



Description: Introduction to using GenBroker communications with TCP/IP protocol.

OS Requirement: Windows 2003/XP/2000/NT, 9x/Me

General Requirement:

- General knowledge of GENESIS32, which should be installed on the machines being configured.
- General knowledge of windows networking.

Why Use GenBroker?

First there was DCOM, a promising technology embraced by ICONICS and the OPC Foundation. The DCOM interprocess communication is fully utilized in the GENESIS32 products. It allows a seamless connection to the OPC Servers available to the automation market.

The unparalleled capabilities of GENESIS32 and DCOM led to impressive cost reductions in advanced engineering projects. However, DCOM also exposed some technical limitations, which made it hard to use in complicated networks where OPC servers were being connected to the factory's intranet or to the public Internet. To provide a robust industrial solution, DCOM needed a helping hand. GenBroker was developed by ICONICS, and incorporates OPC and Security communications through TCP/IP or SOAP/XML channels. It achieves real-time and secure communications to and from OPC servers (data access, alarm & event, historical data).

DCOM Problems:

- The complex settings often make its configuration ineffective when the client is remote.
- It does not work well across NT domains.
- Connection over slow or unreliable networks (e.g., the Internet) is practically impossible.
- It is not easily configurable for passing data through firewalls.

The GenBroker Solution:

- 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.

Any computer running OPC Servers can now be quickly intranet or Internet enabled by simply running the GenBroker service. The computer will then only need some kind of network connection such as a dialup modem, network card or other Internet connection.

GenBroker is natively available to GENESIS32. Because of this, the user can even browse the intranet and Internet for OPC

Servers and OPC Tags that may be required in a GENESIS32 application. In fact, even third party OPC Data Access client applications can be instantly WEB enabled using GenBroker technology. These clients can address their Data Access OPC tags through DataWorX32.

GenBroker communications also makes ICONICS WebHMI possible. Because of its web nature, WebHMI is used in applications where thin clients need to get access to SCADA/HMI graphics, alarms, and trend data. The SCADA/HMI engineers can develop their process interface using the standard GENESIS32 development tools from any remote location. They can then publish the developed application to the WebHMI server. The visitors of this web site will have instant access to the application including full security, read/write capabilities, and alarm acknowledgement, just like GENESIS32.

GenBroker Philosophy

The basic philosophy of GenBroker communications lies in its use of reliable **Request / Response** protocols: The client sends requests, and the servers reply to them. GenBroker applies to all GENESIS32 applications.

For example, in WebHMI a thin client sends out a request over the Internet to the WebHMI server. The thin client uses GenBroker to transmit the request. The WebHMI server's response to the request is also returned via GenBroker. But if the thin client's request requires a response from a remote server in a local area network (LAN), WebHMI could be configured to use DCOM. The machine will use DCOM to return the response to the WebHMI server, which will use GenBroker to relay the response to the thin client.

GenBroker enables you to designate which method is to be used for networked communications, as well as to define a central License and/or Security server for a networked set of workstations.

GenBroker/GenClient Architecture

The GenClient/GenBroker architecture consists of two main parts: GenClient.dll and GenBroker.exe. The GenClient.dll resides on the client side. It can communicate directly with OPC servers (using the traditional DCOM method), or it can establish a connection with GenBroker, which then communicates with the OPC servers on behalf of the client. Unlike DCOM, GenBroker can communicate over the Internet.

The client computer that wants to reach out to this OPC Data will have a GenBroker configuration set up to identify the computers that expose OPC Data. An example can be seen in **Fig. 1**, where a computer called **Myopcsrver** exposes relevant



OPC data via TCP/IP Communications. This figure also illustrates the four different channels previously discussed.

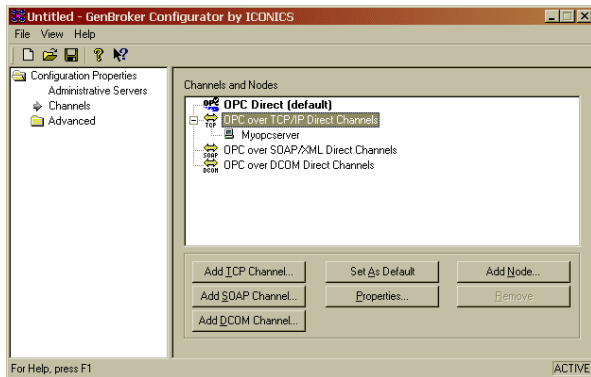


Fig. 1: Example GenBroker Configuration (.gbx) File

GenBroker for GENESIS32 to GENESIS32 Communication

GenBroker Communications can be used for OPC communications between two GENESIS32 machines. This type of GenBroker communications is illustrated in **Fig. 2**. In this example, the GENESIS32 machine at the top of the flow chart runs all of the GENESIS32 OPC Client applications, and the GENESIS32 machine at the bottom of the flow chart runs the OPC Servers and GENESIS32 Server applications. GenBroker communications can be enabled on any GENESIS32 system to make OPC data on one GENESIS32 machine available to other GENESIS32 machines. For more information on how to set this up, refer to the application note entitled “GenBroker Quick Start”.

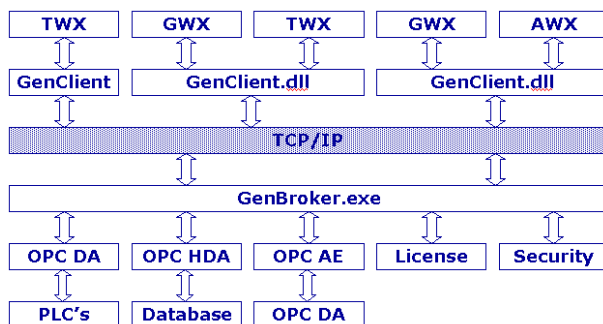


Fig. 2: GenBroker Communications Flow Chart

GenBroker for WebHMI Communication

As previously mentioned, GenBroker Communications is also used in ICONICS WebHMI to communicate OPC data from the WebHMI server to the clients. The means of communications is virtually the same in this case as it is between two GENESIS32 systems. In the example shown in **Fig. 2**, the top of the flow

chart represents client applications running on the WebHMI client, and the bottom of the flow chart represents applications running on the WebHMI server.

NOTE: GenBroker communications does not have to be enabled for GENESIS32 to GENESIS32 OPC communication in order to use it for WebHMI and vice versa. Each of the two types of GenBroker communications requires their own individual .gbx file, and they are completely independent of one another.