

Siemens TCP/IP Server Ethernet Driver

© 2023 PTC Inc. All Rights Reserved.

Table of Contents

Siemens TCP/IP Server Ethernet Driver	1
Table of Contents	2
Siemens TCP/IP Server Ethernet Driver	3
Overview	3
Setup	4
Channel Properties — General	4
Tag Counts	5
Channel Properties — Ethernet Communications	5
Channel Properties — Write Optimizations	5
Channel Properties — Advanced	6
Channel Properties — Communications Properties	7
Device Properties — General	7
Operating Mode	8
Tag Counts	9
Device Properties — Scan Mode	9
Device Properties — CPU Settings	10
Siemens Client Device Configuration	10
Internal Tags	10
Data Types Description	12
Address Descriptions	13
Event Log Messages	18
Failure to start unsolicited communications. Port number = <number>.	18
Appendix: Configuring Connections Using the SIMATIC Manager	19
Step One: Creating a New Project	19
Step Two: Configuring the Siemens Client and PC Station	22
Step Three: Connecting the Siemens Client and the Siemens Server Driver	35
Step Four: Inserting Function Blocks	41
Step Five: Creating the DB3 Data Block	47
Step Six: Inserting PUT FB	48
Step Seven: Downloading to the PLC	53
Index	58

Siemens TCP/IP Server Ethernet Driver

Help version 1.041

CONTENTS

Overview

What is the Siemens TCP/IP Server Ethernet Driver?

Setup

How do I configure a device for use with this driver?

Data Types Description

What data types does this driver support?

Address Descriptions

How do I address a data location on a Siemens TCP/IP Ethernet device?

Event Log Messages

What messages does the driver produce?

Overview

The Siemens TCP/IP Server Ethernet Driver provides a reliable way to connect Siemens TCP/IP server devices to client applications; including HMI, SCADA, Historian, MES, ERP, and countless custom applications. This driver acts as a simulated Siemens PLC. It is intended for simulation of Siemens S7-300.

Setup

Supported Protocols

S7 Messaging on Industrial Ethernet (ISO 8073 Class 0) over TCP/IP. This is defined in RFC1006.

Supported Commands

FB14-GET (S7-300)
 FB15-PUT (S7-300)
 SFB14-GET (S7-400)
 SFB15-PUT (S7-400)

Channel and Device Limits

The maximum number of channels supported by this driver is 1. The maximum number of virtual devices supported by this driver is 256 per channel.

Libraries

This driver requires a standard Ethernet card. No special libraries or hardware are needed.

● **Note:** To communicate with this driver, devices require specialized ladder programming.

[Channel Properties](#)

[Device Properties](#)

[Siemens Client Device Configuration](#)

[Appendix: Configuring Connections Using the SMATIC Manager](#)

Channel Properties — General

This server supports the use of multiple simultaneous communications drivers. Each protocol or driver used in a server project is called a channel. A server project may consist of many channels with the same communications driver or with unique communications drivers. A channel acts as the basic building block of an OPC link. This group is used to specify general channel properties, such as the identification attributes and operating mode.

Property Groups General Write Optimizations Advanced	<table border="1"> <tr> <td colspan="2">[-] Identification</td> </tr> <tr> <td>Name</td> <td></td> </tr> <tr> <td>Description</td> <td></td> </tr> <tr> <td>Driver</td> <td></td> </tr> <tr> <td colspan="2">[-] Diagnostics</td> </tr> <tr> <td>Diagnostics Capture</td> <td>Disable</td> </tr> <tr> <td colspan="2">[-] Tag Counts</td> </tr> <tr> <td>Static Tags</td> <td>10</td> </tr> </table>	[-] Identification		Name		Description		Driver		[-] Diagnostics		Diagnostics Capture	Disable	[-] Tag Counts		Static Tags	10
[-] Identification																	
Name																	
Description																	
Driver																	
[-] Diagnostics																	
Diagnostics Capture	Disable																
[-] Tag Counts																	
Static Tags	10																

Identification

Name: Specify the user-defined identity of this channel. In each server project, each channel name must be unique. Although names can be up to 256 characters, some client applications have a limited display window when browsing the OPC server's tag space. The channel name is part of the OPC browser information. The property is required for creating a channel.

● For information on reserved characters, refer to "How To... Properly Name a Channel, Device, Tag, and Tag Group" in the server help.

Description: Specify user-defined information about this channel.

Many of these properties, including Description, have an associated system tag.

Driver: Specify the protocol / driver for this channel. Specify the device driver that was selected during channel creation. It is a disabled setting in the channel properties. The property is required for creating a channel.

Note: With the server's online full-time operation, these properties can be changed at any time. This includes changing the channel name to prevent clients from registering data with the server. If a client has already acquired an item from the server before the channel name is changed, the items are unaffected. If, after the channel name has been changed, the client application releases the item and attempts to re-acquire using the old channel name, the item is not accepted. Changes to the properties should not be made once a large client application has been developed. Utilize proper user role and privilege management to prevent operators from changing properties or accessing server features.

Diagnostics

Diagnostics Capture: When enabled, this option makes the channel's diagnostic information available to OPC applications allows the usage of statistics tags that provide feedback to client applications regarding the operation of the channel. Because the server's diagnostic features require a minimal amount of overhead processing, it is recommended that they be utilized when needed and disabled when not. The default is disabled.

Note: This property is not available if the driver does not support diagnostics.

For more information, refer to "Communication Diagnostics" and "Statistics Tags" in the server help.

Tag Counts

Static Tags: Provides the total number of defined static tags at this level (device or channel). This information can be helpful in troubleshooting and load balancing.

Channel Properties — Ethernet Communications

Ethernet Communication can be used to communicate with devices.

Property Groups	Ethernet Settings	
General	Network Adapter	Default
Ethernet Communications		
Write Optimizations		
Advanced		

Ethernet Settings

Network Adapter: Specify the network adapter to bind. When left blank or Default is selected, the operating system selects the default adapter.

Channel Properties — Write Optimizations

The server must ensure that the data written from the client application gets to the device on time. Given this goal, the server provides optimization properties to meet specific needs or improve application responsiveness.

Property Groups	<input type="checkbox"/> Write Optimizations	
General	Optimization Method	Write Only Latest Value for All Tags
Write Optimizations	Duty Cycle	10

Write Optimizations

Optimization Method: Controls how write data is passed to the underlying communications driver. The options are:

- **Write All Values for All Tags:** This option forces the server to attempt to write every value to the controller. In this mode, the server continues to gather write requests and add them to the server's internal write queue. The server processes the write queue and attempts to empty it by writing data to the device as quickly as possible. This mode ensures that everything written from the client applications is sent to the target device. This mode should be selected if the write operation order or the write item's content must uniquely be seen at the target device.
- **Write Only Latest Value for Non-Boolean Tags:** Many consecutive writes to the same value can accumulate in the write queue due to the time required to actually send the data to the device. If the server updates a write value that has already been placed in the write queue, far fewer writes are needed to reach the same final output value. In this way, no extra writes accumulate in the server's queue. When the user stops moving the slide switch, the value in the device is at the correct value at virtually the same time. As the mode states, any value that is not a Boolean value is updated in the server's internal write queue and sent to the device at the next possible opportunity. This can greatly improve the application performance.
 - **Note:** This option does not attempt to optimize writes to Boolean values. It allows users to optimize the operation of HMI data without causing problems with Boolean operations, such as a momentary push button.
- **Write Only Latest Value for All Tags:** This option takes the theory behind the second optimization mode and applies it to all tags. It is especially useful if the application only needs to send the latest value to the device. This mode optimizes all writes by updating the tags currently in the write queue before they are sent. This is the default mode.

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

● **Note:** It is recommended that the application be characterized for compatibility with the write optimization enhancements before being used in a production environment.

Channel Properties — Advanced

This group is used to specify advanced channel properties. Not all drivers support all properties; so the Advanced group does not appear for those devices.

Property Groups	<input type="checkbox"/> Non-Normalized Float Handling	
General	Floating-Point Values	Replace with Zero
Write Optimizations	<input type="checkbox"/> Inter-Device Delay	
Advanced	Inter-Device Delay (ms)	0

Non-Normalized Float Handling: A non-normalized value is defined as Infinity, Not-a-Number (NaN), or as a Denormalized Number. The default is Replace with Zero. Drivers that have native float handling may default to Unmodified. Non-normalized float handling allows users to specify how a driver handles non-normalized IEEE-754 floating point data. Descriptions of the options are as follows:

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

● **Note:** This property is disabled if the driver does not support floating-point values or if it only supports the option that is displayed. According to the channel's float normalization setting, only real-time driver tags (such as values and arrays) are subject to float normalization. For example, EFM data is not affected by this setting.

● For more information on the floating-point values, refer to "How To ... Work with Non-Normalized Floating-Point Values" in the server help.

Inter-Device Delay: Specify the amount of time the communications channel waits to send new requests to the next device after data is received from the current device on the same channel. Zero (0) disables the delay.

● **Note:** This property is not available for all drivers, models, and dependent settings.

Channel Properties — Communications Properties

Property Groups	☐ Communications Parameters	
General	Port Number	502
Communications Parameters		
Block Sizes		

Port Number: Specify the port number on which the driver listens. Devices must be configured to connect to this port: messages sent to all other ports are ignored by the driver. The valid range is 0 to 65535. The default setting is TCP/IP: 102 (TSAP).

● **Note:** Non-standard values may be required by routing and firewall issues.

Device Properties — General

A device represents a single target on a communications channel. If the driver supports multiple controllers, users must enter a device ID for each controller.

Property Groups	☐ Identification	
General	Name	
Scan Mode	Description	
	Channel Assignment	
	Driver	
	Model	
	ID Format	Decimal
	ID	2

Identification

Name: Specify the name of the device. It is a logical user-defined name that can be up to 256 characters long and may be used on multiple channels.

● **Note:** Although descriptive names are generally a good idea, some OPC client applications may have a limited display window when browsing the OPC server's tag space. The device name and channel name become part of the browse tree information as well. Within an OPC client, the combination of channel name and device name would appear as "ChannelName.DeviceName".

● **For more information, refer to "How To... Properly Name a Channel, Device, Tag, and Tag Group" in server help.**

Description: Specify the user-defined information about this device.

● Many of these properties, including Description, have an associated system tag.

Channel Assignment: Specify the user-defined name of the channel to which this device currently belongs.

Driver: Selected protocol driver for this device.

Model: Specify the type of device that is associated with this ID. The contents of the drop-down menu depend on the type of communications driver being used. Models that are not supported by a driver are disabled. If the communications driver supports multiple device models, the model selection can only be changed when there are no client applications connected to the device.

● **Note:** If the communication driver supports multiple models, users should try to match the model selection to the physical device. If the device is not represented in the drop-down menu, select a model that conforms closest to the target device. Some drivers support a model selection called "Open," which allows users to communicate without knowing the specific details of the target device. For more information, refer to the driver help documentation.

ID: Specify the device's driver-specific station or node. The type of ID entered depends on the communications driver being used. For many communication drivers, the ID is a numeric value. Drivers that support a Numeric ID provide users with the option to enter a numeric value whose format can be changed to suit the needs of the application or the characteristics of the selected communications driver. The format is set by the driver by default. Options include Decimal, Octal, and Hexadecimal.

● **Note:** If the driver is Ethernet-based or supports an unconventional station or node name, the device's TCP/IP address may be used as the device ID. TCP/IP addresses consist of four values that are separated by periods, with each value in the range of 0 to 255. Some device IDs are string based. There may be additional properties to configure within the ID field, depending on the driver.

Operating Mode

Property Groups	+ Identification	
General	- Operating Mode	
Scan Mode	Data Collection	Enable
	Simulated	No

Data Collection: This property controls the device's active state. Although device communications are enabled by default, this property can be used to disable a physical device. Communications are not attempted when a device is disabled. From a client standpoint, the data is marked as invalid and write operations are not accepted. This property can be changed at any time through this property or the device system tags.

Simulated: Place the device into or out of Simulation Mode. In this mode, the driver does not attempt to communicate with the physical device, but the server continues to return valid OPC data. Simulated stops physical communications with the device, but allows OPC data to be returned to the OPC client as valid data. While in Simulation Mode, the server treats all device data as reflective: whatever is written to the simulated device is read back and each OPC item is treated individually. The data is not saved if the server removes the item (such as when the server is reinitialized). The default is No.

Notes:

1. This System tag (`_Simulated`) is read only and cannot be written to for runtime protection. The System tag allows this property to be monitored from the client.
2. When a device is simulated, updates may not appear faster than one (1) second client.

Simulation Mode is for test and simulation purposes only. It should never be used in a production environment.

Tag Counts

Property Groups	<input type="checkbox"/> Identification <input type="checkbox"/> Operating Mode <input type="checkbox"/> Tag Counts	
General	Static Tags	130

Static Tags: Provides the total number of defined static tags at this level (device or channel). This information can be helpful in troubleshooting and load balancing.

Device Properties — Scan Mode

The Scan Mode specifies the subscribed-client requested scan rate for tags that require device communications. Synchronous and asynchronous device reads and writes are processed as soon as possible; unaffected by the Scan Mode properties.

Property Groups	<input type="checkbox"/> Scan Mode	
General	Scan Mode	Respect Client-Specified Scan Rate ▼
Scan Mode	Initial Updates from Cache	Disable

Scan Mode: Specify how tags in the device are scanned for updates sent to subscribing clients. Descriptions of the options are:

- **Respect Client-Specified Scan Rate:** This mode uses the scan rate requested by the client.
- **Request Data No Faster than Scan Rate:** This mode specifies the value set as the maximum scan rate. The valid range is 10 to 99999990 milliseconds. The default is 1000 milliseconds.
 - **Note:** When the server has an active client and items for the device and the scan rate value is increased, the changes take effect immediately. When the scan rate value is decreased, the changes do not take effect until all client applications have been disconnected.
- **Request All Data at Scan Rate:** This mode forces tags to be scanned at the specified rate for subscribed clients. The valid range is 10 to 99999990 milliseconds. The default is 1000 milliseconds.
- **Do Not Scan, Demand Poll Only:** This mode does not periodically poll tags that belong to the device nor perform a read to get an item's initial value once it becomes active. It is the OPC client's

responsibility to poll for updates, either by writing to the `_DemandPoll` tag or by issuing explicit device reads for individual items. *For more information, refer to "Device Demand Poll" in server help.*

- **Respect Tag-Specified Scan Rate:** This mode forces static tags to be scanned at the rate specified in their static configuration tag properties. Dynamic tags are scanned at the client-specified scan rate.

Initial Updates from Cache: When enabled, this option allows the server to provide the first updates for newly activated tag references from stored (cached) data. Cache updates can only be provided when the new item reference shares the same address, scan rate, data type, client access, and scaling properties. A device read is used for the initial update for the first client reference only. The default is disabled; any time a client activates a tag reference the server attempts to read the initial value from the device.

Device Properties — CPU Settings

Property Groups	☐ CPU Settings	
General	Rack Number	0
Scan Mode	CPU Slot	0
CPU Settings	Maximum PDU Size	960

Rack Number: This property specifies the number of the rack in which the simulated CPU of interest resides. The valid range is 0 to 7. Devices must have unique rack and slot values. The default setting is 0.

CPU Slot: This property specifies the number of the slot in which the simulated CPU of interest resides. The valid range is 0 to 31. Devices must have unique rack and slot values. The default setting is 0.

Maximum PDU Size: This property specifies the maximum size of the Protocol Data Unit which the driver supports. It can be configured to 240, 480, and 960 bytes.

● **Note:** To observe the PDU value negotiated with the device, use the `_CurrentPDUSize` internal tag (see [Internal Tags](#)).

Siemens Client Device Configuration

Siemens PLCs must be programmed to issue read and write commands to the driver and to handle returned data. *For more information, refer to the Siemens PLC programming documentation. For information on preparing the Siemens client device and the unsolicited driver for communications, refer to [Configuring Connections Using the SIMATIC Manager](#).*

Messages must be sent to the IP address of the selected Ethernet adapter of the host computer running the unsolicited driver. To do so, update the channel properties.

● *For more information concerning the port number configured for the simulated device, refer to [Communication Properties](#).*

Internal Tags

Although the following internal tags are not visible in the server configuration, they can be browsed by the OPC client. They can be found under the `<Channel Name>.<Device Name>._InternalTags` group. If the OPC client does not support browsing, or if a non-OPC client is being used, the tags can be created dynamically and statically by using the addresses given below.

Device Address	Description	Range	Data Type	Access
_CurrentPDUSize	Subsequent to connection, this tag shows the size of the Protocol Data Unit which has been negotiated with the device. Prior to connection it shows the maximum configured PDU value.	240, 480, 960	Word	Read

Data Types Description

Data Type	Description
Boolean	Single bit
Byte	Unsigned 8-bit value
Char	Signed 8-bit value
Word	Unsigned 16-bit value bit 0 is the low bit bit 15 is the high bit
Short	Signed 16-bit value bit 0 is the low bit bit 14 is the high bit bit 15 is the sign bit
DWord	Unsigned 32-bit value bit 0 is the low bit bit 31 is the high bit
Long	Signed 32-bit value bit 0 is the low bit bit 30 is the high bit bit 31 is the sign bit
BCD	Two byte packed BCD Value range is 0-9999. Behavior is undefined for values beyond this range.
LBCD	Four byte packed BCD Value range is 0-99999999. Behavior is undefined for values beyond this range.
Float	32-bit floating point value. The driver interprets two consecutive registers as a floating point value by making the second register the high word and the first register the low word.
String	NULL-terminated ASCII string

Address Descriptions

The following information applies to both S7-300 and S7-400 models. The default data types for dynamically defined tags are shown in **bold**.

Address Type	Range	Type	Access
Discrete Inputs	I0.b-I4095.b*	Boolean	Read/Write
	.b is Bit Number 0-7	Byte , Char, String**	Read/Write
	IB0-IB4095	Word , Short, BCD	Read/Write
	IW0-IW4094	DWord, Long	Read/Write
	IW:KT0-IW:KT4094	Word , Short	Read/Write
	IW:KC0-IW:KC4094	DWord , Long, LBCD, Float	Read/Write
Discrete Inputs	E0.b-E4095.b*	Boolean	Read/Write
	.b is Bit Number 0-7	Byte , Char, String**	Read/Write
	EB0-EB4095**	Word , Short, BCD	Read/Write
	EW0-EW4094	DWord, Long	Read/Write
	EW:KT0-EW:KT4094	Word , Short	Read/Write
	EW:KC0-EW:KC4094	DWord , Long, LBCD, Float	Read/Write
Discrete Outputs	Q0.b-Q4095.b*	Boolean	Read/Write
	.b is Bit Number 0-7	Byte , Char, String**	Read/Write
	QB0-QB4095	Word , Short, BCD	Read/Write
	QW0-QW4094	DWord, Long	Read/Write
	QW:KT0-QW:KT4094	Word , Short	Read/Write
	QW:KC0-QW:KC4094	DWord , Long, LBCD, Float	Read/Write
Discrete Outputs	A0.b- A4095.b*	Boolean	Read/Write
	.b is Bit Number 0-7	Byte , Char, String**	Read/Write
	AB0-AB4095	Word , Short, BCD	Read/Write
	AW0-AW4094	DWord, Long	Read/Write
	AW:KT0-AW:KT4094		

Note: I and E access the same memory area.

Note: Q and A access the same memory area.

Address Type	Range	Type	Access
	AW:KC0-AW:KC4094	Word , Short	Read/Write
	AD0-AD4092	DWord , Long, LBCD, Float	Read/Write
Internal Memory	F0.b-F4095.b* .b is Bit Number 0-7	Boolean	Read/Write
	FB0-FB4095	Byte , Char, String* *	Read/Write
	FW0-FW4094	Word , Short, BCD	Read/Write
	FW:KT0-FW:KT4094	DWord, Long	Read/Write
	FW:KC0-FW:KC4094	Word , Short	Read/Write
	FD0-FD4092	DWord , Long, LBCD, Float	Read/Write
Internal Memory	M0.b-M4095.b* .b is Bit Number 0-7	Boolean	Read/Write
	MB0-MB4095	Byte , Char, String* *	Read/Write
	MW0-MW4094	Word , Short, BCD	Read/Write
	MW:KT0-MW:KT4094	DWord, Long	Read/Write
	MW:KC0-MW:KC4094	Word , Short	Read/Write
	MD0- MD4092	DWord , Long, LBCD, Float	Read/Write
	Note: F and M access the same memory area.		
Data Block Boolean	DB1-N:KM0.b-KM4094.b* 1-N is Block Number .b is Bit Number 0-15	Boolean	Read/Write
	<i>Alternates</i>		
	DB1DBX0.b-DBNDBX4094.b* 1-N is Block Number .b is Bit Number 0-15	Boolean	Read/Write
	DB1D0.b-DBND4094.b* 1-N is Block Number .b is Bit Number 0-15	Boolean	Read/Write
Data Block Left Byte	DB1-N:KL0-KL4095 1-N is Block Number	Byte , Char, String* *	Read/Write
	<i>Alternates</i>		
	DB1DBB0-DBNDBB4095 1-N is Block Number	Byte , Char, String* *	Read/Write

Address Type	Range	Type	Access
	DB1DL0-DBNDL4095 1-N is Block Number	Byte , Char, String**	Read/Write
Data Block Right Byte	DB1-N:KR0-KR4094 1-N is Block Number <i>Alternates</i> DB1DR0-DBNDR4094 1-N is Block Number	Byte , Char, String** Byte , Char, String**	Read/Write Read/Write
Data Block Unsigned Word	DB1-N:KH0-KH4094 1-N is Block Number	Word , Short, BCD	Read/Write
Data Block Signed Word	DB1-N:KF0-KF4094 1-N is Block Number <i>Alternates</i> DB1DBW0-DBNDBW4094 1-N is Block Number DB1DW0-DBNDW4094 1-N is Block Number	Word, Short , BCD Word, Short , BCD Word, Short , BCD	Read/Write Read/Write Read/Write
Data Block Signed Long	DB1-N:KD0-KD4092 1-N is Block Number <i>Alternates</i> DB1DBD0-DB1DBD4092 1-N is Block Number DB1DD0-DB1DD4092 1-N is Block Number	DWord, Long , LBCD, Float DWord, Long , LBCD, Float DWord, Long , LBCD, Float	Read/Write Read/Write Read/Write
Data Block Float	DB1-N:KG0-KG4092 1-N is Block Number	Float	Read/Write
Data Block BCD	DB1-N:BCD0-BCD4094 1-N is Block Number	Word , Short	Read/Write
Data Block S5 Timer as DB	DB1-N:KT0-KT4094 1-N is Block Number	DWord, Long	Read/Write
Data Block S5 Counter as DB	DB1-N:KC0-KC4094 1-N is Block Number	Word , Short	Read/Write
Data Block String	DB1:S0.n-DB1:S4095.n* .n is string length. 0<n<=932.	String	Read/Write

* These memory types/subtypes do not support arrays.

** Byte memory types (MB) support strings. The syntax for strings is <address>.<length> where 0<length<=<=932.

Notes:

1. All offsets for memory types I, Q, and F represent a byte starting location within the specified memory type.
2. Use caution when modifying Word, Short, DWord, and Long types. For I, Q and F, each address starts at a byte offset within the device. Therefore, Words FW0 and FW1 overlap at byte 1. Writing to FW0 modifies the value held in FW1. Similarly, DWord, and Long types can also overlap. It is recommended that these memory types be used so that overlapping does not occur. For example, when using DWords, FD0, FD4, FD8 and so on should be used to prevent overlapping bytes.
3. For strings, the total number of bytes requested cannot exceed the data portion of the negotiated PDU size. If raw strings exceed the negotiated PDU size, they may fail to be read or written.

Arrays

All memory types/subtypes support arrays (excepting those discussed above). The valid syntax for declaring an array is described below. If no rows are specified, row count of 1 is assumed.

```
<address>[rows][cols]
<address>.rows.cols
<address>,rows,cols
<address>_rows_cols
```

For Word, Short, BCD and "KT" arrays, the base address+(rows* cols* 2) cannot exceed 4096. The elements of the array are words, and are located on a word boundary. For example, IW0[4] would return IW0, IW2, IW4 and IW6. "KT" subtypes fall into the 16-bit category because the data stored in the PLC is contained within a Word.

For Float, DWord, Long and Long BCD arrays (excluding "KT" subtypes), the base address+(rows* cols* 4) cannot exceed 4096. Keep in mind that the elements of the array are DWords, located on a DWord boundary. For example, ID0[4] returns ID0, ID4, ID8 and ID12.

For all arrays, the total number of bytes being requested cannot exceed the data portion of the negotiated PDU size. For example, given a 960 byte PDU size, the largest single array that may be read or written is 932 bytes.

KL vs. KR vs. DBB

KL and KR determine whether the left byte or right byte of the data block word is returned.

Value	8	9	A	B	C
Byte	0	1	2	3	4

Example 1

```
DB1:KH0=0x89
DB1:KL0=0x8
DB1:KR0=0x9
DB1:DBB0=0x8
```

Example 2

```
DB1:KH1=0x9A
DB1:KL1=0x9
```

DB1:KR1=0xA
DB1:DBB1=0x9

Examples

- To access bit 3 of Internal Memory F20, declare an address as follows: F20.3
- To access Data Block 5 as word memory at byte 30, declare an address as follows: DB5:KH30
- To access Data Block 2 byte 20 and bit 7, declare an address as follows: DB2:KM20.7
- To access Data Block 1 as left byte memory at byte 10, declare an address as follows: DB1:KL10
- To access Internal Memory F20 as a DWord, declare an address as follows: FD20
- To access Input Memory I10 as a Word, declare an address as follows: IW10

Event Log Messages

The following information concerns messages posted to the Event Log pane in the main user interface. Consult the OPC server help on filtering and sorting the Event Log detail view. Server help contains many common messages, so should also be searched. Generally, the type of message (informational, warning) and troubleshooting information is provided whenever possible.

Failure to start unsolicited communications. | Port number = <number>.

Error Type:

Error

Possible Cause:

1. The driver was not able to create a listen socket for unsolicited communications. Another application may be using the port specified.
2. There may be low system resources.

Possible Solution:

1. Use network monitor software to see if another application is using the port. If so, shut down the conflicting application and restart the OPC Server. If the conflicting application is free to pick any available port, make sure the server is always started first so it can claim the required port. If both the PLC programming software and this driver must use the same port, they may not be able to be used simultaneously.
2. Verify there are adequate system resources or release resources from other processes.

See Also:

Channel Setup

Appendix: Configuring Connections Using the SIMATIC Manager

Connections are configured using the SIMATIC Manager software. The following topics provide information on configuring the Siemens TCP/IP Server Ethernet Driver to run in unsolicited mode, and demonstrate a basic setup using the S7-300 PLC as the active partner and the driver as the passive partner.

● **Note:** The Siemens TCP/IP Server Ethernet Driver can configure 256 devices, each with an associated slot/rack. When the active partner (Siemens client) communicates with the passive partner (unsolicited or Siemens server driver), it directs its requests to a specific device in the unsolicited driver. Multiple remote partners can talk to the same device.

To jump to a specific section, select a link from the list below.

[Step One: Creating a New Project](#)

[Step Two: Configuring the Siemens Client and PC Station](#)

[Step Three: Connecting the Siemens Client and the Siemens Server Driver](#)

[Step Four: Inserting Function Blocks](#)

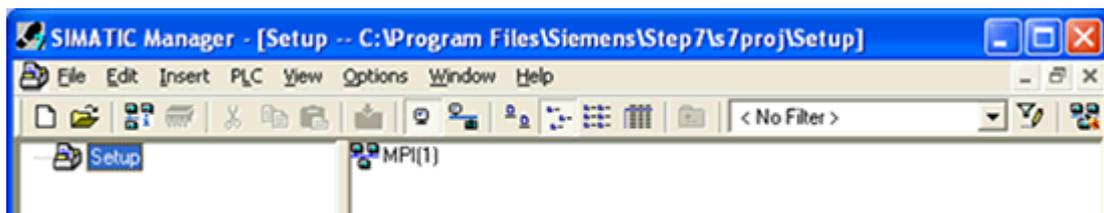
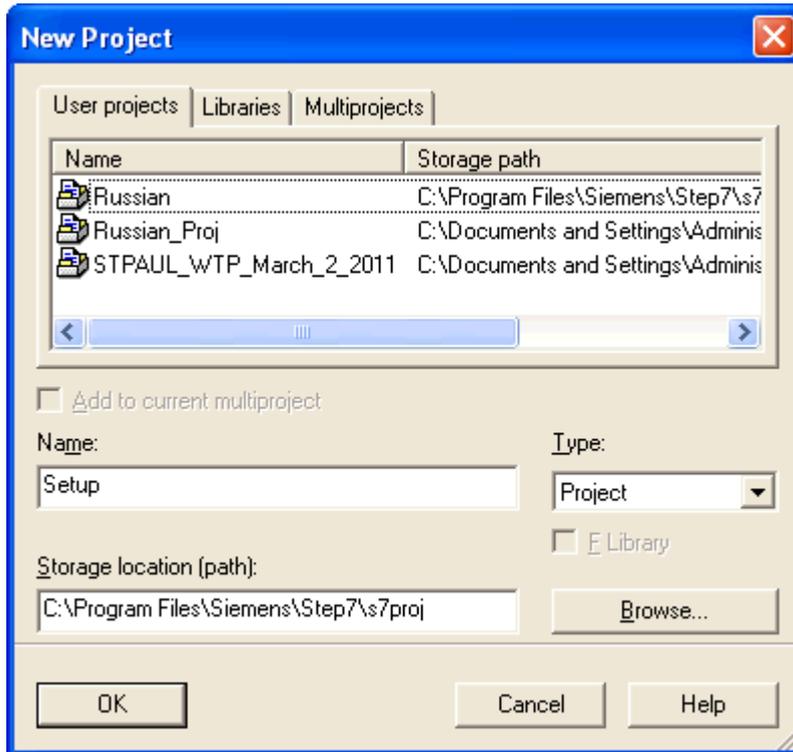
[Step Five: Creating the DB3 Data Block](#)

[Step Six: Inserting PUT FB](#)

[Step Seven: Downloading to the PLC](#)

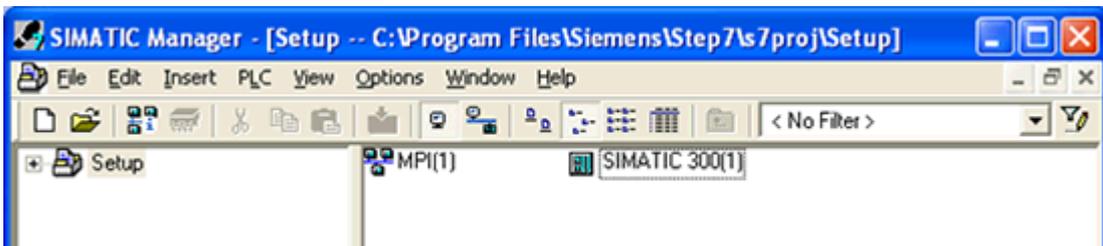
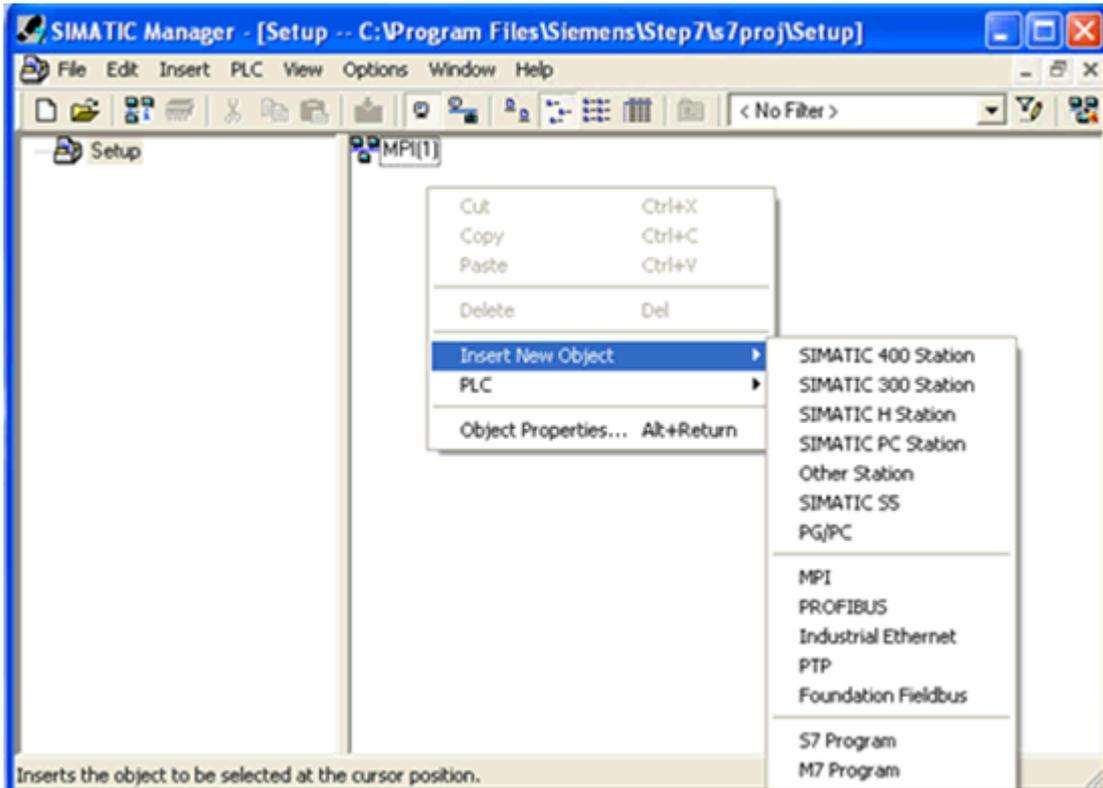
Step One: Creating a New Project

1. To start, open the SIMATIC Manager software and then create a new project. In this example, the project being used is "Setup".

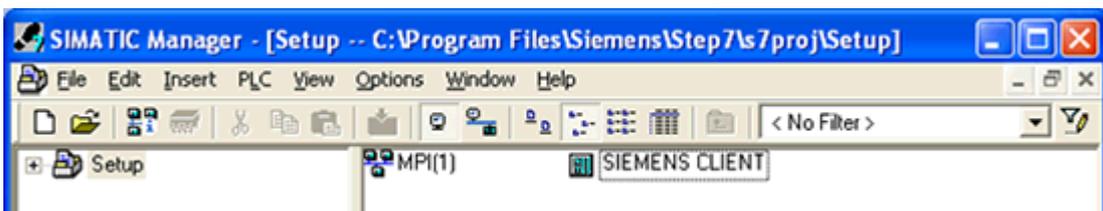


2. Create the Siemens client and PC Station. To do so, right-click in the right pane of the window and then select **Insert New Object | SIMATIC 300 Station**.

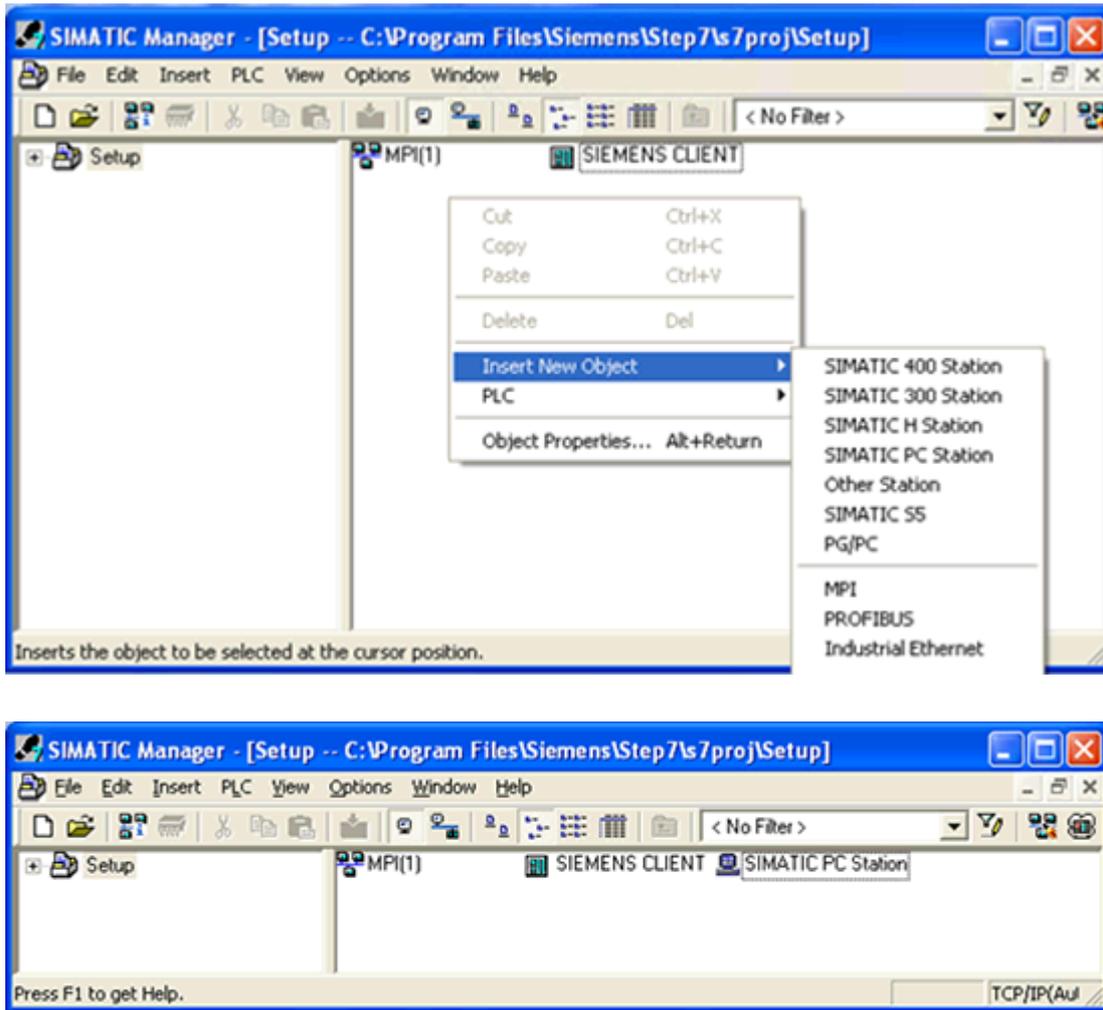
Tip: The Siemens client unit is the active partner or the image of the actual PC. The PC Station is the PC on which the SIMATIC Manager software is running.



3. Name the new station "SIEMENS-CLIENT" because it represents the communication's active partner.



4. Right-click in the right pane of the window and select **Insert New Object | SIMATIC PC Station**.

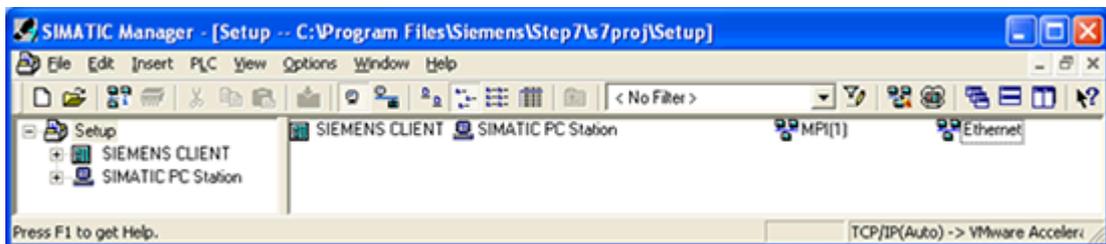
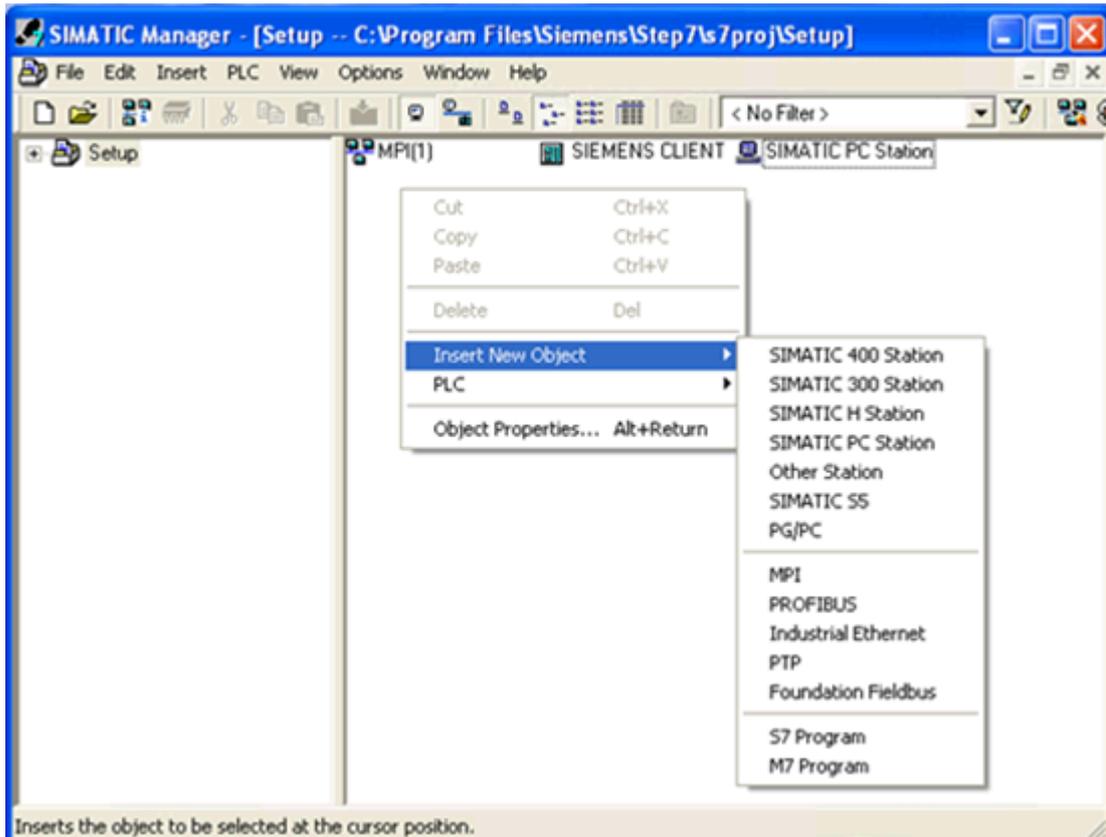


For more information, refer to [Step Two: Configuring the Siemens Client and PC Station](#).

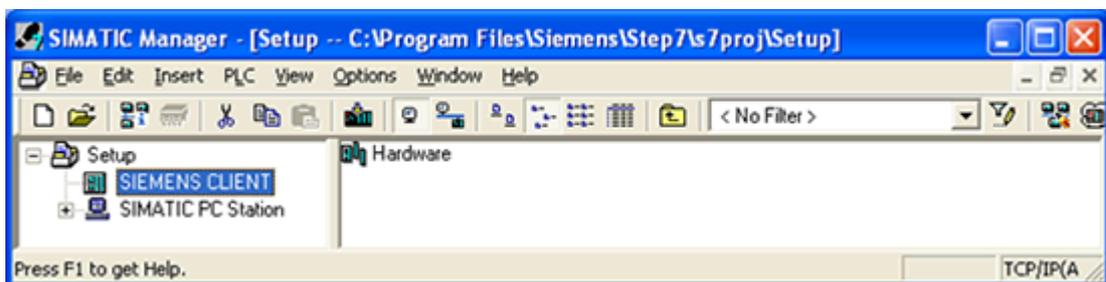
Step Two: Configuring the Siemens Client and PC Station

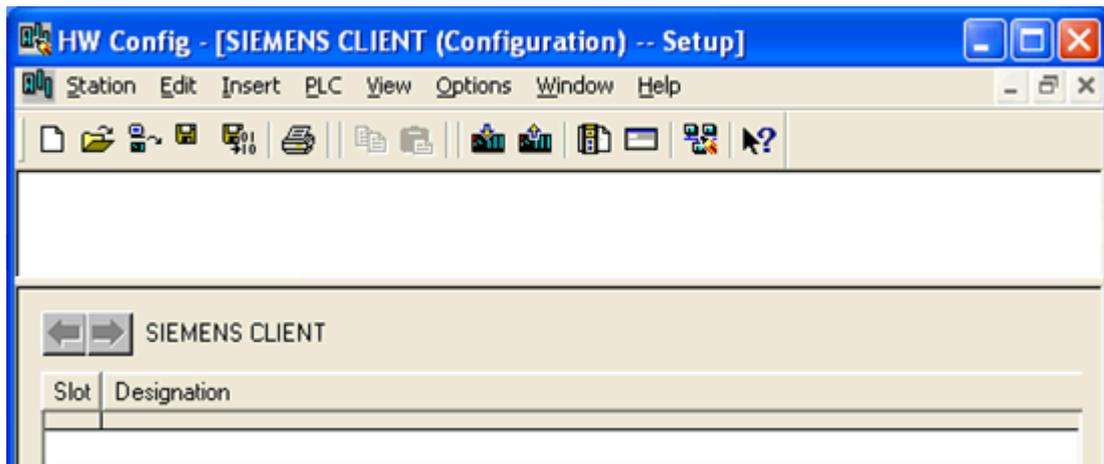
Industrial Ethernet (IE) is the protocol used for communication.

1. To start, right-click in the right pane of the SIMATIC Manager window. Select **Insert New Object | Industrial Ethernet**.



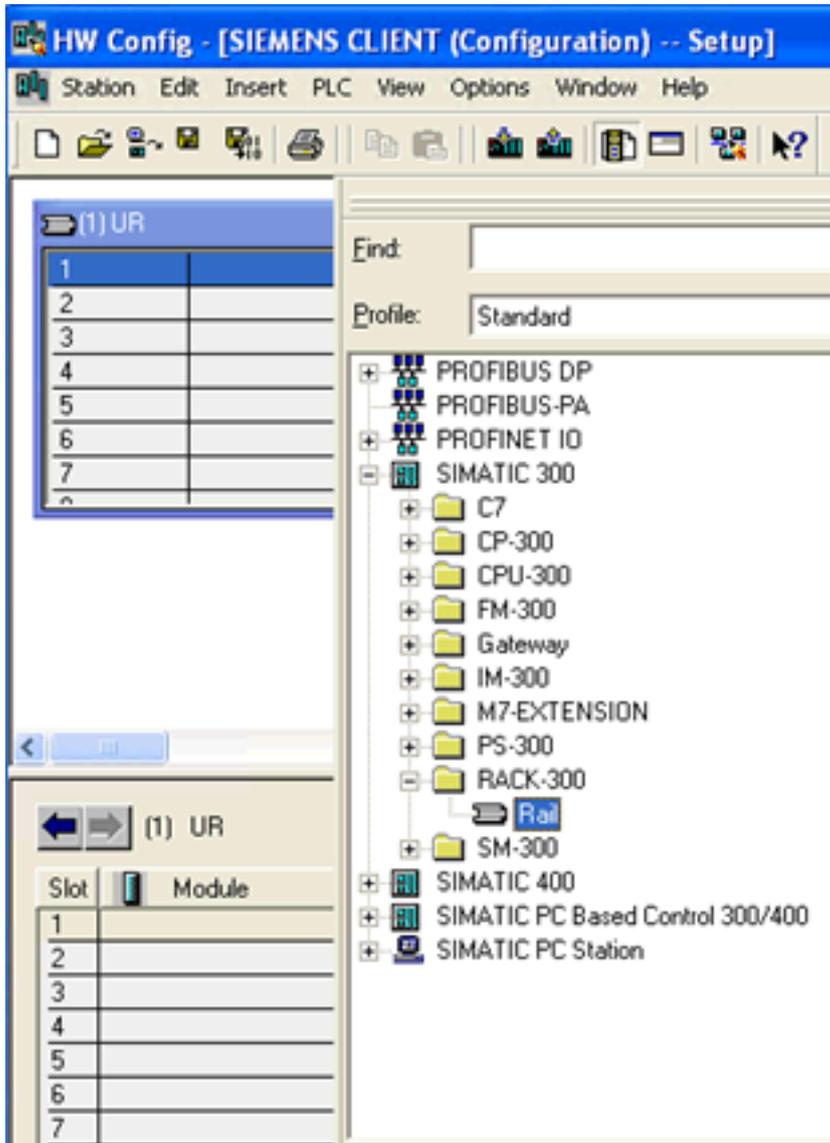
2. Select the SIEMENS CLIENT icon in the left pane of the window and double-click **Hardware**.



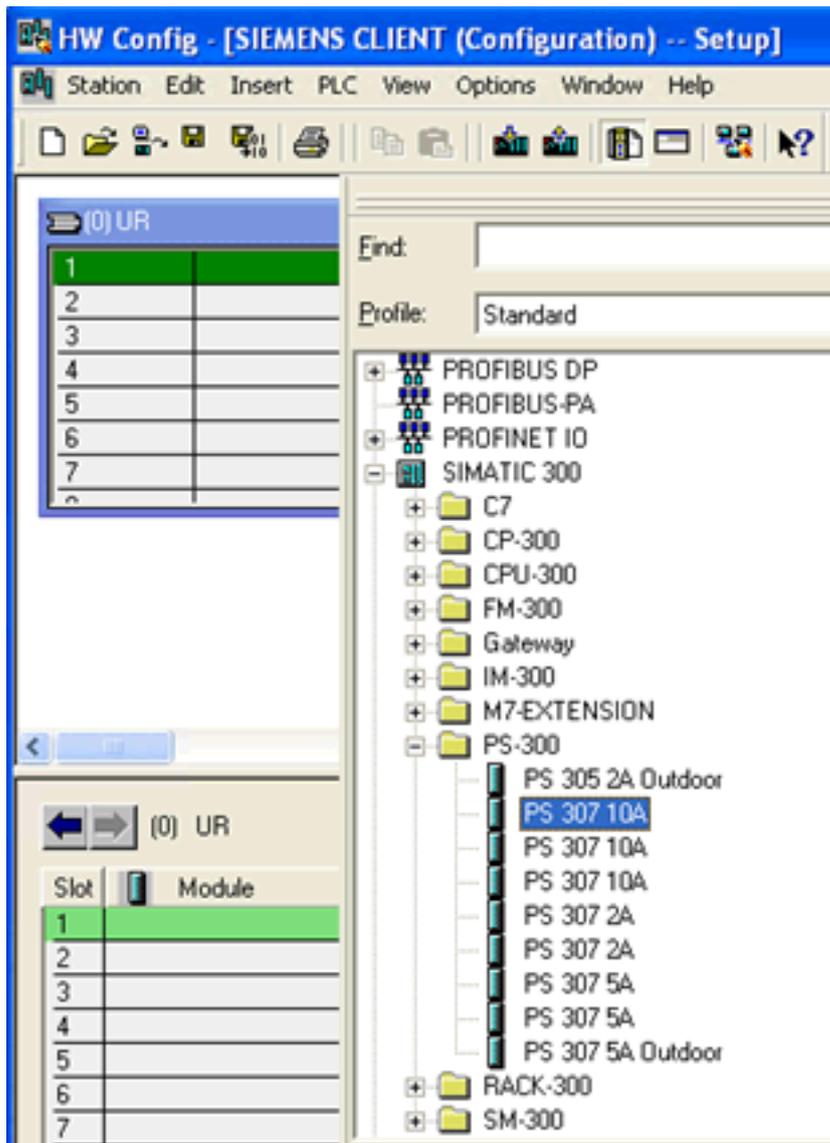


3. Open the **View** tab and select **Catalog**.
4. Expand the **SIMATIC 300** menu and the **Rack 300** menu.

5. To insert the racks, double-click on **Rail**.

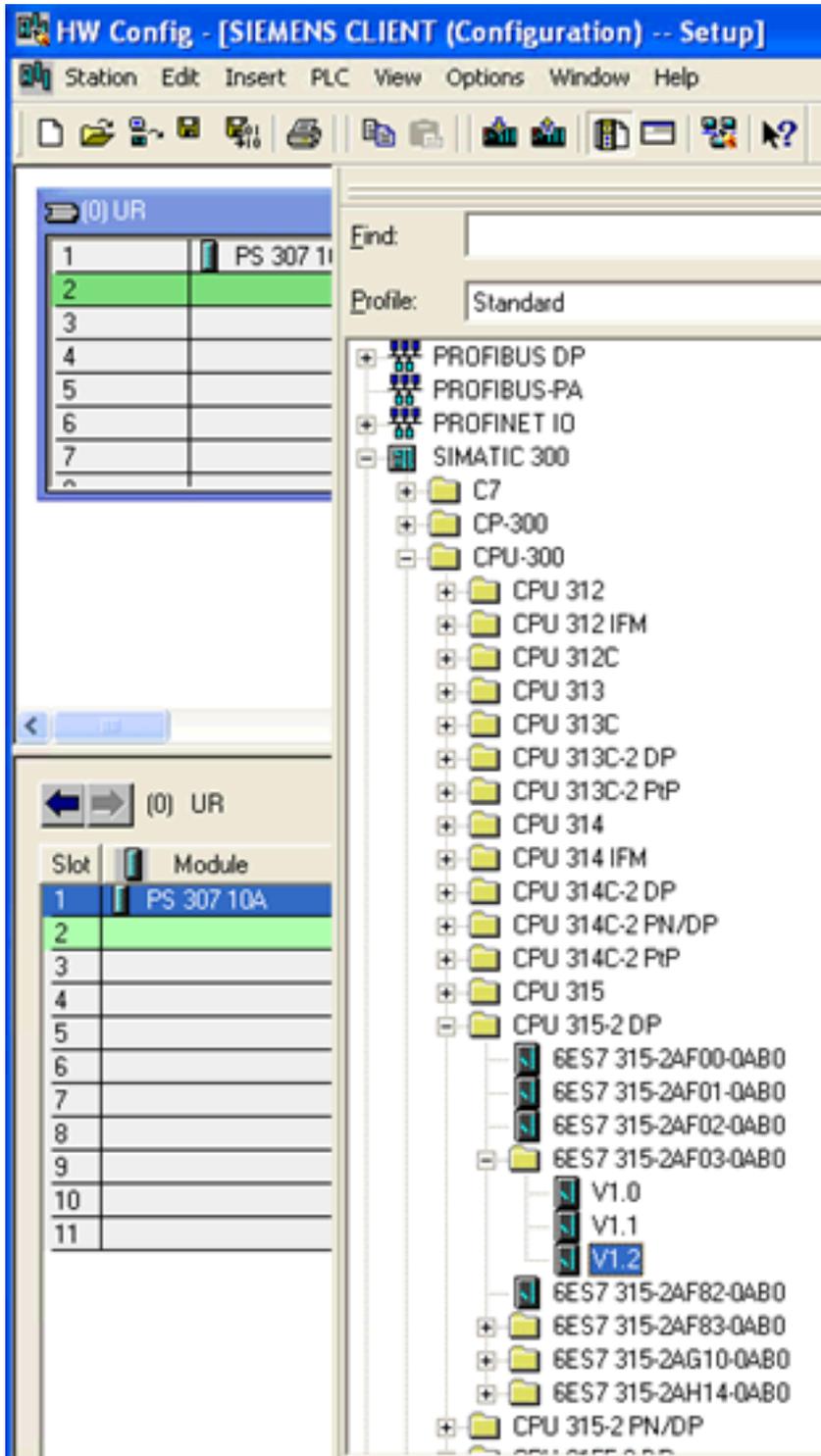


6. Expand the **PS 300** menu. Double-click on **PS 307 10A** or any other suitable option to insert the power supply into slot 1.

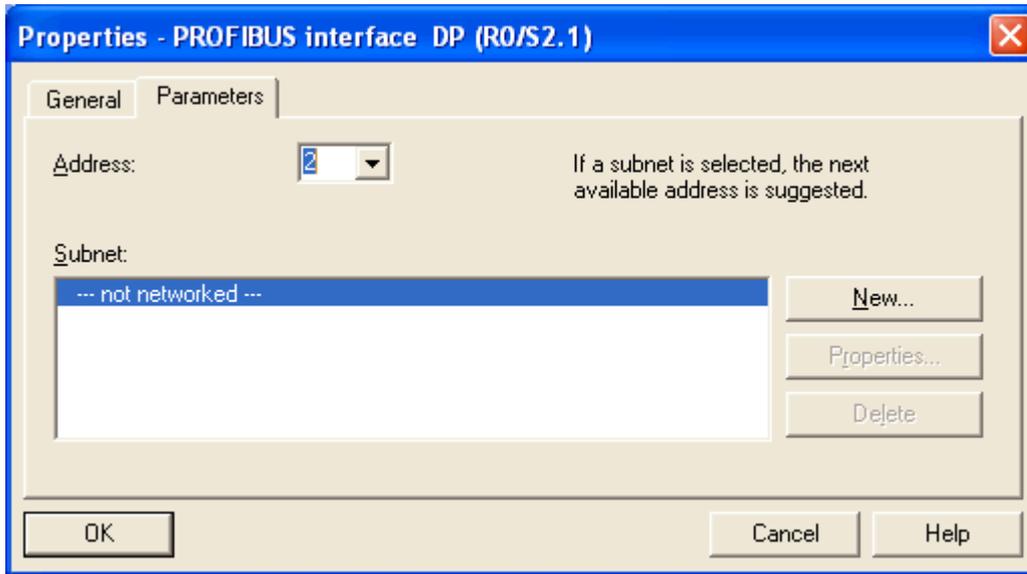


7. To insert the CPU, expand both the **CPU 300** menu and the **CPU 315-2 DP** menu.

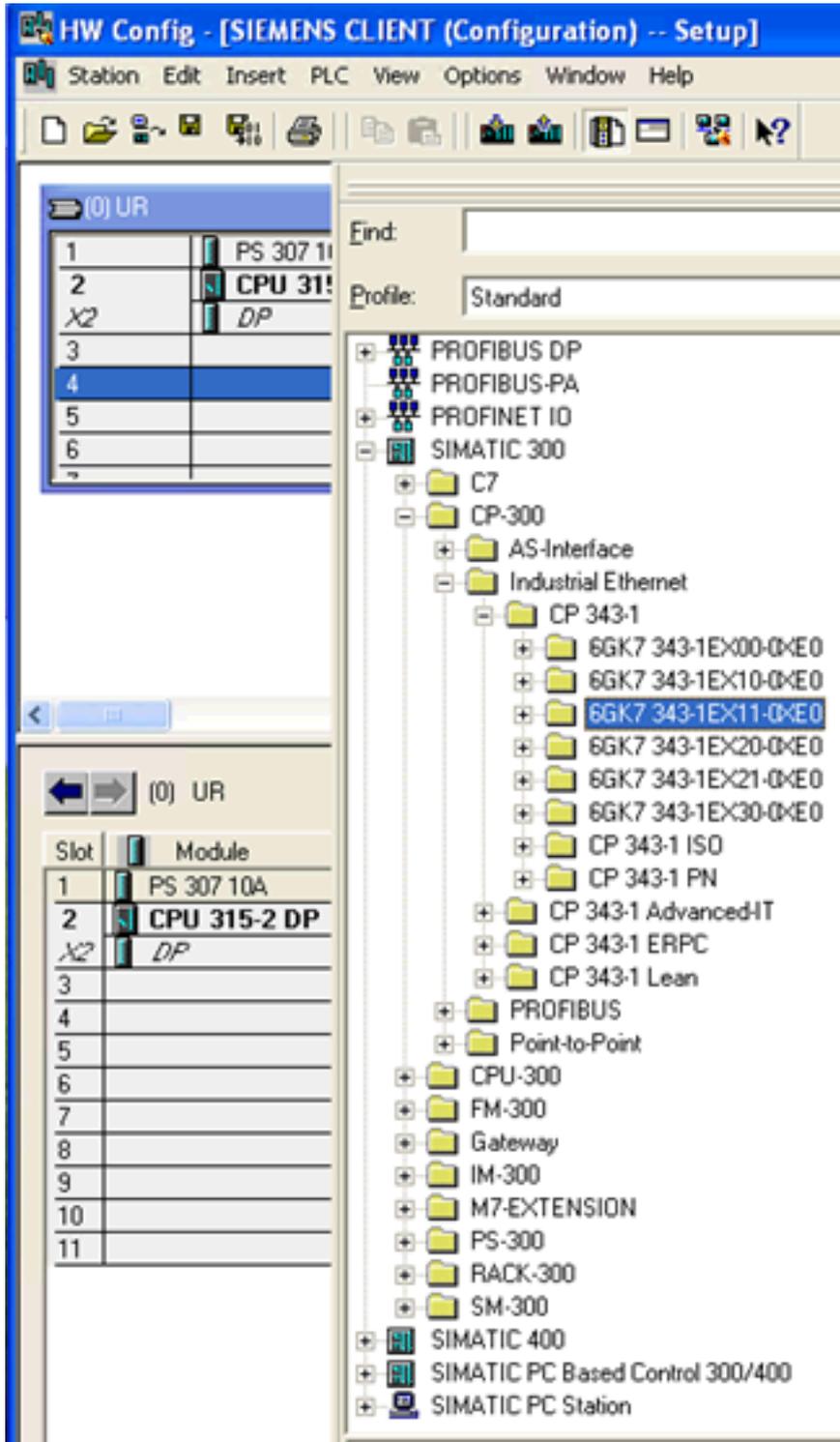
- 8. Double-click on the CPU that matches the hardware.

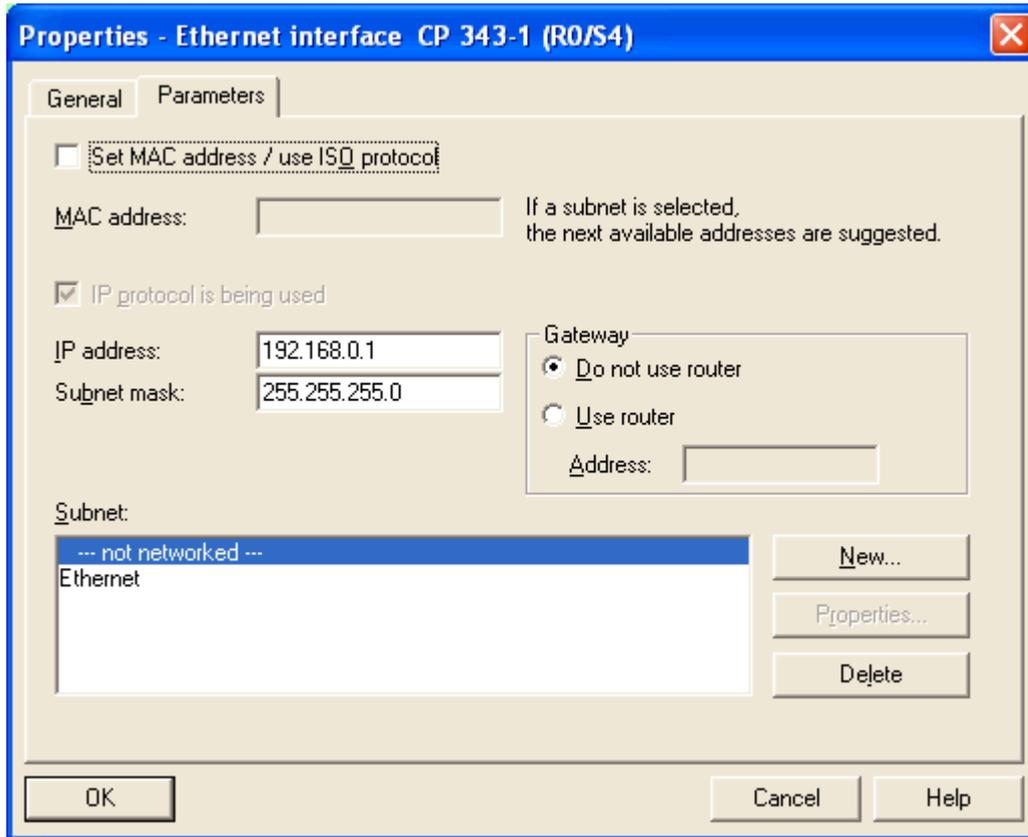


9. To insert the CPU into slot 2, click **OK**.



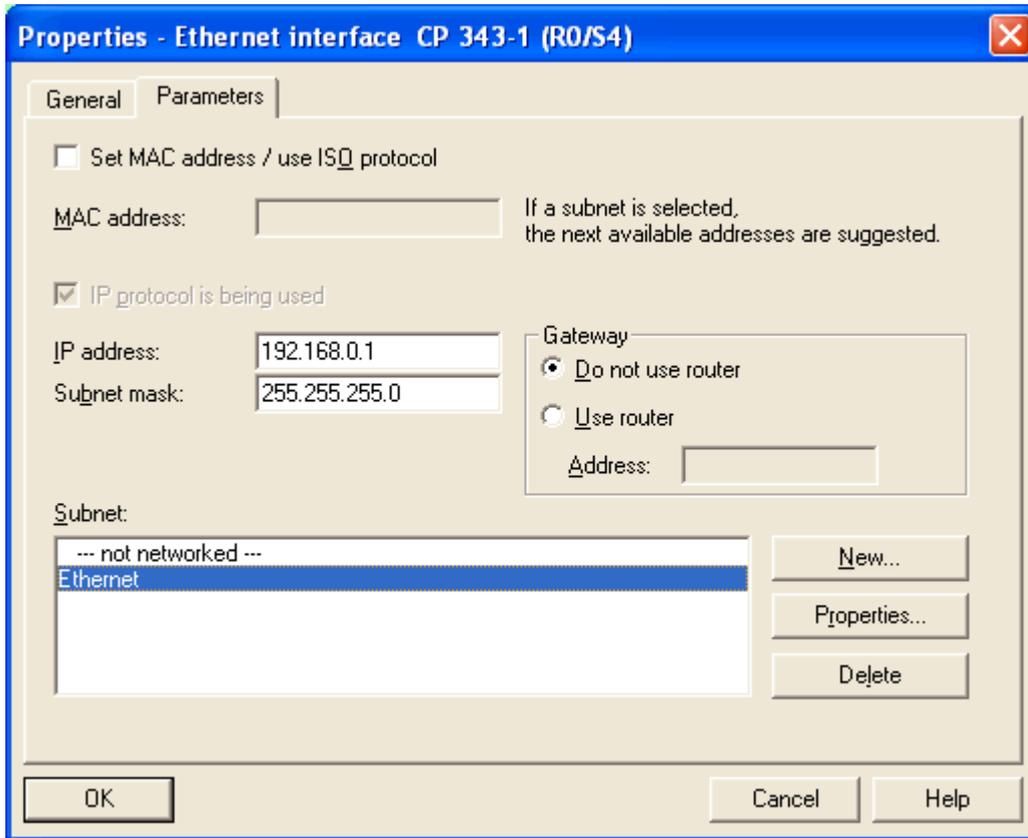
10. To insert the CP, leave slot 3 empty and then click on slot 4 in the racks.
11. Expand both the **CP 300** menu and the **Industrial Ethernet** menu.
12. Double-click on the CP that matches the hardware.



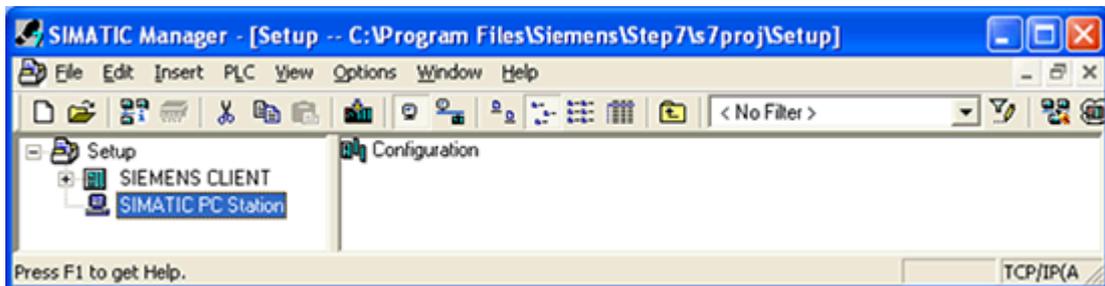


13. Enter the PLC's IP address and subnet mask. Select **Ethernet** from the subnet box.

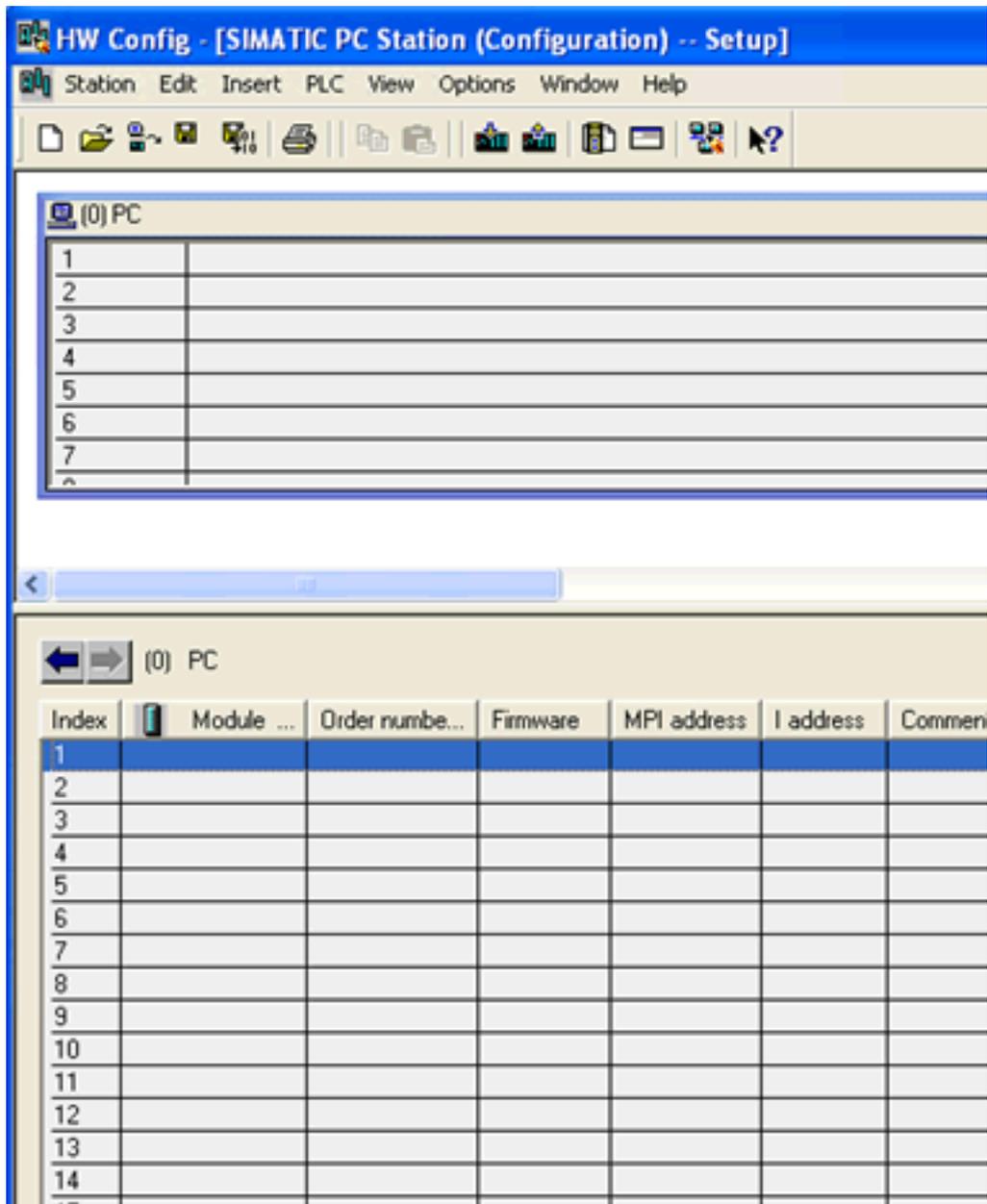
14. Click **OK** to configure the Siemens client.



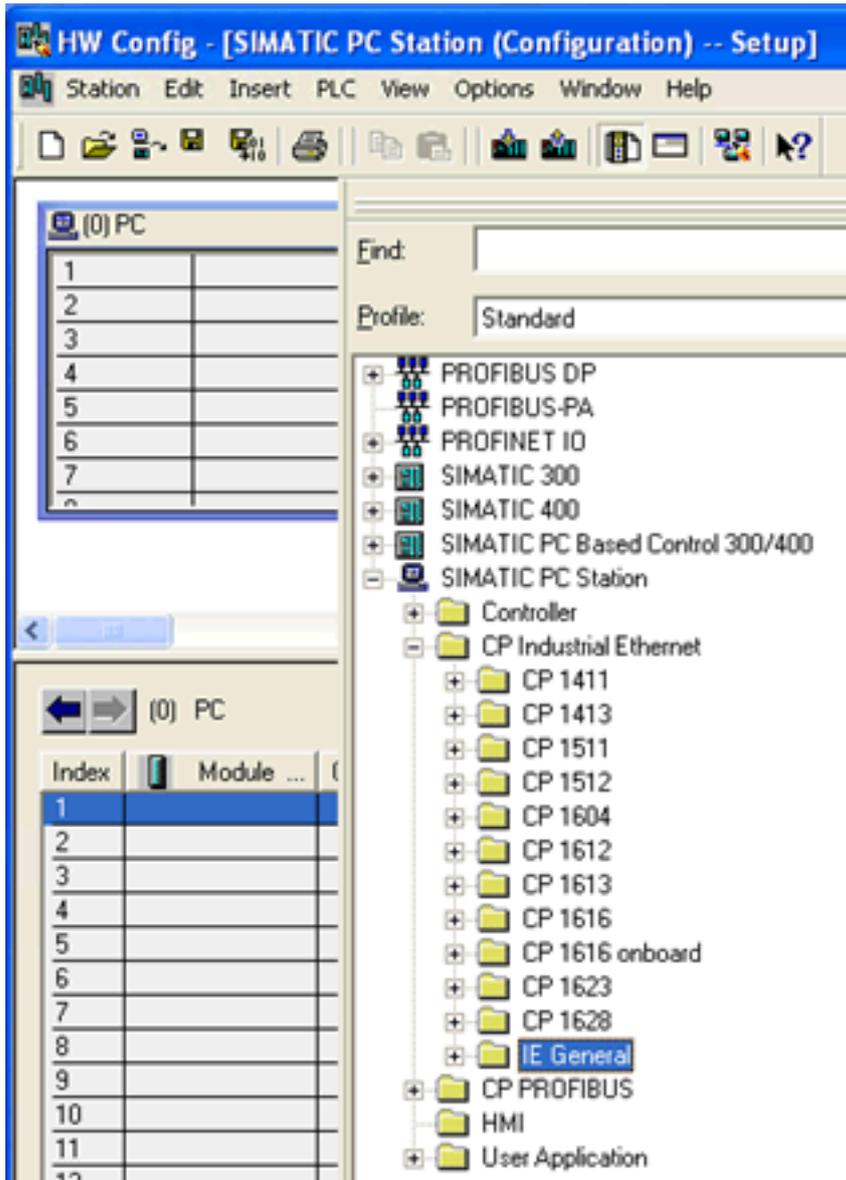
15. Once finished, open the **View** tab and select **Catalog** to hide the catalog window.
16. Save and exit the HW Configuration window.
17. To configure the PC station, click on the SIMATIC PC Station in the left pane of the SIMATIC Manager window.
18. Double-click on **Configuration**.

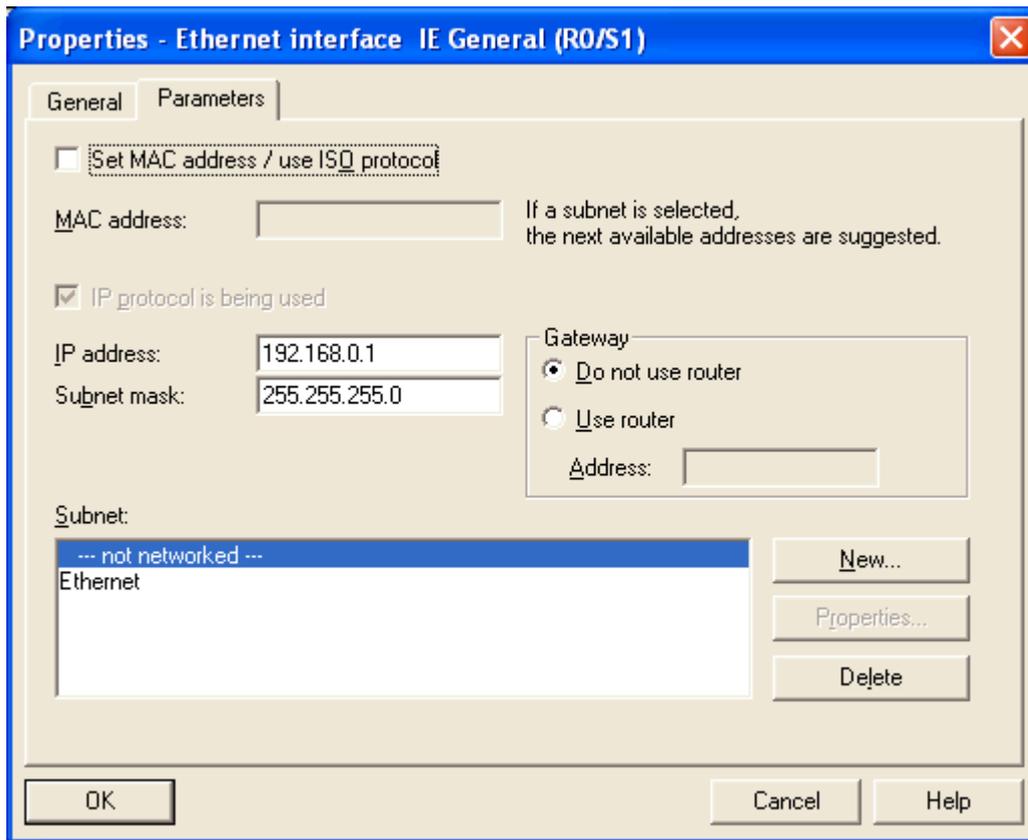


19. Click on the **View** tab and select **Catalog**.



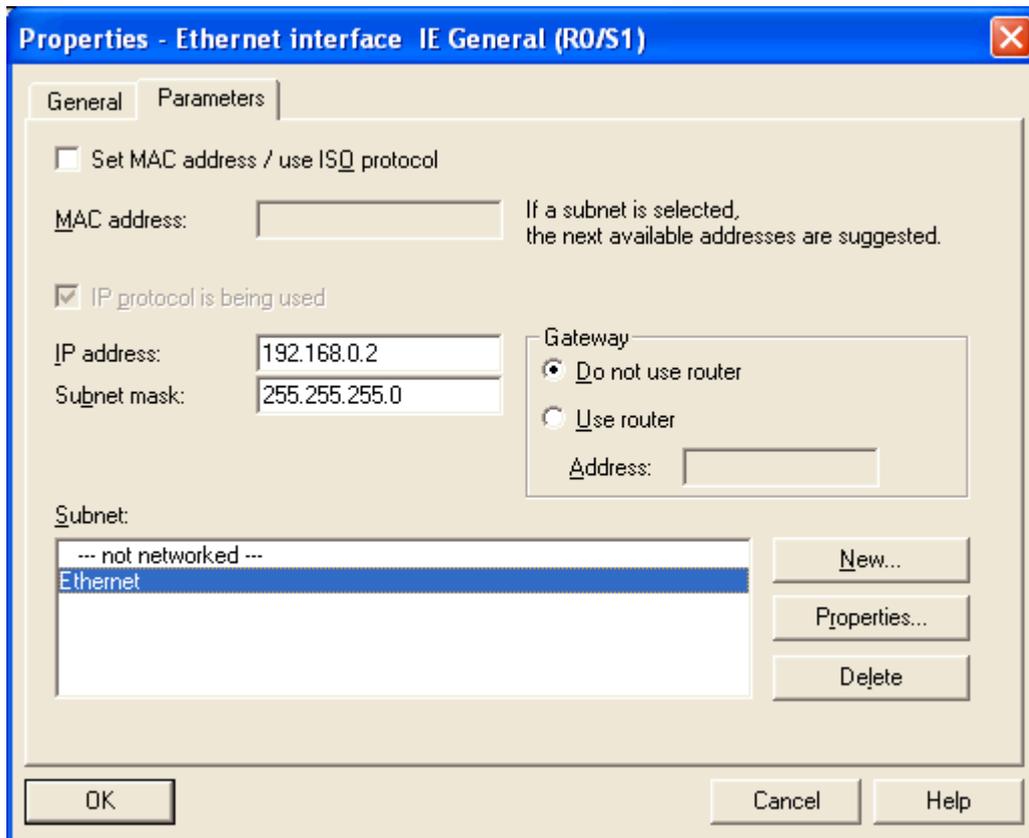
20. Expand both the **SIMATIC PC Station** menu and the **CP Industrial Ethernet** menu.
21. Double-click on **General** or any other suitable option.





22. Enter the IP address of the PC running the SIMATIC Manager software, in addition to the correct subnet mask.
23. Select **Ethernet** from the subnet box.

24. Click **OK** to configure the PC station.



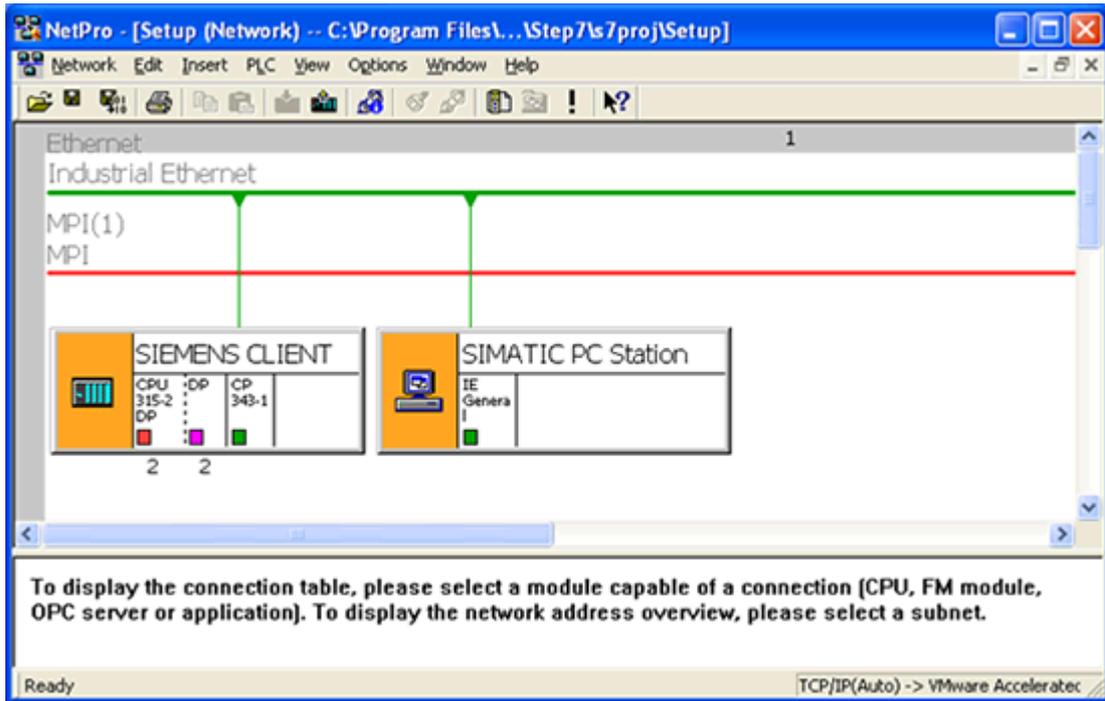
25. Once finished, open the **View** tab and select **Catalog** to hide the catalog window.
26. Save and exit the HW Configuration window.

For more information, refer to [Step Three: Connecting the Client and the Server Driver](#).

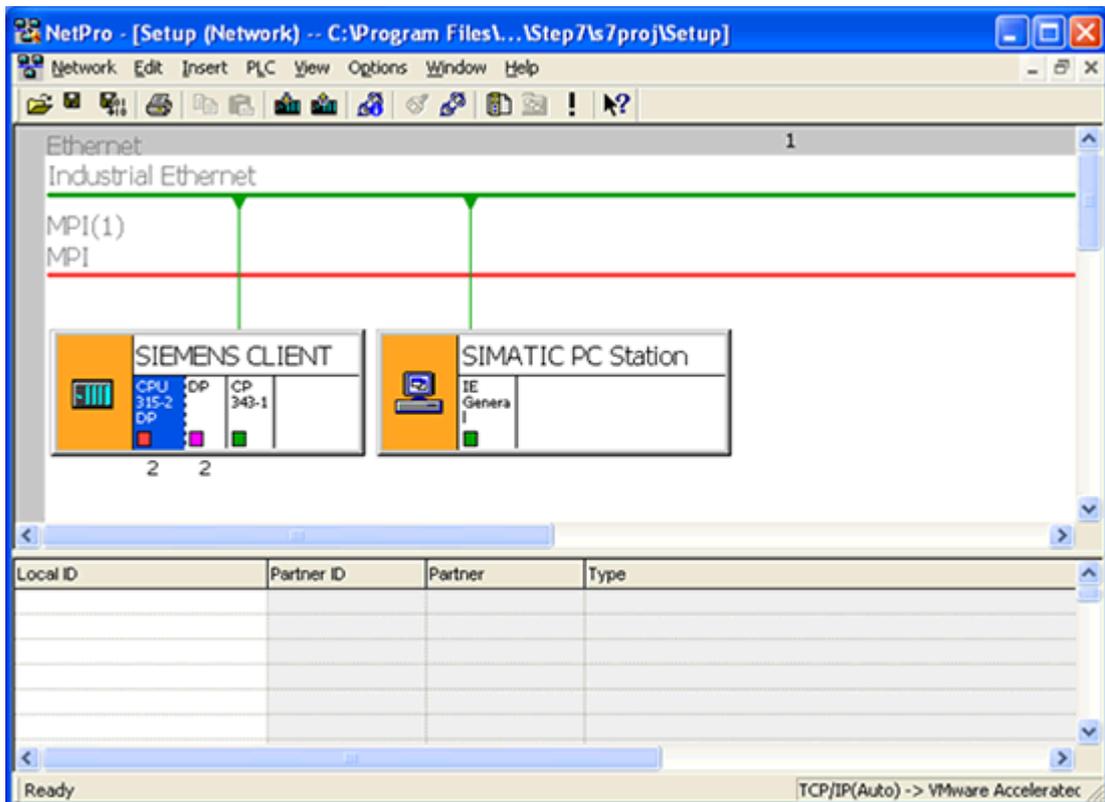
Step Three: Connecting the Siemens Client and the Siemens Server Driver

Once the Siemens client and the PC Station have been successfully configured, the Siemens client and the Siemens server must be connected.

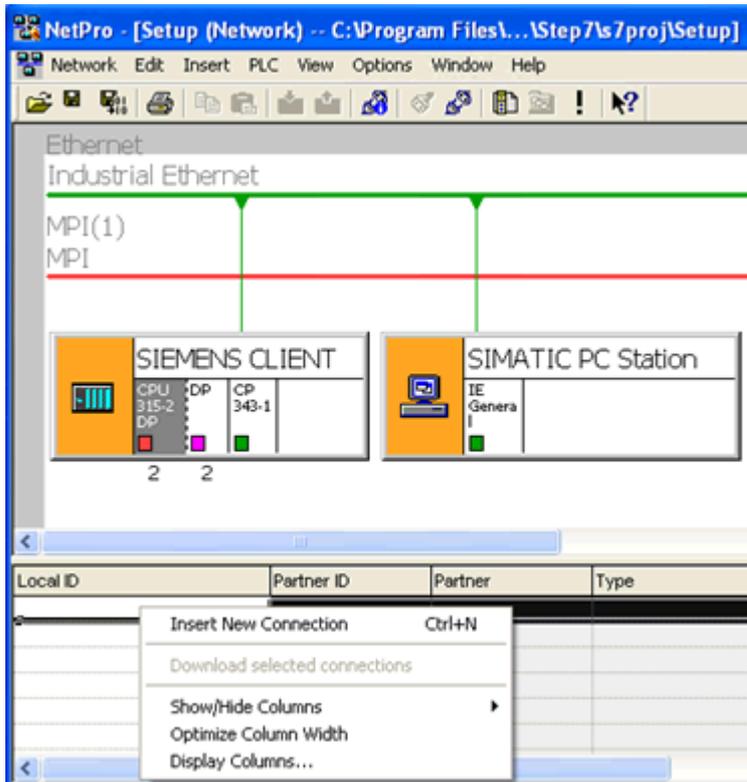
1. To start, open the **Options** tab in the SIMATIC Manager window and select **Configure Network**.



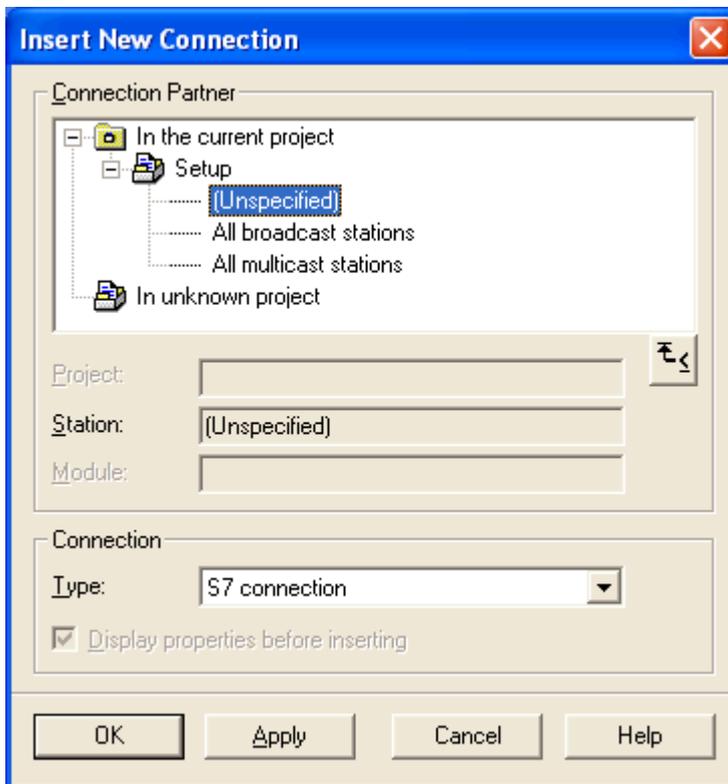
2. Click on the Siemens client's **CPU 315-2 DP** block. A series of rows should be displayed in the lower half of the window.

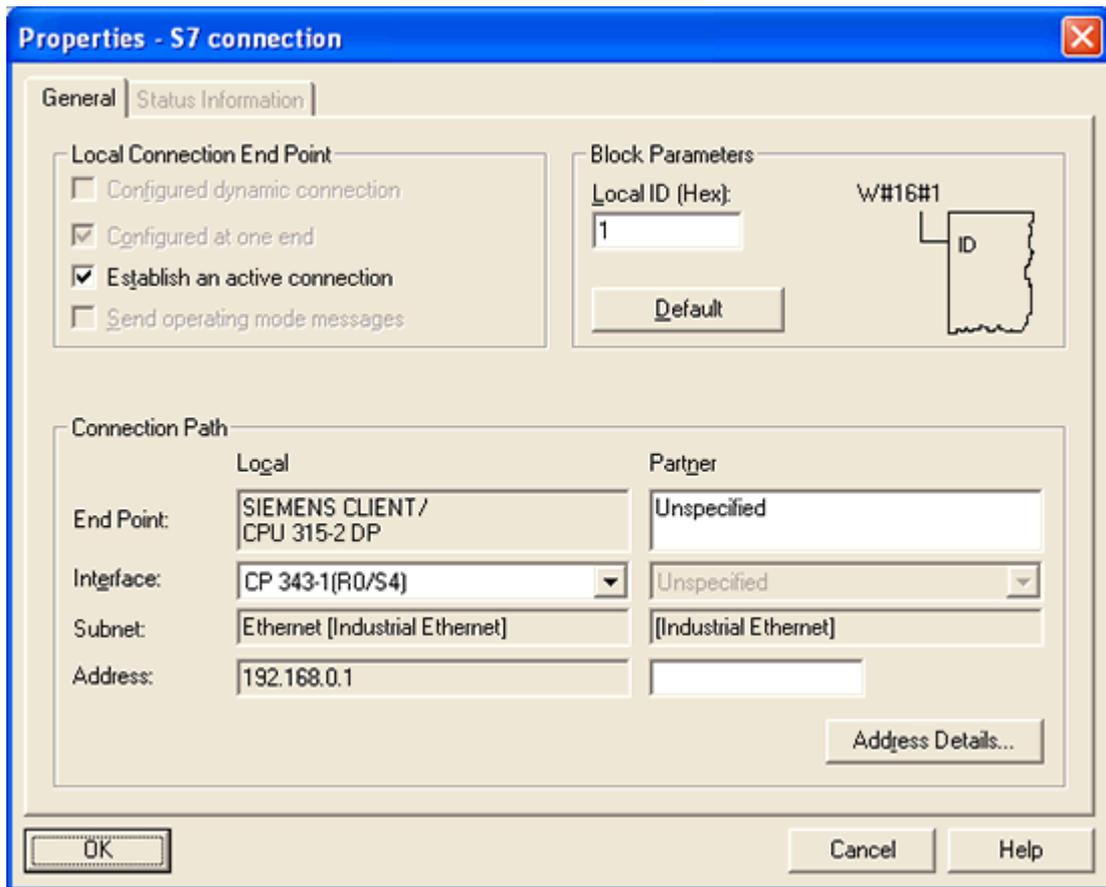


- Right-click on the first row and select **Insert New Connection**.



- Click **OK**.





5. Enter the IP address of the machine on which the Siemens TCP/IP Server Ethernet Driver runs.

Properties - S7 connection

General | Status Information

Local Connection End Point

Configured dynamic connection

Configured at one end

Establish an active connection

Send operating mode messages

Block Parameters

Local ID (Hex): W#16#1

1

Default

Connection Path

	Local	Partner
End Point:	SIEMENS CLIENT / CPU 315-2 DP	Unspecified
Interface:	CP 343-1(R0/S4)	Unspecified
Subnet:	Ethernet (Industrial Ethernet)	Industrial Ethernet
Address:	192.168.0.1	192.168.111.6

Address Details...

OK Cancel Help

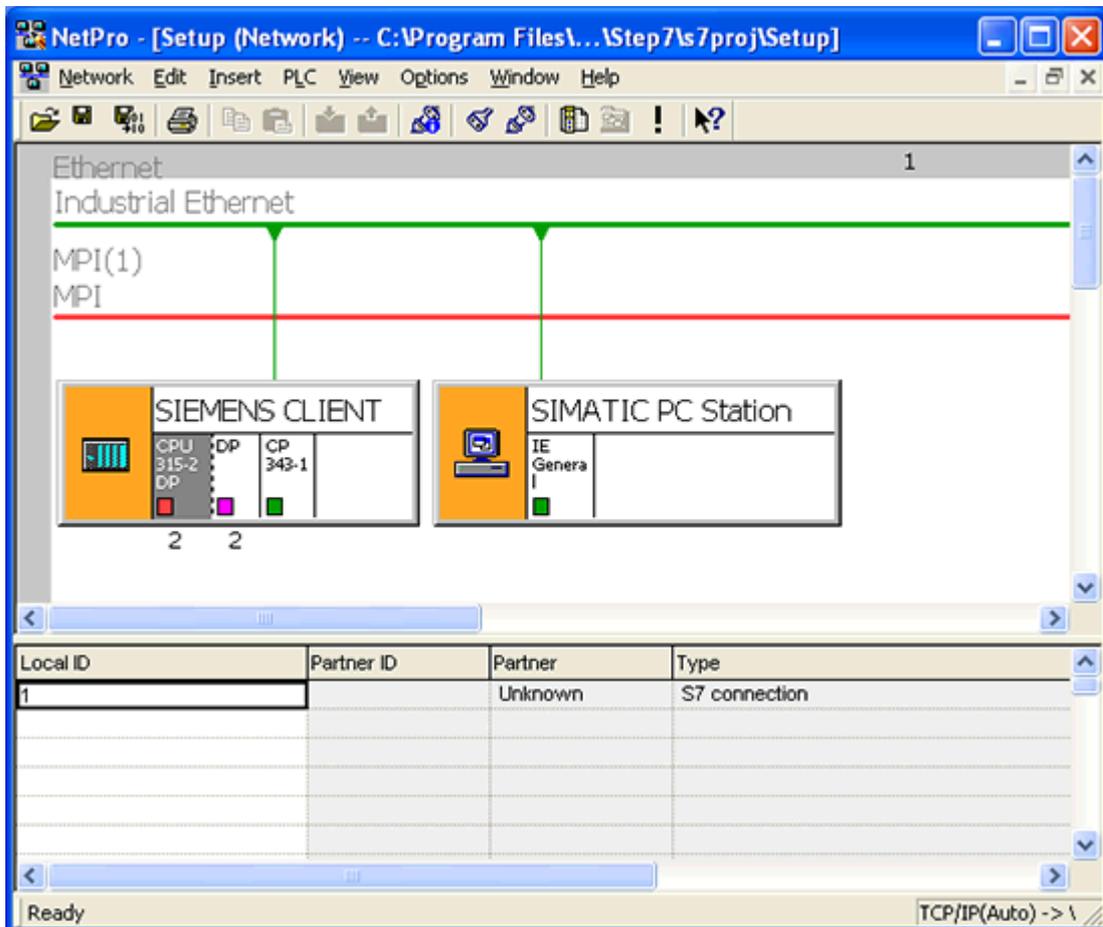
6. Click **Address Details** and enter the rack/slot values of the device in the unsolicited driver with which the Siemens client should communicate.

Address Details

	Local	Partner
End Point:	SIEMENS CLIENT / CP 343-1	Unspecified
Rack/Slot:	0 4	0 4
Connection Resource (hex):	10	03
TSAP:	10.04	03.04
S7 Subnet ID:	0071 - 0002	-

OK Cancel Help

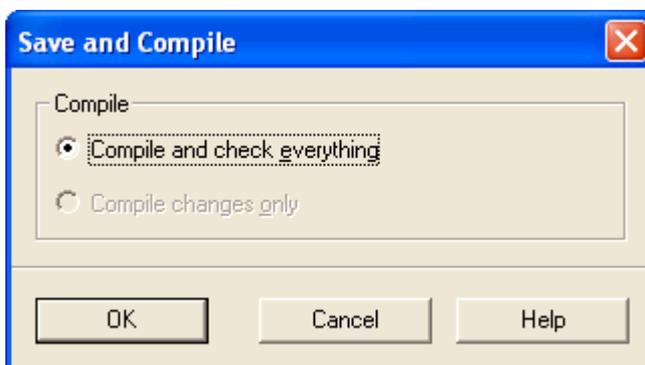
7. Click **OK** twice to successfully connect the Siemens client and server drivers. The Siemens client uses these settings to communicate with the destination device at rack 0 and slot 2.



● **Note:** The Local ID number (=1) identifies the connection between the two partners. This number is used later when creating function blocks for reading and writing data.

8. Save and compile the data by opening the **Network** tab and selecting **Save and Compile**. Click **OK**.

● **Note:** There should be no errors on compilation.

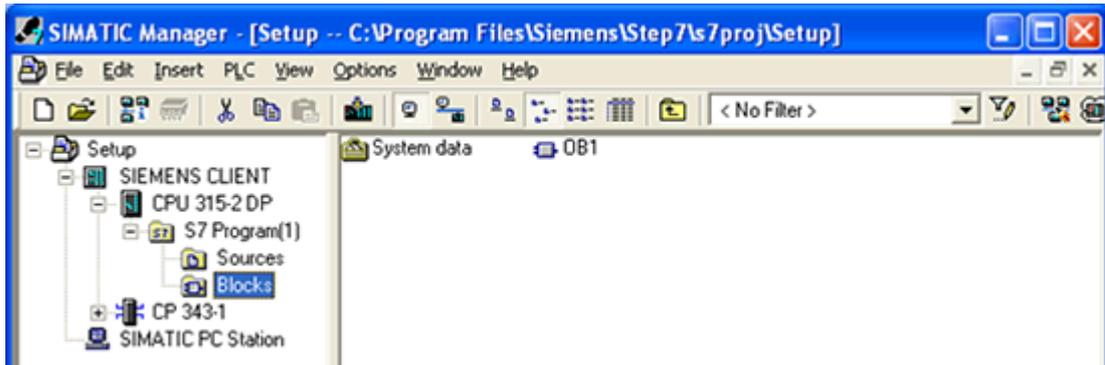


● For more information, refer to [Step Four: Inserting Function Blocks](#).

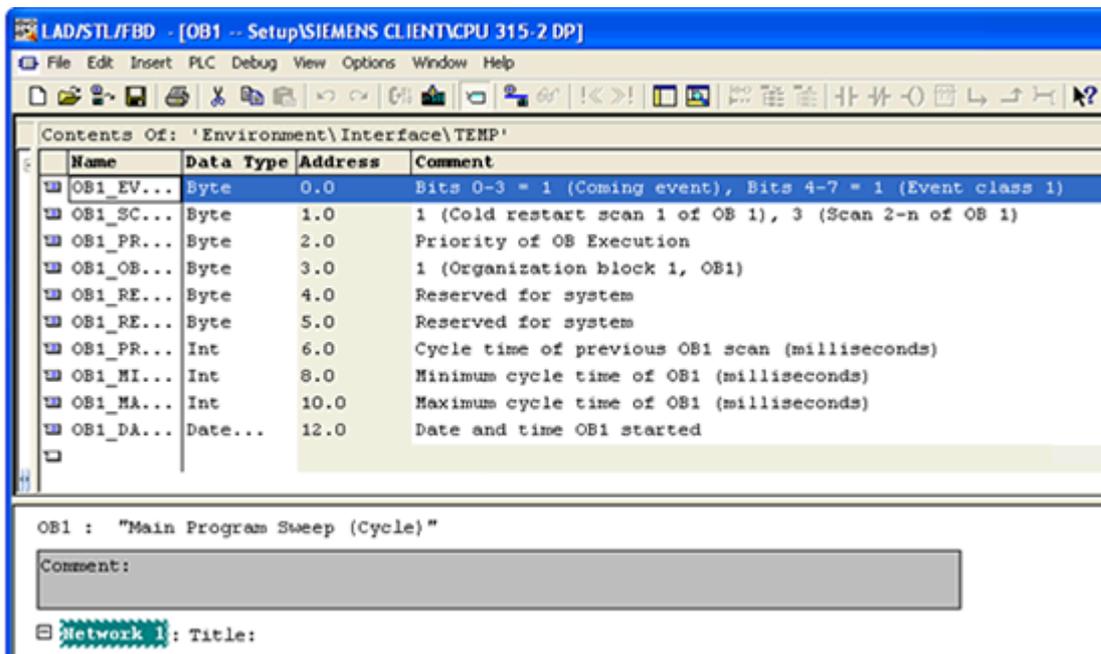
Step Four: Inserting Function Blocks

Once the Siemens client has been configured and connected with the Siemens server or unsolicited driver, it must also be prepared to generate requests for the unsolicited partner. This is done by creating function blocks, which can be used to read data from or write data to an unsolicited driver. The function block (FB) used for reading data in this example is FB14 (GET). The function block (FB) for writing data is FB15 (PUT).

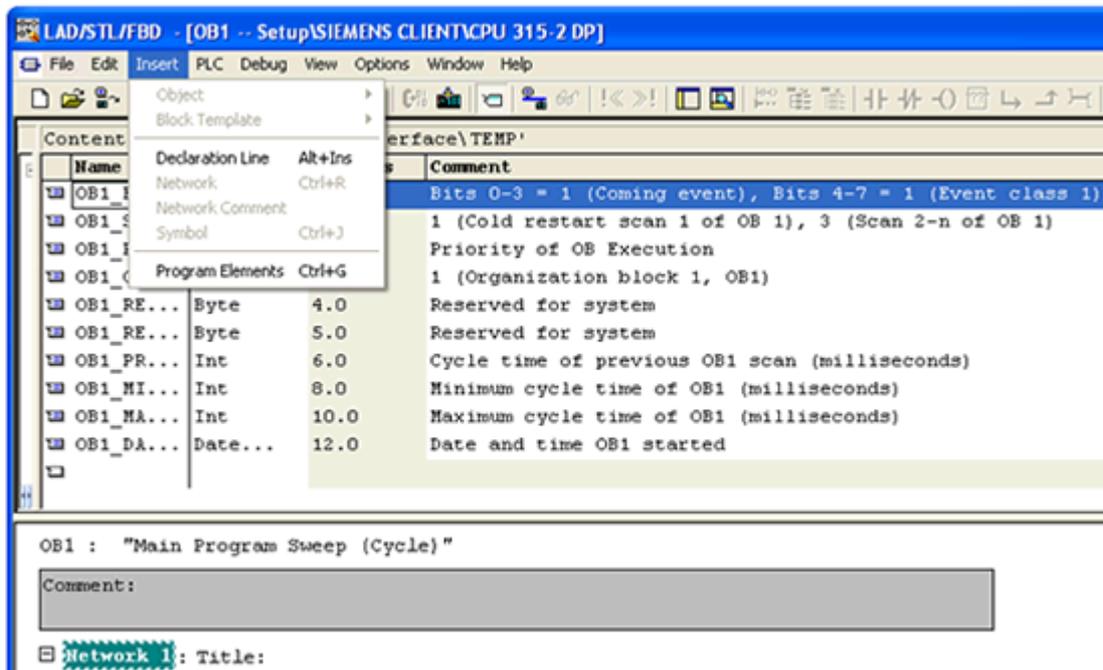
1. Expand the Siemens client menu, the **CPU 315-2 DP** menu, and the **S7 Program[1]** menu.
2. Double-click on **Blocks** and **OB1**.



3. LAD, STL, or FBD can be used to create function blocks. In this example, FBD is used. In the LAD/STL/FBD window, click on the **Insert** menu.

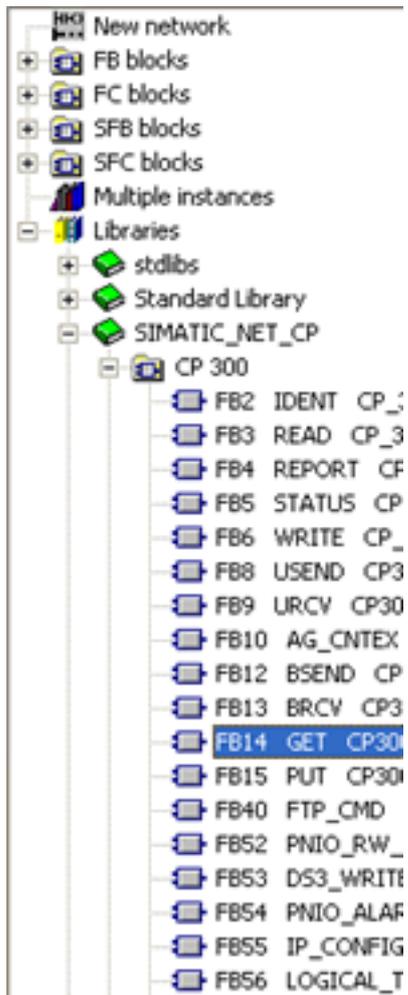


4. Click **Program Elements**.



5. Expand the **Libraries**, **SIMATIC_NET_CP**, and **CP 300** menus.

6. Double-click on **FB14 GET** to insert a function block to read data.



7. Close the **Program Elements** window. "FB14" should be inserted as shown below.

The screenshot shows the SIMATIC Manager interface. The top part displays a table of OB1 parameters:

Name	Data Type	Address	Comment
OB1_EV...	Byte	0.0	Bits 0-3 = 1 (Coming event), Bits 4-7 = 1 (Event class 1)
OB1_SC...	Byte	1.0	1 (Cold restart scan 1 of OB 1), 3 (Scan 2-n of OB 1)
OB1_PR...	Byte	2.0	Priority of OB Execution
OB1_OB...	Byte	3.0	1 (Organization block 1, OB1)
OB1_RE...	Byte	4.0	Reserved for system
OB1_RE...	Byte	5.0	Reserved for system
OB1_PR...	Int	6.0	Cycle time of previous OB1 scan (milliseconds)
OB1_MI...	Int	8.0	Minimum cycle time of OB1 (milliseconds)
OB1_MA...	Int	10.0	Maximum cycle time of OB1 (milliseconds)
OB1_DA...	Date...	12.0	Date and time OB1 started

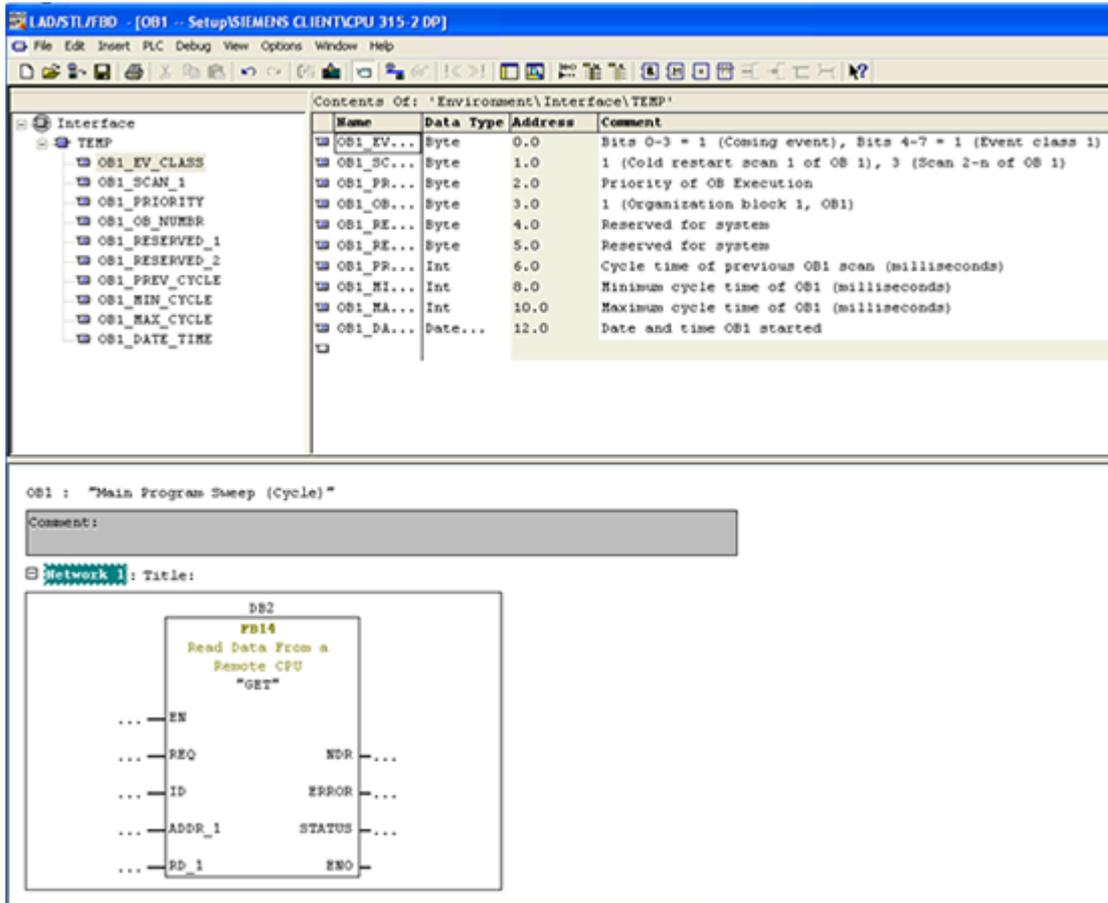
Below the table, the OB1 configuration is shown as "OB1 : 'Main Program Sweep (Cycle)'" with a comment field. The network editor shows a function block "GET" with the following connections:

- EN: ...
- REQ: ...
- ID: ...
- ADDR_1: ...
- RD_1: ...
- NDR: ...
- ERROR: ...
- STATUS: ...
- ENO: ...

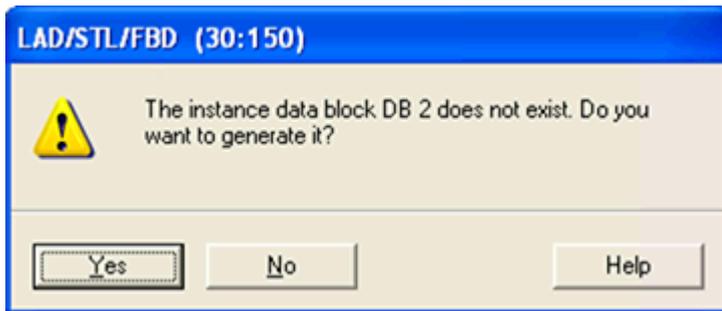
Three red question marks are positioned above the "GET" block, indicating the need to associate a data block (DB) with the function block (FB).

8. Associate a data block (DB) with the function block (FB). To do so, click above the FB where there are three red question marks.

9. Enter the name of a data block. In this example, it is "DB2".



10. Click **Yes** to create the data block.



11. Fill in the other details as appropriate for the fields in the function block. Users should consider the following:

- "ADDR_1" is the address on the destination device in the unsolicited driver.
- "RD_1" is the address local to the PLC.
- The value at the remote address specified by "ADDR_1" is written (GET) to the local address specified by "RD_1".

- Enter the Local ID number that was generated when setting up the connection between the Siemens client and the Siemens server driver in the ID field. In this example, the Local ID number is 1.
- The number of bytes in both the "ADDR_1" and "RD_1" fields should be same for the unsolicited driver to respond correctly. Otherwise, an error occurs.

The screenshot shows the SIMATIC Manager interface. The top window displays the 'Contents Of: 'Environment\Interface\TEMP'' table:

Name	Data Type	Address	Comment
OB1_EV_CLASS	Byte	0.0	Bits 0-3 = 1 (Coming event), Bits 4-7 = 1 (Event class)
OB1_SCAN_1	Byte	1.0	1 (Cold restart scan 1 of OB 1), 3 (Scan 2-n of OB 1)
OB1_PRIORITY	Byte	2.0	Priority of OB Execution
OB1_OB_NUMBR	Byte	3.0	1 (Organization block 1, OB1)
OB1_RESERVED_1	Byte	4.0	Reserved for system
OB1_RESERVED_2	Byte	5.0	Reserved for system
OB1_PREV_CYCLE	Int	6.0	Cycle time of previous OB1 scan (milliseconds)
OB1_MIN_CYCLE	Int	8.0	Minimum cycle time of OB1 (milliseconds)
OB1_MAX_CYCLE	Int	10.0	Maximum cycle time of OB1 (milliseconds)
OB1_DATE_TIME	Date...	12.0	Date and time OB1 started

The bottom window shows the 'OB1 : "Main Program Sweep (Cycle)"' block. The 'Network' title is 'Title:'. The ladder logic diagram shows the following connections:

- MO.0 (EN) connected to the EN input of FB14.
- MO.1 (REQ) connected to the REQ input of FB14.
- MW16#1 (ID) connected to the ID input of FB14.
- PI 0.0 (ADDR_1) connected to the ADDR_1 input of FB14.
- PI 0.0 (RD_1) connected to the RD_1 input of FB14.
- Q0.2 (MO.2) connected to the NO output of FB14.
- Q0.3 (MO.3) connected to the ERROR output of FB14.
- Q0.1 (MO.1) connected to the STATUS output of FB14.
- Q0.0 (ENO) connected to the ENO output of FB14.

• **Note:** Now that the GET function block has been created successfully, users must remember that the block gets executed / triggered only on a rising edge (REQ).

12. Click **Save** and close the **LAD/STL/FBD** window.

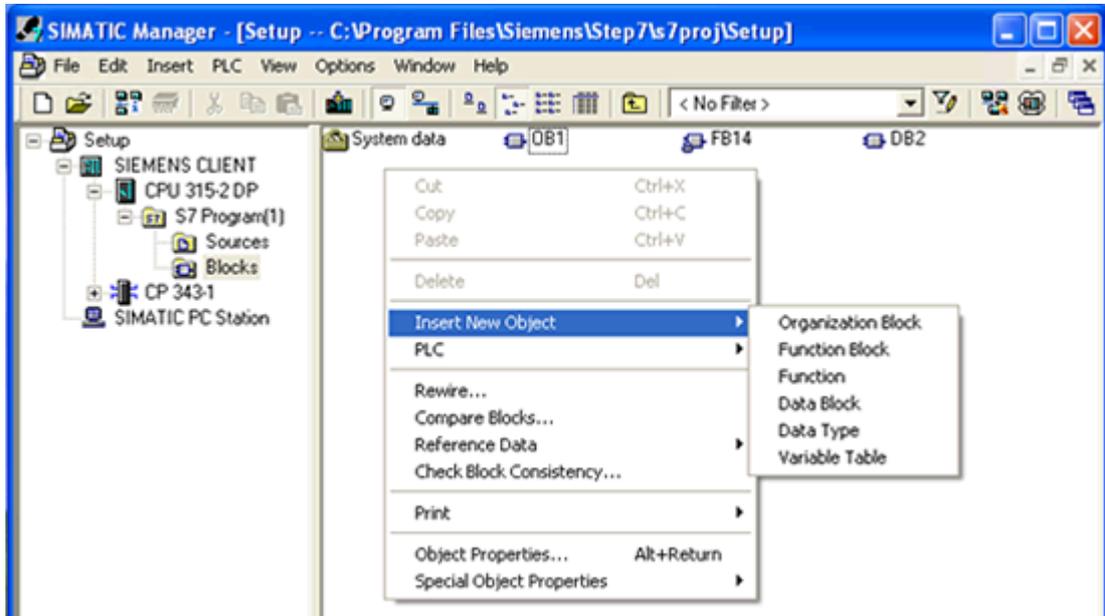
The screenshot shows the SIMATIC Manager interface. The project tree on the left is expanded to show the 'Setup' folder, which contains 'SIEMENS CLIENT', 'CPU 315-2 DP', 'S7 Program(1)', 'Sources', and 'Blocks'. The 'Blocks' folder is expanded, showing 'DB1', 'FB14', and 'DB2'.

• For more information, refer to [Step Five: Creating the DB3 Data Block](#).

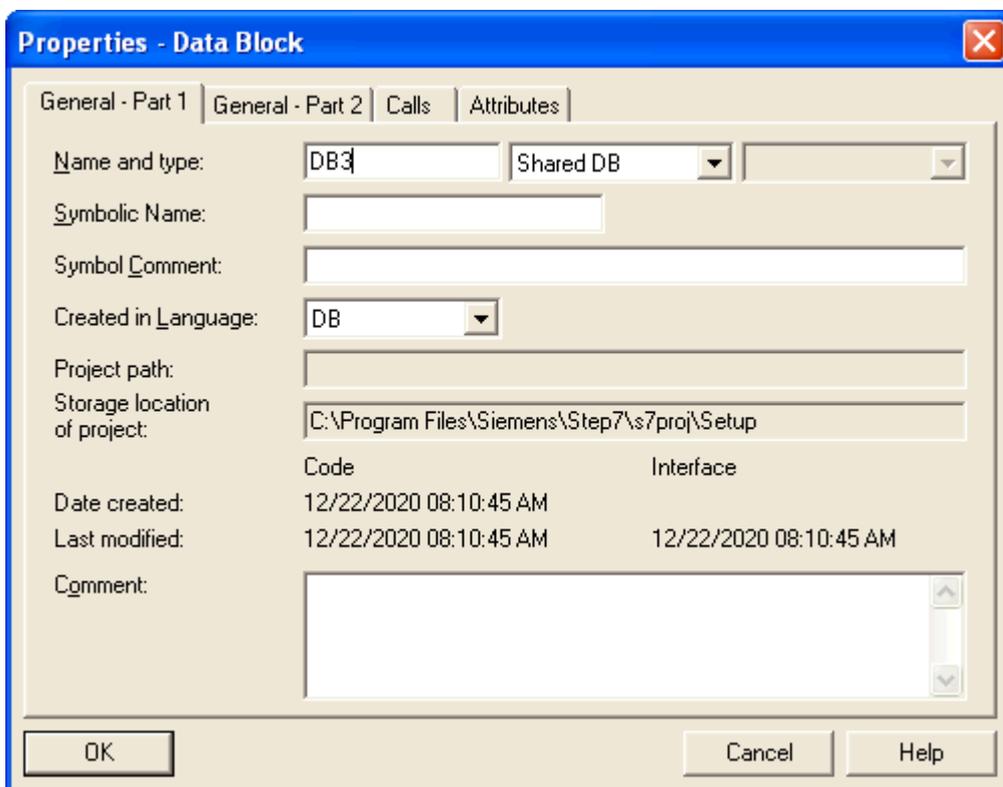
Step Five: Creating the DB3 Data Block

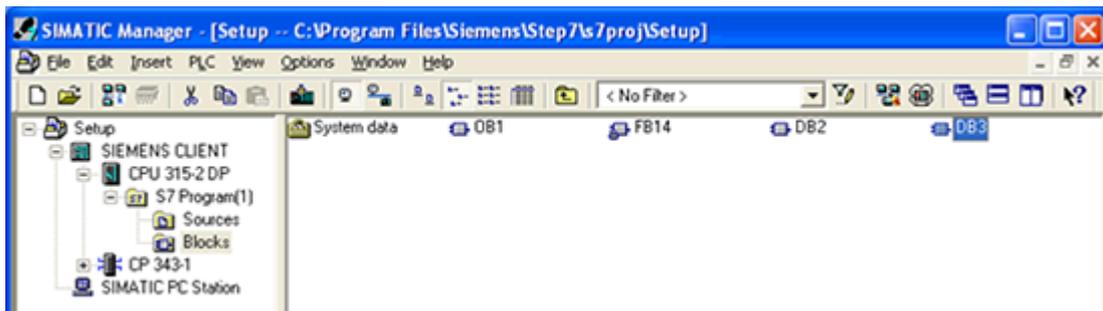
While configuring GET FB, the data block "DB3" was used for the "RD_1" field. This is the data block that stores read values.

1. Right-click in the right pane of the SIMATIC Manager window and then select **Insert New Object | Data Block**.

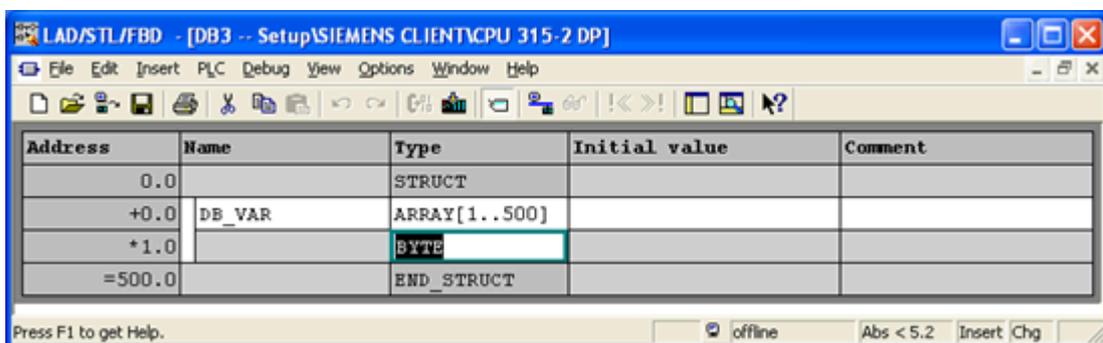


2. Change the name to "DB3."





3. Double-click on **DB3**. To assign some memory to the data block, users can make changes similar to those shown below. Although the array size in this example is arbitrary, values should be specified to fit a particular need.

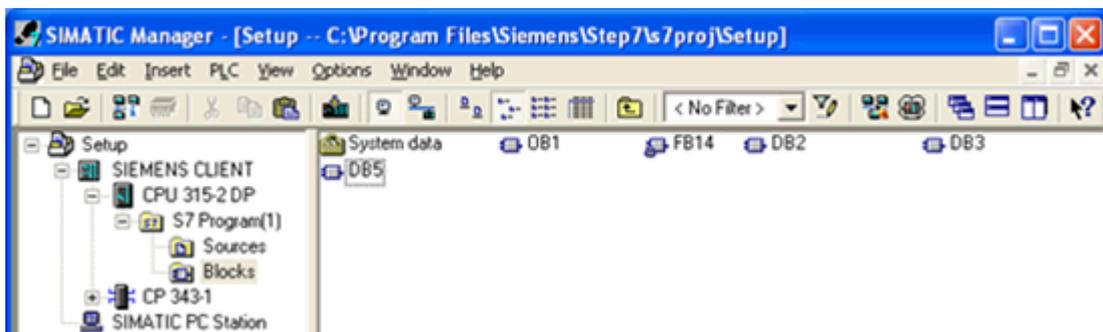


4. Save and close the **LAD/STL/FBD** window.

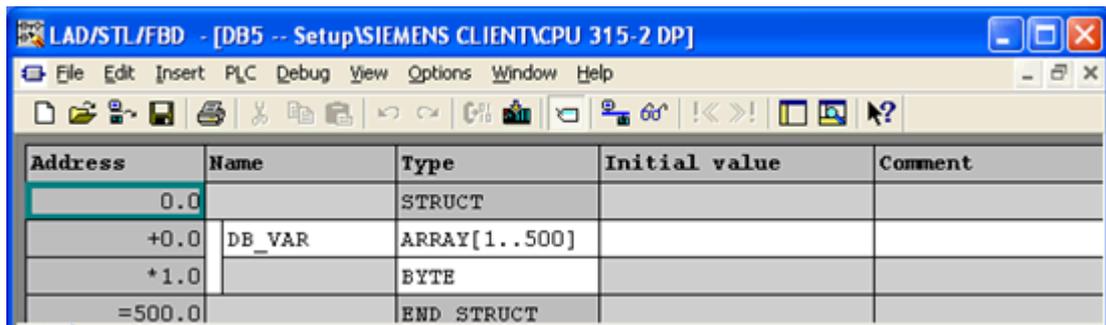
For more information, refer to [Step Six: Inserting PUT FB](#)

Step Six: Inserting PUT FB

1. Create a separate data block for the PUT FB, which holds the data that is written to the remote partner. To insert this new data block, follow the steps in [Step Five: Creating the DB3 Data Block](#) but name it "DB5."



2. Double-click on **DB5**, then specify a memory size. Although the array size in this example was chosen arbitrarily, the values should be specified to fit a particular need.



The screenshot shows the SIMATIC Manager LAD/STL/FBD editor window. The title bar reads "LAD/STL/FBD - [DB5 -- Setup\SIEMENS CLIENT\CPU 315-2 DP]". The menu bar includes File, Edit, Insert, PLC, Debug, View, Options, Window, and Help. The toolbar contains various icons for file operations and editing. Below the toolbar is a table with the following data:

Address	Name	Type	Initial value	Comment
0.0		STRUCT		
+0.0	DB_VAR	ARRAY[1..500]		
+1.0		BYTE		
=500.0		END STRUCT		

3. To insert the PUT FB, double-click on **OB1** in the SIMATIC Manager window.
4. In **LAD/STL/FBD**, right-click in the blank space below **GET FB**.

- Click **Insert Network** and select the blank space below.

The screenshot shows the SIMATIC Manager interface. At the top, the title bar reads "LAD/STL/FBD - [OB1 -- Setup\SIEMENS CLIENT\CPU 315-2 DP]". Below the menu bar, the "Contents Of: 'Environment\Interface\TEMP'" window displays a table of OB1 parameters:

Name	Data Type	Address	Comment
OB1_EV_CLASS	Byte	0.0	Bits 0-3 = 1 (Coming event), Bits 4-7 = 1 (Event class 1)
OB1_SCAN_1	Byte	1.0	1 (Cold restart scan 1 of OB 1), 3 (Scan 2-n of OB 1)
OB1_PRIORITY	Byte	2.0	Priority of OB Execution
OB1_OB_NUMBR	Byte	3.0	1 (Organization block 1, OB1)
OB1_RESERVED_1	Byte	4.0	Reserved for system
OB1_RESERVED_2	Byte	5.0	Reserved for system
OB1_PREV_CYCLE	Int	6.0	Cycle time of previous OB1 scan (milliseconds)
OB1_MIN_CYCLE	Int	8.0	Minimum cycle time of OB1 (milliseconds)
OB1_MAX_CYCLE	Int	10.0	Maximum cycle time of OB1 (milliseconds)
OB1_DATE_TIME	Date...	12.0	Date and time OB1 started

Below the table, the OB1 configuration is shown as "OB1 : 'Main Program Sweep (Cycle)'" with a "Comment:" field. Underneath, "Network 1" is displayed with the title "Network 1 : Title:". The network diagram features a function block "FB14 Read Data From a Remote CPU 'GET'" within a data block "DB2". The inputs and outputs are as follows:

- Inputs: M0.0 (EN), M0.1 (REQ), W#16#1 (ID), P#I 0.0 (ADDR_1), P#DB3. (RD_1), DEX0.0 (RD_1).
- Outputs: NDR (M0.2), ERROR (M0.3), STATUS (M0.1), ENO.

Below Network 1, "Network 2" is shown with the title "Network 2 : Title:" and a large empty rectangular area for editing.

- Click **Insert | Program Elements**.
- Expand the **Libraries, SIMATIC_NET_CP, and CP 300** menus.
- To insert a function block to write data, double-click on **FB15 PUT**.

- Close the **Program Elements** window.

The screenshot shows the Siemens STEP 7 LAD editor interface. The top window displays the 'Contents Of: 'Environment\Interface\TEMP'' table:

Name	Data Type	Address	Comment
OB1_EV_CLASS	Byte	0.0	Bits 0-3 = 1 (Coming event), Bits 4-7 = 1 (Event class 1)
OB1_SCAN_1	Byte	1.0	1 (Cold restart scan 1 of OB 1), 3 (Scan 2-n of OB 1)
OB1_PRIORITY	Byte	2.0	Priority of OB Execution
OB1_OB_NUMBR	Byte	3.0	1 (Organization block 1, OB1)
OB1_RESERVED_1	Byte	4.0	Reserved for system
OB1_RESERVED_2	Byte	5.0	Reserved for system
OB1_PREV_CYCLE	Int	6.0	Cycle time of previous OB1 scan (milliseconds)
OB1_MIN_CYCLE	Int	8.0	Minimum cycle time of OB1 (milliseconds)
OB1_MAX_CYCLE	Int	10.0	Maximum cycle time of OB1 (milliseconds)
OB1_DATE_TIME	Date...	12.0	Date and time OB1 started

Below the table, a ladder logic network is shown. It includes a function block call for 'FB15 Write Data to a Remote CPU "PUT"'. The network is titled 'Network 2: Title:'. The function block has several inputs and outputs:

- EN: Enable input
- REQ: Request input
- ID: Local ID input
- ADDR_1: Remote address input
- SD_1: Local address input
- DONE: Done output
- ERROR: Error output
- STATUS: Status output
- ENO: Enable output

Three red question marks are placed above the 'FB15' block, indicating that the data block (DB) has not yet been associated with the function block.

- Associate a data block (DB) with the function block (FB). To do so, click above the FB where there are three red question marks.
 - Specify a name. In this example, "DB4" is used.
 - A window prompt requests confirmation of data block creation. Click **Yes**.
 - Fill in the other details as appropriate. Users should consider the following:
 - "ADDR_1" address is on the destination device in the unsolicited driver.
 - "SD_1" is the address local to the PLC.
 - The value at the local address specified by "SD_1" is written (PUT) to the remote address specified by "ADDR_1".
 - Enter the Local ID number that was generated when setting up the connection between the Siemens client and the Siemens server driver in the **ID** field. In this example, the Local ID number is 1.
- Important:** The number of bytes in both the "ADDR_1" and "SD_1" fields should be same

for the unsolicited driver to respond correctly. Otherwise, an error occurs.

The screenshot shows the LAD/STL/FBD editor for a Siemens PLC. The top part displays a table of OB1 parameters:

Name	Data Type	Address	Comment
OB1_EV_CLASS	Byte	0.0	Bits 0-3 = 1 (Coming event), Bits 4-7 = 1 (Event class 1)
OB1_SCAN_1	Byte	1.0	1 (Cold restart scan 1 of OB 1), 3 (Scan 2-n of OB 1)
OB1_PRIORITY	Byte	2.0	Priority of OB Execution
OB1_OB_NUMBR	Byte	3.0	1 (Organization block 1, OB1)
OB1_RESERVED_1	Byte	4.0	Reserved for system
OB1_RESERVED_2	Byte	5.0	Reserved for system
OB1_PREV_CYCLE	Int	6.0	Cycle time of previous OB1 scan (milliseconds)
OB1_MIN_CYCLE	Int	8.0	Minimum cycle time of OB1 (milliseconds)
OB1_MAX_CYCLE	Int	10.0	Maximum cycle time of OB1 (milliseconds)
OB1_DATE_TIME	Date...	12.0	Date and time OB1 started

Below the table, two ladder logic networks are shown:

Network 1: A network with inputs `W#16#1` (ID), `P#I 0.0`, and `BYTE 10` (ADDR_1). It contains a network symbol with `NDR` (MO.2), `ERROR` (MO.3), and `STATUS` (M#1) outputs, and an `ENO` output.

Network 2: A network titled "Network 2: Title:" containing a function block `FB15` (Write Data to a Remote CPU "PUT") within a data block `DB4`. It has inputs `M0.0` (EN), `M0.1` (REQ), `W#16#1` (ID), `P#Q 0.0`, and `BYTE 10` (ADDR_1). It has outputs `DONE` (MO.4), `ERROR` (MO.5), and `STATUS` (M#2), and an `ENO` output.

14. Click **Save** and close LAD/STL/FBD.

The screenshot shows the SIMATIC Manager Setup window. The left pane shows the project structure:

- Setup
 - SIEMENS CLIENT
 - CPU 315-2 DP
 - S7 Program(1)
 - Sources
 - Blocks
 - CP 343-1
 - SIMATIC PC Station

The right pane shows the system data:

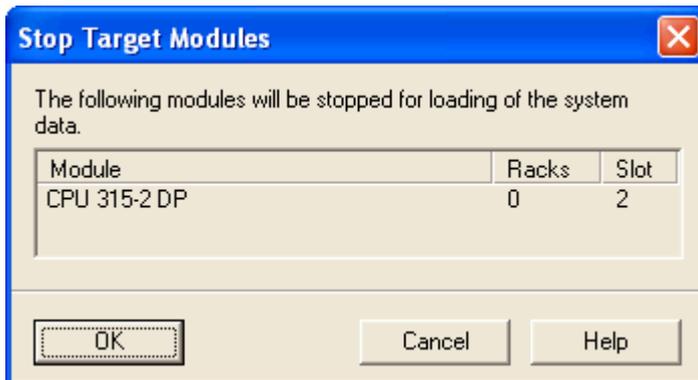
- System data
 - OB1
 - OB2
 - OB3
 - OB4
 - OB5
 - FB14
 - FB15
 - DB2
 - DB3
 - DB4
 - DB5

For more information, refer to [Step Seven: Downloading to the PLC](#).

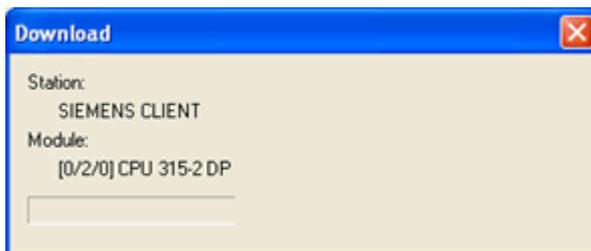
Step Seven: Downloading to the PLC

Once the Siemens client has been prepared to generate Read / Write requests for the remote unsolicited partner, the information must be downloaded to the PLC.

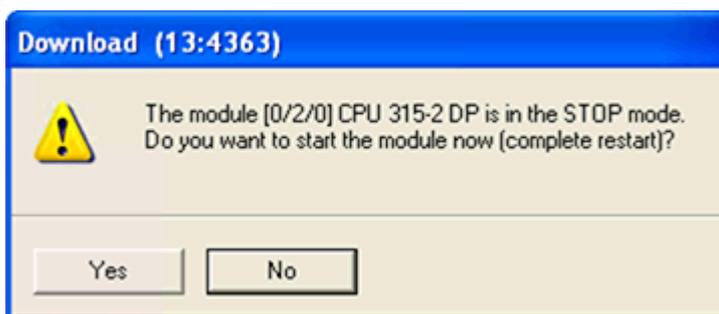
1. Click **Siemens client** in the left pane of the SIMATIC Manager window.
2. Select the **PLC** menu.
3. Select **Download** to begin downloading the project to the PLC.



4. Click **OK**.

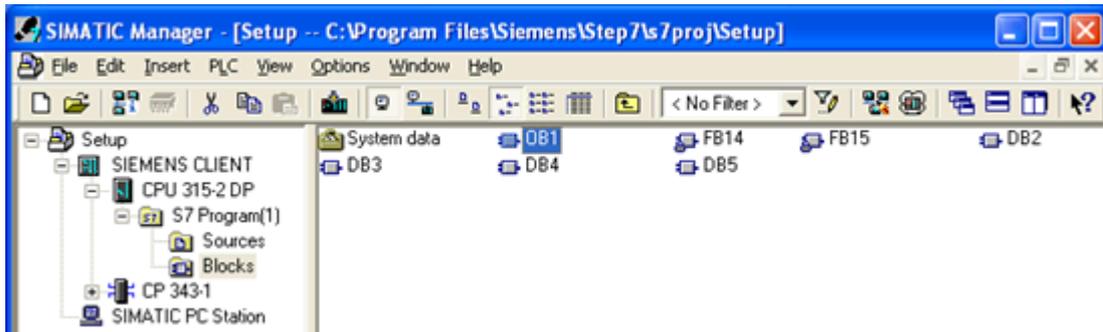


5. Click **Yes**.

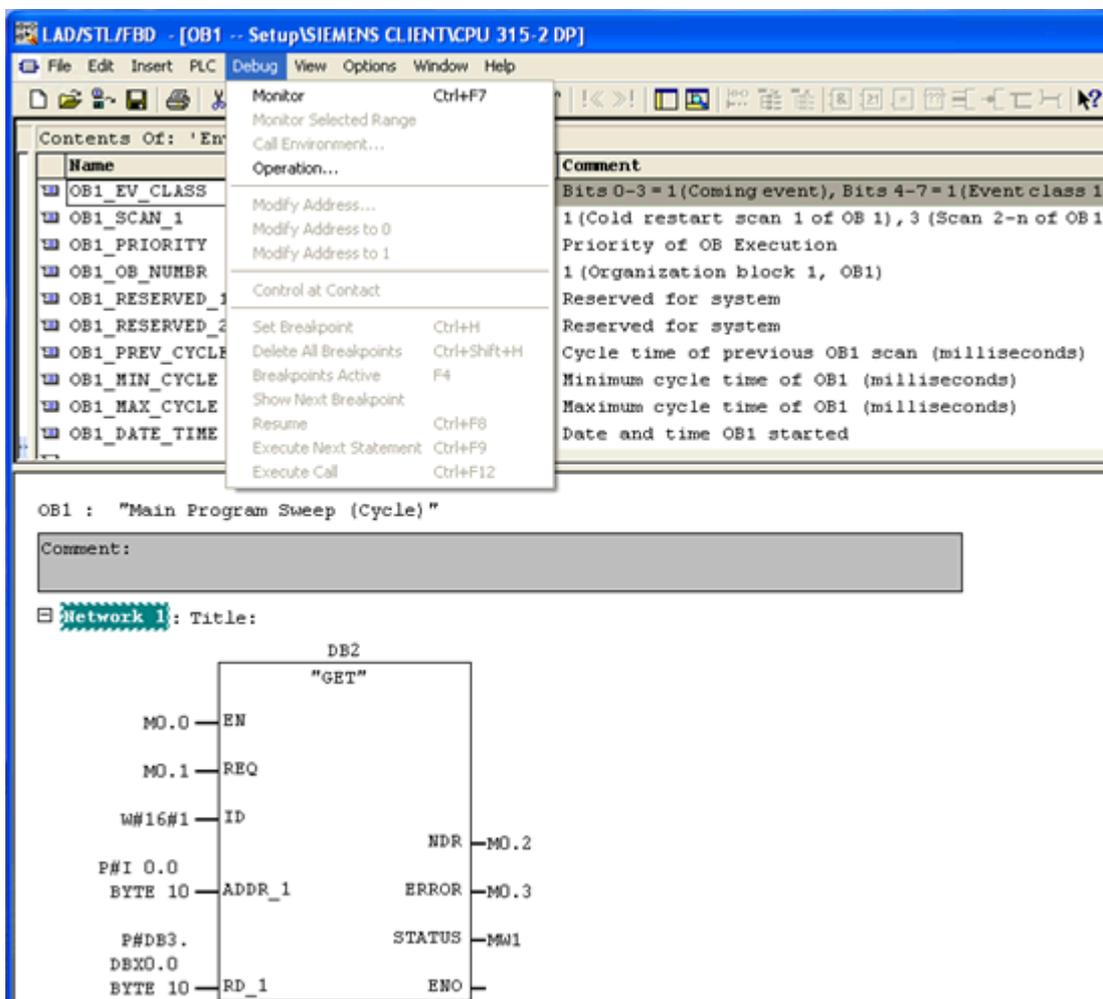


Note: The Siemens client must be run to trigger the function blocks that generate Read / Write requests.

6. Double-click on **OB1** in the SIMATIC Manager window.



7. In LAD/STL/FBD, click **Debug | Monitor**.



- **Note:** LAD/STL/FBD should appear in Online Mode.

LAD/STL/FBD - [OB1 -- SetupSIEMENS CLIENTCPU 315-2 DP ONLINE]

File Edit Insert PLC Debug View Options Window Help

Contents Of: 'Environment\Interface\TEMP'

Name	Data Type	Address	Comment
OB1_EV_CLASS	Byte	0.0	Bits 0-3 = 1 (Coming event), Bits 4-7 = 1 (Event class)
OB1_SCAN_1	Byte	1.0	1 (Cold restart scan 1 of OB 1), 3 (Scan 2-n of OB 1)
OB1_PRIORITY	Byte	2.0	Priority of OB Execution
OB1_OB_NUMBR	Byte	3.0	1 (Organization block 1, OB1)
OB1_RESERVED_1	Byte	4.0	Reserved for system
OB1_RESERVED_2	Byte	5.0	Reserved for system
OB1_PREV_CYCLE	Int	6.0	Cycle time of previous OB1 scan (milliseconds)
OB1_MIN_CYCLE	Int	8.0	Minimum cycle time of OB1 (milliseconds)
OB1_MAX_CYCLE	Int	10.0	Maximum cycle time of OB1 (milliseconds)
OB1_DATE_TIME	Date...	12.0	Date and time OB1 started

OB1 : "Main Program Sweep (Cycle)"

Comment:

Network 1: Title:

Network 2: Title:

- To execute **GET/PUT FBs**, change the **REQ** value to 0 and then 1 to indicate the rising edge. To do so, right-click on the **REQ** field and select **Modify to 0** to force a zero to the field.

The screenshot shows the Siemens LAD/STL/FBD editor interface. The top window displays the contents of the 'Environment\Interface\TEMP' directory, listing various OB1 parameters:

Name	Data Type	Address	Comment
OB1_EV_CLASS	Byte	0.0	Bits 0-3 = 1 (Coming event), Bits 4-7 = 1 (Event class 1)
OB1_SCAN_1	Byte	1.0	1 (Cold restart scan 1 of OB 1), 3 (Scan 2-n of OB 1)
OB1_PRIORITY	Byte	2.0	Priority of OB Execution
OB1_OB_NUMBR	Byte	3.0	1 (Organization block 1, OB1)
OB1_RESERVED_1	Byte	4.0	Reserved for system
OB1_RESERVED_2	Byte	5.0	Reserved for system
OB1_PREV_CYCLE	Int	6.0	Cycle time of previous OB1 scan (milliseconds)
OB1_MIN_CYCLE	Int	8.0	Minimum cycle time of OB1 (milliseconds)
OB1_MAX_CYCLE	Int	10.0	Maximum cycle time of OB1 (milliseconds)
OB1_DATE_TIME	Date...	12.0	Date and time OB1 started

Below the table, the OB1 configuration is shown as "Main Program Sweep (Cycle)". A comment field is present but empty.

The main workspace displays "Network 1" with a title field. A right-click context menu is open over the 'MO.1' field, with the 'Modify to 1' option selected. The network diagram includes a DB2 block labeled "GET" and various input/output fields like MO.0, MO.1, MO.2, MO.3, P#I 0.0, P#DB3, DBX0.0, and BYTE 1.

- Right-click on the **REQ** field and select **Modify to 1** to force a value of one to the field.

The screenshot shows the SIMATIC Manager interface with the following components:

- Contents Table:**

Name	Data Type	Address	Comment
OB1_EV_CLASS	Byte	0.0	Bits 0-3 = 1 (Coming event), Bits 4-7 = 1 (Event class 1)
OB1_SCAN_1	Byte	1.0	1 (Cold restart scan 1 of OB 1), 3 (Scan 2-n of OB 1)
OB1_PRIORITY	Byte	2.0	Priority of OB Execution
OB1_OB_NUMBR	Byte	3.0	1 (Organization block 1, OB1)
OB1_RESERVED_1	Byte	4.0	Reserved for system
OB1_RESERVED_2	Byte	5.0	Reserved for system
OB1_PREV_CYCLE	Int	6.0	Cycle time of previous OB1 scan (milliseconds)
OB1_MIN_CYCLE	Int	8.0	Minimum cycle time of OB1 (milliseconds)
OB1_MAX_CYCLE	Int	10.0	Maximum cycle time of OB1 (milliseconds)
OB1_DATE_TIME	Date...	12.0	Date and time OB1 started
- OB1 Description:**

OB1 : "Main Program Sweep (Cycle)"

Comment:
- Network 1:**

Network 1: Title:

The network diagram shows a DB2 block titled "GET" with the following connections:

 - MO.0 (EN) connected to the EN input of DB2.
 - MO.1 (REQ) connected to the REQ input of DB2.
 - DB2 outputs: NDR (MO.2), ROR (MO.3), TUS (MW1), and ENO.

Note: Both of the FBs must next be configured to respond to the same rising edge for the SIMATIC Manager's variables to be locally monitored and modified.

10. In **LAD/STL/FBD**, click on **PLC** and select **Monitor/Modify Variables**.

11. Enter the variables to be monitored. To view the changes made to this window, execute the function blocks.

Note: Remember that the slot / rack value of the remote device with which the Siemens client is communicating is "rack:0 slot:2". The values can be changed from the NetPro window. Users must make sure that the Siemens server or unsolicited driver on the other end has a device with these values and is running.

Index

A

Address Descriptions 13

Arrays 16

B

BCD 12, 15

Boolean 12

C

Channel Assignment 8

Channel Properties — Advanced 6

Channel Properties — Ethernet Communications 5

Channel Properties — General 4

Channel Properties — Write Optimizations 6

Communications Properties 7

CPU Settings 10

CPU Slot 10

D

Data Block Boolean 14

Data Collection 8

Data Types Description 12

Device Properties — General 7

Diagnostics 5

Discrete Inputs 13

Discrete Outputs 13

Do Not Scan, Demand Poll Only 10

Driver 8

Duty Cycle 6

DWord 12

E

Ethernet 4

Ethernet Settings 5

Event Log Messages 18

Examples 17

F

Failure to start unsolicited communications. | Port number = <number>. 18

Float 12, 15

G

General 7

H

Help Contents 3

I

ID 8

Identification 4, 8

Initial Updates from Cache 10

Inter-Device Delay 7

Internal Memory 14

Internal Tags 10

ISO 8073 Class 0 4

L

LBCD 12

Libraries 4

Long 12

M

Maximum PDU Size 10

Model 8

N

Name 8

Network Adapter 5

Non-Normalized Float Handling 7

O

Operating Mode 8

Optimization Method 6

Overview 3

P

Port Number 7

Protocols 4

R

Rack Number 10

Replace with Zero 7

Respect Tag-Specified Scan Rate 10

RFC1006 4

S

S5 Counter 15

S5 Timer 15

Scan Mode 9

Setup 4

Short 12

Siemens Client Device Configuration 10

Siemens S7-300 3

Signed Word 15
SIMATIC Manager 19
Simulated 9
Step Five: Creating the DB3 Data Block 47
Step Four: Inserting Function Blocks 41
Step One: Creating a New Project 19
Step Seven: Downloading to the PLC 53
Step Six: Inserting PUT FB 48
Step Three: Connecting the Siemens Client and the Siemens Server Driver 35
Step Two: Configuring the Siemens Client and PC Station 22
String 12, 15
Supported Commands 4

T

Tag Counts 5, 9

U

Unmodified 7
Unsigned Word 15

W

Word 12
Write All Values for All Tags 6
Write Only Latest Value for All Tags 6
Write Only Latest Value for Non-Boolean Tags 6