

## Accessing Data on a Rockwell Automation Logix PLC With the ETH-1000 Ethernet Gateway

The ICC ETH-1000 Multiprotocol Ethernet Gateway has the ability to act as a client to remote EtherNet/IP-enabled devices, which include not only I/O or low-level slaves, but PLCs as well. There are two types of data transfer mechanisms that exist within the EtherNet/IP protocol: TCP/IP-based "explicit" messaging, and UDP/IP-based "implicit" (or "I/O") messaging.

EtherNet/IP explicit messaging is a point-to-point, request/response (client/server) protocol typically used for unscheduled (on-demand) "information" messaging. On the other hand, implicit messaging is a multicast, producer/consumer protocol designed for scheduled "control" data transfer. In ControlLogix parlance, I/O messaging is the transfer of I/O data with an I/O device in the I/O configuration tree, or the transfer of Tag data via a "Produced Tag" connection.

Legacy Rockwell Automation PLCs, such as the PLC5E and SLC5/05, only support a subset of explicit messaging: data on these units can be accessed from the ETH-1000 by configuring an EtherNet/IP client service object with a "Typed Read/Write" command. Logix PLCs (such as the ControlLogix), however, support both explicit and implicit mechanisms.

Retrieving data from a Logix PLC can be achieved by several methods. This document will explain three methods for accessing data on a Logix PLC with provided examples. The first and simplest method directly accesses the "Controller Tags". The second method maps "Controller Tags" to File Numbers, and uses explicit messaging data transfers. The third method requires the use of a Produced Tag connection on the PLC, and uses implicit messaging data transfers. For an ETH-1000 with a coprocessor firmware V2.320 and higher, use Method 1. For older firmware versions, it is strongly recommended to upgrade to the latest firmware version to use Method 1. Otherwise, use Method 2 and Method 3.

In places, this document shows some steps to take using Rockwell Automation software products (such as RSLogix 5000). This is not intended to be a complete tutorial on using these products. It is assumed that the reader has familiarity with the basic use of the controllers and software to program them. The information presented here applies to all members of the ControlLogix family: CompactLogix, DriveLogix, SoftLogix and FlexLogix, etc.

**ETH-1000 with coprocessor firmware V2.320 and higher should use Method 1.**

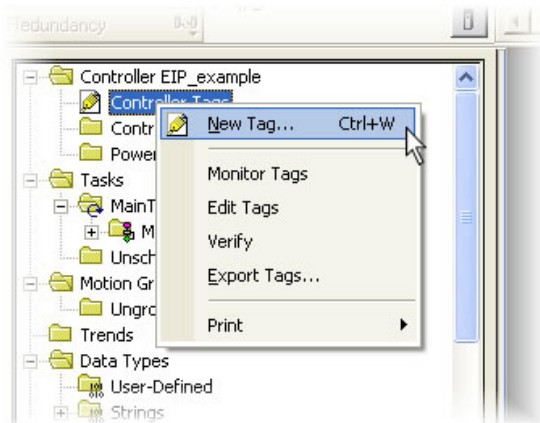
### **Method 1: Direct Tag Access**

The ETH-1000 service object can be configured with a Data Table Read/Write explicit messaging command to directly target "Controller Tags" with a data type of SINT, INT, and DINT. This method provides full read/write capability from the ETH-1000 to the PLC memory.

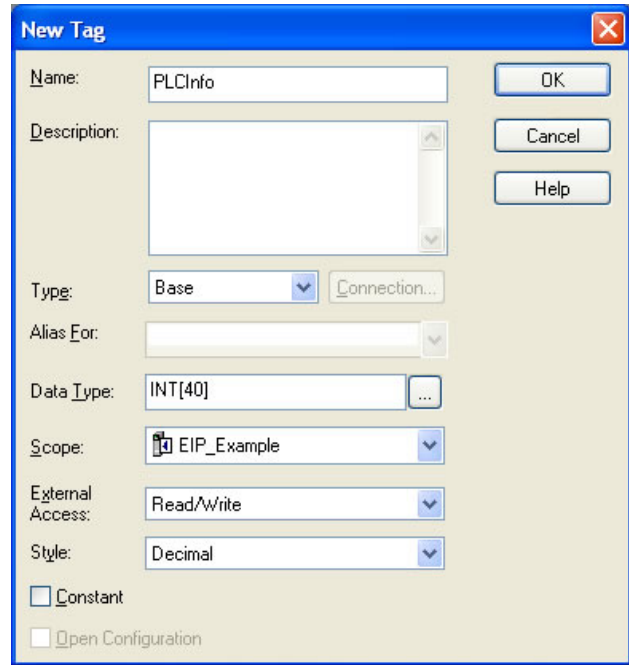
In the following example, we will create a tag called "PLCInfo", which is structured as an array of 40 elements of data type INT. Note that in practice, this tag could be a standard tag or an alias tag. We will then access the tag via a Data Table Read/Write explicit messaging command from the ETH-1000.

### **Example**

1. In RSLogix 5000, create a new tag by right-clicking on the "Controller Tags" menu in the project tree and selecting "New Tag..." (refer to Figure 1).
2. Define an array named "PLCInfo", which is comprised of 40 elements of data type INT (refer to Figure 2). Click "OK".

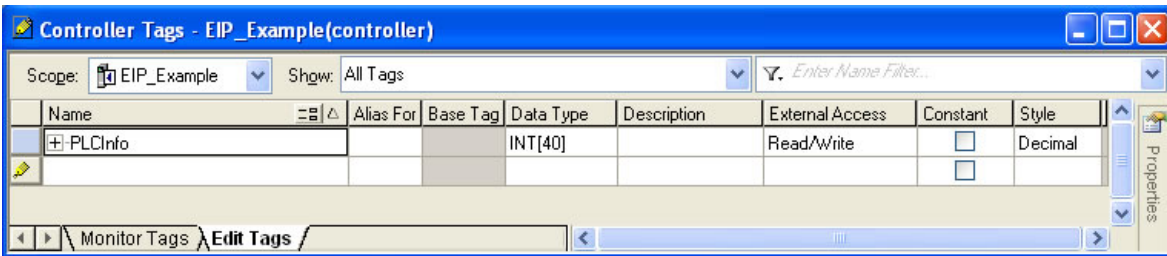


**Figure 1: New Tag Menu**



**Figure 2: Create PLCInfo Array Tag**

- The PLCInfo tag should now be visible in the “Controller Tags” window (refer to Figure 3).



**Figure 3: Viewing the PLCInfo Tag**

- The PLCInfo tag can now be accessed by configuring a service object on the ETH-1000 with a “Data Table Read/Write” message type targeting the PLCInfo tag. Refer to Figure 4 for a screenshot of an ETH-1000’s web server, showing the configuration of a connection object targeting the ControlLogix PLC (residing at IP address 192.168.16.199), and a service object (called “PLCInfo Access”) which is configured to access the PLCInfo tag. The Slot field must be set to 0, to correspond with the location of the controller in the Logix rack. The Element Type is set to 16-bit signed to correspond with the PLCInfo data type of INT. The Element Type must match the size of the tag’s data type. If the tag’s data type is SINT, then the Element Type must be set to 8-bit signed/unsigned. Likewise, if the tag’s data type is DINT, the Element Type must be set to 32-bit signed/unsigned. Refer to the *ETH-1000 User’s Manual* for more detailed information regarding configuration of EtherNet/IP connection objects and service objects.

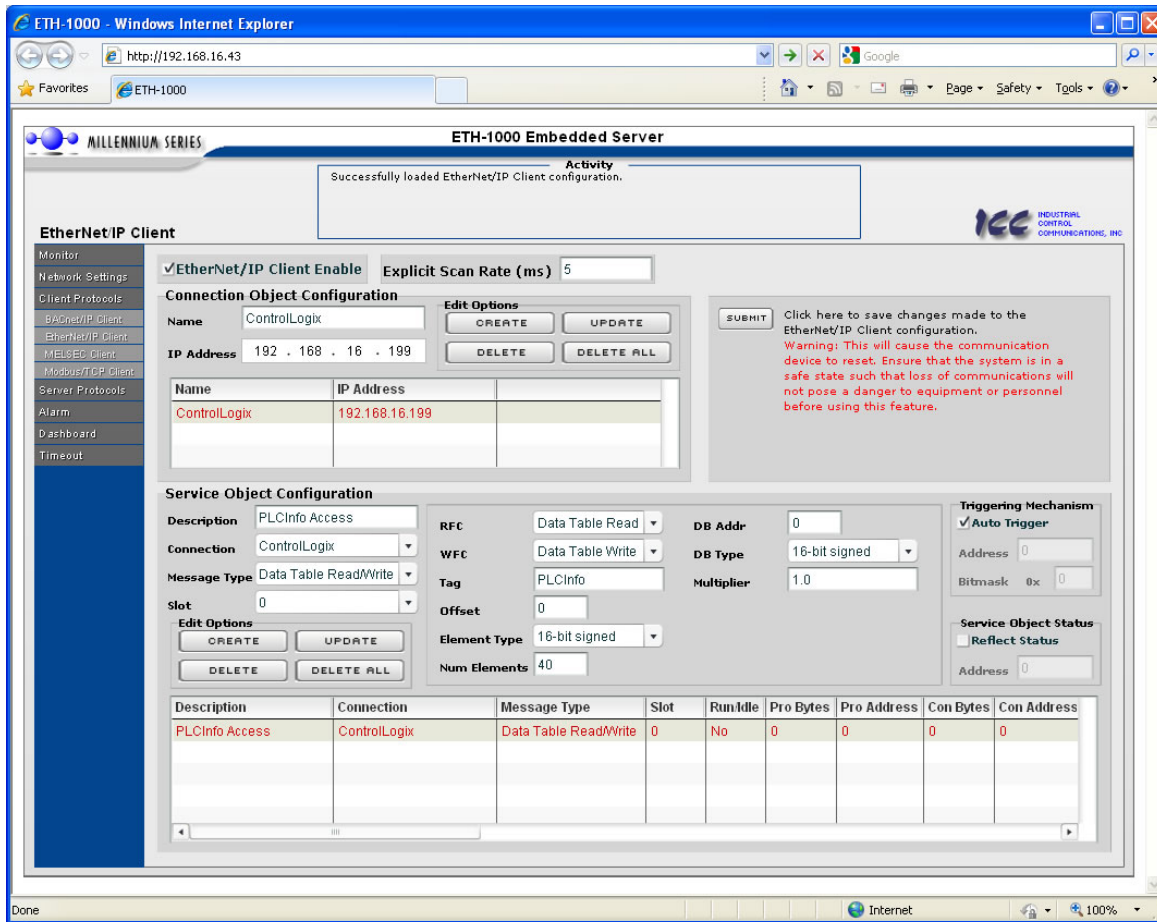


Figure 4: ETH-1000 Connection Object and PLCInfo Access Service Object Configuration

**ETH-1000 with coprocessor firmware V2.300 and older can use Method 2 and Method 3. To use Method 1, upgrade to the latest firmware version.**

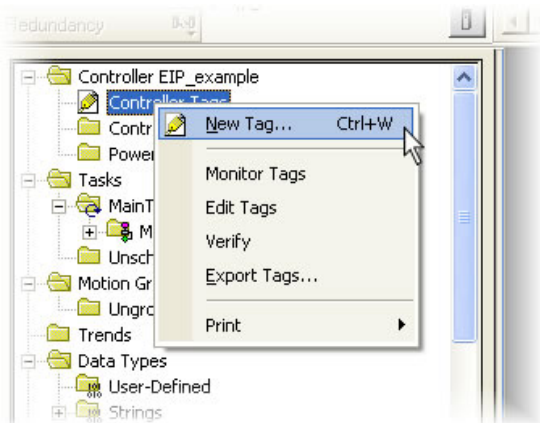
## **Method 2: PLC/SLC Mapping**

The PLC/SLC mapping method is achieved by mapping a Controller Tag to a File Number. Once this is completed, an ETH-1000 service object can be configured with a Typed Read/Write explicit messaging command to target the File Number, and hence access the associated Controller Tag. This method provides full read/write capability from the ETH-1000 to the PLC memory.

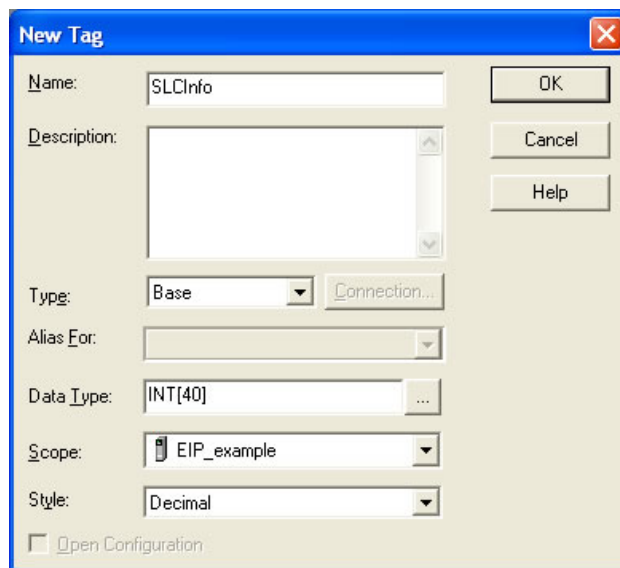
In the following example, we will create a tag called “SLCInfo”, which is structured as an array of 40 elements of data type INT. Note that in practice, this tag could be a standard tag or an alias tag. We will then expose the SLCInfo tag to the EtherNet/IP network by mapping it to File Number 10, which will provide PLC/SLC-style direct access to the underlying tag via a Typed Read/Write explicit messaging command from the ETH-1000.

## **Example**

1. In RSLogix 5000, create a new tag by right-clicking on the “Controller Tags” menu in the project tree and selecting “New Tag...” (refer to Figure 5).
2. Define an array named “SLCInfo”, which is comprised of 40 elements of data type INT (refer to Figure 6). Click “OK”.



**Figure 5: New Tag Menu**



**Figure 6: Create SLCInfo Array Tag**

- The SLCInfo tag should now be visible in the “Controller Tags” window (refer to Figure 7).

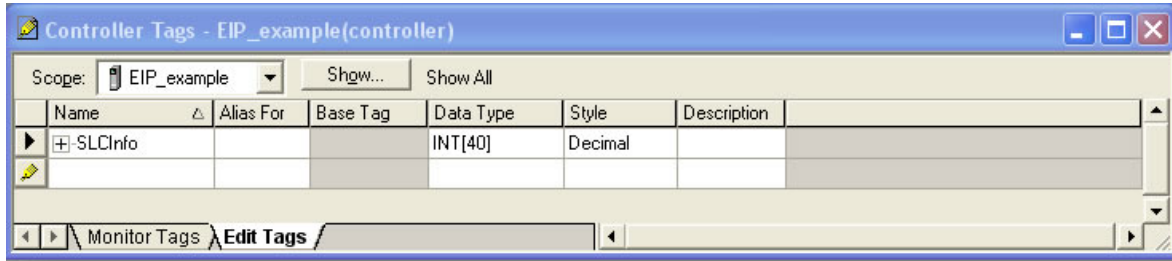


Figure 7: Viewing the SLCInfo Tag

- Navigate to *Logic...Map PLC/SLC Messages...* (refer to Figure 8). The “PLC2,3,5 / SLC Mapping” window will appear.

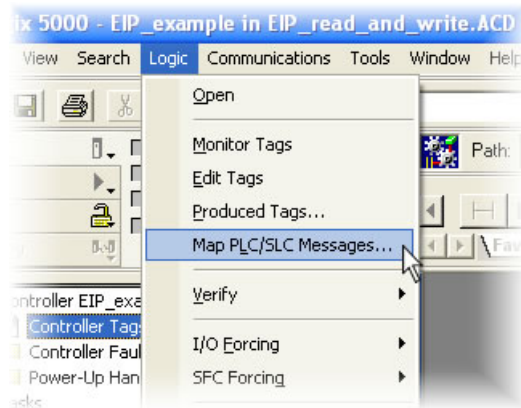


Figure 8: Map Messages Menu

5. In the first row under the "File Number" column, enter a value of 10. On the same row, click on the dropdown box in the "Tag Name" column and select the SLCInfo tag (refer to Figure 9). Click "OK".
6. The mapping is complete. The SLCInfo tag can now be accessed by configuring a service object on the ETH-1000 with a "Typed Read/Write" message type targeting File Number 10. Refer to Figure 10 for a screenshot of an ETH-1000's web server, showing the configuration of a connection object targeting the ControlLogix PLC (residing at IP address 192.168.16.199), and a service object (called "SLCInfo Access") which is configured to access the SLCInfo tag. Refer to the *ETH-1000 User's Manual* for detailed information regarding configuration of EtherNet/IP connection objects and service objects.

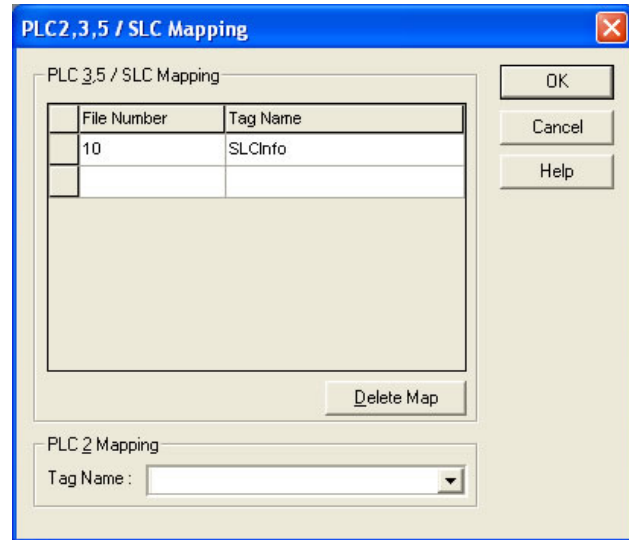


Figure 9: Mapping the SLCInfo Tag

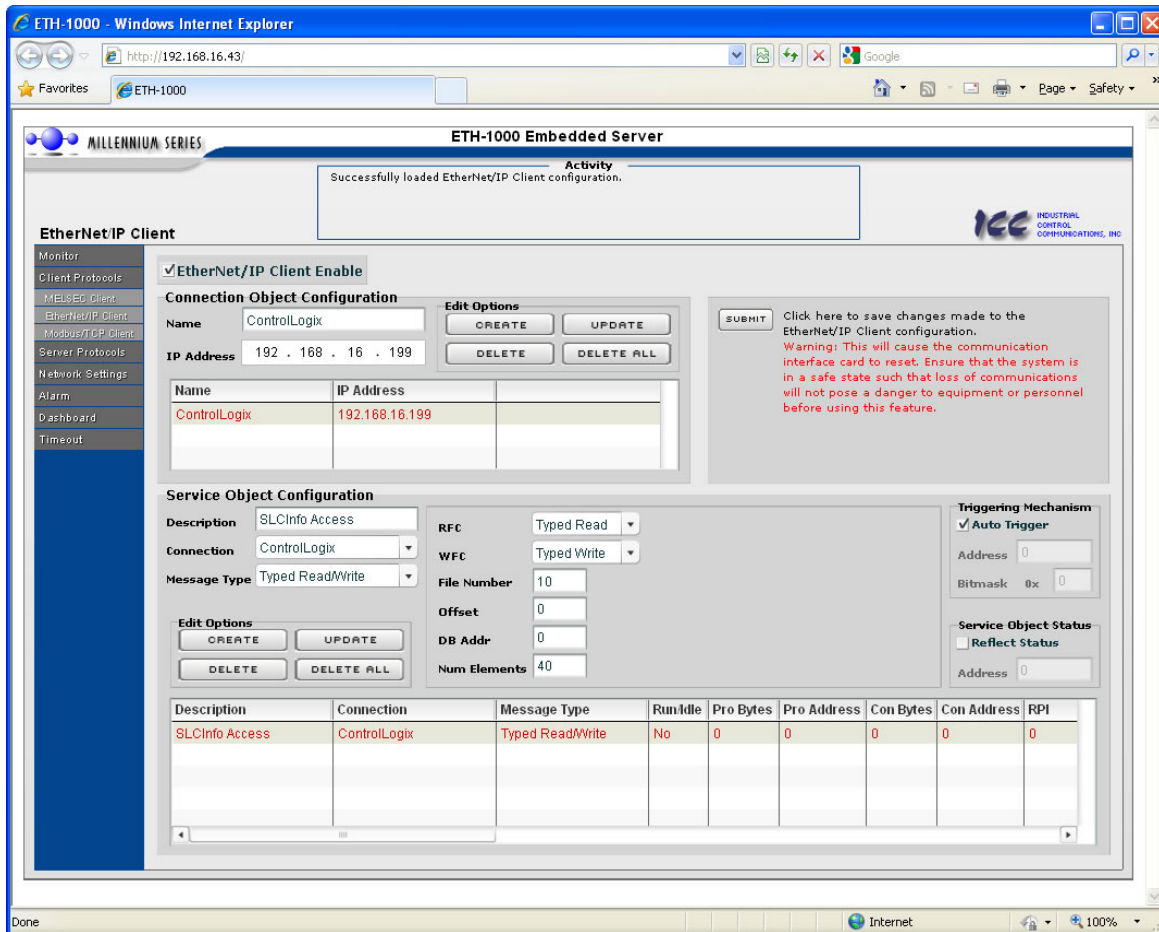


Figure 10: ETH-1000 Connection Object and SLCInfo Access Service Object Configuration

### **Method 3: Produced Tag Connection**

A Produced Tag connection allows another device (the ETH-1000, in this case) to make a direct connection to a tag in the PLC's database. Like standard implicit (I/O) data transfers, these connections are class 1 cyclic multicast transfers. The main difference is that rather than connecting to a predetermined I/O assembly, any tag in the PLC that the user designates as a Produced Tag can be accessed. The user must select the Produced Tag option when creating the tag in RSLogix 5000. This designation makes the tag known to the PLC's Connection Manager as a potential connection target. Note that it is only possible to share controller-scoped tags.

Produced Tag connections use unidirectional data flow (from the PLC to the ETH-1000), with a "heartbeat" in the return direction. The ETH-1000 does not need to be present in the PLC's I/O configuration, although this is certainly allowed if required by the application, as the ETH-1000 is capable of simultaneously acting as both an EtherNet/IP client and server.

Produced tags must have a data type that is cumulatively 4 bytes or larger. Therefore, single 8-bit (SINT) and 16-bit (INT) tags cannot be configured as produced tags. Use DINT, REAL, or user-defined data types. Any user-defined data types must be at least 4 bytes or larger. An array of DINT, REAL, or user-defined data types are also permitted. Produced tags must be less than 500 bytes in total size.

#### **--- Special Note for ETH-1000 Coprocessor Firmware V2.000 ---**

ETH-1000 coprocessor firmware V2.000 requires the user to manually add and configure the service object for a Produced Tag connection directly in the appropriate XML file, and then FTP that XML file to the gateway's filesystem. Once this customized XML file is loaded into the gateway, do not modify the EtherNet/IP client configuration via the embedded web server, or download any configuration via the USB connection, as performing either of these actions will overwrite the customized XML configuration file with a resulting loss of the Produced Tag connection configuration.

The following XML schema represents the format of the Produced Tag connection service object as it must be encoded in the gateway's EIPC\_10750A.XML file:

```
<eipclient id="" msgtype="4" rpi="" mult="0" cbytes="" caddr="" tag="" desc=""
tauto="1" taddr="0" tmask="0" rsen="0" rsaddr="0"/>
```

#### Where

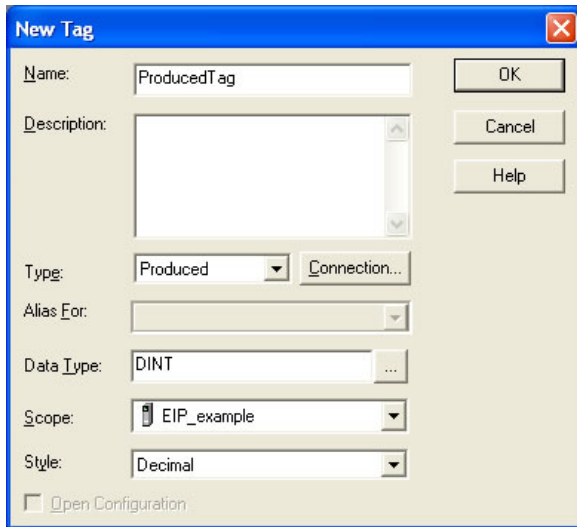
- id ..... Must match the id of the appropriate connection object.
- msgtype..... Must be set to a value of 4 (Produced Tag connection).
- rpi ..... Requested packet interval in milliseconds. Valid values range from 10 to 4294967295 and must be specified in hexadecimal (A to FFFFFFFF).
- mult ..... Timeout multiplier. A value of 0 is recommended (this equates to a 4X multiplier).
- cbytes..... Consumed bytes. Size of the PLC's designated Producing Tag in bytes. Valid values range from 4 to 500 bytes, and must be specified in hexadecimal (4 to 1F4).
- caddr ..... Consumed address. The location in the ETH-1000's internal database to store Produced Tag data. Valid values range from 0 to 4095, and must be specified in hexadecimal (0 to FFF).
- tag ..... Name of the Produced Tag in the PLC.
- desc ..... Description of service object (16 characters max). Can be left as an empty string ("").
- tauto..... Auto trigger enable. For simplicity, it is recommended to set this to a value of 1 (auto trigger enabled), but manual enabling/disabling of the service object via a designated trigger bit is possible if desired (refer to the *ETH-1000 User's Manual*).
- taddr ..... Trigger address. Set to 0 if tauto is 1.
- tmask ..... Trigger mask. Set to 0 if tauto is 1.
- rsen ..... Reflect status enable. For simplicity, set to 0 (can be enabled if desired, however).
- rsaddr ..... Reflect status address. Set to 0 if rsen is 0.

In the following example, we will create a Produced Tag in the ControlLogix PLC called “ProducedTag”, which is structured as a single DINT (32-bit or 4-byte) data type. We will then manually modify the gateway’s EIPC\_10750A.XML file to instruct the ETH-1000 to establish a Produced Tag I/O connection with the PLC (required for ETH-1000 coprocessor firmware V2.000 only).

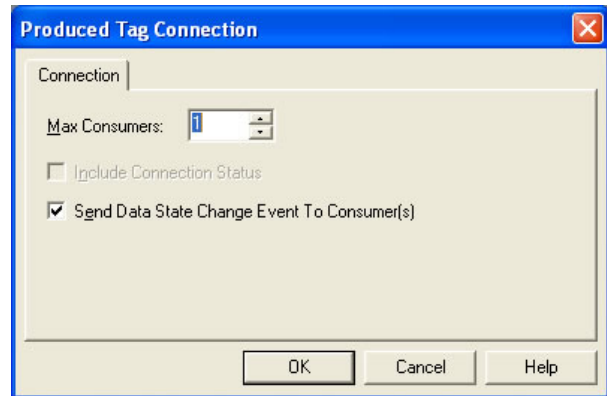
**Example**

1. In RSLogix 5000, create a new tag by right-clicking on the “Controller Tags” menu in the project tree and selecting “New Tag...” (refer to Figure 5).
2. The New Tag window will appear (refer to Figure 11). In the “Name” field, type “ProducedTag”. In the “Type” dropdown box, select Produced. For “Data Type”, select DINT. Click on the “Connection...” button next to the “Type” dropdown box: the Produced Tag Connection properties window as shown in Figure 12 will appear. Depending on the PLC program, decide whether or not to check the “Send Data State Change Event To Consumer(s)” checkbox according to the following criteria:
  - Check this box if the PLC program will use an IOT instruction to send updated tag values to the ETH-1000. Note that if this box is checked, then the ETH-1000 will only be notified of changes to the produced tag’s values when the IOT instruction executes.
  - Uncheck this box to automatically notify the ETH-1000 of changes to the produced tag’s values at the RPI rate (no IOT instruction is necessary).

Click “OK” to close the Produced Tag Connection properties window, and then click “OK” again to close the New Tag window.



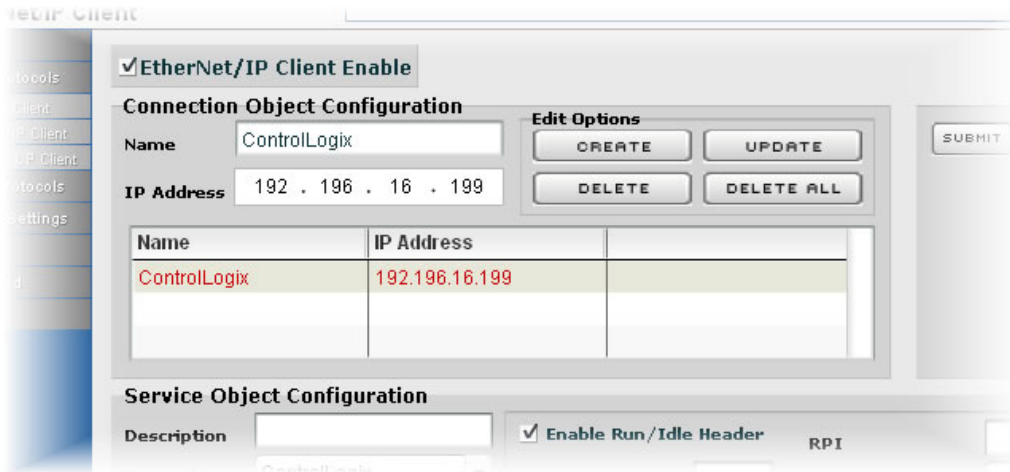
**Figure 11: Create the Produced Tag**



**Figure 12: Produced Tag Connection Properties**

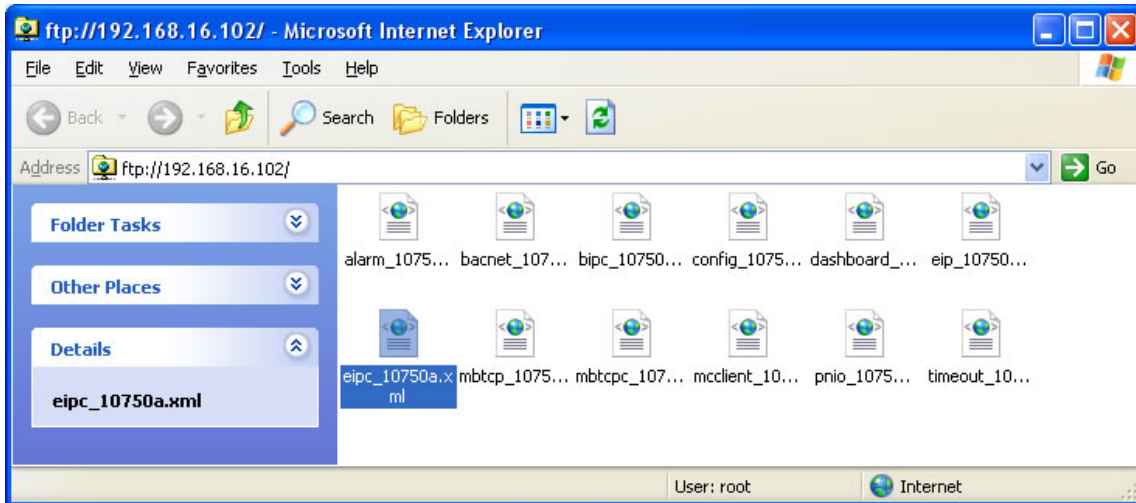
3. The tag is now ready to be accessed by the ETH-1000. If your device has coprocessor firmware V2.100 or later, you can now define the connection object(s) and produced tag service object(s) via the *Gateway Configuration Utility* or the web server. If your device has coprocessor firmware V2.000, continue on to the next step.
4. Open the ETH-1000’s embedded web server and navigate to the EtherNet/IP Client configuration in the menu tree.

5. Enable the EtherNet/IP Client driver and create a connection object named "ControlLogix" with the IP address "192.168.16.199" (the IP address of the targeted PLC in this example). Refer to Figure 13. Submit the configuration by clicking the "submit" button. The ETH-1000 will automatically reboot.



**Figure 13: Creating the Connection Object for Accessing the Produced Tag**

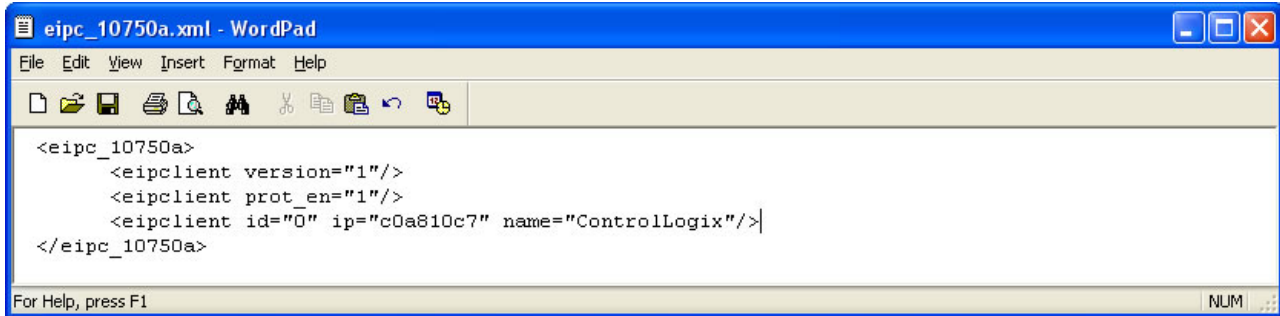
6. When the ETH-1000 has completed its boot cycle, download the configuration file "EIPC\_10750A.XML" from the gateway via FTP (refer to Figure 14).



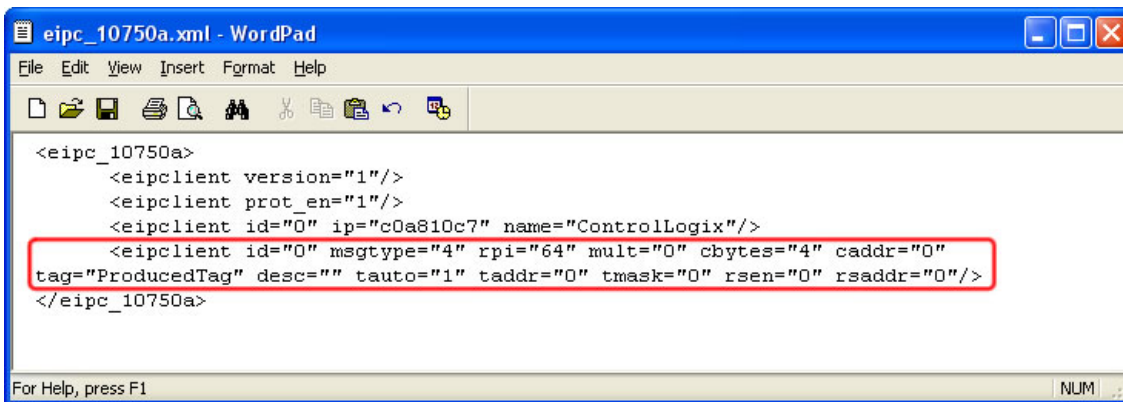
**Figure 14: Download Configuration File via FTP**

7. Edit the file with a text editor of your choice. We will use Wordpad in this example. The unmodified configuration file is shown in Figure 15.
8. Copy the following text and paste it into the file at the location indicated in Figure 16:

```
<eipclient id="0" msgtype="4" rpi="64" mult="0" cbytes="4" caddr="0"
tag="ProducedTag" desc="" tauto="1" taddr="0" tmask="0" rsen="0" rsaddr="0"/>
```



**Figure 15: XML Configuration File Prior to Modification**



**Figure 16: Modified XML Configuration File**

9. Save the XML file and then upload it back to the ETH-1000's filesystem via FTP.
10. Reboot the ETH-1000. The gateway will automatically establish the configured Produced Tag connection with the targeted PLC upon startup. The value of the PLC's tag "ProducedTag" will be sent to the ETH-1000 every 100ms, and stored in the gateway's data array at addresses 0...3.

**References**

- *Establishing I/O Communications with RA ControlLogix Systems on EtherNet/IP (Rev 1.0)* – Rockwell Automation, March 10, 2003
- *Communicating with RA Products Using EtherNet/IP Explicit Messaging (Rev. 1.2)* – Rockwell Automation, June 7, 2001
- *ETH-1000 User's Manual* – Industrial Control Communications, October 30, 2009