ProSoft TECHNOLOGY

Where Automation Connects.

PLX32-EIP-MBTCP-UA

Multi-Protocol Gateway

June 5, 2025

USER MANUAL

Your Feedback Please

We always want you to feel that you made the right decision to use our products. If you have suggestions, comments, compliments or complaints about our products, documentation, or support, please write or call us.

ProSoft Technology, Inc.
+1 (661) 716-5100
+1 (661) 716-5101 (Fax)
www.prosoft-technology.com
ps.support@belden.com

PLX32-EIP-MBTCP-UA User Manual
For Public Use.

June 5, 2025

Content Disclaimer

This documentation is not intended as a substitute for and is not to be used for determining suitability or reliability of these products for specific user applications. It is the duty of any such user or integrator to perform the appropriate and complete risk analysis, evaluation and testing of the products with respect to the relevant specific application or use thereof. Neither ProSoft Technology nor any of its affiliates or subsidiaries shall be responsible or liable for misuse of the information contained herein. Information in this document including illustrations, specifications and dimensions may contain technical inaccuracies or typographical errors. ProSoft Technology makes no warranty or representation as to its accuracy and assumes no liability for and reserves the right to correct such inaccuracies or errors at any time without notice. If you have any suggestions for improvements or amendments or have found errors in this publication, please notify us.

No part of this document may be reproduced in any form or by any means, electronic or mechanical, including photocopying, without express written permission of ProSoft Technology. All pertinent state, regional, and local safety regulations must be observed when installing and using this product. For reasons of safety and to help ensure compliance with documented system data, only the manufacturer should perform repairs to components. When devices are used for applications with technical safety requirements, the relevant instructions must be followed. Failure to use ProSoft Technology software or approved software with our hardware products may result in injury, harm, or improper operating results. Failure to observe this information can result in injury or equipment damage.

For professional users in the European Union
If you wish to discard electrical and electronic equipment (EEE), please contact your dealer or supplier for further information.

Prop 65 Warning – Cancer and Reproductive Harm – www.P65Warnings.ca.gov

Agency Approvals and Certifications

Please visit our website: www.prosoft-technology.com

Open Source Information

Open Source Software used in the product

The product contains, among other things, Open Source Software files, as defined below, developed by third parties and licensed under an Open Source Software license. These Open Source Software files are protected by copyright. Your right to use the Open Source Software is governed by the relevant applicable Open Source Software license conditions. Your compliance with those license conditions will entitle you to use the Open Source Software as foreseen in the relevant license. In the event of conflicts between other ProSoft Technology, Inc. license conditions applicable to the product and the Open Source Software license conditions, the Open Source Software conditions shall prevail. The Open Source Software is provided royalty-free (i.e. no fees are charged for exercising the licensed rights). Open Source Software contained in this product and the respective Open Source Software licenses are stated in the module webpage, in the link Open Source.
If Open Source Software contained in this product is licensed under GNU General Public License (GPL), GNU Lesser General Public License (LGPL), Mozilla Public License (MPL) or any other Open Source Software license, which requires that source code is to be made available and such source code is not already delivered together with the product, you can order the corresponding source code of the Open Source Software from ProSoft Technology, Inc. - against payment of the shipping and handling charges - for a period of at least 3 years since purchase of the product. Please send your specific request, within 3 years of the purchase date of this product, together with the name and serial number of the product found on the product label to:

ProSoft Technology, Inc.
Director of Engineering
9201 Camino Media, Suite 200
Bakersfield, CA 93311
USA

Warranty regarding further use of the Open Source Software

ProSoft Technology, Inc. provides no warranty for the Open Source Software contained in this product, if such Open Source Software is used in any manner other than intended by ProSoft Technology, Inc. The licenses listed define the warranty, if any, from the authors or licensors of the Open Source Software. ProSoft Technology, Inc. specifically disclaims any warranty for defects caused by altering any Open Source Software or the product's configuration. Any warranty claims against ProSoft Technology, Inc. in the event that the Open Source Software contained in this product infringes the intellectual property rights of a third party are excluded. The following disclaimer applies to the GPL and LGPL components in relation to the rights holders:
"This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License and the GNU Lesser General Public License for more details."
For the remaining open source components, the liability exclusions of the rights holders in the respective license texts apply. Technical support, if any, will only be provided for unmodified software.

Table of Contents

1 Start Here

1.1 Overview

This document explains the features of the PLX32-EIP-MBTCP-UA gateway. It covers the configuration, showing how to map data between a device or network, through the gateway, to a PLC or PAC. The ProSoft Configuration Builder (PCB) software creates files to import into the PLC or PAC programming software, integrating the gateway into the system.

Data can also be mapped between areas in the gateway's internal database. This allows data to be copied to different addresses within the gateway database in order to create easier data requests and control.

The PLX32-EIP-MBTCP-UA is a stand-alone DIN-rail mounted unit that provides two Ethernet ports for communications, remote configuration, and diagnostics. The gateway has an SD Card slot (SD card optional) that allows the storage of configuration files used for recovery, transferring the configuration to another gateway, or general configuration backup.

1.2 System Requirements

The ProSoft Configuration Builder software for the PLX32-EIP-MBTCP-UA requires the following minimum system components:

- Windows 7 Professional (32-bit version), 8 GB RAM Intel® Core™ i5 650 (3.20 GHz)
- Windows XP Professional Ver.2002 Service Pack 2, 512 MB RAM Pentium 4 (2.66 GHz)
- Windows 2000 Ver.5.00.2195 Service Pack 2 512 MB RAM Pentium III (550 MHz)

Note: To use PCB under the Windows 7 OS, install PCB using the "Run as Administrator" option. To find this option, right-click on the **Setup.exe** installer program icon. In the context menu, use the "Run as Administrator" option. Using the "Run as Administrator" option allows the PCB installer to create folders and files on the PC with proper permissions and security.

If the "Run as Administrator" option is not used, PCB may appear to install correctly; but will receive numerous, repeating file access errors whenever PCB is running, especially when changing configuration screens. If this happens, completely uninstall PCB and then re-install using the "Run as Administrator" option.

1.3 Package Contents

The following components are included with the PLX32-EIP-MBTCP-UA, and are all required for installation and configuration.

Important: Before beginning the installation, please verify that all the following items are present.

Qty.	Part Name	Part Number	Part Description
1	Mini screwdriver	HRD250	Tool for wiring and securing the power connector
1	Power connector	J180	PLX32-EIP-MBTCP-UA power connector
1	Jumper	J809	Spare jumper for resetting OPC UA configuration

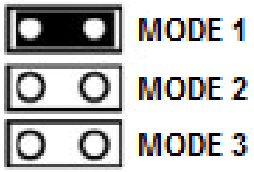
1.4 Mounting the Gateway on a DIN-rail

To mount the PLX32-EIP-MBTCP-UA on a DIN-rail:

1. Position the gateway on the DIN-rail B at a slight angle.
2. Hook the lip on the rear of the adapter onto the top of the DIN-rail, and rotate the adapter onto the rail.
3. Press the adapter down onto the DIN-rail until flush. The locking tab snaps into position and lock the gateway to the DIN-rail.
4. If the adapter does not lock in place, use a screwdriver or similar device to move the locking tab down while pressing the adapter flush onto the DIN-rail and release the locking tab to lock the adapter in place. If necessary, push up on the locking tab to lock.

1.5 Jumper Settings

There are three pairs of jumper pins located on the back of the gateway.



- **MODE 1** - The two pins should be jumpered during normal operation.
- **MODE 2** - Default IP Jumper: This is the middle jumper. The default IP address of the gateway is 192.168.0.250. Set this jumper to put the gateway's IP address back to the default.
- **MODE 3** - If set, this jumper provides a level of security resulting in the following behaviors:
  - This jumper disables ProSoft Configuration Builder (PCB) upload and download functions. If an upload or download request is made through PCB, an error message occurs indicating that these functions are unavailable.
  - This jumper also disables access to the PLX32-EIP-MBTCP-UA web page making it impossible to upgrade the firmware.

Attention: Simultaneously setting jumper **MODE 1** and **MODE 3** will restore the OPC UA configuration to factory defaults.

1.6 SD Card

An optional SD card (Part Number SDI-1G) can be ordered for the PLX32-EIP-MBTCP-UA. In the event of a gateway failure, the SD card can be removed from one gateway to another to resume operation.

The gateway uses the configuration on the SD card if SD card is present when powered up or rebooted.

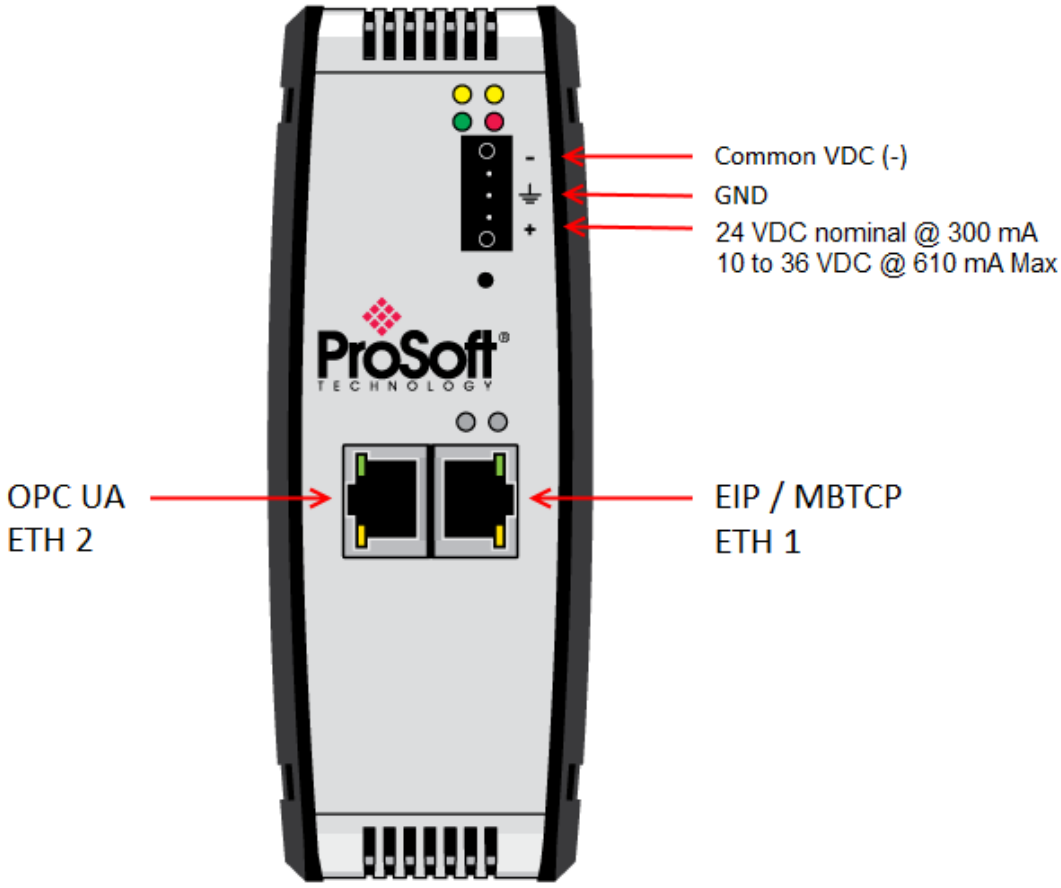
1.6.1 With an SD Card

- The ProSoft Configuration Builder downloads the configuration to the SD Card in the gateway.
- The gateway does not transfer the configuration data from the SD card to internal memory. If the SD card is removed and the gateway is rebooted, the gateway loads the configuration data from the gateway's memory. If there is no configuration data in the gateway's memory, the gateway uses the factory default configuration.

1.6.2 Without an SD Card

- The ProSoft Configuration Builder downloads the configuration to the internal memory of the gateway. The gateway uses the configuration from internal memory.
- If a blank SD Card is inserted into the gateway after the gateway has been configured, the gateway does not use the configuration on the SD card unless the gateway is rebooted. To copy the configuration to the SD card, download the configuration to the gateway while the SD card is in the gateway.

1.7 Connecting Power to the Unit



WARNING: Be sure not to reverse polarity when applying power to the gateway. This causes permanent damage to the gateway's internal power distribution circuits.

1.8 Installing ProSoft Configuration Builder Software

The ProSoft Configuration Builder *(*PCB) software is used to configure the gateway. Download the latest version at: www.prosoft-technology.com.

1. Open a web browser and navigate to www.prosoft-technology.com.
2. Search for '*PCB*' or '*ProSoft Configuration Builder*'.
3. Click on the ProSoft Configuration Builder search result link.
4. From the *Downloads* link, download the latest version of *ProSoft Configuration Builder*.
5. Choose **SAVE** or **SAVE FILE**, if prompted.
6. Save the file to the *Windows Desktop*.
7. When the download is complete, locate and open the file, and then follow the instructions on the screen to install the program.

Note: To use PCB under the Windows 7 OS, install PCB using the "Run as Administrator" option. To find this option, right-click on the **Setup.exe** installer program icon. In the context menu, use the "Run as Administrator" option. Using the "Run as Administrator" option allows the PCB installer to create folders and files on the PC with proper permissions and security.

If the "Run as Administrator" option is not used, PCB may appear to install correctly; but will receive numerous, repeating file access errors whenever PCB is running, especially when changing configuration screens. If this happens, completely uninstall PCB and then re-install using the "Run as Administrator" option.

To ensure a successful installation of ProSoft OPC UA Configuration Manager, a reboot may be required prior to starting the installation. In several test systems, Windows Update Service had to be stopped prior to installation. Once the installation completes, restart the Windows Update service.

Stop Windows Update service

1. Click the Windows *Start* button and enter the following: **services.msc**

2. Scroll down and right-click on Windows Update and choose **STOP**.

Perform the ProSoft OPC UA Configuration Manager setup procedures. Once the setup completes, perform the steps above and choose *Start* for the last step.

2 Using ProSoft Configuration Builder

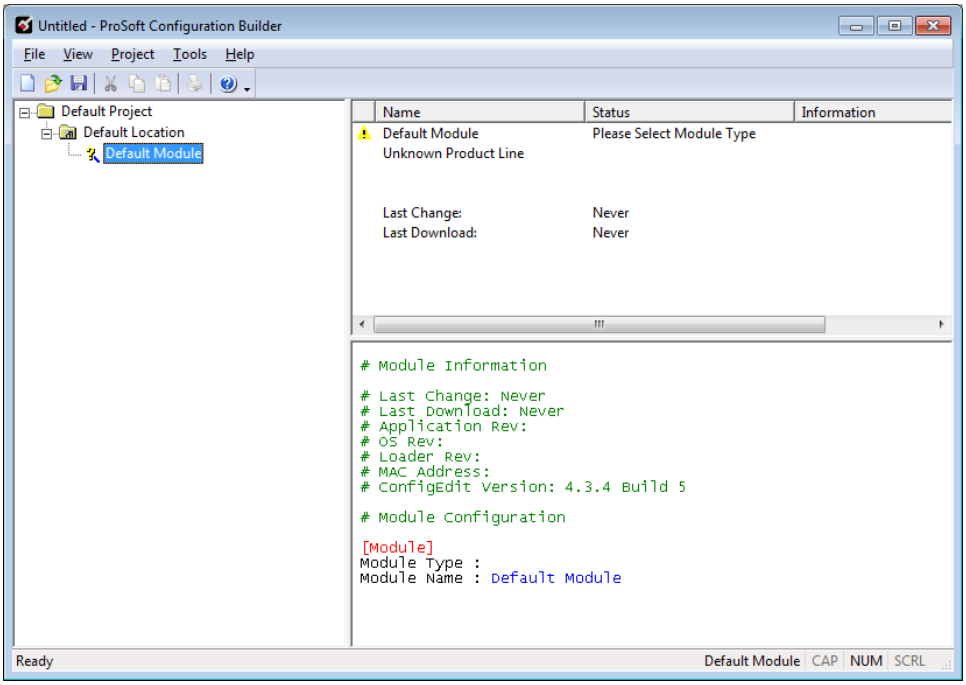
The ProSoft Configuration Builder (PCB) software provides a quick and easy way to manage gateway configuration files customized to the needs of the application. Information from previously installed (known working) configurations can be applied to new projects.

2.1 Connecting the PC to the Gateway

With the gateway securely mounted, connect one end of the Ethernet cable to the **ETH 1** Port, and the other end to an Ethernet hub or switch accessible from the same network as the PC. Or, connect directly from the Ethernet Port on the PC to the **ETH 1** Port on the gateway.

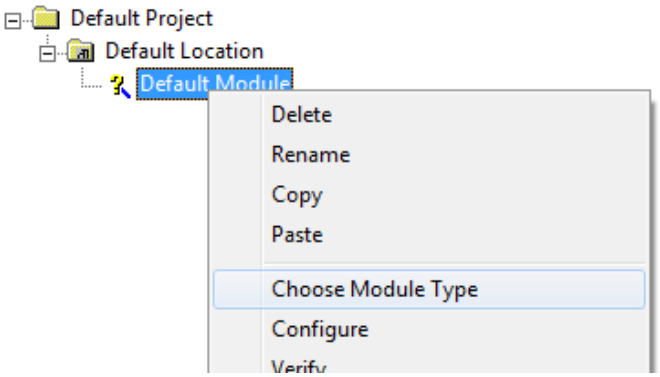
2.2 Creating a New Project

The ProSoft Configuration Builder window consists of a tree view on the left, an information pane, and a configuration pane on the right side of the window. When PCB is first started, the tree view consists of folders for *Default Project*, *Default Location*, and a *Default Module*.

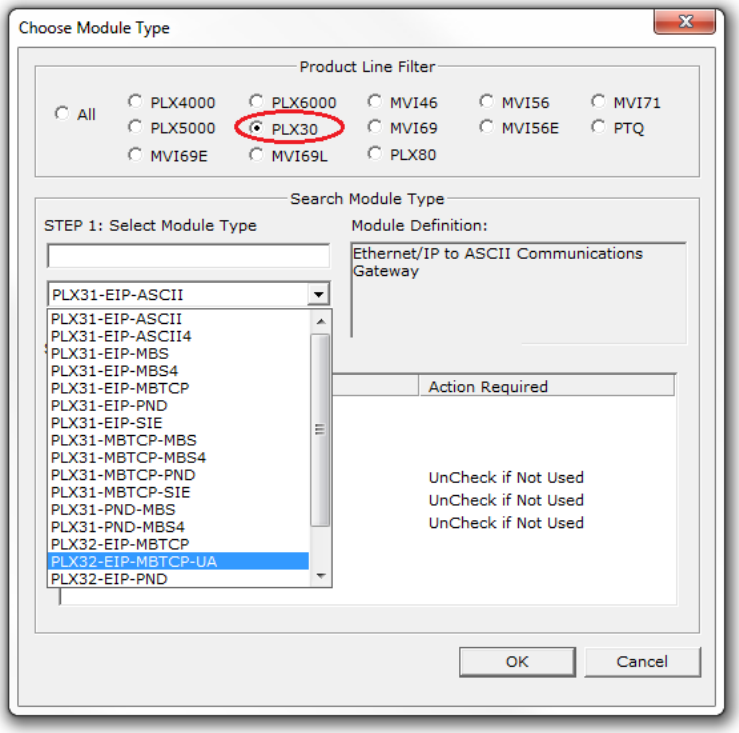


To add the gateway to the project:

1 Right-click **DEFAULT MODULE** in the tree view, and then choose **CHOOSE MODULE TYPE**. This opens the *Choose Module Type* dialog.



2 In the *Product Line Filter* area of the dialog, select the **PLX30** radio button.



3 In the *STEP 1: Select Module Type* dropdown list, select **PLX32-EIP-MBTCP-UA**.

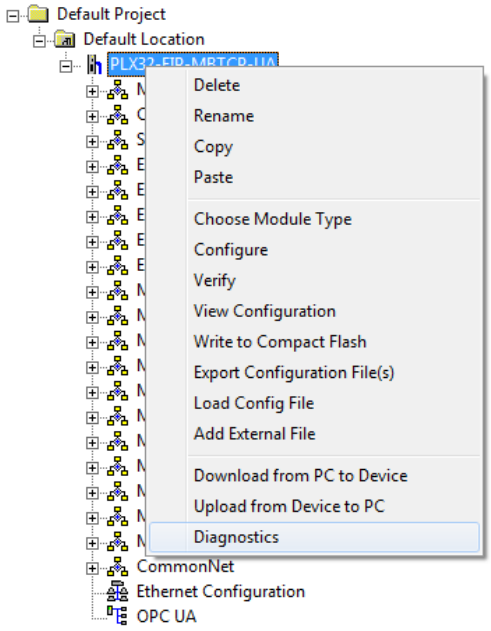
4 Disable one or more drivers on the gateway they are not needed. For more information, see section *2.4 Disabling Gateway Protocol Functionalities*.

5 Click **OK** to save the settings and return to the PCB Main window.

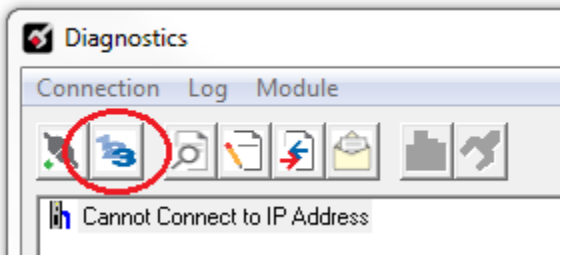
2.3 Setting a Temporary IP Address in the Gateway

Important: ProSoft Discovery Service (PDS) locates the gateway through UDP broadcast messages. PDS is an application that is built into PCB. These messages may be blocked by routers or layer 3 switches. In that case, PDS is unable to locate the gateways.
To use PDS, arrange the Ethernet connection so that there is no router or layer 3 switch between the computer and the gateway OR reconfigure the router or layer 3 switch to allow the routing of the UDP broadcast messages.

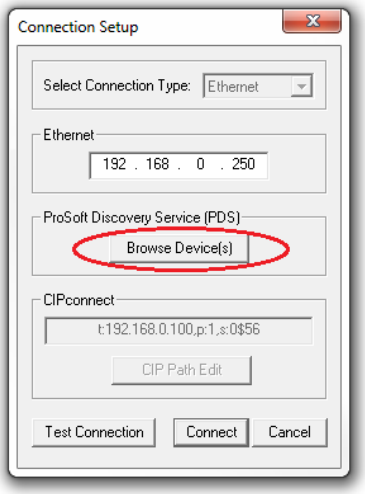
1 To open PDS, right-click on the **PLX32-EIP-MBTCP-UA** icon in PCB and click on **DIAGNOSTICS**.



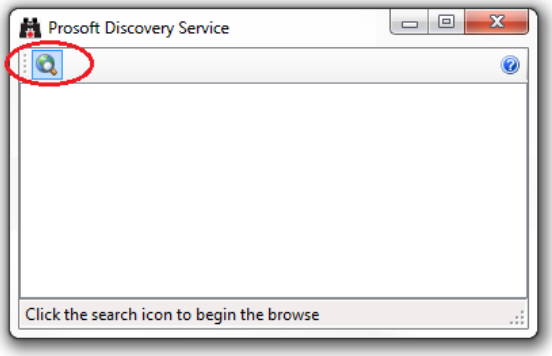
2 In the *Diagnostics* dialog, click on the **CONNECTION SETUP** icon.



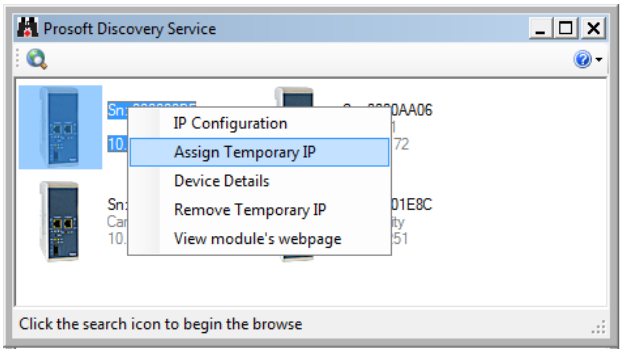
3 In the *Connection Setup* dialog, click the **BROWSE DEVICE(S)** button under the *ProSoft Discovery Service (PDS)* heading.



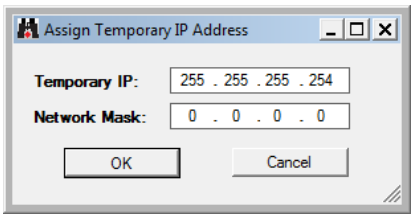
4 In the *ProSoft Discovery Service* dialog, click on the **BROWSE FOR PROSOFT MODULES** icon to search for ProSoft Technology modules on the network.



5 Right-click on the gateway, and then select **ASSIGN TEMPORARY IP**.



6 The gateway's default IP address is **192.168.0.250**.



7 Enter an unused IP within the subnet, and then click **OK**.

8 To set the permanent IP address in the gateway, see section *2.6 Configuring the Ethernet Port*.

2.4 Disabling Gateway Protocol Functionalities

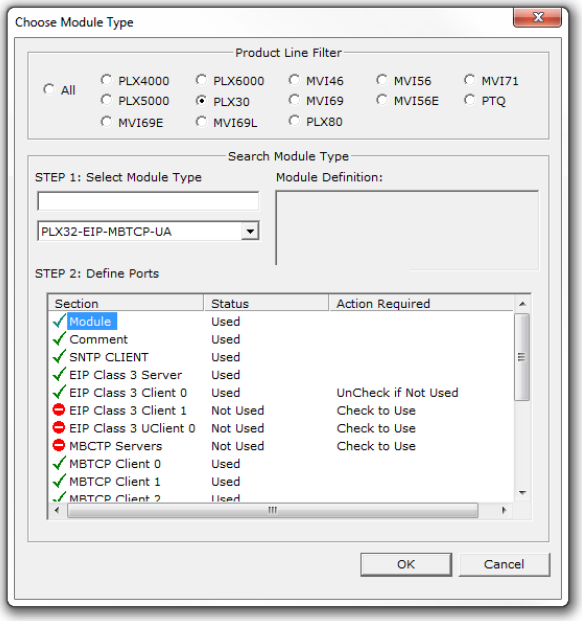
ProSoft Configuration Builder (PCB) has the option to disable one or more driver functionalities if they are not needed. Disabling driver functionalities can simplify the number of configuration options, making it easier to set up the gateway.

It is easiest to disable driver functionalities when the gateway is added to the project in PCB. The functionalities can be enabled and disabled after the gateway is added to the project. Both methods are described in this topic.

Note: Disabling driver functionalities does not affect the performance of the gateway and is not required.

2.4.1 Disabling Driver Functionalities When Adding the Gateway to the Project

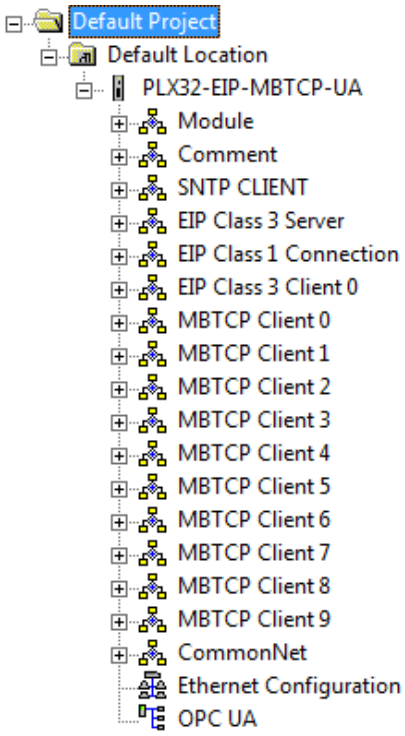
The best time to disable one or more driver functionalities on the gateway is when the gateway is added to the project in PCB. Disable them by clicking on a driver name in the *Choose Module Type* dialog. The following image gives an example.



There are three driver functionalities disabled. Please note the following:

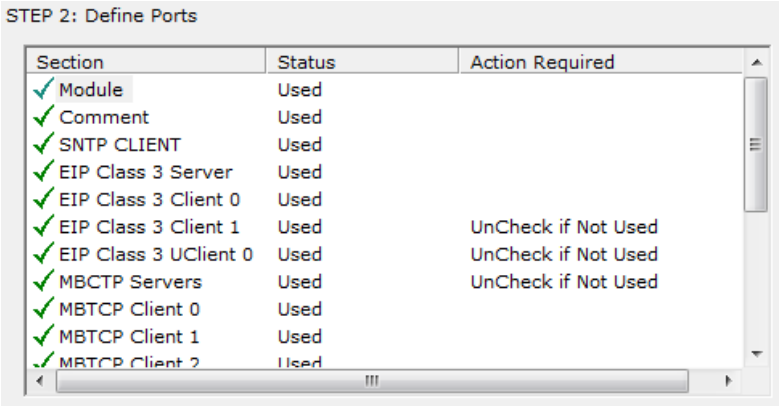
- Drivers that are disabled have **UNCHECK IF NOT USED** in the **ACTION REQUIRED** column.
- When a driver is disabled, a red circle replaces the green checkmark.
- If there are multiple drivers of the same type, only the last one has the *UnCheck if not Used* message. Disable and enable only in reverse order.
- Finally, to enable a disabled functionality in this dialog, click the driver functionality name again.

After clicking **OK**, PCB inserts the gateway into the tree view with the disabled configuration options hidden.



2.4.2 Disabling or Enabling Driver Functionalities After the Gateway has Been Added to the Project

1 Right-click the **PLX32-EIP-MBTCP-UA** icon in the tree view, and then choose **CHOOSE MODULE TYPE**. This opens the *Choose Module Type* dialog, with the correct **MODULE TYPE**.



Warning: Note that all the drivers are enabled by default, and that the driver state in the *Choose Module Type* dialog does not match the actual state of the drivers.

2 Click the driver functionality name to change its status from **Enabled** to **Disabled**, or vice-versa. The same rules noted above still apply.

3 After clicking **OK**, PCB updates the gateway in the tree view, showing the configuration options for the enabled functionalities, and hiding the disabled functionalities.

2.5 Configuring Gateway Parameters

1. Click the **[+]** sign next to the module icon to expand gateway information.
2. Click the **[+]** sign next to any icon to view gateway information and configuration options.
3. Double-click any icon to open an *Edit* dialog.
4. To edit a parameter, select the parameter in the left pane and make changes in the right pane.
5. Click **OK** to save the changes.

2.5.1 Renaming PCB Objects

Rename objects such as the *Default Project* and *Default Location* folders in the tree view. The **MODULE** icon can be renamed to customize the project.

1. Right-click the object to rename and then choose **RENAME.**
2. Type the new name for the object and press **Enter**.

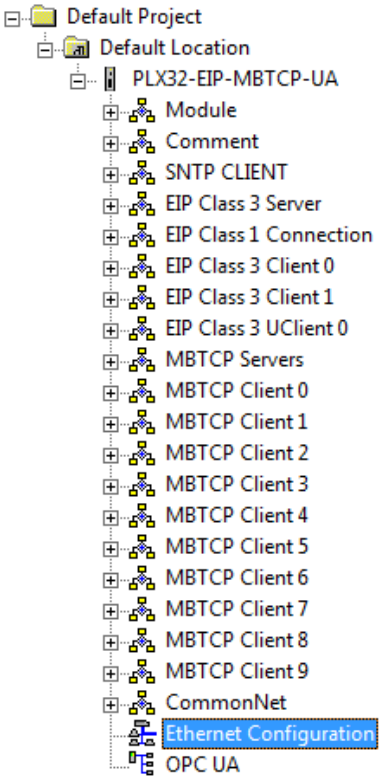
2.5.2 Printing a Configuration File

1. In the main PCB window, right-click the **PLX32-EIP-MBTCP-UA** icon and then choose **VIEW CONFIGURATION**.
2. In the *View Configuration* dialog, click the **FILE** menu and click **PRINT**.
3. In the *Print* dialog, choose the printer to use from the drop-down list, select the printing options, and click **OK**.

2.6 Configuring the Ethernet Port

This section shows how to set the Ethernet port parameters for the PLX32-EIP-MBTCP-UA.

1 In the ProSoft Configuration Builder tree view, double-click on the *Ethernet Configuration* icon.



2 Click any parameter in the *Edit - WATTCP* dialog to change the value. Since the gateway has two Ethernet ports, there are separate configuration options for each port.

Parameter	Description
IP Address	Unique IP address assigned to the gateway
Netmask	Subnet mask of gateway
Gateway	Gateway (if used)

Note: Each Ethernet port must be on a different Ethernet subnet.

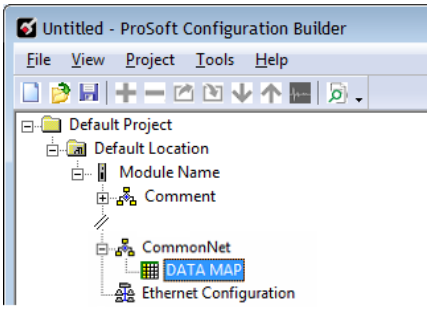
2.7 Mapping Data in PLX32-EIP-MBTCP-UA Memory

Use the *DATA MAP* section in the ProSoft Configuration Builder to copy data between areas in the gateway's internal database. Data can be copied to different addresses within the gateway database in order to create simpler data requests and control. Use this feature for the following tasks.

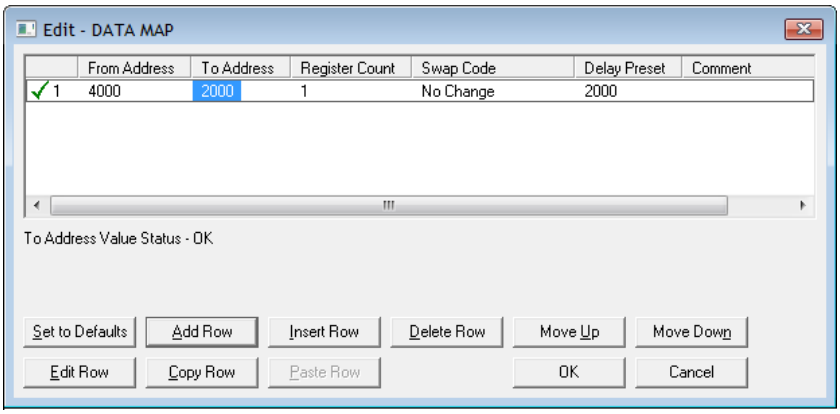
- Copy a maximum of 100 registers per Data Map command, and configure a maximum of 200 separate copy commands.
- Copy data from the error or status tables in upper memory to internal database registers in the user data area.
- Rearrange the byte and/or word order during the copy process. For example, by rearranging byte or word order, convert floating-point values to the correct format for a different protocol.
- Use the Data Map to condense widely dispersed data into one contiguous data block, making it easier to access.

1 In the ProSoft Configuration Builder, expand the module tree by clicking the **[+]** next to the module name.

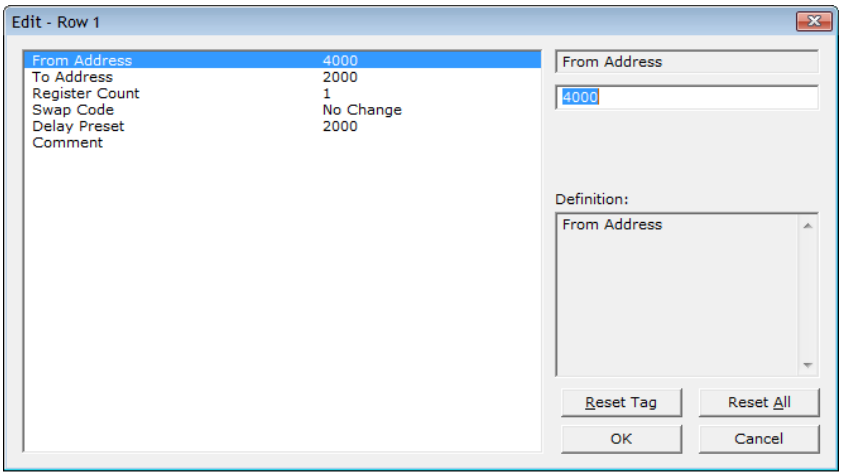
2 Click the **[+]** next to **COMMONNET**, and then double-click **DATA MAP**.



3 In the *Edit - Data Map* dialog, click **ADD ROW**.



4 Click **EDIT ROW** to edit the parameters for the mapping.



5 To change the value of a parameter, click the parameter and enter a new value. Click **OK** when finished.

6 Repeat the above steps to add more memory mappings.

2.7.1 From Address

0 to highest *Status Data* address

Specifies the beginning internal database register address for the copy operation. This address can be any valid address in the user data area or the status data area of the gateway.

2.7.2 To Address

0 to **9999**

Specifies the beginning destination register address for the copy operation. This address must always be within the user data area. Specify a destination address that does not overwrite data that is stored in memory by one of the communication protocols running on the gateway.

2.7.3 Register Count

1 to **100**

Specifies the number of registers to copy.

2.7.4 Swap Code

NO CHANGE, **WORD SWAP**, **WORD AND BYTE SWAP**, **BYTE SWAP**

Specifies the order of the bytes in the registers during the copy process in order to change the alignment of bytes between different protocols. Use this parameter when dealing with floating-point or other multi-register values, because there is no standard for storage of these data types in slave devices.

Swap Code	Description
No Swap	No change is made in the byte ordering (1234 = 1234)
Word Swap	The words are swapped (1234 = 3412)
Word and Byte Swap	The words are swapped, then the bytes in each word are swapped (1234 = 4321)
Bytes	The bytes in each word are swapped (1234 = 2143)

2.7.5 Delay Preset

This parameter sets an interval for each *Data Map* copy operation. The value for the *Delay Preset* is not a fixed amount of time. It is the number of firmware scans that must transpire between copy operations.

The firmware scan cycle can take a variable amount of time, depending on the level of activity of the protocol drivers running on the gateway and the level of activity on the gateway's communication ports. Each firmware scan can take from one to several milliseconds to complete. Therefore, *Data Map* copy operations cannot be expected to happen at regular intervals.

If multiple copy operations (several rows in the *Data map* section) happen too frequently or all happen in the same update interval, they could delay the process scan of the gateway protocols, which could result in slow data updates or missed data on communication ports. To avoid these potential problems, set the *Delay Preset* to different values for each row in the *Data Map* section and set them to higher, rather than lower, numbers.

For example, *Delay Preset* values below 1000 could cause a noticeable delay in data updates through the communication ports. Do not set all *Delay Presets* to the same value. Instead, use different values for each row in the Data Map such as 1000, 1001, and 1002 or any other different *Delay Preset* values. This prevents the copies from happening concurrently and prevents possible process scan delays.

2.8 Downloading the Project to the PLX32-EIP-MBTCP-UA

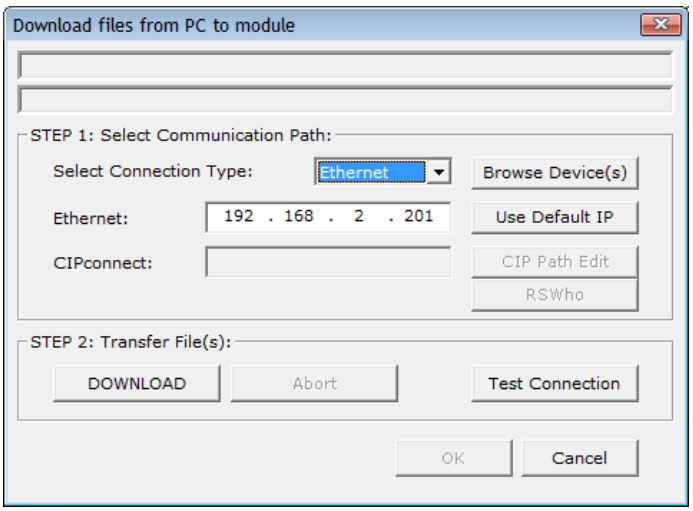
Download the current project in ProSoft Configuration Builder to the PLX32-EIP-MBTCP-UA.

Note: To connect the PLX32-EIP-MBTCP-UA to a PC, see section *2.1 Connecting the PC to the Gateway*.

Note: If jumper 3 of the PLX32-EIP-MBTCP-UA is set, this function is not available.

1. In the tree view in *ProSoft Configuration Builder*, right-click the **PLX32-EIP-MBTCP-UA** icon and then choose **DOWNLOAD FROM PC TO DEVICE.** This opens the *Download* dialog.
2. In the *Download* dialog, in the *Select Connection Type* dropdown box, use the default **ETHERNET** option.

Note: If connected to the gateway using a temporary IP address, the Ethernet address field contains that temporary IP address. *ProSoft Configuration Builder* uses this temporary IP address to connect to the gateway.

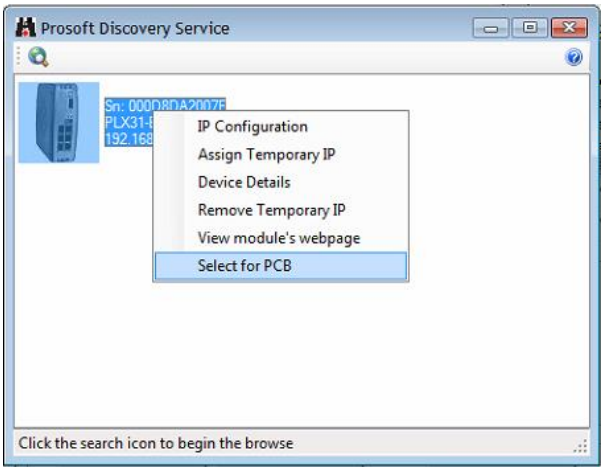


3. Click **TEST CONNECTION** to verify that the IP address allows access to the gateway.
4. If the connection succeeds, click **DOWNLOAD** to transfer the Ethernet configuration to the gateway.

Note: The steps above only downloads or modifies the OPC UA server's IP address and name, it does not download or modify the OPC UA configuration.

If the *Test Connection* procedure fails, follow these steps:

1. Click **OK** to dismiss the error message.
2. In the *Download* dialog, click **BROWSE DEVICE(S)** to open *ProSoft Discovery Service*.



3. Right-click the module and then choose **SELECT FOR PCB**.
4. Close *ProSoft Discovery Service*.
5. Click **DOWNLOAD** to transfer the configuration to the PLX32-EIP-MBTCP-UA.

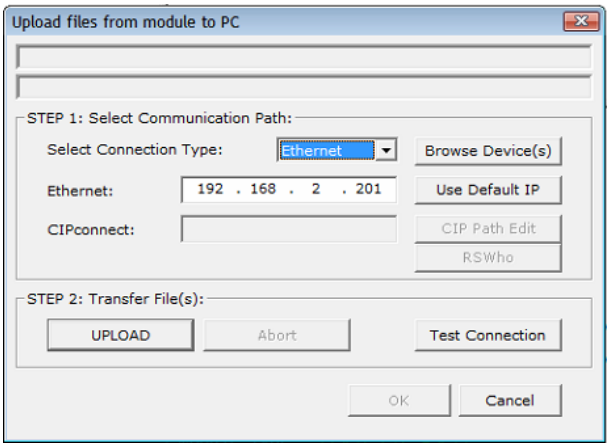
2.9 Uploading the Project from the PLX32-EIP-MBTCP-UA

Upload the project settings from the PLX32-EIP-MBTCP-UA to the current project in ProSoft Configuration Builder.

Note: To connect the PLX32-EIP-MBTCP-UA to a PC, see section *2.1 Connecting the PC to the Gateway*.

1. In the tree view in *ProSoft Configuration Builder*, right-click the **PLX32-EIP-MBTCP-UA** icon and then choose **UPLOAD FROM DEVICE TO PC**. This opens the *Upload* dialog.
2. In the *Upload* dialog, in the *Select Connection Type* dropdown box, use the default **ETHERNET** setting.

Note: If connected to the PLX32-EIP-MBTCP-UA using a temporary IP address, the Ethernet address field contains that temporary IP address. ProSoft Configuration Builder uses this temporary IP address to connect to the gateway.

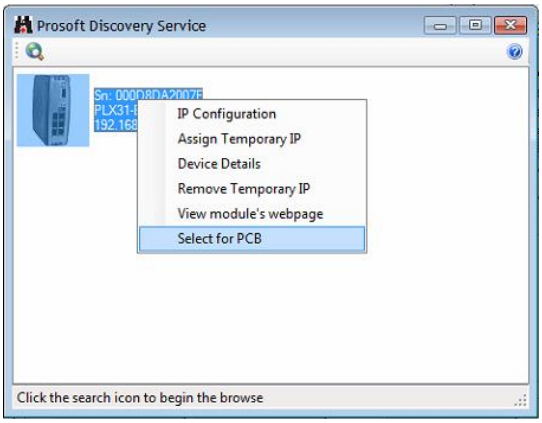


3. Click **TEST CONNECTION** to verify that the IP address allows access to the gateway.
4. If the connection succeeds, click **UPLOAD** to transfer the Ethernet configuration to the PC.

Note: The steps above only uploads or modifies the OPC UA server's IP address and name, it does not upload or modify the OPC UA configuration.

If the *Test Connection* procedure fails, follow these steps.

1. Click **OK** to dismiss the error message.
2. In the *Upload* dialog, click **BROWSE DEVICE(S)** to open *ProSoft Discovery Service*.



3. Right-click the gateway and then choose **SELECT FOR PCB**.
4. Close *ProSoft Discovery Service*.
5. Click **DOWNLOAD** to transfer the configuration to the gateway.

3 Diagnostics and Troubleshooting

Troubleshoot the gateway using several methods:

- Monitor the LED indicators on the gateway.
- Use the Diagnostics functions in ProSoft Configuration Builder (PCB).
- Examine the data in the status data area (upper memory) of the gateway internal memory.

3.1 LED Indicators

The first and quickest is to scan the LEDs on the gateway to determine the existence and possible cause of a problem. The LEDs provide valuable information such as:

- The state of each port
- System configuration errors
- Application errors
- Fault indications

3.1.1 Main Gateway LEDs

This table describes the gateway front panel LEDs.

LED	State	Description
PWR (Power)	Off	Power is not connected to the power terminals or source is insufficient to properly power the gateway (208 mA at 24 VDC is required).
	Solid Green	Power is connected to the power terminals.
FLT (Fault)	Off	Normal operation.
	Solid Red	A critical error has occurred. Program executable has failed or has been user-terminated and is no longer running. Press the **Reset** button or cycle power to clear the error.
CFG (Configuration)	Off	Normal operation.
	Solid Amber	The unit is in configuration mode. Either a configuration error exists, or the configuration file is being downloaded or read. After power-up, the gateway reads the configuration, and the unit implements the configuration values and initializes the hardware. This occurs during power cycle or pressing the **Reset** button.
ERR (Error)	Off	Normal operation.
	Flashing Amber	An error condition has been detected and is occurring on one of the application ports. Check configuration and troubleshoot for communication errors.
	Solid Amber	This error flag is cleared at the start of each command attempt (master/client) or on each receipt of data (slave/adapter/server). If this condition exists, it indicates a large number of errors are occurring in the application (due to bad configuration) or on one or more ports (network communication failures).
NS (Network Status) for EIP protocol only	Off	No power or no IP address
	Solid Red	Duplicate IP address
	Solid Green	Connected
	Flashing Red	Connection timeout
	Flashing Green	IP address obtained; no established connections
	Alternating Red and Green Flash	Self-test
MS (Module Status) for EIP protocol only	Off	No power
	Solid Red	Major fault
	Solid Green	Device operational
	Flashing Red	Minor fault
	Flashing Green	Standby
	Alternating Red and Green Flash	Self-test

3.1.2 Ethernet Port LEDs

This table describes the gateway Ethernet port LEDs.

LED	State	Description
LINK/ACT	Off	No physical network connection is detected. No Ethernet communication is possible. Check wiring and cables.
	Solid Green	Physical network connection detected. This LED must be ON solid for Ethernet communication to be possible.
100 Mbit	Off	No activity on the port.
	Flashing Amber	The Ethernet port is actively transmitting or receiving data.

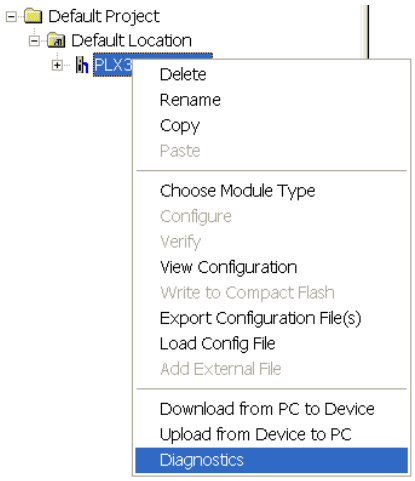
3.2 Using Diagnostics in ProSoft Configuration Builder

ProSoft Configuration Builder has many useful tools to help with diagnostics and troubleshooting. Use PCB to connect to the gateway and retrieve current status values, configuration data and other valuable information.

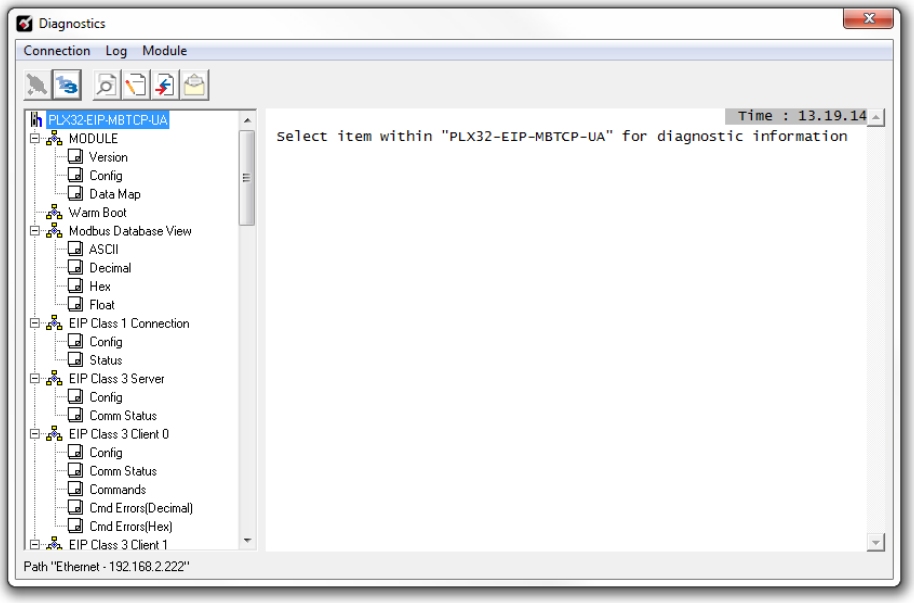
Tip: A ProSoft Configuration Builder Diagnostics window can be open for more than one gateway at a time.

3.2.1 Connecting to the PLX32-EIP-MBTCP-UA Communication Port

1 In PCB, right-click the gateway name and choose **DIAGNOSTICS**.

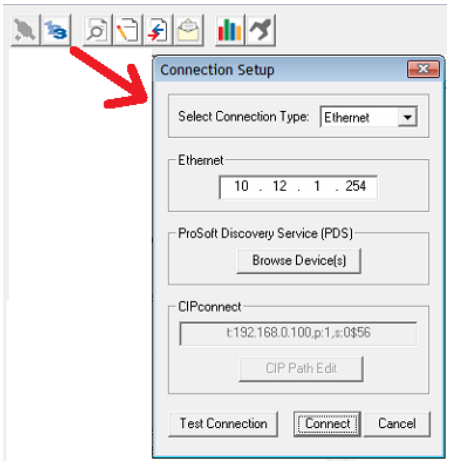


2 This opens the *Diagnostics* window.



If there is no response from the gateway, follow these steps:

1 From the toolbar, click the **SETUP CONNECTION** button.



2 In the *Connection Setup* dialog, select **ETHERNET** from the **SELECT CONNECTION TYPE** list.

3 Type in the gateway's IP address in the **ETHERNET** field.

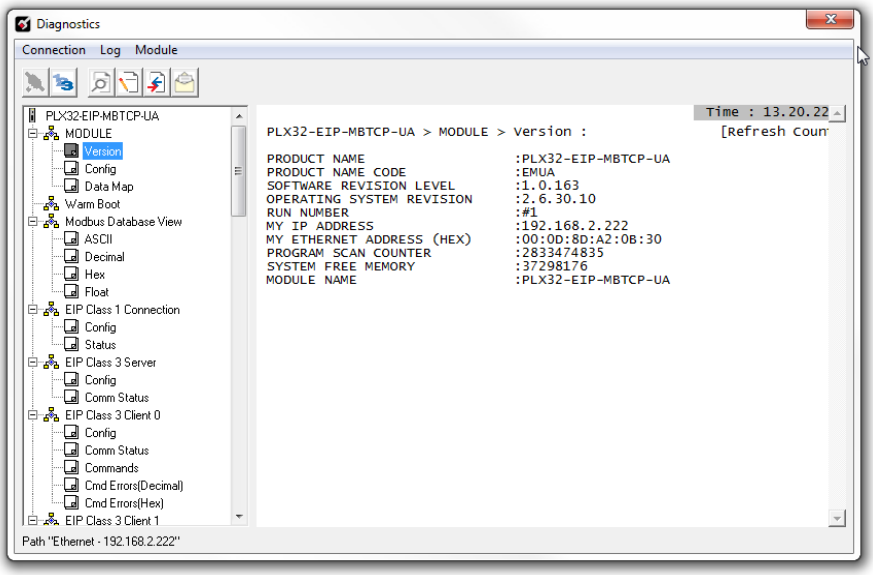
4 Click **CONNECT**.

5 Verify that the Ethernet is connected properly between the computer's communication port and the gateway.

6 If a connection cannot be established, contact ProSoft Technology Technical Support for assistance.

3.2.2 Diagnostics Menu

The *Diagnostics* menu is arranged as a tree structure in the left side of the *Diagnostics* window.



Caution: Some commands in this menu are designed for advanced debugging and system testing only, and can cause the gateway to stop communicating, potentially resulting in data loss or other communication failures. Use these commands only if their potential effects are understood, or if specifically directed to do so by ProSoft Technology Technical Support engineers.

The following menu commands are shown below:

Menu Command	Submenu Command	Description
Module	Version	Displays the gateway's current software version and other important values.
	Config	Displays the status information of the gateway's Ethernet ports, enabled protocols, NTP server, and OPC UA server.
	Data Map	Displays the gateway's Data Map configuration.
Modbus Database View	ASCII	Displays the contents of the gateway's database in ASCII character format.*
	Decimal	Displays the contents of the gateway's database in decimal number format.*
	Hex	Displays the contents of the gateway's database in hexadecimal number format.*
	Float	Displays the contents of the gateway's database in floating-point number format.*

*Use the scroll bar on the right edge of the window to navigate through the database. Each page displays 100 words of data. The total number of pages available depends on the gateway's configuration.

3.2.3 Capturing a Diagnostic Session to a Log File

A Diagnostics session can be captured to a log file. This feature can be useful for troubleshooting and record-keeping purposes, and for communication with ProSoft Technology's Technical Support team.

1. Open a *Diagnostics* window. For more information, see *3.2 Using Diagnostics in ProSoft Configuration Builder*.
2. To log a Diagnostics session to a text file, from the toolbar, click the **LOG FILE** button. Click the button again to stop the capture.

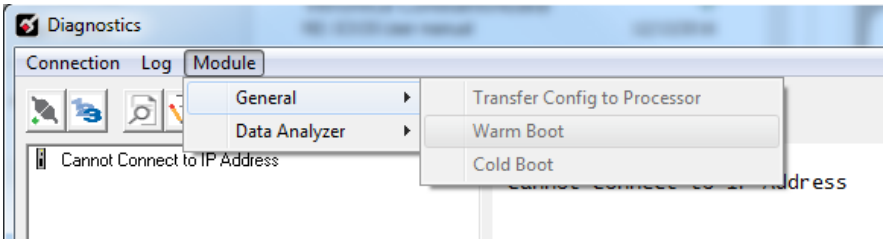
3. To view the log file, from the toolbar, click the **VIEW LOG FILE** button. The log file opens as a text file. It can be renamed and saved to a different location.

4. To email the log file to ProSoft Technology's Technical Support team, from the toolbar, click the **EMAIL LOG FILE** button. This only works if Microsoft Outlook is installed on the PC.)

5. If multiple sequential sessions are captured, PCB appends the new data to the end of the previously captured data. To clear the previous data from the log file, click the **CLEAR DATA** button before the data capture.

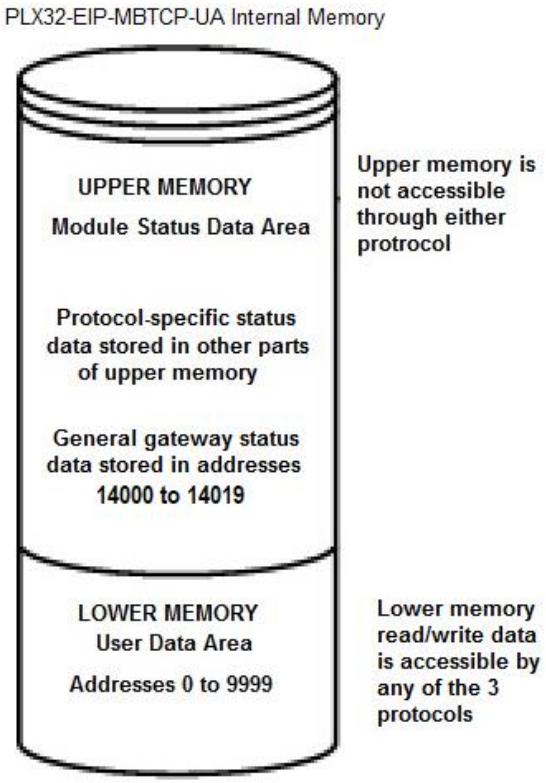
3.2.4 Warm Boot / Cold Boot

Warm and Cold booting the PLX32-EIP-MBTCP-UA can be done by clicking **MODULE** > **GENERAL** > **WARM BOOT** or **COLD BOOT**.



3.3 Gateway Status Data in Upper Memory

The gateway writes useful status data in dedicated upper memory locations in its internal database. The location of this status data area depends on the protocols supported by the gateway. Use the Data Map function in Prosoft Configuration Builder to map this data into the user data area of the gateway's database (registers 0 through 9999). Remote devices, such as HMIs or processors can then access the status data. For more information, see section *2.7 Mapping Data in PLX32-EIP-MBTCP-UA Memory*.



3.3.1 General Gateway Status Data in Upper Memory

The following table describes the contents of the gateway's general status data area.

Register Address	Description
14000 through 14001	Program Cycle Counter
14002 through 14004	Product Code (ASCII)
14005 through 14009	Product Revision (ASCII)
14010 through 14014	Operating System Revision (ASCII)
14015 through 14019	OS Run Number (ASCII)

3.3.2 Protocol-Specific Status Data in Upper Memory

The PLX32-EIP-MBTCP-UA also has upper memory locations for protocol-specific status data. The location of the status data area for the gateway protocol drivers depend on the protocols. For more information, see:

- Section *5.3.2 EIP Status Data in Upper Memory*
- Section *6.3.2 MBTCP Status Data in Upper Memory*

4 Hardware Information

4.1 Hardware Specifications

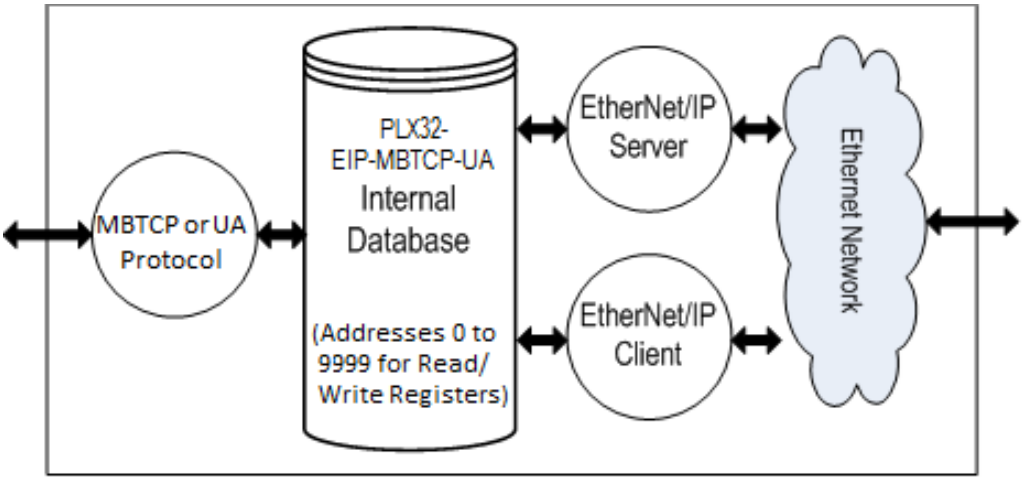
Specification	Description
Power Supply	24 VDC nominal 10 to 36 VDC allowed Positive, Negative, GND Terminals
Current Load	24 VDC nominal @ 300 mA 10 to 36 VDC @ 610 mA maximum
Operating Temperature	-25°C to 70°C (-13°F to 158°F)
Storage Temperature	-40°C to 80°C (-40°F to 176°F)
Relative Humidity	5% to 95% RH with no condensation
Dimensions (H x W x D)	5.38 x 1.99 x 4.38 in 13.67 x 5.05 x 11.13 cm
LED Indicators	Configuration (CFG) and Error (ERR) Communication Status Power (PWR) and Hardware Fault (FLT) Network Status (NS) EtherNet/IP™ Class I or Class III Connection Status (EtherNet/IP Only) Module Status (MS) Module Configuration Status (EtherNet/IP Only) Ethernet Communication Port Link/Activity and 100 mbit
Ethernet Port(s)	10/100 Mbit full-duplex RJ45 Connector Electrical Isolation 1500 Vrms at 50 Hz to 60 Hz for 60 seconds, applied as specified in section 5.3.2 of IEC 60950: 1991 Ethernet Broadcast Storm Resiliency = less than or equal to 5000 [ARP] frames-per-second and less than or equal to 5 minutes duration
Shipped With Each Unit	2.5 mm screwdriver J180 Power Connector

5 EIP Protocol

5.1 EIP Functional Overview

Use the PLX32-EIP-MBTCP-UA to interface many different protocols into the Rockwell Automation family of processors, or other software-based solutions.

The following illustration shows the functionality of the EtherNet/IP protocol.



5.1.1 EtherNet/IP General Specifications

The EIP driver supports the following connections:

Class	Connection Type	Number of Connections
Class 1	I/O	2
Class 3	Connected Client	2
	Unconnected Client	1
	Server	5

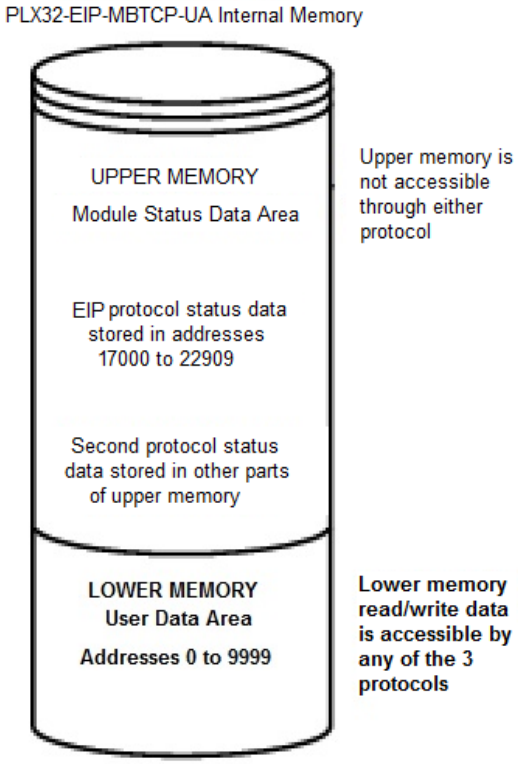
Specification	Description
Supported PLC Types	PLC2, PLC5, SLC, CLX, CMPLX, MICROLX
Supported Message Types	PCCC and CIP
I/O connection sizes in/out	496/496 bytes
Max RPI time	5 ms per connection
CIP Services Supported	0x4C: CIP Data Table Read 0x4D: CIP Data Table Write CIP Generic
Command List	Supports up to 100 commands per client. Each command is configurable for command type, IP address, register to/from address, and word/bit count.
Command Sets	PLC-2/PLC-3/PLC5 Basic Command Set PLC5 Binary Command Set PLC5 ASCII Command Set SLC500 Command Set

5.1.2 EIP Internal Database

The internal database is central to the functionality of the PLX32-EIP-MBTCP-UA. The gateway shares this database between all the communications ports on the gateway and uses it as a conduit to pass information from one protocol to another device on one network to one or more devices on another network. This permits data from devices on one communication port to be accessed and controlled by devices on another protocol.

In addition to data from the client and server, status and error information generated by the gateway can be mapped into the user data area of the internal database. The internal database is divided into two areas:

- Upper memory for the gateway status data area. This is where the gateway writes internal status data for the protocols supported by the gateway.
- Lower memory for the user data area. This is where incoming data from external devices is stored and accessed.



Each protocol in the PLX32-EIP-MBTCP-UA can write data to and read data from the user data area.

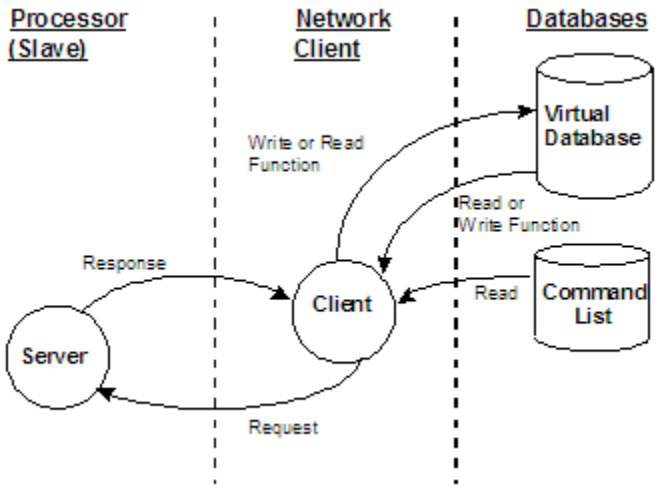
Note: To access gateway status data in the upper memory, use the data mapping feature in the gateway to copy data from the gateway status data area to the user data area. For more information, see section *2.7 Mapping Data in PLX32-EIP-MBTCP-UA Memory*.

Otherwise, use the diagnostic functions in ProSoft Configuration Builder to view gateway status data. For more information on the gateway status data, see section *5.3 Network Diagnostics*.

5.1.2.1 EIP Client Access to Database

The client functionality exchanges data between the gateway's internal database and data tables established in one or more processors or other server based devices. The command list that defined in ProSoft Configuration Builder specifies what data is to be transferred between the gateway and each of the servers on the network. No ladder logic is required in the processor (server) for client functionality, except to assure that sufficient data memory exists.

The following illustration describes the flow of data between the Ethernet clients and the internal database.



5.1.2.2 Multiple Server Access to EIP Database

Server support in the gateway allows client applications (such as HMI software and processors) to read from and write to the gateway's database. The server driver is able to support multiple concurrent connections from several clients.

When configured as a server, the user data area of the internal database in the gateway is the source for read requests and the destination for write requests from remote clients. Access to the database is controlled by the command type received in the incoming message from the client.

The gateway must be correctly configured and connected to the network before any attempt is made to use it. Use a network verification program, such as ProSoft Discovery Service or the command prompt PING instruction, to verify that the gateway can be seen on the network. Use ProSoft Configuration Builder to confirm proper configuration of the gateway and to transfer the configuration files to and from the gateway.

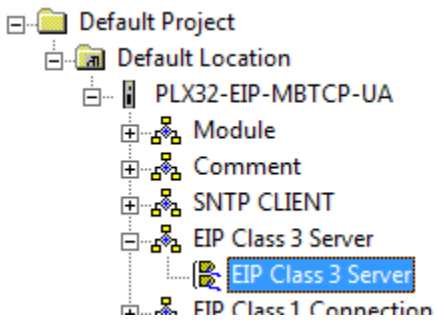
5.2 EIP Configuration

5.2.1 Configuring EIP Class 3 Server

Use the EIP Class 3 Server connection when the gateway is acting as a server (slave) device responding to message instructions initiated from a client (master) device such as an HMI, DCS, PLC, or PAC.

Setting the Server File Size

1. In ProSoft Configuration Builder, click the **[+]** next to the gateway, then click the **[+]** next to *EIP Class 3 Server*.



2. Double-click the second *EIP Class 3 Server* to display the *Edit - EIP Class 3 Server* dialog.
3. Select the **SERVER FILE SIZE** (100 or 1000).
   - For a value of 100, the registers are from N10:0 to N10:99.
   - For a value of 1000, the valid registers are from N10:0 to N10:999.

5.2.1.1 Accessing the Gateway's Internal Memory

The following table refers to the user data area in the gateway's memory:

Data Type	Tag Name	Length of Each Element in CIP Message	Array Range for 10,000 Element Database
BOOL	BOOLData[ ]	1	0 to 159999
Bit Array	BITAData[ ]	4	0 to 4999
SINT	SINTData[ ]	1	0 to 19999
INT	INT_Data[ ]	2	0 to 9999
DINT	DINTData[ ]	4	0 to 4999
REAL	REALData[ ]	4	0 to 4999

5.2.1.1.1 MSG Instruction Type - CIP

The following table defines the relationship of the user data area in the gateway's internal database to the addresses required in the MSG CIP instructions:

Data-base Address	CIP Integer	CIP Boolean	CIP Bit Array	CIP Byte	CIP DINT	CIP Real
0	Int_data [0]	BoolData[0]	BitAData[0]	SIntData[0]	DIntData[0]	RealData [0]
999	Int_data [999]	BoolData[15984]		SIntData[1998]		
1000	Int_data [1000]	BoolData[16000]	BitAData[500]	SIntData[2000]	DIntData[500]	RealData [500]
1999	Int_data [1999]	BoolData[31984]		SIntData[3998]		
2000	Int_data [2000]	BoolData[32000]	BitAData[1000]	SIntData[4000]	DIntData[1000]	RealData [1000]
2999	Int_data [2999]	BoolData[47984]		SIntData[5998]		
3000	Int_data [3000]	BoolData[48000]	BitAData[1500]	SIntData[6000]	DIntData[1500]	RealData [1500]
3999	Int_data [3999]	BoolData[63999]		SIntData[9998]		

5.2.1.1.2 MSG Instruction Type - PCCC

The following table defines the relationship of the user data area in the gateway's internal database to the addresses required in the MSG PCCC instructions:

Database Address	File size 100	Database Address	File size 100
0	N10:0	0	N10:0
999	N19:99	999	N19:99
1000	N20:0	1000	N20:0
1999	N29:99	1999	N29:99
2000	N30:0	2000	N30:0

5.2.1.2 EtherNet/IP Explicit Messaging Server Command Support

The PLX32-EIP-MBTCP-UA supports several command sets.

5.2.1.2.1 Basic Command Set Functions

Command	Function	Definition	Supported in Server
0x00	N/A	Protected Write	X
0x01	N/A	Unprotected Read	X
0x02	N/A	Protected Bit Write	X
0x05	N/A	Unprotected Bit Write	X
0x08	N/A	Unprotected Write	X

5.2.1.2.2 PLC-5 Command Set Functions

Command	Function	Definition	Supported in Server
0x0F	0x00	Word Range Write (Binary Address)	X
0x0F	0x01	Word Range Read (Binary Address)	X
0x0F		Typed Range Read (Binary Address)	X
0x0F		Typed Range Write (Binary Address)	X
0x0F	0x26	Read-Modify-Write (Binary Address)	
0x0F	0x00	Word Range Write (ASCII Address)	X
0x0F	0x01	Word Range Read (ASCII Address)	X
0x0F	0x26	Read-Modify-Write (ASCII Address)	

5.2.1.2.3 SLC-500 Command Set Functions

Command	Function	Definition	Supported in Server
0x0F	0xA1	Protected Typed Logical Read With Two Address Fields	X
0x0F	0xA2	Protected Typed Logical Read With Three Address Fields	X
0x0F	0xA9	Protected Typed Logical Write With Two Address Fields	X
0x0F	0xAA	Protected Typed Logical Write With Three Address Fields	X
0x0F	0xAB	Protected Typed Logical Write With Mask (Three Address Fields)	

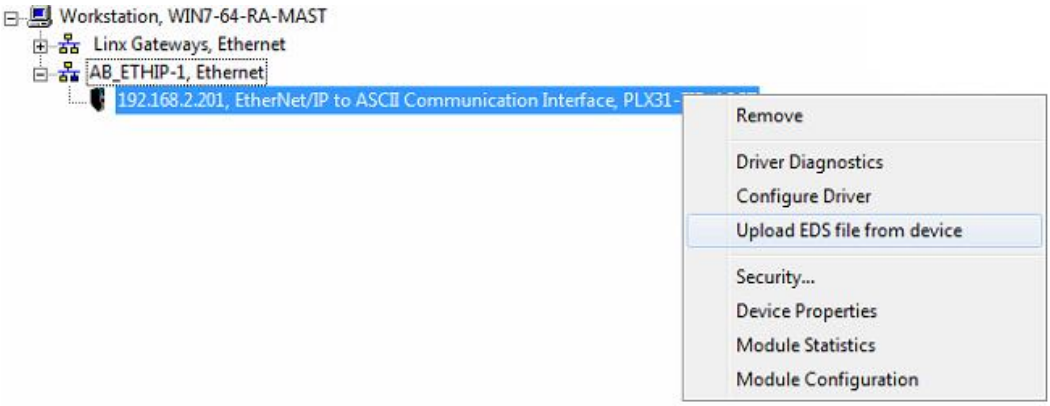
5.2.2 Configuring EIP Class 1 Connection

Use the EIP Class 1 Connection when the gateway acts as an EIP adapter transferring data to and from a PLC (the EIP scanner) using a direct I/O connection. Direct I/O connections can transfer large amounts of data quickly.

The PLX32-EIP-MBTCP-UA can handle up to eight I/O connections (depending on the model), each with 248 words of input data and 248 words of output data.

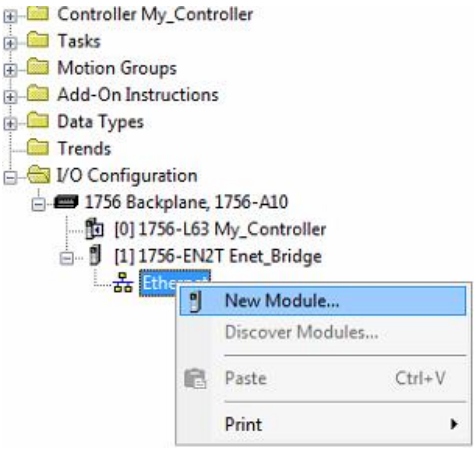
5.2.2.1 Adding the Gateway to RSLogix5000 v20

1. Start Rockwell Automation RSLinx and browse to the PLX32-EIP-MBTCP-UA.
2. Right-click the gateway and then choose **UPLOAD EDS FROM DEVICE**.



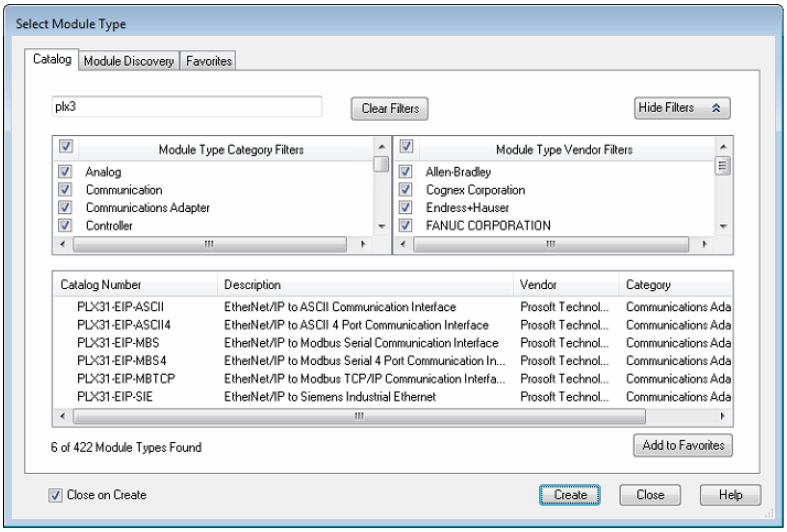
Note: RSLogix5000 may need to be restarted to complete the EDS installation.

3. After RSLogix 5000 is restarted, open the desired RSLogix 5000 project.
4. In the *Controller Organizer*, right-click the EtherNet/IP bridge in the I/O tree and choose **NEW MODULE**.

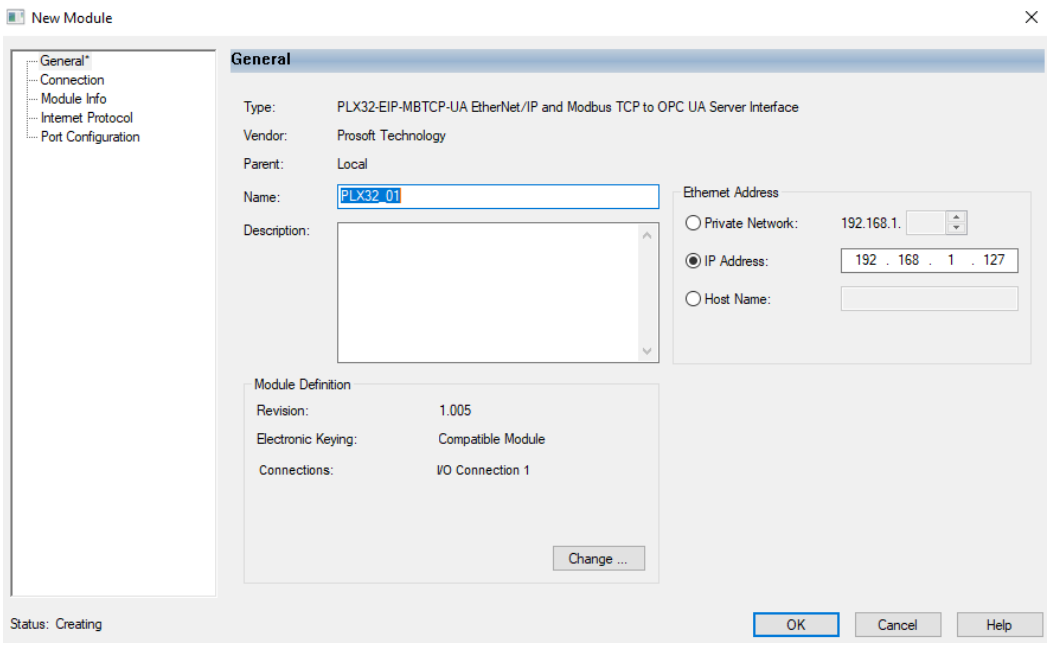


5 In the *Select Module Type* dialog, in the *Enter search text* box, type **PLX3**.

6 Select the **PLX32-EIP-MBTCP-UA**, and then click **CREATE**.

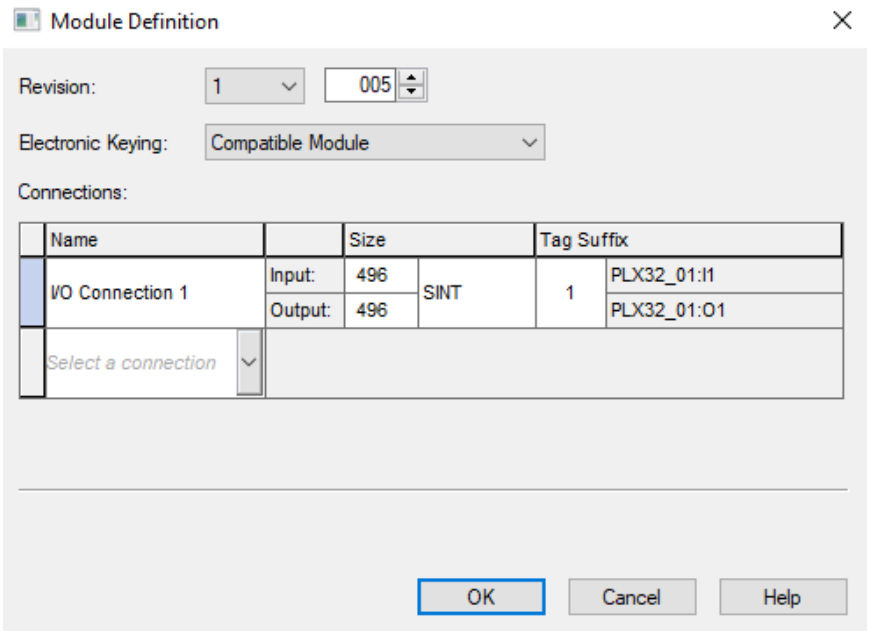


7 In the *New Module* dialog, enter a name for the gateway, then enter the IP address of the PLX32-EIP-MBTCP-UA.



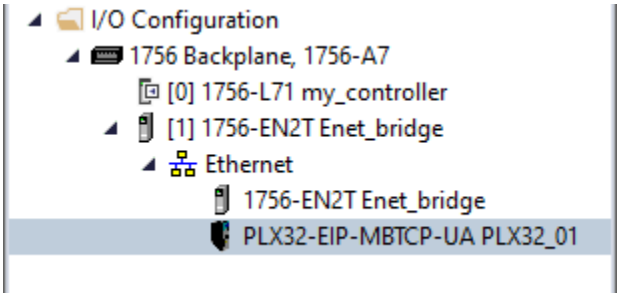
8 To add I/O connections click **CHANGE.**

9 In the *Module Definition* dialog, enter the I/O connections. Up to eight I/O connections can be added. The I/O connections have a fixed size of **496** bytes of input data and **496** bytes of output data. Click **OK**.



10 In the *Module Properties* dialog, click the **CONNECTION** tab to configure each I/O connection with its own RPI time. Click **OK**.

11 The new gateway appears in the *Controller Organizer* under the EtherNet/IP bridge.



5.2.2.2 Adding the Gateway to RSLogix5000 v16 Through v19

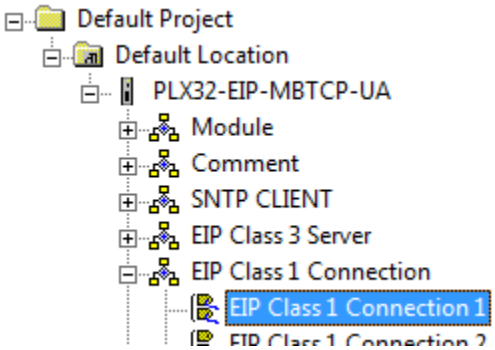
Note: Class 1 connections are not supported in RSLogix v15 and older

1. Start Rockwell Automation RSLogix 5000.
2. In the *Controller Organizer*, right-click the EtherNet/IP bridge in the I/O tree and choose **NEW MODULE**.
3. In the *Select Module Type* dialog, click **FIND.** Search for *Generic EtherNet Bridge*, click *Generic Ethernet Bridge*, and then click **CREATE**.
4. In the *New Module* dialog, enter a name for the gateway, then enter the IP address of the PLX32-EIP-MBTCP-UA. This creates the communication path from the processor to the PLX32-EIP-MBTCP-UA.
5. Add a new module under the *Generic EtherNet Bridge* and add a CIP Connection (*CIP-MODULE*). Specify the parameters for the I/O connection. The input and output sizes need to match the input and output sizes configured in PCB. The **ADDRESS** field value represents the connection number in PCB. By default all of the connections have 248 Input words, 248 Output words, and 0 Configuration words. Set the Comm format to Data type INT, and set the Assembly instances to be "1" for input, "2" for output, and "4" for configuration.
6. Add and configure a CIP Connection for each I/O connection.

5.2.2.3 Configuring EIP Class 1 Connections in PCB

After the PLX32-EIP-MBTCP-UA has been configured in RSLogix 5000, configure the connections in the gateway.

1. In *ProSoft Configuration Builder*, click the **[+]** next to the gateway, then click the **[+]** next to *EIP Class 1 Connection [x]*.



2. Double-click the *EIP Class 1 Connection [x]* to display the *Edit - EIP Class 1 Connection [x]* dialog.
3. In the dialog, click a parameter and then enter a value for the parameter. There are four configurable parameters for each I/O connection in ProSoft Configuration Builder.

Parameter	Value Range	Description
Input Data Address	0 to 9999	Specifies the starting address within the gateway's virtual database for data transferred from the gateway to the PLC.
Input Size	0 to 248	Specifies the number of Integers being transferred to the PLC's input image (248 integers max).
Output Data Address	0 to 9999	Specifies the starting address within the gateway's virtual database for data transferred from the PLC to the gateway.
Output Size	0 to 248	Specifies the number of integers being transferred to the PLC's output image (248 integers max).

5.2.3 Configuring EIP Class 3 Client[x]/UClient Connection

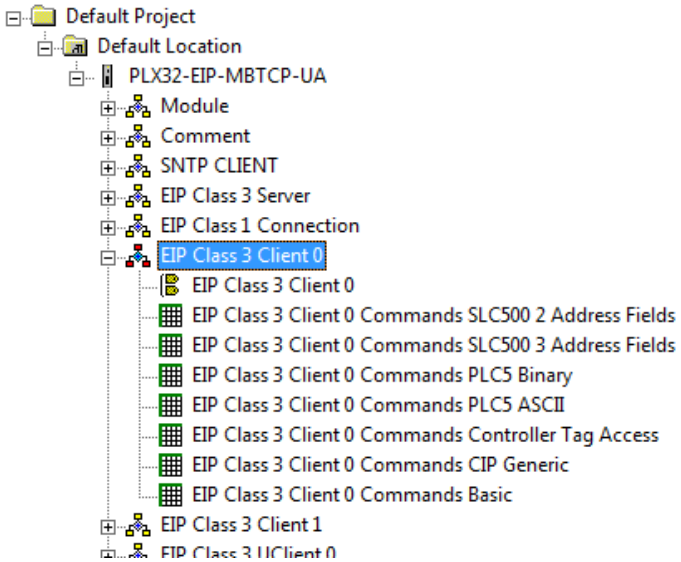
The PLX32-EIP-MBTCP-UA supports two connected clients and one unconnected client (most devices use connected clients; be sure refer to the user manual for the target device for verification).

- Use the *EIP Class 3 Client [x]* connections when the gateway is acting as a client/master initiating message instructions to the server/slave devices. The PLX32-EIP-MBTCP-UA EIP protocol supports three connected client connections. Typical applications include SCADA systems, and SLC communication.
- Use the *EIP Class 3 UClient* connection when the gateway is acting as a client/master initiating message instructions to the server/slave devices. The PLX32-EIP-MBTCP-UA EIP protocol supports one unconnected client connection. Unconnected messaging is a type of EtherNet/IP explicit messaging that uses TCP/IP implementation. Certain devices, such as the AB Power Monitor 3000 series B, support unconnected messaging. Check the device documentation for further information about its EtherNet/IP implementation.

5.2.3.1 Class 3 Client[x]/UClient

To configure Class 3 Client/UClient [x] connections:

1 In ProSoft Configuration Builder, click the **[+]** next to the gateway, then click the **[+]** next to *EIP Class 3 Client [x]* or *EIP Class 3 UClient [x]*.



2 Double-click the second *EIP Class 3 Client [x]* to display the *Edit - EIP Class 3 Client [x]* dialog.

3 In the dialog, click any parameter to change its value.

The following table specifies the configuration for the EIP client (master) device on the network port:

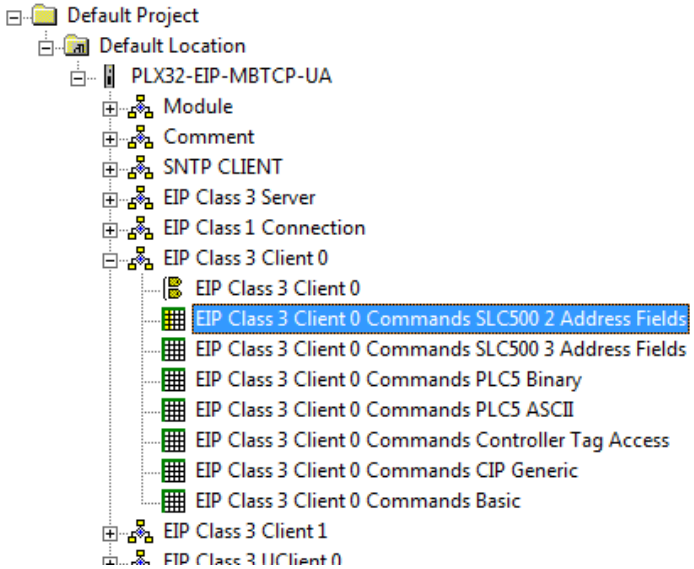
Parameter	Value	Description
Minimum Command Delay	0 to 65535 milliseconds	Specifies the number of milliseconds to wait between the initial issuances of a command. This parameter can be used to delay all commands sent to servers to avoid "flooding" commands on the network. This parameter does not affect retries of a command as they will be issued when failure is recognized.
Response Timeout	0 to 65535 milliseconds	Specifies the amount of time in milliseconds that a Client will wait before re-transmitting a command if no response is received from the addressed server. The value to use depends on the type of communication network used, and the expected response time of the slowest device connected to the network.
Retry Count	0 to 10	Specifies the number of times a command will be retried if it fails.

5.2.3.2 Class 3 Client[x]/UClient Commands

There is a separate command list for each of the different message types supported by the protocol. Each list is processed from top to bottom, one after the other, until all specified commands are completed, and then the polling process begins again.

This section defines the EtherNet/IP commands to be issued from the gateway to server devices on the network. Use these commands for data collection and control of devices on the TCP/IP network. In order to interface the virtual database with Rockwell Automation Programmable Automation Controllers (PACs), Programmable Logic Controllers (PLCs), or other EtherNet/IP server devices, construct a command list using the command list parameters for each message type.

1. In ProSoft Configuration Builder, click the **[+]** next to the gateway, then click the **[+]** next to *EIP Class 3 Client [x]* or *EIP Class 3 UClient [x]*.



2. Double-click the desired command type to display the *Edit - EIP Class 3 Client [x] Commands* or *Edit - EIP Class 3 UClient [x] Commands* dialog.
3. Click **ADD ROW** to add a new command.
4. Click **EDIT ROW** or double-click the row to display the *Edit* dialog to configure the command.

5.2.3.2.1 Class 3 Client/UClient [x] Commands SLC500 2 Address Fields

Parameter	Value	Description
Enable	Enable Disable Conditional Write	Specifies if the command should be executed and under what conditions. **ENABLE** - The Command is executed each scan of the command list **DISABLE** - The command is disabled and will not be executed **CONDITIONAL WRITE** - The Command executes only if the internal data associated with the command changes
Internal Address	0 to 9999	Specifies the database address in the gateway's internal database to be associated with the command. If the command is a read function, the data received in the response message is placed at the specified location. If the command is a write function data used in the command is sourced from specified data area.
Poll Interval	0 to 65535	Specifies the minimum interval to execute continuous commands. The parameter is entered in 1/10 of a second. If a value of 100 is entered for a command, the command executes no more frequently than every 10 seconds.
Reg Count	0 to 125	Specifies the number of data points to be read from or written to the target device.
Swap Code	None Word swap Word and Byte swap Byte swap	Specifies if the data from the server is to be ordered differently than it was received. This parameter is typically used when dealing with floating-point or other multi-register values. **NONE** - No change is made (abcd) **WORD SWAP** - The words are swapped (cdab) **WORD AND BYTE SWAP** - The words and bytes are swapped (dcba) **BYTE SWAP** - The bytes are swapped (badc)
IP Address	xxx.xxx.xxx.xxx	Specifies the IP address of the target device to be addressed.
Slot	-1	Specifies the slot number for the device. Use a value of -1 when interfacing to an SLC 5/05. These devices do not have a slot parameter. When addressing a processor in a CLX or CMPLX rack, the slot number corresponds to the slot containing the controller being addressed.
Func Code	501 509	Specifies the function code to be used in the command. **501** - Protected Typed Read **509** - Protected Typed Write
File Type	Binary Counter Timer Control Integer Float ASCII String Status	Specifies the file type to be associated with the command.
File Number	-1	Specifies the PLC-5 file number to be associated with the command. If a value of -1 is entered for the parameter, the field will not be used in the command, and the default file will be used.
Element Number		Specifies the element in the file where the command will start.
Comment		Optional 32 character comment for the command.

5.2.3.2.2 Class 3 Client[x]/UClient Commands SLC500 3 Address Fields

This command is typically used when accessing data in a Timer or Counter.
i.e. **T.1.1.2** is the address of the accumulator in Timer 1.

Parameter	Value	Description
Enable	Enable Disable Conditional Write	Specifies if the command should be executed and under what conditions. **ENABLE** - The Command is executed each scan of the command list **DISABLE** - The command is disabled and will not be executed **CONDITIONAL WRITE** - The Command executes only if the internal data associated with the command changes
Internal Address	0 to 9999	Specifies the database address in the gateway's internal database to be associated with the command. If the command is a read function, the data received in the response message is placed at the specified location. If the command is a write function data used in the command is sourced from specified data area.
Poll Interval	0 to 65535	Specifies the minimum interval to execute continuous commands. The parameter is entered in 1/10 of a second. If a value of 100 is entered for a command, the command executes no more frequently than every 10 seconds.
Reg Count	0 to 125	Specifies the number of data points to be read from or written to the target device.
Swap Code	None Word swap Word and Byte swap Byte swap	Specifies if the data from the server is to be ordered differently than it was received. This parameter is typically used when dealing with floating-point or other multi-register values. **NONE** - No change is made (abcd) **WORD SWAP** - The words are swapped (cdab) **WORD AND BYTE SWAP** - The words and bytes are swapped (dcba) **Byte swap** - The bytes are swapped (badc)
IP Address	xxx.xxx.xxx.xxx	Specifies the IP address of the target device to be addressed by this command.
Slot	-1	Specifies the slot number for the device. Use a value of -1 when interfacing to an SLC 5/05. These devices do not have a slot parameter. When addressing a processor in a ControlLogix or CompactLogix, the slot number corresponds to the slot in the rack containing the controller being addressed.
Func Code	502 510 511	Specifies the function code to be used in the command. **502** - Protected Typed Read **510** - Protected Typed Write **511** - Protected Typed Write w/Mask
File Type	Binary Counter Timer Control Integer Float ASCII String Status	Specifies the file type to be associated with the command.
File Number	-1	Specifies the SLC 500 file number to be associated with the command. If a value of -1 is entered for the parameter, the field will not be used in the command, and the default file will be used.
Element Number		Specifies the element in the file where the command will start.
Sub Element		Specifies the sub-element to be used with the command. Refer to the AB documentation for a list of valid sub-element codes.
Comment		Optional 32-character comment for the command.

5.2.3.2.3 Class 3 Client[x]/UClient Commands PLC5 Binary

Parameter	Value	Description
Enable	Enable Disable Conditional Write	Specifies if the command should be executed and under what conditions. **ENABLE** - The Command is executed each scan of the command list **DISABLE** - The command is disabled and will not be executed **CONDITIONAL WRITE** - The Command executes only if the internal data associated with the command changes
Internal Address	0 to 9999	Specifies the database address in the gateway's internal database to be associated with the command. If the command is a read function, the data received in the response message is placed at the specified location. If the command is a write function data used in the command is sourced from specified data area.
Poll Interval	0 to 65535	Specifies the minimum interval to execute continuous commands. The parameter is entered in 1/10 of a second. If a value of 100 is entered for a command, the command executes no more frequently than every 10 seconds.
Reg Count	0 to 125	Specifies the number of data points to be read from or written to the target device.
Swap Code	None Word swap Word and Byte swap Byte swap	Specifies if the data from the server is to be ordered differently than it was received. This parameter is typically used when dealing with floating-point or other multi-register values. **NONE** - No change is made (abcd) **WORD SWAP** - The words are swapped (cdab) **WORD AND BYTE SWAP** - The words and bytes are swapped (dcba) **BYTE SWAP** - The bytes are swapped (badc)
IP Address	xxx.xxx.xxx.xxx	Specifies the IP address of the target device to be addressed by this command.
Slot	-1	Specifies the slot number for the device. Use a value of -1 when interfacing to a PLC5 These devices do not have a slot parameter. When addressing a processor in a ControlLogix or CompactLogix, the slot number corresponds to the slot in the rack containing the controller being addressed.
Func Code	100 101 102	Specifies the function code to be used in the command. **100** - Word Range Write **101** - Word Range Read **102** - Read-Modify-Write
File Number	-1	Specifies the PLC5 file number to be associated with the command. If a value of -1 is entered for the parameter, the field will not be used in the command, and the default file will be used.
Element Number		Specifies the element in the file where the command will start.
Sub Element		Specifies the sub-element to be used with the command. Refer to the AB documentation for a list of valid sub-element codes.
Comment		Optional 32-character comment for the command.

5.2.3.2.4 Class 3 Client[x]/UClient Commands PLC5 ASCII

Parameter	Value	Description
Enable	Enable Disable Conditional Write	Specifies if the command should be executed and under what conditions. **ENABLE** - The Command is executed each scan of the command list **DISABLE** - The command is disabled and will not be executed **CONDITIONAL WRITE** - The Command executes only if the internal data associated with the command changes
Internal Address	0 to 9999	Specifies the database address in the gateway's internal database to be associated with the command. If the command is a read function, the data received in the response message is placed at the specified location. If the command is a write function data used in the command is sourced from specified data area.
Poll Interval	0 to 65535	Specifies the minimum interval to execute continuous commands. The parameter is entered in 1/10 of a second. If a value of 100 is entered for a command, the command executes no more frequently than every 10 seconds.
Reg Count	0 to 125	Specifies the number of data points to be read from or written to the target device.
Swap Code	None Word swap Word and Byte swap Byte swap	Specifies if the data from the server is to be ordered differently than it was received. This parameter is typically used when dealing with floating-point or other multi-register values. **NONE** - No change is made (abcd) **WORD SWAP** - The words are swapped (cdab) **WORD AND BYTE SWAP** - The words and bytes are swapped (dcba) **BYTE SWAP** - The bytes are swapped (badc)
IP Address	xxx.xxx.xxx.xxx	Specifies IP address of the target device to be addressed by this command.
Slot	-1	Specifies the slot number for the device. Use a value of -1 when interfacing to a PLC5 These devices do not have a slot parameter. When addressing a processor in a ControlLogix or CompactLogix, the slot number corresponds to the slot in the rack containing the controller being addressed.
Func Code	150 151 152	Specifies the function code to be used in the command. **150** - Word Range Write **151** - Word Range Read **152** - Read-Modify-Write
File String		Specifies the PLC-5 Address as a string. For example, N10:300
Comment		Optional 32-character comment for the command.

5.2.3.2.5 Class 3 Client[x]/UClient Commands Controller Tag Access

Parameter	Value	Description
Enable	Enable Disable Conditional Write	Specifies if the command should be executed and under what conditions. **ENABLE** - The Command is executed each scan of the command list **DISABLE** - The command is disabled and will not be executed **CONDITIONAL WRITE** - The Command executes only if the internal data associated with the command changes
Internal Address	0 to 9999	Specifies the database address in the gateway's internal database to be associated with the command. If the command is a read function, the data received in the response message is placed at the specified location. If the command is a write function data used in the command is sourced from specified data area.
Poll Interval	0 to 65535	Specifies the minimum interval to execute continuous commands. The parameter is entered in 1/10 of a second. If a value of 100 is entered for a command, the command executes no more frequently than every 10 seconds.
Reg Count	0 to 125	Specifies the number of data points to be read from or written to the target device.
Swap Code	None Word swap Word and Byte swap Byte swap	Specifies if the data from the server is to be ordered differently than it was received. This parameter is typically used when dealing with floating-point or other multi-register values. **NONE** - No change is made (abcd) **WORD SWAP** - The words are swapped (cdab) **WORD AND BYTE SWAP** - The words and bytes are swapped (dcba) **BYTE SWAP** - The bytes are swapped (badc)
IP Address	xxx.xxx.xxx.xxx	Specifies the IP address of the target device to be addressed by this command.
Slot	-1	Specifies the slot number for the device. Use a value of -1 when interfacing to a PLC5 These devices do not have a slot parameter. When addressing a processor in a ControlLogix or CompactLogix, the slot number corresponds to the slot in the rack containing the controller being addressed.
Func Code	332 333	Specifies the function code to be used in the command. **332** - CIP Data Table Read **333** - CIP Data Table Write
Data Type	Bool SINT INT DINT REAL DWORD	Specifies the data type of the target controller tag name.
Tag Name		Specifies the controller tag in the target PLC.
Offset	0 to 65535	Specifies the offset database where the value corresponds to the Tag Name parameter
Comment		Optional 32-character comment for the command.

5.2.3.2.6 Class 3 Client[x]/UClient Commands CIP Generic

Parameter	Value	Description
Enable	Disabled Enabled Conditional Write	Specifies the condition to execute the command. **DISABLED** - The command is disabled and will not be executed. **ENABLED** - The command is executed on each scan of the command list if the *Poll Interval* is set to zero. If the *Poll Interval* is non-zero, the command is executed when the interval timer expires. **CONDITIONAL WRITE** - The command executes only if the internal data value(s) to be sent has changed.
Internal Address	0 to 9999	Specifies the database address in the gateway's internal database to be associated with the command. If the command is a read function, the data received in the response message is placed at the specified location. If the command is a write function, data used in the command is sourced from specified data area.
Poll Interval	0 to 65535	Specifies the minimum interval to execute continuous commands. The parameter is entered in 1/10 of a second. For example, if a value of '100' is entered for a command, the command executes no more frequently than every 10 seconds.
Reg Count	0 to 125	Specifies the number of data points to read/write to the target device.
Swap Code	None Word swap Word and Byte swap Byte swap	Specifies if the data from the server is to be ordered differently than it was received. This parameter is typically used when dealing with floating-point or other multi-register values. **NONE** - No change is made (abcd) **WORD SWAP** - The words are swapped (cdab) **WORD AND BYTE SWAP** - The words and bytes are swapped (dcba) **BYTE SWAP** - The bytes are swapped (badc)
IP Address	xxx.xxx.xxx.xxx	Specifies the IP address of the target device to be addressed by this command.
Slot	-1	Use '-1' to target a connected device. Use > -1 to target a device in a specific slot number within the rack.
Func Code	CIP Generic	Used to read/write the attributes of any object by using an explicit address
Service Code	00 to FF (Hex)	An integer identification value which denotes a particular Object Instance and/or Object class function. For more information refer to ODVA CIP specification.
Class	00 to FFFF (Hex)	An integer identification value assigned to each Object Class accessible from the network. For more information, refer to ODVA CIP specification.
Instance	Application-dependent	An integer identification value assigned to an Object Instance that identifies it among all Instances of the same Class. For more information, refer to ODVA CIP specification.
Attribute	00 to FFFF (Hex)	An integer identification value assigned to a Class and/or Instance Attribute. For more information, refer to ODVA CIP specification.
Comment		This field can be used to give a 32-character comment to the command. The ":" and "#" characters are reserved characters. It is strongly recommended not be use in the comment section.

Note: Due to the behavior of Connected Clients, please note the following:

- Multiple commands with different Class objects cannot be configured to the same device.
- Multiple commands with different Class objects cannot be configured to different devices.
- Configure multiple commands using the Get_Attribute_Single of the same Class and address different Attributes.
- If there are commands in any of the other command types (i.e. Controller Tag Access) and configure a CIP Generic command to the same device, it will not work due to the Connected Client having an active connection to a device. However, both Controller Tag Access and CIP Generic can be used if the target devices are different.
- To avoid any or all these scenarios, it is recommended to use the Unconnected Client to send commands to different devices, since these connections are reset/closed after each command is executed.

5.2.3.2.7 Class 3 Client[x]/UClient Commands Basic

Parameter	Value	Description
Enable	Enable Disable Conditional Write	Specifies if the command should be executed and under what conditions. **ENABLE** - The command is executed each scan of the command list **DISABLE** - The command is disabled and will not be executed **CONDITIONAL WRITE** - The command executes only if the internal data associated with the command changes
Internal Address	0 to 9999	Specifies the database address in the gateway's internal database to be associated with the command. If the command is a read function, the data received in the response message is placed at the specified location. If the command is a write function data used in the command is sourced from the specified data area.
Poll Interval	0 to 65535	Specifies the minimum interval to execute continuous commands. The parameter is entered in 1/10 of a second. If a value of 100 is entered for a command, the command executes no more frequently than every 10 seconds.
Reg Count	0 to 125	Specifies the number of data points to be read from or written to the target device.
Swap Code	None Word swap Word and Byte swap Byte swap	Specifies if the data from the server is to be ordered differently than it was received. This parameter is typically used when dealing with floating-point or other multi-register values. **NONE** - No change is made (abcd) **WORD SWAP** - The words are swapped (cdab) **WORD AND BYTE SWAP** - The words and bytes are swapped (dcba) **BYTE SWAP** - The bytes are swapped (badc)
IP Address	xxx.xxx.xxx.xxx	Specifies the IP address of the target device to be addressed by this command.
Slot	-1	Use a value of -1 when interfacing to an SLC 5/05. These devices do not have a slot parameter. When addressing a processor in a ControlLogix or CompactLogix, the slot number corresponds to the slot in the rack containing the controller being addressed.
Func Code	1 2 3 4 5	Specifies the function code to be used in the command. **1** - Protected Write **2** - Unprotected Read **3** - Protected Bit Write **4** - Unprotected Bit Write **5** - Unprotected Write
Word Address		Specifies the word address where to start the operation.
Comment		Optional 32-character comment for the command.

5.3 EIP Network Diagnostics

5.3.1 EIP PCB Diagnostics

The best way to troubleshoot the EIP driver is to use ProSoft Configuration Builder to access the diagnostic capabilities of the gateway.

The following table summarizes the status information available in PCB for the EIP driver:

Connection Type	Submenu Item	Description
EIP Class 1	Config	Configuration settings for Class 1 Connections.
	Status	Status of the Class 1 Connections. Displays any configuration error, as well as the number of Class 1 Connections.
EIP Class 3 Server	Config	Configuration settings for Class 3 Server Connections.
	Comm Status	Status information for each Class 3 Server Connection. Displays port numbers, IP addresses, socket status, and read and write counts.
EIP Class 3 Client/UClient [x]	Config	Configuration settings for Class 3 Client/UClient Connections.
	Comm Status	Status information for Class 3 Client/UClient [x] commands. Displays a summary of all the errors resulting from Class 3 Client/UClient [x] commands.
	Commands	Configuration for the Class 3 Client/UClient [x] command list.
	Cmd Errors (Decimal)	Current error codes for each command on the Class 3 Client/UClient [x] command list in decimal number format. A zero means there is currently no error for the command.
	Cmd Errors (Hex)	Current error codes for each command on the Class 3 Client/UClient [x] command list in hexadecimal number format. A zero means there is currently no error for the command.

For specific information on error codes, see section *5.3.3 EIP Error Codes*.

5.3.2 EIP Status Data in Upper Memory

The EIP driver has an associated status data area located in the upper memory of the PLX32-EIP-MBTCP-UA. The *Data Map* functionality of the PLX32-EIP-MBTCP-UA can be used to map this data into the normal user data range of the PLX32-EIP-MBTCP-UA database.

Note that all the status values are initialized to zero (0) at power-up, cold boot and during warm boot.

5.3.2.1 EIP Client Status Data

The following table lists the addresses in upper memory the PLX32-EIP-MBTCP-UA stores general error and status data for each EIP connected and unconnected client:

EIP Client	Address Range
Connected Client 0	17900 through 17909
Connected Client 1	18100 through 18109
Unconnected Client 0	22800 through 22809

The content of each client's status data area is structured in the same way. The following table describes the content of each register in the status data area:

Offset	Description
0	Number of Command Requests
1	Number of Command Responses
2	Number of Command Errors
3	Number of Requests
4	Number of Responses
5	Number of Errors Sent
6	Number of Errors Received
7	Reserved
8	Current Error Code
9	Last Error Code

5.3.2.2 EIP Client Command List Error Data

The PLX32-EIP-MBTCP-UA stores a status/error code in upper memory for each command in each EIP client's command list. The following table lists the addresses in upper memory where the gateway stores the command list error data for each EIP client:

EIP Client	Address Range
Connected client 0	17910 through 18009
Connected client 1	18110 through 18209
Unconnected client 0	22810 through 22909

The first word in each client's command list error data area contains the status/error code for the first command in the client's command list. Each successive word in the command error list is associated with the next command in the list. Therefore, the size of the command list error data area depends on the number of commands defined. The structure of the command list error data area (which is the same for all clients) is displayed in the following table:

Offset	Description
0	Command #1 Error Code
1	Command #2 Error Code
2	Command #3 Error Code
3	Command #4 Error Code
4	Command #5 Error Code
.	.
.	.
.	.
97	Command #98 Error Code
98	Command #99 Error Code
99	Command #100 Error Code

5.3.2.3 EIP Class 1 Server Status Data

The following table lists the addresses in upper memory where the PLX3x gateway stores the Open Connection Count for each EIP Class 1 server.

EIP Class 1 Server	Address Range	Description
	17000	Bit map of PLC State for each Connection 1 to 8. 0 = Run 1 = Program
1	17001	Open Connection Count for Connection 1
2	17002	Open Connection Count for Connection 2
3	17003	Open Connection Count for Connection 3
4	17004	Open Connection Count for Connection 4
5	17005	Open Connection Count for Connection 5
6	17006	Open Connection Count for Connection 6
7	17007	Open Connection Count for Connection 7
8	17008	Open Connection Count for Connection 8

5.3.2.4 EIP Class 3 Server Status Data

The following table lists the addresses in upper memory where the PLX32-EIP-MBTCP-UA stores status data for each EIP server:

EIP Server	Address Range
0	18900 through 18915
1	18916 through 18931
2	18932 through 18947
3	18948 through 18963
4	18964 through 18979

The content of each server's status data area is structured the same. The following table describes the content of each register in the status data area:

Offset	Description
0 through 1	Connection State
2 through 3	Open Connection Count
4 through 5	Socket Read Count
6 through 7	Socket Write Count
8 through 15	Peer IP

5.3.3 EIP Error Codes

The PLX32-EIP-MBTCP-UA stores error codes returned from the command list process in the command list error memory region. A word is allocated for each command in the memory area. The error codes are formatted in the word as follows: The least-significant byte of the word contains the extended status code and the most-significant byte contains the status code.

Use the error codes returned for each command in the list to determine the success or failure of the command. If the command fails, use the error code to determine the cause of failure.

Warning: The gateway-specific error codes (not EtherNet/IP/PCCC compliant) are returned from within the gateway and never returned from an attached EtherNet/IP/PCCC slave device. These are error codes that are part of the EtherNet/IP/PCCC protocol or are extended codes unique to the PLX32-EIP-MBTCP-UA. The most common EtherNet/IP/PCCC errors are shown below:

5.3.3.1 Local STS Error Codes

Code (Int)	Code (Hex)	Description
0	0x0000	Success, no error
256	0x0100	DST node is out of buffer space
512	0x0200	Cannot guarantee delivery (Link Layer)
768	0x0300	Duplicate token holder detected
1024	0x0400	Local port is disconnected
1280	0x0500	Application layer timed out waiting for response
1536	0x0600	Duplicate node detected
1792	0x0700	Station is offline
2048	0x0800	Hardware fault

5.3.3.2 Remote STS Error Codes

Code (Int)	Code (Hex)	Description
0	0x0000	Success, no error
4096	0x1000	Illegal command or format
8192	0x2000	Host has a problem and will not communicate
12288	0x3000	Remote node host is missing, disconnected, or shut down
16384	0x4000	Host could not complete function due to hardware fault
20480	0x5000	Addressing problem or memory protect rungs
24576	0x6000	Function not allowed due to command protection selection
26872	0x7000	Processor is in Program mode
-32768	0x8000	Compatibility mode file missing or communication zone problem
-28672	0x9000	Remote node cannot buffer command
-24576	0xA000	Wait ACK (1775-KA buffer full)
-20480	0xB000	Remote node problem due to download
-16384	0xC000	Wait ACK (1775-KA buffer full)
-12288	0xD000	Not used
-8192	0xE000	Not used
	0xF0nn	Error code in the EXT STS byte (nn contains EXT error code)

5.3.3.3 EXT STS Error Codes

Code (Int)	Code (Hex)	Description
-4096	0xF000	Not used
-4095	0xF001	A field has an illegal value
-4094	0xF002	Fewer levels specified in address than minimum for any address
-4093	0xF003	More levels specified in address than system supports
-4092	0xF004	Symbol not found
-4091	0xF005	Symbol is of improper format
-4090	0xF006	Address does not point to something usable
-4089	0xF007	File is wrong size
-4088	0xF008	Cannot complete request
-4087	0xF009	Data or file is too large
-4086	0xF00A	Transaction size plus word address is too large
-4085	0xF00B	Access denied, improper privilege
-4084	0xF00C	Condition cannot be generated - resource is not available
-4083	0xF00D	Condition already exists - resource is already available
-4082	0xF00E	Command cannot be executed
-4081	0xF00F	Histogram overflow
-4080	0xF010	No access
-4079	0xF011	Illegal data type
-4078	0xF012	Invalid parameter or invalid data
-4077	0xF013	Address reference exists to deleted area
-4076	0xF014	Command execution failure for unknown reason
-4075	0xF015	Data conversion error
-4074	0xF016	Scanner not able to communicate with 1771 rack adapter
-4073	0xF017	Type mismatch
-4072	0xF018	1171 Gateway response was not valid
-4071	0xF019	Duplicate label
-4070	0xF01A	File is open; another node owns it
-4069	0xF01B	Another node is the program owner
-4068	0xF01C	Reserved
-4067	0xF01D	Reserved
-4066	0xF01E	Data table element protection violation
-4065	0xF01F	Temporary internal problem

5.3.3.4 EIP Error Codes

Code (Int)	Code (Hex)	Description
-1	0xFFFF	CTS modem control line not set before transmitting
-2	0xFFFE	Timeout while transmitting message
-10	0xFFF6	Timeout waiting for DLE-ACK after request
-11	0xFFF5	Timeout waiting for response after request
-12	0xFFF4	Reply data does not match requested byte count
-20	0xFFEC	DLE-NAK received after request
-21	0xFFEB	DLE-NAK sent after response
-200	0xFF38	DLE-NAK received after request

5.3.3.5 TCP/IP Interface Error Codes

Error (Int)	Error (Hex)	Description
-33	0xFFDF	Failed to connect to target
-34	0xFFDE	Failed to register session with target (timeout)
-35	0xFFDD	Failed forward open response timeout
-36	0xFFDC	PCCC/Tag command response timeout
-37	0xFFDB	No TCP/IP connection error

5.3.3.6 Common Response Error Codes

Error (Int)	Error (Hex)	Description
-40	0xFFD8	Invalid response length
-41	0xFFD7	CPF item count is incorrect
-42	0xFFD6	CPF address field error
-43	0xFFD5	CPF packet tag invalid
-44	0xFFD4	CPF bad command code
-45	0xFFD3	CPF status error reported
-46	0xFFD2	CPF incorrect connection ID value returned
-47	0xFFD1	Context field not matched
-48	0xFFD0	Incorrect session handle returned
-49	0xFFCF	CPF not correct message number

5.3.3.7 Register Session Response Error Codes

Error (Int)	Error (Hex)	Description
-50	0xFFCE	Message length received not valid
-51	0xFFCD	Status error reported
-52	0xFFCC	Invalid version

5.3.3.8 Forward Open Response Error Codes

Error (Int)	Error (Hex)	Description
-55	0xFFC9	Message length received not valid
-56	0xFFC8	Status error reported

5.3.3.9 PCCC Response Error Codes

Error (Int)	Error (Hex)	Description
-61	0xFFC3	Message length received not valid
-62	0xFFC2	Status error reported
-63	0xFFC1	CPF bad command code
-64	0xFFC0	TNS in PCCC message not matched
-65	0xFFBF	Vendor ID in PCCC message not matched
-66	0xFFBE	Serial number in PCCC message not matched

5.4 EIP Reference

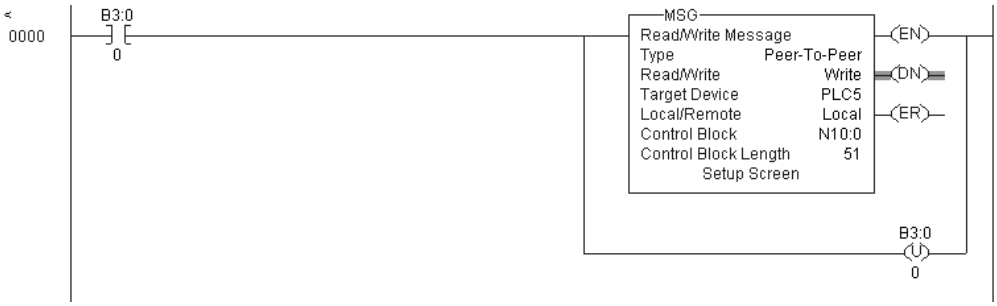
5.4.1 SLC and MicroLogix Specifics

5.4.1.1 Messaging from a SLC 5/05

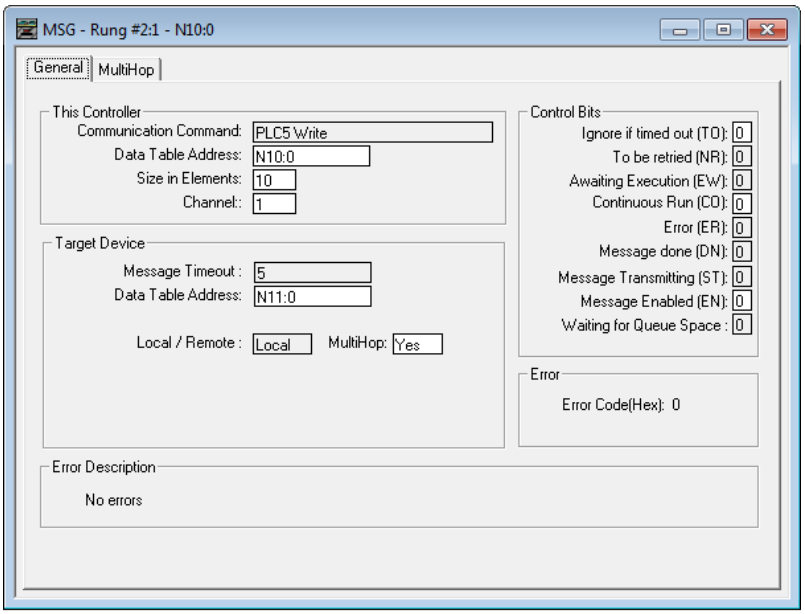
The PLX32-EIP-MBTCP-UA can receive messages from a SLC 5/05 containing an Ethernet interface. The gateway supports both read and write commands.

5.4.1.1.1 SLC5/05 Write Commands

Write commands transfer data from the SLC processor to the gateway. The following diagram shows an example rung to execute a write command.



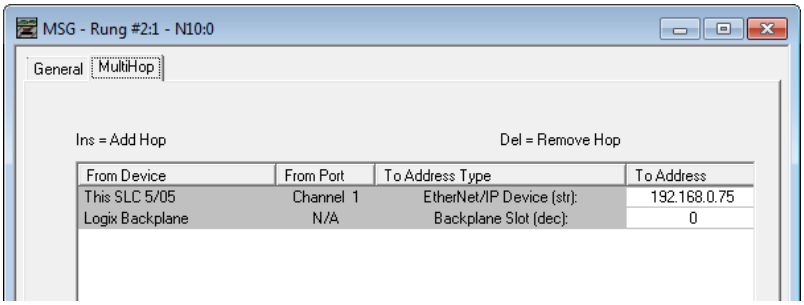
1. Set the **READ/WRITE** parameter to **WRITE.** The gateway supports a **TARGET DEVICE** parameter value of **500CPU** or **PLC5**.
2. In the MSG object, click **SETUP SCREEN** in the MSG object to complete the configuration of the MSG instruction. This displays the following dialog.



3 Set the **TARGET DEVICE DATA TABLE ADDRESS** to a valid file element (such as, N11:0) for SLC and PLC5 messages.

4 Set the **MULTIHOP** option to **YES**.

5 Complete the **MULTIHOP** tab portion of the dialog shown in the following image.

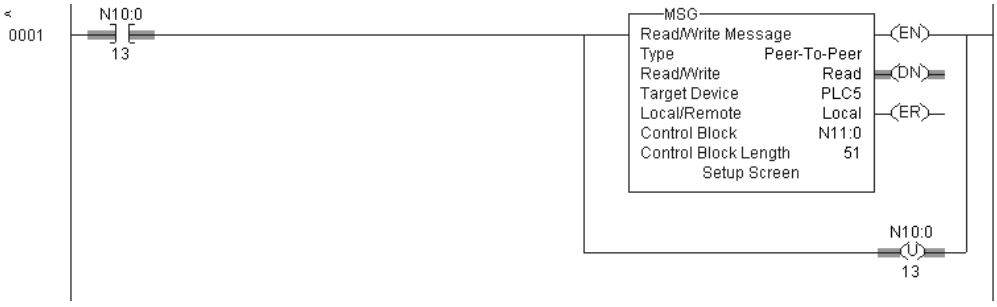


6 Set the **TO ADDRESS** value to the gateway's Ethernet IP address.

7 Press the **INS** key to add the second line for ControlLogix Backplane and set the slot number to **0**.

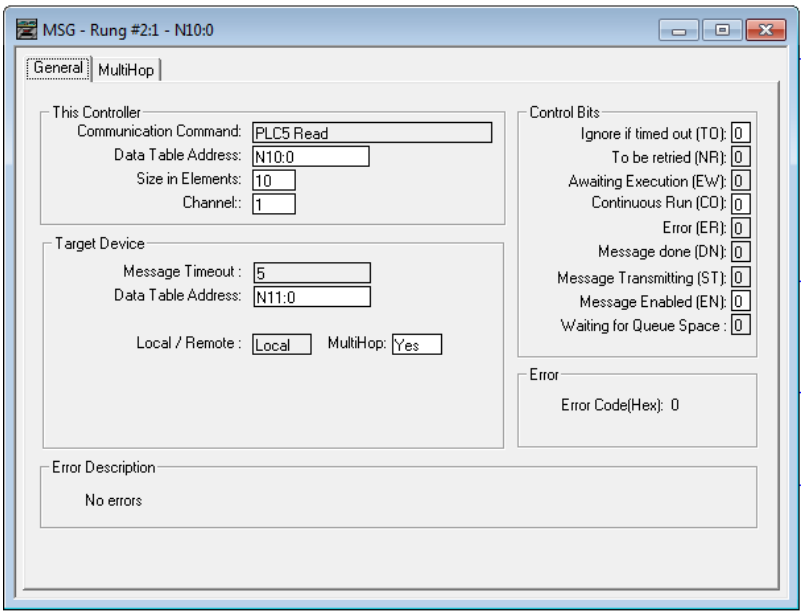
5.4.1.1.2 SLC5/05 Read Commands

Read commands transfer data to the SLC processor from the gateway. The following diagram shows an example rung to execute a read command.



1 Set the **READ/WRITE** parameter to **READ.** The gateway supports a **TARGET DEVICE** parameter value of **500CPU** or **PLC5.**

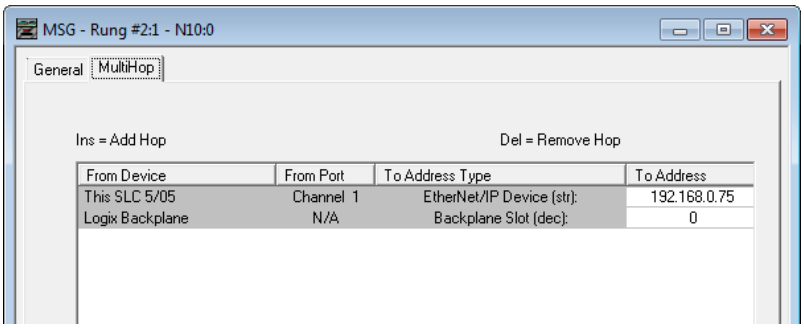
2 In the MSG object, click **SETUP SCREEN** in the MSG object to complete the configuration of the MSG instruction. This displays the following dialog.



3 Set the **TARGET DEVICE DATA TABLE ADDRESS** to a valid file element (such as, N11:0) for SLC and PLC5 messages.

4 Set the **MULTIHOP** option to **YES.**

5 Fill in the **MULTIHOP** tab portion of the dialog as shown in the following image.



6 Set the **TO ADDRESS** value to the gateway's Ethernet IP address.

7 Press the **INS** key to add the second line for ControlLogix Backplane and set the slot number to zero.

5.4.1.2 SLC File Types

This information is specific to the SLC and MicroLogix family or processors used with the PCCC command set. The SLC and MicroLogix processor commands support a file type field entered as a single character to denote the data table to use in the command. The following table defines the relationship of the file types accepted by the gateway and the SLC file types.

File Type	Description
S	Status
B	Bit
T	Timer
C	Counter
R	Control
N	Integer
F	Floating-point
Z	String
A	ASCII

The *File Type Command Code* is the ASCII character code value of the File Type letter. This is the value to enter for the **FILE TYPE** parameter of the PCCC Command configurations in the data tables in the ladder logic.

Additionally, the SLC specific functions (502, 510 and 511) support a sub-element field. This field selects a sub-element field in a complex data table. For example, to obtain the current accumulated value for a counter or timer, set the sub-element field to 2.

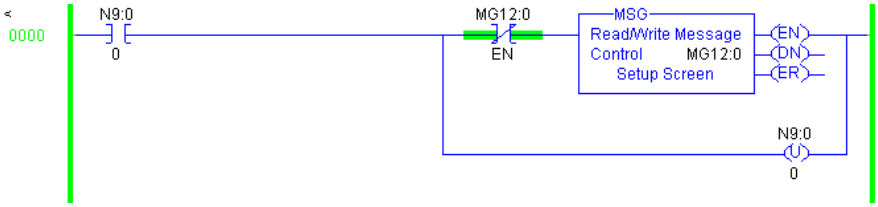
5.4.2 PLC5 Processor Specifics

5.4.2.1 Messaging from a PLC5

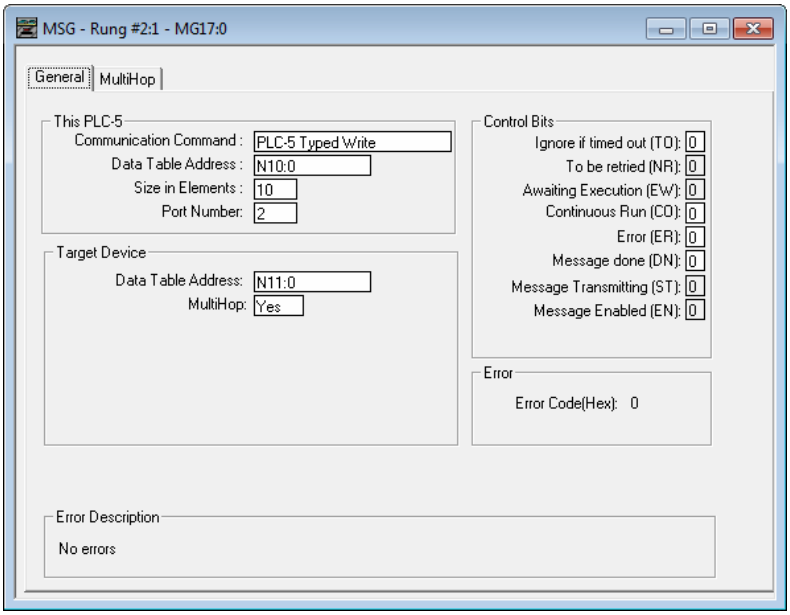
The gateway can receive messages from a PLC5 containing an Ethernet interface. The gateway supports both read and write commands.

5.4.2.1.1 PLC5 Write Commands

Write commands transfer data from the PLC5 processor to the gateway. The following diagram shows an example rung to execute a write command.



1 In the MSG object, click **SETUP SCREEN** in the MSG object to complete the configuration of the MSG instruction. This displays the following dialog.



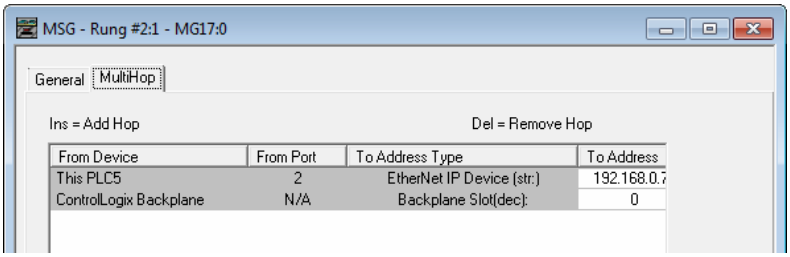
2 Select the **COMMUNICATION COMMAND** to execute from the following list of supported commands.

- PLC5 Type Write
- PLC2 Unprotected Write
- PLC5 Typed Write to PLC
- PLC Typed Logical Write

3 Set the **TARGET DEVICE DATA TABLE ADDRESS** to a valid file element (such as,N11:0) for SLC and PLC5 messages. For the PLC2 Unprotected Write message, set the address to the database index (such as 1000) for the command.

4 Set the **MULTIHOP** option to **YES**.

5 Complete **MULTIHOP** tab portion of the dialog as shown in the following image.

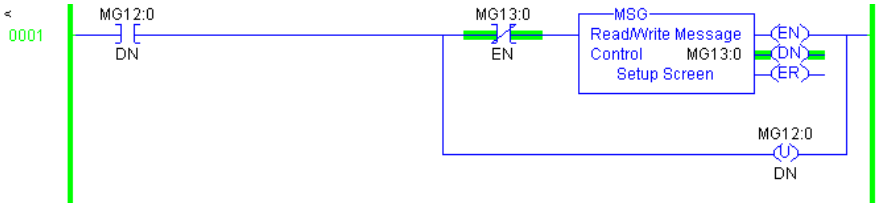


6 Set the **TO ADDRESS** value to the gateway's Ethernet IP address.

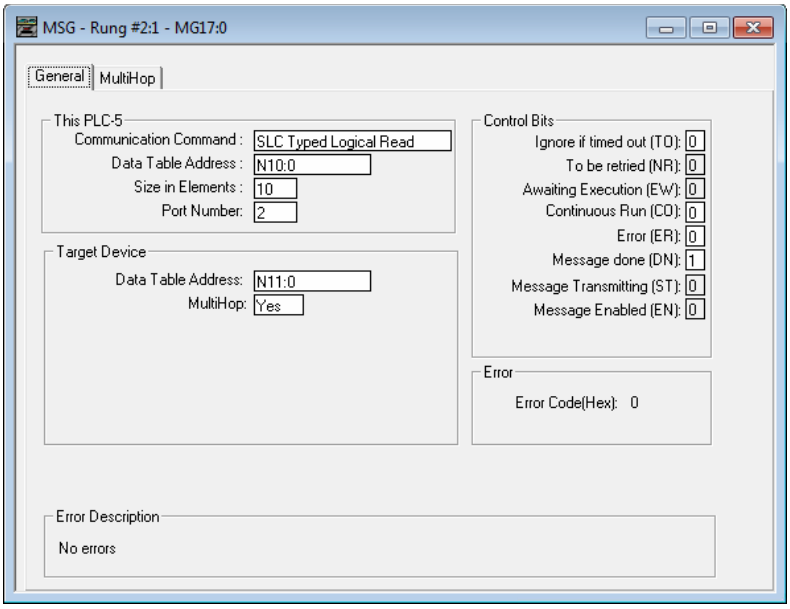
7 Press the **INS** key to add the second line for ControlLogix Backplane and set the slot number to **0**.

5.4.2.1.2 PLC5 Read Commands

Read commands transfer data to the PLC5 processor from the gateway. The following diagram shows an example rung that executes a read command.

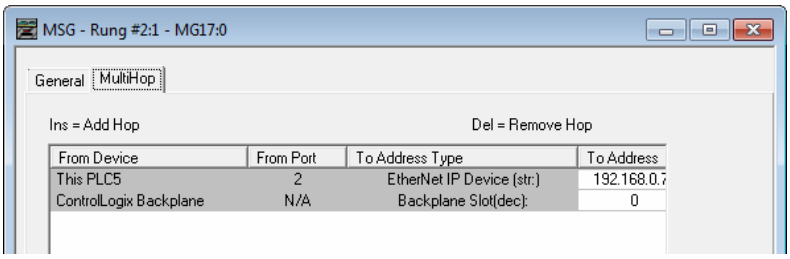


1. In the MSG object, click **SETUP SCREEN** in the MSG object to complete the configuration of the MSG instruction.



2. Select the **COMMUNICATION COMMAND** to execute from the following list of supported commands.
   - PLC5 Type Read
   - PLC2 Unprotected Read
   - PLC5 Typed Read to PLC
   - PLC Typed Logical Read
3. Set the **TARGET DEVICE DATA TABLE ADDRESS** to a valid file element (such as, N11:0) for SLC and PLC5 messages. For the PLC2 Unprotected Read message, set the address to the database index (such as, 1000) for the command.
4. Set the **MULTIHOP** option to **YES.**

5 Complete the **MULTIHOP** tab portion of the dialog as shown in the following image.



6 Set the **TO ADDRESS** value to the gateway's Ethernet IP address.

7 Press the **INS** key to add the second line for ControlLogix Backplane and set the slot number to **0**.

5.4.2.2 PLC-5 Sub-Element Fields

This section contains information specific to the PLC-5 processor when using the PCCC command set. The commands specific to the PLC-5 processor contain a sub-element code field. This field selects a sub-element field in a complex data table. For example, to obtain the current accumulated value for a counter or timer, set the sub-element field to 2.

The following tables contain the sub-element codes for PLC-5 complex data tables.

5.4.2.2.1 Timer / Counter

Code	Description
0	Control
1	Preset
2	Accumulated

5.4.2.2.2 Control

Code	Description
0	Control
1	Length
2	Position

5.4.2.2.3 PD

All PD values are floating point values, they are two words long.

Code	Description
0	Control
2	SP
4	Kp
6	Ki
8	Kd
26	PV

5.4.2.2.4 BT

Code	Description
0	Control
1	RLEN
2	DLEN
3	Data file #
4	Element #
5	Rack/Grp/Slot

5.4.2.2.5 MG

Code	Description
0	Control
1	Error
2	RLEN
3	DLEN

5.4.3 ControlLogix and CompactLogix Processor Specifics

5.4.3.1 Messaging from a ControlLogix or CompactLogix Processor

Use the MSG instruction to exchange data between a Control/CompactLogix processor and the gateway. There are two basic methods of data transfer supported by the gateway when using the MSG instruction:

- Encapsulated PCCC messages
- CIP Data Table messages.

5.4.3.2 Encapsulated PCCC Messages

This section contains information specific to the Control/CompactLogix processor when using the PCCC command set. The current implementation of the PCCC command set does not use functions that can directly access the Controller Tag Database. In order to access this database, use the table-mapping feature in RSLogix 5000. RSLogix 5000 permits assigning Controller Tag Arrays to virtual PLC 5 data tables. The PLX32-EIP-MBTCP-UA using the PLC 5 command set defined in this document can then access this controller data.

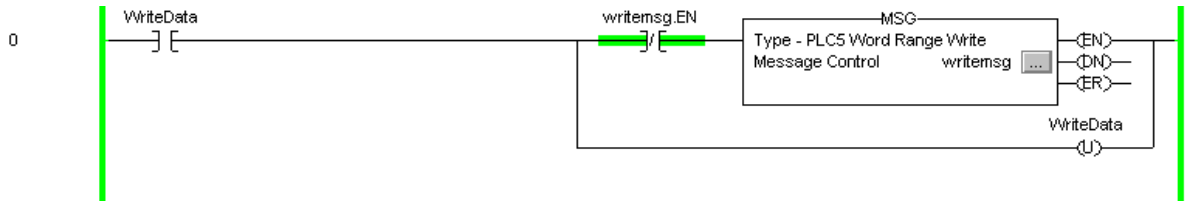
PLC5 and SLC5/05 processors containing an Ethernet interface use the encapsulated PCCC message method. The gateway simulates these devices and accepts both read and write commands.

5.4.3.2.1 Encapsulated PCCC Write Message

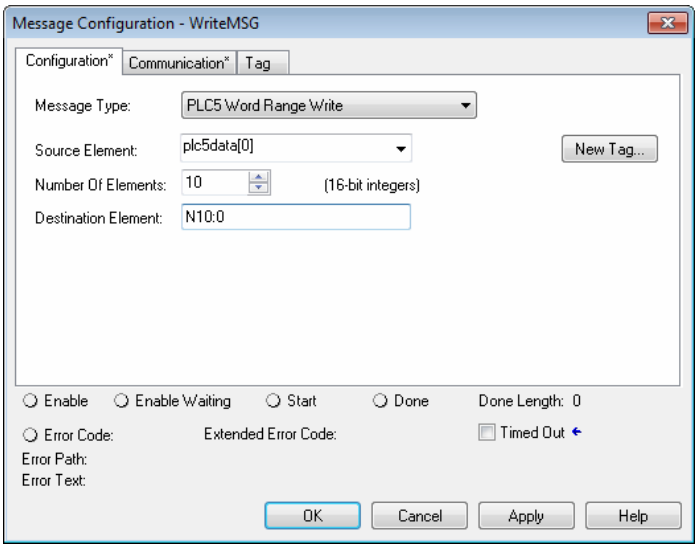
Write commands transfer data from the processor to the gateway. The gateway supports the following encapsulated PCCC commands:

- PLC2 Unprotected Write
- PLC5 Typed Write
- PLC5 Word Range Write
- PLC Typed Write

The following diagram shows an example rung that executes a write command.



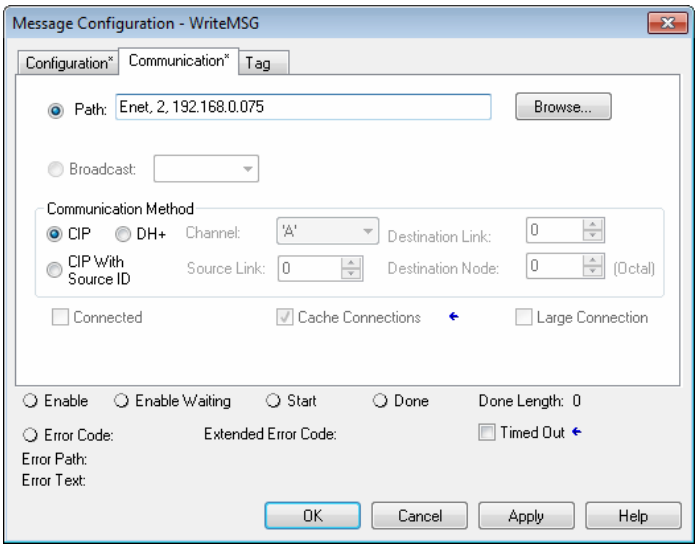
1 In the *Message Configuration* dialog, define the data set to be transferred from the processor to the gateway as shown in the following image.



2 Complete the dialog for the data area to be transferred.

- For PLC5 and SLC messages, set the **DESTINATION ELEMENT** to an element in a data file (such as, N10:0).
- For the PLC2 Unprotected Write message, set the **DESTINATION ELEMENT** to the address in the gateway's internal database. This cannot be set to a value less than ten. This is not a limitation of the gateway but of the RSLogix software.
- For a PLC2 Unprotected Write or Read function, enter the database address in octal format.

3 Click the **COMMUNICATION** tab and complete the communication information as shown in the following image.



4 Select **CIP** as the **COMMUNICATION METHOD**. The **PATH** specifies the message route from the processor to the EIP gateway. Path elements are separated by commas. In the example path shown:

- The first element is "**Enet**", which is the user-defined name given to the 1756-ENET gateway in the chassis (the slot number of the ENET gateway cannot be substitued for the name)
- The second element, "**2**", represents the Ethernet port on the 1756-ENET gateway.
- The last element of the path, "**192.168.0.75**" is the IP address of the gateway, which is the target for the message.

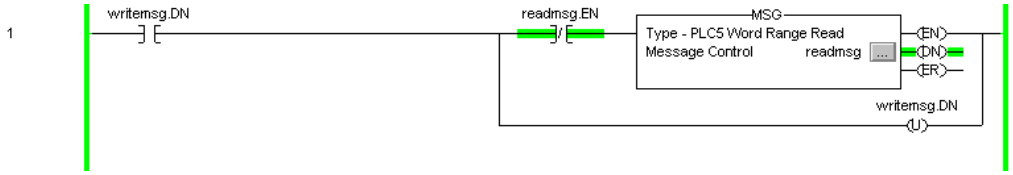
More complex paths are possible for routing to other networks using multiple 1756-ENET gateways and racks. Refer to the ProSoft Technology Technical Support Knowledgebase for more information on Ethernet routing and path definitions.

5.4.3.2.2 Encapsulated PCCC Read Message

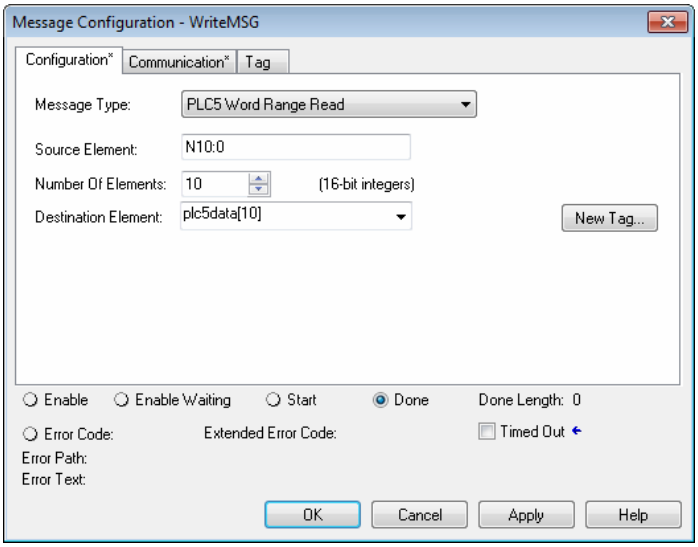
Read commands transfer data from the gateway to a processor. The gateway supports the encapsulated PCCC commands:

- PLC2 Unprotected Read
- PLC5 Typed Read
- PLC5 Word Range Read
- PLC Typed Read

The following diagram shows an example rung that executes a read command.



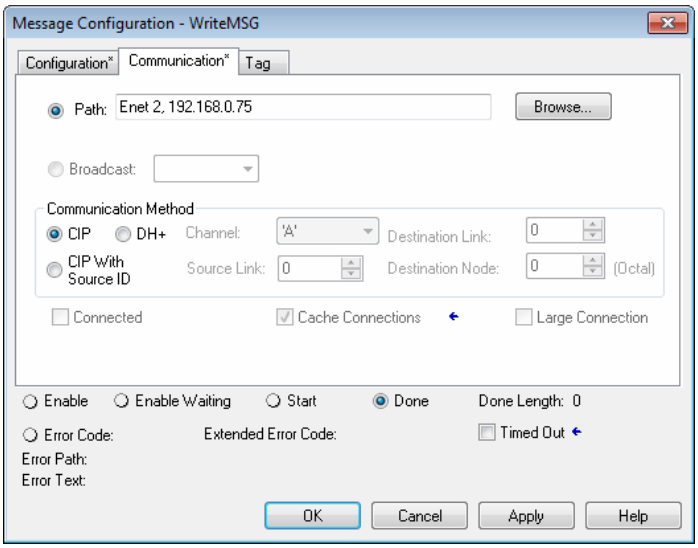
1 In the *Message Configuration* dialog, define the data set to be transferred from the processor to the gateway as shown in the following image.



2 Complete the dialog for the data area to be transferred.

- For PLC5 and SLC messages, set the **SOURCE ELEMENT** to an element in a data file (such as, N10:0).
- For the PLC2 Unprotected Read message, set the **SOURCE ELEMENT** to the address in the gateway's internal database. This cannot be set to a value less than ten. This is not a limitation of the gateway but of the RSLogix software.

3 Click the **COMMUNICATION** tab and complete the communication information as shown in the following image.



4 Select **CIP** as the **COMMUNICATION METHOD**. The **PATH** specifies the message route from the processor to the EIP gateway. Path elements are separated by commas. In the example path shown:

- The first element is "Enet", which is the user-defined name given to the 1756-ENET gateway in the chassis (substitute the slot number of the ENET gateway for the name)
- The second element, "2", represents the Ethernet port on the 1756-ENET gateway.
- The last element of the path, "192.168.0.75" is the IP address of the gateway, and the target for the message.

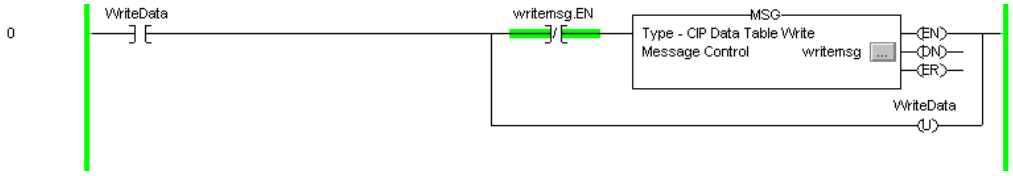
More complex paths are possible if routing to other networks using multiple 1756-ENET gateways and racks. Refer to the ProSoft Technology Technical Support Knowledgebase for more information on Ethernet routing and path definitions.

5.4.3.3 CIP Data Table Operations

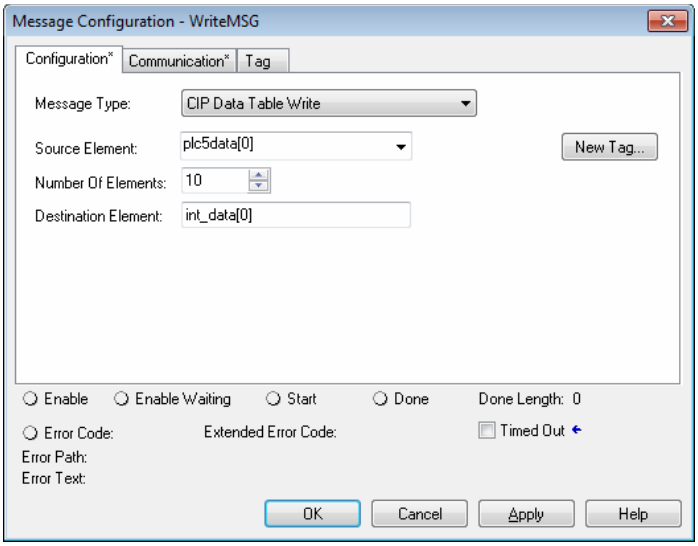
Use CIP messages to transfer data between the ControlLogix or CompactLogix processor and the gateway. Tag names define the elements to be transferred. The gateway supports both read and write operations.

5.4.3.3.1 CIP Data Table Write

CIP data table write messages transfer data from the processor to the gateway. The following diagram shows an example rung that executes a write command.

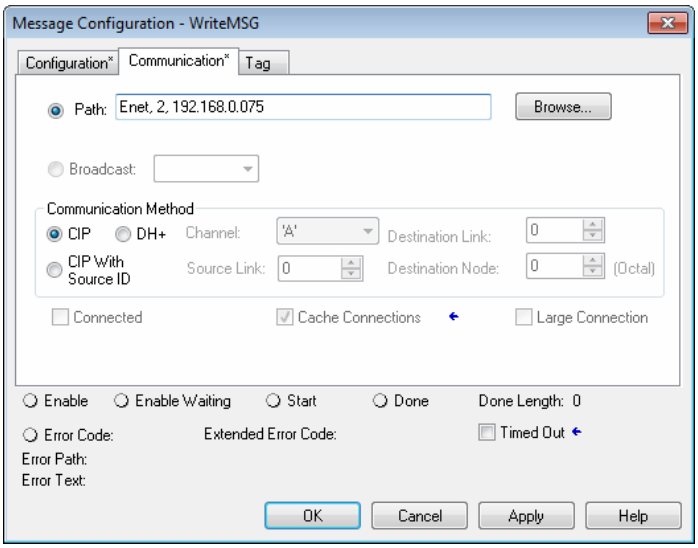


1. In the *Message Configuration* dialog, define the data set to be transferred from the processor to the gateway as shown in the following image.



2. Complete the dialog for the data area to be transferred. CIP Data Table messages require a tag database element for both the source and destination.
   - The **SOURCE TAG** is a tag defined in the Controller Tag database.
   - The **DESTINATION ELEMENT** is the tag element in the gateway.
   - The gateway simulates a tag database as an array of elements defined by the maximum register size for the gateway with the tag name **INT_DATA** (with the maximum value of int_data[3999]).

3 In the previous example, the first element in the database is the starting location for the write operation of ten elements. Click the **COMMUNICATION** tab and complete the communication information as shown in the following image.



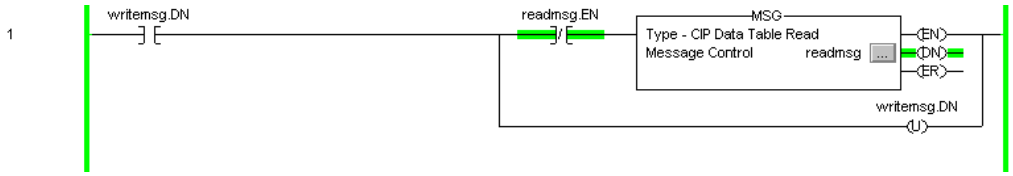
4 Select **CIP** as the **COMMUNICATION METHOD**. The **PATH** specifies the message route from the processor to the EIP gateway. Path elements are separated by commas. In the example path shown:

- The first element is "**Enet**", which is the user-defined name given to the 1756-ENET gateway in the chassis (substitute the slot number of the ENET gateway for the name)
- The second element, "**2**", represents the Ethernet port on the 1756-ENET gateway.
- The last element of the path, "**192.168.0.75**" is the IP address of the gateway, which is the target for the message.

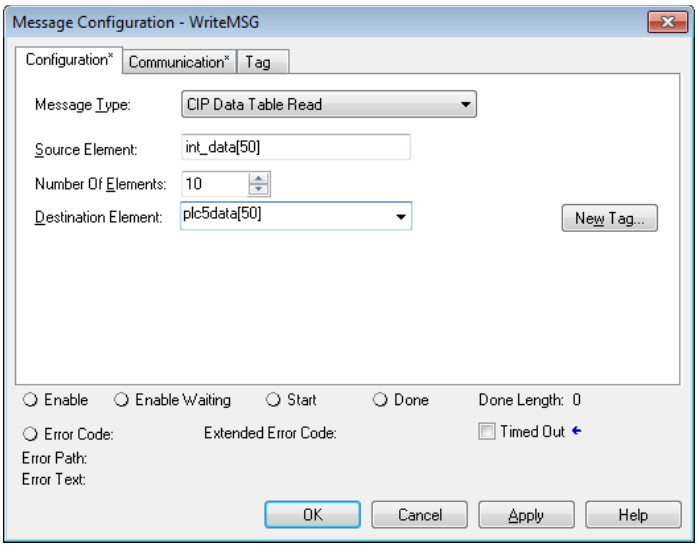
More complex paths are possible if routing to other networks using multiple 1756-ENET gateways and racks. Refer to the ProSoft Technology Technical Support Knowledgebase for more information on Ethernet routing and path definitions.

5.4.3.3.2 CIP Data Table Read

CIP data table read messages transfer data to the processor from the gateway. The following diagram shows an example rung that executes a read command.



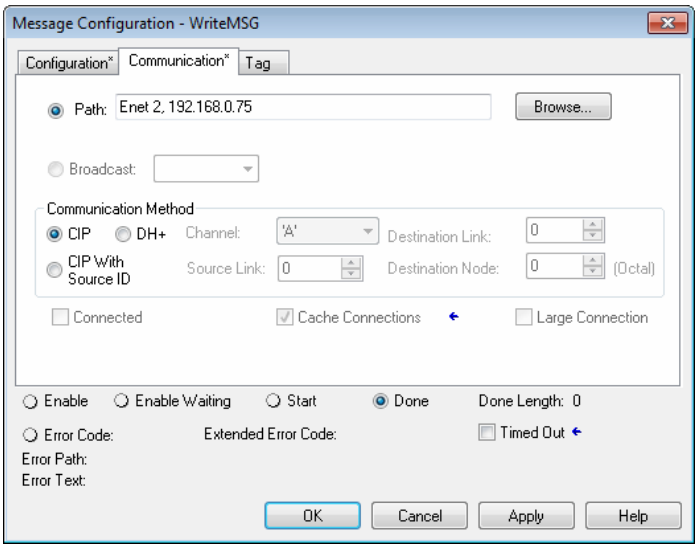
1 In the *Message Configuration* dialog, define the data set to be transferred from the processor to the gateway as shown in the following image.



2 Complete the dialog for the data area to be transferred. CIP Data Table messages require a tag database element for both the source and destination.

- The **DESTINATION TAG** is a tag defined in the Controller Tag database.
- The **SOURCE ELEMENT** is the tag element in the gateway.
- The gateway simulates a tag database as an array of elements defined by the maximum register size for the gateway (user configuration parameter "Maximum Register" in the [Gateway] section) with the tag name **INT_DATA.**

3 In the previous example, the first element in the database is the starting location for the read operation of ten elements. Click the **COMMUNICATION** tab and complete the communication information as shown in the following image.



4 Select **CIP** as the **COMMUNICATION METHOD**. The **PATH** specifies the message route from the processor to the EIP gateway. Path elements are separated by commas. In the example path shown:

- The first element is "Enet", which is the user-defined name given to the 1756-ENET gateway in the chassis (substitute the slot number of the ENET gateway for the name)
- The second element, "2", represents the Ethernet port on the 1756-ENET gateway.
- The last element of the path, "192.168.0.75" is the IP address of the gateway, which is the target for the message.

More complex paths are possible if routing to other networks using multiple 1756-ENET gateways and racks. Refer to the ProSoft Technology Technical Support Knowledgebase for more information on Ethernet routing and path definitions.

6 MBTCP Protocol

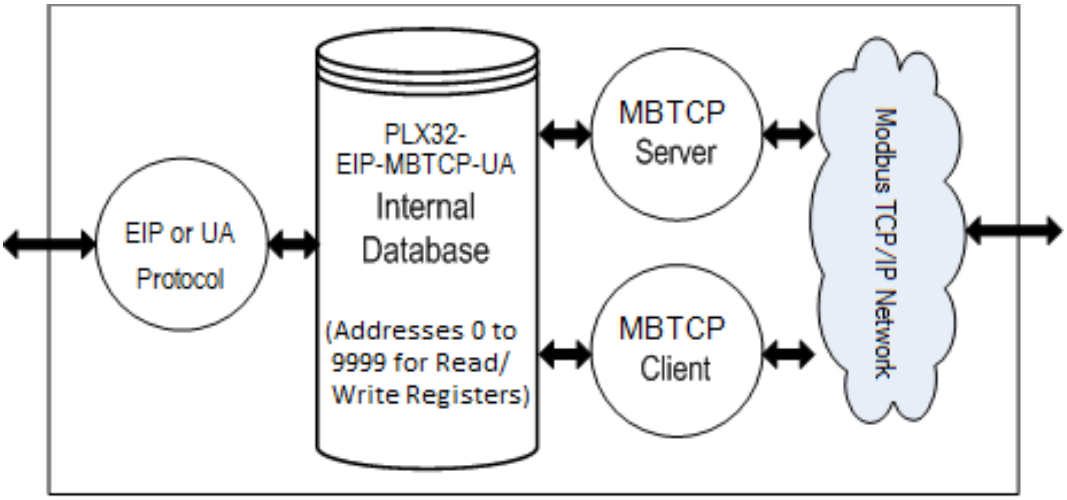
6.1 MBTCP Functional Overview

Use the PLX32-EIP-MBTCP-UA Modbus TCP/IP (MBTCP) protocol to interface many different protocols into the Schneider Electric Quantum family of processors as well other devices supporting the protocol. The MBTCP protocol supports both client and server connections.

The gateway supports a client connection on the TCP/IP network to interface with processors (and other server based devices) using a command list of up to 100 entries. The gateway stores the write commands for remote processors in the gateway's lower memory. This is also where the gateway stores data from read commands from other devices. For more information, see section *6.1.2 MBTCP Internal Database*.

Data in the lower memory of the gateway's internal database is accessible for read and write operations by any node on the network supporting the MBAP (Service Port 502) or MBTCP (Service Ports 2000/2001) TCP/IP protocols. The MBAP protocol (Port 502) is a standard implementation defined by Schneider Electric and used on their Quantum processor. This open protocol is a modified version of the Modbus serial protocol. The MBTCP protocol is an embedded Modbus protocol message in a TCP/IP packet. The gateway supports up to five active server connections on Service Ports 502, five additional active server connections on Service Port 2000, and one active client connection.

The following illustration shows the functionality of the Modbus TCP/IP protocol.



6.1.1 MBTCP General Specifications

The Modbus TCP/IP protocol allows multiple independent, concurrent Ethernet connections. The connections may be all clients, all servers, or a combination of both client and server connections.

- 10/100 MB Ethernet Communication port
- Supports Enron version of Modbus protocol for floating-point data transactions
- Configurable parameters for the client including a minimum response delay of 0 to 65535 ms and floating-point support
- Supports five independent server connections for Service Port 502
- Supports five independent server connections for Service Port 2000
- All data mapping begins at Modbus register 400001, protocol base 0.
- Error codes, error counters, and port status data available in user data memory

6.1.1.1 Modbus TCP/IP Client

- Actively reads data from and writes data to Modbus TCP/IP devices using MBAP
- Up to 10 client connections with multiple commands to talk to multiple servers

6.1.1.2 Modbus TCP/IP Server

- The server driver accepts incoming connections on Service Port 502 for clients using Modbus TCP/IP MBAP messages and connections on Service Port 2000 (or other Service Ports) for clients using Encapsulated Modbus messages.
- Supports multiple independent server connections for any combination of Service Port 502 (MBAP) and Service Port 2000 (Encapsulated)
- Up to 20 servers are supported

Parameter	Description	
Modbus Commands Supported (client and server)	1: Read Coil Status 2: Read Input Status 3: Read Holding Registers 4: Read Input Registers 5: Force (Write) Single Coil 6: Preset (Write) Single Holding Register	15: Force (Write) Multiple Coils 16: Preset (Write) Multiple Holding Registers 22: Mask Write Holding Register (Slave Only) 23: Read/Write Holding Registers (Slave Only)
Configurable Parameters: (client and server)	• Gateway IP Address • PLC Read Start Register (%MW) • PLC Write Start Register (%MW) • Number of MBAP and MBTCP servers • Gateway Modbus Read Start Address • Gateway Modbus Write Start Address	
Configurable Parameters: (client only)	• Minimum Command Delay • Response Timeout • Retry Count • Command Error Pointer	
Command List	Up to 160 Modbus commands (one tag per command)	

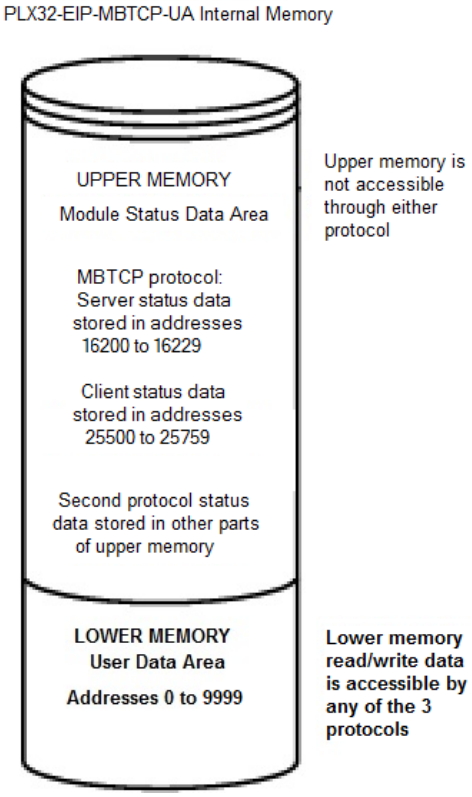
Status Data	• Error codes reported individually for each command. • High-level status data available from Modbus TCP/IP client (Ex: PLC)
Command List Polling	Each command can be individually enabled or disabled; write-only-on-data-change is available

6.1.2 MBTCP Internal Database

The internal database is central to the functionality of the PLX32-EIP-MBTCP-UA. The gateway shares this database between all the communications ports on the gateway and uses it as a conduit to pass information from one protocol to another device on one network to one or more devices on another network. This permits data from devices on one communication port to be accessed and controlled by devices on another communication port.

In addition to data from the client and server, status and error information generated by the gateway can be mapped into the user data area of the internal database. The internal database is divided into two areas:

- Upper memory for the gateway status data area. This is where the gateway writes internal status data for the protocols supported by the gateway.
- Lower memory for the user data area. This is where incoming data from external devices is stored and accessed.



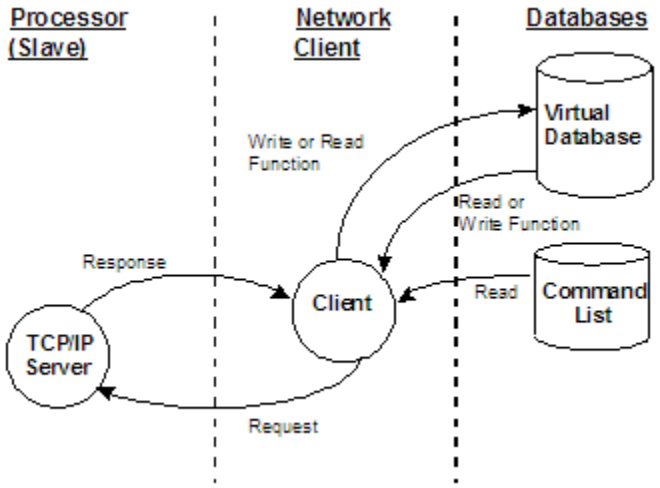
Each protocol in the PLX32-EIP-MBTCP-UA can write data to and read data from the user data area.

Note: To access gateway status data in the upper memory, use the data mapping feature in the gateway to copy data from the gateway status data area to the user data area. For more information, see section *2.7 Mapping Data in PLX32-EIP-MBTCP-UA Memory*. Otherwise, use the diagnostic functions in ProSoft Configuration Builder to view gateway status data. For more information on the gateway status data, see section *6.3 MBTCP Network Diagnostics*.

6.1.2.1 Modbus TCP/IP Client Access to Database

The client functionality exchanges data between the PLX32-EIP-MBTCP-UA's internal database and data tables established in one or more Quantum processors or other server based devices. The command list that defined in ProSoft Configuration Builder specifies what data is to be transferred between the gateway and each of the servers on the network. No ladder logic is required in the processor (server) for client functionality, except to ensure that sufficient data memory exists.

The following illustration describes the flow of data between the Ethernet clients and the internal database.



6.1.2.2 Multiple Server Access to Database

The MBTCP gateway provides server functionality using reserved Service Port 502 for Modbus TCP/IP MBAP messages, as well as Service Ports 2000 and 2001 to support the TCP/IP Encapsulated Modbus version of the protocol used by several HMI manufacturers. Server support in the gateway permits client applications (for example: HMI software, Quantum processors, etc) to read from and write to the gateway's database. This section discusses the requirements for attaching to the gateway using client applications.

The server driver supports multiple concurrent connections from several clients. Up to five clients can simultaneously connect on Service Port 502 and five more can simultaneously connect on Service Port 2000. The MBTCP protocol uses Service Port 2001 to pass Encapsulated Modbus commands through from the Ethernet port to the gateway's serial port.

When configured as a server, the gateway uses its internal database as the source for read requests and the destination for write requests from remote clients. Access to the database is controlled by the command type received in the incoming message from the client. The following table specifies the relationship of the gateway's internal database to the addresses required in the incoming Modbus TCP/IP requests.

Database Address	Modbus Address
0	40001
1000	41001
2000	42001
3000	43001
3999	44000

The following virtual addresses are not part of the normal gateway user database and are not valid addresses for standard data. However, these addresses may be used for incoming commands that are requesting floating-point data.

To use addresses in this upper range requires the configuration of the following parameters in *Prosoft Configuration Builder* (PCB):

- Set the *Float Flag* in the MBTCP server configuration to **YES**
- Set the *Float Start* to a database address in the range below
- Set the *Float Offset* to a database address in the gateway user memory area shown above.

All data above the *Float Start* address must be floating-point data. For more information, see *6.2.1 Configuring MBTCP Servers*.

Database Address	Modbus Address
4000	44001
5000	45001
6000	46001
7000	47001
8000	48001
9000	49001
9999	50000

The gateway must be correctly configured and connected to the network before any attempt is made to use it. Use a network verification program, such as ProSoft Discovery Service or the command prompt PING instruction, to verify that other devices can find the gateway on the network. Use ProSoft Configuration Builder to confirm proper configuration of the gateway and to transfer the configuration files to and from the gateway.

6.1.2.3 Modbus Message Routing: Port 2001

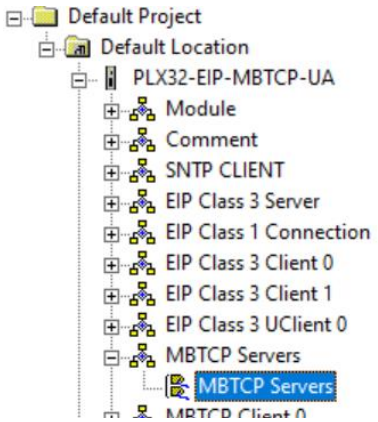
When Modbus messages are sent to the PLX32-EIP-MBTCP-UA over the TCP/IP connection to port 2001, the messages are routed by the gateway directly out the serial communication port (Port 0 if it is configured as a Modbus master). The commands (whether a read or a write command) are immediately routed to the slave devices on the serial port. Response messages from the slave devices are routed by the gateway to the TCP/IP network to be received by the originating host.

6.2 MBTCP Configuration

6.2.1 Configuring MBTCP Servers

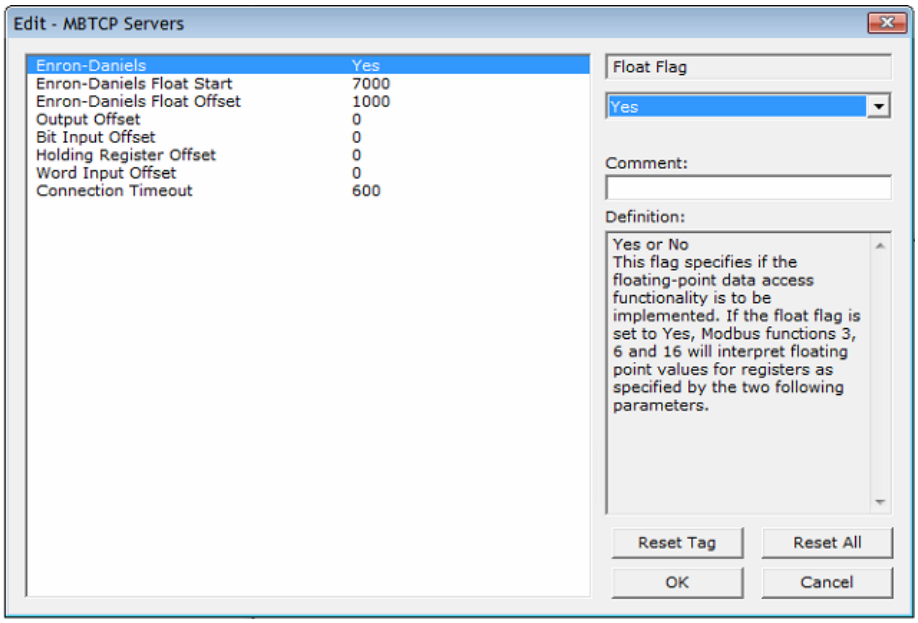
This section contains database offset information used by the PLX32-EIP-MBTCP-UA MBTCP server when accessed by external clients. Use these offsets to segment the database by data type.

1 In *ProSoft Configuration Builder*, click the **[+]** next to the gateway, then click the **[+]** next to *MBTCP Servers*.



2 Double-click the second *MBTCP Servers* to display the *Edit - MBTCP Servers* dialog.

3 In the dialog, click a parameter and enter a value for the parameter. Note that the *Float Start* and *Float Offset* parameters only appear if the *Float Flag* parameter is set to *Yes*.

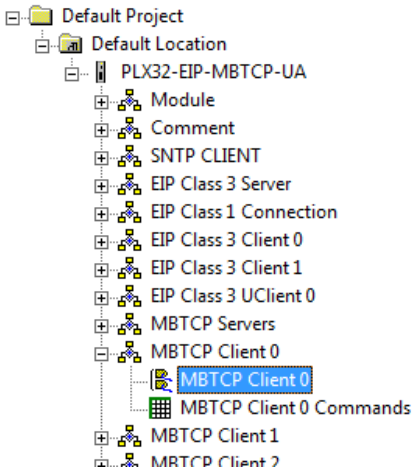


Parameter	Value	Description
Float Flag (Enron-Daniels)	Yes or No	Specifies if the floating-point data access functionality is active. **YES** - Modbus functions 3, 6, and 16 interpret floating-point values for registers as specified by *Float Start* and *Float Offset*. **NO** - The gateway does not use floating point functionality.
Float Start	0 to 65535	This parameter only appears if *Float Flag* is Yes. Specifies the first register of floating-point data if *Float Flag* is Yes. All requests with register values greater than or equal to this value are considered floating-point data requests. For example, a value of 7000 considers all requests for registers 7000 and above as floating-point data.
Float Offset	0 to 9999	This parameter only appears if *Float Flag* is Yes. Specifies the start register for floating-point data in the internal database. For example, a value of 3000 and the *Float Start* is set to 7000, data requests for register 7000 will use the internal Modbus register 3000.
Output Offset	0 to 9999	This parameter applies if the port is configured as a slave. Specifies the internal database address to use as the zero address or starting point for binary output Coil data. Coil data is read by Modbus Function Code 1 commands (Read Coils) and written by Function Codes 5 (Force Single Coil) or Function Code 15 (Force Multiple Coils). For example, if the value is set to 50 and the gateway receives a Function Code 1 command requesting Coil address 0 (virtual Modbus Coil address 00001 or 000001), the gateway returns the value at register 50, bit 0 in the gateway's database.
Bit Input Offset	0 to 9999	Specifies the offset address in the internal Modbus database for network requests for Modbus function 2 commands. For example, if the value is set to 150, an address request of 0 returns the value at register 150 in the database.
Holding Register Offset	0 to 9999	Specifies the offset address in the internal Modbus database for network requests for Modbus functions 3, 6, or 16 commands. For example, if the value is set to 50, an address request of 0 returns the value at register 50 in the database.
Word Input Offset	0 to 9999	Specifies the offset address in the internal Modbus database for network requests for Modbus function 4 commands. For example, if the value is set to 150, an address request of 0 returns the value at register 150 in the database.
Connection Timeout	0 to 1200	Specifies the number of seconds the server waits to receive new data. If the server does not receive any new data during this time, it closes the connection.

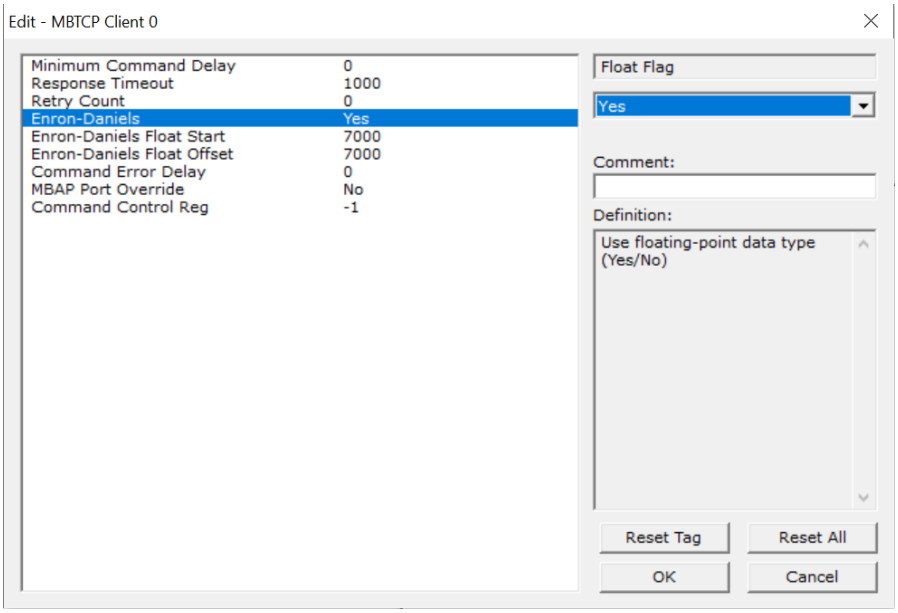
6.2.2 Configuring MBTCP Client [x]

The *MBTCPClient [x]* section of the configuration specifies the parameters for the client to be emulated on the gateway. The command list for the client is entered in a separate section.

1. In *ProSoft Configuration Builder*, click the **[+]** next to the gateway, then click the **[+]** next to *MBTCP Client [x]*.



2. Double-click the second *MBTCP Client [x]* to display the *Edit - MBTCP Client [x]* dialog.
3. In the dialog, click a parameter and then enter a value for the parameter. Note that the *Float Start* and *Float Offset* parameters only appear if *Float Flag* is set to **YES**.



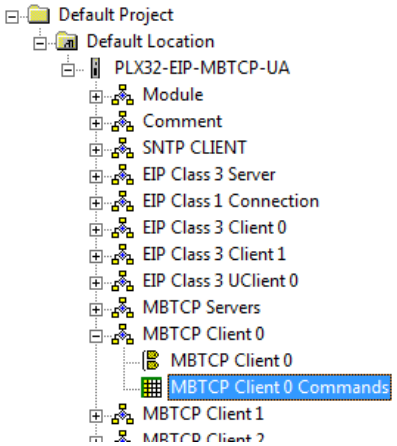
Parameter	Value	Description
Minimum Command Delay	0 to 32767	Specifies the number of milliseconds to wait between the initial issuance of a commands. Use this to delay all commands sent to slaves to avoid "flooding" commands on the network. This parameter does not affect retries of a command as they will be issued when failure is recognized.
Response Timeout	0 to 65535	Specifies the time in milliseconds that a client waits before re-transmitting a command if no response is received from the addressed server. The value depends on the type of communication network, and the expected response time of the slowest device on the network.
Retry Count	0 to 10	Specifies the number of times the gateway retries a command if it fails.
Float Flag (Enron-Daniels)	Yes or No	Specifies if the floating-point data access functionality is active. **YES** - Modbus functions 3, 6, and 16 interpret floating-point values for registers as specified by *Float Start* and *Float Offset*. **NO** - The gateway does not use floating point functionality.
Float Start	0 to 32767	This parameter only appears if *Float Flag* is Yes. Specifies the first register of floating-point data. The gateway considers all requests with register values greater-than or equal to this value as floating-point data requests. For example, if a value of 7000 is entered, the gateway considers all requests for registers 7000 and above as floating-point data.
Float Offset	0 to 9998	This parameter only appears if *Float Flag* is Yes. Specifies the starting register for floating-point data in the gateway internal database. For example: • If Float Offset is set to 3000 and Float Start is set to 7000, the gateway returns data as floating-point data for register 47001 (or 407001) comes from internal gateway registers 3000 and 3001. • If the requested address is 47002 (407002), the gateway returns data from internal registers 3002 and 3003. • If the requested address is 47101 (407101), the gateway returns data from internal registers 3200 and 3201; and so on.
Command Error Delay	0 to 300	Specifies the number of 100 millisecond intervals to turn off a command in the error list after an error is recognized for the command. If it is set to 0, there is no delay.
MBAP Port Override	Yes or No	Specifies whether to override the default port settings. **YES** - The gateway uses MBAP format messages for all Server Port values. The gateway does not use RTU through TCP. **NO** - The gateway uses standard Server Port 502 with MBAP format messages. All other Server Port values use encapsulated Modbus message format (RTU through TCP).
Command Control Reg	0 to 9840, -1 = Disable	This parameter allows the control of command execution in the MBTCP Client Command List. This parameter reserves 100 registers, starting at the value entered. **Note:** This feature allows a command to be enabled, disabled, etc. regardless of how it is configured in the client command list. A value of **0**, **1**, or **2** can be entered into each command control register: **0** = The command will be disabled. **1** = The command will continuously execute.

Parameter	Value	Description
		2 = The command will be enabled for conditional writing, which will cause the command to execute only when the value to be written has changed.

6.2.3 Configuring MBTCP Client [x] Commands

To interface the PLX32-EIP-MBTCP-UA with Modbus TCP/IP server devices, a command list must be created. The *MBTCP Client [x] Commands* section defines the Modbus TCP/IP commands to be issued from the gateway to server devices on the network. Use these commands for data collection and/or control of devices on the TCP/IP network.

1. In *ProSoft Configuration Builder*, click the **[+]** next to the gateway, then click the **[+]** next to *MBTCP Client [x]*.



2. Double-click *MBTCP Client [x] Commands* to display the *Edit - MBTCP Client [x] Commands* dialog.
3. In the dialog, click **ADD ROW** to add a command, then click **EDIT ROW** to enter values for the command.

The commands in the list specify the server device to be addressed, the function to be performed (read or write), the data area in the device to interface with and the registers in the internal database to be associated with the device data. The client command list supports up to 16 commands per client. The gateway processes the command list from top (command #0) to bottom.

The following table describes the command list configuration parameters:

Parameter	Value	Description
Enable	YES NO CONDITONAL	Specifies if the command is to be executed and under what conditions. **NO (0)** - the command is disabled and is not executed in the normal polling sequence. **YES (1)** - the command is executed upon each scan of the Command List if the *Poll Interval* is set to zero (0). If the *Poll Interval* is set to a non-zero value, the command is executed when the interval timer for that command expires. **CONDITIONAL (2)** - the command is executed only if the internal bit data associated with the command changes. This parameter is valid for write commands (FC 5, 6, 15 and 16).
Internal Address	0 to 9999 (for register-level addressing) or 0 to 159999 (for bit-level addressing)	Specifies the database address in the gateway's internal database to use as the destination for data from a read command, or as the source for data sent by a write command. The database address is interpreted as a bit address or a 16-bit register (word) address, depending on the Modbus Function Code used in the command. • For Modbus functions 1, 2, 5, and 15, this parameter is interpreted as a bit-level address. • For Modbus functions 3, 4, 6, and 16, this parameter is interpreted as a register-level address.
Poll Interval	0 to 65535	Specifies the minimum interval between executions of continuous commands. The value is in 1/10 second. If a value of 100 is entered, the command executes no more frequently than once every 10 seconds.
Reg Count	1 to 125 (for registers) or 1 to 2000 (for coils)	Specifies the number of 16-bit registers or binary bits to be transferred by the command. • Modbus functions 5 and 6 ignore this field as they apply only to a single data point. • For Modbus functions 1, 2, and 15, this parameter sets the number of bits (inputs or coils) transferred by the command. **Note:** For Modbus functions 1 and 2; 2000 coils are supported. For Modbus function 15; 1968 coils are supported. • For Modbus functions 3, 4, and 16, this parameter sets the number of registers transferred by the command.
Swap Code	No Change Word Swap Word and Byte Swap Byte Swap	Specifies if and how the order of bytes in data received or sent is to be rearranged. Different manufacturers store and transmit multi-byte data in different combinations. Use this parameter when dealing with floating-point or other multi-byte values, as there is no standard method of storing these data types. Set this parameter to rearrange the byte order of data received or sent into an order more useful or convenient for other applications. **NO CHANGE (0)** - No change is made in the byte ordering (1234 = 1234). **WORD SWAP (1)** -The words are swapped (1234=3412). **WORD AND BYTE SWAP (2)** - The words are swapped, then the bytes in each word are swapped (1234=4321). **BYTE SWAP (3)** - The bytes in each word are swapped (1234=2143). These swap operations affect 4-byte (2-word) groups of data. Therefore, data swapping using *Swap Codes* should be done only when using an even number of words, such as 32-bit integer or floating-point data.
Node IP Address	xxx.xxx.xxx.xxx	IP address of the device being addressed by the command.

Parameter	Value	Description
Serv Port	502 or other supported port on server	Service Port on which communication will occur. Use a value of 502 when addressing Modbus TCP/IP servers that are compatible with the Schneider Electric MBAP specifications (this will be most devices). If the server device supports another Service Port, enter the Service Port value for this parameter.
Slave Address	1 to 255 (0 is a broadcast)	Specifies the node address of a remote Modbus Serial device through a Modbus Ethernet to Serial converter. **Note:** Most Modbus devices only accept addresses in the range of 1 to 247, so check with the slave device manufacturer to see if the slave device can use addresses 248 to 255. If the value is set to zero, the command will be a broadcast message on the network. The Modbus protocol permits broadcast commands for *write* operations. **Do not** use node address 0 for *read* operations.
Modbus Function	1, 2, 3, 4, 5, 6, 15, or 16	Specifies the Modbus Function Code to be executed by the command. These function codes are defined in the Modbus protocol. More information on the protocol is available from www.modbus.org or see section *6.4.1 About the Modbus Protocol*. The following function codes are supported by the gateway. **1** - Read Coil Status **2** - Read Input Status **3** - Read Holding Registers **4** - Read Input Registers **5** - Force (Write) Single Coil **6** - Preset (Write) Single Register **15** - Force Multiple Coils **16** - Preset Multiple Registers
MB Address in Device	Varies	Specifies the starting Modbus register or bit address in the server to be used by the command. Refer to the documentation of each Modbus server device for the register and bit address assignments valid for that device. The Modbus Function Code determines whether the address is a register-level or bit-level OFFSET address into a given data type range. The offset is the target data address in the server minus the base address for that data type. Base addresses for the different data types are: • 00001 or 000001 (0x0001) for bit-level Coil data (Function Codes 1, 5, and 15). • 10001 or 100001 (1x0001) for bit-level Input Status data (Function Code 2) • 30001 or 300001 (3x0001) for Input Register data (Function Code 4) • 40001 or 400001 (4x0001) for Holding Register data (Function Codes 3, 6, and 16). • Address calculation examples: • For bit-level Coil commands (FC 1, 5, or 15) to read or write a Coil 0X address 00001, specify a value of 0 (00001 - 00001 = 0). • For Coil address 00115, specify 114 (00115 - 00001 = 114) • For register read or write commands (FC 3, 6, or 16) 4X range, for 40001, specify a value of 0 (40001 - 40001 = 0).

Parameter	Value	Description
		• For 01101, 11101, 31101 or 41101, specify a value of 1100. (01101 - 00001 = 1100) (11101 - 10001 = 1100) (31101 - 30001 = 1100) (41101 - 40001 = 1100) **Note:** If the documentation for a particular Modbus server device lists data addresses in hexadecimal (base16) notation, convert the hexadecimal value to a decimal value for this parameter. In such cases, it is not usually necessary to subtract 1 from the converted decimal number, as this addressing scheme typically uses the exact offset address expressed as a hexadecimal number.
Comment		Optional 32-character comment for the command.

6.3 MBTCP Network Diagnostics

6.3.1 MBTCP PCB Diagnostics

Troubleshoot the MBTCP driver by using ProSoft Configuration Builder to access the diagnostic capabilities through the Ethernet debug port. For instructions on how to access the diagnostics, see chapter *3 Diagnostics and Troubleshooting*.

The following table summarizes the status information available in PCB for the MBTCP driver:

Connection Type	Submenu Item	Description
MBTCP Server	Config	Configuration settings for Server Connections.
	Comm Status	Status of the Server Connections. Displays a summary of the requests, responses, and errors.
MBTCP Client [x]	Config	Configuration settings for Client [x] Connections.
	Comm Status	Status information for Client [x] commands. Displays a summary of all the errors resulting from Client [x] commands.
	Modbus Commands	Configuration for the Client [x] Modbus command list.
	Modbus Cmd Errors (Decimal)	Current error codes for each command on the Client [x] command list in decimal number format. A zero means there is currently no error for the command.
	Modbus Cmd Errors (Hex)	Current error codes for each command on the Client [x] command list in hexadecimal number format. A zero means there is currently no error for the command.

6.3.2 MBTCP Status Data in Upper Memory

The MBTCP driver has an associated status data area located in the PLX32-EIP-MBTCP-UA's upper memory. The Data Map functionality of the PLX32-EIP-MBTCP-UA can be used to map this data into the normal user data range of the PLX32-EIP-MBTCP-UA's database.

Note that all the status values are initialized to zero (0) at power-up, cold boot and during warm boot.

6.3.2.1 MBTCP Server Status Data

The following table lists the addresses in upper memory where the PLX32-EIP-MBTCP-UA stores status data for MBTCP servers:

Server Port	Address Range
2000	16200 through 16209
502	16210 through 16219
2001	16220 through 16229

The content of each server's status data area is structured the same. The following table describes the content of each register in the status data area:

Offset	Description
0	Number of Command Requests
1	Number of Command Responses
2	Number of Command Errors
3	Number of Requests
4	Number of Responses
5	Number of Errors Sent
6	Number of Errors Received
7	Configuration Error Word
8	Current Error Code
9	Last Error Code

6.3.2.2 MBTCP Client Status Data

The following table lists the addresses in upper memory where the PLX32-EIP-MBTCP-UA stores status data for each MBTCP Client:

Client	Address Range
0	25500 through 25509
1	25526 through 25535
2	25552 through 25561
.	.
.	.
8	25708 through 25717
9	25734 through 25743

The content of each Client's status data area is structured the same. The following table describes the content of each register in the status data area:

Offset	Description
0	Command Request Count (total Client commands sent)
1	Command Response Count (total command responses received)
2	Command Error Count
3	Number of Request Packets
4	Number of Response Packets
5	Errors Sent
6	Errors Received
7	Reserved
8	Current Error
9	Last Error

- Offsets 8 and 9 contain information about the most recent communication errors.
- The Current Error (offset 8) has a non-zero value if the currently executing client command experiences an error.

- The Last Error (offset 9) stores the most recent non-zero value error code that was reported by the client the last time it experienced an error. Note that this value is protected. This register holds the last error value until the memory is cleared by a restart, reset, cold-boot, or warm-boot operation. Therefore, any value here may be from an error that occurred at any time since the PLX32-EIP-MBTCP-UA was last restarted.

6.3.2.3 MBTCP Client Command List Error Data

The PLX32-EIP-MBTCP-UA stores a status/error code in upper memory for each command in each MBTCP client's command list. The following table lists the addresses in upper memory where the PLX32-EIP-MBTCP-UA stores the command list error data for each MBTCP Client:

Client	Address Range
0	25510 to 25525
1	25536 to 25551
2	25562 to 25577
.	.
.	.
8	25718 to 25733
9	25744 to 25759

The first word in each client's command list error data area contains the status/error code for the first command in the client's Command List. Each successive word in the Command Error List is associated with the next command in the client Command List. Therefore, the number of valid error values depends on the number of commands defined.

The structure of the command list error data area (which is the same for all Clients) is displayed in the following table:

Offset	Description
0	Command #1 Error Code
1	Command #2 Error Code
2	Command #3 Error Code
.	.
.	.
14	Command #15 Error Code
15	Command #16 Error Code

A non-zero error code for a command indicates an error.

6.3.3 MBTCP Error Codes

6.3.3.1 Standard Modbus Exception Code Errors

These error codes are generated or returned on both the Controller and slave ports. These codes are the standard Modbus errors.

Code	Description
1	Illegal Function
2	Illegal Data Address
3	Illegal Data Value
4	Failure in Associated Device
5	Acknowledge
6	Busy, Rejected Message

6.3.3.2 MBTCP Client Specific Errors

These error codes are specific to the MBTCP client.

Code	Description
-33	Failed to connect to server specified in command
-35	Wrong message length in the response
-36	MBTCP command response timeout (same as -11)
-37	TCP/IP connection ended before session finished

6.3.3.3 MBTCP Communication Error Codes

The gateway detects these command-specific error codes during initial command list loading at gateway power-up or reset and are stored in the *Command Error List* memory region.

Code	Description
-2	Timeout while transmitting message
-11	Timeout waiting for response after request (same as -36)
253	Incorrect slave/server address in response
254	Incorrect function code in response
255	Invalid CRC/LRC value in response

6.3.3.4 MBTCP Command List Error Codes

The PLX32-EIP-MBTCP-UA detects these command-specific error codes during initial command list loading at PLX32-EIP-MBTCP-UA power-up or reset and are stored in the *Command Error List* memory region.

Code	Description
-40	Too few parameters
-41	Invalid enable code
-42	Internal address > maximum address
-43	Invalid node address (<0 or >255)
-44	Count parameter set to 0
-45	Invalid function code
-46	Invalid swap code

6.4 MBTCP Reference

6.4.1 About the Modbus Protocol

Modbus is a widely-used protocol originally developed by Modicon in 1978. Since that time, the protocol has been adopted as a standard throughout the automation industry.

The original Modbus specification uses a serial connection to communicate commands and data between master and server devices on a network. Later enhancements to the protocol allow communication over Ethernet networks using TCP/IP as a "wrapper" for the Modbus protocol. This protocol is known as Modbus TCP/IP.

Modbus TCP/IP is a client/server protocol. The master establishes a connection to the remote server. When the connection is established, the master sends the Modbus TCP/IP commands to the server. The PLX32-EIP-MBTCP-UA simulates up to 30 masters, and works both as a master and a server.

Aside from the benefits of Ethernet versus serial communications (including performance, distance, and flexibility) for industrial networks, the Modbus TCP/IP protocol allows for remote administration and control of devices over an Internet connection. It is important to note that not all Internet protocols are implemented in the gateway; for example, HTTP and SMTP protocols are not available. Nevertheless, the efficiency, scalability, and low cost of a Modbus TCP/IP network make this an ideal solution for industrial applications.

The PLX32-EIP-MBTCP-UA acts as an input/output gateway between devices on a Modbus TCP/IP network and the Rockwell Automation backplane and processor. The gateway uses an internal database to pass data and commands between the processor and the master and server devices on the Modbus TCP/IP network.

6.4.1.1 Supported Function Codes

The format of each command in the list depends on the Modbus Function Code being executed. The following table lists the Function Codes supported by the PLX32-EIP-MBTCP-UA.

Function Code	Definition	Supported as master	Supported as server
1	Read Coil Status 0x	X	X
2	Read Input Status 1x	X	X
3	Read Holding Registers 4x	X	X
4	Read Input Registers 3x	X	X
5	Set Single Coil 0x	X	X
6	Single Register Write 4x	X	X
8	Diagnostics		X
15	Multiple Coil Write 0x	X	X
16	Multiple Register Write 4x	X	X
17	Report Slave ID		X
22	Mask Write 4X		X
23	Read/Write		X

Each command list record has the same general format. The first part of the record contains the information relating to the communication gateway and the second part contains information required to interface to the Modbus server device.

7 OPC UA Server

This chapter covers the configuration of the PLX32-EIP-MBTCP-UA's OPC UA Server and OPC UA Client connectivity. The highly reliable and secure OPC UA Server supports 10 simultaneous OPC UA sessions that provide access to EtherNet/IP and Modbus TCP/IP data from the respective networks.

7.1 UA Server Configuration Manager Software

The UA Server is configured using the UA Server Configuration Manager (PSW-UACM) software. PSW-UACM is launched within the ProSoft Configuration Builder (PCB) software.

Note: To ensure a successful installation of ProSoft OPC UA Configuration Manager, a reboot may be required prior to starting the installation. In several test systems, Windows Update Service had to be stopped prior to installation. Once the installation completes, restart the Windows Update service.

Stop Windows Update service

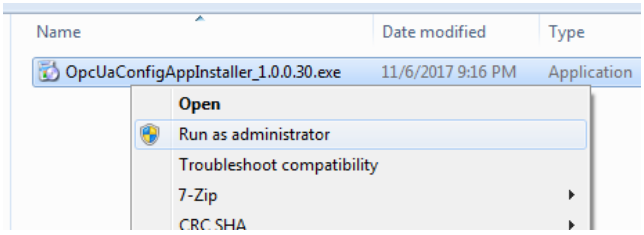
1. Click the Windows **START** button and enter the following: **services.msc**

2. Scroll down and right-click on Windows Update and choose **STOP**.

Perform the ProSoft OPC UA Configuration Manager setup procedures. Once the setup completes, perform the steps above and choose *Start* for the last step.

7.1.1 Installation

1. Download the latest version of PSW-UACM from www.prosoft-technology.com and save it to the local hard drive.
2. Right-click on the file and select **RUN AS ADMINISTRATOR**.



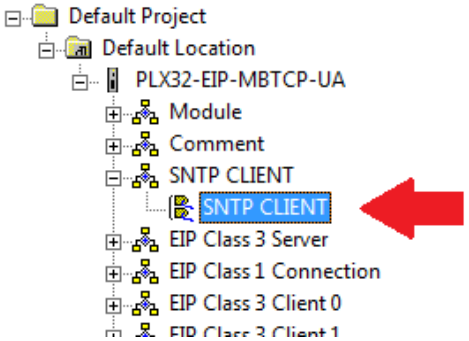
3. Follow the prompts in the Setup Wizard to complete the installation.
4. When installation is complete, click **FINISH**.

7.1.2 NTP Server Time Synchronization

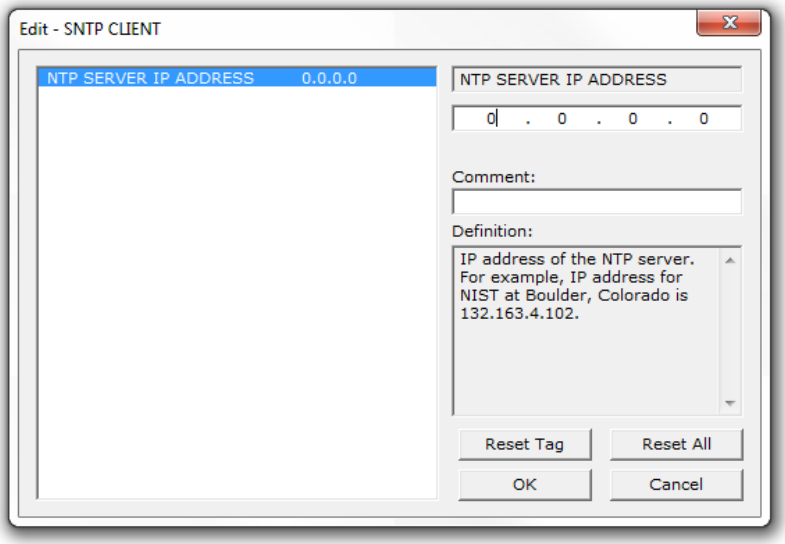
It is important to first set the system time of the PLX32-EIP-MBTCP-UA. It is best to maintain the PLX32-EIP-MBTCP-UA system time by synchronizing with an NTP (Network Time Protocol) server.

Important: If the NTP Server IP address is a public IP address, the NTP packet will ONLY be routed through OPC UA Ethernet port 2. Therefore, Ethernet Port 2 will need access to the public IP addresses.

1 In PCB, double-click on the **SNTP CLIENT** icon within the PLX32-EIP-MBTCP-UA tree.



2 In the *Edit – SNTP CLIENT* dialog, enter the IP address of an NTP server. Please consult the IT administrator for a suitable NTP server.



3 Click **OK**.

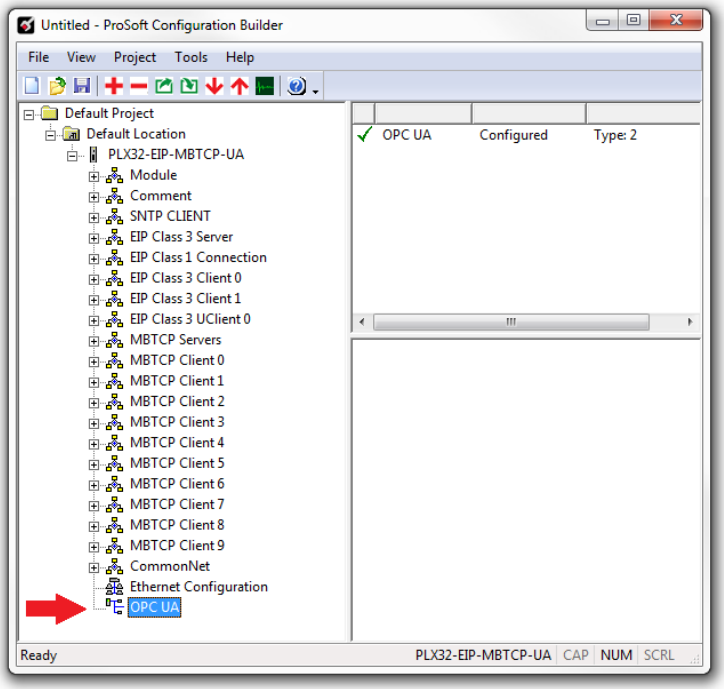
4 Save the PCB project, then download it to the PLX32-EIP-MBTCP-UA, as shown in section *2.8 Downloading the Project to the PLX32-EIP-MBTCP-UA*.

If there is no NTP server available, then the *NTP Server IP Address* field must be set to **0.0.0.0**. Setting it to an arbitrary IP address where there is no NTP server listening on port 123 is not recommended.

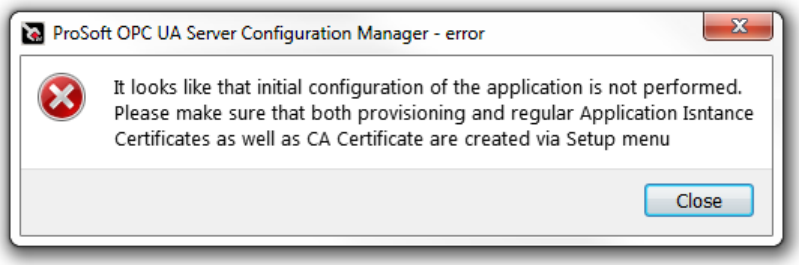
The system time can also be set manually by using the *Set Date and Time* function on the PLX32-EIP-MBTCP-UA web page. When the page is loaded, the default values for date and time are set to 3 seconds ahead. When the **UPDATE DATE AND TIME** button is clicked immediately after the web page loads, the system time sets to the PC's time within a few seconds of accuracy.

7.1.3 Launching PSW-UACM

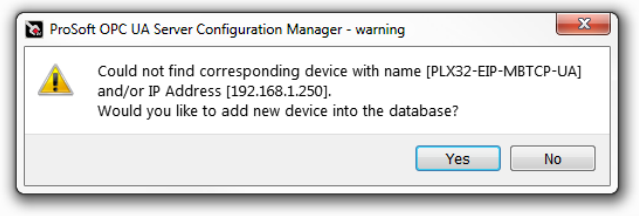
1 In PCB, double-click on the **OPC UA** icon within the PLX32-EIP-MBTCP-UA tree.



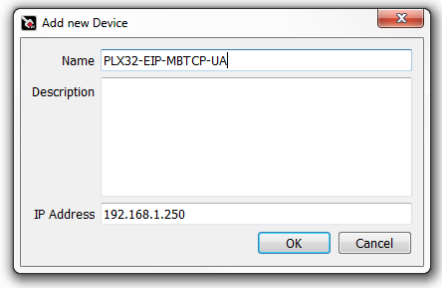
2 An error message may appear. This error is addressed later in this chapter. Click **CLOSE** to continue.



3 A warning message may appear. It indicates that the PSW-UACM has not found the PLX32-EIP-MBTCP-UA on the local network. Click **YES** to add a new device into the UA Server database.

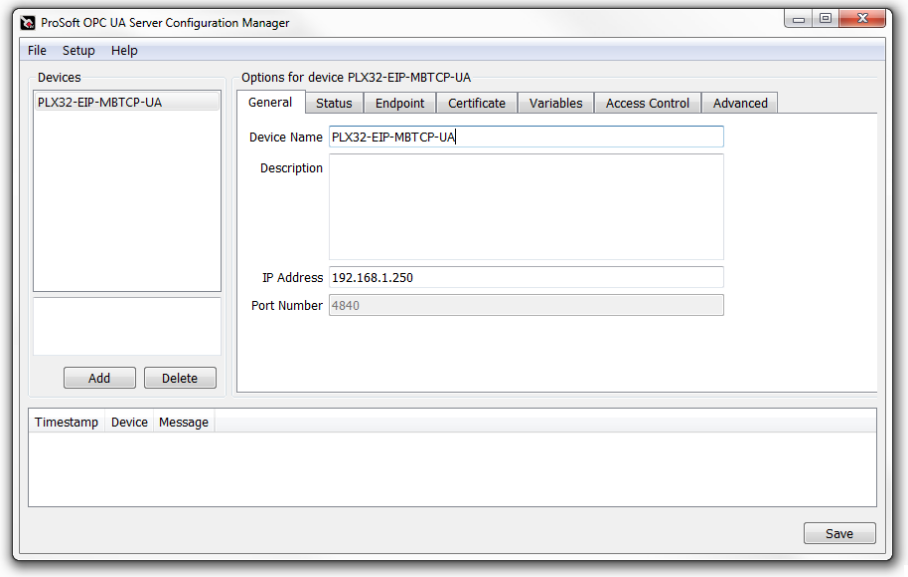


4 Enter the *Name*, *Description* (optional), and unique *IP Address* of the new device. Click **OK**.



Note: The IP Address can also be edited in the *General* tab of PSW-UACM.

5 Once complete, the main window of PSW-UACM displays.



Note that the device in the PSW-UACM is associated with the PLX32-EIP-MBTCP-UA name in PCB. If PSW-UACM cannot find the PLX32-EIP-MBTCP-UA with the same name as the PCB module name, it looks for a device with the same IP address. If the corresponding device cannot be found, then PSW-UACM prompts to create a new device.

If a PLX32-EIP-MBTCP-UA in PCB is renamed, the corresponding device's name in PSW-UACM should be modified to the same name to maintain the correct association.

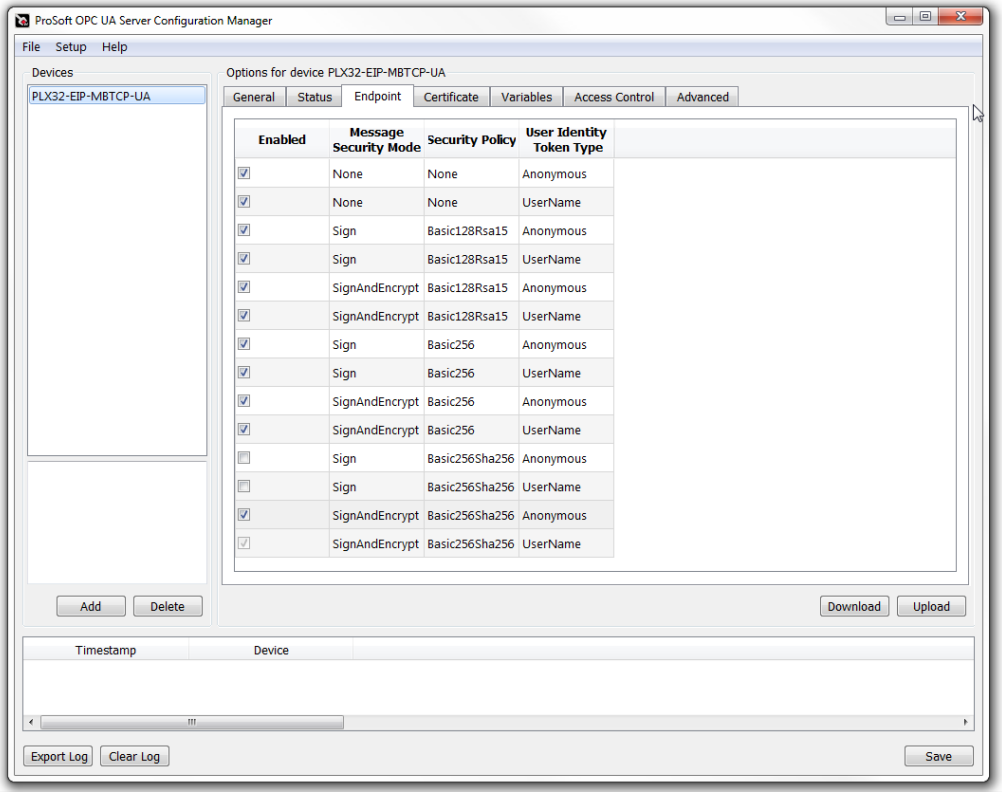
7.2 Certificates

There are three certificates that need to be generated by PSW-UACM. This section covers these steps.

7.2.1 Security Policy

Prior to provisioning the OPC UA server, the Endpoint configuration needs to be modified for the type of security policy to be implemented.

1 In PSW-UACM, click on the *Endpoint* tab and select each type of security modes to enable.



2 Click **SAVE** when complete.

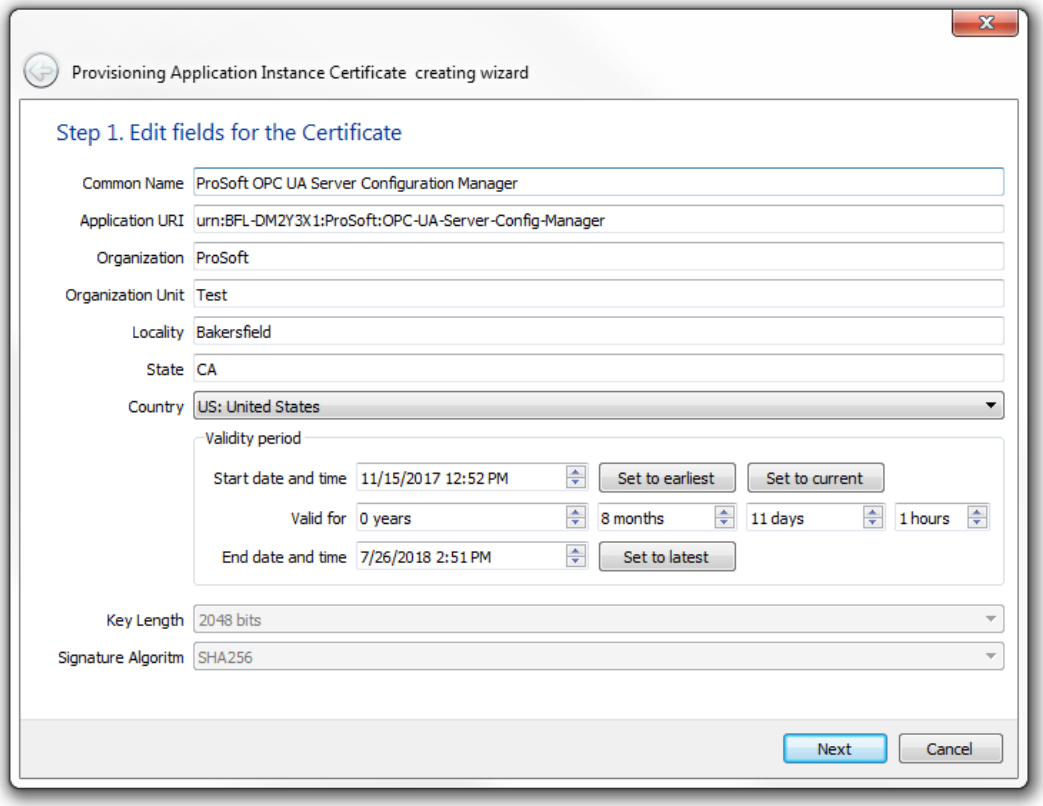
Important: The certificate provisioning steps in the following sections must be completed before using the **DOWNLOAD** option in this tab.

7.2.2 Creating a Provisioning Application Instance Certificate

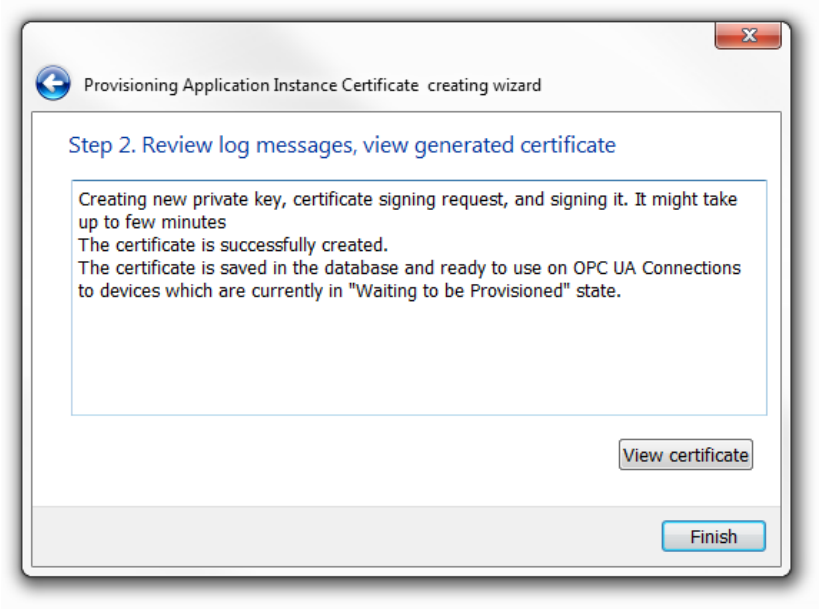
This certificate is used by the PSW-UACM as its OPC UA Application Instance Certificate for device connections, which are in their original state: *Waiting to be provisioned*. This certificate is signed by a factory-supplied provisioning CA Certificate (included in PSW-UACM installation), as well as preinstalled on all devices' default trust list. Therefore, devices will allow a secure connection from PSW-UACM. No OPC UA clients other than PSW-UACM can connect to the device until it is provisioned and the trust list is configured to allow such clients to connect.

The Provisioning Application Instance Certificate can be re-created multiple times, if required.

1. In PSW-UACM, click on **SETUP** > **CREATE PROVISIONING APPLICATION INSTANCE CERTIFICATE**.
2. Enter the application information in the setup wizard. Click the **NEXT** button.



3 The software creates the certificate with the application information. Click **FINISH**.



7.2.3 Creating a CA Certificate

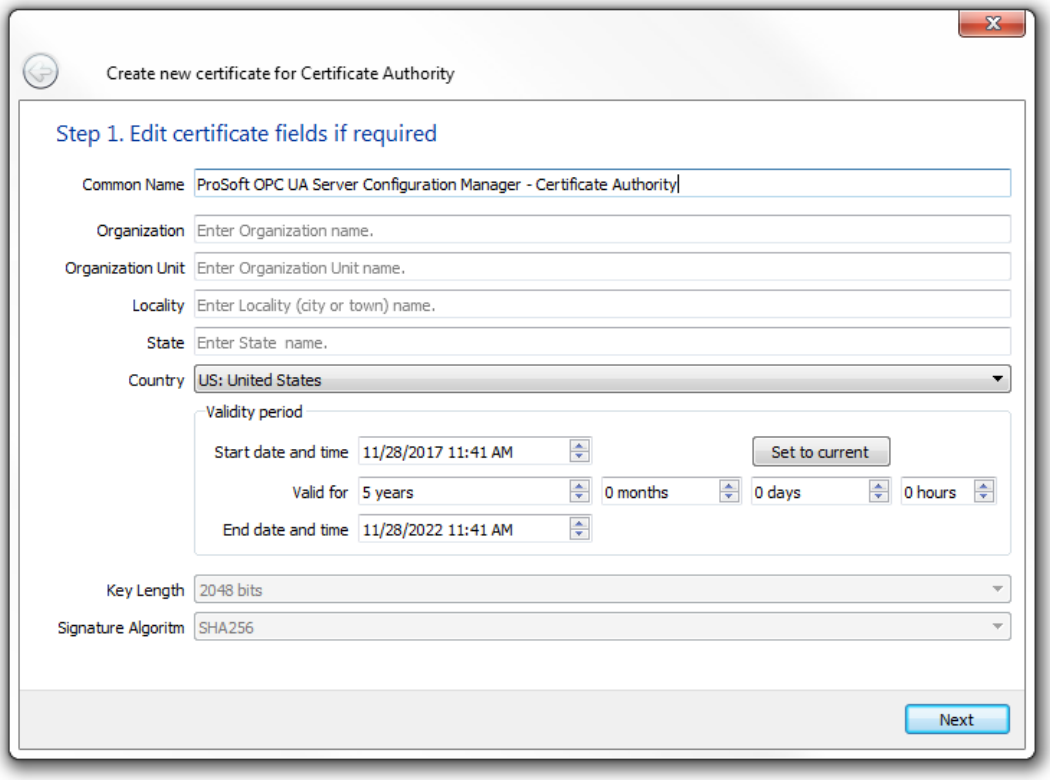
The CA Certificate is used by PSW-UACM to sign newly created device certificates. It is also installed into a device's trust list as part of the provisioning process.

The CA Certificate is used to sign PSW-UACM's own OPC UA Application Instance Certificate, which is used on connections from PSW-UACM to provisioned devices.

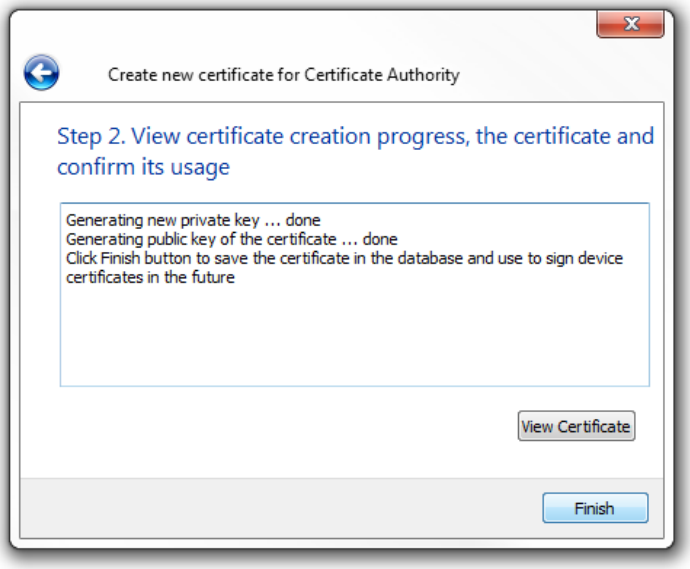
Note that the current version of the gateway does not support the renewal of the CA Certificate. Therefore, it must be created with a lengthy validity period. Once the CA Certificate expires, devices will switch to "not provisioned" mode on the next reboot.

The CA Certificate should be created once, and not re-created until a new version of the PSW-UACM with support of CA Certificate renewal is available.

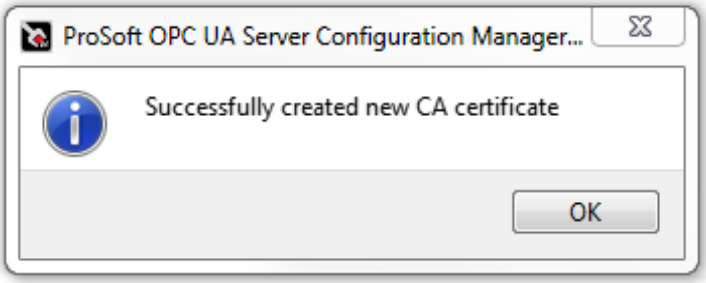
1. In PSW-UACM, click on **SETUP** > **CREATE CA CERTIFICATE**
2. Enter the application information in the setup wizard. Click the **NEXT** button.



3 The software creates the CA certificate with the application information. Click **FINISH**.



4 When prompted, click **OK**.



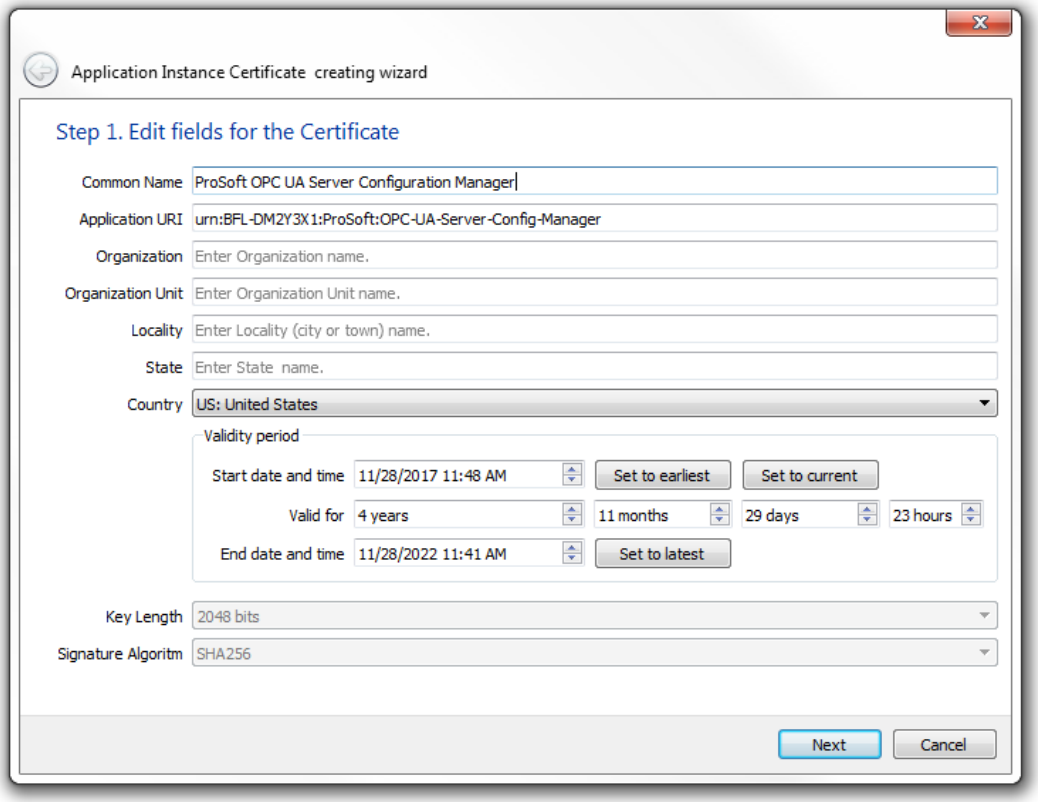
7.2.4 Creating an Application Instance Certificate

This certificate is used by the PSW-UACM as its OPC UA Application Instance Certificate for its connections to devices that are in the *Provisioned* state.

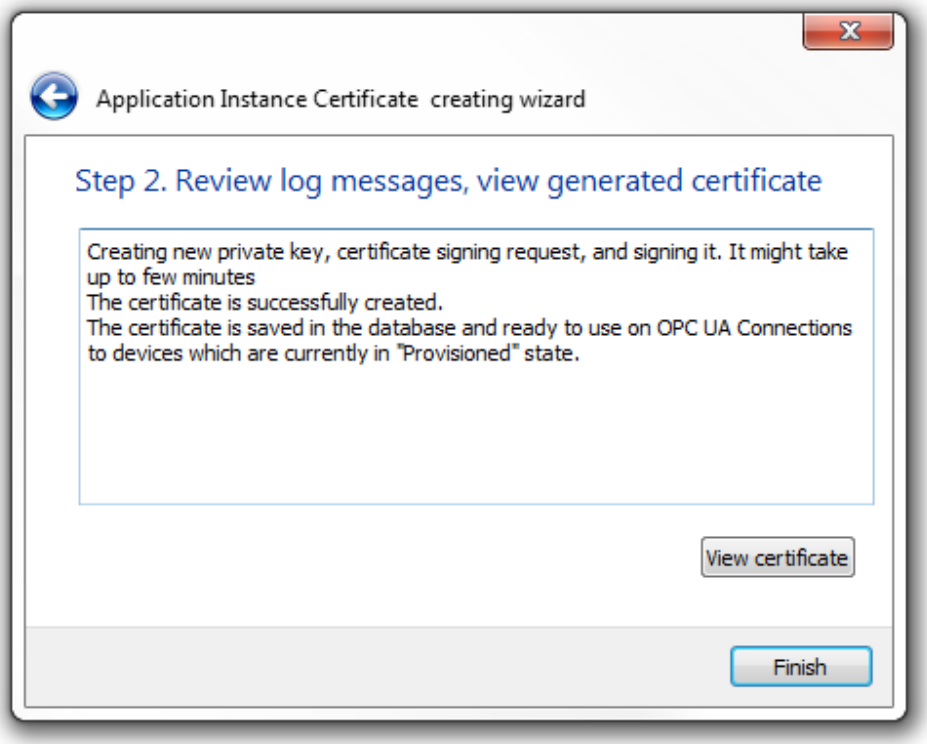
This certificate can be recreated multiple times, if needed.

1. In PSW-UACM, click on **SETUP** > **CREATE APPLICATION INSTANCE CERTIFICATE.**
2. Enter the application information in the setup wizard. Click the **NEXT** button.

Note: The *Application URI* parameter must not contain an underscore '_' character.



3 The software creates the Application Instance certificate with the application information. Click **FINISH**.



7.2.5 Refreshing the Status Tab

Once the certificates have been created, the PSW-UACM software is ready to connect and configure OPC UA Servers.

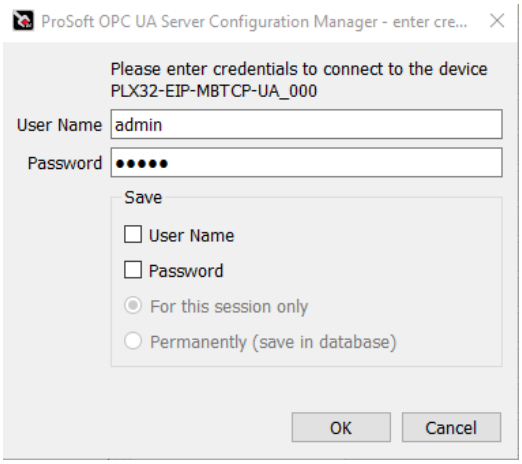
The *Status* tab is used to display the current status of the selected device. Also, a test can be done to check if PSW-UACM can establish connection with the device.

1. In the *Devices* pane of PSW-UACM, select the device to connect to and click on the *Status* tab. This tab displays the current status of the PLX32-EIP-MBTCP-UA Server. Initially, the parameters are blank.
2. In order to connect to the selected device, enter the user name and password in the *Enter Credentials* dialog window. The default credentials for a user with administrator rights are:

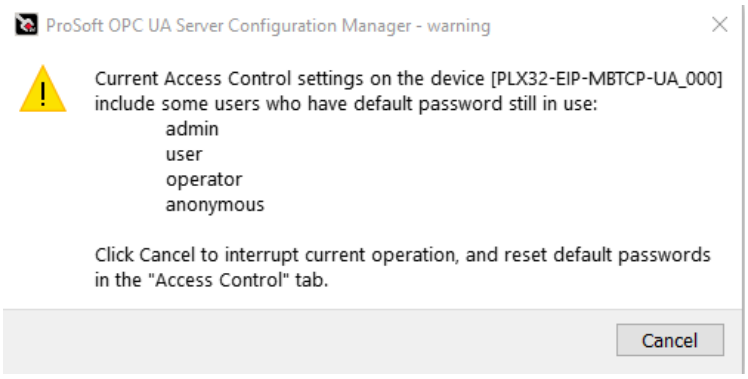
   User Name: **admin**

   Password: **12345**

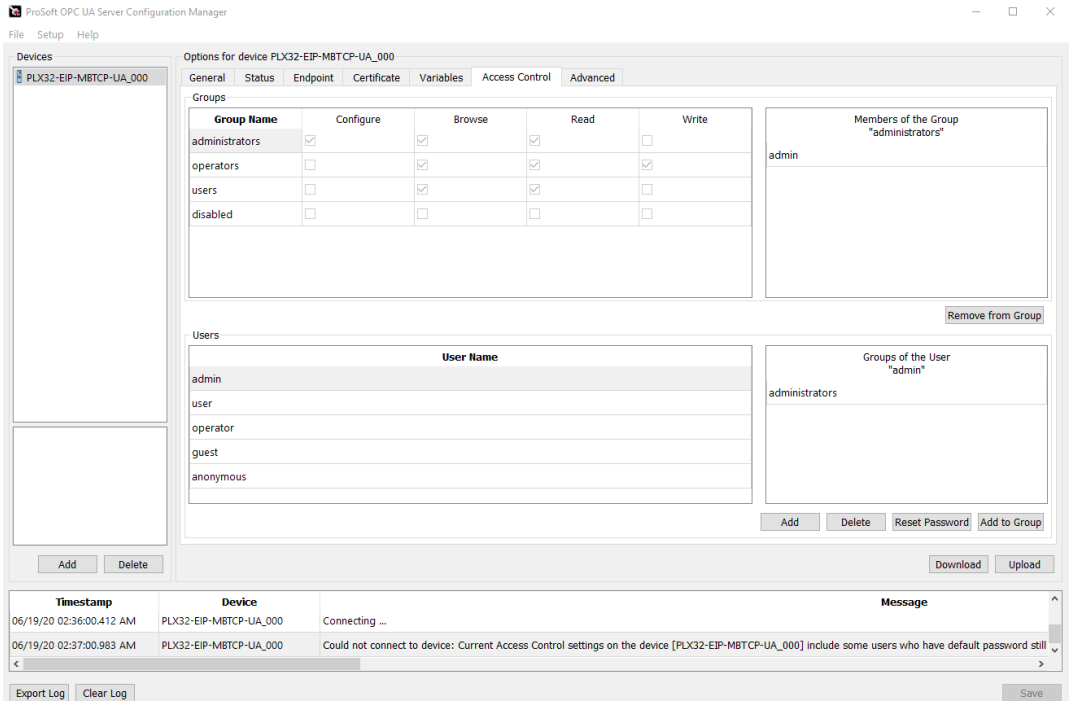
   Save the *User Name* and *Password* for future use by checking the boxes.



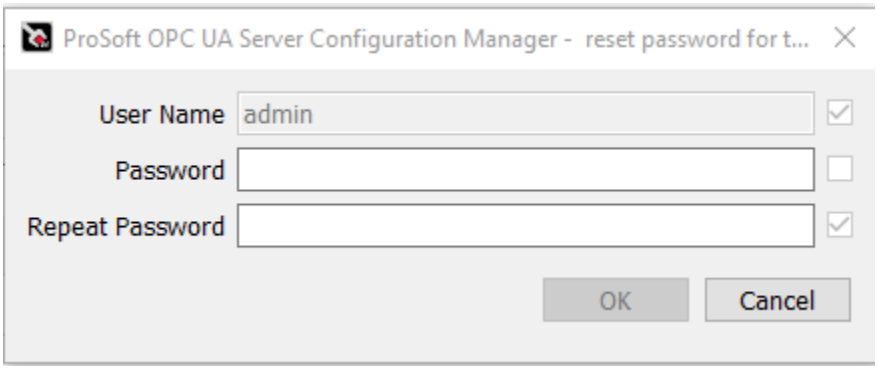
3. Click **OK**.
4. If the default credentials were authenticated, the following warning will appear:



5 From *Access Control* tab, navigate to the *Users* panel and select **admin**.



6 Click on the **RESET PASSWORD** button to assign a new password.

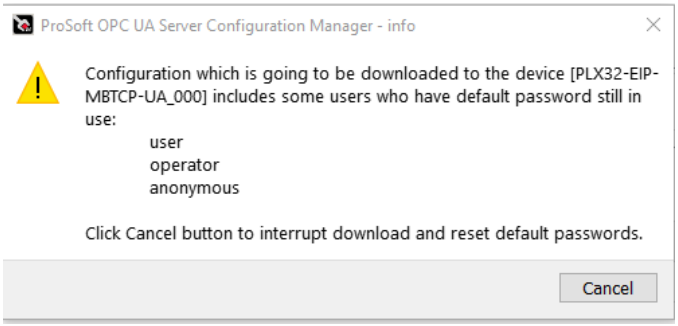


7 The new *Password* and *Repeat Password* entries must match, and meet the following criteria before being changed:

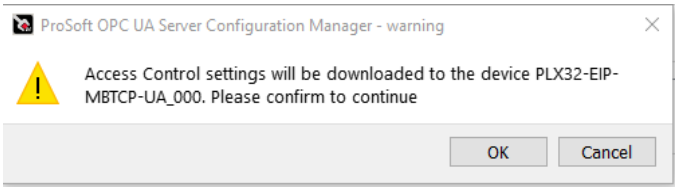
- Minimum 5 characters and maximum 32 characters
- Upper case letter
- Lower case letter
- Digit
- Special character #?!@$%^&*()[]

8 Repeat this process for **user**, **operator**, and **anonymous** users (if needed).

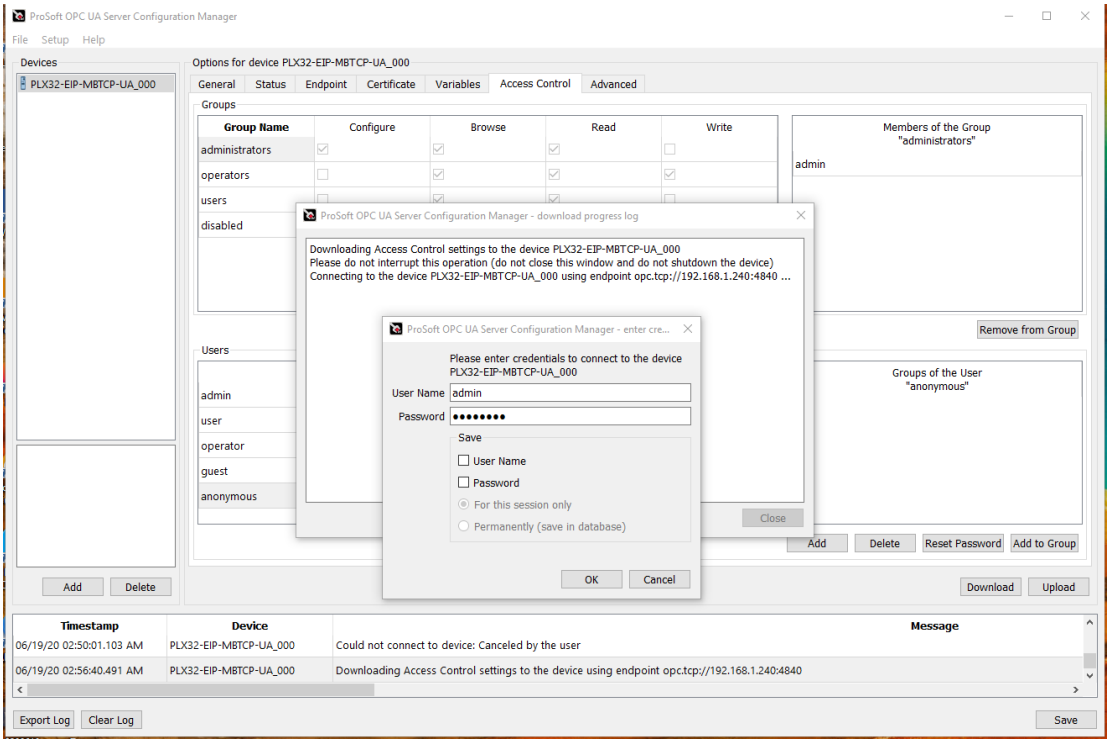
9 If any other user is detected using default documented credentials, the following message will appear to change the default password:



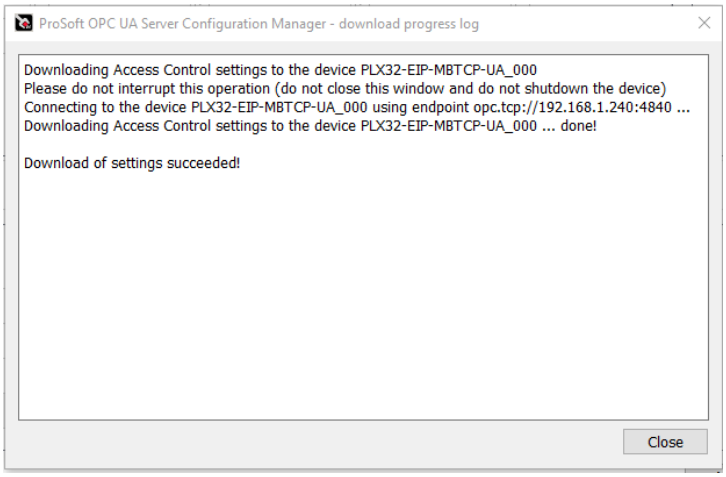
10 When ready, click the **DOWNLOAD** button to download the new credentials to the device.



11 When downloading the new security settings to the device, use the default password (the current credentials stored on the device).

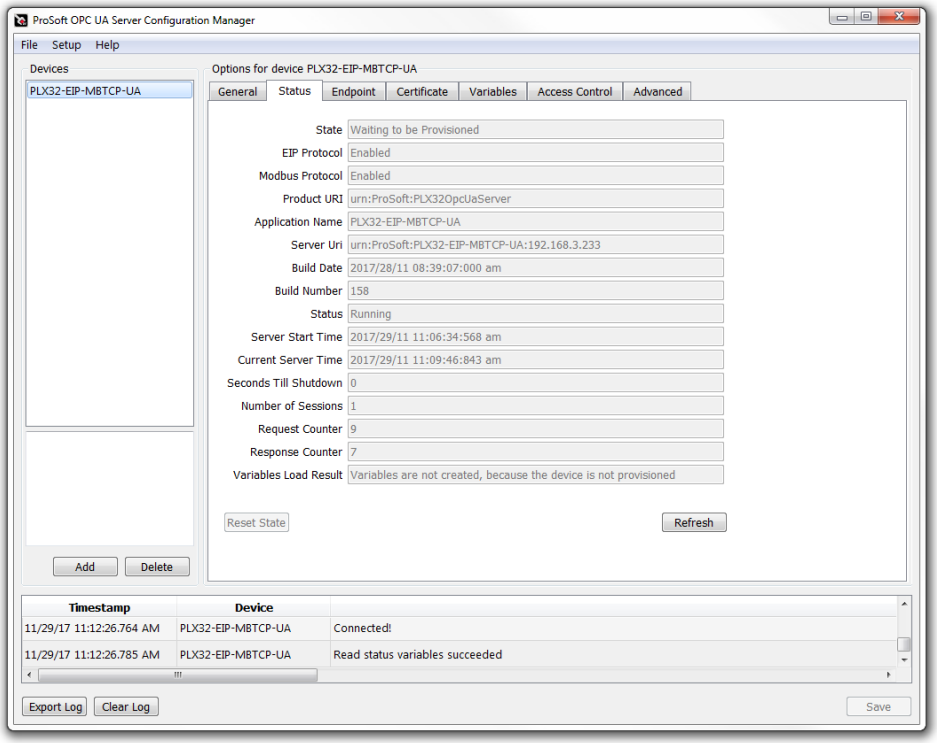


12 The download progress will indicate when the downloaded security settings have been applied.



13 Once default passwords have been changed, resetting password to the default documented password is not permitted since it does not meet validation password requirements.

14 The *Status* parameters are now populated.



Note: Click on the **REFRESH** button to update the *Status* tab.

7.2.6 Creating and Signing a New Certificate

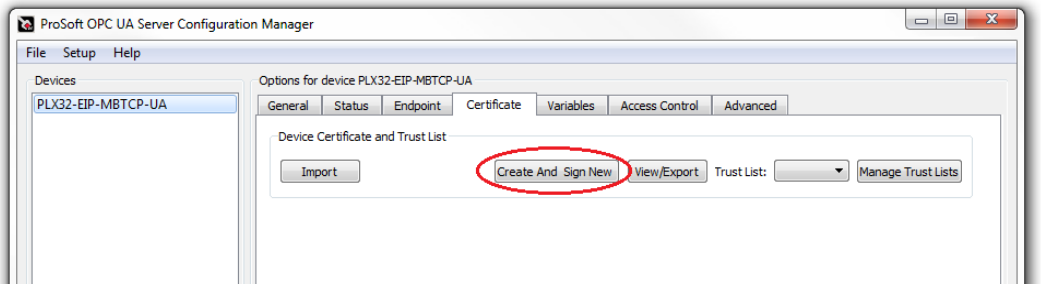
Initially, the PLX32-EIP-MBTCP-UA has a self-signed certificate that is used as its OPC UA Application Instance Certificate. The purpose of this self-signed certificate is for the initial secured communication with the PSW-UACM. In order to enable access to variables, the device's certificate must be replaced by a certificate signed by a CA Certificate. This process is called 'provisioning'. Once the device is provisioned, its variables become available in OPC UA address space.

By default, only the most secured endpoint is enabled. If less-secure endpoints are required, they must be enabled before provisioning starts.

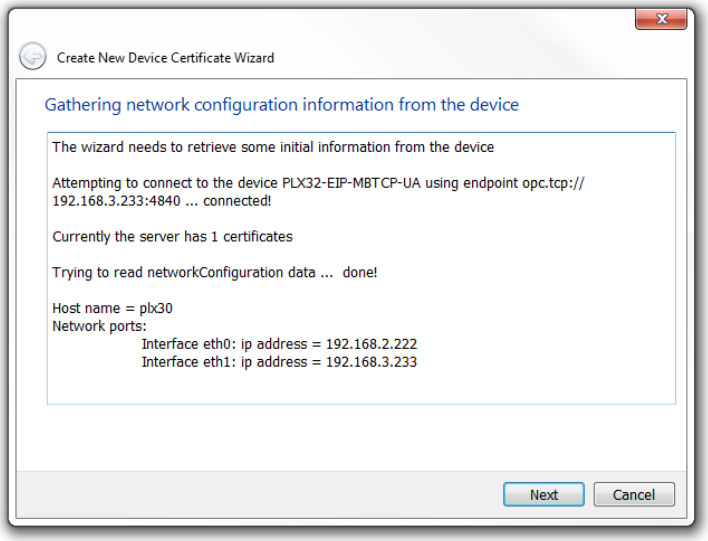
To support all possible endpoints, 2 certificates are required. First, the certificate with type 2048 bit SHA256 must be created, then type 2048 bit RSA128 can be created.

Important: For RSA128 Certificates, downloading the endpoints to the device is mandatory after saving endpoints.

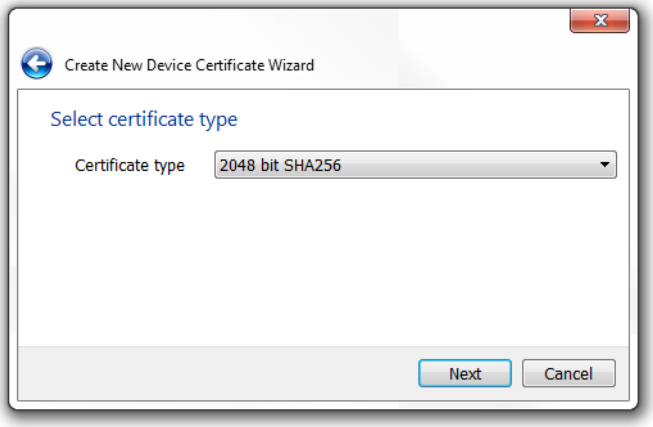
1 From the *Certificate* tab in PSW-UACM, click on **CREATE AND SIGN NEW** button.



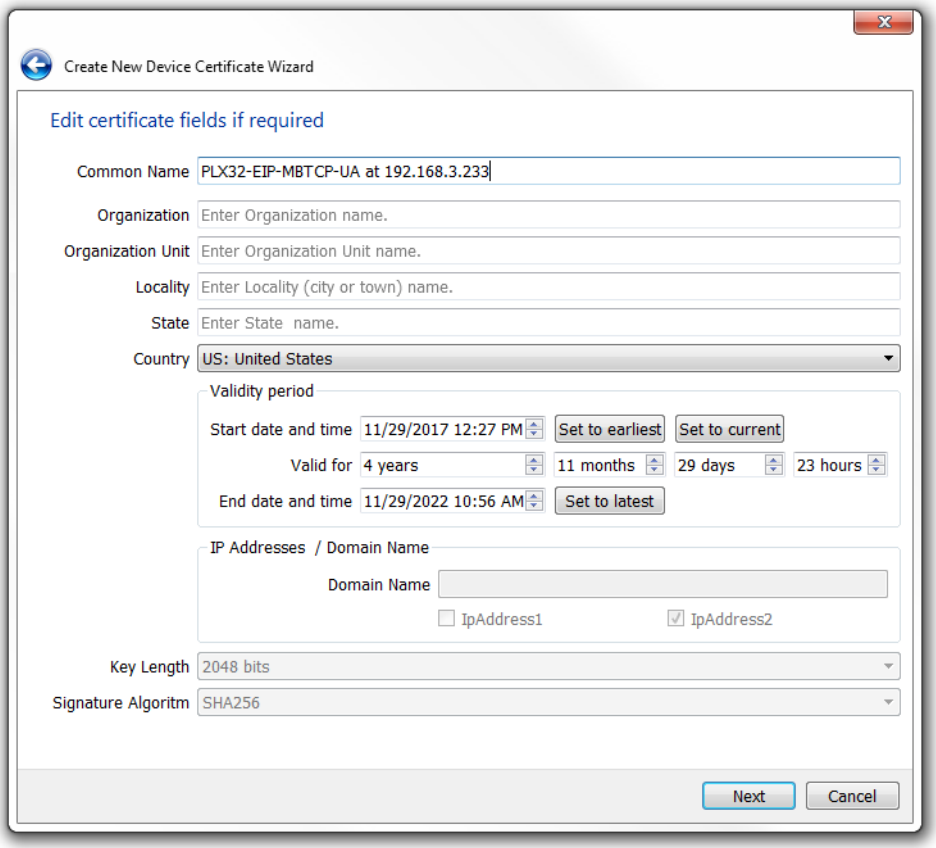
2 Upon connection, click **NEXT**.



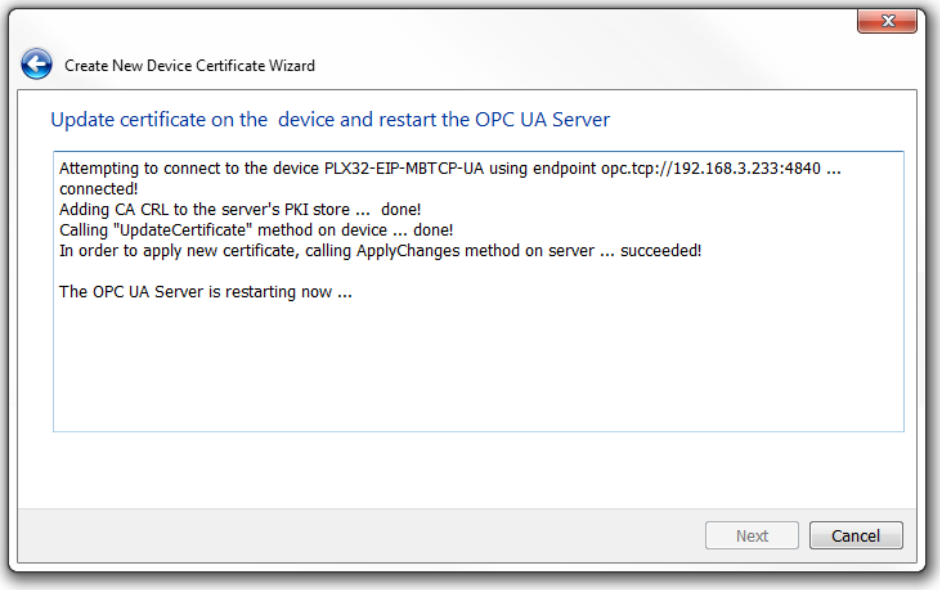
3 Select the *Certificate Type*, then click **NEXT**.



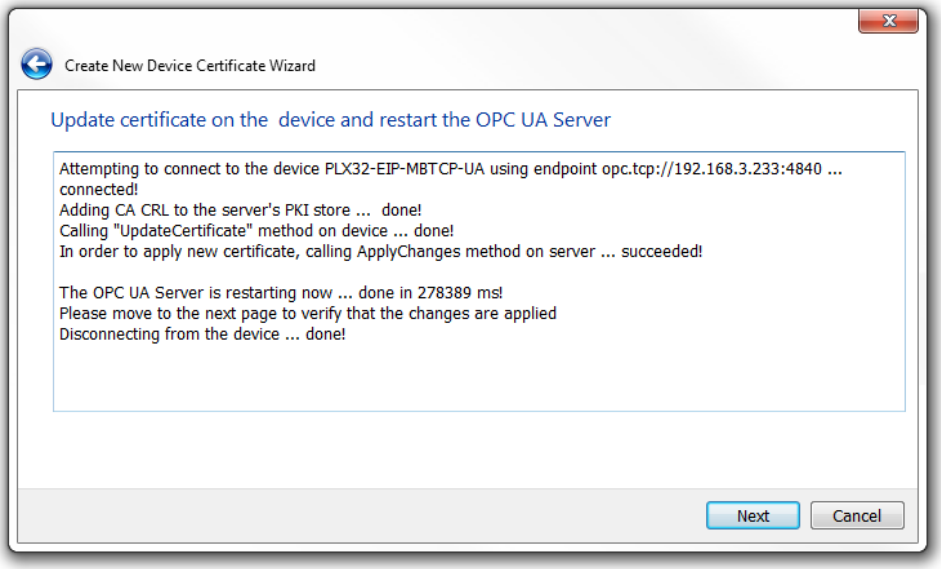
4 Enter the application information in the wizard. Click the **NEXT** button.



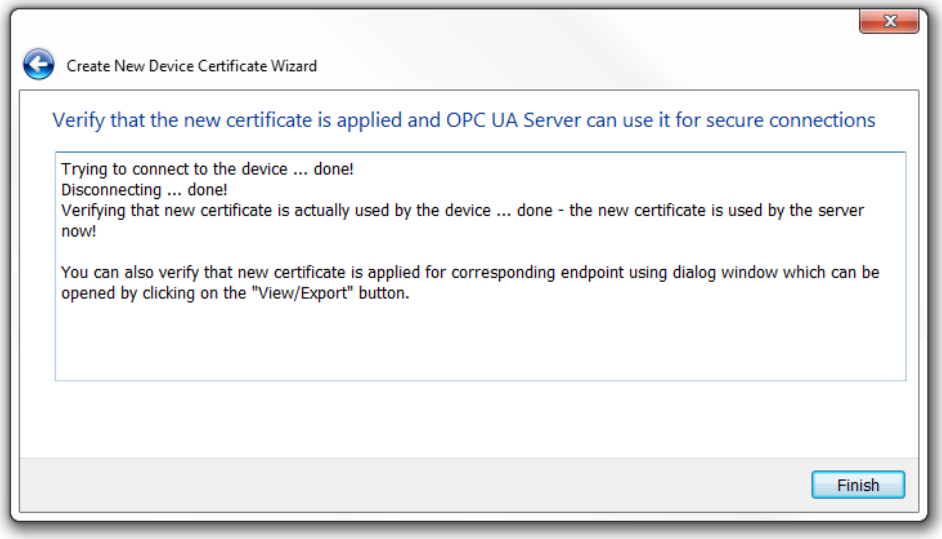
5 Upon successful connection, the new certificate is applied. The OPC UA Server driver of the PLX32-EIP-MBTCP-UA automatically reboots.



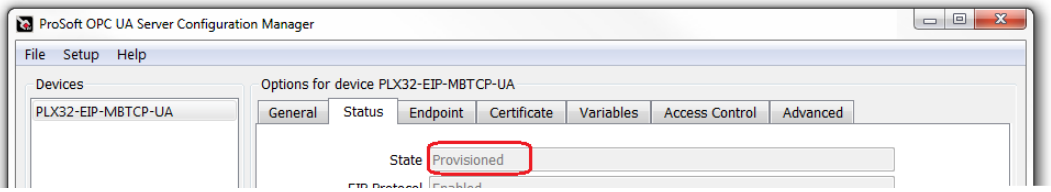
6 After the PLX32-EIP-MBTCP-UA reboots, click **NEXT**.



7 The OPC UA Server verifies and applies the new certificate. Click **FINISH**.



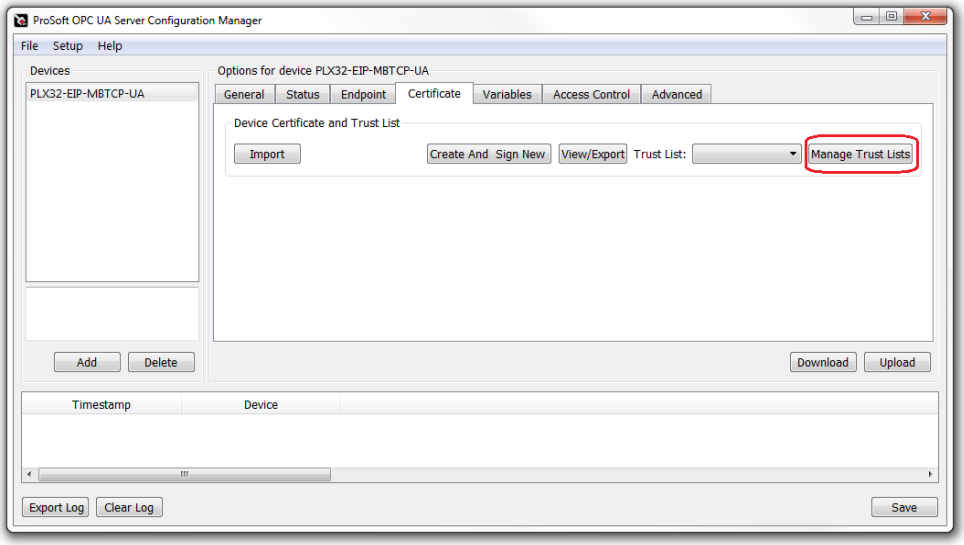
8 The PLX32-EIP-MBTCP-UA Server is now in a Provisioned state. Verify this in the *State* and *Variable Load Result* parameters in the Status tab.



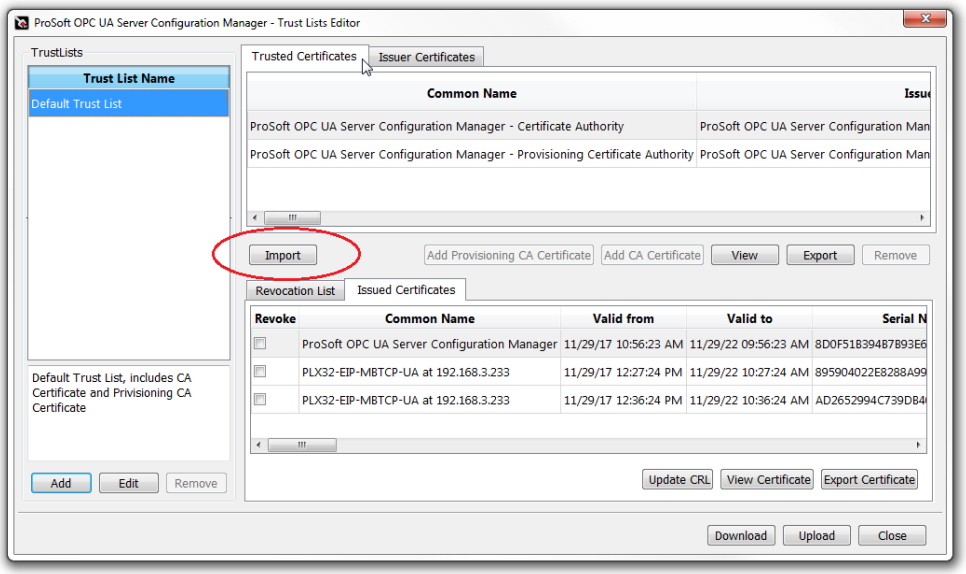
7.3 Importing a Certificate Public Key File

This section describes how to import the Certificate Public Key File of an OPC UA Client into PSW-UACM. This import is required to add the OPC UA Client's Application Instance Certificate into the trust list of the device's OPC UA Server. This allows the client to connect in a secure mode.

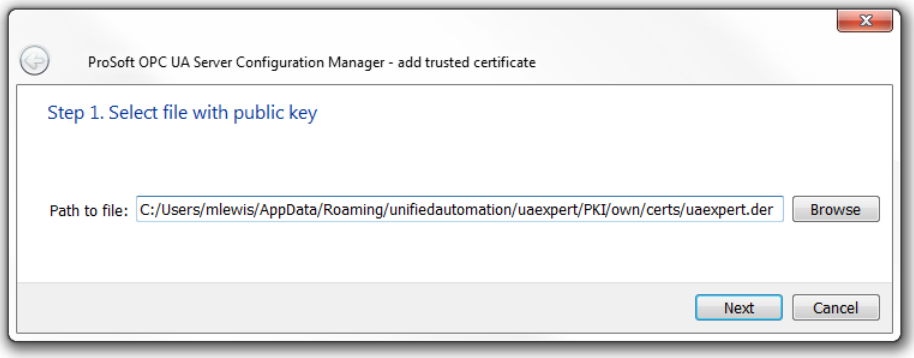
1 From the *Certficate* tab in PSW-UACM, click on the **MANAGE TRUST LISTS** button.



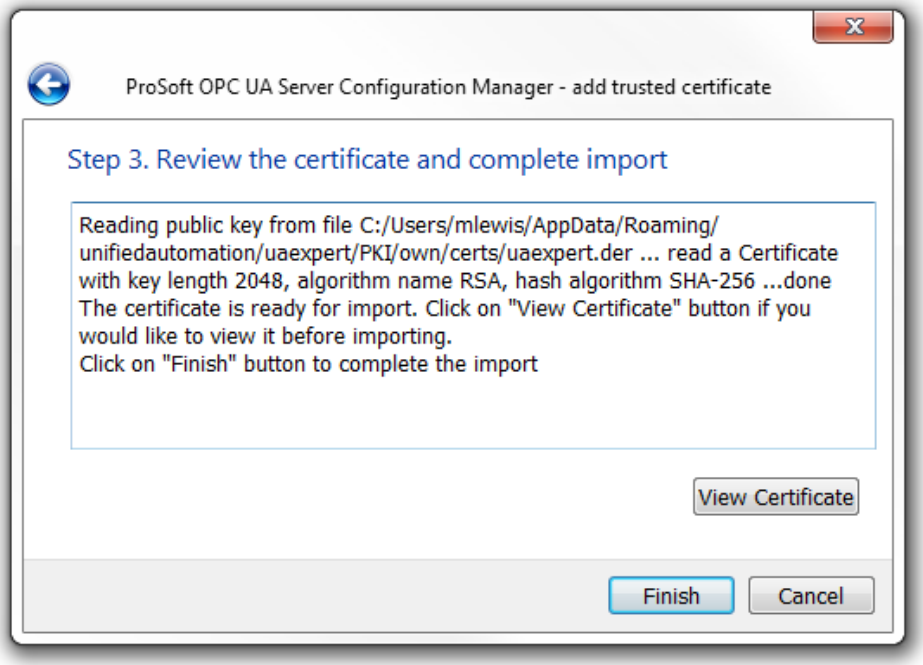
2 The *Trust Lists Editor* dialog opens. Under the *Trusted Certificates* tab, click on the **IMPORT** button.



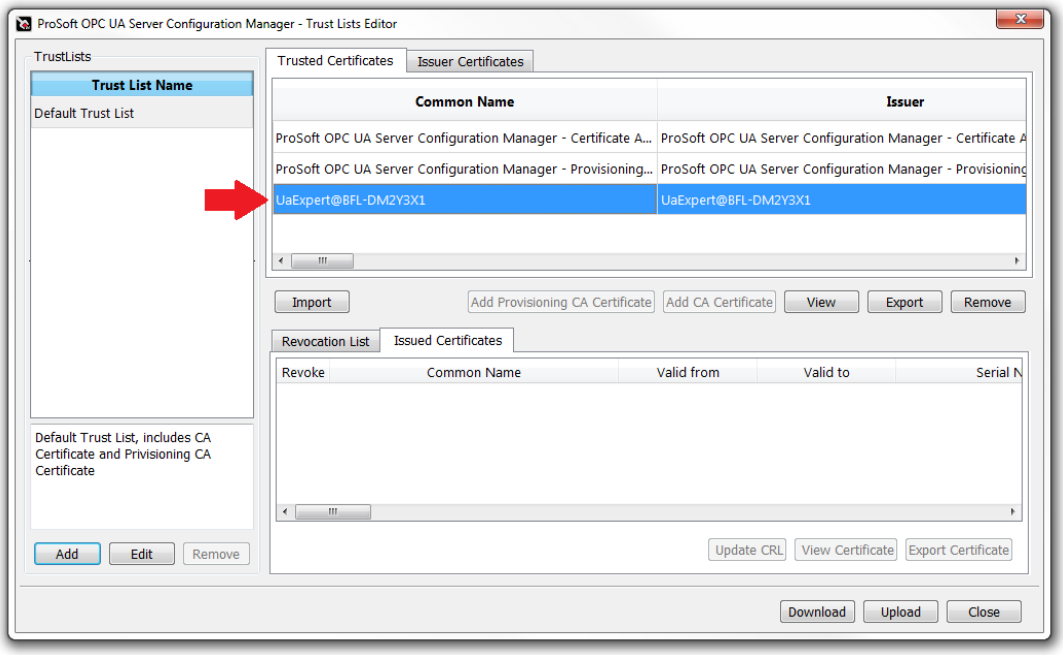
3 In the *Add Trusted Certificate* dialog, browse to the OPC Client's Certificate Public Key **.der* file and click **NEXT**.



4 When the certificate is ready for import, click **FINISH**.



6 The OPC Client certificate is now listed as a trusted certificate.



If the OPC UA Client's certificate chain is longer than 1 (i.e. it is signed by an external CA Certificate, which in turn can be signed by another CA Certificate and so on), then one certificate at a desired trust level should be imported into the *Trusted Certificates* table. The rest of the CA Certificates are placed into the *Issuer Certificates* table. Each of the CA Certificate's corresponding valid CRL must be imported too.

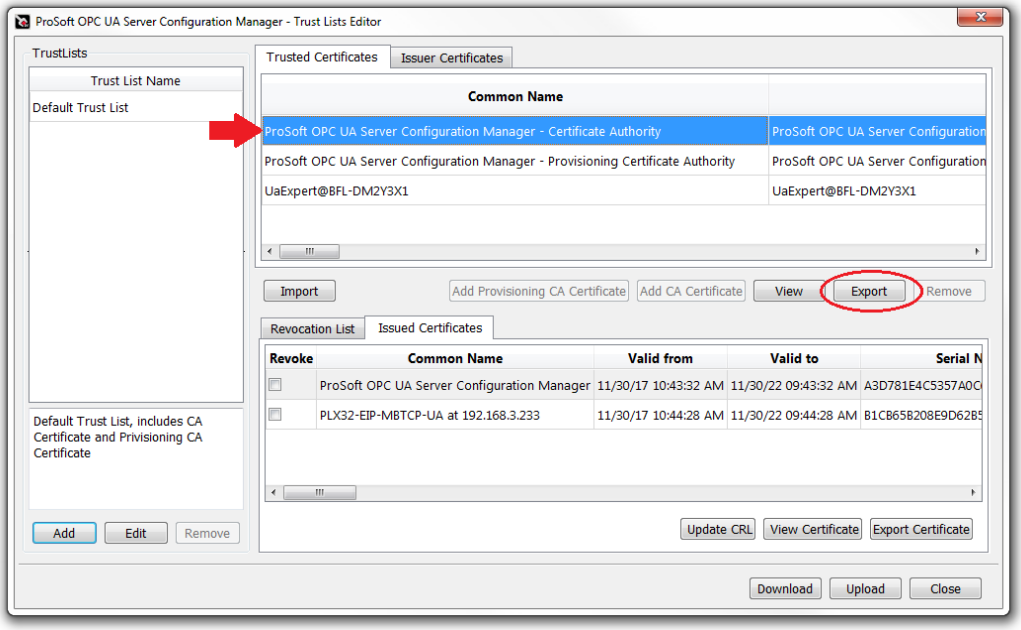
The use of self-signed Application Instance Certificates by OPC UA Server/Client applications is not recommended. If there is no dedicated certificate management system in the company, then PSW-UACM can be used as a central Certificate Authority to sign client certificates. This is done by clicking the **SETUP** > **CREATE INSTANCE CERTIFICATE FOR THIRD PARTY OPC UA APPLICATION** option in the main page of the PSW-UACM.

If clients use such certificates, there is no need to configure the gateway's OPC UA Server to trust these clients. They are trusted based on their certificate (located in the trust list) that is signed by the CA Certificate.

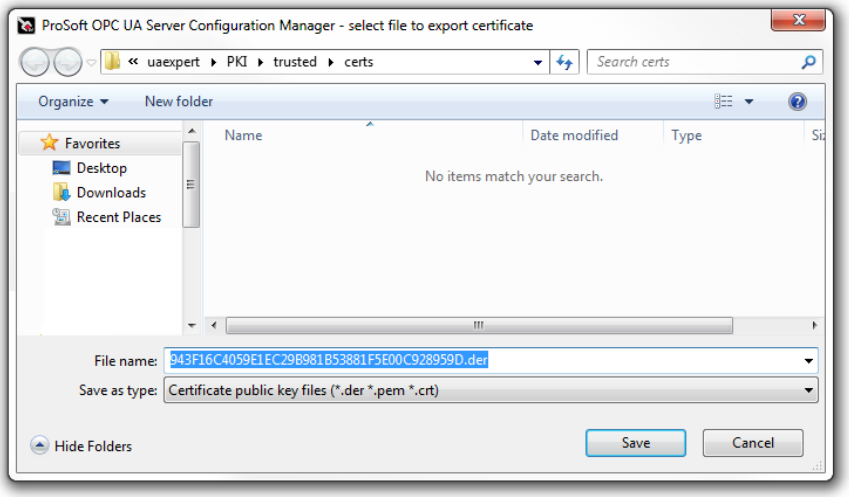
7.4 Exporting the CA Certificate to the OPC Client

To provide a full certificates chain available for an OPC UA client, the Certificate Authority (CA) Certificate must be added to the *Trusted Certificates* list of the OPC Client.

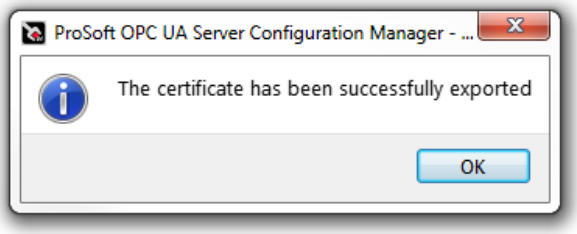
1 In the *Trust Lists Editor* dialog, under the *Trusted Certificates* tab, click and highlight the **ProSoft OPC UA Server Configuration Manager – Certificate Authority**. Click the **EXPORT** button.



2 Browse to the proper OPC Client trusted certificates folder location and save the PLX32-EIP-MBTCP-UA CA Certificate **.der* file. Click **SAVE**.



3 After successful export, click **OK**.



7.5 Revocation List

The Revocation List is similar to a blacklist. It is a list of clients that are not allowed to connect/configure the UA Server. Without the Revocation List, the CA Certificate is not considered valid by the OPC UA Server.

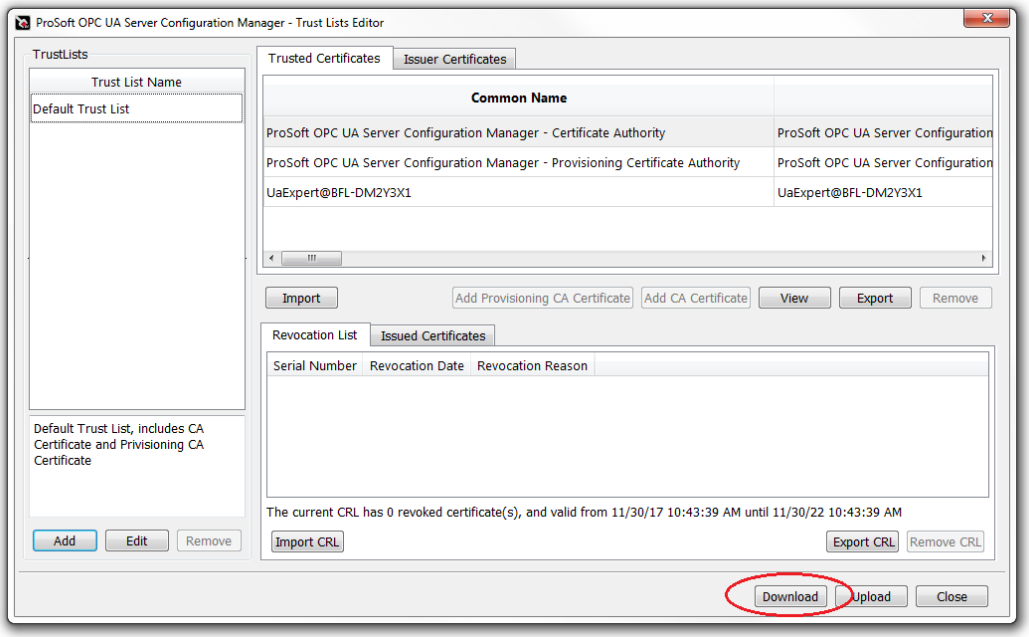
Note: The Certificate Revocation List field is blank when it does not contain a listing of Revoked Certificates.

1 In the *Trust Lists Editor* dialog, under the *Revocation List* tab, click on the **EXPORT CRL** button to add the certificate of the revoked UA Client.
2 Follow the prompts to complete the export.

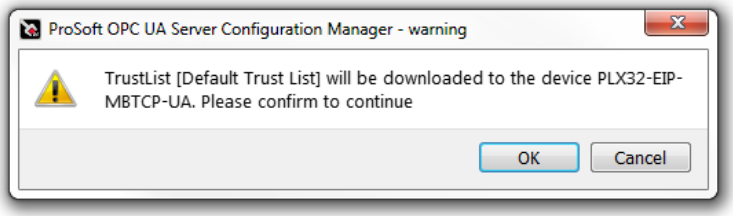
7.6 Downloading the UA Server Configuration to the Gateway

Once the client certificates are added to the trust list, it is ready to be downloaded to the gateway.

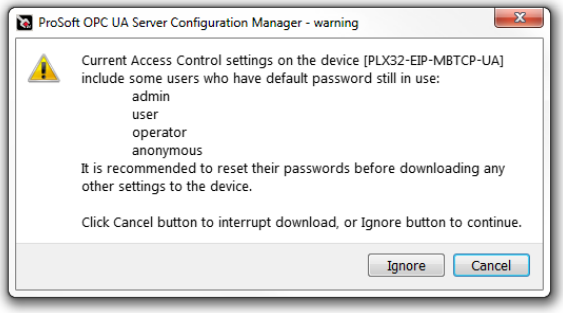
1 In the *Trust Lists Editor* dialog, click on the **DOWNLOAD** button.



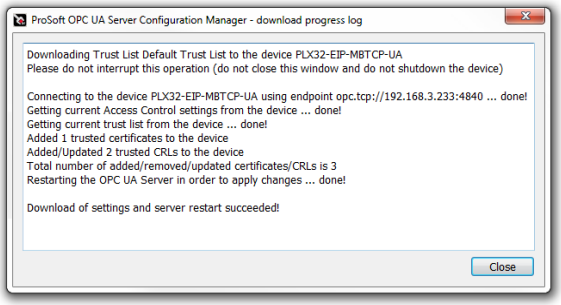
2 A TrustList warning appears, click **OK**.



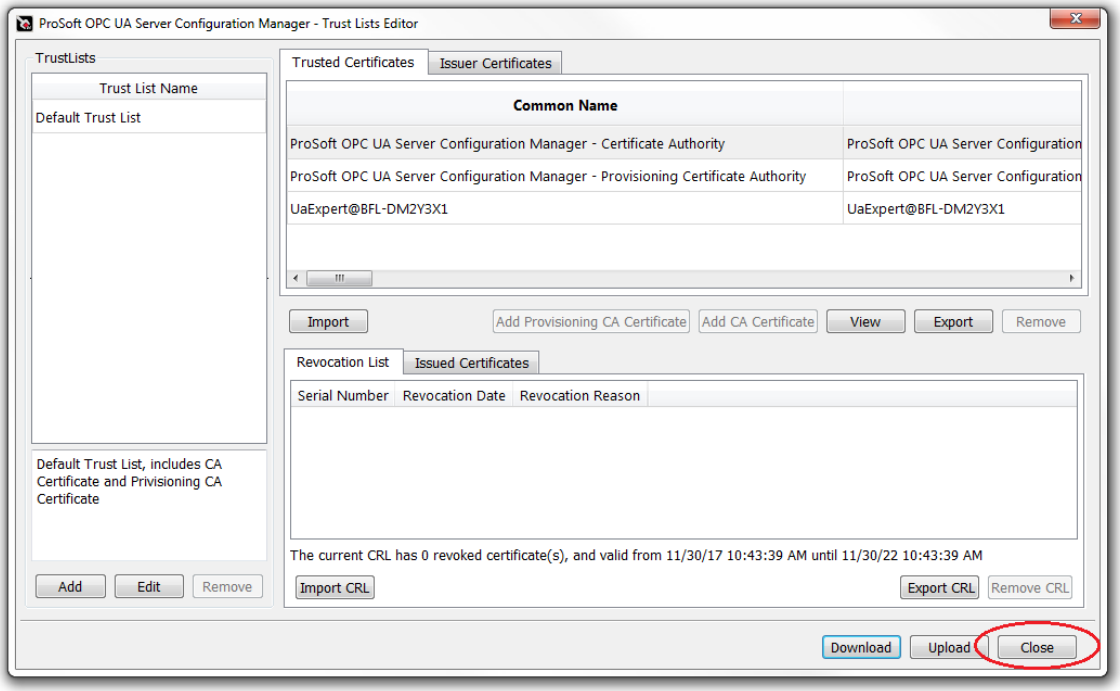
3 An *Access Control* warning may appear. In a production environment, default passwords must be reset to custom passwords to reduce a security threat. Click the **IGNORE** button to continue without changing the default password.



4 The download process reboots the UA Server driver. Click the **CLOSE** button when complete.



5 When the configuration and download of the trust list is complete, click the **CLOSE** button in the *Trust Lists Editor* dialog.



7.7 User Access Control

The *Access Control* tab contains the configuration of the permissions for assigned users. An individual account is set up as a *User*. Each *User* is then placed in a *Group* with certain permissions.

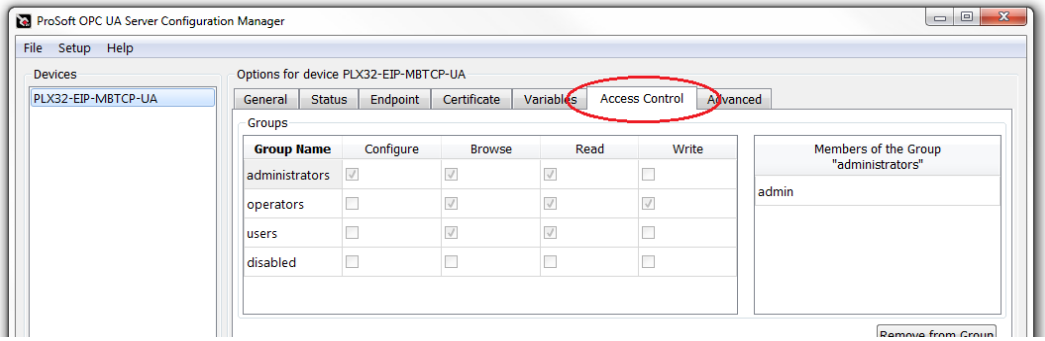
The default credentials are as follows:

User: **admin**

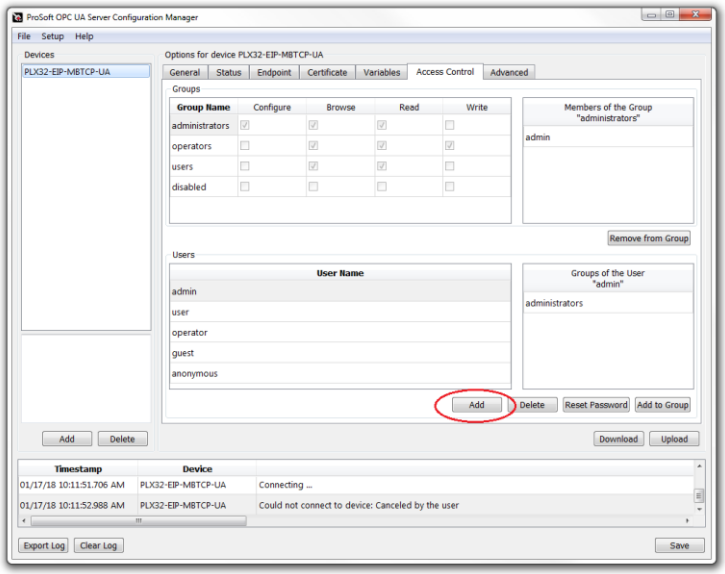
Group: **administrators**

7.7.1 Adding a User

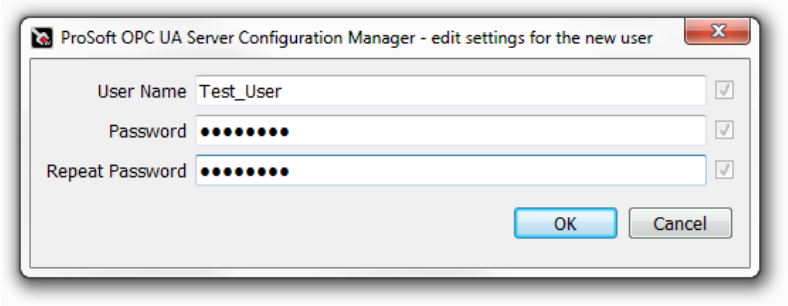
1 In PSW-UACM, click on the *Access Control* tab.



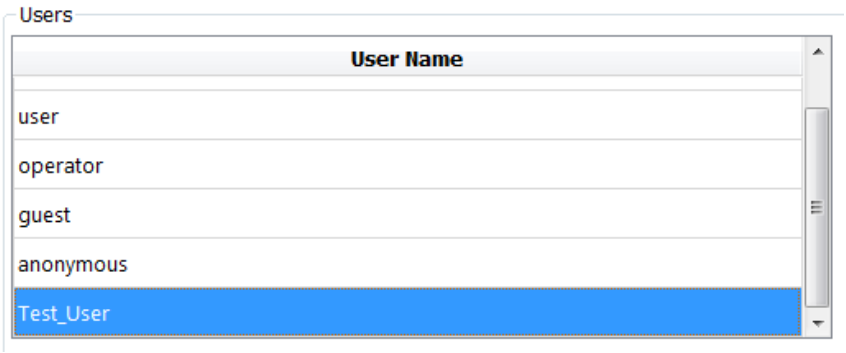
2 Click the **ADD** button to add a new *User*.



3 Enter a *User Name* and *Password* and click **OK**.

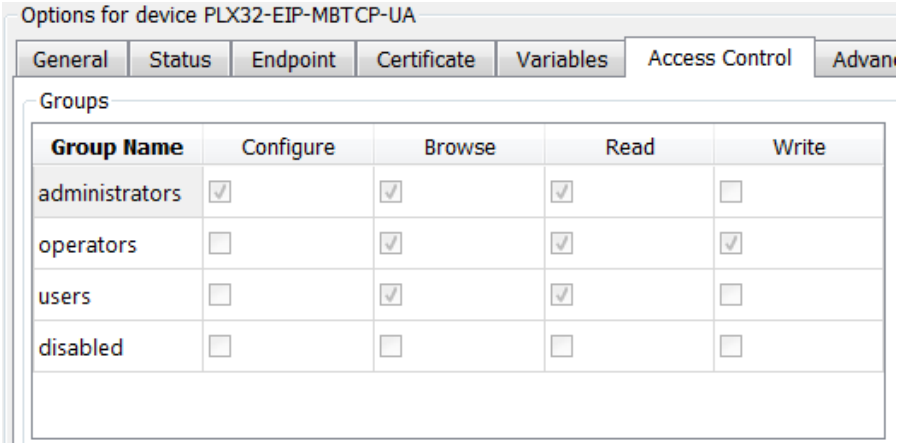


4 The new account now appears in the *Users* section.

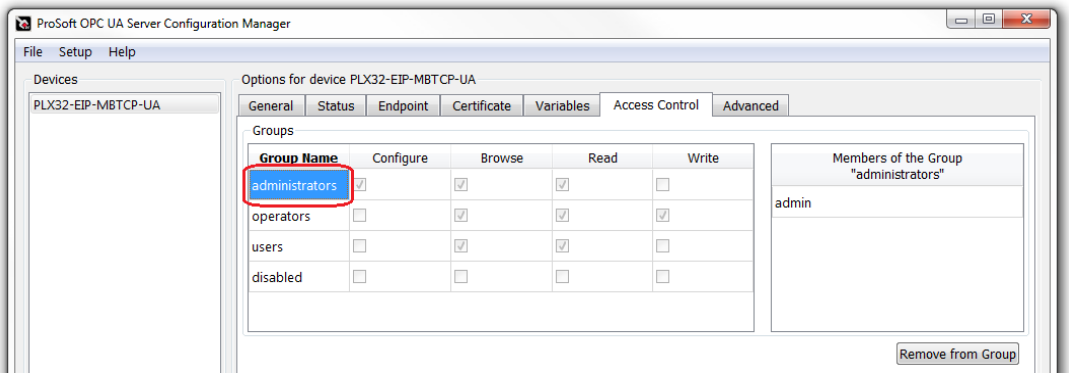


7.7.2 Adding a User to a Group

Each *Group* has their own set of permissions that individual *Users* can be assigned to. The default *Group* permissions cannot be edited.

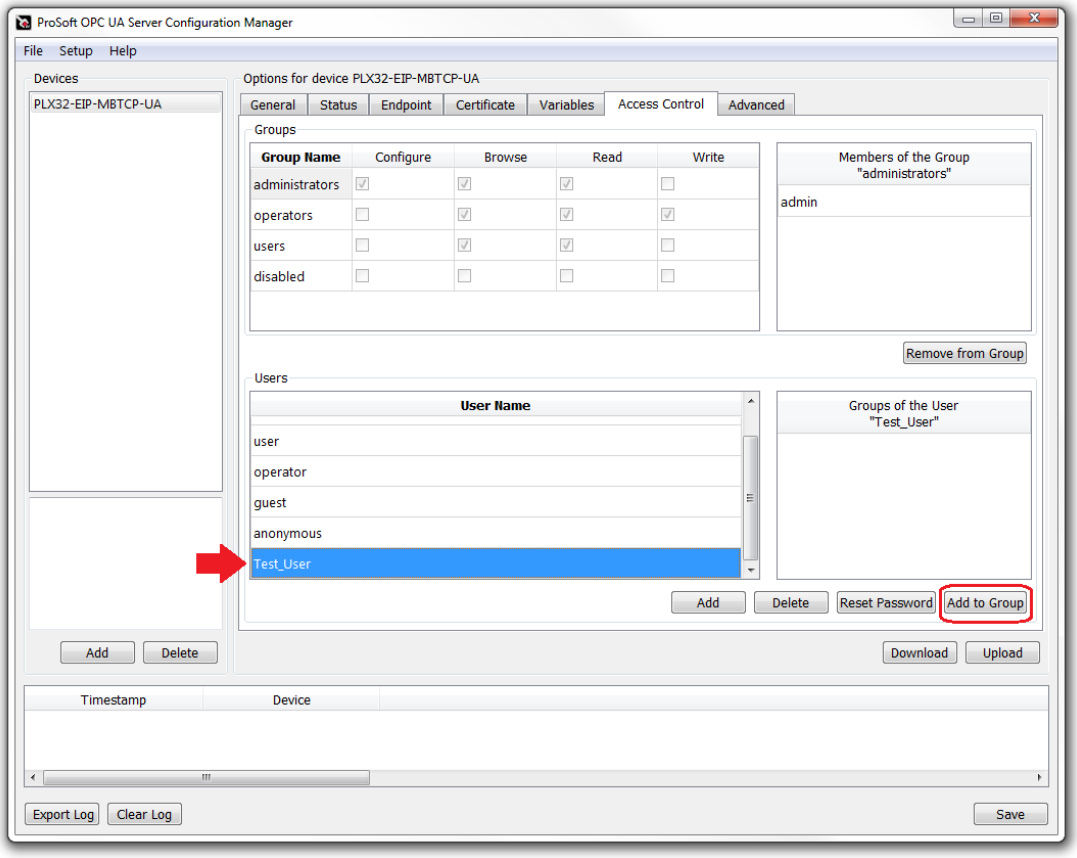


1 Under the *Groups* section in the *Access Control* tab, highlight the desired *Group Name* that the *User* will be placed into.

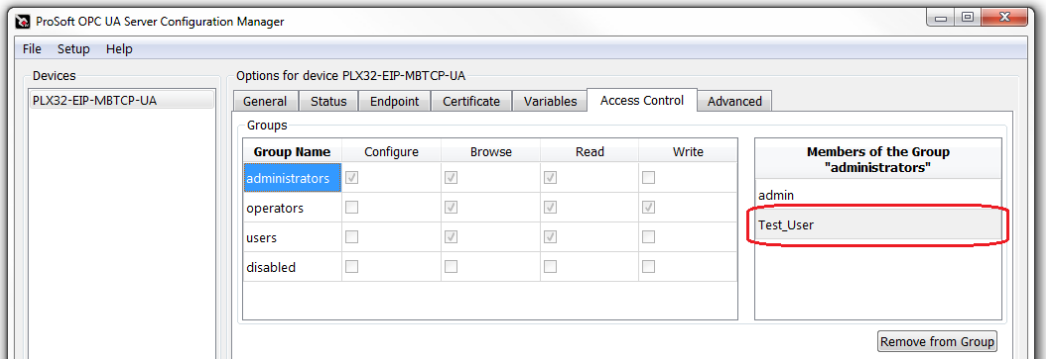


Notice the “Members of the Group” section on the right side of the window. These are the current *Users* that are assigned in the highlighted *Group*.

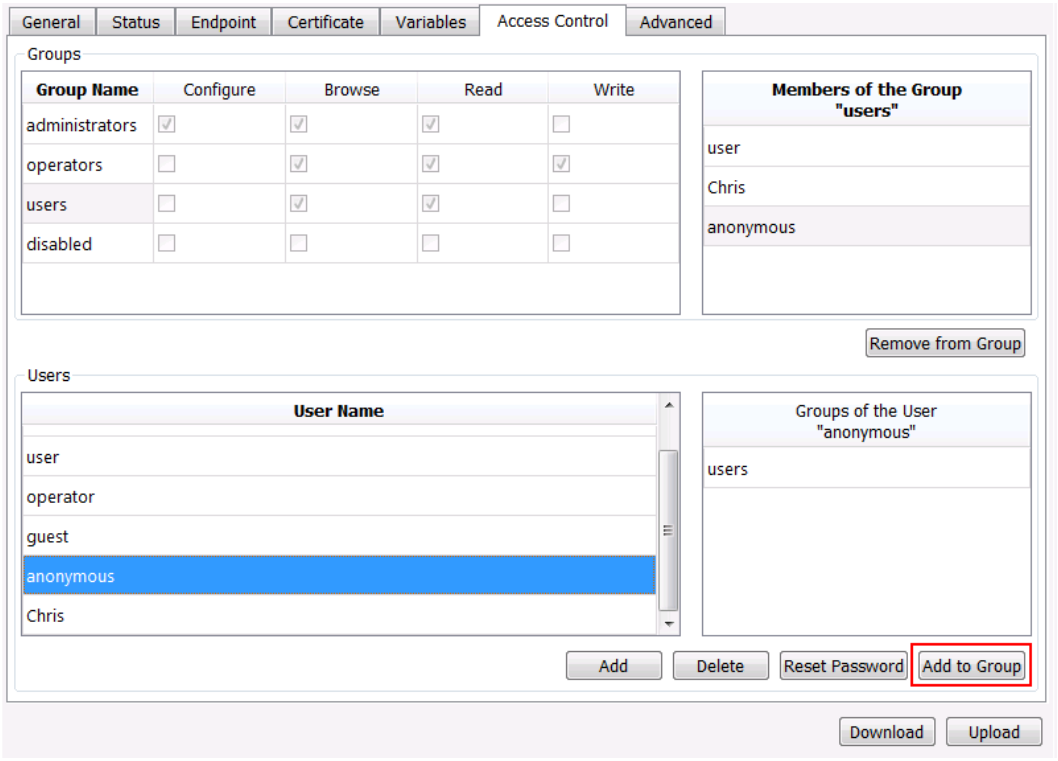
2 Highlight the desired *User* to add to the *Group*, then click the **ADD TO GROUP** button.



3 The *User* has now been added to the *Group*.



4 **Tip for *anonymous* users:**
If the selected *Endpoint* security mode uses the *Anonymous* Token Type, be sure to add the *Anonymous* user to the appropriate group.



In the image above, the anonymous user is added to the “users” group, which has Browse and Read access only.

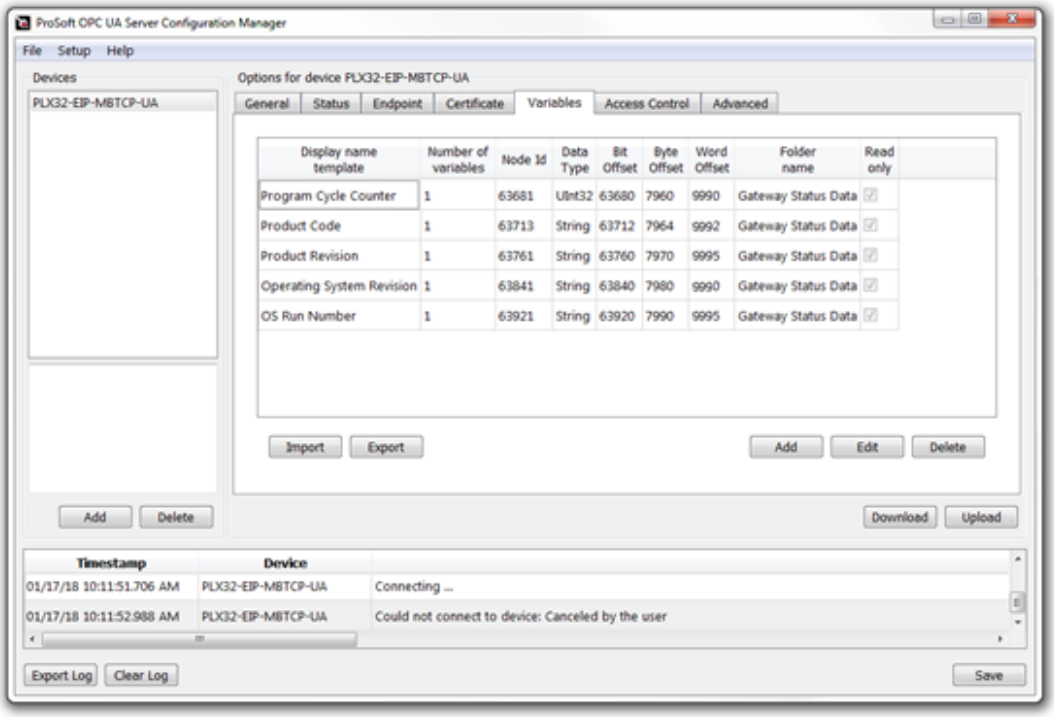
Important: The Anonymous user must be added to a group; otherwise, the connection to the UA Server may become unstable.

5 When ready, download the changes to the PLX32-EIP-MBTCP-EIP by clicking the **DOWNLOAD** button. *Access Control* setting edits do not require OPC UA Server reboot.

7.8 Creating Tags

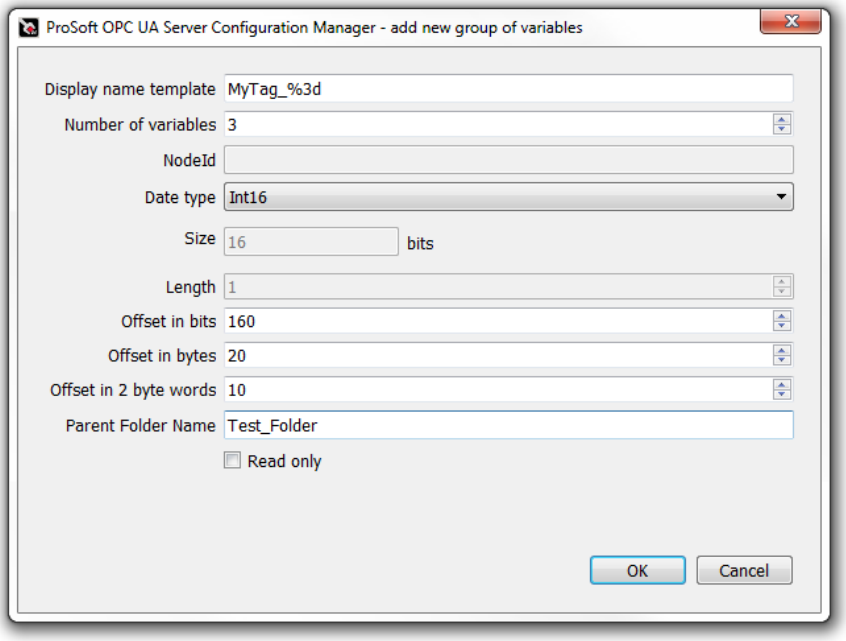
The UA Client accesses data within the UA Server of the PLX32-EIP-MBTCP-UA by using tags. These tags are mapped to the PLX32-EIP-MBTCP-UA's 10,000-register user data range. Creating such tags are done in the PSW-UACM software.

1. Within PSW-UACM, click on the *Variables* tab. This tab displays the current tags that are mapped to the PLX32-EIP-MBTCP-UA Server. By default, there are five PLX32-EIP-MBTCP-UA gateway status tags mapped within the 10,000-register user data range. This data in this range is accessible by the EIP and MBTCP drivers, allowing data exchange across different protocols.



2. To create a new tag, click the **ADD** button.

3 The *Add New Group of Variables* dialog opens.

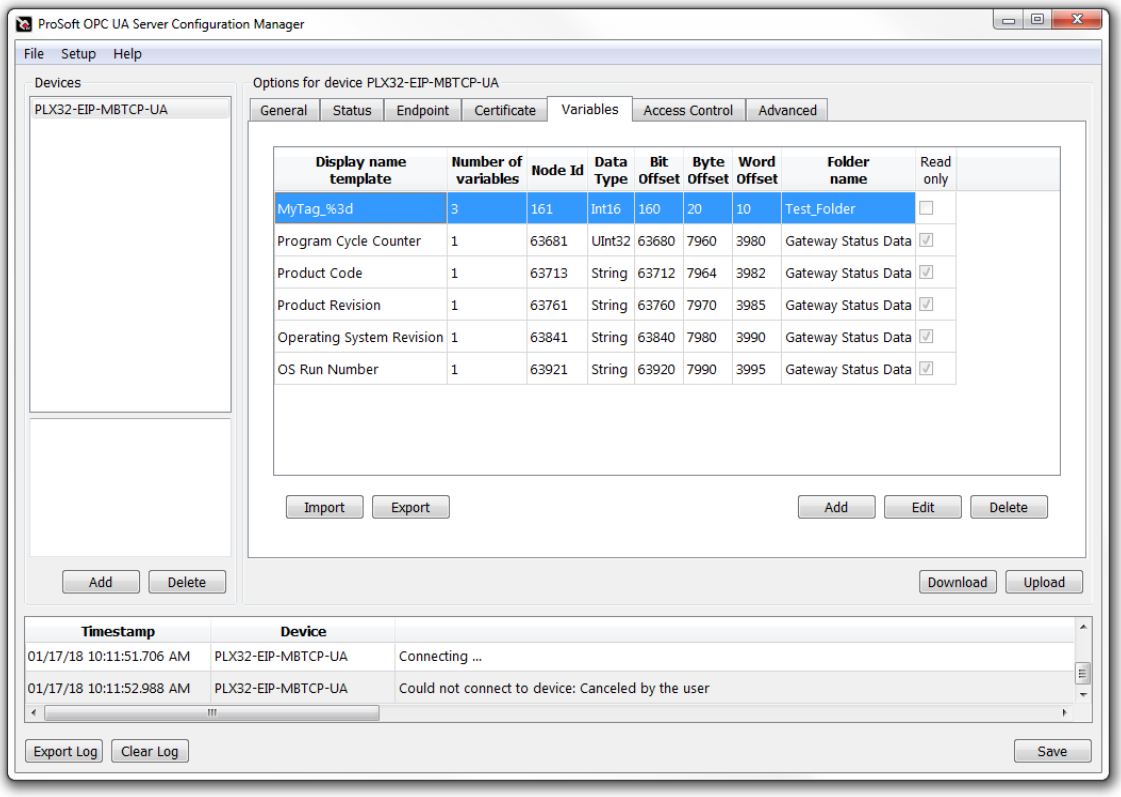


4 Enter the values for each parameter.

Parameter	Description
Display Name Template	Template string used to generate display name of OPC UA variable(s). Note that 1 or more variables can be created for each record in the *Variables* tab. If more than 1 variable is created, the desired number of variables in entered in the *Number of variables* field. Also, the *Display Name Template* field must contain a C-style format string to include the index of the variable, starting from 1. **Example:** If the *Display Name Template* is defined as **MyTag_%03d**, and the *Number of Variables* is set to '3', then the following tags are produced: MyTag_001 MyTag_002 MyTag_003
Number of Variables	Number of variables in the group. In the example of above; 3
NodeId	Node ID of variable. This parameter is only applicable if: *Number of Variables* = **1** *Data Type* = **ByteString** Once the Variable is created, its *NodeId* is automatically assigned a value from 1 to 63992. It is dependent on the starting bit index of the byte (8 bits) or word (16 bits) that the variable is assigned to within the PLX32-EIP-MBTCP-UA database. **Example:** If a set of three Int16-type variables is assigned to PLX32-EIP-MBTCP-UA database index 2000, 2001 and 2002, the NodeID = 32001 for the 1st variable, NodeID = 32017 for the 2nd variable, and NodeID = 32033 for the 3rd variable.
Data Type	Data type of variable
Size	Size of data area used by a single variable in gateway memory, in bits
Length	Length of variable, in bytes. This parameter is only applicable if: *Data Type* = **String** or **ByteString**
Offset in Bits	Variable's position, starting at gateway memory 0, in bits.
Offset in Bytes	Variable's position, starting at gateway memory 0, in bytes.
Offset in 2-byte Words	Variable's position, starting at gateway memory 0, in words (2 bytes).
Parent Folder Name	Name of the parent folder, relative to the folder **Root/Objects/PLX32-EIP-MBTCP-UA**. If it is empty, the variables are created in this folder. This parameter is applicable in the UA Client's Address Space. It displays the variables that the UA Client is receiving from the UA Server. **Example:** If the *Parent Folder Name* is "MyInt16TypeTags" and there are three "Int16Tags_%03d" tags configured, the UA Client displays the tags in the following manner: PLX32-EIP-MBTCP-UA |_ MyInt16TypeTags |_ Int16Tags_001 |_ Int16Tags_002 |_ Int16Tags_003

5 Click **OK**.

6 The new variable is now displayed in the *Variables* tab.



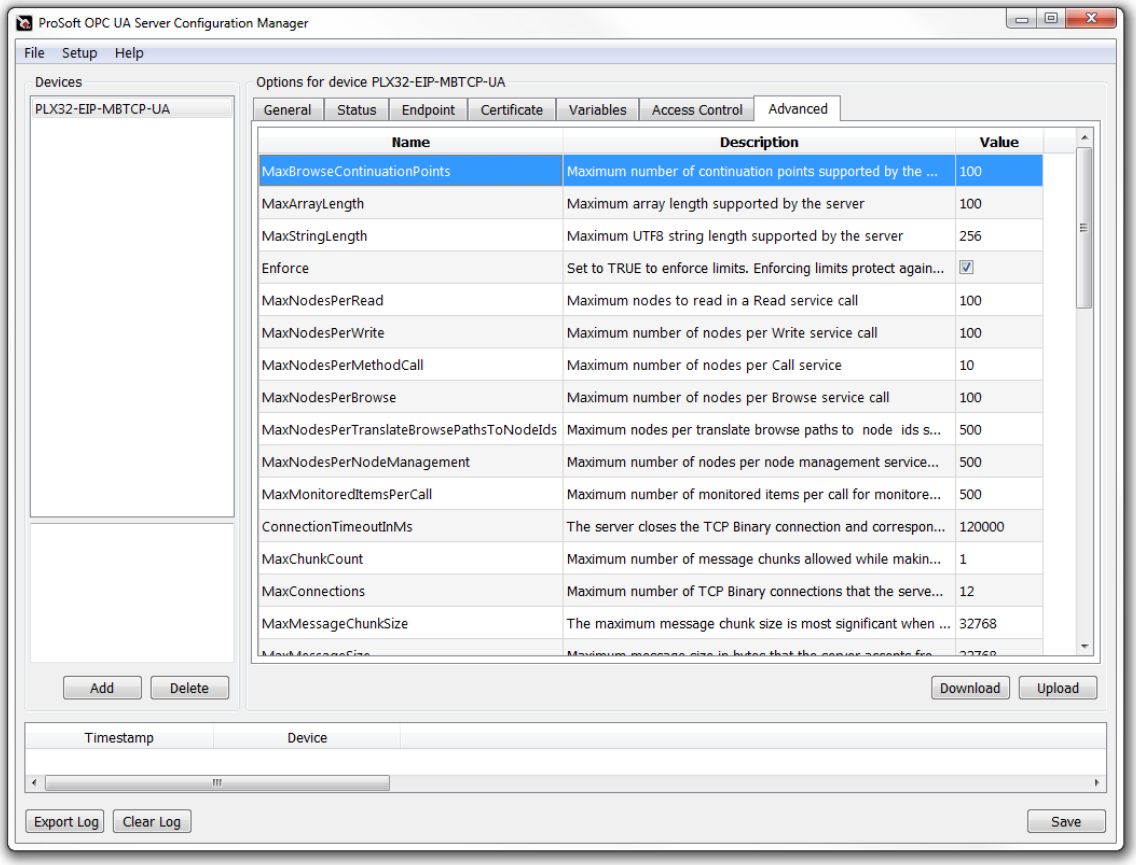
7 In the example above, the new tags (**MyTag_001, MyTag_002** and **MyTag_003**) occupy three 16-bit words, starting at PLX32-EIP-MBTCP-UA gateway memory 10. The EIP and MBTCP drivers can then access these tags.

8 When ready, save the configuration by clicking the **SAVE** button.

9 PSW-UACM does not perform a complete validation of tag configuration settings. If the same memory area is assigned to more than one variable, all variable configurations will fail to load upon OPC UA Server reboot. This error is reported in the *Variables Load Result* field of the *Status* tab.

7.9 Advanced Tab

The *Advanced* tab contains OPC UA operational parameter names, descriptions, and adjustable values.



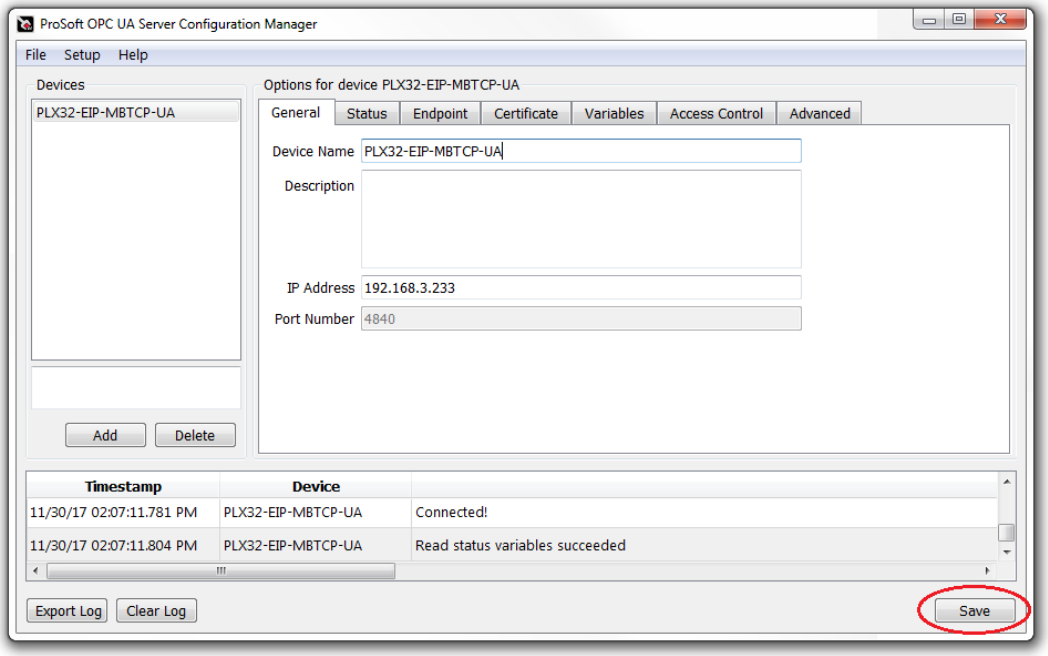
Name	Value	Description
MaxBrowseContinuationPoints	1 to 100	Maximum number of continuation points supported by the browse service.
MaxArrayLength	1 to 1000	Maximum array length supported by the server.
MaxStringLength	10 to 4096	Maximum UTF8 string length supported by the server.
Enforce	True / False	Set to TRUE to enforce limits. Enforcing limits protect against out of memory conditions but may affect interoperability with clients that do not observe these values.
MaxNodesPerRead	1 to 500	Maximum nodes to read in a Read service call.
MaxNodesPerWrite	1 to 500	Maximum number of nodes per Write service call.
MaxNodesPerMethodCall	1 to 10	Maximum number of nodes per Call service.
MaxNodesPerBrowse	1 to 100	Maximum number of nodes per Browse service call.
MaxNodesPerTranslateBrowse PathsToNodeIds	1 to 500	Maximum nodes per translate browse paths to node ID's service call.
MaxNodesPerNodeManagement	1 to 500	Maximum number of nodes per node management service calls.

MaxMonitoredItemsPerCall	1 to 500	Maximum number of monitored items per call for monitored items services.
ConnectionTimeoutInMs	1000 to 600000	The server closes the TCP Binary connection and corresponding TCP socket if this duration elapses with no messages from the client.
MaxChunkCount	1 to 10	Maximum number of message chunks allowed while making up a message. This number multiplied by the message chunk size must be > = the maximum message size.
MaxConnections	1 to 50	Maximum number of TCP Binary connections that the server supports. This value must equal at least to the number of secure channels required plus 1.
MaxMessageChunkSize	4096 to 262144	The maximum message chunk size is most significant when using security as the overall message is chunked into pieces which are signed and optionally encrypted.
MaxMessageSize	4096 to 262144	Maximum message size in bytes that the server accepts from a client. This must be at least as big as a single message chunk.
IpAddress	xxx.xxx.xxx.xxx	IP Address on which OPC UA Server is listening for client connections. Empty value as well as "0.0.0.0" means listining to all addresses. Use placeholder [IpAddress1] to bind to the first interface (eth0 for example), and [IpAddress2] to bind to the last interface (eth1).
MaxChannels	2 to 50	Maximum number of secure channels that the server supports. This value must equal at least the number of concurrent client connections required plus 1 extra channel for the integrated discovery endpoint.
MaxRevisedChannelLifetimeInMs	1000 to 86400000	Maximum acceptable value for channel lifetime requested by a client when the client connects to the server. If the client requests a lifetime greater than this value, then the server restricts the negotiated lifetime to this value.
MinRevisedChannelLifetimeInMs	100 to 60000	Minimum acceptable value for channel lifetime requested by a client when the client connects to the server. If the client requests a lifetime less than this value, then the server restricts the negotiated lifetime to this value.
CyclicRateInMs	5 to 60000	All samplings that the server performs can only be a multiple of the server cyclic rate.
MinRevisedSessionTimeoutInMs	100 to 3600000	Minimum session timeout.
MaxRevisedSessionTimeoutInMs	10000 to 86400000	Maximum session timeout.
MaxSessions	2 to 50	Maximum number of sessions that the server supports. This value must equal at least to the number of concurrent client connections required.
MaxSubscriptionsPerSession	1 to 50	Maximum number of sessions that the server supports. This value must equal at least to the number of concurrent client connections required.
MinMaxKeepAliveCount	1 to 1000	Minimum for maximum keep alive count.
MaxMaxKeepAliveCount	10 to 10000	Maximum for maximum keep alive count.

MaxNotificationRetransmission QueueSize	1 to 100	Maximum retransmission queue size.
MaxPublishRequestsPerSession	1 to 10	Maximum publish requests per session.
MaxBrowseRefsPerNode	1 to 10000	When the server responds to a browse request, it can limit the number of references included in the response and prompt the client to make another service call to get further references. This allows the server to limit the size of browse response messages and slows down the browse process as more messages are required.
MaxMonitoredItems	1 to 100000	Maximum number of monitored items that the server can support across all sessions.
MaxMonItemQueueSize	1 to 100	Maximum sample queue length for monitored items. Where a client is only interested in the latest live data value, the sample queue length can be set to 1. Where a client may be interested in logging all values and/or sampling monitored items at a slower rate than the containing subscription reports results, this value may be set to greater than 1.
PercentDeadbandDisable	True / False	To enable/disable monitored item percent dead band filtering for all variables.
PercentDeadbandFloatingPoint Disable	True / False	To enable/disable monitored item percent dead band filtering for all floating-point variables.
PercentDeadbandDoubleDisable	True / False	To enable/disable monitored item percent dead band filtering for all double precision floating point variables.
PercentDeadbandDoubleVectors Disable	True / False	To enable/disable monitored item percent dead band filtering for all double precision floating point vector variables.
UseSinglePrecisionFloatForAllFilter Operations	True / False	To enable/disable single precision float for all filter operation.
MinPublishingIntervalInMs	1 to 60000	Minimum acceptable value for subscription publishing intervals.
MinSamplingIntervalInMs	10 to 60000	Minimum acceptable value for sampling intervals.
SendApplicationInstanceCertificate ForSecurityPolicyNone	True / False	Set to true to send the application instance certificate in the GetEndpoints service call result for endpoints with security policy None.
SuppressCertificateExpired	True / False	Ignore expiration date of the certificate at certificate validation.
SuppressUriMismatch	True / False	Ignore mismatch between URI of the client application, defined in its certificate, and Application URI defined in its Application Description.
SuppressRevocationStatus Unknown	True / False	Ignore CRLs at certificate validation.
CheckRevocationStatusOffline	True / False	Check CRLs located in file system at certificate valdiation.
PortNumber	0 to 65000	TCP port number of server's endpoint.

7.10 Saving the UA Server Configuration

1 In the main window of the PSW-UACM, click the **SAVE** button.



2 The UA Server is now ready for communications with a UA Client.

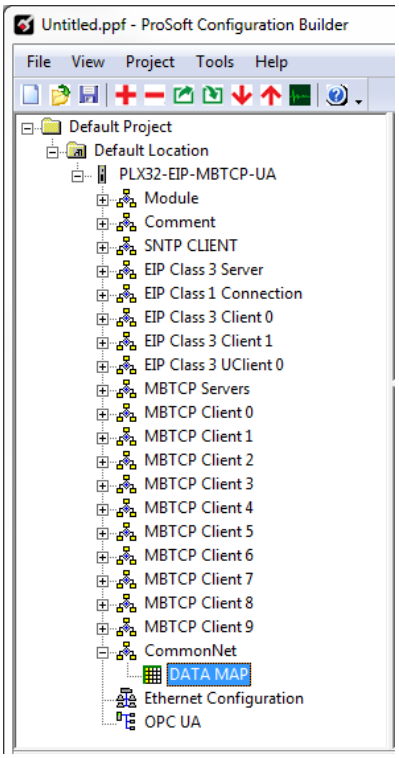
7.11 UA Client Connectivity

The following example shows how the UA Client can read data from the gateway's UA Server.

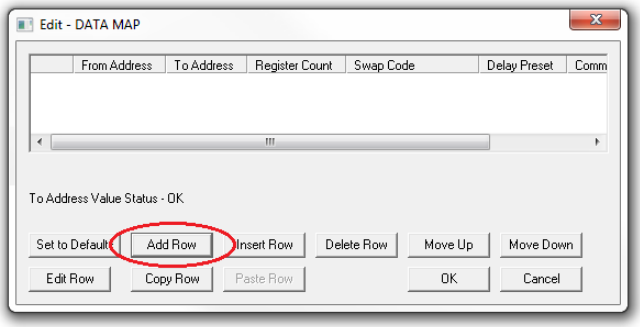
7.11.1 Data Map Example

In this example, the live data will be mapped into the gateway's user data range (0 to 9999). The mapped data includes the *Revision Number* and *Program Cycle Counter* of the PLX32-EIP-MBTCP-UA. The *DATA MAP* feature in PCB is used to accomplish this.

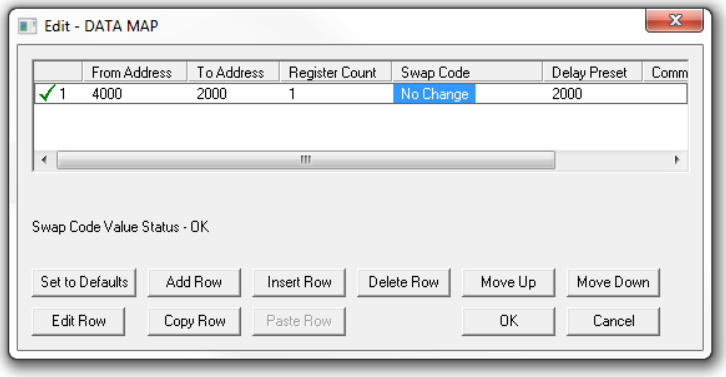
1 In PCB, double-click on the **DATA MAP** icon.



2 The *Edit – DATA MAP* dialog opens. Click on the **ADD ROW** button.

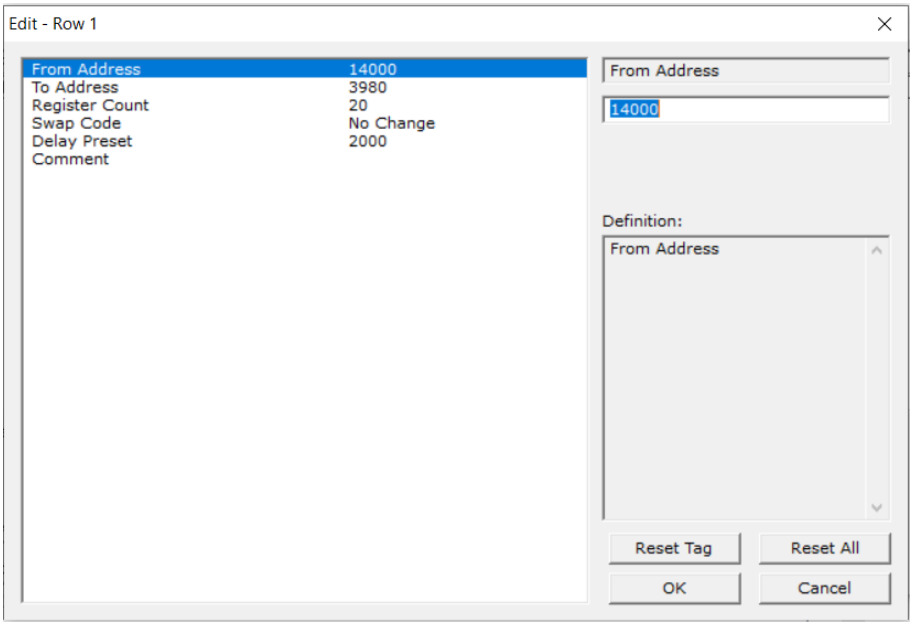


3 The first row populates with a default data mapping. Double-click on this row to edit it.



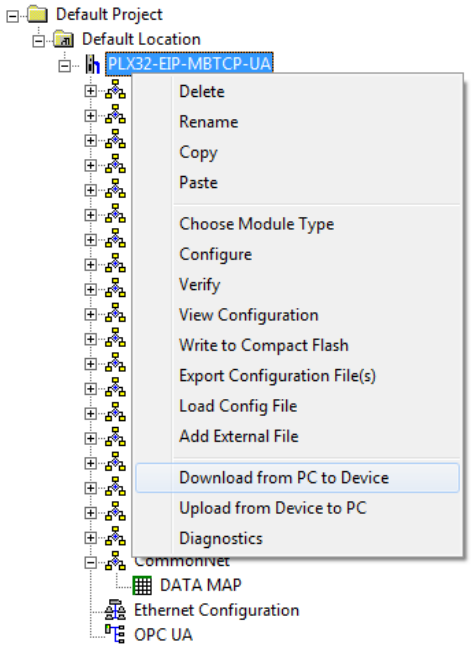
4 In the *Edit – Row 1* dialog, edit the following parameters:
From Address parameter to: **14000**
To Address parameter to: **3980**
Register Count parameter to: **20**

5 The PLX32-EIP-MBTCP-UA status data is located at gateway memory 14000 to 14019. The *DATA MAP* feature copies these 20 registers down to 3980 to 3999. This is where the UA Client will read from.

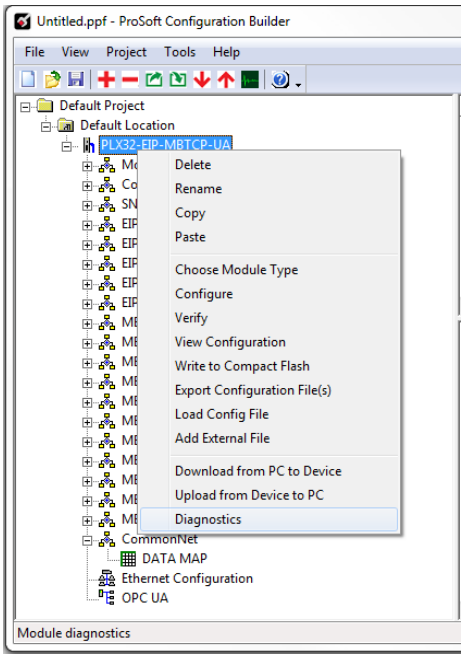


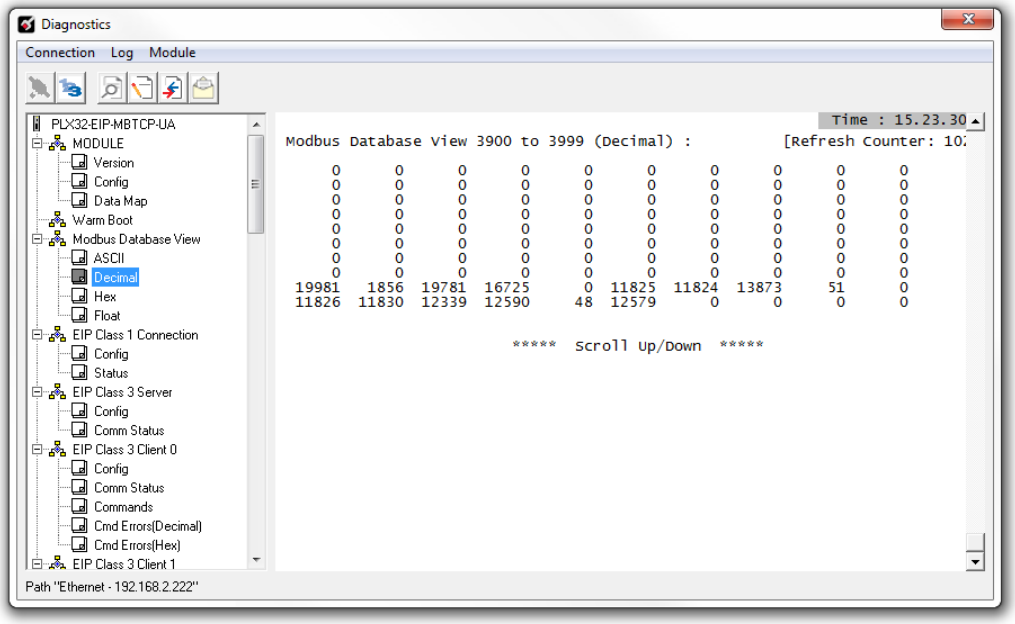
6 Click the **OK** button.

7 When ready, download the PCB configuration to the gateway.

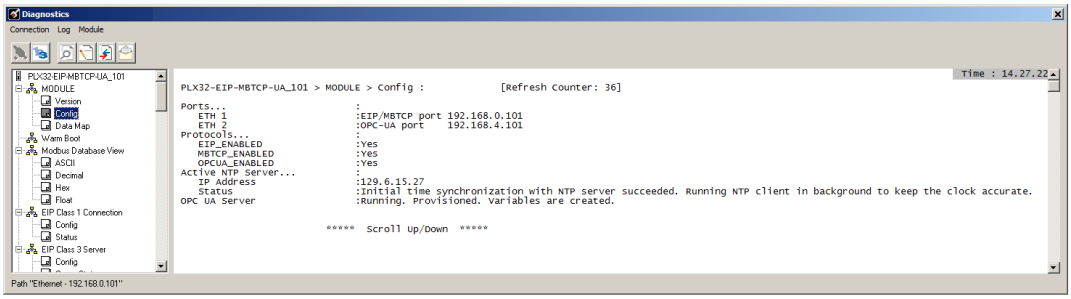


8 Upon (automatic) reboot after PCB download, the mapped data can be viewed in the *Database View* of the PCB *Diagnostics* window.





9 In **MODULE** > **Config**, verify the SNTP client is being updated properly in preparation to connect the UA Server to the UA Client.



10 The PLX32-EIP-MBTCP-UA is ready to communicate with the UA Client.

7.11.2 UA Client Setup

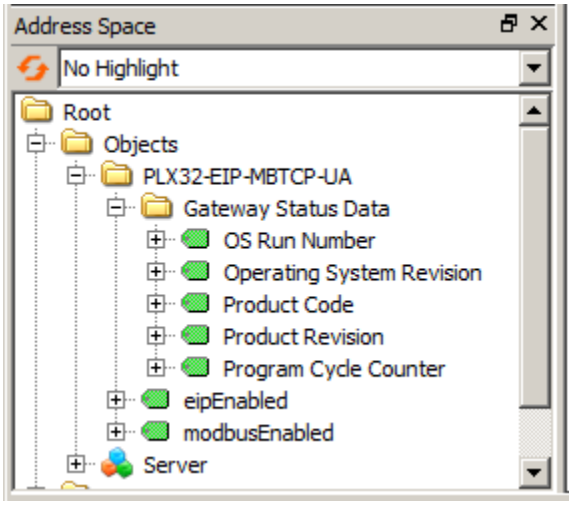
Since UA Client configurations differ from one-another, this section covers the general setup of the UA Client to communicate to the UA Server.

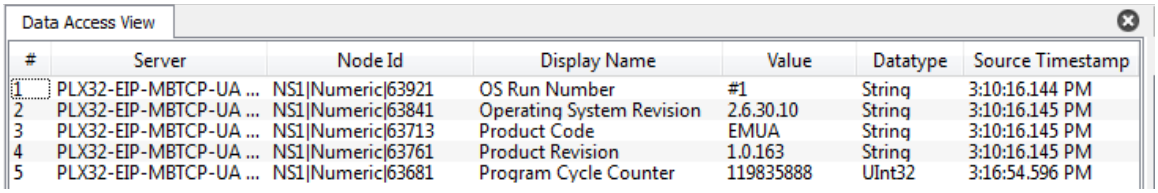
When entering the URL of the UA Server, use the following: **opc.tcp://xxx.xxx.xxx.xxx** format where *xxx.xxx.xxx.xxx* is the IP address of the UA Server.

It is recommended to use the most secure endpoint with **Sign & Encrypt** security mode and **Basic256Sha256** security policy (enabled by default).

Once the connection has been made, locate the *Gateway Status Data* variables. These variables include the following PLX32-EIP-MBTCP-UA status parameters:

- OS Run Number
- Operating System Revision
- Product Code
- Product Revision
- Program Cycle Counter





#	Server	Node Id	Display Name	Value	Datatype	Source Timestamp
1	PLX32-EIP-MBTCP-UA ...	NS1|Numeric|63921	OS Run Number	#1	String	3:10:16.144 PM
2	PLX32-EIP-MBTCP-UA ...	NS1|Numeric|63841	Operating System Revision	2.6.30.10	String	3:10:16.145 PM
3	PLX32-EIP-MBTCP-UA ...	NS1|Numeric|63713	Product Code	EMUA	String	3:10:16.145 PM
4	PLX32-EIP-MBTCP-UA ...	NS1|Numeric|63761	Product Revision	1.0.163	String	3:10:16.145 PM
5	PLX32-EIP-MBTCP-UA ...	NS1|Numeric|63681	Program Cycle Counter	119835888	UInt32	3:16:54.596 PM

7.12 Troubleshooting and Maintenance of OPC UA Server

7.12.1 Status Tab

The *State* field indicates the expected state of the device according to device settings stored in the PSW-UACM database. It might not match with the actual state of the device.

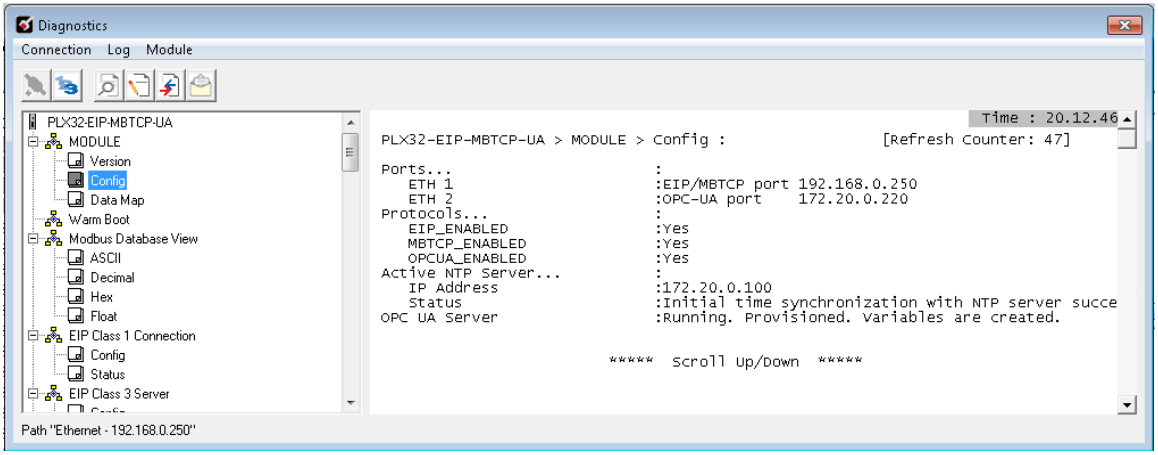
The *Variables Load Result* field reflects the actual state of the device. A value of **Success** means that variables are loaded successfully, which is only possible when a device is provisioned.

7.12.2 Communication Errors Log

If communication errors exist, error messages may appear in the logging table at the bottom of the PSW-UACM window. These messages can be exported to a text file for troubleshooting.

7.12.3 PCB Gateway Diagnostics

The *Config* option in the PCB *Diagnostics* window provides information about configured IP addresses, NTP server configuration, NTP server communications status, and current status of the OPC UA Server.



7.12.4 Reset of State Back to "Waiting to be provisioned"

If the PLX32-EIP-MBTCP-UA becomes locked (lost password, expired PSW-UACM certificate, expired CRL, etc.), the device's OPC UA Server can be reset to the default state.

To reset the OPC UA Server to the default state (includes all OPC UA Server configuration files and trust lists), short both jumpers 1 and 3, then reboot the PLX32-EIP-MBTCP-UA.

To match the actual state of the device, the state of the device stored in the PSW-UACM database needs to be set to "not provisioned" as well. This is done by clicking the **RESET STATE** button located in the *Status* tab.

7.12.5 Backup of PSW-UACM Configuration Database

The PSW-UACM configuration database is in the *C:\ProgramData\Prosoft Technology\OPC UA Server Configuration Manager* folder. To create a backup copy, close PSW-UACM, and copy the folder content into a backup destination.

7.12.6 Moving the PSW-UACM Installation to a Different Machine

A single installation of the PSW-UACM must be used to configure all PLX32-EIP-MBTCP-UA devices. This setup will not work if another copy of the PSW-UACM is installed on another PC. If PSW-UACM must be migrated to another PC, then:

- Create a backup of the PSW-UACM database as described in section *7.12.5 Backup of PSW-UACM Configuration Database*
- Install PSW-UACM on another PC
- Restore backup files on the PC into the *C:\ProgramData\Prosoft Technology\OPC UA Server Configuration Manager* folder (PSW-UACM must be closed during restore).

8 Support, Service, and Warranty

8.1 Contacting Technical Support

ProSoft Technology, Inc. is committed to providing the most efficient and effective support possible. Before calling, please gather the following information to assist in expediting this process:

1. Product Version Number
2. System architecture
3. Network details

If the issue is hardware related, we will also need information regarding:

1. Module configuration and associated ladder files, if any
2. Module operation and any unusual behavior
3. Configuration/Debug status information
4. LED patterns
5. Details about the interfaced serial, Ethernet or Fieldbus devices

North America (Corporate Location)	**Europe / Middle East / Africa Regional Office**
Phone: +1 661-716-5100 ps.prosofttechnology@belden.com Languages spoken: English, Spanish REGIONAL TECH SUPPORT ps.support@belden.com	Phone: +33.(0)5.34.36.87.20 ps.europe@belden.com Languages spoken: English, French, Hindi, Italian REGIONAL TECH SUPPORT ps.support.emea@belden.com
Latin America Regional Office	**Asia Pacific Regional Office**
Phone: +52.222.264.1814 ps.latinam@belden.com Languages spoken: English, Spanish, Portuguese REGIONAL TECH SUPPORT ps.support.la@belden.com	Phone: +60.3.2247.1898 ps.asiapc@belden.com Languages spoken: Bahasa, Chinese, English, Hindi, Japanese, Korean, Malay REGIONAL TECH SUPPORT ps.support.ap@belden.com

For additional ProSoft Technology contacts in your area, please see:
www.prosoft-technology.com/About-Us/Contact-Us

8.2 Warranty Information

For details regarding ProSoft Technology's legal terms and conditions, please see:
www.prosoft-technology.com/ProSoft-Technology-Legal-Terms-and-Conditions

For Return Material Authorization information, please see:
www.prosoft-technology.com/RMA