Introduction

The Purpose of this Supplementary Manual

This KEP***Direct*** Setup manual provides enough of the basics to install the OPC* server* software without having to read the complete online help documentation that covers all the details of KEP***Direct***. Exercise caution: This is not intended to replace reading the online help documentation. This is intended only as a supplement. This is only a quick start guide.

Who Can and Should Use KEP*Direct*?

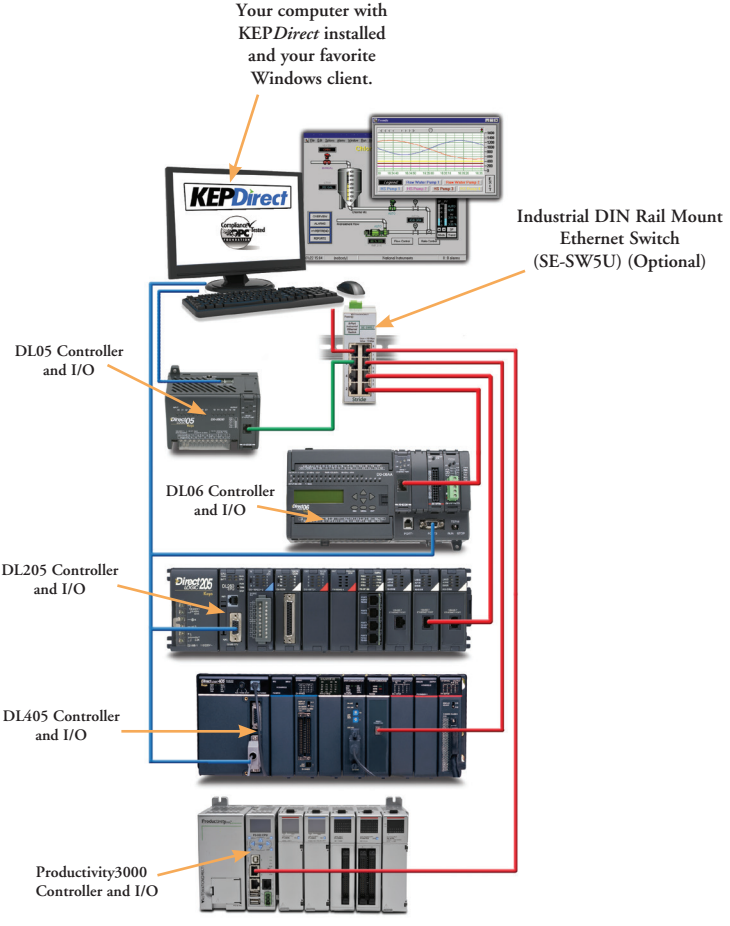
KEP***Direct*** is a software based server that is designed for accurate communication, quick setup and unmatched interoperability between client applications, industrial devices and systems. This server provides a wide range of plug-in device drivers and components that suit most communication needs. KEP***Direct*** can interface your favorite Windows client with most **AUTOMATIONDIRECT** control systems. Compatible products are listed in the "Supported Devices" and "Supported I/O Modules" sections of this chapter.

Using KEP***Direct*** with your **AUTOMATIONDIRECT** I/O system provides the following advantages:

- Versatility: Connect your favorite HMI/SCADA* software to a variety of **AUTOMATIONDIRECT** I/O products.
- Cost Savings: Design a low-cost data acquisition/monitoring system using affordable **AUTOMATIONDIRECT** controllers and I/O modules.
- Convenience: Remotely control **AUTOMATIONDIRECT** systems from any client PC. For instance, connect to one of our DURApulse or GS series AC drives via a GS-EDRV module and have the ability to control start/ stop, speed and many other drive functions remotely.
- Custom Interfacing: Interface your custom VisualBASIC or Visual C++ application to **AUTOMATIONDIRECT** I/O systems and eliminate time-consuming driver development.

**See Glossary*

Diagram Showing the Basic System Compatibility



Preparing for Installation

Check your PC Hardware Requirements

The OPC server has minimum system requirements for both software and hardware. These requirements must be met in order for the application to operate as designed.

Minimum System Requirements

The OPC server requires the following hardware at a minimum:

- 2.0 GHz Processor
- 1 GB installed RAM
- 180 MB available disk space
- Ethernet Card (required for Ethernet drivers or serial drivers that utilize Ethernet encapsulation)
- Super VGA (800 x 600) or higher resolution video adapter and monitor
- Keyboard and Microsoft mouse or compatible pointing device

Supported Operating Systems

This application supports the following Microsoft Windows operating systems:

- Windows 10*
- Windows 7 Pro and Ultimate*
- Windows XP* SP2 (strongly recommended for industrial settings)
- Windows Server 2003* SP2 (strongly recommended for industrial settings)
- Vista Business/Ultimate*
- Windows Server 2008* (strongly recommended for industrial settings)

* Includes support for 64 bit operating systems.

Supported Devicesw

KEP*Direct* Compatible DL05/06/105 Series Controllers, Ports and Communications Modules				
CPU	**CPU Port 1**	**CPU Port 2**	**D0-DCM**	**H0-ECOM(*)**
D0-05(*)	RS-232	RS-232	RS-232/422/485	Ethernet
D0-06(*)	RS-232	RS-232/422/485	RS-232/422/485	Ethernet
F1-130(*)	RS-232	—	—	—

KEP*Direct* Compatible DL205 Series Controllers, Ports and Communications Modules					
CPU	**CPU Port 0**	**CPU Port 1**	**CPU Port 2**	**D2-DCM**	**H2-ECOM(*)**
D2-230	—	RS-232	—	—	—
D2-240	—	RS-232	RS-232	RS-232/422	Ethernet
D2-250-1	—	RS-232	RS-232/422	RS-232/422	Ethernet
D2-260	—	RS-232	RS-232/422/485	RS-232/422	Ethernet
H2-EBC(*)	—	Ethernet	—	—	—

KEP*Direct* Compatible DL305 Series Controllers, Ports and Communications Modules					
CPU	**CPU Port 1**	**CPU Port 2**	**D3-DCM**	**D3-232-DCU**	**D3-422-DCU**
D3-330	—	—	—	RS-232	RS-422
D3-340	RS-232	RS-232	—	RS-232	RS-422
D3-350	RS-232	RS-232/422	RS-232/422	—	—

KEP*Direct* Compatible DL405 Series Controllers, Ports and Communications Modules						
CPU	**CPU Port 0**	**CPU Port 1**	**CPU Port 2**	**CPU Port 3**	**D4-DCM**	**H4-ECOM(*)**
D4-430	RS-232	RS-232/422	—	—	RS-232/422	Ethernet
D4-440	RS-232	RS-232/422	—	—	RS-232/422	Ethernet
D4-450	RS-232	RS-232/422	RS-232	RS-232/422	RS-232/422	Ethernet
D4-454	RS-232	RS-232/422	RS-232	RS-232/422	—	—

KEP*Direct* Compatible Productivity Series Controllers, Ports and Communications Modules					
CPU	**CPU Port**	**CPU Port**	**CPU Port**	**P3-RS**	**P3-ESCM**
P3-550	Ethernet	RS-232	RS-485	RS-232/485	RS-422/485, Ethernet
P2-550	Ethernet	RS-232	RS-485	__	__
P2-RS	—	RS-232	RS-485	__	__
P1-540	Ethernet	RS-232	RS-485	__	__

KEP*Direct* Compatible GS Series Drives, Ports and Communications Modules		
Drive	**Drive Port**	**GS-EDRV**
GS1	__	Ethernet
GS2	__	Ethernet
DURAPulse	__	Ethernet

An asterisk suffix (*) means there are multiple part numbers. Example: part numbers H2-ECOM, H2-ECOM100, H2-ECOM-F are collectively referred to as H2-ECOM(*) in the chart.

Supported I/O Modules

H2- EBC(-F), H2-EBC100 and DL205	
Module	**Description**
D2-08ND3	8 Pt 12–21 VDC Input
D2-16ND3-2	16 Pt 24VDC Input
D2-32ND3	32 Pt 24VDC Input
D2-32ND3-2	32 Pt 5–15 VDC Input
D2-08NA-1	8 Pt 110VAC Input
D2-08NA-2	8 Pt 170–265 VAC Input, 2 Commons
D2-16NA	16 Pt 110VAC Input
F2-08SIM	8 Pt Switch Slide Simulator
D2-04TD1	4 Pt 12–24 VDC Sink Output
D2-08TD1	8 Pt 12–24 VDC Sink Output
D2-16TD1-2	16 Pt 12–24 VDC Sink (2 Terminals)
D2-08TD2	8 Pt 12–24 VDC Source Output
D2-32TD2	32 Pt 24VDC Source output
D2-16TD2-2	16 Pt 12–24 VDC Output (2 Terminals)
D2-32TD1	32 Pt 24VDC Output
D2-08TA	8 Pt 18–220 VAC Output
D2-12TA	12 Pt 18–110 VAC Output
D2-04TRS	4 Pt Isolated Relay 5–30 VDC or 5–250 VAC Output
D2-08TR	8 Pt Relay Output 12–28 VDC / 12–250 VAC
F2-08TA	8 Pt 24–140 VAC Output, 1.5 A/Pt.
F2-08TR	8 Pt Relay Output, 10A/Common 12–28 VDC / 12–250 VAC
F2-08TRS	8 Pt Relay Output 12–28 VDC or 12–250 VAC
D2-12TR	12 Pt Relay Output 5–30 VDC or 5–250 VAC
D2-08CDR	Combo 4 Pt 24VDC Input and 4 Pt Relay Output
F2-04AD-1	4 CH Analog In 4–20 mA 12-Bit Res
F2-04AD-1L	4 Pt Analog Input 4–20 mA 12-Bit 12 VDC Power Needed
F2-04AD-2	4 CH Analog Input Volt 12-Bit Res
F2-04AD-2L	4 Pt Analog Input Volt 12-Bit 12 VDC Power Needed
F2-08AD-1	8 CH Analog Input 4–20 mA 12-Bit
F2-08AD-2	8 CH Analog Input Voltage 12-Bit
F2-04RTD	4 CH RTD, 0.1 Deg C Resolution
F2-04THM	4 CH Thermocouple or 16-bit V Input
F2-02DA-1	2 CH Analog Output 4–20 mA 12-Bit
F2-02DA-2	2 CH Analog Output Voltage 12-Bit
F2-02DA-1L	2 CH Analog Output 12-Bit Use with 12VDC Power Supply
F2-02DA-2L	2 CH Analog Output 12-Bit Use with 12VDC Power Supply
F2-08DA-1	8 CH Analog Output 4–20 mA 12-Bit
F2-08DA-2	8 CH 0–5 VDC or 0–10 VDC 12-Bit Analog Output
F2-02DAS-1	2 CH Isolated 4–20 mA 16-Bit Analog Output
F2-02DAS-2	2 CH Isolated 4–20 mA 16-Bit Analog Output
F2-4AD2DA	4 CH In & 2 CH Out Analog 4–20 mA 12-Bit
F2-8AD4DA-1	8 CH Analog Input 0–20 mA 12-,14- or 16-Bit 4 CH Analog Output 4–20 mA 16-Bit
F2-8AD4DA-2	8 CH Analog In 0–5 VDC or 0–10 VDC 12-, 14- or 16-Bit. 4 CH Analog Output 0–5 VDC or 0–10 VDC 15- or 16-Bit.
H2-CTRIO	High Speed Counter/Pulse Output Module

H4-EBC(-F) and DL405	
Module	**Description**
D4-08ND3S	8 Pt 24–48 VDC Source Input
D4-16ND2	16 Pt 12–24 VDC Source Input
D4-16ND2F	16 Pt 12–24 VDC In Fast Response
D4-16SIM	16 Pt Switch Slide Simulator
D4-32ND3-1	32 Pt 24 VDC Sink Source Input
D4-32ND3-2	32 Pt 5–12 VDC Sink Source Input
D4-64ND2	64 Pt 20–28 VDC Source Input
D4-08NA	8 Pt 110–220 VAC Input
D4-16NA	16 Pt 110VAC Input
D4-16NA-1	16 Pt 220VAC Input Module
D4-16NE3	16 Pt 12–24 VAC/VDC Sink Input
F4-08NE3S	8 Pt 90–150 VAC/DC Sink Iso. Input
D4-08TD1	8 Pt 12–24 VDC Sink Output
F4-08TD1S	8 Pt 24–150 VDC SinkSource Isolated Output
D4-16TD1	16 Pt 5–24 VDC Sink Output
D4-16TD2	16 Pt 12–24 VDC Source Output
D4-32TD1	32 Pt 5–24 VDC Sink Output
D4-32TD1-1	32 Pt 5–24 VDC Sink Output
D4-32TD2	32 Pt 12–24 VDC Source Output
D4-64TD1	64 Pt 5–24 VDC Sink Output
D4-08TA	8 Pt 18–220 VAC Output
D4-16TA	16 Pt 18–220 VAC Output
D4-08TR	8 Pt Relay 5–30 VDC or 5–250 VAC Output
F4-08TRS-1	8 Pt Relay 12–30 VDC or 12–250 VAC Output
F4-08TRS-2	8 Pt Relay 12–30 VDC or 12–250 VAC Output
D4-16TR	16 Pt Relay 5–30 VDC or 5–250 VAC Output
F4-04AD	4 CH Analog Input Voltage/Current
F4-04ADS	4 CH Iso. Analog In Voltage/Current
F4-08AD	8 CH Analog Input Voltage/Current
D4-02DA	2 CH Analog Output Voltage/Current
F4-04DA	4 CH Analog Output Voltage/Current
F4-04DA-1	4 CH Analog Output Current
F4-04DA-2	4 CH Analog Output Voltage
F4-04DAS-1	4 CH Isolated. 4–20mA 16-Bit Analog Output
F4-04DAS-2	4 CH Isolated 16-Bit Analog Voltage Output
F4-08DA-1	8 CH Analog Output Current
F4-08DA-2	8 CH 0–5 VDC or 0–10 VDC 12-Bit Analog Output
F4-16DA-1	16 CH Analog Output Current
F4-16DA-2	16 CH 0–5 VDC or 0–10 VDC 12-Bit Analog Output
F4-08THM	8 CH Thermo Module For Type (J,E,K,R,S,T,B,N,C)
F4-08THM-n	8 CH Thermo Module For Type (J,E,K,R,S,T,B,C,P)
F4-08RTD	8 CH RTD Module
H4-CTRIO	High Speed Counter/Pulse Output Module

Supported I/O Modules (continued)

Terminator I/O	
Module	**Description**
T1K-08ND3	8 Pt 12–24 VDC Sink Source Input
T1K-16ND3	16 Pt 12–24 VDC Sink Source Input
T1K-08NA-1	8 Pt 110VAC Input
T1K-16NA-1	16 Pt 110VAC Input
T1K-08TD1	8 Pt 12–24 VDC Sink Output
T1K-08TD2-1	8 Pt 12–24 VDC Source Output
T1K-16TD1	16 Pt 12–24 VDC Sink Output
T1K-16TD2-1	16 Pt 12–24 VDC Source Output
T1K-08TA	8 Pt 110–240 VAC Output
T1K-16TA	16 Pt 110–240 VAC Output
T1K-08TAS	8 Pt 110–240 VAC Output Isolated Commons
T1K-08TR	8 Pt Relay Output 5–30 VDC or 5–240 VAC
T1K-16TR	16 Pt Relay Output 5–30 VDC or 5–240 VAC
T1K-08TRS	8 Pt Isolated Relay Output 5–30 VDC or 5–240 VAC
T1F-08AD-1	8 CH Analog Input 4–20 mA 14-Bit Resolution
T1F-08AD-2	8 CH Analog Input 14-Bit Resolution
T1F-16AD-1	16 CH Analog Input 4–20 mA 14-Bit Resolution
T1F-16AD-2	16 CH Analog Input Voltage 14-Bit Resolution
T1F-08DA-1	8 CH Analog Output 4–20 mA 12-Bit Resolution
T1F-08DA-2	8 CH Analog Output Voltage 12-Bit Resolution
T1F-16DA-1	16 CH Analog Output 4–20 mA 12-Bit Resolution
T1F-16DA-2	16 CH Analog Output Voltage 12-Bit Resolution
T1F-16RTD	16 CH. RTD
T1F-14THM	14 CH Thermocouple 16 Bit Resolution
T1F-8AD4DA-1	Terminator I/O 8 CH Analog Input 4 CH Analog Output Current
T1F-8AD4DA-2	Terminator I/O 8 CH Analog Input 4 CH Analog Output Voltage
T1H-CTRIO	High Speed Counter/Pulse Output Module

DL05/ DL06	
Module	**Description**
F0-08SIM	8 Pt Input Simulator
D0-10ND3	10 Pt 12–24 VDC Sink Source Input
D0-10ND3F	10 Pt Fast 12–24 VDC Sink Source Input
D0-16ND3	16 Pt 20–28 VDC Sink Source Input
F0-08NA-1	8 Pt 110VAC Input
D0-10TD1	10 Pt 12–24 VDC Sink Output
D0-16TD1	16 Pt 6–27 VDC Sink Output
D0-10TD2	10 Pt 12–24 VDC Source Output
D0-16TD2	16 Pt 12–24 VDC Source Output
D0-07CDR	4 Pt 12–24 VDC Input, 3 Pt Relay Output
D0-08TR	8 Pt Relay Output 6–27 VDC or 6–240 VAC
D0-08CDD1	4 Pt 12–24 VDC Input, 4 Pt 12–24 VDC Sink Output
F0-04TRS	4 Pt High Current Relay Output 5–30 VDC or 5–125 VAC
F0-04AD-1	4 CH Analog Input 0–20 mA or 4–20 mA 12 Bit Resolution
F0-08ADH-1	8 CH Analog Input 0–20 mA 16 Bit Resolution
F0-04AD-2	4 CH Analog Input 0–5 V or 0–10 V 12 Bit Resolution
F0-08ADH-2	8 CH Analog Input 0–5 V or 0–10 V 16 Bit Resolution
F0-04DAH-1	4 CH Analog Output 4–20 mA 16 Bit Resolution
F0-08DAH-1	8 CH Analog Output 4–10 mA 16 Bit Resolution
F0-04DAH-2	4 CH Analog Output 0–10 VDC 16 Bit Resolution
F0-08DAH-2	8 CH Analog Output 0–10 VDC 16 Bit Resolution
F0-4AD2DA-1	4 CH Input/2–CH Output Analog Combination 0–20 mA or 4–20 mA 12 Bit Resolution
F0-2AD2DA-2	2 CH Input/2–CH Output Analog Combination 0–5 VDC or 0–10 VDC 12 Bit Resolution
F0-4AD2DA-2	4 CH Input/2–CH Output Analog Combination 0–5 VDC or 0–10 VDC 12 Bit Resolution
F0-04RTD	4 CH RTD
F0-04THM	4 CH Thermocouple 16 Bit Resolution
H0-CTRIO	High Speed Counter/Pulse Output Module

Supported I/O Modules (continued)

DL-305 PLC	
Module	**Description**
D3-08ND2	8 Pt 18–36 VDC Source Input
D3-16ND2-1	16 Pt 18–36 VDC Source Input
D3-16ND2F	16 Pt Fast 18–36 VDC Source Input
D3-08NA-1	8 Pt 85–132 VAC Input
D3-08NA-2	8 Pt 220VAC Input
D3-16NA	16 Pt 110VAC Input
D3-08NE3	8 Pt 20–28 VAC/ VDC Sink Source Input
D3-16NE3	16 Pt 14–30 VAC/ VDC Sink Source Input
F3-04ADS	4 CH Analog Input 0–5 VDC, 0–10 VDC, –5–+5 VDC, –10– +10 VDC, 1–5 VDC, 0–20 mA, 4–20mA 12-Bit Resolution
F3-08AD-1	8 CH Analog Input 4–20mA 12-Bit Resolution
F3-16AD	16 CH Analog Input –5– +5 VDC, –10– +10 VDC, 0–10 VDC, 0–20 mA 12-Bit Resolution
D3-08SIM	8 Pt Input Simulator
D3-08TD1	8 Pt 5–24 VDC Sink Output
F3-16ND3F	16 Pt 5VDC (TTL & CMOS) or 12–24 VDC Sink Source Input
D3-08TD2	8 Pt 5–24 VDC Source Output
D3-16TD1-1	16 Pt 5–24 VDC Sink Output
D3-16TD1-2	16 Pt 5–24 VDC Sink Output
D3-16TD2	16 Pt 5–24 VDC Source Output
D3-04TAS	4 Pt 80–265 VAC Output
F3-08TAS	8 Pt 12–250 VAC Output
F3-08TAS-1	8 Pt 20–125 VAC Output
D3-08TA-1	8 Pt 80–265 VAC Output
D3-08TA-2	8 Pt 80–265 VAC Output
F3-16TA-2	16 Pt 20–125 VAC Output
D3-16TA-2	16 Pt 15–265 VAC Output
F3-04DAS	4 CH Analog Output 0–5 VDC, 0–10 VDC, –5 – +5 VDC, –10– +10 VDC, 0–20 mA, 4–20 mA 12-Bit Resolution
F3-04DA-1	4 Ch Analog Output 0–5 VDC, 0–10 VDC, 4–12 mA, 4–20 mA 12-Bit Resolution
D3-08TR	8 Pt Relay Output 5–30 VDC / 5–265 VAC
F3-08TRS-1	8 Pt Relay Output 12–30 VDC or 12–250 VAC
F3-08TRS-2	8 Pt Relay Output 12–30 VDC or 12–125 VAC
D3-16TR	16 Pt Relay Output 5–30 VDC / 5–265 VAC
F3-08THM-x	8 CH Thermocouple

Productivity3000	
Module	**Description**
P3-16SIM	16 Pt Input Simulator
P3-08ND3S	8 Pt 12–24 VDC Sink Source Input
P3-16ND3	16 Pt 12–24 VDC Sink Source Input
P3-32ND3	32 Pt 24VDC Sink Source Input
P3-64ND3	64 Pt 24VDC Sink Source Input
P3-08NAS	8 Pt 110–240 VAC Input
P3-16NA	16 Pt 110–240 VAC Input
P3-08TD1S	8 Pt 6–27 VDC Sink Output
P3-08TD2S	8 Pt 6–27 VDC Source Output
P3-16TD1	16 Pt 6–27 VDC Sink Output
P3-16TD2	16 Pt 6–27 VDC Source Output
P3-32TD1	32 Pt 6–27 VDC Sink Output
P3-32TD2	32 Pt 6–27 VDC Source Output
P3-64TD1	64 Pt 6–27 VDC Sink Output
P3-64TD2	64 Pt 6–27 VDC Source Output
P3-08TAS	8 Pt 110–240 VAC Output
P3-16TA	16 Pt 110–240 VAC Output
P3-08TRS	8 Pt Isolated Relay Output 6–27 VDC or 6–240 VAC
P3-16TR	16 Pt Relay Output 6–27 VDC or 6–240 VAC
P3-08TRS-1	8 Pt Isolated Relay Output 5–30 VDC or 5–240 VAC
P3-04ADS	4 CH Iso. Analog In Voltage/Current
P3-08AD	8 CH Analog Input Voltage/Current
P3-16AD-1	16 CH Analog Input Current
P3-16AD-2	16 CH Analog Input Voltage
P3-08RTD	8 CH RTD Module
P3-08THM	8 CH Thermo Module For Type (J,E,K,R,S,T,B,N,C)
P3-04DA	4 CH Analog Output Voltage/Current
P3-08DA-1	8 CH Analog Output Current
P3-08DA-2	8 CH Analog Output Voltage
P3-06DAS-1	6 CH Isolated Analog Output Currentw
P3-06DAS-2	6 CH Isolated Analog Output Voltage
P3-16DA-1	16 CH Isolated Analog Output Current
P3-16DA-2	16 CH Isolated Analog Output Voltage
P3-8AD4DA-1	Combo 8 CH Analog Input Current and 4 CH Analog Output Current
P3-8AD4DA-2	Combo 8 CH Analog Input Voltage and 4 CH Analog Output Voltage

Go to the **Automationdirect.com** website for information on any device introduced after publication of this manual.

Installation of KEP*Direct* OPC Server

The KEP***Direct*** software installs as a demo package with full development capabilities, but with a restricted Runtime application. The Runtime application in the demo version is restricted to two hours of operation. However, with the purchase of a software license, this restriction is removed. Download the demo of the software from AutomationDirect.com(http://support.automationdirect.com/demos.html) The following section outlines the step by step instructions for the install.

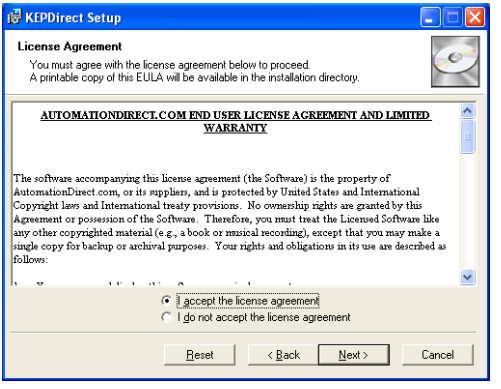
Step 1: Welcome Window

The KEP***Direct*** install wizard starts and opens the **Welcome** window shown below. This window issues a reminder to exit all other Window applications. If you are unsure of the programs that may be running, open the Task Manager by pressing the Ctrl-ALT-Delete keys at the same time. Close any opened applications by selecting them and clicking on the **End Task** button of the Task Manager. Click on the **Next** button to proceed with the installation.

Step 2: License Agreement

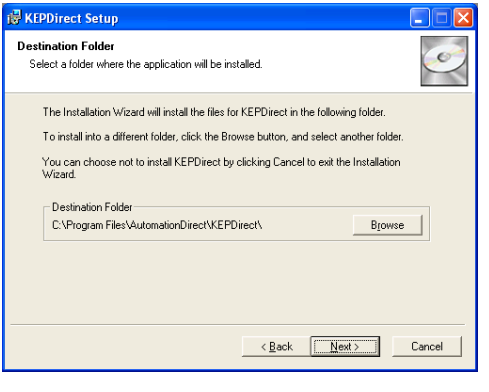
The wizard displays the **License Agreement** shown below. Read the agreement and select **I accept the license agreement** if you agree with the terms and conditions.

Select Next to continue with the installation.



Step 3: Select the Installation Directory

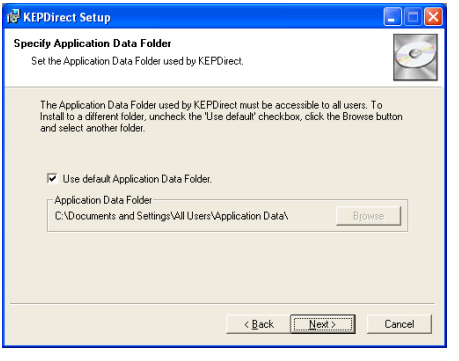
The **Destination Folder** window displays the folder (or directory) where the KEP***Direct*** program files will be installed. You may choose a different folder by clicking on the **Browse** button and selecting a preferred directory.



If you accept the displayed folder or select a preferred directory, click on the **Next** button to continue with the KEP***Direct*** installation.

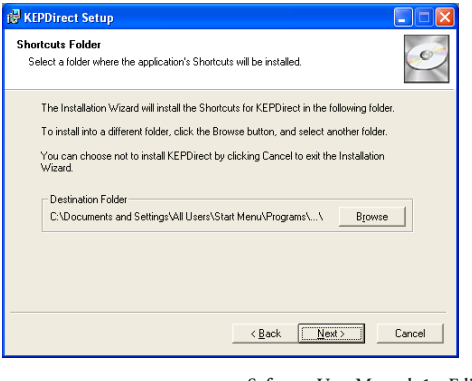
Step 4: Select the Application Directory

The **Specify Application Data Folder** window displays the folder (or directory) where the KEP***Direct*** application files will be installed.



Application files such as security files and certifications, will be stored in this folder. Although it is not recommended, you may choose a different folder by clicking on the **Browse** button and selecting a preferred directory. If you accept the displayed folder or select a preferred directory, click on the **Next** button to continue with the KEP***Direct*** installation.

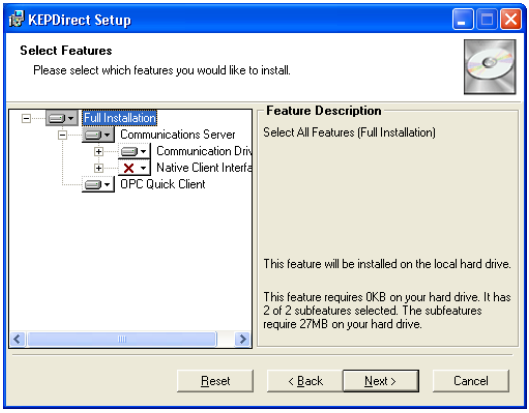
Step 5: Select the Shortcuts Directory



The **Shortcuts Folder** window displays the folder (or directory) where the KEP***Direct*** shortcuts will be stored. You may choose a different folder by clicking on the **Browse** button and selecting a preferred directory. If you accept the displayed folder or select a preferred directory, click on the **Next** button to continue with the KEP***Direct*** installation.

Step 6: Select Features

The wizard now displays the **Select Features** window. This window allows you to select the components to install for use with KEP***Direct***.

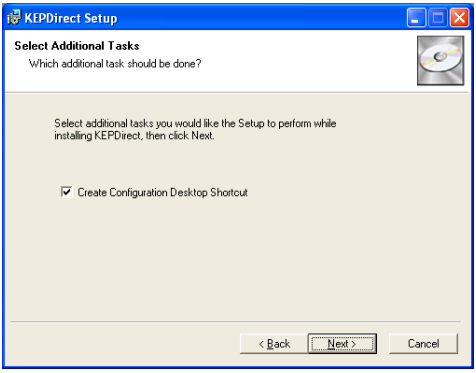


When a main component is selected, all sub-components are selected automatically. This window includes a **Description** box which provides a brief description of each component when it is selected.

When all selections have been completed, click on the **Next** button to proceed with installation.

Step 7: Select Additional Tasks

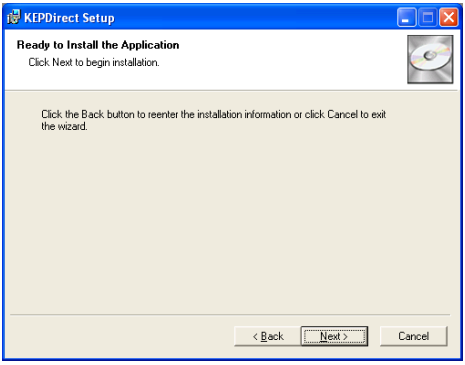
The **Select Additional Tasks** window allows you to select any other tasks you would like the setup to perform while installing the KEP***Direct*** program. Check or uncheck the boxes next to the options that you wish to accept or decline. Once all selections are made, click on the **Next** button to continue with the installation.



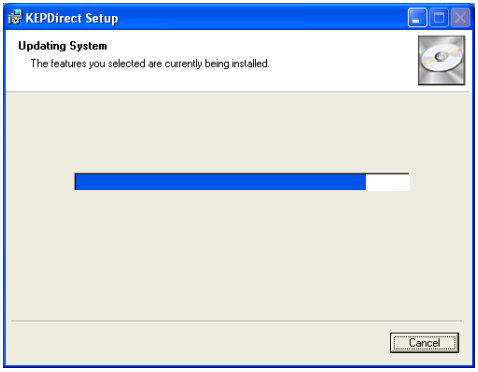
Step 8: Ready to Install the Application

The wizard now displays the **Ready to Install the Application** window shown below. This window is an alert that the actual install will begin next. Verify that the installation information is correct before continuing. If incorrect, click on the **Back** button to return to previous windows. If correct, click on the **Next** button to start copying the KEP***Direct*** files to your PC's hard drive.

Step 9: System Status

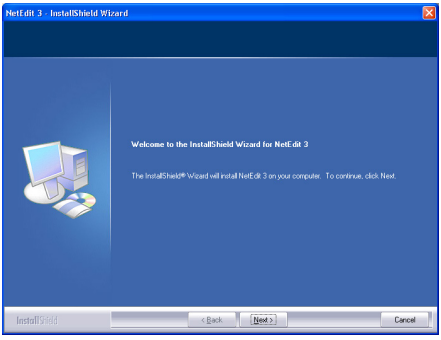


The installation **Updating System** window opens and displays the progress of the installation and setup of the KEP***Direct*** files. A progress bar is displayed indicating the amount of the process that is completed.



Step 10: Install NetEdit

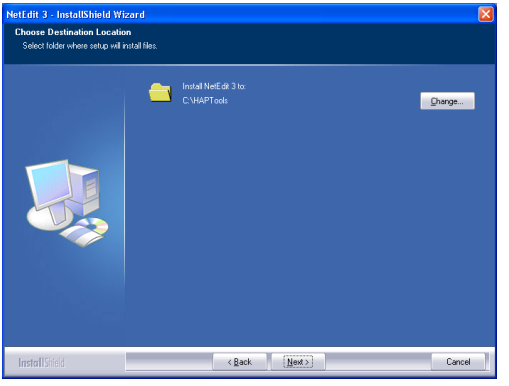
After the Kep***Direct*** files have been copied to your PC, the **NetEdit 3** install wizard will appear.



NetEdit 3 is a tool for configuring and troubleshooting EBC or ECOM modules for ***Direct***LOGIC systems. If you are planning to use EBC or ECOM modules please install this application if it's not already on your PC. Click on the **Next** button to proceed with the installation.

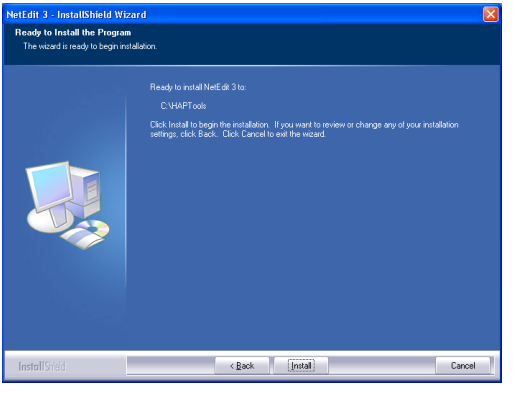
NOTE: *NetEdit3 is required to setup DirectLOGIC Ethernet devices.*

For those who continue with the **NetEdit 3** installation, you will be asked to choose a destination folder where the install will store needed files. The default destination folder is shown.

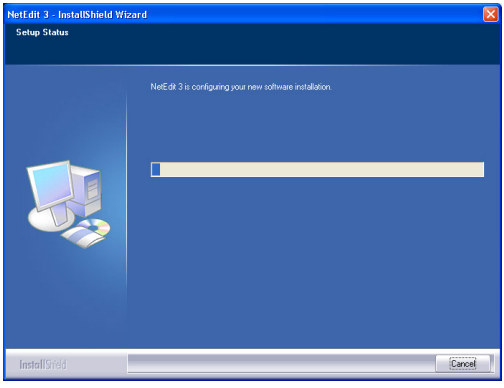


Click the **Change** button to choose a different location (it is not recommended to change the location of the destination folder). Select **Next** to continue with the installation.

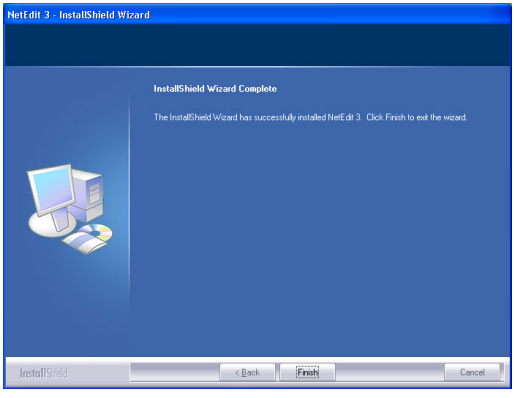
The **NetEdit 3** wizard is now ready to install the application on your PC. **The Ready to Install the Program** screen will appear allowing you to review and make any changes needed before the download begins. If changes are needed click the **Back** button, if no changes are needed, click the **Install** button.



The following **Setup Status** window will show the status of the install. The progress bar indicates how much of the process is completed.



The following window is displayed once the install is completed. It is an indicator that the **NetEdit 3** installation was successful. Clicking **Finish** will close the **NetEdit 3** installation wizard and return you to the Kep***Direct*** installation wizard.



Step 11: Installation Complete

After exiting the NetEdit 3 installation, the KEP***Direct*** Setup window below will be displayed. This **Installation Complete** window indicates that the KEP***Direct*** install was successful. Click **Finish** to close the window and exit the installation.

Step 12: KEPDirect OPC Server Registration

Upon purchase of the full package of KEP***Direct***, the purchaser will be sent an email containing the Activation ID and a link to the Licensing Portal. With the new software installed on your PC, run the license utility and enter the Activation ID Code from the email. The license utility will then generate an activation_request.txt file.

NOTE: *If someone other than you purchased the software, the ID Code email would have been sent to them. If you have no access to the ID Code email and need assistance, have your purchase information available and call AutomationDirect's Tech Support at (770) 844-4200.*

Open the Licensing Portal by clicking the link supplied in the email or use the link provided on the AutomationDirect software download page. Complete the required information and select the previously created activation_request.txt file. Click the "Generate Activation Response File" button. A response.txt file will be generated which you will then use in the license utility of your KEP***Direct*** software to activate the full version. Only the machine that generated the activation_ request.txt file will be activated by the response.txt file.

The following section describes the activation procedure in greater detail.

After purchasing the full version of KEP***Direct*** OPC Server, you will receive an Activation ID email that will resemble the following:

Dear valued customer,

Please find your installation key code(s) and registration procedures for your recently purchased KEP***Direct*** software, PC-KEPOPC.

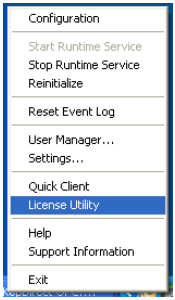
Activation ID Code: AAAAAAAA-BBBB-CCCC-DDDD-EEEEEEEEEEE

** Use this code for installing your software **

Your KEP***Direct*** OPC Server software installs initially in a Demo mode. In this mode you can configure, connect and run the server; however, it will time out 2 hours after starting the device driver. Please follow these steps to register your software online and activate your full, unrestricted software license.

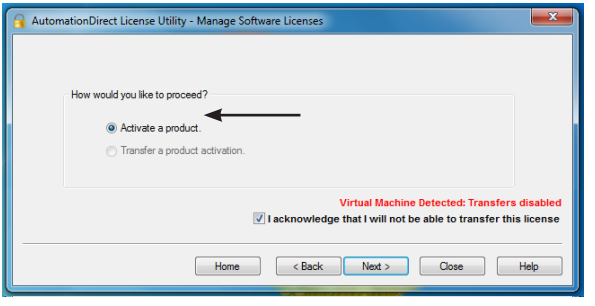
(The software should be installed prior to following these steps)

1. To access the License Utility, right-click on the server Administration menu (located in the system tray) and then select License Utility.

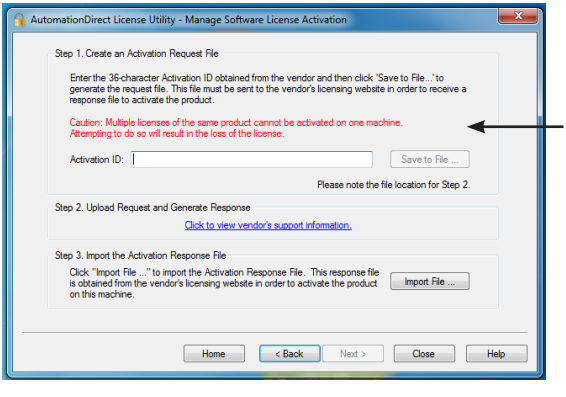


2. With the License Utility opened, Select Manage Software Licenses from the Select an option drop down, and then click Next.

3. Select the Activate a product option, and then click Next.

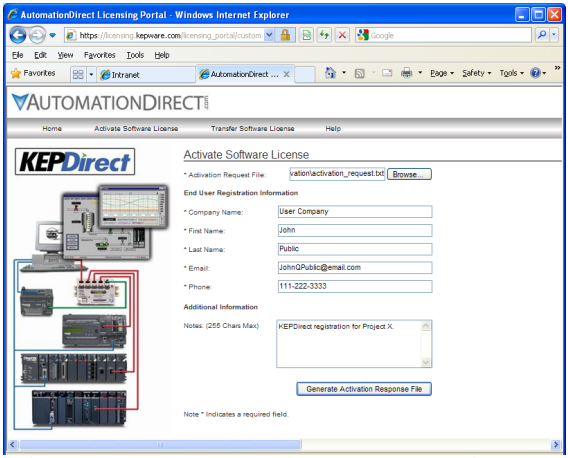


4. Enter the Activation ID, which was assigned by AutomationDirect with the purchased product, in the Create Activation Request File section and then click the Save to File button.



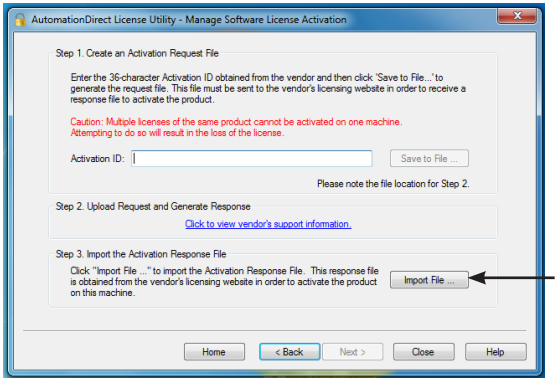
5. The License Utility will display a file Save As dialog, allowing you to save the generated activation request file, “activation_request.txt”. Save the activation request file, making note of its location as it will be sent to the AutomationDirect licensing portal in the next step.

6. Using the online license management tool (Licensing Portal) at https://licensing.kepware.com/licensing_portal/customer-center.aspx?id=AutomationDirect, browse to the activation request file. Complete all required fields for the End User Registration Information and click the Generate Activation Response File button.

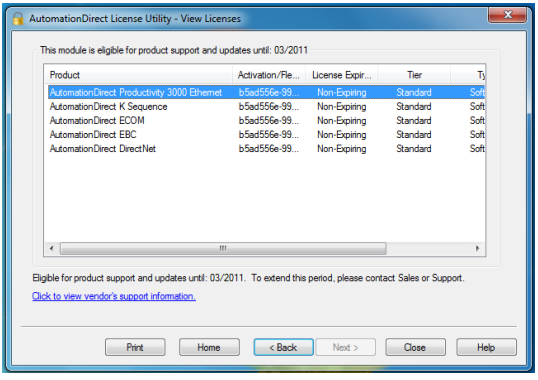


A file save dialog will pop up and an activation response file will be returned. Save the activation response file, making note of its location. If you do not see this file save dialog, please check your pop-up blocker settings and try again.

7. Next, return to the License Utility and click the Import File button located on the Manage Software License Activation page. Browse to the Activation Response File you received from the online license management tool, and then click the Open button.



8. Upon successful activation, the View Licenses page will be displayed in the License Utility showing the licensing information for the products activated.



An Emergency Activation ID is provided with each purchased software license to be used in the event of a machine/system failure occurring outside of standard business hours. The emergency license can be activated through the online license management tool and will allow the product to run for a period of 7 days. The emergency activation ID is a unique, one-time use license. A separate emergency activation ID must be used for each product that requires emergency activation. After using the emergency license, users must contact AutomationDirect Technical Support to discuss options for re-commissioning the application. 770-844-4200

Emergency Activation ID: AAAAAAAA-BBBB-CCCC-DDDD-EEEEEEEEE
** Use only in emergency situation as described above **

Best regards,

AutomationDirect Customer Service
www.automationdirect.com
1-800-633-0405

KEPDirect OPC Server Setup Summary

KEP***Direct*** OPC Server is a software driver that provides a means of communication between other software (OPC* clients) and components. Another way of looking at it is that KEP***Direct*** is a translator server that provides the communication bridge between the software (OPC* clients) and components.

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 and LinkMaster* products provided in this package have 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.

LinkMaster has the capabilities of both a “server” and a “client*” application, allowing it to access, collect, organize, and link data from other OPC servers and offer that data to any OPC/DDE client. It provides the means of linking data between OPC servers; thus, acting as a universal bridge for OPC server/client components.

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

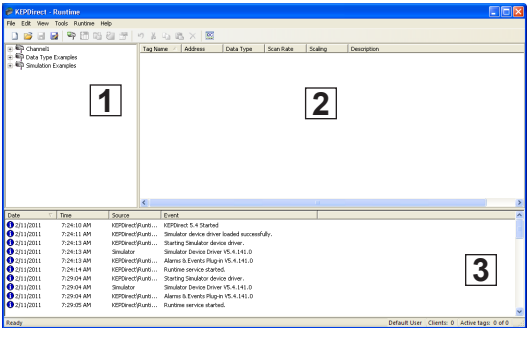
** See Glossary*

Create a New Project for Serial Devices

Step 1: Launch KEP*Direct* and Get to Know the Work Areas

Click on the KEP***Direct*** OPC Server icon, shown on the left, to launch the application. The application will start and open the window shown below.

When the KEP***Direct*** OPC Server application is launched for the first time, it opens with an operational simulation program.



This is a sample program which can be used to practice adding devices and channels. The window is divided into three areas as explained below (see window above for reference):

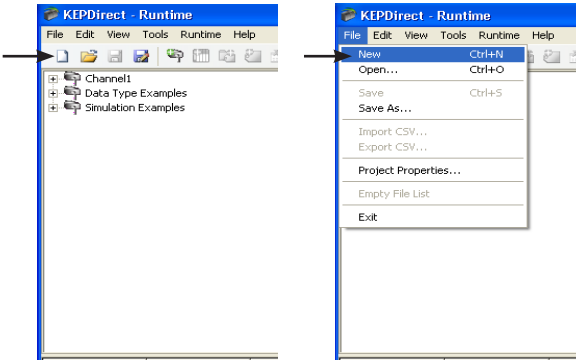
1. This area displays any existing channels, devices, and groups in a project. In addition, this area allows new channels, devices and groups to be created within a project.
2. This is the tag entry window of the server. Tags that you enter for a given device or tag group will be displayed here.
3. This area is the event log window of the server. Any of the messages generated by the server or the underlying driver will be displayed here.

Step 2: Things to confirm before Adding and Configuring a Channel

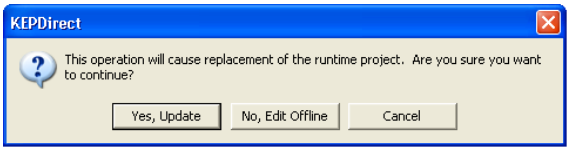
Before adding a new channel to your new project, first confirm that the following items have been completed or are available:

1. KEP***Direct*** OPC Server has been installed and launched properly.
2. Any devices to be added to your project must be installed with modules in place, powered up, and working properly.
3. In order to communicate serially between the KEP***Direct*** OPC Server and a device, such as a PLC, the device's serial port must be established. Please refer to the proper user manual in order to setup the serial communications port on your device.

Once all the above items are confirmed and ready, either click the **New Project** icon on the tool bar or click **File > New** from the menu bar to start your new project.



When you choose to open a new project the following window will appear asking you if you would like to replace the runtime project. The runtime project is currently a simulation that loaded on startup. Select **Yes, Update** to continue with a new project.

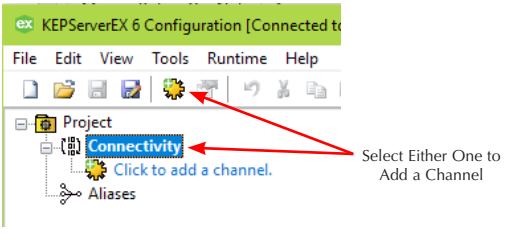


Step 3: Adding and Configuring a Channel

A channel refers to a specific communications driver. You can define a number of channels within a single project for organizational purposes. A channel acts as the basic building block of an OPC link. Each channel name must be unique in a KEP***Direct*** project. The channel name can be up to 256 alphanumeric characters in length (no spaces or special characters).

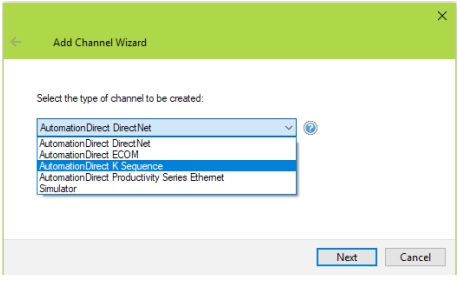
To add a new channel to your project, click on either the "**Click to add a channel"** text in the OPC Server window, the **New Channel** icon on the toolbar, **Edit > Connectivity > New Channel** from the menu bar or right click on the Connectivity selection in the OPC Server window.

As soon as you click on either of the above, the **New Channel** window shown below will open.



Step 4: Select the Device Driver

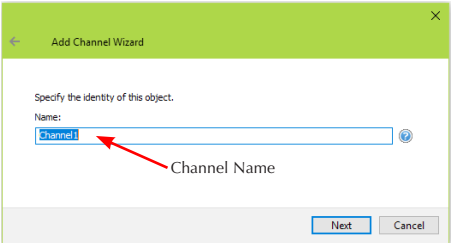
The **Device Driver** window will open. The arrow to the right side of the name provides a drop down list of device drivers that are available on your system. From this list, select **AUTOMATIONDIRECT *Direct*NET or K Sequence** (K Sequence is the preferred serial driver). After selecting the device driver, click on the **Next** button to go to the next step.



Step 5: Name the Channel

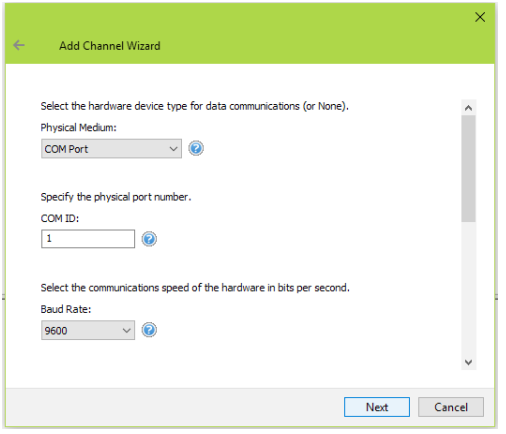
As previously mentioned, each channel name in a KEP***Direct*** project must be unique and is limited to 256 characters. While using long descriptive names is generally a good idea, keep in mind that some OPC client applications may have a limited display window when browsing the tag space of an OPC server. The channel name entered here will be part of the OPC browser information.

The **Identification** window prompts you to type a unique name for the channel by typing over the default **Channel1** label. Once the name has been typed, click on the **Next** button.



Step 6: Communications Setup

Next, the wizard will prompt you for the communications parameters. Set up the parameters in the **Communications** window shown below.



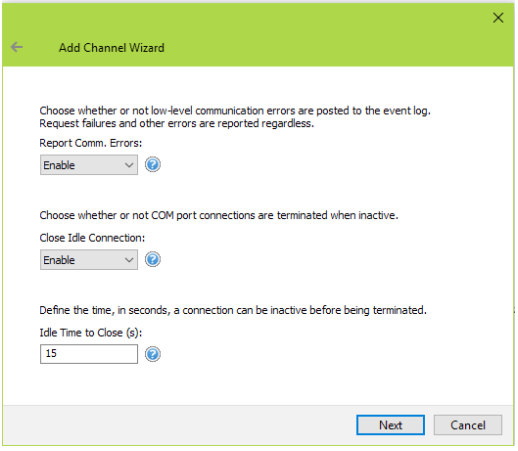
Physical Medium:

- **COM Port:** Select Com Port to display and configure the Serial Prot Settings section.
- **Use Ethernet Encapsulation:** Many serial drivers also support Ethernet Encapsulation mode, which uses an Ethernet based serial port gateway instead of the normal PC based serial port.
- **COM ID:** specifies the communications ID that will be used when communicating with devices assigned to the channel. The valid range is 1 to 9991 to 16. The default is 1.
- **Baud Rate:** specifies the baud rate that should be used with the selected communications port.

Once completed, click on the **Next** button to continue with the setup.

Step 7: Communications Setup, continued

Next, the wizard will prompt you for the communications parameters. Set up the parameters in the **Communications** window shown below.



- **Report Comm. Errors:** turns the reporting of low level communications errors On or Off. When enabled, low-level errors (such as parity, framing and overrun errors) are posted to the Event Log when they occur.

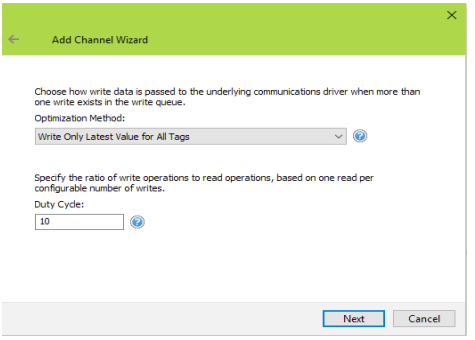
- Close Idle Connection: Choose to close the connection when there are no longer any tags being referenced by a client on the channel. The default is Enable.

- Idle Time to Close (s): Specify the amount of time that the server waits once all tags have been removed before closing the COM port. The default is 15 seconds.

Once completed, click on the Next button to continue with the setup.

Step 8: Communications Setup, continued

Optimization Method: controls how write data is passed to the underlying communications driver.



The options are:

- Write All Values for All Tags: This option forces the server to attempt to write every value to the controller. In this mode the server continues to gather OPC write requests and add them to the server internal write queue. The server processes the write queue and attempts to empty the queue by writing data to the device as quickly as possible.

  This mode insures that everything written from the client applications is sent to the target device. This mode should be selected if the order of your write operations or the content of every write item must uniquely be seen at the target device.

- Write Only Latest Value for Non-Boolean Tags: Many consecutive writes to the same value can accumulate in the write queue due to the time required to actually send the data to the device. If the server updates a write value that has already been placed in the write queue, far fewer writes are needed to reach the same final output value. In this way, no extra writes accumulate in the server's queue. When the user stops moving the slide switch, the value in the device is at the correct value at virtually the same time. As the mode states, any value that is not a Boolean value is updated in the server's internal write queue and sent to the device at the next possible opportunity. This can greatly improve the application performance.

NOTE: *This option does not attempt to optimize writes to Boolean values. It allows users to optimize the operation of HMI data without causing problems with Boolean operations, such as a momentary push button.*

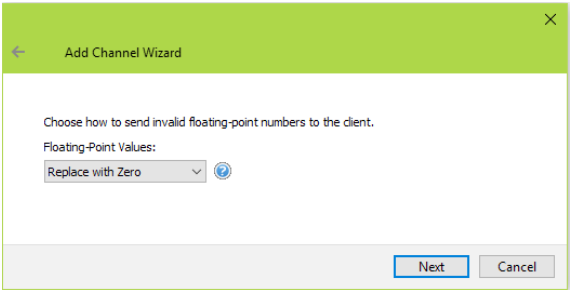
- **Write Only Latest Value for All Tags:** This option takes the theory behind the second optimization mode and applies it to all tags. It is especially useful if the application only needs to send the latest value to the device. This mode optimizes all writes by updating the tags currently in the write queue before they are sent. This is the default mode.

Duty Cycle: is used to control the ratio of write to read operations. The ratio is always based on one read for everyone to ten writes. The duty cycle is set to ten by default, meaning that ten writes occur for each read operation. Although the application is performing a large number of continuous writes, it must be ensured that read data is still given time to process. A setting of one results in one read operation for every write operation. If there are no write operations to perform, reads are processed continuously. This allows optimization for applications with continuous writes versus a more balanced back and forth data flow.

NOTE: *It is strongly recommended that you characterize your application for compatibility with these write optimization enhancements before using them in a production environment.*

Once completed, click on the **Next** button to continue with the setup.

Step 9: Handling Invalid Floating Point Numbers

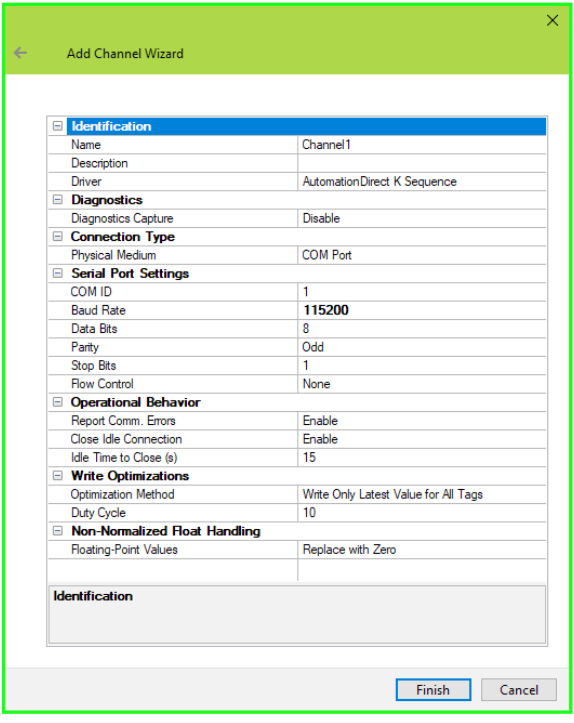


- **Replace with Zero:** This option allows a driver to replace non-normalized IEEE-754 floating point values with zero before being transferred to clients.
- **Unmodified:** This option allows a driver to transfer IEEE-754 de-normalized, normalized, non-number, and infinity values to clients without any conversion or changes.

Once completed, click on the Next button to continue with the setup.

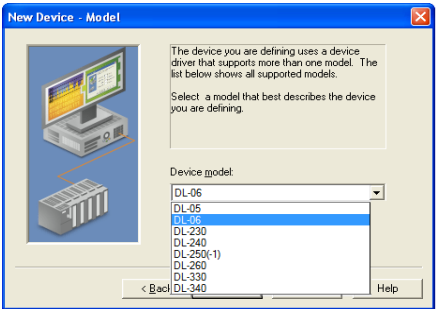
Step 10: Review the Channel Setup Summary

On the Add Channel Wizard final summary screen you will be able to view all the settings assigned in previous windows. There are additional Serial Port settings available that can be chosen on this screen.



Step 11: Select the Device Model

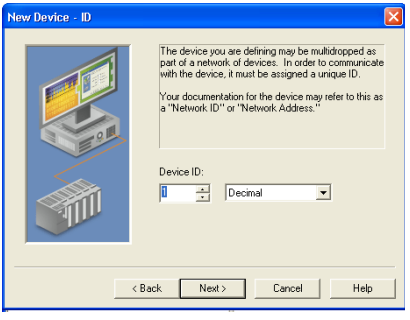
The next window which opens will be the **Model** window shown below. Click on the arrow to open the drop down list of the models available for setup. Select the model of the device which you are adding, i.e. DL06 PLC.



Once the device model is selected, click on the **Next** button to continue with the device installation.

Step 12: Type the Device ID

The next window to open is the **ID** window shown below. The device



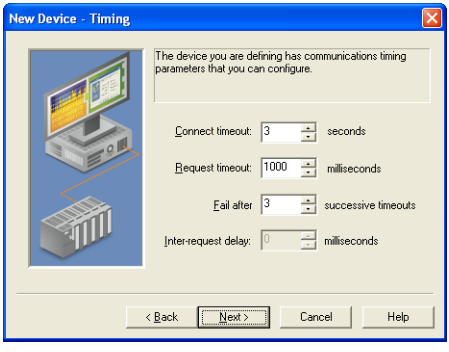
ID parameter allows you to specify the driver specific station or node for a given device. Up to 1024 devices may be defined on a given channel. For serial drivers, the ID is a numeric value.

The format of the entered numeric value can be changed to suit the needs of either your application or the characteristics of the chosen

communication driver. By default, the format is set by the driver; either Decimal, Octal or Hexadecimal.

Step 13: Set the Device's Communication Timing Parameters

The **Timing** window shown below opens and displays the available communication settings.



This screen provides three different communication settings for the device being installed. The following list explains the purpose of each available setting.

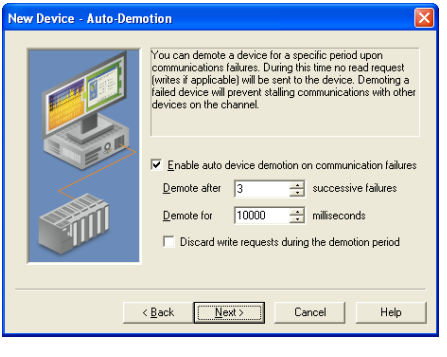
- **Connection Timeout**: The Connection timeout is used by Ethernet based drivers and therefore is disabled during this setup.
- **Request Timeout:** This is used to determine how long the driver will wait for a response from the target device. The request timeout has a valid range of 100 to 30000 milliseconds. The default is typically 1000 milliseconds but can vary depending on the specific nature of the chosen driver. The default timeout for most serial drivers is based on a baud rate of 9600 baud or better. When using the driver at lower baud rates, you may need to increase the timeout to compensate for the increased time required to acquire data.
- **Fail After:** This parameter is used to determine how many times the driver will retry a communications request before considering the request to have failed. The valid range is 1 to 10 retries. The default is typically three retries but can vary depending on the specific nature of the driver. The number of retires configured for your application is dependent largely on your communications environment.

If your environment is prone to noise induced communication failures you may want to increase the number of retries the driver performs. Keep in mind, however, that when the driver does encounter a communication issue, it will attempt to reacquire the data for the lost request. Based on the Request timeout and the Fail after count the driver will pause on a specific request until either the device responds or the timeout and retires have been exceeded. With this in mind you wouldn't want to set the timeout to 30000 milliseconds and 10 retires with the hope of covering every possible issue as this would result in a potential communications pause of 5 minutes.

- **Inter-Request Delay:** This parameter is not supported with Automation Direct products.

Once the three communication parameter settings have been selected, click on the **Next** button to continue with the device installation.

Step 14: Set the Auto-Demotion Parameters



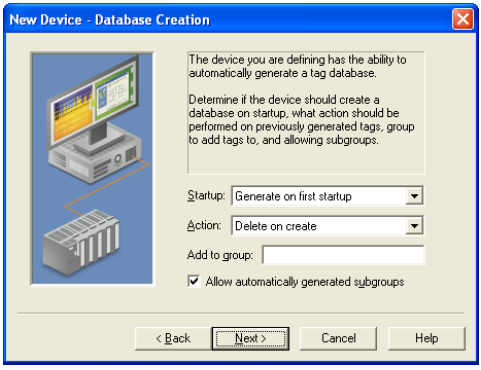
The **Auto-Demotion** parameters allow a driver to temporarily place a device off-scan in the event that a device is not responding. By placing a nonresponsive device offline for a specific time period, the driver can continue to optimize its communications with other devices on the same channel. After the time period has been reached, the driver will reattempt to communicate with the nonresponsive device. If the device is responsive, the device will be placed on-scan; otherwise, it will restart its off-scan time period.

Step 15: Set the Database Settings

The KEP***Direct*** I/O Server has a feature to automatically generate an OPC tag database. This has been designed to make the setup of your OPC application a Plug and Play operation. Communications drivers can be configured to automatically build a list of OPC tags within the server that correspond to device specific data. The automatically generated OPC tags can then be browsed from your OPC client. The OPC tags that are generated depend on the nature of the supporting driver.

If the target device supports its own local tag database, the serial driver can import the device's tag information in the form of a CSV file and then use this data to generate OPC tags within the server. Refer to the "Creating Tags" section of this manual for more information on CSV files.

The mode of operation for automatic tag database generation is completely configurable. The following **Database Creation** window, shown below, will open next. It allows you to configure how the server and the associated communications driver will handle automatic OPC tag database generation.



The following list explains the purpose of each available setting and group name.

- **Startup:** This selection allows you to configure when OPC tags will be automatically generated. There are three possible selections:
  - **Do not generate on startup:** This selection prevents the driver from adding any OPC tags to the tag space of the OPC Server. This is the default selection.

- **Always generate on startup:** This selection causes the driver to always import the device's tag information and to add OPC tags to the tag space of the server each time the runtime file is initialized. The runtime file is initialized whenever a new or existing project is opened or by selecting **Runtime > Connect** or **Runtime > Reinitialize** from the toolbar.
- **Generate on first startup:** This selection will cause the driver to import the target device's tag information the first time the runtime file is initialized and to add any OPC tags to the server tag space as needed.

NOTE: *When the Startup is selected, any tags that are added to the server's tag space must be saved with the project.*

When Startup (automatic tag generation) is enabled, the server needs to be told how to handle driver generated OPC tags and tags created by the user.

- **Action:** This selection allows you to control how the server will handle OPC tags that were automatically generated and currently exist in your project. This feature prevents automatically generated tags from piling up in the server.
  - **Delete on create:** This is the default condition which allows the server to remove any tags that had previously been added to the tag space before the communications driver can add any new tags.
  - **Overwrite as necessary:** This condition allows the server to remove only the tags that the communications driver is replacing with new tags. Any tags that are not being overwritten will remain in the server's tag space.
  - **Do not overwrite:** This condition prevents the server from removing any tags that had been previously generated or that may have already existed in the server. With this selection, the communications driver can only add completely new tags.
  - **Do not overwrite, log error:** This condition has the same effect as the previous condition but it also adds an error message that is posted to the OPC Server's event log when a tag overwrite would have occurred.

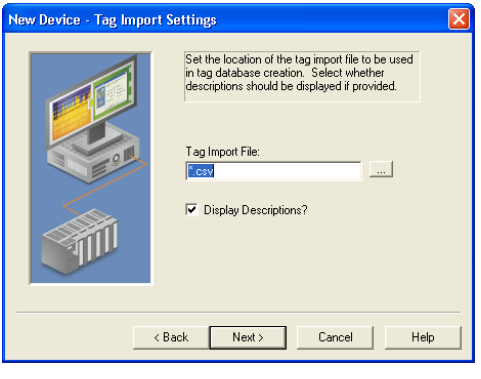
NOTE: *The removal of OPC tags affects tags that have been automatically generated by the communications driver and any tags you have added using names that match generated tags. It is recommended that you try to avoid adding your own tags to the server using names that match tags that may be automatically generated by the driver.*

- **Add to group**: This parameter can be used to aid in keeping automatically generated tags from mixing with tags that you may enter manually. With this parameter you may specify a sub group that will be used when adding all automatically generated tags for this device. The name of the sub group can be up to 256 characters in length.

Once the database settings are completed, click on the **Next** button to continue with the device installation.

Step 16: Tag Import Settings

The next dialog the wizard presents is the **Tag Import Settings** window shown below.



Here you are to enter the exact location of the ***Direct***SOFT export file you want to import tags from (see the "Creating Tags" section for more information on creating CSV files).

Click on the button next to the file entry window to select the file to be imported. You can also type the location and the file name if you wish. Only two types of files can be imported:

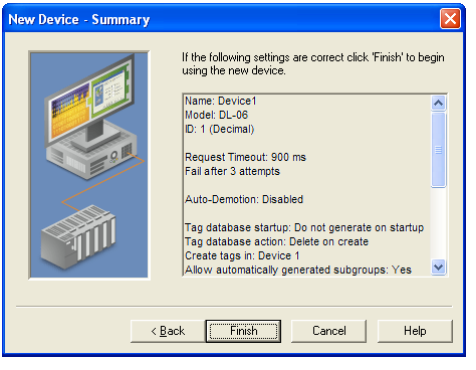
- Program (via export), .txt extension
- Element Documentation (via export), Standard Format, .csv extension

After selecting the import tag file, you have the option to have the tag descriptions imported or not. Check the **Display Descriptions?** box to have the descriptions imported. If necessary, a description will be given to tags with long names stating the original tag name.

Click the **Next** button to continue.

Step 17: Review the Device Setup Summary

The final window of the device setup is the **Summary** window shown below.



Review the summary and click on **Finish** to complete the device setup. If server tag database was set to automatically generate, then the CSV file selected in the previous step will be used to create the database as soon as the device setup is completed. If the database creation was set to manual, refer to the "Creating Tags" section for information on adding tags to the project.

Step 18: Repeat These Steps for Additional Devices

Repeat these steps if more serial devices are to be added to the selected channel.

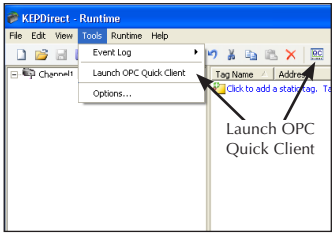
Using OPC Quick Client to Test a Serial Device Setup

Once all of your channels, devices and tags have been created, your project is ready to launch the **OPC Quick Client**. Before launching the client, you will want to save your project to the disk. Do this by clicking on **File > Save As**. Name the project and save it.

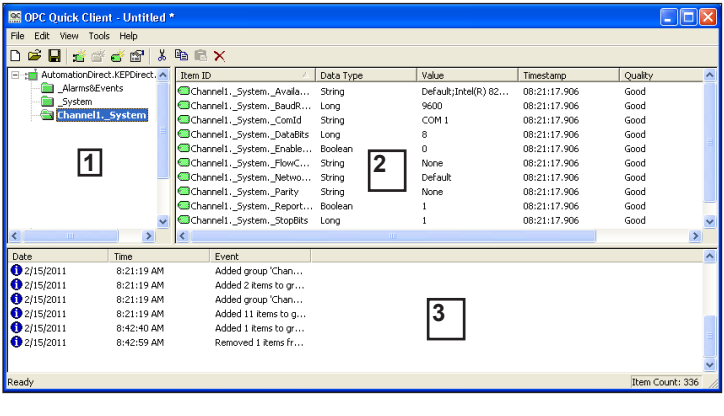
The following steps will show how to test a serial device with the **OPC Quick Client** application.

Step 1: Launch the OPC Quick Client

From the KEP***Direct*** OPC Server program window, either click on the **Tools** menu and select the "**Launch OPC Quick Client"** text or click on the **Launch OPC** icon on the tool bar shown in the figure below to launch the **OPC Quick Client** window.



This will open the **OPC Quick Client** window shown here.



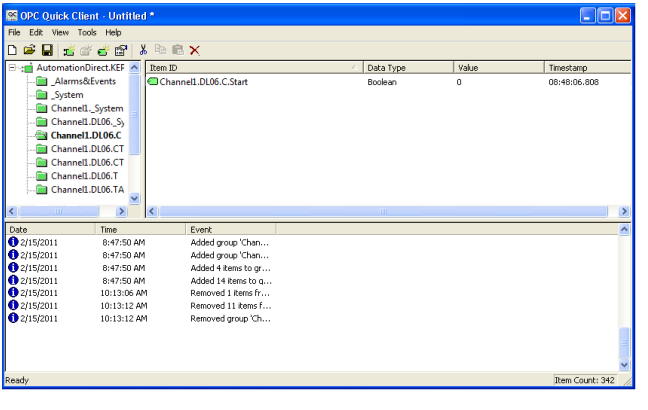
Panel 1, shown on the previous page, is the **Group Window**. This window manages a list of server connections along with group objects. Right clicking on any of the items in the list will display a pop-up menu of options. Double clicking on the item in the list will bring up the property sheet for that item.

Panel 2, the larger panel, is the **Item Window**. This window displays a list of items (tags) for the selected item in the Group Window. Right clicking on an item name in the list will display a pop-up menu of options. Double clicking on the item will display the property sheet for that item. Clicking on the Item ID column heading will cause the item list to be sorted based on Item ID. The column headings can be resized.

Panel 3 is the **Message Window**. This window logs status messages generated by the application. This window does not support sorting, but the column headers can be resized.

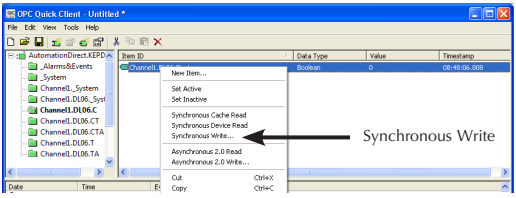
Step 2: Selecting a Tag to Test

A simple test of your setup is to read and write to an output or a data location in the serial device. For our example, the device is a ***Direct***LOGIC06 PLC. The DL06 tags have been previously imported (refer to the "Creating Tags" section for more information). Click on **Channel1.DL06.C** to open the tag(s) in the Item ID window as seen below.

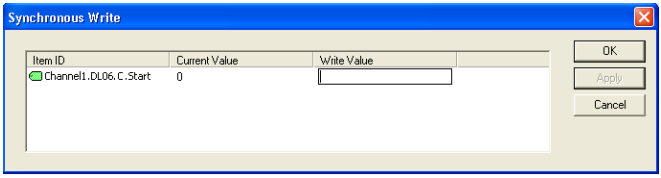


Step 3: Writing to a Tag

Right click on the tag to test and a drop down menu will be displayed.



From the drop down menu, select either **Synchronous Write** or **Asynchronous Write** by clicking on the text. Selecting **Synchronous Write** will open the window shown below.



The cursor will be blinking in the **Write Value** area of the window. Type the number "1" and click on the **Apply** button. The Write Value now displays a "1" reflecting the synchronous write that was just applied. The **Current Value** should also change to a "1". Follow the same procedure when writing a zero to turn off the tag.

Step 4: Repeat the Test for all Tags

To test any remaining tags within the device, repeat Step 3 and select each tag to be tested. Also, tags in other devices can be tested in the same manner.

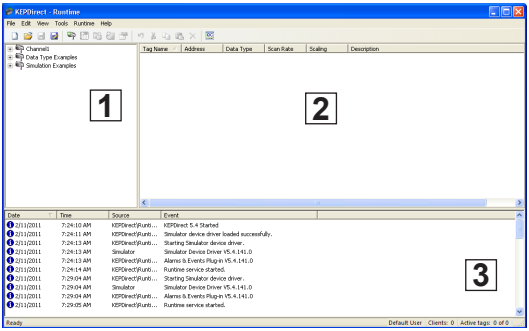
To learn about other ways to use the OPC Quick Client, refer to the **Help** file selection on the tool bar.

Create a New Project for ECOM Modules

Step 1: Launch KEP*Direct* and Get to Know the Work Areas

Click on the KEP***Direct*** OPC Server icon, shown on the left, to launch the application. The application will start and open the window shown below.

When the KEP***Direct*** OPC Server application is launched for the first time, it opens with an operational simulation program.



This is a sample program which can be used to practice adding devices and channels. The window is divided into three areas as explained below (see window above for reference):

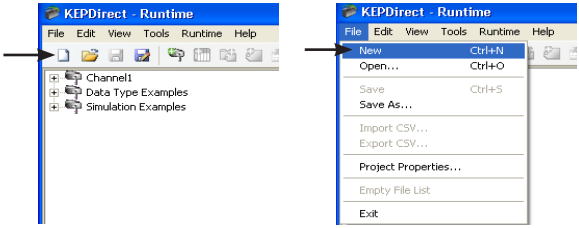
1. This area displays any existing channels, devices, and groups in a project. In addition, this area allows new channels, devices and groups to be created within a project.
2. This is the tag entry window of the server. Tags that you enter for a given device or tag group will be displayed here.
3. This area is the event log window of the server. Any of the messages generated by the server or the underlying driver will be displayed here.

Step 2: Things to confirm before Adding and Configuring a Channel

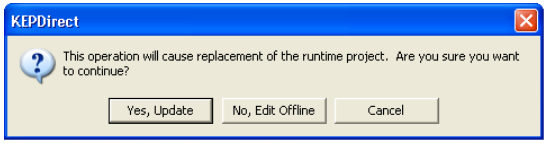
Before adding a new channel to your new project, first confirm that the following items have been completed or are available:

1. KEP***Direct*** OPC Server has been installed and launched properly.
2. Any devices to be added to your project must be installed with modules in place, powered up, and working properly.
3. The IP address of each ECOM module must be configured in order to establish a communications link to the server. Refer to the proper ECOM manual for instructions on how to set up IP addresses using NetEdit 3. Make note of each devices' IP address for they will be needed during the KEP***Direct*** OPC Server setup.
4. If you will be using more than one device to setup your project, an industrial Ethernet Switch such as the SE-SW5U from **AUTOMATIONDIRECT** is strongly recommended.
5. Ping each ECOM module from the Windows command prompt to verify the communications link.

Once all the above items are confirmed and ready, either click the **New Project** icon on the tool bar or click **File > New** from the menu bar to start your new project.



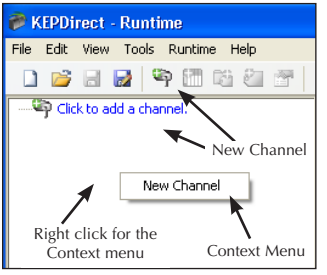
When you choose to open a new project the following window will appear asking you if you would like to replace the runtime project. The runtime project is currently a simulation that loaded on startup. Select **Yes, Update** to continue with a new project.



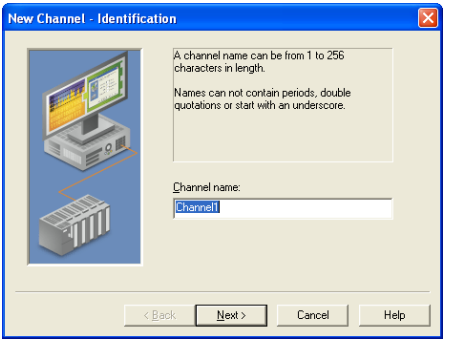
Step 3: Adding and Configuring a Channel

A channel refers to a specific communications driver. You can define a number of channels within a single project for organizational purposes. A channel acts as the basic building block of an OPC link. Each channel name must be unique in a KEP***Direct*** project. The channel name can be up to 256 alphanumeric characters in length (no spaces or special characters).

To add a new channel to your project, click on either the "**Click to add a channel"** text in the OPC Server window, the **New Channel** icon on the toolbar, **Edit > Devices > New Channel** from the menu bar or right click on the OPC Server window to select **New Channel** from the context menu.



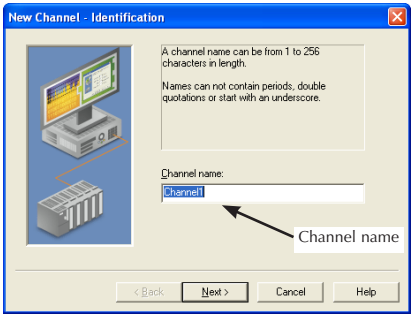
As soon as you click on either of the above, the **New Channel** window shown below will open.



Step 4: Name the Channel

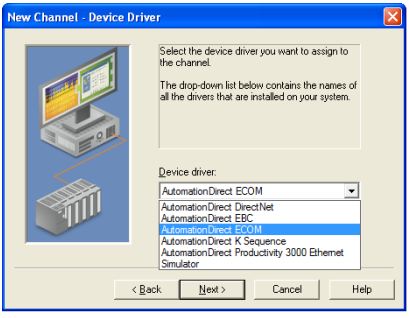
As previously mentioned, each channel name in a KEP***Direct*** project must be unique and is limited to 256 characters. While using long descriptive names is generally a good idea, keep in mind that some OPC client applications may have a limited display window when browsing the tag space of an OPC server. The channel name entered here will be part of the OPC browser information.

The **Identification** window prompts you to type a unique name for the channel by typing over the default **Channel1** label. Once the name has been typed, click on the **Next** button.



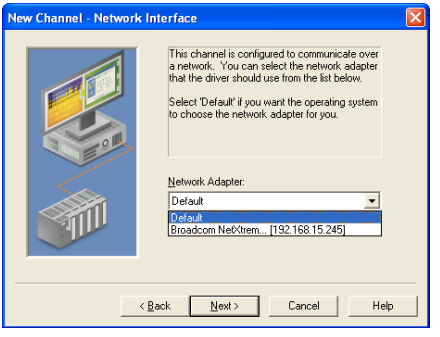
Step 5: Select the Device Driver

The **Device Driver** window will open. The arrow to the right side of the name provides a drop down list of device drivers that are available on your system. From this list, select **AUTOMATIONDIRECT ECOM.** After selecting the device driver, click on the **Next** button to go to the next step.



Step 6: Select the Network Interface

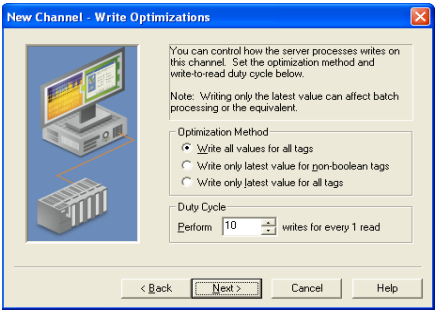
Next, the wizard will prompt you to select a "Network Adapter"and the **Network Interface** window will open.



Click on the down arrow to open the drop down list. Select the Ethernet **Network Adapter** that you would like to use to connect to the ECOM module. This list displays any detected network cards from your computer. Either select the adapter you wish to use or select **Default** and allow the operating system to select the adapter.

Once completed, click on the **Next** button to continue with the setup.

Step 7: Write Optimizations Setup



The **Write Optimizations** window will now open. This window provides three choices for an **Optimization Method**. These choices are described in the following list.

- **Write All Values for All Tags:** It forces the server to attempt to write every value to the controller. In this mode the server will continue to gather OPC write requests and add them to the server's internal write queue. The server will then process this write queue and attempt to empty the queue by writing data to the device as quickly as possible.

  This mode insures that everything written from your OPC client applications will be sent to the target device. This mode should be selected if the order of your write operations or the content of every write item must uniquely be seen at the target device.

- **Write Only Latest Value for Non-Boolean Tags:** Any value that is not a Boolean value will be updated in the server's internal write queue and will then be sent to the device at the next possible opportunity.

  This feature must be used with a clear understanding of how it will affect the operation of your application. This mode does not attempt to optimize writes to Boolean values. This allows you to optimize the operation of HMI data, such as a slide switch, without causing problems with Boolean operations like a momentary push button.

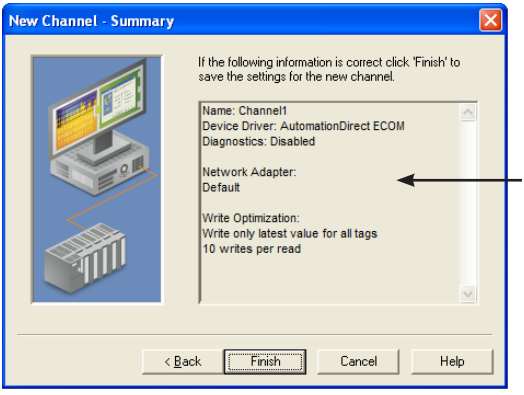
- **Write Only Latest Value for All Tags:** This is the default mode. The final write optimization mode takes the operation described for the second mode and applies it to all tags. If your application needs only to send the latest value to your device, this mode will optimize all writes by updating the tags currently in the write queue before they are sent.

The **Duty Cycle** selection allows you to control the ratio of write operations to read operations. The ratio is always based on one read for every one to ten writes. By default, the duty cycle is set to ten. This means that ten writes will occur for each read operation. If your application is doing a large number of continuous writes, but you need to insure that read data is still given time to process, you may want to reduce the duty cycle. A setting of one will result in one read operation for every write operation. In all cases, if there are no write operations to perform, reads will be processed continuously.

NOTE: *It is strongly recommended that you characterize your application for compatibility with these write optimization enhancements before using them in a production environment.*

Step 8: Review the Channel Setup Summary

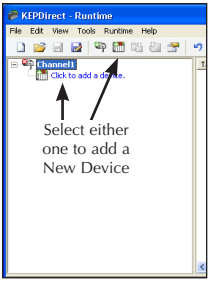
After setting up the write optimizations, click on the **Next** button. The process ends with a view of the new channel setup. Review the **Summary**, then click on **Finish** to complete the setup.



Step 9: Add a Device(s)

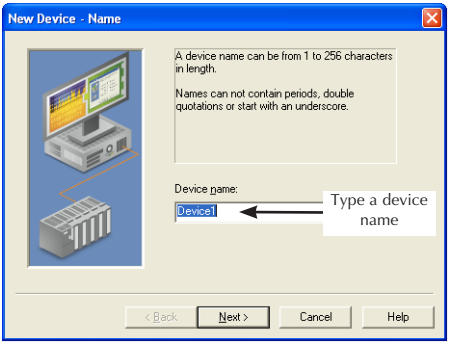
Once the channel setup is complete, the application window displays a directory tree on the screen showing the newly created channel name. A new device can now be added to the channel. Since we are using the **AUTOMATIONDIRECT ECOM** driver for this project, our devices will be ECOM modules and their associated PLC's

The quickest way to add a device is to select the "**Click to add device"** text found next to the device symbol as shown in the figure on the right. Other ways of adding a new device are to select **Edit > Devices > New Device** from the menu bar, or by selecting the **New Device** icon from the toolbar. Any of these methods will initiate the device setup and open the **New Device** window.



Step 10: Name the Device

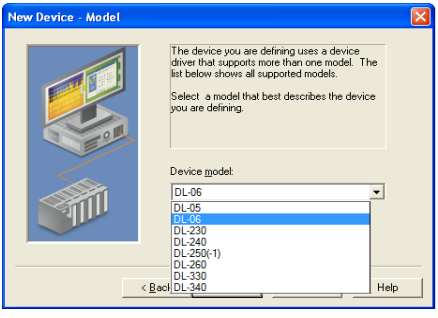
The **Name** window allows you to type a name from 1 to 256 characters long for your device . The default name is **Device1** as shown below.



Once your device name has been entered, click on the **Next** button to continue with the device installation.

Step 11: Select the Device Model

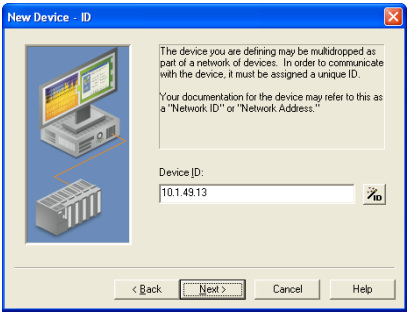
The next window which opens will be the **Model** window shown below. Click on the arrow to open the drop down list of the models available for setup. Select the model of the device which you are adding, i.e. DL06 PLC.



Once the device model is selected, click on the **Next** button to continue with the device installation.

Step 12: Type the Device ID

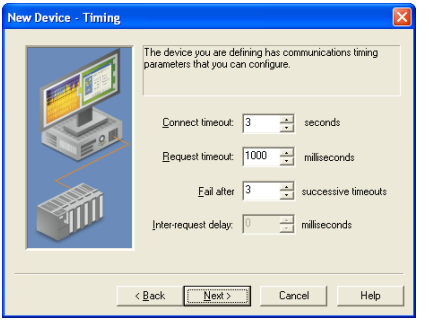
Since the ECOM driver is an Ethernet based driver, the device ID entered here is an IP address. The NetEdit 3 configuration software supplied can be used to configure the IP address of an ECOM module. For this driver, the device ID has the following format: YYY.YYY.YYY. YYY with each YYY byte in the range of 0–254.



Enter the IP address of the ECOM module you wish to monitor. It is important that the IP address is typed correctly and that it corresponds to the device being installed. Click on the **Next** button to continue with the device installation.

Step 13: Set the Device's Communication Timing Parameters

The **Timing** window shown below opens and displays the available communication settings.



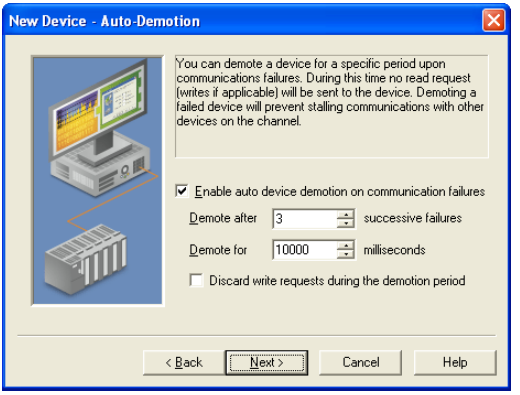
This screen provides three different communication settings for the device being installed. The following list explains the purpose of each available setting.

- **Connection Timeout**: 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. The default is typically 3 seconds but can vary depending on the specific nature of the chosen driver.
- **Request Timeout:** This is used to determine how long the driver will wait for a response from the target device. The request timeout has a valid range of 100 to 30000 milliseconds. The default is typically 1000 milliseconds but can vary depending on the specific nature of the chosen driver.
- **Fail After:** This parameter is used to determine how many times the driver will retry a communications request before considering the request to have failed. The valid range is 1 to 10 retries. The default is typically three retries but can vary depending on the specific nature of the driver. The number of retires configured for your application is dependent largely on your communications environment.

  If your environment is prone to noise induced communication failures you may want to increase the number of retries the driver performs. Keep in mind, however, that when the driver does encounter a communication issue, it will attempt to reacquire the data for the lost request. Based on the Request timeout and the Fail after count the driver will pause on a specific request until either the device responds or the timeout and retires have been exceeded. With this in mind you wouldn't want to set the timeout to 30000 milliseconds and 10 retires with the hope of covering every possible issue as this would result in a potential communications pause of 5 minutes.
- **Inter-Request Delay:** This parameter is not supported with Automation Direct products.

Once the three communication parameter settings have been selected, click on the **Next** button to continue with the device installation.

Step 14: Set the Auto-Demotion Parameters



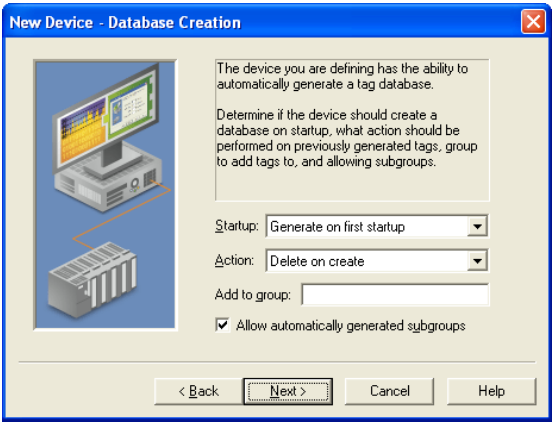
The **Auto-Demotion** parameters allow a driver to temporarily place a device off-scan in the event that a device is not responding. By placing a nonresponsive device offline for a specific time period, the driver can continue to optimize its communications with other devices on the same channel. After the time period has been reached, the driver will reattempt to communicate with the nonresponsive device. If the device is responsive, the device will be placed on-scan; otherwise, it will restart its off-scan time period.

Step 15: Set the Database Settings

The KEP***Direct*** I/O Server has a feature to automatically generate an OPC tag database. This has been designed to make the setup of your OPC application a Plug and Play operation. Communications drivers can be configured to automatically build a list of OPC tags within the server that correspond to device specific data. The automatically generated OPC tags can then be browsed from your OPC client. The OPC tags that are generated depend on the nature of the supporting driver.

If the target device supports its own local tag database, the ECOM driver can import the device's tag information in the form of a CSV file and then use this data to generate OPC tags within the server. Refer to the "Creating Tags" section of this manual for more information on CSV files.

The mode of operation for automatic tag database generation is completely configurable. The following **Database Creation** window, shown below, will open next. It allows you to configure how the server and the associated communications driver will handle automatic OPC tag database generation.



The following list explains the purpose of each available setting and group name.

- **Startup:** This selection allows you to configure when OPC tags will be automatically generated. There are three possible selections:
  - **Do not generate on startup:** This selection prevents the driver from adding any OPC tags to the tag space of the OPC Server. This is the default selection.
  - **Always generate on startup:** This selection causes the driver to always import the device's tag information and to add OPC tags to the tag space of the server each time the runtime file is initialized. The runtime file is initialized whenever a new or existing project is opened or by selecting **Runtime > Connect** or **Runtime > Reinitialize** from the toolbar.
  - **Generate on first startup:** This selection will cause the driver to import the target device's tag information the first time the runtime file is initialized and to add any OPC tags to the server tag space as needed.

NOTE: *When the Startup is selected, any tags that are added to the server's tag space must be saved with the project.*

When Startup (automatic tag generation) is enabled, the server needs to be told how to handle driver generated OPC tags and tags created by the user.

- **Action:** This selection allows you to control how the server will handle OPC tags that were automatically generated and currently exist in your project. This feature prevents automatically generated tags from piling up in the server.
  - **Delete on create:** This is the default condition which allows the server to remove any tags that had previously been added to the tag space before the communications driver can add any new tags.
  - **Overwrite as necessary:** This condition allows the server to remove only the tags that the communications driver is replacing with new tags. Any tags that are not being overwritten will remain in the server's tag space.
  - **Do not overwrite:** This condition prevents the server from removing any tags that had been previously generated or that may have already existed in the server. With this selection, the communications driver can only add completely new tags.
  - **Do not overwrite, log error:** This condition has the same effect as the previous condition but it also adds an error message that is posted to the OPC Server's event log when a tag overwrite would have occurred.

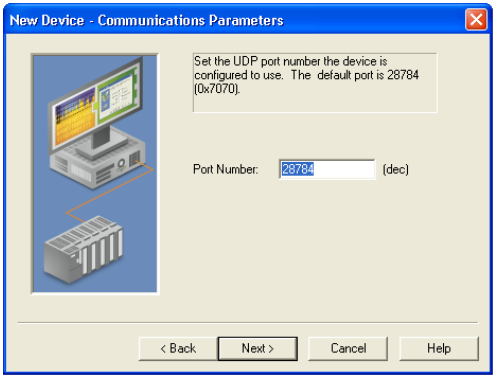
NOTE: *The removal of OPC tags affects tags that have been automatically generated by the communications driver and any tags you have added using names that match generated tags. It is recommended that you try to avoid adding your own tags to the server using names that match tags that may be automatically generated by the driver.*

- **Add to group**: This parameter can be used to aid in keeping automatically generated tags from mixing with tags that you may enter manually. With this parameter you may specify a sub group that will be used when adding all automatically generated tags for this device. The name of the sub group can be up to 256 characters in length.

Once the database settings are completed, click on the **Next** button to continue with the device installation.

Step 16: Communications Parameters

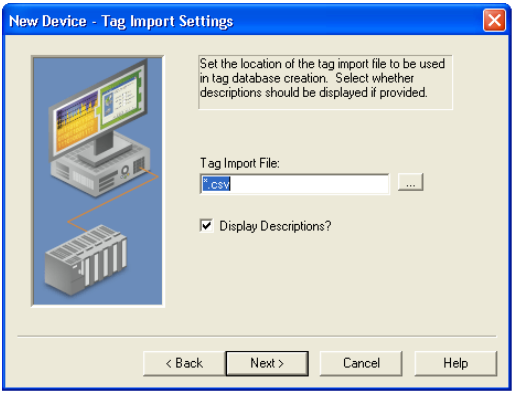
Next, the wizard will show the **Communications Parameters** dialog shown below.



This parameter is used to select the port number to be used by the driver for communicating with the remote device. The default value is 28784 (0x7070). Legal port number values are between 0-65535.

Step 17: Tag Import Settings

The next dialog the wizard presents is the **Tag Import Settings** window shown below.



Here you are to enter the exact location of the ***Direct***SOFT export file you want to import tags from (see the "Creating Tags" section for more information on creating CSV files). Click on the button next to the file entry window to select the file to be imported. You can also type the location and the file name if you wish.

Only two types of files can be imported:

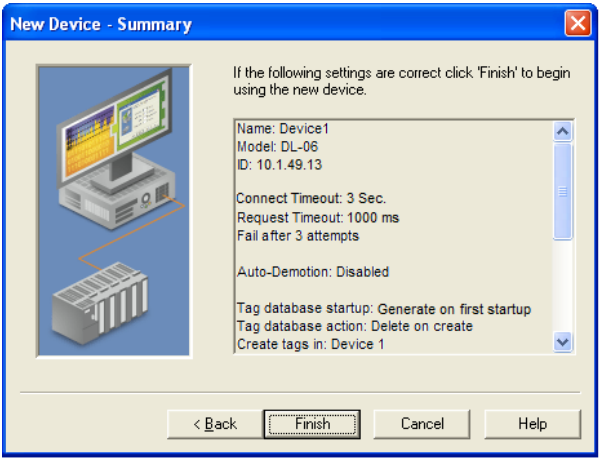
- Program (via export), .txt extension
- Element Documentation (via export), Standard Format, .csv extension

After selecting the import tag file, you have the option to have the tag descriptions imported or not. Check the **Display Descriptions?** box to have the descriptions imported. If necessary, a description will be given to tags with long names stating the original tag name.

Click the **Next** button to continue.

Step 17: Review the Device Setup Summary

The final window of the device setup is the **Summary** window shown below.



Review the summary and click on **Finish** to complete the device setup. If server's tag database was set to automatically generate, then the CSV file selected in the previous step will be used to create the database as soon as the device setup is completed. If the database creation was set to manual, refer to the "Creating Tags" section for information on adding tags to the project.

Step 18: Repeat These Steps for Additional Devices

Repeat these steps if more ECOM modules are to be added to the selected channel.

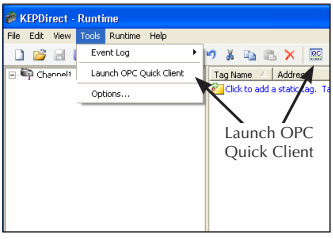
Using OPC Quick Client to Test an ECOM Module Setup

Once all of your channels, devices and tags have been created, your project is ready to launch the **OPC Quick Client**. Before launching the client, you will want to save your project to the disk. Do this by clicking on **File > Save As**. Name the project and save it.

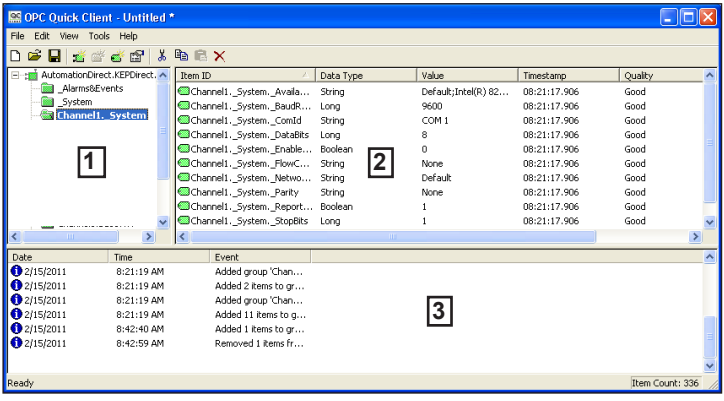
The following steps will show how to test an ECOM module with the **OPC Quick Client** application.

Step 1: Launch the OPC Quick Client

From the KEP***Direct*** OPC Server program window, either click on the **Tools** menu and select the "**Launch OPC Quick Client"** text or click on the **Launch OPC** icon on the tool bar shown in the figure below to launch the **OPC Quick Client** window.



This will open the **OPC Quick Client** window shown here.



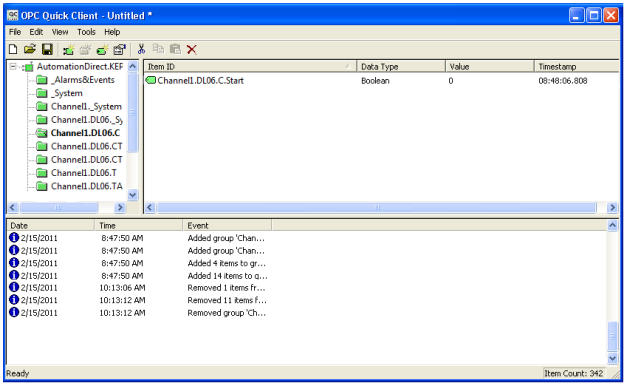
Panel 1, shown on the previous page, is the **Group Window**. This window manages a list of server connections along with group objects. Right clicking on any of the items in the list will display a pop-up menu of options. Double clicking on the item in the list will bring up the property sheet for that item.

Panel 2, the larger panel, is the **Item Window**. This window displays a list of items (tags) for the selected item in the Group Window. Right clicking on an item name in the list will display a pop-up menu of options. Double clicking on the item will display the property sheet for that item. Clicking on the Item ID column heading will cause the item list to be sorted based on Item ID. The column headings can be resized.

Panel 3 is the **Message Window**. This window logs status messages generated by the application. This window does not support sorting, but the column headers can be resized.

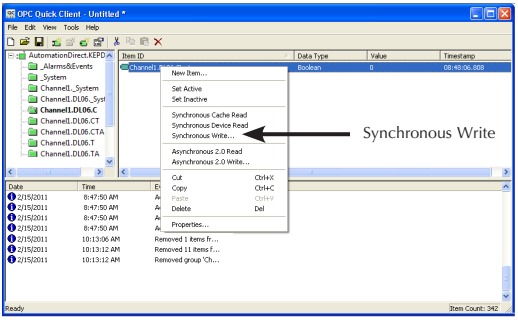
Step 2: Selecting a Tag to Test

A simple test of your setup is to read and write to an output or a data location in the ethernet device. For our example, the ECOM module is in the rack with a ***Direct***LOGIC06 PLC. The DL06 tags have been previously imported (refer to the "Creating Tags" section for more information). Click on **Channel1.DL06.C** to open the tag(s) in the Item ID window as seen below.

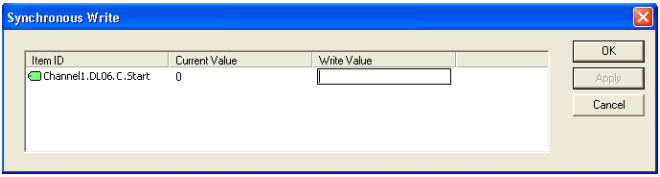


Step 3: Writing to a Tag

Right click on the tag to test and a drop down menu will be displayed.



From the drop down menu, select either **Synchronous Write** or **Asynchronous Write** by clicking on the text. Selecting **Synchronous Write** will open the window shown below.



The cursor will be blinking in the **Write Value** area of the window. Type the number "1" and click on the **Apply** button. The Write Value now displays a "1" reflecting the synchronous write that was just applied. The **Current Value** should also change to a "1". Follow the same procedure when writing a zero to turn off the tag.

Step 4: Repeat the Test for all Tags

To test any remaining tags within the device, repeat Step 3 and select each tag to be tested. Also, tags in other devices can be tested in the same manner.

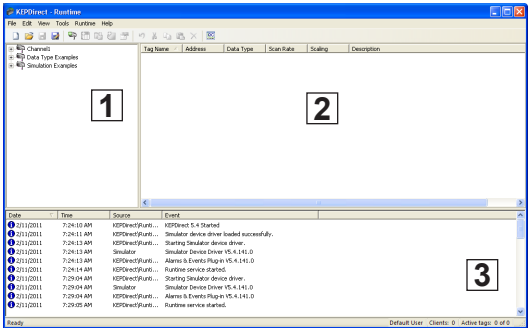
To learn about other ways to use the OPC Quick Client, refer to the **Help** file selection on the tool bar.

Create a New Project for EBC Modules

Step 1: Launch KEP*Direct* and Get to Know the Work Areas

Click on the KEP***Direct*** OPC Server icon, shown on the left, to launch the application. The application will start and open the window shown below.

When the KEP***Direct*** OPC Server application is launched for



the first time, it opens with an operational simulation program. This is a sample program which can be used to practice adding devices and channels. The window is divided into three areas as explained below (see window above for reference):

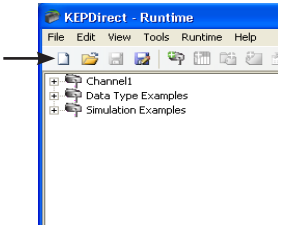
1. This area displays any existing channels, devices, and groups in a project. In addition, this area allows new channels, devices and groups to be created within a project.
2. This is the tag entry window of the server. Tags that you enter for a given device or tag group will be displayed here.
3. This area is the event log window of the server. Any of the messages generated by the server or the underlying driver will be displayed here.

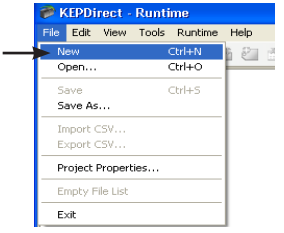
Step 2: Things to confirm before Adding and Configuring a Channel

Before adding a new channel to your new project, first confirm that the following items have been completed or are available:

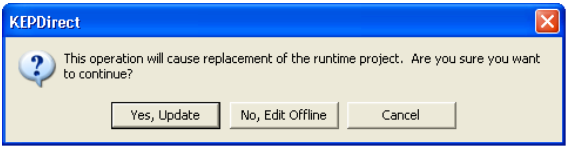
1. KEP***Direct*** OPC Server has been installed and launched properly.
2. Any devices to be added to your project must be installed with modules in place, powered up, and working properly.
3. The IP address of each EBC module must be configured in order to establish a communications link to the server. Refer to the proper EBC manual for instructions on how to set up IP addresses using NetEdit 3. Make note of each devices' IP address for they will be needed during the KEP***Direct*** OPC Server setup.
4. If you will be using more than one device to setup your project, an industrial Ethernet Switch such as the SE-SW5U from **AUTOMATIONDIRECT** is strongly recommended.
5. Ping each EBC module from the Windows command prompt to verify the communications link.

Once all the above items are confirmed and ready, either click the **New Project** icon on the tool bar or click **File > New** from the menu bar to start your new project.





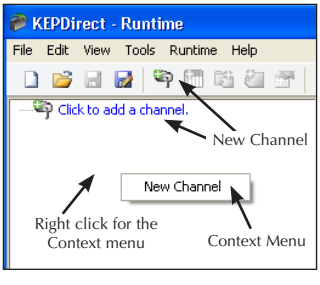
When you choose to open a new project the following window will appear asking you if you would like to replace the runtime project. The runtime project is currently a simulation that loaded on startup. Select **Yes, Update** to continue with a new project.



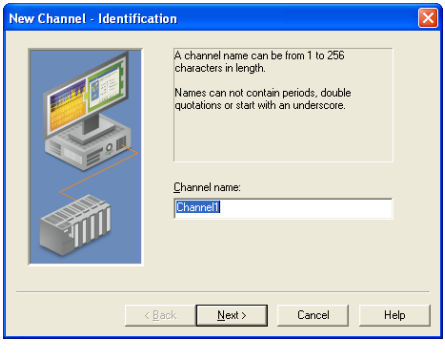
Step 3: Adding and Configuring a Channel

A channel refers to a specific communications driver. You can define a number of channels within a single project for organizational purposes. A channel acts as the basic building block of an OPC link. Each channel name must be unique in a KEP***Direct*** project. The channel name can be up to 256 alphanumeric characters in length (no spaces or special characters).

To add a new channel to your project, click on either the "**Click to add a channel"** text in the OPC Server window, the **New Channel** icon on the toolbar, **Edit > Devices > New Channel** from the menu bar or right click on the OPC Server window to select **New Channel** from the context menu.



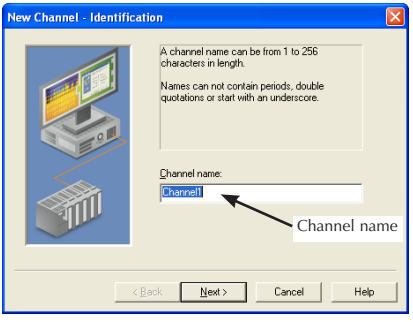
As soon as you click on either of the above, the New Channel window shown below will open.



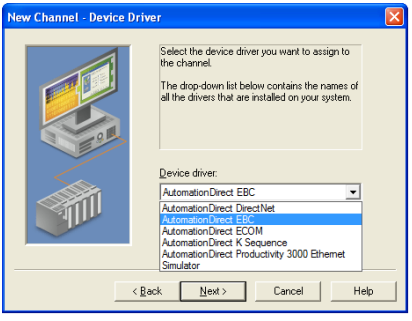
Step 4: Name the Channel

As previously mentioned, each channel name in a KEP***Direct*** project must be unique and is limited to 256 characters. While using long descriptive names is generally a good idea, keep in mind that some OPC client applications may have a limited display window when browsing the tag space of an OPC server. The channel name entered here will be part of the OPC browser information.

The **Identification** window prompts you to type a unique name for the channel by typing over the default **Channel1** label. Once the name has been typed, click on the **Next** button.



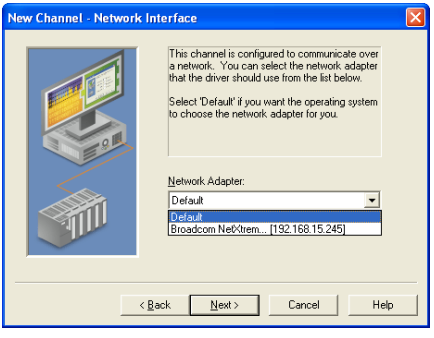
Step 5: Select the Device Driver



The **Device Driver** window will open. The arrow to the right side of the name provides a drop down list of device drivers that are available on your system. From this list, select **AUTOMATIONDIRECT EBC.** After selecting the device driver, click on the **Next** button to go to the next step.

Step 6: Select the Network Interface

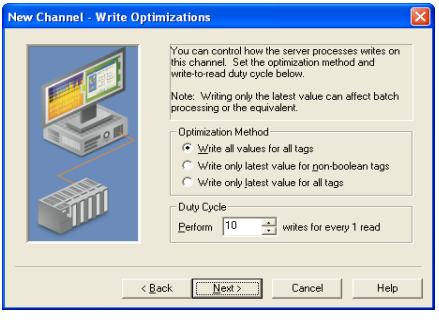
Next, the wizard will prompt you to select a "Network Adapter"and the **Network Interface** window will open.



Click on the down arrow to open the drop down list. Select the Ethernet **Network Adapter** that you would like to use to connect to the EBC module. This list displays any detected network cards from your computer. Either select the adapter you wish to use or select **Default** and allow the operating system to select the adapter.

Once completed, click on the **Next** button to continue with the setup.

Step 7: Write Optimizations Setup



The **Write Optimizations** window will now open. This window provides three choices for an **Optimization Method**. These choices are described in the following list.

- **Write All Values for All Tags:** It forces the server to attempt to write every value to the controller. In this mode the server will continue to gather OPC write requests and add them to the server's internal write queue. The server will then process this write queue and attempt to empty the queue by writing data to the device as quickly as possible.

  This mode insures that everything written from your OPC client applications will be sent to the target device. This mode should be selected if the order of your write operations or the content of every write item must uniquely be seen at the target device.

- **Write Only Latest Value for Non-Boolean Tags:** Any value that is not a Boolean value will be updated in the server's internal write queue and will then be sent to the device at the next possible opportunity.

  This feature must be used with a clear understanding of how it will affect the operation of your application. This mode does not attempt to optimize writes to Boolean values. This allows you to optimize the operation of HMI data, such as a slide switch, without causing problems with Boolean operations like a momentary push button.

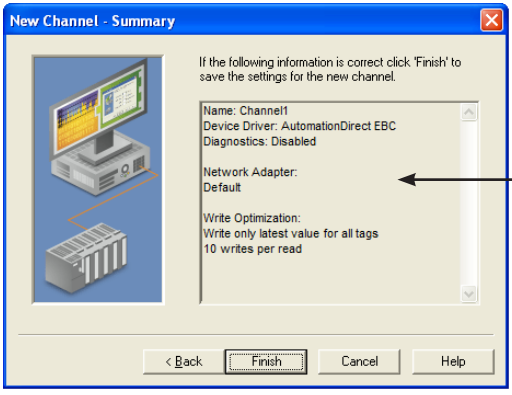
- **Write Only Latest Value for All Tags:** This is the default mode. The final write optimization mode takes the operation described for the second mode and applies it to all tags. If your application needs only to send the latest value to your device, this mode will optimize all writes by updating the tags currently in the write queue before they are sent.

The **Duty Cycle** selection allows you to control the ratio of write operations to read operations. The ratio is always based on one read for every one to ten writes. By default, the duty cycle is set to ten. This means that ten writes will occur for each read operation. If your application is doing a large number of continuous writes, but you need to insure that read data is still given time to process, you may want to reduce the duty cycle. A setting of one will result in one read operation for every write operation. In all cases, if there are no write operations to perform, reads will be processed continuously.

NOTE: *It is strongly recommended that you characterize your application for compatibility with these write optimization enhancements before using them in a production environment.*

Step 8: Review the Channel Setup Summary

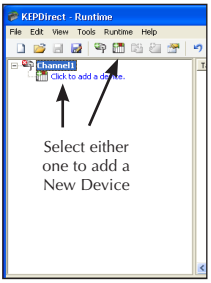
After setting up the write optimizations, click on the **Next** button. The process ends with a view of the new channel setup. Review the **Summary**, then click on **Finish** to complete the setup.



Step 9: Add a Device(s)

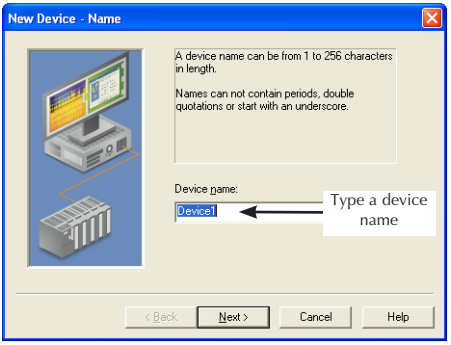
Once the channel setup is complete, the application window displays a directory tree on the screen showing the newly created channel name. A new device can now be added to the channel. Since we are using the **AUTOMATIONDIRECT EBC** driver for this project, our devices will be GS-EDRV modules or EBC modules and their associated I/O.

The quickest way to add a device is to select the "**Click to add device"** text found next to the device symbol as shown in the figure on the right. Other ways of adding a new device are to select **Edit > Devices > New Device** from the menu bar, or by selecting the **New Device** icon from the toolbar. Any of these methods will initiate the device setup and open the **New Device** window.



Step 10: Name the Device

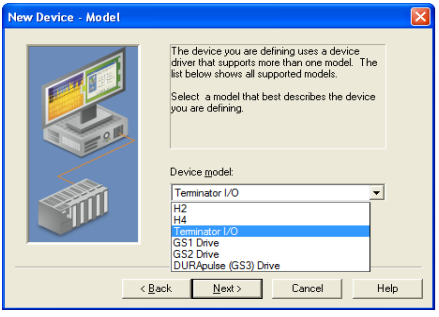
The **Name** window allows you to type a name from 1 to 256 characters long for your device . The default name is **Device1** as shown below.



Once your device name has been entered, click on the **Next** button to continue with the device installation.

Step 11: Select the Device Model

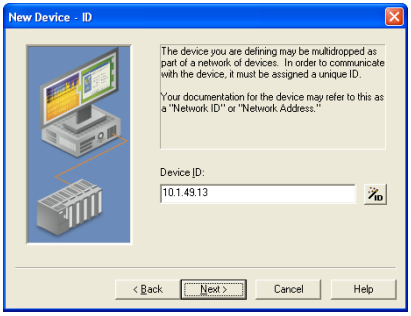
The next window which opens will be the **Model** window shown below. Click on the arrow to open the drop down list of the models available for setup. Select the model of the device which you are adding, i.e. Terminator I/O.



Once the device model is selected, click on the **Next** button to continue with the device installation.

Step 12: Type the Device ID

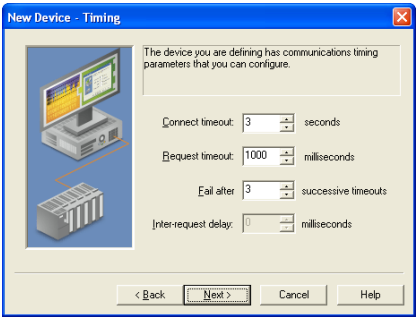
Since the EBC driver is an Ethernet based driver, the device ID entered here is an IP address. The NetEdit 3 configuration software supplied can be used to configure the IP address of an EBC module. For this driver, the device ID has the following format: YYY.YYY.YYY.YYY with each YYY byte in the range of 0–254.



Enter the IP address of the EBC module you wish to monitor. It is important that the IP address is typed correctly and that it corresponds to the device being installed. Click on the **Next** button to continue with the device installation.

Step 13: Set the Device's Communication Timing Parameters

The **Timing** window shown below opens and displays the available communication settings.



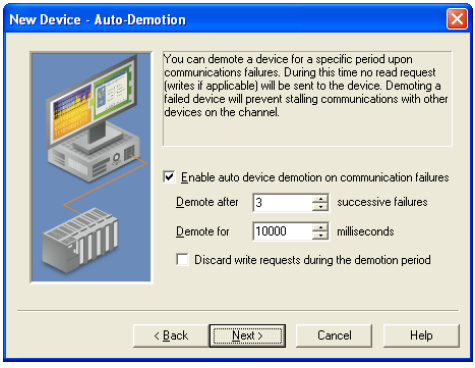
This screen provides three different communication settings for the device being installed. The following list explains the purpose of each available setting.

- **Connection Timeout**: 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. The default is typically 3 seconds but can vary depending on the specific nature of the chosen driver.
- **Request Timeout:** This is used to determine how long the driver will wait for a response from the target device. The request timeout has a valid range of 100 to 30000 milliseconds. The default is typically 1000 milliseconds but can vary depending on the specific nature of the chosen driver.
- **Fail After:** This parameter is used to determine how many times the driver will retry a communications request before considering the request to have failed. The valid range is 1 to 10 retries. The default is typically three retries but can vary depending on the specific nature of the driver. The number of retires configured for your application is dependent largely on your communications environment.

  If your environment is prone to noise induced communication failures you may want to increase the number of retries the driver performs. Keep in mind, however, that when the driver does encounter a communication issue, it will attempt to reacquire the data for the lost request. Based on the Request timeout and the Fail after count the driver will pause on a specific request until either the device responds or the timeout and retires have been exceeded. With this in mind you wouldn't want to set the timeout to 30000 milliseconds and 10 retires with the hope of covering every possible issue as this would result in a potential communications pause of 5 minutes.
- **Inter-Request Delay:** This parameter is not supported with Automation Direct products.

Once the three communication parameter settings have been selected, click on the **Next** button to continue with the device installation.

Step 14: Set the Auto-Demotion Parameters



The **Auto-Demotion** parameters allow a driver to temporarily place a device off-scan in the event that a device is not responding. By placing a nonresponsive device offline for a specific time period, the driver can continue to optimize its communications with other devices on the same channel. After the time period has been reached, the driver will reattempt to communicate with the nonresponsive device. If the device is responsive, the device will be placed on-scan; otherwise, it will restart its off-scan time period.

Step 15: Set the Database Settings

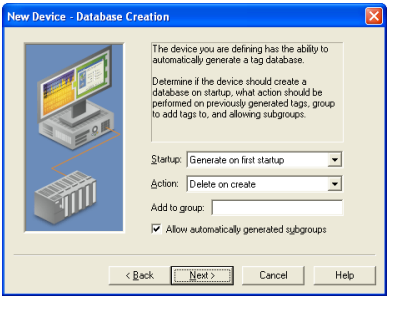
The KEP***Direct*** I/O Server has a feature to automatically generate an OPC tag database. This has been designed to make the setup of your OPC application a Plug and Play operation. Communications drivers can be configured to automatically build a list of OPC tags within the server that correspond to device specific data. The automatically generated OPC tags can then be browsed from your OPC client. The OPC tags that are generated depend on the nature of the supporting driver.

In the case of an Ethernet I/O system that supports detection of its own available I/O module types, like the EBC, the driver will automatically generate OPC tags in the server that are based on the types of I/O modules plugged into the Ethernet I/O rack.

NOTE: *For Terminator analog modules, the EBC driver will generate both bit level and word level tags for each analog channel. Many of these tags are duplicates and may not be needed for your particular application.*

NOTE: *The H2-EBC(-F) and Terminator I/O EBCs automatically detect I/O modules in the base on power up. The H4-EBC(-F), however, only detects discrete modules. Use NetEdit 3 to manually configure the H4-EBC(-F) base when using analog modules.*

The mode of operation for automatic tag database generation is completely configurable. The following **Database Creation** window, shown below, will open next. It allows you to configure how the server and the associated communications driver will handle automatic OPC tag database generation.



The following list explains the purpose of each available setting and group name.

- **Startup:** This selection allows you to configure when OPC tags will be automatically generated. There are three possible selections:
  - **Do not generate on startup:** This selection prevents the driver from adding any OPC tags to the tag space of the OPC Server. This is the default selection.
  - **Always generate on startup:** This selection causes the driver to always import the device's tag information and to add OPC tags to the tag space of the server each time the runtime file is initialized. The runtime file is initialized whenever a new or existing project is opened or by selecting **Runtime > Connect** or **Runtime > Reinitialize** from the toolbar.
  - **Generate on first startup:** This selection will cause the driver to import the target device's tag information the first time the runtime file is initialized and to add any OPC tags to the server tag space as needed.

NOTE: *When the Startup is selected, any tags that are added to the server's tag space must be saved with the project.*

When Startup (automatic tag generation) is enabled, the server needs to be told how to handle driver generated OPC tags and tags created by the user.

- **Action:** This selection allows you to control how the server will handle OPC tags that were automatically generated and currently exist in your project. This feature prevents automatically generated tags from piling up in the server. This would occur in an Ethernet I/O system if you continued to change the I/O modules in the rack with the server configured to always generate new OPC tags on startup. Under this condition, every time the communications driver detected a new I/O module, the tags would be added to the server. If the old tags are not removed, a number of unused tags could accumulate in the server's tag space.
  - **Delete on create:** This is the default condition which allows the server to remove any tags that had previously been added to the tag space before the communications driver can add any new tags.
  - **Overwrite as necessary:** This condition allows the server to remove only the tags that the communications driver is replacing with new tags. Any tags that are not being overwritten will remain in the server's tag space.
  - **Do not overwrite:** This condition prevents the server from removing any tags that had been previously generated or that may have already existed in the server. With this selection, the communications driver can only add completely new tags.
  - **Do not overwrite, log error:** This condition has the same effect as the previous condition but it also adds an error message that is posted to the OPC Server's event log when a tag overwrite would have occurred.

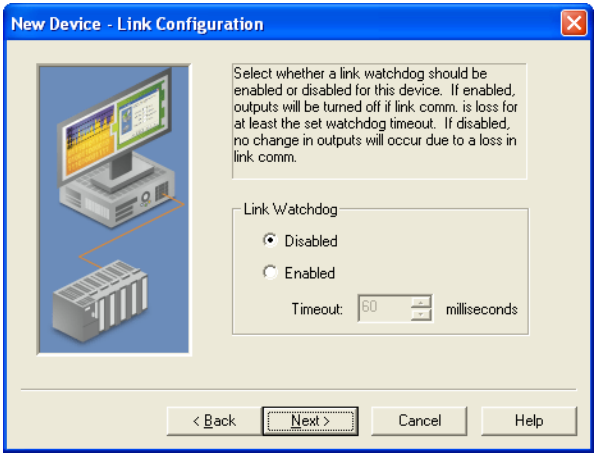
NOTE: *The removal of OPC tags affects tags that have been automatically generated by the communications driver and any tags you have added using names that match generated tags. It is recommended that you try to avoid adding your own tags to the server using names that match tags that may be automatically generated by the driver.*

- **Add to group**: This parameter can be used to aid in keeping automatically generated tags from mixing with tags that you may enter manually. With this parameter you may specify a sub group that will be used when adding all automatically generated tags for this device. The name of the sub group can be up to 256 characters in length.

Once the database settings are completed, click on the **Next** button to continue with the device installation.

Step 16: Set the Link Configuration

The next window is the **Link Configuration** window shown below. This window offers the options to disable or enable the Link Watchdog.



The Link Watchdog monitors the vital communications link between the PC and the I/O device. It is possible for the communications link to "break" during its use. A "break" in the link causes a loss of communication between devices which can be permanent or temporary. An example of this would be a physical break in the network, i.e., a severed cable. On the other hand, a lack of communication would be a significant time interval between exchanges that is temporary. This interval would have to exceed a set timeout before being considered a loss of communication.

For the link watchdog, sometimes referred to as a link monitor, there are two options:

- **Disable:** When the link watchdog is disabled, a "break" in the communication link has no effect on the device's I/O outputs. The outputs will maintain the state they were assigned prior to the "break" until communication is restored and outputs are altered via the client application.

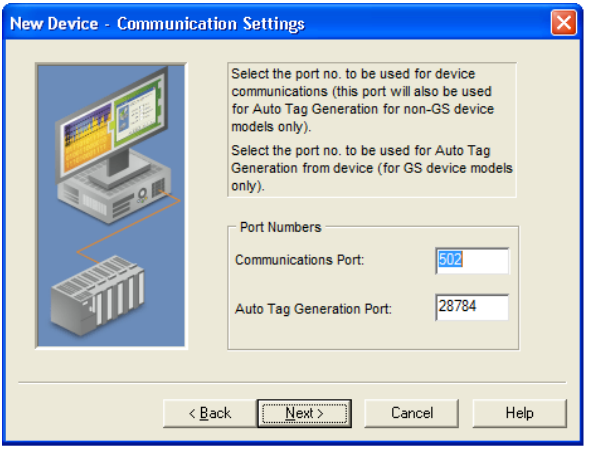
- **Enable:** When link watchdog is enabled, it will continuously monitor the link. If a "break" in the link occurs, a timer is initiated. When the timer reaches the Watchdog Timeout value, all device I/O outputs will be turned off.

NOTE: *The Link Configuration options will be disabled for GS device modules.*

Once the link configuration is completed, click on the **Next** button to continue with the device installation.

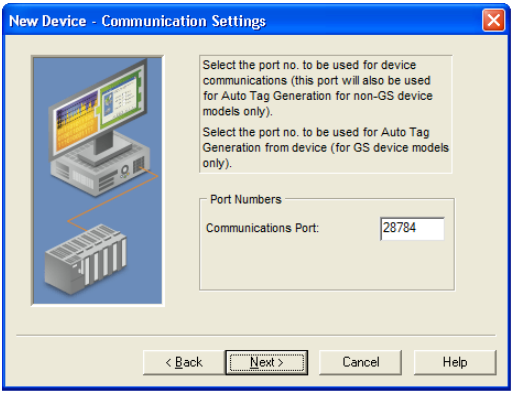
Step 17: Communication Settings

Next, the wizard will show the one of the following **Communication Settings** dialogs. If the EBC driver is monitoring a GS series drive, the following window will appear.



- The Communications Port parameter is the port number to be used by the driver for communicating with the remote device. The default value for GS device models (GS1, GS2 and GS3) is 502. Legal port number values are between 0-65535.
- The Auto Tag Generation Port is the port number to be used by the driver for performing automatic tag database generation from the remote device. The default value is 28784 (0x7070) and legal port number values are between 0-65535.

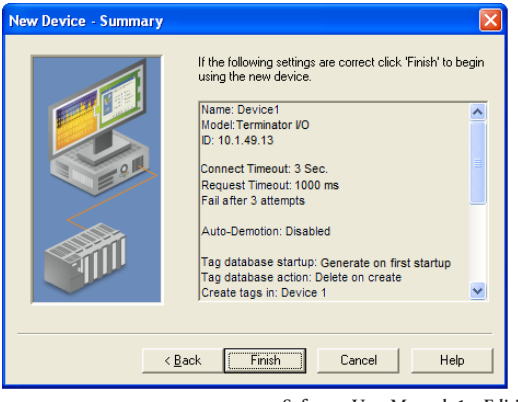
If the EBC driver is not monitoring a GS series drive, then the following dialog will appear.



- The Communications Port parameter is the port number to be used by the driver for communicating with the remote device. This port will also be used for performing automatic tag database generation. The default value is 28784 (0x7070). Legal port number values are between 0-65535.

Step 18: Review the Device Setup Summary

The final window of the device setup is the **Summary** window shown below.



Review the summary and click on **Finish** to complete the device setup. If server's tag database was set to automatically generate, then the server will attempt to connect to the device and create a tag database using the device's configuration. If the database creation was set to manual, refer to the "Creating Tags" section for information on adding tags to the project.

Step 19: Repeat These Steps for Additional Devices

Repeat these steps if more EBC modules are to be added to the selected channel.

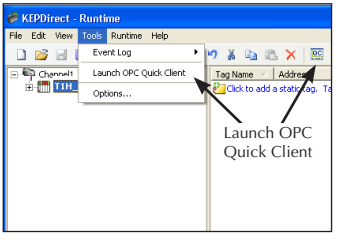
Using OPC Quick Client to Test an EBC Module Setup

Once all of your channels, devices and tags have been created, your project is ready to launch the **OPC Quick Client**. Before launching the client, you will want to save your project to the disk. Do this by clicking on **File > Save As**. Name the project and save it.

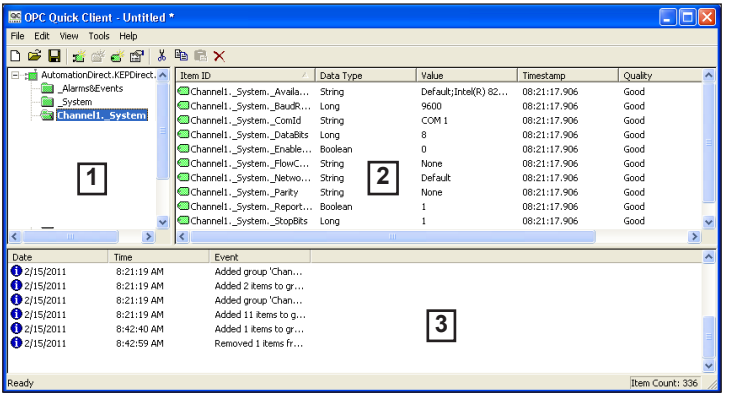
The following steps will show how to test an EBC module with the **OPC Quick Client** application.

Step 1: Launch the OPC Quick Client

From the KEP***Direct*** OPC Server program window, either click on the **Tools** menu and select the "**Launch OPC Quick Client"** text or click on the **Launch OPC** icon on the tool bar shown in the figure below to launch the **OPC Quick Client** window.



This will open the **OPC Quick Client** window shown here.

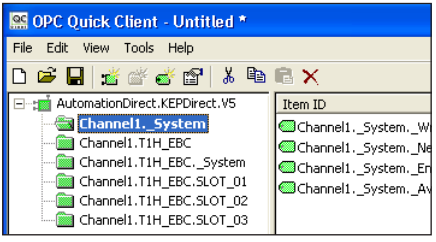


Panel 1, shown on the previous page, is the **Group Window**. This window manages a list of server connections along with group objects. Right clicking on any of the items in the list will display a pop-up menu of options. Double clicking on the item in the list will bring up the property sheet for that item.

Panel 2, the larger panel, is the **Item Window**. This window displays a list of items (tags) for the selected item in the Group Window. Right clicking on an item name in the list will display a pop-up menu of options. Double clicking on the item will display the property sheet for that item. Clicking on the Item ID column heading will cause the item list to be sorted based on Item ID. The column headings can be resized.

Panel 3 is the **Message Window**. This window logs status messages generated by the application. This window does not support sorting, but the column headers can be resized.

Looking at the sample below, the group window shows that a device is connected to **Channel1** and the device name is **T1H_EBC**. Also that the EBC device has three input/output modules indicated as **SLOT_01, SLOT_02** and **SLOT_03**.

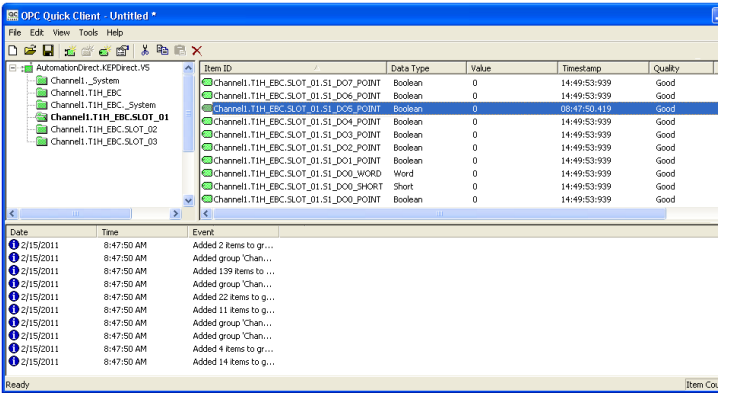


Step 2: Selecting a Tag to Test

From the available slots, select a slot to test. For our sample we are selecting **Slot 01** as shown in the figure on the following page.

When the slot is selected, the panel on the right displays the available input/output tags for the device. The tag for output point 5 is chosen and it is shown highlighted.

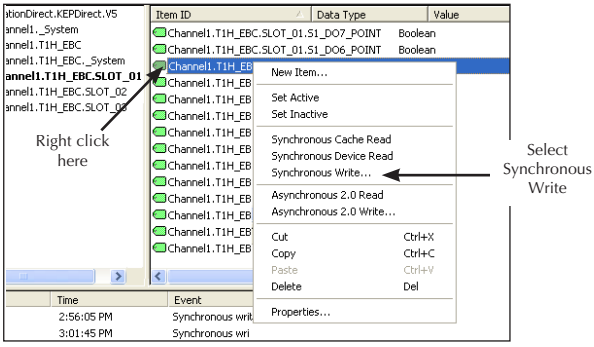
NOTE: *KEPDirect maps I/O for each slot in several different ways. A 16 point discrete input module will have points 0-15 mapped individually. These same points will also be mapped together in Word, DWord and Byte formats. Select the format that you need and disregard the others.*



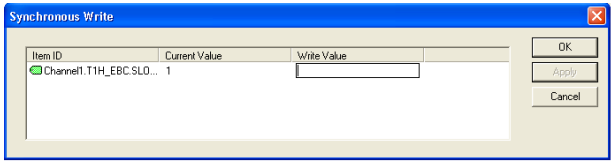
- **T1H_EBC:** Device description connected to Channel 1.
- **Slot_01:** Identifies the location of the I/O module
- **S1_DO05_POINT:** Identifies the connection point or terminal number. On this label, **DO** stands for **Discrete Output** and the **05** indicates point **5** of the I/O module. If the terminal is an input point, the **DO** would be a **DI** representing a **Discrete Input**.
- **Boolean:** Shows the data type used. When selecting outputs to test, ensure that the selected output is a Boolean data type.
- **0:** This is the present value of the output. When the value is **0**, the output is off (deactivated).

Step 3: Writing to a Tag

Right click on the tag to test and a drop down menu will be displayed.



From the drop down menu, select either **Synchronous Write** or **Asynchronous Write** by clicking on the text. Selecting **Synchronous Write** opens the **Synchronous Write** window shown below.



The cursor will be blinking in the **Write Value** area of the window. Type the number "1" and click on the **Apply** button. The Write Value now displays a "1" reflecting the synchronous write that was just applied. The **Current Value** should also change to a "1". Follow the same procedure when writing a zero to turn off the output tag.

Step 4: Repeat the Test for all Tags

To test any remaining tags within the device, repeat Step 3 and select each tag to be tested. Also, tags in other devices can be tested in the same manner.

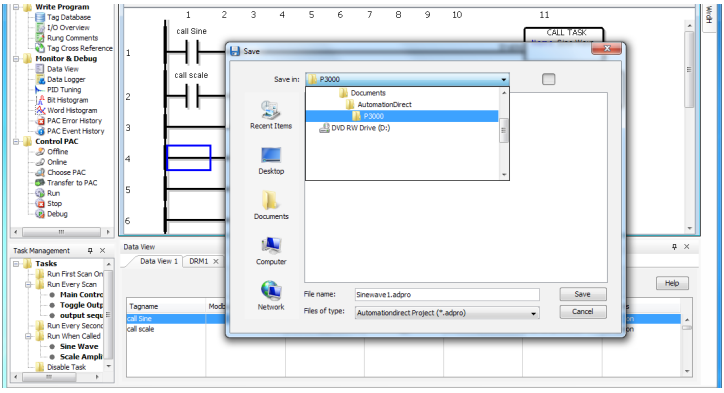
To learn about other ways to use the OPC Quick Client, refer to the **Help** file selection on the tool bar.

Create a New Project for Productivity3000

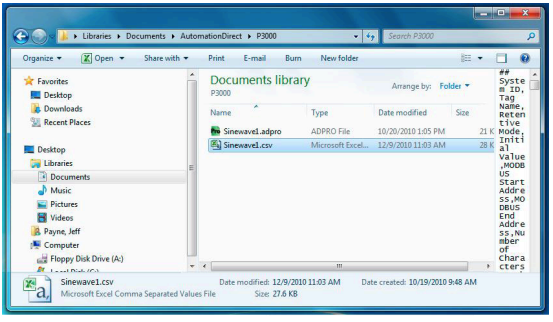
This is a general example of how to setup KEPDirect OPC Server to communicate with a Productivity3000 over Ethernet. A more detailed analysis of the setup options can be found in the ***Direct***LOGIC setup section.

Step 1: Create and save your Productivity3000 project

Note the location of the saved file. The default directory for your file is C:\My Documents\AutomationDirect\ …



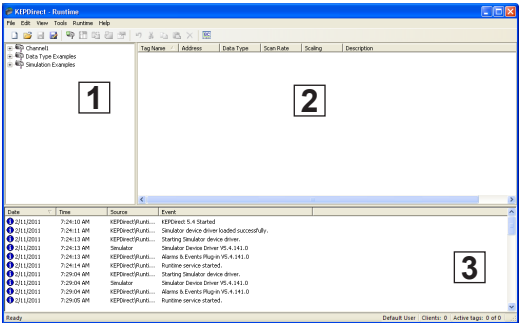
Each time you save your Productivity3000 project file, a CSV file is created with the same name and is located in the same directory as your project file. This is your Tag database file.



Step 2: Launch KEP*Direct* and Get to Know the Work Areas

Click on the KEP***Direct*** OPC Server icon, shown on the left, to launch the application. The application will start and open the window shown below.

When the KEP***Direct*** OPC Server application is launched for the first time, it opens with an operational simulation program.



This is a sample program which can be used to practice adding devices and channels. The window is divided into three areas as explained below (see window above for reference):

1. This area displays any existing channels, devices, and groups in a project. In addition, this area allows new channels, devices and groups to be created within a project.
2. This is the tag entry window of the server. Tags that you enter for a given device or tag group will be displayed here.
3. This area is the event log window of the server. Any of the messages generated by the server or the underlying driver will be displayed here.

Step 3: Things to confirm before Adding and Configuring a Channel

Before adding a new channel to your new project, first confirm that the following items have been completed or are available:

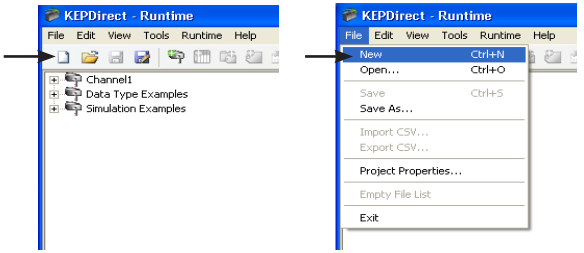
1. KEP***Direct*** OPC Server has been installed and launched properly.
2. Any devices to be added to your project must be installed with modules in place, powered up, and working properly.

3. **Ethernet:** You must have the correct IP addresses for each of the devices to be used in your project. Please refer to the proper user manuals in order to setup the IP addresses for your devices. Once all of the IP addresses have been configured, verify each communications link by pinging the P3-550's using the Windows command prompt. Make note of each devices' IP address for they will be needed during the KEP***Direct*** OPC Server setup.

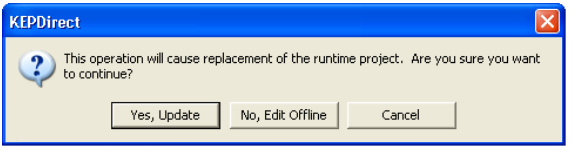
   Serial: In order to communicate serially between KEP***Direct*** OPC Server and a device, the device's serial port must be established. Please refer to the proper user manual in order to setup the serial communications port.

4. If you will be using more than one Ethernet device to setup your project, an industrial Ethernet Switch such as the SE-SW5U from **AUTOMATIONDIRECT** is strongly recommended.

Once all the above items are confirmed and ready, either click the **New Project** icon on the tool bar or click **File > New** from the menu bar to start your new project.



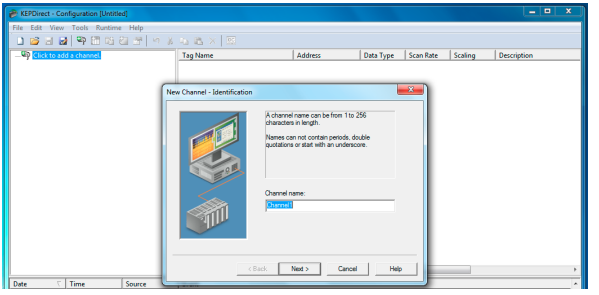
When you choose to open a new project the following window will appear asking you if you would like to replace the runtime project. The runtime project is currently a simulation that loaded on startup. Select



Yes, Update to continue with a new project.

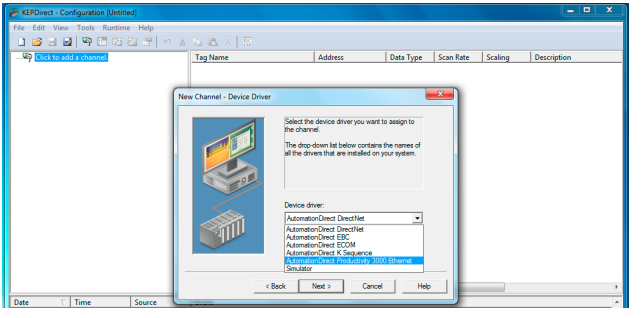
Step 4: Adding and Configuring a Channel

The Channel defines the communication media that will be used to communicate with the PLC such as Serial or Ethernet. In this example we will be using Ethernet. Select the Edit menu to add a New Channel to the server or click where it says "Click to add a channel". When you select New Channel, it will open a wizard to walk you through setting up the channel. The first step in the wizard is giving the channel a unique name. The name can be any alpha/numeric string up to 256 characters. Do not use special characters. In this example we will leave it at default, and click "Next".



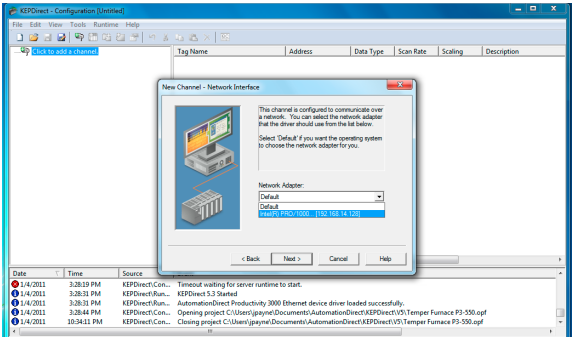
Step 5: Select the Device Driver

Since we are using Ethernet to connect to the Productivity3000, we will select the AutomationDirect Productivity3000 Ethernet as the Device Driver and click "Next".



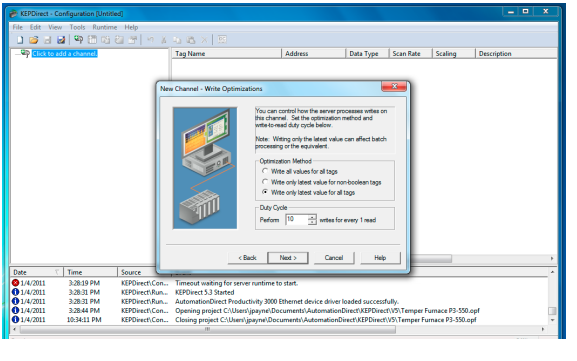
Step 6: Select the Network Interface

Since we are using Ethernet, you will have to select the Network Adapter that the Productivity3000 CPU (P3-550) is connected to. If you only have one Network Adapter in your computer, you can select Default. If you have multiple network cards in your PC, select the one that the P3-550 is connected to. You can determine this by contacting your Network Administrator. Select the correct adapter and click "Next".



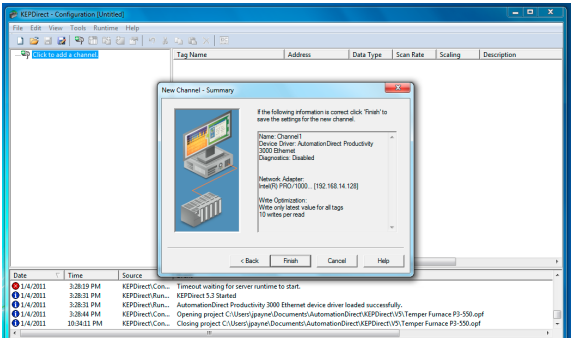
Step 7: Write Optimizations Setup

The Write Optimization allows the user to define how the software processes read and write commands from the client software. These parameters are application dependent; in most cases the default settings are sufficient. Click "Next" to continue.



Step 8: Review the Channel Setup Summary

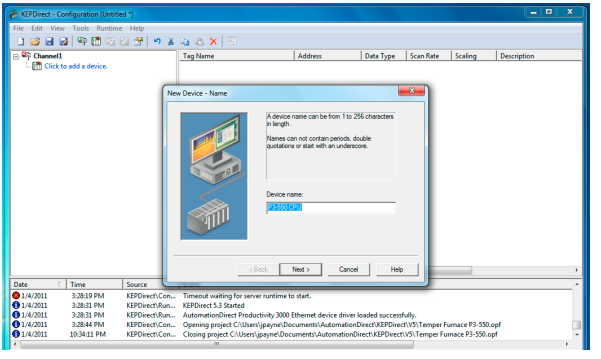
The **Summary** screen allows the user to review all the settings before it is applied to the server. If all the settings are OK, then select "Finish" to apply the settings to the Server.



Step 9: Add a Device(s)

From the Edit menu, select New Device or click where it says "Click to add a device".

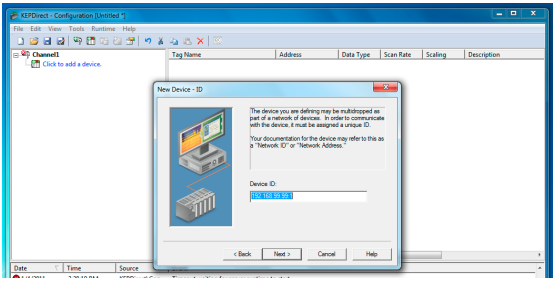
When you select New Device, it will open a wizard to walk you through setting up the device. The first step in the wizard is giving the device a unique name. The name can be any alpha/numeric string up to 256 characters. Do not use special characters. In this example we will leave it at default, and click "Next".



Step 10: Type the Device ID

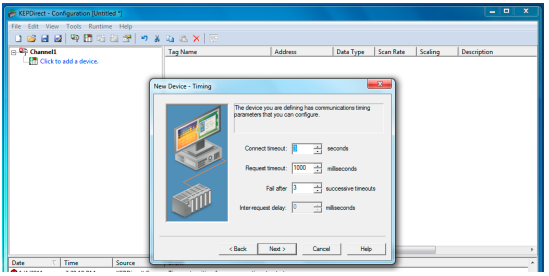
The Device ID specifies the IP address and node number that you assign the P3-550. The Device ID has the following format: YYY.YYY. YYY.YYY:1 with each YYY byte in the range of 0–254. The ':1' is needed to make the Device ID valid. Enter the IP address then click "Next".

NOTE: *The wizard will not continue with the setup until you have entered a valid IP address with a node number extension.*



Step 11: Set the Device's Communication Timing Parameters

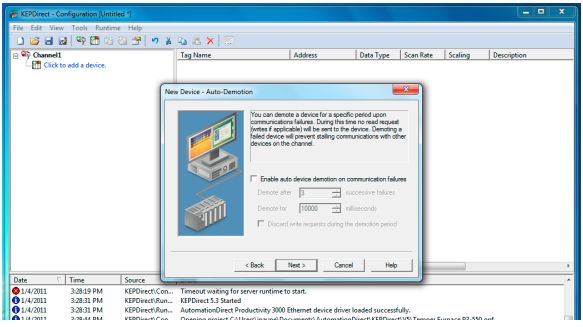
The Communication Parameters are used to set the server Time Out parameters for connection failures. The defaults are sufficient in most



applications, but you may need to adjust them if you are going through gateways, routers or bridges. Adjust as necessary and click "Next".

Step 12: Set the Auto-Demotion Parameters

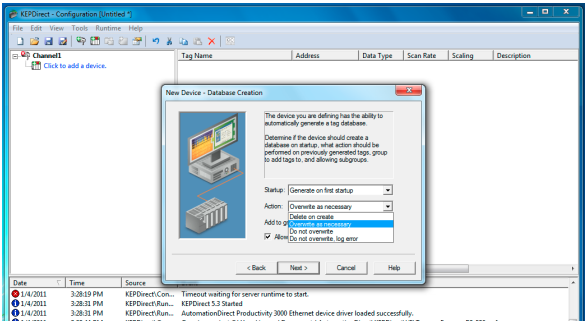
Auto-Demotion gives you the option to place less significant devices off scan that are not responding. By doing this for a specific amount of time, the driver can continue to communicate with other devices on the same channel. For initial system testing this option is not necessary. Leave the option unchecked as default or make the necessary changes and click "Next".



Step 13: Set the Database Settings

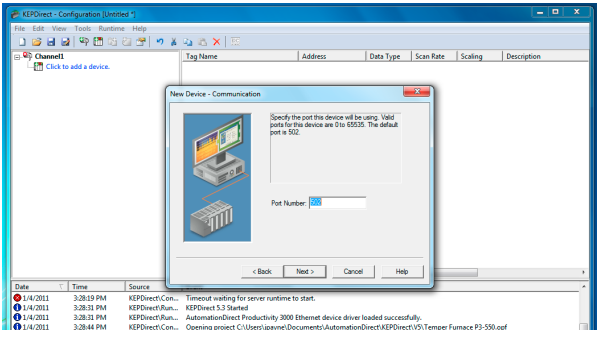
The Productivity3000 controller has its own local tag database; therefore, the device driver will use the tag names found in the P3-550 CSV file to build the server's OPC tags. Refer to the "Creating Tags" section of this manual for more information on CSV files.

These settings are used to configure when the OPC tags will be generated and what action to take on the OPC tag database. Configure your preferred option and click "Next".



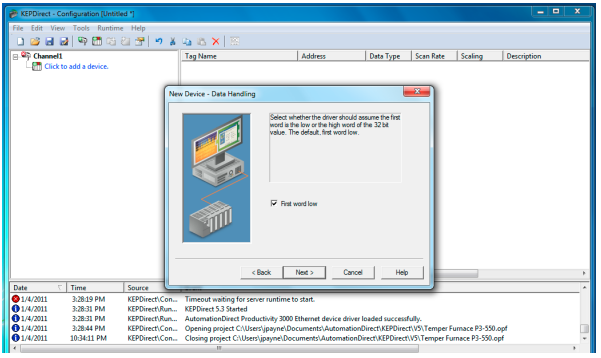
Step 14: Set the Port Number

This option is used to specify the port number for the device. The default port is 502 and unless specifically changed in the P3-550 does not need to be changed in the OPC server. Leave as default or adjust as necessary and click “Next”.



Step 15: Set the Data Handling Options

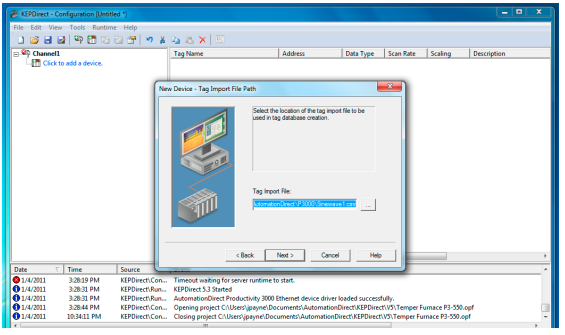
By default the driver assumes the first register used to construct a 32 bit value is the low word. This is correct for typical communications to the P3-550. Leave the default option checked and click “Next”.



Step 16: Tag Import Settings

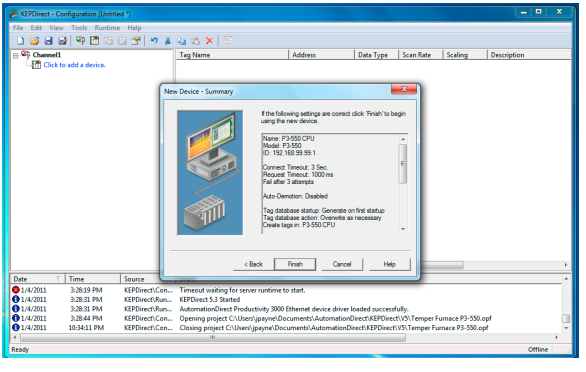
This is an important step for connecting to your Productivity3000.

The Tag Import File Path directs the OPC server to your P3-550 project tag database file which will be used to create your OPC tags. Click the button with the "…" to browse for the CSV file created by the Productivity3000 project. Remember, by default this is located along with your project file in the C:\My Documents\AutomationDirect directory.



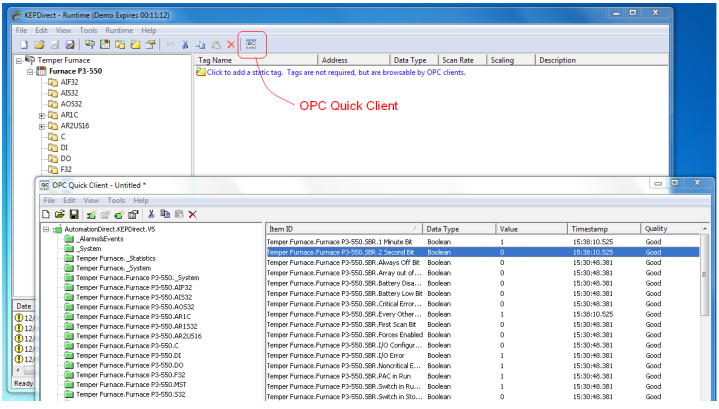
Step 17: Review the Device Setup Summary

The **Summary** screen allows the user to review all the settings before it is applied to the server. If all the settings are OK, then select "Finish" to apply the settings to the Server.



Using OPC Quick Client to Test a Productivity3000 Setup

If you configured the system to automatically generate the tag database on startup, you should now see a folder list of all data types under your device. If the tag database wasn't set to automatically generate, refer to the "Creating Tags" section for information on manually creating tags. We can now check your connection with the OPC Quick Client. Click the OPC Quick Client icon on the toolbar. This will open the OPC Quick Client in another window as seen below.



There should be a pre-populated folder list for all data types on the left. They will be listed by the Channel & Device name you gave your configuration.

Click on the channel_name.Device_name.SBR" folder. This will give you a list of the System Bit Read tags in the right window. If you have created the server correctly, you will see the 2 Second Bit toggle and the Tag Quality will be "Good".

NOTE: *The update rate of the tags in the OPC client are based on sample rates and server configuration. These values may update slower than the controller tags.*

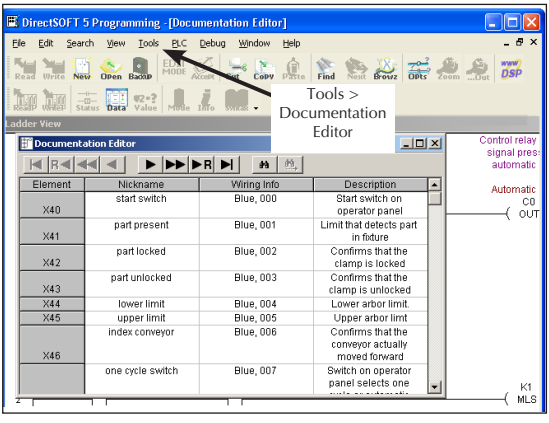
If all goes as planned you can start to develop your client application after this. The OPC Quick Client is a quick and easy way to verify your controller link before you start to develop your client application.

Creating Tags

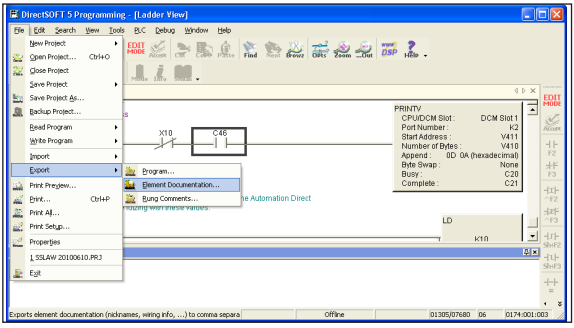
Importing CSV Files

This server supports the import and export of tag data in a CSV (comma separated value) file. The CSV functions are only available when a Device or Tag Group is selected. When using CSV import and export, tags can be created quickly in the application of your choice. Creating a CSV file from your ***Direct***SOFT or ProductivitySuite project is the best way to import the element nicknames and descriptions to your KEPDirect project. Importing the nicknames and devices is done in two parts. First, is to create an export file from ***Direct***SOFT or ProductivitySuite, then to generate the tag database in the OPC server. As mentioned in the previous section, ProductivitySuite creates a CSV file for your Productivity3000 project each time it is saved. That CSV file can be found in the same directory as the project file. ***Direct***LOGIC CSV files are not created automatically and the following steps will demonstrate how to create the file.

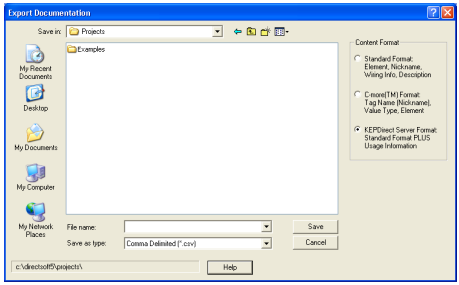
Open your ***Direct***SOFT project that contains the tags (elements) you wish to export to the OPC server. The ***Direct***SOFT project should have all the elements documented, if not, open the Documentation Editor by selecting **Tools,** then click on **Documentation Editor**. To add elements, enter the nickname and description for each element of interest.



Next, click on **File** to open the drop-down menu, then select **Export > Element Documentation...** to open the **Export Documentation** dialog.



The Export Documentation dialog, shown below, will appear for you to select the location and file where you want to save the exported CSV file. Select **Comma Delimited (*csv)** for **Save as type** and either **Standard**



Format or **KepDirect Server Format**, then click on **Save**.

Your CSV file will look similar to the diagram on the following page.

Microsoft Excel - Tag example.csv

File Edit View Insert Format Tools Data Window Help Acrobat

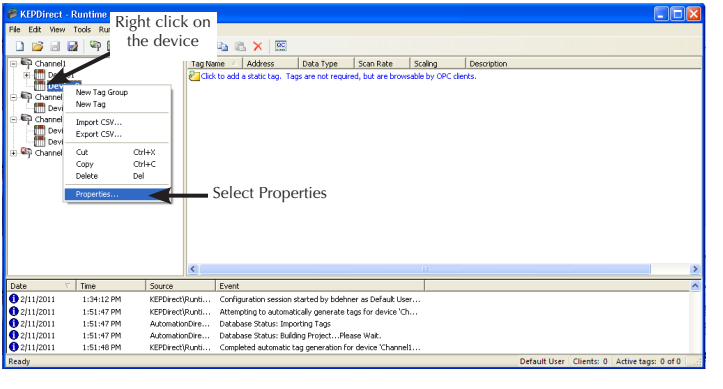
A29 =

	A	B	C	D	E	F	G	H	I	J
1	X40	start switc	Blue, 000	Start switch on operator panel						
2	X41	part preser	Blue, 001	Limit that detects part in fixture						
3	X42	part locke	Blue, 002	Confirms that the clamp is locked						
4	X43	part unlock	Blue, 003	Confirms that the clamp is unlocked						
5	X44	lower limit	Blue, 004	Lower arbor limit.						
6	X45	upper limit	Blue, 005	Upper arbor limt						
7	X46	index conv	Blue, 006	Confirms that the conveyor actually moved forward						
8	X47	one cycle	Blue, 007	Switch on operator panel selects one cycle or automatic operation						
9	X50	stop switc	Blue, 008	Stop Switch on operator panel						
10	Y40	clamp	Red, 000	Clamp to hold part in place						
11	Y41	arbor dowr	Red, 001	Output for downard movement of the arbor						
12	Y42	conveyor	Red, 002	Motor starter for conveyor motor						
13	C0	Automatic mode		Control relay used to signal press is in automatic mode						
14	C1	press complete		Control relay showing that the part has been pressed						
15	C2	release clamp		Control relay that controls the releasing of the fixture clamp						
16	C3	1 cycle mode		Control relay used to signal that press is in one cycle mode						
17	T0	Conveyor delay		Delay timer for conveyor						
18	CT0	Part Counter		Self resetting parts counter to count number of parts made						
19										
20										
21										
22										

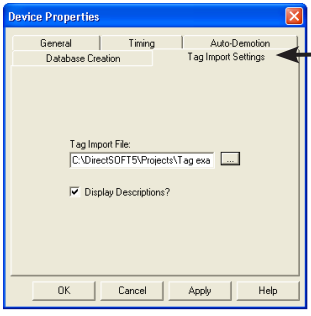
Now that you have the ***Direct***SOFT nicknames and descriptions in a CSV file, they can be imported to your OPC Server project.

The following steps will demonstrate how to import a ***Direct***SOFT or ProductivitySuite CSV file into your KEP***Direct*** project.

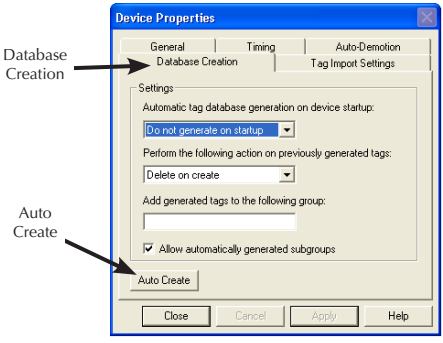
First, if no devices have been created in your KEP***Direct*** project, then this CSV file will be needed when setting up the tag import settings of the device. If your project has devices already configured, then start by opening the **Device Properties** window from the KEP***Direct*** Server main window. This is done by right clicking on the device and selecting **Properties**.



With the **Device Properties** window open, select the **Tag Import Settings** tab. Browse to the location of the previously exported CSV file. Click **Apply** once the file has been selected.



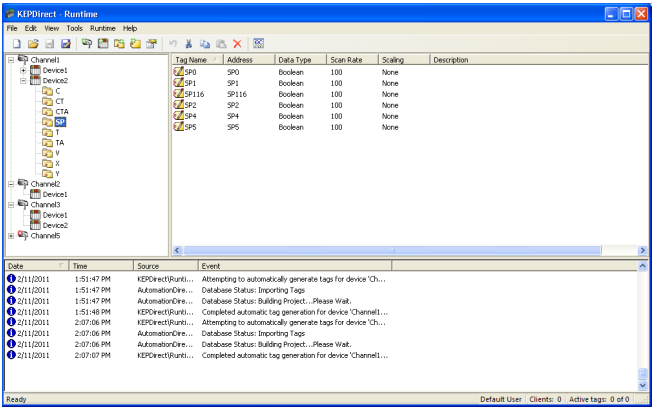
Next, select the **Database Creation** tab. Click **Auto Create**. Then click **Close.**



NOTE: *The database can be generated in either a group or in the device.*

The OPC Server will attempt to create the tag database while posting messages to the event log on the status of the import. When finished, all elements exported out of ***Direct***SOFT or ProductivitySuite will appear in the OPC Server in the main window layout.

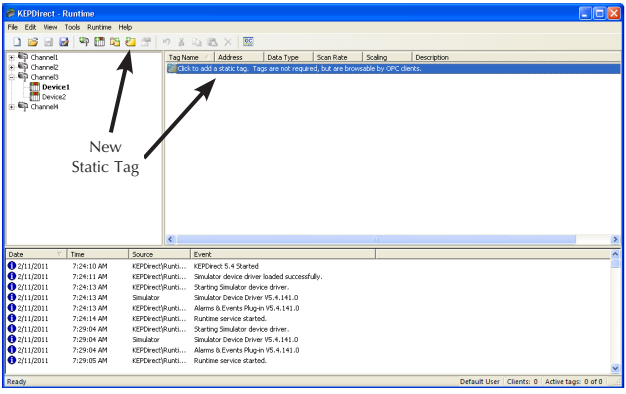
The KEP***Direct*** Server window should look like the diagram below. The OPC tags generated are given meaningful names in the OPC Server and are based on the variables imported. These tags are also placed in meaningful tag groups to provide a structured and manageable interface to the tags. The end result is a well organized OPC server project that directly reflects the variable import file.



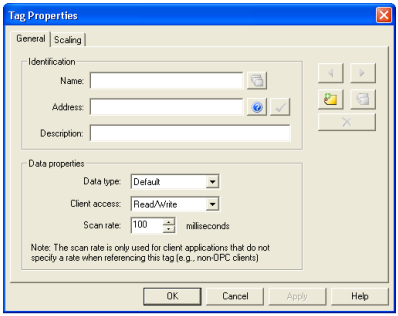
Static Tags

Besides CSV importing, the server allows user defined static tags. User defined static tags are created in the server and benefit the user by allowing the tag to be browsed from OPC clients that support tag browsing. The user defined tags also support tag scaling.

To add a static tag to your project, either click on the text **Click to add a static tag** or click on the **New Tag** tool on the tool bar.

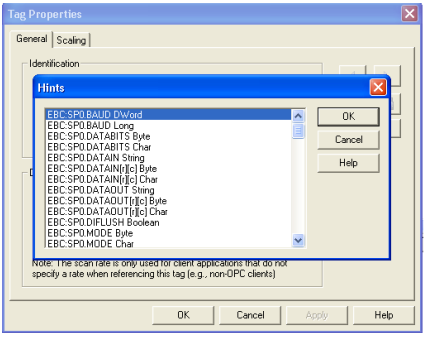


The **Tag Properties** dialog, shown below, will appear for you to enter the **Name, Address** and **Description** for the tag being entered.



The **Name** parameter allows you to enter the string that will represent the data available from this tag. The tag name can be up to 256 characters in length. While using long descriptive names is generally a good idea, keep in mind that some OPC client applications may have a limited display window when browsing the tag space of an OPC server. The tag name is part of the OPC browse data. Tag names must be unique within a given device branch or tag group branch. If your application is best suited for using blocks of tags with the same names, you can use **tag groups** to segregate the tags.

The **Address** parameter allows you to enter the desired driver address for this tag. The format of the address entered here is based entirely upon the driver being used. To determine how an address should be entered you can use the "Hints" button to open a pop-up window with a list of addresses and ranges for the particular device that you are using.



Hints provide a quick reference guide to the address format of the driver. The primary driver Help can also be invoked from the hints dialog if needed. The address entered can be up to 128 characters in length. Once you have entered an address, you can test it using the check address button. When pressed, the check address button attempts to validate the address with the driver. If the driver accepts the address as entered no message will be displayed. If an error is detected, a pop-up window will inform you of the error. Keep in mind that some errors will be related to the data type selection and not the address string.

DL05/06, DL105, DL205 Series, D3-350 and DL405 Series General Addressing - Data Types				
	Boolean		Word	
Address Type	Address Form	Bit Number (Octal)	Address Form	Word Number (Octal)
Input Points (READ ONLY)	X<xxx>	xxx	V<xxxxx>	xxxxx
Output Points	Y<xxx>			
Control Relays	C<xxx>			
Special Relays	SP<xxx>			
Timer Status Bits	T<xxx>			
Counter Status Bits	CT<xxx>			
Stages	S<xxx>			
Timer Current Values	None	None		
Counter Current Values				
Data Words				
Data Words Non-Volatile				
System Parameters				

NOTE: Bit Access to V-memory

Bit information can be directly accessed within V-memory registers. To access a bit within a V-memory register, a bit number can be appended to any V-memory address. V-memory addressing with bit access would appear as follows: V<xxxxx>.<yy> where xxxxx is the V-memory location and y is the bit number (0 to 15) within that register. If the V-memory location is either a Long or DWord, the bit number yy can be (0 to 31).

Examples

V40401	Bits 20-27 (octal) of X Input
V41100	Timer status bits 0-17 (octal)
CT165	Counter contact 165
S57	Stage control bit 57
V2000.1	Bit access to V2000 bit 1
V2000.30@Long	Bit access to V2000 as a Long bit 30

The optional **Description** parameter allows you to attach a comment to this tag. A string of up to 64 characters can be entered for the description. If you are using an OPC client that supports Data Access 2.0 Tag Properties, the description parameter will be accessible from the Item Description property of the tag.

The **Data Type** selection allows you to specify the format of this tag's data as it is found in the physical device. In most cases, this is also the format of the data as it is returned to the client. The data type setting is an important part of how a communication driver reads and writes data to a device. For many drivers, the data type of a particular piece of data is rigidly fixed. In these cases, the driver knows what format it needs to use when reading data from the device. In some cases, however, the interpretation of device data is largely in the user's hands. An example would be a device that uses 16 bit data registers. Normally this would indicate that the data is either a Short or a Word. Many register-based devices also support values that span two registers. In these cases, the double register values could be a Long, Dword, or Float. When the driver you are using supports this level of flexibility, you must tell the driver how you want to read data for this tag. By selecting the appropriate data type you are telling the driver to read either one register or two, or possibly a Boolean value. The driver governs the data format you choose. You can access the driver's help system through the Hints button to get specific help on what data types are available for a given driver. Below is a chart of available data type selections.

Data Types Description		
Type	**Description**	**Prefix**
Boolean	Single bit	X, Y, C, SP, T, CT, IO
Word	Unsigned 16 bit value	V
Short	Signed 16 bit value	V
DWord	Unsigned 32 bit value	V
Long	Signed 32 bit value	V
Float	32 bit Real value IEEE format	V
Double	64 bit Real value IEEE format	V
String	Null terminated ASCII string	V
BCD	Two byte packed BCD value,range is 0-9999	V
LBCD	Four byte packed BCD value, range is 0-99999999	V
Byte	Unsigned 8 bit value	R
Char	Signed 8 bit value	R

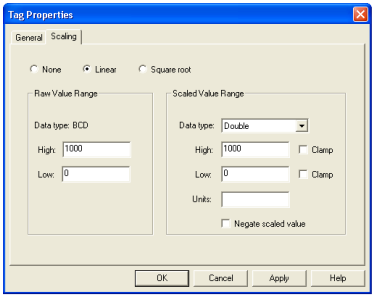
NOTE: *Reading a HEX value with BCD format results in an invalid number.*

The **Client access** selection allows you to specify whether this tag is **Read Only** or **Read/Write**. By selecting **Read Only**, you can prevent client applications from changing the data contained in this tag. By selecting **Read/Write**, you are allowing client applications to change this tag's value as needed. The **Client access** selection also has an effect upon how this tag will appear in the browse space of an OPC client. Many OPC client applications allow you to filter tags based on their attributes. Changing the access method of this tag may change how and when the tag will appear in the browse space of your OPC client.

The **Scan rate** parameter allows you to specify the update interval for this tag when used with a non-OPC client. OPC clients can control the rate at which data is scanned by using the update rate that is part of all OPC groups. Normally, non-OPC clients don't have that luxury. The server allows you to specify an update rate on a tag per tag basis for non-OPC clients. By using the scan rate, you can tailor the bandwidth requirements of the server to suit the needs of your application. If, for example, you need to read data that changes very slowly, there is no reason to read the value very often. By using the scan rate, this tag can be forced to read at a slower rate reducing the demand on the communications channel. The valid range is 10 to 99999990ms, with 10ms increments. The default is 100 milliseconds.

Scaling Static Tags

Scaling allows raw data from your device to be scaled to a more appropriate range for your application.



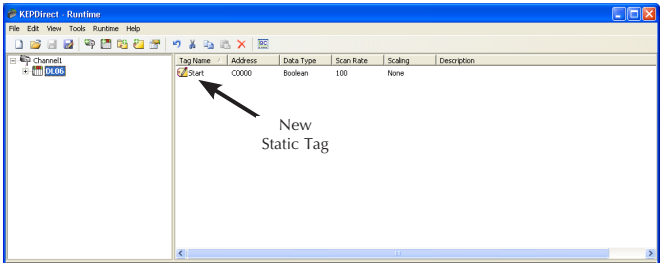
Selecting either **Linear** or **Square Root** will enable the scaling parameters. The **Raw Value Range** allows you to specify the range of raw data from the device. The valid range is dependent upon the data type of the raw tag value. If, for example, the data type is set to Short, the valid range of the raw value would be -32768 to 32767. The raw high range must be greater than the raw low range.

Normally, a scaled value is assumed to result in a floating-point value. The server does not make that assumption for you. The data type of the **Scaled Value Range** can be set to any valid OPC data type. This gives you the ability to scale from a raw data type, such as, Short to an engineering value with a data type of Long if needed. The default scaled data type is Double. The **Scaled Value Range** allows you to specify the range of the resulting scaled value. The valid range is dependent upon the data type of the scaled value. If, for example, the scaled Data type is set to Long, the valid range is -2147483648 to 2147483647. The scaled high range must be greater than the scaled low range.

In many cases the raw data from the device exceeds the range you have specified for the raw data. When this occurs, the scaled value is also forced outside of the range you have established. To prevent this, the **High** and **Low Clamps** can be used to constrain the scaled value to the range specified. The server also allows a **Units** string to be assigned to a scaled tag. The **Units** string can be up to 32 characters long. The last option for the scaled value is the **Negate Scaled Value** option. This option forces the resulting value to be negated before being passed to the client.

The server supports the OPC tag properties available in the 2.0 Data Access specifications. If the OPC client that you are using supports these properties, it can automatically configure the range of objects like user input objects or displays, using the data entered here.

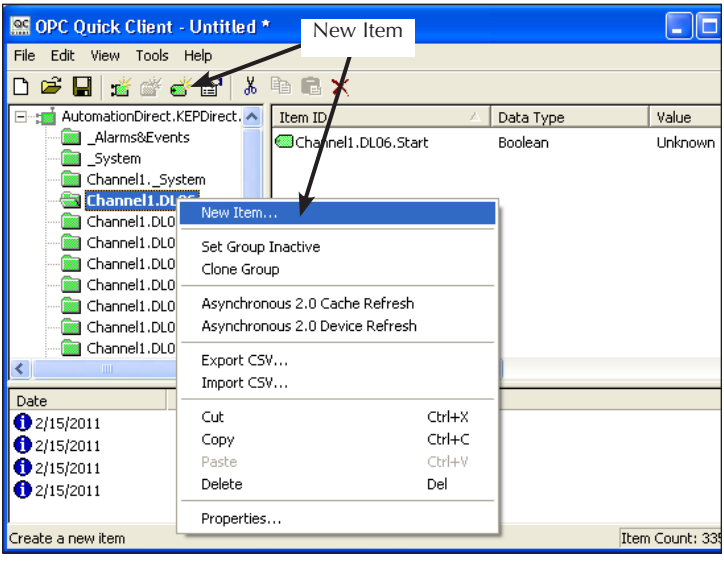
Once created, the new static tag is displayed in the tag entry window as seen below.



Dynamic Tags

Another method for entering and defining tags is called **Dynamic Tag** addressing. Dynamic tags allow you to define tags strictly in the client application. Instead of creating a tag item in your client that addresses another tag item you have created in the server, you need only to create tag items in the client that directly accesses the device driver's addresses. On client connect, the server will create a virtual tag for that location and start scanning for data automatically.

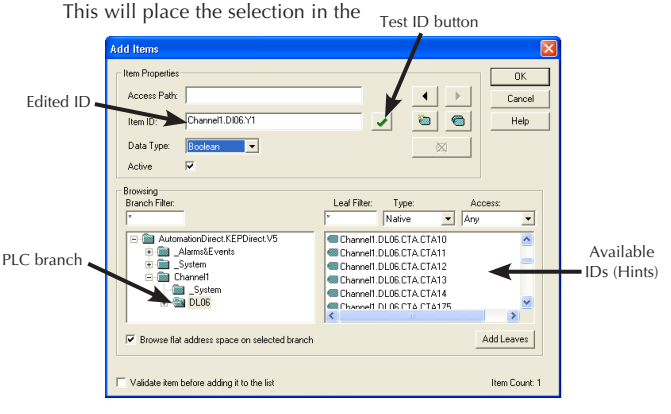
With the OPC client running, click on the device in order to add the new tag to the Item ID panel. Either select **New Item...** from the drop down menu or click on the **New Item** toolbar button to define an item using the item editor dialog.



The **Add Items** dialog, shown below, will appear.

The **Access Path** is required by some OPC servers to complete an item definition. Entry of a path is not necessary.

Item ID is the OPC server item used to reference the data. KEP*Direct* OPC Server supports tag browsing, so open the server's tree branches on the lower left portion of the dialog to locate the device branch. This will place the available IDs on the lower right portion of the dialog. You can browse through the list until the proper ID is located. Click on the ID to select it, then click on the **Add Leaves** button. This will place the selection in the



Item ID window. Edit the ID by removing "Hints" and by typing the correct address. In this case, we are using the address Y1. Clicking on the √ button will test the ID. If the ID is not correct, an error message window will pop-up. If the ID is correct, the **Data Type** will be automatically chosen.

The **Data Type** is used when communicating between the OPC server and the OPC Quick Client. This should be specified to agree with the size and type of the register or memory location that is being addressed. Click on the down arrow next to the Data Type window to open a list of supported data types and select the proper type.

Active is checked by default. The item ID will be active after the **OK** button is pressed.

Clicking on the **OK** button will write the new item ID to the OPC Item ID panel.

Specify Optional Data Types

To specify an optional data type, append one of the strings from the Data Types chart below after the '@' symbol.

Data Types Description		
Type	**Description**	**Prefix**
Boolean	Single bit	X, Y, C, SP, T, CT, IO
Word	Unsigned 16 bit value	V
Short	Signed 16 bit value	V
DWord	Unsigned 32 bit value	V
Long	Signed 32 bit value	V
Float	32 bit Real value IEEE format	V
Double	64 bit Real value IEEE format	V
String	Null terminated ASCII string	V
BCD	Two byte packed BCD value,range is 0-9999	V
LBCD	Four byte packed BCD value, range is 0-99999999	V
Byte	Unsigned 8 bit value	R
Char	Signed 8 bit value	R

If you omit the data type, the driver will choose a default data type based on the device and address you are referencing. The default data types for all locations are documented in the individual driver help files. If the data type specified is not valid for the device location, the server will not accept the tag and an error will be posted in the Event Log window.

The use of the @[Data Type] modifier is not normally required. OPC clients can specify the desired data type as part of the request when registering a link for a specific data item. The data type specified by the OPC client will be used if the communication driver supports the data type. The @[Data Type] modifier can be handy when you want to insure that a communication driver interprets a piece of data exactly as you desire.

Example:

OPC Client Using Dynamic Addressing:

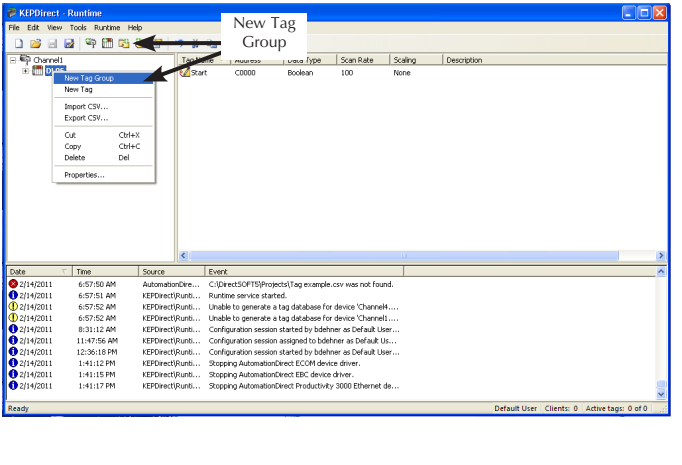
1. Start the OPC client application and connect to the server.
2. Create a channel (called channel1) and device (called Device1) using the simulator driver.
3. In the client application, define an item name of "Channel1. Device1.V3000@BCD".

4. The client project will automatically start receiving data. The default data type for address V3000 in the device is 'Word'. To override this, the "@BCD" has been appended to select a data type of BCD.

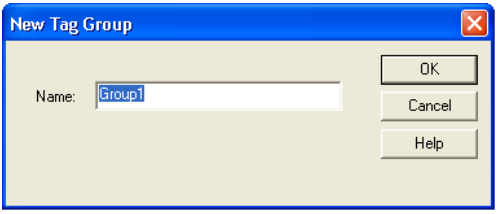
NOTE 1: *The server creates a special Boolean tag for every device in a project that can be used by a client to determine whether that device is functioning properly. To use this tag you would specify the item in the link as Error. The value of this tag is zero if the device is communicating properly otherwise it is one.*
NOTE 2: *If the device address is used as the item of a link such that the address matches the name of a user-defined tag in the server, the link will reference the address pointed to by the user-defined tag.*
NOTE 3: *In order to scale your data in the server you must use static tags.*

Tag Groups

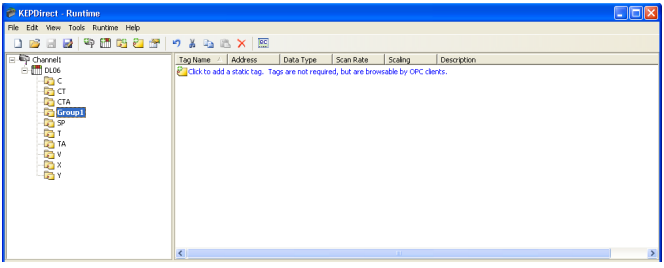
A tag group is used to organize a collection of tags (items) with a common set of properties. Tag groups allow you to tailor the layout of OPC data in logical groupings that fit the needs of your application. Using tag groups allows multiple sets of identical tags to be added under the same device. This can be very convenient when a single device handles a number of similar machine segments. To add a new tag group to your project, either right click on an existing device and select **New Tag Group** from the context menu or click on the **New Tag Group** tool on the toolbar as shown in the figure below.



The following dialog will appear. You can either type in a unique name or use the default name.



After clicking the **OK** button, the main KEP***Direct*** window will appear with the new group branch and the text **Click to add a static tag** as shown below.



Static tags can now be added to the new group by using the method previously described.

Tag groups can be added at any level from the device level down. Multiple tag groups can be nested together to fit the needs of your application.

Glossary

Client:

The computers/devices that use the server connection are called the clients. The client devices need to be physically connected to the server via a hub or a serial cable.

DDE:

Dynamic Data Exchange (DDE) is a generic client server technology provided by Microsoft. DDE provides a basic architecture that allows many Windows applications from a wide range of vendors to share data.

EBC:

This is the abbreviation for Ethernet Based Controller. An EBC is an Ethernet based slave I/O interface.

LinkMaster:

LinkMaster is a fast and robust Windows application that requires no programming knowledge, simply "drag and drop" to create your links. Built-in scaling, user-access manager, error tracking, and write optimization capabilities, provide total control of your data flow and application access.

OLE:

This is an abbreviation of Object Linking and Embedding. OLE is a compound document standard developed by Microsoft Corporation. It enables you to create objects with one application and then link or embed them in a second application. Embedded objects retain their original format and links to the application that created them.

OPC:

This is an abbreviation for Object Process Control. This is the same as OLE (defined above), but for Process Control.

SCADA:

Supervisory Control and Data Acquisition (SCADA). A SCADA system normally consists of a host software program, an HMI unit, a telemetry system, and Remote Terminal Units (RTUs) and/or Programmable Logic Controllers (PLCs) that are typically installed in remote areas and communicate over long distances to the Host system.

Server:

A server is a computer/device, which provides information or services to computers on a network.