GENESIS64 FrameWorX
GenBroker Advanced Setup

APPLICATION NOTE

September 2013

Description: Guide to setting up GenBroker for remote communications to a legacy OPC server
OS Requirement: Windows XP x64/Server 2003 x64/Vista x64/ Server 2008 x64/Windows 8 x64/Server 2012 x64
General Requirement: installation of GENESIS64

Introduction

All GENESIS64 clients to server OPC data communications goes through the FrameWorX64 server, which is an OPC UA server. Communicating through this server has many benefits, however there will be times that you need to use the GenBroker/GenClient technology for communication.

Consider the scenario where you are running an OPC DA server on are remote machine, let's call it PC1. On PC2, you have installed GENESIS64 and you need to get data from the PC1's OPC DA server. GenBroker is running on PC1 and normally, when you are using a GENESIS32 system that's all you need. In GENESIS64, the information needs to go through the FrameWorX server. There is backwards compatibility built into the FrameWorX server. If it detects the remote communication is to a legacy server (e.g. OPC DA, HDA, or A&E), it will switch to use the GenBroker/GenClient technology to obtain the data.

In this document, we will detail the steps necessary for you to receive data in a GENESIS64 system in the scenario described above.

Server Side Settings

On the machine where your OPC server is running, the only thing you need to do, apart from DCOM configuration, is to install GenBroker and run it. Remember, all GenBroker configuration should be done on the client side, and this is true regardless of the version you are using.

> **Note:** For 3rd party OPC server communication you can use DataWorX32 Tunneling. You can refer to application note called *DataWorX32 - OPC Tunneling*.

Client Side Settings

On the machine where GENESIS64 is installed is where we need to do most of the work. For the most part, it works out of the box, but you can follow the steps below to make sure that it is setup correctly.

1. Start Workbench by going to Start → Programs → ICONICS GENESIS64 → Workbench.
2. In the Project Explorer section of the Workbench, click on FrameWorX Configuration button.
3. Expand localhost -> Classic OPC and right-click on GenBroker to select "Edit on New Tab".

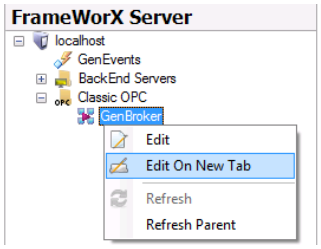

Figure 1 - FrameWorX Configuration: GenBroker

4. The GenBroker configuration should open in a new tab on the right-hand side displaying the Channels tab.
5. The default channel is **OPC over TCP/IP Direct Channels**. If you would like to use a different channel for default, you can right-click on the channel and select "Set as Default". If you are not sure which channel to select as the default one, you can refer to the application note called *GenBroker - Which Communications Channel is Right for You* for more information.
6. On this tab, you can also create new channels. You generally will not need to create one. However, in some cases where there is a firewall between your GENESIS64 machine and the remote OPC server, for example, you may need to create a mediator node. In such cases, you can simply click on the prefered channel of communication and enter your mediator node name or IP address.
7. With this information configured, you should be able to get data from your remote machine.

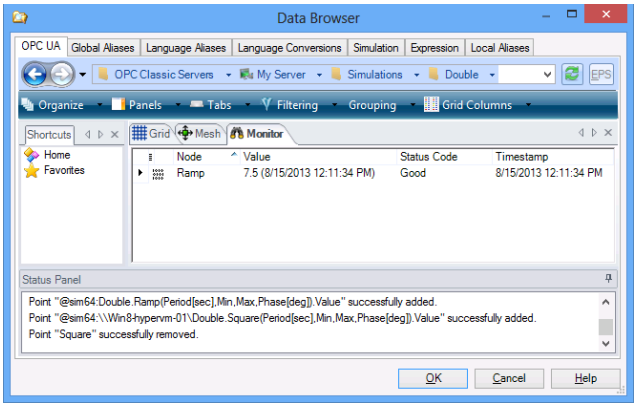

Figure 2 - Remote Tag Value in Monitor View of Data Browser

Additional GenBroker Settings

There are other settings that allow you to tune in GenBroker for the client. These settings are the same as they were in GENESIS32. Please find them under Advanced tab (as you can see in screenshots below).

Tag Syntax

Use the Tag Syntax tab to configure tag extension syntax and XML DA syntax. You use this tab to redefine separators for OPC item names. Tag Extension Syntax field can be used to redefine default separators for OPC item names over TCP\IP, DCOM and OPC Direct channels communication. XML DA Syntax can be used for redefine separators for SOAP/XML channel communication.
Describing whole syntax is too complex for Application note, please, refer to Help file.

Note. Any changes you make can be undone by clicking the Default button.

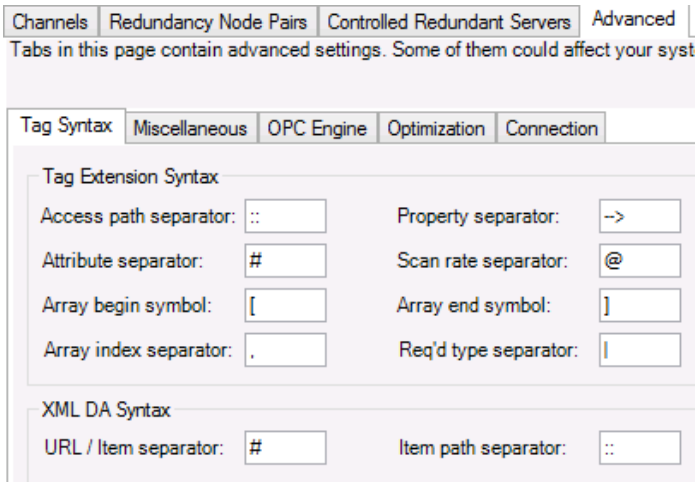

Figure 3 – Tag Syntax

Miscellaneous

Miscellaneous dialog, shown in the figure below, configures the following settings:

- Client Side Cache
- Tag Retry Rate
- Sync Call Timeout

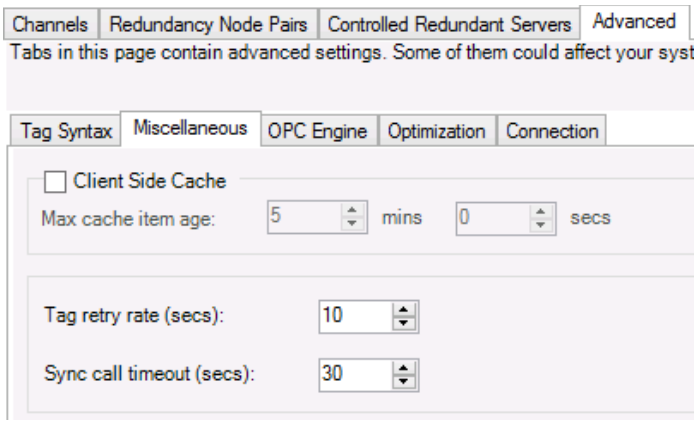

Figure 3 – Miscellaneous

Client Side Cache

Checking this check box allows you to specify the amount of time (in minutes and seconds) to cache data on the client side.

Tag Retry Rate

The Tag Retry Rate defines the rate (in seconds) at which the client tries to request failed tags or OPC data items from the server. Failed alarm and event subscriptions are also requested at the same rate.

Sync Call Timeout

The Sync Call Timeout specifies the default timeout rate (in seconds) for synchronous calls. Some of the GenBroker functions (e.g. registering applications, login and logout to and from the Security Server) fail if the relevant server does not respond in the time specified.

OPC Engine

The OPC Engine dialog, shown in the figure below, sets the OPC server connection control parameters.

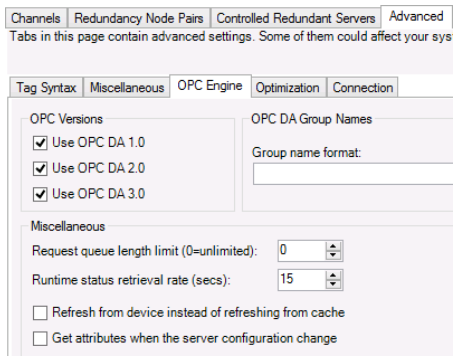

Figure 4 – OPC Engine

OPC Versions
The OPC Versions section tells the GenClient which OPC version it is allowed to use. Select a version of OPC Data Access. Check all that apply.

OPC DA Group Names
Specify an OPC Data Access group name in the Group Name Format field.
When GenClient creates an OPC group on a DA server, it may specify its name depending on the format defined here. Asterisks (*) are replaced by the group numbers, so each group has a unique name. If there are no asterisks in the format, the group number is appended. When no group name is specified here, GenClient does not specify the names and lets the server create them.

Note: The group names are not needed for normal operation. Some servers may require them for some specific purposes (e.g. security).

Miscellaneous
A server may get stuck or so slow that it would not be able to process requests from GenBroker, but it still may be considered alive. Then the requests to the server are stored in a queue in GenBroker. The **Request Queue Length Limit** defines the maximum size of the queue. It is a good practice to have the queue greater than the number of points on the largest display.

GenBroker now supports optional OPC item caching. You can specify a time the cache will keep an item requested. When a client releases an OPC item (e.g. a tag), the item is not actually removed from the server but instead is moved into the item cache. Then, when the client requests the same item again, the item is retrieved from the cache instead of being retrieved from the OPC server again. However, if **Refresh from Device instead of refreshing from cache** is checked, the item is retrieved from the i/o device to which the OPC server is connected.

The **Get Attributes when the server configuration change** option may be used to force the client to re-acquire attributes (ranges, canonical data type, access rights). If this check box is checked and the server sends a bad quality value with substatus "configuration error", the client re-acquires the attributes. If this check box is not checked, then the attributes are only obtained at the beginning.

Optimization

The Optimization dialog, shown in the figure below, sets the server-client connection parameters.

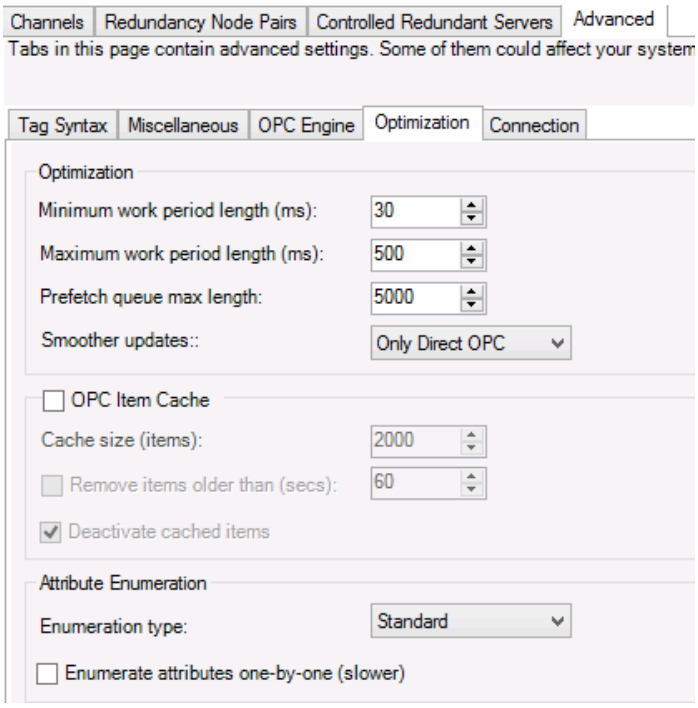

Figure 6 – Optimization

Work Period Limits
GenBroker periodically collects requests from the clients. It uses an adaptive algorithm to set the collection rate, but the rate (in milliseconds) is always within ranges defined in **Work Period Limits**. The longer the period, the more requests GenBroker may collect. Requests collected in one period are processed together, which is more efficient than processing them separately. Also the longer the period is, the longer the clients have to wait for responses on the requests. So a short period means quicker responses but less efficient processing of the requests. With a longer period, the DCOM calls may be more efficient, but the response delay is longer.

When GenBroker is idle, it may prepare updates in advance. As a result, it will send more updates to the client, so it will run more smoothly. But it also means more traffic over the network and more CPU utilization. Turn this behavior on with the **Smoother Updates** drop-down list. **Prefetch Queue Max Length** limits the number of updates prepared in advance.

OPC Item Cache
GenBroker supports OPC item caching. You can specify a time the cache will keep an item requested. When a client releases an OPC item (e.g. a tag), the item is not actually removed from the server but instead is moved into the item cache. Then, when the client requests the same item again, the item is retrieved from

the cache instead of being retrieved from the OPC server again. You can define how long OPC items should be stored in the cache. Optionally, GenBroker may deactivate the items put in the cache, thereby decreasing the number of updates coming in from the OPC servers (otherwise the items in the cache are also updated).

The cache mechanism is optional and may be switched on or off. It improves the speed of screen switching. Caching takes place for both OPC Direct operation and operation via GenBroker channels.
The **OPC Item Cache** check box enables/disables the cache. When the cache is enabled, released OPC items are then put into the cache rather than removed from the server. When the items are requested again, they will be taken from the cache. **Cache Size** specifies the maximum number of items that will be stored in the cache. When **Deactivate Cached Items** is checked, the items are deactivated when put into the cache. **Remove Items Older Than (X) Seconds** defines how long (in seconds) the items should be kept in the cache.

Attribute Enumeration
The Attribute Enumeration section specifies how enumeration is made. Enumeration is a way to get OPC items' ranges and access rights:

- **None**: Turns enumeration off so the attributes are not obtained. When ranges are not required, this could improve display switching speed.
- **Standard**: Enumerates all items in the current OPC group. When an item is added to a group (e.g. an item in a pop-up window), the whole group is enumerated again.
- **Intelligent**: A separate group is created for the newly added items, and the new group is enumerated and then removed. Intelligent enumeration only enumerates the newly added items, but it needs some overhead to create and destroy the auxiliary group. Standard enumeration does not have this overhead, but it also enumerates items already on the display. With large displays with pop-up windows, intelligent enumeration should be faster. When just switching between displays, standard enumeration will be faster because of no overhead.
- **Enumerate Attributes One By One**: Forces GenBroker to ask for attributes one by one rather than getting them all together in one call. In the past, some servers had compatibility problems with gathering all attributes.

Connection

The Connection dialog, shown in the figure below, sets the OPC server connection control parameters.

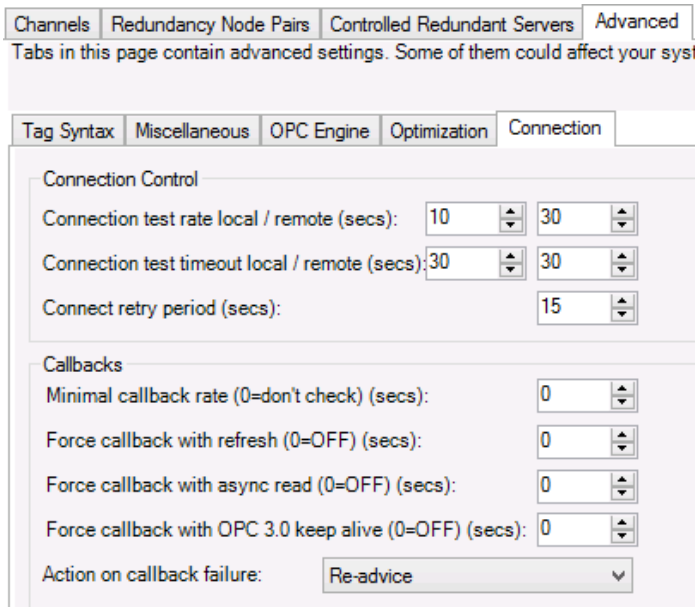

Figure 5 – Connection

Connection Control and Callbacks
The Connection dialog define the following parameters for server crash detection:

- **Connection Test Rate**: Defines, how often (in seconds) GenBroker should check if the OPC servers are "alive." The test is performed by calling the GetStatus() function of the servers.
- **Connection Test Timeout**: Specifies how long (in seconds) the call to GetStatus() may take for an OPC server to be considered "alive."

  Note: Both the Connection Test Rate and Connection Test Timeout values can be set differently for local and remote OPC servers. A remote server is actually local for GenBroker located remotely, but on the same node as the server.
- **Connect Retry Period**: When GenBroker cannot connect to a server, it will retry after the time specified (in seconds) in Connect Retry Period.
- **Minimal Callback Rate**: Specifies how often (in seconds) OPC servers should update values to be considered alive. A value of 0 switches this test off.

  Note: Not sending updates is OK for a server when its values are constant. But with Forced Refresh Rate on, the server must

refresh even the constant values. These options can be used for detecting a situation, when the connection to the server works fine, but the callback connection (from the server to the client) has failed.

- **Force Callbacks with Refresh**: Specifies if and how often (in seconds) GenBroker should force OPC servers to refresh the data values. A value of 0 turns this off.
- **Force Callbacks with Async Read**: Forces GenClient to periodically asynchronously read one item in each OPC group to check the state of the callback connection. This is a preferred method before forced refresh. A value of 0 turns this off.
- **Force Callbacks with OPC 3.0 Keep Alive Calls**: Switches on using Keep Alive callbacks defined in OPC 3.0. A server conforming to OPC DA 3.0 may periodically send empty callbacks so the client can check that the callback connection is OK without the need to read or refresh anything. A value of 0 turns this off.
- **On Callback Failure**: Tells GenClient what to do when a callback connection failure is detected: Re-advise, Re-create OPC Group, or Re-Create Server.