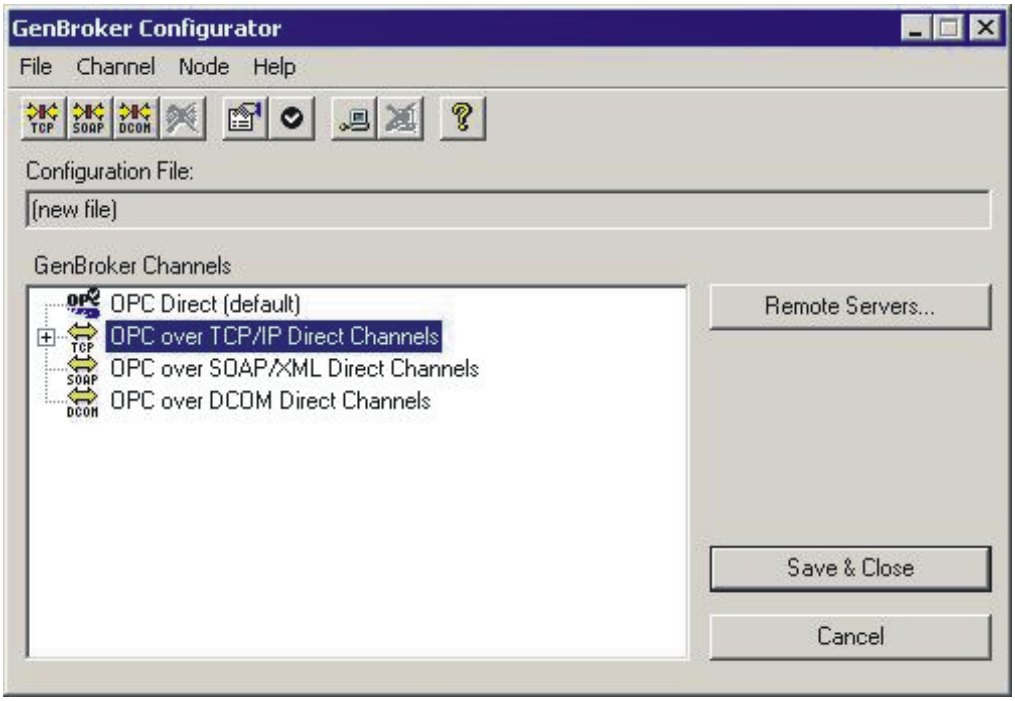
GenBroker

smar

FIRST IN FIELDBUS

USER`S GUIDE

GenBroker



JUN / 04

GenBroker

VERSION 7.1

FOUNDATION

PVI EWGBKME

smar

web: www.smar.com

Specifications and information are subject to change without notice.
For the latest updates, please visit the SMAR website above.

BRAZIL
Smar Equipamentos Ind. Ltda.
Rua Dr. Antonio Furlan Jr., 1028
Sertãozinho SP 14170-480
Tel.: +55 16 3946-3510
Fax: +55 16 3946-3554
e-mail: smarinfo@smar.com

ARGENTINA
Smar Argentina
Soldado de La Independencia, 1259
(1429) Capital Federal – Argentina
Telefax: 00 (5411) 4776 -1300 / 3131
e-mail: smarinfo@smarperifericos.com

CHINA
Smar China Corp.
3 Baishiqiao Road, Suite 30233
Beijing 100873, P.R.C.
Tel.: +86 10 6849-8643
Fax: +86-10-6894-0898
e-mail: info@smar.com.cn

FRANCE
Smar France S. A. R. L.
42, rue du Pavé des Gardes
F-92370 Chaville
Tel.: +33 1 41 15-0220
Fax: +33 1 41 15-0219
e-mail: smar.am@wanadoo.fr

GERMANY
Smar GmbH
Rheingaustrasse 9
55545 Bad Kreuznach
Germany
Tel: + 49 671-794680
Fax: + 49 671-7946829
e-mail: infoservice@smar.de

MEXICO
Smar México
Cerro de las Campanas #3 desp 119
Col. San Andrés Atenco
Tlalnepantla Edo. Del Méx - C.P. 54040
Tel.: +53 78 46 00 al 02
Fax: +53 78 46 03
e-mail: ventas@smar.com

SINGAPORE
Smar Singapore Pte. Ltd.
315 Outram Road
#06-07, Tan Boon Liat Building
Singapore 169074
Tel.: +65 6324-0182
Fax: +65 6324-0183
e-mail: info@smar.com.sg

USA
Smar International Corporation
6001 Stonington Street, Suite 100
Houston, TX 77040
Tel.: +1 713 849-2021
Fax: +1 713 849-2022
e-mail: sales@smar.com

Smar Laboratories Corporation
10960 Millridge North, Suite 107
Houston, TX 77070
Tel.: +1 281 807-1501
Fax: +1 281 807-1506
e-mail: smarlabs@swbell.net

Smar Research Corporation
4250 Veterans Memorial Hwy.
Suite 156
Holbrook , NY 11741
Tel: +1-631-737-3111
Fax: +1-631-737-3892
e-mail: sales@smarresearch.com

Index

GenBroker

Introduction to GenBroker Communications

DCOM is the method of communication for ProcessView. There are, however, certain disadvantages to using DCOM. To overcome DCOM timeout problems, for example, it was necessary to eliminate Windows callbacks and reduce the number of different function calls over Windows DCOM communications.

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 practically impossible.
- It is not easy to configure to pass through firewalls.

In response to these difficulties, Smar has enhanced ProcessView 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 browser clients and WebHMI servers. Everything except the method of communication remains the same as before.

GenBroker Fundamentals

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 ProcessView applications.

For example, 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 technology has been incorporated to make the setup of Internet and intranet communications easier. In addition, GenBroker provides easier Internet connectivity and operation with firewalls and routers.

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.

GenClient/GenBroker 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.

Advantages of GenClient/GenBroker TCP Communication

GenClient/GenBroker TCP 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.

Communication Types

GenBroker uses the following basic methods of networked communications:

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

Most channels, either direct or indirect, translate the OPC COM calls into one of two protocols and transmit them to the remote node, where GenBroker can execute them. (The OPC Direct channel is the only exception; it uses OPC calls.) Protocols that every channel can use for these transmissions are:

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

Note
In order to use SOAP/XML channels in GenBroker, you must first install Microsoft Internet Information Server (IIS) on the GenBroker Server node.

OPC Direct

In an OPC direct channel, the client applications, such as GraphWorX, TrendWorX, and AlarmWorX, communicate directly with OPC servers. As, shown below, this method uses standard fast DCOM over an intranet or local area network (LAN). This method of communication is similar to that of OLExpress.

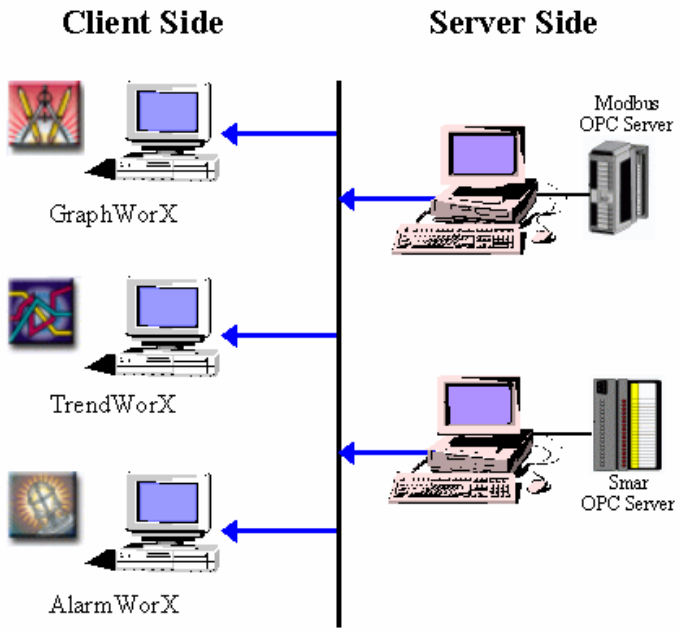


Figure 1. OPC Direct (Intranet Only)

Using a Direct Channel Over DCOM

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

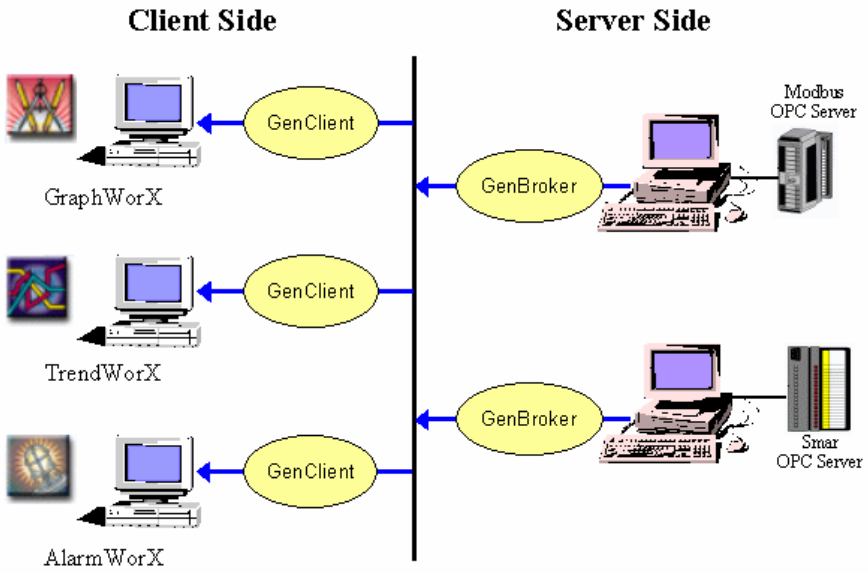


Figure 2. OPC Direct Channel Over DCOM (Intranet Only)

Using a Direct Channel Over TCP/IP

In a direct channel over TCP/IP, the client applications communicate with GenBroker software located on remote nodes on an intranet or LAN as well as over the Internet, as shown below. This method allows you to set a connection retry period and timeout settings. GenBroker then communicates (locally) with the target OPC servers on behalf of the client application. There are predefined direct channels that users cannot delete.

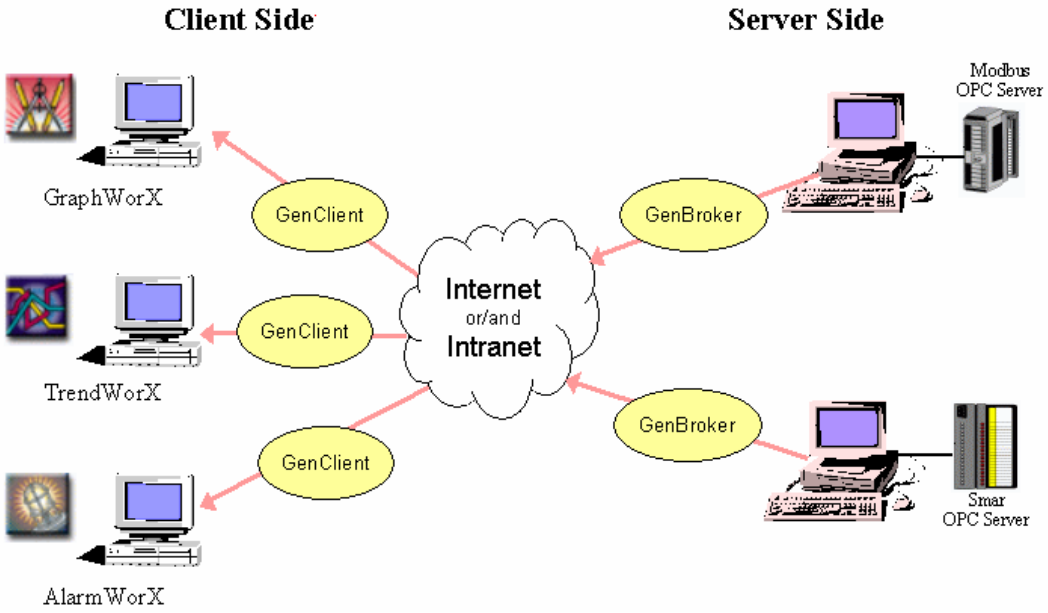


Figure 3. OPC Direct Channel Over TCP/IP (Intranet and/or Internet)

Using a Direct Channel Over SOAP/XML

In a direct channel over SOAP/XML, GenBroker responds to requests from clients that use SOAP/XML communications. SOAP (Simple Object Access Protocol) is a powerful Internet communications protocol that uses XML (e**X**tensible **M**arkup **L**anguage) transmitted over HTTP (Hyper Text Transfer Protocol). The main advantage of using SOAP is that, because SOAP operates at the HTTP level (above TCP/IP), data can pass through a firewall without having to change the firewall settings. In other words, SOAP uses the same protocol that we usually use to get Web pages. HTTP uses port 80 to establish a communication with a server, and this port is usually enabled on most firewalls. The disadvantage of SOAP is that it is slightly slower than TCP/IP.

XML has become a preferred format for encoding and moving data in an open, system-independent way. XML is a highly structured text format that is both machine- and human- readable. XML is often transported via HTTP just like ordinary HTML Web pages, but XML is also easily transported via other Internet protocols, such as FTP and SMTP.

Note: In order to use SOAP/XML channels in GenBroker, you must first install Microsoft Internet Information Server (IIS) on the GenBroker Server node. The SOAP/XML server can be running only on Windows NT-based systems (i.e. Windows NT 4.0, Windows 2000, or Windows XP). SOAP/XML support files are included in the ProcessView installation. GenBroker over SOAP/XML uses a SOAP/XML ISAPI listener that uses a WSDL (Web Services Description Language) file.

For more information about SOAP, go to http://msdn.microsoft.com/Soap.

Using an Indirect Channel Via a Mediator Node

In an indirect channel, the client applications do not communicate directly with the final node, but instead communicate over the Internet with a **mediator node.** As shown below, the mediator node communicates with the final node on behalf of the client. The mediator can communicate with OPC servers over the intranet or LAN. The client applications cannot have any direct connection with the node where the GenBroker utility they use is located. You can create and delete indirect channels using the GenBroker Configurator. The GenBroker Configurator allows you to specify a mediator node for the channel. If you want to route all requests for nonspecified nodes through this mediator, then set the channel as default.

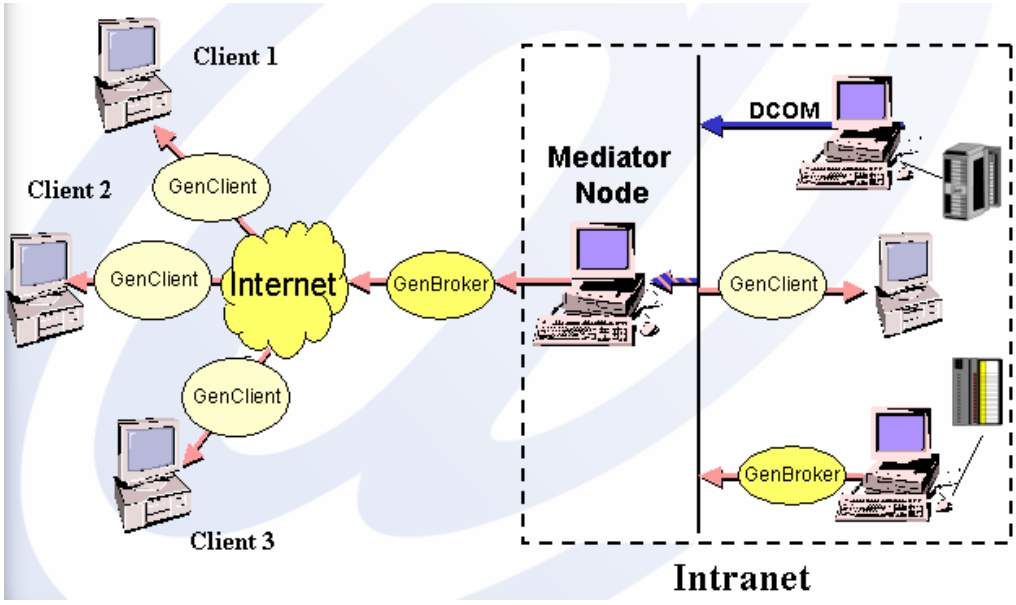


Figure 4. OPC Indirect Channel Via a Mediator Node

Configuring GenBroker

The GenBroker Configurator is a tool that enables you to customize your network architecture. The GenBroker Configurator allows you to:

- 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 (.gbc or .gbx) file. You can store multiple configuration files. Configuration files tell GenBroker Server how to communicate when it interacts with various clients. The .gbx configuration file is the preferred format for version 7.0. The .gbc format only contains primary nodes of security and licensing servers, while .gbx contains the primary and backup nodes for all kinds of Smar servers. The .gbx file is stored in XML, while the .gbc is binary. The .gbc file is used for backward compatability with version 6.x. It is recommended that you use the .gbx format in version 7.0.

Note
GenBroker has already been configured prior to installation. Configure (or reconfigure) GenBroker only if you wish, for example, to change servers**.**

Note
GenBroker is configured for both the client side and the server side. For WebHMI, the GenBroker configuration (.gbc or .gbx) file used by the client resides on the server side. Once the client connects to the server, the file is downloaded from the server. ProcessView clients have their own .gbc or .gbx file. Note that a GenBroker Server does not need to be a GenBroker client or a Web server.

Note.
For information about WebHMI client configuration, please see the WebHMI help documentation

Basic Steps in GenBroker Configuration

An example of a basic GenBroker configuration includes a server node connected to three PCs on a network that are connected to the Internet, as shown in the figure below.

PC1 - Server Node: This PC has WebHMI, ProcessView (with Security Server), and the ProcessView License Utility installed.

PC2 and PC3 - Client Nodes: These PCs have a Web browser, such as Microsoft Internet Explorer or Netscape Navigator, installed.

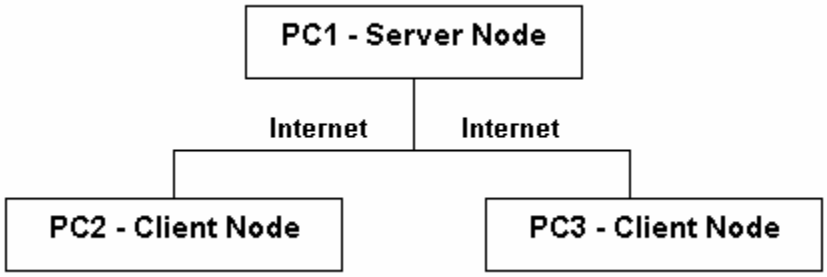


Figure 5. Basic Network Configuration

Basic Steps for Server Side Configuration

On the server node (PC1):

1. Create a display in GraphWorX.

2. Save the display and then publish the display to the Web server using the Web Publishing Wizard.

3. Configure GenBroker Server using the GenBroker Configurator, using the OPC over TCP/IP direct channel (to enable communication over the Internet) as the default channel. Save the GenBroker configuration (.gbc or .gbx) file.

4. Set GenBroker active on the server node using the ProcessView Tray in ProcessView.

Note.
You may also want to configure the Security Server on the server node using the Security Configurator. You need to specify the location of the Security Server and the License Server in the GenBroker Configurator.

Client Side Configuration: Viewing the Published Display File

Once GenBroker is running, you can view the GraphWorX display that you have published to the Web server from any client Web browser:

1. On the client machine (PC2 or PC3), open the Web browser (e.g. Microsoft Internet Explorer).

2. Enter the URL for the published GraphWorX display in the **Address** field of the Web browser window, and then click **Go.**

3. The Web page is downloaded from the Web server across the Internet and appears in the client's Web browser window.

GenBroker Server Configuration

To start the GenBroker Configurator:

1. From the Windows **Start** menu, select **Programs > Smar > ProcessView > Tools > GenBroker Configurator.**

2. This opens the **GenBroker Configurator** dialog box. Click on the **GenBroker Server** tab, shown below.

3. You can enable or disable the OPC communication channels (i.e. TCP/IP, SOAP/XML, or DCOM.

Note
It is recommended that you accept the default settings on the **GenBroker Server** tab. Change the settings only if you do not wish to use some of the interfaces or if you should wish to change them.

Note
In order for clients to communicate with the GenBroker Server, you must set GenBroker active on the server node using ProcessView Tray. For more information, please see "Starting GenBroker Server."

Clicking the **Default** button restores all default property installation settings.

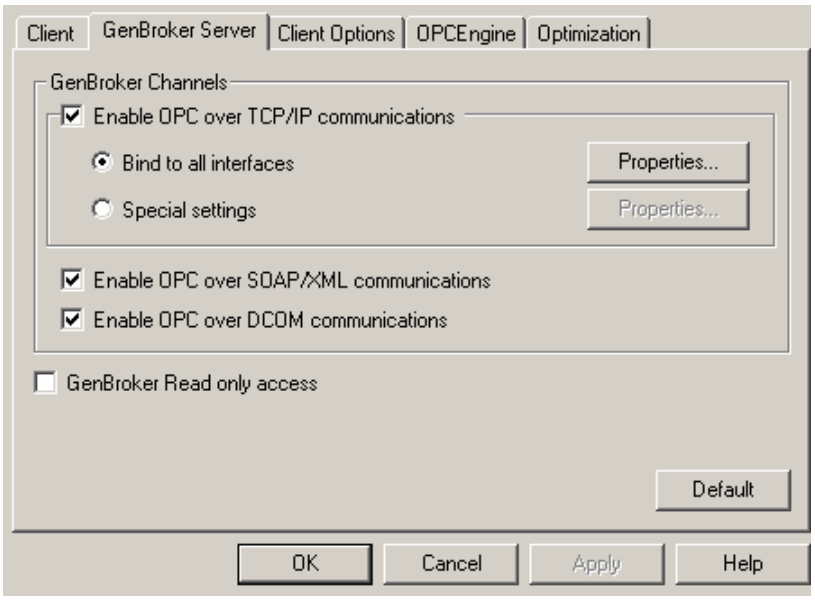


Figure 6. GenBroker Properties Dialog Box: GenBroker Server Tab

Enable OPC Over TCP/IP Communications

When **OPC Over TCP/IP Communications** is enabled on the **GenBroker Server** tab, the client applications communicate with GenBroker software located on remote nodes on an intranet or LAN as well as over the Internet. You have two options for configuring TCP/IP interfaces:

- Bind to All Interfaces
- Special Settings

Bind to All Interfaces

When **Bind to All Interfaces** is selected, GenBroker uses all available interfaces (i.e. network cards) on the local machine for communication. Clicking **Properties** on the **GenBroker Server** tab opens the **Interface Properties** dialog box, as shown in the figure below. The **IP Address** is set to **<All>,** and the interface **Port** number is 38080. The **No Delay** check box allows you to control a delay before a short packet of data is sent. **No Delay** is checked by default to ensure the smoothest data interface exchange between clients and the GenBroker Server.

Note
The **Port** number has changed to 38080. (In ProcessView Version 6.1, the port number was 8080.)

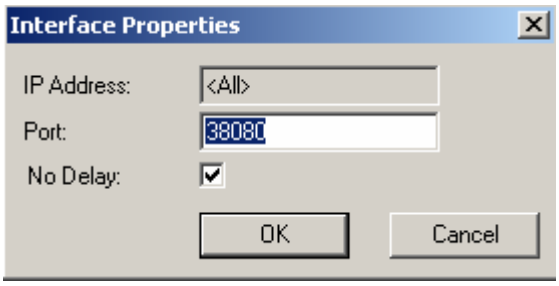


Figure 7. Properties For All Interfaces

Special Settings

When **Special Settings** is selected, you can specify which interfaces GenBroker will use for communication. For example, if you have multiple network cards on the local machine, you can select which network card to use. Clicking **Properties** on the **GenBroker Server** tab opens the **TCP/IP Interfaces** dialog box, as shown in the figure below. This dialog box lists all available interfaces on the local machine. Simply check the interfaces you want to use for GenBroker communications.

By default, the IP address for the local host "loop back" interface is 127.0.0.1. This interface is used to test TCP/IP communications locally.

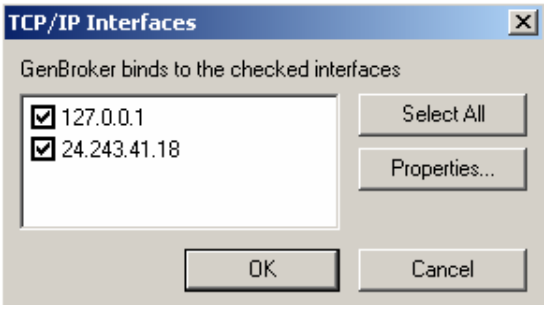


Figure 8. Properties For Specific Interfaces

To edit an interface's settings, highlight the interface and click **Properties.** This opens the **Interface Properties** dialog box for the selected interface, as shown in the figure below. The **IP Address** is specified, and the interface **Port** number is 38080. The **No Delay** check box allows you to control a delay before a short packet of data is sent. **No Delay** is checked by default to ensure the smoothest data interface exchange between clients and the GenBroker Server.

Note
The **Port** number has changed to 38080. (In ProcessView Version 6.1, the port number was 8080.)

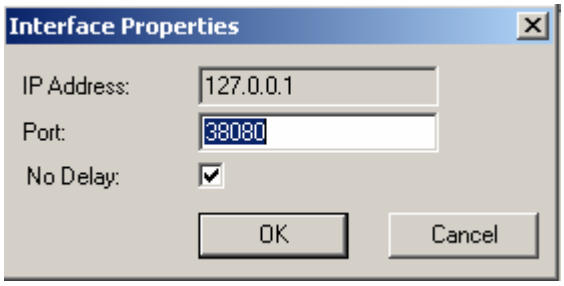


Figure 9. Properties For Selected Interface

Enable OPC Over SOAP/XML Communications

When **OPC Over SOAP/XML Communications** is enabled on the **GenBroker Server** tab, GenBroker responds to requests from clients that use the SOAP/XML protocol. SOAP uses XML to communicate over HTTP. The main advantage of using SOAP is that, because SOAP operates on the HTTP level (over TCP/IP), data can pass through a firewall without having to change the firewall settings. The disadvantage of SOAP is that it is slightly slower than TCP/IP. SOAP uses port 80 by default.

Enable OPC Over DCOM Communications

When **OPC Over DCOM Communications** is enabled on the **GenBroker Server** tab, client applications communicate with GenBroker software located on remote nodes on an intranet or LAN. This method allows you to set a connection retry period and timeout settings. GenBroker then communicates (locally) with the target OPC servers on behalf of the client application.

GenBroker Read Only Access

When the **GenBroker Read Only Access** check box is checked on the **GenBroker Server** tab, clients that communicate with the GenBroker Server cannot perform write requests to an OPC Data Access server. For example, GraphWorX could read a process point's value but could not respond to the value. Alarm acknowledgement for OPC Alarm and Event servers and insertion of annotations into Historical Data Access servers can be also disabled with this option.

Starting GenBroker Server

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 set GenBroker active on the server node using ProcessView Tray:

1. Launch ProcessView Tray from the Windows **Start** menu by selecting **Programs > Smar > ProcessView > Tools > ProcessView Tray.**

2. When ProcessView Tray opens, the triangle icon appear in the Windows tool tray. Click on the triangle and select **GenBroker > Start** from the pop-up menu, as shown in the figure below. This activates the GenBroker Server.

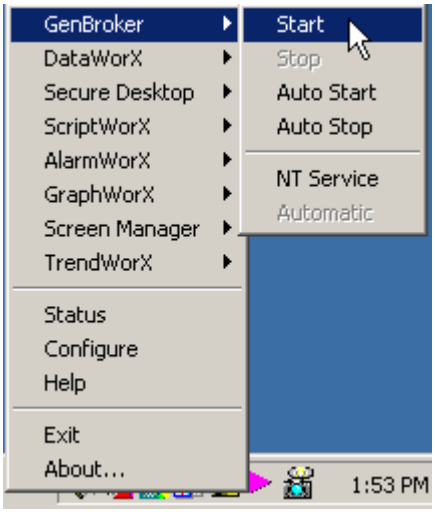


Figure 10. Starting the GenBroker Server on the Server Node

Note
ProcessView Tray 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 ProcessView Tray. Instead of running GenBroker from ProcessView Tray, 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.

GenBroker Client Configuration

In the **Client** tab of the **GenBroker Configurator** dialog box, shown in the figure below, set GenBroker active for the client. You have two general options for configuring communications:

- OPC Communications
- GenBroker Communications

Clicking the **Default** button restores all default property installation settings.

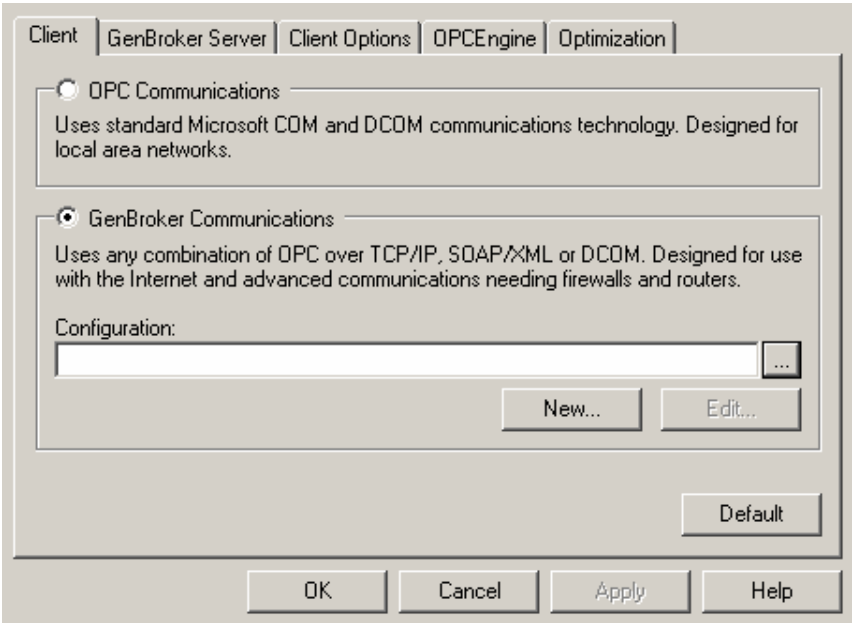


Figure 11. GenBroker Properties Dialog Box: Client Tab

OPC Communications

OPC Communications uses standard COM and DCOM technology and is standard with past versions of ProcessView. OPC communication does not require GenBroker on the server side. Client applications talk directly to the OPC servers using the standard OPC interface. All options like Security Server and Licensing Server locations are taken from the system registry. It is designed for local area networks with DCOM.

GenBroker Communications

GenBroker Communications may use TCP/IP, SOAP/XML or DCOM channels as well as direct OPC communication. Any combination of the above is also allowed. Since GenBroker needs some information about the network (it needs to know how to communicate with particular nodes), you should provide a GenBroker configuration file, which can be created with the **New** button and edited with the **Edit** button. The GenBroker configuration file can be stored in the binary form (compatible with ProcessView version 6.x) in a .gbc file, or it can be stored in an XML format in a .gbx file (new to version 7.0).

Note
If no configuration file is specified, OPC Direct communication is used for all nodes. In this case, GenBroker communication behaves the same way (for OPC Data Access, Security, and Licensing, applications using other subsystems may work a little bit differently) as OPC communication, except it does not pick up the servers' location from the system registry (as mentioned above). Instead it assumes the servers are local.

Client Options

The **Client Options** tab of the **GenBroker Configurator** dialog box, shown in the figure below, sets the **Tag Extension Syntax, Tag Retry Rate,** and **Sync Call Time Out** parameters for the client.

Clicking the **Default** button restores all default property installation settings.

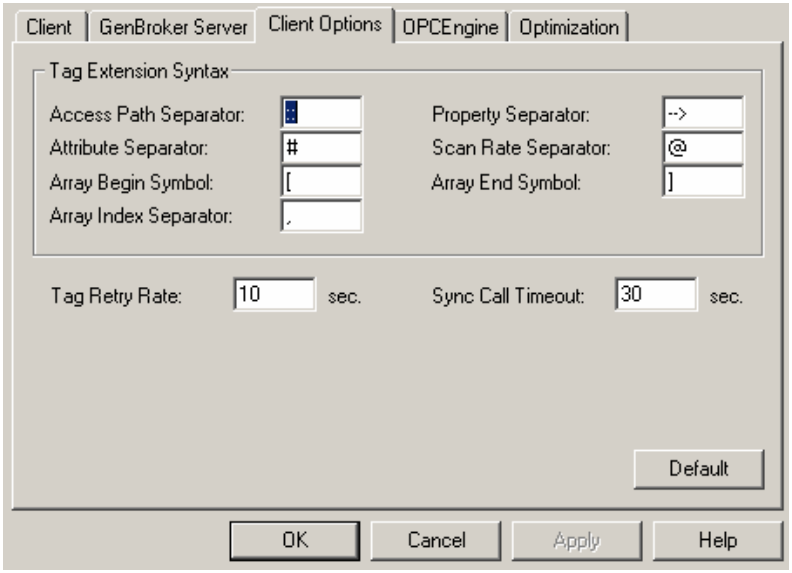


Figure 12. GenBroker Properties Dialog Box: Client Options Tab

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.

Tag Extension Syntax

The controls under the **Tag Extension Syntax** field of the **Client Options** tab can be used to redefine default separators for OPC item names. The section below explains the Tag Extension Syntax using the default separators.

The **Tag Extension Syntax** is a powerful feature of ProcessView. According to the OPC Specification, an OPC item or tag on a specific server is identified by an ItemID. ItemIDs can contain any characters.

The standard syntax for a tag in ProcessView client applications is:

[\\<node>\]<server>\<itemid>

where <*node*> is a name or IP address of a node in the network. The node name is optional. When omitted, the local node is used.

<server> is a class ID (CLSID) or ProgID of an OPC DA Server.

<itemid> is an OPC ItemID; this may contain any characters.

Example: \\PC1\Smar.Simulator\SimulatePLC.Sine

The **Tag Extension Syntax** allows you to specify tags more precisely, telling GenBroker to parse the value incoming from the server.

To use the extension syntax, type **tag:** before the tag name. Then the separators defined in this property page may be used. You may modify the separators if they conflict with the ItemIDs of the used OPC Server.

Note: The **Access Path Separator (::)** and **Property Separator (-->)** could also be used with the previous versions of ProcessView, but they might conflict with some servers' item IDs. Now these are part of the Tag Extension Syntax, and you have to explicitly use tag: to use them.

The **Tag Extension Syntax** is:

tag:[<*node*>***\]***<*server*>******<*itemid*>***[::***<*accpath*>*][- ->*<*prop*>*][*<*idx*>*][#*<*attr*>*][@*<*rate*>*]***

<*node*>, <*server*> and <*itemid*> is the same as in the standard syntax.

<accpath> is an optional access path string. This can be used (according to the OPC Specification) to give the server a "hint" how to get the OPC item specified by its <*itemid*>.

<prop> is a property ID as defined in the OPC 2.0 Specification. When specified, GenBroker uses the server's function *IOPCItemProperties::LookupItemIDs()* to get the property value. Note that not all OPC 2.0 compliant servers support this function, so the usage is limited. Besides specifying the property ID, it is possible to specify a property name listed below.

<idx> is an index or indexes (separated by commas) for an array. If the specified OPC item is of an array type, this allows for accessing one of its elements. The element is read-only. The indexes should be enclosed in brackets ([and]) and separated by commas (,).

<attr> specifies an attribute that should be returned instead of the item value. In OPC, values always come with a quality value and timestamp. The attributes defined here are derived from these values. The supported are attributes listed below. Unlike properties (see <*prop*> above), this option works with any OPC server, even OPC 1.0. The attributes are read-only.

<rate> specifies a scan rate for the given item. Usually the scan rate is defined for a group of items, eg. in GraphWorX it is defined for the whole display. This option can override the display scan rate for one particular item. The scan rate is in milliseconds.

Options in the tag name can be combined, but they must be used in the order indicated above.

Property Names Defined by GenBroker

Property names supported by GenBroker are listed in the table below. Their meaning is defined in the OPC Specification 2.0. The property names are equal to their IDs.

Property Name	ID	Property Name	ID	Property Name	ID
DataType	1	OpenLabel	107	AlPrimArea	303
Value	2	TimeZone	108	CondLogic	304
Quality	3	DefDisplay	200	LimExceeded	305
Timestamp	4	FgColor	201	Deadband	306
AccessRights	5	BkColor	202	HiHiLimit	307
ScanRate	6	Blink	203	HiLimit	308
EUUnits	100	BMPFile	204	LoLimit	309

Property Name	ID	Property Name	ID	Property Name	ID
Description	101	SoundFile	205	LoLoLimit	310
HighEU	102	HTMLFile	206	ROCLimit	311
LowEU	103	AVIFile	207	DevLimit	312
HighIR	104	CondStatus	300	ItemID	5000
LowIR	105	AlQuickHelp	301	Name	5001
CloseLabel	106	AlAreaList	302	UsageCnt	5002

Attributes Defined by GenBroker

When any of the following attributes is defined, GenBroker returns the value of the attribute rather than the value of the specified item. Possible attributes are listed in the table below.

Attribute	Data type	Meaning
quality	INTEGER	Quality of the item – see OPC Specification for more details.
valid	BOOLEAN	Flag if the item value is valid. True for GOOD and UNCERTAIN qualities and for BAD quality with LAST KNOWN VALUE substatus.
timestamp	DATE/TIME	Time of the last change of the item value.
qualgood	BOOLEAN	True when the item value has GOOD quality, False otherwise.

Tag Extension Syntax Examples

The following examples show how the Tag Extension Syntax is used.

\\PC1\Smar.Simulator\SimulatePLC.Sine::Path1

Requests tag ”SimulatePLC.Sine::Path1“ on Smar.Simulator located on PC1.

tag:\\PC1\Smar.Simulator\SimulatePLC.Sine::Path1

Requests tag ”SimulatePLC.Sine“ on Smar.Simulator located on PC1. Uses access “Path1.“ Note that it is necessary to explicitly switch on the OPC Item Extension Syntax by placing **tag:** at the beginning of the string.

tag:\\PC1\Smar.Simulator\SimulatePLC.Sine-->Description

tag:\\PC1\Smar.Simulator\SimulatePLC.Sine-->101

Both request the Description property of tag ”SimulatePLC.Sine“ on Smar.Simulator located on PC1. This particular example would not work, because Smar.Simulator does not support the IOPCItemProperties::LookupItemIDs() function necessary for proper function. OPC Servers based on OPC ToolWorX 3.0 do support this function.

tag:\\PC1\Smar.Simulator\SimulatePLC.SineArray[3]

Requests tag ”SimulatePLC.SineArray“ on Smar.Simulator located on PC1. This tag is supposed to be an array. GenBroker then extracts the third element of the array.

tag:\\PC1\Smar.Simulator\SimulatePLC.Sine#qualgood

Requests tag "SimulatePLC.Sine" on Smar.Simulator located on PC1. GenBroker then uses its quality and tests if it is good. If so, GenBroker returns TRUE, otherwise it returns FALSE.

tag:\\PC1\Smar.Simulator\SimulatePLC.Sine#qualgood@1000

Same as before, but requests the tag "SimulatePLC.Sine" on the scan rate of 1000 milliseconds, not regarding the display setting (in GraphWorX).

tag:\\PC1\Smar.Simulator\SimulatePLC.Sine@1000#qualgood

This would not work; the options must be ordered properly.

OPC Engine

The **OPC Engine** tab of the **GenBroker Configurator** dialog box, shown in the figure below, sets the OPC server connection control parameters.

Clicking the **Default** button restores all default property installation settings.

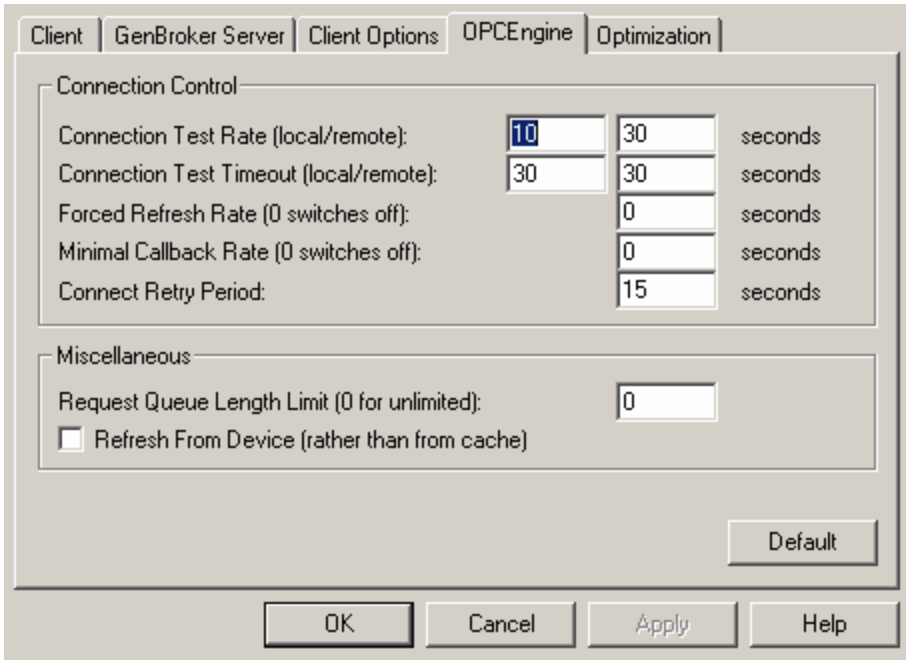


Figure 13. GenBroker Properties Dialog Box: OPC Engine Tab

Connection Control Parameters

The **Connection Control** field defines 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.

- **Forced Refresh Rate:** Specifies if and how often (in seconds) GenBroker should force OPC servers to refresh the data values.
- **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.

- **Connect Retry Period:** When GenBroker cannot connect to a server, it will retry after the time specified in **Connect Retry Period.**

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 released from the OPC server again. However, if **Refresh From Device (rather than from cache)** is checked, the item is retrieved from the i/o device to which the OPC server is connected.

Optimization

The **Optimization** tab of the **GenBroker Configurator** dialog box, shown in the figure below, sets the server-client connection parameters.

Clicking the **Default** button restores all default property installation settings.

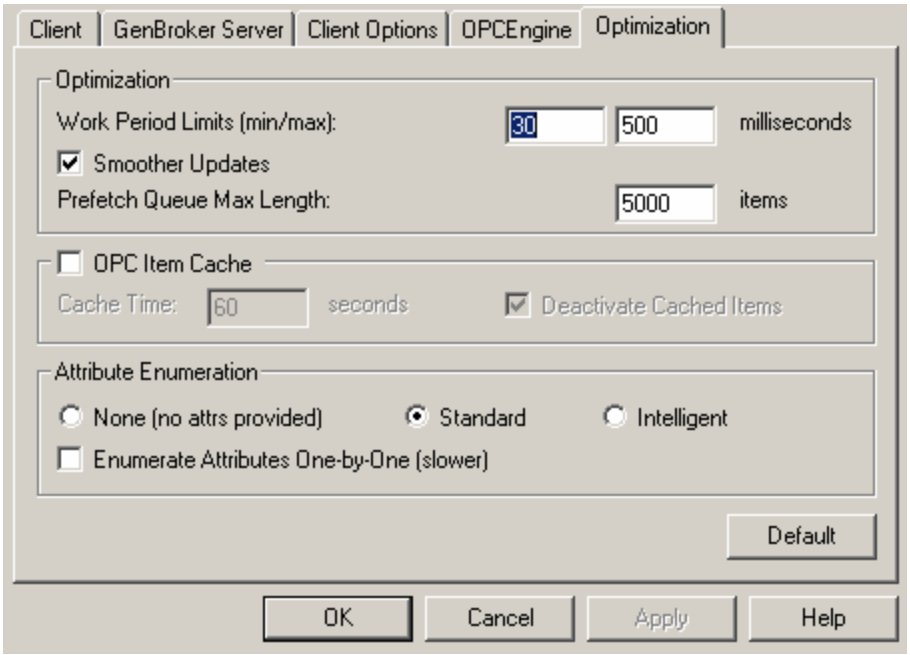


Figure 14. GenBroker Properties Dialog Box: Optimization Tab

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 **Smoother Updates** check box. **Prefetch Queue Max Length** limits the number of updates prepared in advance.

OPC Item Cache

GenBroker now 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 released 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 Time** defines how long (in seconds) the items should be kept in the cache. When **Deactivate Cached Items** is checked, the items are deactivated when put into 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 Enumeration:** 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 auxillary group. Standard enumeration does not have this overhead, but it enumerates also 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.

Creating a GenBroker Configuration File

To create a new GenBroker configuration file, select **GenBroker Communications** on the **Client** tab of the GenBroker Configurator, and then click the **New** button, as shown in the figure below.

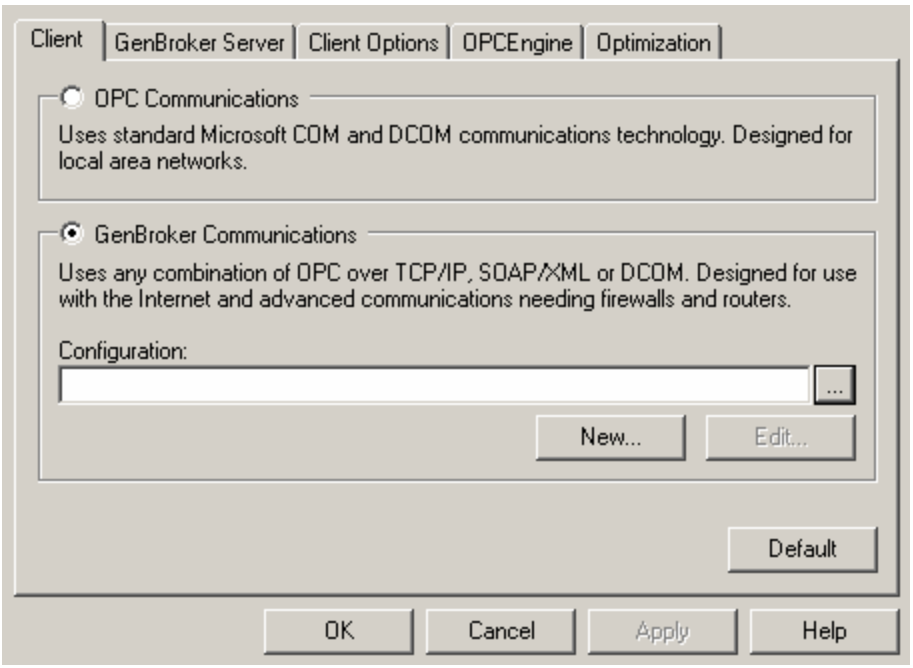


Figure 15. GenBroker Properties Dialog Box: Client Tab

The **GenBroker Configurator** screen appears, as shown in the figure below. Note that you can also search for an existing .gbc or .gbx file by clicking the **Browse** button on the **Client** tab. Clicking the **Edit** button on the **Client** tab allows you to modify an existing configuration file.

As shown in the figure below, the GenBroker Configurator contains four predefined sets of channels that cannot be deleted:

- OPC direct
- OPC over TCP/IP direct
- OPC over SOAP/XML direct
- OPC over DCOM direct

However, you can add, remove, or reconfigure nodes and channels using the commands and functions described in the following sections.

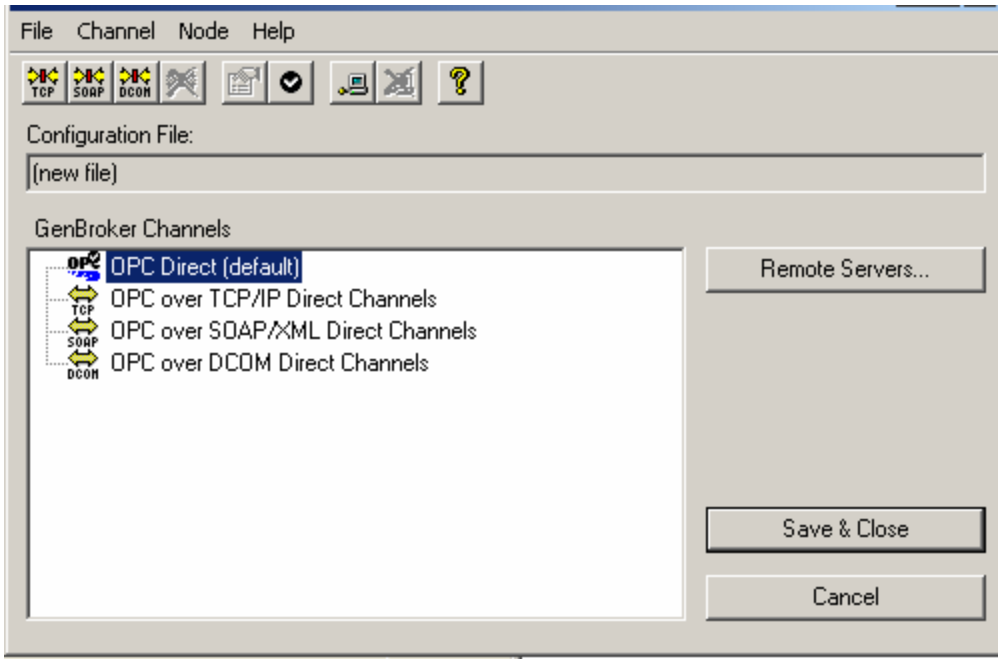


Figure 16. GenBroker Configurator Screen

Toolbar

The GenBroker Configurator toolbar, shown below, contains the following command functions. For more information about these functions, please refer to the **Menus** section.

- **TCP/IP:** Adds a new TCP/IP indirect (mediator) channel.
- **SOAP/XML:** Adds a new SOAP/XML indirect (mediator) channel.
- **DCOM:** Adds a new DCOM indirect (mediator) channel.
- **Remove Channel:** Deletes the selected channel.
- **Properties:** Shows the properties of the selected channel.
- **Default:** Sets the selected channel as the default channel.
- **Add Node:** Adds a node to the selected channel.
- **Remove Node:** Removes a node from the selected channel.
- **About:** Opens the About Box, which contains information about the application.

Figure 17. GenBroker Configurator Toolbar

Menus

The GenBroker Configurator contains the following menus:

- **File**
- **Channel**
- **Node**
- **Help**

File Menu

The **File** menu contains the following commands:

Command	Function
Save As	Opens the **Save As** dialog box, which allows you to save the GenBroker Configuration (.gbc or .gbx) file.
Save and Close	Saves the GenBroker Configuration (.gbc or .gbx) file and then exits the GenBroker Configurator.
Exit Without Saving	Closes the GenBroker Configurator.

Channel Menu

The **Channel** menu contains the following commands:

Command	Function
Add TCP/IP Channel	Adds a new TCP/IP indirect (mediator) channel to the configuration. This opens the **Select a Node** dialog box, which allows you to specify a mediator node for the newly created channel.
Add SOAP/XML Channel	Adds a new SOAP/XML indirect (mediator) channel to the configuration. This opens the **Select a Node** dialog box, which allows you to specify a mediator node for the newly created channel.
Add DCOM Channel	Adds a new DCOM indirect (mediator) channel to the configuration. This opens the **Select a Node** dialog box, which allows you to specify a mediator node for the newly created channel.
Set As Default	Sets the selected channel as the default channel.
Properties	Displays the configuration settings for the selected channel.
Remove	Deletes the selected channel from the configuration.

Node Menu

The **Node** menu contains the following commands:

Command	Function
Add	Opens the **Select a Node** dialog box, which allows you to add a new node that will be accessed via the selected channel.
Remove	Removes the selected node from a channel.

Help Menu

The **Help** menu contains the following commands:

Command	Function
Help Topics	Opens the help documentation associated with this program.
About Application	Opens the About Box, which provides the version number and copyright information for this software.

Assigning Nodes to Channels

In order for GenBroker to know which channel should be used for a particular node, you must assign nodes to the channel. Nodes that are not explicitly defined in the configuration will use the default channel. You may set up properties for every channel, depending on the communication protocol used. The following sections describe how to assign nodes to channels and how to change the channels' properties.

Note
These instructions apply ONLY when you have selected **GenBroker Communications** from the **Client** tab of the **GenBroker Configurator** dialog box.

Choosing the Default Channel

Before assigning nodes to channels, choose your default channel:

1. Right-click the communication channel (for example, **OPC over TCP/IP Direct Channels),** and select **Set As Default** from the pop-up menu, as shown in the figure below.

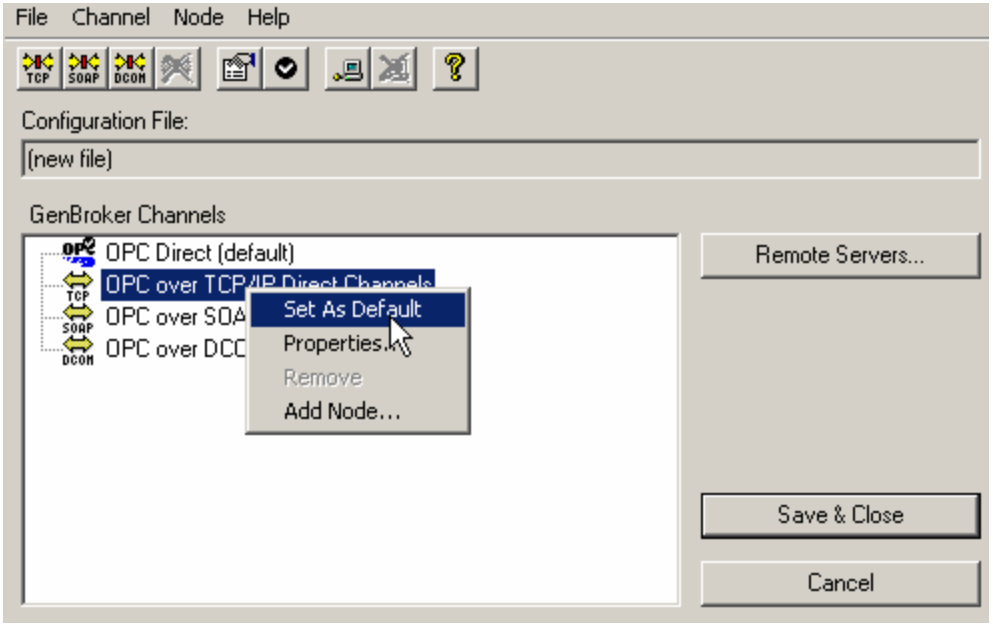


Figure 18. Setting the Default Channel

2. The new default channel is indicated in the display by "(default)," as shown in the figure below.

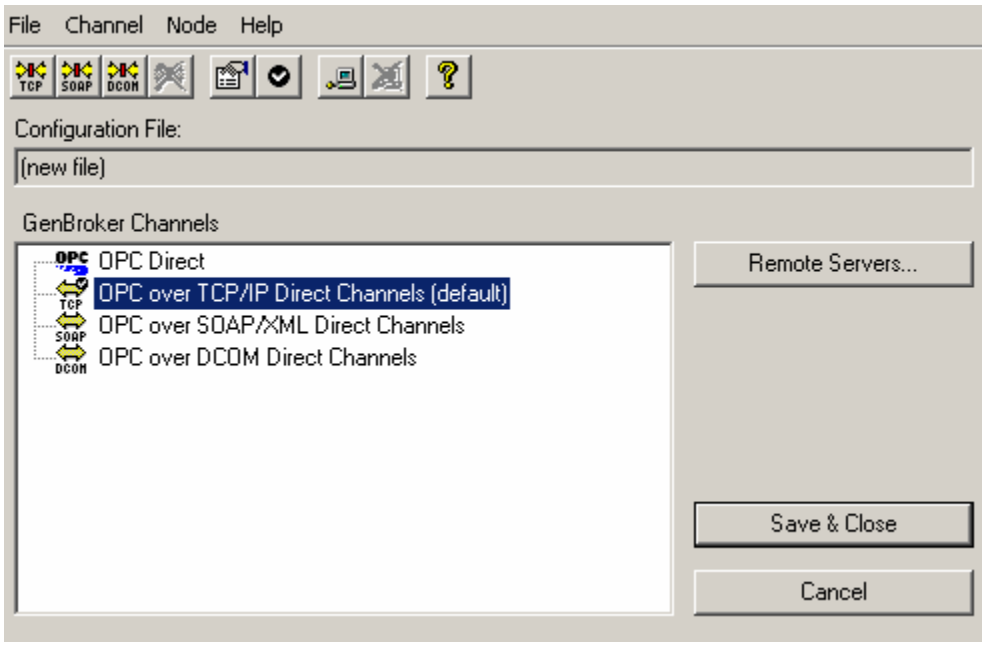


Figure 19. Default Channel Changed

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.

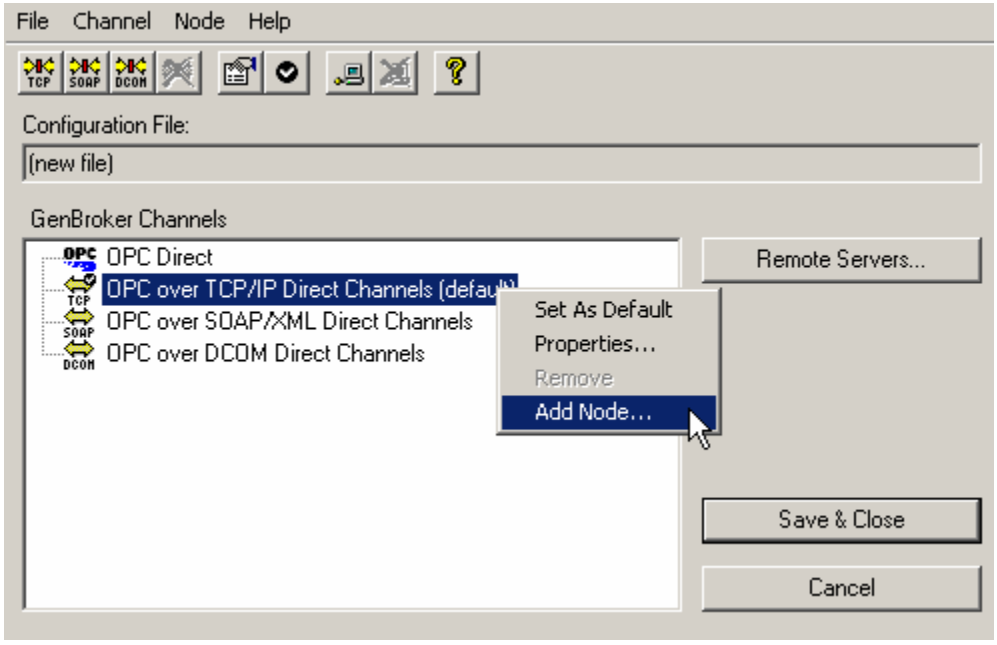


Figure 20. Adding a Node to a Channel

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.

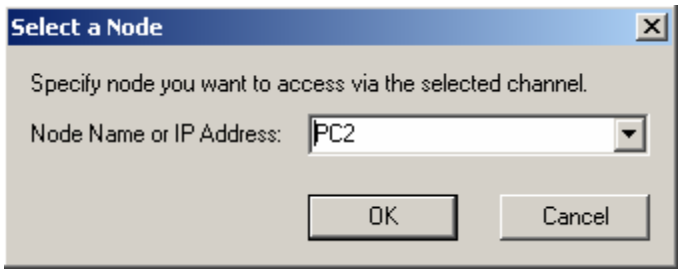


Figure 21. Specifying a Node to Access Via the Selected Communication Channel

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

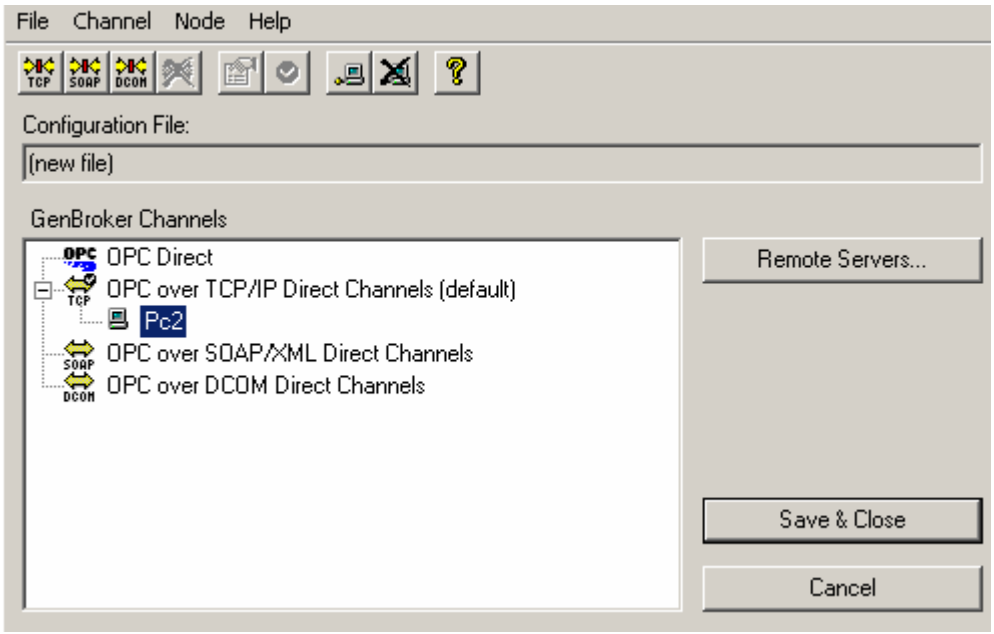


Figure 22. Node Name Added to Communication Channel

Removing Nodes From Channels

To remove a node from a channel, right-click the node that you wish to remove, and then select **Remove** from the pop-up menu, as shown in the figure below.

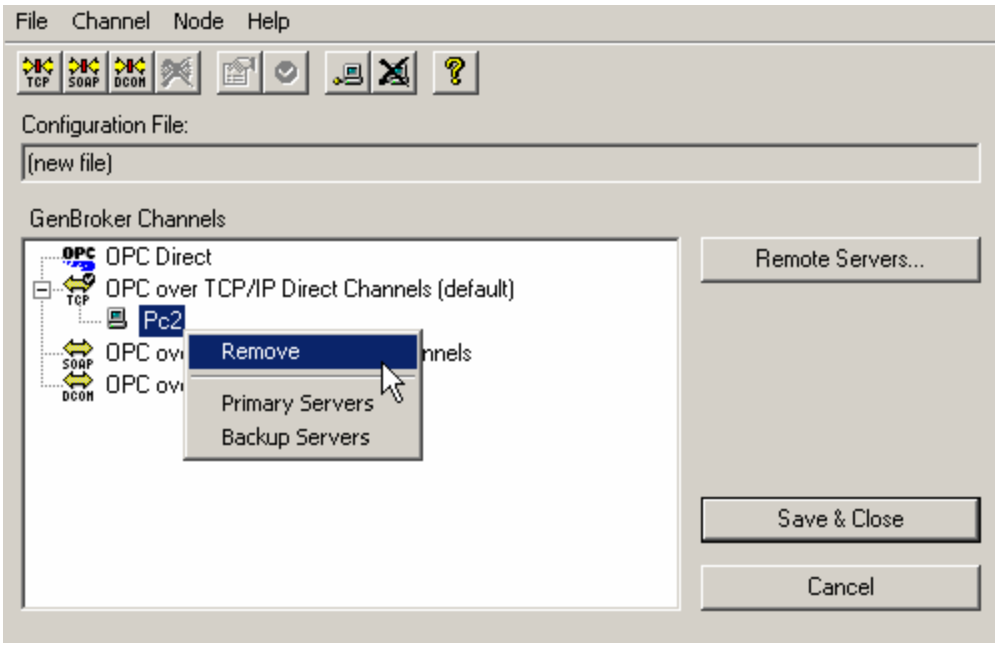


Figure 23. Removing a Node From a Channel

Channel Properties

To view the properties for a channel, right-click the channel and select **Properties** from the pop-up menu, as shown in the figure below. This opens the **Channel Properties** dialog box. The **Channel Properties** dialog box, which varies slightly for DCOM, TCP/IP, and SOAP/XML, has several parameters that you can configure for the selected communication method. For example, you can define which port is used for the TCP/IP traffic. If a mediator node has been selected, the mediator node name is also displayed.

You can also configure the settings for retrieving data from the OPC server. In case the connection between the client and the server fails, you can set the connection retry period, which is the time between successive connection trials. You can also set the polling rate, which is the time during which GenClient periodically flushes internal request queues. In addition, you can set the timeout rate, which is the time limit for establishing a connection with the server. If the connection fails, the client will wait for the connection retry period to elapse before trying to reconnect to the server.

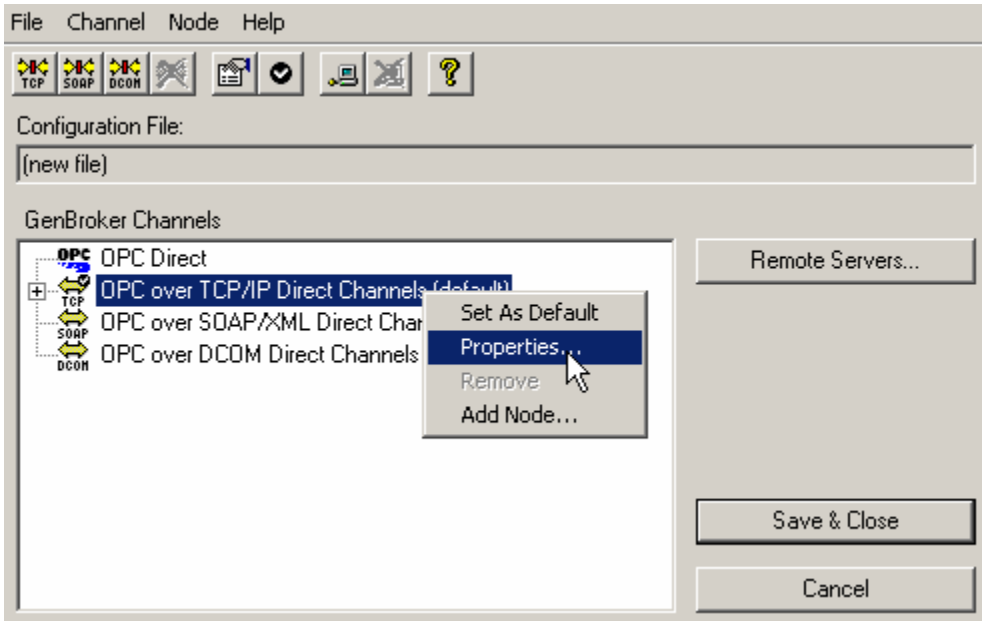


Figure 24. Viewing Channel Properties

DCOM Channel Properties

Right-clicking on a DCOM channel and selecting **Properties** opens the **DCOM Channel Properties** dialog box, shown below.

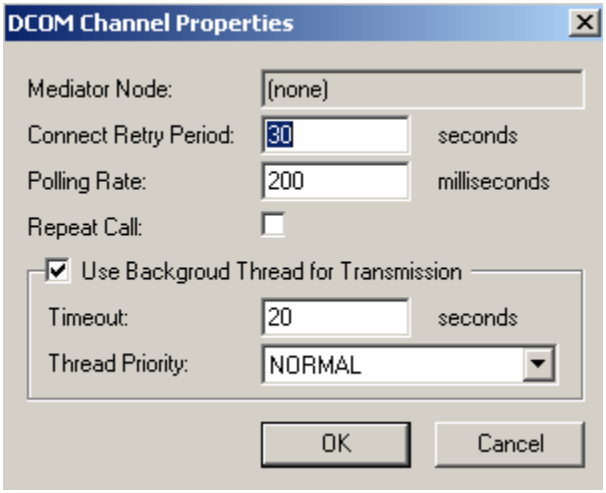


Figure 25. DCOM Channel Properties Dialog Box

The **DCOM Channel Properties** dialog box configures the following settings for the DCOM channel:

- **Mediator Node:** If a mediator node has been selected, the node name will be displayed in this field.
- **Connect Retry Period:** In case the connection between the GenClient and the remote GenBroker server fails, you can set the connection retry period, which is the time (in seconds) between successive connection trials.
- **Polling Rate:** Number of milliseconds during which GenClient periodically flushes internal request queues.
- **Repeat Call:** When this box is checked, the client keeps requesting data from the server until all data from the server have been sent to the client. Subsequent calls are made at the specified **Polling Rate** interval.
- **Use Background Thread for Transmission:** When this option is checked, GenClient uses another thread to transmit data.
- **Timeout:** The maximum time (in seconds) that the client waits for a response from the server.
- **Thread Priority:** Performance-tuning parameter. Select from Lowest, Below Normal, Normal, Above Normal, and Highest.

TCP/IP Channel Properties

Right-clicking on a TCP/IP channel and selecting **Properties** opens the **TCP/IP Channel Properties** dialog box, shown below.

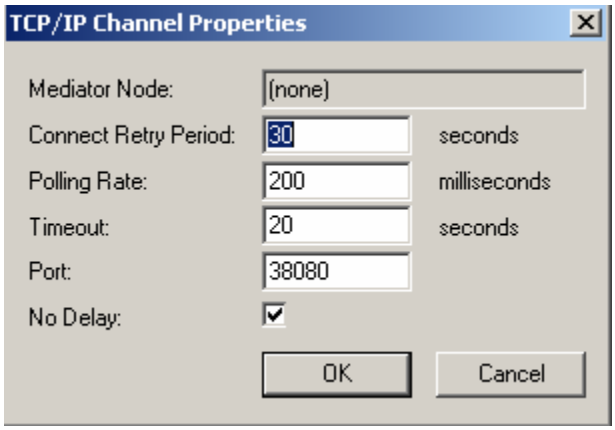


Figure 26. TCP/IP Channel Properties Dialog Box

The **TCP/IP Channel Properties** dialog box configures the following settings for the TCP/IP channel:

- **Mediator Node:** If a mediator node has been selected, the node name is displayed in this field.
- **Connect Retry Period:** In case the connection between the GenClient and the remote GenBroker server fails, you can set the connection retry period, which is the time (in seconds) between successive connection trials.
- **Polling Rate:** Number of milliseconds during which GenClient periodically flushes internal request queues.
- **Timeout:** The maximum time (in seconds) that the client will wait for a response from the server.
- **Port:** The TCP/IP port number that GenClient tries in order to establish communication with the GenBroker server.

Note: The **Port** number has changed to 38080. (In ProcessView Version 6.1, the port number was 8080.)

- The **No Delay** check box allows you to control a delay before a short packet of data is sent. **No Delay** is checked by default to ensure the smoothest data interface exchange between clients and the GenBroker Server.

SOAP/XML Channel Properties

Right-clicking on a SOAP/XML channel and selecting **Properties** opens the **SOAP Channel Properties** dialog box, shown below.

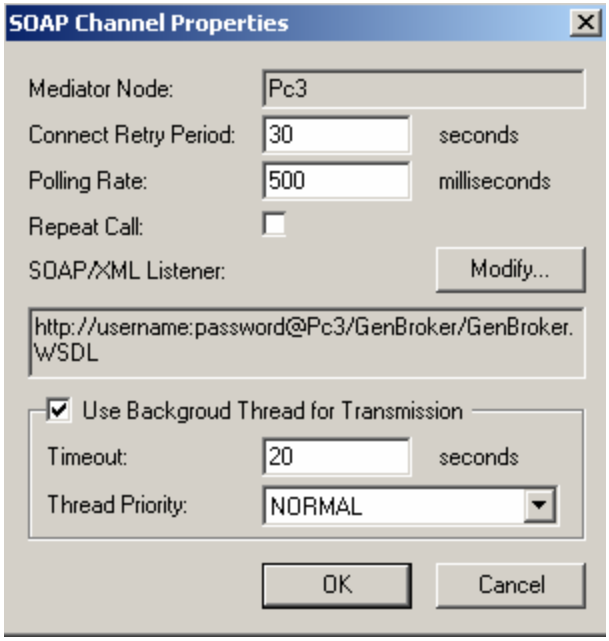


Figure 27. SOAP Channel Properties Dialog Box

The **SOAP Channel Properties** dialog box configures the following settings for the TCP/IP channel:

- **Mediator Node:** You can create a special SOAP channel with a mediator node through which GenBroker requests will be routed to a group of nodes. If a mediator node has been selected, the node name is displayed in this field.
- **Connect Retry Period:** In case the connection between the GenClient and the remote GenBroker server fails, you can set the connection retry period, which is the time (in seconds) between successive connection trials.
- **Polling Rate:** Number of milliseconds during which GenClient periodically flushes internal request queues.
- **Repeat Call:** When this box is checked, the client keeps requesting data from the server until all data from the server have been sent to the client. Subsequent calls are made at the specified **Polling Rate** interval.
- **Use Background Thread for Transmission:** When this option is checked, GenClient uses another thread to transmit data.
- **Timeout:** The maximum time (in seconds) that the client waits for a response from the server.
- **Thread Priority:** Performance-tuning parameter. Select from Lowest, Below Normal, Normal, Above Normal, and Highest.
- **Timeout:** The maximum time (in seconds) that the client will wait for a response from the server.

The SOAP/XML channel allows you to send data using the SOAP communication protocol. The channel's properties are similar to the TCP/IP and DCOM channel properties. The primary difference is that you must specify the listener for the SOAP/XML channel in the URL address to which you want to connect. Any SOAP server application must have a **listener.**

Note
In order to use SOAP/XML channels in GenBroker, you must first install Microsoft Internet Information Server (IIS) on the GenBroker Server node. The SOAP/XML server can be running only on Windows NT-based systems (i.e. Windows NT 4.0, Windows 2000, or Windows XP). SOAP/XML support files are included in the ProcessView installation. GenBroker over SOAP/XML uses a SOAP/XML ISAPI listener that uses a WSDL (Web Services Description Language) file.

The URL address for the SOAP/XML channel is specified in the **SOAP/XML Listener** field. Clicking the **Modify** button on the **SOAP Channel Properties** dialog box opens the URL editor, as shown in the figure below.

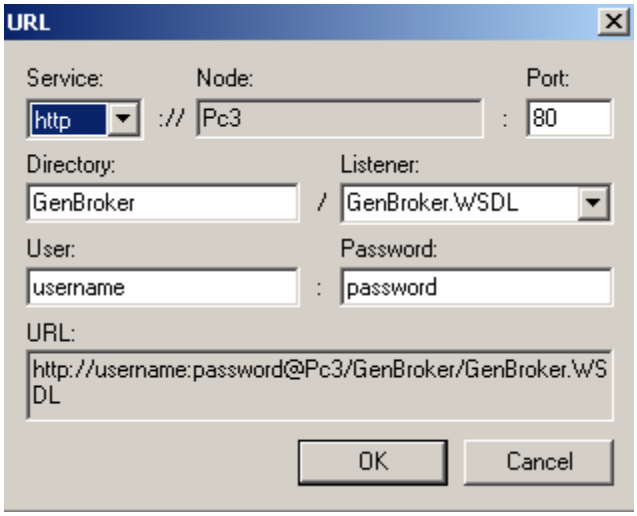


Figure 28. SOAP URL Editor

Suppose that the *Mediator Node, Virtual Directory* and *Listener* determine a whole URL for the listener. The whole listener address is then:

http://<*Mediator Node*>/<*Virtual directory*>/<*Listener*>

This editor allows you to change the URL path components while maintaining the proper syntax for the URL address (e.g. http://soapserver.company.com/GenBroker/GenBroker.WSDL for the ISAPI listener or http://soapserver.company.com/GenBroker/GenBroker.asp for the ASP listener) of the SOAP/XML channel:

- **Service:** Specifies either HTTP or HTTPS (secure). Select a protocol type from the drop-down list.
- **Node:** If a mediator node has been selected, the node name is displayed in this field.
- **Port:** SOAP uses port 80 by default.
- **Directory:** Specifies the virtual directory on the mediator node through which the SOAP/XML channel is routed.
- **Listener:** Select a protocol type from the drop-down list (e.g. GenBroker.WSDL or GenBroker.asp).
- **URL:** Displays the full URL path with proper syntax.

For more information about SOAP, go to http://msdn.microsoft.com/Soap.

Remote Servers

You can use the GenBroker Configurator to assign locations (node names or IP addresses) to the following primary and redundant (backup) servers:

- Security Server
- License Server
- Global Aliasing Server
- Language Aliasing Server
- GenEvent Server

Assigning a Location to Remote Servers

You can assign a node name or IP address to a remote server. Your system will then know where to get information about security and licensing, as well as language aliasing and global aliasing.

1. In the GenBroker Configurator, click the **Remote Servers** button, as shown in the figure below.

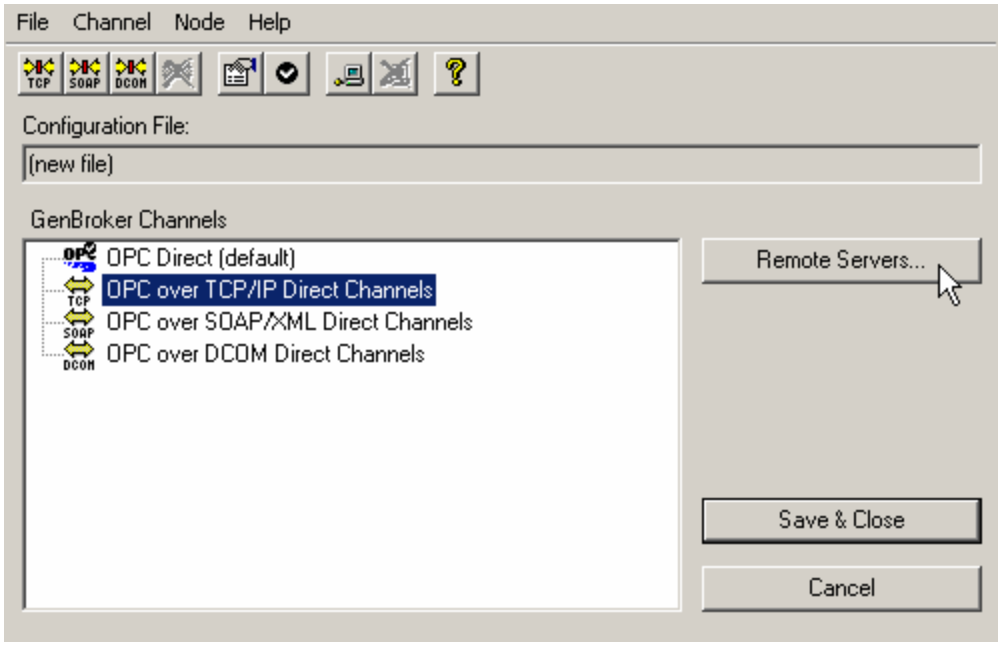


Figure 29. Assigning a Location to Remote Servers

2. This opens the **Remote Server List** dialog box, shown below. This dialog lists the server types and the location of the primary and backup servers for each type. Clicking the **Default** button assigns the Primary server node to the local machine. You can also specify the server locations from the registry information by clicking the **From Registry** button. Click OK.

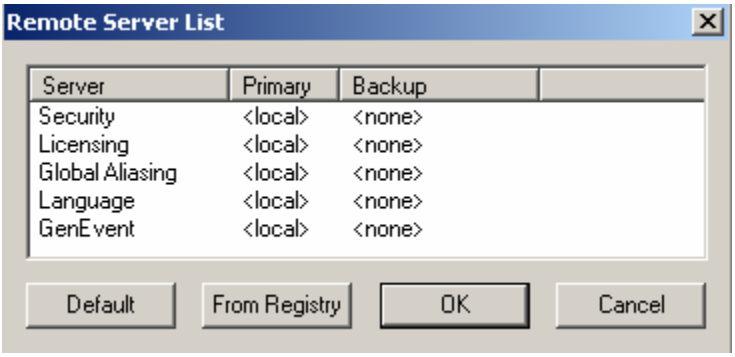


Figure 30. Remote Server List

3. Right-click a node name under a communication channel and select **Primary Servers** or **Backup Servers** from the pop-up menu, as shown below. This assigns the node to the primary or backup servers, respectively.

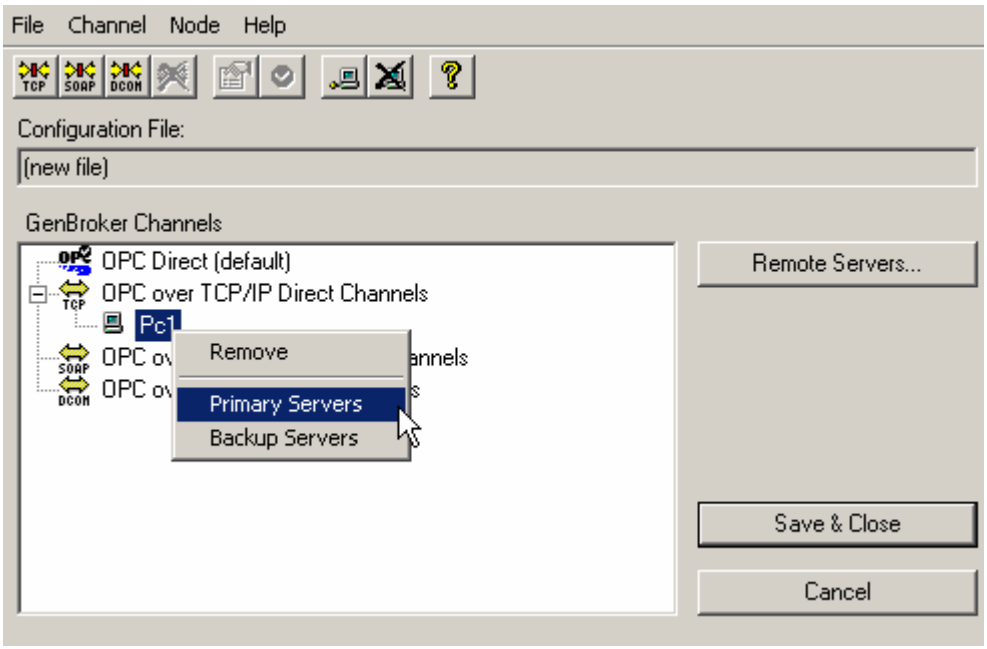


Figure 31. Assigning a Node to a Remote Server

4. Click the **Remote Servers** button. The assigned node name or IP address appears in the **Remote Server List** field, as shown in the figure below.

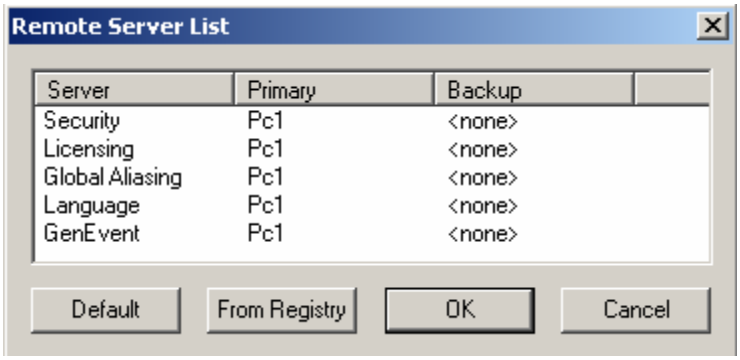


Figure 32. Assigned Node Name in Remote Server List

Activating the GenBroker Configuration File

You can make the GenBroker configuration (.gbc or .gbx) file active by saving it:

1. Click the **Save & Close** button on the GenBroker Configurator, as shown in the figure below.

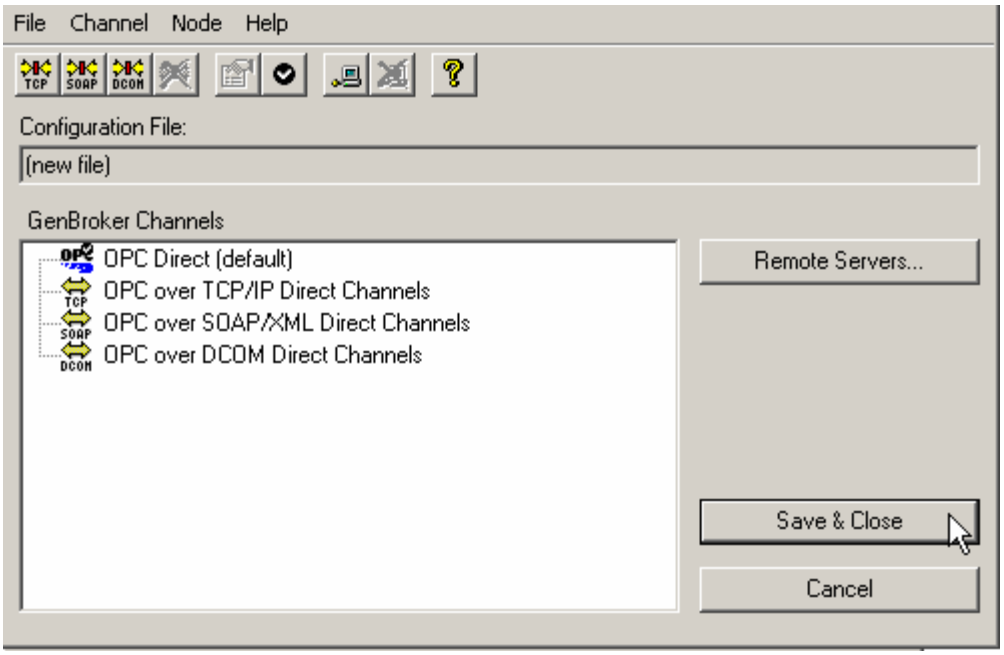


Figure 33. Activating the GenBroker Configuration File

2. This opens the **Save As** dialog box, as shown below. Browse to the location where you wish to save the file. Type the file name in the **File name** field. Click the **Save** button.

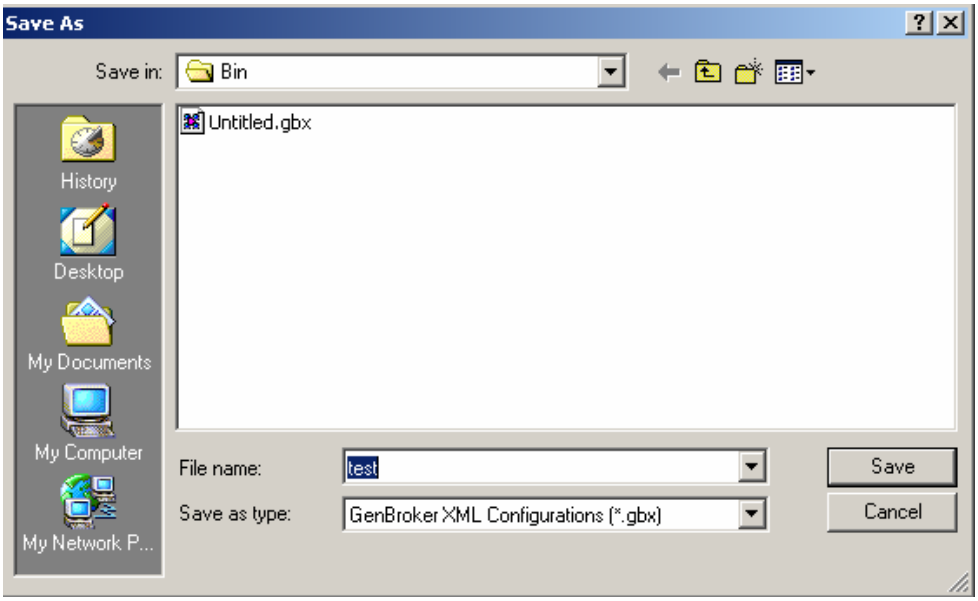


Figure 34. Saving the GenBroker Configuration File

3. The GenBroker Configurator **Client** tab appears, as shown below. The file name and pathway appear in the **Configuration** field. Click the **Apply** button to finish saving the GenBroker configuration file.

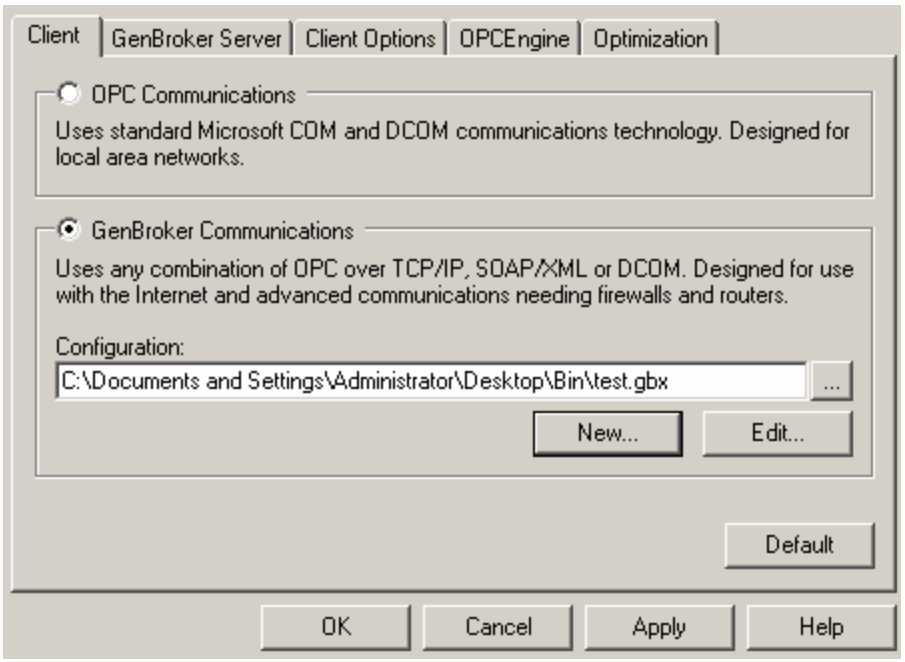


Figure 35. Configuration File Name Specified in the Client Tab

4. If you wish to edit the configuration file, click the **Edit** button, as shown below. After you click the **Edit** button, the **GenBroker Configurator** screen will reappear, and the file name will appear in the **Configuration** field. After you have edited the configuration file in the **GenBroker Configurator,** click the **Save & Close** button again. The **Client** tab will reappear. To close GenBroker, click **OK** on the **Client** tab.

Example Configurations

This section shows some example GenBroker configurations for the sample network setup shown below. The network includes five different PCs, or nodes. The example GenBroker configuration shown here is for PC1, which communicates with PC2, PC3, PC4, and PC5 according to the settings in the GenBroker Configurator. The following section demonstrates several different ways the GenBroker Configurator can be used to configure PC1 to communicate with the four other nodes.

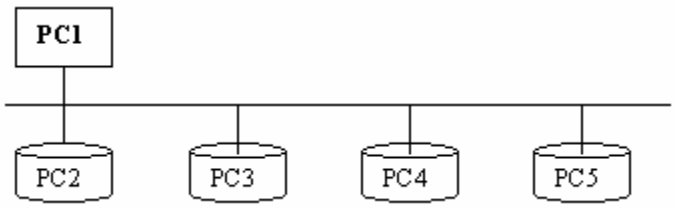


Figure 36. Example Network Configuration

Example 1: Communication Using OPC Over TCP/IP Direct

In the first example, PC1 communicates with PC2, PC3, PC4, and PC5 using only OPC over TCP/IP direct channels. Right-click **OPC over TCP/IP Direct Channels** and select **Set As Default** from the pop-up menu, as shown below. All nodes being requested by the client programs (e.g. ProcessView, WebHMI browsers, DataWorX, etc.) will communicate OPC data over a TCP/IP channel.

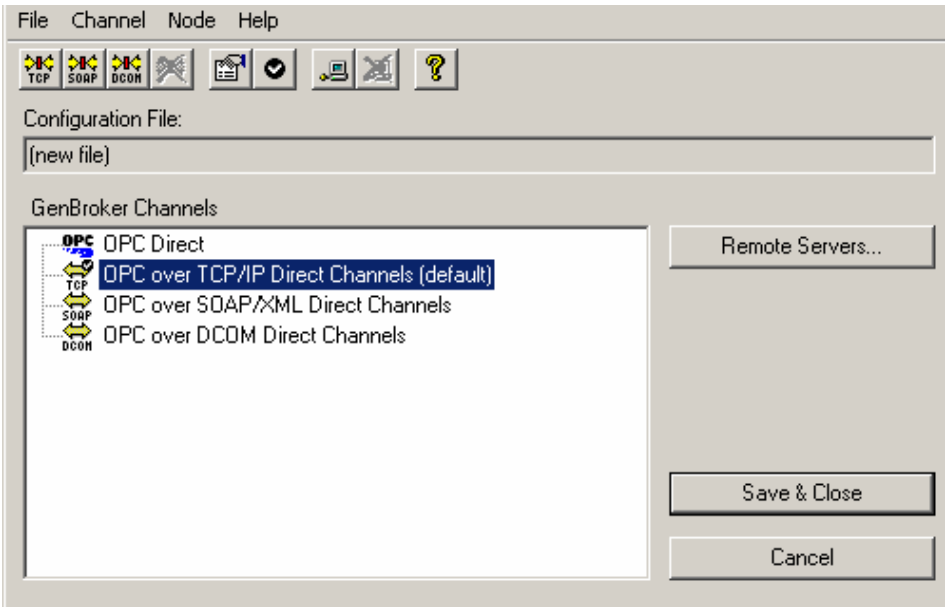


Figure 37. OPC Over TCP/IP Direct Channel

Example 2: OPC Over TCP/IP and OPC Over DCOM

In the second example, PC1 communicates with PC2 and PC3 using an OPC over TCP/IP direct channel.

1. First set OPC over TCP/IP as the default channel. Right-click **OPC over TCP/IP Direct Channels** and select **Set as Default** from the pop-up menu.

2. PC1 also communicates with PC4 and PC5 using an OPC over DCOM direct channel. Because OPC over TCP/IP is the default method of communication, you must now specify which nodes will communicate using OPC over DCOM. Right-click **OPC over DCOM Direct Channels** and select **Add Node** from the pop-up menu. This opens the **Select a Node** dialog box. Enter **PC4** in the **Node Name or IP Address** field, and then click **OK.** The node name will appear below the channel, as shown below. Repeat this procedure for **PC5.**

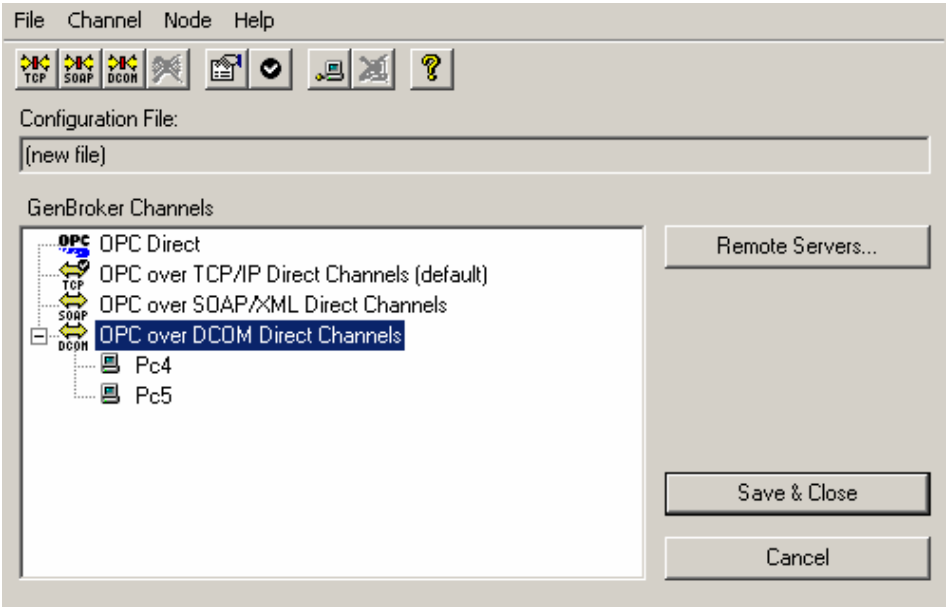


Figure 38. OPC Over TCP/IP (Default) and OPC Over DCOM

Example 3: OPC Direct, OPC Over TCP/IP, and OPC Over DCOM

In the third example, PC1 communicates with PC2 using OPC direct.

1. First set OPC direct as the default channel. Right-click **OPC Direct** and select **Set As Default** from the pop-up menu.

2. PC1 also communicates with PC3 using OPC over TCP/IP. Right-click **OPC over TCP/IP Direct Channels** and select **Add Node** from the pop-up menu. This opens the **Select a Node** dialog box. Enter **PC3** in the **Node Name or IP Address** field, and then click **OK.** The node name will appear below the channel, as shown below.

3. PC1 also communicates with PC4 and PC5 using OPC over DCOM. Right-click **OPC over DCOM Direct Channels** and select **Add Node** from the pop-up menu. This opens the **Select a Node** dialog box. Enter **PC4** in the **Node Name or IP Address** field, and then click **OK.** The node name will appear below the channel, as shown below. Repeat this procedure for **PC5.**

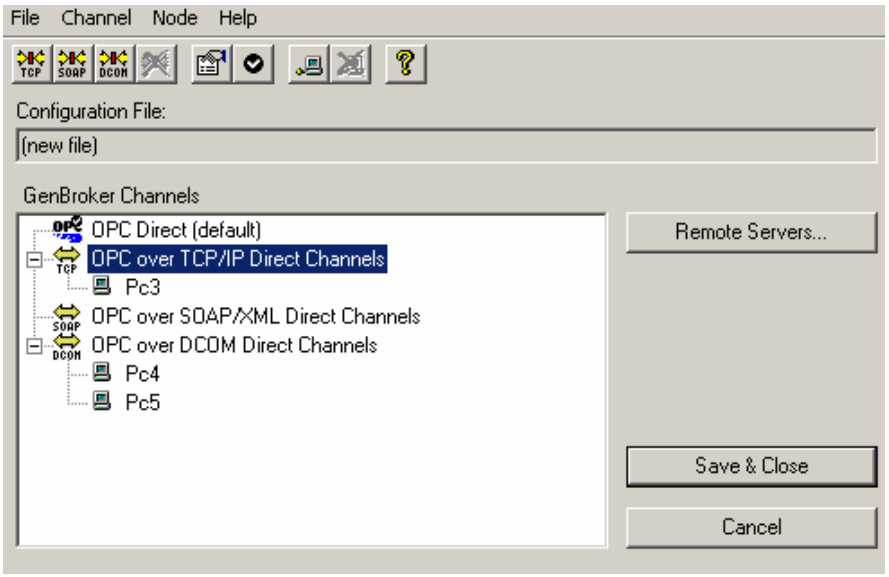


Figure 39. OPC Over TCP/IP Direct Channel

Example 4: Communication Using a Mediator Node

The final example demonstrates how communication between PC1 and the other four PCs can be routed through a mediator node, as shown below. PC1 will not communicate directly with the remote computers, but it instead will establish a connection with the mediator computer. This mediator will then route the communications to the final destinations (in this case PC2, PC3, PC4, or PC5). The mediator node may be located on a remote computer, which can only be reached through a RAS (remote access service) connection or the Internet.

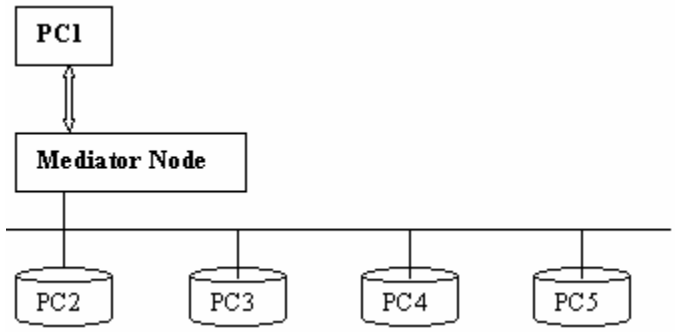


Figure 40. Communication via Mediator Node

The GenBroker configuration on the mediator node determines how the mediator will connect to PC2, PC3, PC4, and PC5. This may use OPC Direct, OPC over DCOM, or OPC over TCP/IP. To configure communication through the mediator node on a TCP/IP channel, for example, do the following:

1. Select **Add TCP/IP Channel** from the **Channel** menu. This opens the **Select a Node** dialog box. Specify a name for the mediator node (for example, "Mediator") in the **Node Name or IP Address** field, and then click **OK.** The new channel will appear in the GenBroker Configurator screen as **OPC over TCP/IP Channel via Mediator,** and the new mediator node will appear below the new channel, as shown below.

2. Now add PC2, PC3, PC4, and PC5 to the new TCP/IP channel. Right-click **OPC over TCP/IP Channel via Mediator** and select **Add Node** from the pop-up menu. This opens the **Select a Node** dialog box. Enter **PC2** in the **Node Name or IP Address** field, and then click **OK.** The node name will appear below the channel, as shown below. Repeat this procedure for PC3, PC4, and PC5.

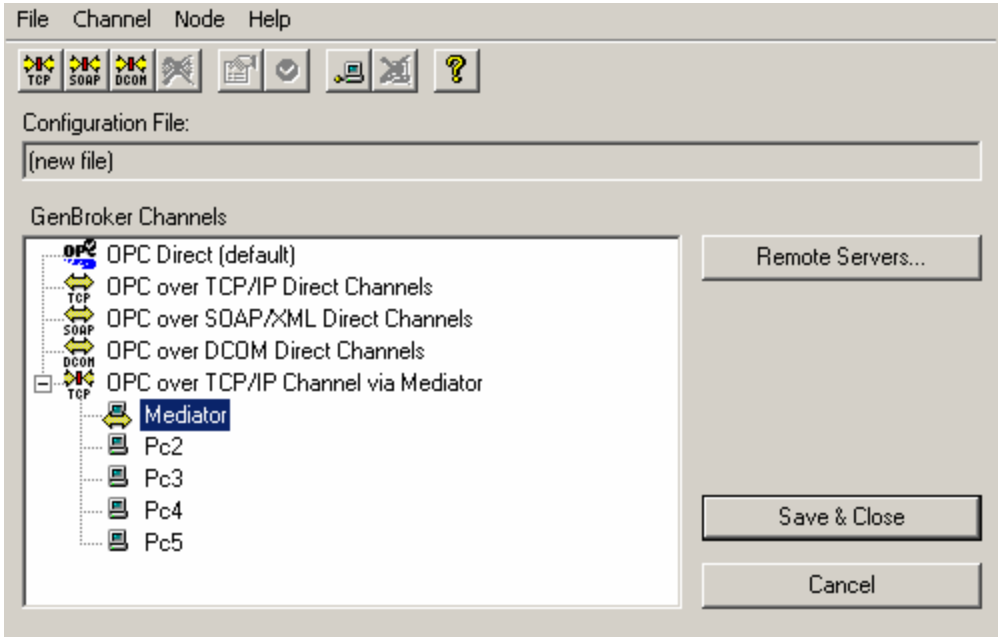


Figure 41. Configuring a Mediator Node

Example 5: OPC Direct, OPC Over SOAP/XML, and OPC Over DCOM

In the fifth example, PC1 communicates with PC2 using OPC direct.

1. First set OPC direct as the default channel. Right-click **OPC Direct** and select **Set As Default** from the pop-up menu.

2. PC1 also communicates with PC3 using OPC over SOAP/XML. Right-click **OPC over SOAP/XML Direct Channels** and select **Add Node** from the pop-up menu. This opens the **Select a Node** dialog box. Enter **PC3** in the **Node Name or IP Address** field, and then click **OK.** The node name will appear below the channel, as shown below.

3. PC1 also communicates with PC4 and PC5 using OPC over DCOM. Right-click **OPC over DCOM Direct Channels** and select **Add Node** from the pop-up menu. This opens the **Select a Node** dialog box. Enter **PC4** in the **Node Name or IP Address** field, and then click **OK.** The node name will appear below the channel, as shown below. Repeat this procedure for **PC5.**

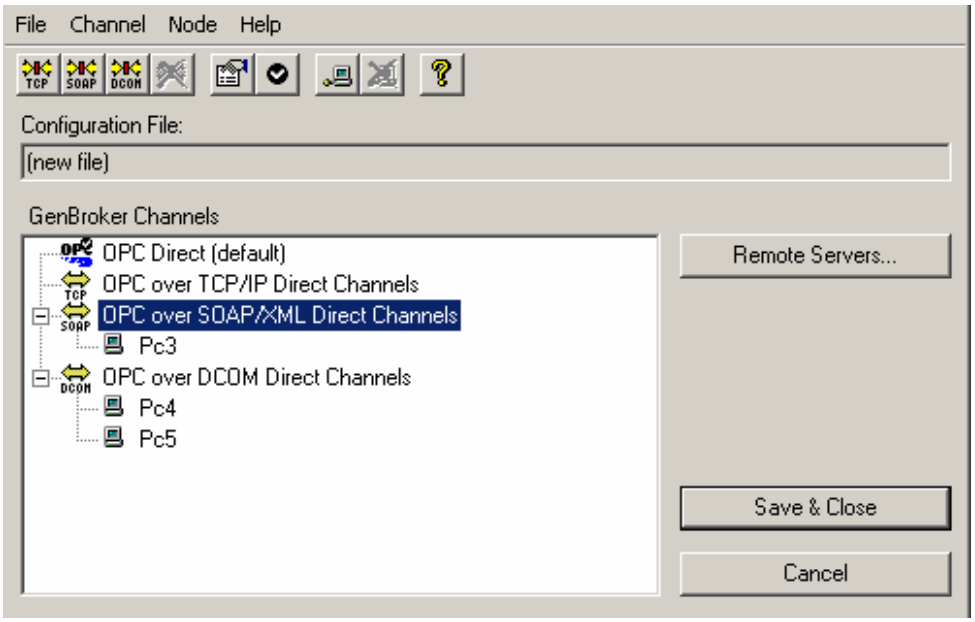


Figure 42. OPC Over SOAP/XML Direct Channel

WebHMI Client Configuration

WebHMI Client/Server Architecture

GenBroker acts as a bridge that links WebHMI clients to the WebHMI server over the Internet. The WebHMI clients are used to view Web pages downloaded from the WebHMI server through a Web browser (Internet Explorer or Netscape Navigator). As shown in the figure below, GenBroker enables communication between the Web clients and the WebHMI server over TCP/IP.

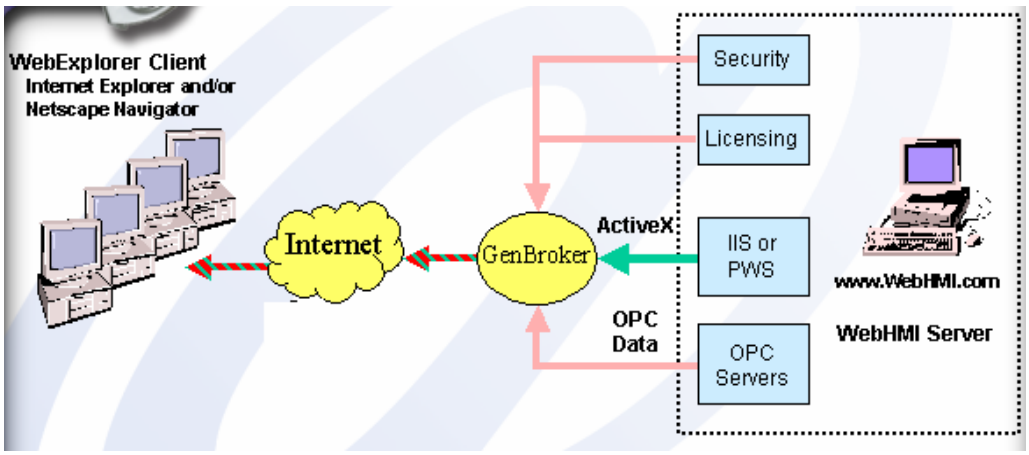


Figure 43. WebHMI Client/Server Architecture

Configuring GenBroker for the Client Side

GenBroker is configured for both the client side and the server side. Here we are concerned with WebHMI client configuration. For information on how to configure GenBroker for the server side, and for example GenBroker configurations, please see the GenBroker Help documentation.

WebHMI uses a configuration (.gbc or .gbx) file for GenBroker. The GenBroker configuration file used by the client resides on the server side. Once the client connects to the server, the .gbc or .gbx file will be downloaded from the server.

The first step in the WebHMI client configuration is to specify which .gbc or .gbx file to use. You can use a default file provided in the WebHMI installation, or you can you can create a new file. To work with a .gbc or .gbx file, you must start the WebHMI client configuration from the Windows **Start** menu by selecting **Programs > Smar > WebHMI> Client Configuration,** as shown in the figure below.

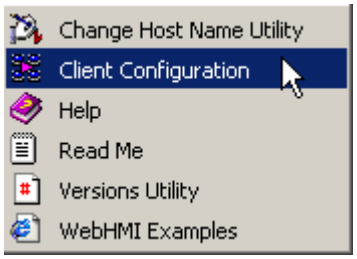


Figure 44. Starting the Client Configuration

This opens the **WebHMI** tab of the **GenBroker Properties** dialog box, as shown in the figure below. To use the default configuration file, click the **...** button to the right of the **GenBroker Configuration to Upload** field. This allows you to select a .gbc file. Browse to "InetPub\wwwroot\WebHMI\Samples" and select **Default.gbx.**

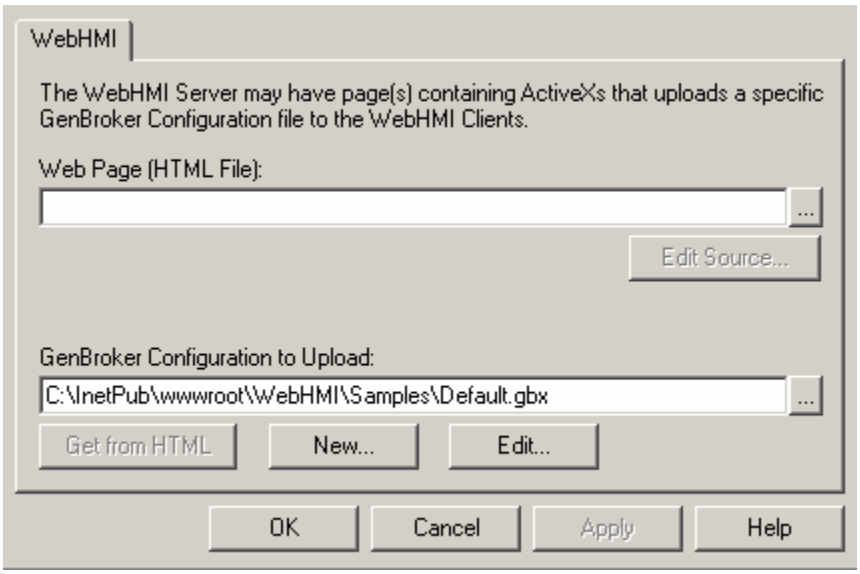


Figure 45. WebHMI Client Configuration

The file pathway will appear in the text box, as shown in the figure above. You do not have to reconfigure the default file except to:

- Redirect channels of a WebHMI server.
- Add a channel for a new node whose resident OPC server provides data to client PC browsers.

If you want to set up a different or more advanced client/server communication architecture, you can either reconfigure the “default.gbx” file or create a new .gbc or .gbx file.

Editing the Default Configuration File

To reconfigure the default configuration, click the **Edit** button on the **WebHMI** tab of the **GenBroker Properties** dialog box. This opens the GenBroker Configurator screen, shown in the example in the figure below. Notice that the file pathway is displayed under the **Configuration File** field. In the example, the IP address "99.99.99.99" is the default mediator node and is the node assigned to the Security server and the License server. You can add, remove, or reconfigure nodes using the commands and functions described in the following sections.

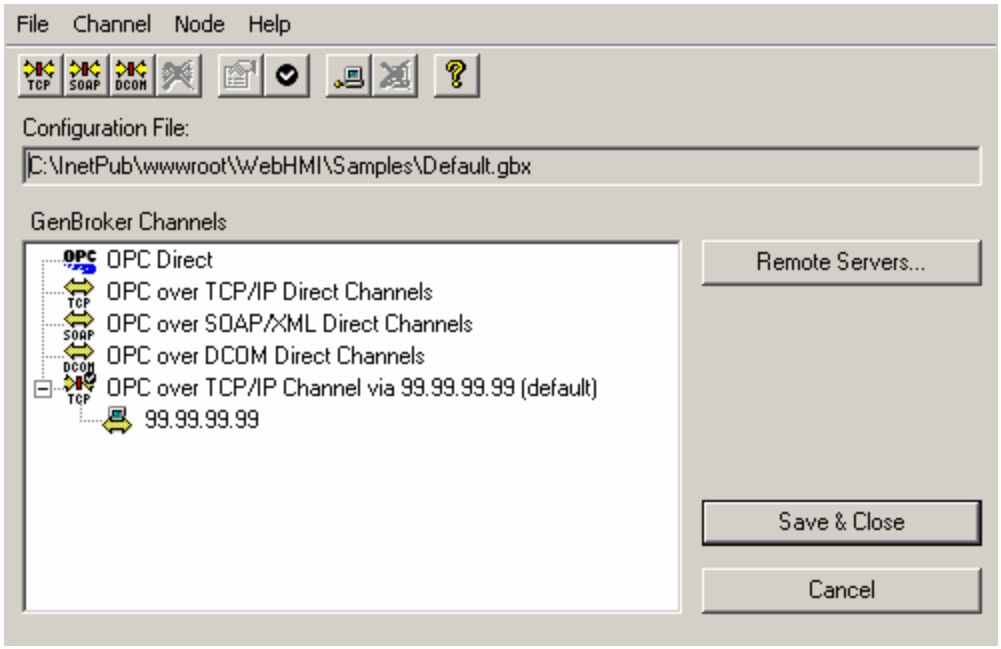


Figure 46. Editing the Default GenBroker Configuration

Creating a New GenBroker Configuration File

You can also create a new configuration by clicking the **New** button on the **WebHMI** tab. This opens the GenBroker Configurator screen, shown in the figure below. You can add and configure nodes using the commands and functions described in the GenBroker Help documentation. To view the GenBroker Help, select **Help Topics** from the **Help** menu.

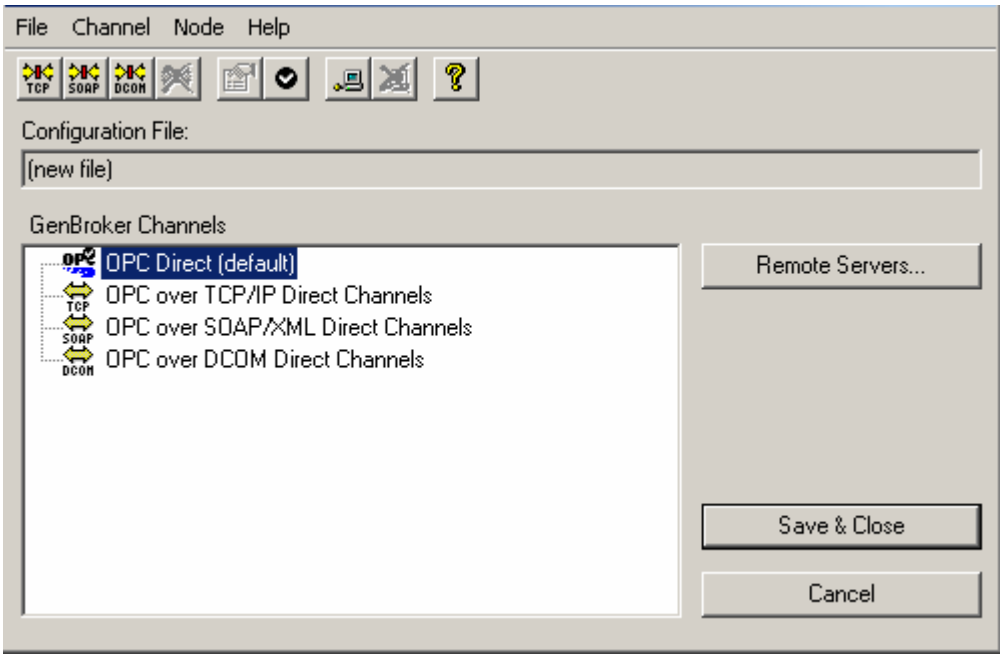


Figure 47. Creating a New GenBroker Configuration

Retrieving a GenBroker Configuration File From a Web Page

Alternatively, you can search for a GenBroker configuration file in an HTML file. In many cases, .gbc or .gbx files are referenced in the source code of HTML files. Click the **...** button to the right of the **Web Page (HTML File)** field on the **WebHMI** tab of the **GenBroker Properties** dialog box. Browse to select an HTML file, and then click the **Open** button. The HTML file pathway will now be displayed in the **Web Page (HTML File)** text box. In addition, the **Edit Source** button will be enabled. Clicking **Edit Source** allows you to view the source code for the selected HTML file. Once you have selected the HTML file, click the **Get From HTML** button to retrieve the .gbc or .gbx file that is referenced in the HTML source code.