

Kepware Products for Windows 95™, 98™, 2000™, NT™ and XP™

KepserverEx Client Connectivity Guide



KTSM-00001

Copyright © 2001, Kepware Technologies

KEPWARE END USER LICENSE AGREEMENT AND LIMITED WARRANTY

The software accompanying this license agreement (the Software) is the property of Kepware Inc, and is protected by United States and International Copyright laws and International treaty provisions. No ownership rights are granted by this Agreement or possession of the Software. Therefore, you must treat the Licensed Software like any other copyrighted material (e.g., a book or musical recording), except that you may make a single copy for backup or archival purposes. Your rights and obligations in its use are described as follows:

1. You may use and display this software on a single computer.
2. You may make one copy of the software for archival purposes or you may copy the software onto your hard disk and hold the original for archival purposes.
3. You may not modify or attempt to reverse engineer the software, or make any attempt to change or even examine the source code of the software.
4. You may transfer the software to another computer using the utilities provided. However, the software must be used on only a single computer at one time.
5. You may not give or distribute copies of the software or written materials associated with the software to others.
6. You may not sub-license, sell, or lease the software to any person or business.

Return Policy

The original licensee of the software can return it within sixty (60) days of purchase. Please call us for a Return Material Authorization Number.

Limited Warranty

Kepware does not warrant that the Software will be error free, that it will satisfy your planned applications or that all defects in the Software can be corrected. If Kepware provides information or assistance regarding the use of the Software or otherwise, Kepware is not assuming the role of engineering consultant. Kepware disclaims responsibility for any errors or omissions arising in connection with engineering in which its Software or such information or assistance is used.

The foregoing is the sole and exclusive warranty offered by Kepware.

Kepware disclaims all other warranties, express or implied, including but not limited to the implied warranties of merchantability and fitness for a particular purpose, with regard to the licensed software and all accompanying materials.

In no event shall Kepware be liable for incidental or consequential damages, including lost profit, lost savings, lost opportunities, or other incidental or consequential damages arising out of the use or inability to use the licensed software, even if Kepware has been advised of the possibility of such damages.

Kepware's entire liability shall be, at Kepware's option, either (a) return of the price paid for the Software (or component), or (b) repair or replacement of the Software (or component) that does not meet Kepware's Limited Warranty and which is returned to Kepware within the warranty period. This shall be the sole and exclusive obligation of Kepware and your sole and exclusive remedy with respect to any such failure. The Limited Warranty is void if failure of the Software (or component) has resulted from accident, abuse or misapplication.

Support

Kepware provides *unlimited e-mail* support for all Software whether a demo or registered license. Kepware will provide a total of *two hours* free phone support for all registered Software after paying the applicable license fees. Kepware will provide *unlimited phone* support so long as you have paid Kepware any applicable maintenance or support fees and subject to the terms of those agreements. All corrections and maintenance releases will be made available through Kepware's Internet site. All major product releases of the Software are subject to upgrade fees. At no time will on-site support be provided without advance payment to Kepware for a minimum of two days on-site engineering support services, plus all expenses.

Trademarks

Microsoft® and Microsoft Excel® are registered trademarks, Windows™ is a trademark of Microsoft Corporation.

32 Bit KepserverEx Connectivity Guide

Kepware Technologies

81 Bridge Street
Yarmouth, Maine 04096

Sales: (207) 846-5881

Technical Support: (207) 846-5881

Fax: (207) 846-5947

E-mail: Sales@kepware.com or

Technical.Support@kepware.com

Home Page: <http://www.OPCSource.com>

Copyright © 2001, Kepware Technologies

Table of Contents

INTRODUCTION TO KEPSERVEREX	1
ACCESSING KEPSERVEREX	1
USING KEPSERVEREX DRIVERS	2
AUTOMATIONDIRECTS LOOKOUT™ DIRECT AS AN OPC CLIENT	1
CONNECT TO THE SERVER FROM LOOKOUT™ DIRECT	1
CREATE A NEW PROCESS	1
CREATE AN OPCCLIENT OBJECT	3
ADD A DISPLAY OBJECT TO THE PANEL	6
VIEW LIVE DATA	9
CUTLER-HAMMER'S PANELMATE PC PRO AS AN OPC CLIENT	11
ABOUT PANELMATE PC PRO	11
<i>Creating a PanelMate PC Project</i>	<i>11</i>
<i>Create a Connection to the Server</i>	<i>12</i>
<i>Configure a Display Page</i>	<i>14</i>
<i>Save your configuration</i>	<i>17</i>
<i>Run your PanelMate PC Pro Project</i>	<i>17</i>
GE'S CIMPLICITY® AS AN OPC CLIENT	19
CONNECT TO THE SERVER FROM CIMPLICITY V5.0	19
<i>Create a New Project</i>	<i>19</i>
<i>Run the Project Wizard</i>	<i>21</i>
<i>Running the Project and Connecting to KepserverEx</i>	<i>25</i>
OPTIMIZING YOUR CIMPLICITY 5.0 PROJECT	28
<i>Adding Additional Points to your Project</i>	<i>30</i>
CONNECT TO THE SERVER FROM CIMPLICITY V4.01	33
<i>Create a New Project</i>	<i>33</i>
<i>Create an OPC Port</i>	<i>36</i>
<i>Create a Device</i>	<i>37</i>
<i>Create a Data Point</i>	<i>39</i>
<i>Check the OPC Connection</i>	<i>41</i>
UPGRADING A PROJECT FROM KEPSERVER V3.2 TO V4.0	41
ICONICS' GENESIS32® AS AN OPC CLIENT	43
CONNECT TO THE SERVER FROM GENESIS32 COMPONENTS	43
<i>Select a Process Point</i>	<i>43</i>
Browse for an Available Server	44
Select Tag from Browser	45
<i>Check the OPC Connection and View Server Data</i>	<i>47</i>
INTELLUTION'S FIX DYNAMICS® AND OPC POWERTOOL® AS AN OPC CLIENT	49
CONNECT TO THE SERVER FROM IFIX	49
<i>Configure the SCADA Properties</i>	<i>49</i>
<i>Enable the Fix OPC Client</i>	<i>51</i>
<i>Add a Server to the OPC PowerTool</i>	<i>52</i>
Browse for Available Servers	52
<i>Add an OPC Group</i>	<i>53</i>
<i>Add an OPC Item/Tag</i>	<i>54</i>
Browse for Available Tags	55
KEPWARE'S OPC QUICK CLIENT AS AN OPC CLIENT	57

CONNECT TO KEPSERVEREX FROM KEPWARE'S OPC QUICK CLIENT.....	57
<i>Auto Launch and Generate a Quick Client Project</i>	57
<i>Manually Create a Quick Client Project</i>	58
Browse for KepsServerEx	58
Add a Tag/Item Group.....	59
Set Group Properties.....	59
Add Tags/Items	59
Check OPC Connection.....	60
OSI PI AS AN OPC CLIENT.....	63
CONNECT TO THE SERVER FROM OSI PI.....	63
<i>Start the OSI Pi Process Services</i>	63
<i>Create PI Tags in the PI Pointbuilder</i>	63
STARTING THE OPC INTERFACE.....	65
CREATE A DISPLAY WITH PI PROCESSBOOK.....	67
<i>Open the PI ProcessBook</i>	67
<i>Create a New ProcessBook Project</i>	67
<i>Create a Display in the ProcessBook</i>	68
<i>Create a Trend Object in the Display</i>	69
Defining the Trend Object Properties.....	69
<i>Viewing Tag Data in the Trend</i>	73
SAVE AND CLOSE THE PROJECT	74
ROCKWELL SOFTWARE'S RSVIEW32® AS AN OPC CLIENT	75
CONNECT TO THE SERVER FROM RSVIEW32.....	75
<i>Create a Device Node</i>	75
Browse for an Available Server.....	77
<i>Create a Tag</i>	78
Browse for Tags in the Server	79
<i>Check the OPC Connection</i>	81
UPGRADING YOUR CONNECTIVITY FROM KEPSERVER TO KEPSERVEREX.....	83
<i>Converting From KepsServer OPC to KepsServerEx OPC</i>	83
<i>Converting From KepsServer Advanced DDE to KepsServerEx OPC</i>	86
<i>Converting From KepsServer Advanced DDE to KepsServerEx Advanced DDE</i>	86
SIEMENS' WINCC AS AN OPC CLIENT	89
CONNECT TO KEPSERVEREX FROM WINCC	89
<i>Create a New WinCC Project</i>	89
Add a Driver to the Project.....	90
Create Driver/Group Connection.....	91
Create A Tag	93
<i>Create Graphic Display</i>	94
<i>View Data and Check the OPC Connection</i>	97
THINK & DO'S LIVE! AS AN OPC CLIENT	99
ABOUT THINK & DO LIVE!	99
CONNECT TO KEPSERVEREX FROM THINK & DO'S LIVE!	99
<i>Configure Tag Link to server</i>	103
<i>Create a Display Window</i>	103
WONDERWARE'S INTOUCH® AS A FASTDDE® OR SUITELINK® CLIENT	107
CONNECT TO KEPSERVEREX WITH FASTDDE® OR SUITELINK®	107
<i>Verify FastDDE/SuiteLink Support in the Server</i>	107
<i>Add an Alias (DDE Topic)</i>	108
<i>Create an InTouch® project</i>	109
Add Access Name Links to your Project.....	110

Add Tags to the Tagname Dictionary	113
<i>Create a Window to Show the Device Values</i>	114
Add and Animate a Text Object.....	115
CHECK THE CONNECTION TO THE SERVER AND VIEW DATA	117
APPENDIX A	119
CIMPPLICITY MASTER OPC INI FILES	119
<i>About the Master OPC INI File</i>	119
<i>The INI File Sections</i>	119
PortLevel Section	120
Device1 Section	121
DefaultPoll Section	122
GLOSSARY OF TERMS	125

Introduction to KepserverEx

KepserverEx is a 32 bit windows application that provides a means of bringing data and information from a wide range of industrial devices and systems into client applications on your windows PC. KepserverEx falls under the category of a "Server" application. It is very common to hear the term "client/server application" in use across many software disciplines and business segments. In the industrial market, it has usually come to mean the sharing of manufacturing or production data between a variety of applications ranging from human machine interface software and data historians, to large MES and ERP applications.

Regardless of the business segment served, client/server applications have one thing in common: a standardized method of sharing data. In the industrial segment, many client/server technologies have been developed over the last ten years. Initially, some of these technologies were proprietary. In many cases these proprietary client/server architectures were in wide use but remained unavailable to third party applications. Early in the development of windows, Microsoft provided a generic client/server technology called DDE or Dynamic Data Exchange. DDE did provide a basic architecture that would allow many windows applications from a wide range of vendors to share data, but there was one problem. DDE was not designed for the industrial market. It lacked much of the speed and robustness desired in an industrial setting. However, this did not stop DDE from becoming a dominant client/server architecture, largely due to its availability in most windows applications. In time, variations on Microsoft's DDE were developed by some of the leading vendors in the market. These variations addressed some of the speed and reliability issues of DDE but many people in the industrial segment agreed that a better system needed to be developed.

With the advent of 32 bit Operating Systems, and the use of Ethernet to provide communications between devices, there was a need for quicker and cleaner data transfer between software applications. This is where OPC saw its birth into the industry.

OPC (OLE for Process and Control) servers provide a standardized method of allowing multiple industrial applications to share data in a quick and robust manner. The OPC server provided in this package has been designed to meet the demanding requirements found in the industrial environment.

This OPC server has been designed as a two-part program. The primary component provides all of the OPC and DDE connectivity as well as the user interface functions. The second part is comprised of plug-in communications drivers. This two-part design allows you to add multiple communications options to your SCADA application while utilizing a single OPC server product thus reducing your learning curve as your project grows.

OPC technology reflects the move from closed proprietary solutions to open architectures that provide more cost-effective solutions based on established standards.

Accessing KepserverEx

A Windows based client application must be used to view data from the KepserverEx application. In this section we will cover the basics of connecting a number of common OPC clients to KepserverEx. While we cannot possibly cover every client application that exists, we believe that after reviewing this document you should be able to deal with most client applications.

The intention of this section is to show connectivity to KepserverEx. It is assumed that you have already either configured your KepserverEx application by selecting the appropriate driver and settings or you have run the Simulator demo (Simdemo.opf) which is included with KepserverEx. For simplicity, the Simdemo project will be used for all examples contained in this section.

Before beginning any of the examples, start the KepserverEx application by selecting it from your Start Menu or from its desktop icon. Once the server is loaded, use the File|Open command to

load the “Simdemo” project. The KepserverEx application is always active once you have opened an existing project or configured at least one channel and device in a new project. After you have selected a project, in this case the Simdemo project, KepserverEx will automatically load this project when an OPC client application invokes KepserverEx’s OPC server component.

Users have always had the ability to create what we refer to as “user defined tags” in their KepserverEx application. Prior to OPC, defined tags gave a DDE application designer the ability to create a label for device data. Assume register 1000 contained the value of parts made, without defined tags a DDE application would have directly accessed register 1000. Using defined tags a label can be created like “PartsMade”. Now the DDE application could access the data via this new label, removing the machine level knowledge from the client application and keeping it at the server level where it belongs. This label, while useful for DDE is a necessity for OPC clients. For OPC clients, defined tags take on a greater role. Like the DDE example, defined tags allow you to create labels for your device data and keep the configuration of those tags in the server. OPC clients have a major advantage over DDE clients. OPC clients can browse the defined tags you create in your KepserverEx application, which allows you to simply point and click on a tag to add it to your OPC client project.

OPC Tag Browsing allows you to see a list of the defined tags you have created in your KepserverEx application, directly within your OPC client application.

For more information on defined tags see the “Designing a Project” section of the KepserverEx help file, which can be accessed from the Help/Contents menu selection of the KepserverEx application.

Using KepserverEx Drivers

Part of the innovative design of Kepware’s OPC/DDE Server Technology is the separation of the Hardware Protocol Driver from the Server Technology. This separation allows the user to use one or more drivers in the server at the same time. Each driver has its own help file which provides information on devices supported, communications parameters, cabling, addressing, and error messages.

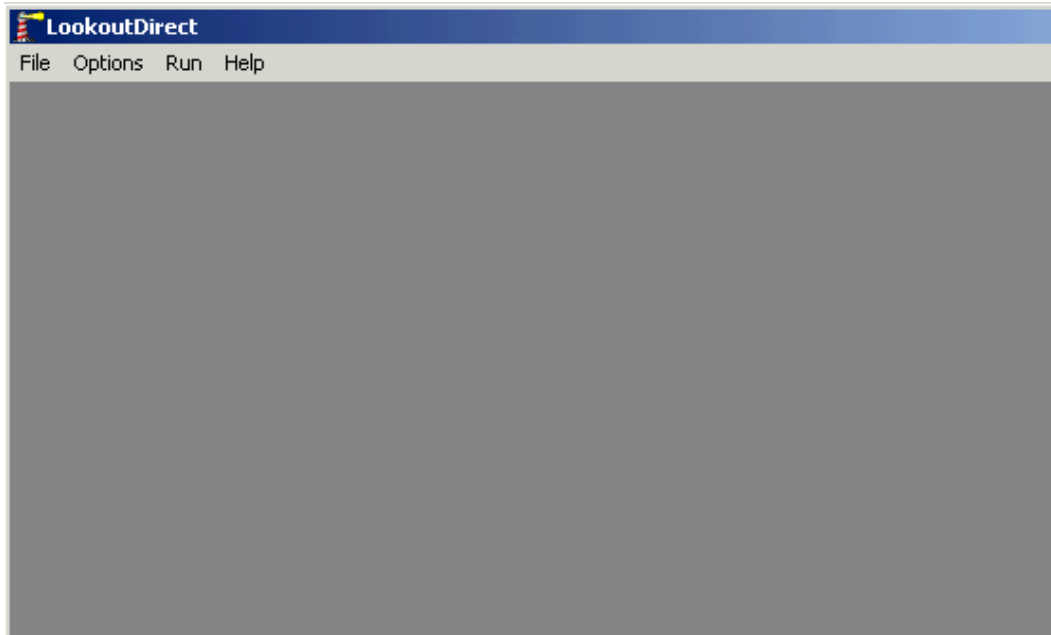
The driver help file should contain all of the information you will need to connect your device to the PC so that we can talk to it. If you do not connect to the device be sure to check the error messages and look up their meaning in the help file.

Automationdirects Lookout™ Direct as an OPC Client

Connect to the Server from Lookout™ Direct

Automationdirects Lookout Direct as an OPC client is one of many HMI's that can be used to connect to KepserverEx. The Lookout Direct version used for this example was version 4.5.1 Build 18. The following steps will show you how to create an OPC connection to the Server from Lookout Direct.

First you will need to open your copy of Lookout Direct and click on File|New to start a new process in Lookout Direct.

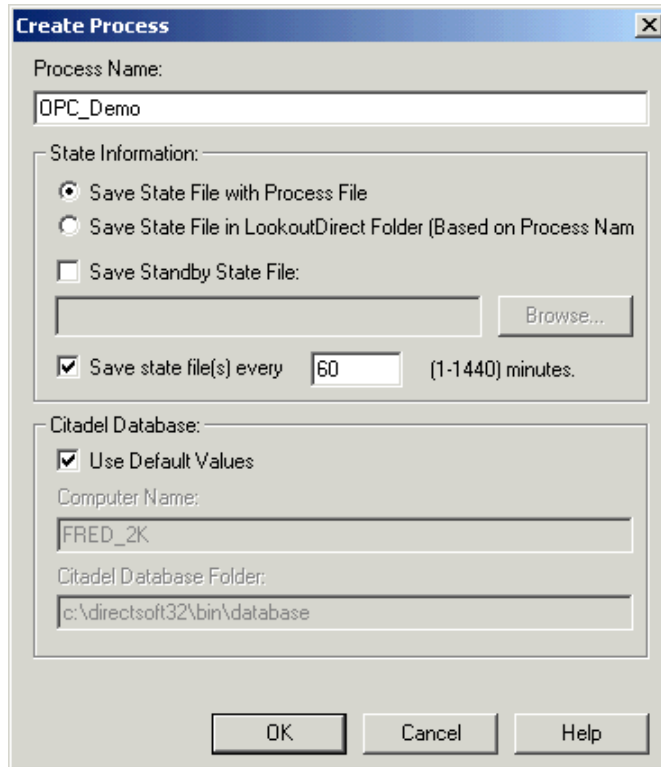


KepserverEx is capable of being an OPC or DDE server to Lookout Direct. If you define an OPCClient object in Lookout Direct, then KepserverEx will be an OPC server to it. Likewise, if you define a DDE object in Lookout Direct, KepserverEx will be a DDE server

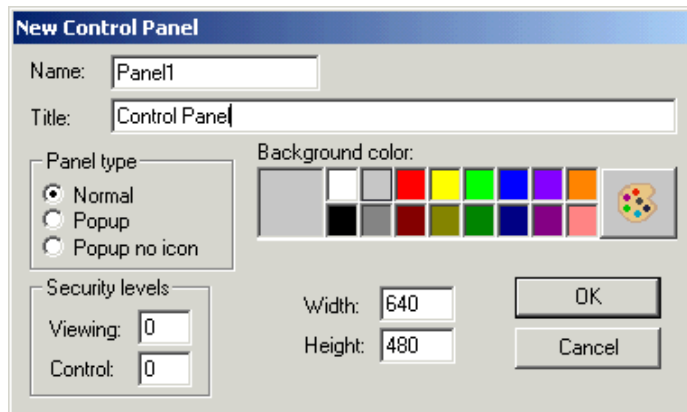
Create a New Process

You will now see a Create Process dialog box. Enter a name for your process in the Process Name field. We named ours "OPC_Demo" and accepted the defaults for the rest of the parameters. For more information on the other parameters in the Create Process dialog box see the Lookout Direct Users manual.

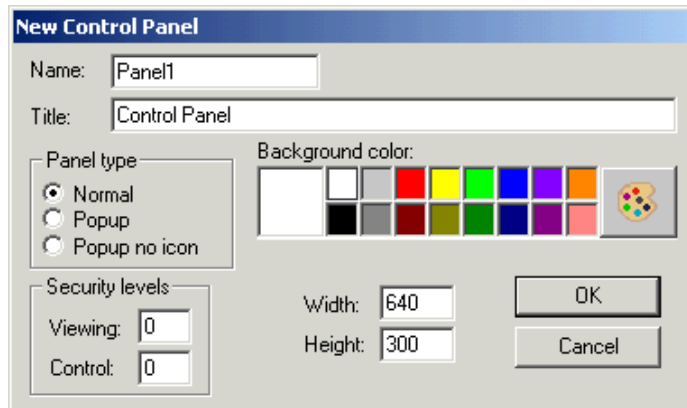
Lastly, click OK to finish.



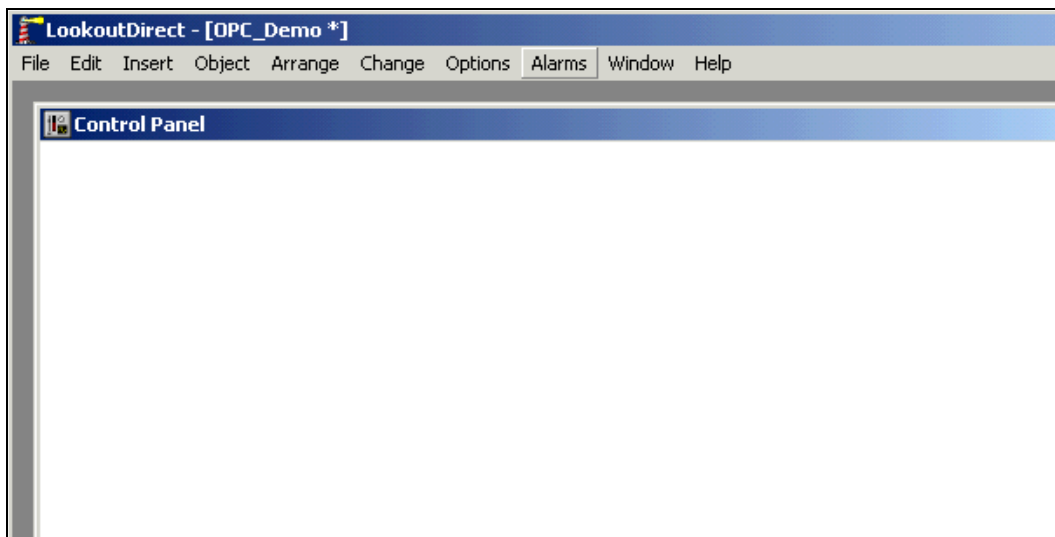
Next, Lookout Direct will prompt you to create a new control Panel.



For instruction purposes you can accept the defaults. If you are not creating a full screen project you may wish to change the panel size, we changed ours to 640 by 300.

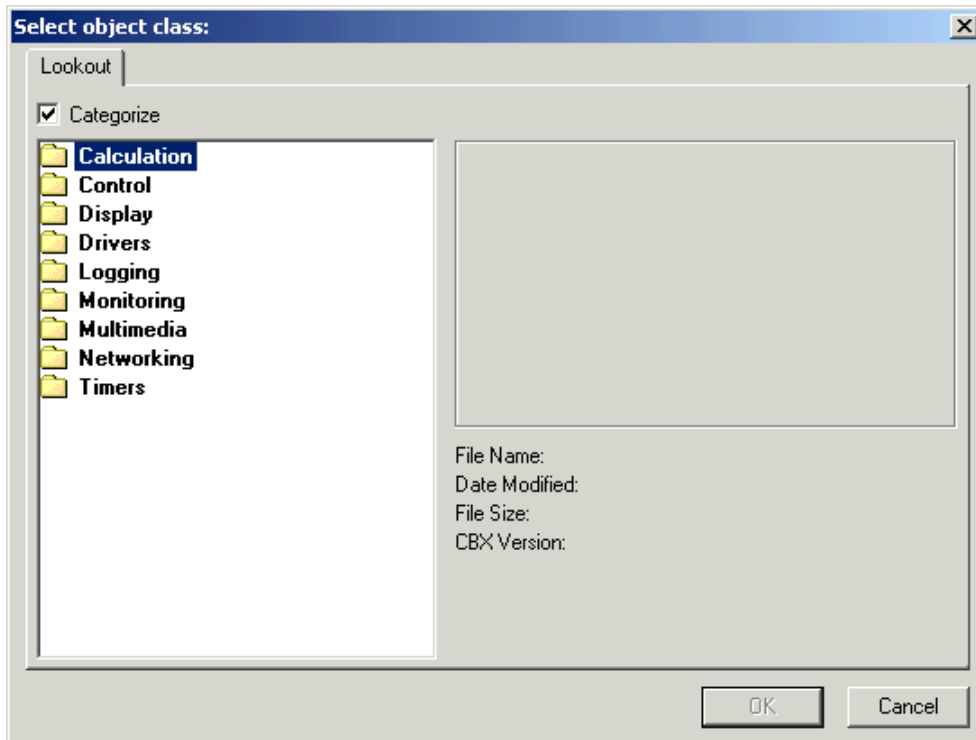


Click OK to accept the changes that were made to the panel and create it as an object in the process.

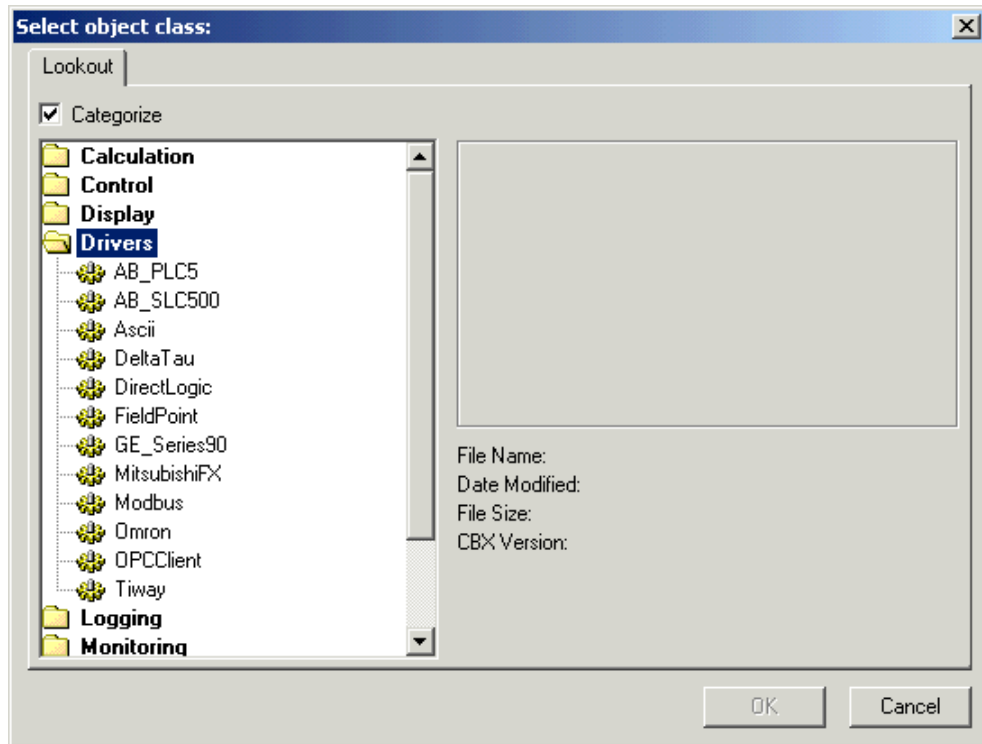


Create an OPCClient Object

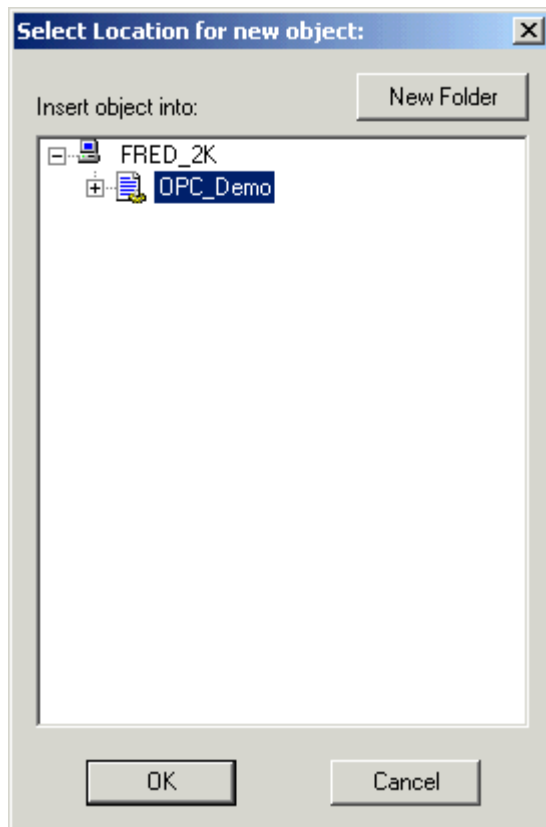
Now that you have a process started you will need to get data into it for display and control. To do this in Lookout Direct you first need to create an object class to handle the communications between KepserverEx and Lookout Direct. To do this select Object|Create... from the Lookout Direct Main Menu.



In the Select Object Class dialog double click on the Drivers Folder and then select the OPCClient and click OK to accept it.



Next you will get a Select Location for new object dialog box. Select the process that you created earlier and then click OK to finish.



Now you will get a Create OPC Client dialog. You can accept the default Name or you can create your own. In the OPC Server Settings section you are going to use the drop down list to select

KepserverEx. If you have more than one OPC server installed on your PC the Server Name you will select is "KEPware.KEPServerEx.V4".

Create OPC Client

Name:

OPC Server Settings

Server Name:

In-Process Server
 Local Server
 Remote Server

Browsing

Disabled Flat Hierarchical

Use Asynchronous I/O
 Force Refresh after Write

OPC Group Settings

Update Rate: milliseconds
Deadband: percent
Poll Device = (optional)
Default Access Path: (optional)

Communication alarm priority:

If you are going to connect to KepserverEx running on another PC then you would select the Remote Server radio button and type the name or IP address of the PC that is running the server. Unlike NetDDE connections you do not use backslashes in the name. You will also need to properly configure DCOM on all PC's involved in the connection. For details on Configuring DCOM for KepserverEx see the DCOM configuration guide which is on your installation CD or available for download at www.kepware.com/support.

For now we will assume that you are connecting to a local server so you should be able to click on OK to accept the client settings. Look in your Lookout Direct Help for an explanation of the other fields in the OPCClient dialog.

Create OPC Client

Name:

OPC Server Settings

Server Name:

In-Process Server
 Local Server
 Remote Server

Browsing

Disabled Flat Hierarchical

Use Asynchronous I/O
 Force Refresh after Write

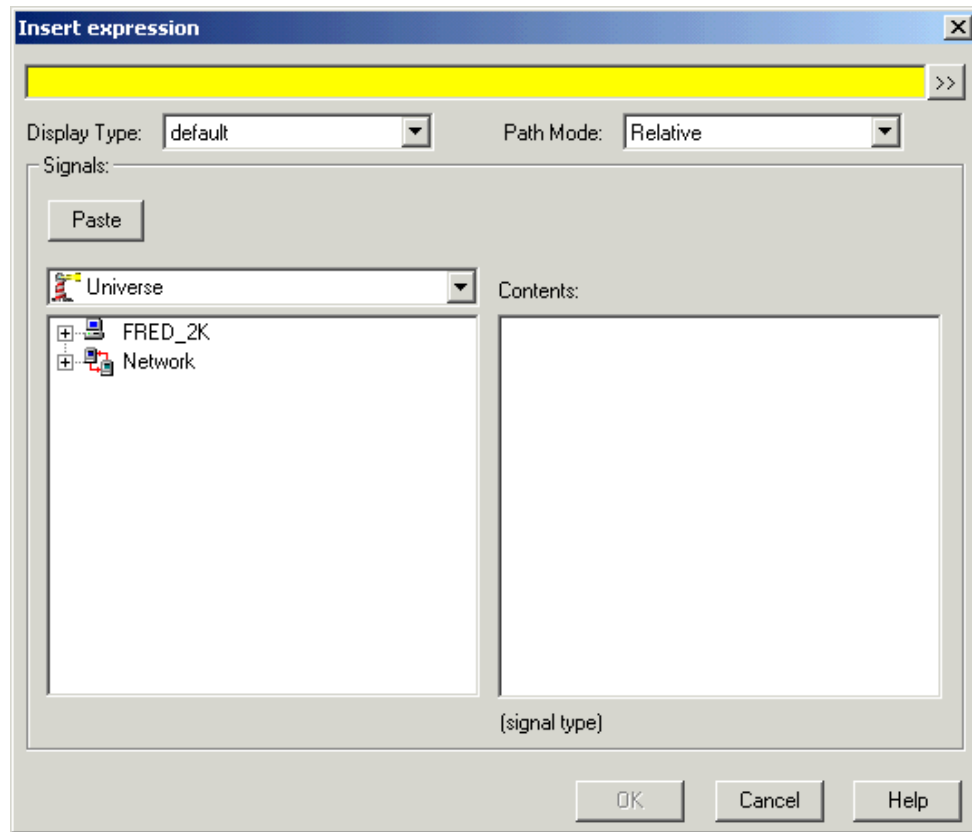
OPC Group Settings

Update Rate: milliseconds
Deadband: percent
Poll Device = (optional)
Default Access Path: (optional)

Communication alarm priority:

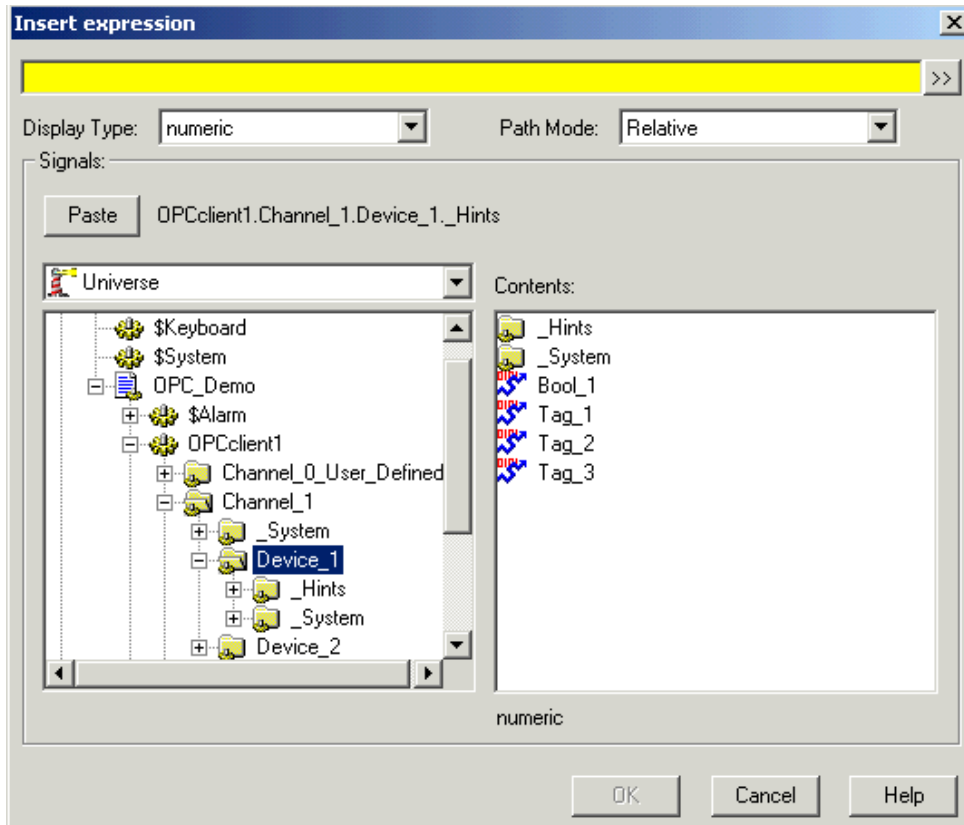
Add a Display Object to the Panel

Next, you need to display a value from the server in the Control Panel. To do this you will click on Insert|Expression... in the Lookout Direct Main Menu.

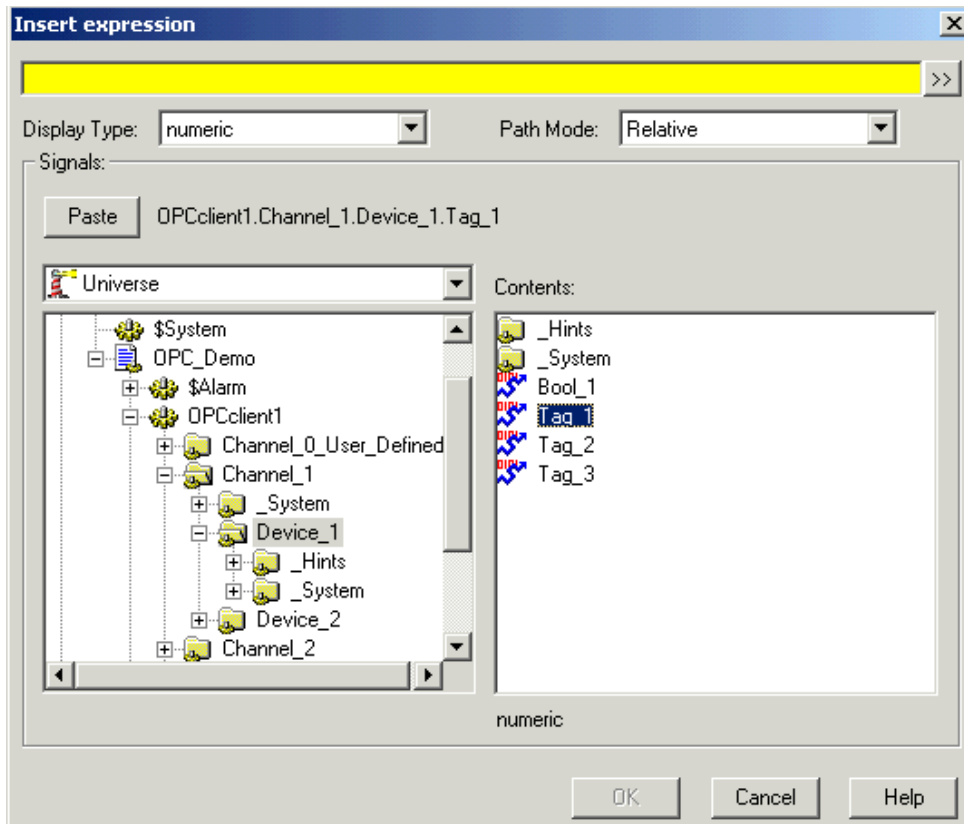


In the Signals section of the Insert Expression dialog you are going to expand the Objects in the left hand pane until you can see the folders under the OPCClient Object that you created. These folders are the Channels and Devices that were created in the KepserverEx project.

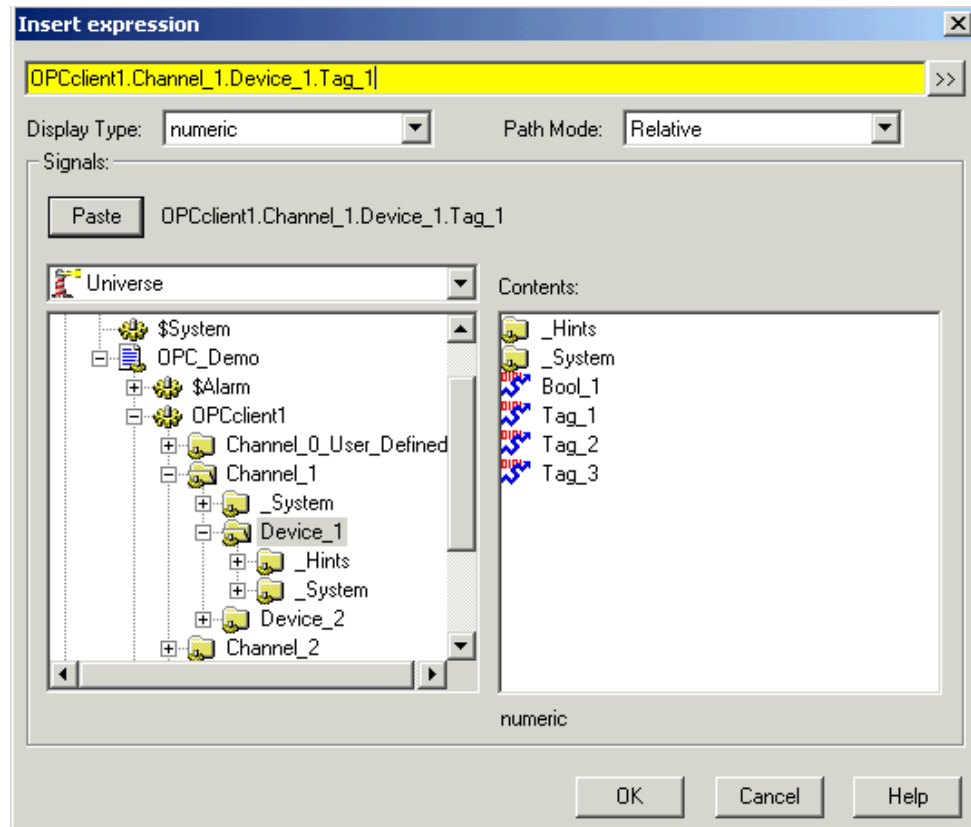
Select the Device Folder that contains the tag you wish to display in the control panel. In the Right Hand pane of the dialog you will see folders for internal tags specific to the device and you will see the tags that were added to the device in the KepserverEx.



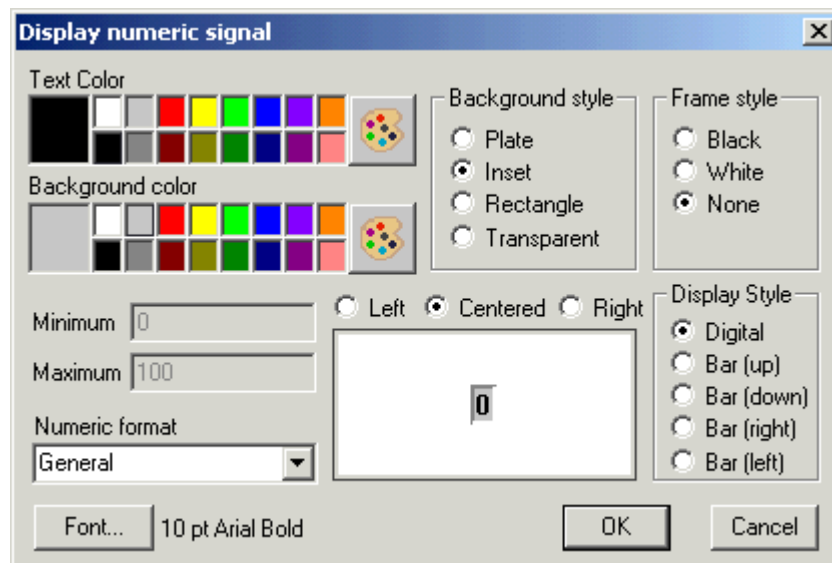
Select the tag you wish to display in the panel and click on the Paste button in the dialog to add the tag to the Expression field at the top of dialog box.



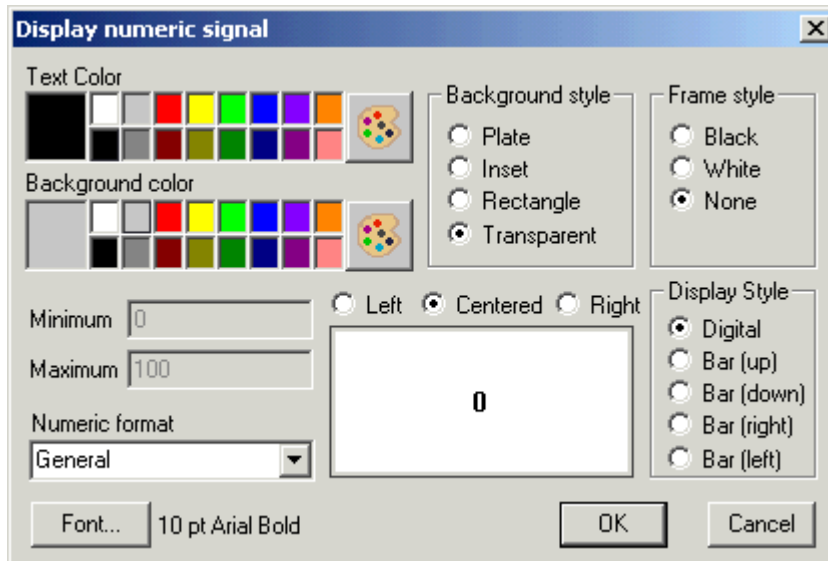
Click OK to accept the expression.



In the Display Numeric Signal dialog you will determine how the tag you selected will be displayed. In our example we select a tag that is a Word so we will accept the defaults except for the Background Style which we will change to Transparent.

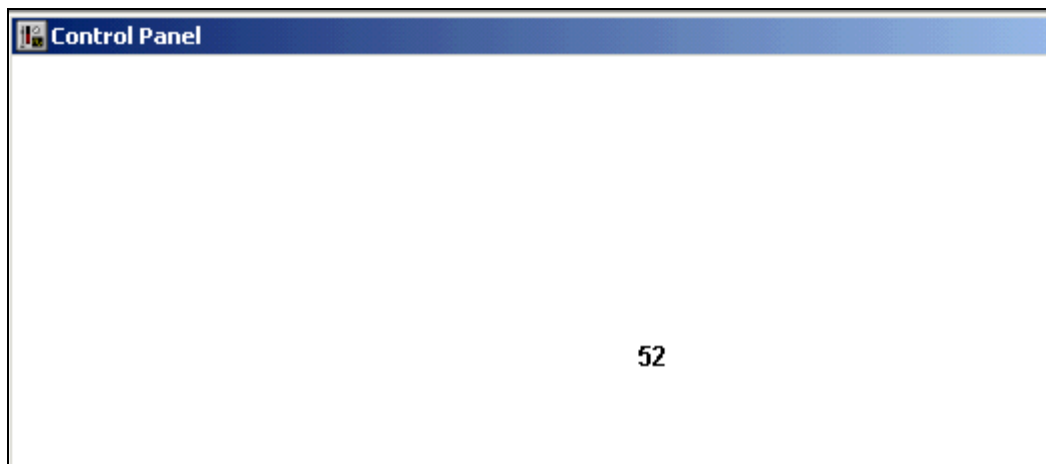


Click OK to accept.



View Live Data

At this time you should be seeing data displayed in the Control Panel.



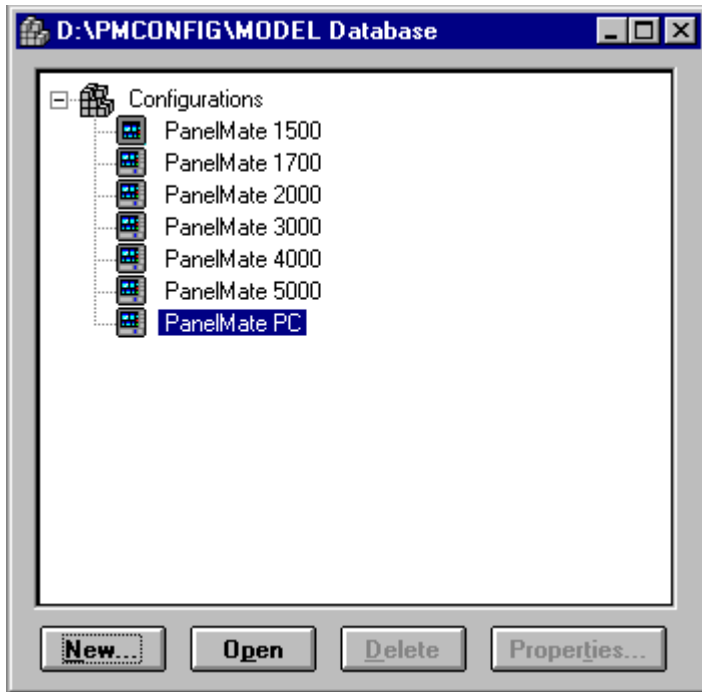
Cutler-Hammer's PanelMate PC Pro as an OPC Client

About PanelMate PC Pro

Cutler-Hammer's PanelMate PC Pro OPC client is one of many clients that can be used to connect to KepserverEx. The PanelMate PC Pro OPC client version used for this example was version 1.10. All PanelMate PC Pro versions 1.10 or greater support OPC client connectivity. The following steps will show you how to create an OPC connection to the Server from PanelMate PC Pro.

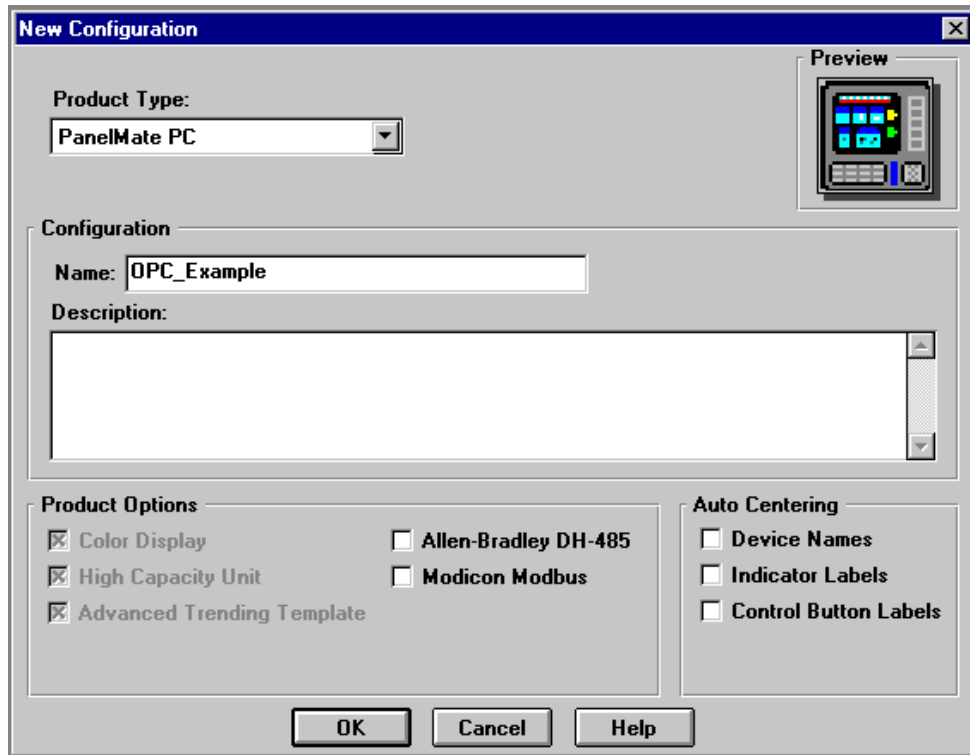
Creating a PanelMate PC Project

1. First, go to the Start menu folder for PanelMate and click on PanelMate Power Pro Software.

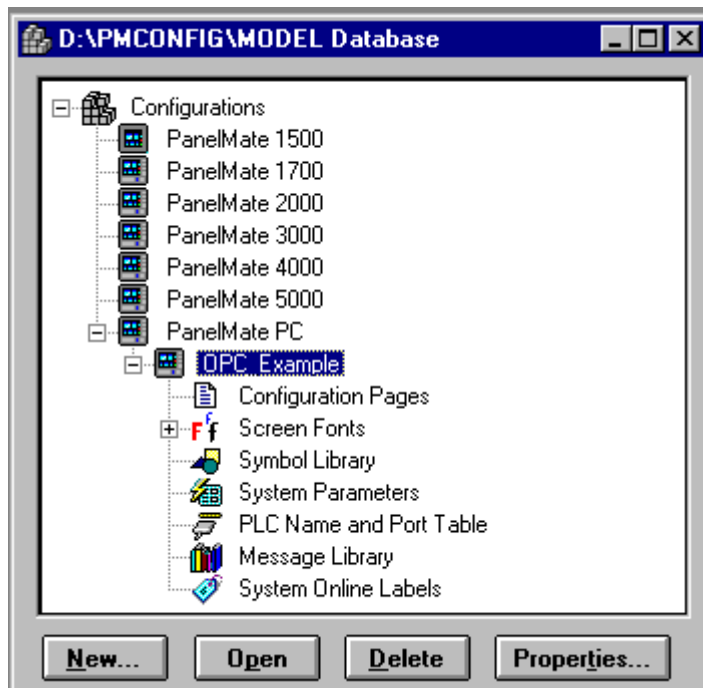


KepserverEx is capable of being an OPC or DDE server to PanelMate. If your node is defined as an OPC node in PanelMate, then KepserverEx will be an OPC server to PanelMate. Likewise, if the node is defined as a DDE node in PanelMate, KepserverEx will be an Advanced DDE server.

2. The PanelMate PC Configuration Editor will open with a default Database model Window.
3. Select the PanelMate PC icon and click the New... button.



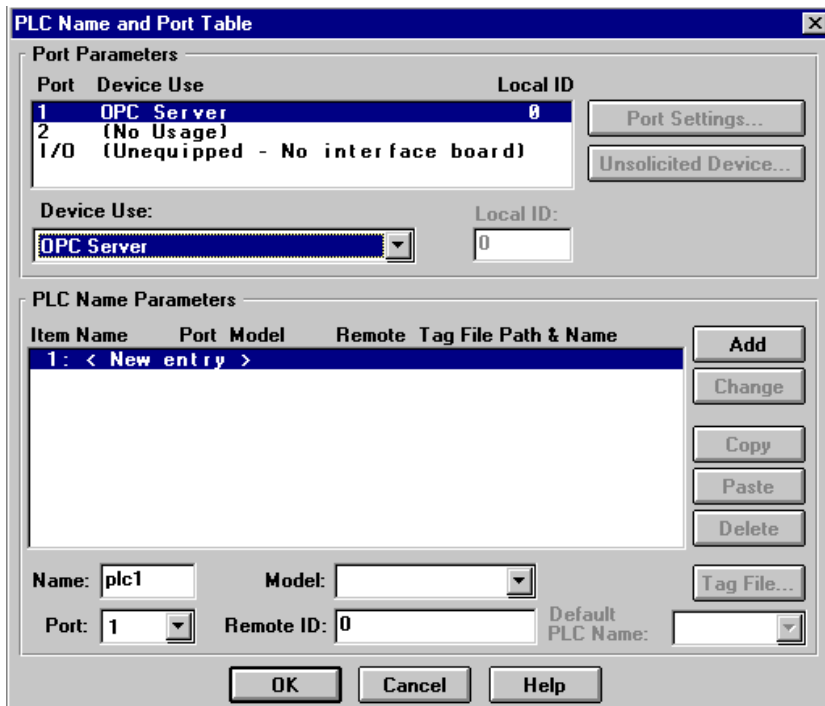
4. The Configuration Name is similar to a project name, so assign a unique name and click ok.



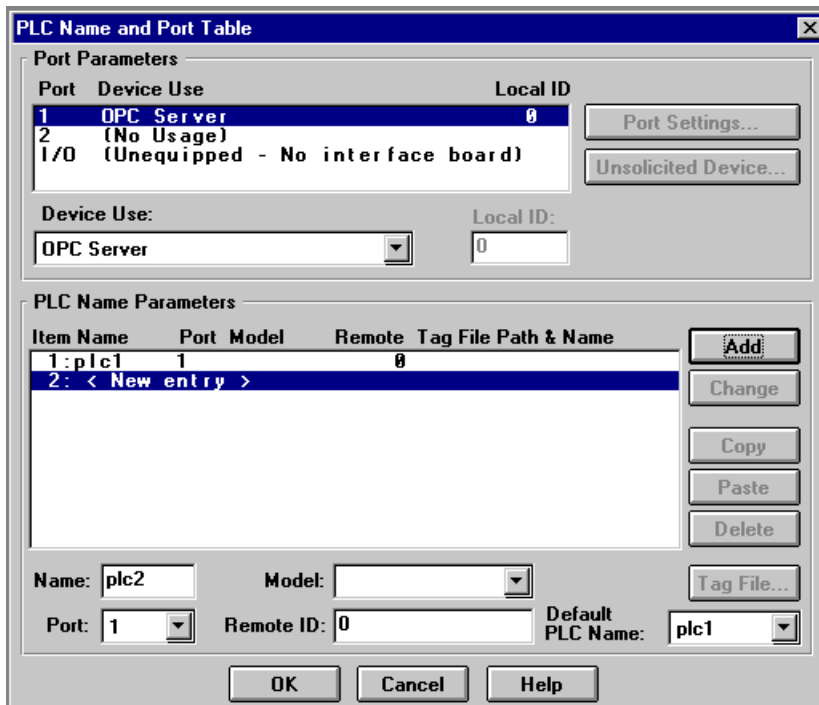
Create a Connection to the Server

5. Click on the + symbol to the left of the configuration name that you just created and drop down the tree view options. Double click on the PLC Name and Port Table option to open and edit the table.

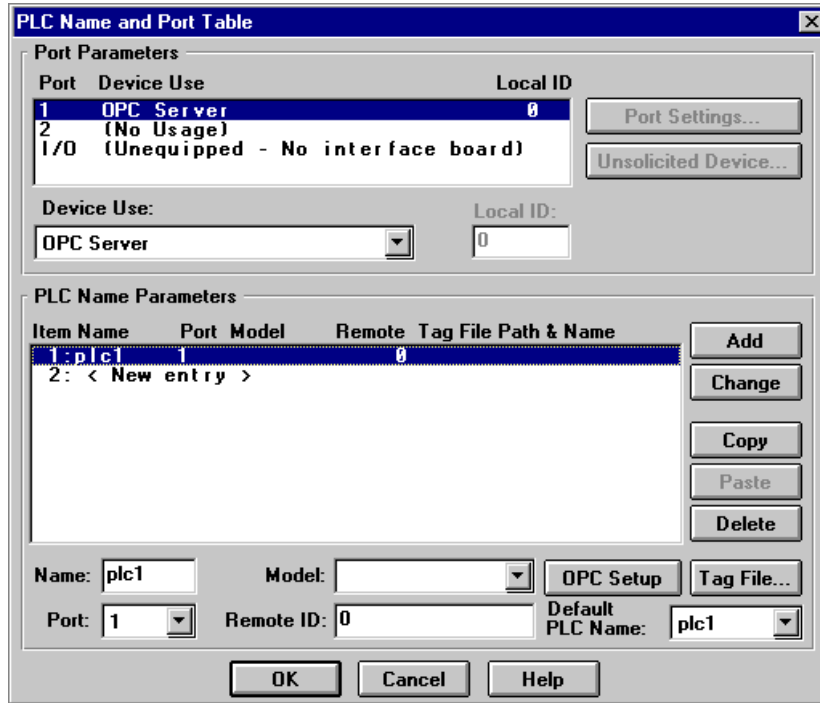
6. Select OPC Server from the Device Use: drop down list located in the middle left side of the window.
7. Port 1 will be displayed as OPC Server in the Port Parameters list at the top of the window.



8. Next, click on the Add button to the middle right of the window. Item 1 will be PLC1 on port 1 in the PLC Name Parameters window.



- Select Item 1 in the PLC Name Parameter list and the OPC Setup button in the lower right will become available.

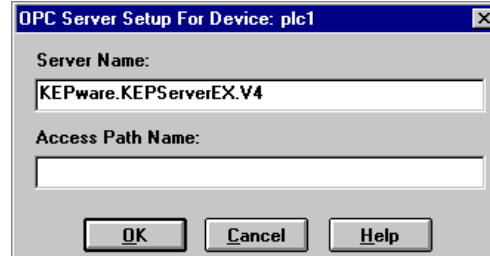


- Click on the OPC Setup button and the OPC Server Setup window will open.

Settings 1:



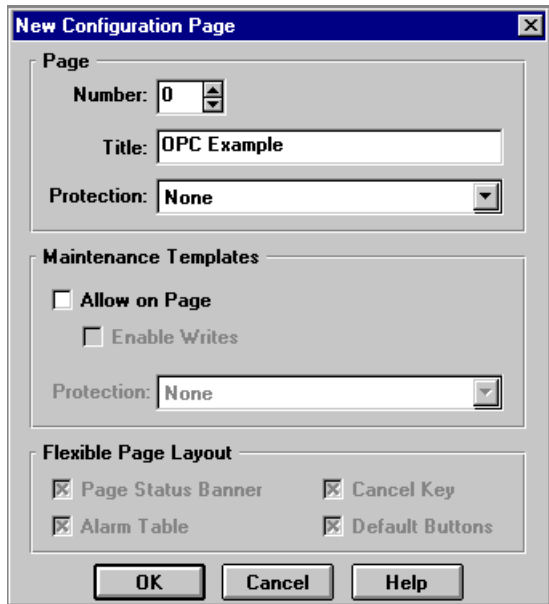
Settings 2:



- Enter "KEPware.KEPServerEx.V4" as the Server Name.
- Next, if you are connecting to only one device from PanelMate then you can enter an Access Path for the device as we did in Settings 1. As you can see the one we entered points to Device_1 on Channel_1.
- If you prefer to assign the Access Path Name in your item references then you will want to leave the Access Path blank as we have done in Settings 2.
- When you finish making the entries click OK to accept them. Then click OK again to close the PLC Name and Port Table window.

Configure a Display Page

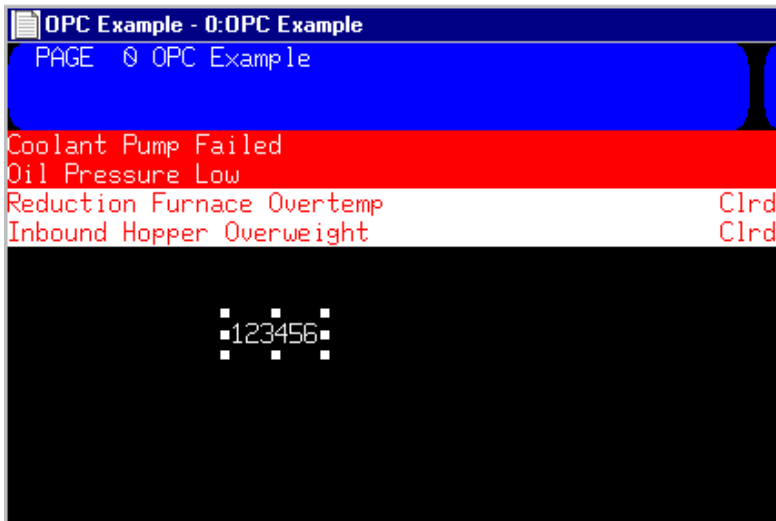
- To configure a window (page), select the Configuration Pages branch and then click on the New button. Give the page a name (Title) and accept the defaults by clicking the OK button. In this example we used a page title of "OPC Example".



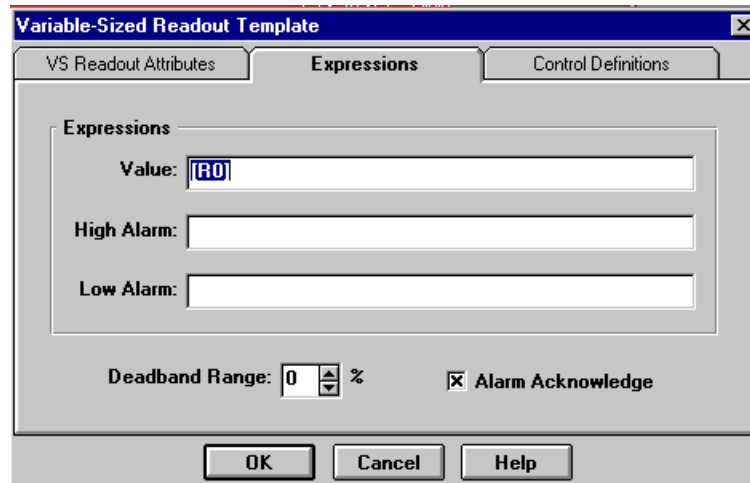
- Next to open the page, double click on the new page that has appeared under Configuration Page in the tree view.



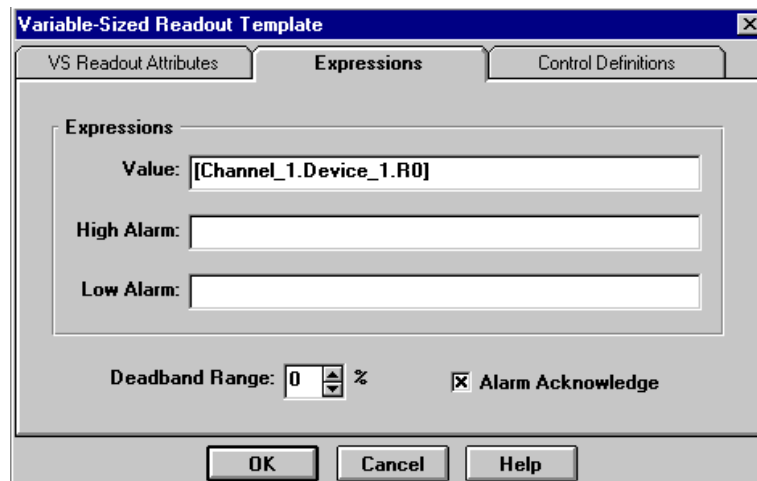
- Click on the “VS Readout Template” button from the bottom middle of the Tool Box and place a template on your page.



18. Double click on the template and enter a valid simulator tag in the Value: field of the Expressions tab. Make sure the item reference is surrounded by square brackets. If you entered an access path when setting up your server connection you would enter “[R0]” as the item.



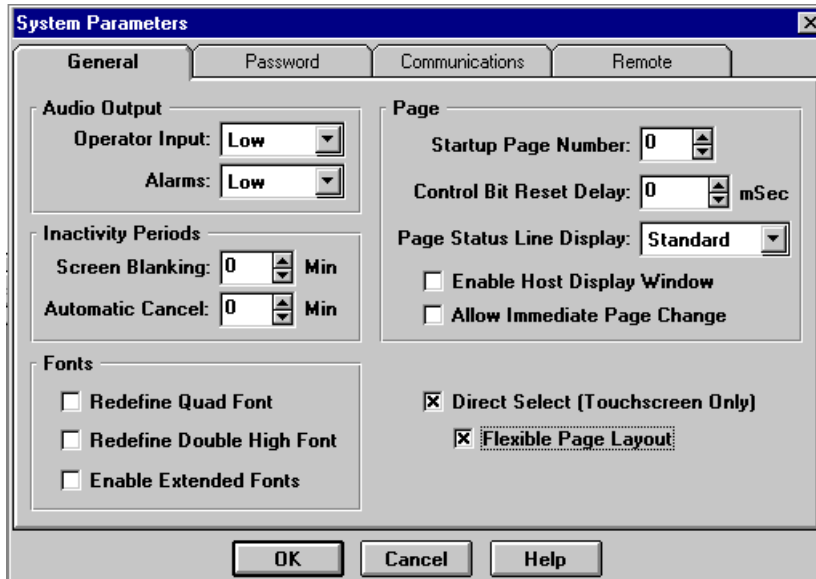
19. If you left the access path field blank in the OPC Server setup (step 11) then you would enter the item as “[Channel_1.Device_1.R0]”.



In both examples we are adding tags to the server dynamically by asking for data addresses that exist without having been defined in the device in the server project. This is fine if you wish to work with the default data type for that address. In our example the tag would be created as a word (also known as a 16bit unsigned integer). Please refer to the [Addressing](#) section of your KepserverEx driver help file for more information on default data types. Tags that do not use the default data type must be defined in the server first to set the appropriate data type. For example, if you needed to look at the R8 register of the Simulator as a float, you would need to define a tag in the KepserverEx and give it an address of R8 and a data type of float. If you had given this User Defined tag a name of “float8” then that is what you would use for your tag reference in PanelMate PC Pro. For more information on User Defined tags please refer to the “Designing a KepserverEx Project” section of the KepserverEx|Help|Contents selection.

20. To reference User Defined tags the format would be “[Tag_1]” or “[Channel_1.Device_1.Tag_1]. Remember, Tag_1 is a tag that was created in the server Simdemo project already.

21. Now click OK to accept the expression you have entered.
22. Next, double click on the System Parameters branch.



23. In the lower right place a check in the boxes for the Direct Select and Flexible Page Layout options then click OK to accept the parameters.

Save your configuration

24. Save your configuration by selecting the Configuration Name branch that resides under the PanelMate PC icon, then go to the File menu and select Save. It will save the configuration with the name you entered when creating this project earlier.

Run your PanelMate PC Pro Project

1. To run a PanelMate PC Pro project you need to export your configuration as a .PPS file. From the file menu select Export and give your PPS file a name (it is easiest to use the same name as was used for your configuration file). Note that although the PanelMate Configuration Editor runs under all Windows operating systems, PanelMate PC Pro requires either a Windows NT 4.0 or Windows 2000 runtime environment.
2. Save the .PPS file in the default directory (Pmconfig\Online\Cfg).
3. Make sure that the KEPServerEx SimDemo project is running, then from your windows explorer double click on your PPS file.




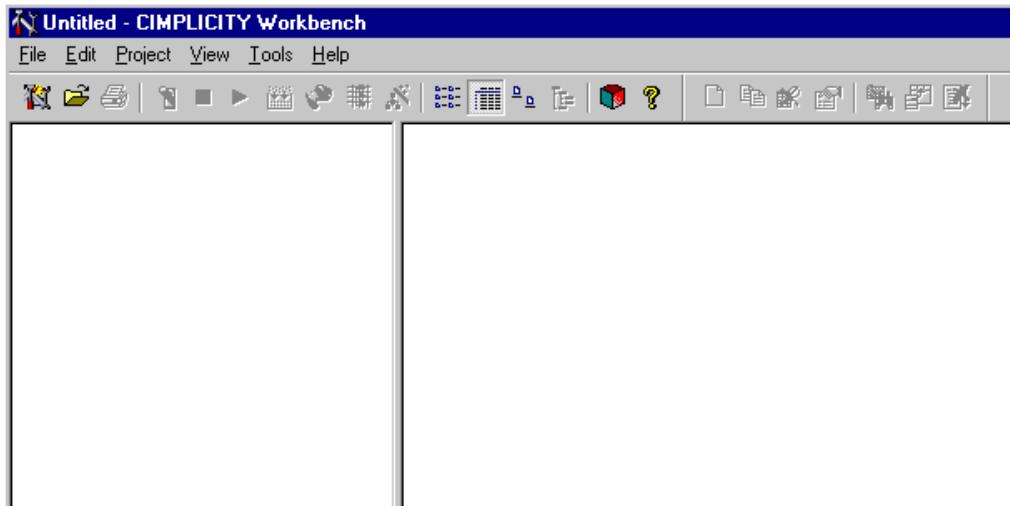
GE's Cimplicity® as an OPC Client

Connect to the Server from Cimplicity V5.0

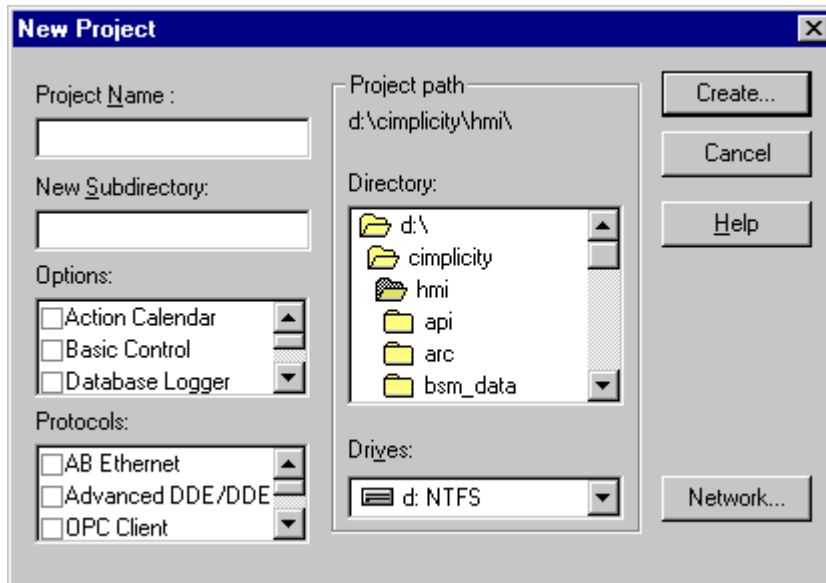
The following steps will show you how to create an OPC connection to the Server from Cimplicity V5.0. Cimplicity's Workbench is the project control center.

Create a New Project

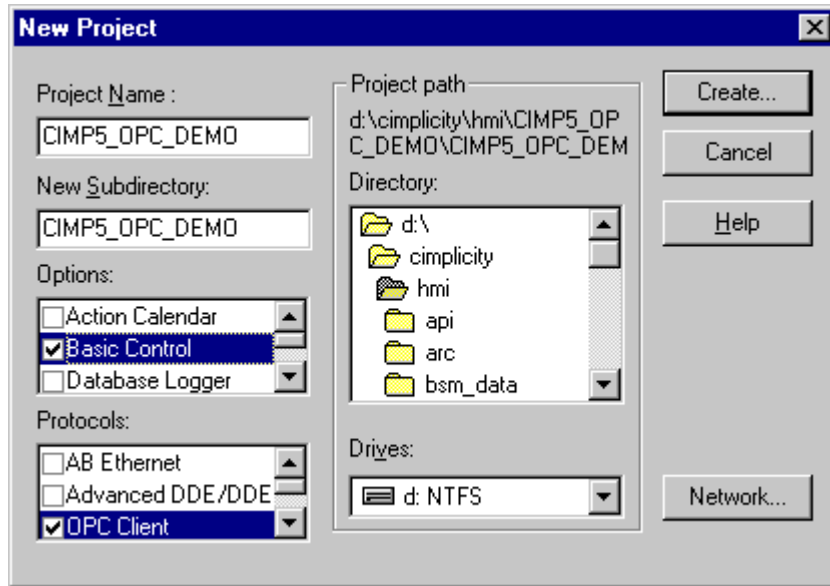
1. Start Workbench and click on the New Project button  or click File|New|Project on the Main Menu.



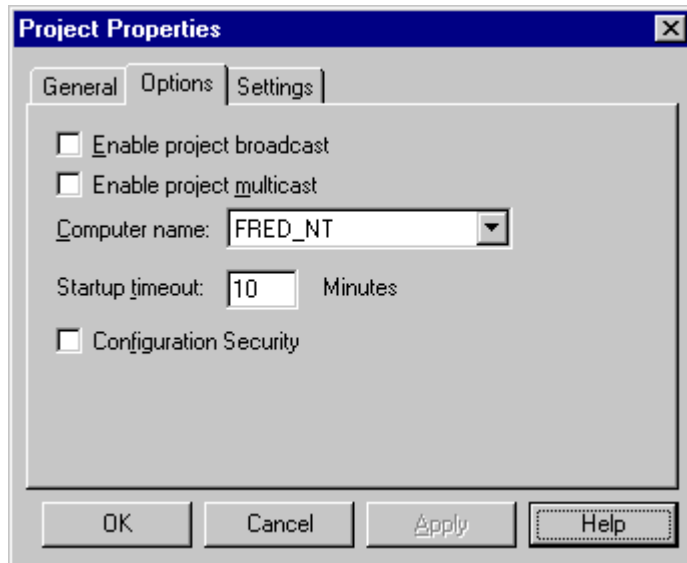
2. The New Project Dialog box will appear.



3. First enter a Project Name for your new project. By default the Sub Directory is auto filled with the Project Name
4. Next select the Options that you wish to include in your project. For our example we chose only Basic Controls.
5. Then select OPC Client as the Protocol you want to use in your project



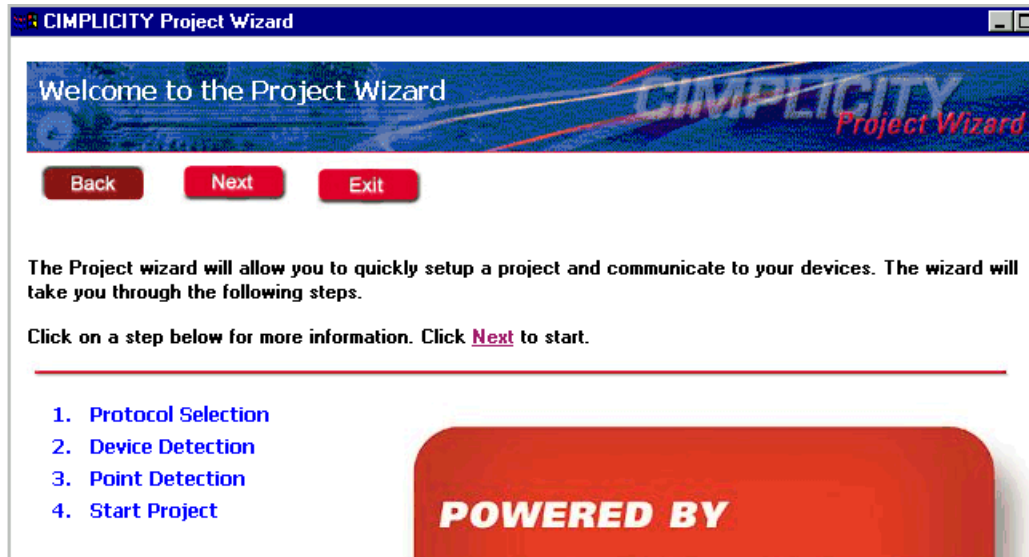
6. Lastly, select the Project Path and click Create... to create your project.



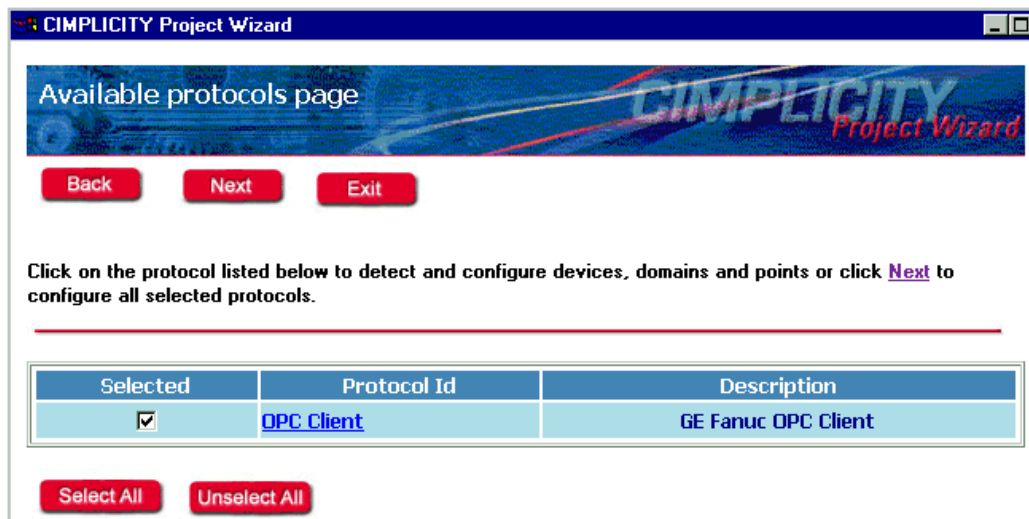
7. The Project Properties dialog box will open. If you plan to provide information from the project to the rest of your network then you will want to check Enable Project Broadcast, otherwise click OK to accept the Project Properties.


Run the Project Wizard

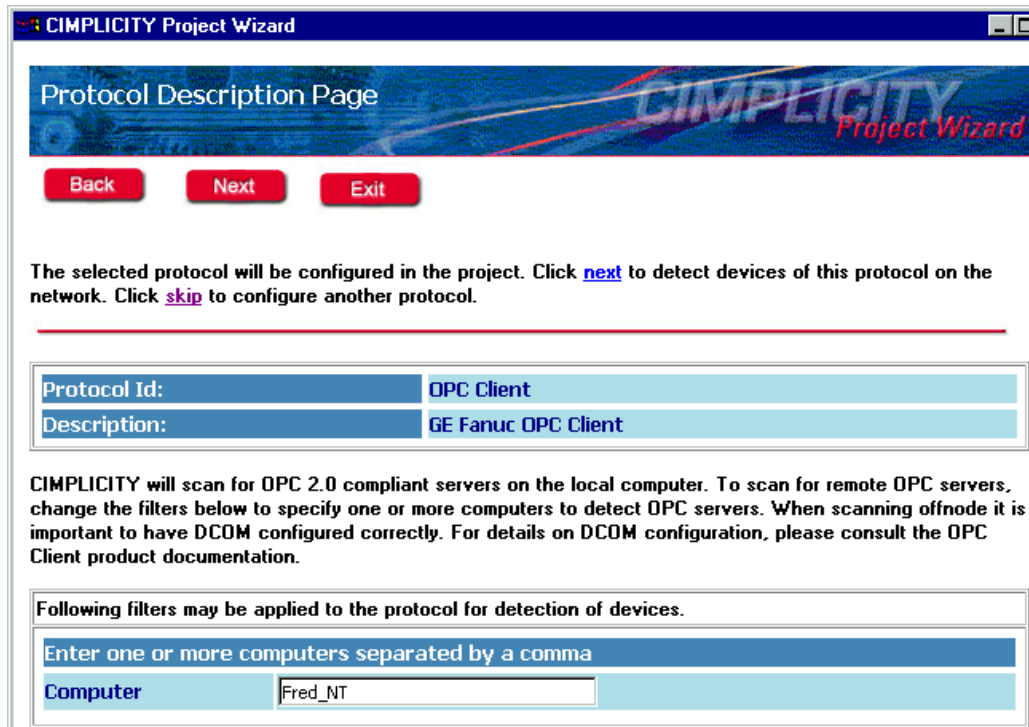
- When you create a new project that has OPC Client protocol enabled the Project Wizard should start automatically. To start the wizard, manually click on Project|ProjectWizard in the Main Menu.




- Click the Next button  to start the first step of the wizard.



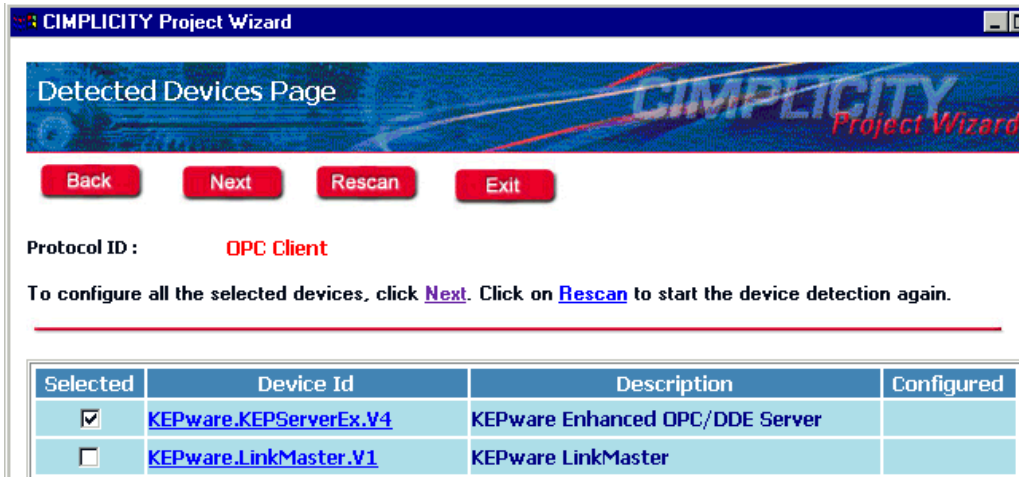
- When we created the project we selected the single protocol of OPC Client which is the protocol we now wish to configure.
- Click the Next button  to configure the selected protocol.




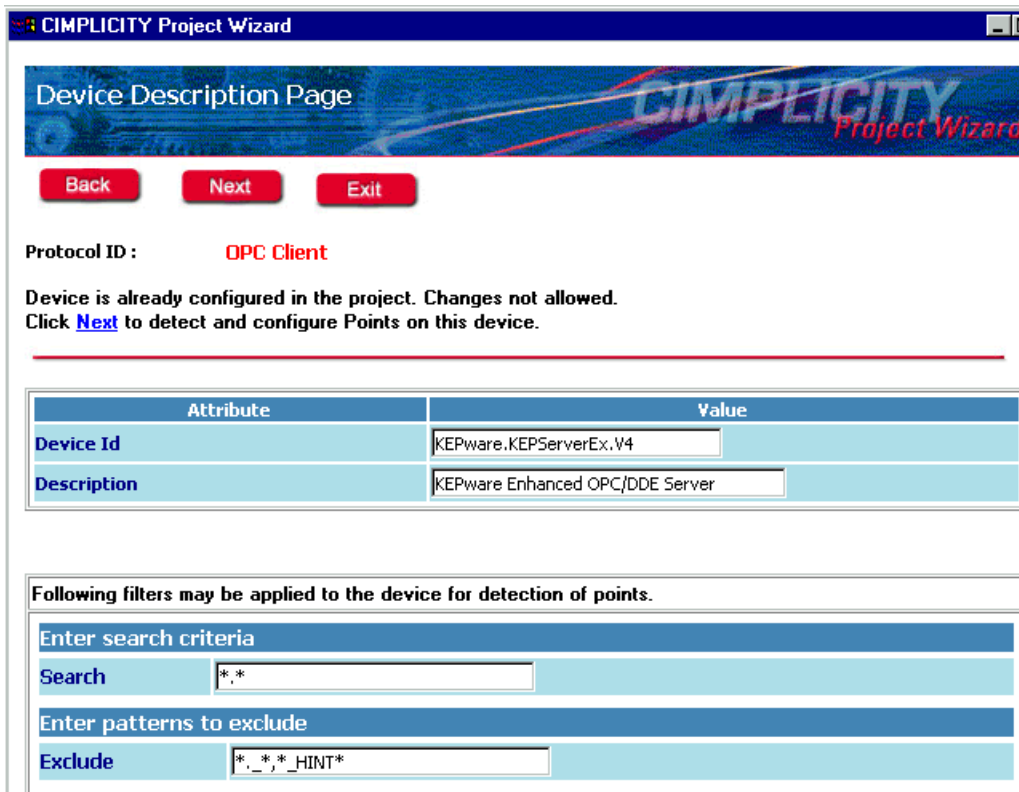
12. Verify that you have selected the correct protocol. You should also see at least your computer named in the list of computers to be searched for valid devices. By default the wizard sets its server browse filter to look on the local PC only for servers. If you want to look on another PC you would replace the local PC's name with that of the remote one. You could also browse for servers on multiple PC's by entering each name separated by a comma. If you doing a remote connection to the server you will need to configure DCOM. See Kepware's DCOM Configuration for KepserverEx guide for details.
13. Click the Next button  to detect valid devices.




14. You will notice that the Project Wizard has detected all of the devices (servers) that are installed and available to it
15. At this point we are only concerned with connecting to and configuring KepserverEx (Kepware Enhanced OPC/DDE Server). Uncheck all of the other devices except it.

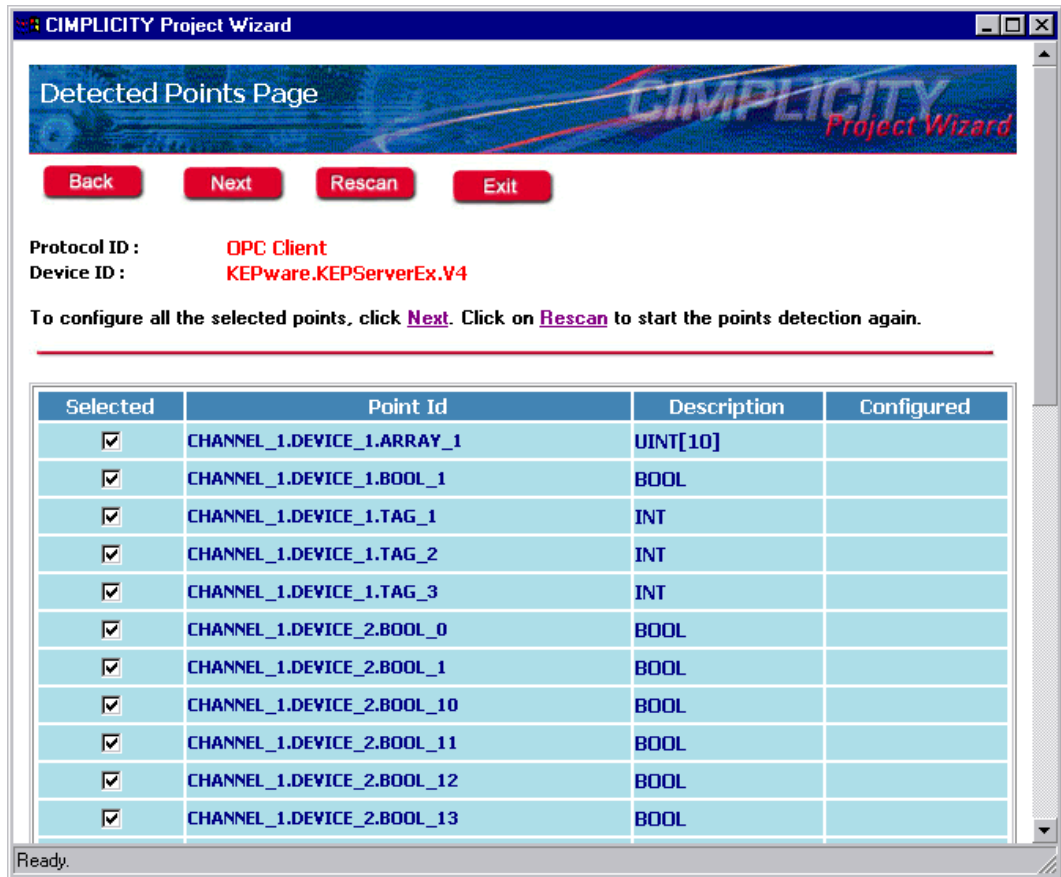


16. Click the Next button  to show the device details.




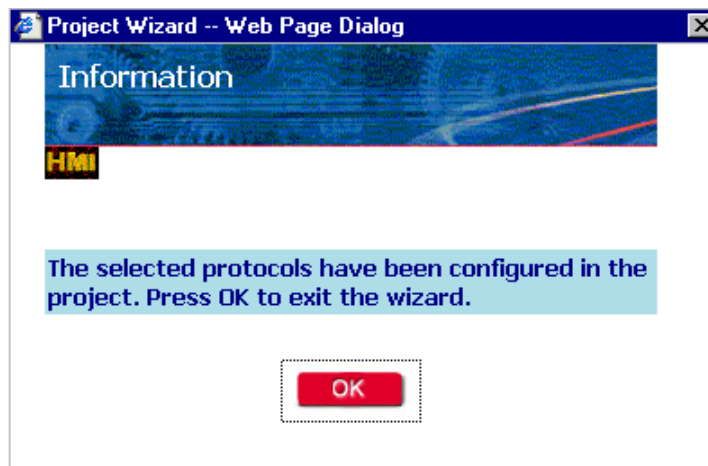
17. Verify that you have indeed selected the correct server. Next enter or edit the search for the points in the device. You may also wish to exclude certain points or groups of points. The Project Wizard was tested with KepserversEx so you can see that by default, the wizard excludes *System tags* and *Hints*.

18. Click the Next button  to detect the points in the server that meet the search criteria and exclusions.




At this point you should see all of the points configured in the server that meet your search criteria. If you do not see all of your tags, then may want to go back one step and re-evaluate your search and exclusion criteria, then select the rescan option to browse the server again. If you add new tags to the server you will want to restart the Project Wizard and rescan the server.

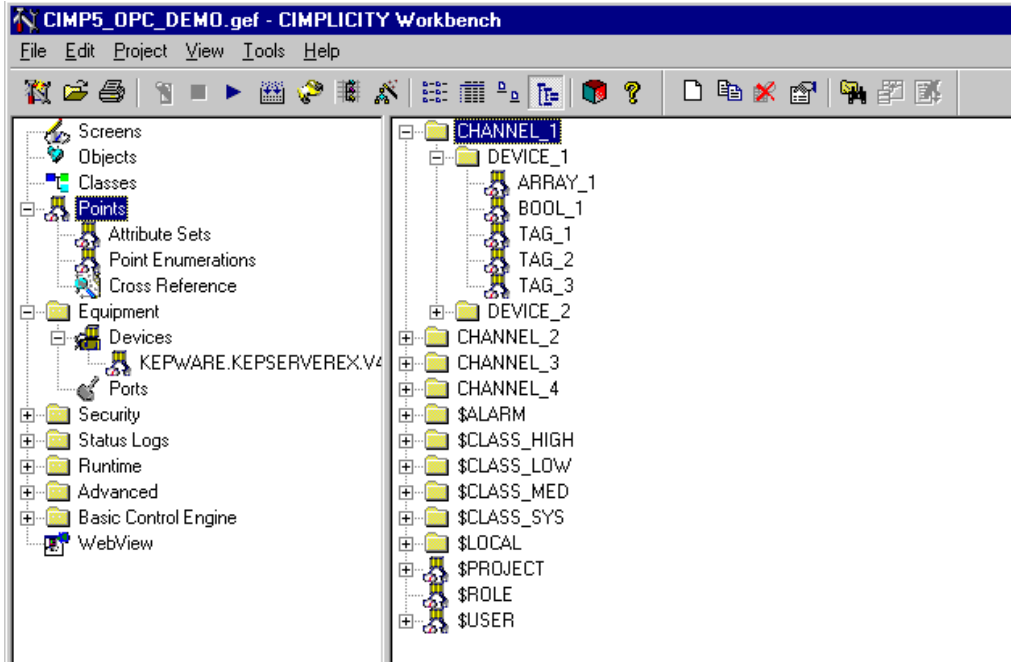
19. By default all of the points will be checked or selected. At this time you may uncheck any points that you do not wish to configure
20. Click the Next button  to configure the selected points



21. Once the Project Wizard has extracted the point data from the server and added it to your project you will see the completion window.


22. Click on the OK button  to close the Project Wizard
23. In the project select Points in the left-hand pane of the project window and expand the folders under Channel_1 in the right hand window to display the points that were added.

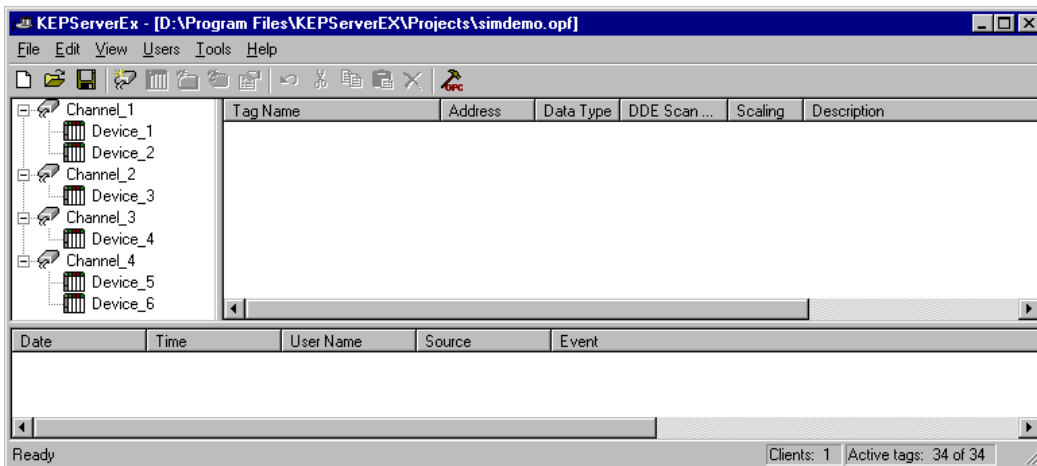
This is the Tree view representation of the “Point ID”. If you were to look at the point “Array_1” in the Detail view instead of the Tree view the point would have a Point ID of “Channel_1.Device_1.Array_1”.



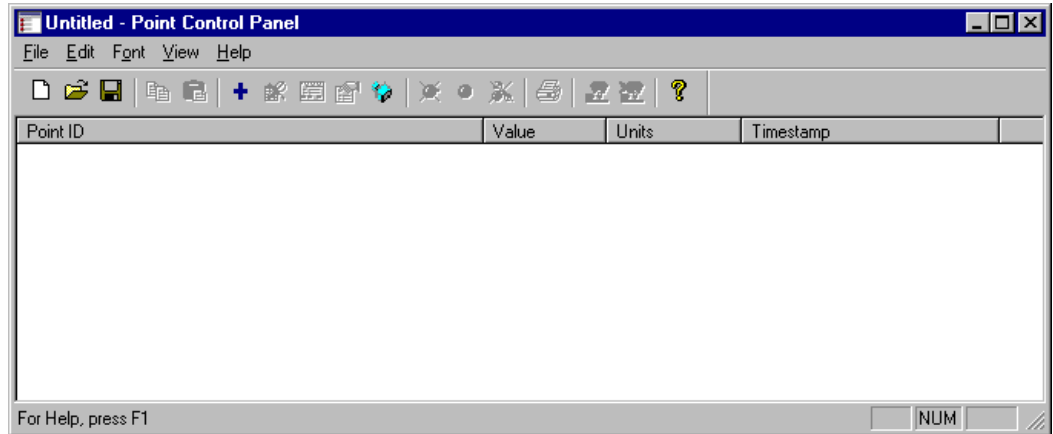
Running the Project and Connecting to KepserverEx

Now is a good time to verify that you can poll data through KepserverEx.

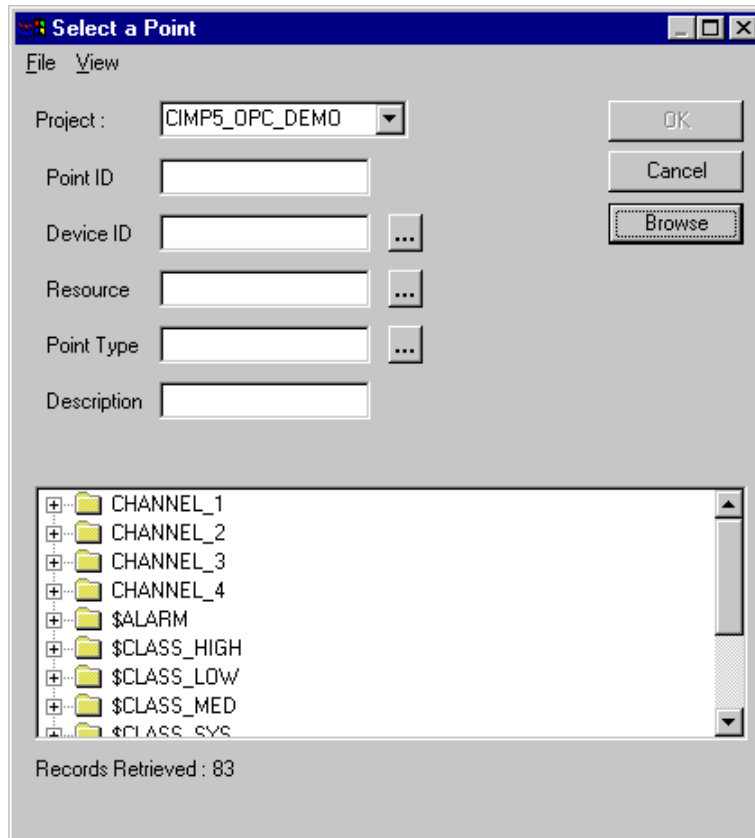
1. Click on the Run button  to start the Cimplicity project
2. Once your project is running, look at KepserverEx and check the Status Bar at the bottom right edge of the server. You should see at least one Client and 1 of 1 Active tags. In our example we have 34 Active tags out of 34 because we selected all of the available tags in our project wizard. If the Connection Status bar is blank, then Cimplicity is not connected to the server.



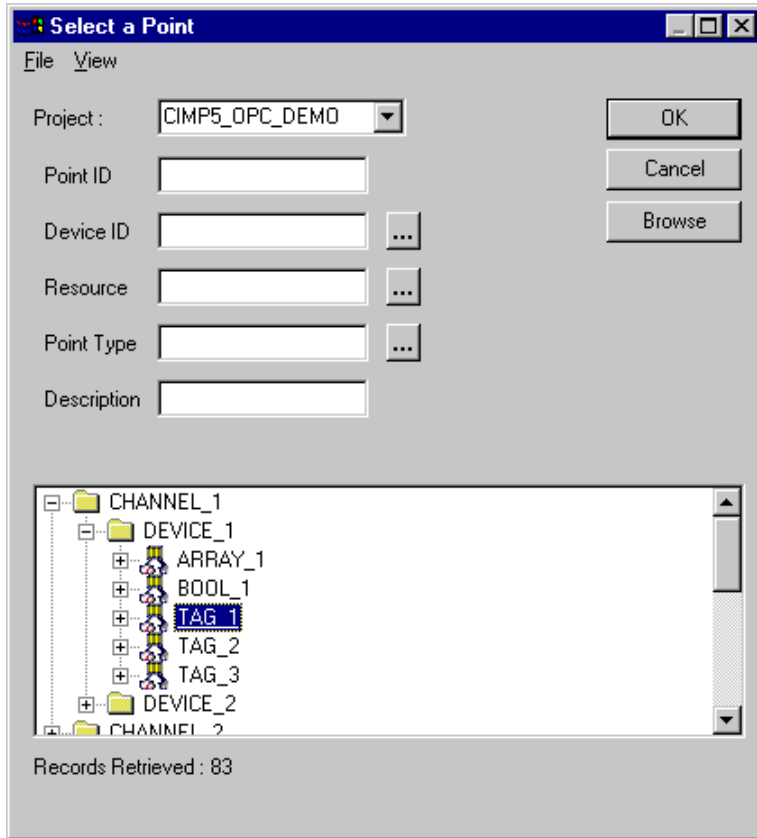
3. Back in your Cimplicity project click on the “+” sign to the left of Runtime in the project tree and double click on Point Control Panel.



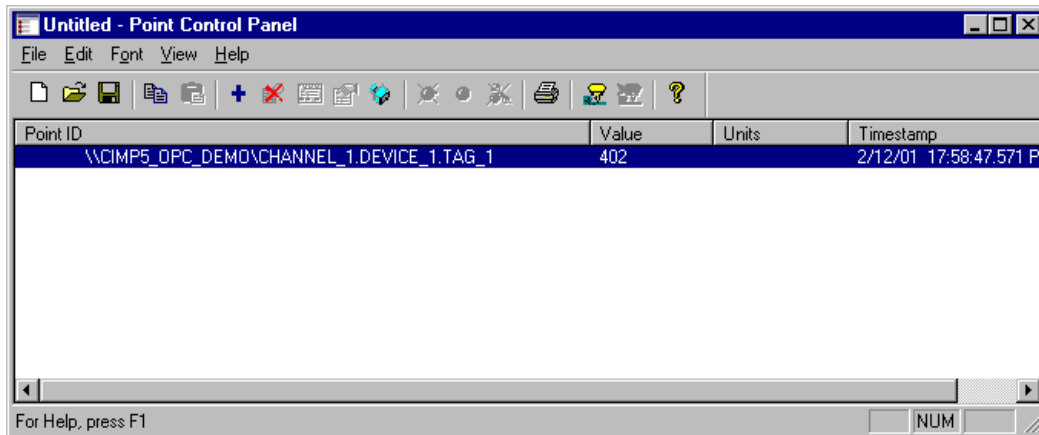
4. Unless you have already done so you will probably be prompted to enter a login before you are allowed to browse the points database and select the point or points you wish to view.



5. In the Tree view of the Point Selection dialog box expand the branches of the tree and select the point or points you want to view and then click OK. As you can see we have chosen point “Tag_1”.



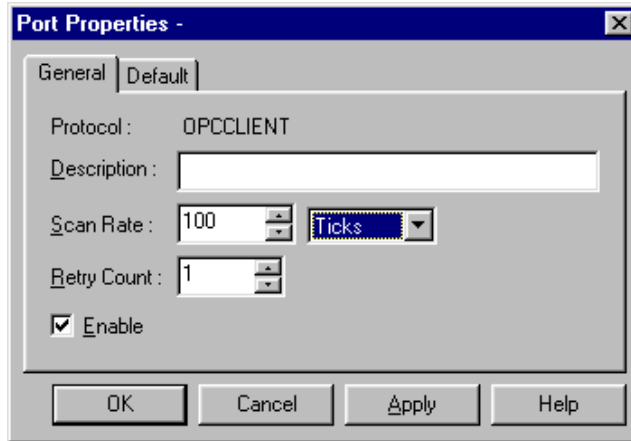
You should now see the point displayed in the panel. In this example we used our sample simulation project called “Simdemo” which simulates changing data for TAG_1. If you are also using the Simdemo project and you are viewing TAG_1, you should see its value ramping up.



You have probably noticed that the point is incrementing once per second. That is because the simulator we are using increments the data values every time they are scanned by a client application. In the next section we will explain how to increase the scan rate (group update rate) in your Cimplicity project.

Optimizing Your Cimplicity 5.0 Project

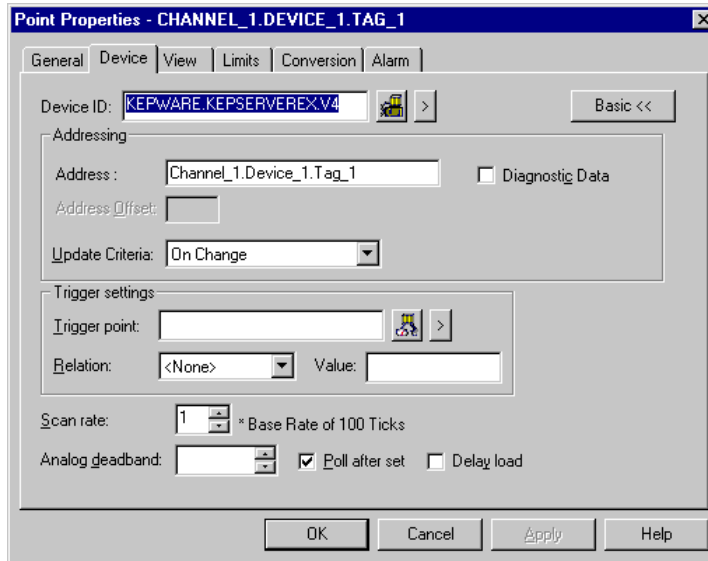
In Cimplicity 4.01 you would set the update and scan rate in the Port Properties by setting the Scan Rate to a certain number of Ticks. Each Tick is approximately 10 msec so in our example the server is scanned and this project is updated at 1000 msec.



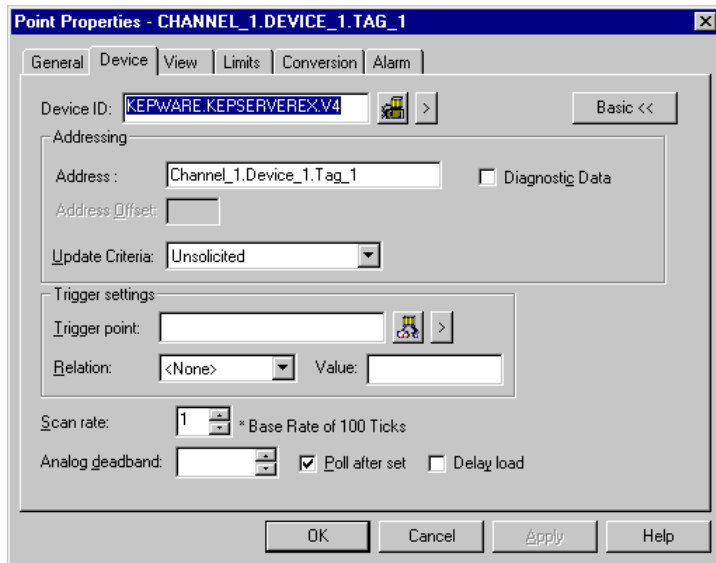
Scan rates are handled a bit differently in Cimplicity 5.0. Cimplicity 5.0 now uses an INI file for each port that allows you to establish the rate at which data is scanned

To explain this a bit we will need to step back for a moment, to when we added the points to our project with the Project Wizard. When the wizard adds the points it uses a default Update Criteria of “On_Change”. This Update Criteria instructs your project to use Synchronous read/write operations with the server

Synchronous Read/Write – The client sends a read or write request to the server and waits for a response before making the next request. If your application reads and writes large amounts of data, the use of Synchronous operations could significantly hinder your application’s performance..



We strongly recommend changing the update criteria to “Unsolicited” which will instruct your project to use Asynchronous read/write operations with the server.



Asynchronous Read/Write – The client sends a read or write request to the server and continues processing. The server processes the request and upon confirmation from the device that it was successful, it notifies the client via the Callback() function.

Note: In the KepserverEx writes are executed as soon as they are received from the client.

Ok, back to the update rate that you set in the Port Properties. That is the rate at which the points are updated from the servers cache when using the “On Change” update criteria. The Scan rate is controlled from an INI configuration file created for each port. This file can be found in the data folder located within the project folder. If you have a port named Master_OPC_0 then you should find a file named “Master_OPC_0.ini”.

; This INI file is for use by the OPC Client devcom provided with CIMPLICITY 4.0.08 or higher, ; ; and CIMPLICITY 5.0.1

[PortLevel]

EightByteReals=0
 UseServerTimeStamp=1
 MessageTicks=50
 CircularLog=1
 UseDataTypePromotion=0

[DEVICE1]

StartupDelay=0
 ReadDelay=0
 PingInterval=5000
 PingTimeout=3000
 AbortShutdown=0
 UseLocalReg=0
 DCOMTimeoutThreshold=10000
 PingBeforePoll=1
 PingBeforeWrite=1
 ForceOPC1Server=0
 HRBothActive=0

[DEVICE2]

StartupDelay=0
 ReadDelay=0
 PingInterval=5000
 PingTimeout=3000
 AbortShutdown=0
 PingBeforePoll=1
 PingBeforeWrite=1

[DEFAULTPOLL]

ScanRate=1000
 NoAccessPath=0
 DeviceReadAfterSet=1

[MYGROUP01]
ScanRate=1000
DeviceReadAfterSet=1
NoAccessPath=0

[DEFAULTUNSO]
ScanRate=1000
DeviceReadAfterSet=1
NoAccessPath=0

You will immediately notice that your file may not look like the one we have displayed here. In our example we have removed all of the comments.

When you use the “On_Change” update criteria, if you change the ScanRate value under [DEFAULTPOLL] it will increase or decrease the number of times your project requests data from the server.

Our demo project currently has an update rate of 100 Ticks (1000Msec) with update criteria of “On_Change” and a scan rate of 1000Msec. This means that we see one update for every scan. If we were to change the scan rate to 100Msec then we would update our data in the project once for every 10 scans.

If you decide to stay with this configuration then you will want to adjust the two scan rates so that you update the client more frequently

As we said earlier if you are reading large amounts of information from the server then you will want to change the tag update criteria to “Unsolicited”. This update criteria uses a different set of defaults and updates as fast as the scan rate. You will want to change the ScanRate under [DEFAULTUNSO] to optimize your processing. Remember that the project is updated at the same rate that the scan occurs.

Check your KepserverEx driver help file for additional ways to optimize the server’s performance and your Cimplicity user manual for other ways to optimize Cimplicity’s performance.

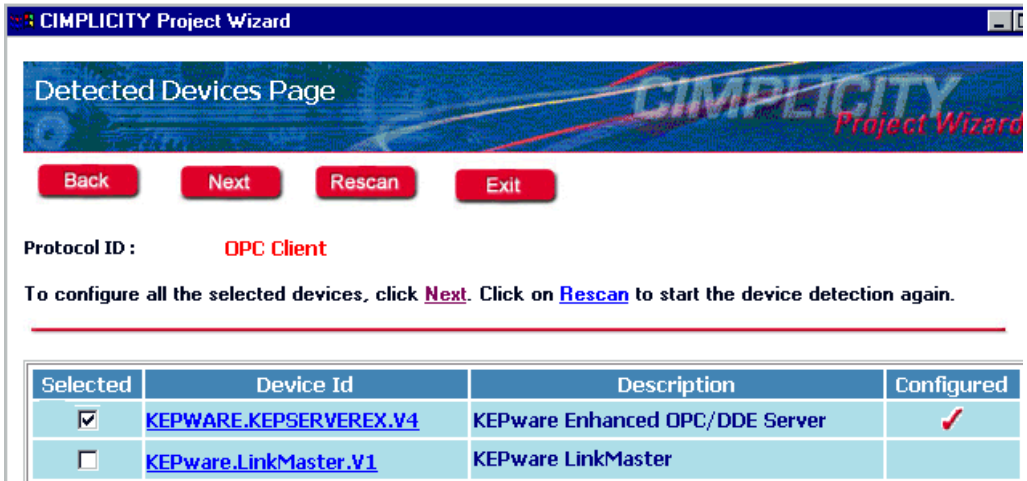
Adding Additional Points to your Project



Suppose your project has been running for several months in its current configuration but you need to add more points to it to monitor new hardware. The easiest way to do this is to use the Project Wizard to rescan the server for the new tags or, you can add them one at a time if you like

1. Click on Project|Project Wizard in the Main Menu to start the wizard. You will notice that when you get to the Detected Devices page that it lists KepserverEx as configured and leaves it unchecked.




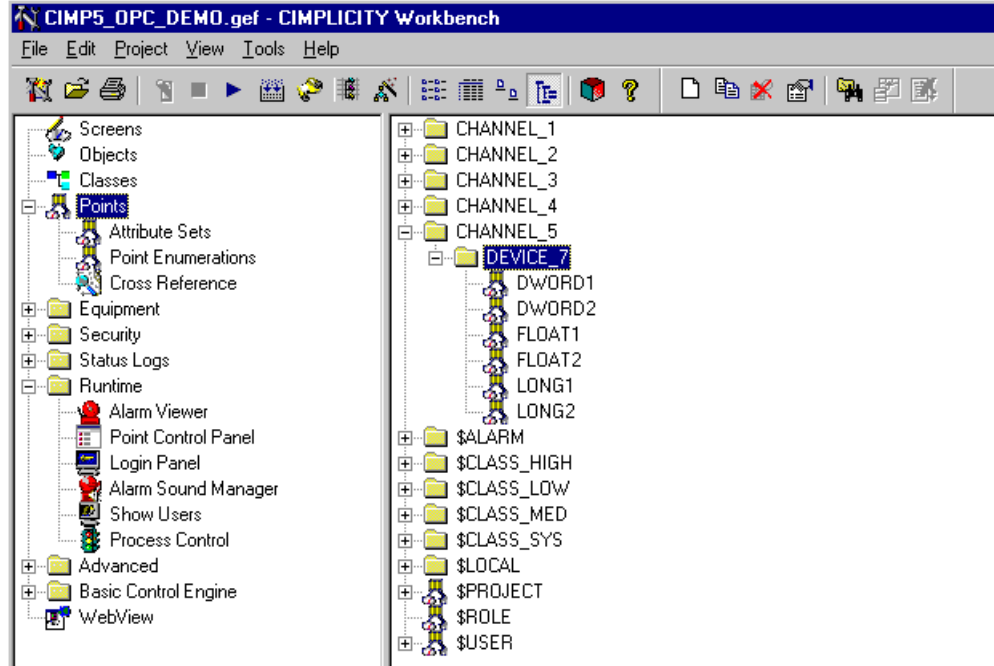
2. Deselect all the devices you do not want to configure and make sure that you re-select the KepsServerEx.



3. Click the Next button  to verify the device description and search criteria then Click the Next button  to detect the points in the server that meet the search criteria and exclusions.



- The wizard will detect all of the points that are currently available and it will check the points that are not currently configured in you project. Uncheck the points that you do not wish to add to your project and click the Next button  to configure the selected points.
- Once the points have been added click OK in the completion window to close the wizard. Click View/Refresh in the main menu to refresh the project display and verify that your tags have been added.



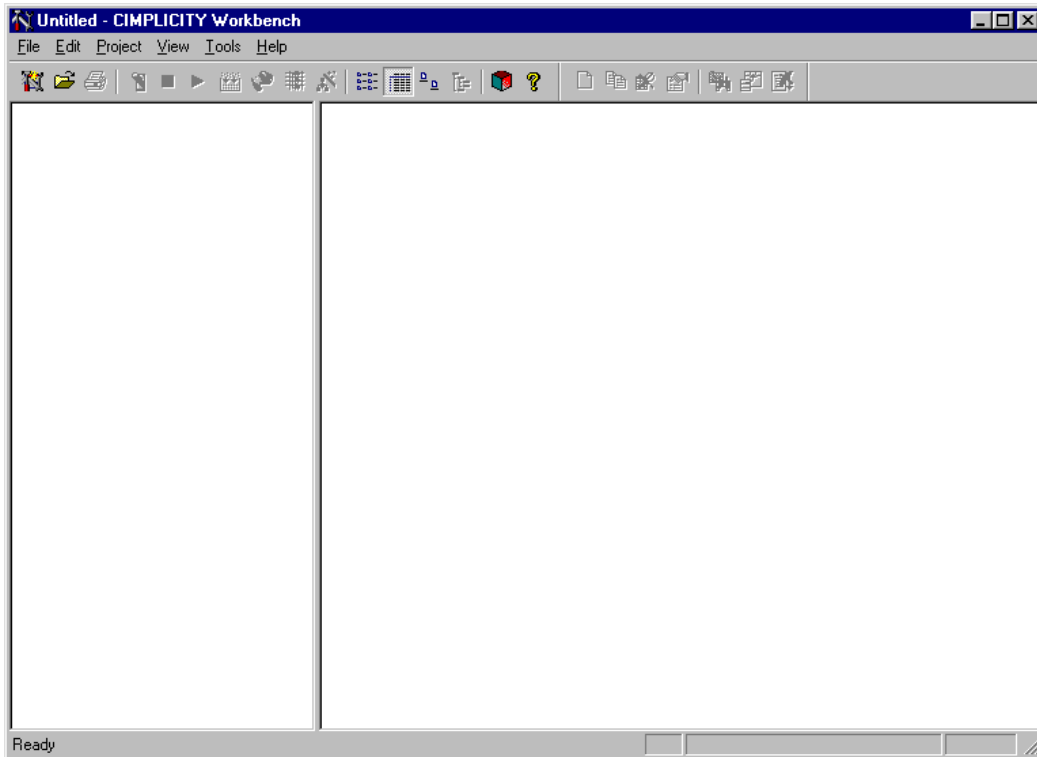
- If you have recently upgraded from Cimplicity Version 4.01 to Version 5.0 you will be able to add tags to your project in the same way.

Connect to the Server from Cimplicity V4.01

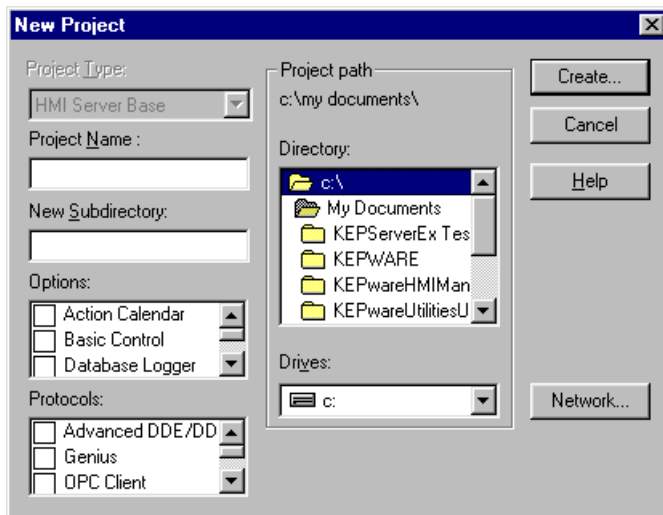
The following steps will show you how to create an OPC connection to the Server from Cimplicity V4.01. Cimplicity's Workbench is the project control center.

Create a New Project

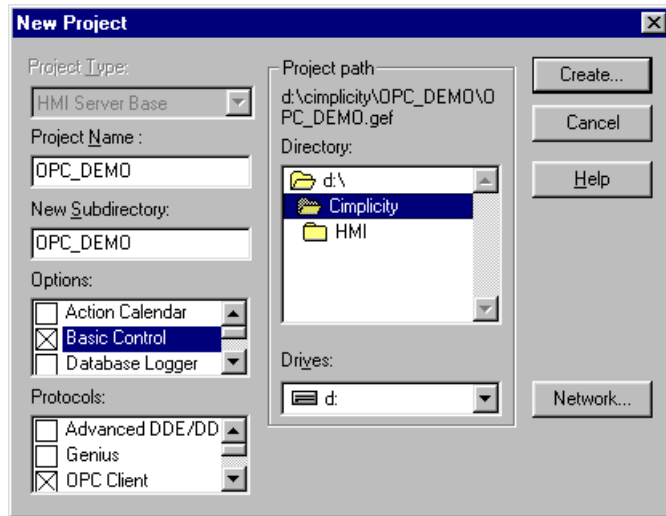
1. Start Workbench and click on New Project.



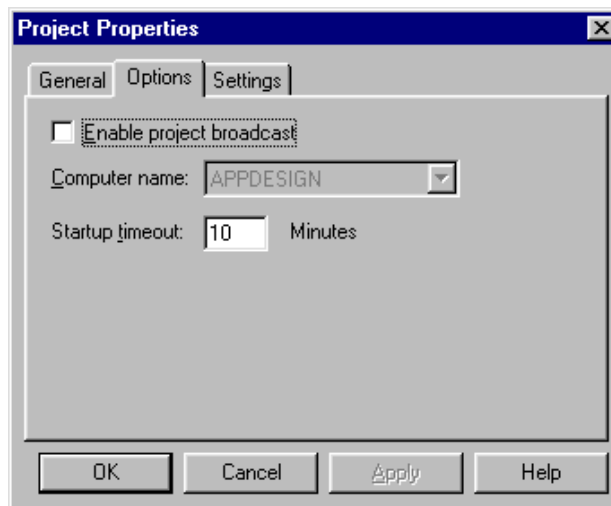
2. The New Project Dialog box will appear.



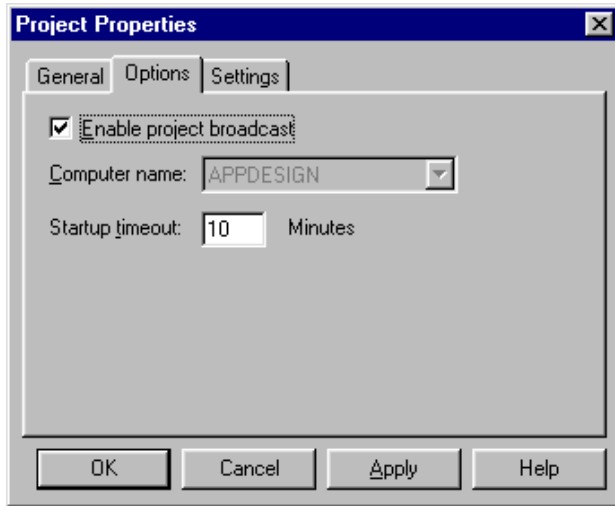
3. First enter a Project Name for your new project. By default the Sub Directory is auto filled with the Project Name.
4. Next select the Options that you wish to include in your project. For our example we chose only Basic Controls
5. Then select OPC Client as the Protocol you want to use in your project.



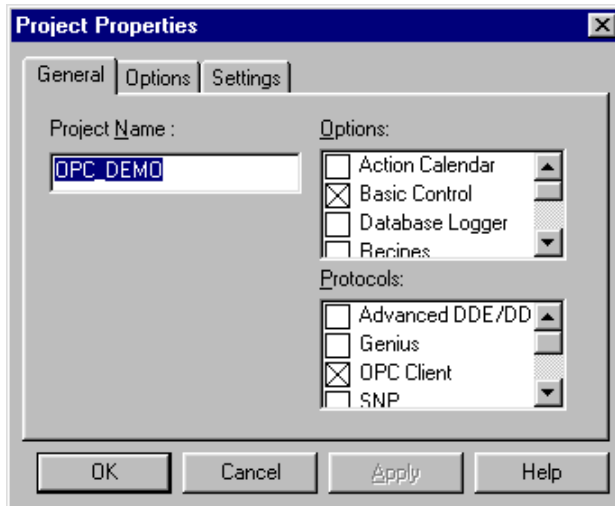
6. Lastly, select the Project Path and click Create... to create your project.



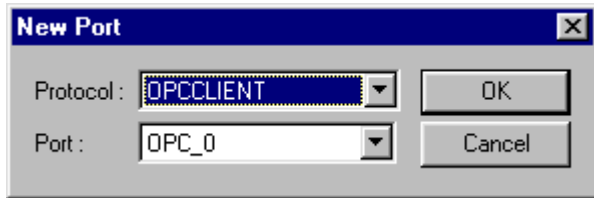
7. The Project Properties dialog box will open. If you plan to provide information from the project to the rest of your network then check Enable Project Broadcast.



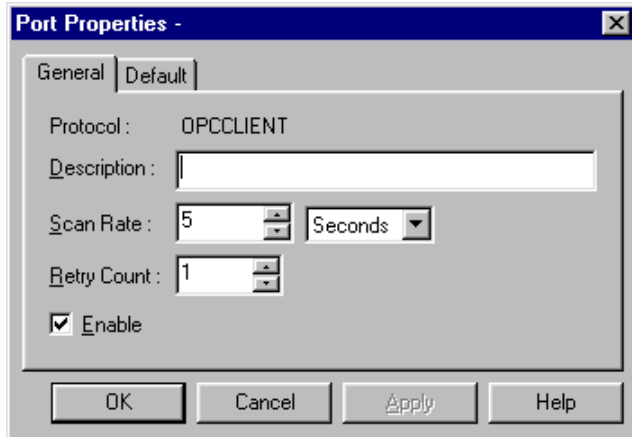
8. You can modify any general project options by selecting the General Tab. In our example we accepted the defaults



9. Selecting the Settings Tab allows you to setup specific areas of your project. In our example we accepted the defaults



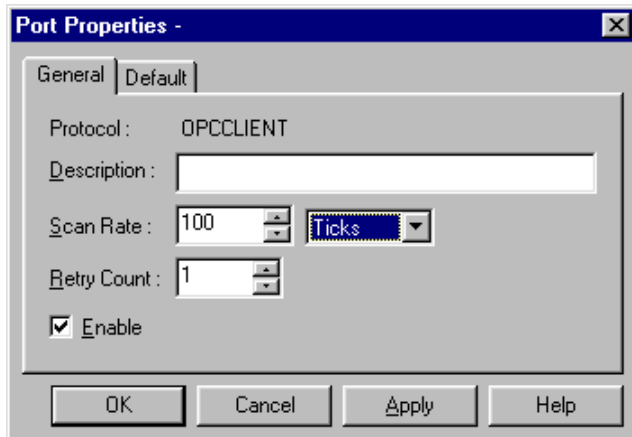
14. Click OK to accept



15. In the Port Properties dialog box you can enter a Description of the new port to help identify it.

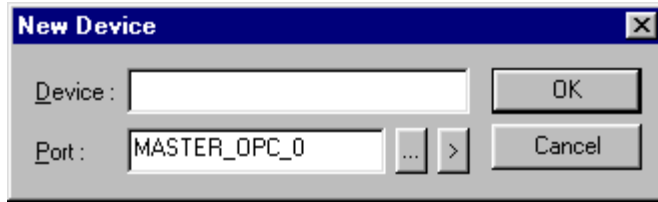
16. Enter a base Scan Rate for the port.

17. Click Enable to enable the port and then click OK.

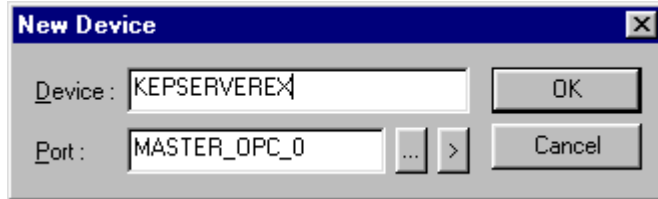


Create a Device

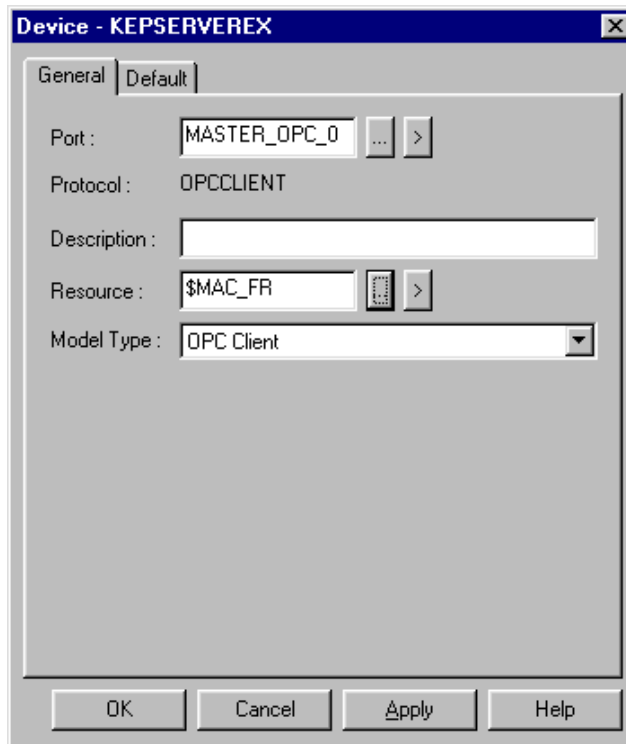
18. In the project tree view click expand Equipment then right click on Devices, click New to add a new Device for the OPC connection.



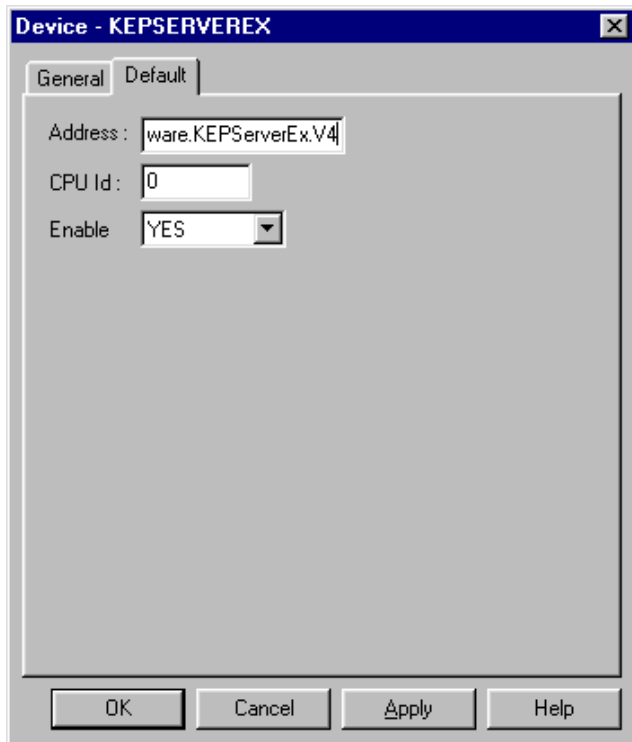
19. In the New Device dialog box enter a Device name for the new device. We chose KEPSERVEREX.
20. If you have created more than one Port then be sure to select the proper one for this device now and click OK.



21. In the Device Properties dialog box you may add a Description for the device.

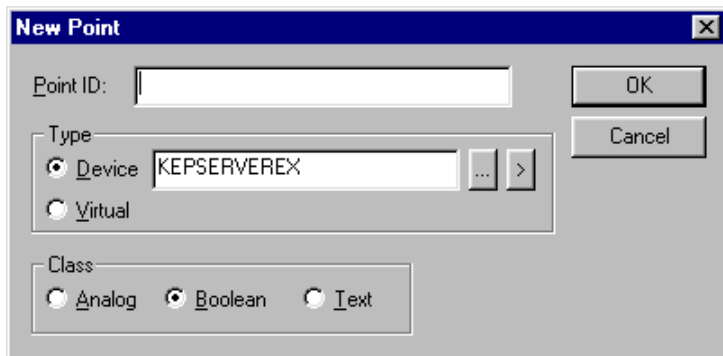


22. Next select a Resource to log device events to, then click the Default tab. See your Cimplicity user manual for details.
23. In the Default tab assign an Address to the device. You will need to enter "KEPware.KEPServerEx.V4". This is the OPC Server id.
24. Next be sure to set Enable to Yes and click OK.

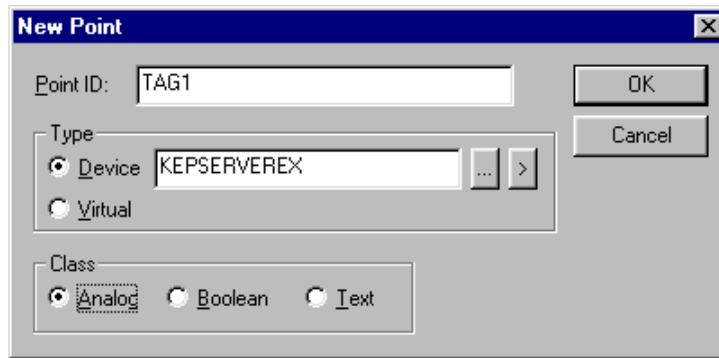


Create a Data Point

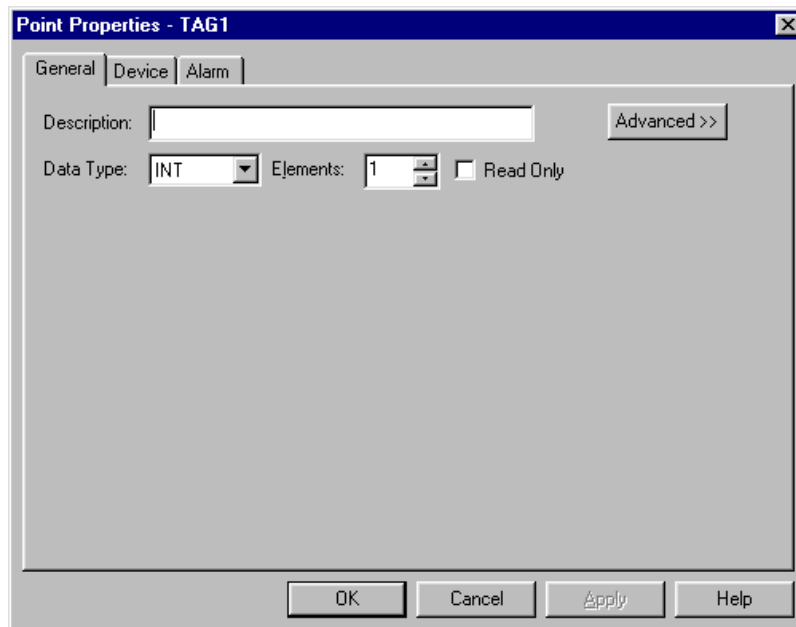
25. In the project tree view click expand Equipment then right click on Points, click New to add a new Device for the OPC connection.



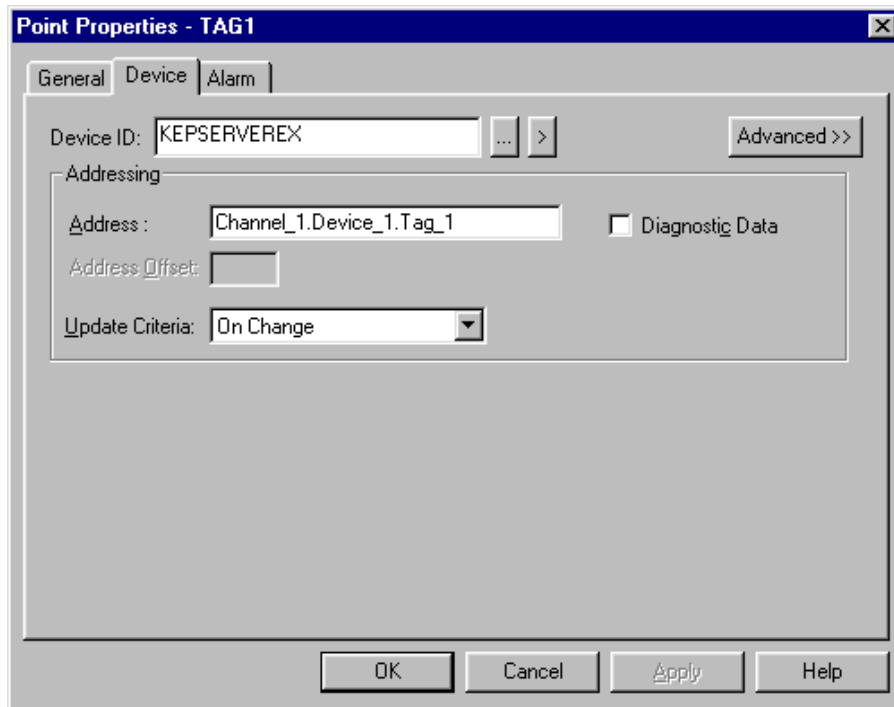
26. Enter a Point ID to identify the point
27. The Type of point will be Device. If you have created more than one device be sure to select KEPSERVEREX.
28. Select a Class for the point and then click OK. We are creating an Analog tag for our example



29. In the Point Properties dialog box you may enter a Description for the point.
30. Next select the Data Type, if the point is for display only, click Read Only and then click on the Device tab.



31. If the proper Device ID is not selected then do so now.
32. Next enter the Address of the OPC point. We are using the KepserverEx SimDemo project so the address for this point will be "Channel_1.Device_1.Tag_1".
33. Lastly, select the Update Criteria for the point. We selected On Change so that we will see the newest data all of the time.
34. Now click OK to add the point and you are ready to use it in your project.



Check the OPC Connection

35. As a quick test, if you select run in the project it will connect the points to the server. Next, check the Status bar at the lower right edge of the KepserverEx window, you should see at least one Active Item. You can also view the data points in a Process Control Panel.
36. Now click OK to add the point and you are ready to use it in your project.

Upgrading a Project from Kepserver V3.2 to V4.0

Some user may be upgrading projects that were using the Kepware legacy server V3.2. If you are one of these users then you will want to read the following steps closely to ensure a quick and easy upgrade to the new server. Since you are upgrading your project to use the Kepware Enhanced OPC/DDE Server you have probably have already seen that the Kepserver model was redesigned. This was done to take advantage of improved technology and methods.

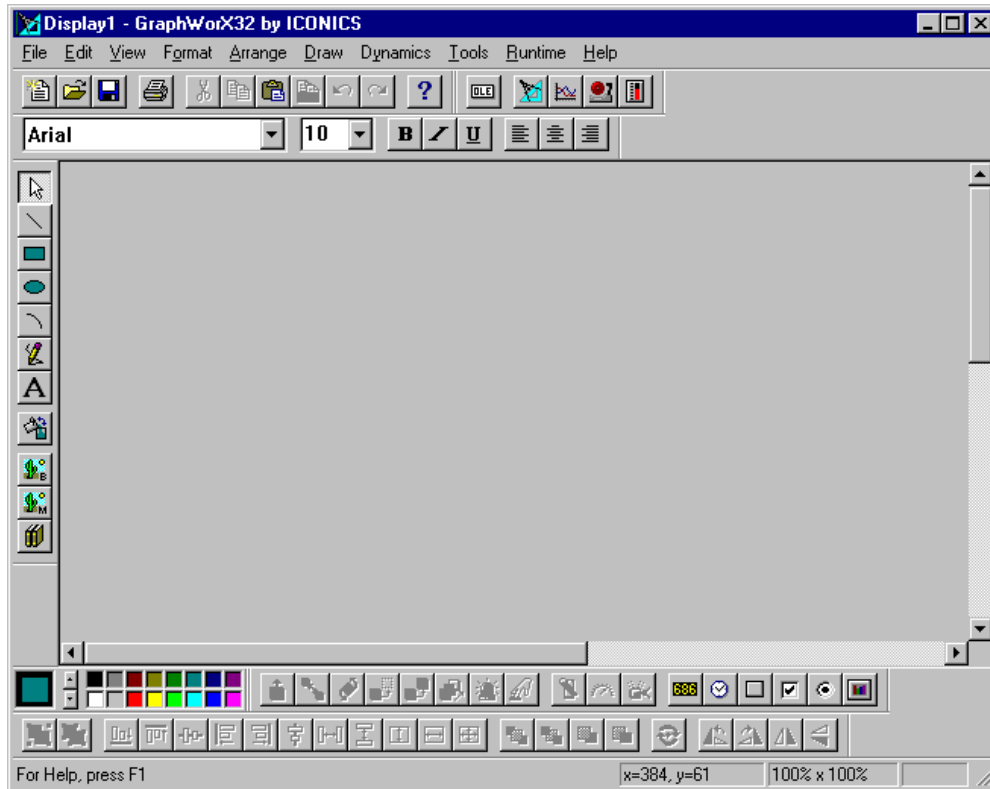
1. The biggest impact to you as a user is the change in the way you connect to the server.
2. Once you have installed KepserverEx on your PC start it and open your old server project file in the new server and save it. Your project is now running you no longer have to turn it on and off line.
3. Next you will want to open your Cimplicity project and select you the device that you created and edit its Device properties.
4. Select the Default tab and edit the device address.
5. You will notice that the old ID is “Kepserver”. The new server’s address is “KEPware.KEPServerEx.V4”. Enter the new address and click OK to accept the changes.
6. You should now be able to run your project and connect to the new server.

Iconics' Genesis32® as an OPC Client


Connect to the Server from Genesis32 Components

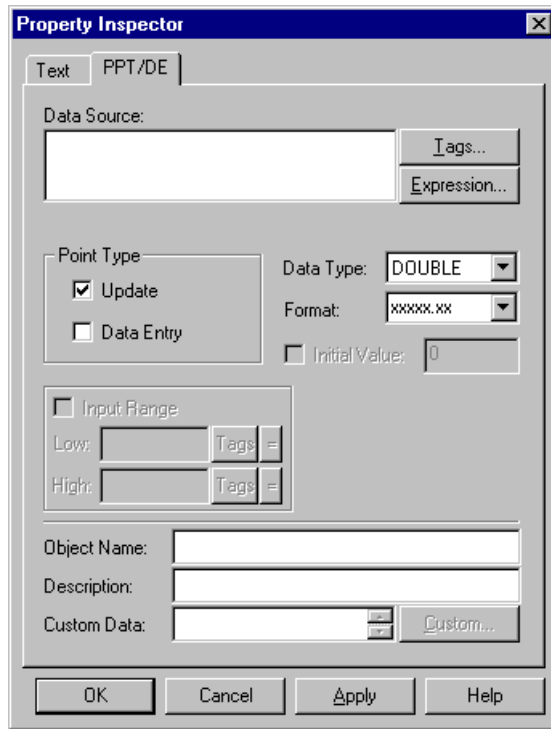
The following steps will show you how to create an OPC connection to the KepserverEx from GraphWorX32. The Genesis Version used in this example is Version 6.01.

1. Start GraphWorX32 from the start menu.



Select a Process Point

2. In the new display window select Process Point  and place it in the window.
3. The Property Inspector for the object will open automatically.

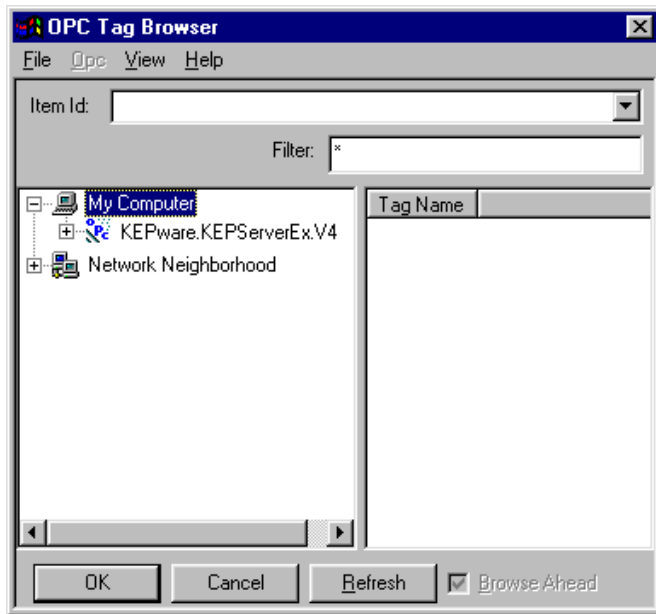


Browse for an Available Server

4. In the Property Inspector click on Tags... to select an OPC tag for the object



5. In the OPC Tag Browser window select and expand the “My Computer“ tree view for a list of the available OPC Servers on your PC.

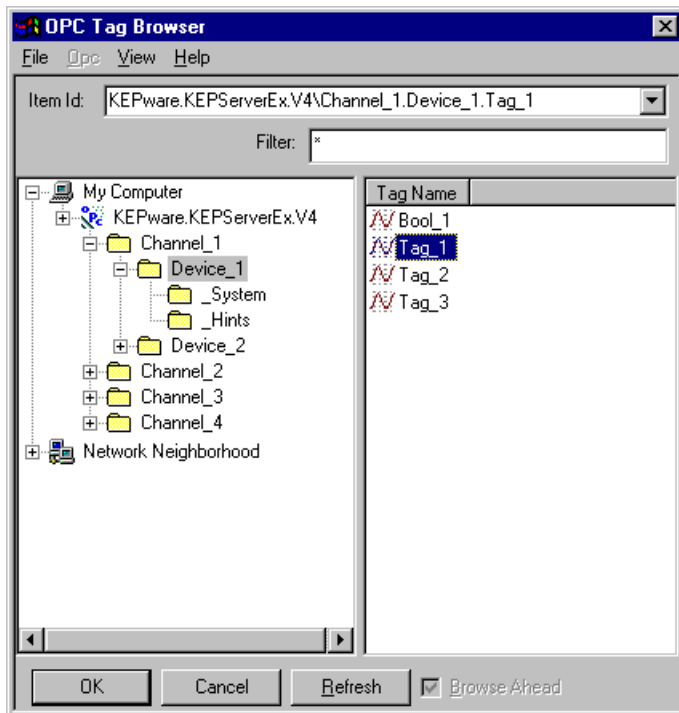


- In the OPC Tag Browser tree view expand KEPware.KEPServerEx.V4 and then select Device_1 under Channel_1. A list of the tags available under Device_1 will appear in the right-hand pane of the browser.

If you are connecting to a server located on another PC then you would browse through the Network Neighborhood, select the PC running the server and then select the server from the available list. You will need to configure DCOM for the remote connection. See Kepware's DCOM Configuration for KepserverEx guide for details.

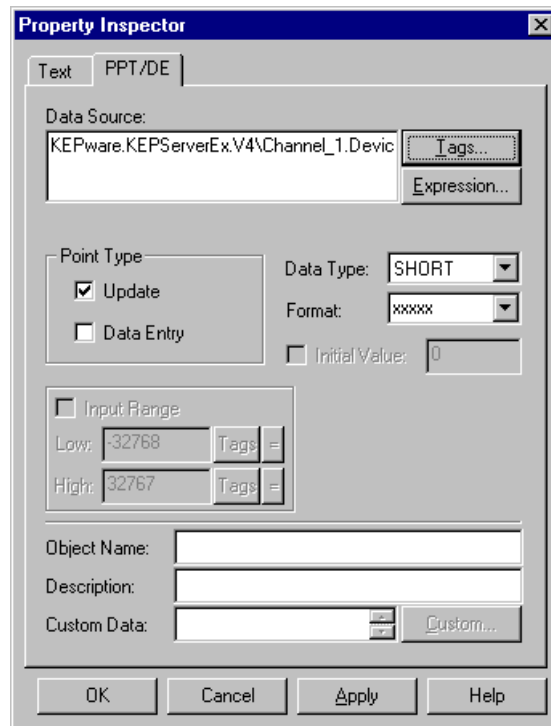
Select Tag from Browser

- Select Tag_1 and click OK to close the browser.

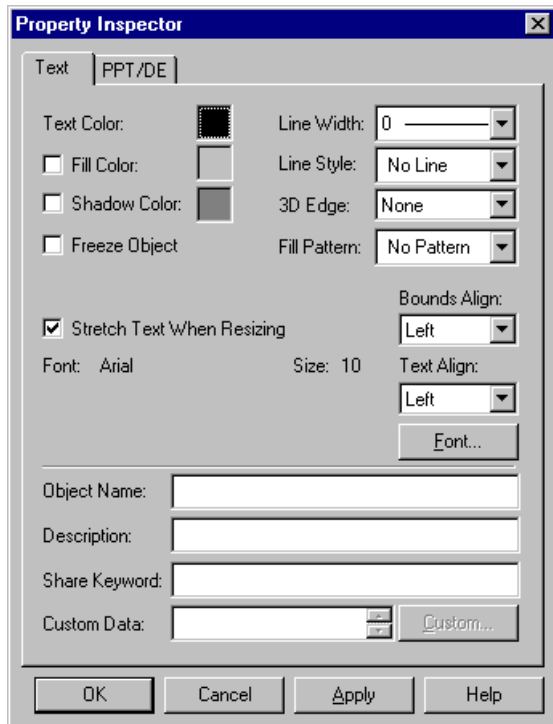


The Item ID for the Tag is KEPware.KEPServerEx.V4\Channel_1.Device_1.Tag_1. If you had browsed to another PC in the network to select the tag the PC name would be included in the Item ID as follows \\Remote_Station\KEPware.KEPServerEx.V4\Channel_1.Device_1.Tag_1.

8. Now that you have your tag you will want to set the Point Type. We have left the default of Update.
9. Next, if you are using an older version of Iconics you may need to set the Data Type. This particular tag in the KepserverEx is a Short.

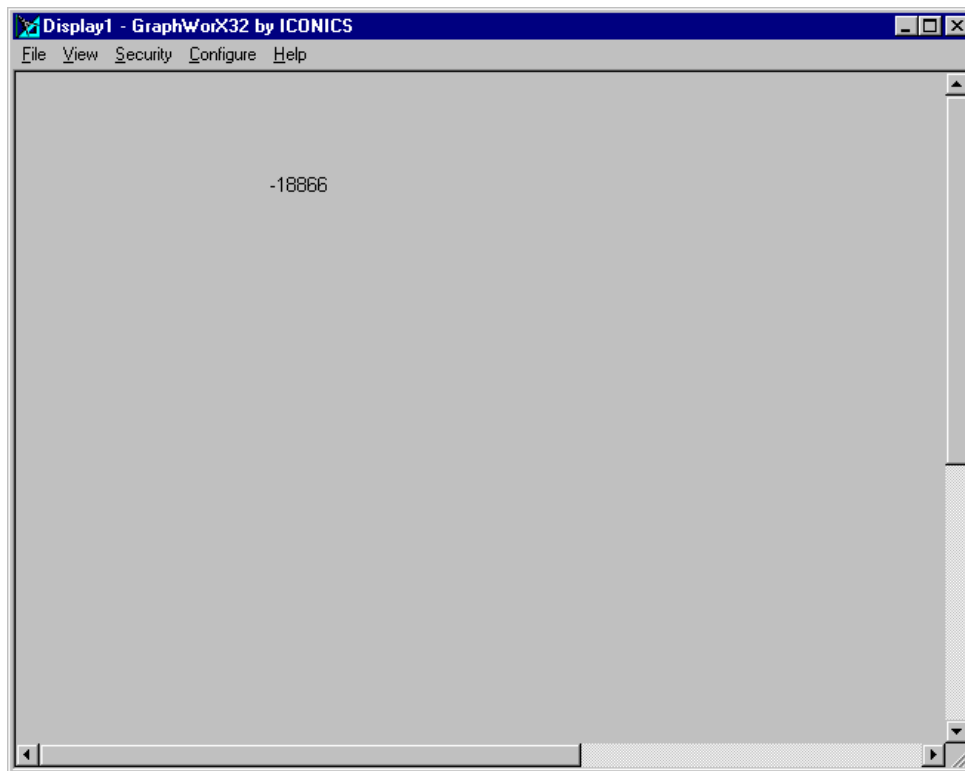


10. Now click the Text tab to define how you want the point to display.
11. Select the properties you want for your object then click OK.



Check the OPC Connection and View Server Data

1. In the GraphWorX32 window you will see the new object on the window. Click Runtime to run the window and connect to the server.



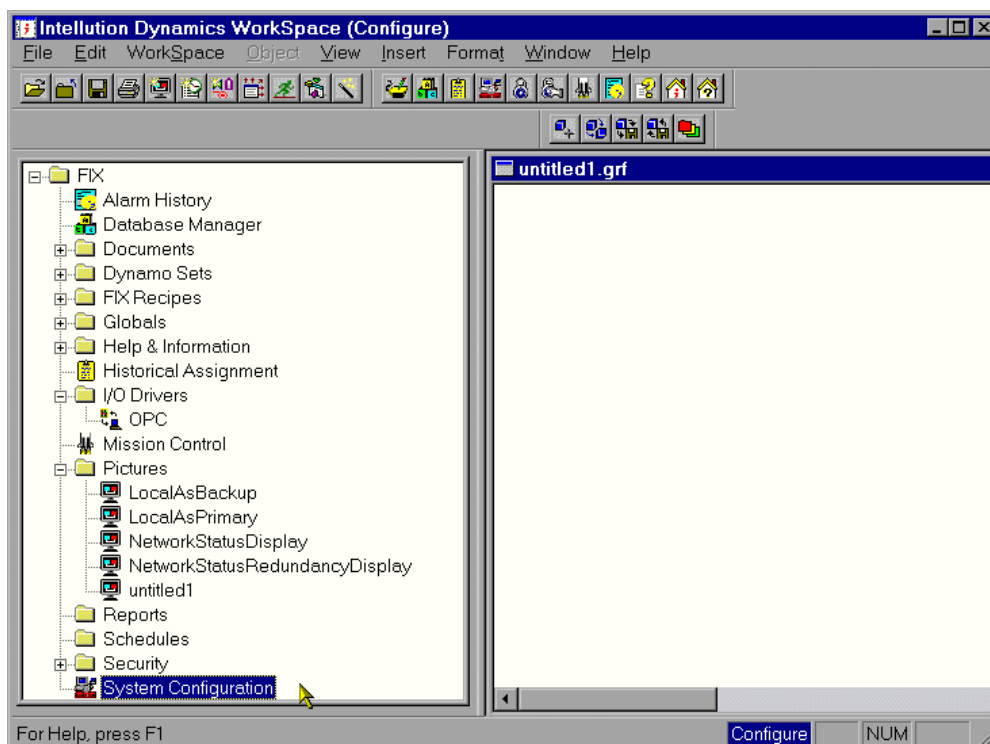
2. At this time you should see a changing number in the display window.

Intellution's Fix Dynamics® and OPC PowerTool® as an OPC Client

The following steps will show you how to create an OPC connection to the Server from the OPC PowerTool and Fix Dynamics. Although Dynamics will run on any 32bit operating system, it is very important to note that the OPC PowerTool must be used on a Windows NT based platform.

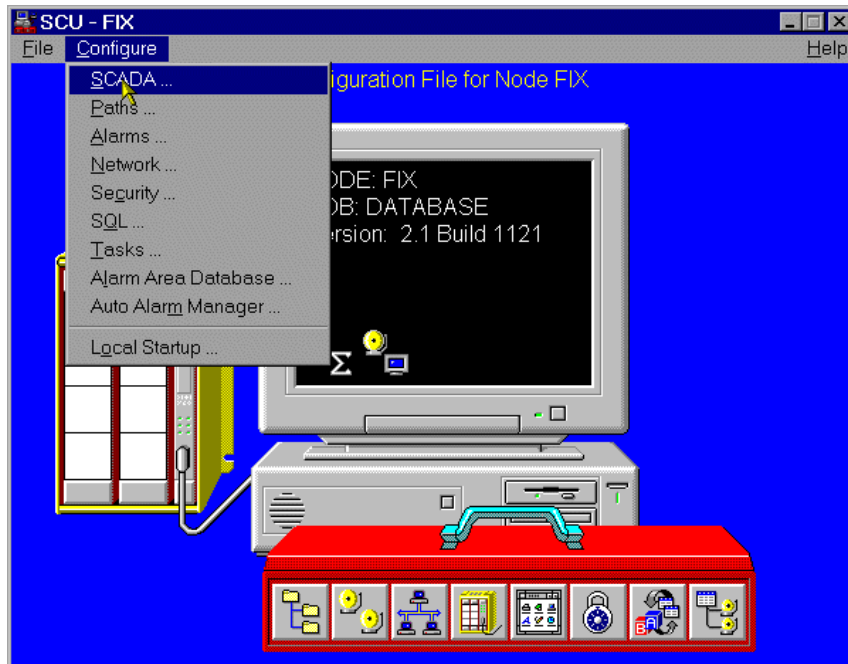
Connect to the Server from iFix

1. Open your copy of Dynamics to start a new project.

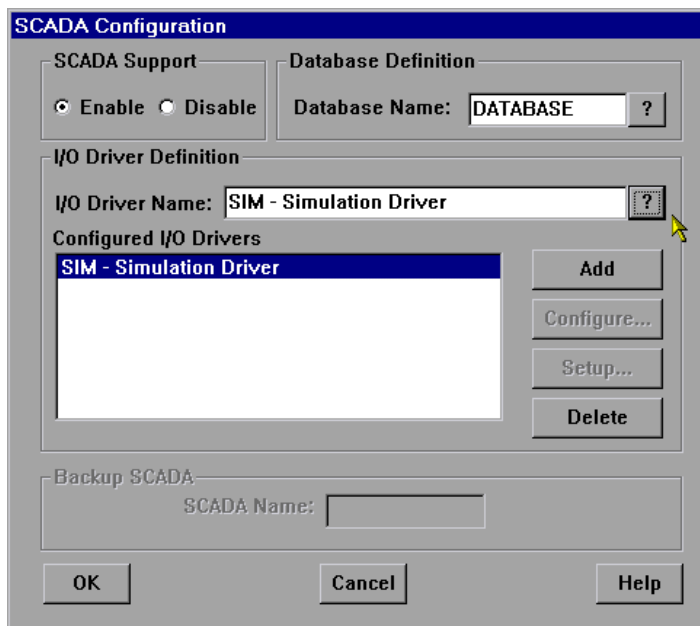


Configure the SCADA Properties

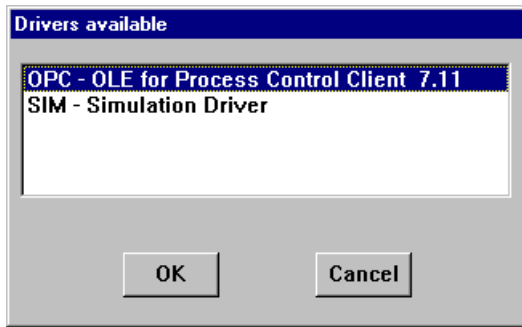
2. In Fix Dynamics, select System Configuration from the Tree. This will start the System Configuration Utility (SCU).



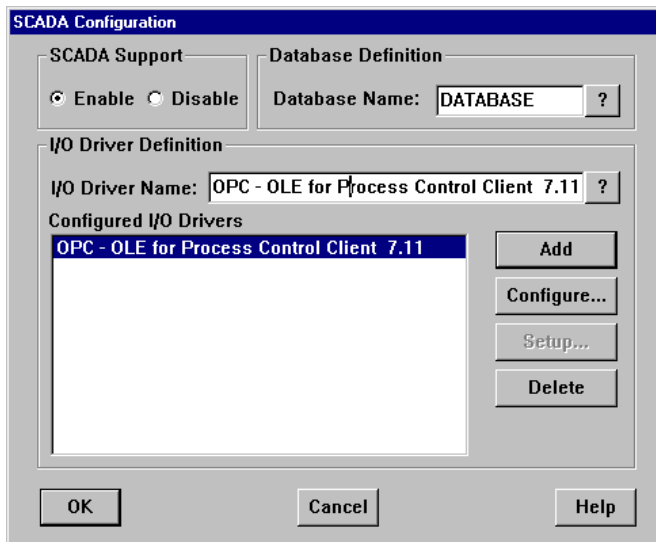
3. In the SCU select Configure|Scada... from the Main Menu.



4. In the SCADA Configuration window click on the Enabled radio button. This will activate the Database Definition section and the I/O Driver Definition. For our example we will accept the default Database Definition
5. Click on the I/O Driver Name selection button to see a list of available drivers.
6. Select the OPC Client driver and click OK.

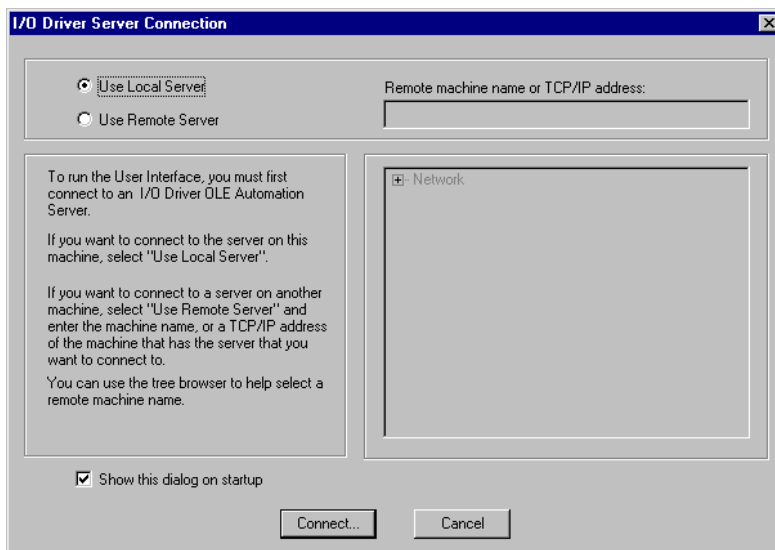


7. Click Add to place the OPC Client driver into the list of configured drivers for this project.

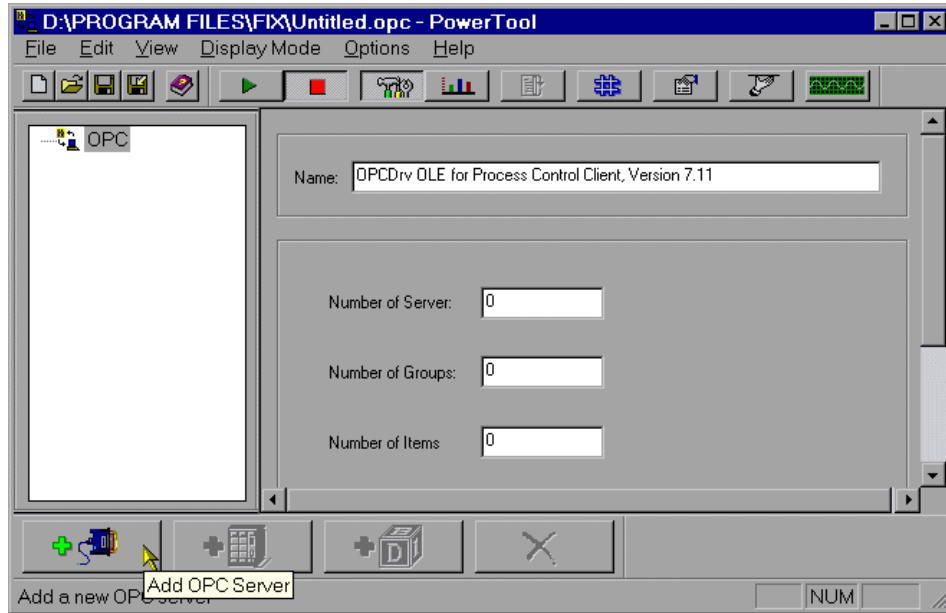


Enable the Fix OPC Client

8. Highlight the driver in the list and then click on Configure... to set it up.



9. In the I/O Driver Server Connection Window, enable the Use Local Server radio button and click Connect... to connect to the OPC PowerTool.

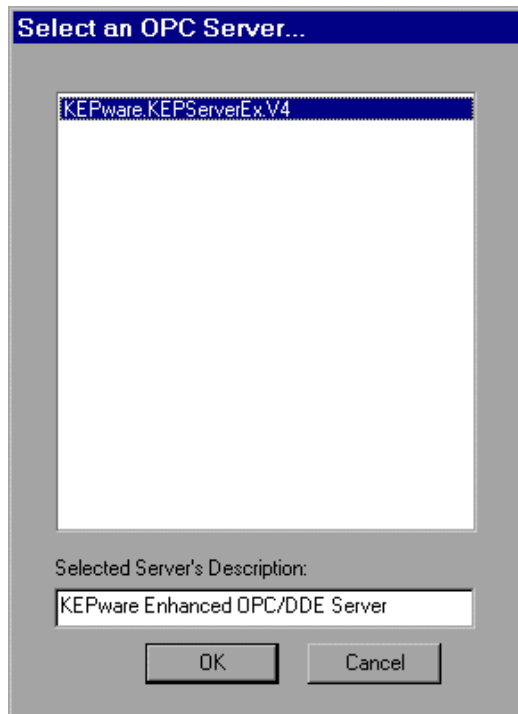


Add a Server to the OPC PowerTool

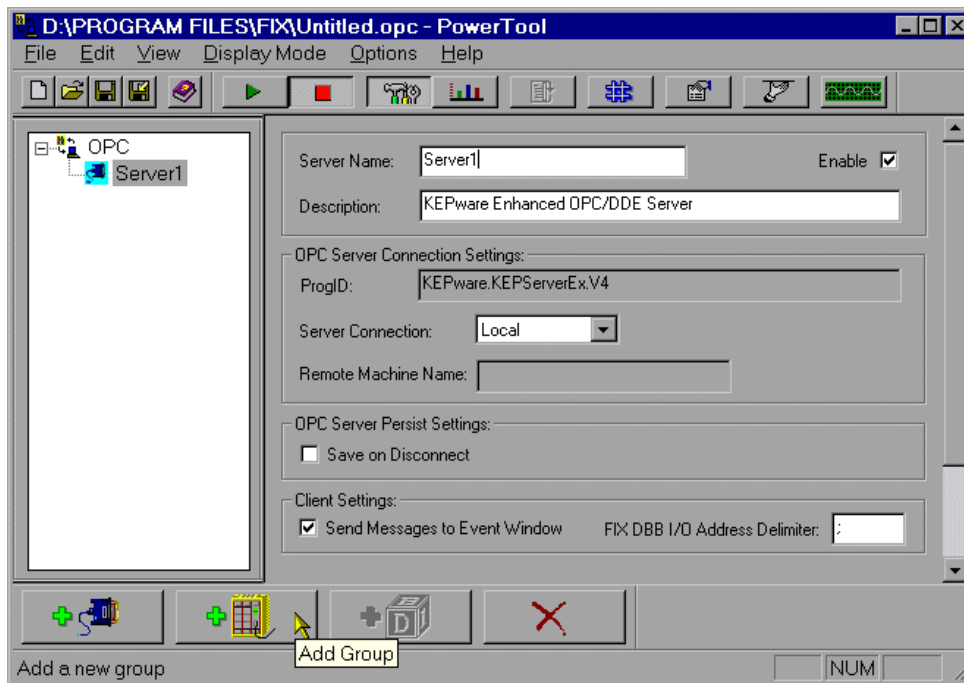
1. In the OPC PowerTool window you will add the Servers that you wish to connect to, Tag Groups related to those servers, and Tags within each Tag Group. To start you will need to add a Server.

Browse for Available Servers

2. Click the Add Server button located in the lower left-hand corner of the OPC PowerTool window, located first from the left.



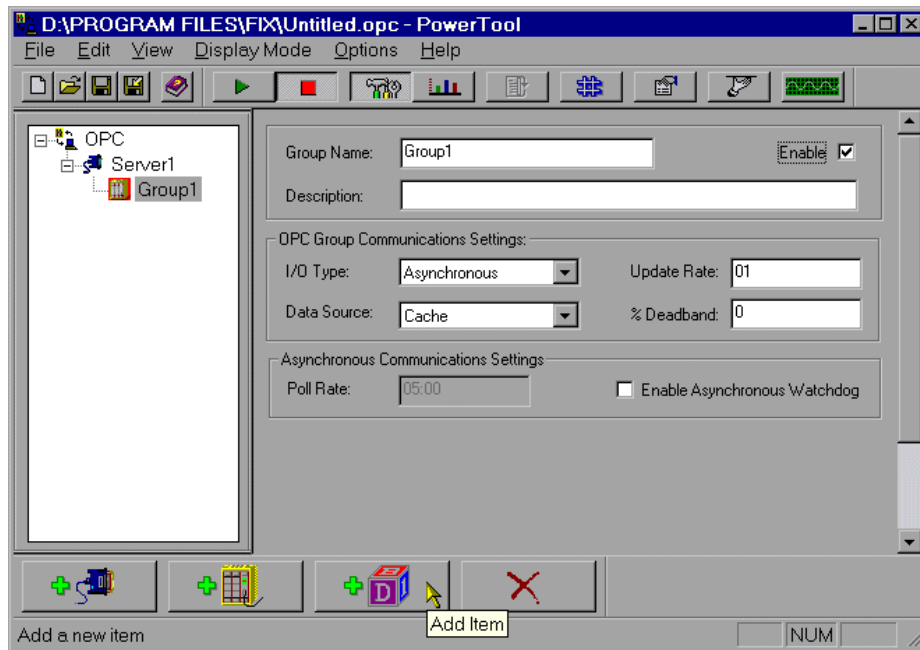
3. Select an OPC Server from the browser window and click OK. In our example we have only one server loaded on our machine.



4. Enter a Server Name and Description to identify the server within Fix Dynamics.
5. Make sure Enable is checked, otherwise you will not be able to communicate with any devices in this server.
6. Leave the Save on Disconnect box unchecked. The KepserverEx does not support this function. Leave the Client Settings at their default.

Add an OPC Group

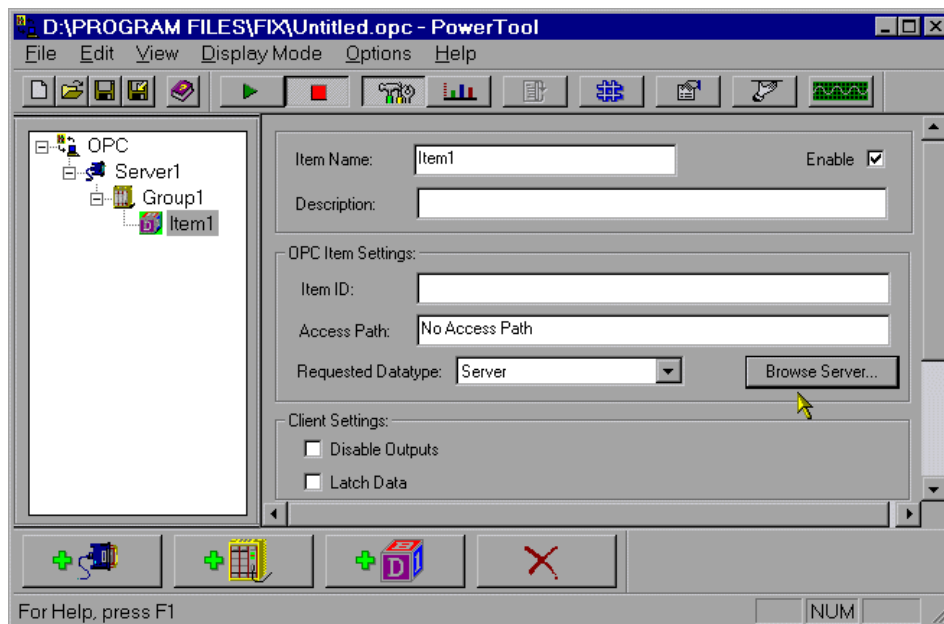
1. Next, click the Add Group button located at the bottom left-hand side of the OPC Power Tools window, second from the left.



2. Enter a Group Name and Description to identify the tag group in Fix Dynamics.
3. Make sure you check the Enable box. Otherwise, you will be unable to communicate with the group or any items attached to it.
4. Select Synchronous or Asynchronous as an I/O Type.
5. Select Cache as a Data Source type. Cache reads are typically faster and recommended.
6. Set the Update Rate, % Deadband, and Poll Rates as your project requires. See the OPC Help file for more information.

Add an OPC Item/Tag

1. Next, click the Add Item button located at the bottom left-hand side of the OPC Power Tools window, third from the left.

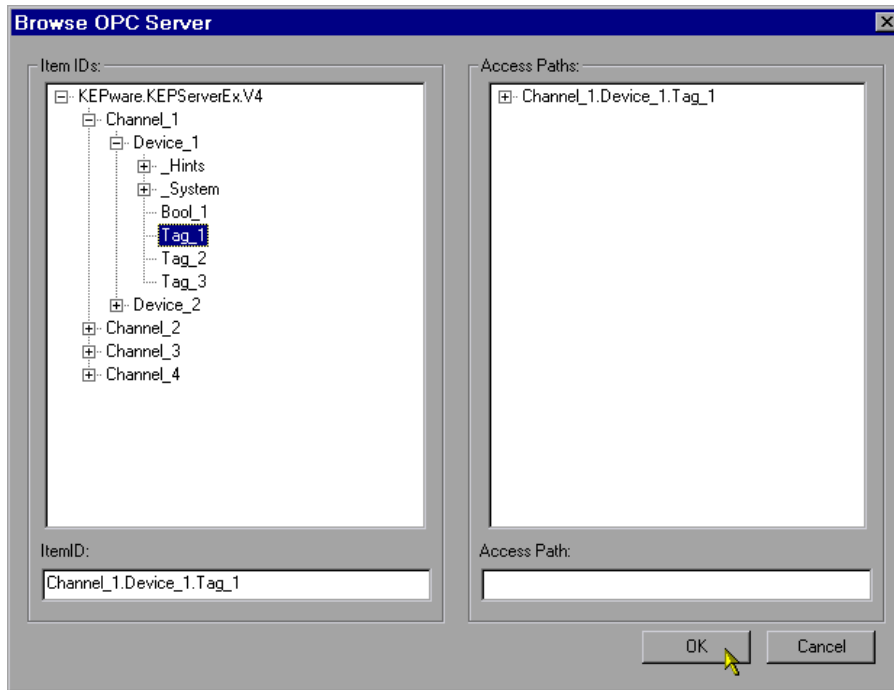


2. Enter an Item Name and Description to identify the tag group, in Fix Dynamics.

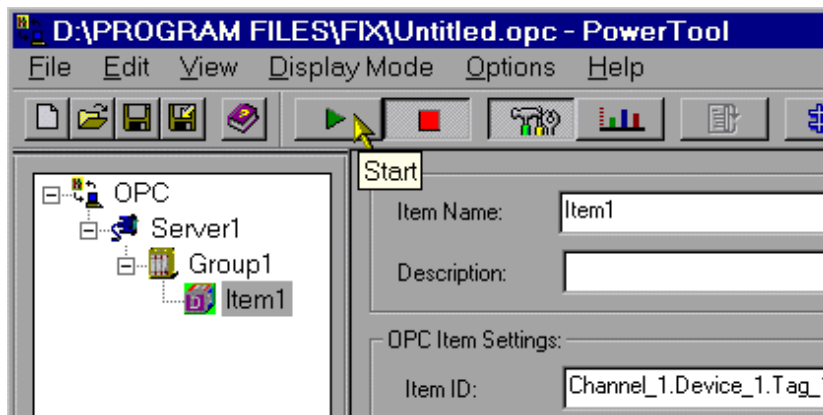
3. Make sure you check the Enable box. Otherwise, you will be unable to communicate with the item.

Browse for Available Tags

4. Click on the Browser Server button to select an Item Id or User Defined tag.



5. Select an ItemID from the server tree list and click OK.
6. You will notice that the Access Path was left blank. If you did not add tags to the server then you can add them dynamically from here as well, the proper format is “<Channel>.<Topic>.<Item>”. Remember in KepserverEx the <Topic> is the device name and <Item> is the defined or dynamic tag.
7. Click Save to save the server setup you have just created.



8. Once you have saved you can click Start and then click Statistics to be able to verify your connection to the server.


Kepware's OPC Quick Client as an OPC Client

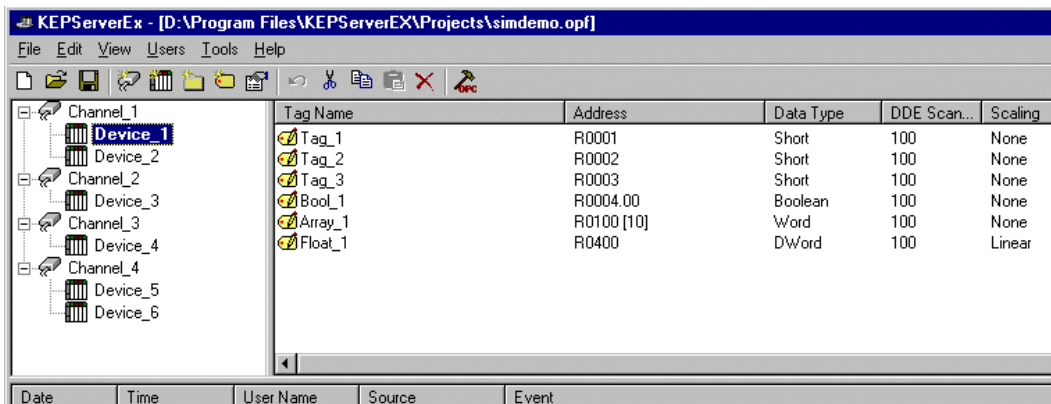
Connect to KepserverEx from Kepware's OPC Quick Client

Kepware's OPC Quick Client is an interface that can be used to connect to KepserverEx. For this example, the OPC Server Version used is V4.41.163 and the OPC client version used is version 4.20.66. The following steps will show you how to create an OPC connection to the KepserverEx from the Test Client either by using the auto project launch form the server or by manually starting the client and creating a project. Both examples use the "SimDemo.opf" project provided with the server install.

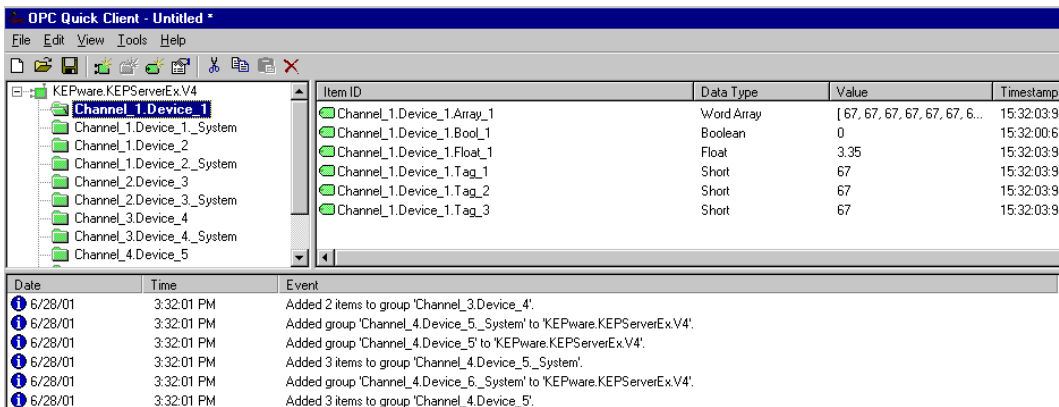
Auto Launch and Generate a Quick Client Project

In most cases when you are using the Quick Client you are testing to see if you can see all of the tags you have defined in the server. To aid in this process we provide the ability to launch the client from the server and automatically generate the client project.

1. In the Server, once you have added all of the tags you currently want, click on the Launch button  or select "Tools|Launch OPC Quick Client".

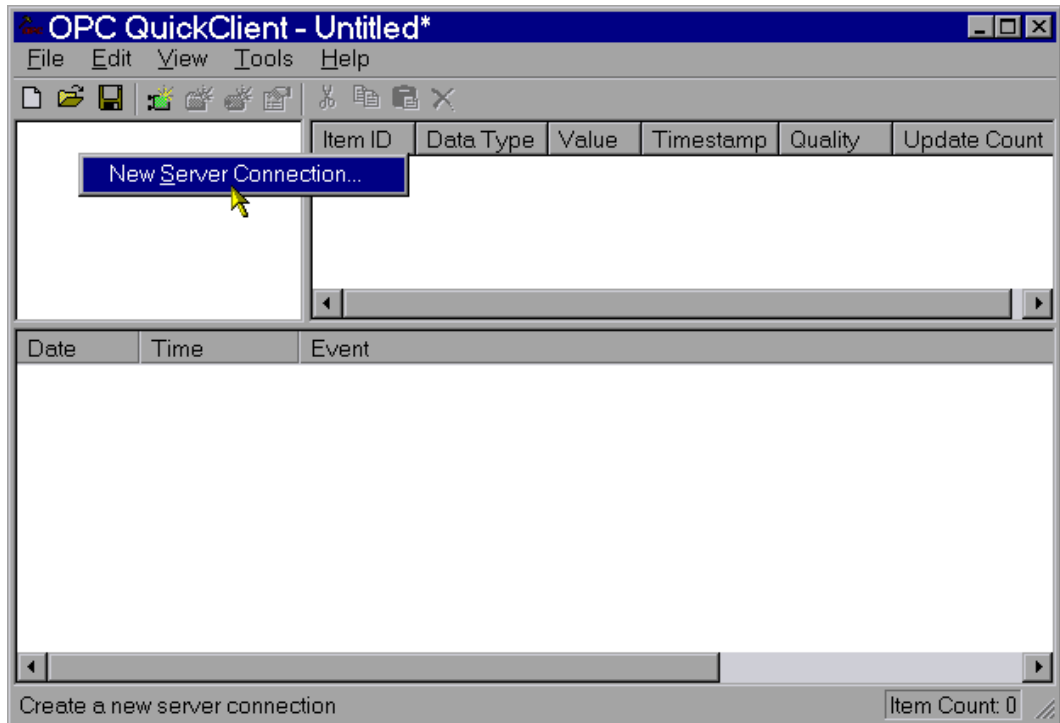


2. You will notice that the client project automatically opens to display the tags in the first Device group.



Manually Create a Quick Client Project

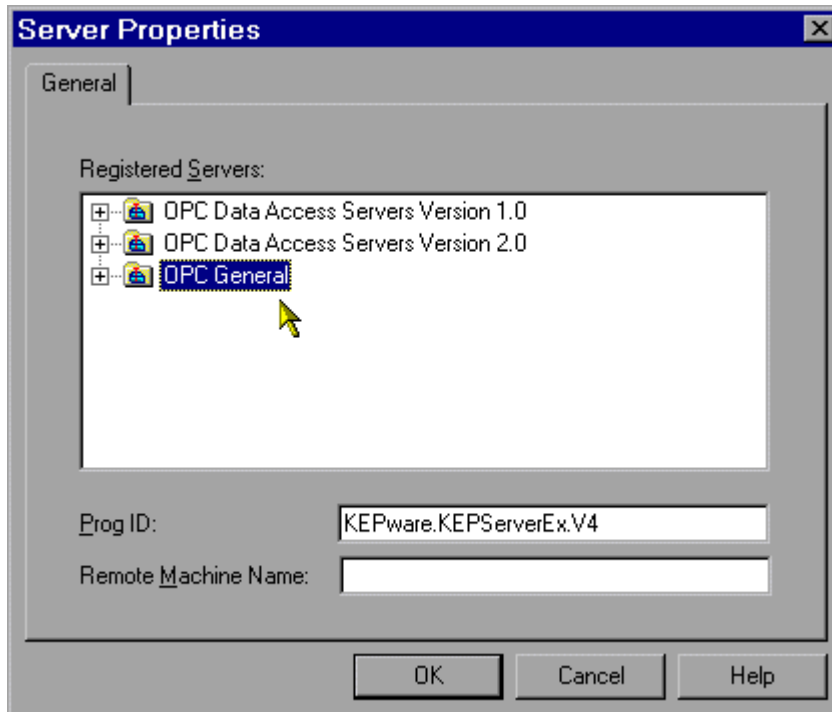
1. To start a new project, select **New Server Connection...** from the **Edit** menu or right click in the upper left display box and select **New Server Connection...**(shown below).



Browse for KepserverEx

2. In the Server Properties window, click **OK** because the KepserverEx is set as the default server. If you were going to connect to the server running on another PC then you would enter the name of that PC in the **Remote Machine Name** field of the Server Properties window without any back slashes “\\”.

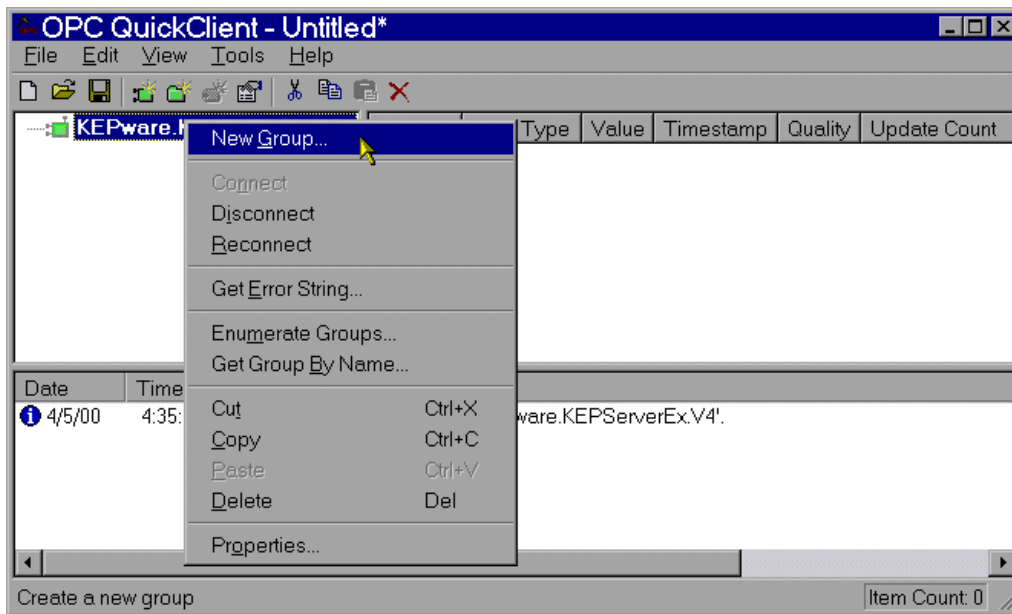
For remote connections you have to be sure that you have DCOM properly configured. For information on how to configure DCOM see the two guides located either on the installation CD or at the Kepware web site.



Add a Tag/Item Group

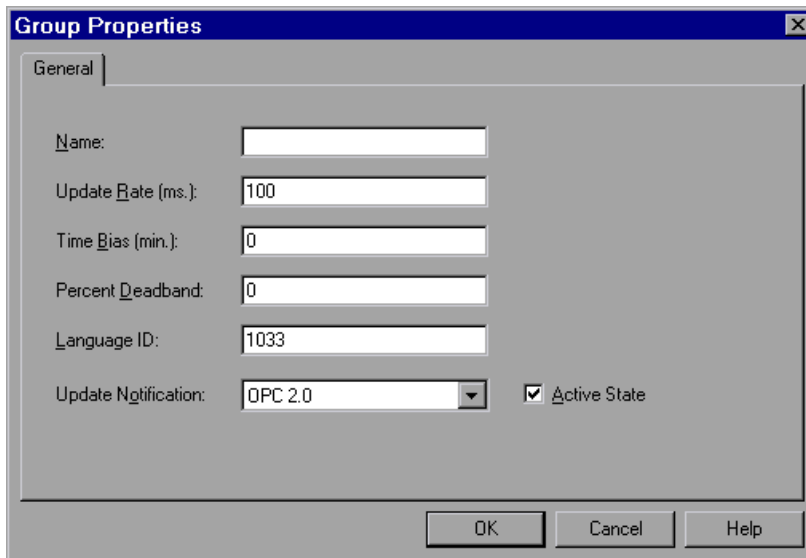
After choosing KepserverEx, there should be a visual representation of the client connection to the server provided in the upper left-hand display.

3. Right click on this connection and choose New Group or select New Group from the Edit menu.



Set Group Properties

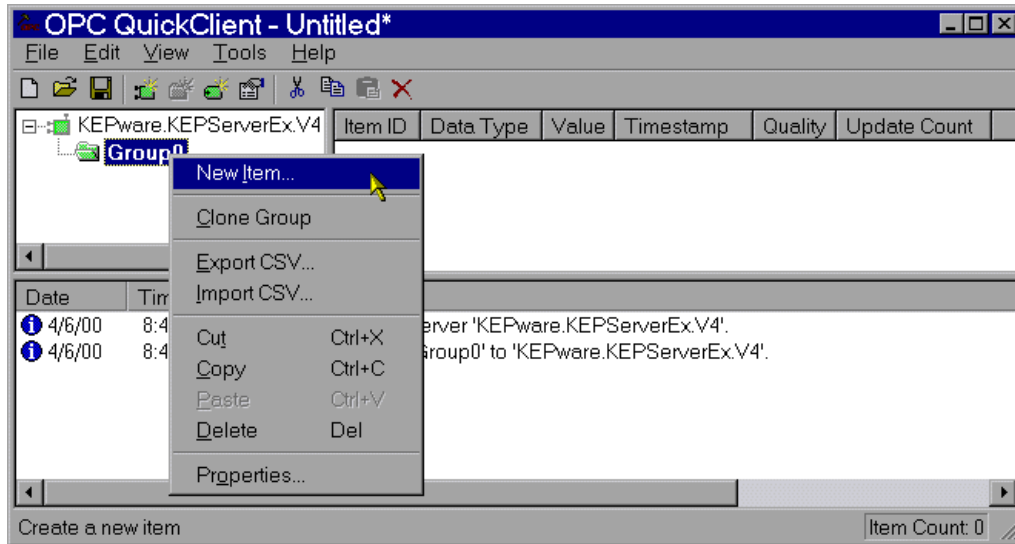
4. In the Group Properties window, enter a Name to identify the group. If no group name is entered, the server will generate a unique name for you. Also, make sure the Active State is enabled so the client will show active items for the group. Finally, click OK.



Add Tags/Items

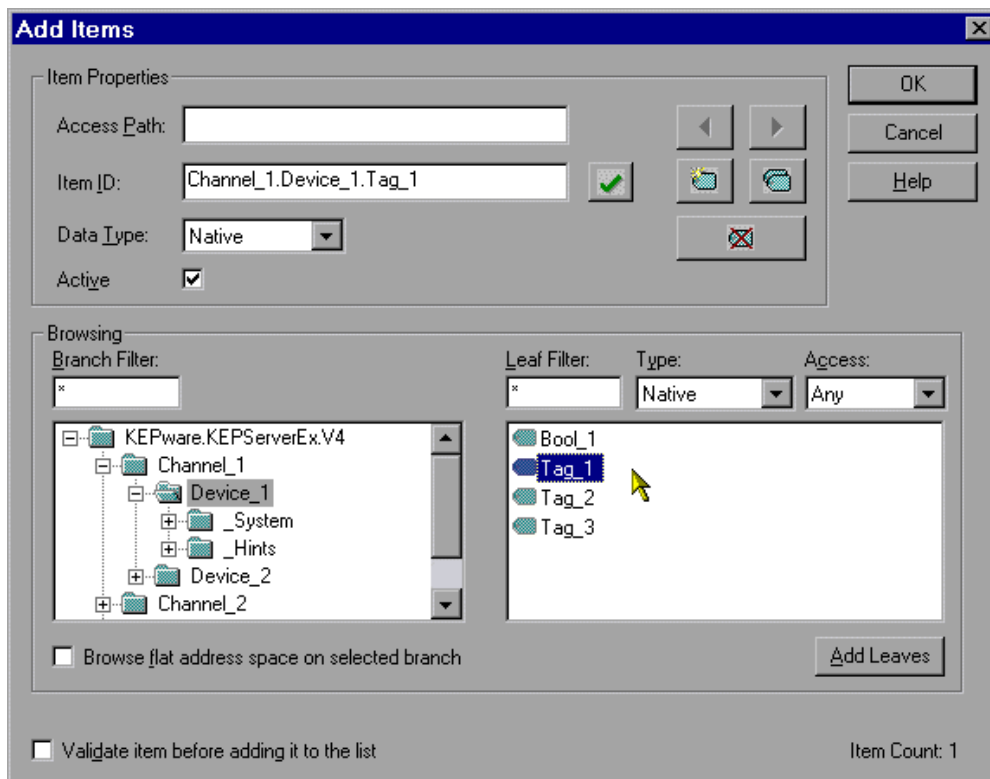
Now that a group has been created it is possible to add items.

5. Right click over the Group Name in the left display or select New Item from the Edit menu to open the Add Items window.



To learn more about using Kepware's OPC QuickClient with KepserverEx, see the KepserverEx help file "Designing a Project (Saving and Testing the Project)".

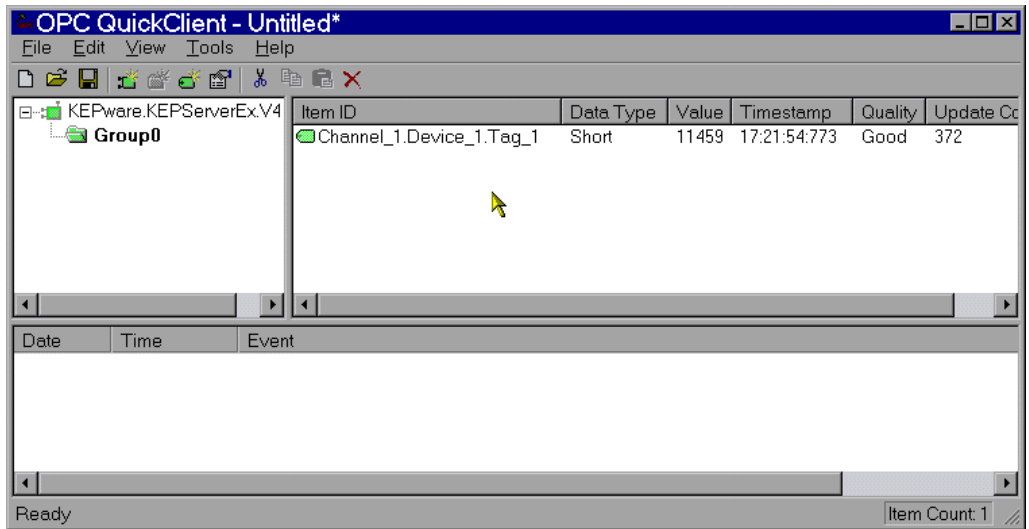
- Using the tree view in the left display of the OPC Address Browser, select the Device or Group from which you wish to select Address items/tags. In the right hand pane, double click to add an item. In this example, Tag_1 in Device_1 on Channel_1 has been selected. Remember that these are items that have already been created in our Server Demo Project (Simdemo.opf). Also, you can view all possible address types and create a dynamic item by choosing "_Hints" under the device name in the tree menu.



- Once you have selected all of the items to be displayed by the client, click OK

Check OPC Connection

- Next we are going to verify that we are getting data from KepserverEx. You should see a Quality of "Good" in the Tag Monitor window for the tag and a Value that is ramping or incrementing very quickly.



9. You now have a good OPC connection to the Server from the OPC Quick Client.

OSI PI as an OPC Client

Connect to the Server from OSI PI

The following steps will show you how to create an OPC connection to the Server from OSI PI. OSI PI works a little differently than other clients in that it uses several programs running in the background to handle connectivity. We will show you how to start the processes and how to get and display data from the server. All of our examples will use our “Simdemo.opf” project that is provided with every software installation.

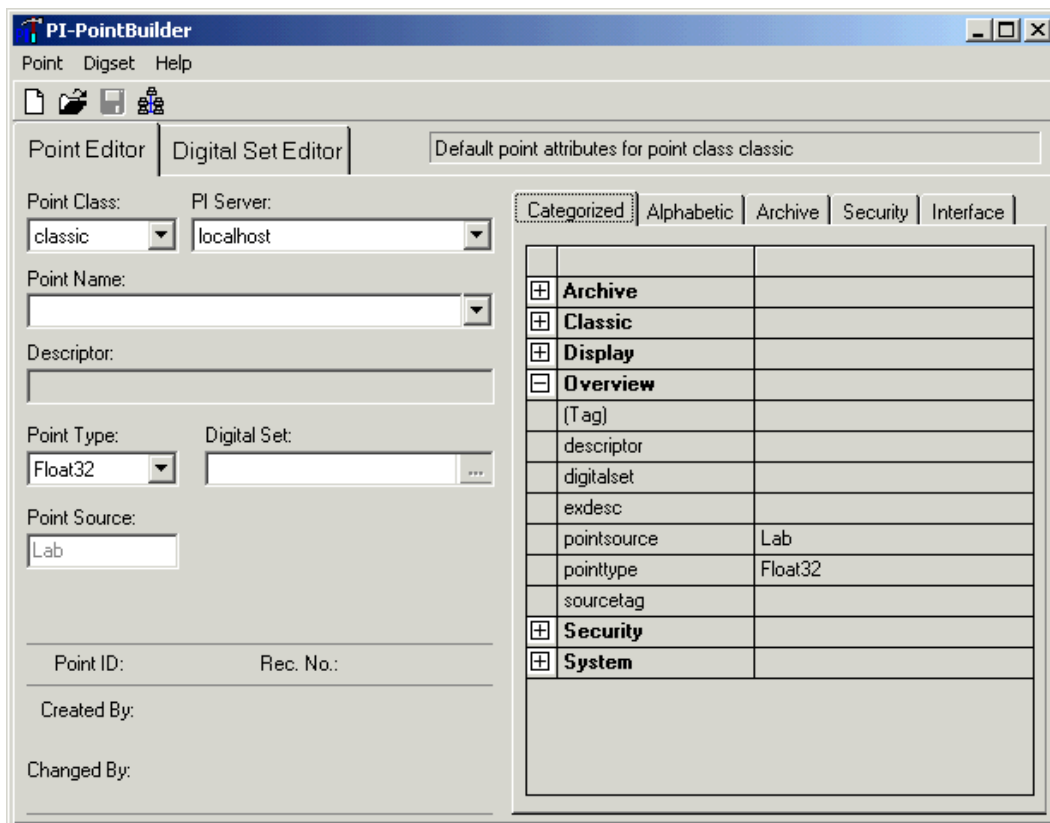
Start the OSI Pi Process Services

First you will need to start the PI Process Services. This can take a few minutes so it is best if you start them at boot-up. You will want to create a shortcut in your Startup folder that points the “PISRVSSTART.BAT” file located in “C:\PI\ADM”. This file must run from the ADM folder.

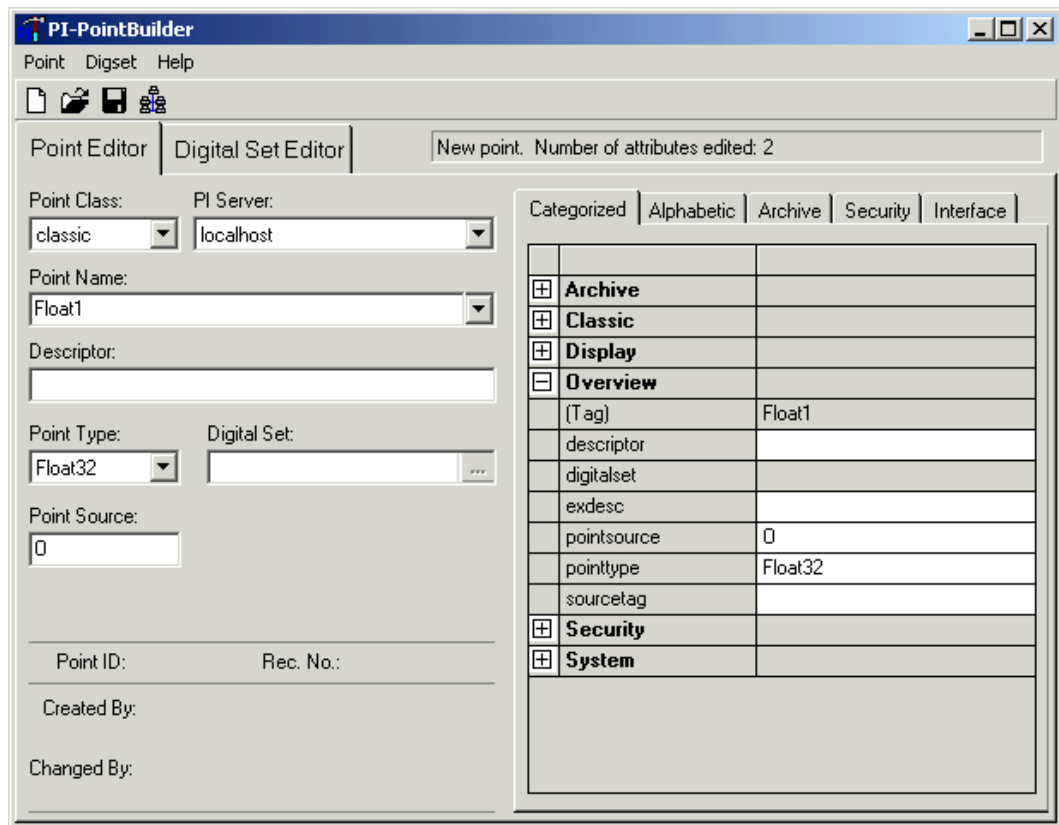
All of the PI software was installed using the default installation folders. If you did not use the defaults then you may have to use different path references.

Create PI Tags in the PI Pointbuilder

Now you need to create a point for use in your ProcessBook Project. To do this you need to open the PI Pointbuilder by selecting it from the PI System Start Folder.



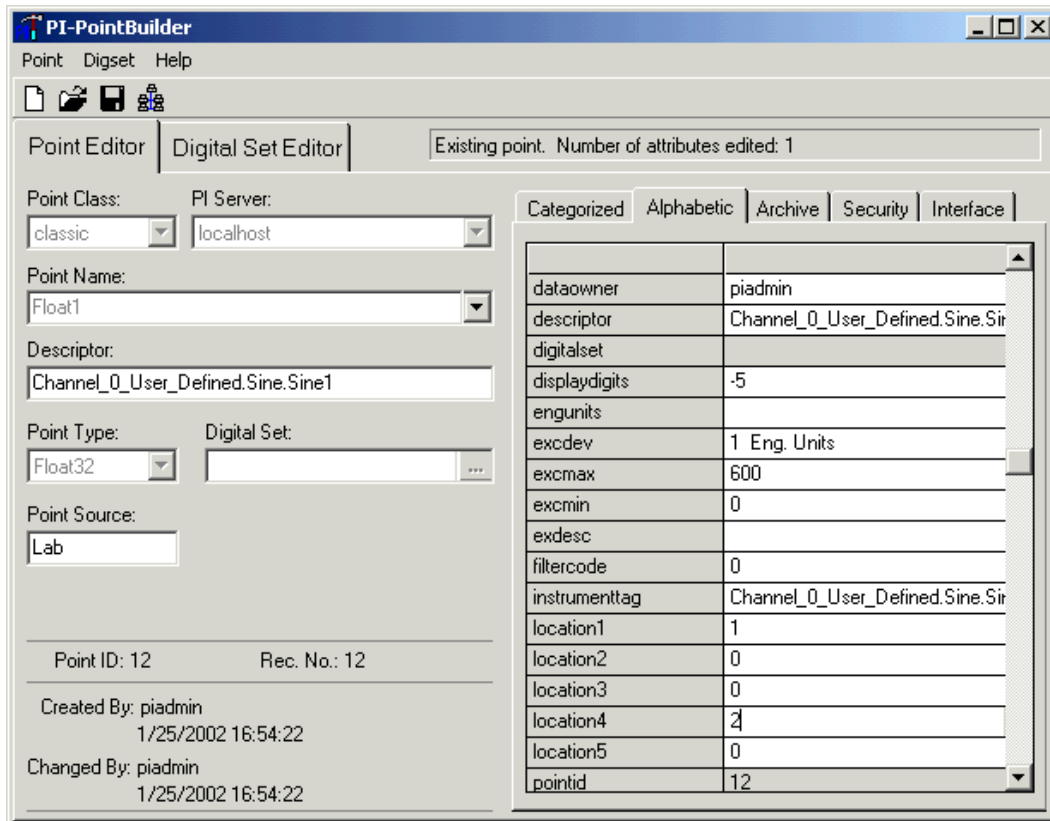
By default, the Pointbuilder will open to add a new tag. First you will want to enter a Point Name (we named our tag “Float1”). Once you enter a Point Name, the Point Source field becomes available to edit. You will want to enter a capital “O” for OPC in the Point Source field. All tags defined in the PI database and used by the OPC interface must share a common Point Source. Leave the Point Type at its default, since we are creating a Float.



Next you are going to select the Alphabetic tab. This is where you will define the item in the server that you wish to use in PI. First, you will need to enter the full Item ID of the server item you want to use in the Instrumenttag field. We are using “Channel_0_User_Defined.Sine.Sine1”. Next, copy the contents of the Instrumenttag to the Descriptor field, this provides text to display when using the tag if needed.

Now you are going to write a “1” to the Location1 field. This field specifies the interface instance number. In the **opcint.bat** file, there is a parameter for interface ID, /ID=#, where # is any number. The Location1 field for each tag must match that number, or the tag will be ignored. We will discuss the “.bat” file later.

The last thing you are going to do is write a “2” to Location4. This is the scan class number. Scan classes are defined in the interface startup file; each scan class defines an update period. This location code defines which scan class period is used to update the PI point. We are assigning this point to the second scan class period



Once you have entered all of information you can save the PI Point to the database.

Starting the OPC Interface

If you have never used PI before, you will find a generic “.bat” file named “**opcint.bat**” in the “C:\Program Files\PIPC\Interfaces\OPCint” folder. You need to modify this file to allow the server to connect. The current file is configured as follows:

```

rem
rem Startup file for the OPC interface to PI
rem The ^ marks are continuation characters, they allow
rem you to have a command be split between multiple lines.
rem There must not be ANYTHING after the ^ on each line.
rem This is only a sample of the options available, the user
rem manual has the list and descriptions for them all.
rem
rem /ps=O           The pointsource -- this should match the pointsource for
your tags
rem /ec=10         The event counter number for IORATES
rem /er=00:00:03  The requested update rate for event triggered tags
rem /id=1          The identifier string used in the pipc.log file for messages
rem               from this interface. It must match Location1 on the tags.
rem /SERVER=OPC.OSI.1 The OPC server name; format hostname::servername or
rem               just servename if it is local
rem /host=mabel:5450 The PI server name and port
rem /MA=Y          Should we try to add tags in large batches rather than
rem               singly ?
rem /ts=a          Where do we get timestamps ? (Y/N/A/U)
rem /stopstat      Write a status to PI tags when the OPC server goes away ?
rem /f=00:00:01   Scan classes. The first one is for Read On Change tags...

```

```

opcint ^
/ps=O ^
/ec=10 ^
/er=00:00:03 ^
/id=1 ^
/SERVER=OPC.OSI.1 ^
/host=localhost:5450 ^
/MA=Y ^
/ts=Y ^
/OPCstopstat ^
/f=00:00:01 ^
/f=00:00:01 ^
/f=00:00:01 ^
/f=00:00:02

```

TextBox 1-1: Generic OPCint.bat

Now you will modify the “.bat” file. First you are going to change the “/Server” parameter to read “KEPware.KEPServerEx.V4”. This is the program id of the KepserverEx. Next, we are going to change the second “/f” scan class to read “00:00:00.1”. This directs the OPC Interface to request data from the server every 100 msec.

The last parameter you will need to know is how to change the “/host” parameter. This tells the OPC Interface where to look for the server. If you are running the server on the local PC then you will leave it set to “localhost”. However, if you are running the server on another PC then you should change it to the name of the PC running the server. You will also need to make sure that you have properly configured DCOM. You can find details on how to do that in the DCOM documents provided on the Kepware Technologies installation CD, at our Web site or in the OPCinc.doc (word) document provide by PI with the installation of the OPC interface. When you are done, you must save the edited batch file.

```

opcint ^
/ps=O ^
/ec=10 ^
/er=00:00:03 ^
/id=1 ^
/SERVER=KEPware.KEPServerEx.V4 ^
/host=localhost:5450 ^
/MA=Y ^
/ts=Y ^
/OPCstopstat ^
/f=00:00:01 ^
/f=00:00:00.1 ^
/f=00:00:01 ^
/f=00:00:02

```

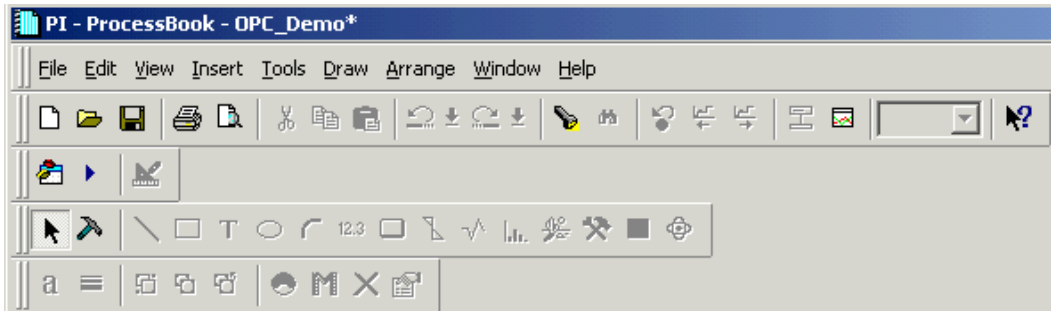
The last thing you are going to do is run the batch file to start the OPC interface. This can take a few minutes if you are loading a large number of tags.

Create a Display With PI Processbook

We will now create a PI Process Book in which to monitor the server tag that we created.

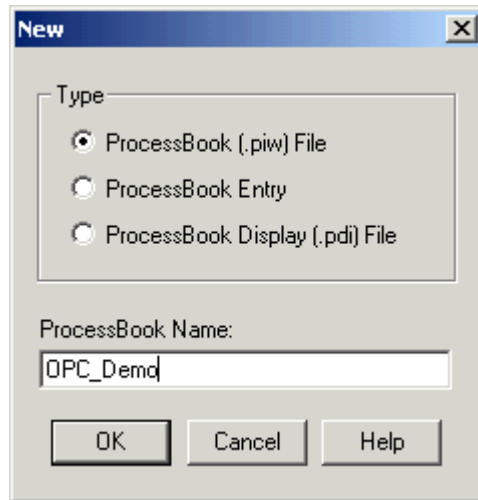
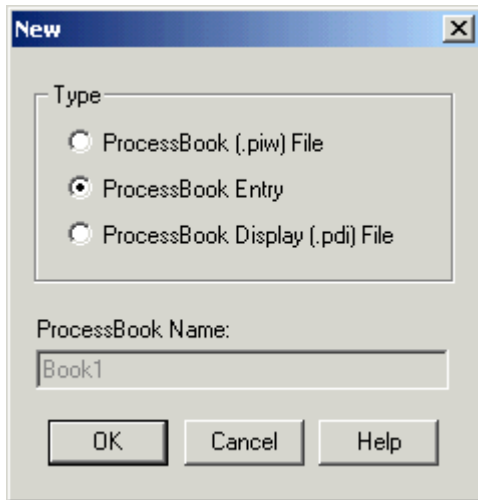
Open the PI ProcessBook

From the PI Systems start folder, click on “PI ProcessBook” to start it. By default, it will open with the PI demo processbook file showing. You can close this and then click on **F**ile|**N**ew in the main menu to create a new Processbook file.



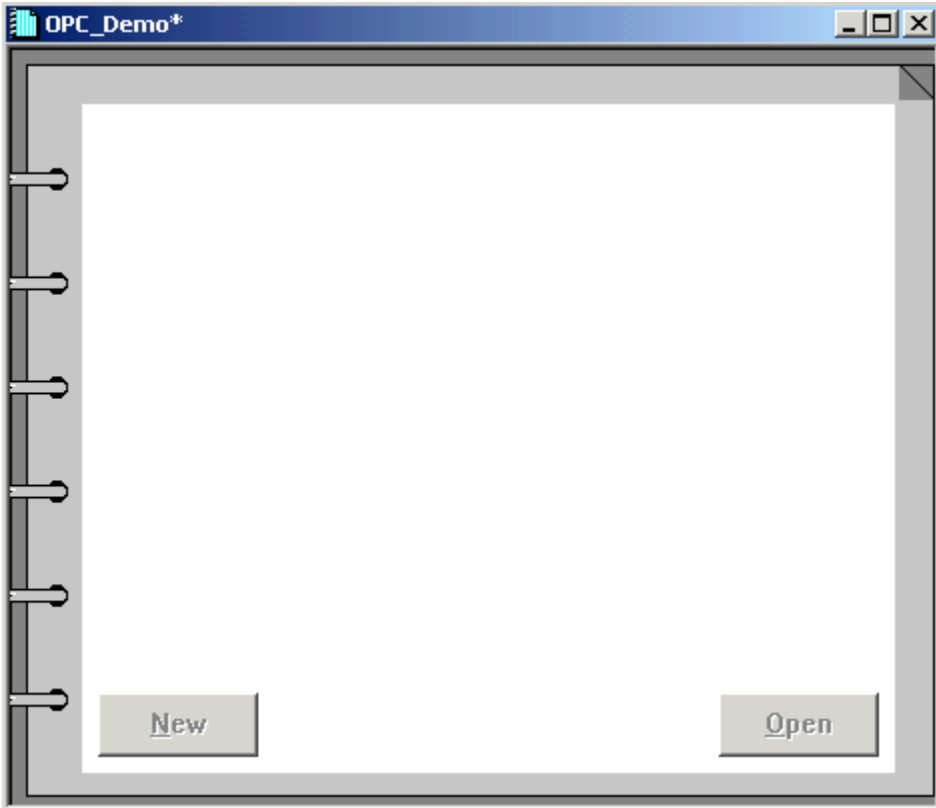
Create a New ProcessBook Project

In the New dialog box you will want to select the radio button for “ProcessBook (.piw) file” and then give your new processbook file a name. We chose “OPC_Demo” for our file name.

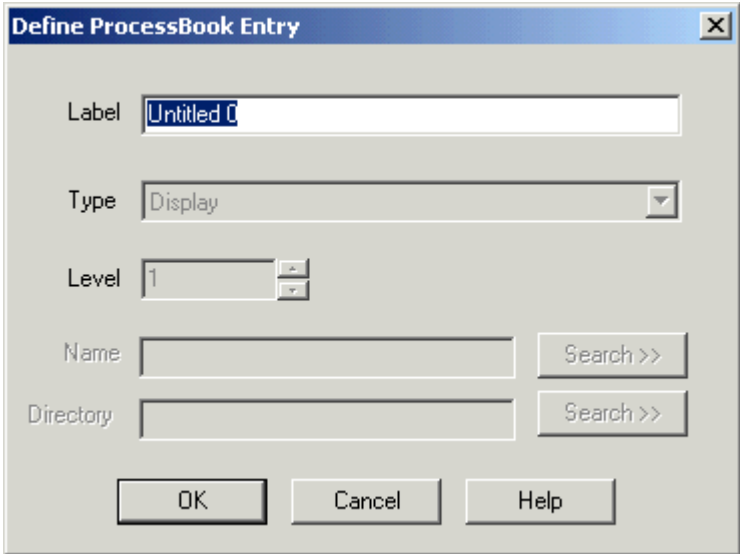


Create a Display in the ProcessBook


When you click Ok in the dialog box you will get your new Process book. Next with your new process book selected, you are going to select Insert|Display from the main menu to add a display to the Process book.

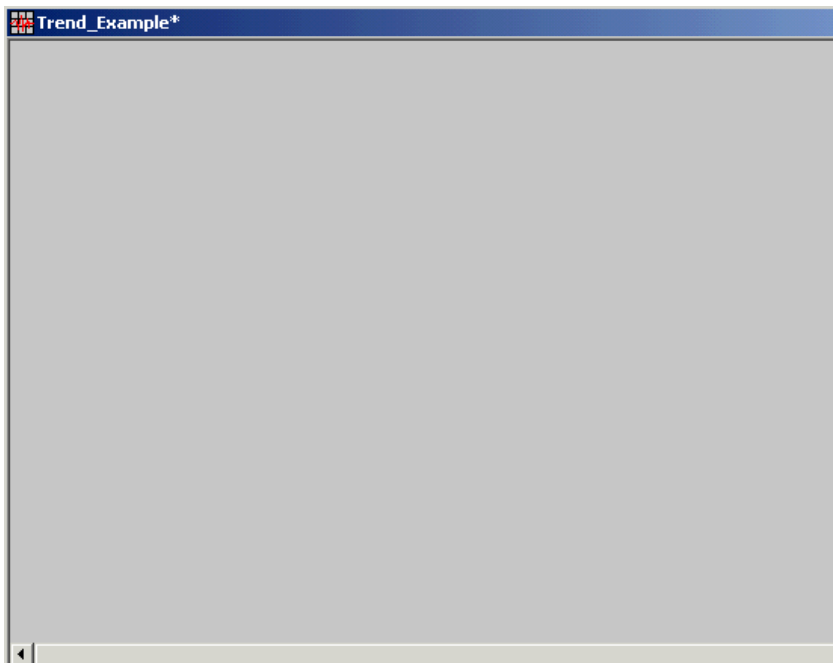


In the Define ProcessBook Entry dialog box you will need to enter a name for your display. We named our display "Trend_Example". Click OK when you are done.



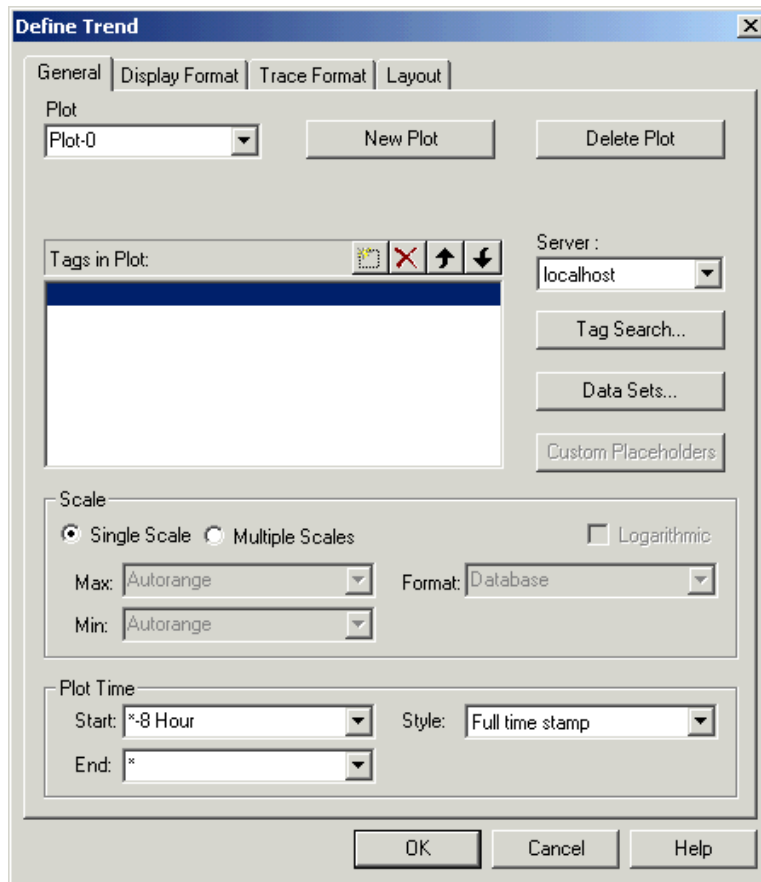
Create a Trend Object in the Display

The new display should be open. As you may have guessed by the name of the display, we are going to put a trend object in our display. Click the Trend icon  from the button menu. Then with the mouse, draw a rectangle in the display that is the size you want for your trend.

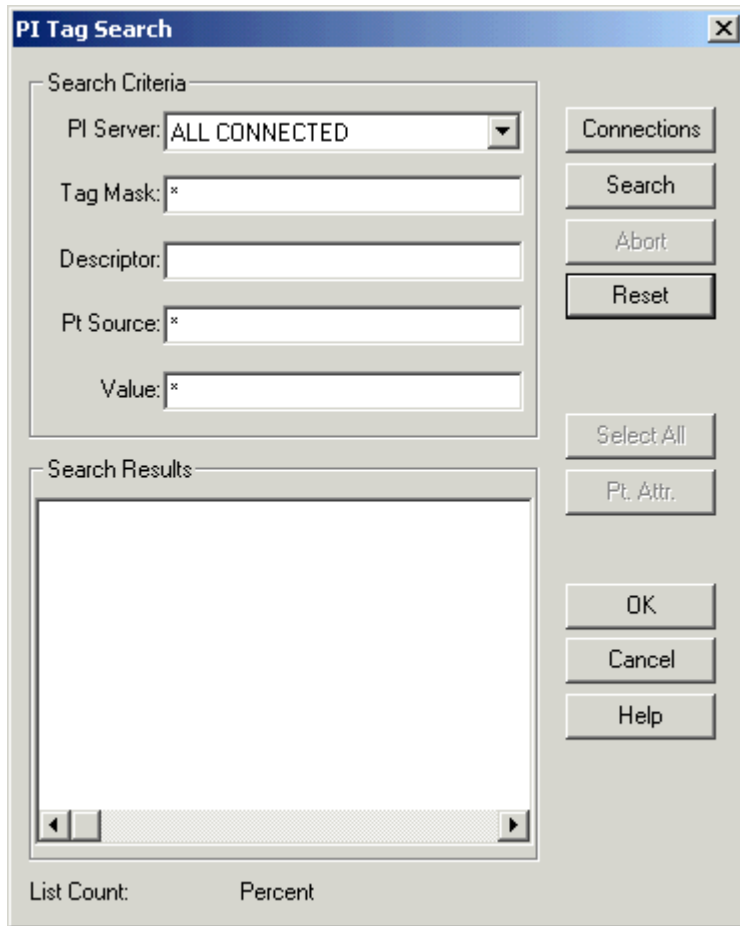


Defining the Trend Object Properties

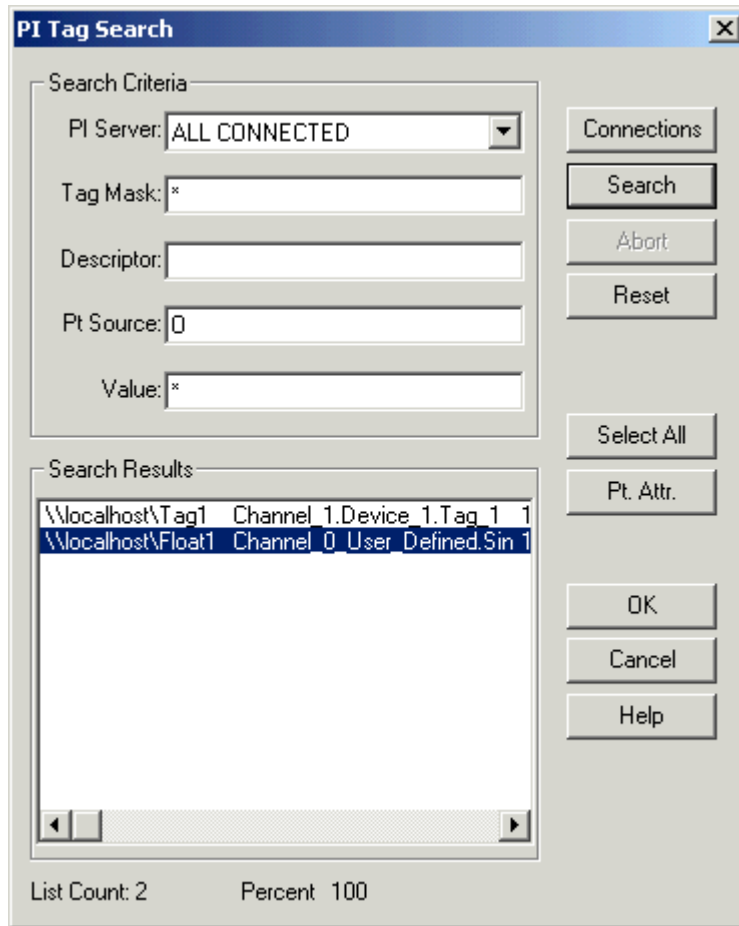
When you release the mouse button, the Define Trend dialog will automatically open. The first thing you need to do is tell the trend what data you are displaying in it. Click on the Tag Search... button to get the tag(s) that you will display.



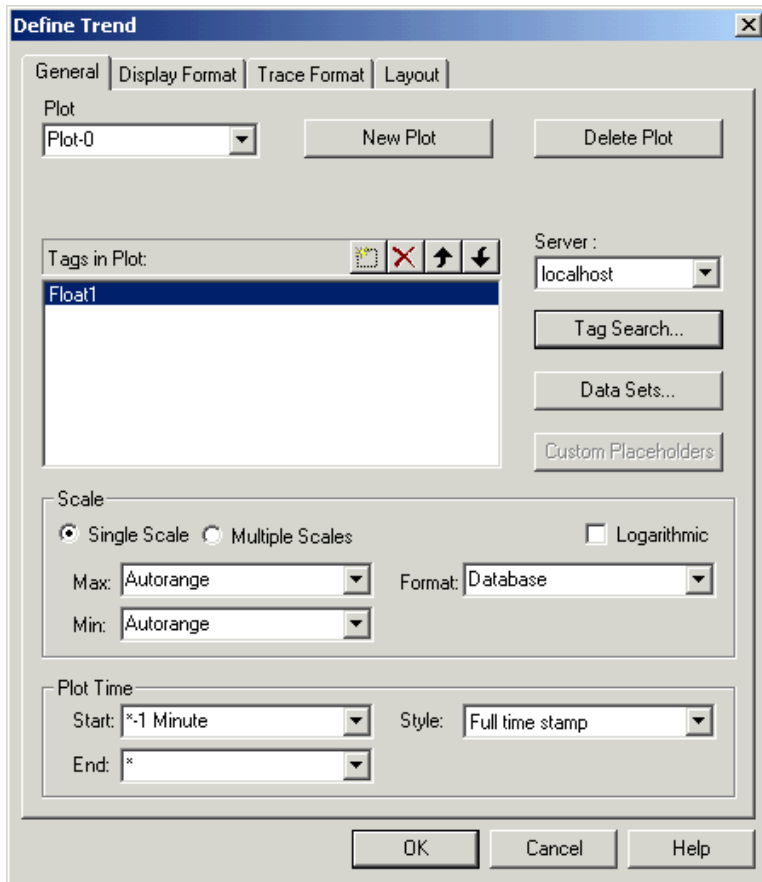
You should now see the PI Tag Search Dialog box. If you click Search you will get all the tags that have been created in the PI Points database. Since you only want the tags we created, you will enter an "O" in the Pt Source field then click on Search.



Now you should see the Float_1 tag that we created, and any other tags you may have entered. You will want to highlight it by clicking on it with the mouse and then clicking OK to add it to the Trend properties.

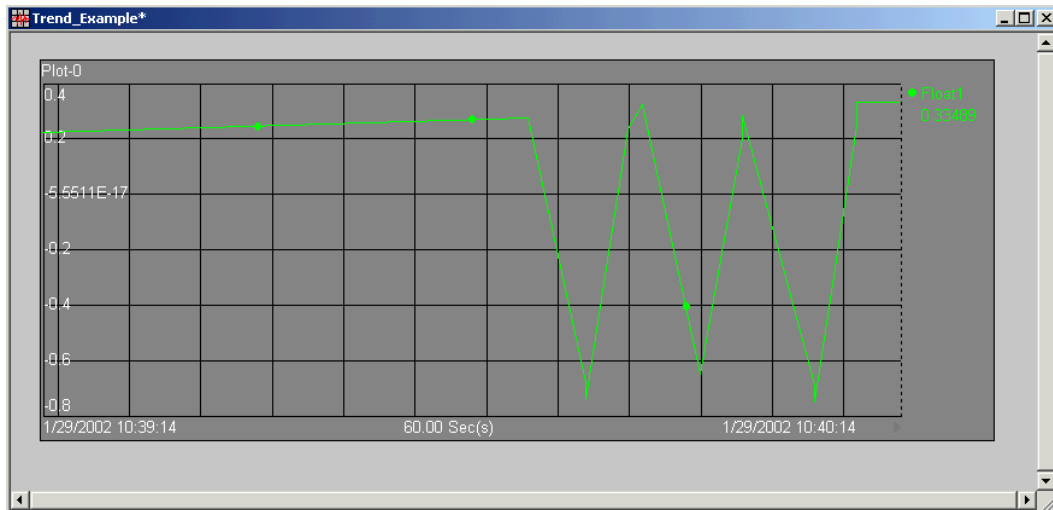


The tag name will now appear in the “Tags in plot” field of the Define Trend dialog. You still have one thing left to do. The tag that you are using in the server changes value about every 100msec. You want to be able to see these changes in the trend. The trend’s default display width is 8 hours. Change the Plot Start value to read “*-1 Minute”, then click OK.



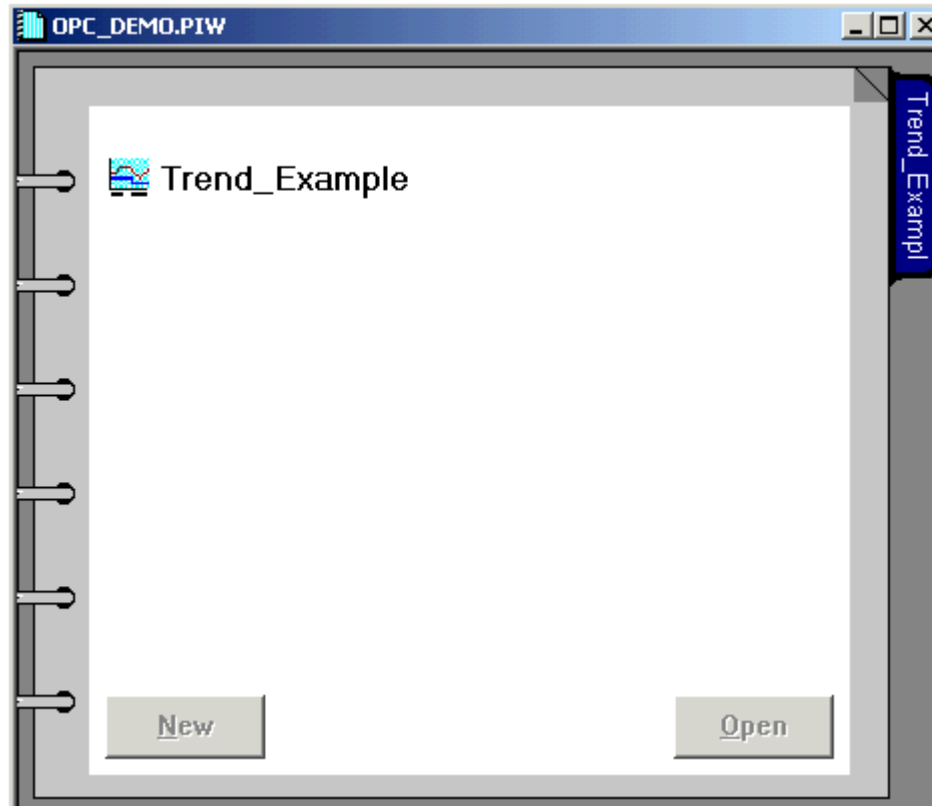
Viewing Tag Data in the Trend

You should start seeing values plotted in the display right away. You may want to adjust plot times to get your data displayed correctly.



Save and Close the Project

Once you save and close the display window, you will see it displayed in your process book. Don't forget to save your ProcessBook as well.



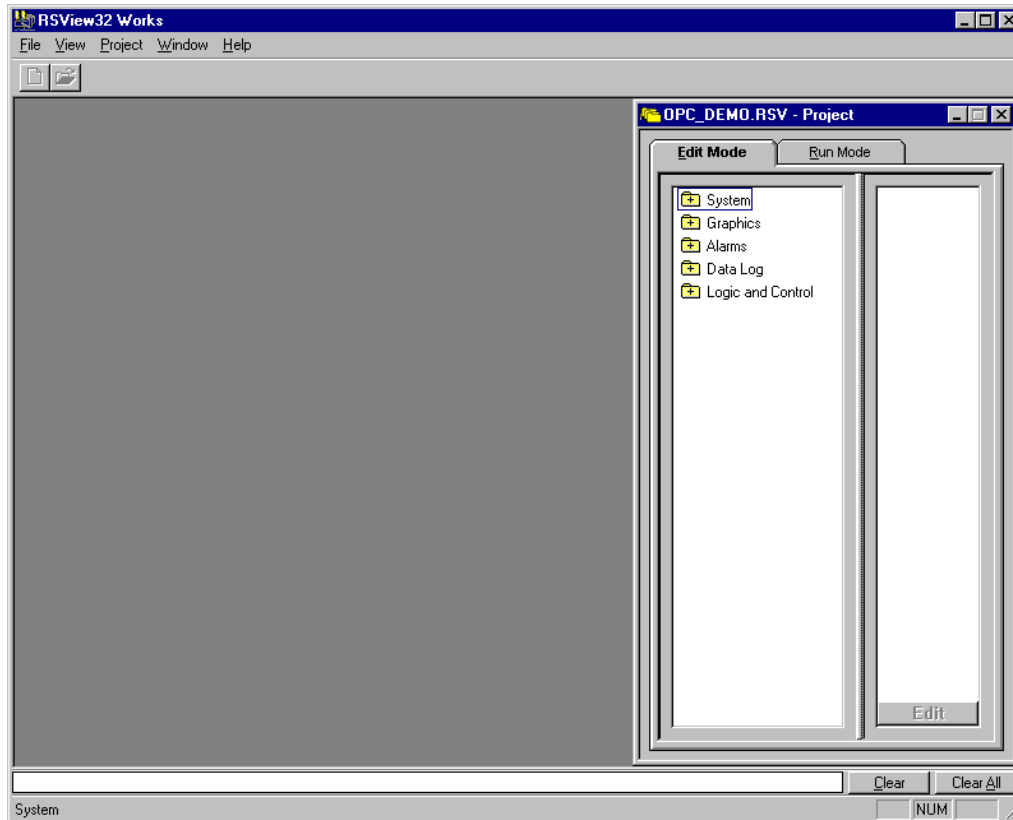
You should now be able to connect to, and display data from the KepserverEx with OSI PI. For more information on using PI you will need to contact your local OSI PI representative.

Rockwell Software's RSVIEW32® as an OPC Client

Connect to the Server from RSVIEW32

Rockwell's RSVIEW32 OPC client is one of many HMI's that can be used to connect to KepserverEx. The RSVIEW32 OPC client version used for this example was version 6.3. The following steps will show you how to create an OPC connection to the Server from RSVIEW32.

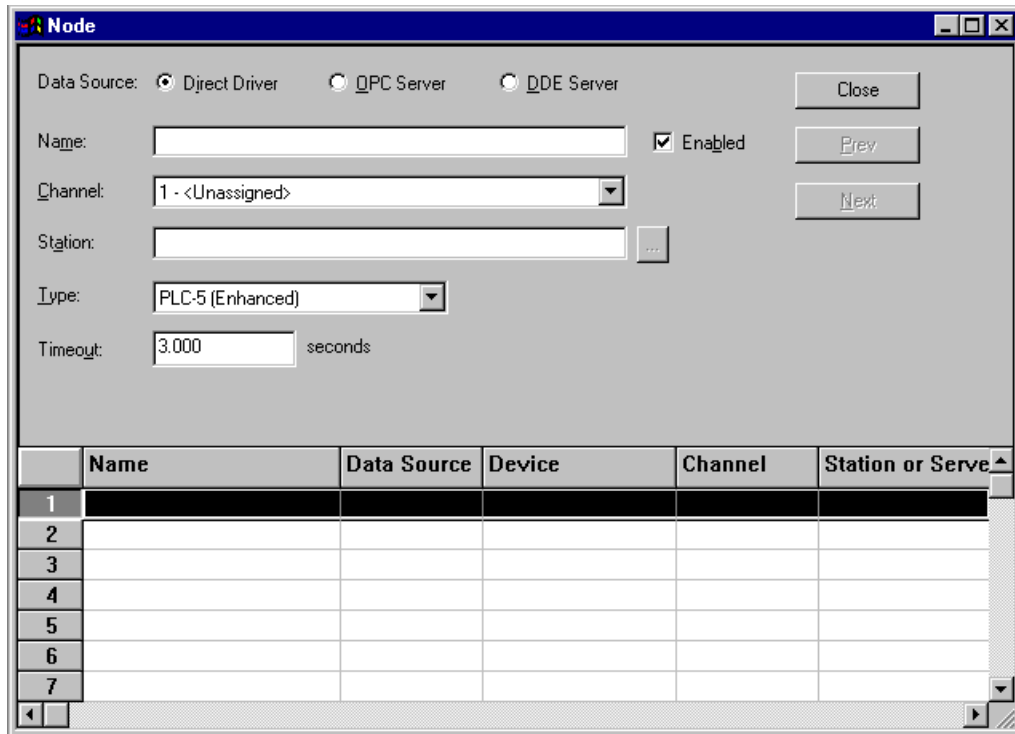
1. Open your copy of RSVIEW32 and start a new project.



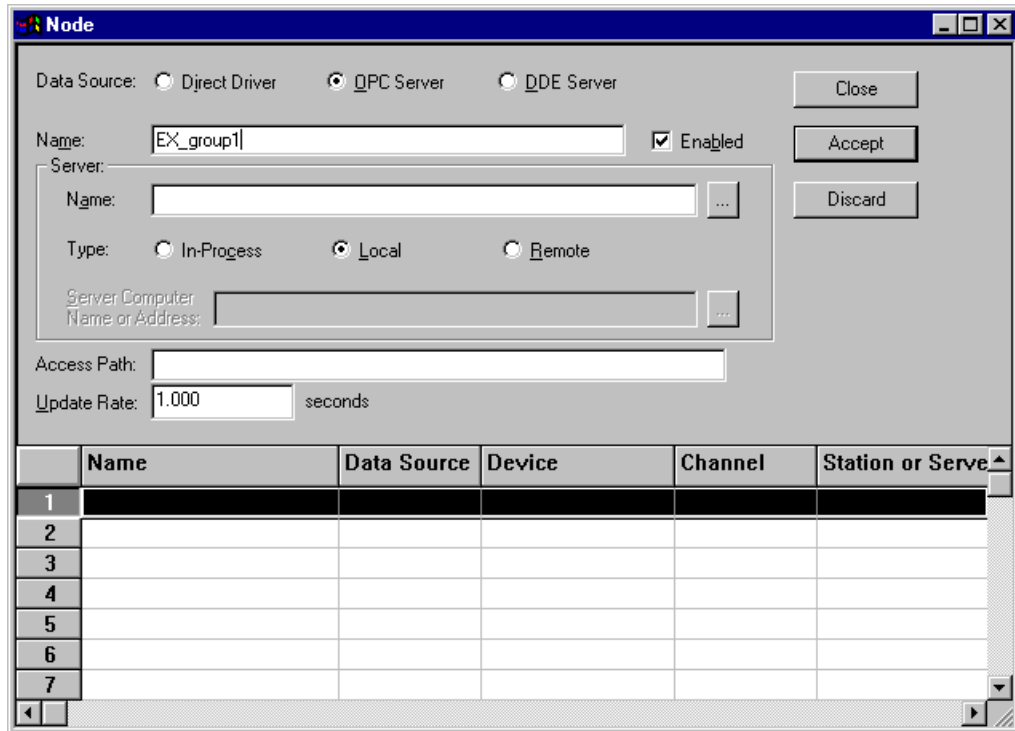
KepserverEx is capable of being an OPC or DDE server to RSVIEW. If your node is defined as an OPC node in RSVIEW, then KepserverEx will be an OPC server to RSVIEW. Likewise, if the node is defined as a DDE node in RSVIEW, KepserverEx will be an Advanced DDE.

Create a Device Node

2. In RSVIEW32 click on the System folder and then click on Node in the Project Control Panel. If you have multiple devices in the KepserverEx project, you may want to create multiple nodes in RSVIEW and link them to the devices in the server.




3. First we will select a Data Source for the Node; in this case it will be OPC Server.



4. Next you will assign a name by which RSView will refer to the node. In this case we called it "Ex_group1" but you can assign a name of your choice.

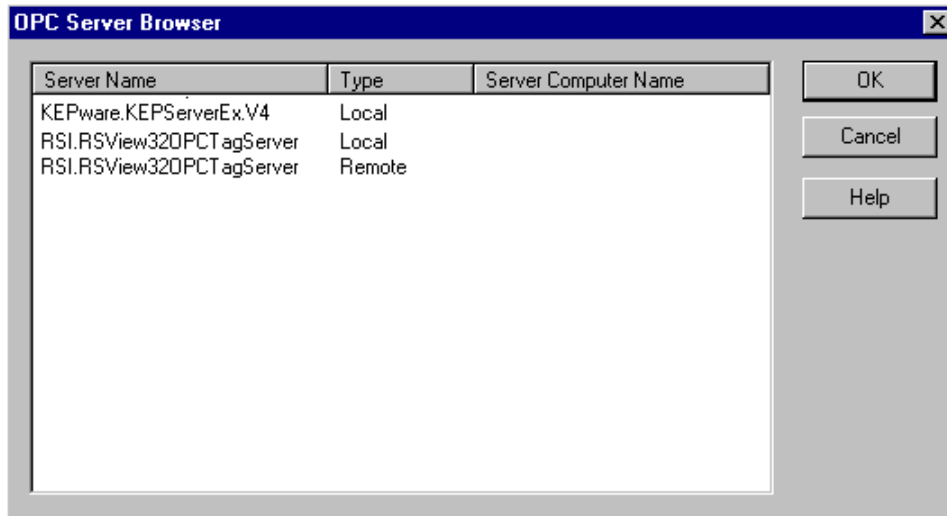
Browse for an Available Server

5. In the Server section you need to perform several tasks. First, you will click on the browse button  and select a valid server Program ID for a local or remote server. .

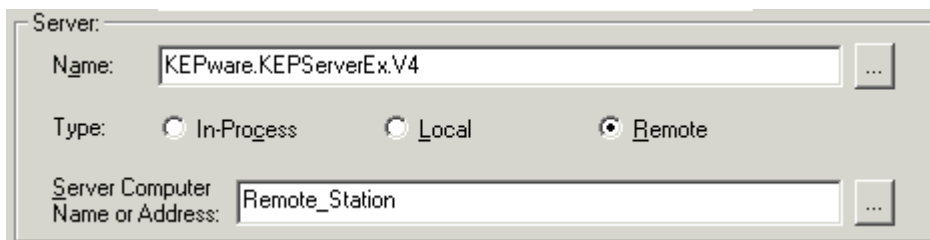
The list displayed in the OPC Server Browser is pulled from the local PC even for remote connections. Because of this you will have to configure the DCOM settings for the server, see Kepware's "DCOM Configuration for KepserverEx" guide

If you have your server running on a Win 95 or Win 98 box you will have to have DCOM installed, configured, and running, in order to allow access to the server.

6. Select "KEPware.KEPServerEx.V4" from the list of servers.



7. In the Server section select the Type of connection you are going to make to KepserverEx. Select Local for connection to a server on the same PC as RSView. Select Remote to connect to a server running on another PC. Selecting Remote will enable the Server Computer Name or Address field. You can enter either a name like "Remote_Station" or IP address like "120.150.11.90" for the PC running the server.

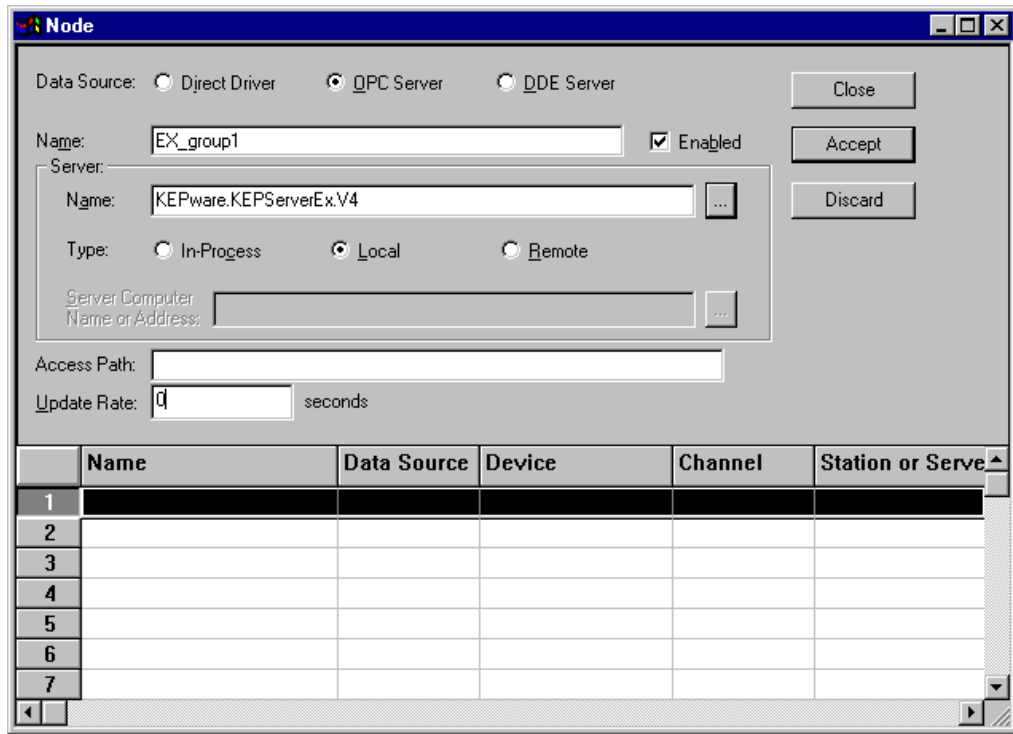


RSView allows you to create the node with or without an Access Path. We have left this field blank. By leaving the Access Path blank RSView is able to link to any tag within the server from this node. This means you can access tags from multiple devices in the KepserverEx application. If we had entered an Access Path, then only tags from a single device could be accessed through this node.

Make sure that Enabled is checked so that the node will connect to KepserverEx when you run your RSView project.

8. Set the Update Rate to 0.

This tells RSView that KepserverEx will use the fastest possible rate to send data. If you design a large RSView project with multiple nodes, you may want to slow down the update rates of your nodes

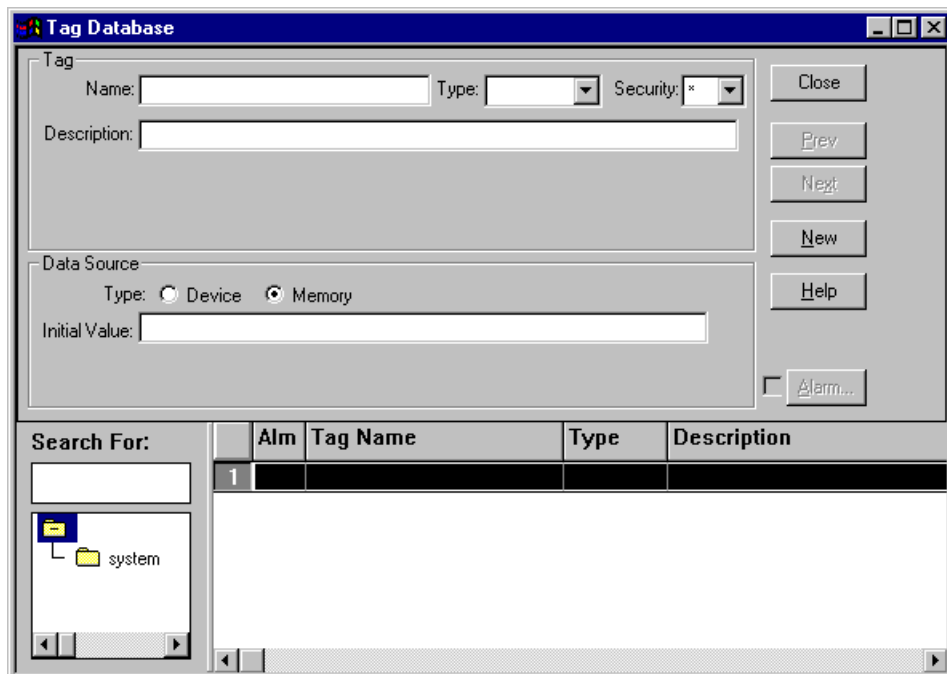


9. Click Accept and Close in the Node dialog box.

Create a Tag

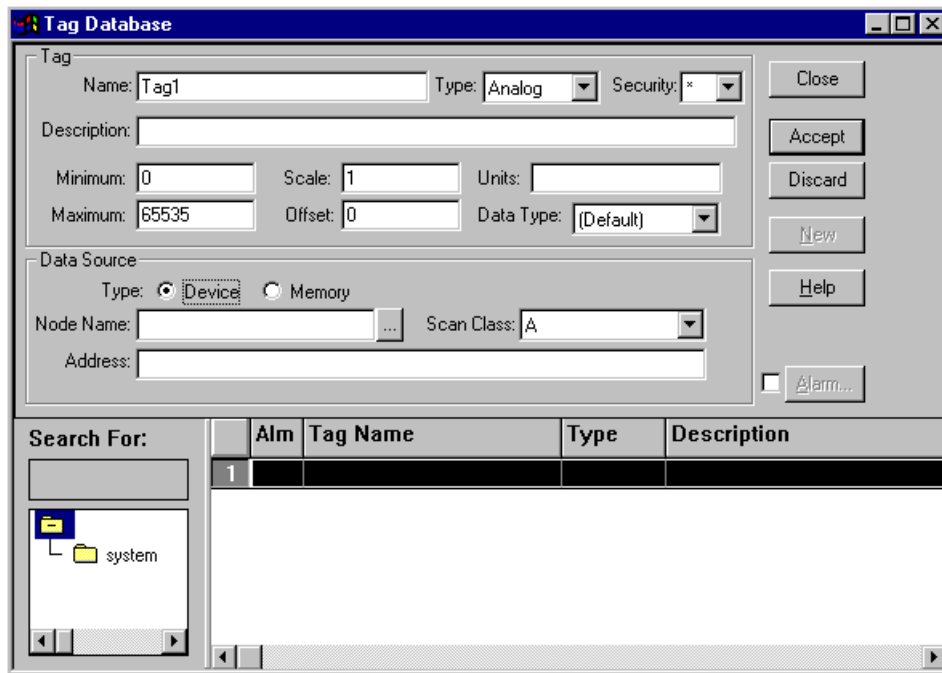
Next, create a tag or tags to access the data in the devices.

10. In RSView32 click on the System folder and then click on Tag Database in the Project Control and open the Tag Database dialog box.

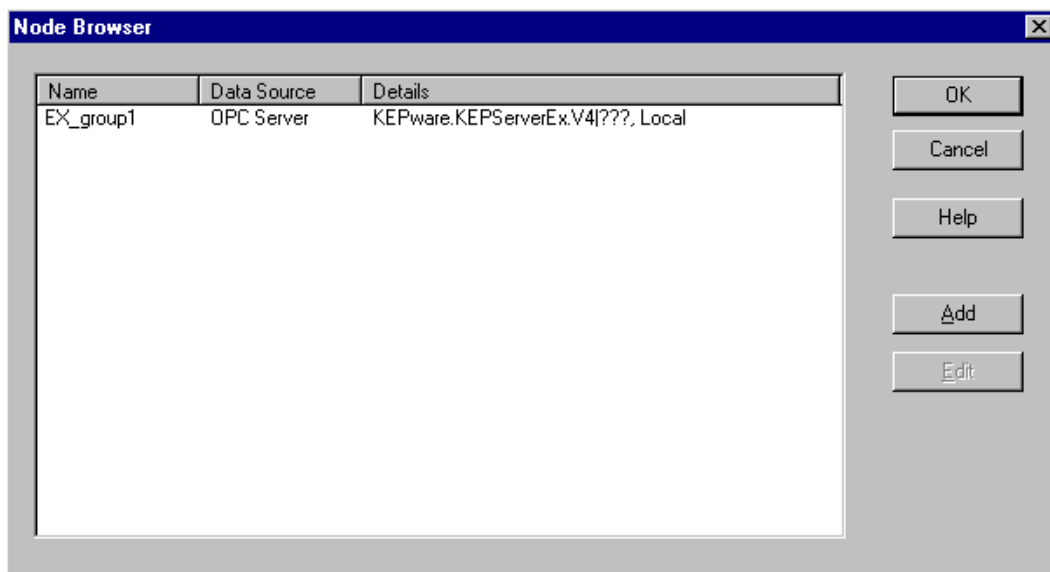


11. In the Tag Database, enter the tag description information as you would for any Rockwell driver

12. In the Data Source section Click on the Device radio button.



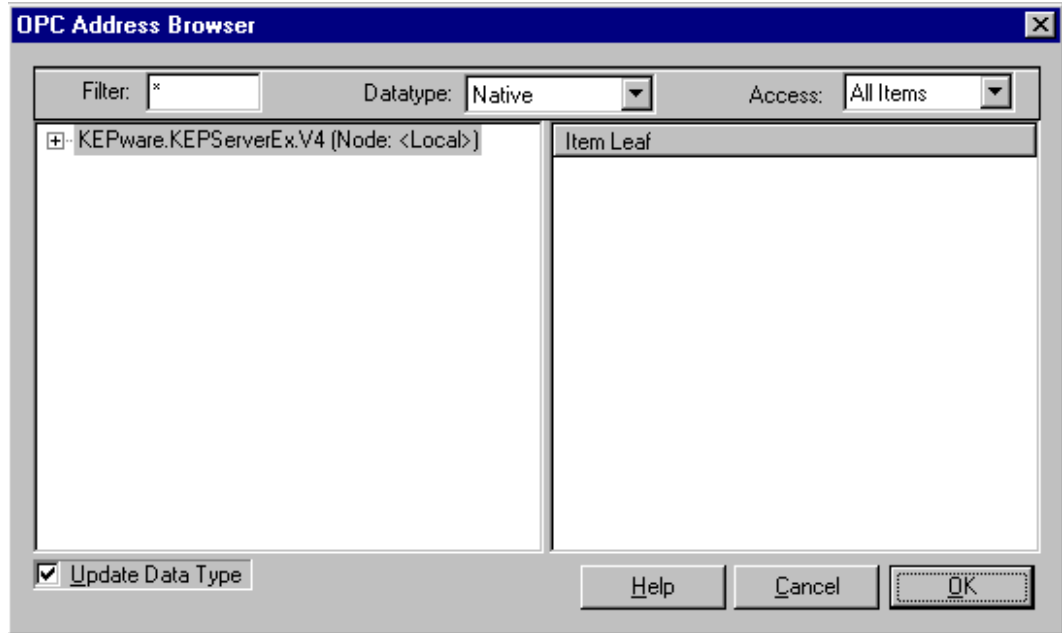
13. Click on the Node Browser button  to display the Node Browser window.



14. In the Node Browser select a Node Name and click OK to assign the node to your tag. In our example we have only the “Ex_group1” node that we added

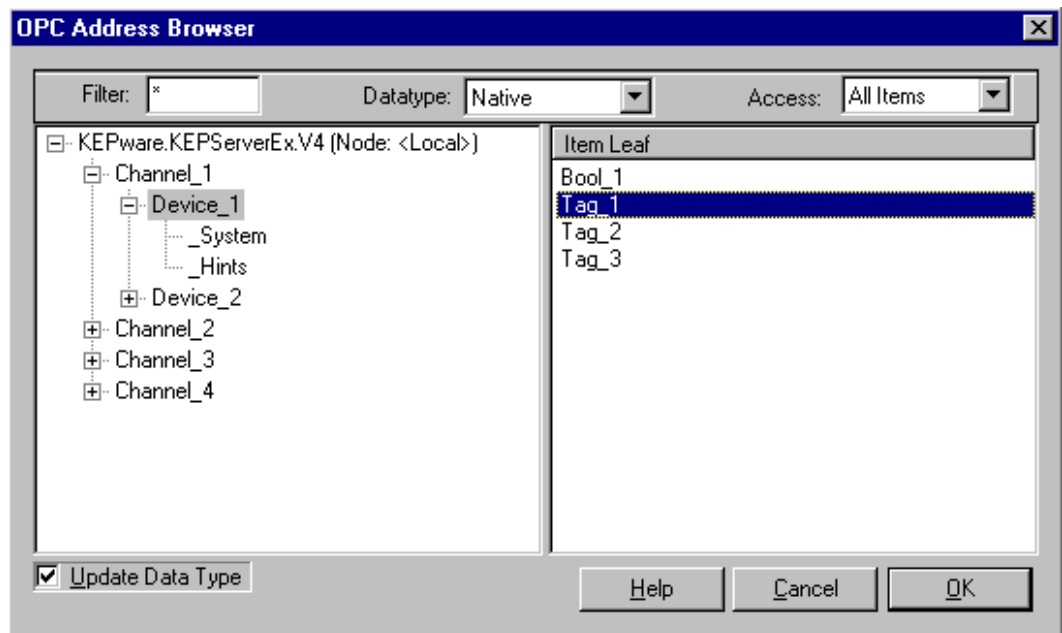
Browse for Tags in the Server

15. Click on the Address Browser button  located to the right of the Address field.

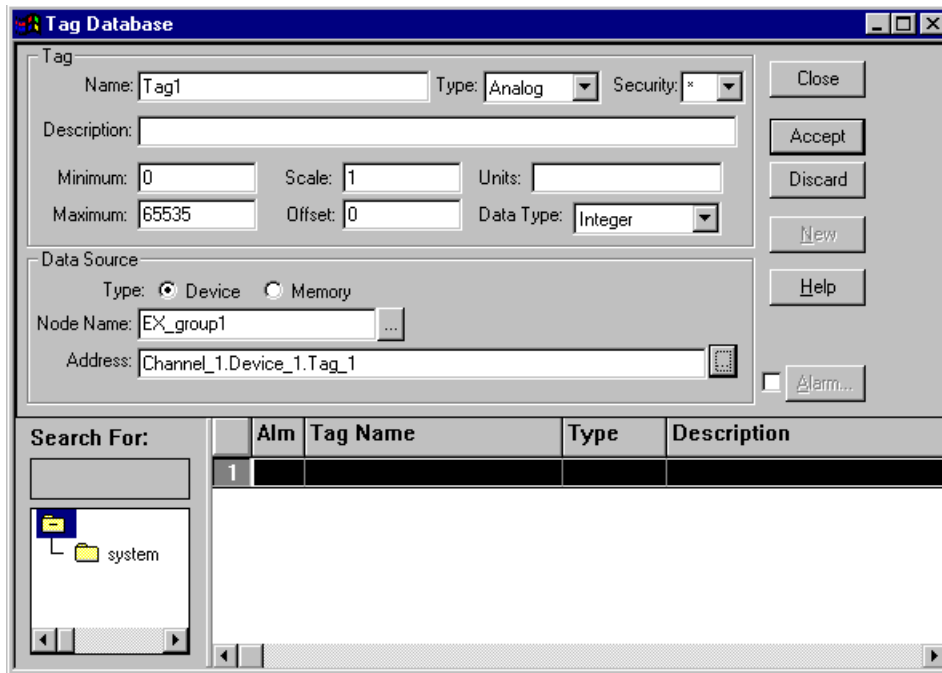


- Using the tree view in the left pane of the OPC Address Browser window, select the Device or Group from which you wish to select Address items/tags. We are selecting Tag_1 in Device_1 on Channel_1.

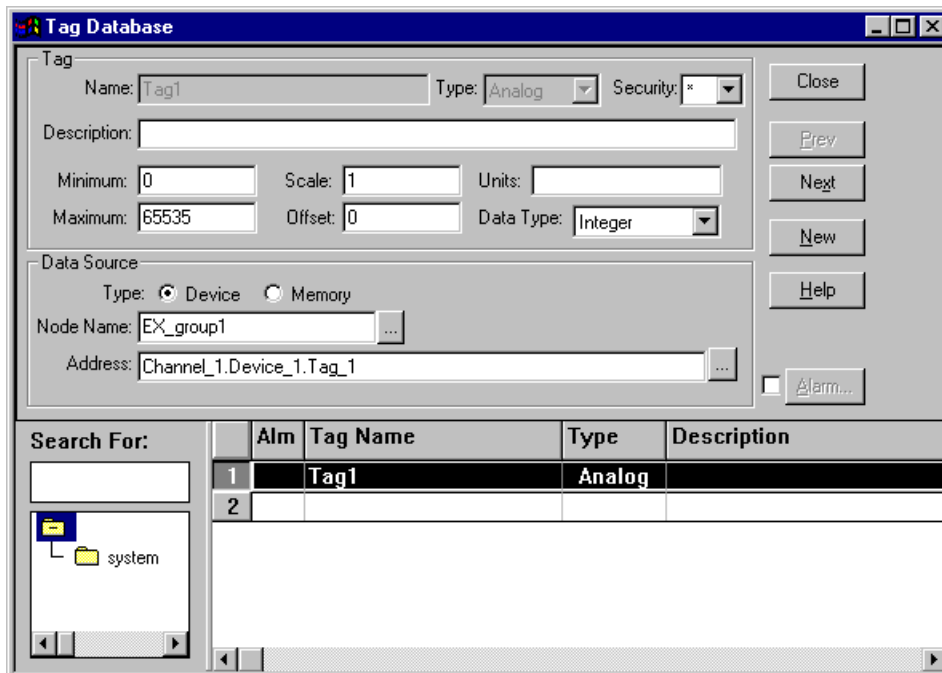
Remember that these are tags that have already been defined in our Simdemo.opf project.



- After you highlight the tag, click **OK** to add it.



18. Now click Accept to add the tag to the RSView tag database.



19. Click Close to close the tag database.

Check the OPC Connection

Next we are going to verify a connection to the KepserverEx “Simdemo” project.

20. To do this, double click on Tag Monitor in the System Folder.

	?	Tag Name	Value	State	Description
1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					

21. Enter the Tag Name or names that you want to check. In this case we entered "Tag1" which is the name of our tag that we added.
22. Hit the Enter key after typing in the tag name.

	?	Tag Name	Value	State	Description
1	Y	Tag1	1220	valid	
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					

23. You should see a State of "valid" in the Tag Monitor window for the tag and a Value that is ramping or incrementing very quickly.
24. You have now established a good OPC connection to KepserverEx from RSView

Upgrading your connectivity from Kepserver to KepserverEx

KepserverEx is the latest generation of Kepware's OPC server technology. Building upon the original KEPserver, KepserverEx has incorporated many of the features requested by Kepware's customers. In addition to customer driven enhancements, many technological changes have occurred. These features and enhancements have all been made with the goal of providing an OPC server that demonstrates unparalleled compatibility and performance.

If you are an existing Kepserver user you may want to consider upgrading your server to KepserverEx. Please contact Kepware to inquire about upgrade pricing.

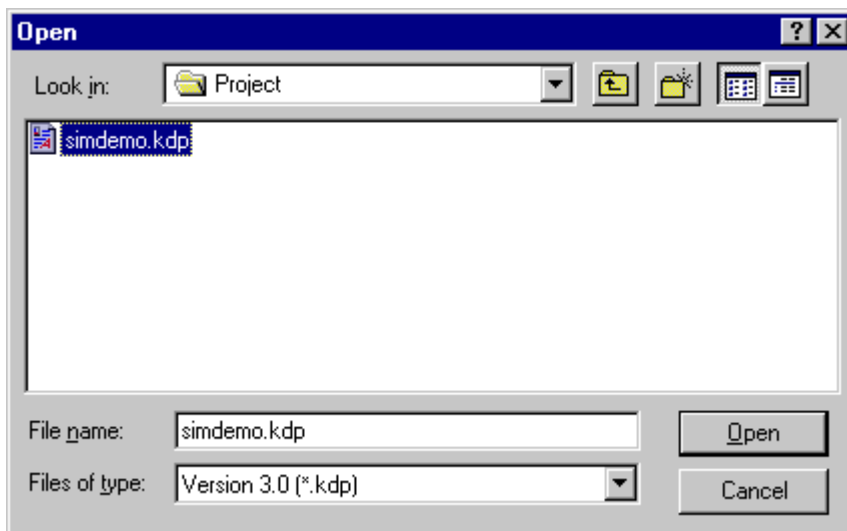
If you have a copy of KepserverEx and you would like to test connectivity with your existing RSView application please use this section to convert your Kepserver and RSView applications. Although these upgrades should be very simple, we recommend saving backup copies of your projects in case you encounter any problems.

This section is for users who are interested in upgrading from an existing Kepserver version 3.21 or earlier to Kepware's latest enhanced OPC server KepserverEx V4.0.

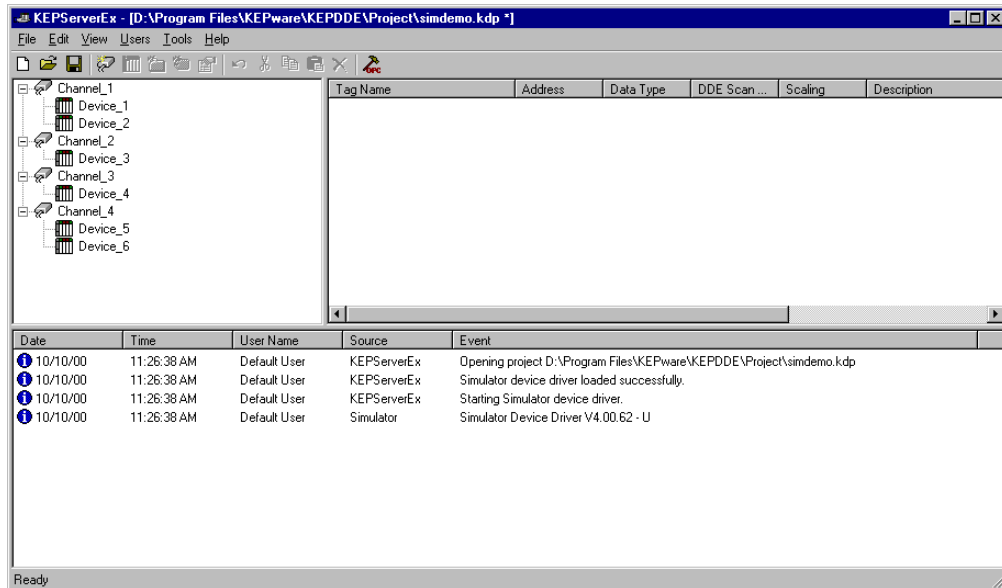
This section is for users who are interested in upgrading from an existing Kepserver version 3.21 or earlier to Kepware's latest enhanced OPC server KepserverEx V4.0.

Converting From Kepserver OPC to KepserverEx OPC

1. The most important thing to know is that you do not have to recreate your server project. All that you have to do is open your existing .kdp Kepserver project file in KepserverEx. Select File|Open from the Main menu of KepserverEx.



2. In the File Open dialog select "Version 3.0 (.kdp)" from the Files of type drop down box.

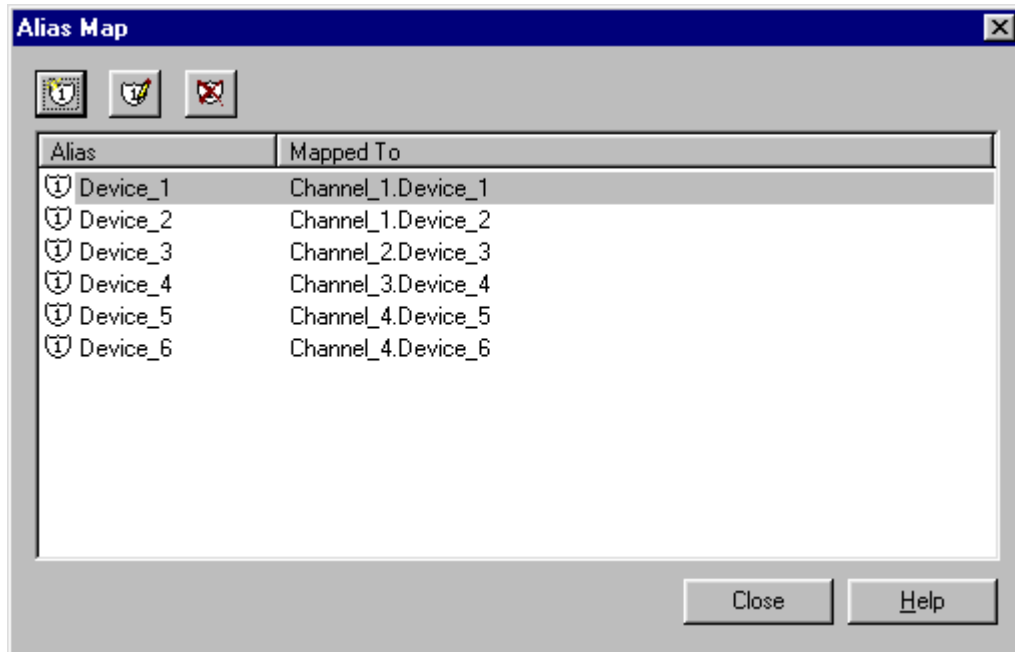


Once the project is open in the new server you will need to save it as a new server project. This provides two functions for you. First it converts the project to the new server project model (with a .opf file extension). Secondly, it enables the backward compatibility in your project so that the impact of converting to the new server is minimal.

3. Click File|Save to save your project.

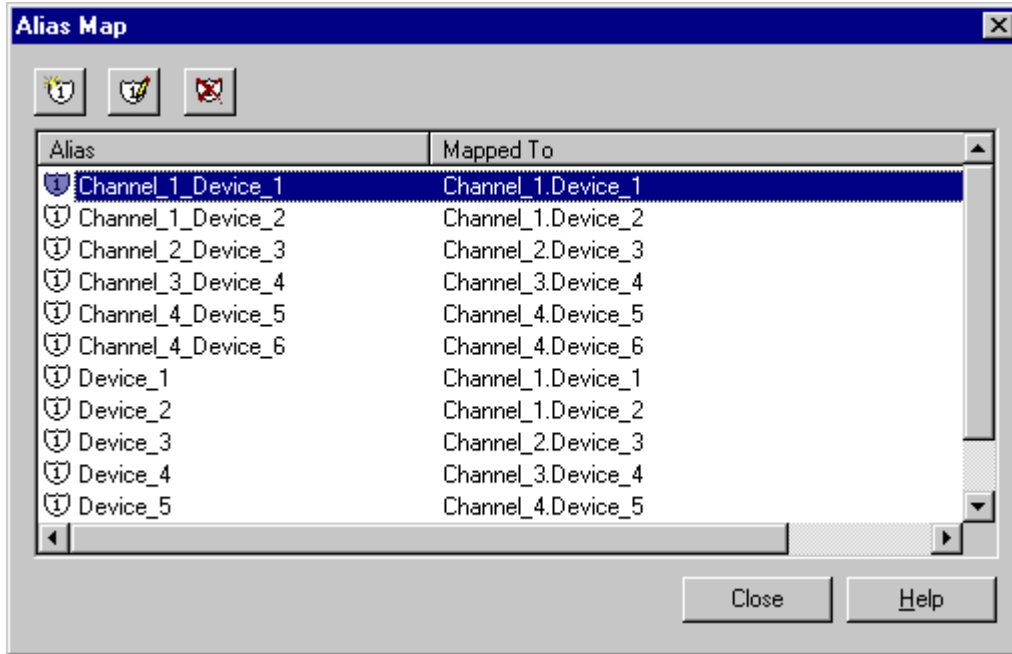
If you refer to the Simdemo.opf project that installs with KepsServerEx, you will be able to directly refer to the Alias map shown in the next step.

4. To see how this is accomplished click on Edit|Alias Map... in the main menu.



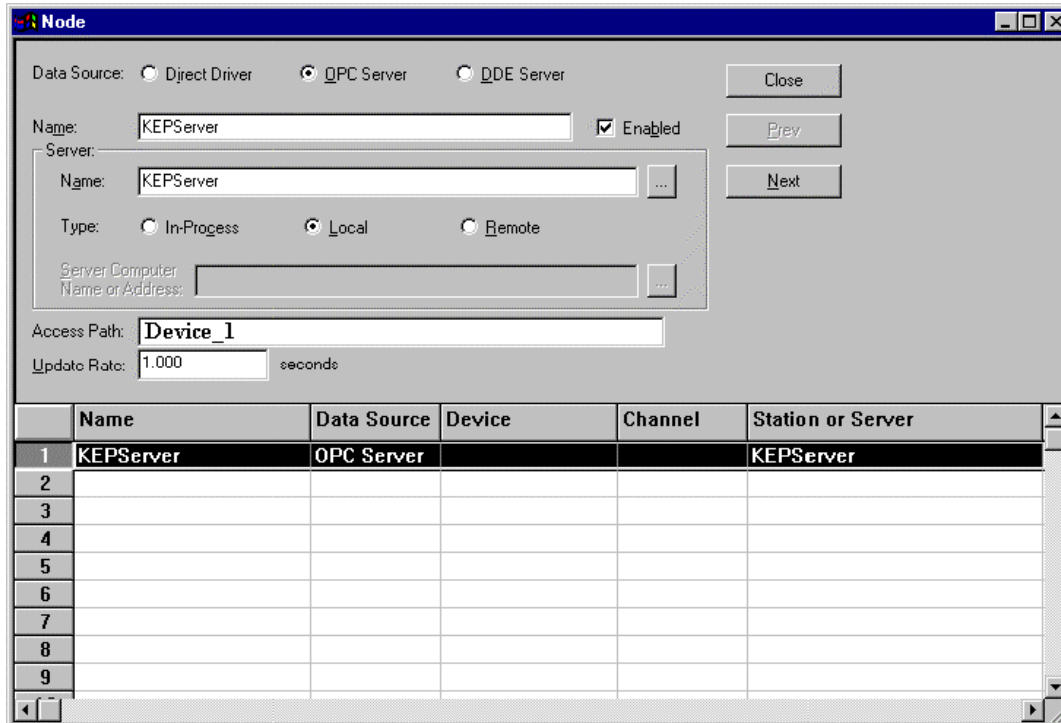
You will notice that an Alias was created for every device in the project and that each Alias was mapped to its appropriate channel and device. In KepsServer V3.21 and earlier, items were referenced at the device level ("device.item"). In KepsServerEx V4.0 and newer, items are referenced through the channel and device ("channel.device.item"). The beauty of this is that you can upgrade without having to make any changes at the tag level of your OPC client application. You only need to make reference changes at the node level, as you will see in the next step.


If you were to save the server project and re-open it and look at the aliases, you would notice the addition of aliases. These aliases are automatically created and updated by the server whenever you add, modify or remove channels and devices in the server in any project.



In your RSVIEW32 project you will need to edit the node and update the OPC Server Name that is referenced.

- Go to nodes from the System folder then select the Kepservers node.



- In the Server selection area, either click the browse button  and select the new server or enter “KEPware.KepserversEx.V4” as the new server name.

In Kepware documentation, this server name is also referred to as the Program ID or Prog ID. If you have an existing Access Path entry, you can update it to reflect the newer item connection

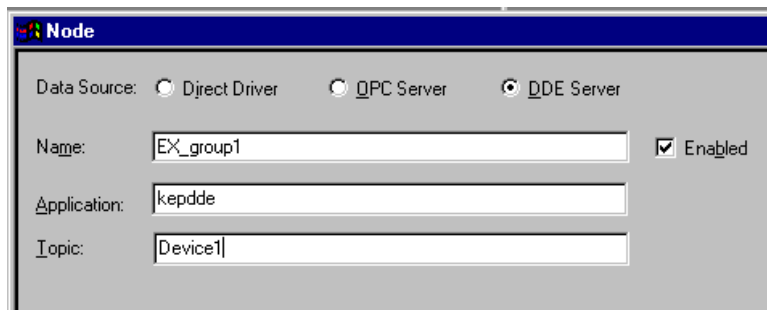
format (i.e. channelname.devicename). However, it is not mandatory to edit the Access Path entry.


7. Click Close to accept the new changes.

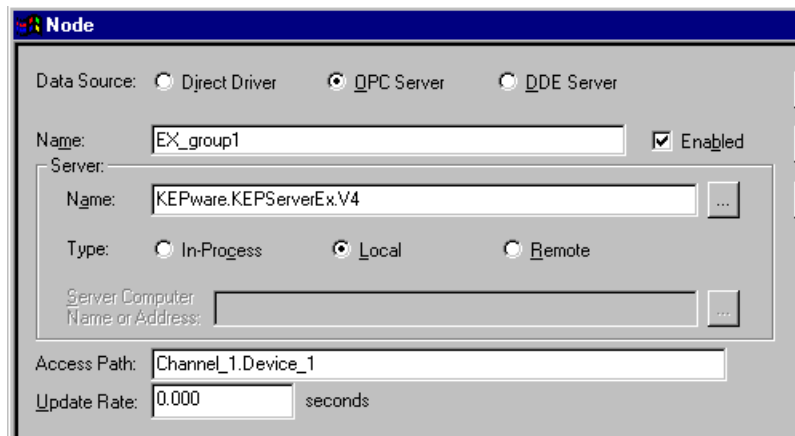
Now you should be ready to save and run your project. You should not need to make any other changes to your RSView32 project, the Alias names created in KepserverEx should take care of all existing tag references

Converting From Kepserver Advanced DDE to KepserverEx OPC

1. See steps 1-7 in “Converting from Kepserver OPC to KepserverEx OPC” about converting your old server project to the new server model
2. Next, in RSView32 go to Nodes
3. Select your existing DDE Node. Remember that a DDE node points at the specific device whereas an OPC node points at the server



4. Next you will click on the OPC Server radio button.
5. Click on the Browse button  and select “KEPware.KepserverEx.V4”. Next, you will need to enter an access path that points the server to the device that it was looking at in DDE, ours is “Channel_1.Device_1”.



6. Click Accept to accept the changes to the node and then close the Node dialog.
7. You should now be able to save and run the project.

Converting From Kepserver Advanced DDE to KepserverEx Advanced DDE

1. Although we highly recommend the performance gains of OPC over DDE, in some cases you may want to convert to the new server but not use OPC. In that case you should follow steps 1-7 in “Converting from Kepserver OPC to KepserverEx OPC”.
2. It is very important to convert the project because the new server does DDE differently than the old one did. In the Old Server you connected to each device individually as a

topic. In the new server there is one topic “_ddedata” and all items are accessed via the placing the full path in front of the actual item. For example Tag_1 on Device 1 would look like this “Channel_1.Device_1.Tag_1”.

3. To make it easy we use the alias so that we do not have to change every item address.
4. You should now be able to save and run the project.

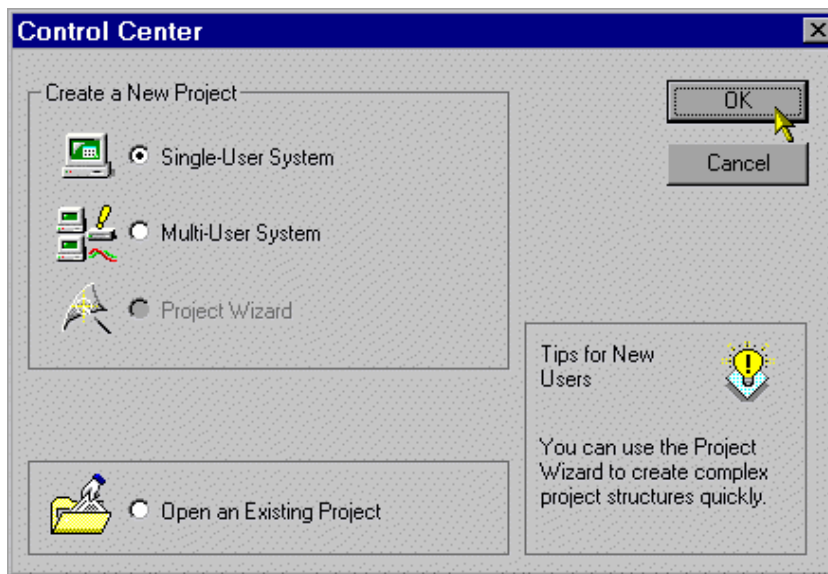
Siemens' WinCC as an OPC Client

Connect to KepserverEx from WinCC

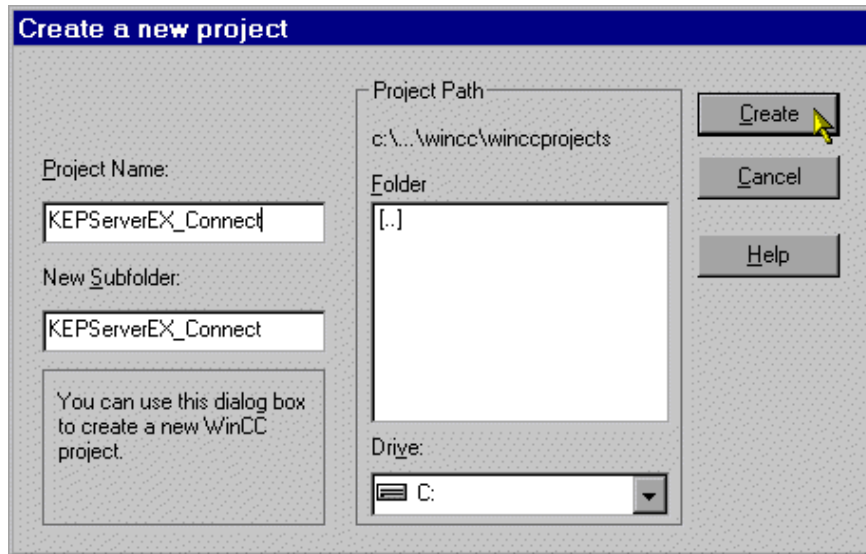
The following steps will show you how to create an OPC connection to the Server from WinCC.
Our example uses WinCC version 4.0.2.

Create a New WinCC Project

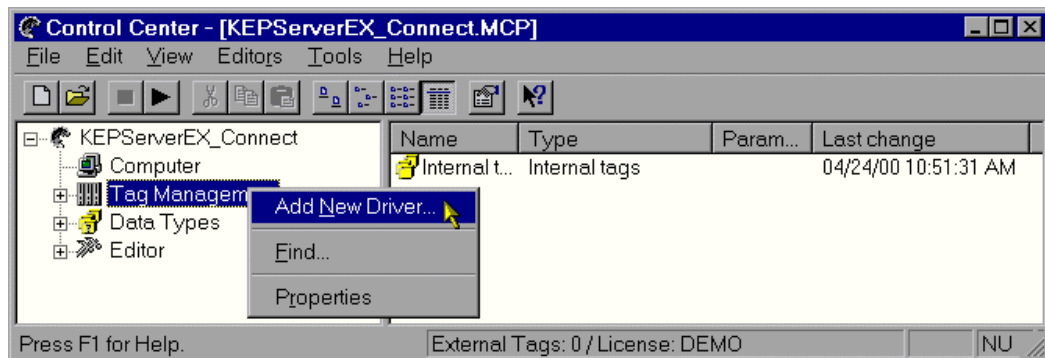
1. Start up the KepserverEx "Simdemo.opf" project.
2. Open your copy of WinCC to start a new project.
3. If there are no existing projects, the Create A New Project dialog will automatically appear, otherwise choose File from the Main Menu and New from the drop down.



4. For this example, select the Single-User System radio button and then OK.

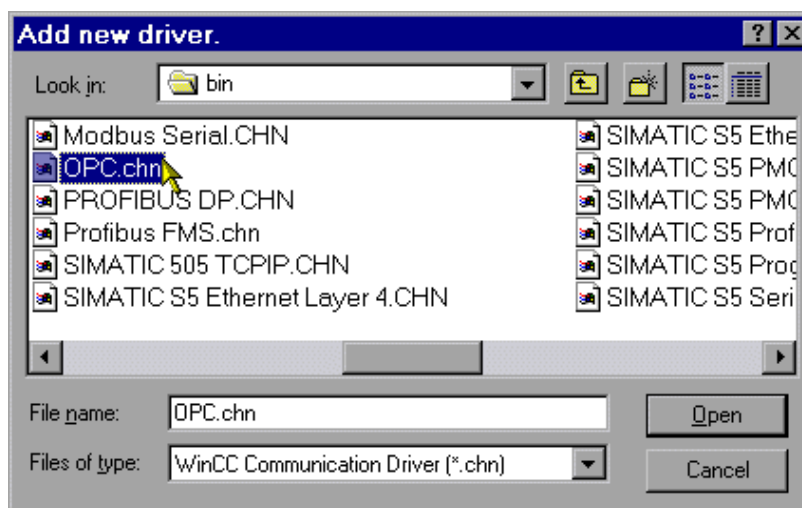


5. The next step is to create a project name and directory path. Here, the Project and Subfolder are both named “KEPServerEx_Connect”. Choose Create when finished.



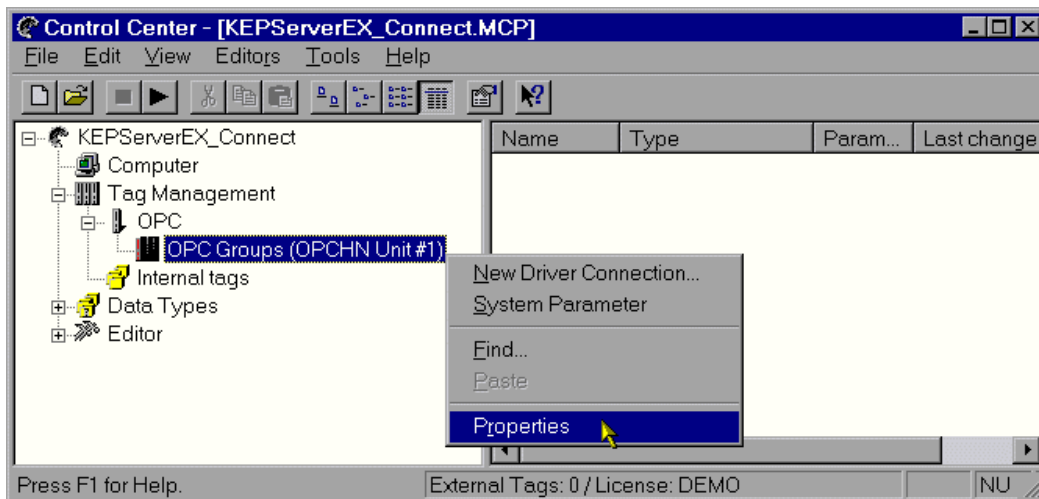
Add a Driver to the Project

6. In the “tree view” of the main Control Center window, right click on the Tag Management module and select Add New Driver from the drop down menu.



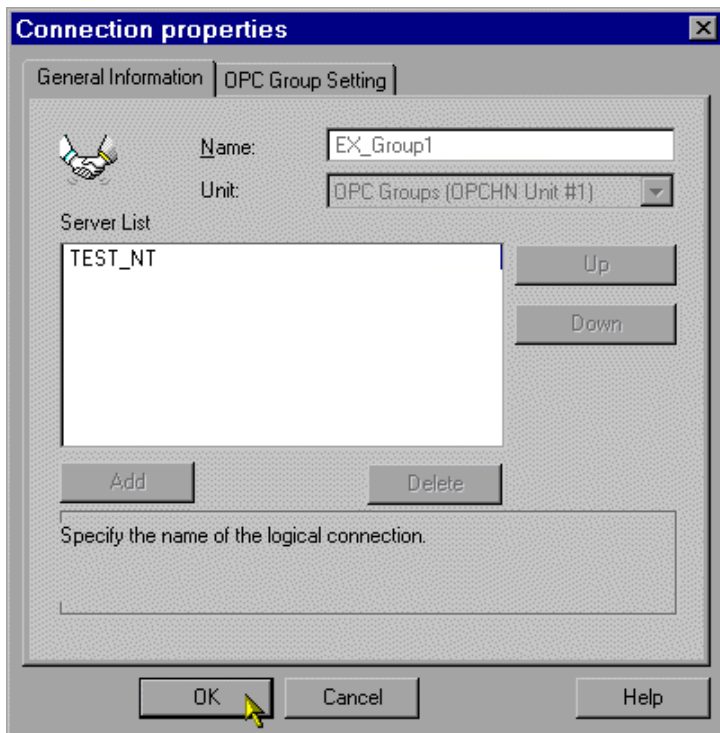
7. Select the driver named “OPC.Chn” (OPC Channel) and choose Open.

- Next, right click on the newly created OPC Groups module available under Tag Management\OPC. Choose Properties from the right click menu.

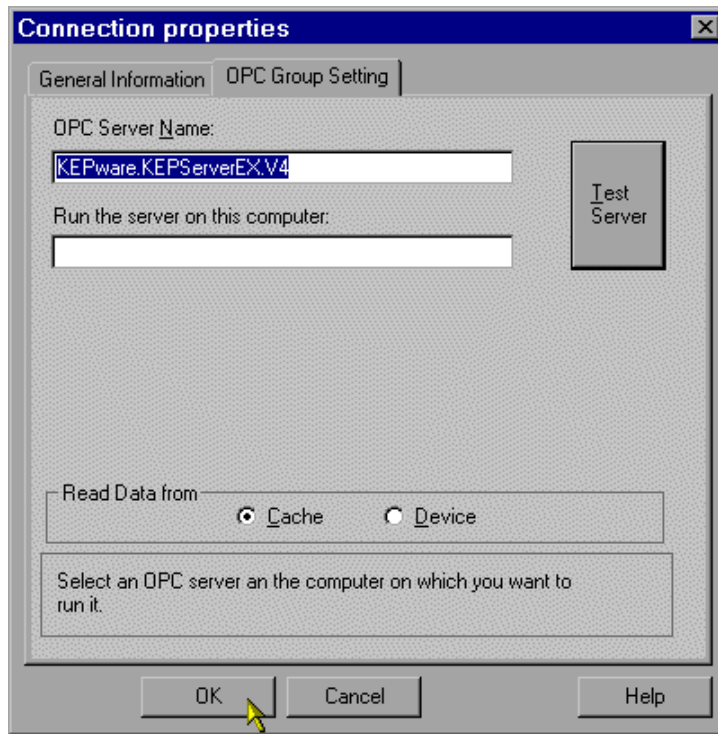


Create Driver/Group Connection

- Click on New... in the Channel Unit Properties window to get to the main Connection properties window.

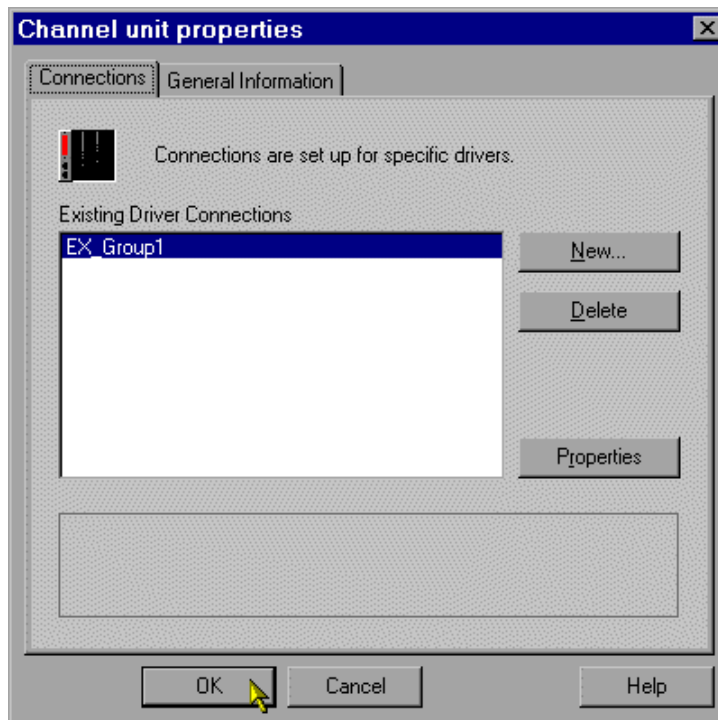


- In this example we have named the OPC connection as “Ex_Group1” while our computer name remains as “TEST_NT”. After appropriately filling in the General Information tab, click on the OPC Group Setting tab at the top of the dialog box.

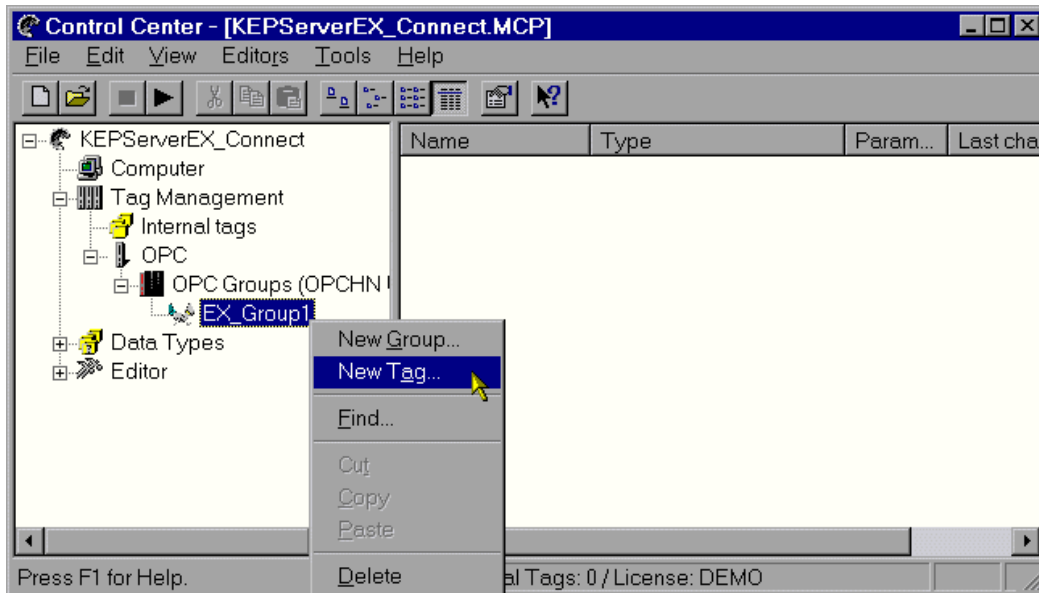


If the server test fails, check the OPC Server Name field to make sure it matches our example.

11. In the OPC Group Setting dialog, enter “KEPware.KEPServerEx.V4” as the OPC server name. Press the server test button to make sure that Control Center can properly invoke the server. Finally, choose which data source to receive data from (Cache is recommended). Click OK when done.

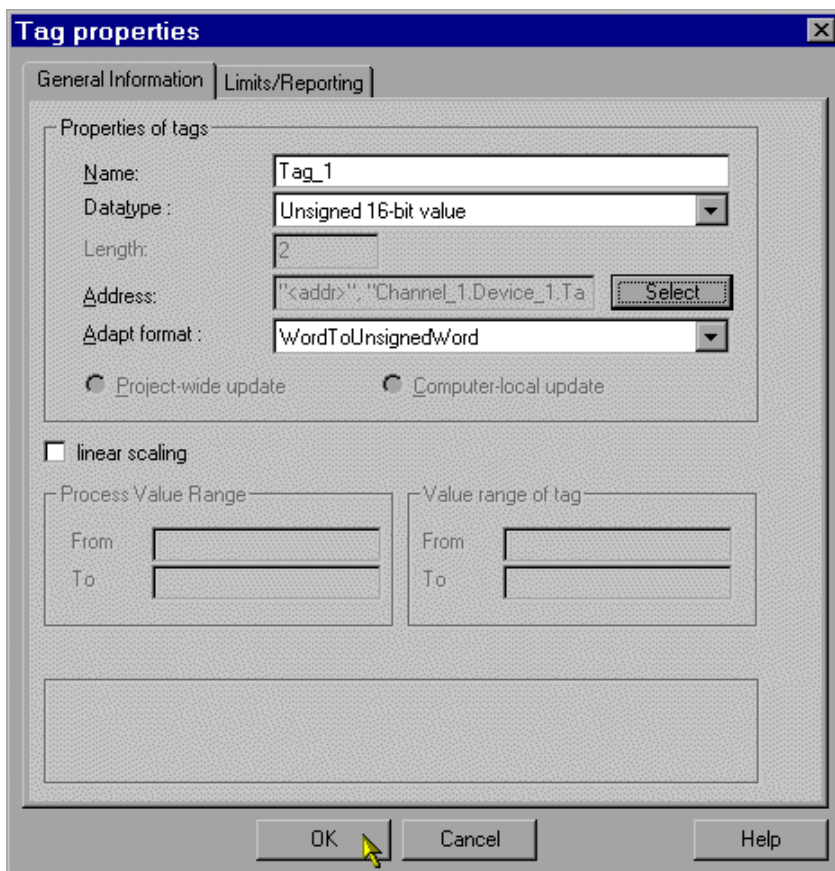


12. The Channel unit properties dialog box now contains the newly created OPC group connection. Click OK to continue.

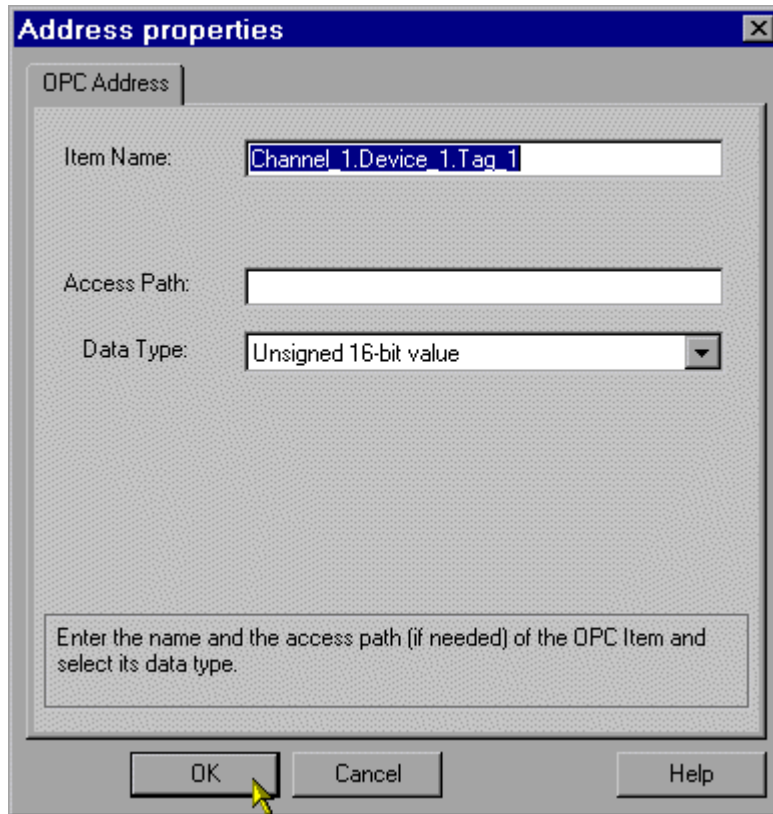


Create A Tag

- The next step is to create a tag under the new OPC group. Right click on Ex_Group1 and choose New Tag... from the right click menu.

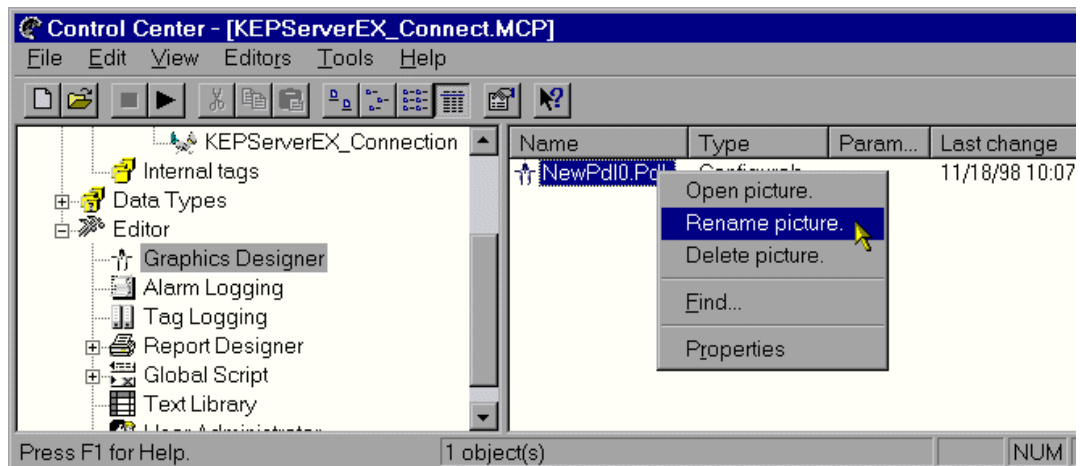


- In this example we use an Item that has been predefined in KepserverEx (known as a User Defined Tag) called "Tag_1". Click the Select button to define a tag item.



15. In the item name field, enter the address of the tag item. Also, choose the most appropriate Data type for the tag item and click OK.

Note: In the Item Name field, you may also ask for a dynamic tag by replacing the tag name with an actual address. e.g. Channel_1.Device_1.R001

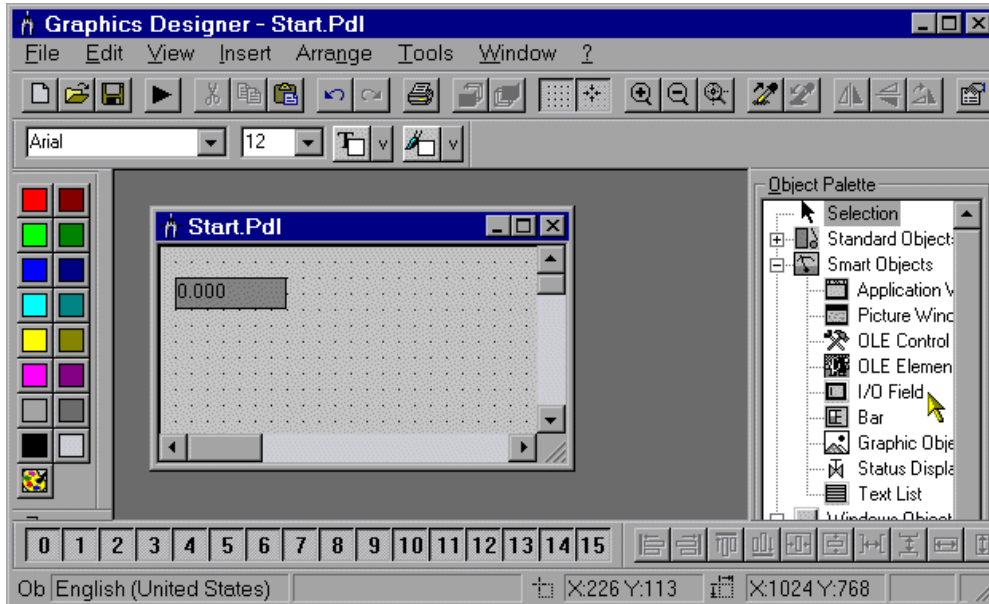


Create Graphic Display

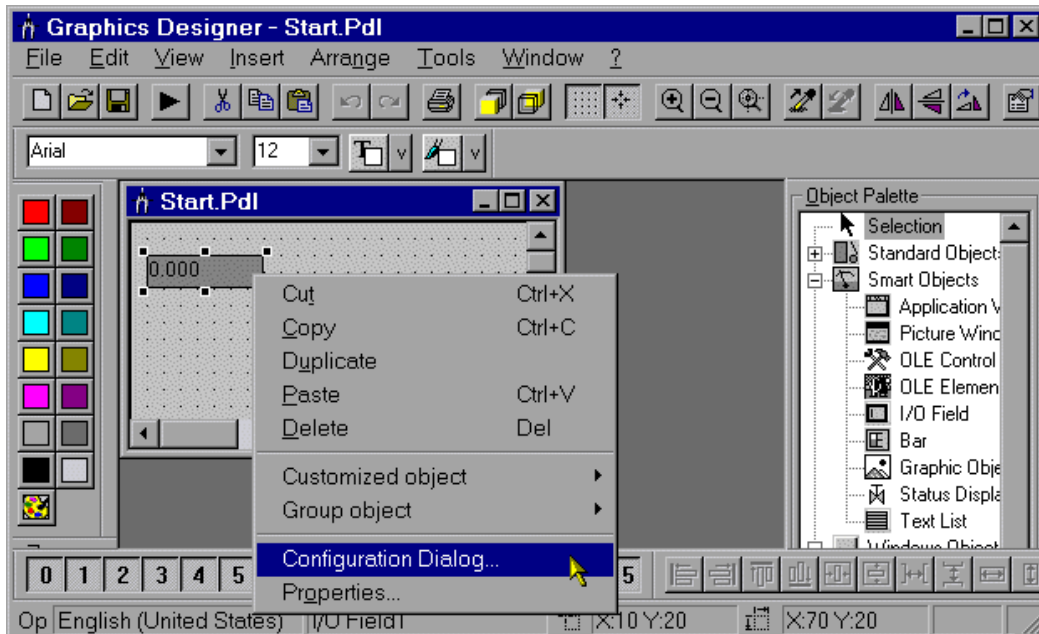
1. Now that a connection and tag have been defined, we must now create a graphic display of the tag item. Right click on Graphics Designer under Editor in the tree view. Open a picture and right click on that picture to rename it. Rename the picture to "Start.pdl", as this is looked for when WinCC executes runtime. Lastly, click on OK.



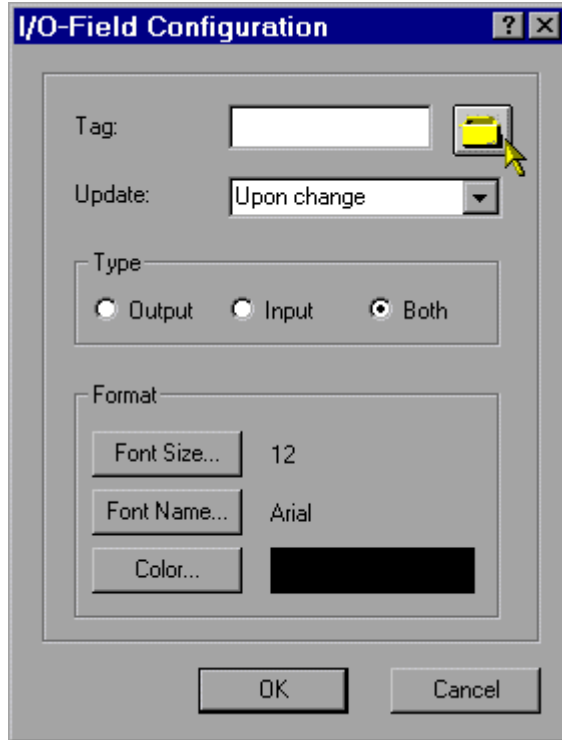
- Now double-click on the picture's name to start up the Graphics Designer.




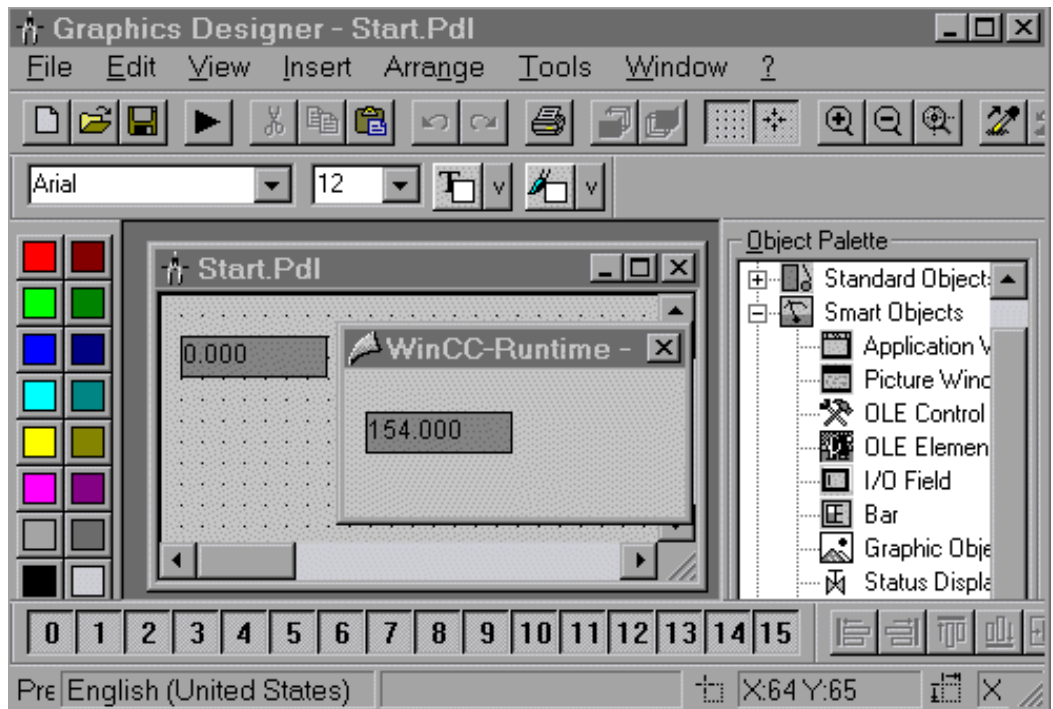
- In the Graphic Design window, choose I/O Field object from the Object Palette. Next, right click on the picture window to create the I/O Field display object that will be used to display data during runtime in the window. In order for this display object to function correctly, there must be link created from the object to a defined tag item.




- Right click on the I/O Field object and choose Configuration Dialogue... from the right click menu.



- In the I/O – Field Configuration dialog, click on the Tag button  to select a tag in the Select Tag dialogue box. Once a tag has been chosen, click OK in each dialog box to exit to the main Graphics Designer screen.



View Data and Check the OPC Connection

6. The display object is now linked to a tag item. Tag data can now be displayed in runtime by pressing the Play button  on either the Graphics Designer or Control Center menu bar. As a quick test, check the Status Bar at the lower right edge of the KepserverEx window. You should see at least one Active Item.

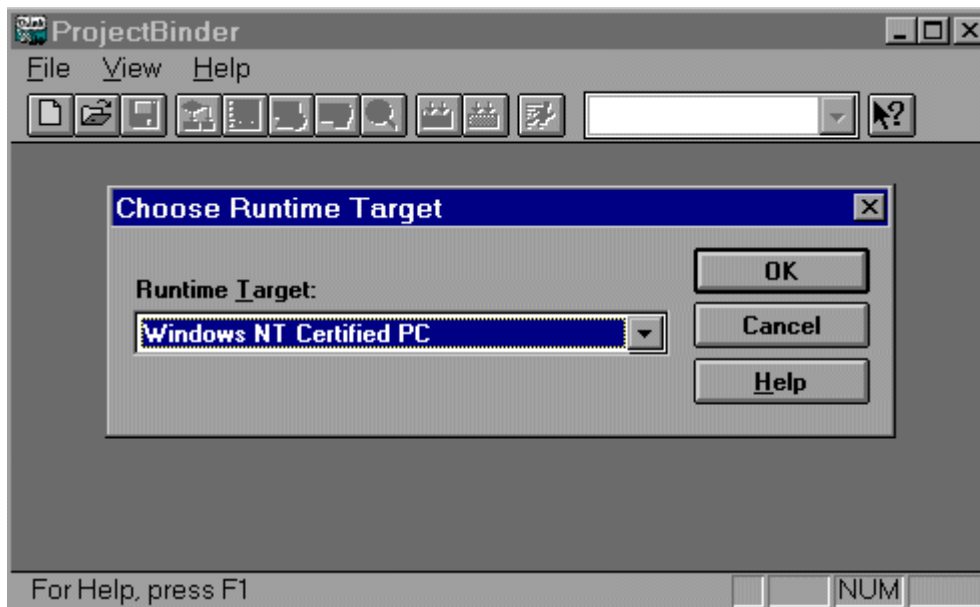
Think & Do's Live! as an OPC Client

About Think & Do Live!

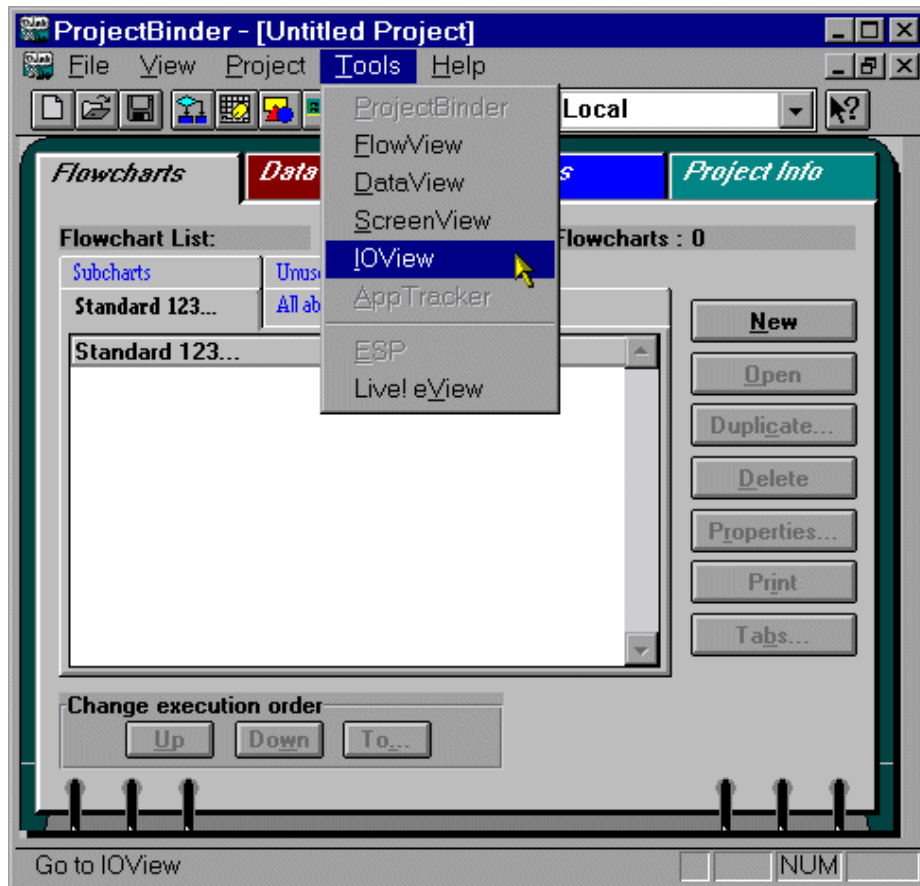
Think & Do uses a graphical flow chart approach to creating connectivity and data acquisition and connectivity. This allows the user to clearly visualize the flow of information from Server to the data display object. The following steps will show you how to create an OPC connection to the KepserverEx from Think & Do Live!.

Connect to KepserverEx from Think & Do's Live!

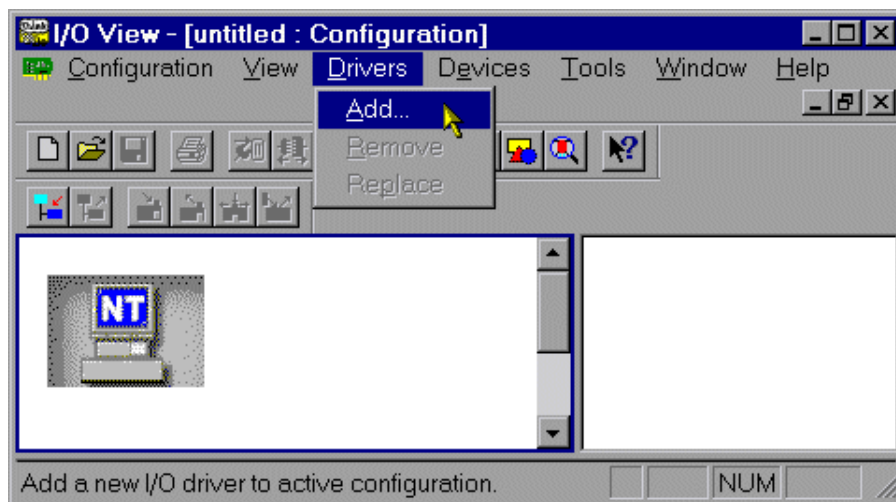
1. Start up the KepserverEx "Simdemo.opf" project.
2. Open your copy of Think & Do's ProjectBinder to start a new project.
3. Choose File|New from the main menu.



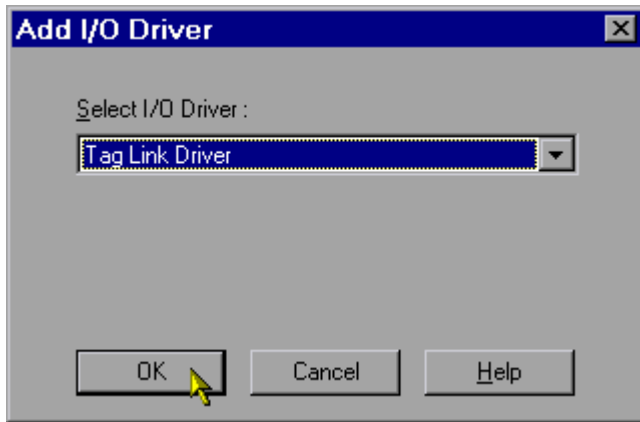
4. Select the target Operating System in which your project will be running. In this example we selected Windows NT Certified PC as the runtime target. Click OK when finished.
5. The ProjectBinder should now appear in the window. The next step is to choose Tools|OView from the main menu.



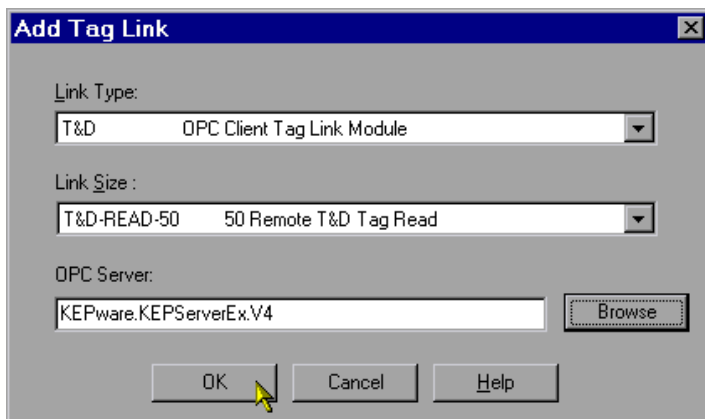
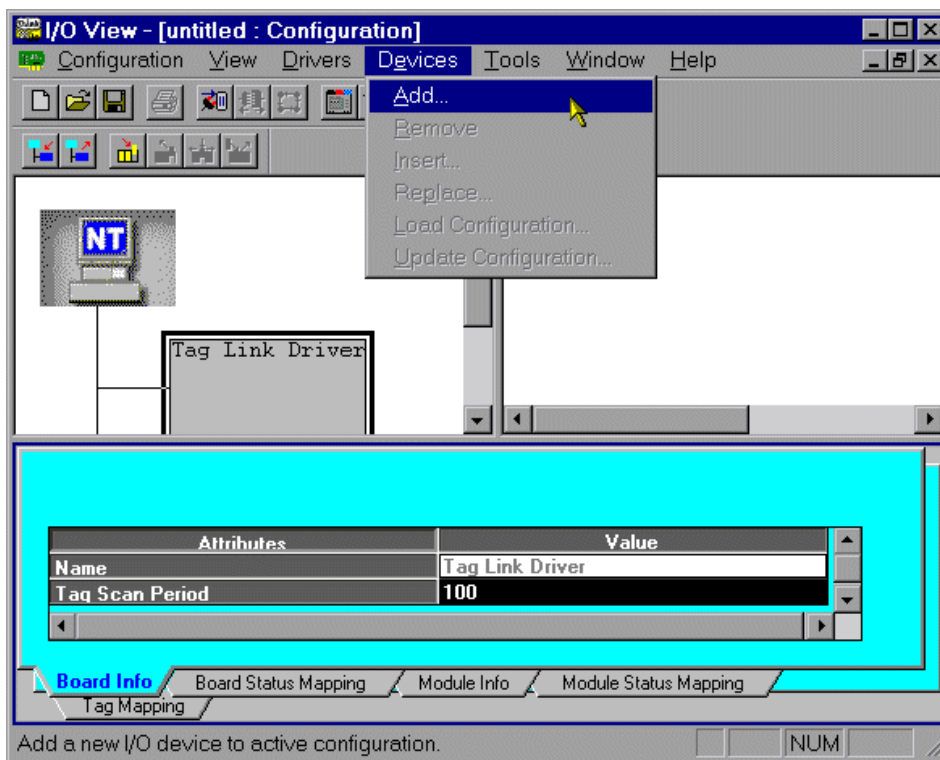
6. In the I/O View window, select Drivers|Add from the main menu.



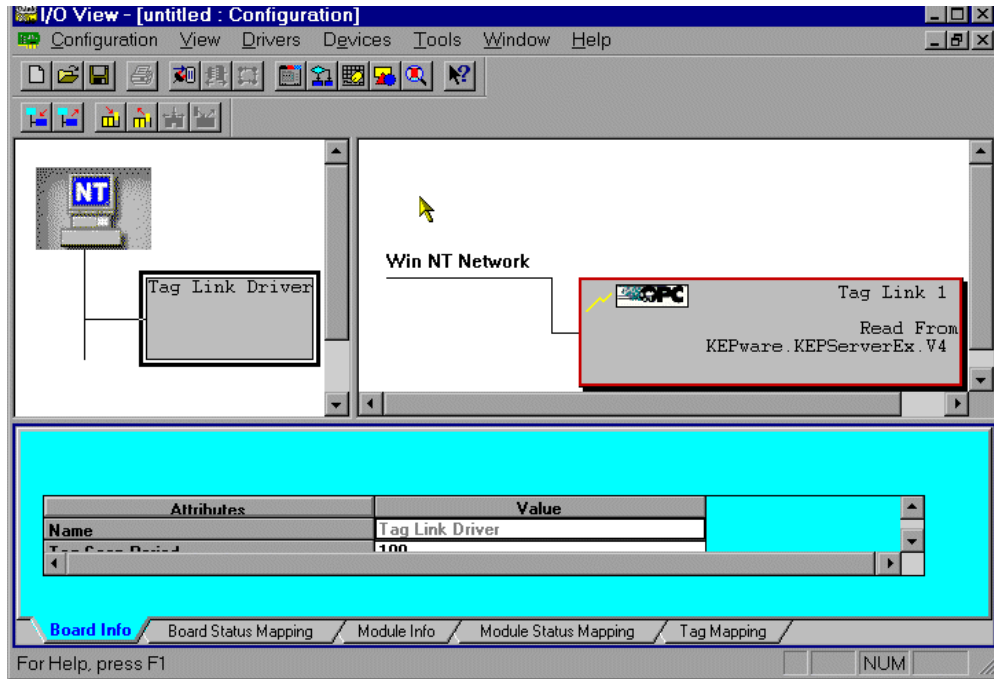
7. From the drop down menu of the Add I/O Driver dialogue, select “Tag Link Driver” and click OK.



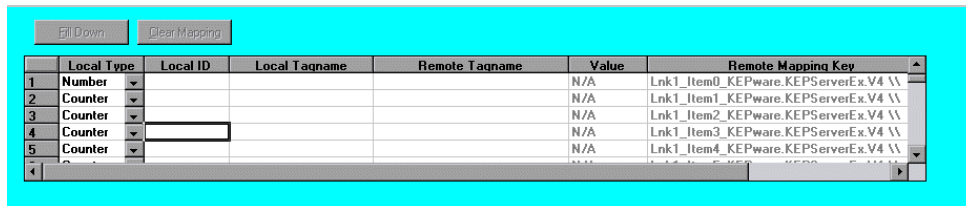
8. Now select Devices|Add from the main menu.



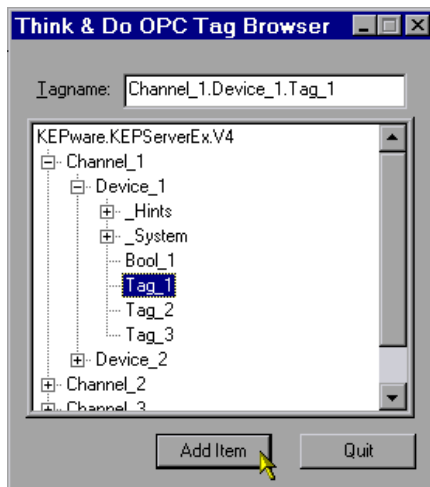
9. In the Add Tag Link dialog, choose the OPC Client Tag Link Module under Link Type.
10. Next, Browse for KepserversEx. You should select and return KEPware.KEPServerEx.v4, then click OK.



11. Now, at the bottom of the I/O View window, choose the Tag Mapping tab.



12. Right click on the first Remote Tagname field to browse for a tag in KepserversEx.

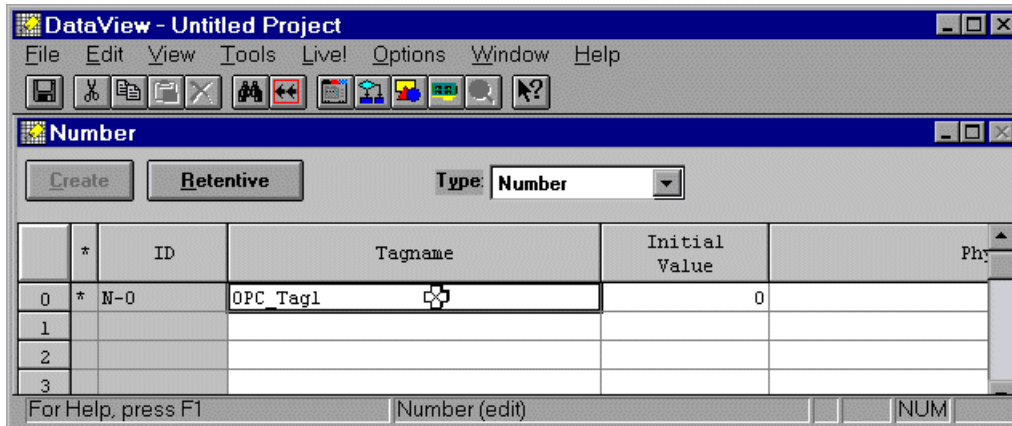


- Browse KepserverEx for the appropriate predefined tag item. When you have found it click Add Item, and then click Quit.

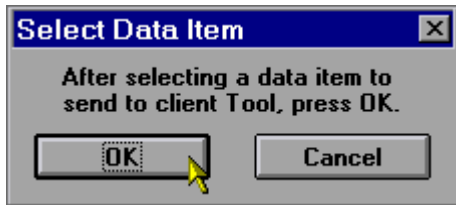
	Local Type	Local ID	Local Tagname	Remote Tagname	Value	Remote Mapping Key
1	Number			Channel_1.Device_1.Tag_1	N/A	Lnk1_Item0 KEPware.KEPSeverEx.V4
2	Counter				N/A	Lnk1_Item1 KEPware.KEPSeverEx.V4
3	Counter				N/A	Lnk1_Item2 KEPware.KEPSeverEx.V4
4	Counter				N/A	Lnk1_Item3 KEPware.KEPSeverEx.V4

- In the same row that you added the Remote Tagname, under Local Type, browse a data type most appropriate for the item. In this example “Number” is used. Next, double click on Local Tagname to enter a local name for the remote tag.

For performance purposes, make sure tag update rates are set as low as possible in the I/O View.



- In the DataView window, choose the correct data Type, then enter the desired name for the local tag. In this example the Tagname is “OPC_Tag1”. Click OK on the Select Data Item dialogue.



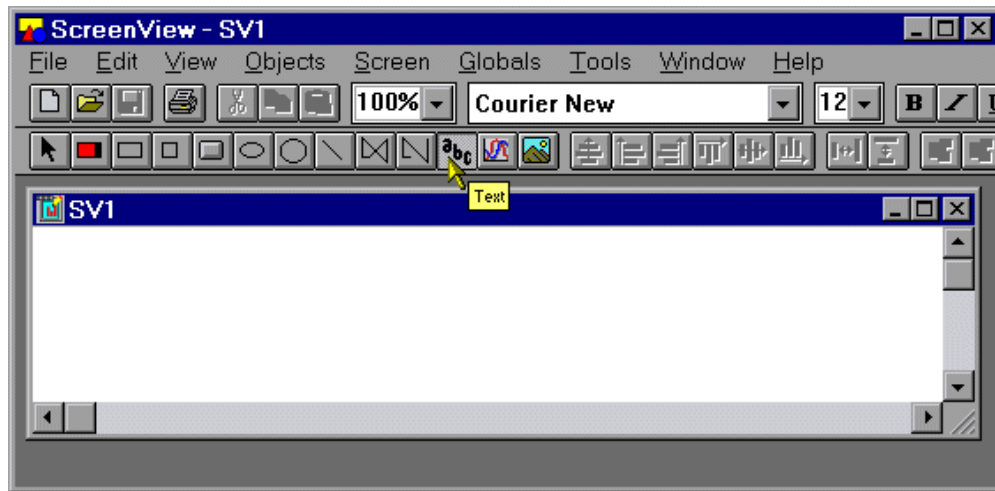
	Local Type	Local ID	Local Tagname	Remote Tagname	Value	Remote Mapping Key
1	Number	N-0	OPC_Tag1	Channel_1.Device_1.Tag_1	N/A	Lnk1_Item0 KEPware.KEPSeverEx.V4
2	Counter				N/A	Lnk1_Item1 KEPware.KEPSeverEx.V4
3	Counter				N/A	Lnk1_Item2 KEPware.KEPSeverEx.V4

Configure Tag Link to server

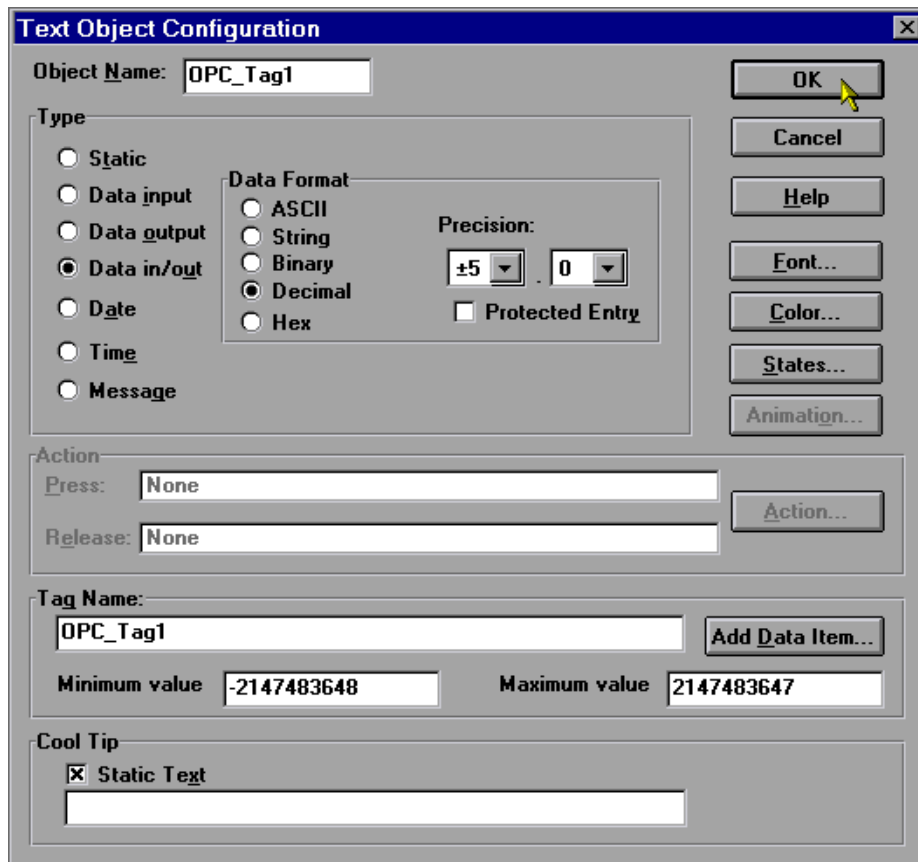
- Now that a tag has been linked, a connection can be made from the Tag Link Driver to the KepserverEx Simdemo project. Select Configuration from the I/O View main menu. In the drop down , select Connect and then Disconnect.

Create a Display Window


- The next step is to create a display object in ScreenView. Click Tools on the menu bar and select ScreenView in the drop down menu.

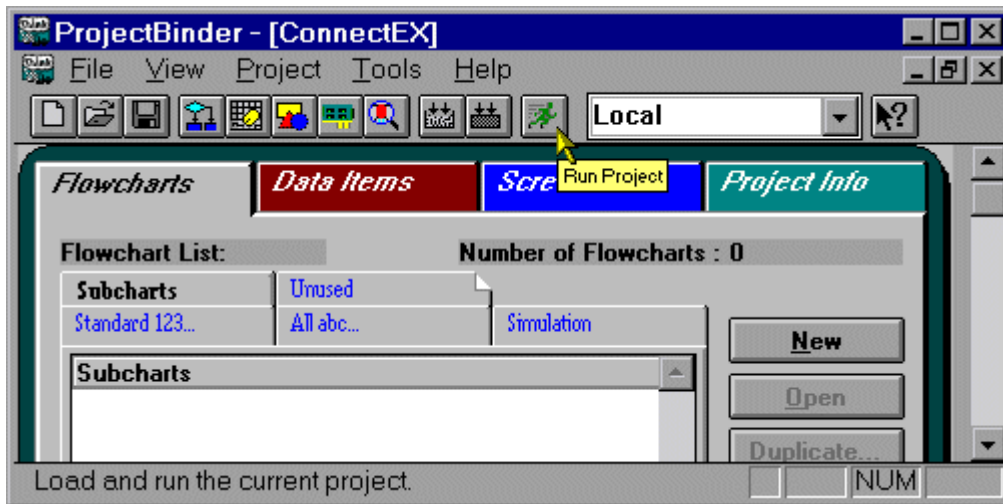


2. In the ScreenView window select **F**ile|**N**ew from the main menu. A view screen will be created.
3. Next, choose the text button **abc** from the tool bar, place the cursor on the view screen and press the left mouse button. This should place a text object on the screen.
4. Double click on the object and select **P**roperties from the drop down menu.

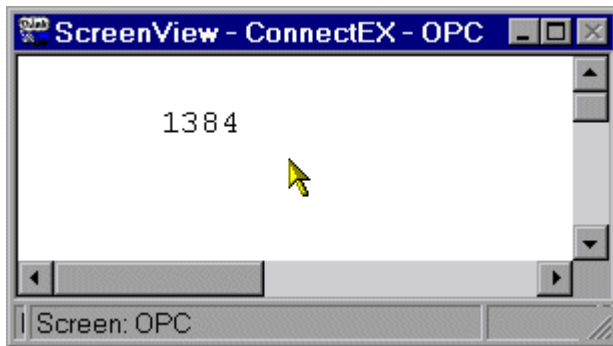


5. Fill in the preferred Object Name, Click the Data in/out radio button.
6. Enter a local tag name in the Tag Name field by clicking on the Add Data Item button and browsing the local tags that have been added. We selected "OPC_Tag1".

7. Save this view screen by selecting **File|Save** on the main menu bar of ScreenView.
8. On the main menu bar of the ScreenView, select **Tools|ProjectBinder**.
9. Now select the Screens tab on the binder. Click on the **Start Up** button to allow the newly created view screen to load when the project is executed.
10. Select **File|Save** from the ProjectBinder main menu to save the project.
11. Next, click the Build button  to build a runtime version of the project.



12. Press the Run button  on the tool bar to run the project.



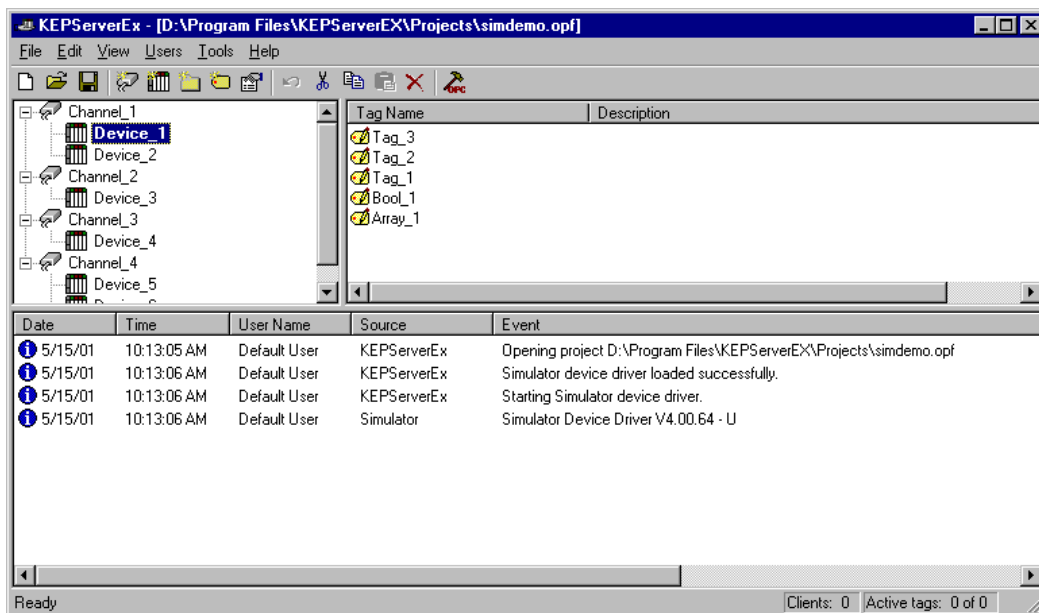
Wonderware's InTouch® as a FastDDE® or SuiteLink® Client

Connect to KepserverEx with FastDDE® or SuiteLink®

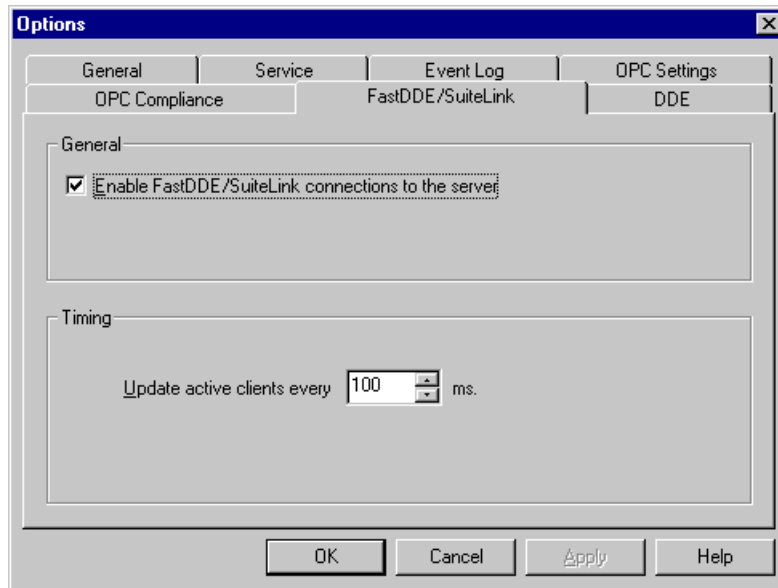
Wonderware provides several ways to connect to third party servers like KepserverEx. At the core of all Wonderware connectivity is FastDDE and SuiteLink. FastDDE and SuiteLink allow Wonderware applications such as InTouch to receive data from servers like KepserverEx. The following steps will show you how to create FastDDE and SuiteLink connections to KepserverEx. For our example we used InTouch V7.1 and KepserverEx V4.40.154.

Verify FastDDE/SuiteLink Support in the Server

1. Start up the KepserverEx "Simdemo.opf" project.



2. It is important to make certain that the server properly detected that you have Wonderware installed and that it enabled FastDDE and SuiteLink support. You can do this by selecting Tools|Options... from the server main menu.
3. If Wonderware was properly initialized on your PC and the server detects it, you should see the FastDDE/SuiteLink tab in the Options dialog box. If the tab exists then click on it.



4. Verify that Enable FastDDE/SuiteLink connections to the server is checked. The client update rate is set to 100msec by default and can be changed at any time.
5. Click OK to close the Options dialog.

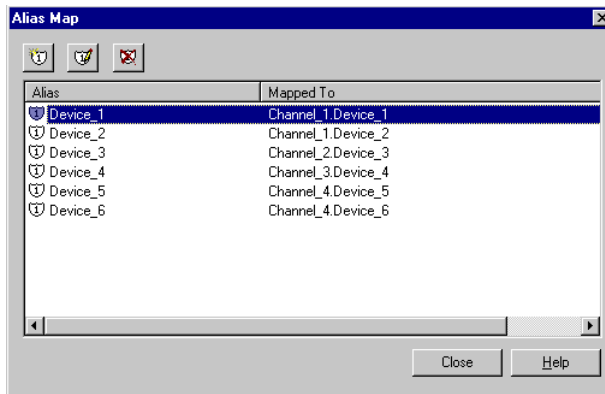
Add an Alias (DDE Topic)

You cannot use dynamic tags with tag groups. See the KepserverEx help file for an explanation of Dynamic and Static tags.

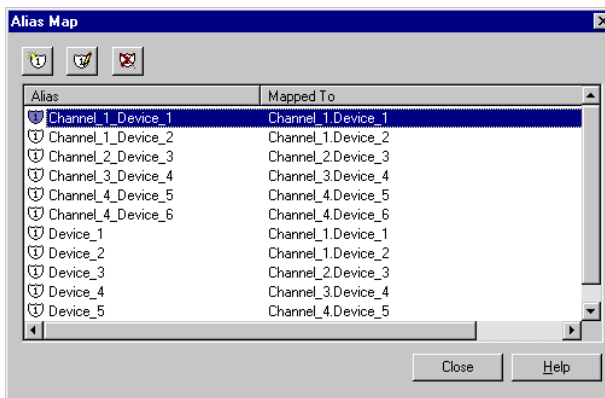
DDE connections consist of three components, an Application name, a Topic, and an Item. When connecting to KepserverEx via Fastdde or Suitelink the Application name will always be "Servermain". This name is automatically defined in the server and cannot be changed for Fastdde/Suitelink connections. There are two ways to reference the Topic component. The first way is via a Topic reference of "_ddedata". If you would like more info on this method please refer to the "How Do I..." section of the KepserverEx main help file. The preferred method to reference the DDE Topic is using the Alias Map feature of the server. The Alias Map is a special feature that simplifies the use of server data in DDE applications. Additionally, the Alias Map provides a mechanism for backwards compatibility with legacy Kepserver V3 applications. In this example we will merely explain how to create or use an Alias, if you would like more information please refer to the KepserverEx main help file.

If you are upgrading from a legacy Kepserver V3 project, simply select a file type of .kdp from the server File|Open menu option and then save your project into the new .opf format. When you do this, Alias references will be created automatically. To view these alias references select Edit|Alias Map... from the server main menu.

In the two examples below you will see aliases made to ensure backwards compatibility and you will see the aliases that the server automatically creates for a project. You will notice that in KepserverEx the automatic alias is a combination of the channel and device name. The server will modify these automatic aliases whenever you make changes to the Channel or Device name. For a detailed explanation on creating aliases see Designing a KepserverEx Project in the KepserverEx help file.



Alias map of imported legacy Kepserver V3 project file. Notice that the Alias name is the same as the device name.



Example of imported project after it has been re-opened. Notice that the server automatically creates an Alias composed of the channel and device name.

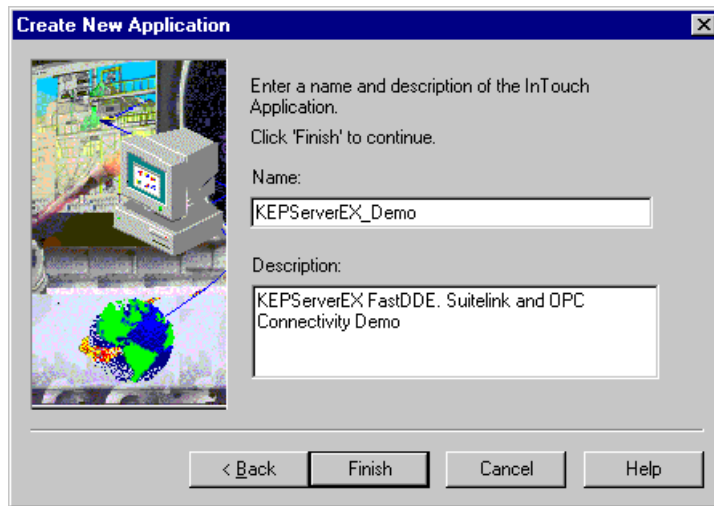
6. In this example we are using the sample KepserverEx project called “Simdemo.opf”, so you will not need to create an alias.

If you wish to learn how to create aliases for your device and group folders, then go to “Designing a KepserverEx Project” in the Server Help file.

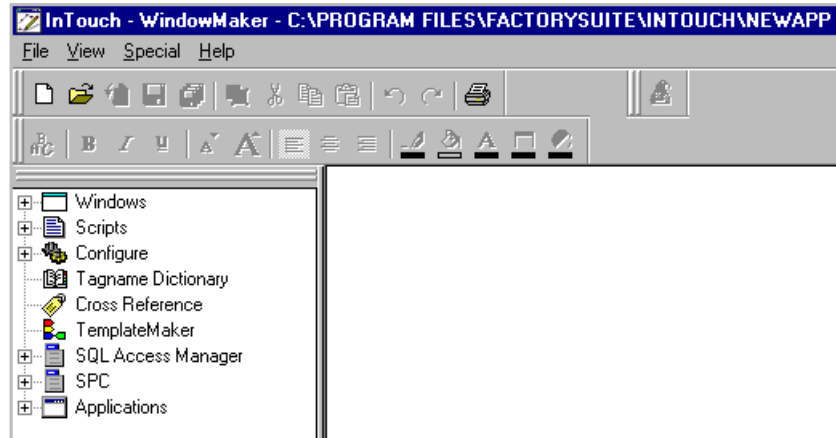
Create an InTouch® project

Now we need to create an InTouch Application Project. The InTouch project will act as the DDE client to the KepserverEx. If you are using the Simdemo.opf server project it will return simulated data to InTouch. If you are using one of the more than 50 actual device drivers in the server, the server will use the device driver to poll the PLC or device and return the data to the DDE client.

1. Start the InTouch Application Manager and then create a new project by selecting File|New... from the main menu.

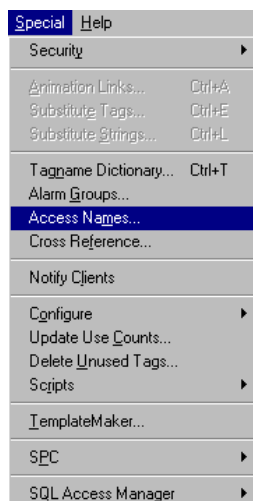


2. In the Create New Application dialog box enter a Name for your new InTouch project. You may also enter a Description if you wish. Once your are done, click Finish to continue and start InTouch WindowMaker.



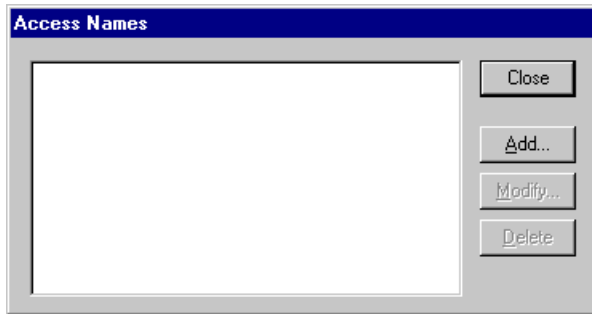
Add Access Name Links to your Project

Now that we have a new application we need to link it to Kepware's server.

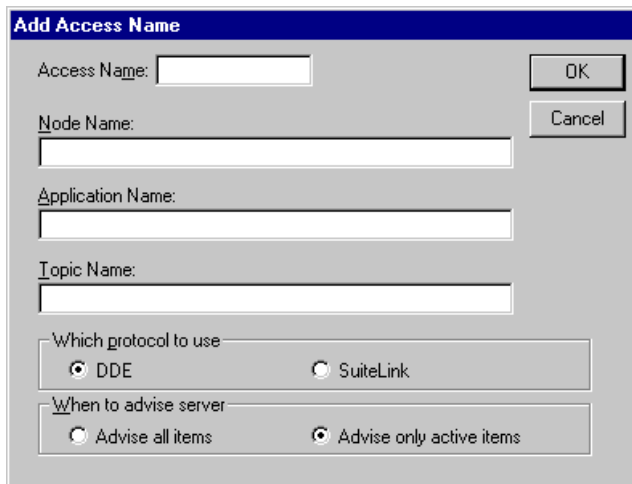


1. For multiple server devices, remember to create a separate Access Name for each Device in your KepserverEx project. In this example we will create a single Access Name, which will be linked to the first Device in the KepserverEx project (Device_1). To do this select Special|Access Names... in the WindowMaker main menu.

See the InTouch Users Guide and Reference Manuals for details on creating and using Access Names in applications.



2. Click on Add to add a new Access Name.



3. Enter a unique Access Name, we made ours “KEPServerEx_FS”. The Application Name will always be “Servermain”. The Topic Name will be the alias that we created for the first device “Device_1”. Leave the Node Name blank.
4. This will be a DDE connection so we will select that protocol option.
5. We will select Advise only active items. See the InTouch Users manual for an explanation of the two Advise options.

If we were connecting to a remote PC then we would enter its Name in Node Name.

Add Access Name

Access Name:

Node Name:

Application Name:

Topic Name:

Which protocol to use

DDE SuiteLink

When to advise server

Advise all items Advise only active items

6. Click OK to add the access name. For our example we will also add an access name for a SuiteLink connection.
7. For SuiteLink connections to other PC's you should enter a Node Name which will be the name of the PC that is running the server project you wish to connect to, otherwise leave it blank.

Add Access Name

Access Name:

Node Name:

Application Name:

Topic Name:

Which protocol to use

DDE SuiteLink

When to advise server

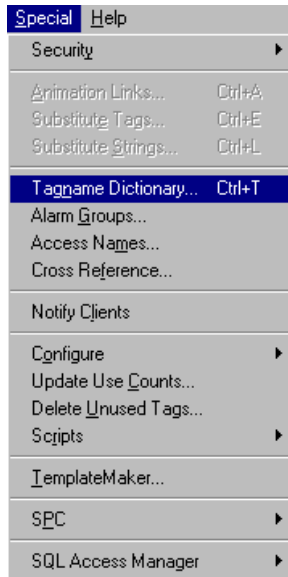
Advise all items Advise only active items

8. Once you have added all of the Access Names click Close to return to your project.

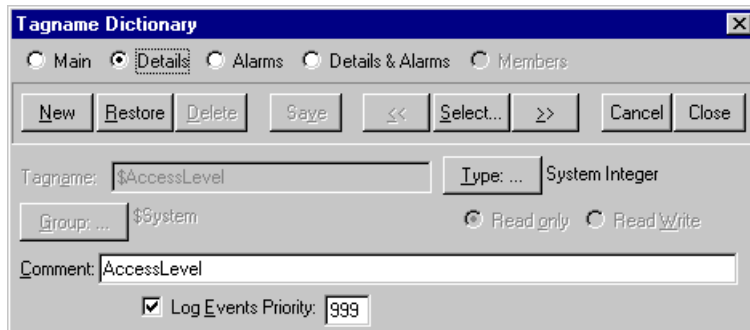
Access Names

Add Tags to the Tagname Dictionary

Now that we have created the links to the server we can add tags to the Tagname Dictionary.



1. Select Special|Tagname Dictionary... from the WindowMaker main menu.



2. In the Tagname Dictionary click on New to add a new tag.
3. Because we started adding tags while we were on an internal tag the new tag form opens with a tag type of Memory Integer. We will be using an Integer as well so click on Type ... and select I/O Integer from the list.
4. Next click on Access Name ... to select where the data for this tag will be coming from. We chose KEPServerEx_FS for this tag.
5. Enter a Tagname for the tag, we entered KEPServerEx_INT1 for this tag.
6. For the Item field at the bottom left of the tag property window, we used a reference of Tag_1. This tag had been pre-defined in the server Simdemo project and is called a User Defined tag. In the server the data type is defined as Short which is a signed 16-bit integer.
7. Click on Save to accept the new tag. At this point you can go to the next section and display data if you wish.

See the InTouch Users Guide and Reference Manuals for details on creating and using Tags in applications.

The screenshot shows the 'Tagname Dictionary' dialog box with the following details:

- Buttons: New, Restore, Delete, Save, <<, Select..., >>, Cancel, Close
- Tagname: KEPServerEX_INT1
- Type: I/O Integer
- Group: \$System
- Access: Read only (selected), Read Write
- Comment: AccessLevel
- Log Data: Log Events: Retentive Value: Retentive Parameters:
- Initial Value: 0
- Min EU: -32768
- Max EU: 32767
- Deadband: 0
- Min Raw: -32768
- Max Raw: 32767
- Eng Units: (empty)
- Conversion: Linear (selected), Square Root
- Access Name: KEPServerEX_FS
- Item: Tag_1
- Use Tagname as Item Name: Log Deadband: 0

- Next we are going to add a second tag. This is an example of a dynamic tag.
- The new tag will have a Tagname of KEPServerEx_INT4_Dynamic and the item will be R200, which is a valid address within the server. We assigned this one to our SuiteLink node, Access Name KEPServerEx_SL.

The screenshot shows the 'Tagname Dictionary' dialog box with the following details:

- Buttons: New, Restore, Delete, Save, <<, Select..., >>, Cancel, Close
- Tagname: KEPServerEX_INT4_Dynamic
- Type: I/O Integer
- Group: \$System
- Access: Read only, Read Write (selected)
- Comment: AccessLevel
- Log Data: Log Events: Retentive Value: Retentive Parameters:
- Initial Value: 0
- Min EU: 0
- Max EU: 65535
- Deadband: 0
- Min Raw: 0
- Max Raw: 65535
- Eng Units: (empty)
- Conversion: Linear (selected), Square Root
- Access Name: KEPServerEX_SL
- Item: R200
- Use Tagname as Item Name: Log Deadband: 0

If you would like more info on Dynamic and User-Defined tags, please refer to the "Designing a KepserverEx Project (Step5)" of the server main help file.

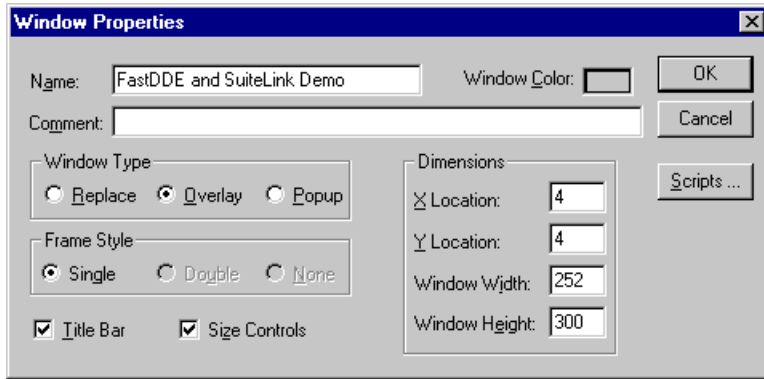
- Once you have added your tags click Close to close the Tagname Dictionary.

Create a Window to Show the Device Values

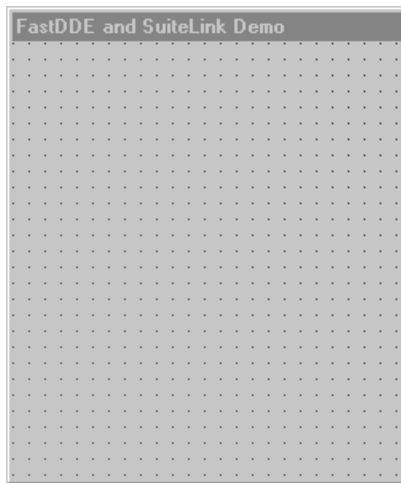
Now we need to display our data in our project.

- In the WindowMaker main menu click on New|Window to add a new window to the project.

See the InTouch Users Guide and Reference Manuals for details on creating windows in applications.



2. Give your window a name that will be displayed on the Title Bar, if checked, and click OK to accept the properties.



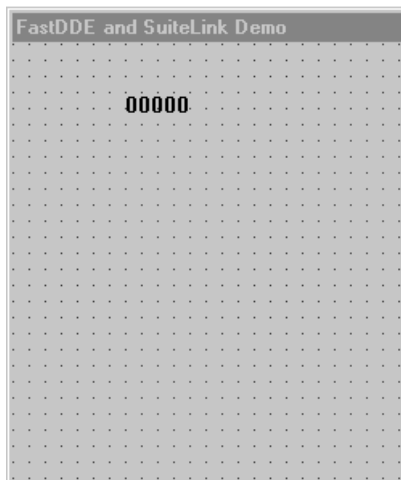
Add and Animate a Text Object

Now we need to add a text object to show the value of one of our tags.

3. On the WindowMaker Toolbar select the Text icon.



4. Click on the object window and create a data object by entering the maximum number of "0" digits that the item data will display.



- Double click on the new text object to add an animation link to it.
- For the purposes of our demo we are selecting Analog User Input under Touch Links.

The screenshot shows a configuration dialog for 'Touch Links'. It is divided into several sections:

- Touch Links:**
 - User Inputs:** Discrete (unchecked), **Analog (checked)**, String (unchecked).
 - Sliders:** Vertical (unchecked), Horizontal (unchecked).
 - Touch Pushbuttons:** Discrete Value (unchecked), Action (unchecked), Show Window (unchecked), Hide Window (unchecked).
- Line Color:** Discrete (unchecked), Analog (unchecked), Discrete Alarm (unchecked), Analog Alarm (unchecked).
- Fill Color:** Discrete (unchecked), Analog (unchecked), Discrete Alarm (unchecked), Analog Alarm (unchecked).
- Text Color:** Discrete (unchecked), Analog (unchecked), Discrete Alarm (unchecked), Analog Alarm (unchecked).
- Object Size:** Height (unchecked), Width (unchecked).
- Location:** Vertical (unchecked), Horizontal (unchecked).
- Percent Fill:** Vertical (unchecked), Horizontal (unchecked).
- Miscellaneous:** Visibility (unchecked), Blink (unchecked), Orientation (unchecked), Disable (unchecked).
- Value Display:** Discrete (unchecked), Analog (unchecked), String (unchecked).

- Clicking the Analog button will also check the use box and open the animation properties dialog for it.

The screenshot shows the 'Input -> Analog Tagname' dialog box with the following settings:

- Object type:** Text
- Tagname:** (empty text field)
- Key equivalent:** Ctl (unchecked), Shift (unchecked), **Key...** (selected), None (selected)
- Msg to User:** (empty text field)
- Keypad?:** Yes (unchecked), **No** (checked)
- Min Value:** -32768
- Max Value:** 32767
- Input Only:** (unchecked)

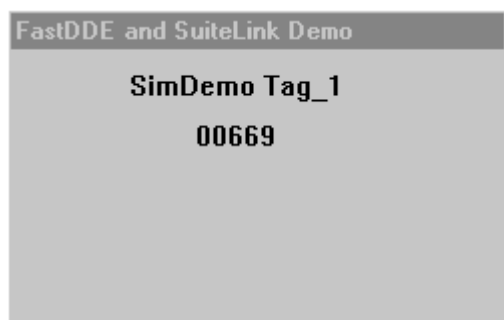
- Enter the Tagname to display, we chose to use Tag KEPServerEx_INT1.

This screenshot is identical to the previous one, but with the 'Tagname' field containing the text 'KEPServerEX_INT1'.

- Click OK to exit.

Check the Connection to the Server and View Data

1. To view active data, click on runtime in the upper right corner of WindowMaker. You should now see data ramping in the WindowViewer as well as an active client and items on the Connection Status Bar at the lower right edge of the KepserverEx project window.



2. If you do not see a ramping value, compare the configuration steps in this documentation to your project to verify that everything is configured correctly. Check the server Connection Status Bar to confirm that your InTouch application is connecting to the server, and then look in the Event window in the bottom of the server application view to check for error messages. If you have questions, please refer to the Support section of www.kepware.com for technical support assistance.

Appendix A

Cimplicity Master OPC INI files

This section explains the use of some of the sections and parameters found in the Master OPC INI file. Unless otherwise specified in the connectivity guide you will use the defaults when connecting to KepserverEx.

About the Master OPC INI File

This INI file is for use by the OPC Client devcom provided with CIMPLICITY 4.0.08 or higher, and CIMPLICITY 5.0.1.

This file must reside in the “data” subdirectory of the project directory. There is one “INI” file per OPC port configured in the project. The name reflects the port's ID. For example “MASTER_OPC_0.INI” for the first port, and “MASTER_OPC_1.INI” for the second port configured, etc.

The purpose of this file is to allow the user to manually configure the OPC Port, Devices, and Groups. Unlike the global parameters file (GLB_PARMS) where settings applied to all devices or groups, settings here apply only to the specified device or group.

The sample provided here contains all of the possible parameters. These parameters are set to their default values! If this INI file is not present, or if a particular Device or Group is not specified by a [Section Header], then the default values are used.

For typical BOOLEAN parameters, a value of 1 means TRUE/ENABLE. A value of 0 means FALSE/DISABLE.

Note: The OPC client always creates TWO GROUPS where ALL points are placed unless you configure them otherwise. These groups are named: DEFAULTPOLL and DEFAULTUNSO. You can set parameters for these groups as well as the groups you define.

Note: There are a couple of parameters that apply to the 'Port' as a whole. These are placed in the section called [PortLevel].

The INI File Sections

[PortLevel]

EightByteReals=0

UseServerTimeStamp=1

MessageTicks=50

CircularLog=1

UseDataTypePromotion=0

[DEVICE1]
StartupDelay=0
ReadDelay=0
PingInterval=5000
PingTimeout=3000
AbortShutdown=0
UseLocalReg=0
DCOMTimeoutThreshold=10000
PingBeforePoll=1
PingBeforeWrite=1
ForceOPC1Server=0
HRBothActive=0

[DEVICE2]
StartupDelay=0
ReadDelay=0
PingInterval=5000
PingTimeout=3000
AbortShutdown=0
PingBeforePoll=1
PingBeforeWrite=1

[DEFAULTPOLL]
ScanRate=10
NoAccessPath=0
DeviceReadAfterSet=1

[MYGROUP01]
ScanRate=1000
DeviceReadAfterSet=1
NoAccessPath=0

[DEFAULTUNSO]
ScanRate=10
DeviceReadAfterSet=1
NoAccessPath=0

PortLevel Section

EightByteReals – *Default = 0*

Should Cimplicity treat Real values as four byte (float) or 8 byte (double) if the OPC server sends single float values, then setting this to one (1) will have no affect on precision. If the server sends doubles, setting this to one (1) will maintain the precision as the value is passed into Cimplicity.

UseServerTimeStamp – *Default = 1*

Should CIMPLICITY use the timestamp passed from the OPC server or provide its own timestamp at the time the update is received.

MessageTicks – *Default = 50*

Controls the rate in Ticks (1 tick = 1/100 seconds) that Windows Messages are pumped by the OPC client. While the client has no use for these messages, it is required by the OS to process them. By default, it drains the message queue once every half second.

CircularLog – *Default = 1*

By default, tracing is to a circular file. If disabled, logging goes to the output file.

UseDataTypePromotion – *Default = 0*

Conversion of data types for many older servers was accomplished via a method called data type promotion. Because the VARIANT was not able to hold signed eight bit (SINT), unsigned sixteen bit (UINT), and unsigned thirty-two bit (UDINT) values, these types were 'promoted' to the next larger type that the VARIANT was capable of holding.

As of Service Pack 4 for Windows NT, the VARIANT was redefined to allow these data types, making data type promotion obsolete. While newer OPC servers are making use of the new VARIANT types many older OPC servers still expect data type promotion to occur.

Since data type promotion is no longer typical, this parameter defaults to disabled.

Device1 Section**StartupDelay** – *Default = 0 (Value is milliseconds)*

Some OPC servers take extra time to load their own point databases. Often, points added to the server by Cimplicity are declared invalid because the server was not ready to process additions.

This delay occurs after the OPC server is started, but, before any points are added by Cimplicity.

ReadDelay – *Default = 0 (Value is milliseconds)*

Some OPC servers take extra time, after a client has added points, getting initial values for those points. If a poll occurs too soon values with "Bad Quality" are provided. To avoid this, the Read Delay causes Cimplicity to hesitate after adding points but before the very first poll.

PingInterval – *Default = 5000 (Value is milliseconds)*

Cimplicity now pings the OPC server on a regular basis. The Ping Interval determines how often this ping occurs.

PingTimeout – *Default = 3000 (Value is milliseconds)*

If the method call made by the Ping (IOPCServer::GetStatus) takes too long to return, Cimplicity declares a communication error. The Ping Timeout determines how long to wait for this method call.

AbortShutdown – *Default = 0*

If this parameter is TRUE (1), Cimplicity will shut down (on project shutdown) without releasing any references to objects in the OPC server. This is strictly against the rules of OLE!!!

However, it will allow the Cimplicity project to shut down in a minimum amount of time. The OPC server will most likely not shut down at all, since it will believe that our client is still attached.

UseLocalReg – *Default = 0*

If this parameter is TRUE (1), Cimplicity will access the local registry information to find the OPC Server ID first. If the devcom fails to find the server information in local registry, it will then log warning message and search the information in the remote node. This parameter is used to resolve the multi-sessions issue.

DCOMTimeoutThreshold – *Default = 10000 (Value is milliseconds)*

After a ping fails, this is the number of milliseconds that the client will wait after detecting a lack of response from the server via a ping before it will force an abort of the connection

PingBeforePoll – *Default = 1*

When a network connection is broken between the Cimplicity and server; in a DCOM situation, the very next method call the client makes (usually a Poll [ISyncIO::Read]) can hang for upwards of 90 seconds. This causes all device communication in Cimplicity to hang until the method call times out. By pinging the OPC server immediately before the method call, Cimplicity eliminates most of these potential DCOM timeouts (since Pinging occurs on a different thread).

This does add the overhead of an extra call to the server per poll request (or write request) but with a DCOM timeout of 90 seconds this may be well worth it.

PingBeforeWrite – *Default = 1*

Same as above but is done after the poll.

ForceOPC1Server – *Default = 0*

Cimplicity by default establish a connection using OPC 2.0 methods if the Server will support it. (It will use 1.0 if the server does not.) To force the use of 1.0 methods, enable this variable.

HRBothActive – *Default = 0*

In Host Redundant environments, by default, the acting master will advise for the points that are defined to be unsolicited/unsolicited on change/poll once/poll once on change (i.e. the server polls the data and notifies the client only of those items that have changed). Some servers may be slow in collecting the data resulting in transition times for the unsolicited data. All data will be advised and refreshed (unless refresh is disabled) on host transition. Enabling this variable will result in both servers advising the client. The result will be more traffic with a lower transition time. It is possible that data that changes during the transition between hosts may be lost for points with an unsolicited or unsolicited on change update criteria.

DefaultPoll Section

ScanRate – *Default = 1000*

This is the Scan Rate passed to the OPC server for this group. It tells the OPC server how often to update the items in this group with fresh device data. It also determines how often to fire DataChange events to Cimplicity. The unit of measure is in milliseconds and the default value is 1 second.

If you require faster value updates, you can decrease this value to a value as low as zero. Zero tells the OPC server to update values 'as fast as possible'. For servers able to collect rapidly changing data very rapidly, in a DCOM environment, it is possible to send the data faster than can be supported by the hardware/operating system on a sustained basis.

Caution is urged before setting this to a value of zero.

NOTE: As the scan rate decreases CPU usage may rise (somewhat significantly) and a corresponding increase in network traffic may be observed in a DCOM environment.

The [DEFAULTPOLL] section controls scanning for "On Change" events and issues Synchronous reads and writes to the server. The [DEFAULTUNSO] section controls scanning for unsolicited events and issues Asynchronous reads and writes to the server. The latter type is much more efficient.

NoAccessPath – *Default = 0*

Some OPC servers do not use the Access Path when adding points. Instead, the Access Path must be pre-pended in front of the Item ID. Enabling this feature allows these OPC servers to have the Item IDs include the access path, separated by a period.

DeviceReadAfterSet – *Default = 1*

For polled operations, Cimplicity will retrieve required data from the Servers cache. For other operations, (i.e. poll after set) by default, it will request that the OPC Server read the data from the device. This can significantly affect performance where the functionality is fully supported. To force all reads from the cache, disable this variable. It is enabled by default.

Glossary of Terms

Address

1) A character string that uniquely identifies a memory location or the physical location of an input or output circuit. 2) A character string that uniquely identifies a node (device) on a network

ANSI

American National Standards Institute. An organization that develops and publishes voluntary industry standards in the United States.

ASCII

American Standards Code for Information Interchange. It is a 7-bit code with an optional parity bit used to represent alphanumeric, punctuation marks, and control-code characters.

Asynchronous Transmission

A method of serial transmission where data may be transmitted at unequal time intervals. It requires that each data packet contain start/stop timing elements (extra bits) so the receiver can properly detect the start and end of each packet.

Asynchronous Writes

With Async writes the client sends the write to the server and continues processing. The Server/driver process the write and upon confirmation from the device that the write was successful it notifies the client via the Callback() function.

Bandwidth

The bit per second transmission capability of a channel.

Bar Code

A series of horizontal stripes or bars of varying width which represent a string of characters that can be read by a bar code reader (scanner).

Buffer

1) In software terms, a register or group of registers used for temporary storage of data, to compensate for transmission rate differences between the transmitter and receiving device. 2) In hardware terms, an isolating circuit used to avoid the reaction of one circuit with another.

Bus

A single path or multiple parallel paths for power or data signals to which several devices may be connected at the same time. A bus may have several sources of supply and/or several sources of demand.

Bus Network

A network where all communications travel along a common path. Each device in the network handles its own communications control.

Communications Protocol

The rules governing the exchange of information between devices on a data link.

CRC (Cycling Redundancy Check)

An error detection schema in which the block check character is the remainder after dividing all the serialized bits in a transmission block by a predetermined binary number.

Data Area

An area of PLC memory allocated to the storage of data not programs.

DCS (Data Communication System)

An electronic system that transmits data over communication lines from one location to another.

DCE (Data Communication Equipment)

1) Equipment that provides the functions required to establish, maintain, or terminate a connection. 2) The signal conversion and coding required for communication between data terminal equipment and data circuits. DCE may or may not be an integral part of a computer.

Data Transfer Protocol

See DDE (Dynamic Data Exchange) or OPC (OLE for Process Control)

Data Type

A type which describes the format and value range of data in a memory location. I.E. Boolean, Word, Long, etc.

Device

A piece of hardware used to control a process or collect data. These can range from PLC's to smart switches. All of them have the ability to communicate in some form

Diagnostic Program

A user program designed to help isolate hardware malfunctions in the programmable controller and application equipment.

Diagnostic

Pertains to the detection and isolation of an error or malfunction.

Dip Switches

1) Switches for opening and closing leads between devices. 2) Switches used to set board lever parameters in smart cards and other configurable devices.

DCOM (Distributed COM)

Based on Microsoft's COM technology, OPC servers can share data with remote client applications using DCOM (Distributed COM). DCOM allows you to use a single OPC server to provide data to client applications running both locally and on remote machines.

DH-485

Data Highway 485 link. An Allen-Bradley token-passing base band link for a local area network based on the RS-485 standard.

DLL (Dynamic Link Library)

A program file that, although it cannot be run as a stand alone executable, can be utilized by one or more applications or programs as a common service. DLL files have a *.DLL extension. DLL's comprise a number of stand-alone functions

DTE

Data-Terminal Equipment. Equipment that is attached to a network to send or receive data, or both.

Driver

The software which controls the communication to a device. Commonly these are DLL files.

Dynamic Data Exchange

DDE was implemented in 1986 to exchange data between applications. In the automation industry there are four types of DDE. 1)CF_Text 2)XL_Table 3)Advanced DDE 4)FastDDE

Dynamic Tags

Dynamic tags allow you to define tags once in the client application. Instead of providing the server with a tag name as the OPC/DDE item, you would provide the device address (and optionally a data type.) The server will create a tag for that location and start scanning for data automatically.

Fieldbus

Fieldbus is a generic term used to describe a common communications protocol for control systems and/or field instruments. Although some standard forms have been agreed for instruments, the DCS industry as a whole has so far no agreed fieldbus. Often, the Foundation Fieldbus is referred to simply as the Fieldbus.

Full Duplex

A mode of operation for a point-to-point link with two physical circuits, in which messages or transmission blocks can be sent in both directions at the same time.

Half Duplex

A mode of operation for a point-to-point or multi-point base band link with two physical circuits, in which messages or transmission blocks can be sent in one direction or the other, but not both at the same time.

Hand Shaking

Exchange of predetermined signals between two devices establishing a connection.

Interactive User

Specifies that the application will run using the security context of the user who is currently logged onto the computer (the interactive user). The application runs as this user in order to be authenticated in the domain. The interactive user may be the same as the launching user. If this is selected and a user is not logged on, the application will not start.

Launching User

Specifies that the application will run using the security context of the user who started the application (the launching user). The application runs as this user in order to be authenticated in the domain. The launching user may be the same as the interactive user.

Line Turn Around Time

The time between the end of a frame transmitted by one station to the start of the next frame transmitted by another station, as referenced by the signals on the bus. For example, from the last bit of an end flag to the first bit of a start flag.

Master (Communications)

A form of communication in which one station has master status, which allows it to initiate communication with any other station on the link; while each of the other stations has slave status such that it can only send replies to commands from the master, and send command messages to the master in response to being polled by the master.

OPC (OLE for Process Control)

In 1994 a group of vendors representing a broad spectrum of disciplines in industrial segment formed what is now known as the OPC Foundation. The OPC Foundation put forth the goal of developing a single client/server specification that would allow any vendor to develop software and applications that could share data in a fast, robust fashion, and do it in a way that would eliminate the proprietary schemes that forced these same vendors to duplicate development efforts. The OPC Foundation developed the first specification called Data Access Specification 1.0a which was released in early 1996. Using this specification, vendors were able to quickly develop client server software.

Parity Check

A check of the sum of the parity bit plus all of the data bits in each word to determine whether the sum is even or odd. A failure of the parity check indicates that a data bit has been corrupted.

Peer-to-Peer

A form of communication in which messages are exchanged between entities having equal access to all the resources.

Point-to-Point

A link between two and only two pieces of equipment.

Producer/Consumer Model

A communication model in which data is identified by its content rather than by its source or destination. Devices that need the data (consumers) recognize the data they need and consume it. Therefore, data only needs to be sent out on the network in a single message no matter how large the number of nodes to which it needs to go.

Protocol

A set of conventions governing the format and timing of data between communication devices.

RS-232

Defines three types of connections: electrical, functional, and mechanical. The RS-232 interface is ideal for the data-transmission range of 0–20 kbps/50 ft. (15.2 m). It employs unbalanced signaling and is usually used with DB25 connectors to interconnect DTEs (computers, controllers, etc.) and DCEs (modems, converters, etc.). Serial data exits through an RS-232 port via the Transmit Data (TD) lead and arrives at the destination device's RS-232 port through its Receive Data (RD) lead. RS-232 is compatible with these standards: ITU V.24, V.28; ISO IS2110.

RS-422

Defines a balanced interface with no accompanying physical connector. Manufacturers who adhere to this standard use many different connectors, including screw terminals, DB9, DB25 with nonstandard pinning, DB25 following RS-530, and DB37 following RS-449. RS-422 is commonly used in point-to-point communications conducted with a dual-state driver.

RS-485

Resembles RS-422. It may be used in multipoint applications where one computer controls many different devices. Up to 64 devices may be interconnected with RS-485.

Serial Transmission

The most common transmission mode in which information bits are sent sequentially on a single data channel.

Service

An application running as a Service on NT or Win2K operating systems allows remote clients to connect to and acquire data from the server regardless of who is logged onto the PC. Services run as soon as the operating system is started and run under the system user space.

Short Haul Modem

A signal converter which conditions a digital signal to ensure reliable transmission over DC continuous private line metallic circuits without interfering with adjacent pairs in the same cable.

Slave (communications)

On a communication link, a station that cannot initiate communication. Only a master can initiate communication.

Static Tags

A set of tags defined in the server project. The names you assign to each tag then become the item of each OPC/DDE link between the client and the server. The primary benefit to this method is that all user-defined tags are available for browsing within OPC clients. Additionally, user defined tags also support scaling.

Straight Through Pinning

RS-232 and RS-422 configuration that matches DTE to DCE, pin for pin.

Synchronous Transmission

Transmission in which the sending and receiving stations operate continuously at the same frequency and are held in a desired phase relationship.

Synchronous Writes

With Sync writes the client sends the write and then waits for the response back from the device before continuing processing. In most cases there will be no visible difference between the two. However, if the device is connected via a modem at 1200baud you would probably see a delay in processing.

Timeout (Connection)

Used primarily by Ethernet based drivers. The connection timeout allows the time required to establish a socket connection to a remote device to be adjusted. In many cases the connection time to a device can take longer than normal communications request to that same device. The valid range is 1 to 30 seconds.

Timeout (Request)

Used by all drivers to determine how long the driver will wait for a response from the target device.

User Defined Tags

See "Static Tags"