

Description: Guide to setting up Advanced Redundancy Settings

OS Requirement: Win 2000, XP Pro, Server 2003, Vista, Server 2008, Windows 7

General Requirement: GENESIS32 Version 9.20, DataWorX32 Professional Redundancy license on Primary and Secondary servers; Administrative rights on primary and secondary server, SQL Server 2005 or 2008

Introduction

It is highly recommended to apply the latest service packs and hot-fixes to your system before implementing DataWorX32 Professional Redundancy. Please contact ICONICS Technical Support for details.

This document provides some advanced settings for GENESIS32 applications with regard to DataWorX32 Professional Redundancy in order to optimize the overall system performance.

Recommendations

Hardware and Operating System

ICONICS recommends that the same hardware and operating system be used on both nodes in a DataWorX32 Professional redundancy pair. This also includes all system updates and service packs from Microsoft.

SQL Server Configuration

If the server machine is running on a 64bit operating system, it’s strongly recommend to use Microsoft SQL Server 64 bit, since it improves performance by at least 25% over the 32 bit version.

SQL Server must be configured properly to help ensure efficient data logging. Microsoft SQL Server will utilize all of the available memory by default. If SQL Server is installed on the same machine as your ICONICS servers (AlarmWorX32, TrendWorX32, DataWorX32, OPC Servers, the Unified Data manager, etc.), then we suggest following:

- Make sure that Microsoft SQL Server is NOT using more than 50% of available system RAM

- Since Microsoft SQL Server treats Virtual memory and real (physical) RAM as one pool of memory resources, make sure that the Virtual RAM (paging file) is large enough to cover Microsoft SQL Server and all of the applications you will have running on a normal basis. Setting it to a high value (i.e. 4 GB or more across multiple drives) can significantly enhance performance.

For alarm and trend history logging ICONICS recommends the following settings:

SQL Server Memory Options

On dedicated SQL Server machine, the minimum memory should be set to at least 50% of the available RAM. The maximum should be unlimited. Since the machine will be dedicated for SQL Server we can allow it to use much more memory than in the previous section.

Open SQL Server management studio and right click on the server name in the Object Explorer to open server properties.

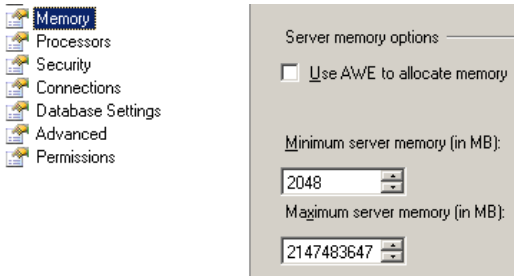


Figure 1 - SQL Server Memory Options (4 GB Available)

NOTE: A limitation of any of these settings may decrease the SQL Server performance when doing a large amount of data logging.

SQL Server Database Recovery Model

The performance of SQL Server database can be increased by changing the default recovery model from FULL to SIMPLE. In the SQL Server management studio navigate to your database, right click on the database name and select Properties →Option.

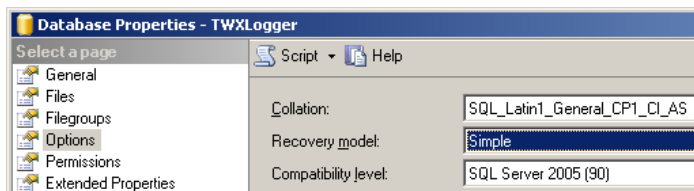


Figure 2 - Recovery Mode of the TWXLogger Database

Database Redundancy Considerations

RAID 5 should be considered to help mitigate the effect of hard drive failures.

It is recommended that the redundant GENESIS32 application use a highly available database storage medium like a SQL cluster.

SQL Server Database File Options

When storing large amount of data into a database, it is recommended to set the initial size and growth file options to appropriate levels such as:

- Initial size : 8192MB
- Auto growth : 1024MB

In the SQL Server management studio navigate to your database, right click on the database name and select Properties → Files or define this setting during creating the database.

Database files:				
Logical Name	File Type	Filegroup	Initial Size (MB)	Autogrowth
TwXLogger	Data	PRIMARY	8192	By 1024 MB, unrestricted growth
TwXLogger_log	Log	Not Applica...	1024	By 10 percent, restricted growth to 4096 MB

Figure 3 - Database Size Options

NOTE: After making these changes it will be necessary to restart the SQL service.

GENESIS32 Settings

The following settings are recommended for highly loaded GENESIS32 applications.

GenBroker

Before you start, it is recommended to review the GenBroker redundancy settings. Open GenBroker configurator and check or configure these settings:

1. **Channels and Nodes** – Set the OPC over TCP/IP Direct Channels as default (assuming there is no mediator node needed)

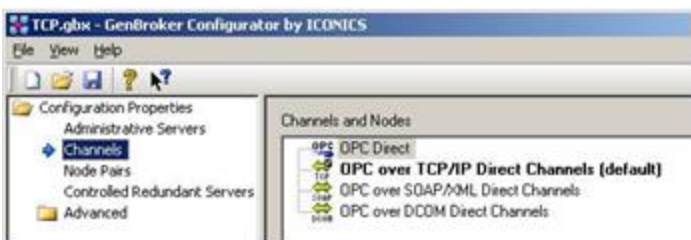


Figure 4 - GenBroker Channels Settings

2. **Node pairs** – Review the settings of the redundant Node Pairs on all servers and all clients.

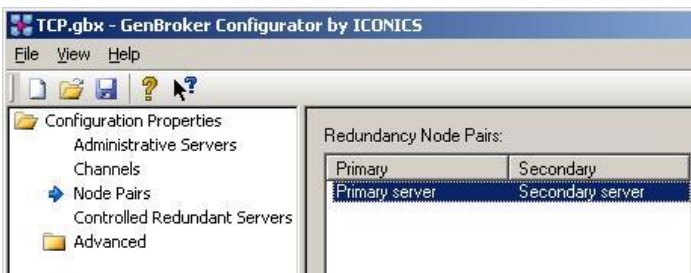


Figure 5 - GenBroker Node Pair Settings

3. Double click on the Node pair to open the Node Pair properties. For fastest switchover we recommend checking “Pre-request points on the secondary node”.

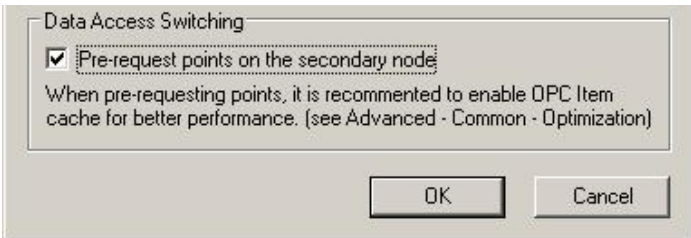


Figure 6 - GenBroker Node Pair Settings

4. When you enable this option, it is also recommended to enable OPC Item cache. Click OK to the Node Pair properties, then select Advanced → Common → Optimization and here setup the Item Cache.



Figure 7 - Cache Size Setting

ICONICS Remoting Service32

If you are not using Workbench32, you can disable the ICONICS Remoting Service 32.

To do this, go to Start → Run and type “services.msc”. This command will open the Services Control Panel. Double-click on ICONICS Remoting Service 32 and set the startup type to Disabled.

AlarmWorX32 Logger

In order to achieve efficient archiving of large tables with logged alarms and events the following entries should be entered into the default IcoSetup.ini file:

```
[AWX32\AWXLog32]
ArchivePeriod=900000
SetArchiveMode=2
MaxRecordChunk=50000
StartupDelay = 20000
```

- **ArchivePeriod** sets the time period for the logger to check if archiving needs to be performed (in milliseconds).
- **SetArchiveMode** = 2 enables high performance alarm archiving (for more information about SetArchiveMode values please refer to the application note entitled *AlarmWorX32 Logger - Efficient and Robust Alarm Archiving*).
- **MaxRecordChunk** defines the number of records that will be archived within one single write command. The archiving will automatically perform multiple write commands as needed.
- **StartupDelay** sets the time period for the logger to wait after it starts up before it begins logging. Sometimes the AlarmWorX32 Server is not ready due the redundancy switch, so the logger will wait for a period of time to help ensure it doesn't start before the AlarmWorX32 Server is ready.

AlarmWorX32 Server

You may want to set the Scan Period in the AlarmWorX32 Server to something slightly different than the update rate of your OPC DA server to avoid missing updates if they happen to come in just a millisecond after the AlarmWorX32 Server updates. For example, if your OPC DA server updates its values once every 500 milliseconds you may want to set the Scan Period in the AlarmWorX32 server to 505 milliseconds or something even shorter, like 256 milliseconds.

To change this in the AlarmWorX32 Server Configurator, click on your configuration and change the Scan Period to an appropriate value.

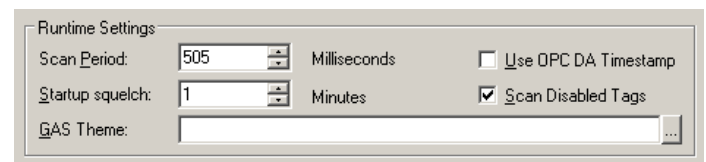


Figure 8 - AlarmWorX32 Server Scan Period

TrendWorX32 Logger

Tables that are too large or too small can cause performance issues, therefore it's recommended to configure your logging groups to create new tables on a fairly regular basis. For heavy logging it's usually good to create a new table every day.

The data collection and logging rate should be set according to the TrendWorX32 documentation. Remember that the data logging period should be is significantly longer than the data collection period for best performance.

MonitorWorX

MonitorWorX is the runtime module that evaluates and determines the state of ICONICS servers (AlarmWorX32 Server, AlarmWorX32 Logger, TrendWorX32 Logger, TrendWorX32 HDA Server) on all connected nodes in a GENESIS32 redundancy applications in a decentralized manner. In addition to standard features you can add the following to the IcoSetup.ini file:

```
[MonitorWorX]
Gravity=1
```

- **Gravity** feature sets the ICONICS Servers (AlarmWorX32 Server, AlarmWorX32 Logger, TrendWorX32 Logger, TrendWorX32 HDA Server) to subscribing to OPC DA data sources on the local machine. This will provide a better performance when a redundancy switch occurs for any of the ICONICS Servers named above. The default value is 0. This



DataWorX32 Professional – Recommended Redundancy Settings



APPLICATIONS NOTE

October 2013

feature has been designed namely for cases when the Auto Fail-back settings has been turned off.

[RedundancyDll]

AdapterName=Local Area Network

- AdapterName parameter should be set equal to the network name as shown in your Control panel / Network connections. This network manages the connection to the counterpart node in redundancy setup. By using this feature the local machine will detect a local network disconnection. Then after reconnection it will use the most recent states of operations from the counterpart machine. This feature has been designed only for cases when the Auto Fail-back settings has been turned off.

Disabling redundancy for individual items

Sometimes users want to disable redundancy switching. This may be desired for cases such as:

- Diagnostic purposes, such as determining when the primary OPC server is offline or to compare the values seen by the primary and secondary OPC servers.
- In combination with GenBroker Controlled Redundant Servers feature (see the help files for more information about this feature).

GENESIS32 allows the user to prefix the Primary server machine name with an exclamation mark (“!”) in order to disable the logic that will switch the client to the redundant secondary OPC server should the primary go offline. Tags or subscriptions that use the exclamation point will be treated as if the Node Pair in the GenBroker Configuration did not exist.

As an example of how to use this syntax, assume we have a GraphWorX32 display with an OPC DA point. Under standard conditions the user subscribes a process point with this syntax:

`\\PrimaryServer\ICONICS.Simulator.1\SimulatePLC.Sine`

If there is a Node Pair configured in GenBroker for PrimaryServer, when PrimaryServer went offline, this tag would begin looking at the secondary server automatically. The user may not even be aware that a failure has occurred.

For this example, we want to be able for the user to tell that the primary server has failed, so we will change the tag to use this syntax instead:

`\\!PrimaryServer\ICONICS.Simulator.1\SimulatePLC.Sine`

Note that the PrimaryServer machine name has been prefixed by the exclamation mark “!”. This will treat that one tag as if there was no Node pair configured for PrimaryServer.

The same principle can be used in any ICONICS module that subscribes to OPC DA points, such as AlarmWorX32 Server, DataWorX32 registers, GraphWorX32 dynamics, etc.

This syntax can also be used in an ICONICS AlarmWorX32 Viewer or AlarmWorX32 Logger subscribed to an ICONICS AlarmWorX32 Server. Adding an exclamation point to the node name will force the subscription not to fail over, as if there were no Node Pair defined for *Primary Server*:

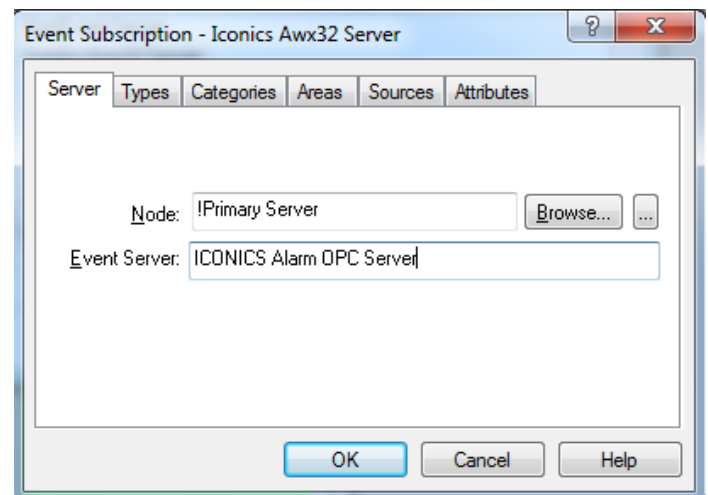


Figure 9 – Disabling redundancy for AlarmWorX32 Server Subscription