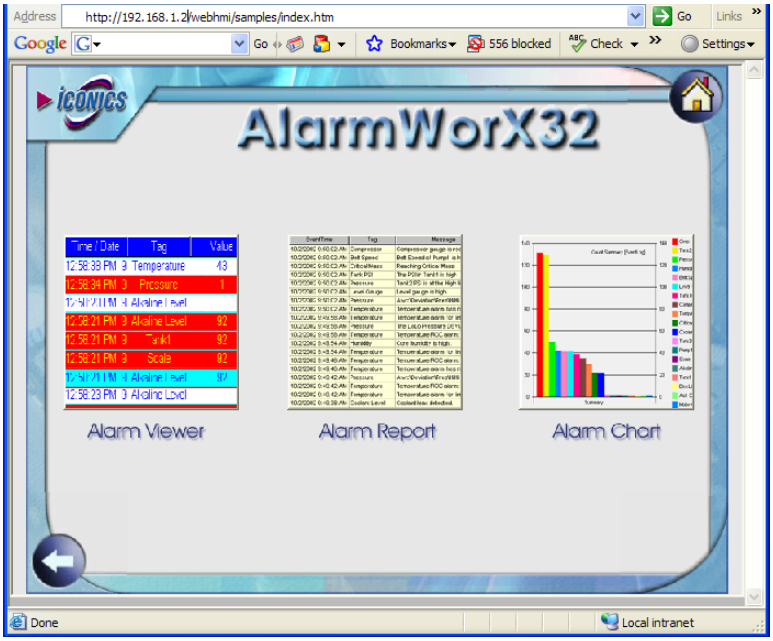

Figure L8.21: WebHMI AlarmWorX32 Sample Page

23. Click on the **Alarm Viewer** link to open a display with an AWXView32 ActiveX control, which will connect to live alarm and event data on the WebHMI server. At this point, the second HTML page you edited is being loaded: **alarmviewer.htm**. You should be prompted to download AWXView32 before the display loads. After the download, the display should look similar to **Figure L8.22.**

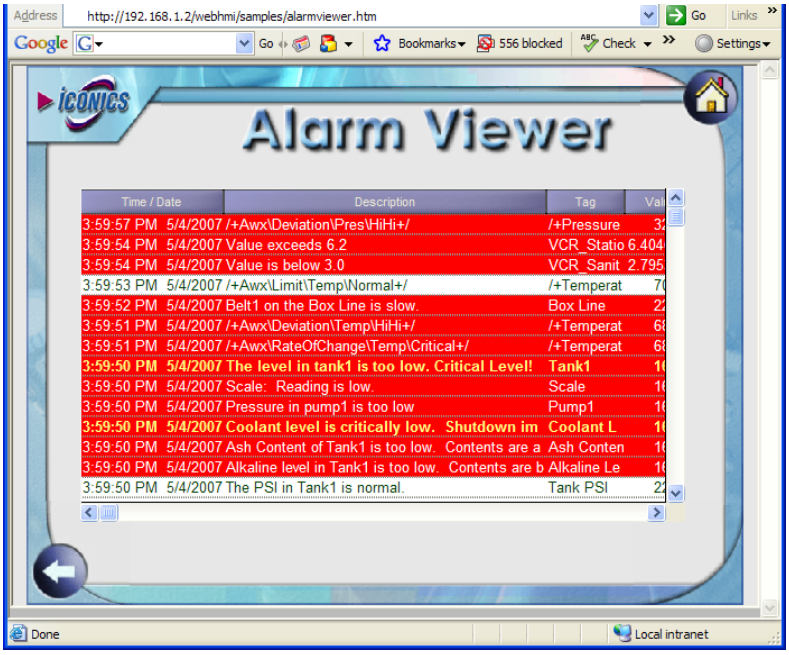

Figure L8.22: WebHMI Alarm Viewer Sample Page

24. Click the back arrow to return to the AlarmWorX32 sample page, and continue to view **Alarm Report** display. Like the Alarm Viewer, this page is not simply loaded through a pick action, but a new HTML page is loaded: **alarmreport.htm**.

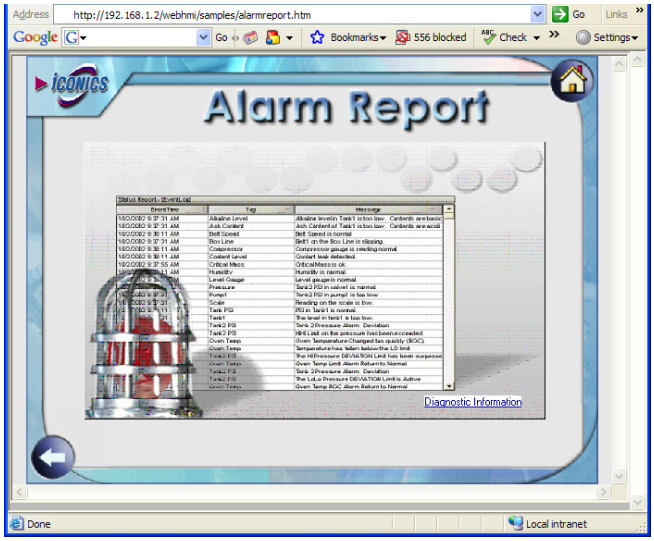

Figure L8.23: WebHMI Alarm Report Sample Page

25. Finally, hit the back arrow button until you return to the main WebHMI page. Click the **Default Samples** link; choose the **TrendWorX32** link, and then the **Time Plot** link.

26. The final HTML page that was edited **timeplot.htm** will be loaded, you will be prompted to download a TrendWorX32 ActiveX control, and then the display should look similar to **Figure L8.24.**

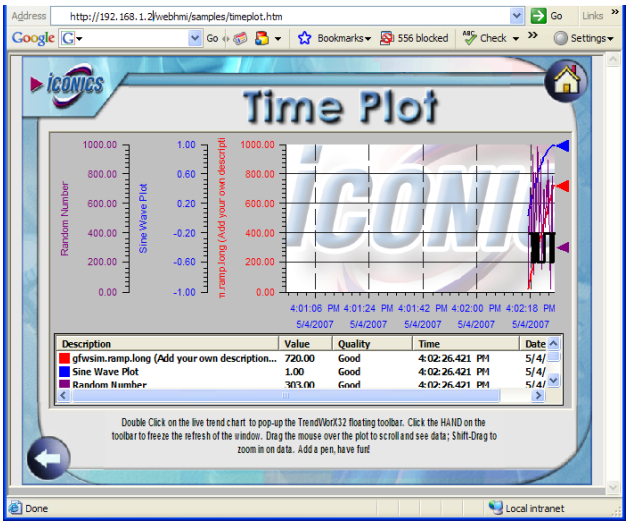

Figure L8.24: WebHMI Time Plot Sample Page

8.4 Preparing Displays for the Advanced Network

Just as with the basic application example, the displays used in this WebHMI system must have the correct network-aware setup so that the data being requested can successfully be found across the network or Web.

The configuration is slightly different for this advanced example, but we will use the same displays and just edit them again.

In the basic application example completed in Chapter 5, all OPC data points requested, alarm servers subscribed to, and databases connected to were done so using the node name of the machines, since the application was developed to be used on a local intranet or LAN.

For this advanced example, not only do the node definitions need to be changed to IP addresses, but recall that the OPC DA and HDA data is now coming from a separate GENESIS32 machine.

Selecting Data Sources from the Remote OPC Server:

1. Open the **Operator3.gdf** display that to begin modifying it for this advanced project.

   NOTE: We will edit a few connections manually to show how it would be browsed for via GenBroker communications if the display were being freshly created. The remainder of the edits will be done through tools provided by GraphWorX32.

2. Select the tank level size dynamic for **Tank1,** edit the symbol, and double click the cut away object to open the **Property Inspector**. Select the **Size** tab and click the **Data Tags** button to edit the data source as shown in **Figure L8.25.**

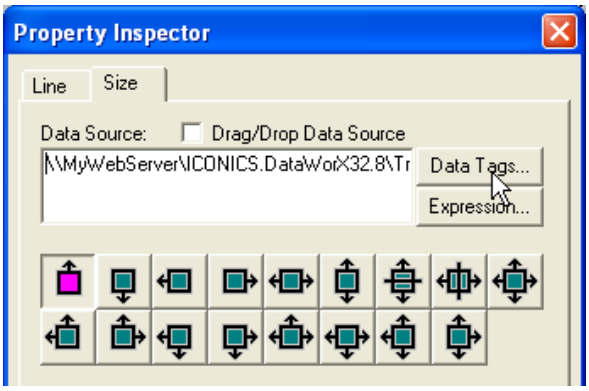

Figure L8.25: Editing the Data Source of a Process Point

3. In the **Unified Data Browser** there are three major branches: **My Computer**, **Network Neighborhood**, and **Internet**.

4. Tags are browsed for through **My Computer** when the displays are used locally and through **Network Neighborhood** when the OPC Server is on the same domain (as we used in the basic example). The **Internet** branch is used for browsing tags when GenBroker communications are being used to get data outside the domain or workgroup.

5. Expand the **Internet** branch of the **Unified Data Browser** to see any channels that are currently in use, as shown in **Figure L8.26.**

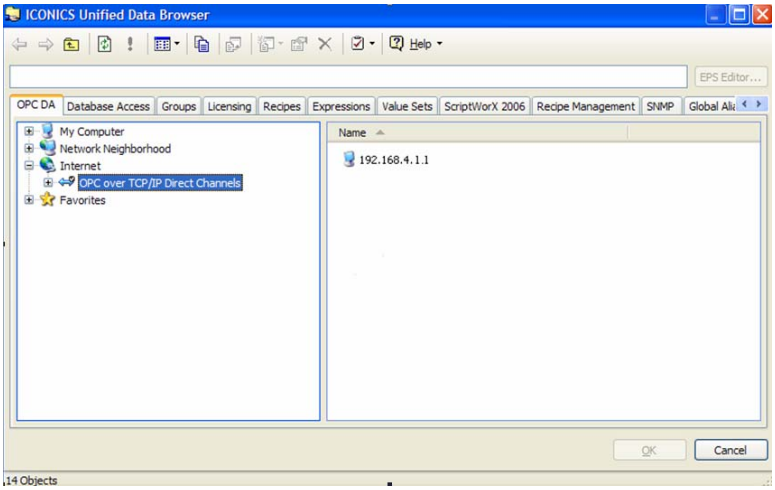

Figure L8.26: Internet Branch of the Universal Tag Browser

6. The **OPC over TCP/IP** branch is available because we configured that channel in our GenBroker configuration.

7. Browse into the **OPC over TCP/IP** channel to find the IP address of the **Simopcserver** machine (**192.168.1.1**). This IP address is available here because we added it to the channel in our GenBroker configuration.

8. You can browse into this node and select the OPC tags as you would through any other machine at this point. Choose the **Tank1.Level** tag as shown in **Figure L8.27.**

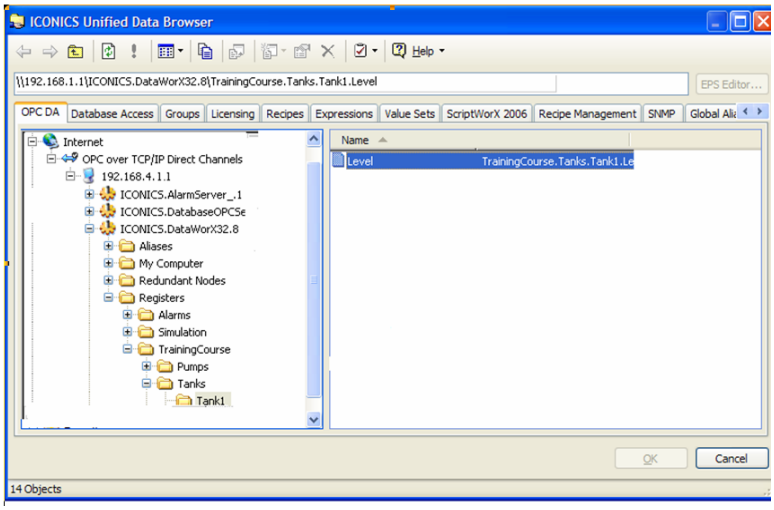

Figure L8.27: Browsing OPC Tags over TCP/IP via GenBroker

9. Click **Ok** to select the tag as the data source when finished and return to the Property Inspector. Notice the IP address in the tag definition.

NOTE: The same method would be used to select an alarm subscription in an alarm viewer, or to select historical pens in a trend viewer.

Replacing Node Information in Existing Displays:

10. Because our three displays (containing OPC data, alarm subscriptions, and historical data) are already created, it would not be efficient to go through and edit each piece of data manually. For this reason, the GraphWorX32 **Replace** feature will be used.

11. Select **Replace** from the **Edit** menu in GraphWorX32. The **Replace** dialog will open in its default state, as shown in **Figure L8.28.**

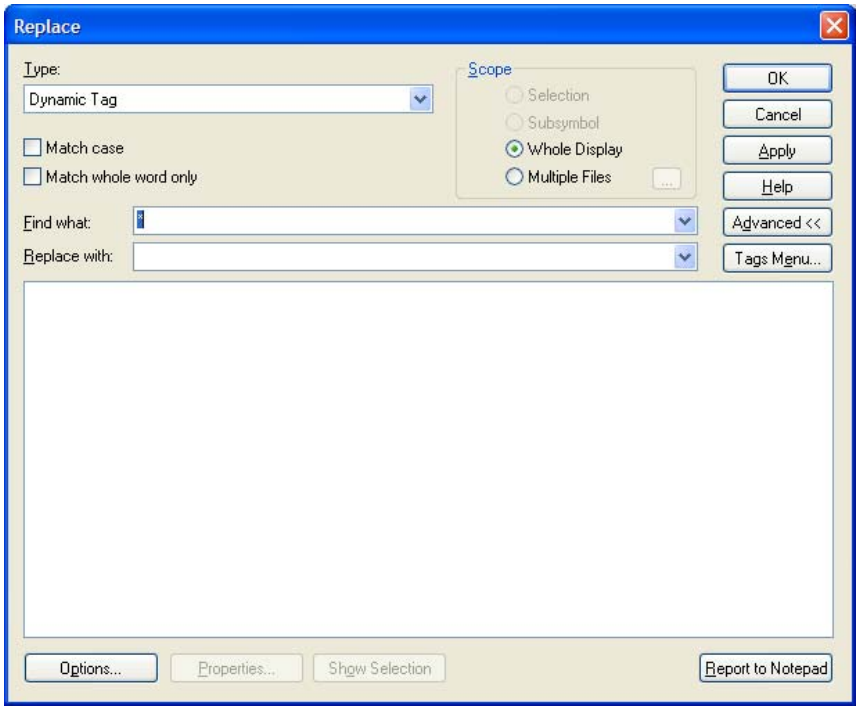

Figure L8.28: Default State of the Replace Tool

12. In all of the existing displays, the node **Mywebserver** was included to be network-aware. Now all OPC data access tags and historical pens are being requested from **Simopcserver**. Alarm subscriptions, however, will still be requested from **Mywebserver**.

13. Since the application is being prepared for use over the World Wide Web, just replacing **Mywebserver** with **Simopcserver** will not be sufficient; IP addresses must be used.

14. We will edit all the OPC DA and HDA tags, so we will leave **Dynamic Tag** as the **Type**.

15. Enter **Mywebserver** as the **Find what:** variable, and the IP address of the Simopcserver machine **192.168.1.1** as the **Replace with:** variable.

16. We want to change all dynamics and process points in the operator displays, as well as the trend pens in the HistTrendViewer display. For this reason, select **Multiple Files** as the **Scope**. More than one display can be edited at once by using the **Multiple Files** option. Click the ... button to select the files to search, as shown in **Figure L8.29.**

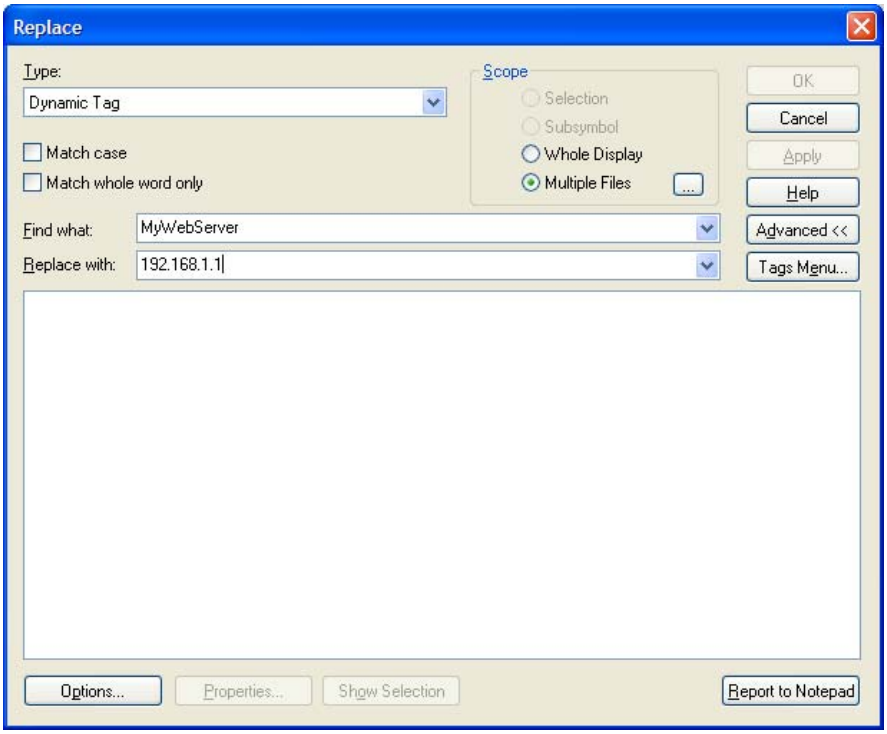

Figure L8.29: Editing Node Definitions of Data Access Tags

17. In the **Select Files** dialog shown in **Figure L8.30,** click the **Add** button to add files to be searched. Browse to and select the files you would like (all operator displays and the historical trend viewer display) so that they are listed in the **Select Files** window. Click **Ok** when finished.

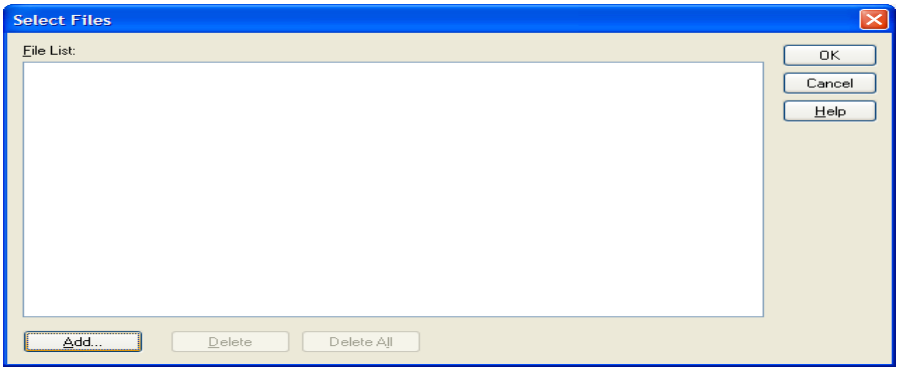

Figure L8.30: Select Files Dialog

18. Click **Apply** to begin replacing the information. You will be prompted with the dialog shown in **Figure L8.31** for each item it searches for information to replace.

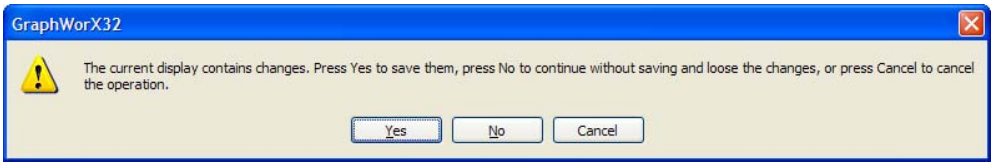

Figure L8.31: Confirming Each Replacement

19. Click **Yes** a few times to see how it moves through each object searching the dynamic tag path for the information to be replaced. If you are comfortable with it, you can click **Yes to All** to allow the tool to change the information everywhere it is found without confirming with you first.

20. When the replacements are complete, each replacement is reported to you in the window of the **Replace** dialog. This information can be reported to Microsoft Notepad in order to save the information if desired by clicking the **Report to Notepad** button.

21. Save this display, as all necessary changes have been made to the data sources of DA and HDA tags.

Changing the Alarm Subscription Node Configuration:

22. Open the **AlarmViewer.gdf** file in order to alter the subscription and database connections as necessary. Unfortunately, the GraphWorX32 Replace tool does not have the capabilities to go into the properties of the alarm ActiveX controls, so they will have to be edited manually.

23. Double-click the **AWXView32 ActiveX Control** to open the properties dialog, and select the **Subscriptions** tab.

24. Double-click the **ICONICS AWX32 Server** and enter 192.168.1.2 into the node field. When finished, this server's properties should look like **Figure L8.32.** Click **Ok** when and then close the Properties dialog to return to the display.

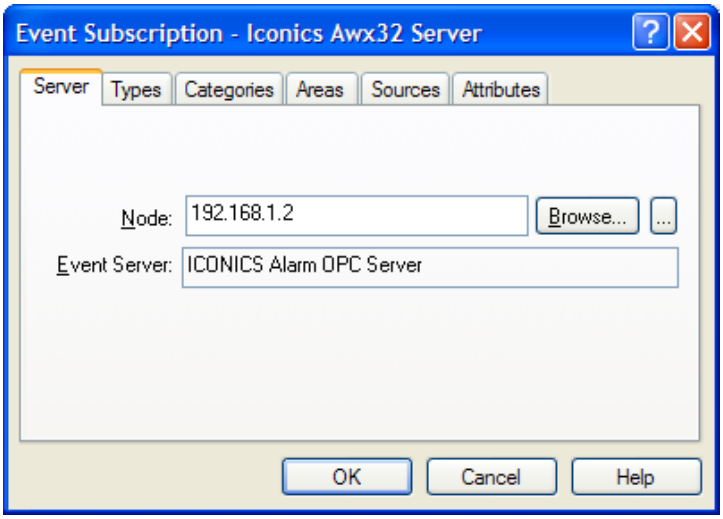

Figure L8.32: AWX32 Server Subscription Properties

25. Save this display, and repeat steps 22-24 to edit the alarm subscription in the **LevelPopup.gdf** file.

26. All displays have now been prepared for the World Wide Web, and altered so that all the data is being requested from the correct server machines.

8.5 Publishing the Displays

Once network-aware displays have been created, they need to be published to the Web Server so that they are available to your WebHMI clients.

When publishing, set the **Publish to Web Server** URL to <**http://12.13.14.15/WebHMI/AdvExample**> (using the IP address of the firewall). This will ensure that the resulting HTML page requests all the necessary files from the firewall IP address, which will in turn redirect the requests to the WebHMI server.

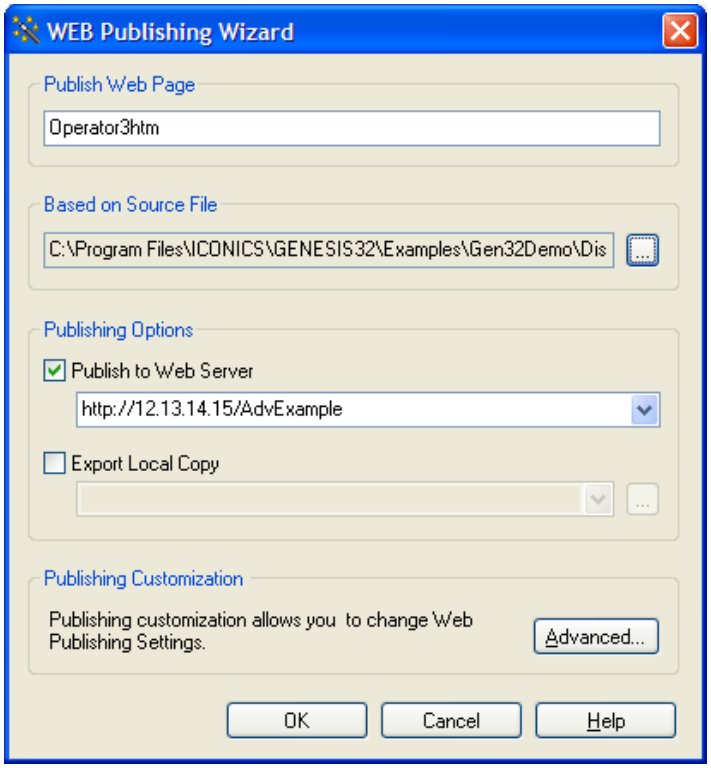

Figure L8.33: Publishing for the Advanced Network

In the **Advanced** settings of the Web Publishing Wizard, make sure that the new .gbx file is chosen as the **User Defined** GenBroker configuration file with the URL **<http://12.13.14.15/WebHMI/AdvExample/AdvancedCFG.gbx>,** as shown in **Figure L8.34.**

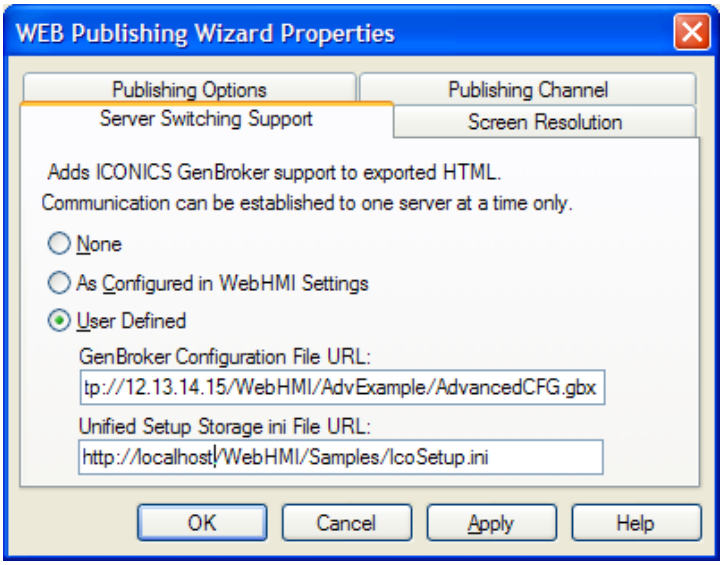

Figure L8.34: Choosing the Advanced GenBroker Configuration File

Please refer to Chapter 6 for information and instructions on publishing displays.

If your displays are already published, you can use the **Change Host Name Utility** to change the host name to **12.13.14.15**, as explained in **Appendix A**.

8.6 Viewing the WebHMI Displays

Using the same WebHMI client as used during testing of the Advanced Application WebHMI server (or one with the same characteristics), view the WebHMI application that was developed in this chapter. Go through the same testing steps you used in Chapter 7. You may want to test the application on machines both on the LAN and across the World Wide Web to be sure it works in both cases.

Make sure that GenBroker is running on the WebHMI server **Mywebserver**. Also make sure that GenBroker is running on the **Simopcserver** machine so that OPC data can be requested as necessary.

On the WebHMI client, enter the Web address for the HTML page that the main display was published to:

<<http://12.13.14.15/webhmi/MyAdvancedProject/Operator3.htm>>

Using the Change Host Name Utility A

A.1 Introduction to the Change Host Utility

At some point, you might find that you wish to switch Web servers or use an OPC server on a different node. The WebHMI Change Host Name Utility enables you to do this easily. To open the Change Host Utility from the Windows **Start** menu, select **Programs → ICONICS WebHMI → Change Host Name Utility**. The **Change Host Name** dialog box will appear, as shown in **Figure A.1.**

NOTE: The process of running the Change Host Utility may require several minutes, and during this time the circle associated with each file type shown in Figure 8.1 (on the right side) will change color. The circle will be red while processing a specific file type, and it will be green when the specific file type is completed with success. The screen may actually flash for an instant while the utility is "operating" on a file. It is possible to see the file currently processed by reading the file name in the bottom right part of the dialog.

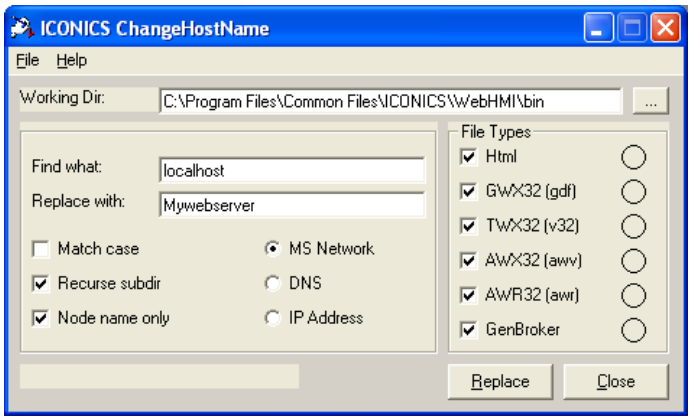

Figure A.1: Change Host Name Dialog Box

A.2 Changing the Directory Name

Change the directory name by typing it in the **Working Dir** field, as shown in **Figure A.2.** You can also change the directory name by clicking on the button to the right of the **Working Dir** field or by selecting **Set Working Directory** from the **File** menu.

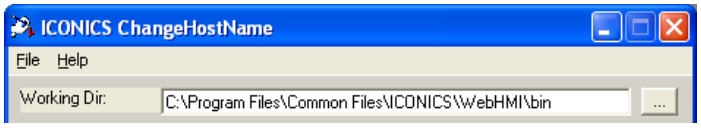

Figure A.2: Directory Name

A.3 Finding and Replacing Files

Functioning like any "find-and-replace" command, the **Change Host Name** dialog box searches ICONICS files for:

- **GWX32:** GraphWorX32 (.gdf) display files
- **TWX32:** TrendWorX32 (.v32) files
- **AWX32:** AlarmWorX32 Viewer (.awv) files
- **AWR32:** AlarmWorX32 Report (.awr) files
- **GenBroker:** GenBroker configuration (.gbc or .gbx) files
- **HTML:** .htm files

Choose the type of file for which you want to search by clicking on the check boxes in the right column, as shown in Figure I.1. The **Change Host Name** utility searches for the text typed in the **Find what** field and replaces it with the text typed in the **Replace with** field, as shown in **Figure A.3.**

Figure A.3: Find What and Replace With Fields

⇒ **Check Boxes**

Check the **Match case** check box to do case-sensitive searches, and check the **Recurse Subdir** check box to search all subdirectories of the working directory. A close-up view of this section of the dialog box is shown in **Figure A.4.**

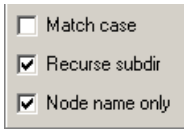

Figure A.4: Check Boxes

Node name only is a special check box that uses **Find String** and **Replace String** and adds "\\" or "//" at the beginning and "\" or "/" at the end of these strings. This confirms that you have really replaced network node names.

You can modify the AlarmWorX32 files and GenBroker Configuration files only by selecting **Node name only.** There is no other text information that you can change. The check boxes for the AlarmWorX32 and GenBroker Configuration files will be grayed out if you uncheck the **Node name only** check box.

Thus, when you uncheck **Node name only,** you can use the Change Host Name utility to change, for example, a point name in GraphWorX32 displays and TrendWorX32 files at the same time. You must then enter the original name in the **Find what** field and the new name in the **Replace with** field.

⇒ **Other Options**

The three options to the right of the check boxes allow you to change the name or address in the **Replace with** field. As shown in **Figure A.5,** a name appears in the **Replace with** field if you select the **MS Network** option.

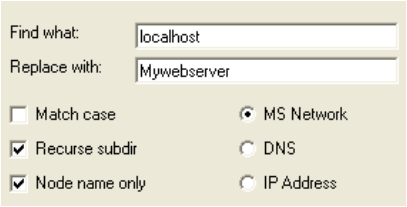

Figure A.5: MS Network Option

Similarly, another name or address will appear in the **Replace with** field if you select the **DNS** or **IP Address** option, as shown in **Figure A.6** and **Figure A.7,** respectively. Click the **Close** button to exit from the Change Host Name Utility.

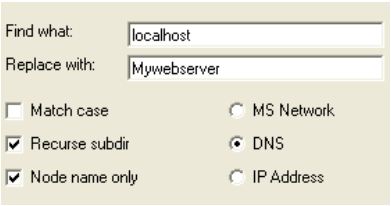

Figure A.6: Domain Name Services Option

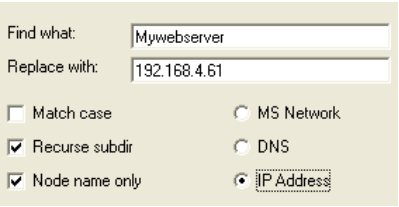

Figure A.7: IP Address Option

Using the Versions Utility B

B.1 Introduction to Version Support and the Versions Utility

Version support is a feature in WebHMI that allows for the auto-update of ActiveX components. When ActiveX components are downloaded from a WebHMI server to a WebHMI client over the Internet, the ActiveX components are silently installed on the client machine. The ActiveX components are delivered into the .cab file archive. The .cab files are referenced in the HTML code near the corresponding ActiveX "clsid". Internet Explorer can detect whether ActiveX components are present on the client machine. Thus, if an ActiveX is already installed on the client machine, then there is no need to download a 2 MB .cab file for that ActiveX.

For example, suppose that a client machine is running Version 9.0, and you want to connect the client machine to a WebHMI Version 9.1 server. In this case, the client machine requires GenBroker Version 9.1 support in order to communicate with the WebHMI 9.1 server. It also requires the new GraphWorX32, AlarmWorX32, TrendWorX32, and all other ActiveX components from Version 9.1. The ICONICS ActiveX components are backward compatible; you can open a Version 9.0 file from a Version 9.1 ActiveX component, but you cannot open a Version 9.1 sample file if you have a Version 9.0 ActiveX component.

Now, for example, suppose that the client has already been connected to a WebHMI 9.0 server, and the client has already installed all the ActiveX support for GENESIS32 Version 9.0. Also suppose that you want to connect the same client to a WebHMI 9.1 server. If there is no version information in an HTML page, Internet Explorer will not download the new ActiveX .cab files because the ActiveX with the same ID is already registered on the client machine.

To avoid such compatibility problems, WebHMI adds the correct version number information to HTML pages. This is accomplished with the **Versions Utility.** This utility gets the version information from an information (.inf) file called **WebHMI.inf**, which contains information about the entire WebHMI installation, including which archives are installed in which directories, what component they contain, and the version numbers of the components installed.

The WebHMI Versions Utility manages all versioning information on the server side. The utility allows you to modify HTML files by adding version information for ActiveX components. The version information in HTML files is necessary for the automatic update of ActiveX components. If an older version of an ActiveX is

already installed and registered on your machine, a newer version will be downloaded, but only if the version information is placed in the HTML file.

The Versions Utility is similar to other "find-and-replace" utilities that search in a specified directory for entered files. The utility searches for the occurrence of ActiveX objects in HTML files. If it finds any, it then searches the WebHMI.inf file for the updated version and inserts the new version information into the HTML files.

B.2 Running the Versions Utility

To open the Versions Utility from the Windows **Start** menu, select **Programs → ICONICS WebHMI → Versions Utility,** as shown in **Figure B.1.** The **Update Versions** dialog box will appear, as shown **Figure B.2.**

NOTE: The Version Utility is Web-server oriented and should be used by the Web server (WebHMI server) administrator only.

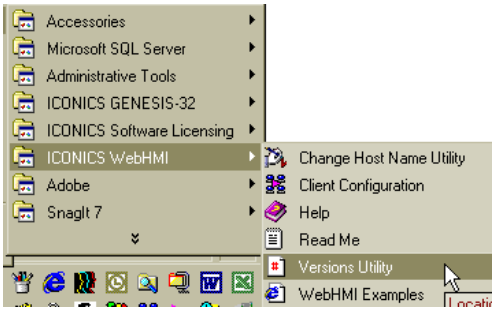

Figure B.1: Starting the Versions Utility

The **WebHMI Inf File** field specifies a path to the information file, which contains version information for ActiveX components. The default file is the WebHMI.inf file previously described. If you want to change the file or its location, you can simply type the new location, or click the ... button to browse for the file.

The **In Files/File Types** field contains a list of files and/or file types to search for. The default file types are *.htm and *.html; the utility will search for all .htm and .html files in the specified location.

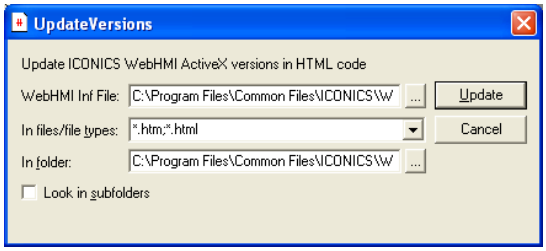

Figure B.2: Update Versions Dialog Box

The **In Folder** field contains the name of the folder where the search will be started. You can also use the button to change the folder. If the **Look In Subfolders** check box is checked, the search will be recursive for all subdirectories of the selected folder.

When you click **Update,** the utility will search for the specified files in the selected directories. If the utility finds any of them, it will open each file and search the file for the occurrence of an ActiveX. If an ActiveX is found, the utility will search for version information for that ActiveX in the WebHMI.inf file.

If the information is found, the updated version information will be automatically appended to the .cab file address in the HTML code. For example, a GraphWorX32 Viewer ActiveX (GWXView32.cab) with version 9.00.138.00 in a Web page will be appended to the "CODEBASE" attribute of the "OBJECT" block in the HTML code as follows:

CODEBASE=http://www.WebHMI.com/WebHMI/cabs/**GWXview32.cab#Version=9.00.138.00**

Securing Your Web Server C

C.1 Introduction to Web Server Security

The World Wide Web is probably the most popular channel used to make information available to a large group of people. ICONICS WebHMI enables just this and helps make your SCADA/HMI screens publicly accessible through the Internet. However, if stringent security is not configured and implemented, then your public web site is vulnerable to a multitude of security problems. One example of such a problem is that any intruder/hacker can change the contents of your Process Control WebPages. To overcome risks associated with such security problems, ICONICS recommends that your web server be secured in ways additional to the ICONICS Security applications. This document details the various ***levels*** of security that can be implemented on an ICONICS WebHMI server.

- **Level 1 or Base Line Security**: Web server security using Internet Information Server and Windows security
- **Level 2 or Enhanced Security:** Web server security using Secure Socket Layer Certificate & a firewall with ICONICS GenBroker
- **Level 3 or Security for a Demilitarized Zone (DMZ)**- Web server security using 2 firewalls around the webserver

C.2 Level 1: Base Line Security

Internet Information Services (IIS) is a high-speed web server used to publish and distribute WWW-based content to standard browsers. ICONICS WebHMI requires the installation of a web server like IIS. IIS can be installed on Windows NT/2000 & XP. For help on installing IIS on the different OSs, please refer to ICONICS Whitepaper titled "WebHMI.pdf" on your product CD under the E:\GENESIS32\Manuals\files directory. IIS application runs as a service and can be seen in the Services list in your Administrative Tools. IIS can be managed from an application called Internet Services Manager (can be opened form Control Panel-Administrative Tools).

This document explains security set up on the Windows 2000 OS. (Please note that if you are using Windows XP or Windows 2003 server, the screen shots may vary from what is in this document)

Level 1 security is recommended for a network similar to figure shown in **Figure C.1.**

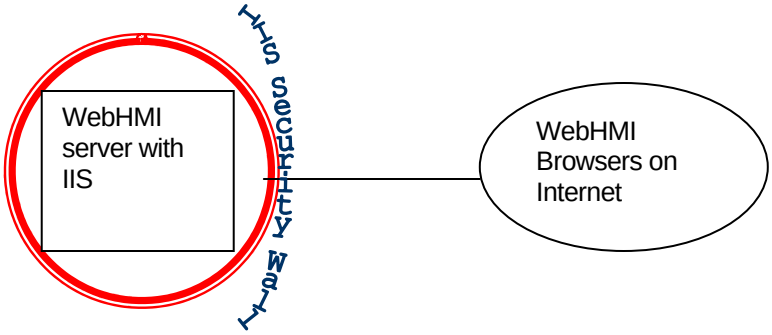

Figure C.1: Level 1 - WebHMI Server with IIS

IIS security is tightly tied together to the operating system. File permissions, registry settings, password usage, user rights, and other issues associated with Windows 2000 security have a direct impact on IIS security.

C.2.1 Configuring Internet Information Services (IIS)

To secure the information published on your web server it is recommended that you prevent unauthorized users from accessing IIS by following these steps:

1. Open the IIS configurator from **Start → Settings → Control Panel → Administrative Tools → Internet Services Manager**. The **Internet Information Services** window will open. **Figure C.2** shows the window slightly expanded.

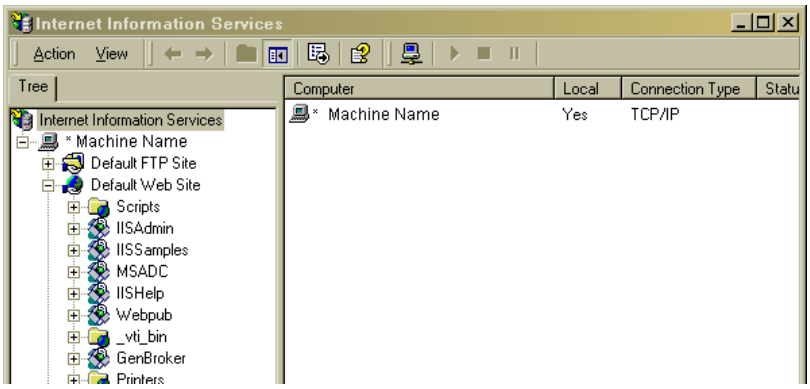

Figure C.2: Internet Information Services Configurator

2. Upon installation, WebHMI will create a folder under the **Default Web Site** as shown in **Figure C.2.** Properties can be set for this folder that will decide the level of security you give to the public when accessing this folder and its contents from the Web.

3. Right click on the **WebHMI** virtual directory to open the property dialog shown in **Figure C.3.**

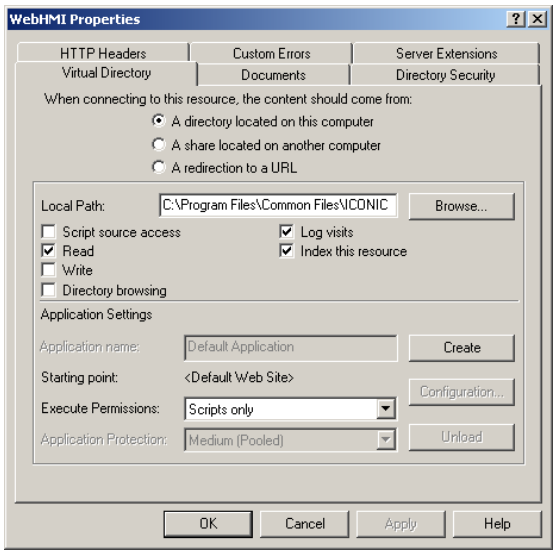

Figure C.3: WebHMI Properties

⇒ **Directory Tab:**

4. When connecting from the Internet, you want the content to come from the correct L**ocal Path**: **C:\Program Files\Common Files\ICONICS\WebHMI**.

5. **Read** permissions are essential in order to allow the WebHMI users to view your displays.

6. It is sufficient to give **Execute permissions** for Scripts only, so that only scripts, such as ASP scripts, can be run.

7. **Application Protection** can be set to **Medium**, so that applications are run in an isolated pooled process in which other applications are also run. (in **Low,** applications are run in the same process as Web services, and in **High,** applications are run in an isolated process separate from other processes.)

8. Clicking the **Configuration** button (if enabled) brings up the **Application Configuration dialog** shown in **Figure C.4.**

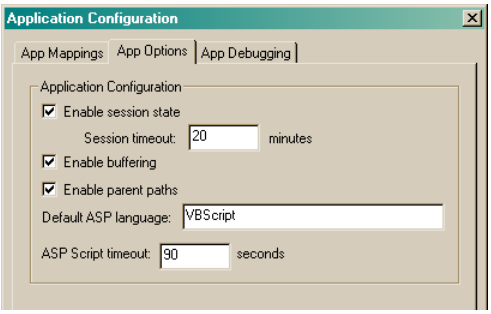

Figure C.4: Application Configuration

9. The **App Options** tab can be used to set a **Session Timeout**. If session state is enabled, Active Server Pages maintains a record of each user who accesses an ASP application so that you can identify the user across pages in the application. ASP scripts store information in the **Session** object or use the **Session_OnStart** or **Session_OnEnd** events.

⇒ **Directory Security Tab**

10. Close out the **Application Configuration** window and switch to the **Directory Security** tab of the **WebHMI Properties** window, as shown in **Figure C.5.** This tab allows you to configure **Anonymous access and Authentication control**.

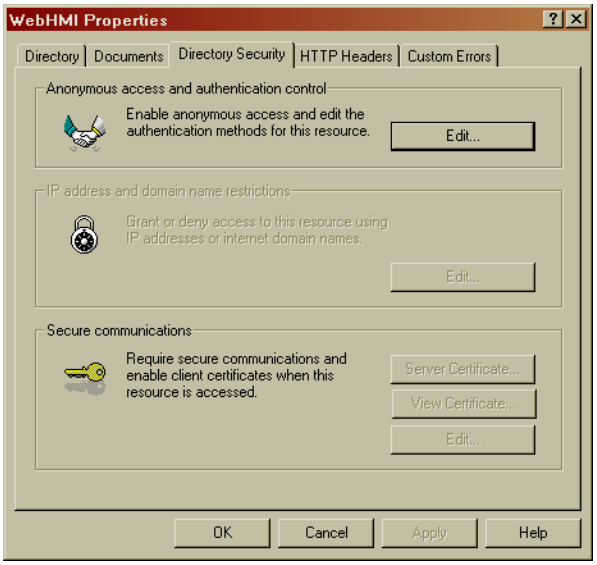

Figure C.5: WebHMI Properties: Directory Security

11. Click the **Edit** button to open the **Authentication Methods** window shown in **Figure C.6** in order to change your authentication options.

12. Uncheck the **Anonymous access** option to prevent unauthenticated users from accessing your web server.

13. You now have an option of using either **Integrated Windows Authentication** or **Basic Authentication**.

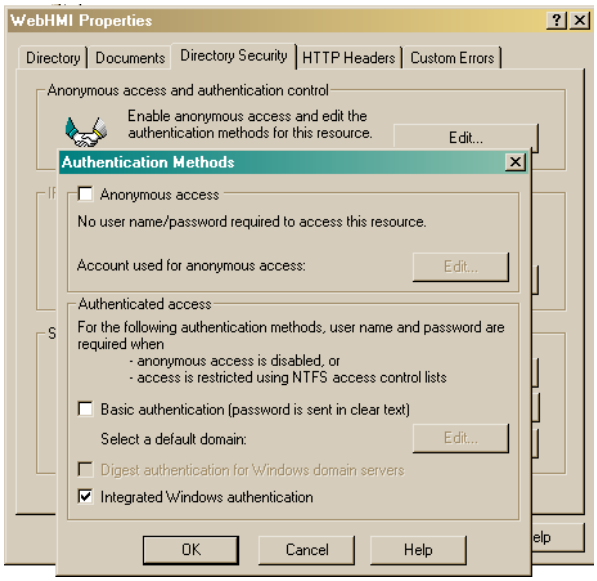

Figure C.6: Authentication Methods

> **Important!** ICONICS recommends to use **Integrated Windows Authentication** with a specific user name and password such as Scada and genesis32 or similar.

14. Checking **Integrated Windows Authentication** option will allow ONLY authenticated windows users on the web server machine to view the web server machine over the Internet. On the WebHMI client side, the current Windows user information is used for the integrated Windows authentication. If this fails, then it requires users to type in a user name and password that can be a validated account on the Web server machine. You may save the password the first time so frequent login prompts don't pop up.

> **Warning**! Selecting **Basic Authentication** results in the transmission of passwords across the network in an unencrypted form. This makes it easy for computer hackers to intercept user names & passwords. **ICONICS** does not recommend using Basic Authentication.

C.2.2 Windows Files Security

The above process of configuring IIS permissions will allow access to the web server node from the web client node alone. In addition it is important to give access to the WebHMI files themselves.

Right-click on the WebHMI folder (or the folder that contains all your HTML files) in **C:\Program Files\Common Files\ICONICS** (if the default installation path was chosen during the WebHMI installation). On the **Folder Properties** dialog, go to the Security tab and **ADD** the allowed user/users (and example user SCADA is shown in **Figure C.7**) to this list with **Read** and **Write** permissions.

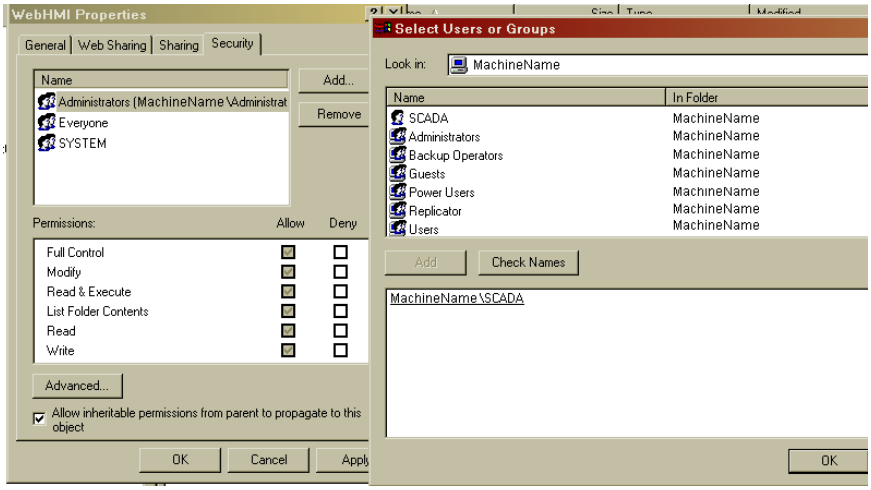

Figure C.7: Adding User Permissions to File Security

C.3 Level 2: Enhanced Security

It is possible to secure your WebHMI server behind a firewall. This firewall can be a software or hardware **firewall**. **ICONICS GenBroker** technology can be used to pass information through a firewall. A firewall combined with SSL or **Secure Socket Layer** implementation in your IIS will provide an "enhanced" level of security.

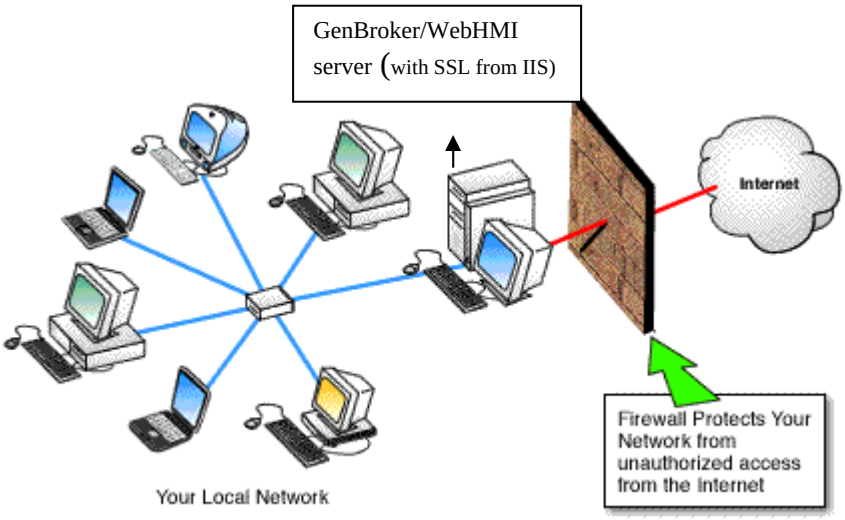

Figure C.8: WebHMI Server Behind a Firewall

Visualize Your Enterprise™

C.3.1 Setting up ICONICS Applications on the WebHMI Server

To setup GenBroker through a firewall (router), we will assume the following setup:

- A WebHMI server machine inside your network with an internal IP address (we will use 192.168.1.1 in our example)
- A router or firewall with an external IP address (we will use 12.13.14.15).
- All publishing will be done to the internal address. You will most likely need to publish locally because if you try to publish to the external interface of the firewall you will have a loopback issue causing the publishing to fail.

  NOTE: Once finished, clients inside the network will NOT be able to use these same published WebHMI pages as those outside the firewall or router.

- The HTTP port (80 by default) needs to be open and redirected to the WebHMI server machine for typical web serving communications.
- The GenBroker port (38080 by default) needs to be open for two-way TCP/IP communication and redirected to the WebHMI server machine.
- Although these ports need to be opened, security is not compromised because all data from ICONICS GenBroker is encrypted and so cannot be deciphered by a port sniffer.

The following steps will help you to set up the ICONICS applications on your WebHMI server for use with the network setup just discussed:

15. The GenBroker client configuration file needs to be setup properly. Open the GenBroker Client Configuration tool from **Start → Programs → ICONICS WebHMI → Client Configuration**.

16. Open your active WebHMI .gbx file, or create a new one and configure it as shown in **Figure C.9.**

17. The mediator channel has the external address as the via node, and the internal address is added as a node to that channel. This is similar to the configuration in our advanced WebHMI example the Practice Lab of this book.

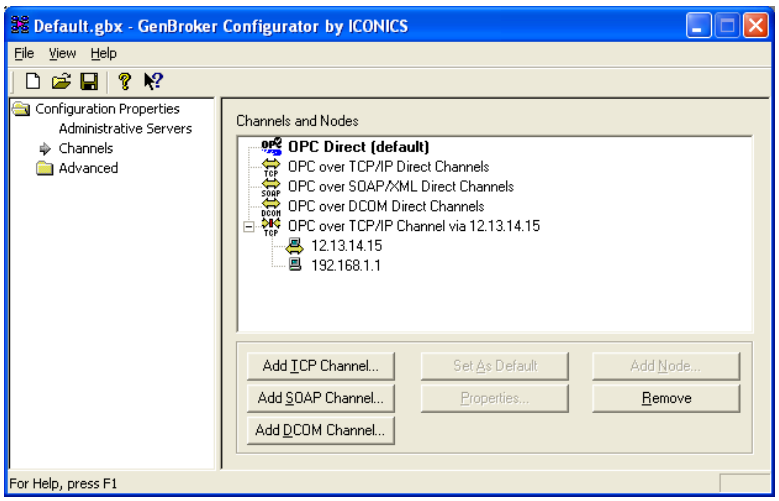

Figure C.9: GenBroker Setup Over a Firewall

18. The Primary Servers have the internal address, as shown in **Figure C.10.**

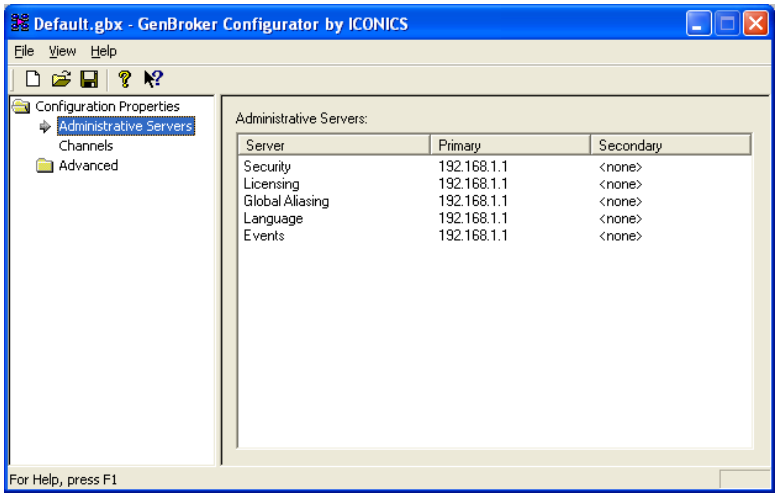

Figure C.10: Primary Servers All Point to IP Internal to the Firewall

19. The GWX displays all need to use the internal addresses when referencing OPC tags.

20. The published HTML pages use the external address for HTTP servers. The Change Host Name utility can be used for existing HTML pages, as shown in **Figure C.11.**

Visualize Your Enterprise™

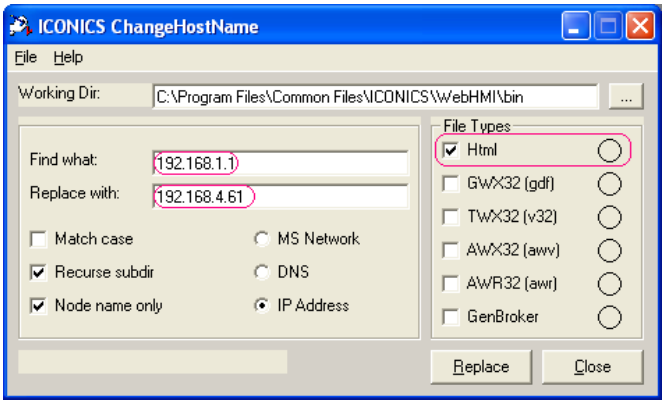

Figure C.11: All HTML Pages use the IP External to the Firewall

C.3.2 Setting up Secure Socket Layer in IIS

Start by opening the IIS Manager from **Start → Settings → Control Panel → Administrative Tools → Internet Services Manager** and take the following steps to set up a Secure Socket Layer in IIS:

1. Right-click on the **Default Web Site** folder and choose **Properties**.

2. On the **Directory Security** tab, there is an option to add what is called a server certificate that enables secure communications via the SSL. To create a server certificate request using the new **Certificate Wizard**, click **Server Certificate**.

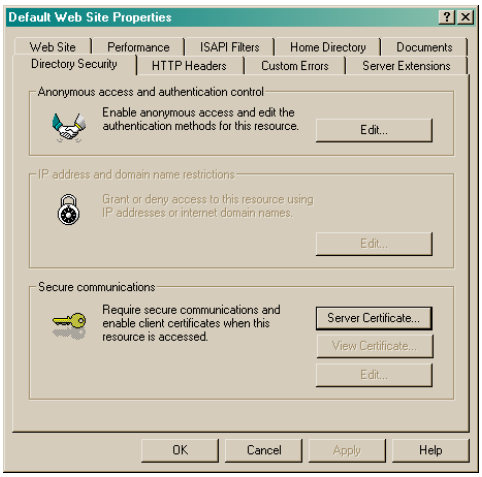

Figure C.12: SSL Certificate from IIS

3. You will not be able to use your Web server's secure communications feature unless you have a valid server certificate installed. For more information, see IIS help.

4. VeriSign & Geotrust are Top Enterprise Class SSL Providers. For more information, visit www.verisign.com and http://www.geotrust.com/

C.4 Level 3: Security for a Demilitarized Zone (DMZ)

A DMZ, short for ***dem**ilitarized **z**one*, typically comprises of a computer that sits between a trusted internal network such as a corporate private LAN, and an untrusted external network such as the public Internet. Typically, the DMZ contains a device accessible to Internet traffic, such as a DNS server.

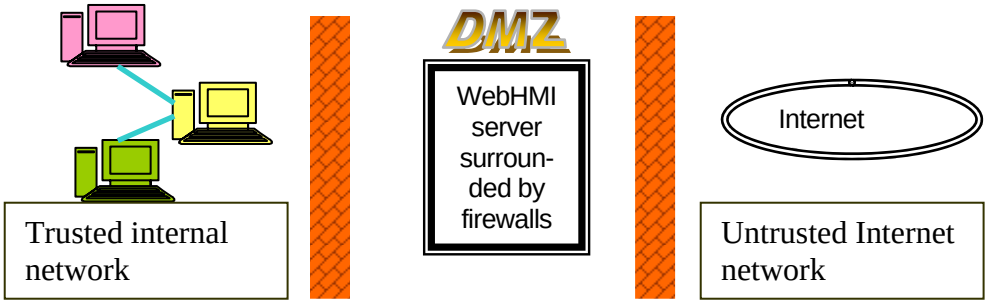

Figure C.13: WebHMI Server Setup - Demilitarized Zone

In a setup that resembles **Figure C.13,** the trusted intranet will have a DNS name with which to access the Web server machine. The firewall on this side of the DMZ (to the left in **Figure C.13**) will route all calls for that DNS name to the Internal IP address of the Web server machine.

Similarly, the untrusted Internet will also have a DNS name with which to access the Web server machine. The firewall on this side of the DMZ (to the right in **Figure C.13**) will route all calls for that DNS name to the Internal IP address of the Web server machine as well.

ICONICS GenBroker will need a special setup, explained in the next section, for WebHMI to function across a DMZ.

⇒ **GenBroker Setup**

The general architecture is that the published displays will use one DNS name (such as webhmi.mycompany.com) that is resolvable both internally and externally to the same WebHMI server.

For this example we will assume the following:

- The common DNS name is **webhmi.mycompany.com**

- The internal IP address of the WebHMI server is **192.168.1.10**
- The external IP address of the firewall/router that takes care of redirection is **12.15.10.100**
- For simplicity, both the internal and external IP addresses refer to the same machine (the WebHMI server) in this example.

Ultimately, the internal and external DNS servers need to resolve webhmi.mycompany.com to point to the WebHMI server, either through 192.168.1.10 or 12.15.10.100 respectively. The external DNS server will handle all the requests coming from outside the firewall and resolve Webhmi.mycompany.com to 12.15.10.100. The internal DNS server will handle all the requests coming from inside the firewall and resolve Webhmi.mycompany.com to 192.168.1.10.

Once communication with the WebHMI server is established and the client begins requesting data, the firewall/router needs to take the incoming requests and route them to the WebHMI server accordingly. If the SOAP/XML channel is being used then port 80 needs to be open and if the TCP/IP channel is being used then port 80 and the GenBroker port (38080 by default) need to be open.

The GenBroker Configuration file *.gbx that is downloaded to the WebHMI client needs to be configured with a VIA channel (either SOAP or TCP/IP depending on your setup). The name of the channel needs to be the DNS name, webhmi.mycompany.com in our example.

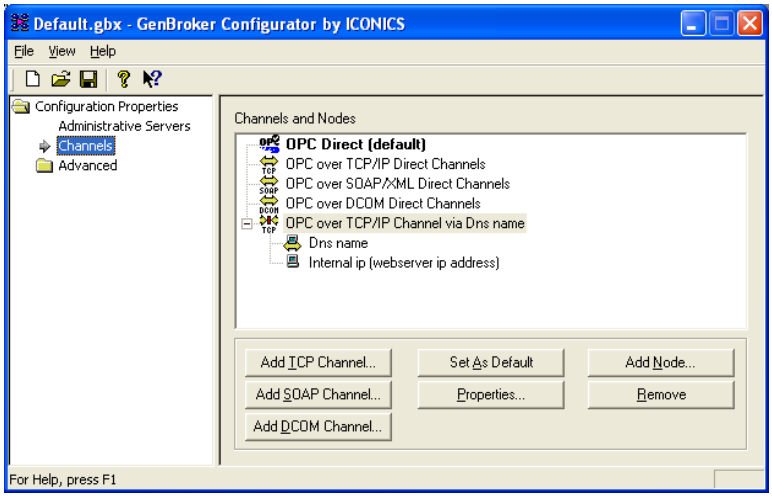

Figure C.14: GenBroker Setup for DMZ

Added to that channel should be the internal IP address (or the local node name) of the WebHMI server, and the mediator channel should be made the default. This ensures that all client GenBroker requests are channeled through the DNS name. All Remote Servers (e.g. Licensing and Security) should be set to the internal IP address or the local node name of the WebHMI server.

In your displays, the internal IP address or node names should be used for all OPC server names. This will be in tag names and in server names for the AlarmWorX32 Viewer and in the TrendWorX32 Viewer.

WebHMI Security and Licensing D

D.1 Using Security and Licensing with WebHMI

Two important parts of GENESIS32 are the ICONICS Security Server and License Monitor. The Security Server allows you to add many layers of security to your applications from locking down particular OPC points on a single display to locking down the entire desktop including the Start Menu. The License Monitor allows you to view your current active license and monitor its usage by all the applications on your system.

Both of these utilities can be used over WebHMI, allowing added security and license monitoring of your GENESIS32 application via the Web.

D.2 WebHMI Security ActiveX

An example of WebHMI Security is included in the WebHMI sample screens. To access the samples, select **Start → Programs → ICONICS WebHMI → WebHMI Samples.** The **WebHMI Samples Home Page** appears, as shown in **Figure D.1.**

Figure D.1: WebHMI Samples Home Page

Click on **Default Samples** to open the WebHMI **Default Samples** page, as shown in **Figure D.2.** Click on the **Security** button to open the Security sample page shown in **Figure D.3.** The **Symbol Library** in GraphWorX32 contains a symbol category file

called **WebHMI Security Login.sdf**, which contains several symbols that, when dragged into a GraphWorX32 display, allow users to gain access to the Security Server.

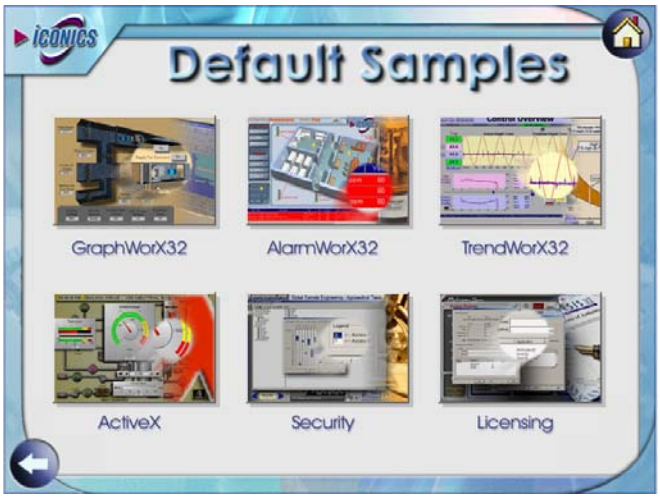

Figure D.2: WebHMI Default Samples Page

All the symbols use VBScript to call the Security Server on the remote WebHMI Server and retrieve security information, and are shown on the **Security** page of the WebHMI samples.

Figure D.3: WebHMI Security Samples Page

You do not need to know VBScript to use any of these WebHMI Security symbols. You can directly drag and drop the symbol that you need from the Symbol Library into your display, but you also have the freedom to use the Script Editor toolbar in GraphWorX32 to change the source code associated with each of these symbols. You can also copy the code and attach it to your own symbols.

All of the scripts associated with the WebHMI Security symbols create an instance of the **ICONICS Login ActiveX** and call methods of this object or access properties. The complete automation for the ICONICS Login ActiveX is described later in this appendix.

D.3 Logging Into the Security Server

The **WebHMI Security Login ActiveX** symbol button, shown in **Figure D.4,** enables WebHMI users on remote client machines to log in to the Security Server. For example, if the Login symbol button is placed in a GraphWorX32 display, the user can simply click on the symbol (button) in runtime mode to launch the **WebHMI Security Login** dialog box, shown in **Figure D.5.**

Figure D.4: Security Login Symbol Button

The **WebHMI Security Login** dialog is basically the same as that of the Security Login application located on the server machine, except that the **Advanced** login mode is disabled, as shown in the **Figure D.5.** The WebHMI Security Login ActiveX also includes full keypad support (ideal for touch-screen systems). The Login ActiveX allows simultaneous login of many users (this must be enabled on the Security Server global settings).

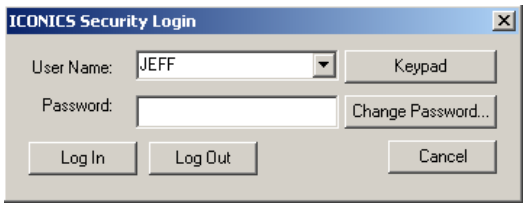

Figure D.5: Logging Into the Security Server

The drop-down list for the user name can show:

- The complete list of users in the system.
- The list of the currently logged users.
- The name of the last logged user.

All of these features must be enabled on the Security Server in order to work.

For more information, please see the Security Configurator Help documentation.

NOTE: Unlike the Security Server on a full GENESIS32 system, when you log into the Security Server using the Login ActiveX you do not get any warning messages when the security session is about to expire. If your security session expires, then the Login ActiveX will automatically be displayed again.

D.4 Changing the Security Server Password via WebHMI

If you should wish to change your password from a WebHMI client, you can do so by clicking on the **Change Password** button on the login ActiveX dialog, or by using the **Change Password** symbol button in one of your own published displays, as shown in **Figure D.6.**

Figure D.6: Change Password Symbol Button

Clicking the **Change Password** button opens the **Security Password Change** dialog box shown in **Figure D.7.** After confirming your current password, type your new password in the **New Password** and **Retype Password** fields, and then click the **OK** button.

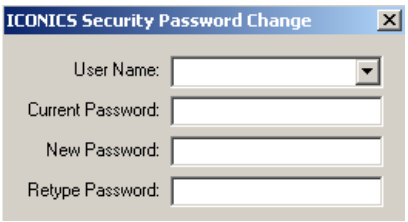

Figure D.7: Changing the Security Password

D.5 Viewing the Logged User List

To view a list of users currently logged in to the Security Server, use the **Logged User List** symbol button in your GraphWorX32 display, as shown in **Figure D.8.**

Figure D.8: Logged Users List Button

The **Security** window will appear, as shown in **Figure D.9.** The **Security** window allows you to view the list of users that have logged in.

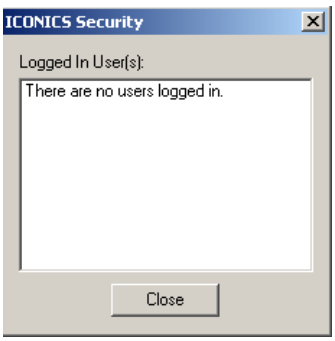

Figure D.9: List of Logged Users

D.6 Logging out of the Security Server

To log out everyone who has logged in, you can use the **Log Out All Users** symbol button shown in **Figure D.10.**

Figure D.10: Logout All Button

You can also logout one specific user with a simple click on the **Logout User** button, as shown in **Figure D.11.**

Figure D.11: Logout User Button

NOTE: You have to specify the user to be logged out in the VBScript code associated with this button. You can do it by editing the script with the Script Editor toolbar in GraphWorX32.

D.7 Security OLE Automation

The OLE Automation interface for the WebHMI Security ActiveX is compatible with VBA and VBScript, although only VBScript (not VBA) is supported in WebHMI. You can perform login/logout operations directly through scripting without displaying any user interface. The WebHMI Security ActiveX contains the following OLE Automation interfaces:

LoginDlg()
Launches the login dialog.
ChangePwdDlg()
Launches the dialog to change the password.
ShowLoggedInUsers()

Launches the dialog to show a list of the user currently logged into the Security Server.

Logout()

Logs out all currently logged users.

SetTimeout(LONG nSec)

Sets the timeout for all of the GenClient calls to the Security Server.

ShowResultMsgs(BOOL bShow)

Enables/disables the message box with the result (i.e. "failed to log on to the Security Server").

LoginUser(BSTR username, BSTR password)

Logs in one specific user through code.

LogoutUser (BSTR username)

Logs out a specific user through code.

D.8 Viewing Licensing Information Using WebHMI

An example of WebHMI Licensing is included in the WebHMI Sample screens. To access the samples, select **Start → Programs → ICONICS WebHMI → WebHMI Examples.** The **WebHMI Samples Home Page** will appear, as shown in **Figure D.12.**

Figure D.12: WebHMI Samples Home Page

Click on **Default Samples** to open the WebHMI Default Samples page, as shown in **Figure D.13.**

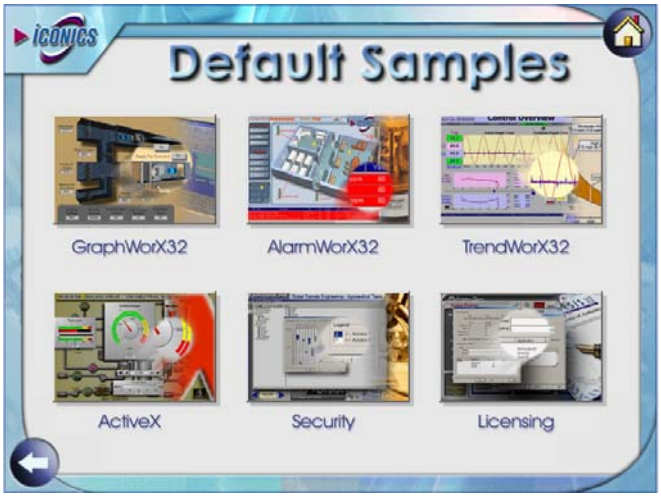

Figure D.13: WebHMI Default Samples Page

Clicking on the **Licensing** button opens the Licensing page, which allows you to launch the GENESIS32 License Monitor on the WebHMI client. The License Monitor allows you to get an immediate view of the number of Client Units your enterprise is using at a given time.

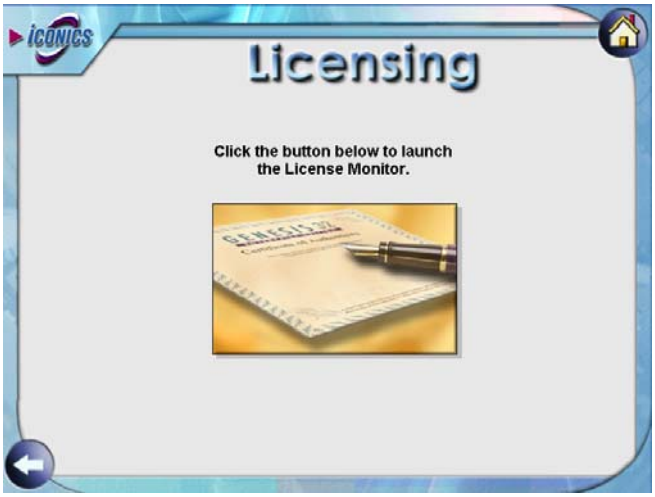

Figure D.14: Launching the License Monitor

Click the **Show License** button shown in **Figure D.14** to launch the **License Monitor,** shown in **Figure D.15.** You will be provided with information about the Client Units and points used to run specific GENESIS32 applications.

The **Client Units** field lists the number of Client Units that are currently being used in all GENESIS32 applications. The number of **Available** Client Units (which are not currently in use), and the **Total** number of Client Units (the number of available Client Units plus the number of Client Units) in use are also listed. For example, if you have 700 total Client Units and 300 Clients Units are currently being used, then

400 Client Units are available. Similarly, the Points field provides information about the number of OPC points that are currently being used (in runtime mode) in all GENESIS32 applications.

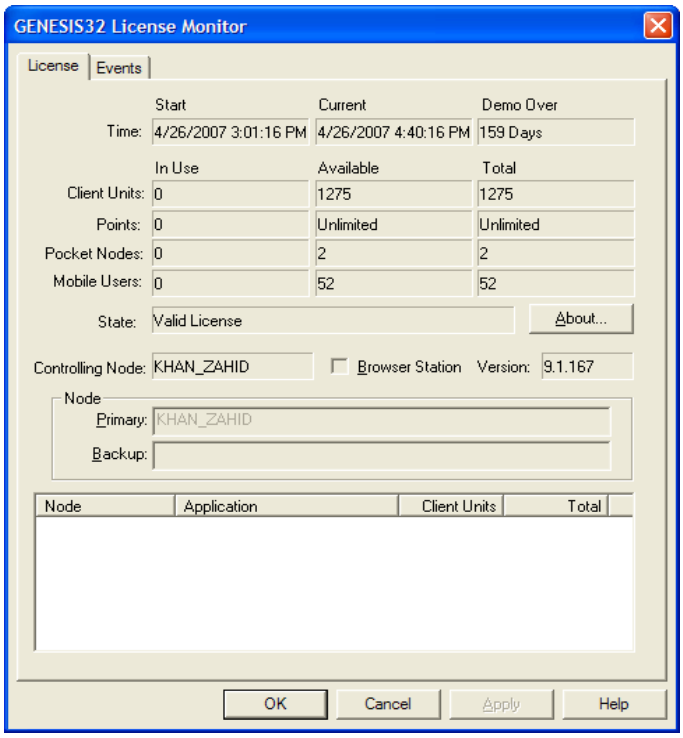

Figure D.15: GENESIS32 License Monitor

D.9 License for the WebHMI Server

Before installing WebHMI you must have a licensed GENESIS32 enterprise installed on the designated WebHMI server. Likewise, the ICONICS Software License Utility must also be installed on the WebHMI server.

Otherwise, the Software License Utility and its installation are the same for WebHMI as they are for any other GENESIS32 application. The License Utility will show a check next to the WebHMI option when a valid license covers WebHMI.

D.10 Obtaining Additional Connection Units

If you desire additional client units so that more clients can run applications at a time, you will need additional licensing for the WebHMI server. To obtain additional units, contact your local ICONICS distributor. For distributor telephone numbers, please call 508-543-8600.

Visualize Your Enterprise™

100 Foxborough Blvd., Foxborough, MA 02035
508.543.8600 • FAX: 508.543.1503
www.iconics.com • E-mail: info@iconics.com

© 2011 ICONICS, Inc. All Rights Reserved.
Specifications subject to change without notice.