



ICONICS, Inc.
100 Foxborough Blvd.
Foxborough, MA 02035
Phone: (508) 543-8600
Fax: (508) 543-1503
[Worldwide Support Centers](#)
www.iconics.com



© ICONICS, Inc. All rights reserved.

Specifications are subject to change without notice. GENESIS, GENESIS32, Pocket GENESIS, BizViz and their respective modules, OPC-To-The-Core, and Visualize Your Enterprise are trademarks of ICONICS, Inc. Other product and company names mentioned herein may be trademarks of their respective owners.

Starting the DataWorX Configuration 1

Introduction to DataWorX32

DataWorX32 is a 32-bit, multithreaded, OPC-compliant client and server application providing multiple functionality. DataWorX32 is a component of the GENESIS32 product family, and it serves as a project-level data system for GENESIS32 applications. Acting as a bridge between various OPC servers, DataWorX32 provides different OPC data channels. Once multiple I/O channels are established between PCs, DataWorX32 will switch between a primary PC (node) and a backup PC on the network. Should the primary PC be disabled, DataWorX32 will automatically (should the options be specified) default to the backup PC and vice versa. Another feature of DataWorX32 is the use of global variables that are accessible from multiple clients.

DataWorX32 is offered in two license versions:

- **DataWorX32 Standard Edition** with OPC Tunneling
- **DataWorX32 Professional Edition** with Redundancy for mission-critical applications

DataWorX32 Standard Edition

DataWorX32 Standard Edition, offered as a stand-alone installation on the GENESIS32 Options CD, features **OPC Tunneling**. This connects any remote OPC server to a client in a robust and secure manner, allowing for OPC server real-time data to be redirected to more than one location. The easy-to-use graphical configuration interface allows for a centralized place to manage all remote connections.

DataWorX32 Professional Edition

Offered in the **DataWorX32 Professional Edition** is the new DataWorX32 full-featured redundancy, which provides high availability similar to features found in large Distributed Control Systems. DataWorX32 Redundancy is the only product to support the three most important OPC standards, increasing the reliability and availability of OPC data by allowing multiple OPC servers to be configured into redundant pairs. These redundant OPC server pairs seamlessly appear as a single OPC server to any OPC client application. This feature can be added to an existing OPC server/client application, without any reconfiguration of those applications, keeping your processes going without any down time. **DataWorX32 Professional Edition** is installed as part of the full GENESIS32 Automation Suite installation.

Installing DataWorX32

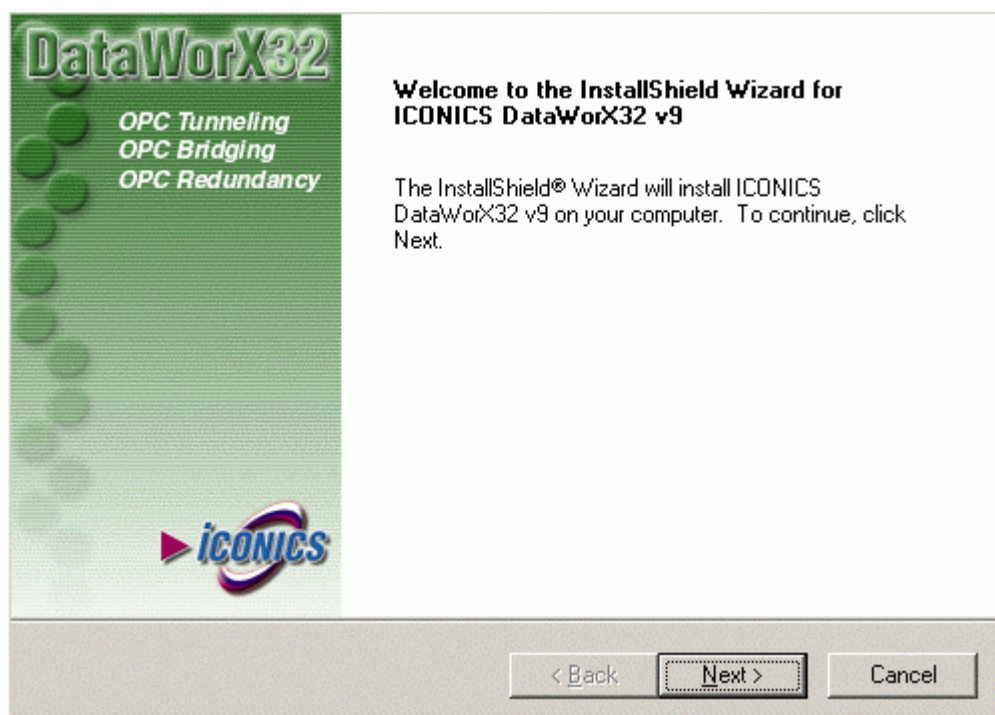
DataWorX32 can be installed as a stand-alone component or as part of the full GENESIS32 Automation Suite installation. **DataWorX32 Standard Edition** is offered as a stand-alone installation on the GENESIS32 Options CD. **DataWorX32 Professional Edition** is installed as part of the full GENESIS32 Automation Suite installation.

Note: For information about installing GENESIS32, please refer to the CDReadme on the GENESIS32 CD or the GENESIS32 Getting Started Help documentation.

The following steps detail the **DataWorX32 Standard Edition** stand-alone installation:

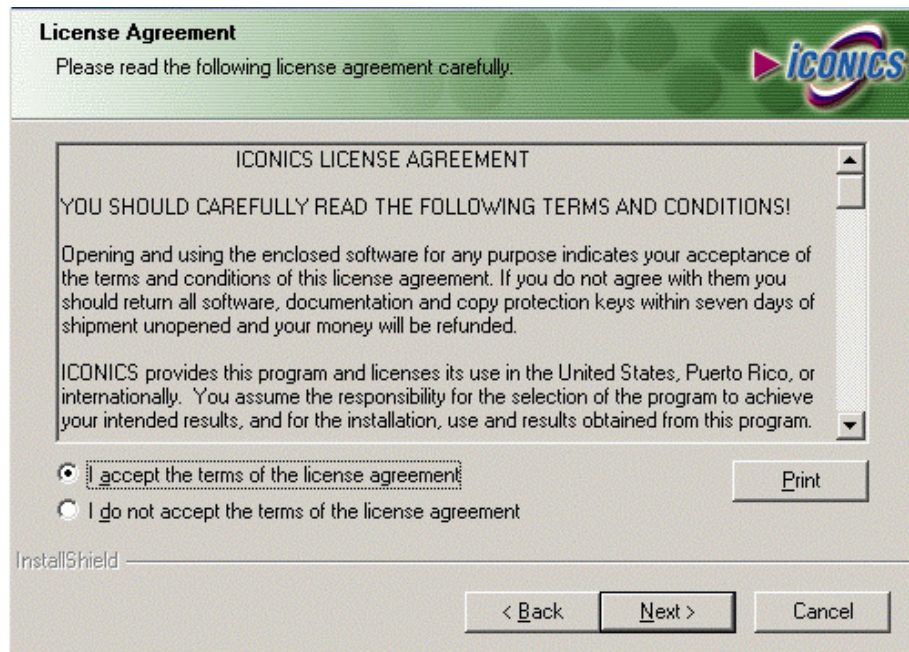
Note: If your operating system (e.g., Windows 2000) requires a login name, you must log in with administrator capability before installing DataWorX32.

1. Before installing the software, be sure that all other applications are closed and/or disabled.
2. Run the DataWorX32 installation from the GENESIS32 Options CD. The **Welcome** screen appears. Click the **Next** button to continue.



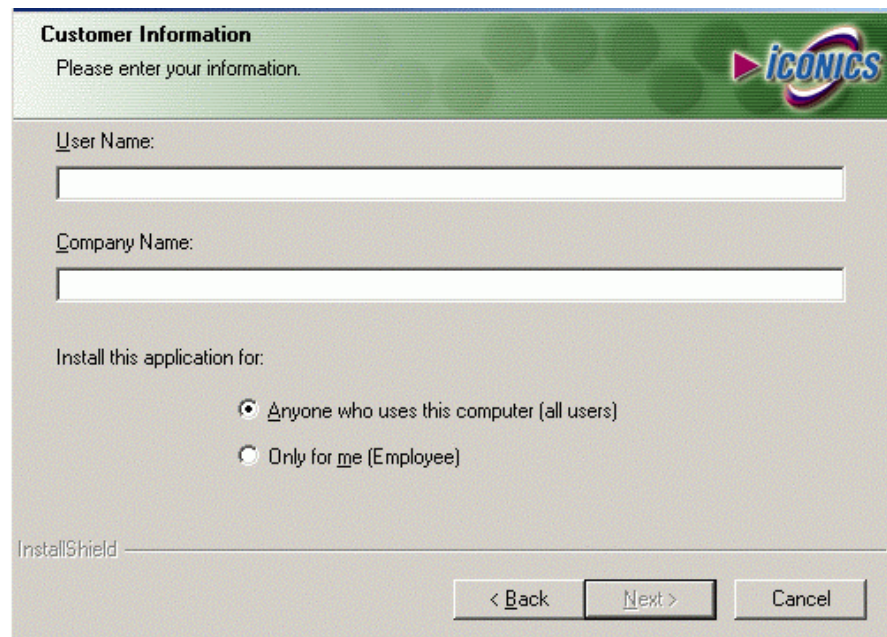
Welcome Screen

3. The **Software License Agreement** dialog box appears. Read the License Agreement. Click **Next** if you accept the terms of the agreement.



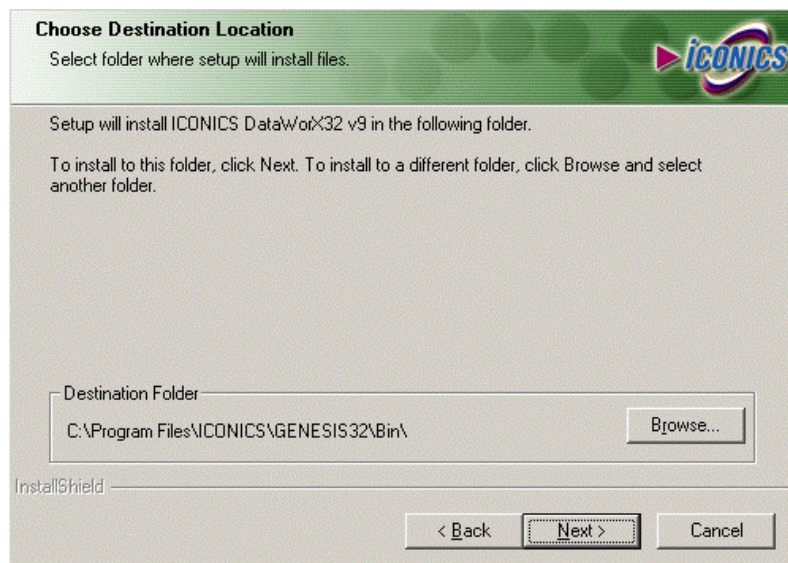
License Agreement

4. The **Customer Information** dialog box appears. Enter your **User Name** and **Company Name**. We recommend that you select to install the application for **Anyone who uses this computer (all users)**. Click the **Next** button to continue the installation.



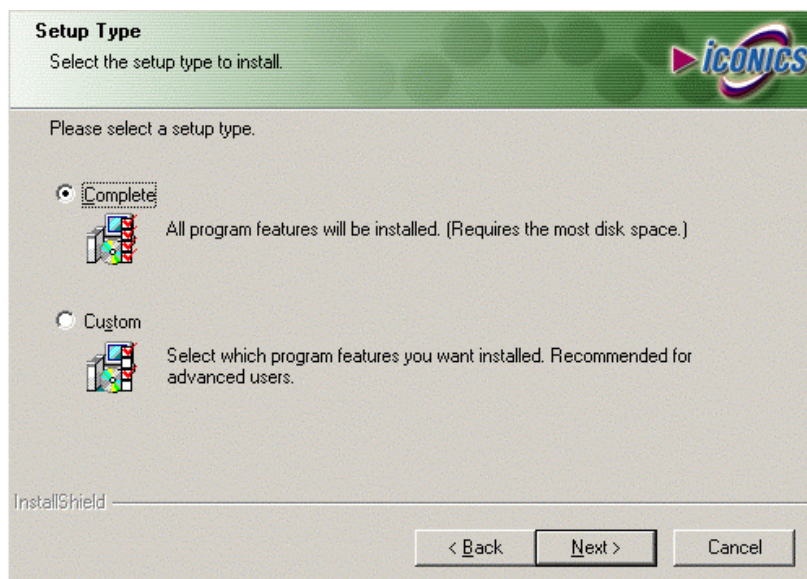
Customer Information

5. The **Choose Destination Location** dialog box appears, as shown in the figure below. By default, the directory for the GraphWorX installation is "..\Program Files\ICONICS\GENESIS32\Bin." You can also click the **Browse** button to specify a different installation folder. Click the **Next** button to continue the installation.



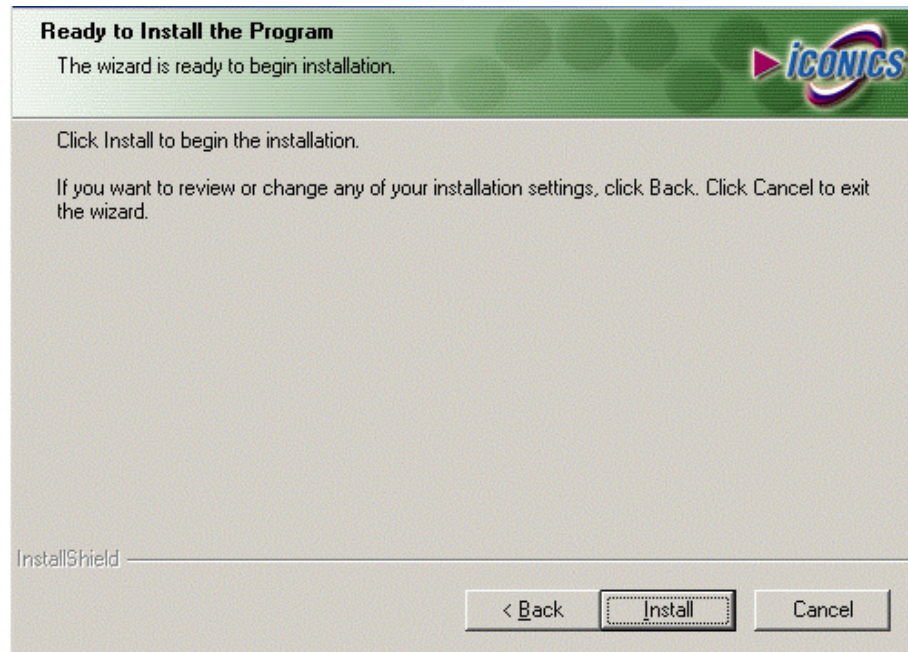
Choose Destination Location

6. The **Setup Type** dialog box appears, as shown in the figure below. Select either **Complete** or **Custom** installation. If you choose **Complete**, all program features will be installed. If you choose **Custom**, you can select which program features you want to install and where they will be installed. Click the **Next** button to continue the installation.



Setup Type

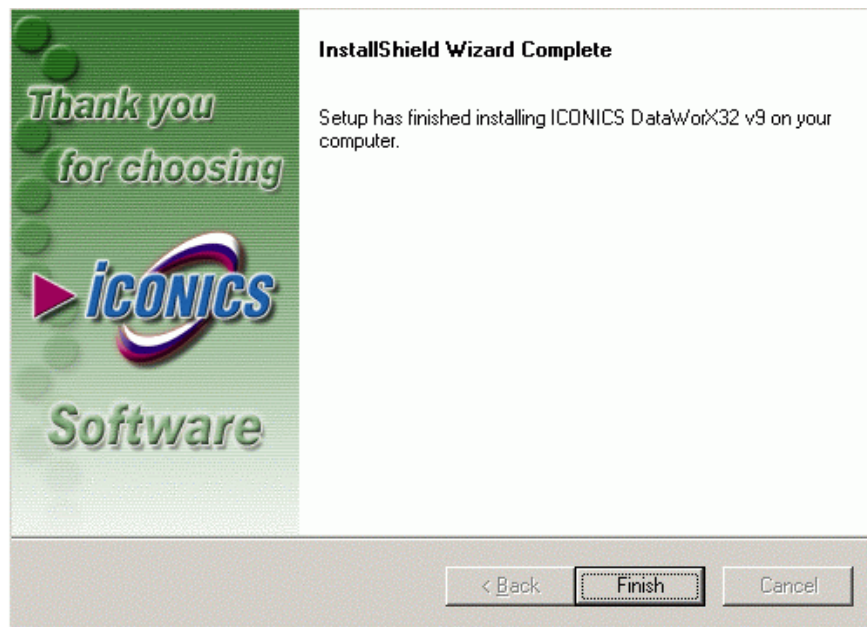
- Click the **Install** button again to proceed with installation, as shown in the figure below.



Confirming Installation

- After all files and components have been installed, click the **Finish** button to complete the installation.

Note: It is recommended that you restart your computer before using the product.



Installation Completed

Software Licensing Protection Necessary for Runtime Mode

DataWorX32 is installable as a stand-alone COM component. It registers itself with the GENESIS32 license protection mechanisms. DataWorX32 permits unlimited configuration capabilities with or without the software license protection present. It requires software license protection to work in runtime mode. If such licensing is not present, DataWorX32 runs for two hours in runtime mode before timing out.

DataWorX32 Interface With the GENESIS32 Security Application

From within the security configuration application, you can restrict access to the various features and/or data available via DataWorX32.

DataWorX32 supports multiple languages. That is, menu items, forms and various other text items are available for translation into other international languages through the GENESIS32 Language Configurator.

Key DataWorX32 Capabilities

DataWorX32 is an OPC-compliant server and client that conforms to Microsoft COM/DCOM program practice. The main features of DataWorX32 include:

- OPC data-bridging and sharing between OPC servers.
- OPC server redundancy with auto switchover.
- OPC data aggregation for better performance and more efficient data management.
- Global variables for OPC aliasing and expressions.
- OPC Tunneling connects remote OPC servers to a local computer and also supports third-party OPC server-to-OPC client communications.
- New OPC Redundancy enables DataWorX32 to be used with any OPC server, adding real-time OPC data management and redundancy to any OPC application

Features of DataWorX32

The main features of DataWorX32 include the following:

- Serves as an OPC data bridge between OPC servers.
- OPC server redundancy with auto-switchover performance.
- OPC client request aggregation.
- Global variables.
- Register grouping and organization.
- Can be run as a service.
- OPC interface on both the client side and server side.
- Switch alias functionality.
- OLE automation interface.

- Conditions that can be used as register inputs.
- Time stamp and quality to points accessed via automation.
- Output to NT Event Logger.
- Support for GenBroker OPC over TCP/IP and SOAP/XML communications.
- Filter support.
- Statistics and performance analyzer.
- Startup with a specified file.
- Primary node status register.
- CSV file import and export.
- XML file import and export.
- Runtime command line option.
- Global aliasing support for OPC inputs and outputs.
- Changes made to DataWorX32 registers that are made together (in a single call to the OPC interface) are also made together. OPC clients connected to DataWorX32 are notified about all the changes in a single update.
- NATIVE register type for array support.
- Optional refresh mechanisms.
- Integrated Expression Editor.
- Disable OPC propagation support.
- New way of registering running instances.
- New OPC Tunneling for secure, transparent tunneling of OPC data between any networked endpoints.
- New OPC Redundancy enables DataWorX32 to be used with any OPC server, adding real-time OPC data management and redundancy to any OPC application.

DataWorX32 As an "OPC Bridge"

Customers sometimes need to pass information from one device on to another. For example, data from one brand of PLC need to be shared with another I/O device. In the past, users had to write their own programs to translate and move data from one server to another. DataWorX32 provides this server-to-server data exchange, serving as an "OPC bridge" between two or more servers.

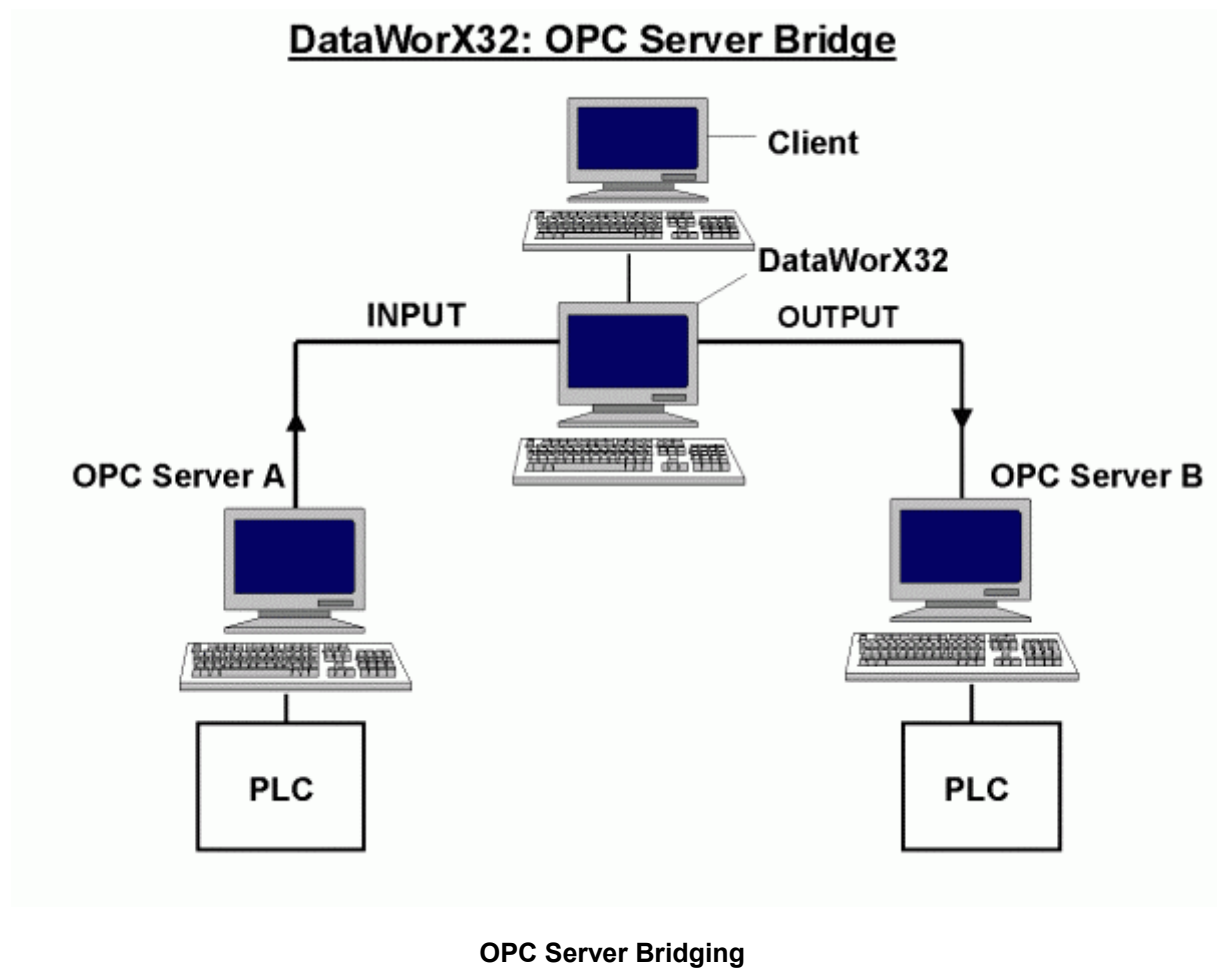
OPC bridging is a unique and powerful feature of DataWorX32. As its name suggests, its purpose is to provide "bridging" between OPC servers of various types.

Should you be required to scan input from one OPC server and supply it to another OPC server, you may "perform" expressions before sending the output to the other server via DataWorX32.

OPC bridging in DataWorX32 involves the following:

- A register is defined in the DataWorX32 Configurator.
- The input and output for the register are defined in the register settings.
- This register is also available to other clients while, depending on the options specified for the register input, DataWorX32 continues to write values to output tags.
- There can be more than one output tag. This way DataWorX32 performs "bridging" between OPC servers by simultaneously reading values from one or more OPC servers and writing to one or more OPC servers.
- No scripting or programming is required for bridging.

The figure below illustrates the functionality of DataWorX32 as an OPC bridge between clients and servers. For another example of bridging, open the **DataWorX32_Bridging.gdf** display in the "ICONICS\GENESIS-32\Examples\Gen32Demo" directory.

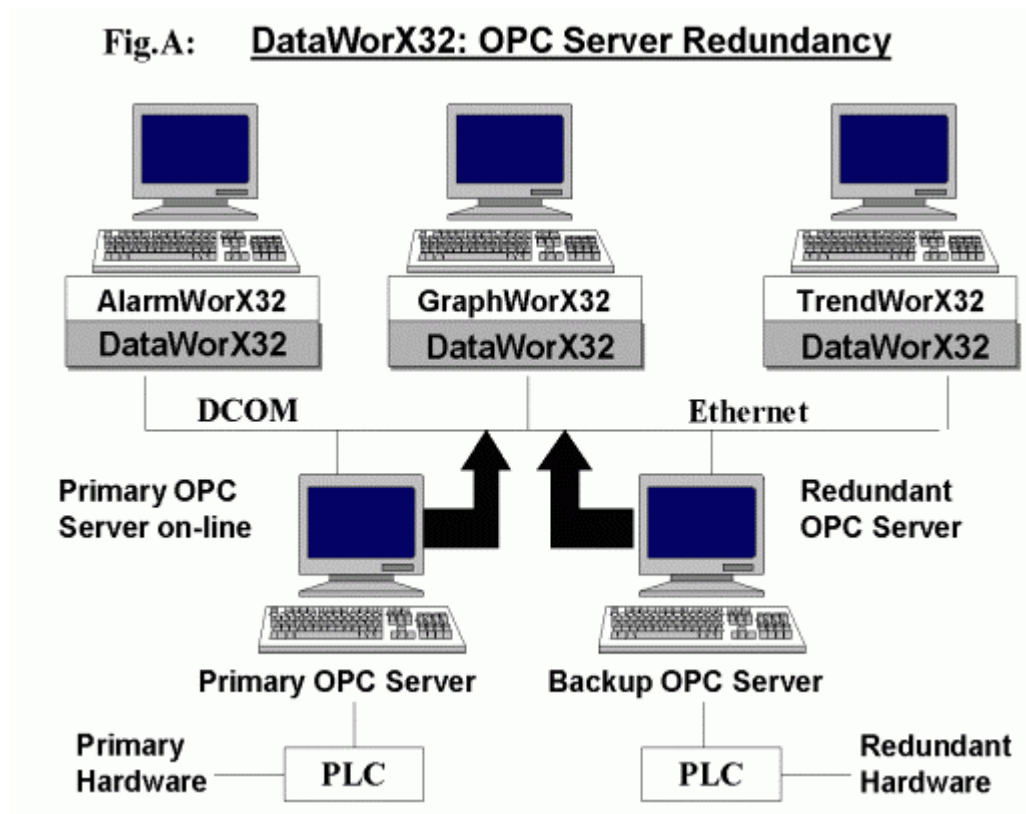


DataWorX32 OPC Server Redundancy

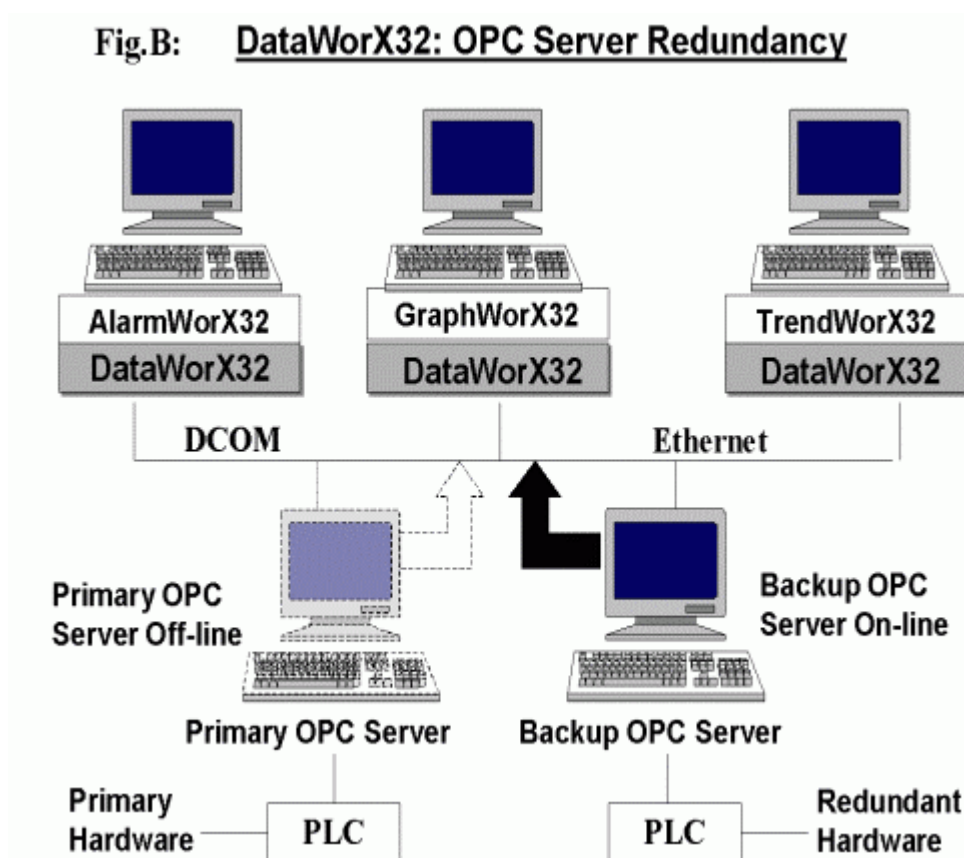
DataWorX32 provides 100 percent OPC server redundancy using OPC servers to any OPC client through the network. This means that users can designate alternative machines as backup servers should a designated Primary server go offline.

DataWorX32 scans the OPC server status and switches to the Backup node in case of Primary node failure. This means that, once a Primary server does go offline, DataWorX32 will default to the Backup server or servers in the order in which the backup servers were designated. A special digital tag is provided to start events in case of a switchover from the Backup to the Primary server. If the **Automatic Switch Back to Primary Server** option is selected in the **Redundant Server Configuration** dialog box, DataWorX32 will default to the Primary server once it returns online.

Figure A and Figure B shown below illustrate DataWorX32 OPC Server Redundancy. For another example of redundancy, open the **DataWorX32_Redundancy.gdf** display in the "ICONICS/GENESIS-32/Examples/GEN32DEMO" directory.



Redundancy When the Primary OPC Server Is Online

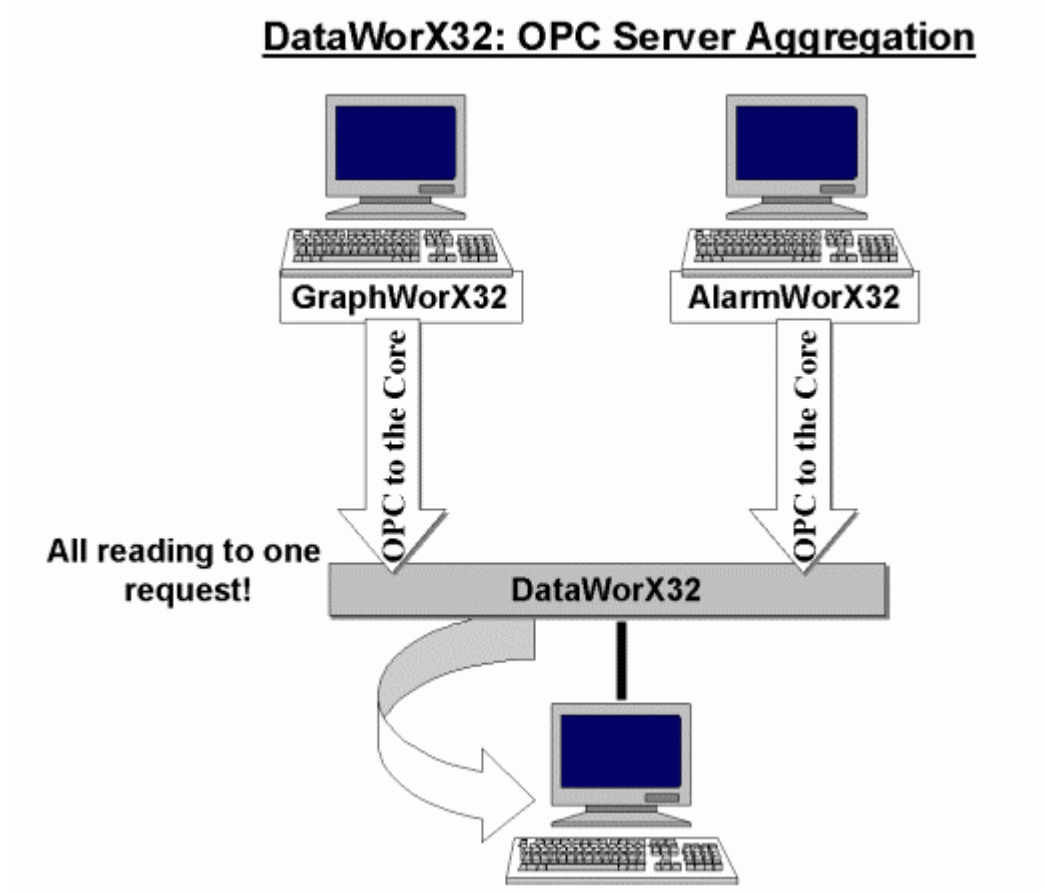
Fig.B: DataWorX32: OPC Server Redundancy**Redundancy When the Primary OPC Server Is Offline**

1. You must designate one OPC server as the "Primary" server in each set.
2. You may designate one or more OPC servers as the "Backup" servers in each set. (This number of servers is not restricted by DataWorX32 itself; rather it is limited only by system resources). If more than one backup server is specified, they should be ordered (2nd, 3rd, 4th, etc.) You will see a message outlining the details of the discrepancies, and will be allowed to either accept it as is or permit reconfiguration.
3. The various OPC client applications request data from DataWorX32, rather than from the OPC server directly. This way if a Primary OPC server failure occurs (due to any number of conditions), an automatic switchover to the Backup OPC server occurs.

DataWorX32 OPC Aggregation

Often in very large projects, several OPC client applications request the same points from an OPC server. For example, GraphWorX32 may need to display a tank level value, and AlarmWorX32 may need to monitor and alarm that same value. This may increase the load of the OPC server, as it now has to provide the same data more than once. Thus, when multiple clients request data from an OPC server, DataWorX32 monitors the OPC server and aggregates the data to the requesting clients.

Often it is desirable to optimize the work performed by the lower-level I/O servers (for example, greater throughput can be achieved). DataWorX32 can serve as a "middle-man" between clients and servers and assist in this optimization process. This is beneficial especially with remote servers over the network. The figure below illustrates the DataWorX32 OPC client request aggregation. For another example of aggregation, open the **DataWorX32_Aggregation.gdf** display in the "ICONICS/GENESIS-32/Examples/GEN32DEMO" directory.



OPC Server Aggregation

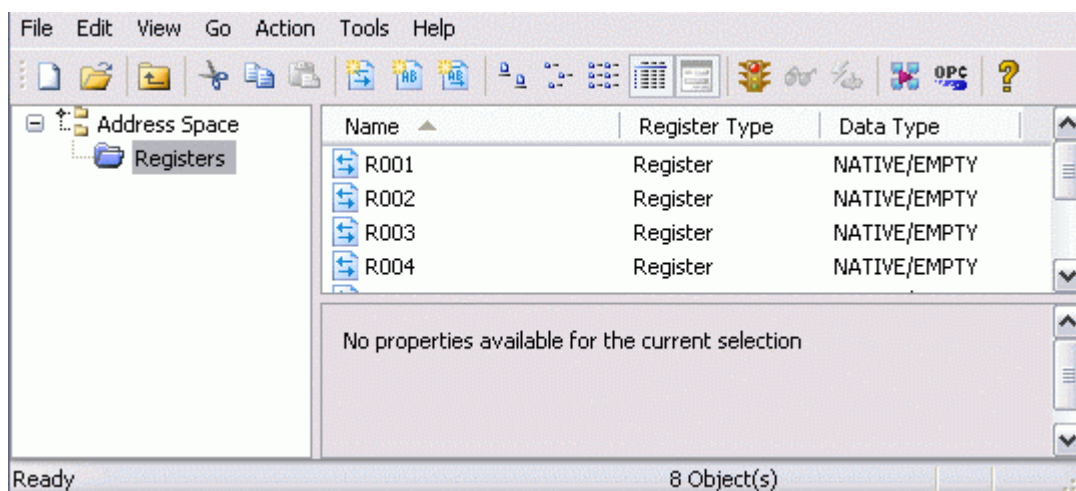
DataWorX32 Global Variables

Many clients require a common variable to share values. DataWorX32 makes it possible to define many variables that are accessible to many clients simultaneously. The DataWorX32 registers can be used as global variables. These variables can also act as aliases for clients. You can create holding registers, data filters, global registers, and switches for your entire project, as well as define data types, ranges, read/write status, and more.

Starting the DataWorX32 Configurator

To start the DataWorX32 Configurator:

1. From the Windows **Start** menu, select **Programs > ICONICS GENESIS32 > DataWorX32 > DataWorX32**.
2. This opens the DataWorX32 Configurator, as shown in the figure below. The screen consists of a split window with a tree control view in the left-hand pane and a configuration view in the right-hand pane. The Configurator provides a standard format for the configuration database, as well as a sample (default) DataWorX32 configuration project. The Configurator also includes a toolbar and menus with many command functions.

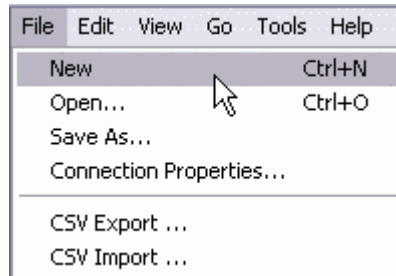


DataWorX32 Configurator Screen

Creating Configuration Databases

The Configurator provides a Configuration Database Wizard for creating new Microsoft Access and SQL Server configuration databases. To create a new configuration database in the Configurator:

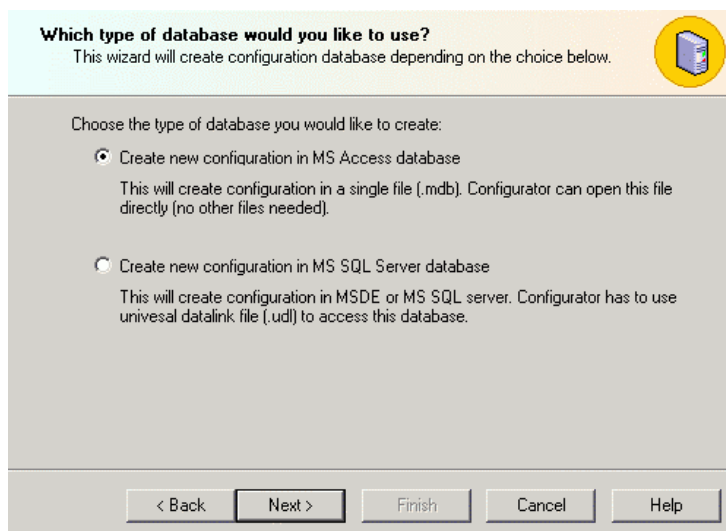
1. Select **New** from the **File** menu, as shown in the figure below.



Creating a Configuration Database

2. The introduction screen for the Configuration Database Wizard appears. Click the **Next** button to continue.
3. You have two options for creating your new database, as shown in the figure below:
 - **Create a new Microsoft Access configuration database:** For a Microsoft Access database, the Configurator uses a single .mdb file.
 - **Create a Microsoft SQL Server database:** The Configurator uses Universal Data Link (.udl) files to connect to a Microsoft SQL Server or MSDE database.

Select the database type you want to create, and then click **Next**.

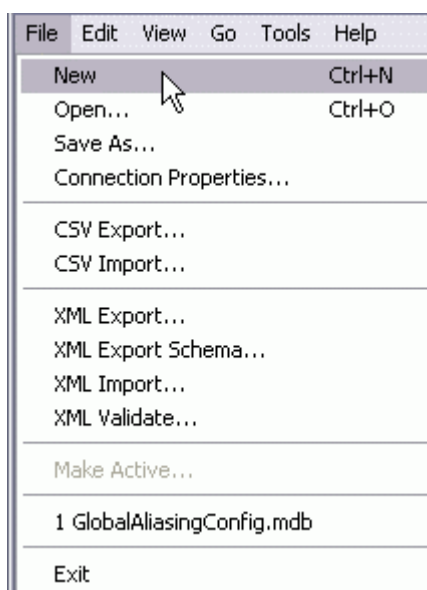


Choosing the Type of Database To Create

Creating a Microsoft Access Configuration Database

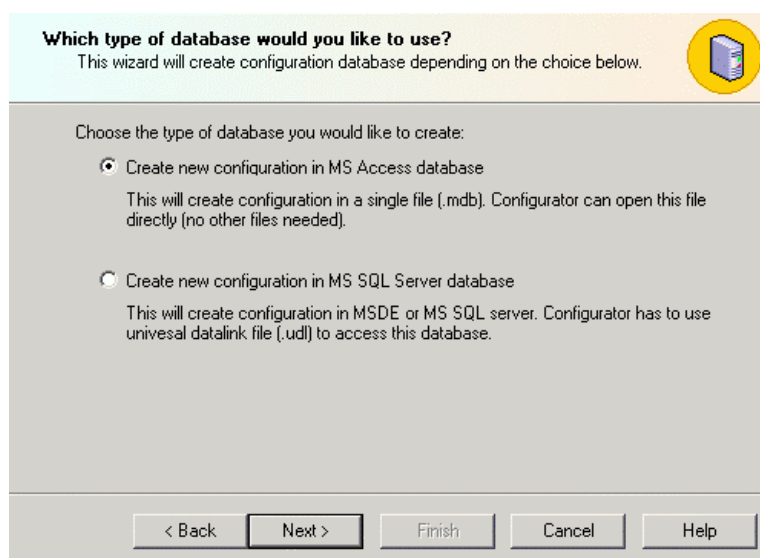
To create a new Microsoft Access configuration database in the Configurator:

1. Select **New** from the **File** menu, as shown in the figure below.



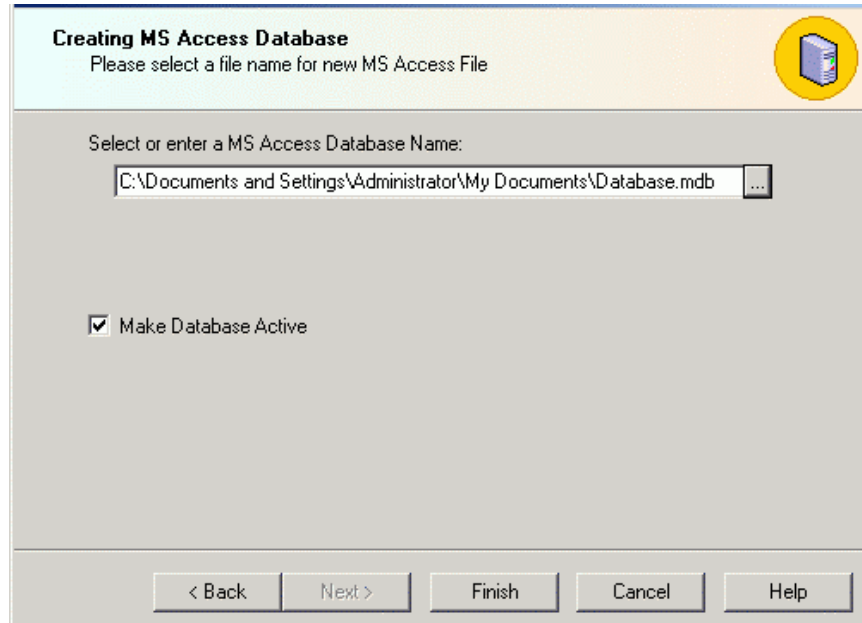
Creating a Configuration Database

2. The introduction screen for the Configuration Database Wizard appears. Click the **Next** button to continue.
3. You have two options for creating your new database, as shown in the figure below. Select **Create new configuration in MS Access database**. Click the **Next** button to continue.



Creating a New Microsoft Access Database

- Specify the directory path and file name for the new database, as shown in the figure below. Click the ... button to browse for a directory. If you want this new database to be the active configuration database, check **Make Database Active**.



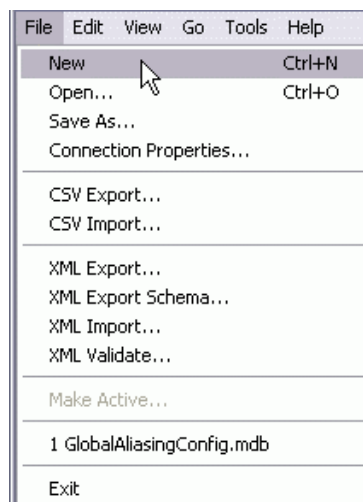
Naming the New Access Database

- Click the **Finish** button. The new database is created and opened in the Configurator.

Creating a Microsoft SQL Server Configuration Database

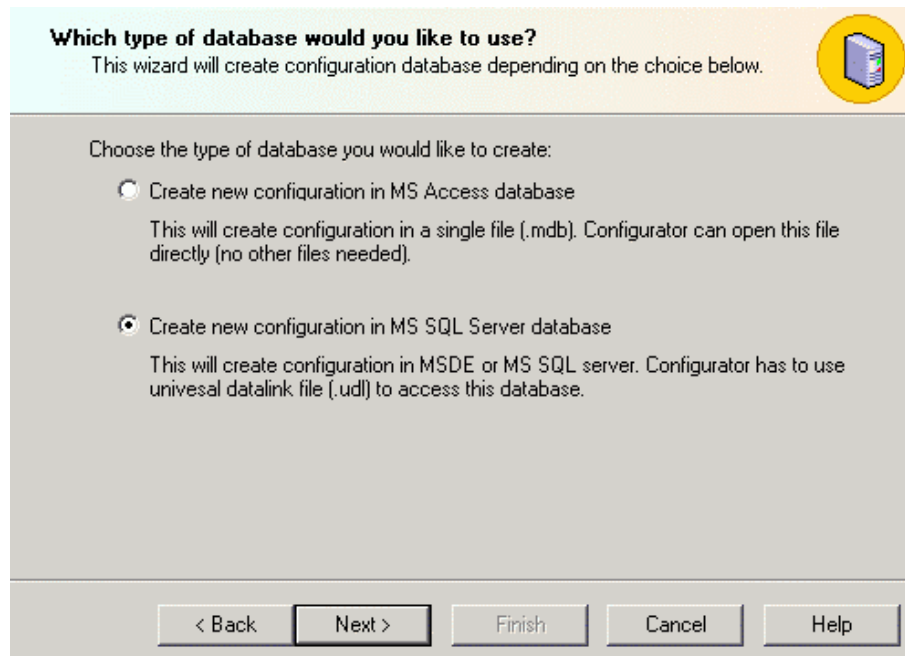
To create a new Microsoft SQL Server configuration database in the Configurator:

- Select **New** from the **File** menu, as shown in the figure below.



Creating a Configuration Database

2. The introduction screen for the Configuration Database Wizard appears. Click the **Next** button to continue.
3. You have two options for creating your new database, as shown in the figure below. Select **Create new configuration in MS SQL Server database**. Click the **Next** button to continue.

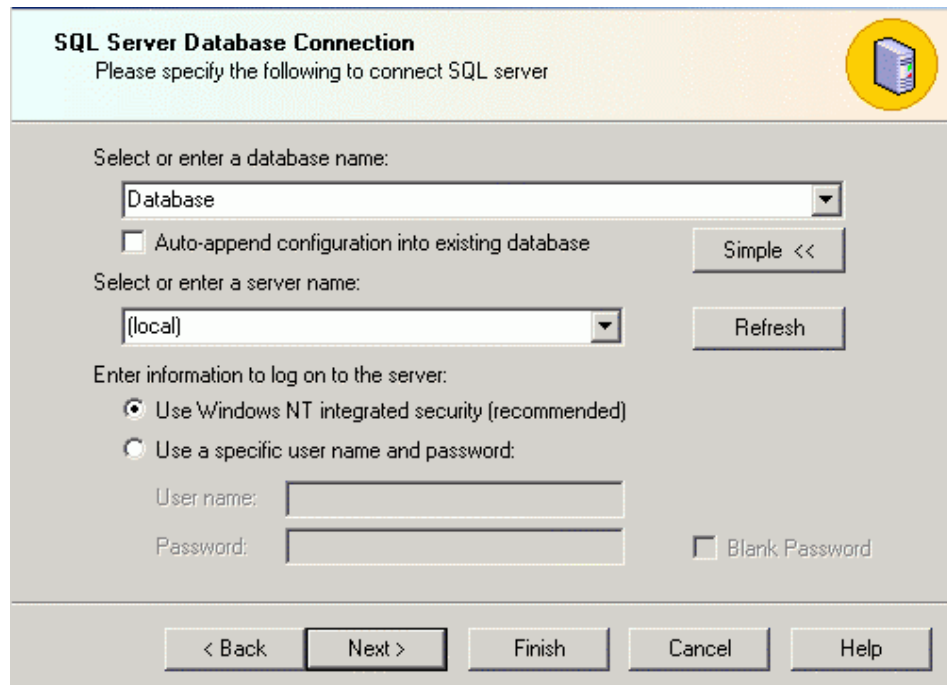


Creating a New Microsoft SQL Server Database

4. To connect to a SQL Server database, either select an existing database from the **Database Name** drop-down list, or type a new name to create a new database, as shown in the figure below. If you choose to use an existing database, you have the option of adding (auto-appending) the configuration to the existing database. (For more information about the auto-append function, please see "Adding the Configuration to an Existing Database.")

In the **SQL Server Name** field, select the local SQL Server on which to create the database. If necessary, enter a user name and password to log on to the SQL Server. (It is recommended that you use Windows NT integrated security.)

Note: Usually you have only one instance of SQL Server running on the local node. In this case, the drop-down list under **SQL Server Name** has only one option: "(local)." However, it is possible to run multiple SQL Server instances on the local node, in which case the **SQL Server** field lists all those SQL Server instances: "(local)" for the default instance and "node_name/instance_name" for all others. The drop-down list may show SQL Server instances on other nodes as well.



SQL Server Database Connection
Please specify the following to connect SQL server

Select or enter a database name:

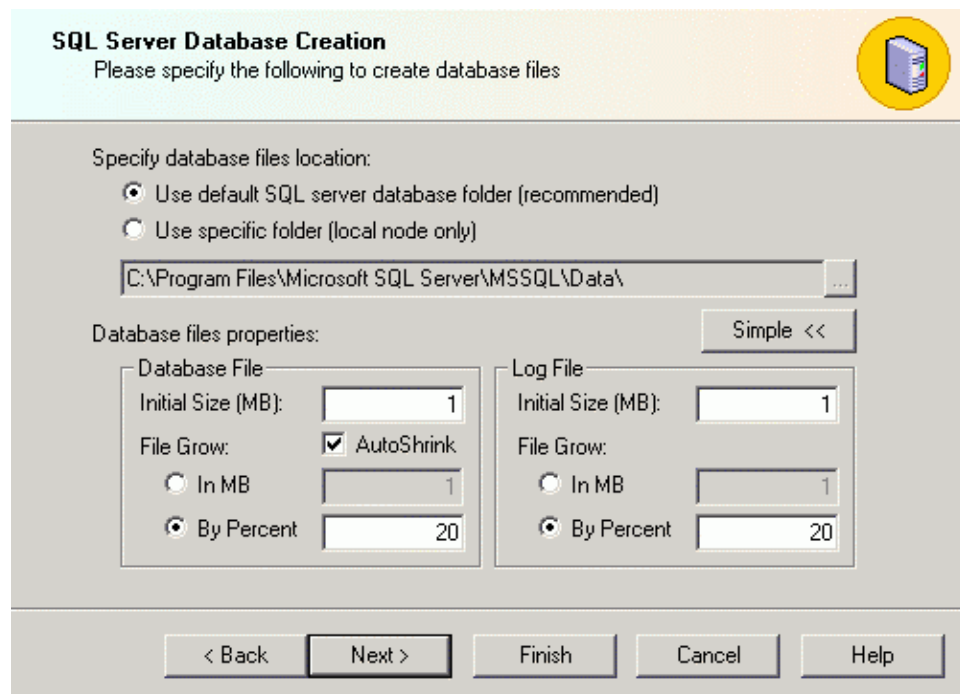
☐ Auto-append configuration into existing database

Select or enter a server name:

Enter information to log on to the server:
☒ Use Windows NT integrated security (recommended)
☐ Use a specific user name and password:
 User name:
 Password: ☐ Blank Password

Connecting to a SQL Server Database

- Specify a directory path location in which to create the database, as shown in the figure below. You can either use the default SQL Server database folder, or you can click the ... button and browse for a specific folder.



SQL Server Database Creation
Please specify the following to create database files

Specify database files location:
☒ Use default SQL server database folder (recommended)
☐ Use specific folder (local node only)

Database files properties:

Database File		Log File	
Initial Size (MB):	1	Initial Size (MB):	1
File Grow:	☒ AutoShrink	File Grow:	☐ AutoShrink
☐ In MB	1	☐ In MB	1
☒ By Percent	20	☒ By Percent	20

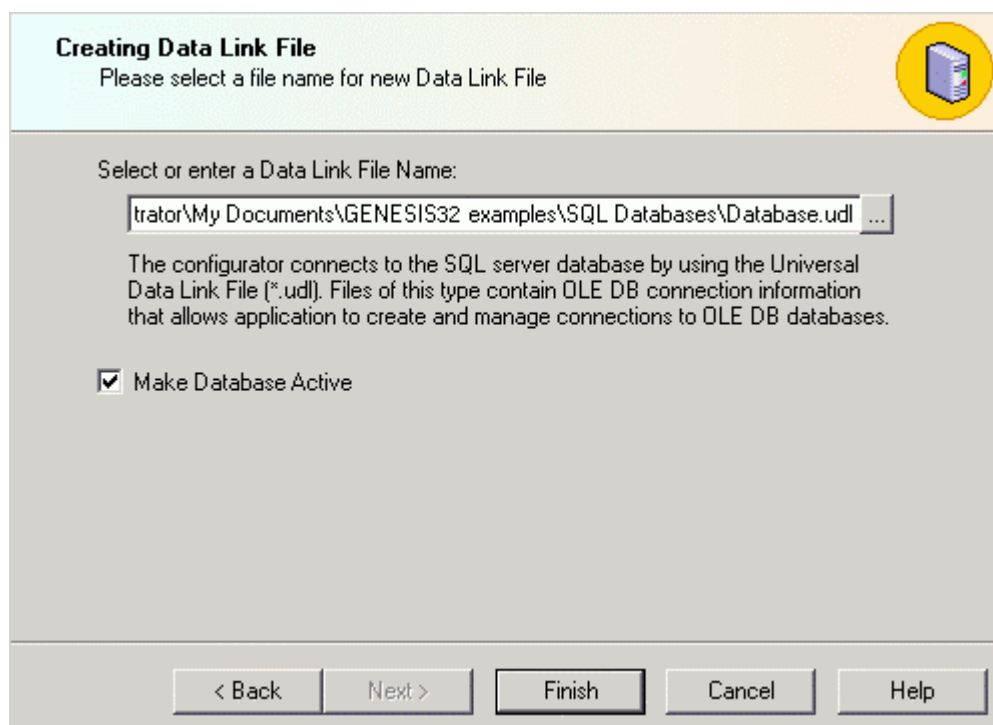
Specifying the Database Location and Properties

Under the **Database Properties** section, specify an initial size for the database, which should be as large as possible. You can also specify a **Database Growth** option (in megabytes) or as a percentage of the total size. MSDE servers are capable of growing the database on the fly to store more data. However, if this operation is performed frequently, the overall system performance may decrease. Choosing an initially large database size and a corresponding database growth option can drastically improve system performance.

Under the **Log File Properties** section, you can also modify the settings for the database transaction log file. Specify a **Log File Growth** option (in megabytes) or as a percentage of the total size. Again, a sufficient initial size setting can greatly improve performance. The default options should be adequate for most applications with a small to medium size load.

Click the **Next** button to create the new SQL Server database.

6. The Configurator uses Universal Data Link (.udl) files to connect to the Microsoft SQL Server database. These .udl files contain OLE database connection information that allows the Configurator to create and manage connections to OLE databases. Enter a name and directory path for the new .udl file in the **Data Link File Name** field, as shown in the figure below. You can browse for a directory by clicking the ... button to the right of this field. Click the **Next** button to continue. If you want this new database to be the active configuration database, check **Make Database Active**.



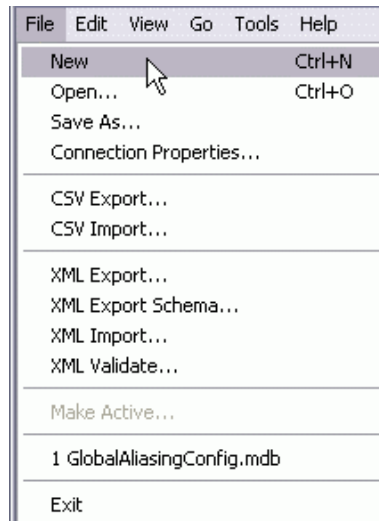
Creating a Universal Data Link File

7. Click the **Finish** button. The new database is created and opened in the Configurator.

Adding a Configuration Database to an Existing SQL Server Database

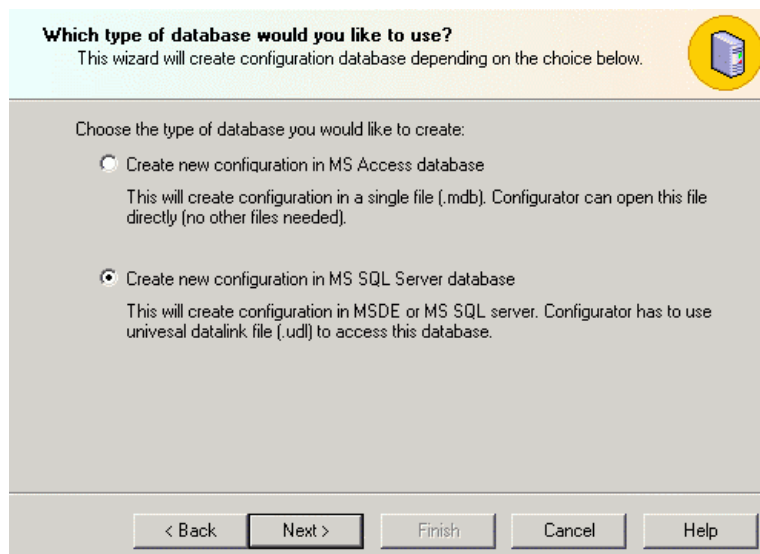
The Configuration Database Wizard also gives you the option to add the configuration database structure to an existing Microsoft SQL Server database:

1. Select **New** from the **File** menu, as shown in the figure below.



Creating a Configuration Database

2. The introduction screen for the Configuration Database Wizard appears. Click the **Next** button to continue.
3. You have two options for creating your new database, as shown in the figure below. Select **Create new configuration in MS SQL Server database**. Click the **Next** button to continue.



Creating a Microsoft SQL Server Database

4. To connect to a SQL Server database, select the existing database from the drop-down list, as shown in the figure below. Check the **Auto-append configuration into existing database** check box. In the **SQL Server** field, select the local SQL Server on which to create the database. Type a name for the database in the **Database Name** field. If necessary, enter a user name and password to log on to the SQL Server. (**Note:** It is recommended that you use Windows NT integrated security.) Click the **Next** button to continue.

Note: Usually you have only one instance of SQL Server running on the local node. In this case, the drop-down list under **SQL Server Name** has only one option: "(local)." However, it is possible to run multiple SQL Server instances on the local node, in which case the **SQL Server** field lists all those SQL Server instances: "(local)" for the default instance and "node_name/instance_name" for all others. The drop-down list may show SQL Server instances on other nodes as well.

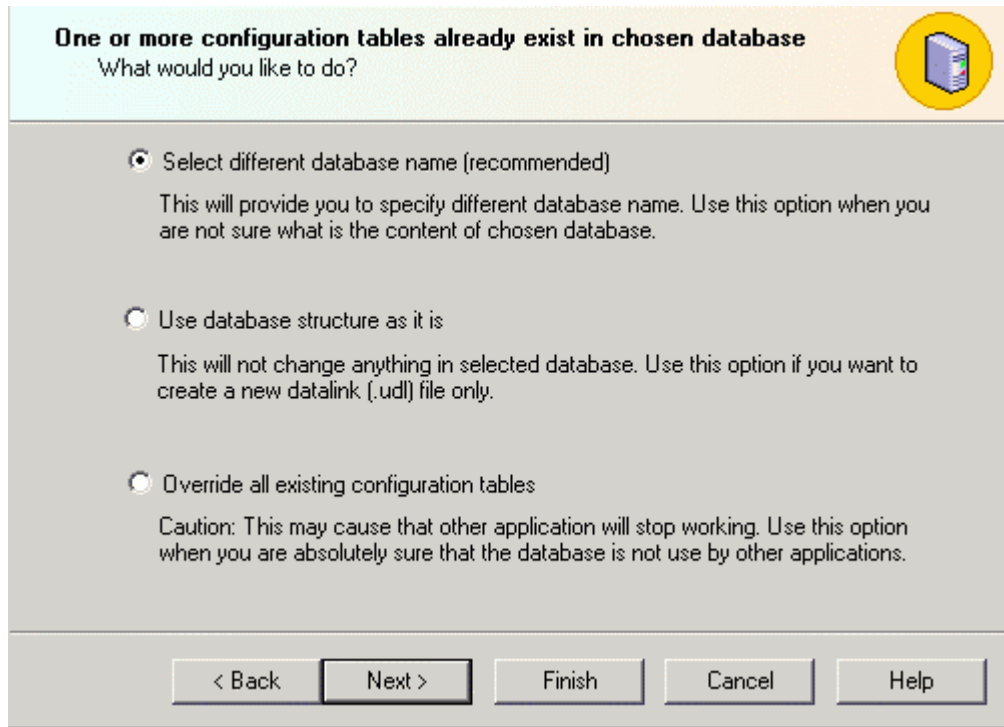
Connecting to an Existing SQL Server Database

5. If the existing database already has configuration tables, you have the following options, as shown in the figure below:
 - **Select a different database name (recommended):** This option allows you to rename the database without affecting the existing database as well as create a new Data Link (.udl) file.
 - **Use the existing database structure:** This option preserves the content of the existing database and allows you to create a new Universal Data Link (.udl) file.

- **Override all existing configuration tables:** This option expunges the content of the existing database and allows you to create a new Universal Data Link (.udl) file.

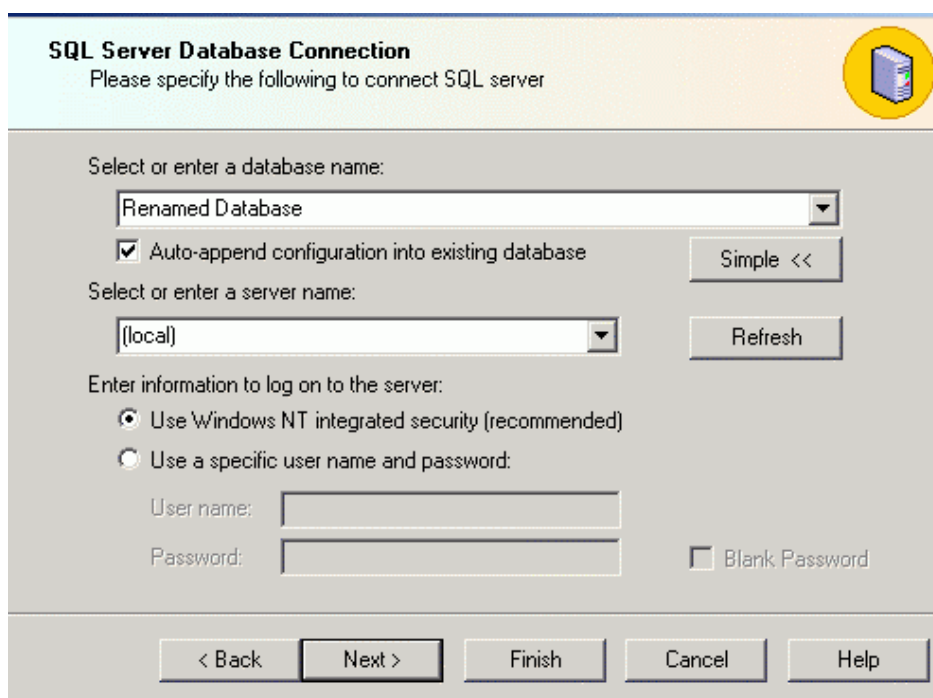
Note: Overriding the database may disable other applications that also use the database.

Click the **Next** button to continue.



Adding the Configuration to an Existing SQL Server Database

6. If you chose **Select a different database name (recommended)**, you are directed back to the SQL Server Database Connection dialog box, as shown in the figure below. Enter a new name for the database, and then click **Next**.



SQL Server Database Connection
Please specify the following to connect SQL server

Select or enter a database name:

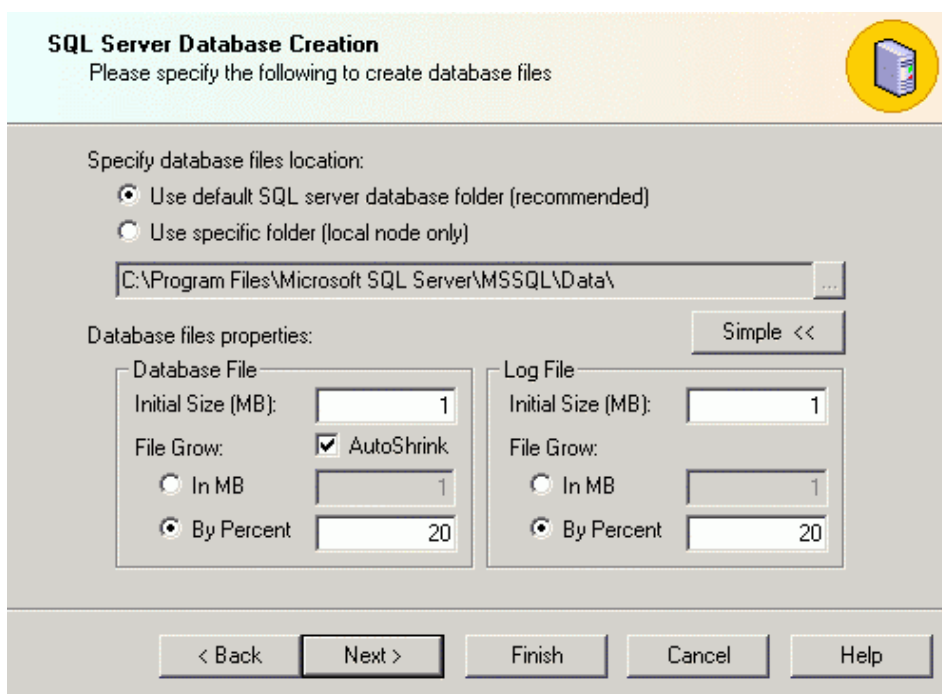
☒ Auto-append configuration into existing database

Select or enter a server name:

Enter information to log on to the server:
☒ Use Windows NT integrated security (recommended)
☐ Use a specific user name and password:
 User name:
 Password: ☐ Blank Password

Renaming the Existing SQL Server Database

- Specify a directory path location in which to create the database, as shown in the figure below. You can either use the default SQL Server database folder, or you can click the ... button and browse for a specific folder.



SQL Server Database Creation
Please specify the following to create database files

Specify database files location:
☒ Use default SQL server database folder (recommended)
☐ Use specific folder (local node only)

Database files properties:

Database File		Log File	
Initial Size (MB):	1	Initial Size (MB):	1
File Grow:	☒ AutoShrink	File Grow:	
☐ In MB	1	☐ In MB	1
☒ By Percent	20	☒ By Percent	20

Specifying the Database Location and Properties

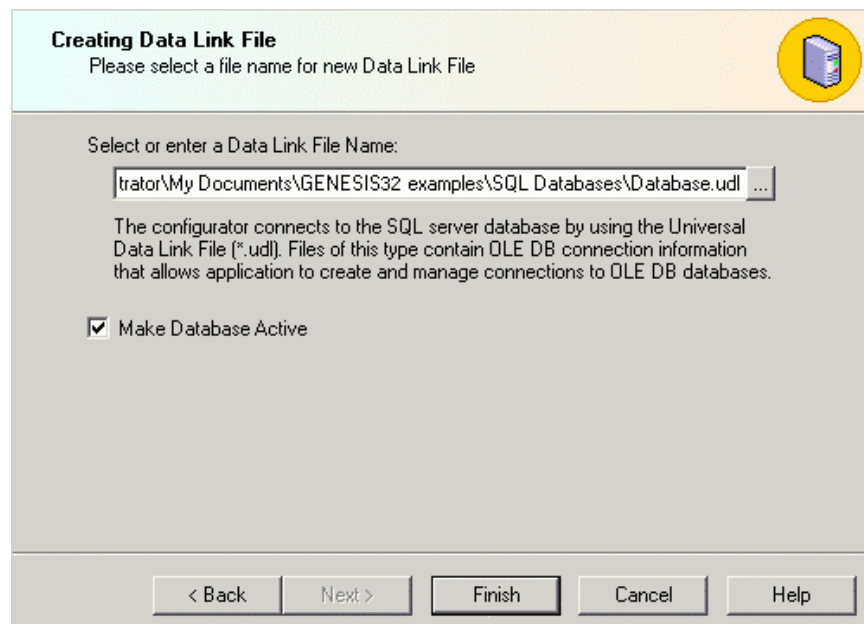
Under the **Database Properties** section, specify an initial size for the database, which should be as large as possible. You can also specify a **Database Growth** option (in megabytes) or as a percentage of the total size. MSDE servers are capable of growing the database on the fly to store more data. However, if this operation is performed frequently, the overall system performance may decrease. Choosing an initially large database size and a corresponding database growth option can drastically improve system performance.

Under the **Log File Properties** section, you can also modify the settings for the database transaction log file. Specify a **Log File Growth** option (in megabytes) or as a percentage of the total size. Again, a sufficient initial size setting can greatly improve performance. The default options should be adequate for most applications with a small to medium size load.

Click the **Next** button to create the new SQL Server database.

8. The Configurator uses Universal Data Link (.udl) files to connect to the Microsoft SQL Server database. These .udl files contain OLE database connection information that allows the Configurator to create and manage connections to OLE databases. Enter a name and directory path for the new .udl file in the **Data Link File Name** field, as shown in the figure below. You can browse for a directory by clicking the ... button to the right of this field. Click the **Next** button to continue. If you want this new database to be the active configuration database, check **Make Database Active**.

Note: If you chose to **Use the existing database structure** or to **Override all existing configuration tables**, you will still need to create a new Data Link file.



Creating a Universal Data Link File

9. Click the **Finish** button. The new database is created and opened in the Configurator.

Toolbar

The Configurator toolbar, shown below, contains the following command buttons. To show or hide the **Standard** toolbar, select **Toolbars > Standard Buttons** from the **View** menu. The **Standard** toolbar, shown below, contains the following command buttons.



New: Creates a new configuration database.

Open: Opens an existing configuration database.

Up One Level: Moves up one level in the tree control.

Cut: Deletes current selection, sending it to the clipboard.

Copy: Copies the current selection to the clipboard.

Paste: Pastes the current contents of the clipboard.

New Register: Creates a new register under the **Address Space** tree control.

New Alias: Creates a new alias under the **Address Space** tree control.

New Switch: Creates a new switch under the **Address Space** tree control.

Large Icons: Displays items as large icons.

Small Icons: Displays items as small icons.

List: Displays items as a list.

Details: Displays items as a list with details.

Dialog View: Displays additional configuration options.

Runtime: Starts/stops the DataWorX32 server.

Monitor View: Displays OPC data in a separate pane.

Update Runtime: Refreshes the DataWorX32 configuration during runtime.

GenBroker Configurator: Launches the GenBroker Configurator.

OPC Tunneling Configurator: Launches the OPC Tunneling Configurator.

About: Displays information about the application.

Menus

The menu bar of the Configurator contains the following menus:

- File
- Edit
- View
- Go
- Action
- Tools
- Help

Note: You can also access many of the menu commands by right-clicking items in the tree control of the Configurator and selecting command functions from the pop-up menus.

File Menu

The **File** menu commands are listed in the table below.

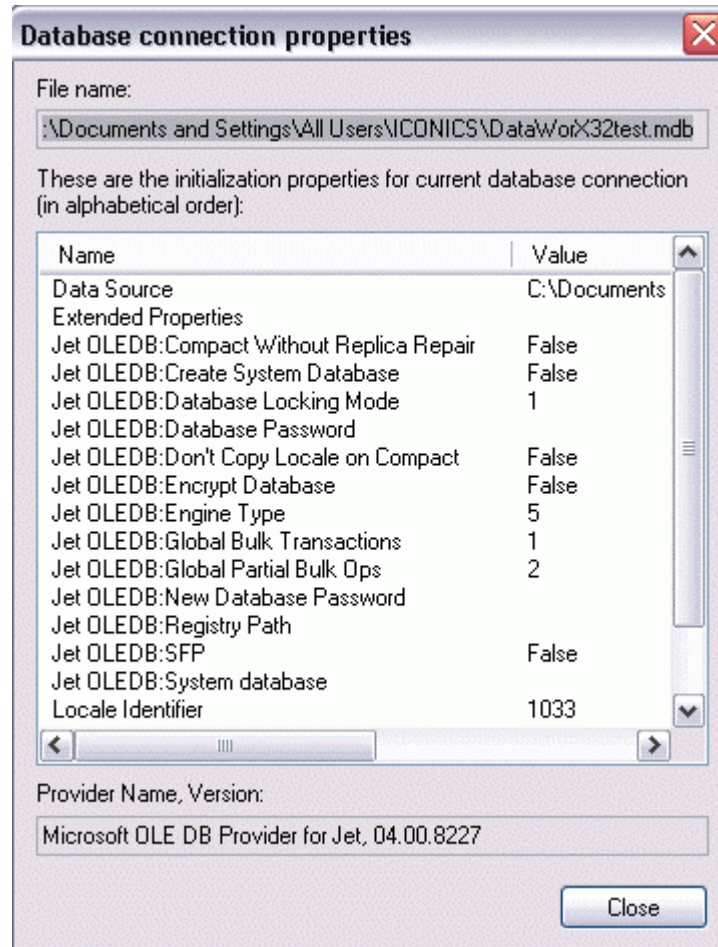
File Menu Commands

Command	Shortcut Keys	Function
New	CTRL+N	Creates a new configuration database.
Open	CTRL+O	Opens an existing configuration database.
Save As		Saves the current database under a different name as a Microsoft Access (.mdb) file.
Connection Properties		Displays the current database connection properties.
Export CSV		Exports configuration data from your database to a text file (.txt) or a Microsoft Excel file (.csv). You can specify the delimiters and what to export.
Import CSV		Imports data into your configuration database from a text file (.txt) or a Microsoft Excel file (.csv). You can then specify the delimiters and choose from the import settings.

XML Export		Exports configuration data to an XML file.
XML Export Schema		Exports configuration data to an XML Schema file.
XML Import		Imports configuration data from an XML file.
XML Validate		This feature does not import an XML data file, but it will try to validate its structure using stored XML schema. Once it passes this validation, the XML file is acceptable for import by the Configurator.
DataWorX Binary Export		Exports configuration data to an older version of DataWorX32 (.dwx files).
DataWorX Binary Import		Imports configuration data from an older version of DataWorX32 (.dwx files).
Make Active		Activates the current database. If this command is not available, then the current database is already the active database.
Update Runtime Configuration		Refreshes the DataWorX32 configuration during runtime.
Exit		Closes the application.

Database Connection Properties

Selecting **Connection Properties** from the **File** menu opens the **Database Connection Properties** dialog box, shown below, which lists the initialization properties for the current database connection.

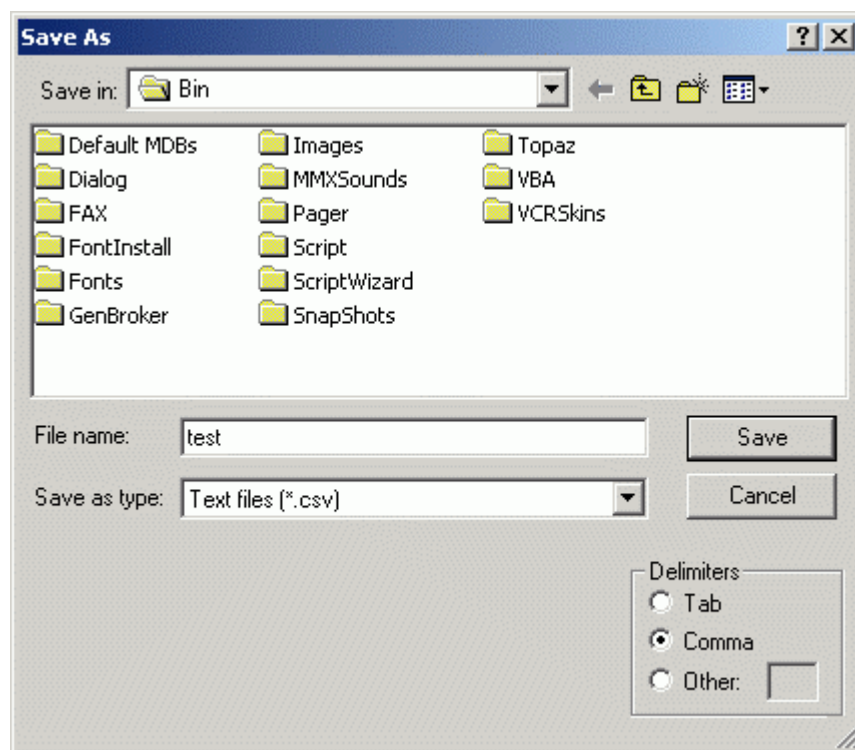


Database Connection Properties Dialog Box

Exporting Configuration Data

Exporting Data to a Text or CSV File

The Configurator offers the flexibility of exporting data from your configuration database to a text (.txt) file or a Microsoft Excel (.csv) file. To export data, select **Export CSV** from the **File** menu. This opens the **Export Configuration Data to File** dialog box, as shown in the figure below. You can then specify the delimiters for exporting the data. Unless you specify delimiters in the **Export Configuration Data to File** dialog box, the file uses **Commas** as delimiters by default. Each group contains headings and columns that provide information about each item, such as descriptions and associated translations and expressions. It also provides the "tree" pathway for each item. Choose the directory to which you want to export the data from your database. In the **Save As Type** field, choose the file type (.txt or .csv) that you would like to save.



Exporting Configuration Data

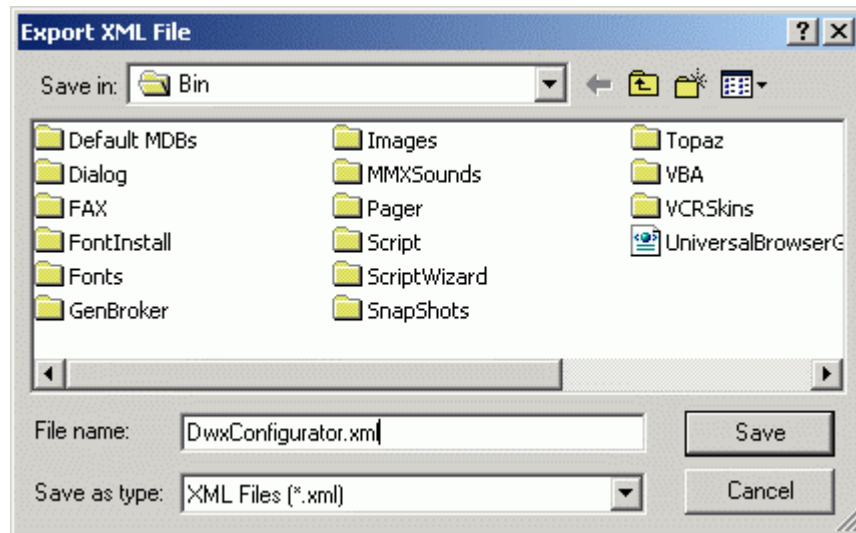
Exporting Data to an XML File

The Configurator also allows you to export data from your configuration database to an XML file. The XML export/import functionality was mainly developed for Windows platforms that do not support databases (e.g. Windows CE and Windows Embedded). XML has the following advantages over the CSV import/export function:

- XML has a standardized format, unlike the text/CSV format, which uses various delimiters (e.g. TAB instead of commas, strings could not accept all characters, etc.)
- XML is language-independent, whereas CSV converts date/time, floats, and currency fields according to local settings in Windows. For example, using CSV, you cannot export data on German Windows and import it on English Windows without making changes
- Windows has an installed automation object that has the capability to work with XML. Thus, programmers can create/modify their configurations outside the Configurator using Visual Basic, if desired.
- XML supports schemas. A **schema** is a special XML file that specifies the data structure of an XML data file.

To export data, select **XML Export** from the **File** menu. This opens the **Export XML File** dialog box, as shown in the figure below. Give the file a name, and then choose the directory to which you want to export the data from your database. Click **Save**.

Note: You can also export configuration data to an XML schema file by selecting **XML Export** from the **File** menu.



Exporting Configuration Data to an XML File

Importing Configuration Data

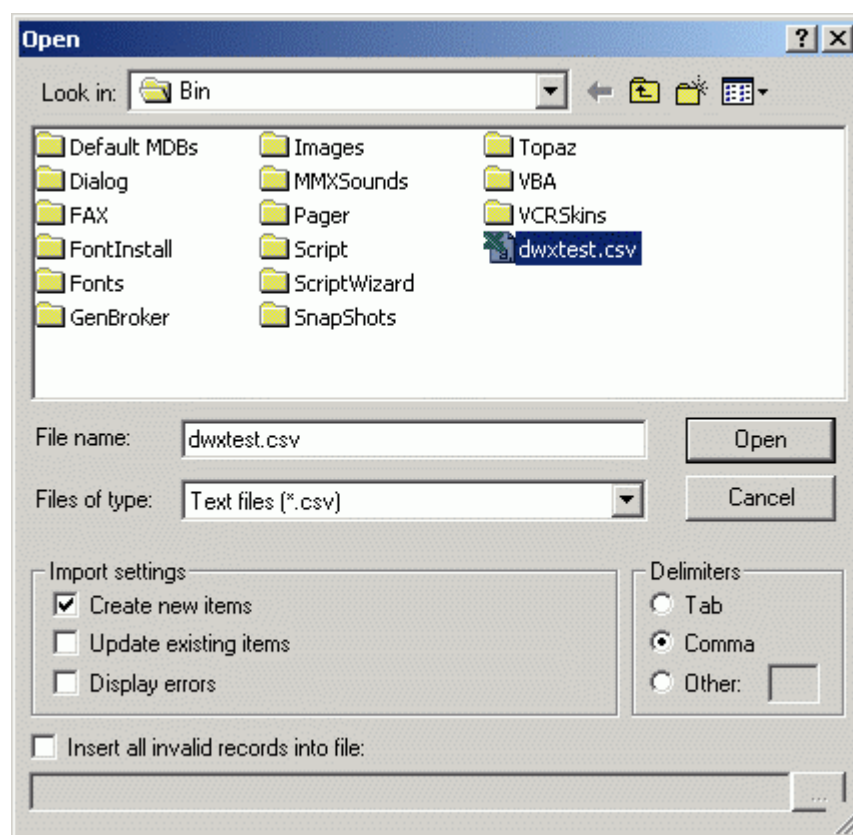
Importing Data From a Text or CSV File

The Configurator offers the flexibility of importing data from a text (.txt) file or a Microsoft Excel (.csv) file to your configuration database. To import data, select **Import CSV** from the **File** menu. This opens the **Import Configuration Data From File** dialog box, shown below. You can then specify the delimiters and choose from the following import settings:

- **Create new items.** When the import file contains items that are not yet in the configuration database, then it creates them. Otherwise it skips these items.
- **Update existing items.** When the import file contains items that are in the configuration database, then it updates them using data from the import file. Otherwise it skips these items.

Note: Either **Create new items** or **Update existing items** must be selected. Otherwise there is nothing to import.

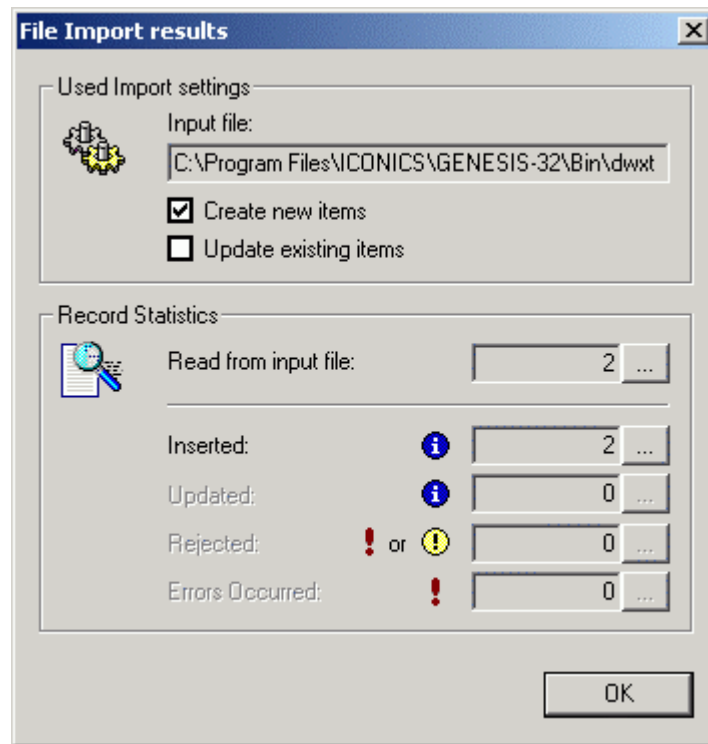
- **Display errors.** When this item is checked, the Configurator shows a dialog box if an error occurs, and then asks you if you want to proceed with the import. When it is not checked, it skips all items where an error occurred.



Import Configuration Data From File Dialog Box

When you have selected a file to import, click **Open**. When the import is completed, the **File Import Results** dialog box opens, as shown below. This shows the import settings, including the input file name. It also provides a summary of the import, including how many items were inserted, updated, or rejected, and shows how many errors occurred.

Click the ... button to the right of each field to get the details view of the import results, as shown below. This view shows the specific items that were inserted, updated, or rejected, as well as a description of any errors that occurred.



File Import Results Dialog Box

Importing Data From an XML File

The Configurator allows you to import data from your configuration database to an XML file. The XML export/import functionality was mainly developed for Windows platforms that do not support databases (e.g. Windows CE and Windows Embedded). XML has the following advantages over the CSV import/export function:

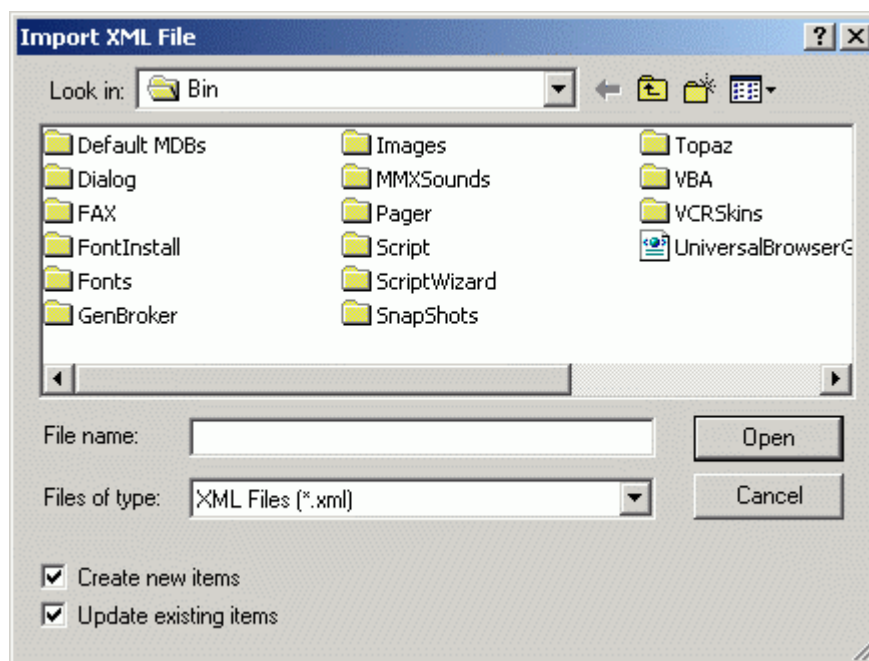
- XML has a standardized format, unlike the text/CSV format, which uses various delimiters (e.g. TAB instead of commas, strings could not accept all characters, etc.)
- XML is language-independent, whereas CSV converts date/time, floats, and currency fields according to local settings in Windows. For example, using CSV, you cannot export data on German Windows and import it on English windows without making changes
- Windows has an installed automation object that has the capability to work with XML. Thus, programmers can create/modify their configurations outside the Configurator using Visual Basic, if desired.
- XML supports schemas. A **schema** is a special XML file that specifies the data structure of an XML data file.

To import data, select **XML Import** from the **File** menu. This opens the **Import XML File** dialog box, as shown in the figure below. Give the file a name, and then choose the directory from which you want to import the data. You can then specify the delimiters and choose from the following import settings. Click **Open**.

- **Create new items.** When the import file contains items that are not yet in the configuration database, then it creates them. Otherwise it skips these items.
- **Update existing items.** When the import file contains items that are in the configuration database, then it updates them using data from the import file. Otherwise it skips these items.

Note: Either **Create new items** or **Update existing items** must be selected. Otherwise there is nothing to import.

Note: Selecting **XML Validate** from the **File** menu does not import an XML data file, but it will try to validate its structure using stored XML schema. Once it passes this validation, the XML file is acceptable for import by the Configurator.

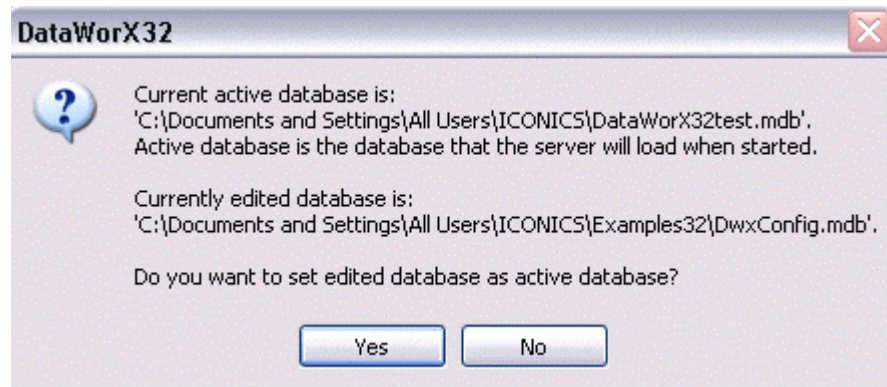


Importing Configuration Data From an XML File

Activating the Database

Once your configuration is complete, you need to make sure that it is the active database. The database that is currently active is the one that the server uses. To make the current database active, select **Make Active...** from the **File** menu. If the **Make Active...** selection is grayed out, then the current database is already the active database.

A dialog box appears showing both the current active database and database that is currently being edited, as shown in the figure below. To set the edited database as the active database, click the **Yes** button.



Making the Database Active

Edit Menu

The **Edit** menu commands are listed in the table below.

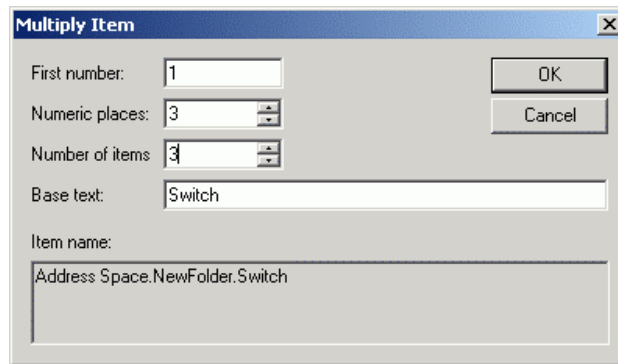
Edit Menu Commands

Command	Shortcut Keys	Function
New		Creates a new item depending on what is selected in the tree control.
Rename		Renames the selected item.
Multiply	CTRL+M	Opens the Multiply Item dialog box, which allows you to multiply an item in the tree control.
Find and Replace	CTRL+H	Opens the Find and Replace dialog box. Enter a string to search for in the Find What field. The Replace operation works in a similar way and replaces the Find What string(s) with the Replace With string(s). All dialog options are the same, except that the Replace dialog box has a field called Replace With .
Delete	CTRL+DEL	Deletes the selected object.
Cut	CTRL+X	Cuts the selected object from the view and places it on the clipboard.
Copy	CTRL+C	Copies the selected object to the clipboard.
Paste	CTRL+V	Pastes the last object placed on the clipboard.
Select All	CTRL+A	Selects all objects in a list. The selection is shown in the upper-right-hand section of the viewer.
Invert Selection		Unselects all selected items and selects all unselected items in a list in the upper-right-hand section of the viewer.

Multiplying Items

The Configurator allows you to multiply items in the tree control. Multiplication provides a simple way of developing configurations where there are many similar items in a given category. To multiply an item:

1. Select the item in the tree control that you wish to multiply.
2. Either right-click the item and select **Multiply** from the pop-up menu, or select **Multiply** from the **Edit** menu. This opens the **Multiply Item** dialog box, shown below.



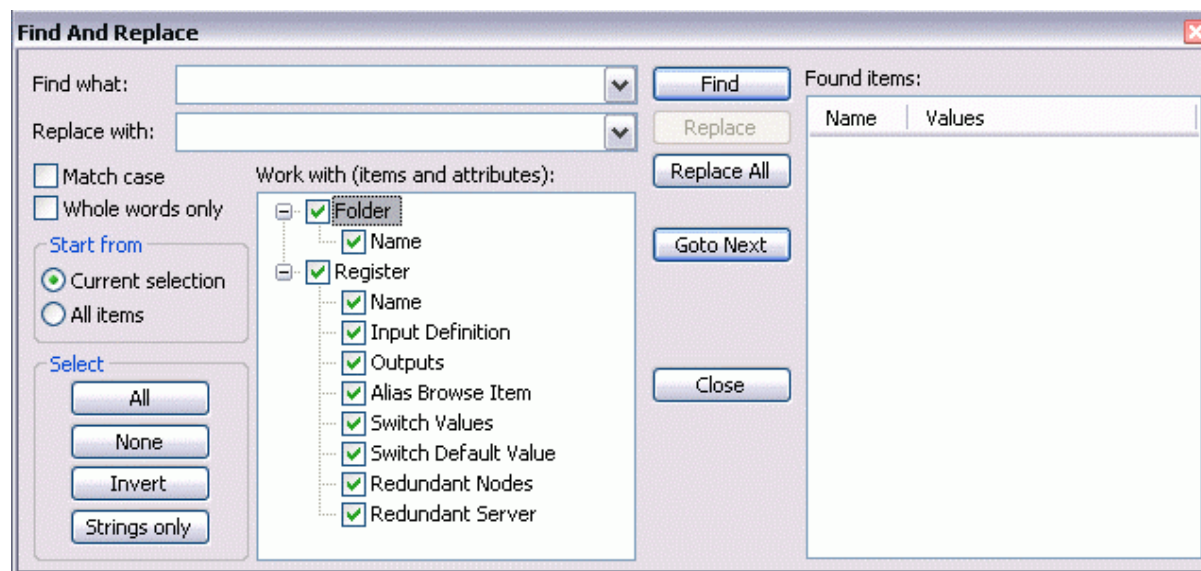
Multiply Item Dialog Box

3. When the items are multiplied, they are all given a base name followed by a number. The default base text is the name of the item selected for multiplication. To modify the base text, change the **Base Text** field appropriately.
4. In the **First Number** field, specify the number to appear next to the first multiplied item.
5. In the **Number of Items** field, specify how many items you wish to create.
6. In the **Numeric Places** field, specify the minimum length of each number to append. Values that take up less space than the specified amount of numeric places will have zeros before the number.
7. If you want to multiply all subfolders as well, check the **Including Subtree** check box.
8. Click the **OK** button to do the multiplication. The example configuration shown in the **Multiply Item** dialog box above creates three new items with the following names:
 - Switch001
 - Switch002
 - Switch003

All subfolders will also be multiplied.

Find and Replace

The **Find and Replace** function on the **Edit** menu allow you to find and replace strings. Selecting **Find and Replace** from the **Edit** menu opens the **Find and Replace** dialog box, as shown in the figure below. Enter a string to search for in the **Find What** field. The replace operation works in a similar way and replaces the **Find What** string(s) with the **Replace With** string(s). When the search or replace operation is completed, the results are displayed in the **Found Items** pane at the right of the **Find and Replace** dialog box



Find/Replace Dialog Box

In the **Work With (Items and Attributes)** field, check one or more types of properties to search for:

- **All:** Selects items and attributes.
- **None:** Unselects all items and attributes.
- **Invert:** Unselects the currently selected items and attributes, and selects the currently unselected items and attributes.
- **Stings Only:** Selects items and attributes that contain strings.

In the **Start From** field, select the scope of the search:

- **Current Selection:** Searches only the current selection.
- **All items:** Searches all items in the Configurator.
- **Current Display:** Searches the entire currently loaded display.
- **Multiple Files:** Searches in multiple display files. Click the ... button to open the **Edit File List** dialog box, which allows you to specify a list of files to search.

You can check one or both of the following search options:

- **Match whole word only:** Restricts the search operation to whole-word matches.
- **Match case:** Restricts the search operation to case-sensitive matches.

Find and Replace Operations

When you are ready to execute a find or replace operation, click one of the following buttons:

- **Go to Next:** Finds the next match.
- **Find:** Finds all occurrences of the search text.
- **Replace:** Replaces a single match and then advances to the next match.
- **Replace All:** Replaces all occurrences of the search text with the replacement text.
- **Close:** Stops an in-progress search or replace operation.

View Menu

The **View** menu commands are listed in the table below.

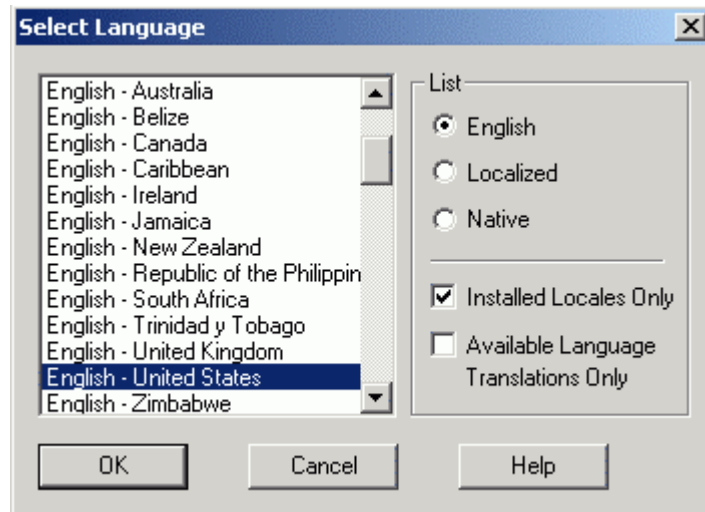
View Menu Commands

Command	Shortcut Keys	Function
Toolbars		Toggles the standard and data manipulation toolbars.
Status Bar		Toggles the status bar.
Large Icons	F7	Displays items as large icons.
Small Icons	F8	Displays items as small icons.
List	F9	Displays items as a list.
Details	F10	Displays items as a list along with detailed information about the configuration of each item.
Dialog View	F11	Toggles the configuration window (right-hand pane).
Monitor View	F12	Displays OPC server data in a separate pane.
Sort By		Displays a list of options for sorting the columns in the right-hand pane of the screen. The options listed depend on the level within the view.
Show/Hide Columns		Displays a list of options that you can choose to show or hide in the view.
Select Language		Opens the Select Language dialog box (see below). Choose the language you wish to use for your system (Unicode version only) and click OK . For navigation purposes, use the buttons and check boxes in the List section.
Global Refresh	F5	Refreshes the data for the entire Configurator screen.
Subtree Refresh	CTRL+F5	Refreshes only the data contained in the currently selected subtree.

Selecting Languages

The **Select Language** function on the **View** menu allows you to choose which language to use in your display. Choosing **Select Language** from the **View** menu opens the **Select Language** dialog box, shown in the figure below.

Note: A language resource .dll is required for language switching.



Select Language Dialog Box

Define the parameters listed in the table below. Then click **OK** to return to the work area.

Select Language Parameters

Parameter	Description
List	Lists available languages. Depending on which item you have selected, the view on the left will change. If English is checked, the languages will appear as their English name. If Localized is checked, the languages will appear with the native country in parentheses (for languages with several dialects only). When Native is checked, the languages are displayed the way they would be written in that language.
Installed Locales Only	If this is checked, local languages appear in the box.
Available Language Translations Only	Checking this box allows you to choose from available language translations only.

Go Menu

The **Go** menu commands are listed in the table below.

Go Menu Commands

Command	Shortcut Keys	Function
Up One Level		Moves the cursor up one level in the tree control.
Next Item	ALT+Down Arrow	Moves the cursor to the next item down in the tree control.
Previous Item	ALT+Up Arrow	Moves the cursor to the next item up in the tree view.
Expand Item	ALT+Left Arrow	Expands an item that contains a submenu.
Collapse Item	ALT+Right Arrow	Collapses an item that contains a submenu.
Page Up	ALT+PgUp	Moves the cursor up to the first item in the tree.
Page Down	ALT+PgDown	Moves the cursor down to the last visible item in the tree.
Home	ALT+Home	Moves the cursor up to the first item in the tree.
End	ALT+End	Moves the cursor down to the last visible item in the tree.
Next Pane	F6	Moves the cursor to the next pane.
Previous Pane	SHIFT+F6	Moves the cursor to the last pane used.

Action Menu

The **Action** menu commands are listed in the table below.

Action Menu Commands

Command	Function
Start Runtime	Starts the DataWorX32 runtime operation.
Stop Runtime	Stops the DataWorX32 runtime operation.

Tools Menu

The **Tools** menu commands are listed in the table below.

Tools Menu Commands

Command	Function
GenBroker Configurator	Launches the GenBroker Configurator.
OPC Tunneling	Launches the OPC Tunneling Configurator.
Options	Opens the Options dialog box, which is described in the section below.

Options

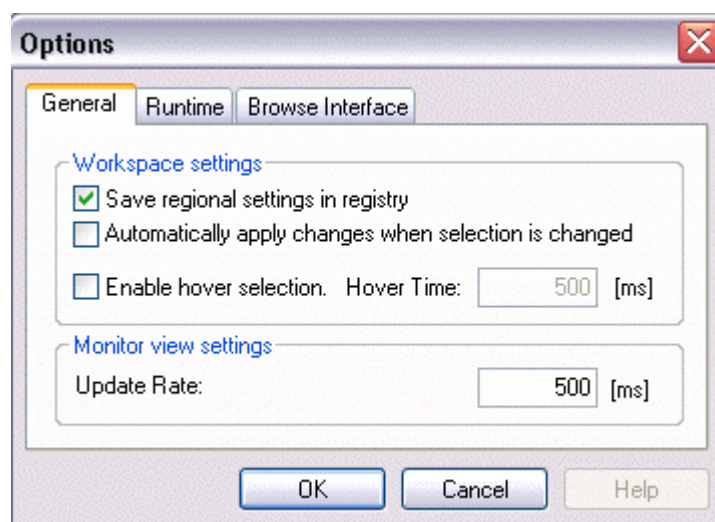
To choose additional settings, select **Options** from the **Tools** menu. This opens the **Options** dialog box, which contains the following tabs:

- General
- Browse Interface

General Tab

The **General** tab of the **Options** dialog box, shown below, contains the following options:

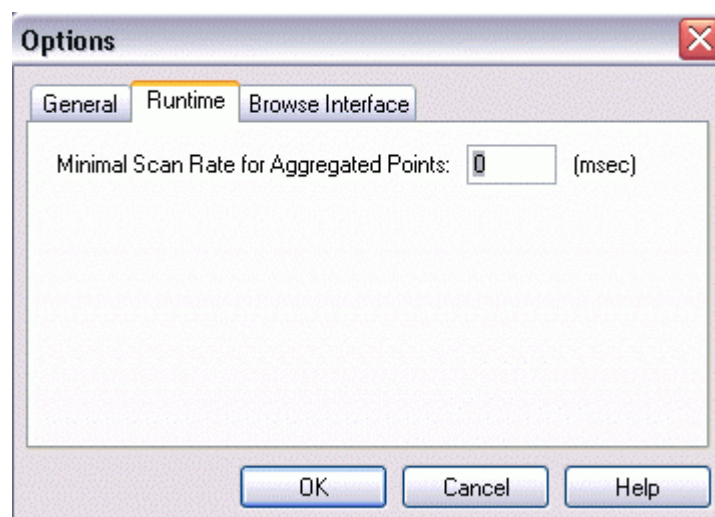
- **Save regional settings in registry:** Checking this option allows you to save regional settings in the registry so that they are applied each time you start the configuration application. This applies to the language settings as well as time and date settings.
- **Automatically apply changes when selection is changed:** Checking this option allows changes to the configuration database to be saved each time you switch dialogs without clicking on the **Apply** button or being shown a message asking if you would like to apply changes.
- **Enable hover selection:** Checking this option allows you to highlight an item by moving the mouse pointer over that item and keeping it there for a specified amount of time (in milliseconds).
- **OPC Monitoring Update Rate:** This specifies the update frequency (in milliseconds) of the items in the Monitor View. The monitor scans the server and displays the tag values at the bottom of the Configurator screen.



Options Dialog Box: General Tab

Runtime Tab

In the **Runtime** tab of the **Options** dialog box, shown below, you can specify a **Minimal Scan Rate for Aggregated Points** (in milliseconds). This is the fastest scan rate for aggregated points. For example, if this is set to 100 msec and someone requests a tag at 50 msec through DataWorX32, he will get it at 100 msec. If someone requests another tag at 500 msec, he will get it at 500. No one gets it at a faster scan rate than defined here.

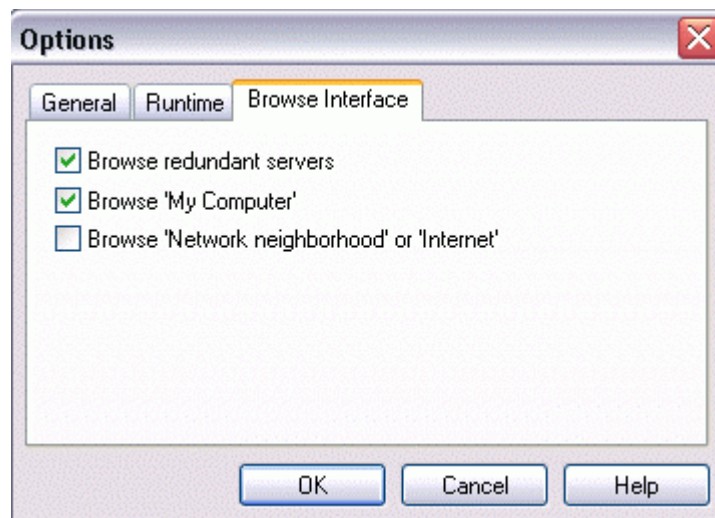


Options Dialog Box: Runtime Tab

Browse Interface Tab

The **Browse Interface** tab of the **Options** dialog box, shown below, contains the following commands:

- **Browse Redundant Servers:** When checked, the items of the servers defined as redundant (in the Redundancy menu) will be shown.
- **Browse My Computer:** When checked, the local OPC server and its items available through DataWorX32 will be shown.
- **Browse Network Neighborhood or Internet:** When checked, the OPC servers located on remote nodes and their items will be shown.



Options Dialog Box: Browse Interface Tab

Help Menu

The **Help** menu commands are listed in the table below.

Help Menu Commands

Command	Shortcut Keys	Function
Help Topics	F1	Launches the online Help for the Configurator.
About Application		Launches the About Box , which contains information about the product version number, copyright, and available disk space. It also contains contact information.

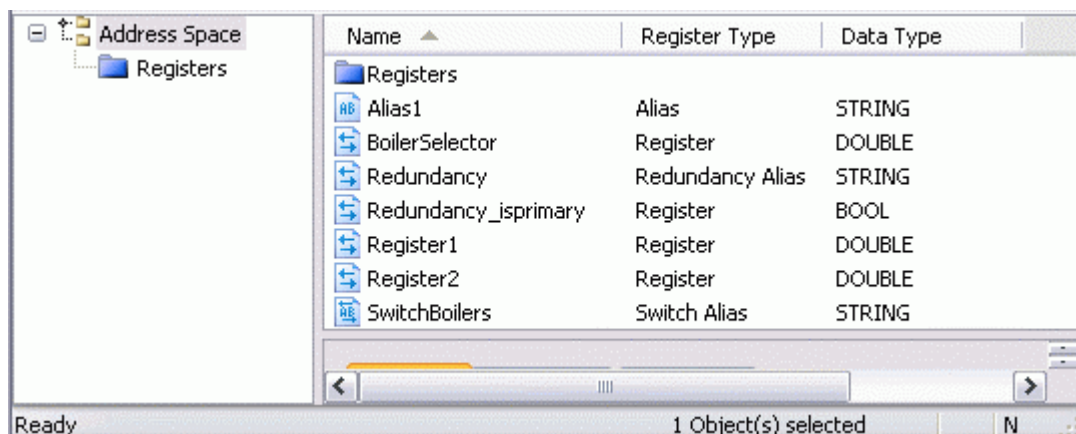
Configuring the Address Space

2

Overview of Address Space Configuration

In the DataWorX32 Configurator, the parameters for data items are set up in the **Address Space** tree control, as shown in the figure below. When configuring the Address space, you can:

- Create folders to organize data items.
- Create and configure registers.
- Create and configure aliases.
- Create and configure switch aliases.
- Create and configure redundancy aliases.



Address Space Tree Control

The **Address Space** tree control in the Configurator sets the properties and connection parameters for all data items, which can be organized hierarchically. It is similar to organizing directories and files on your computer's hard disk. A **folder** can contain additional folders and also data items. The **data items** are always the branches in the tree control hierarchy. The hierarchical structure of the folders and data items helps to organize the registers, aliases, switches, and redundancy aliases.

Folders

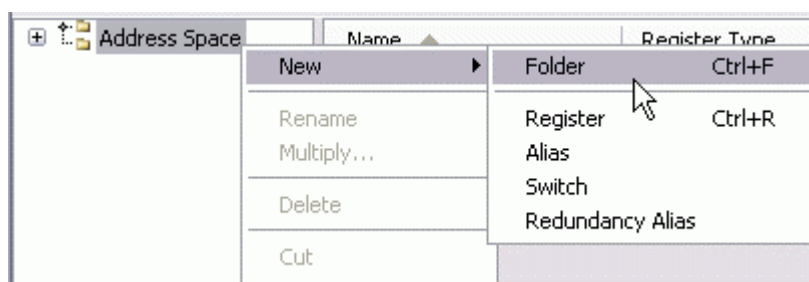
Folders can be used to group items logically. You can configure as many folders as required. Each folder can even have subfolders. The use of folders is not required.

If desirable, the configuration could just contain data items without any folders. But most likely this will only be useful if the application does not demand too many persistent tags.

Creating New Folders

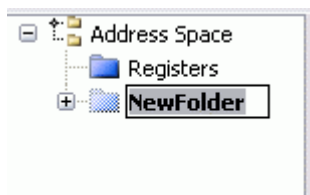
To create a new folder:

1. Right-click the **Address Space** tree control of the Configurator and select **New > Folder** from the pop-up menu, as shown in the figure below.



Creating a New Folder

2. Type a name for the new folder, as shown in the figure below.



Naming the New Folder

3. The new folder appears under the tree control.

Registers

Many clients require a common variable to share values. One of the key functions of DataWorX32 is to provide a mechanism for defining variables and making them available to *all* GENESIS32 clients and OPC clients. DataWorX32 uses the concept of **registers** to achieve this. DataWorX32 makes it possible to define many variables that are accessible to many clients simultaneously because DataWorX32 registers can be used as **global variables**. These variables can also act as aliases for clients. In the various client applications, you can configure global variables for use within that particular application, or possibly only within a particular document for that application.

Register Names

Each register has a unique name composed of only numbers, letters, and the underscore "_" character. DataWorX32 verifies the names are indeed unique, prompting you if this is not the case. The **Register name** field does not accept spaces. Spaces are disabled intentionally because the register names are used by OPC clients as tag names. The OPC specification allows for using spaces, but it is not recommended.

An **Alias Register** is a register that is used to keep the alias value and is of string data type. You can access it like an ordinary register. The [[and]] are used to expand the alias, i.e. to replace the name of the alias with its value. For example:

Register 'REG' = 10; Alias Register 'MODBUS' = "ICONICS.ModbusOPCServer".

'DataWorX32\REG' = 10

'DataWorX32\MODBUS' = "ICONICS.ModbusOPCServer"

*'DataWorX32\[[MODBUS]]Dev1.Tag1' = ICONICS.ModbusOPCServer\Dev1.Tag1
(value)*

DataWorX32 registers are accessible through the OPC Universal Tag Browser.

DataWorX32 registers are accessible from VB using the "GetRegister" function of the Automation interface.

The count limit is not restricted by DataWorX32 itself, but is limited by system resources.

Register Inputs

A register has one **input** (or source) to define its content.

- An input may be assigned to an expression that, when resolved, defines the register's content.
- An input may be assigned to **None** to create a global variable. (The initial value is selectable.)
- An input may be assigned to an OPC data point (including the value itself, quality, and time stamp), using the OLEExpress naming convention.
- An input may be assigned to the output of another DataWorX32 register.
- Besides OPC inputs, registers, and expressions, conditions can also be used as register inputs. The condition itself is connected to multiple OPC items or registers. One of these inputs is chosen depending on the selected criteria. Using conditions as register inputs is particularly useful when used in conjunction with switch aliases.

Register Outputs

When the value of a register's input changes, it is written to all of the outputs assigned to that register (if any).

- An output may be assigned to an item of an OPC server. Thus, the register writes values to the server.
- An output may be assigned to the input of another DataWorX32 register.

- A register may be designated as "Read-Only." The register will still write to its outputs, but the value can only come from an input, such as another OPC server or an expression, not from an OPC client. If this is the case, clients may only view its contents.

Registers, aliases, and other objects in DataWorX32 can be hierarchically organized into groups (folders).

Using Registers

One of the key functions of DataWorX32 is to provide a mechanism for defining variables and making them available to *all* GENESIS32 clients and OPC clients. DataWorX32 uses the concept of **registers** to achieve this.

The "Global Variable" name is simply the register's name. It can be given values through OPC tags, expressions, constants, or even VBA. OPC server data-bridging is accomplished by assigning the input of a register to one OPC data point, and assigning the output to different (even multiple) OPC data points.

This section describes how to configure Registers functions in the DataWorX32 Configurator in order to connect more than two servers. For example, if you want a client PC to send a value from a Modbus OPC server to a Bristol OPC server, you can define the **Register Name** in the **Properties** dialog box, define the **Properties** for that register, set the **Input** to the Modbus OPC server's tag, and set the **Output** to the Bristol OPC server's tag for a Register Tag using the **Register Settings** dialog box.

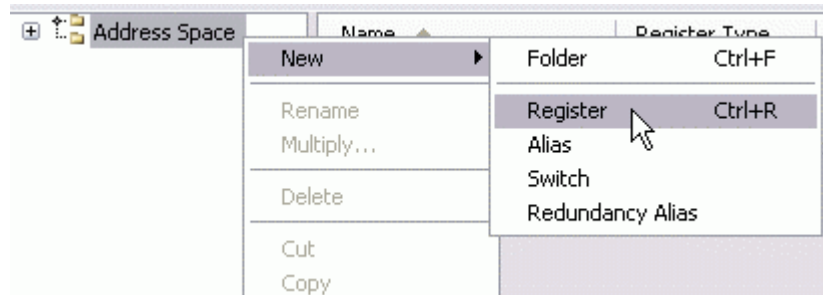
Registering Running Instances

DataWorX32 now registers its dispatch pointer at GenRegistrar on startup and unregisters on exit. Dispatch pointers may be used by Visual Basic or other applications to control a running application. Now, a dispatch pointer to an instance of DataWorX32 running in both configuration mode and runtime can be obtained. In past versions of DataWorX32, this was possible only in runtime.

Creating New Registers

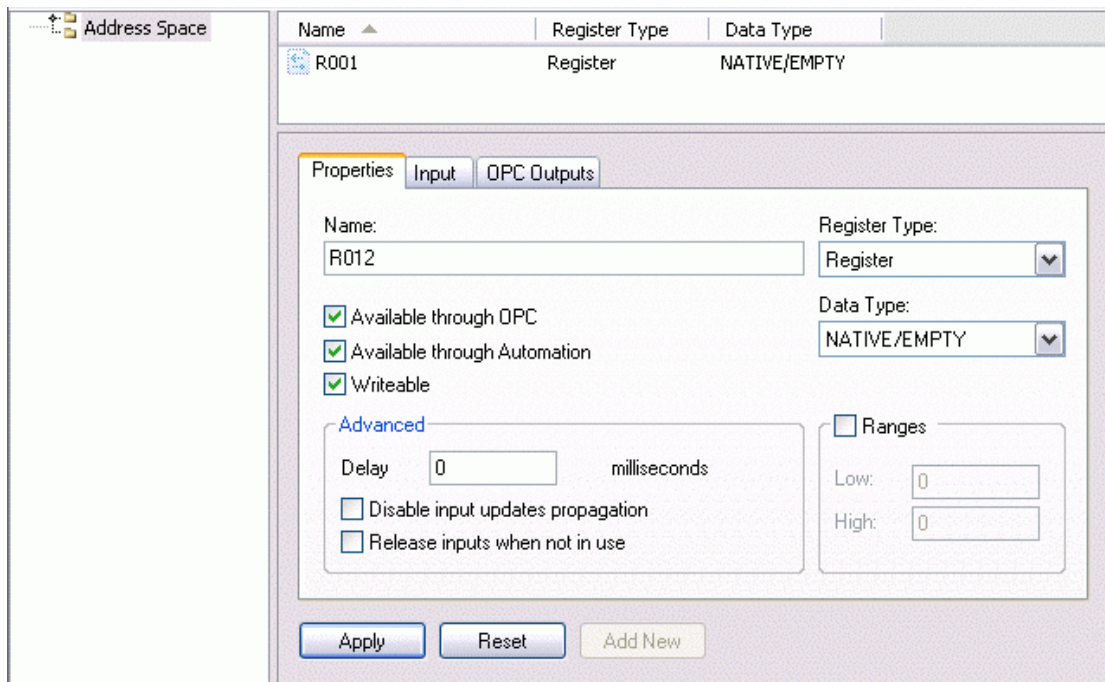
To create a new register:

1. Right-click the **Address Space** tree control of the Configurator and select **New > Register** from the pop-up menu, as shown in the figure below.



Creating a New Register

2. The properties dialog box for the new register appears in the right-hand pane of the Configurator, as shown in the figure below.
3. In the **Name** field, type a name for the new register.
4. When you have finished configuring the register properties, click the **Apply** button. The new register appears under the tree control, as shown in the figure below.



Configuring Register Properties

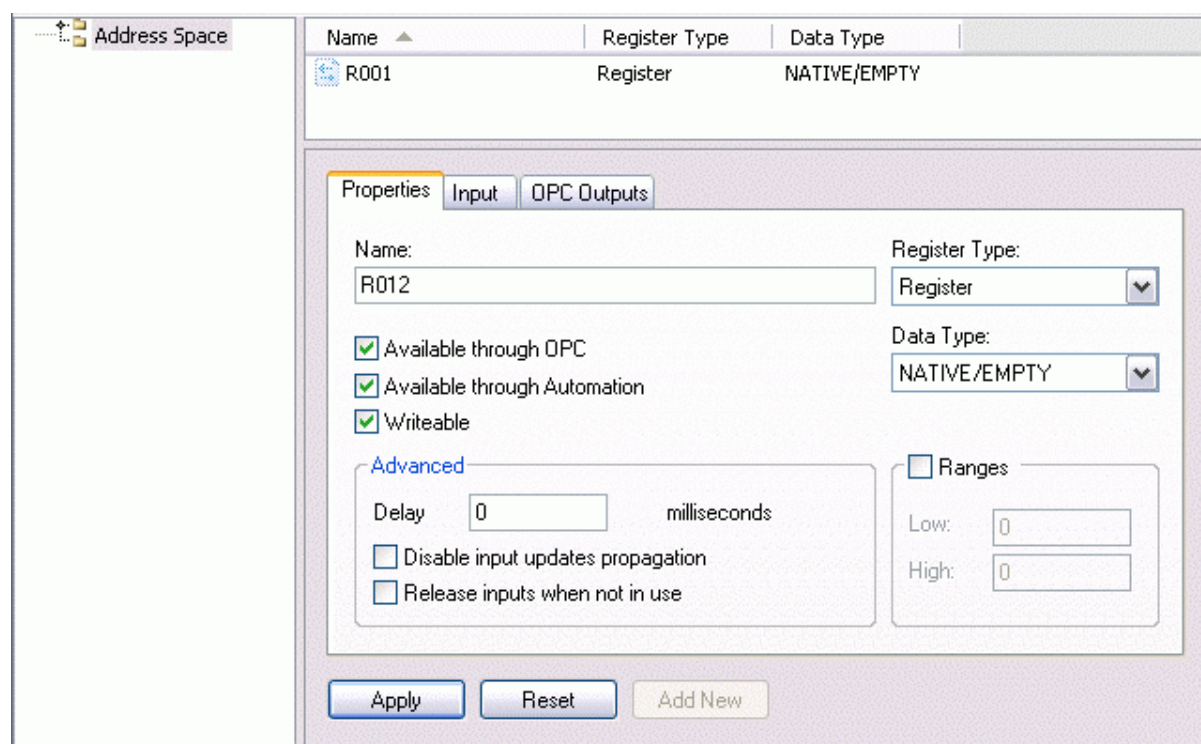
Configuring Register Properties

The register properties dialog box contains the following tabs, as shown in the figure below:

- Properties
- Input
- OPC Outputs

The **Properties**, **Input** and **Output** tabs are the same for different register types. The **Register Type** can be modified on the **Properties** tab by selecting a type from the drop-down list, as shown in the figure below. Depending on this register type, other pages appear or get hidden. For example, aliases can be browsed, so they have the **Browse Interface** tab (unlike registers). On the other hand redundancy aliases, for example, do not have inputs, so the **Input** tab disappears when you select a register to be a redundancy alias. It is important to understand that everything (aliases, switches, etc.) is a register. Its features then depend on the type of register (e.g. the data type for an alias is always STRING).

Note: Each register configuration has a limit of 4 kilobytes (4,096 characters). This limit applies to the entire register record (e.g. register name, expression strings, outputs, list of condition inputs, list of redundancy nodes, etc.).



Configuring Register Properties

Register Settings: Properties Tab

The **Properties** tab in the **Register Settings** dialog box, shown below, allows you to set the following register parameters, as described in the table below.

The screenshot shows the 'Register Settings' dialog box with the 'Properties' tab selected. The 'Name' field contains 'R012'. The 'Register Type' dropdown is set to 'Register'. The 'Data Type' dropdown is set to 'NATIVE/EMPTY'. Three checkboxes are checked: 'Available through OPC', 'Available through Automation', and 'Writeable'. The 'Advanced' section is expanded, showing a 'Delay' of '0' milliseconds and two unchecked checkboxes: 'Disable input updates propagation' and 'Release inputs when not in use'. The 'Ranges' section is also expanded, showing two unchecked checkboxes: 'Low' and 'High', each with an associated input field containing '0'.

Register Settings: Properties Tab

Properties Tab Parameters

Name	A common tag name.
Register Type	Specifies Register, Alias, Switch Alias, or Redundancy Alias.
Available Through OPC	Checking this box makes the register available to OPC clients.
Available Through Automation	Checking this box makes the register available to VB applications so that it could be edited by a VB editor.
Writeable	Checking this box makes the register writeable. Uncheck the box for a read-only OPC client.
Data Type	Allows you to select from the following data types: Native/Empty, Float, Double, Boolean, Byte, Word, DWord, Character, Short, Long, String. All data types available in GraphWorX32 are supported in registers for DataWorX32.
Ranges	Checking this box enables the High and Low ranges fields.
High Range	You can specify a High Range in the box provided.
Low Range	You can specify a Low Range in the box provided.

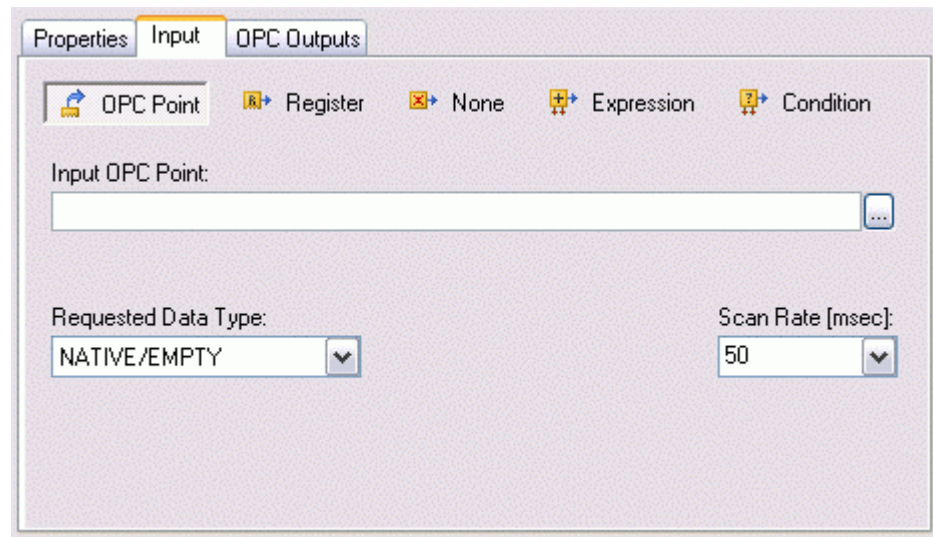
Delay	You can specify a delay time for the register (in milliseconds) in the box provided.
Disable OPC Propagation Support	When an OPC item is connected to a register as both input and output, a fast sequence of writes to that register may cause item value oscillation. (This may happen when two subsequent writes to the register are faster than the OPC item can perform the write operation. The acknowledge of the first write then overwrites the second written value already being stored in the register.) To avoid this behavior, check the Disable input updates propagation check box in the register properties. The registers release their inputs when they are not used. 'Not used' means either referenced from the outside (by an OPC client or through automation), or it means they are having an output. Because a register having an output is actually used for bridging, therefore it cannot release its input. Bridging would not work without it. Checking the Release inputs when not in use check box means the registers have to request for input when a client starts using it and release it when the client stops using it. So the registers with this option checked may respond more slowly.

Note: By default, the data type and range information are obtained from the input of register.

Register Settings: Input Tab

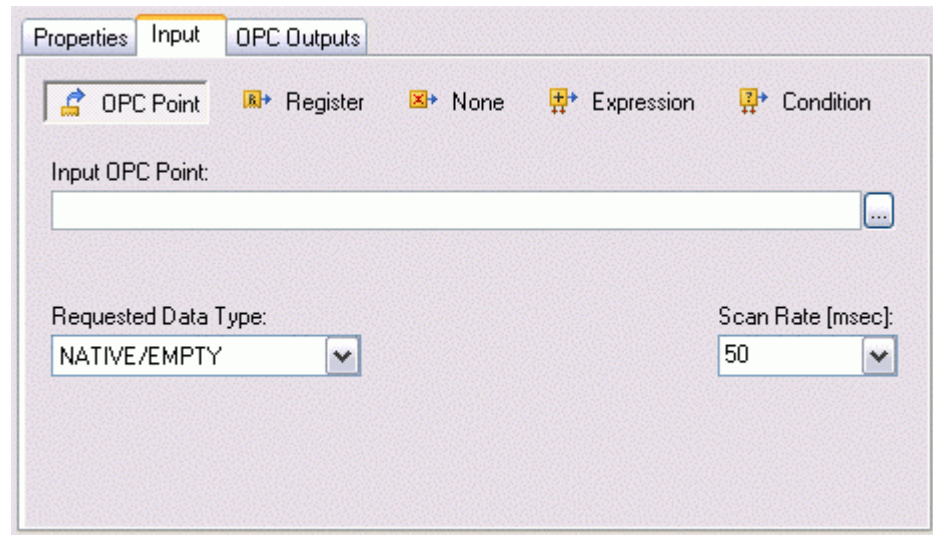
The **Input** tab in the **Register Settings** dialog box, shown below, allows you to set the following register input parameters:

- OPC Point
- Register
- None
- Expression
- Condition Criteria
- Condition Result
- Condition Inputs

**Register Settings: Input Tab**

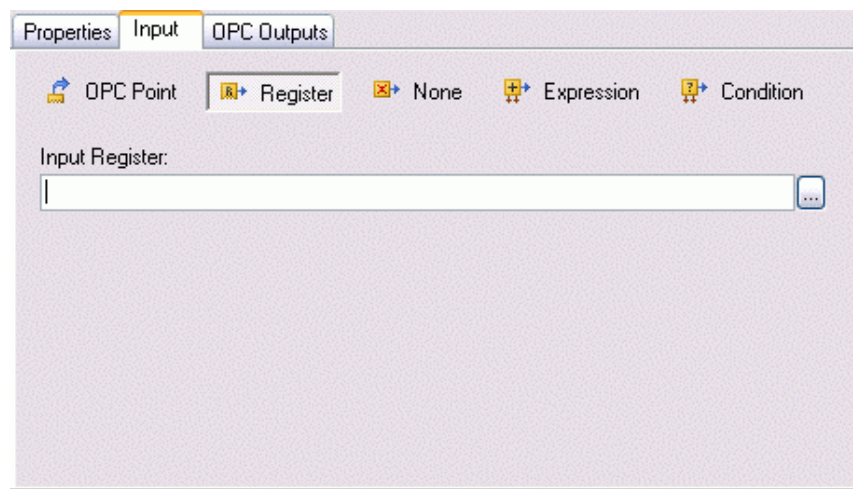
Input Tab: OPC Point

The **OPC Point** section of the **Input** tab, shown in the figure below, specifies that the input is an OPC item. In the **Input OPC Point** field, you can type in an item name (tag name) or browse for one by clicking the ... button, which opens the Unified Data Browser. The **Registered Data Type** field allows you to select from the following data types: Native/Empty, Float, Double, Boolean, Byte, Word, DWord, Character, Short, Long, String. All data types available in GraphWorX32 are supported in registers for DataWorX32. You can also specify the **Scan Rate** for OPC inputs (in milliseconds)

**Input Tab: OPC Point**

Input Tab: Register

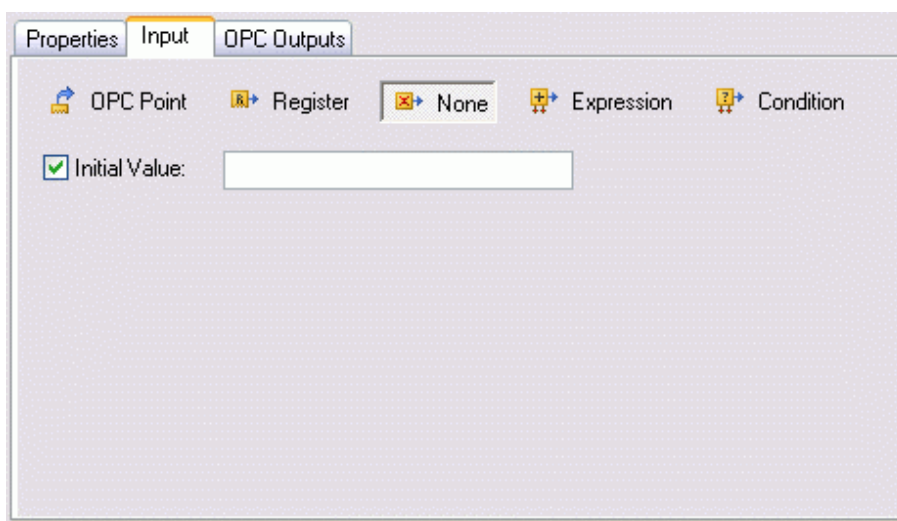
In the **Register** section of the **Input** tab, shown in the figure below, you can type in an **Input Register** or browse for one by clicking the ... button, which opens the Unified Data Browser. This lists the registers defined in DataWorX32. You may select an input coming from another register.



Input Tab: Register

Input Tab: None

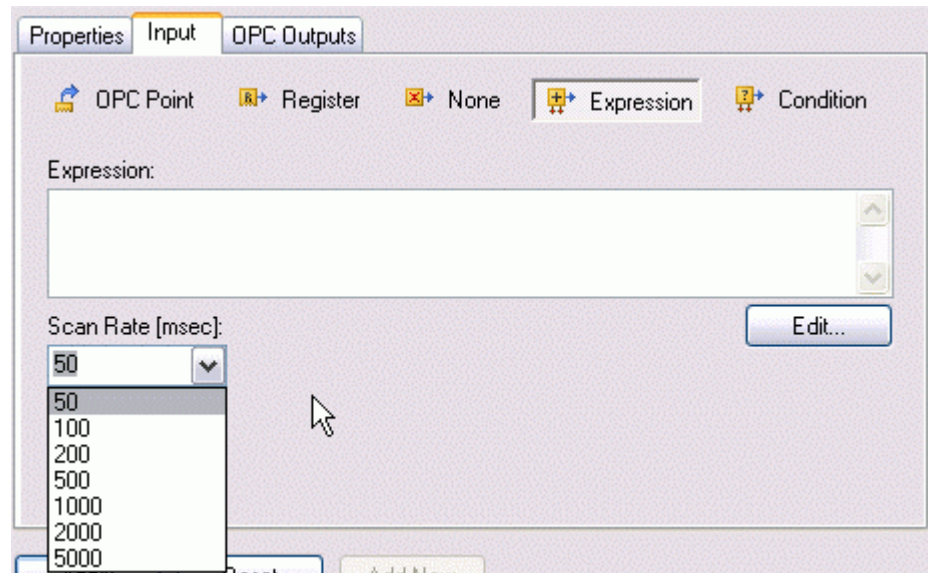
Selecting the **None** section of the **Input** tab, shown in the figure below, means no input will be provided to this register. This could be used when you want to write a value from the client and send it to many OPC servers at the same time. This is also the mechanism for creating a global variable. Checking the **Initial Value** check box allows you to specify an initial value in the field to the right.



Input Tab: None

Input Tab: Expression

The **Expression** section of the **Input** tab, shown in the figure below, allows you to specify an expression in the box provided. Clicking the **Edit** button opens the **Edit Expression** dialog box, which allows you to edit expressions using the Arithmetic, Relational, Logical, Bitwise, and Functions methods, as well as OPC tags and registers. The result of expression includes a quality evaluation. You may specify how the quality should be evaluated. You can also specify the **Scan Rate** for OPC inputs (in milliseconds)



Input Tab: Expression

Input Tab: Condition Criteria

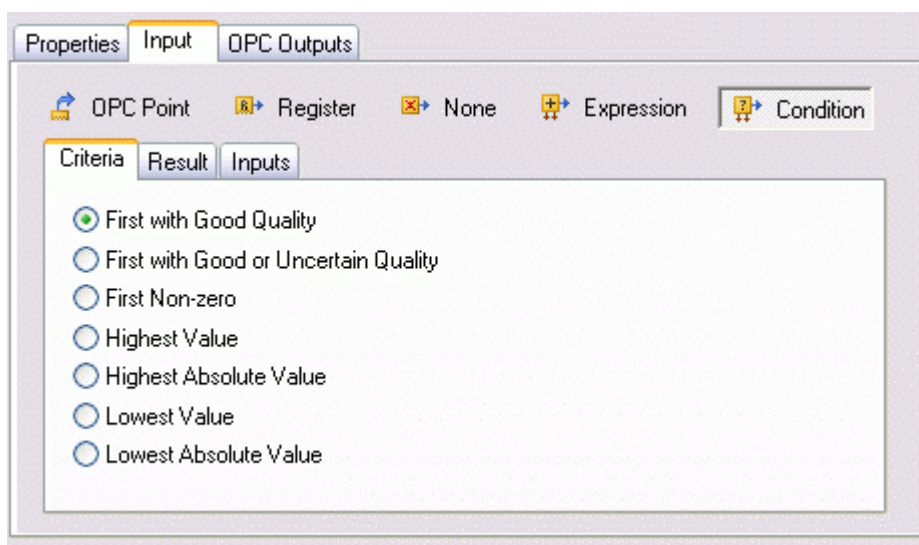
Besides OPC inputs, registers and expressions, **conditions** can be used as register inputs. The conditions themselves may be useful, but their biggest advantage is in use with Switch Aliases. The condition itself is connected to multiple OPC items or registers. One of these inputs is chosen depending on the selected criteria. The result of the condition (i.e. the value of the register having input set to this condition) may be:

- The value of the chosen input; the data type of the register will be the same as the data type of the input (by default).
- The zero-based index of the chosen input; the data type of the register will be a integer number in range 0.. $N-1$, where N is the number of the condition inputs.
- The name of the chosen input; the data type of the register will be a string.

The **Condition Criteria** section of the **Input** tab, shown in the figure below, includes the following possible criteria:

- First with good quality
- First with good or uncertain quality

- First nonzero
- Highest value
- Highest absolute value
- Lowest value
- Lowest absolute value

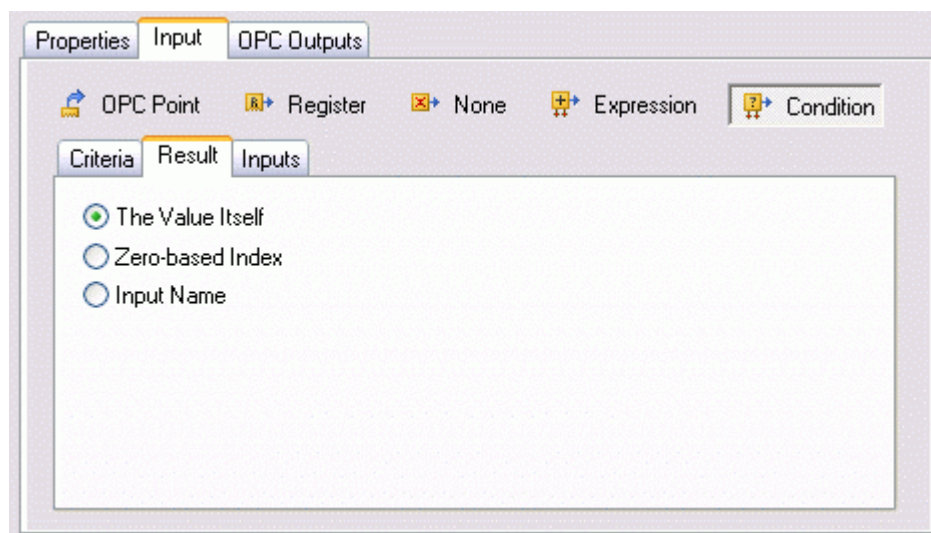


Input Tab: Condition Criteria

Input Tab: Condition Result

The **Condition Result** section of the **Input** tab, shown in the figure below, specifies the result of the condition (i.e. the value of the register having input set to this condition):

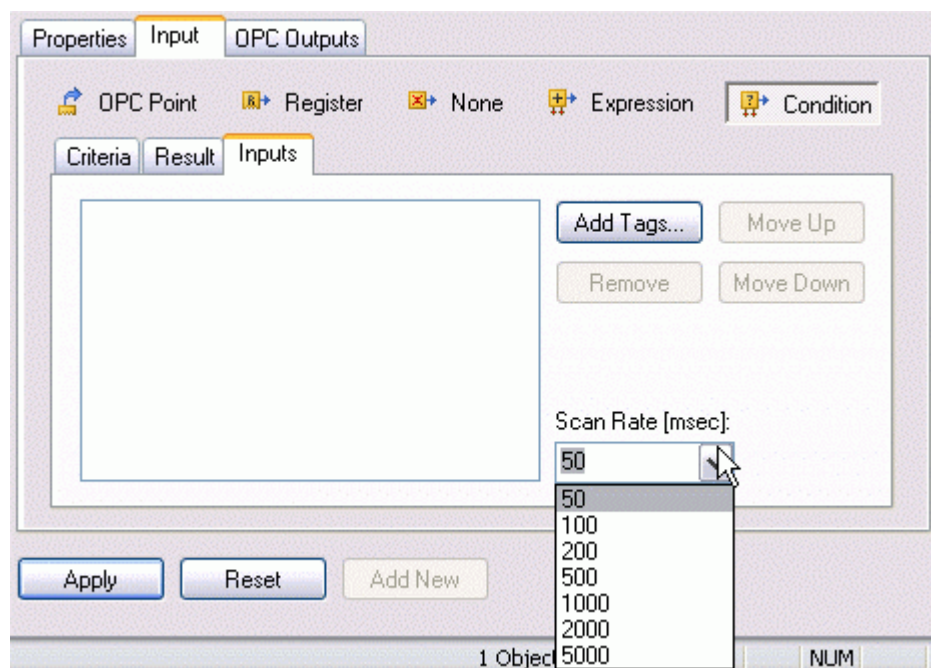
- **The value itself (the value of the chosen input):** The data type of the register will be the same as the data type of the input (by default).
- **Zero-based index of the chosen input:** The data type of the register will be an integer number in range 0.. N -1, where N is the number of the condition inputs.
- **Input Name:** The data type of the register will be a string.



Input Tab: Condition Result

Input Tab: Condition Inputs

The **Condition Inputs** section of the **Input** tab, shown in the figure below, enables you to specify the Input(s) of the condition. You can select an OPC tag from the Unified Data Browser by clicking **Add Tag**. You can also specify the **Scan Rate** for OPC inputs (in milliseconds).



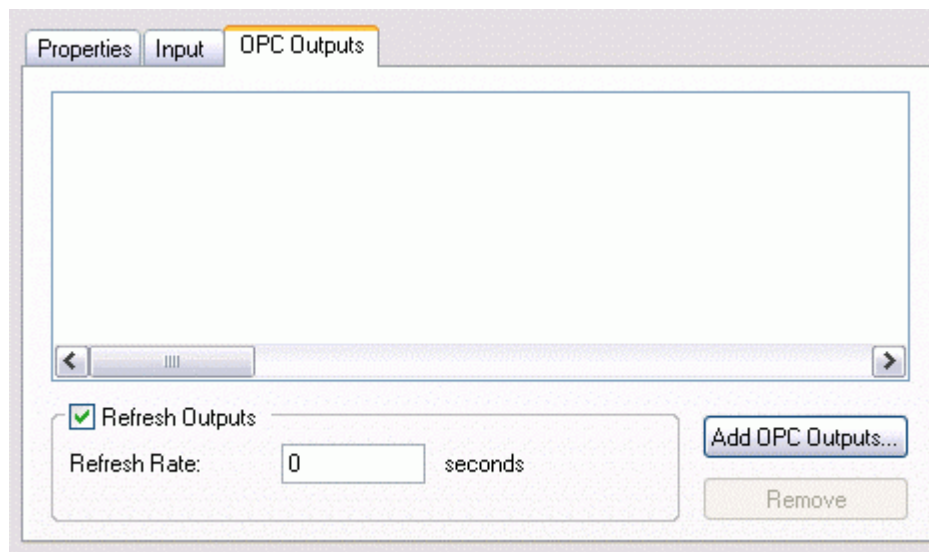
Input Tab: Condition Inputs

Register Settings: OPC Outputs Tab

In the **OPC Outputs** tab in the **Register Settings** dialog box, shown below, you can select OPC tags and /or registers from the Unified Data Browser by clicking **Add OPC Outputs**.

When bridging OPC data, values are written to the outputs only when the input value changes. This is the default behavior. To modify this behavior, you can force DataWorX32 to refresh outputs periodically. Then the value is written to the outputs even if the input value does not change. You can specify a **Refresh Rate** for outputs by checking the **Refresh Outputs** check box and typing the output refresh rate (in seconds) in the edit box.

Note: You may select more than one tag for the output.



Register Settings: OPC Outputs Tab

Aliases

DataWorX32 contains a mechanism for defining aliases. **Aliases** are symbols that are expanded to strings, known as "alias values," during runtime. Aliases are enclosed in double brackets: **[[** and **]]**.

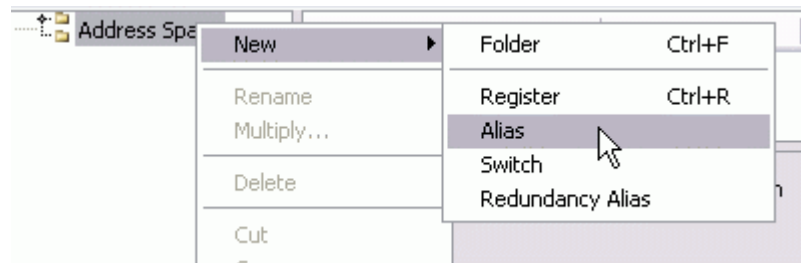
The main requirements for aliasing are as follows:

- Every alias must be defined before it is first used (that is, before the first item is requested with this alias). Otherwise the item name containing the alias is treated as invalid.
- If any name is syntactically valid and contains only aliases that have already been defined, it is treated as valid, even though after expanding aliases the name may not correspond to any existing item.
- The resolution of these aliases ("alias values") can change during runtime mode. When an alias changes, the items having the alias in their names stay valid. They simply change reference to another item after the change is made.

Creating New Aliases

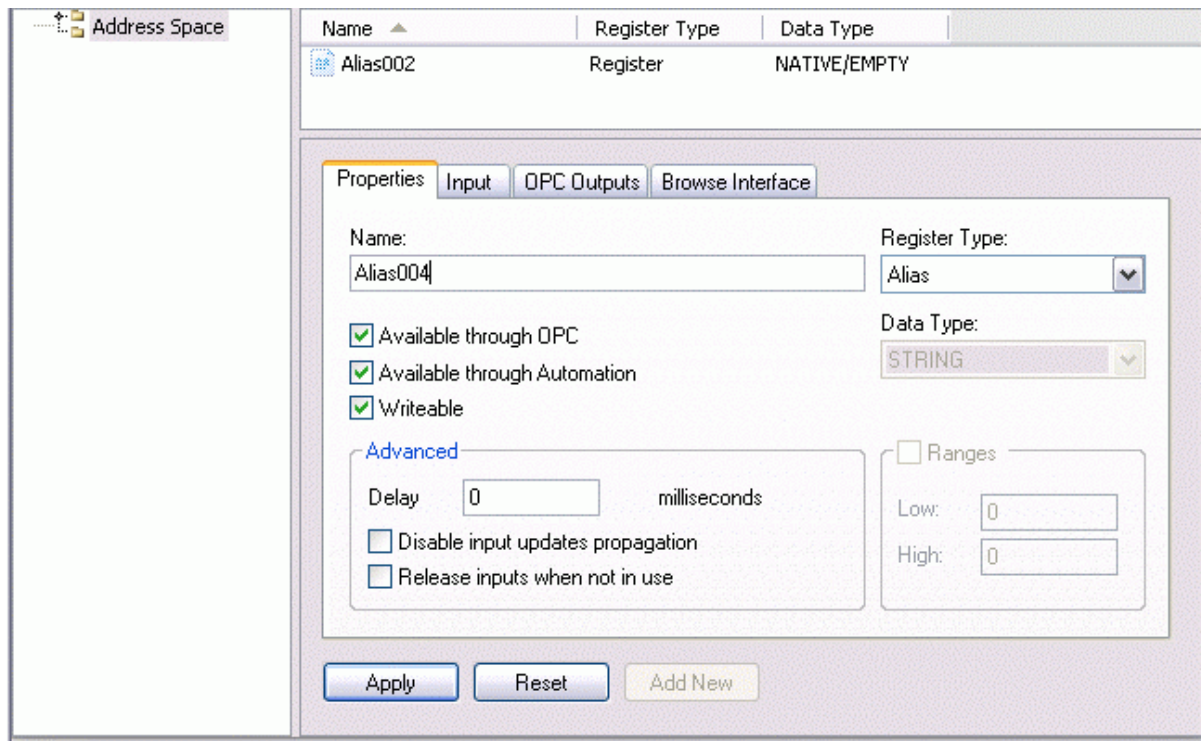
To create a new alias:

1. Right-click the **Address Space** tree control of the Configurator and select **New > Alias** from the pop-up menu, as shown in the figure below.



Creating a New Alias

2. The properties dialog box for the new alias appears in the right-hand pane of the Configurator, as shown in the figure below.
3. In the **Name** field, type a name for the new alias.
4. When you have finished configuring the alias properties, click the **Apply** button. The new alias appears under the tree control, as shown in the figure below.

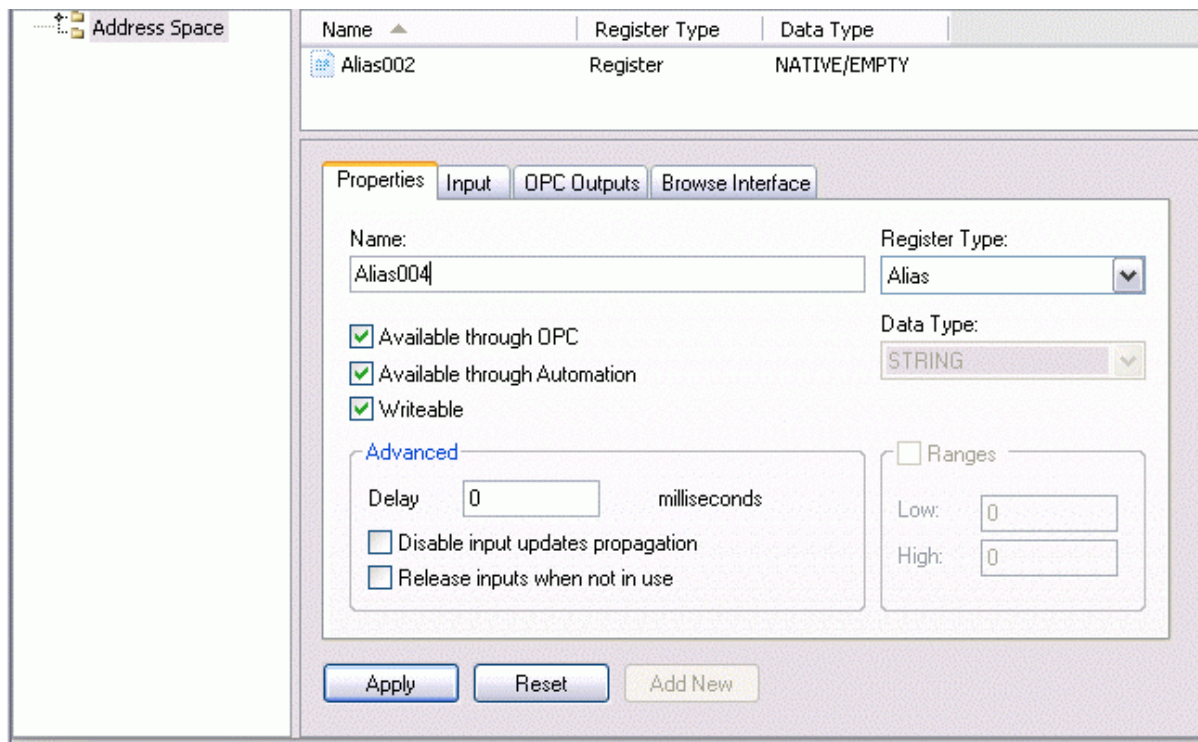


Configuring Alias Properties

Configuring Alias Properties

The alias properties dialog box contains the following tabs, as shown in the figure below:

- Properties
- Input
- OPC Outputs
- Browse Interface



Configuring Alias Properties

Alias Settings: Properties Tab

The **Properties** tab in the **Alias Settings** dialog box, shown below, allows you to set the following alias parameters, as described in the table below.

The screenshot shows the 'Alias Settings' dialog box with the 'Properties' tab selected. The 'Name' field contains 'Alias004'. The 'Register Type' dropdown is set to 'Alias'. The 'Data Type' dropdown is set to 'STRING'. Three checkboxes are checked: 'Available through OPC', 'Available through Automation', and 'Writeable'. The 'Advanced' section is expanded, showing a 'Delay' of 0 milliseconds, and two unchecked checkboxes: 'Disable input updates propagation' and 'Release inputs when not in use'. The 'Ranges' section is also expanded, showing 'Low' and 'High' values both set to 0.

Alias Settings: Properties Tab

Properties Tab Parameters

Name	A common tag name.
Register Type	Specifies Register, Alias, Switch Alias, or Redundancy Alias.
Available Through OPC	Checking this box makes the register available to OPC clients.
Available Through Automation	Checking this box makes the register available to VB applications so that it could be edited by a VB editor.
Writeable	Checking this box makes the register writeable. Uncheck the box for a read-only OPC client.
Data Type	This field is not available for aliases.
Ranges	This field is not available for aliases.
High Range	This field is not available for aliases.
Low Range	This field is not available for aliases.
Delay	You can specify a delay time for the register (in milliseconds) in the box provided.

Disable OPC Propagation Support	When an OPC item is connected to a register as both input and output, a fast sequence of writes to that register may cause item value oscillation. (This may happen when two subsequent writes to the register are faster than the OPC item can perform the write operation. The acknowledge of the first write then overwrites the second written value already being stored in the register.) To avoid this behavior, Check the Disable input updates propagation check box in the register properties. The registers release their inputs when they are not used. 'Not used' means either referenced from the outside (by an OPC client or through automation), or it means they are having an output. Because a register having an output is actually used for bridging, therefore it cannot release its input. Bridging would not work without it. Checking the Release inputs when not in use check box means the registers have to request for input when a client starts using it and release it when the client stops using it. So the registers with this option checked may respond more slowly.
---------------------------------	--

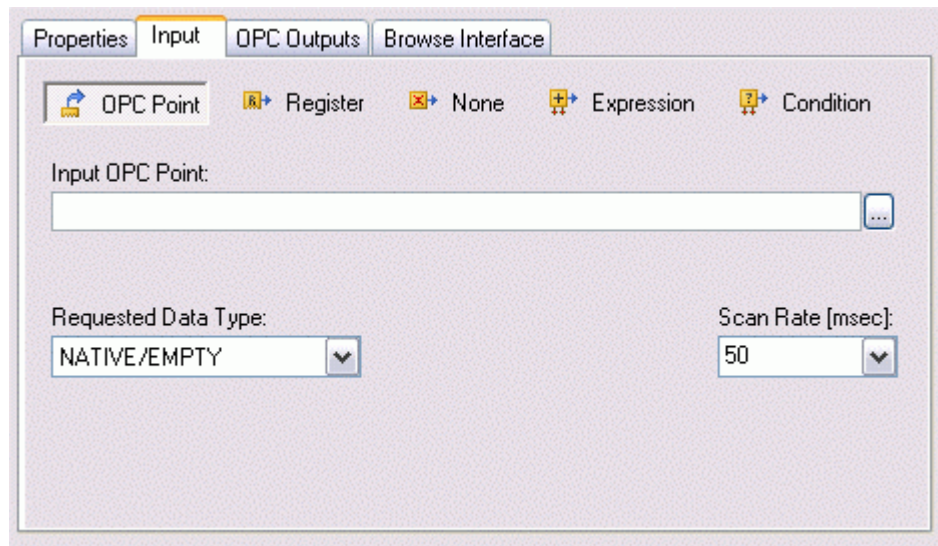
Note: By default, the data type and range information are obtained from the input of register.

Alias Settings: Input Tab

The **Input** tab in the **Alias Settings** dialog box, shown below, allows you to set the following alias input parameters:

- OPC Point
- Register
- None
- Expression
- Condition Criteria
- Condition Result
- Condition Inputs

For more information about input settings, please refer to the "Register Settings: Input Tab" section above.



The dialog box shows the 'Input' tab selected. At the top, there are four buttons: 'OPC Point' (selected), 'Register', 'None', 'Expression', and 'Condition'. Below these buttons is a text field labeled 'Input OPC Point:' with a blue ellipsis button to its right. At the bottom, there are two dropdown menus: 'Requested Data Type:' set to 'NATIVE/EMPTY' and 'Scan Rate [msec]:' set to '50'.

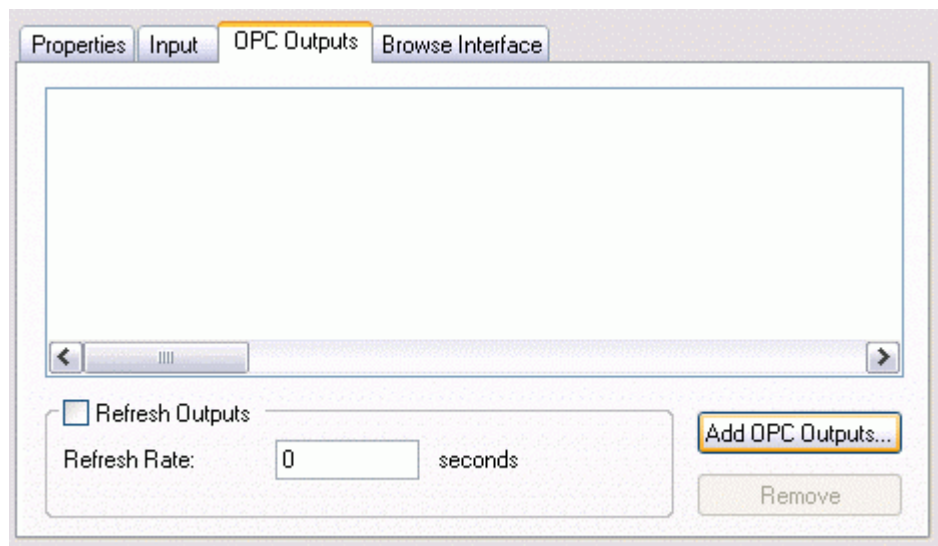
Alias Settings: Input Tab

Alias Settings: OPC Outputs Tab

In the **OPC Outputs** tab in the **Alias Settings** dialog box, shown below, you can select OPC tags and /or registers from the Unified Data Browser by clicking **Add OPC Outputs**.

When bridging OPC data, values are written to the outputs only when the input value changes. This is the default behavior. To modify this behavior, you can force DataWorX32 to refresh outputs periodically. Then the value is written to the outputs even if the input value does not change. You can specify a **Refresh Rate** for outputs by checking the **Refresh Outputs** check box and typing the output refresh rate (in seconds) in the edit box.

Note: You may select more than one tag for the output.



The dialog box shows the 'OPC Outputs' tab selected. It features a large empty list box for output tags. Below the list box is a horizontal scrollbar. At the bottom, there is a checkbox labeled 'Refresh Outputs' which is currently unchecked. To its right is a text field for 'Refresh Rate:' containing the value '0', followed by the word 'seconds'. To the right of these controls are two buttons: 'Add OPC Outputs...' and 'Remove'.

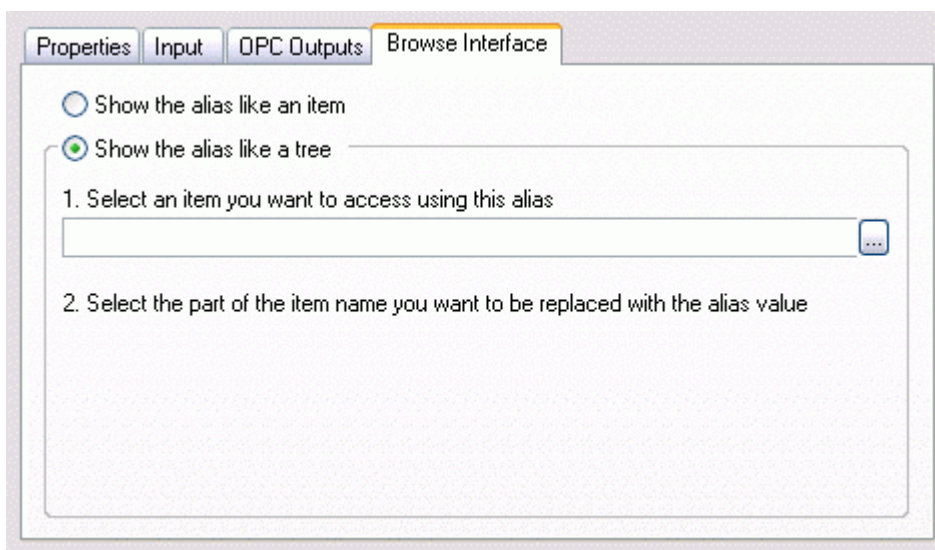
Alias Settings: OPC Outputs Tab

Alias Settings: Browse Interface Tab

In the **Browse Interface** tab in the **Alias Settings** dialog box, shown below, you can define how the alias should appear in the browse interface:

- **Show the alias like an item:** Selecting this option causes the alias to be displayed as a leaf of the tree.
- **Show the alias like a tree:** The subtree of items will appear as a subtree of the alias. In other words, the hierarchy displayed in the browser will mirror the exact tree structure of the alias set up in the group sub-window of DataWorX32.

Then simply follow the two steps indicated on the **Browse Interface** tab. If you choose to show the alias like a tree, first select an item you want to access using the alias. Click the ... button to open the Unified Data Browser. Then select a part of the item name you selected in Step 1. This selected part is the one you intend to replace with the alias.



Alias Settings: Browse Interface Tab

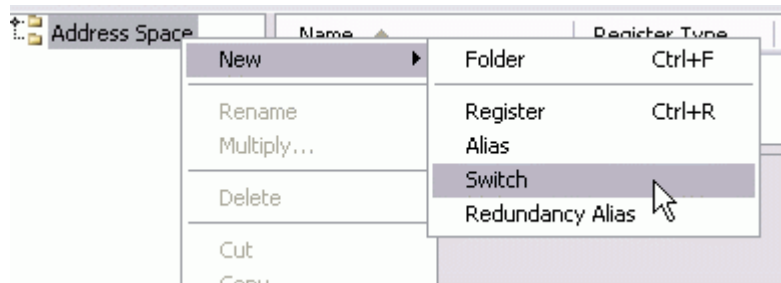
Switch Aliases

A **switch alias** is a special kind of alias. Unlike a regular alias, it has a numeric input. It contains a predefined set of values. The value of the alias is the one of the predefined set that corresponds to the input value.

Creating New Switch Aliases

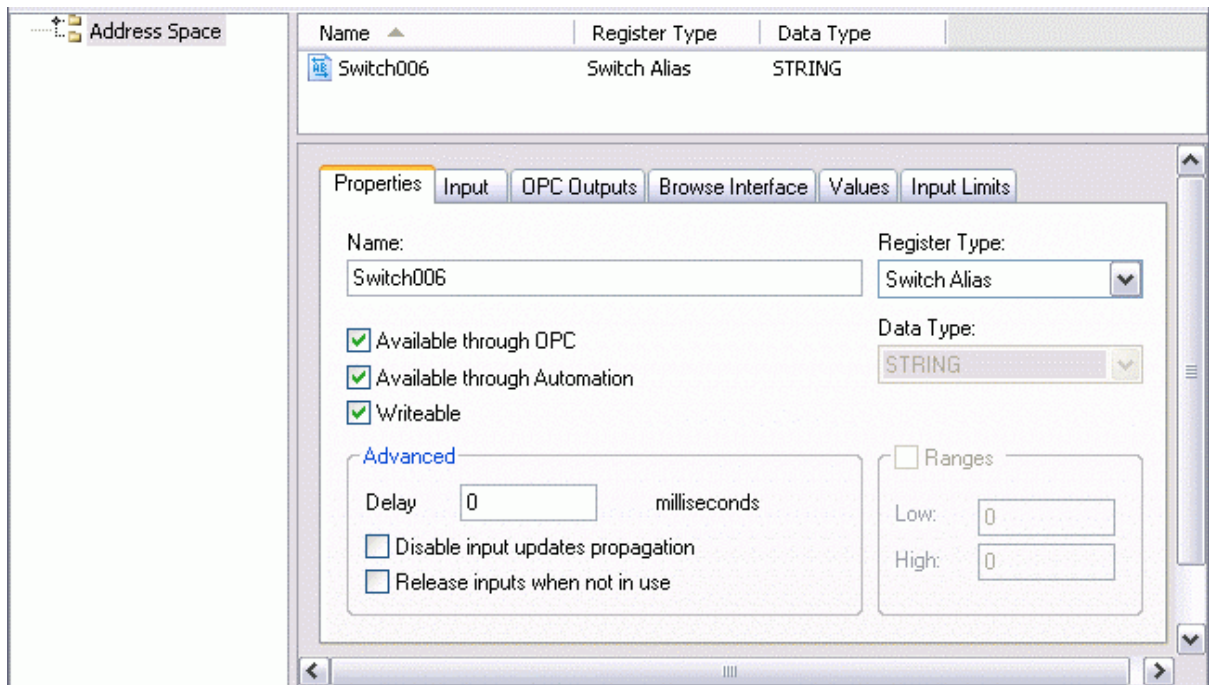
To create a new switch alias:

1. Right-click the **Address Space** tree control of the Configurator and select **New > Switch** from the pop-up menu, as shown in the figure below.



Creating a New Switch Alias

2. The properties dialog box for the new switch alias appears in the right-hand pane of the Configurator, as shown in the figure below.
3. In the **Name** field, type a name for the new switch alias. The name of the switch alias must meet the conditions for register name (unique, consisting of letters, numbers, and underscores).
4. Use the **Values** tab to define the values for the switch alias.
5. When you have finished configuring the switch alias properties, click the **Apply** button. The new switch alias appears under the tree control, as shown in the figure below.

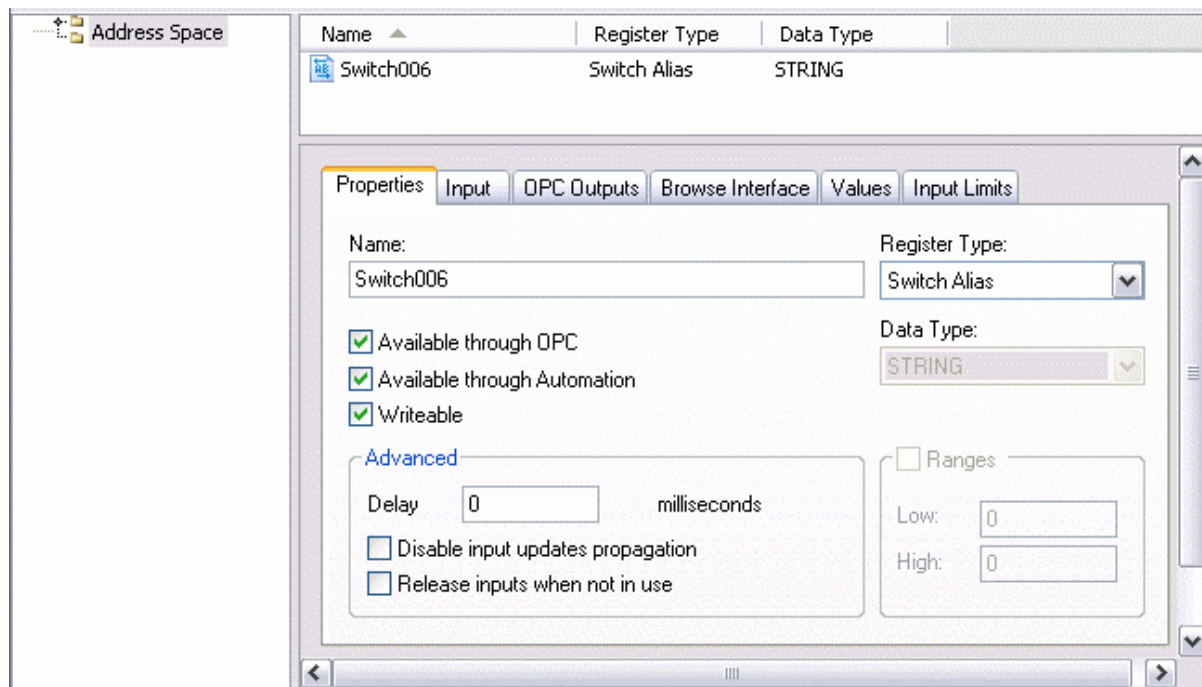


Configuring Switch Alias Properties

Configuring Switch Alias Properties

The switch alias properties dialog box contains the following tabs, as shown in the figure below:

- Properties
- Input
- OPC Outputs
- Browse Interface
- Values
- Input Limits



Configuring Switch Alias Properties

Switch Alias Settings: Properties Tab

The **Properties** tab in the **Switch Alias Settings** dialog box, shown below, allows you to set the following switch alias parameters, as described in the table below.

The screenshot shows the 'Switch Alias Settings' dialog box with the 'Properties' tab selected. The 'Name' field contains 'Switch006'. The 'Register Type' dropdown is set to 'Switch Alias'. The 'Data Type' dropdown is set to 'STRING'. Three checkboxes are checked: 'Available through OPC', 'Available through Automation', and 'Writeable'. The 'Advanced' section is expanded, showing a 'Delay' of 0 milliseconds and two unchecked checkboxes: 'Disable input updates propagation' and 'Release inputs when not in use'. The 'Ranges' section is also expanded, showing 'Low' and 'High' values both set to 0.

Switch Alias Settings: Properties Tab

Properties Tab Parameters

Name	The name of the switch alias must meet the conditions for register name (unique, consisting of letters, numbers, and underscores).
Register Type	Specifies Register, Alias, Switch Alias, or Redundancy Alias.
Available Through OPC	Checking this box makes the register available to OPC clients.
Available Through Automation	Checking this box makes the register available to VB applications so that it could be edited by a VB editor.
Writeable	Checking this box makes the register writeable. Uncheck the box for a read-only OPC client.
Data Type	This field is not available for switch aliases.
Ranges	This field is not available for switch aliases.
High Range	This field is not available for switch aliases.
Low Range	This field is not available for switch aliases.
Delay	You can specify a delay time for the register (in milliseconds) in the box provided.

Disable OPC Propagation Support	When an OPC item is connected to a register as both input and output, a fast sequence of writes to that register may cause item value oscillation. (This may happen when two subsequent writes to the register are faster than the OPC item can perform the write operation. The acknowledge of the first write then overwrites the second written value already being stored in the register.) To avoid this behavior, Check the Disable input updates propagation check box in the register properties. The registers release their inputs when they are not used. 'Not used' means either referenced from the outside (by an OPC client or through automation), or it means they are having an output. Because a register having an output is actually used for bridging, therefore it cannot release its input. Bridging would not work without it. Checking the Release inputs when not in use check box means the registers have to request for input when a client starts using it and release it when the client stops using it. So the registers with this option checked may respond more slowly.
---------------------------------	--

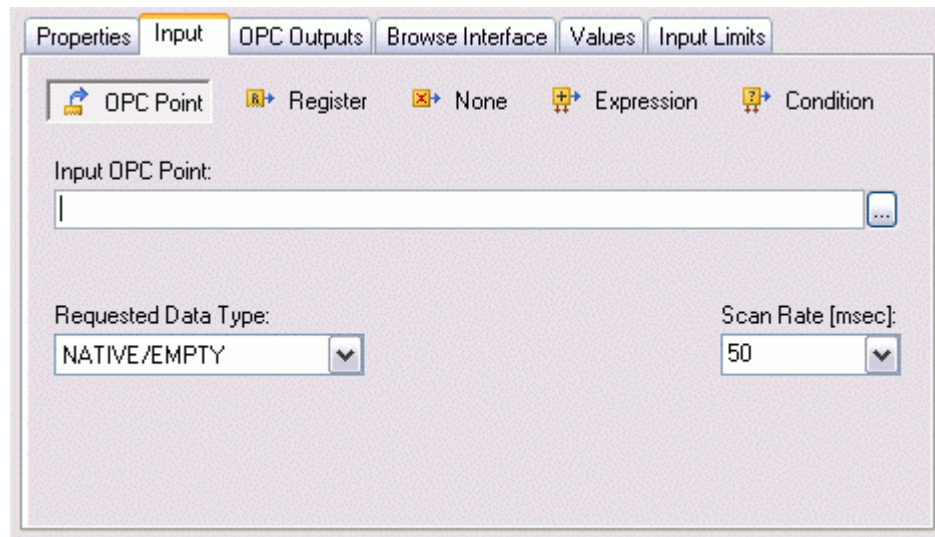
Note: By default, the data type and range information are obtained from the input of register.

Switch Alias Settings: Input Tab

The **Input** tab in the **Switch Alias Settings** dialog box, shown below, allows you to set the following switch alias input parameters:

- OPC Point
- Register
- None
- Expression
- Condition Criteria
- Condition Result
- Condition Inputs

For more information about input settings, please refer to the "Register Settings: Input Tab" section above.



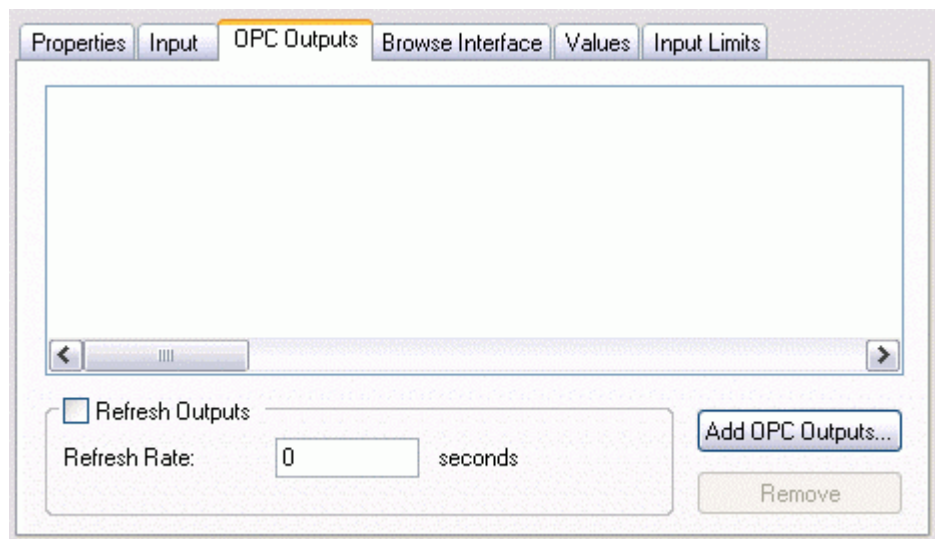
Switch Alias Settings: Input Tab

Switch Alias Settings: OPC Outputs Tab

In the **OPC Outputs** tab in the **Switch Alias Settings** dialog box, shown below, you can select OPC tags and /or registers from the Unified Data Browser by clicking **Add OPC Outputs**.

When bridging OPC data, values are written to the outputs only when the input value changes. This is the default behavior. To modify this behavior, you can force DataWorX32 to refresh outputs periodically. Then the value is written to the outputs even if the input value does not change. You can specify a **Refresh Rate** for outputs by checking the **Refresh Outputs** check box and typing the output refresh rate (in seconds) in the edit box.

Note: You may select more than one tag for the output.



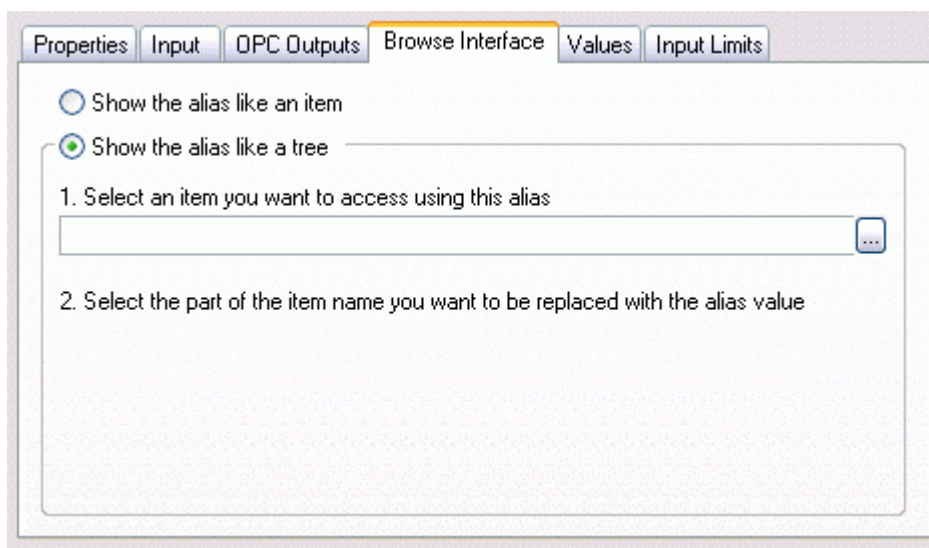
Switch Alias Settings: OPC Outputs Tab

Switch Alias Settings: Browse Interface Tab

In the **Browse Interface** tab in the **Switch Alias Settings** dialog box, shown below, you can define how the switch alias should appear in the browse interface:

- **Show the alias like an item:** Selecting this option causes the alias to be displayed as a leaf of the tree.
- **Show the alias like a tree:** The subtree of items will appear as a subtree of the alias. In other words, the hierarchy displayed in the browser will mirror the exact tree structure of the alias set up in the group sub-window of DataWorX32.

Then simply follow the two steps indicated on the **Browse Interface** tab. If you choose to show the alias like a tree, first select an item you want to access using the alias. Click the ... button to open the Unified Data Browser. Then select a part of the item name you selected in Step 1. This selected part is the one you intend to replace with the alias.



Switch Alias Settings: Browse Interface Tab

Switch Alias Settings: Values Tab

In the **Values** tab in the **Switch Alias Settings** dialog box, shown below, use the **Add New** field and the **Add** button to add new values for the switch alias. Use the **Move Up**, **Move Down**, and **Remove** buttons to change the order of the values. To edit a value, simply click it in the **Value** column. The corresponding numbers for particular values are listed in the **Index** column. The switch alias will be assigned the value corresponding to its numeric input.

Note: The indexes cannot be changed.

The screenshot shows the 'Values' tab of the 'Switch Alias Settings' dialog. At the top, there are tabs: Properties, Input, OPC Outputs, Browse Interface, Values (selected), and Input Limits. Below the tabs, there is an 'Add new:' label followed by an empty text input field. To the right of this field are buttons: 'Add', 'Remove', 'Move Up', 'Move Down', and 'Edit'. Below the input field is a table with two columns: 'Value' and 'In...'. The table is currently empty. At the bottom of the dialog, there is a note: 'The switch alias will be assigned the value corresponding to its numeric input.'

Switch Alias Settings: Values Tab

Switch Alias Settings: Input Limits Tab

When the input is outside the range of the defined indexes, you can select an input type on the **Input Limits** tab in the **Switch Alias Settings** dialog box, shown below. For example, suppose you have defined four values for the switch alias; thus, their indexes are "0," "1," "2," and "3." The **Input Limits** tab allows you to define what should happen when the switch alias's input is less than "0" or greater than "3":

- The value with index "0" can be used
- The MODULUS function can be used.
- It can indicate "bad quality."
- You can also specify a value in the edit box.

The screenshot shows the 'Input Limits' tab of the 'Switch Alias Settings' dialog. At the top, there are tabs: Properties, Input, OPC Outputs, Browse Interface, Values, and Input Limits (selected). Below the tabs, the text 'When input is out of the defined indexes...' is displayed. There are four radio button options: 'Use value with index 0' (selected), 'Calculate index using the modulus function', 'Turn to 'bad quality'', and 'Use this value:'. The 'Use this value:' option has an associated empty text input field.

Switch Alias Settings: Input Limits Tab

Example of Using a Switch Alias

Assume that a Modbus OPC Server with two devices is connected to one PC. Both devices are getting the same process data. The task is to establish redundancy on these devices.

Assume further that the connection from DataWorX32 to the OPC server works well. What can go wrong is the connection from the OPC server to the devices. Redundancy aliases cannot solve this problem as it works on the server level. This is a task for switch aliases and conditions.

1. Assume that a tag connected to the first device is:
ICONICS.ModbusOPCServer\device1.Tag1

The same data can be accessed via the second device as:
ICONICS.ModbusOPCServer\device2.Tag1

2. Now define a switch alias that will switch between the two devices:

Alias name: DEVICE

Set of predefined values: device1, device2

Tag1 mentioned above will be then accessed using the switch alias as:

ICONICS.ModbusOPCServer\[[DEVICE]].Tag1

Depending on the input of the DEVICE alias, either device1 or device2 will be involved.

3. To set up the correct value on the DEVICE alias input, a register with condition will be used. Define such a register:

- Add a new register and choose a name, such as RCOND.
- Go to the **Input** tab, select **Condition > Criteria**.
- Select **First with Good Quality** as the criterion,
- On the **Input** tab, select **Condition > Result**.
- Select **Zero-Based Index** as the result.
- Switch to the **Condition > Inputs** tab, and then add the two tags mentioned above:

ICONICS.ModbusOPCServer\device1.Tag1

ICONICS.ModbusOPCServer\device2.Tag1

The output of the RCOND register is a number that tells which is the first tag with the good quality.

4. Finally, connect the output of the RCOND register to the DEVICE alias.

Redundancy Aliases

The **Redundancy** feature of DataWorX32 allows you to utilize other PCs as backup servers if a primary server goes offline. This means that you can designate alternative PC machines as backup servers if a designated primary server goes offline. Once a primary server does go offline, DataWorX32 will default to the backup server or servers in the sequence in which the backup servers were designated.

If the appropriate option is selected in the dialog box provided for this purpose, DataWorX32 will default to the primary server once it returns online.

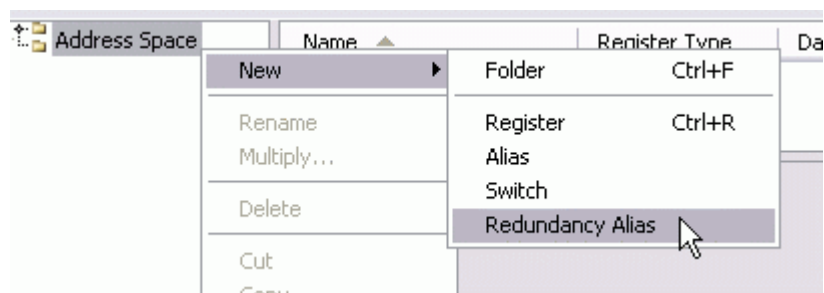
The main qualifying factors for using the redundancy feature in DataWorX32 are as follows:

1. You must designate one OPC server as the "Primary" server in each set.
2. You may designate one or more OPC servers as the "Backup" server(s) in each set. (The number of servers is not restricted by DataWorX32 itself; rather it is limited only by system resources). If more than one backup server is specified, they should be ordered (2nd, 3rd, 4th, etc.) You will see a message outlining the details of the discrepancies and will be allowed to either accept it as is or permit reconfiguration.
3. The various OPC client applications request data from DataWorX32, rather than from the OPC server directly. This way if a primary OPC server failure occurs (due to any number of conditions), an automatic switchover to the backup OPC server occurs.
4. DataWorX32 monitors the OPC servers and aggregates the data to the requesting clients.

Creating New Redundancy Aliases

To create a new redundancy alias:

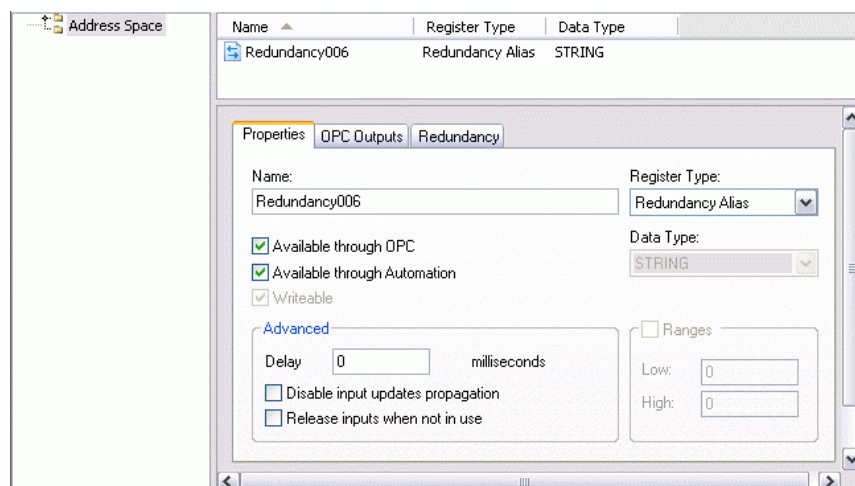
1. Right-click the **Address Space** tree control of the Configurator and select **New > Redundancy Alias** from the pop-up menu, as shown in the figure below.



Creating a New Redundancy Alias

2. The properties dialog box for the new redundancy alias appears in the right-hand pane of the Configurator, as shown in the figure below.
3. In the **Name** field, type a name for the new redundancy alias.

4. When you have finished configuring the redundancy alias properties, click the **Apply** button. The new redundancy alias appears under the tree control, as shown in the figure below.

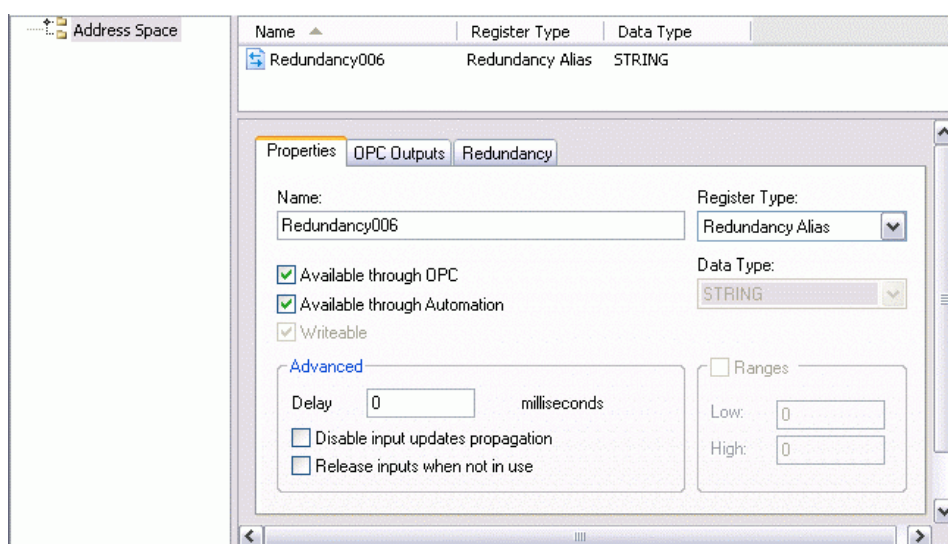


Configuring Redundancy Alias Properties

Configuring Redundancy Alias Properties

The redundancy alias properties dialog box contains the following tabs, as shown in the figure below:

- Properties
- OPC Outputs
- Redundancy



Configuring Redundancy Alias Properties

Redundancy Alias Settings: Properties Tab

The **Properties** tab in the **Redundancy Alias Settings** dialog box, shown below, allows you to set the following redundancy alias parameters, as described in the table below.

The screenshot shows the 'Properties' tab of the 'Redundancy Alias Settings' dialog. The 'Name' field contains 'Redundancy006'. The 'Register Type' dropdown is set to 'Redundancy Alias'. The 'Data Type' dropdown is set to 'STRING'. There are three checked checkboxes: 'Available through OPC', 'Available through Automation', and 'Writeable'. Below these is an 'Advanced' section with a 'Delay' of 0 milliseconds, and two unchecked checkboxes: 'Disable input updates propagation' and 'Release inputs when not in use'. To the right of the 'Advanced' section is a 'Ranges' section which is currently collapsed, showing 'Low' and 'High' range fields.

Redundancy Alias Settings: Properties Tab

Properties Tab Parameters

Name	A common tag name.
Register Type	Specifies Register, Alias, Switch Alias, or Redundancy Alias.
Available Through OPC	Checking this box makes the register available to OPC clients.
Available Through Automation	Checking this box makes the register available to VB applications so that it could be edited by a VB editor.
Writeable	This field is not available for redundancy aliases.
Data Type	This field is not available for redundancy aliases.
Ranges	This field is not available for redundancy aliases.
High Range	This field is not available for redundancy aliases.
Low Range	This field is not available for redundancy aliases.
Delay	You can specify a delay time for the register (in milliseconds) in the box provided.

Disable OPC Propagation Support	When an OPC item is connected to a register as both input and output, a fast sequence of writes to that register may cause item value oscillation. (This may happen when two subsequent writes to the register are faster than the OPC item can perform the write operation. The acknowledge of the first write then overwrites the second written value already being stored in the register.) To avoid this behavior, Check the Disable input updates propagation check box in the register properties. The registers release their inputs when they are not used. 'Not used' means either referenced from the outside (by an OPC client or through automation), or it means they are having an output. Because a register having an output is actually used for bridging, therefore it cannot release its input. Bridging would not work without it. Checking the Release inputs when not in use check box means the registers have to request for input when a client starts using it and release it when the client stops using it. So the registers with this option checked may respond more slowly.
---------------------------------	--

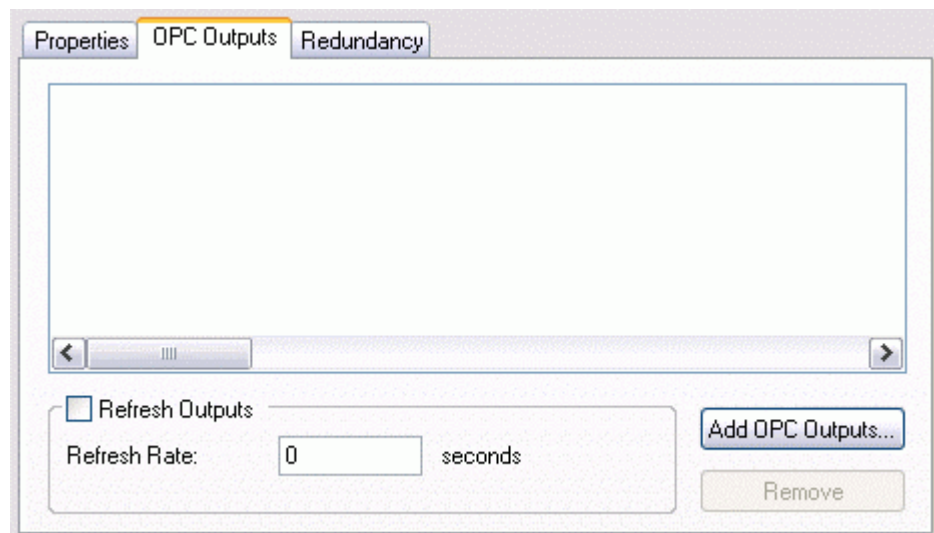
Note: By default, the data type and range information are obtained from the input of register.

Redundancy Alias Settings: OPC Outputs Tab

In the **OPC Outputs** tab in the **Redundancy Alias Settings** dialog box, shown below, you can select OPC tags and /or registers from the Unified Data Browser by clicking **Add OPC Outputs**.

When bridging OPC data, values are written to the outputs only when the input value changes. This is the default behavior. To modify this behavior, you can force DataWorX32 to refresh outputs periodically. Then the value is written to the outputs even if the input value does not change. You can specify a **Refresh Rate** for outputs by checking the **Refresh Outputs** check box and typing the output refresh rate (in seconds) in the edit box.

Note: You may select more than one tag for the output.



Redundancy Alias Settings: OPC Outputs Tab

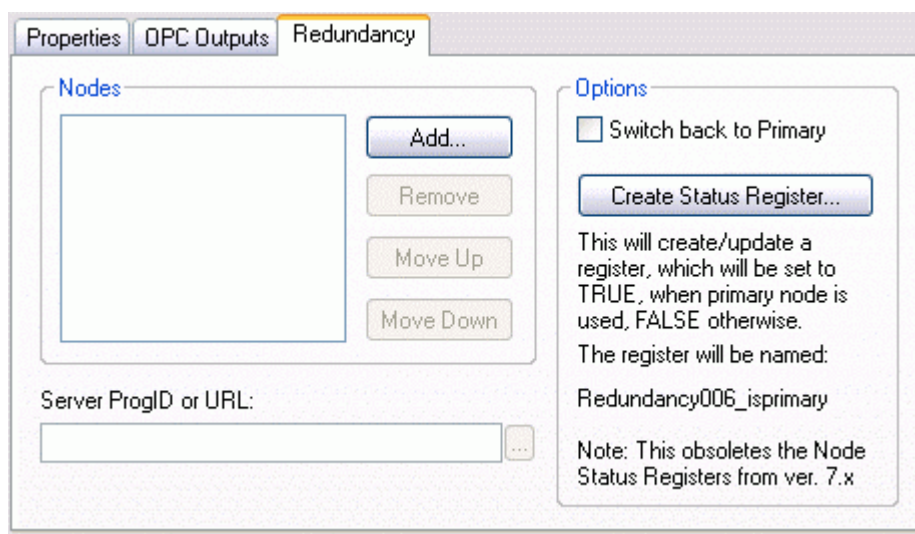
Redundancy Alias Settings: Redundancy Tab

In the **Redundancy** tab in the **Redundancy Alias Settings** dialog box, shown below, you can define select redundant nodes/servers. Click the **Add** button to browse for computers on the network through the Unified Data Browser to be designated as a primary or secondary node. Click the **Move Up** and **Move Down** buttons to change the order of the nodes. Clicking the **Remove** button deletes a network node from the list of primary or secondary nodes.

The **Server ProgID or URL** field displays the name of the server from which the designated PC is receiving its input. Click the ... button to browse a list of OPC servers in order to choose a server from which the PC that will be designated as a primary or secondary node will receive its input.

When DataWorX32 detects that a primary server is running again (after having crashed), it switches back to the primary server from the backup server. If the **Switch Back to Primary** check box is checked, you will be asked before DataWorX32 switches back to the primary server. This is in case you may not want to the primary server for some reason (e.g. the server is in testing mode).

Clicking the **Create Status Register** button allows you to specify whether the primary node is online or offline using the integers "0" or "1." It creates/updates a register that will be set to TRUE when the primary node is used, and set to FALSE otherwise. This register could be very helpful in triggering alarms and starting/stopping data-logging.



Redundancy Alias Settings: Redundancy Tab

Runtime Operations

3

DataWorX32 Runtime Environment

The runtime environment of DataWorX32 lets you view real-time data as they are transmitted between the servers on the network. This is the state in which you can view the live data being received according to the parameters you set in configuration mode. The aggregation feature of DataWorX32 is especially useful during runtime mode. A register with its input connected to an OPC point can be used to "prerequisite" items from the OPC servers. Because registers remain in existence during the entire runtime operation, the items connected to their inputs are also requested for the entire duration of runtime. Because of the feature of OPC aggregation, clients can then request these items without a delay.

During runtime, you can perform the following tasks:

- Open a DataWorX32 file.
- Edit the value of a register.
- Change the value of an alias.
- Edit an alias and change its default value.

Starting and Stopping Runtime Mode

You can switch to runtime mode by selecting **Start Runtime** from the **Action** menu in configuration mode. You can stop runtime mode by selecting **Stop Runtime** from the **Action** menu. Clicking the traffic light button on the toolbar, shown at left, also commences or stops runtime mode. The traffic light icon switches from red to green once you switch from the configuration mode to runtime mode.

Note: The traffic light button on the toolbar starts/stop the runtime mode, not the DataWorX Runtime itself. The DataWorX Runtime (it is an OPC server) is started when needed and runs forever (until stopped by GenTray, or when no clients access it, by the Configurator).

The runtime module loads its configuration from the active configuration database. The Configurator sets this setting by the **Make Active** command on the **File** menu of the Configurator. During runtime mode, the runtime monitor shows live data for the items displayed. It logs the start and stop of runtime as many times as runtime is started and stopped. It also displays messages giving the current status of the tags. Selecting **Update Runtime Configuration** from the **File** menu forces the DataWorX32 runtime engine to reload any changes made in to the configuration database since runtime mode was started.

When the **DwxRuntime** is in memory (running), it is accessible to OPC clients, which may connect to it. If DwxRuntime is in runtime mode, it connects to the underlying OPC servers and gets data, does data bridging, etc. If DwxRuntime is in memory but not in runtime mode, OPC clients can connect to it but they do not get any data.

Monitor View

The Configurator includes a runtime monitor for viewing live server data. To change to the monitor view, select **Monitor View** from the **View** menu. The runtime monitor appears in the bottom pane of the Configurator screen, as shown in the figure below. During runtime, the monitor scans the displays the tag values and other statistics such as date, time, and quality.

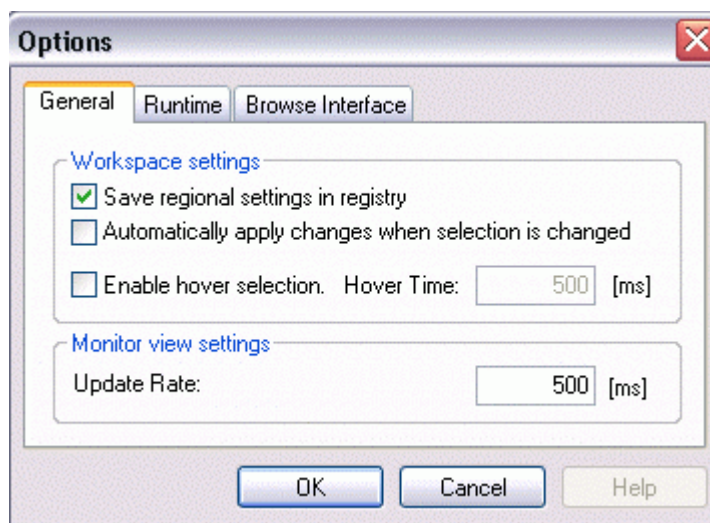
Monitoring is enabled for each item with a check mark next to it. To enable/disable monitoring for an item, you can click on the box to the left of the item. A check mark inside the box means the item is enabled for monitoring. If there is no check mark, then the item is disabled.

Item ID	Value	Timestamp	Quality	Subquality	Lin
☒ LogicalDataItem	0 (VT_BOOL)	09/29/03 17:40:39.558	Good	Non-specific	No
☒ NumericDataItem	0.053987243263039 (V...	09/29/03 17:40:40.058	Good	Non-specific	No
☒ TextualDataItem	"HqM(Ubiqo;q)f" (VT_BS...	09/29/03 17:40:40.058	Good	Non-specific	No

Runtime Monitor View

Monitor View Refresh Rate

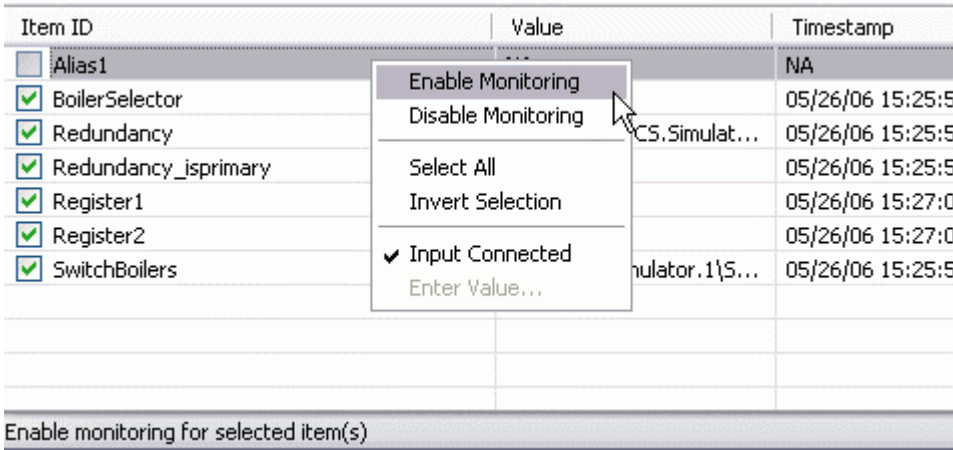
To set the **OPC Monitoring Update Rate** (in milliseconds), select **Options** from the **Tools** menu and click on the **General** tab of the **Options** dialog box, as shown in the figure below.



Options Dialog Box: General Tab

Enable Monitoring

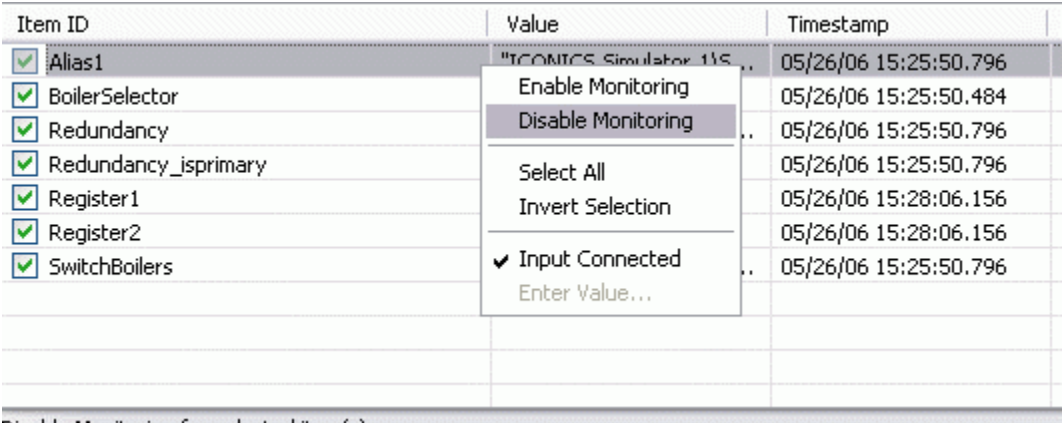
Monitoring is enabled for each item with a check mark next to it. To enable/disable monitoring for an item, you can click on the box to the left of the item. A check mark inside the box means the item is enabled for monitoring. If there is no check mark, then the item is disabled. To enable monitoring for an unchecked item in the monitor view, you can also right-click on the item and select **Enable Monitoring** from the pop-up menu, as shown in the figure below.



Enabling Monitoring for an Item

Disable Monitoring

Monitoring is enabled for each item with a check mark next to it. To enable/disable monitoring for an item, you can click on the box to the left of the item. A check mark inside the box means the item is enabled for monitoring. If there is no check mark, then the item is disabled. To disable monitoring for a checked item in the monitor view, you can also right-click on the item and select **Disable Monitoring** from the pop-up menu, as shown in the figure below.



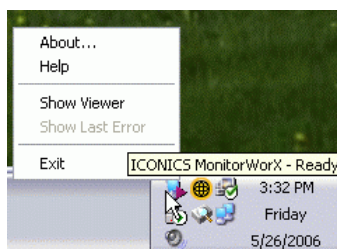
Disabling Monitoring for an Item

Running DataWorX32 As a Service

In general, DataWorX32 can be registered and run as a service as well as a stand-alone COM server. However, the following differences apply:

To configure DataWorX32 to run as a service, do the following:

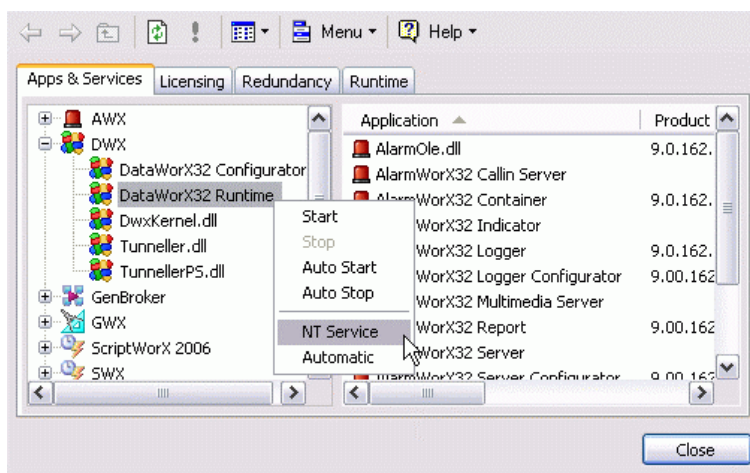
1. Install GENESIS32 or DataWorX32.
2. Use the DataWorX32 Configurator to configure your application.
3. Once the application setup is completed, use MonitorWorX to register DataWorX32 as an NT Service. To do this, start MonitorWorX from the Windows **Start** menu by selecting **Programs > ICONICS GENESIS32 > MonitorWorX**.
4. The MonitorWorX icon appears in the task bar. Right-click on this icon and select **Show Viewer**, as shown in the figure below.



Launching the MonitorWorX Viewer

5. In the **Apps & Services** tab of the MonitorWorX Viewer, expand the **DWX** tree control, as shown in the figure below. Right-click on **DataWorX32 Runtime** and select **NT Service** from the pop-up menu, as shown in the figure below.

Note: You can also set DataWorX32 to start automatically as an NT service the next time the PC is rebooted; select **DataWorX32 Runtime > Automatic** from the pop-up menu.



Registering DataWorX32 As an NT Service

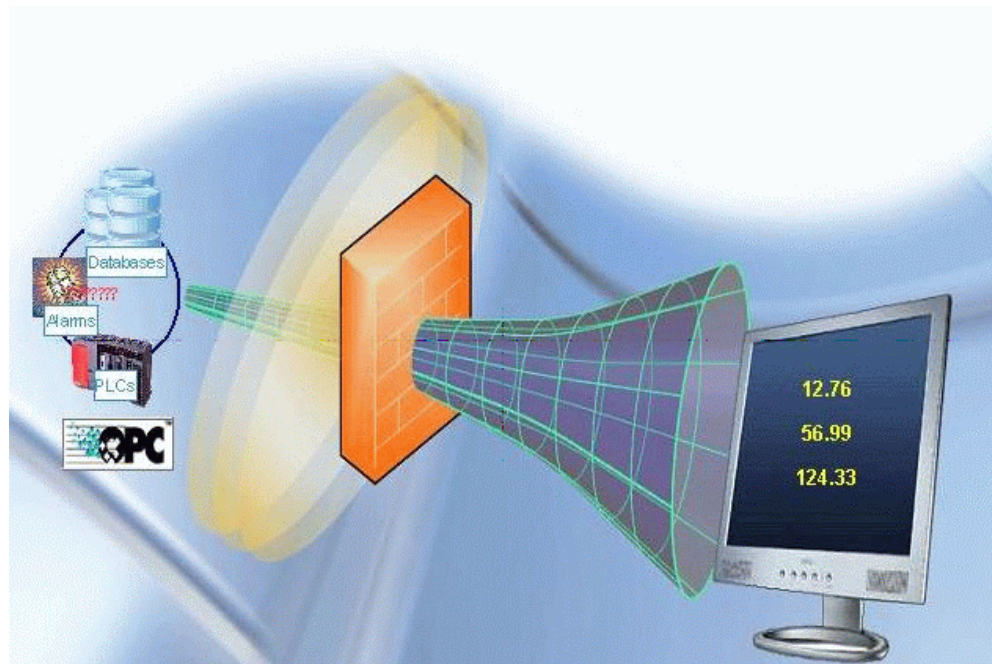
OPC Tunneling

4

Introduction to DataWorX32 OPC Tunneling

DataWorX™32 OPC Tunneling connects remote OPC servers to a local computer and also supports third-party OPC server-to-OPC client communications. OPC Tunneling fully supports the OPC Foundation's latest Data Access (DA), Historical Data Access (HDA), and Alarm and Events (AE) specifications.

OPC Tunneling, which comes as a standard part of DataWorX32 V9, connects a remote OPC server to a local client in a robust and secure manner, allowing for one server to be redirected to more than one location. The powerful graphical user interface allows for easy configuration and a centralized place to manage all remote connections. The underlying technology behind OPC Tunneling is the patented ICONICS **GenBroker™** communication, which provides high-performance and robust communication, providing an alternative to conventional Microsoft DCOM communications. OPC Tunneling is completely OPC-compliant and IT firewall-friendly, supporting communications over LANs, WANs and the Internet with extensive built-in security.



DataWorX32 OPC Tunneling Technology

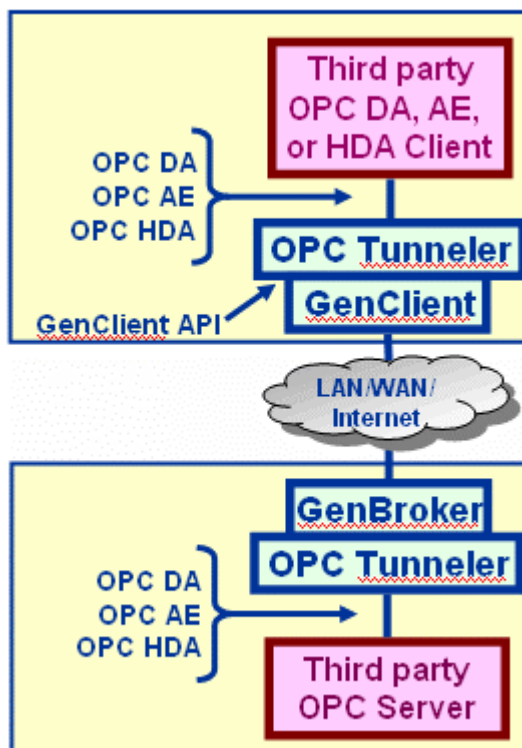
Key Features of DataWorX32 OPC Tunneling Technology

OPC "tunneling" technology is simple to install and deploy to virtually any application requiring remote and secure OPC communications with DataWorX32. DataWorX32 OPC tunneling fully supports the latest open OPC industry standards, including:

- OPC Data Access 3.0
- OPC Alarm and Events 1.1
- OPC Historical Data Access 1.2

Other significant features of DataWorX32 OPC Tunneling include:

- Auto-discovery of remote OPC DA, AE and HDA servers
- Simple to set up and configure
- Supports OPC browser interfaces over LANs, WANs and the Internet
- Provides a robust, secure alternative to standard Microsoft DCOM communications
- Integrated secure communications
- IT-friendly communications through firewalls and Network Address Translators (NAT)
- Supports TCP/IP and SOAP/XML communication protocols



OPC Tunneling Architecture

GENESIS32 Applications Used in OPC Tunneling

DataWorX32 OPC tunneling configuration and runtime operations use the following GENESIS32 applications:

- **GenBroker Configurator:** Enables communication to remote OPC servers via TCP/IP or SOAP/XML channels.
- **OPC Tunneling Configurator:** Connects to remote OPC servers, and enables third-party OPC client and OPC server connectivity.
- **DataWorX32 Configurator:** Launches the GenBroker Configurator and the OPC Tunneling Configurator.
- **Unified Data Browser:** Enables browsing for data tags on OPC servers connected through OPC Tunneling.

Setting up GenBroker for OPC Tunneling

DataWorX32 OPC Tunneling provides a powerful alternative to conventional DCOM communications, enabling remote OPC communications over LANs, WANs and the Internet using TCP/IP and SOAP/XML protocols. DCOM's major disadvantages are:

- Its complex settings often make its configuration ineffective when the client is remote.
- It does not work well across NT domains.
- Its connection over slow and unreliable networks (e.g., the Internet) is difficult.
- It is not easy to configure to pass through firewalls.

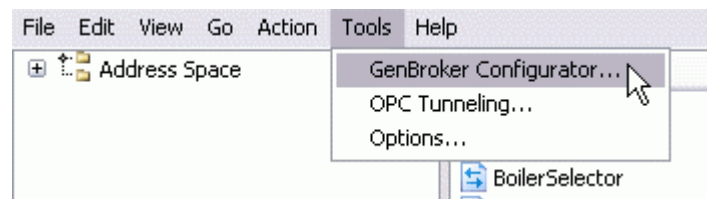
In response to these difficulties, ICONICS has enhanced GENESIS32 with **GenBroker**, a proven, high-speed communications utility. Employing a network of applications, GenBroker uses TCP/IP and SOAP/XML channels to achieve real-time and secure communications between Web clients and servers. GenBroker communication has the following advantages:

- It works well within, across, and in the absence of NT domains.
- It works well on slow and/or unreliable networks (e.g., the Internet).
- The client side works well behind a firewall.

To use OPC Tunneling you must configure GenBroker on the client and start GenBroker on the server, as described in the following sections.

Configuring GenBroker on the Client Side

The **GenBroker Configurator** application enables you to set up TCP/IP and SOAP/XML communication channels for different nodes on your LAN, WAN, or over the Internet. To launch the GenBroker Configurator, select **GenBroker Configurator** from the **Tools** menu in the DataWorX32 Configurator, as shown in the figure below.



Launching the GenBroker Configurator from DataWorX32

For OPC Tunneling, configure GenBroker on the client side to use TCP/IP or SOAP/XML as the default channel, and make the configuration active. GenBroker does not need to be started on the client, but the configuration must be activated.

Setting up Communication Channels in GenBroker Configurator

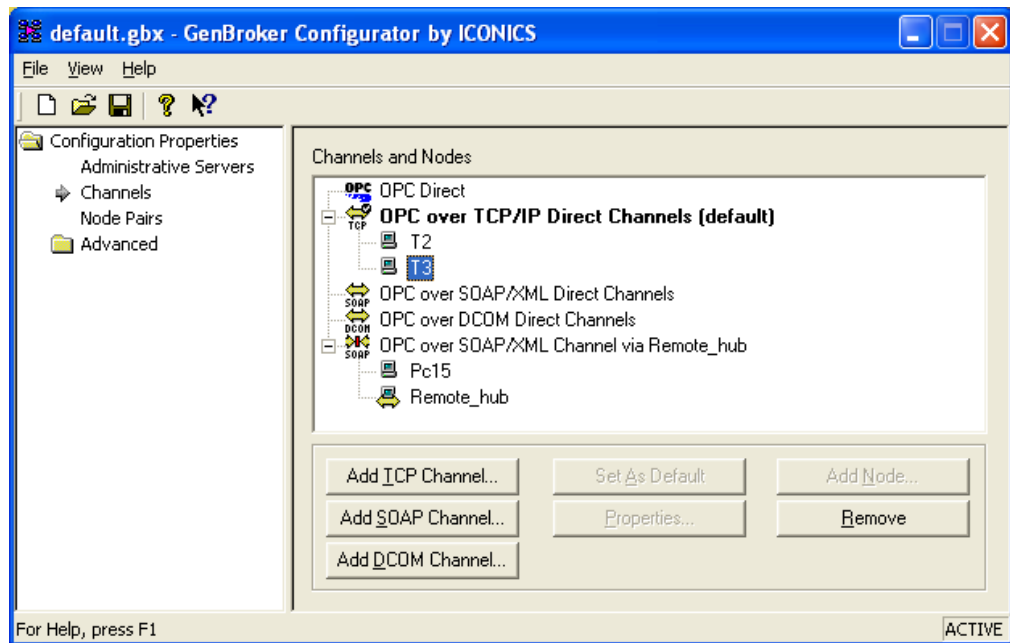
The GenBroker Configurator enables you to customize your network architecture:

- Specify the communication method.
- Configure the GenBroker Client.
- Configure the GenBroker Server.
- Assign nodes to direct channels.
- Assign mediator nodes for indirect channels.
- Specify the default channel.
- Add and remove nodes or IP addresses.
- Configure channel properties.
- Specify the location of primary and redundant remote servers.

The configuration settings are saved in a GenBroker configuration (.gbx) file. You can store multiple configuration files. Configuration files tell GenBroker Server how to communicate when it interacts with various clients.

The figure below shows an example of how different communication channels and also mediator nodes can be set up in the GenBroker Configurator user interface.

Note: For complete information about GenBroker, please refer to the GenBroker Configurator Help documentation.



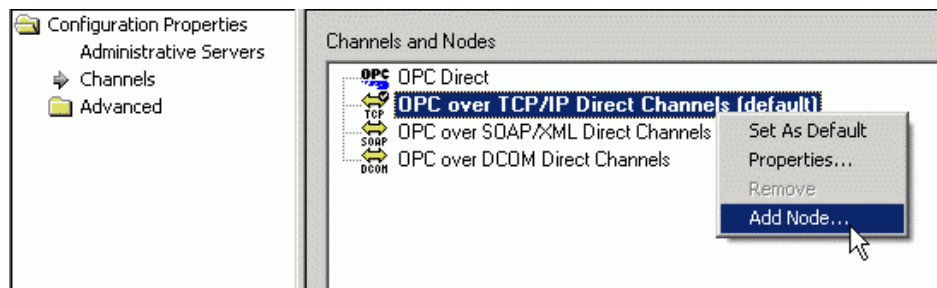
Using GenBrokers to Enable Communication Channels for OPC Tunneling

Adding Nodes to Channels

You can define which nodes on the network will be accessed from the current node that uses the method of communication you have chosen. Adding nodes to a channel allows you the advantage of having more than one node in that direct channel.

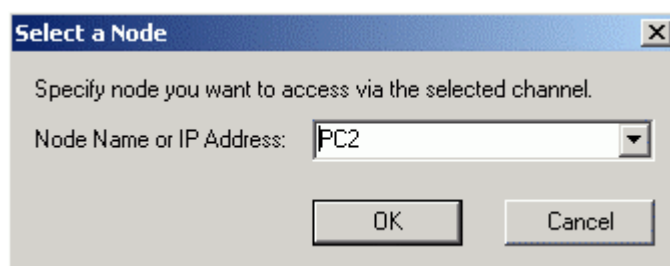
To add a node to a communication channel:

1. Right-click the channel and select **Add Node** from the pop-up menu, as shown in the figure below.



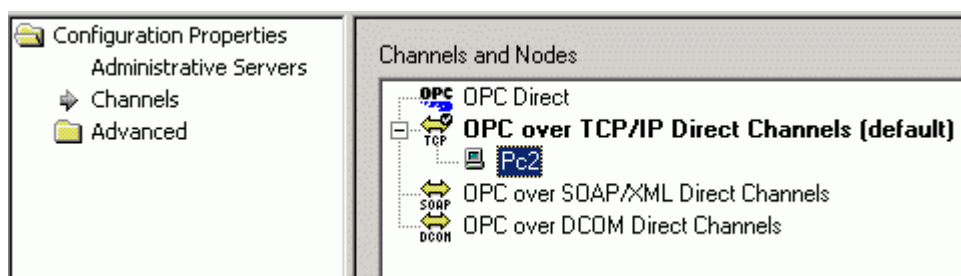
Adding a Node to a Channel

2. The **Select a Node** dialog box opens, as shown in the figure below. There are two ways of choosing a node name and/or IP address. The first way is by choosing one from the drop-down list. If the nodes are not currently connected to the computer being used for the configuration, the second way of choosing a node name or IP address is to type one in the **Node Name or IP Address** field. Click the **OK** button when you have finished choosing a node name or IP address.



Specifying a Node to Access Via the Selected Communication Channel

3. The node name appears below the **Channel** icon on the Configurator screen, as shown in the figure below.

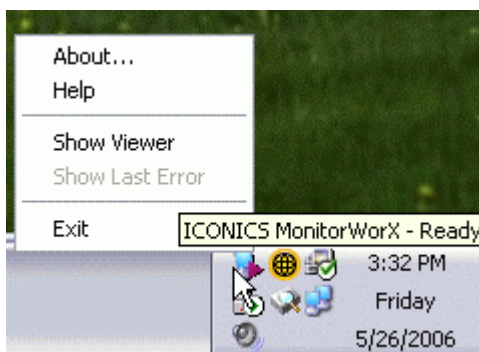


Node Name Added to Communication Channel

Starting GenBroker Server

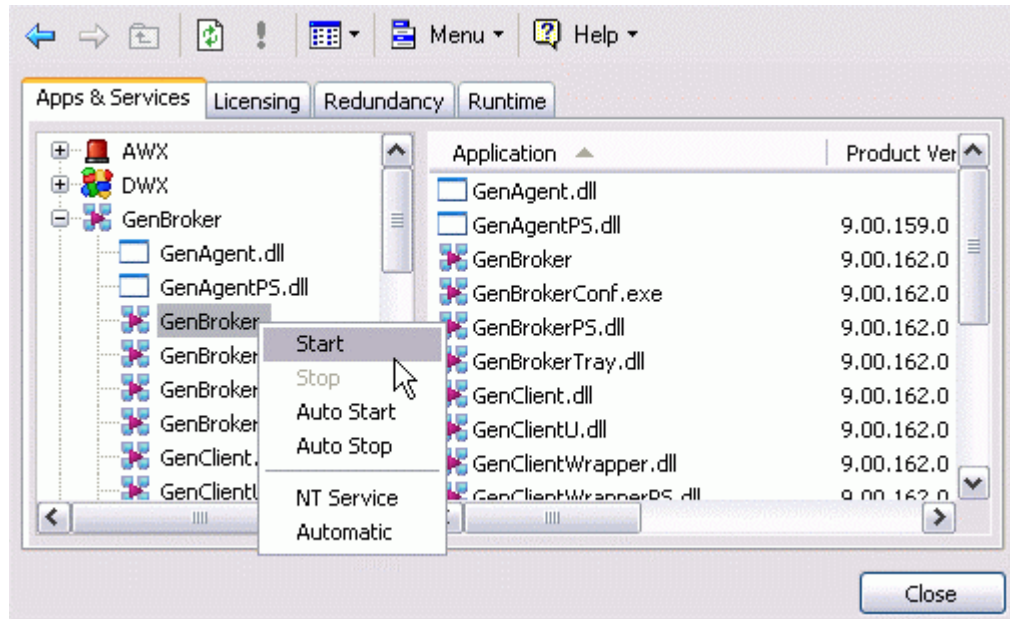
To configure GenBroker for OPC Tunneling on the client side in order for clients to communicate with the GenBroker Server (e.g. to download and view published HTML files on the client Web browser), you must activate GenBroker on the server node.

1. To do this, start MonitorWorX from the Windows **Start** menu by selecting **Programs > ICONICS GENESIS32 > MonitorWorX**.
2. The MonitorWorX icon appears in the task bar. Right-click on this icon and select **Show Viewer**, as shown in the figure below.



Launching the MonitorWorX Viewer

3. In the **Apps & Services** tab of the MonitorWorX Viewer, expand the **GenBroker** tree control, as shown in the figure below. Right-click **GenBroker** and select **Start** from the pop-up menu, as shown in the figure below. This activates the GenBroker Server.



Starting the GenBroker Server on the Server Node

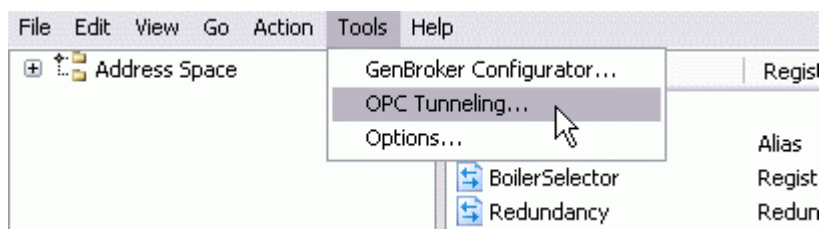
Note: MonitorWorX cannot run when the license is located on a remote machine and the GenBroker on the remote machine is enabled. Even if the GenBroker on the remote machine is not enabled, you will not be able to run GenBroker from MonitorWorX. Instead of running GenBroker from MonitorWorX, in these situations you can run GenBroker from the command prompt. You have to know the full path for the location of the **GenBroker.exe** file. Select **Run** from the Windows **Start** menu, and then run the GenBroker.exe file from the command prompt.

Note: If changes need to be made to the GenBroker configuration on the client, it is recommended that you stop all GENESIS32 processes and then restart them after the changes are completed. Any changes made to the GenBroker configuration while any GENESIS32 clients are running will not take effect until the clients are closed and the processes are restarted.

Connecting to Remote OPC Servers Using OPC Tunneling

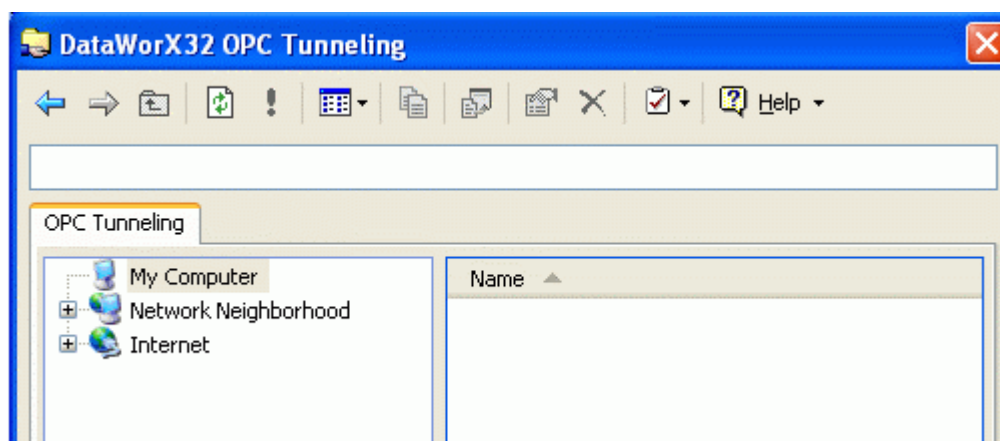
Once you have set up your TCP/IP or SOAP/XML communication channels between network nodes using the GenBroker Configurator, you are ready to create "tunnels" to various OPC DA, AE, and HDA servers. The **OPC Tunneling Configurator** is the application that enables you to connect to remote OPC servers, and also enables connections between third-party (non-ICONICS) OPC clients and OPC servers. To set up a connection to a remote OPC server:

1. Launch the OPC Tunneling Configurator by selecting **OPC Tunneling** from the **Tools** menu in the DataWorX32 Configurator, as shown in the figure below.



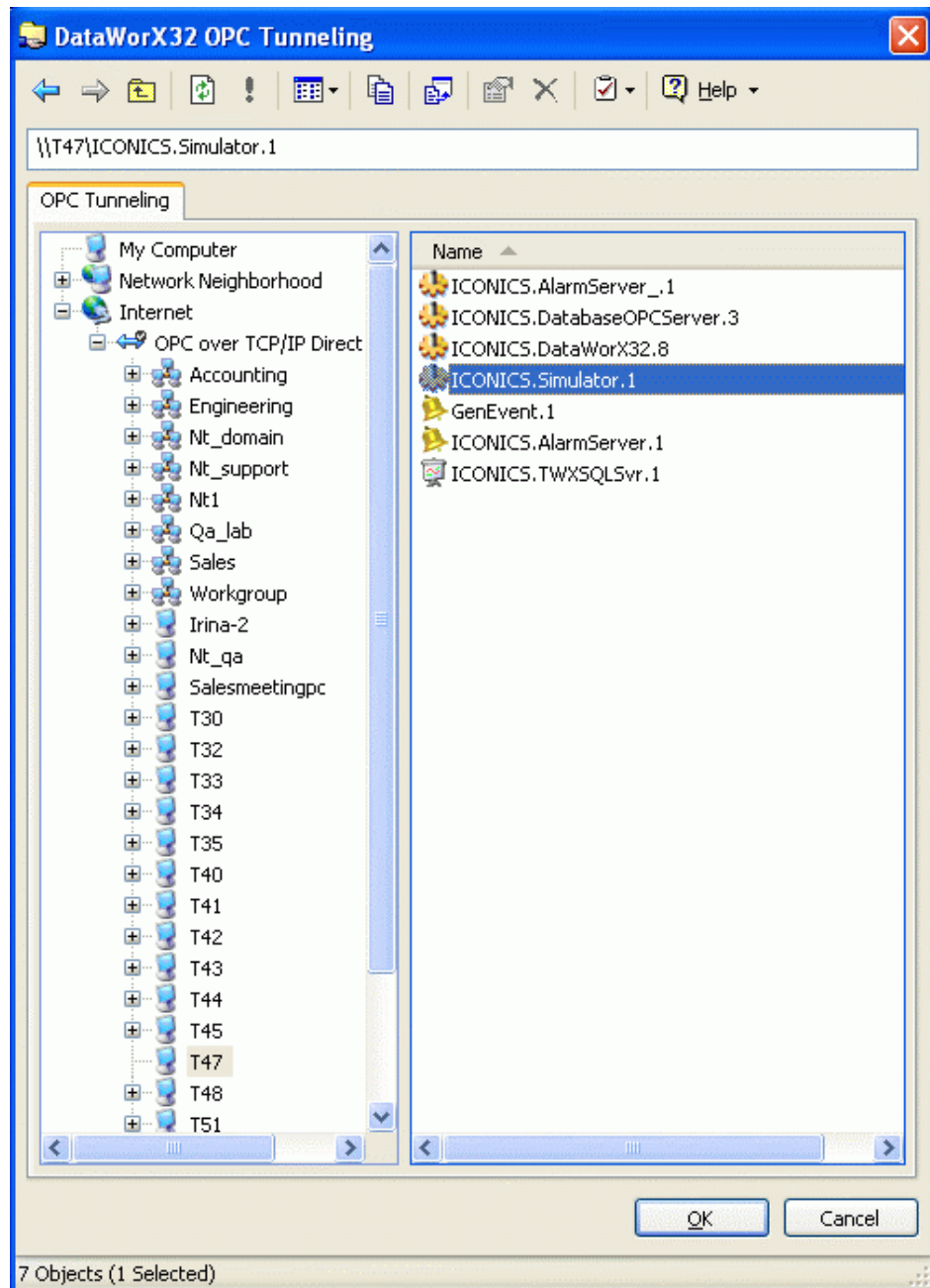
Launching OPC Tunneling From DataWorX32

2. This opens the DataWorX32 OPC Tunneling Configurator interface, as shown in the figure below.



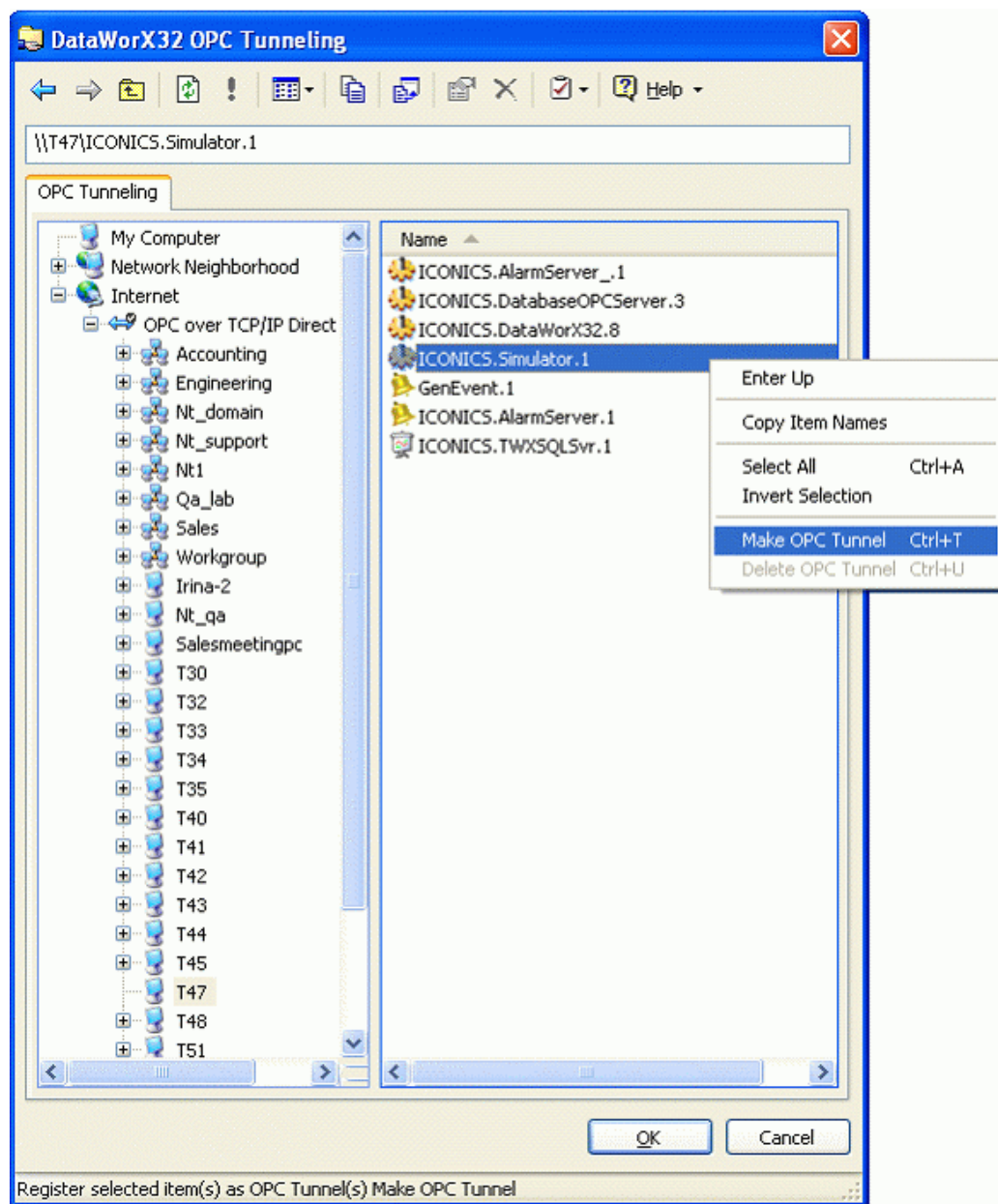
OPC Tunneling Configurator

3. Next you must tell the OPC Tunneling Configurator where to find the target remote OPC server. In the **OPC Tunneling** tab, click on the **Internet** tree control in the left pane and browse to the remote node (computer) on which the target OPC server is located, as shown in the figure below. When you select the node, the OPC servers on that node are displayed in the right pane. For example, in the figure below, the **ICONICS.Simulator.1** OPC server is selected on remote node **T47**. The edit box above the OPC Tunneling tab automatically displays the network path for the selected node and OPC Server. In the example below, the path is **\\T47\ICONICS.Simulator.1**.



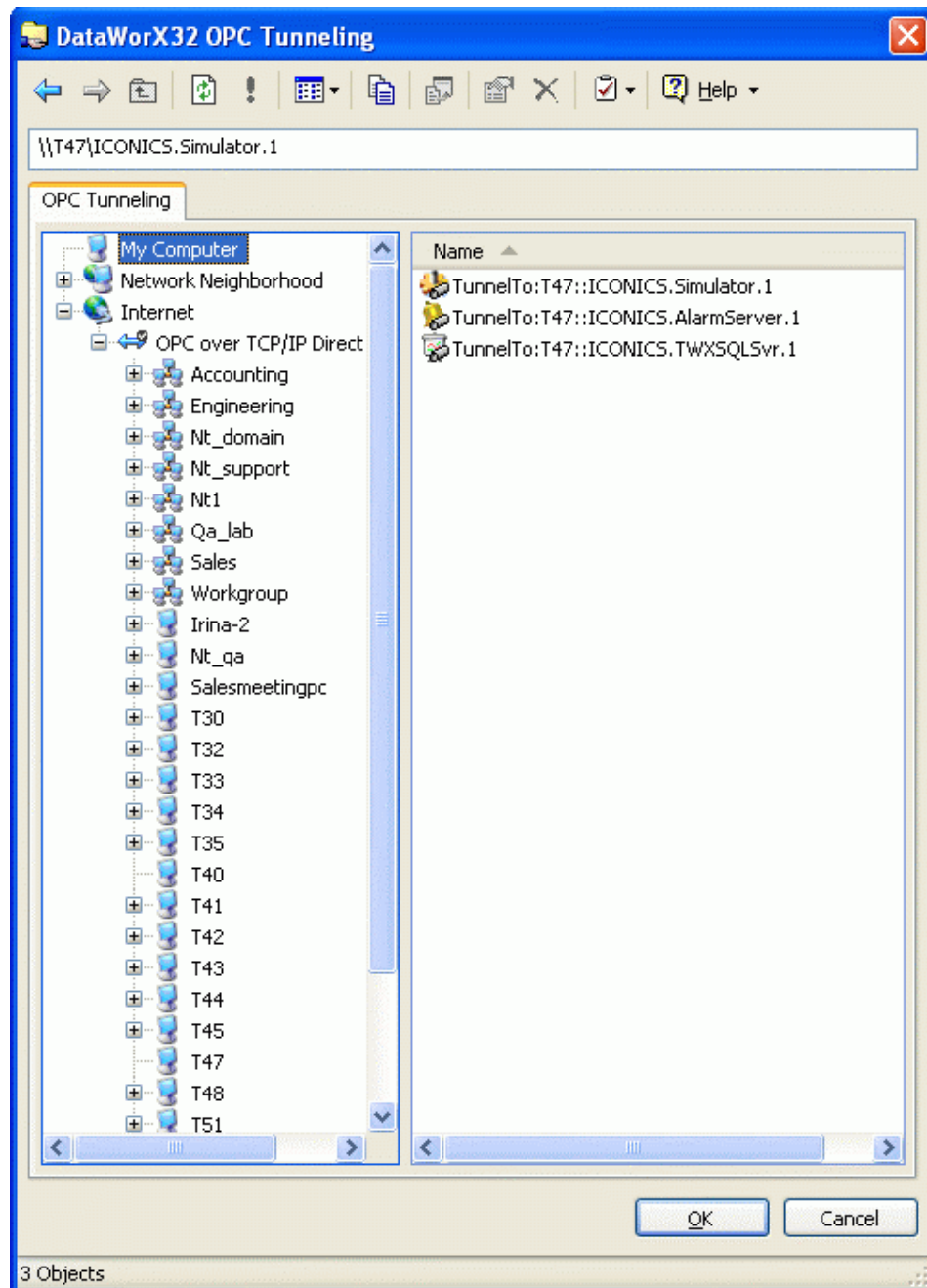
Browsing for a Remote OPC Server

4. To establish the "tunnel" to the selected remote OPC server, right-click on the OPC server and select **Make OPC Tunnel** from the pop-up menu, as shown in the figure below.



Creating a Tunnel to the Remote OPC Server

5. Once you have created a tunnel to the target OPC server, the OPC server appears on the local node (client) computer. To view the tunnel connection, click the **Refresh** button, and then click on **My Computer** (the local client computer) in the tree control. The tunnel connection is displayed in the right pane. In the example below, the tunnel connection appears as **TunnelTo:T47::ICONICS.Simulator.1**. **Note:** The syntax for tunneled servers includes a 'TunnelTo:<ServerNodeName>::' prefix.

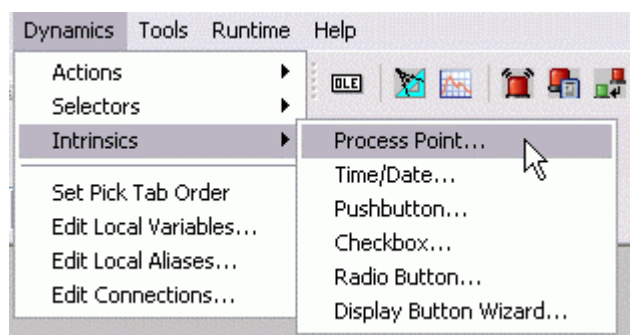


Tunnel Connection Established on Local Computer

Browsing for Data on the Connected Remote OPC Server

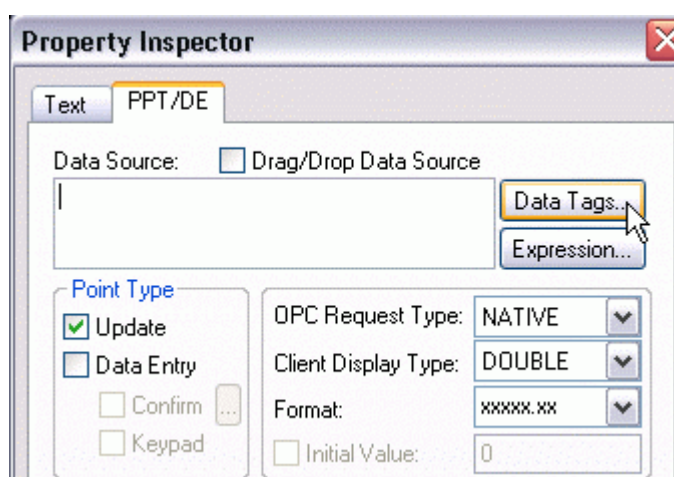
Once the tunnel to the remote OPC server is set up, you can visualize the tunnel connection; it appears as if the OPC server has been installed on your client machine. Thus, the connected OPC server can be accessed by an OPC client application. The OPC server appears in the **Unified Data Browser**, which can be accessed through various GENESIS32 client applications, such as **GraphWorX™32**. From the client side, you can easily connect to OPC tags coming from the remote OPC server on the host (server) node. To do this, you can set up a process point in GraphWorX32, for example:

1. Open GraphWorX32 by from the Windows **Start** menu by selecting **Programs > ICONICS GENESIS-32 > GraphWorX32 > GraphWorX32**.
2. In GraphWorX32, insert a process point into the display by selecting **Intrinsics > Process Point** from the **Dynamics** menu, as shown in the figure below. Click anywhere inside the display.



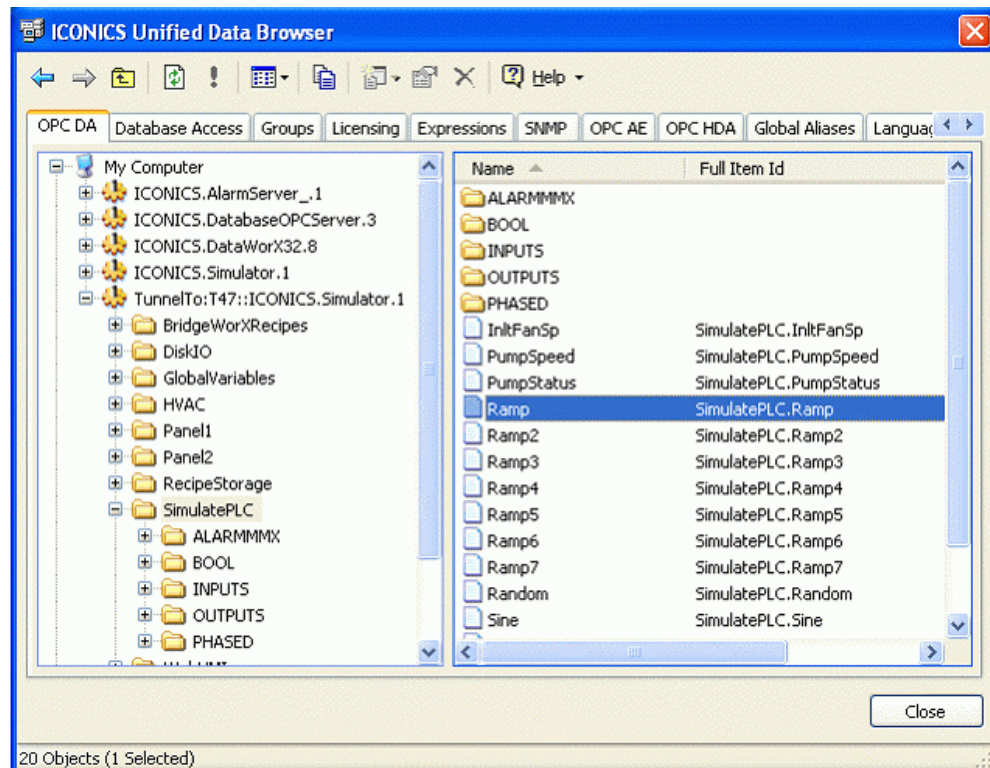
Adding a Process Point in GraphWorX32

3. This opens the **Property Inspector** for the new process point, as shown in the figure below. Click the **Data Tags** button.



Property Inspector for Process Point

4. In the **Unified Data Browser**, shown in the figure below, go to the **My Computer** tree control in the **OPC DA** tab and browse to the remote OPC server tunnel connection (for example, **TunnelTo:T47::ICONICS.Simulator.1**). From the client's perspective, the remote OPC server behaves like a local OPC server, and you can browse for data tags on the remote server just as if the server were located on the local computer.
5. Click **OK** two times to complete the process point configuration.



Selecting OPC Tags From the Tunnel Connection in the Unified Data Browser

Introduction to DataWorX32 Professional

Offered in the **DataWorX32 Professional Edition** is the new DataWorX32 full-featured redundancy, which provides high availability similar to features found in large Distributed Control Systems. DataWorX32 Redundancy is the only product to support the three most important OPC standards, increasing the reliability and availability of OPC data by allowing multiple OPC servers to be configured into redundant pairs. These redundant OPC server pairs seamlessly appear as a single OPC server to any OPC client application. This feature can be added to an existing OPC server/client application, without any reconfiguration of those applications, keeping your processes going without any down time. **DataWorX32 Professional Edition** is installed as part of the full GENESIS32 Automation Suite installation.

DataWorX32 Professional supports redundancy for the following:

- OPC Data Access (DA)
- OPC Alarm and Events (AE): AlarmWorX32 AE Server
- OPC Historical Data Access (HDA): TrendWorX32 HDA Server
- AlarmWorX32 Logger (with store-and-forward technology)
- TrendWorX32 Logger (with store-and-forward technology)

Configuration of DataWorX32 Professional involves the following GENESIS32 applications:

- **DataWorX32 Configurator:** Launches GenBroker node pairs configuration.
- **GenBroker Configurator:** Provides a user interface that enables you to establish a list of node pairs to be used for server redundancy.
- **Unified Data Browser:** Displays the status of redundancy tag connections to subscribed OPC DA, AE and HDA servers. The **Redundancy** tab of the Unified Data Browser automatically generates the proper syntax for redundancy tags.
- **MonitorWorX:** Client-side viewer displays runtime status information about redundant node pairs, and allows you to designate which server is active or standby. Generally, the server on the primary node is the **active** server, while the secondary node runs the server in **standby** mode.

- **AlarmWorX32 Logger Configurator:** Enables you to specify a secondary (redundant) database for logging alarm data, and activate store and forward operation for persistent data caching.
- **TrendWorX32 Logger Configurator:** Enables you to specify a secondary (redundant) database for logging trend data, and activate store and forward operation for persistent data caching.

GENESIS32 already provides some levels of redundancy. DataWorX32 has the ability to switch between a primary PC, managing an OPC Data Access server, and a secondary PC with identical server configurations. Additionally, GenBroker can be configured to provide redundant ICONICS administrative servers such as Security and License servers.

With the new DataWorX32 Professional, redundancy is expanded to include seamless redundancy of OPC Data Access, OPC Alarm and Events, and OPC Historical Data Replay with no need for client-side reconfiguration. This new approach allows for automatic redundancy with no special client-side alias referencing required.

The existing GENESIS32 approach for maintaining redundant administrative servers for Licensing, Security, Global Aliasing, Language Aliasing and GenEvent essentially remains the same.

Redundant Node Pairs

The basic concept of redundancy is to have two identical configurations (e.g. alarm server configuration) running on two separate computers, or nodes. The **Primary** node is the computer that is normally running. If the Primary node breaks, becomes disconnected from the network, or loses power unexpectedly, the system automatically switches over to the **Secondary**, or backup, node. When the Primary node recovers, the system switches back over to the Primary node after the necessary synchronization has occurred.

The AlarmWorX32 AE Server and TrendWorX32 HDA Server provide redundancy when GenBroker communications is enabled and the node on which this server is located is included in the redundant node pair list in the GenBroker configuration (GBX file). An identical GBX configuration, centrally located if desired, is run on both the primary and secondary computers.

For DataWorX32 Professional, the **AlarmWorX32 AE Server** configuration should be identical on both the Primary and Secondary nodes. Ideally, a central configuration should be used, and subscription references should be to the Primary node. Likewise, the **TrendWorX32 HDA Server** configuration should be identical on both the Primary and Secondary nodes.

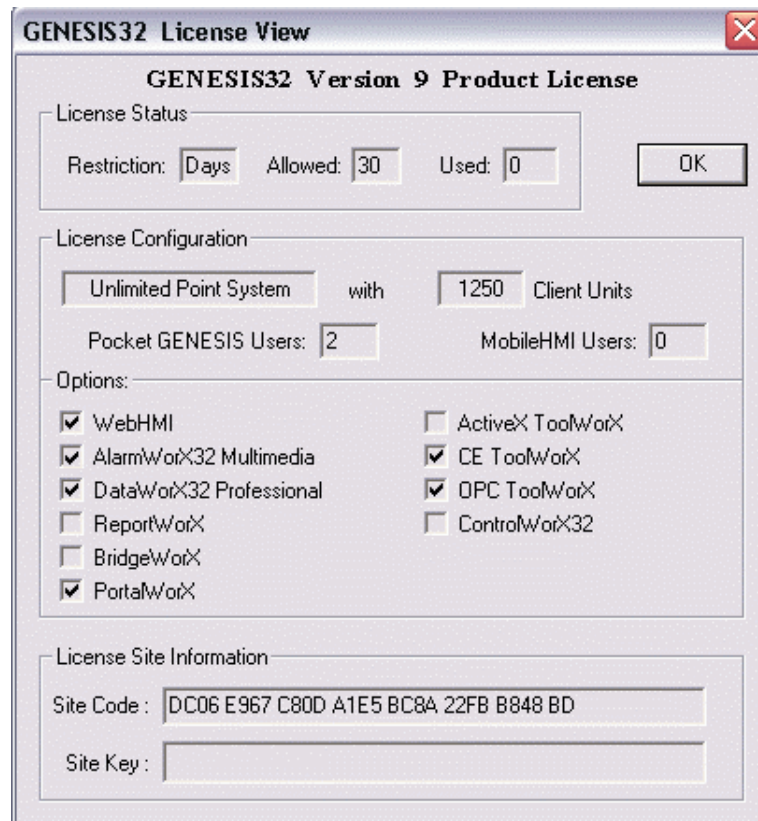
Active vs. Standby

Generally, the primary node hosts the **active server(s)**, while the secondary node hosts the **standby server(s)**. The ICONICS Alarm Server already has embedded redundancy, so it will automatically designate the active/standby status of the alarm servers with respect to all client nodes. For servers in which redundancy is not inherent (OPC DA servers), you can manually specify which one is active with respect to the local client (assuming that both nodes are currently online) using the **Redundancy** tab in MonitorWorX.

DataWorX32 Professional Licensing

All components needed for DataWorX32 Professional are installed by default with GENESIS32, but in order to use Professional you must have an authorized GENESIS32 software license with the DataWorX32 Professional option bit enabled in the License Utility, as shown in the figure below. DataWorX32 Professional does not work with a temporary license; you need a full GENESIS32 license to run DataWorX32 Professional.

Note: For information about authorizing a license, please see the GENESIS32 Software Licensing help documentation.



DataWorX32 Professional Option Bit Enabled in License Utility

Overview of DataWorX32 Professional Configuration

Configuration for DataWorX32 Professional involves the following components:

- **GenBroker Configurator:** Configuration should be identical on paired PCs. A central configuration is recommended. GenBroker should also be configured as an automatic service using MonitorWorX.
- **AlarmWorX32 AE Server:** Configuration in the AlarmWorX32 Server Configurator should be identical on both PCs. A central configuration is recommended. Alarm server subscription references should be to the Primary node. The AlarmWorX32 AE Server should also be configured as a manual service using MonitorWorX.

- **OPC Data Access Server(s):** When connecting to data points on an OPC DA server, OPC tag references should be to the Primary node.
- **TrendWorX32 HDA Replay Server:** The TrendWorX32 HDA Replay Server uses the active TrendWorX32 Logger configuration. Configuration in the TrendWorX32 Logger Configurator should be identical on both PCs. A central configuration is recommended. The TrendWorX32 HDA Server should also be configured as a manual service using MonitorWorX.
- **AlarmWorX32 Logger:** In the AlarmWorX32 Logger Configurator, you may specify a secondary (redundant) database for logging alarm data if the database is not centralized and is not already made redundant via clusters or other means. Store-and-forward technology enables seamless data caching and retrieval. The AlarmWorX32 Logger should also be configured as an automatic service using MonitorWorX, because the AlarmWorX32 Logger is not automatically launched by OPC clients.
- **TrendWorX32 Logger:** In the TrendWorX32 Logger Configurator, you may specify a secondary (redundant) database for logging trend data if the database is not centralized and is not already made redundant via clusters or other means.. Store-and-forward technology enables seamless data caching and retrieval. The TrendWorX32 Logger should also be configured as an automatic service using MonitorWorX, because the TrendWorX32 Logger is not automatically launched by OPC clients.

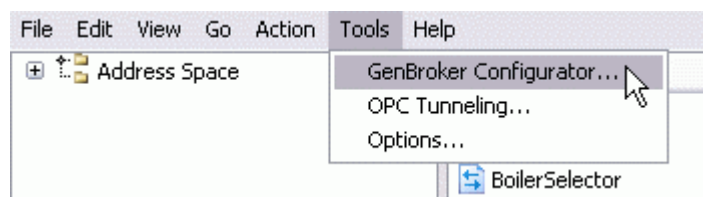
GenBroker Configuration

The GenBroker Configurator provides a user interface that enables you to establish a list of **Node Pairs** to be used for AlarmWorX32 Server OPC Data Access server, and TrendWorX32 HDA Server redundancy. For complete information about GenBroker, please refer to the GenBroker Configurator Help documentation.

Starting the GenBroker Configurator

The **GenBroker Configurator** application enables you to set up TCP/IP and SOAP/XML communication channels for different nodes on your LAN, WAN, or over the Internet. To launch the GenBroker Configurator, select **GenBroker Configurator** from the **Tools** menu in the DataWorX32 Configurator, as shown in the figure below.

Note: DataWorX32 Professional has been tested only with TCP/IP communications. It has not been tested with DCOM or SOAP/XML communications.

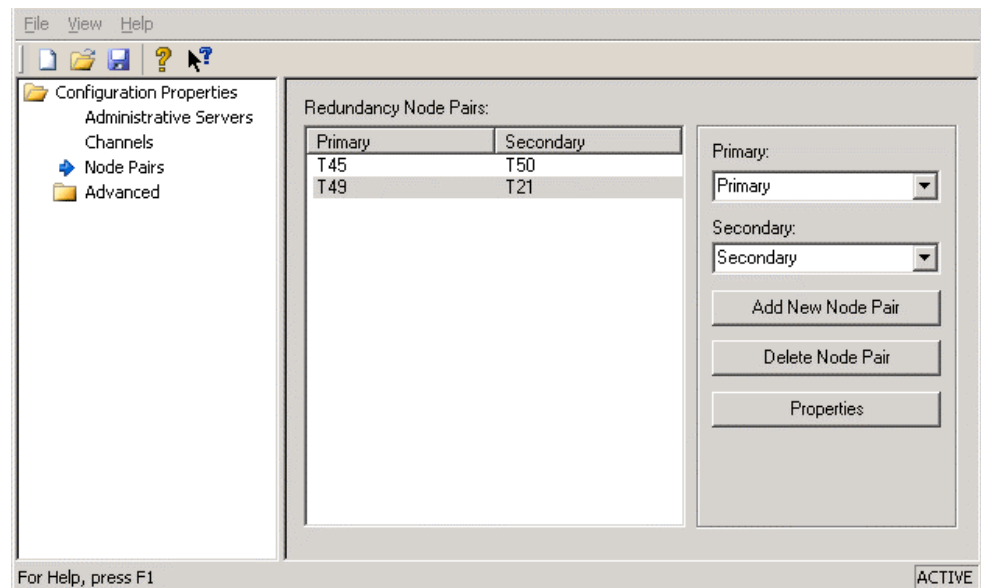


Launching the GenBroker Configurator from DataWorX32

Node Pairs Configuration

The **Node Pairs** section of the GenBroker Configurator enables you to specify one or more pairs of Primary and Secondary (backup) nodes for AlarmWorX32 Server and OPC Data Access server redundancy.

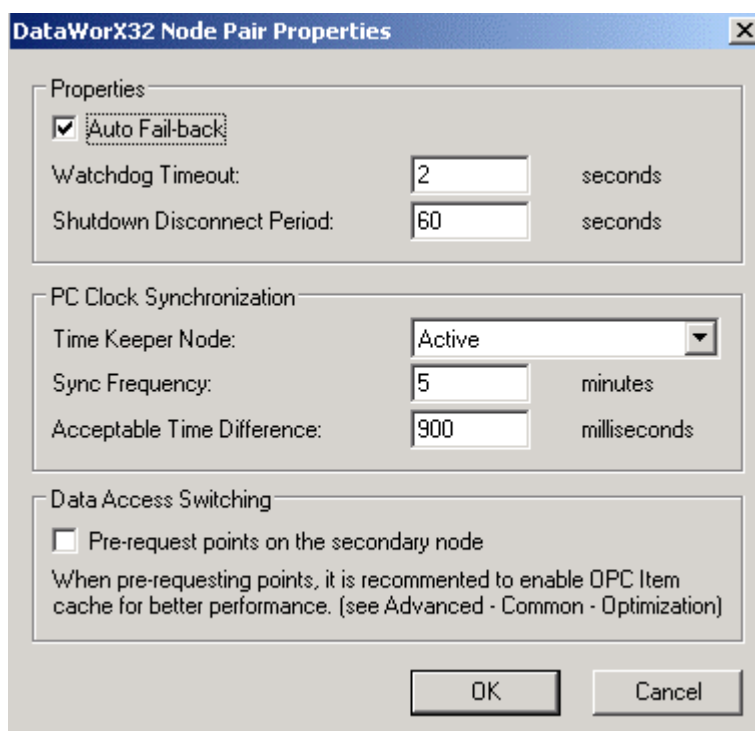
1. In the GenBroker Configurator, expand the **Configuration Properties** tree control and select **Node Pairs**, as shown in the figure below.
2. On the right-hand side there are two drop-down lists. Each list contains the names of nodes (computers) that are currently online. Under **Primary**, select the name of the computer that will serve as the Primary node. Under **Secondary**, select the name of the computer that will serve as the Secondary, or backup, node.
3. Click the **Add New Node Pair** button. This adds the two specified nodes to the **Redundancy Node Pairs** list, as shown in the figure below.



GenBroker Configurator: Node Pairs Configuration

Node Pair Properties

Clicking the **Properties** button opens the **DataWorX32 Node Pair Properties** dialog box, shown in the figure below, which enables you to configure the following settings. When you are finished with your node pairs configuration, close out of the GenBroker Configurator and save your configuration (.gbx) file. Make sure you activate the configuration.



Node Pair Properties

Properties

- **Auto Fail-back:** When checked, the redundant server(s) will revert back to the Primary node after the Primary node is restarted.
- **Watchdog Timeout:** The period of time (in seconds) that an ICONICS redundant server (AlarmWorX32 Server, AlarmWorX32 Logger, TrendWorX32 Logger or TrendWorX32 HDA Server) will allow to transpire before assuming active mode if paired node appears offline.
- **Shutdown Disconnect Period:** The period of time (in seconds) that an ICONICS redundant server will allow to transpire before attempting to reconnect with the paired server. This allows the remote GenTray to shut down an ICONICS redundant server for this period of time without this node attempting to re-launch the server.

PC Clock Synchronization

- **Time Keeper Node:** Determines which node (e.g., Disabled, Primary, Secondary, Active, or Standby) will act as the time keeper for ICONICS redundant servers, such as the AlarmWorX32 Server.
Note: Recommended setting is **Primary** node.
- **Sync. Frequency:** Number of minutes the time check is performed.
Note: Manual time changes are immediately effective.
- **Acceptable Time Difference:** Allowable time difference (in milliseconds) between node pairs.

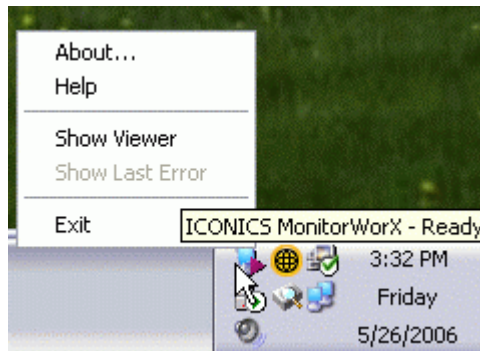
Data Access Switching

Prerequisite points on the secondary node: Allows points to be made ready for fast switchover to the **Secondary** node. This is especially important for AlarmWorX32 Server point requests on remote OPC DA server nodes. If enabled, GenBroker should be configured (on the OPC DA server node) to enable **Advanced > Common > Optimization > OPC Item Cache**. **Cache Size** should be set to a number larger than the number of points read from the DA server. If enabled, **Deactivate Cached Items** will disable the **OPC Group** items so that polling can be suspended. Best switchover performance occurs with **Deactivate Cached Items** disabled.

Running GenBroker As an Automatic Service

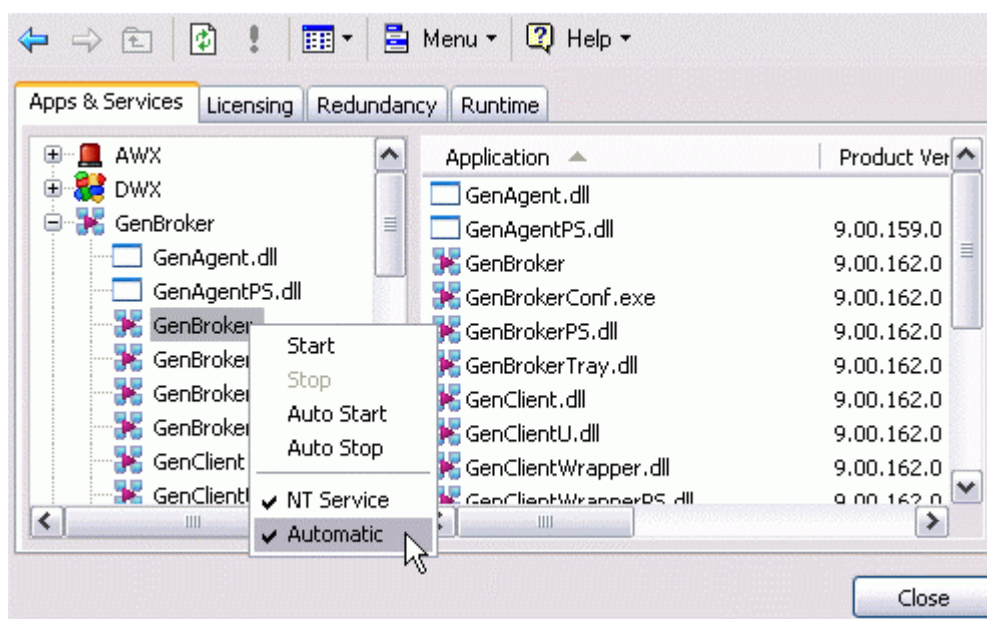
In order for clients to communicate with the GenBroker Server, you must activate GenBroker on the server node using MonitorWorX. For DataWorX32 Professional, it is recommended that GenBroker be configured as an automatic service using MonitorWorX.

1. Launch MonitorWorX from the Windows **Start** menu by selecting **Programs > ICONICS GENESIS32 > MonitorWorX**.
2. To display the MonitorWorX user interface, right-click on the MonitorWorX icon in the Windows taskbar and select **Show Viewer** from the pop-up menu, as shown in the figure below.



Displaying the MonitorWorX Viewer

3. Go to the **Apps & Services** tab, as shown in the figure below. In the **GenBroker** tree control, right-click on **GenBroker** and select **Start** from the pop-up menu. This activates the GenBroker Server. Then select **GenBroker > NT Service**, and then select **GenBroker > Automatic**. Both **NT Service** and **Automatic** should be checked, as shown in the figure below.



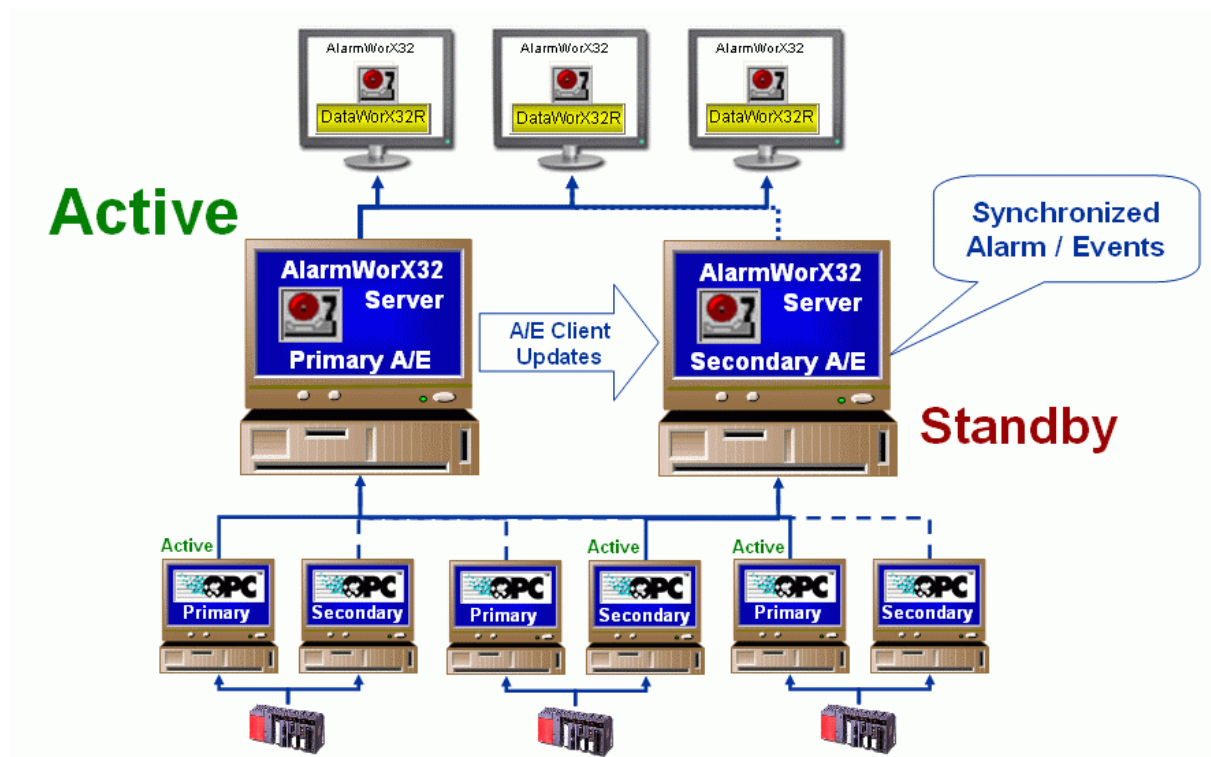
Running GenBroker As an Automatic Service

Note: If the GenBroker configuration file resides on a centralized file server, the GenBroker service should be configured to log on as a user with the proper user account to be able to read the GBX file. It is recommended to log on with a user account that has administrative privileges. The log on account is configured within the Service Control Manager (accessed via the Windows Control Panel). Select GenBroker and right-click to access the GenBroker service properties. Then use the **Log On** tab to assign a user account.

AlarmWorX32 Server Configuration

The **AlarmWorX32 Server Configurator** is used to create and activate alarm configurations. For DataWorX32 Professional, the active AlarmWorX32 Server configuration database should be identical on both the Primary and Secondary nodes. Ideally, a central configuration should be used, and subscription references should be to the Primary node.

Note: For more information about configuring the Alarm Server, please refer to the AlarmWorX32 Server Configurator Help documentation.

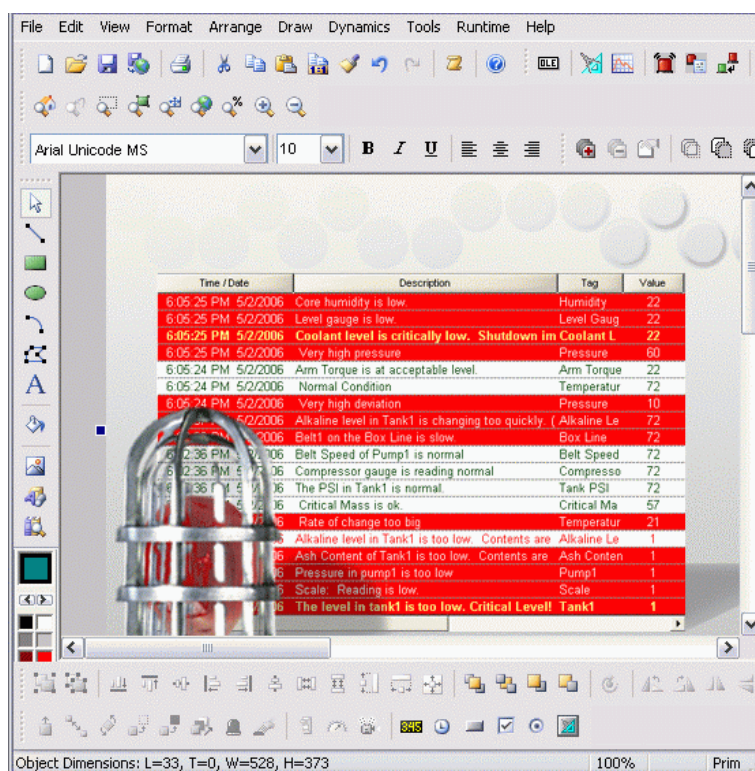


AlarmWorX32 AE Server Redundancy

Connecting to the Alarm Server From the Client Side

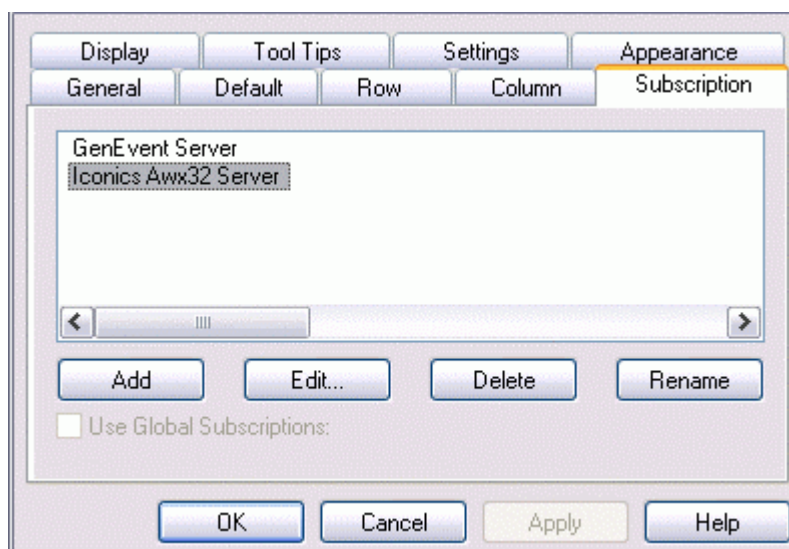
Once you have specified your redundant node pair in the GenBroker Configurator and you have set up your identical active alarm server configurations on both nodes, you can create an alarm subscription on the client side that connects to the alarm server on the Primary node. To do this, you need to use the AlarmWorX32 Viewer ActiveX:

1. Open GraphWorX32 by from the Windows **Start** menu by selecting **Programs > ICONICS GENESIS32 > GraphWorX32 > GraphWorX32**.
2. In GraphWorX32, insert an Alarm Viewer ActiveX into the display by clicking the **Alarm Viewer** button on the **ActiveX** toolbar, as shown in the figure below.



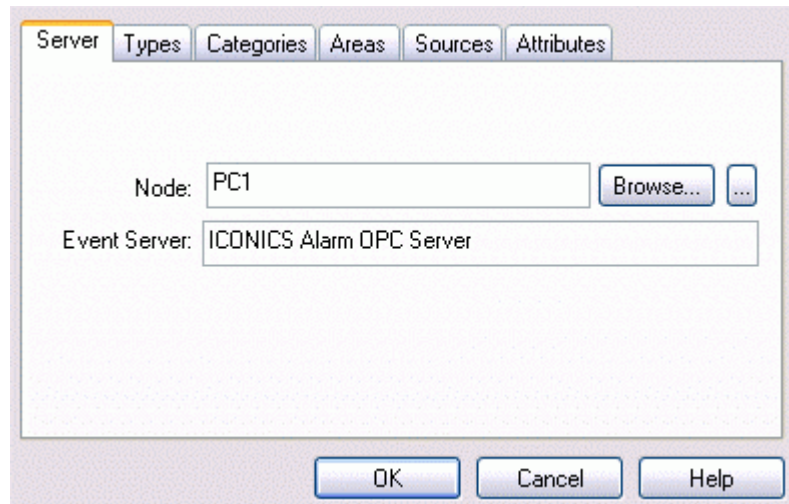
Alarm Viewer ActiveX Added to GraphWorX Display

- Double-click on the Alarm Viewer ActiveX to open the **Properties** dialog box. Go to the **Subscription** tab, select the **ICONICS AWX32 Server**, and click the **Edit** button, as shown in the figure below.



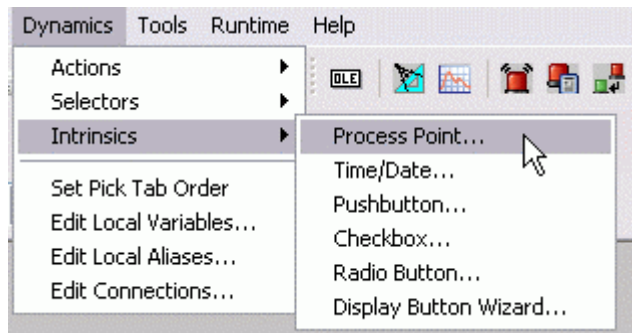
Alarm Subscription Properties

4. This opens the **Event Subscription** dialog box. On the **Server** tab, enter the name of the Primary node (e.g., PC1) in the **Node** field, as shown in the figure below.
5. Click **OK** two times to exit the Alarm Viewer properties.



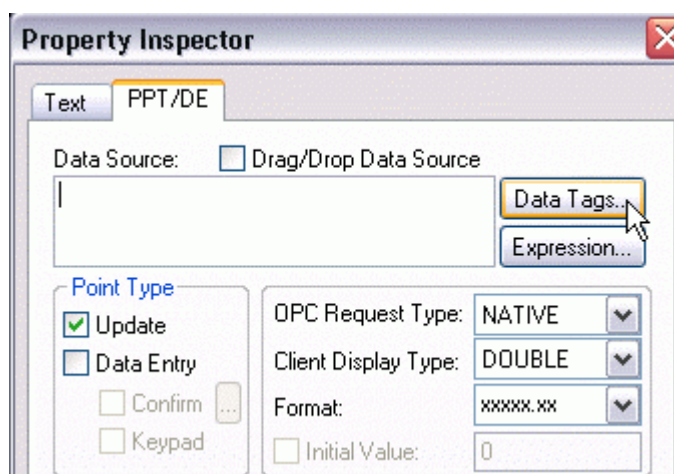
Alarm Subscription Properties: Server Properties

6. Now you are ready to set up your OPC Alarm and Event tags to the subscribed alarm server (i.e. the primary node). Create a process point in the GraphWorX32 display by selecting **Intrinsics > Process Point** from the **Dynamics** menu, as shown in the figure below.



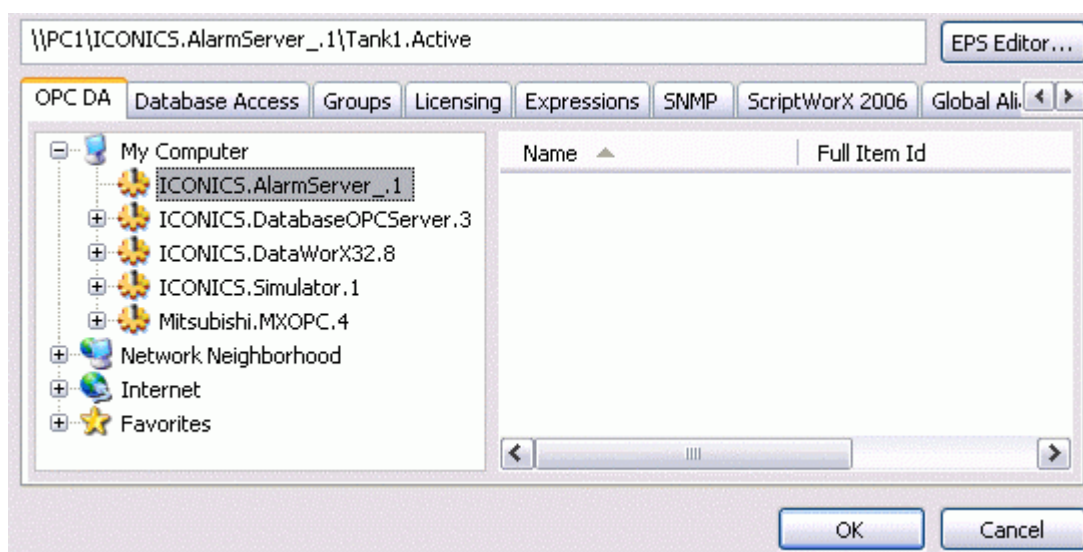
Adding a Process Point in GraphWorX32

7. Click on the display to open the **Property Inspector**, as shown in the figure below. Click the **Data Tags** button.



Property Inspector for Process Point

8. In the **Unified Data Browser**, shown in the figure below, go to the **OPC DA** tab and browse to **ICONICS.AlarmServer_.1** under **My Computer**. Select an alarm tag, and then add the name of the primary redundant node at the beginning of the tag name using the **\\NodeName** syntax. For example **\\PC1\ICONICS.AlarmServer_.1\Tank1.Active**, where the primary node name is PC1. This tells the client computer to get alarms from the subscribed alarm server on the primary node.
9. Click **OK** two times to complete the process point configuration.

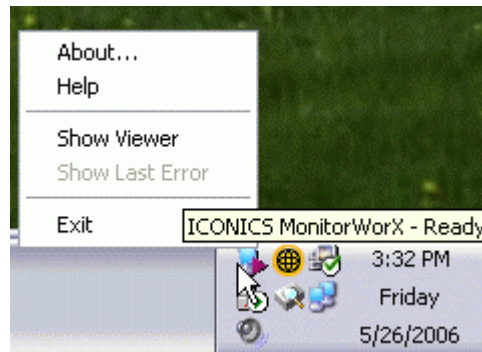


Selecting Alarm Tags in the Unified Data Browser

Running AlarmWorX32 Server As a Manual Service

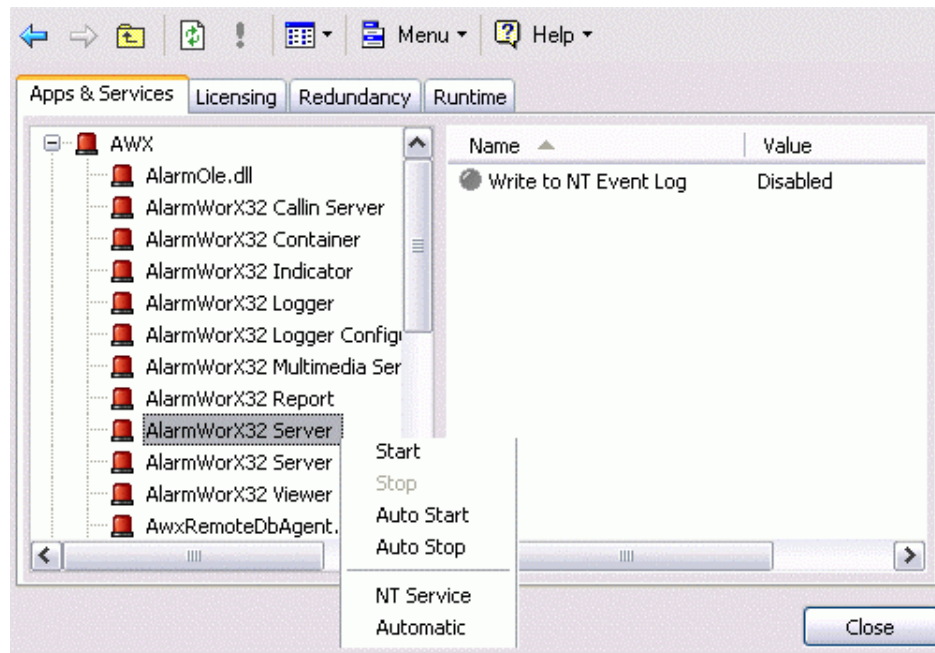
For DataWorX32 Professional, the AlarmWorX32 AE Server should be configured as a manual service using MonitorWorX.

1. Launch MonitorWorX from the Windows **Start** menu by selecting **Programs > ICONICS GENESIS32 > MonitorWorX**.
2. To display the MonitorWorX user interface, right-click on the MonitorWorX icon in the Windows taskbar and select **Show Viewer** from the pop-up menu, as shown in the figure below.



Displaying the MonitorWorX Viewer

3. Go to the **Apps & Services** tab, as shown in the figure below. In the tree control, select **AWX**, and then right-click on **AlarmWorX32 Server** and select **Start** from the pop-up menu. This activates the AlarmWorX32 Server. Then select **AlarmWorX32 Server > NT Service**.



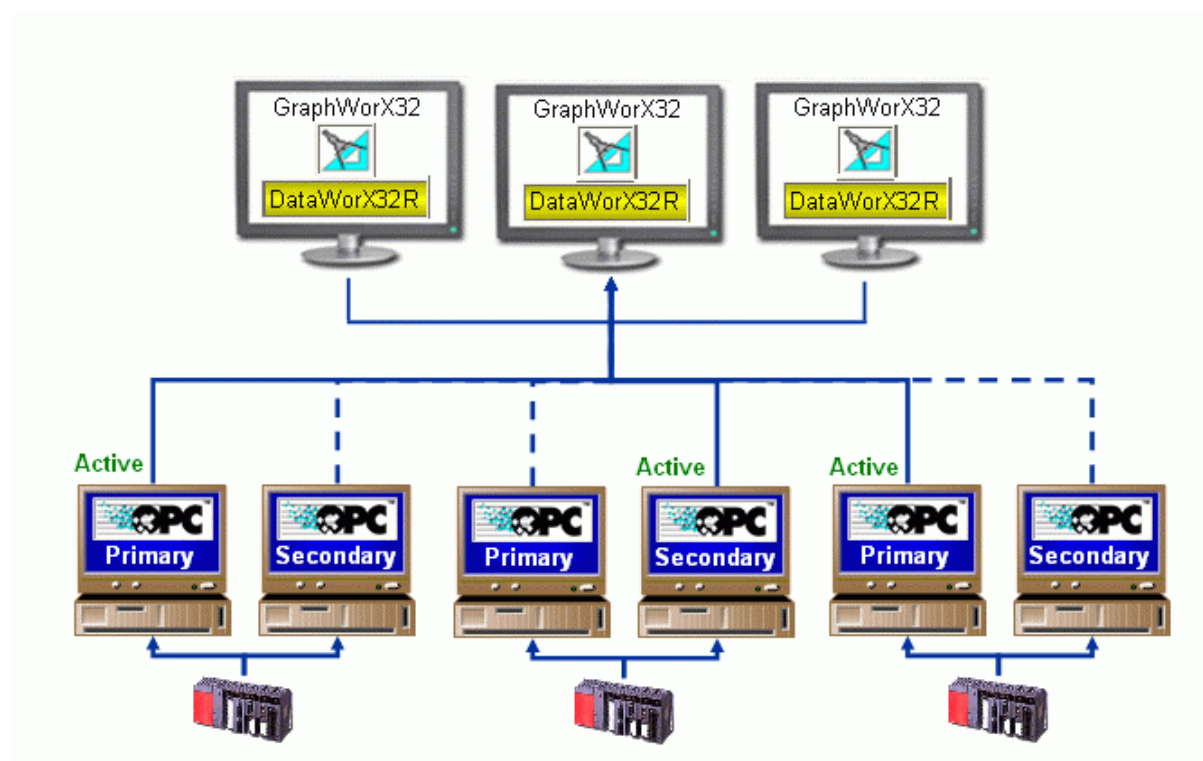
Running AlarmWorX32 Server As a Manual Service

Note: The AlarmWorX32 Server service should be configured to log on using an appropriate user account, as noted in the section above for GenBroker.

Connecting to OPC Data Access Servers

DataWorX32 Professional supports redundancy for OPC Data Access (DA). You can browse for ICONICS or third-party OPC DA servers and connect to OPC DA tags through GraphWorX32 using the Unified Data Browser.

Note: Redundancy is not inherent in OPC Data Access servers, however, so you can manually specify which one is active with respect to the local client (assuming that both nodes are currently online) using the **Redundancy** tab in MonitorWorX.

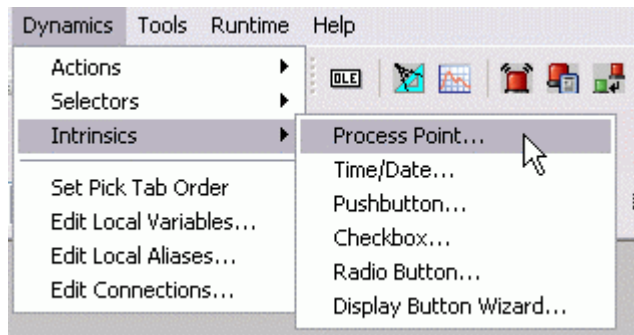


OPC DA Server Redundancy

Connecting to an OPC Data Access Server From the Client Side

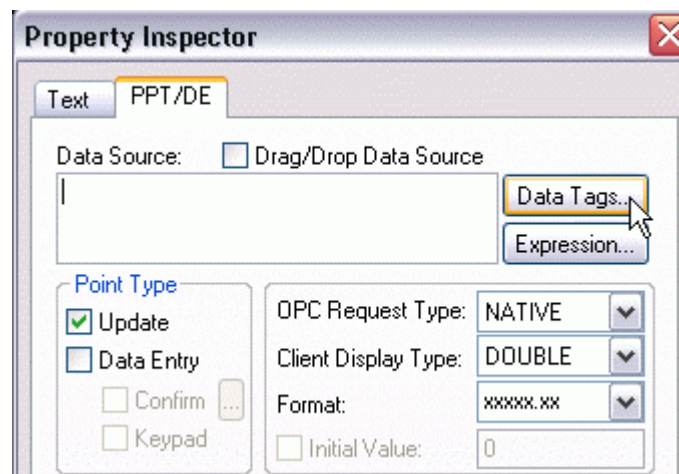
Once you have specified your redundant node pair in the GenBroker Configurator and you have set up your identical configurations on both nodes, you can connect to OPC tags from the client side that connect to the OPC DA server on the Primary node. To do this, you can set up a process point in GraphWorX32:

1. Open GraphWorX32 by from the Windows **Start** menu by selecting **Programs > ICONICS GENESIS32 > GraphWorX32 > GraphWorX32**.
2. In GraphWorX32, insert a process point into the display by selecting **Intrinsics > Process Point** from the **Dynamics** menu, as shown in the figure below. Click anywhere inside the display.



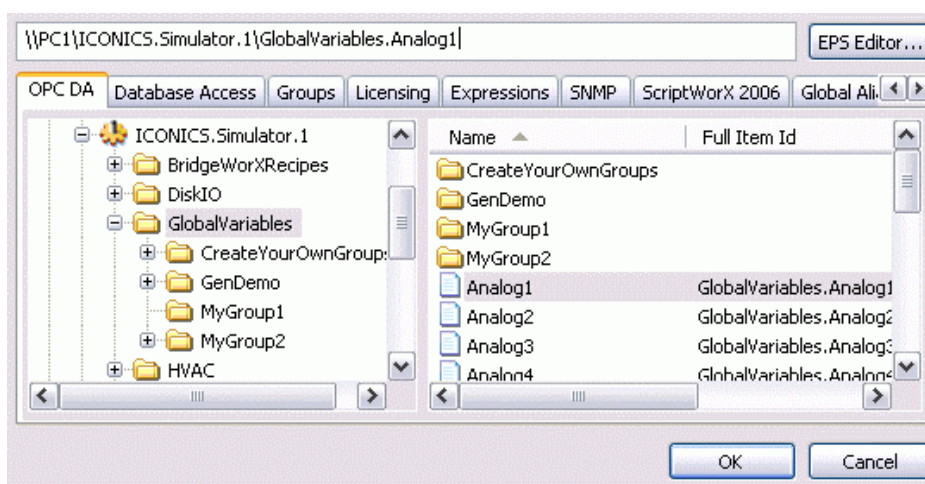
Adding a Process Point in GraphWorX32

3. This opens the **Property Inspector** for the new process point, as shown in the figure below. Click the **Data Tags** button.



Property Inspector for Process Point

4. In the **Unified Data Browser**, shown in the figure below, go to the **OPC DA** tab and browse to an OPC DA server (e.g., **ICONICS.Simulator.1**) under **My Computer**. Select a data access tag, and then add the name of the primary redundant node at the beginning of the tag name using the **\\NodeName** syntax. For example **\\PC1\ICONICS.Simulator.1\GlobalVariables.Analog1**, where the primary node name is PC1. This tells the client computer to get data from the OPC DA server on the primary node.
5. Click **OK** two times to complete the process point configuration.

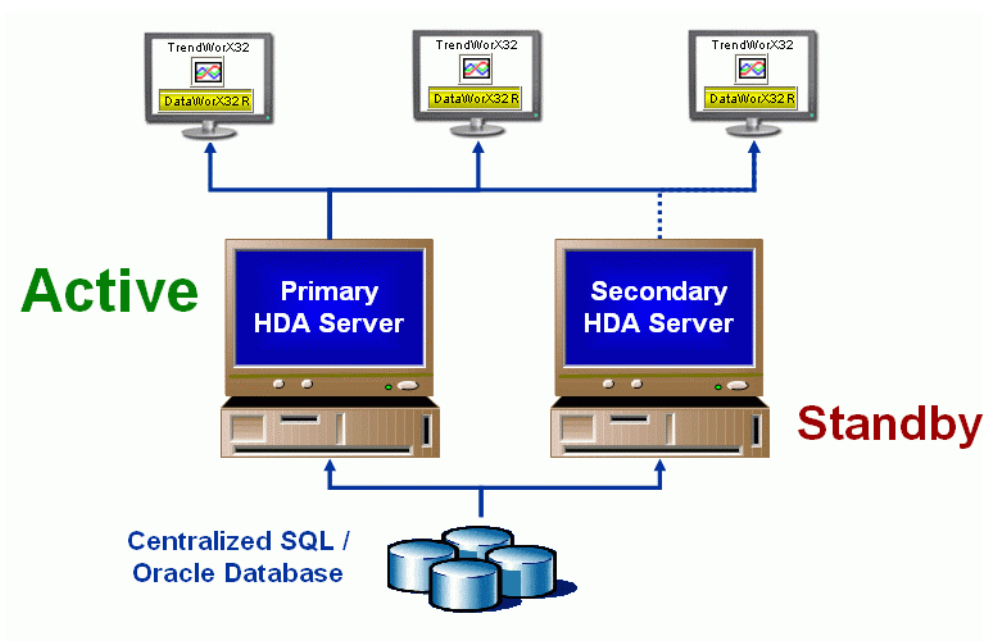


Selecting OPC DA Tags in Unified Data Browser

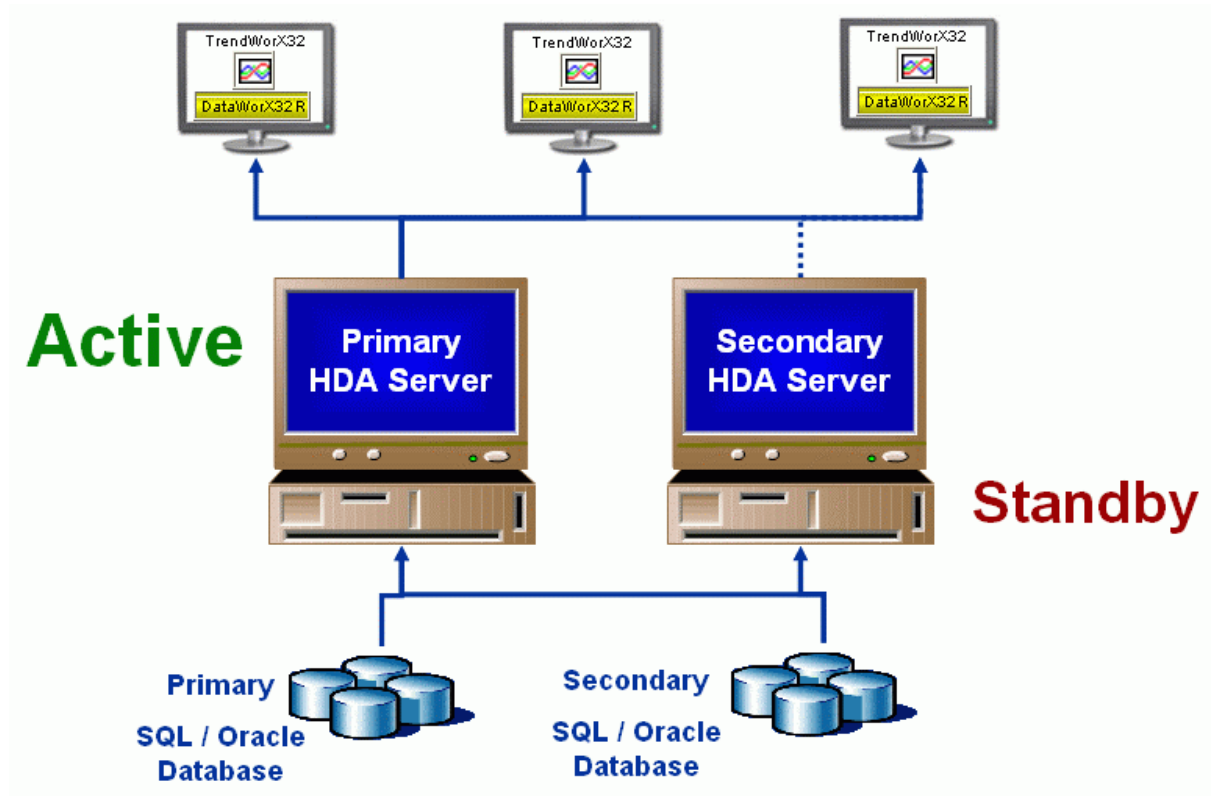
TrendWorX32 HDA Server Configuration

The TrendWorX32 HDA Replay Server uses the active TrendWorX32 Logger configuration. For DataWorX32 Professional, the active TrendWorX32 Server configuration database should be identical on both the Primary and Secondary nodes. A central configuration is recommended. The TrendWorX32 HDA Server should also be configured as a manual service using MonitorWorX. DataWorX32 Professional supports TrendWorX32 HDA Server redundancy for both centralized and duplicate database configurations, as shown in the figure below.

Note: For more information, please refer to the TrendWorX32 Logger Configurator Help documentation.



TrendWorX32 HDA Server Redundancy - Central Database

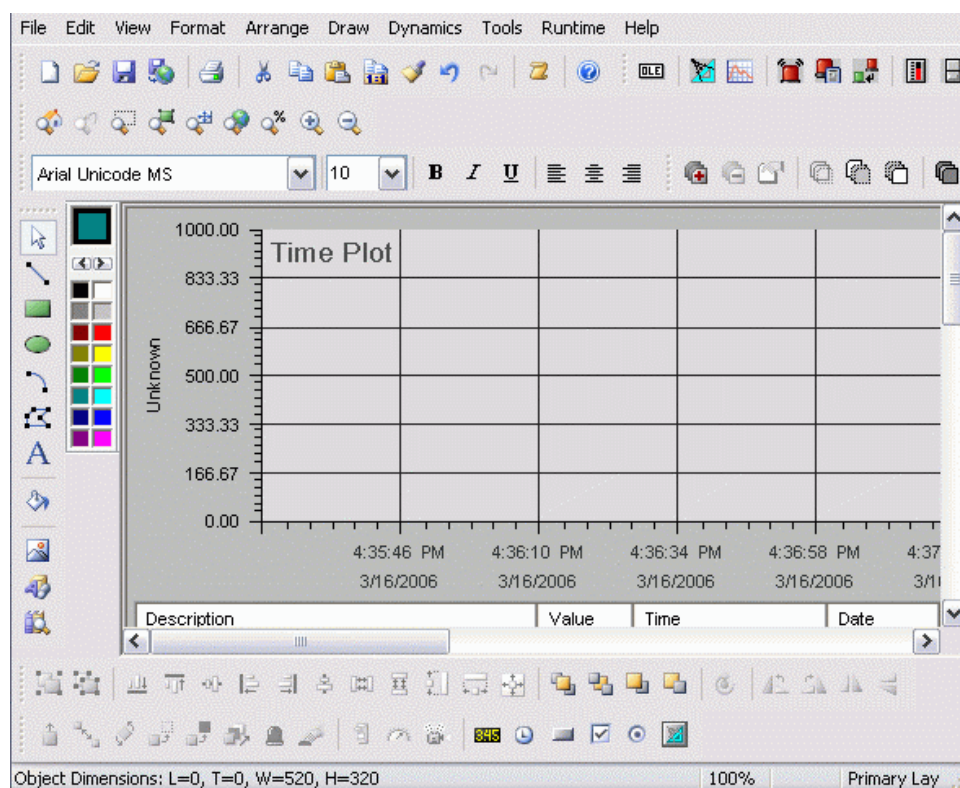


TrendWorX32 HDA Server Redundancy - Duplicate Databases

Connecting to the TrendWorX32 HDA Server From the Client Side

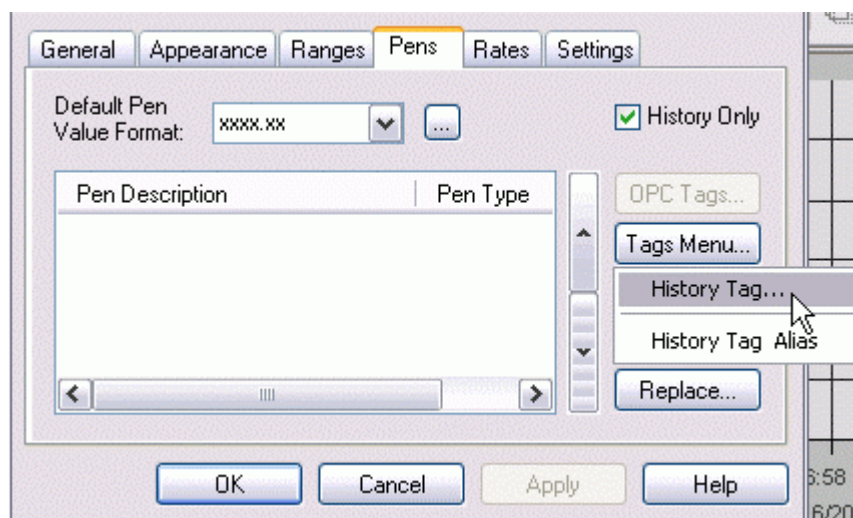
Once you have specified your redundant node pair in the GenBroker Configurator and you have set up your identical configurations on both nodes, you can connect to OPC tags from the client side that connect to the OPC HDA server on the Primary node. To do this, you can set up historical replay using the TrendWorX32 Viewer ActiveX:

1. Open GraphWorX32 by from the Windows **Start** menu by selecting **Programs > ICONICS GENESIS32 > GraphWorX32 > GraphWorX32**.
2. In GraphWorX32, insert a Trend Viewer ActiveX into the display by clicking the **Trend Viewer** button on the **ActiveX** toolbar, as shown in the figure below.



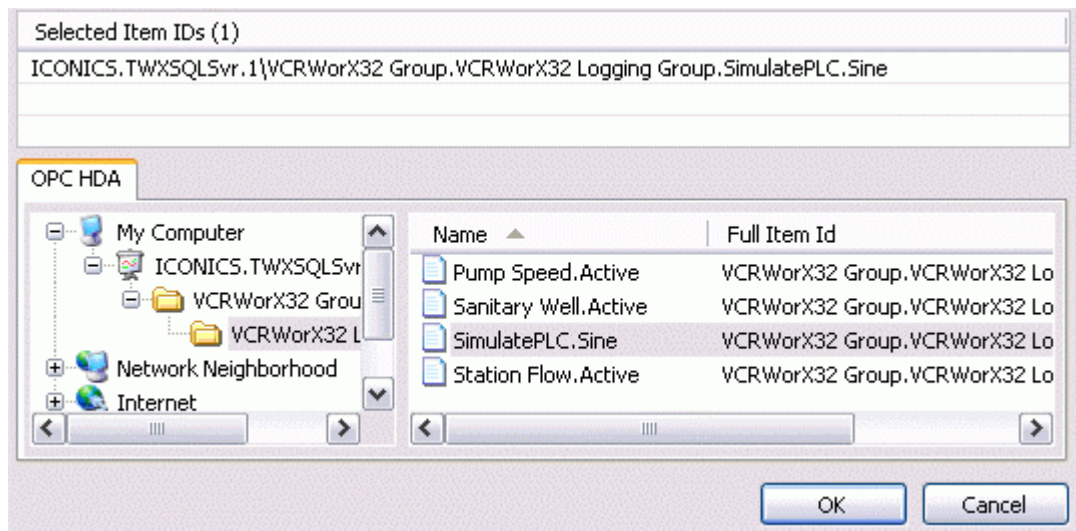
Trend Viewer ActiveX Added to GraphWorX Display

3. Double-click on the Trend Viewer ActiveX to open the **Properties** dialog box. Go to the **Pens** tab, as shown in the figure below. To configure the Viewer for **History Only Replay**, you must first check **History Only** in the **Pens** tab of the **Properties** dialog box, as shown in the figure below, prior to making any signal connections



Trend Viewer Pen Properties

4. In this mode, you can only connect to historical data. Once the Viewer is committed to animation (runtime) mode, it will populate the screen with the historical data and automatically set the Viewer into freeze mode. You can then retrieve historical data by scrolling back using the scroll bar. In addition, for finer control over historical replay, you can double-click the time window or click the **Set Right Time** button on the **Trend Viewer** toolbar to set the Viewer's right historical time of reference.
5. To add historical pens, click **Tags Menu** and select **History Tags** to open the **Unified Browser**, which allows you to browse all of the installed historical servers locally or remotely, as shown in the figure below.
6. In the **Unified Data Browser**, shown in the figure below, go to the **OPC HDA** tab and browse to an OPC HDA server (e.g., **ICONICS.TWXSQLSvr.1**) under **My Computer**. Select a historical data access tag, and then add the name of the primary redundant node at the beginning of the tag name using the **\\NodeName** syntax. For example **\\PC1\\ICONICS.TWXSQLSvr.1\\VCRWorX32 Group.VCRWorX32 Logging Group.SimulatePLC.Sine**, where the primary node name is PC1. This tells the client computer to get data from the OPC HDA server on the primary node.
7. Click **OK** two times to complete the trend configuration.

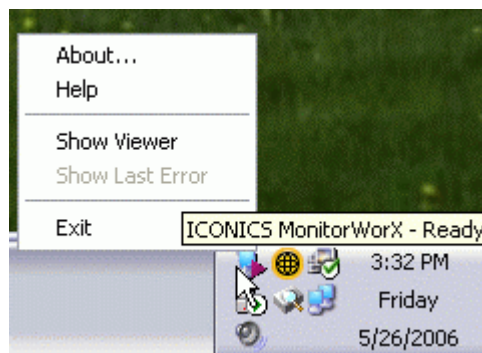


Selecting Trend Tags in the Unified Data Browser

Running TrendWorX32 HDA Server As a Manual Service

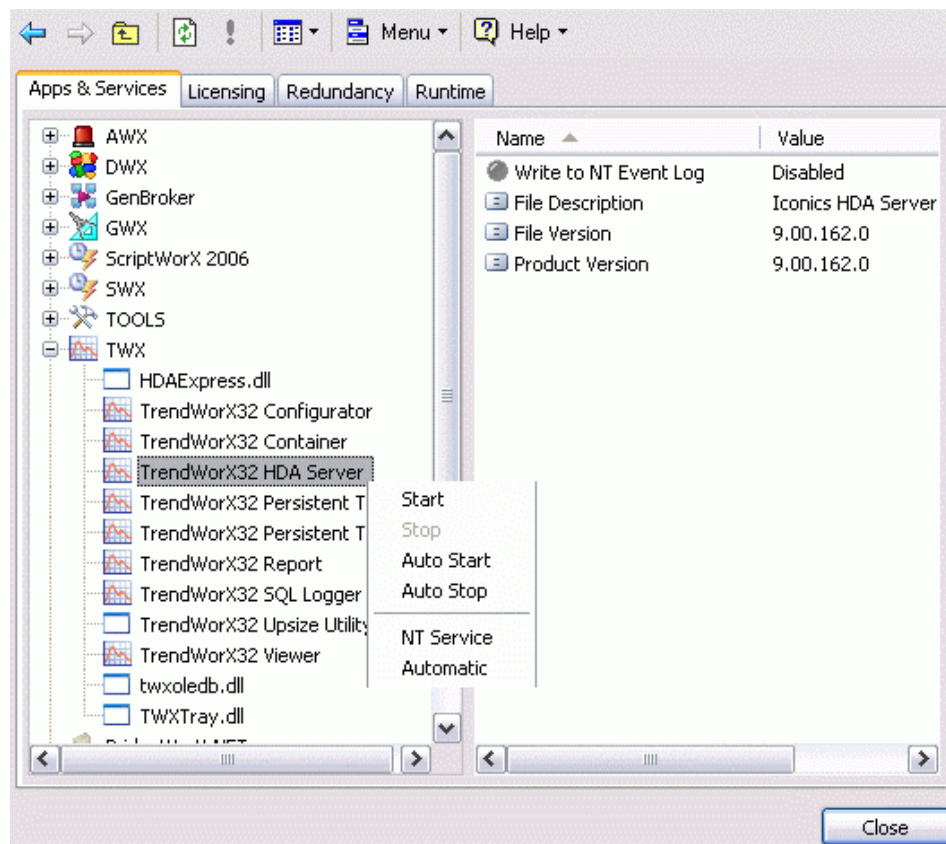
For DataWorX32 Professional, the TrendWorX32 HDA Server should be configured as a manual service using MonitorWorX.

1. Launch MonitorWorX from the Windows **Start** menu by selecting **Programs > ICONICS GENESIS32 > MonitorWorX**.
2. To display the MonitorWorX user interface, right-click on the MonitorWorX icon in the Windows taskbar and select **Show Viewer** from the pop-up menu, as shown in the figure below.



Displaying the MonitorWorX Viewer

- Go to the **Apps & Services** tab, as shown in the figure below. In the tree control, select **TWX**, and then right-click on **TrendWorX32 HDA Server** and select **Start** from the pop-up menu. This activates the TrendWorX32 Server. Then select **TrendWorX32 HDA Server > NT Service**, as shown in the figure below.



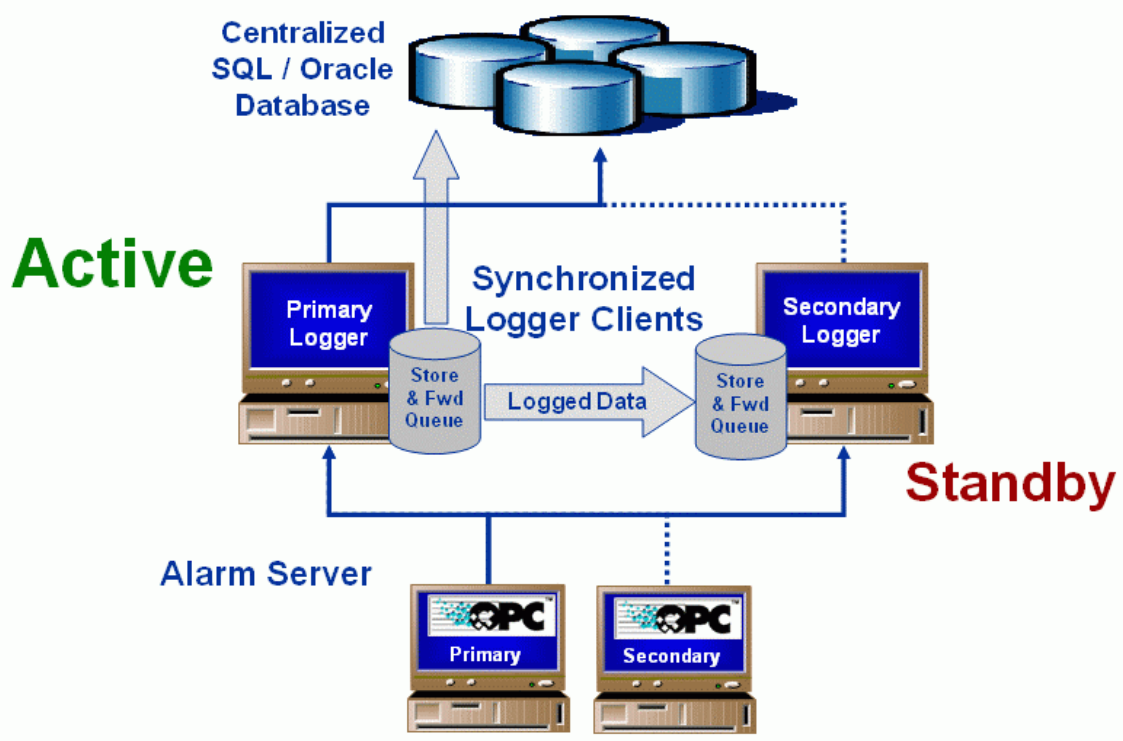
Running TrendWorX32 Server As a Manual Service

Note: The TrendWorX32 Server service should be configured to log on using an appropriate user account, as noted in the section above for GenBroker.

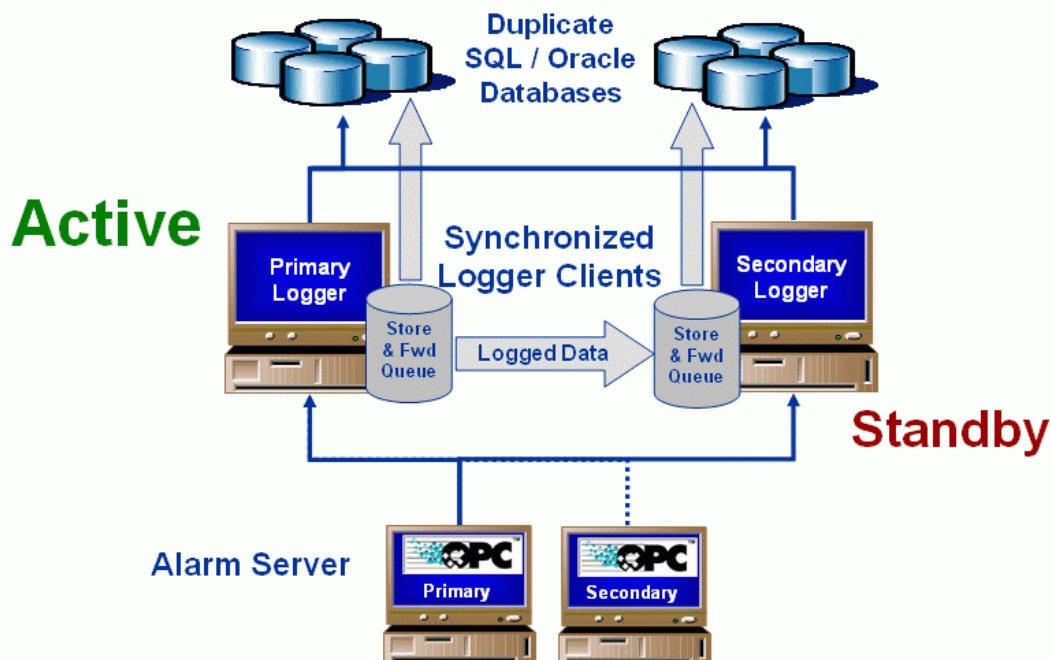
AlarmWorX32 Logger Configuration

In the **AlarmWorX32 Logger Configurator**, you may specify a secondary (redundant) database for logging alarm data if the database is not centralized and is not already made redundant via clusters or other means. Store-and-forward technology enables seamless data caching and retrieval for data synchronization. The AlarmWorX32 Logger should also be configured as an automatic service using MonitorWorX, because the AlarmWorX32 Logger is not automatically launched by OPC clients. DataWorX32 Professional supports AlarmWorX32 Logger redundancy for both centralized and duplicate database configurations, as shown in the figures below.

Note: For more information, please refer to the AlarmWorX32 Logger Configurator Help documentation.



AlarmWorX32 Logger Redundancy with Store-and-Forward Technology - Central Database



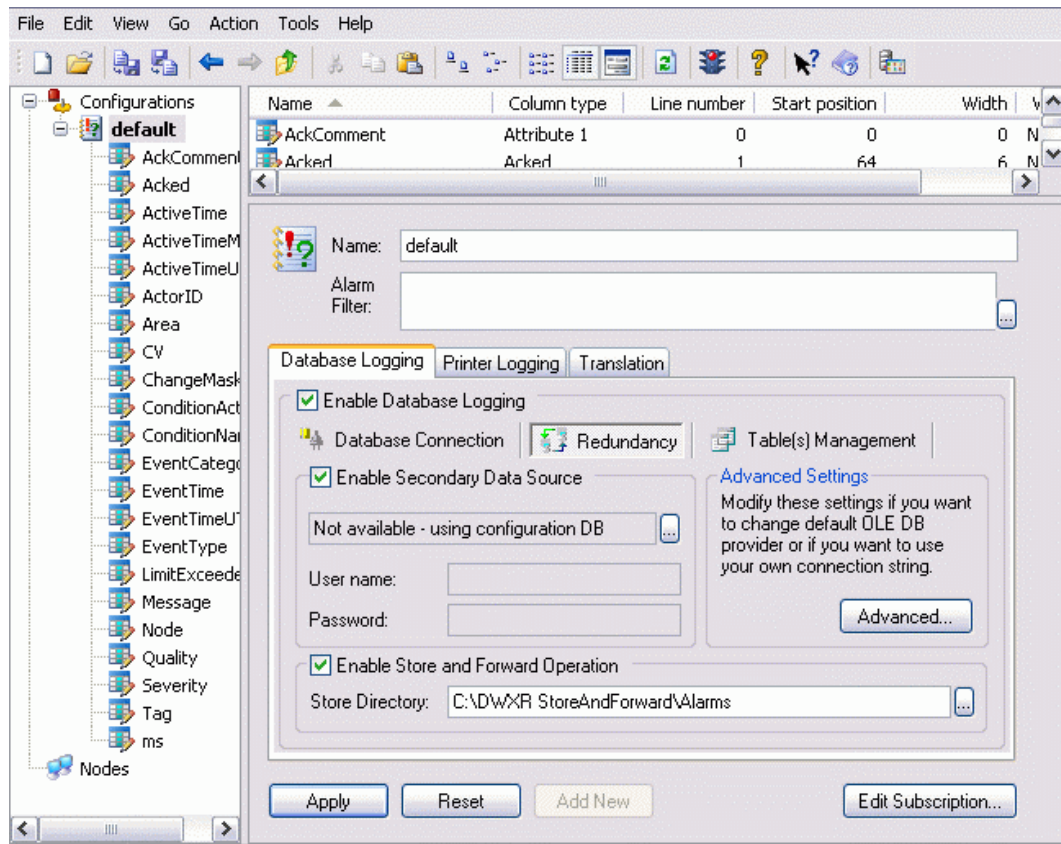
AlarmWorX32 Logger Redundancy with Store-and-Forward Technology - Duplicate Databases

To launch the AlarmWorX32 Logger Configurator from the Windows **Start** menu, select **Programs > ICONICS GENESIS32 > AlarmWorX32 > Alarm Logger Configurator**.

In the Alarm Logger configuration properties, go to the **Database Logging** tab and select **Redundancy**, as shown in the figure below. Check the **Enable Secondary Data Source** check box, and then click the ... button to establish a secondary connection to an ODBC data source (database), which will serve as the redundant alarm logger database. A duplicate set of data will be logged to the specified redundant alarm logger database.

Note: In order to log alarm data to a database, you must check the **Enable Database Logging** check box, as shown in the figure below. Before you can log alarm data to a secondary (redundant) alarm logger database, you must first establish a primary connection to an ODBC data source (database) in the **Database Connection** section of the **Database Logging** tab. Please refer to the AlarmWorX32 Logger Configurator help documentation for more information.

If you enable a secondary database for alarm logger redundancy, you should also check the **Enable Store and Forward Operation** check box, as shown in the figure below. Store-and-forward technology enables data caching if the connection to a logger database is lost. In the **Store Directory** field, click the ... button to browse for and select a directory for data caching. When the connection to the logger database is restored, the cache will be flushed, and the two (primary and secondary) databases will be automatically synchronized.

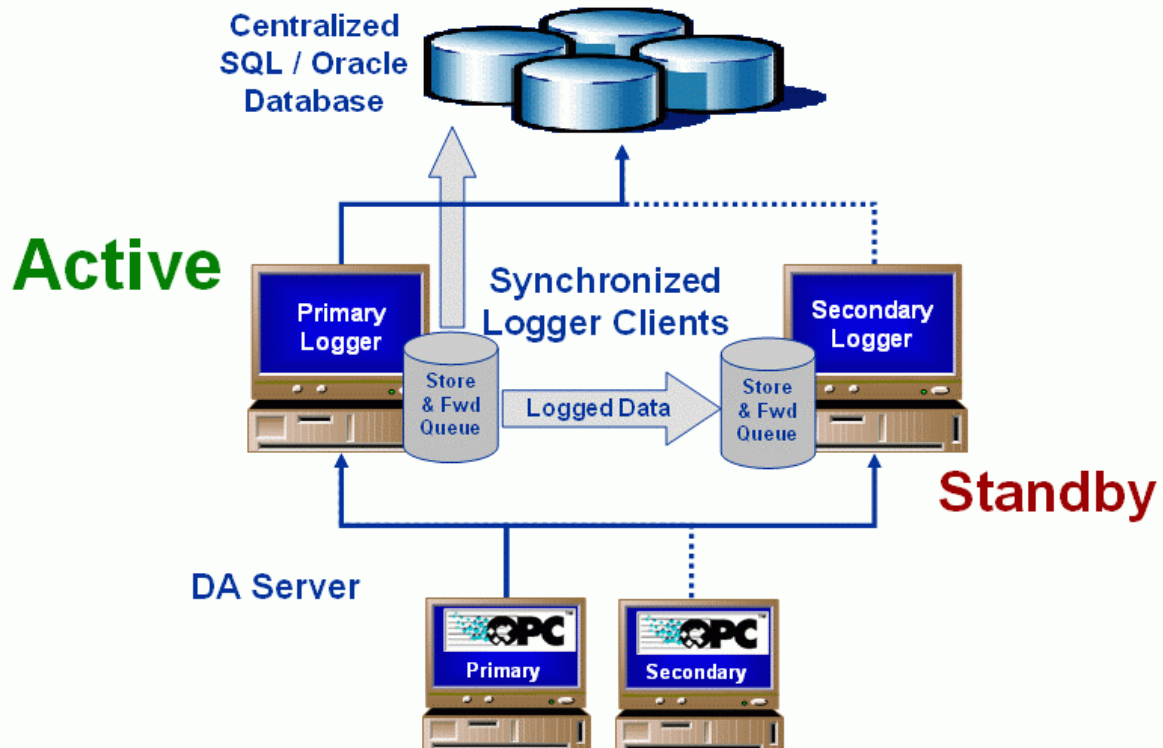


AlarmWorX32 Logger Configuration

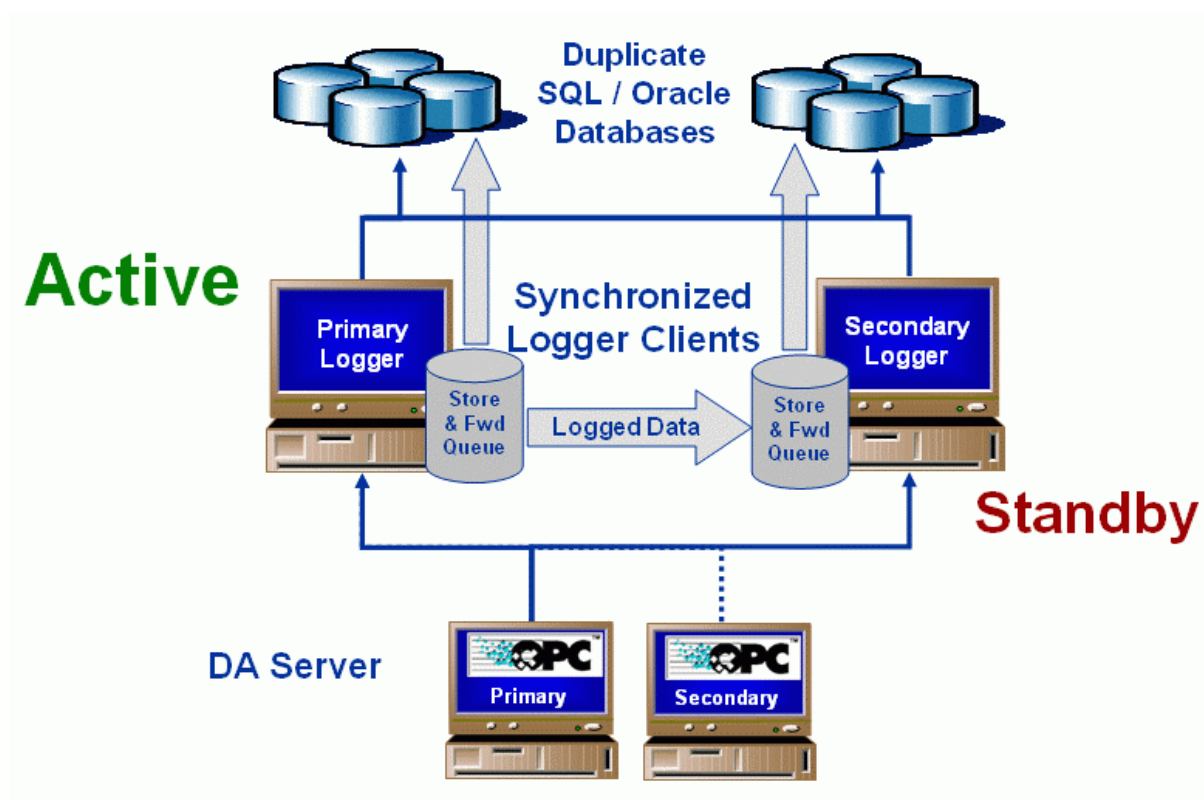
TrendWorX32 Logger Configuration

In the **TrendWorX32 Logger Configurator**, you may specify a secondary (redundant) database for logging trend data if the database is not centralized and is not already made redundant via clusters or other means. Store-and-forward technology enables seamless data caching and retrieval for data synchronization. The TrendWorX32 Logger should also be configured as an automatic service using MonitorWorX, because the TrendWorX32 Logger is not automatically launched by OPC clients. DataWorX32 Professional supports TrendWorX32 Logger redundancy for both centralized and duplicate database configurations, as shown in the figures below.

Note: For more information, please refer to the TrendWorX32 Logger Configurator Help documentation.



TrendWorX32 Logger Redundancy with Store-and-Forward Technology- Central Database



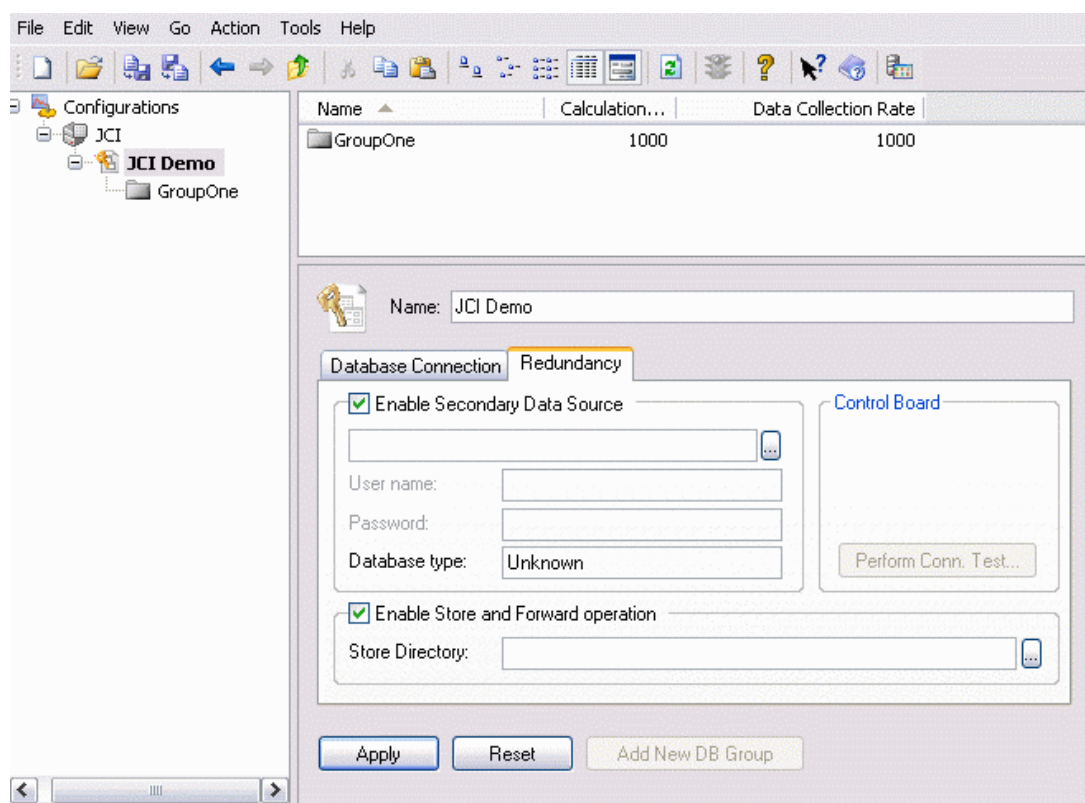
TrendWorX32 Logger Redundancy with Store-and-Forward Technology- Duplicate Databases

To launch the TrendWorX32 Logger Configurator from the Windows **Start** menu, select **Programs > ICONICS GENESIS32 > TrendWorX32 > TrendWorX32 Configurator**.

In the Trend Logger configuration properties, go to the **Redundancy** tab, as shown in the figure below. Check the **Enable Secondary Data Source** check box, and then click the ... button to establish a secondary connection to an ODBC data source (database), which will serve as the redundant trend logger database. A duplicate set of data will be logged to the specified redundant trend logger database.

Note: Before you can log trend data to a secondary (redundant) trend logger database, you must first establish a primary connection to an ODBC data source (database) in the **Database Connection** tab. Please refer to the TrendWorX32 Logger Configurator help documentation for more information.

If you enable a secondary database for trend logger redundancy, you should also check the **Enable Store and Forward Operation** check box, as shown in the figure below. Store-and-forward technology enables data caching if the connection to a logger database is lost. In the **Store Directory** field, click the ... button to browse for and select a directory for data caching. When the connection to the logger database is restored, the cache will be flushed, and the two (primary and secondary) databases will be automatically synchronized.



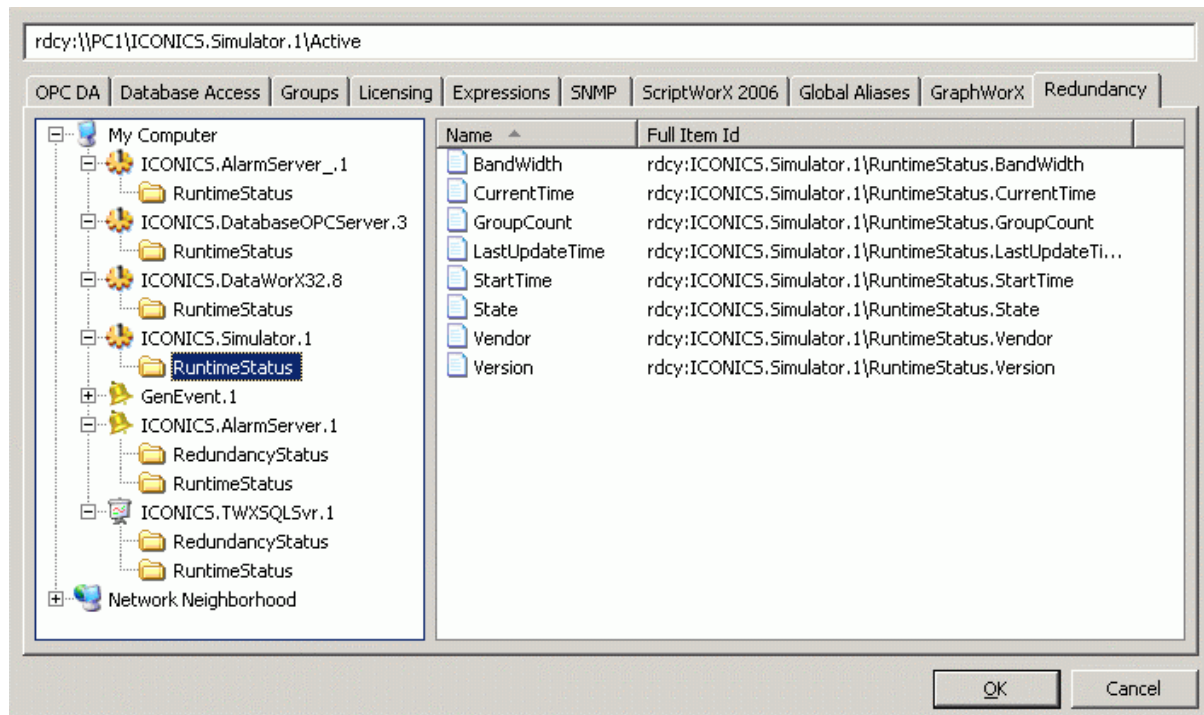
TrendWorX32 Logger Configuration

Redundancy Status Browsing

The **Redundancy Status Browser** supports selection of the tags that are intended to be used to get status information about redundant nodes and alarm, data, and trend servers. **Redundancy status tags** provide the basic status information about the nodes and servers (e.g. whether or not the node or server is online), and also detailed information about the redundant and OPC DA servers (e.g., redundant mode, server status, version, etc.). These status tags:

- Can be used in GraphWorX32 displays to indicate the server status.
- Can be used in scripts.
- Drive the redundancy switchover.

The Redundancy Status Browser is included into the default configuration of the **Unified Data Browser**. There is a special **Redundancy** tab that provides status tags for the redundant OPC DA and other application servers that have integrated DataWorX32 Redundancy support, as shown in the figure below.



Selecting Redundancy Status Tags in Unified Data Browser

Syntax of the Redundancy Status Tags

The general syntax of the redundancy status tags is as follows:

rdcy:[\\<node>\\][<server>\\]<item>[.<subitem>]

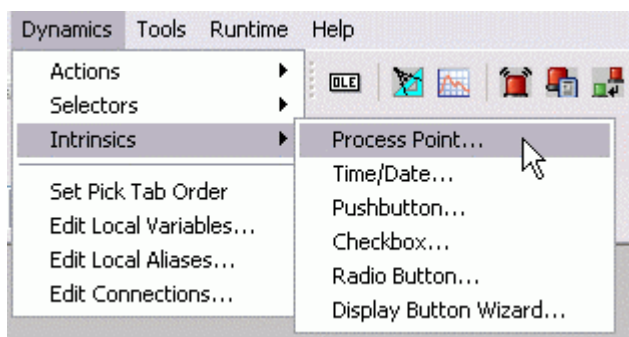
where:

- at least one of <node> or <server> must be specified.
- <item> may be "Online", "Active", "RuntimeStatus", or "RedundancyStatus".
- <subitem> depends on the <item>; should not be specified for "Online".

Connecting to Redundancy Status Tags From the Client Side

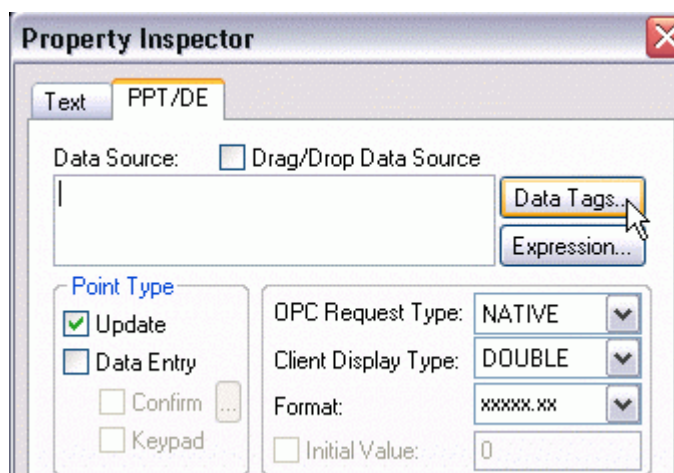
Once you have specified your redundant node pair in the GenBroker Configurator and you have set up your identical configurations on both nodes, you can create and connect to OPC tags from the client side that connect to redundancy status tags on the Primary node. To do this, you can set up a process point in GraphWorX32:

1. Open GraphWorX32 by from the Windows **Start** menu by selecting **Programs > ICONICS GENESIS32 > GraphWorX32 > GraphWorX32**.
2. In GraphWorX32, insert a process point into the display by selecting **Intrinsics > Process Point** from the **Dynamics** menu, as shown in the figure below. Click anywhere inside the display.



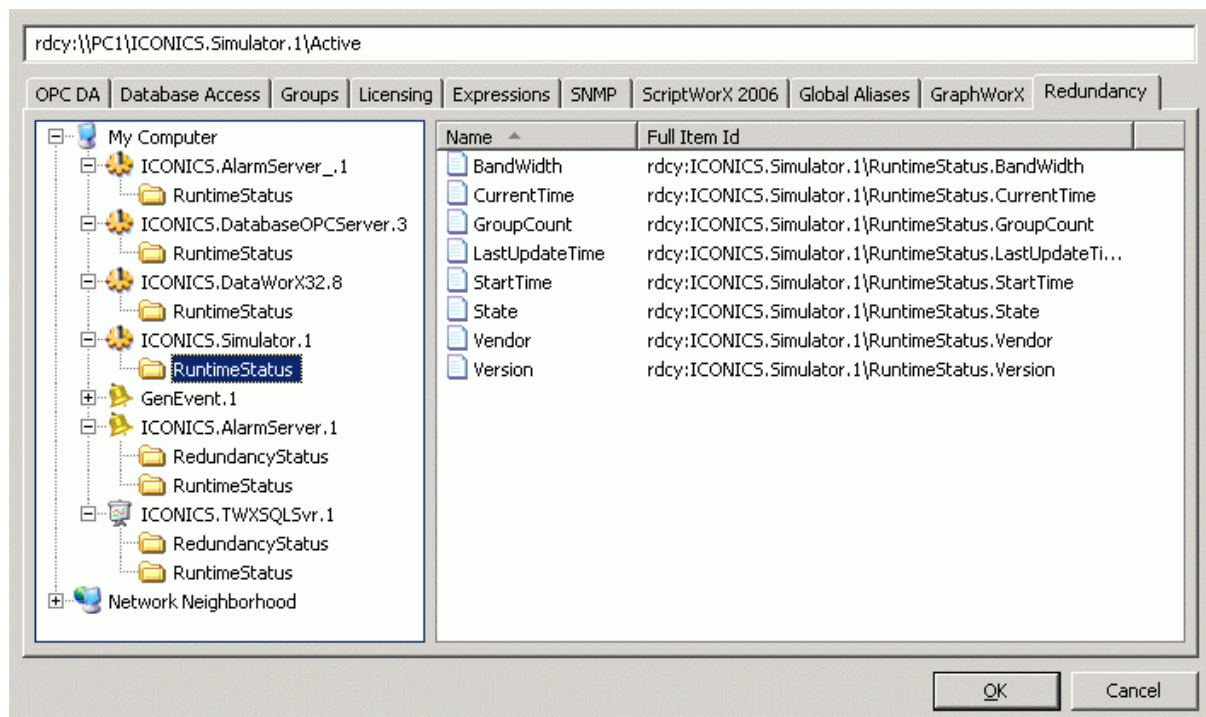
Adding a Process Point in GraphWorX32

3. This opens the **Property Inspector** for the new process point, as shown in the figure below. Click the **Data Tags** button.



Property Inspector for Process Point

4. In the **Unified Data Browser**, shown in the figure below, go to the **Redundancy** tab and browse to an OPC DA or AE server (e.g., **ICONICS.Simulator.1**) under **My Computer**. Select a redundancy status tag, and then append the name of the primary redundant node at the beginning of the tag name using the **\\NodeName** syntax. For example **rdcy:\\PC1\\ICONICS.Simulator.1\\Active**, where the primary node name is PC1. This tells the client computer to get status data from the specified server on the primary node.
5. Click **OK** two times to complete the process point configuration.



Selecting Redundancy Status Tags in Unified Data Browser

Node Status Tags

On each node only the "Online" tag is available. This status tag provides a Boolean value that indicates whether the given node is accessible or not.

The sample tag should look like the following one:

rdcy:\\PC1\\Online

Note: "Online" means that the client is connected to the given node. For example, if the client does not read any values from PC1, the tag "rdcy:\\PC1\\Online" will be false, even if it may be running and serving data to other nodes.

Server Status Tags

The OPC DA and OPC AE servers are listed under the nodes. The server always provides "Online" and "RuntimeStatus" tags.

If the server is on the node configured in the **Redundancy Node Pairs** configuration in the GenBroker Configurator, then the "Active" tag is also available.

If the server is of OPC AE type and is on the Redundancy Node Pair node, then the "RedundancyStatus" tag is available.

Each server (both DA and AE) always has an "Online" Boolean tag that indicates whether or not the given server is accessible. The sample tag should be as follows:

rdcy:\\PC1\\ICONICS.Simulator\\Online

The "RuntimeStatus" tag is also always present. This tag contains the SAFEARRAY of VARIANT values containing all members of the runtime status structure (e.g., state, version, start time, last update time, vendor, etc.).

If the given server is on the node that is configured in the Redundancy Node Pairs configuration in the GenBroker Configurator, then two more tags are listed. The "Active" tag is presented for any type of server, and the "RedundancyStatus" tag is available only for servers that have integrated DataWorX32 Professional support.

The "Active" status tag supports read and write operations and indicates whether the server is running as the active server in the node pair (i.e. the server is currently serving data). The active tag should look like:

```
rdcy:\\PC1\\ICONICS.AlarmServer.1\\Active
```

The Active tag is recommended for applications that require the use of a Boolean tag to monitor or manually switch the client between Primary and Secondary nodes.

The "RedundancyStatus" tag contains the SAFEARRAY of VARIANT values containing all members of the redundancy status structure (e.g. redundant mode, start time, elapse time, version, manufacturer, etc.).

Note: The "RuntimeStatus" and "RedundancyStatus" are also presented as folders listed on the server level.

Server Runtime Status Tags

The **Runtime Status** tags are available for any server. These tags are listed under the "RuntimeStatus" folder and provide the single members of the runtime status structure with status information provided by the given server. The structure of the runtime status depends on the server type, there are several structure versions for:

- OPC DA Server
- OPC AE Server
- OPC HDA Server

The sample runtime status tag syntax is as follows:

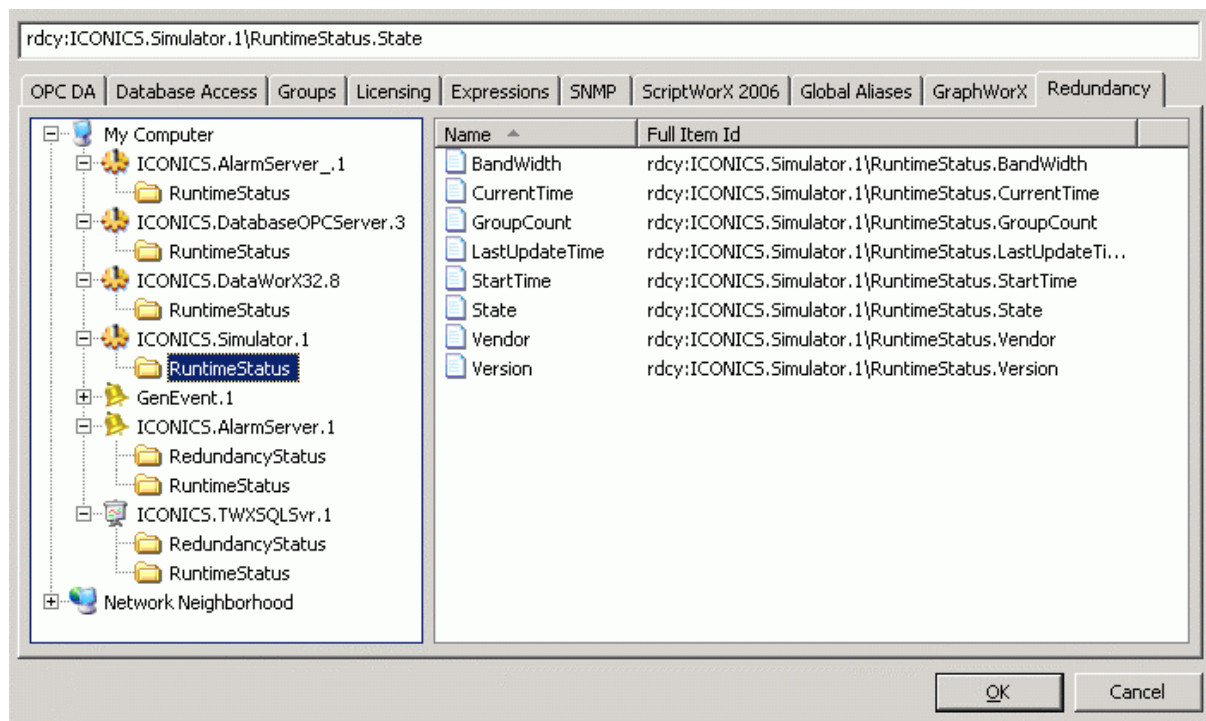
```
rdcy:ICONICS.Simulator\\RuntimeStatus.StartTime
```

The following sections describe the runtime status tags for the various server types. The set of tags is server-specific, so the following tables do not cover all the possibly available tags.

OPC DA Server Runtime Status Tags

Tag Name	Description
StartTime	Date/time value of the server start time
CurrentTime	Date/time value of the last update time
StartTime	Date/time value of the server start time
State	Integer value with the state of the server: State value Meaning 1 Running

	2	Failed
	3	No Config
	4	Suspended
	5	Test
GroupCount	Integer value with the DA server group count	
BandWidth	Integer value with the DA server bandwidth	
Vendor	String value with the vendor name	
Version	String value with the server version (e.g. 2.1.1).	
LastUpdateTime	Date/time value with the last update time	

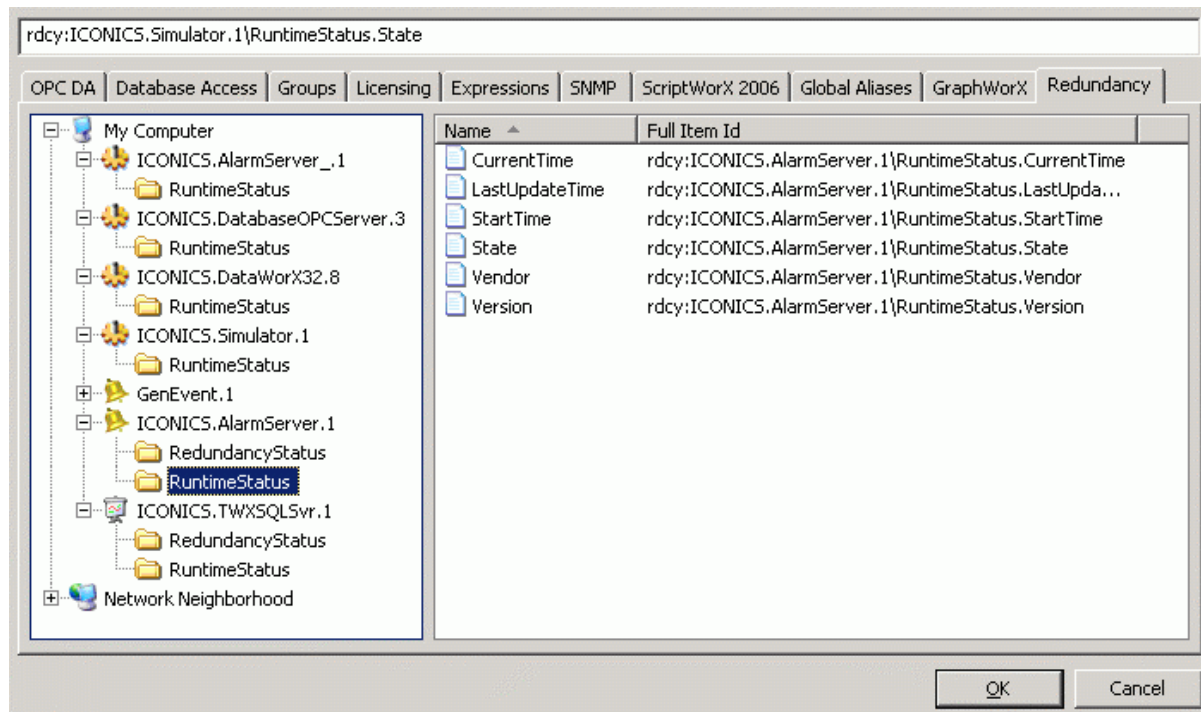


OPC DA Server Runtime Status Tags

OPC AE Server Runtime Status Tags

Tag Name	Description
StartTime	Date/time value of the server start time
CurrentTime	Date/time value of the current time
LastUpdateTime	Date/time value with the last update time

Tag Name	Description
State	Integer value with the state of the server: State value Meaning 1 Running 2 Failed 3 No Config 4 Suspended 5 Test
Vendor	String value with the vendor name
Version	String value with the server version (e.g. 8.2.139).

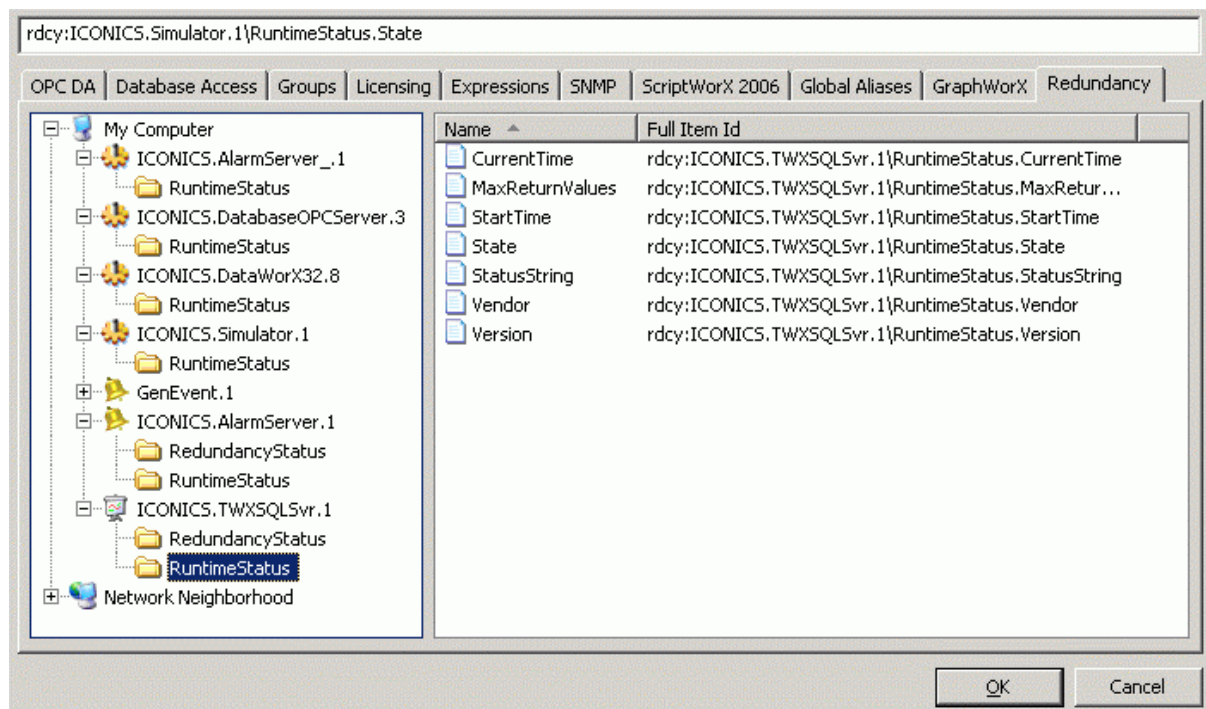


OPC AE Server Runtime Status Tags

OPC HDA Server Runtime Status Tags

Tag Name	Description
StartTime	Date/time value of the server start time
CurrentTime	Date/time value of the current time
State	Integer value with the state of the server:

	State value Meaning 1 Running 2 Failed 3 No Config 4 Suspended 5 Test
MaxReturnValues	Integer value with the HDA server max possible return values
StatusString	Status string of the HDA server
Vendor	String value with the vendor name
Version	String value with the server version (e.g. 2.1.1).



OPC HDA Server Runtime Status Tags

Server Redundancy Status Tags

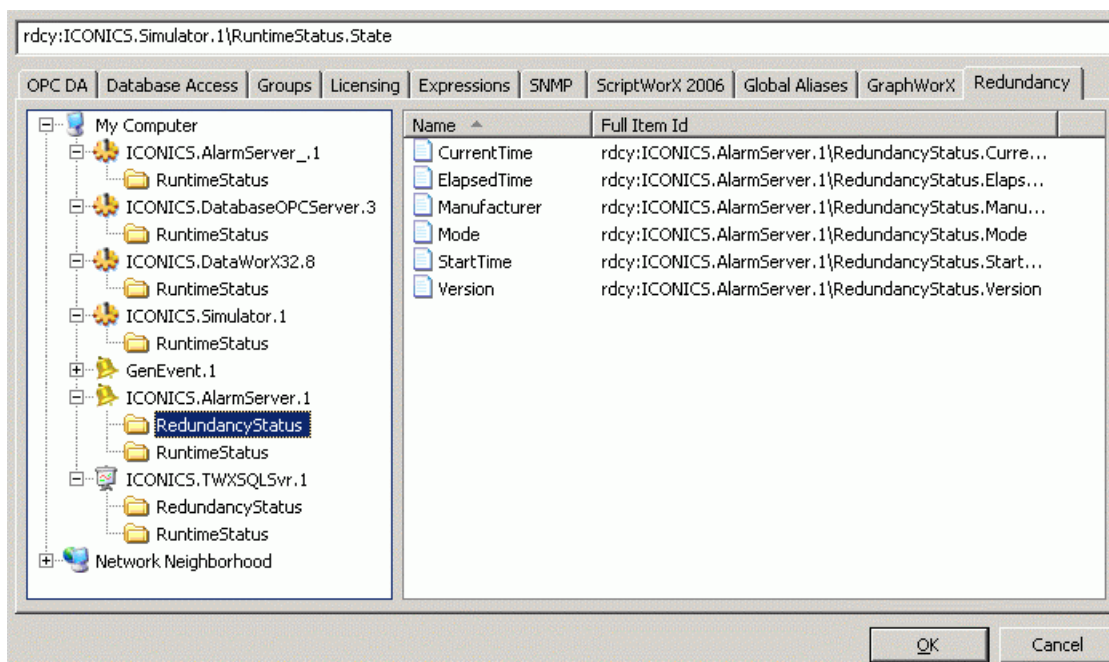
The **Redundancy Status** tags are available only for OPC AE and HDA servers on the nodes configured in the **Redundancy Node Pair** configuration in the GenBroker Configurator. These tags are listed under the "RedundancyStatus" folder and provide the single members of the redundancy status structure with status information provided by the given server (e.g., redundant mode, start time, elapse time, version, manufacturer, etc.). The list of the redundancy status tags is as follows:

Tag Name	Description
----------	-------------

Mode	Mode of the redundant server																
	<table> <tr> <th>Mode value</th><th>Meaning</th></tr> <tr> <td>0</td><td>Offline</td></tr> <tr> <td>1</td><td>Start up</td></tr> <tr> <td>2</td><td>Switch to Standby</td></tr> <tr> <td>3</td><td>Standby</td></tr> <tr> <td>4</td><td>Switch to Active</td></tr> <tr> <td>5</td><td>Active</td></tr> <tr> <td>6</td><td>Shutdown</td></tr> </table>	Mode value	Meaning	0	Offline	1	Start up	2	Switch to Standby	3	Standby	4	Switch to Active	5	Active	6	Shutdown
Mode value	Meaning																
0	Offline																
1	Start up																
2	Switch to Standby																
3	Standby																
4	Switch to Active																
5	Active																
6	Shutdown																
	This is the only writeable tag; writing to it will cause redundancy switchover.																
CurrentTime	Date/time value of the current time																
ElapsedTime	Date/time value of the elapsed time																
StartTime	Date/time value of the server start time																
Version	String value with the server version (e.g. 2.1.1).																
Manufacturer	String value with the manufacturer name																

A sample redundancy status tag is as follows:

rdcy:\\Primary\\ICONICS.AlarmServer\\RedundancyStatus.Mode

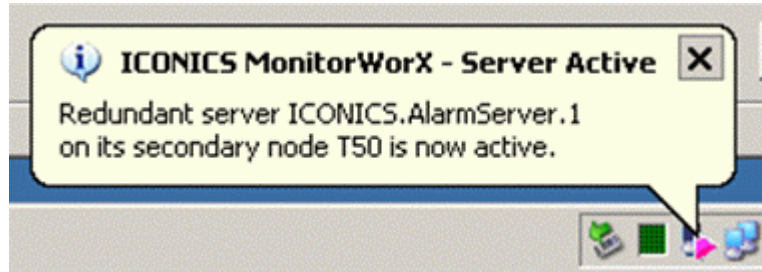


Server Redundancy Status Tags

MonitorWorX

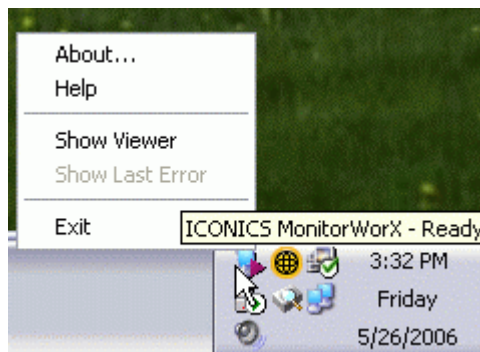
MonitorWorX is a client-side viewer that displays runtime status information about redundant node pairs, and allows you to designate which server is active or standby. Generally, the server on the primary node is the **active** server, while the secondary node runs the server in **standby** mode.

To launch MonitorWorX from the Windows **Start** menu, select **Programs > ICONICS GENESIS32 > MonitorWorX**. When MonitorWorX opens, an icon appears in the Windows tool tray, as shown in the figure below.



MonitorWorX Icon in Windows Tool Tray

To display the MonitorWorX user interface, right-click on the MonitorWorX icon in the Windows taskbar and select **Show Viewer** from the pop-up menu, as shown in the figure below.

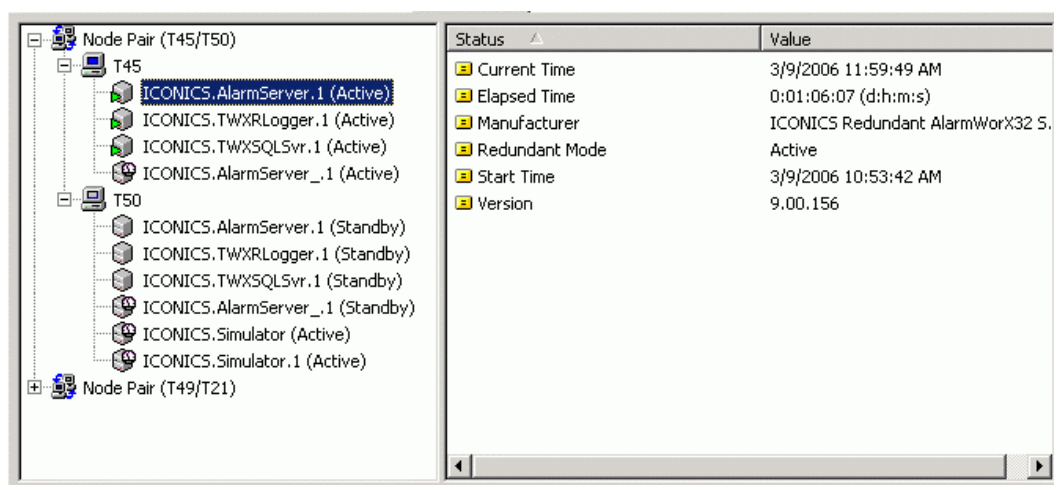


Displaying the MonitorWorX Viewer

Once you have completed configuration of a redundant node pair in the GenBroker Configurator and created connections to OPC tags from the client side (e.g. using GraphWorX32) that connect to the OPC alarm server or data access server on the Primary node, you are ready to start runtime mode. To do this, select the **Runtime** menu in GraphWorX32. When runtime is started, you will be able to see data coming from the subscribed servers on the redundant node pair.

The ICONICS Alarm Server already has embedded redundancy, so it will automatically designate the active/standby status of the alarm server. Redundancy is not inherent in OPC Data Access servers, however, so you can manually specify which data server is active (assuming that both nodes are currently online) using the MonitorWorX **Redundancy** tab.

Note: For more information about MonitorWorX, please see the MonitorWorX help documentation.

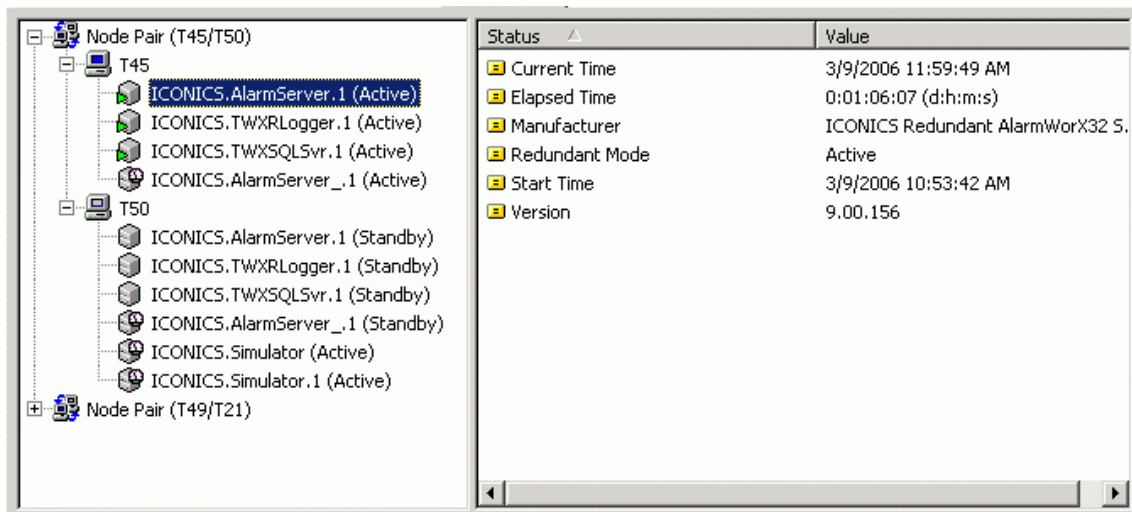


MonitorWorX: Redundancy Tab

Active vs. Standby

The figure below shows a redundant node pair in MonitorWorX. Each node has several servers that are currently running, including the AlarmWorX32 AE Server, a Data Access server, and the TrendWorX32 HDA Server. Each server on each node is indicated as "active" or "standby." The right-hand pane shows the runtime status of the currently selected server (e.g., current time, start time, elapsed time, version, etc.).

Generally, the primary node hosts the **active servers**, while the secondary node hosts the **standby servers**. The AlarmWorX32 AE Server already has embedded redundancy, so it will automatically designate the active/standby status of the alarm servers with respect to all client nodes. Redundancy is not inherent in OPC Data Access servers, however, so you can manually specify which one is active with respect to the local client (assuming that both nodes are currently online) using the **Redundancy** tab in MonitorWorX.

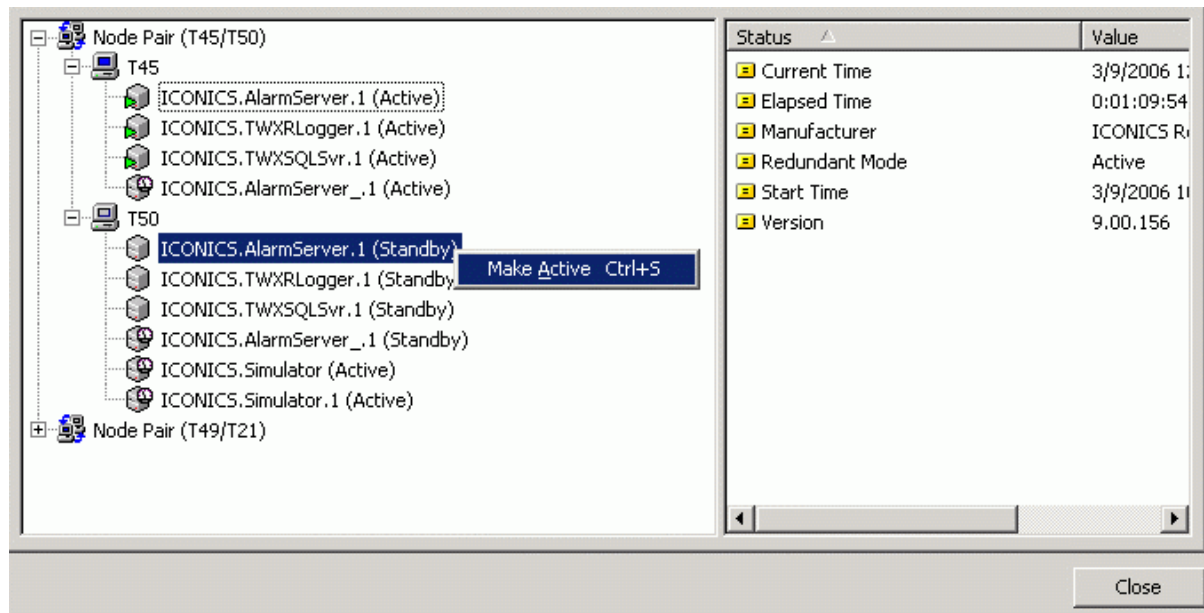


Active vs. Standby

Switching Between Active and Standby

To switch a "standby" server in a redundant node pair to "active," right-click the standby server in the MonitorWorX **Redundancy** tab and select **Make Active**, as shown in the figure below. This changes the status of the standby node to "active," and the currently active node is deactivated and therefore switches to "standby" mode. The active alarm server and trend server are also indicated by a green triangle icon, as shown in the figure below.

Note: When switching between active and standby servers, the amount of time to complete the switchover varies depending on the system size, server location, and configuration size.



Switching Server From 'Active' Mode to 'Standby' Mode

DataWorX32 OLE Automation Reference

DataWorX32 consists of two parts – the Configurator and the Runtime part. Both expose an OLE Automation interface; however the Configurator's interface is designed for use with ProjectWorX32.

The interface of the Runtime part is described below.

The name of the GENESIS32 DwXRuntime Automation Library is **AutoDwXRuntimeLib**. It defines the following classes and interfaces:

- **DwXRuntime**
- **IPoint**
- **IRegister**
- **IRedundancyAlias**

DwXRuntime Object

This object represents the whole DataWorX32 Runtime module. It contains other DataWorX32 objects like points and registers. It can be used to create or access them.

Methods

GetPoint(strName As String, nScanRate As Long) As IPoint

Creates a point object with the specified name and scan rate and returns an interface of it. The point object can be subsequently used to read and/or write values from/to OPC servers.

The **strName** parameter should meet GENESIS32 point name convention:

i.e. [\\node\\]OPCServer\\TagName.

The **nScanRate** is the requested scan rate in milliseconds.

GetRegister(strName As String) As IRegister

Returns an interface of an existing DataWorX32 register. This interface can be used to access the register's value and some of its properties.

GetRedundancyAlias(strName As String) As IRedundancyAlias

Returns an interface of an existing DataWorX32 Redundancy Alias object.

ConfigurationUpdate()

Forces DataWorX32 to reload the active configuration and include changes made since the configuration was loaded last time.

Shutdown()

Forces DataWorX32 to quit immediately. Any reference to DwXRuntime object becomes invalid after calling this function. Users should not call this function – it has been primarily designed for GENESIS32 GenTray.

Property**Runtime As Boolean**

This is a read-write property that tells whether DataWorX32 is in runtime mode. Writing to it starts or stops the runtime operation. When starting the runtime operation, DataWorX32 requests points on OPC servers; the data updates from the OPC servers are propagated during runtime (data bridging).

IPoint Interface

An interface to a point object can be obtained by calling the **DwXRuntime.GetPoint()** function. The point objects represent a non-persistent connection to OPC servers.

Properties**Name As String**

Read-only property; name of the relevant OPC point, in the GENESIS32 point name convention.

Value As Variant

Read-write property; represents value of the relevant OPC point. Writing to this property causes an asynchronous write to the OPC server.

Method**GetValueEtc(Value, Quality As Long, Timestamp As Date, Milliseconds As Long)**

Gets the current value, quality and timestamp of the relevant OPC point object.

IRegister Interface

An interface to a register existing in DataWorX32. It may represent registers of any type (i.e. aliases, switch aliases, etc.) defined in DataWorX32. This interface can be obtained by calling **DwxRuntime.GetRegister()**.

Properties

Name As String

Read-only property; name of the relevant DataWorX32 register.

Value As Variant

Read-write property; represents the value of the relevant DataWorX32 register. Writing to this property changes the value of the register. If the register has defined any output, the new value may be also written into the output.

Ranges As Boolean

Read-write property telling the user whether the ranges of the DataWorX32 register are valid and should be used.

HiRange As Double

LoRange As Double

Read/write properties containing ranges of the register. Writing to these properties modifies the ranges used in runtime to clip the register value. The range changes are not persistent (do not get stored into the active configuration database.)

REGISTER_TYPE RegType

Read-only property; type of the register, one of the REGISTER_TYPE values:

```
RT_REGISTER = 0
RT_ALIAS = 1
RT_REDUNDANCY_ALIAS = 2
RT_REDUNDANCY_FLAG = 3
RT_SWITCH_ALIAS = 4
```

InputConnected As Long

Input of the register can be temporarily disconnected. For example, let's assume a register is connected to "ICONICS.Simulator\Ramp" and you want to write a value through automation into the tag. If you write the value, then the Ramp would overwrite it. So you can temporarily disconnect input (InputConnected = FALSE), write something, and then enable the connection again with InputConnected = TRUE.

Methods

GetValueEtc(Value, Quality As Long, Timestamp As Date, Milliseconds As Long)

Gets the current value, quality and timestamp of the DataWorX32 register.

Sub SetValueQuality(Value, Quality As Long)

Allows you to set both the quality and value of a register.

IRedundancyAlias Interface

An interface to a redundancy alias existing in DataWorX32. It can be obtained by calling **Application.GetRedundancyAlias()**. It can be used to force a redundancy switchover.

Methods

SwitchToPrimary

Forces redundancy switchover to the node defined as primary.

SwitchToNext

Forces redundancy switchover to the backup node following the node that is currently used. If the currently used node is the last backup node, it will switch again to the primary node.

Properties

IsPrimary As Boolean

Returns TRUE if redundancy currently uses the node defined as primary; otherwise returns FALSE.

PrimaryNodeName As String

Returns the name of the node defined as primary.

CurrentNodeName As String

Returns the name of the currently used node.