

---

## Driver and Protocol Objects

This chapter describes Lookout*Direct* Driver and protocol object classes, listed in alphabetical order. Input parameter syntax and data members are documented for each object class, along with a description of the functionality of each object class.

## AB\_PLC5, AB\_SLC500

---

Lookout*Direct* uses the AB object classes to communicate with the Allen-Bradley family of PLC controllers using a variety of interfaces.

Lookout*Direct* can communicate with a member of the PLC-2 family in the following ways:

- Through a Data Highway Plus (DH+) connection to an Allen-Bradley 1785-KA3 PLC-2 adapter module using an Allen-Bradley 1784-KT, 1784-KTx, or 1784-PCMK card, or an S-S Technologies 5136-SD direct-link interface card installed in the computer,
- Through the serial port using an Allen-Bradley KF2 module (which converts serial DF1 to DH+) to an Allen-Bradley 1785-KA3 PLC-2 adapter module, or
- Through a direct DF1 serial connection to the PLC programming port.

Lookout*Direct* can communicate with a member of the PLC-5 family in the following ways:

- Through a direct Ethernet connection to the PLC AUI port,
- Through a direct DH+ connection using a 1784-KT, 1784-KTx, 1784-PCMK, or 5136-SD card installed in the computer,
- Through the serial port via a KF2 module which converts serial DF1 to DH+, or
- Through a direct DF1 serial connection to the PLC programming port.

Lookout*Direct* can communicate with a member of the SLC-500 family in the following ways:

- Through a direct DH+ connection using a 1784-KT, 1784-KTx, 1784-PCMK, or 5136-SD card installed in the computer,
- Through the serial port via a KF2 module which converts serial DF1 to DH+,
- Through the serial port using an Allen-Bradley 1747-KE card (which plugs into the SLC chassis and converts DF1 to DH 485),
- Through the serial port using a stand-alone Allen-Bradley 1770-KF3 communication interface module which converts DF1 to DH 485,
- Through a direct DF1 serial connection to a SLC 5/03 or SLC 5/04 programming port, or

- Through a direct DH485 connection using a 1784-PCMK card in conjunction with either a 1747-AIC or 1761-NET-AIC module.

**PLC Address** refers to the PLC network node address setting as configured on the physical device. If devices share a common **Interface**, they require unique addresses. When using DF1 protocol (serial communications), valid addresses range from 0 to 254 decimal. When using DH+, valid addresses range from 0 to 77 octal.

**PLC Model** specifies the particular type of PLC or SLC you are representing with this object. The **PLC Model** you select determines what native data members comprise the object.

**PollRate** is a numeric expression that determines how often to poll the device. *LookoutDirect* polls the device at the specified time interval. Normally, this is a simple time constant such as 0:01 (one second). See *Numeric Data Members* in Chapter 2, *How LookoutDirect Works*, of the *Getting Started with LookoutDirect* manual for information on entering time constants.

**Poll** is a logical expression. When this expression changes from false to true, *LookoutDirect* polls the device. You can use a simple expression like the signal from a pushbutton, or a complex algorithm.

**Communication alarm priority** determines the priority level of the alarms generated by the AB object.

**Retry attempts** specifies the consecutive number of times *LookoutDirect* attempts to establish communications with a device if it is not getting a valid response. After it tries the number of **Retry attempts** specified, the object generates an alarm and begins to **Skip every  $n$  poll requests after comm failure**. Once *LookoutDirect* reestablishes communications, it polls the device on its regular cycle, as defined by **PollRate**.

**Receive timeout** is the time delay *LookoutDirect* waits for a response from a device before retrying the poll request.

**Interface** identifies the type of communication hardware you are using. The selection you make here determines what protocol parameters you have to specify. The paragraphs that follow describe interface-specific protocol parameters.

## Allen-Bradley Serial Port Interface Parameters

The `KE/KF/Serial` **Interface** selection enables serial port communication via a KE card, a KF3 module or KF2 module. When using your serial port, *LookoutDirect* employs the Allen-Bradley full-duplex (peer-to-peer) DF1 protocol. Figure shows an Allen Bradley object configured for serial communications.

**Serial port** specifies which RS-232C port the object uses for communication to the physical device.

**Data rate, Parity, and Error detection** reference the settings on the hardware device. The AB object classes support both *BCC* (block check character) and *CRC* (cyclic redundancy check) error detection. *BCC* provides a medium level of data security. *CRC* ensures a higher level of data security. Choose the settings as configured on your PLC or SLC.

**Phone number** specifies the number to be dialed if the serial port setting is configured for dial-up. This number only applies to the individual object.

## Allen-Bradley DH+ Interface Parameters



**Note** When you configure an AB\_PLC2, AB\_PLC5, or AB\_SLC500 for DH+, LookoutDirect creates a file called ALLBRAD.INI. This file contains the configuration settings that you enter for your KT card(s). If you plan to run the process file on a LookoutDirect Runtime System, be sure to copy the INI file along with your process (.lkp) file to the target computer.

The 1784-KT, 1784-KTx, 1784-PCMK, and S-S 5136-SD **Interface** selections enable direct connection of your computer to a DH+ network. The following illustration shows an AB\_SLC500 configured for DH+ communications using a 1784-KTx card.

**Card number** selects which network interface card that the PLC is connected to in case your computer has multiple KT or S-S cards.

**Card DH+ node address** identifies the address of the interface card in the DH+ network. Valid addresses range from 0 to 77 octal. The node address of the card must be unique—that is, it must not be the same as the address of any other device on the DH+ network.

**Memory address** specifies the base address location of the selected interface card memory. Your selection should match the settings on the card. If you are using multiple interface cards, be sure each card has a unique address. The network interface cards use dual-ported memory.

For this reason, if you are using a memory manager such as EMM386 it is important to add a memory exclusion statement to your `CONFIG.SYS` file. The table on the following page lists base memory address selections and corresponding exclusions for all legal memory addresses for the 1784-KT card.

Use **Max node address** to maximize the performance of a DH485 network by assigning addresses to nodes on the network using consecutive numbers starting with zero. Set the **Max node address** to the maximum of the assigned node addresses. By default, the value of this variable is 31, which is the largest legal address for any node on a DH485 network.

The 1784-KT interface card has on-board network termination resistors. If you are using such a card and if your computer is the last node on the network and if the cable does not already have a terminating resistor on it, then select the **Enable link termination resistor** check box.

Use the **Card exists in this computer** check box to instruct *LookoutDirect* whether or not to look for the interface card in the computer. *Be sure to check this box when you are ready to start polling your PLCs.* When you check this box and select **OK**, *LookoutDirect* initializes the card, activates its self-test, and downloads its driver firmware. Then polling begins. Leave the **Card exists in this computer** check box deselected (this is the default setting) if the card is not in your computer (for example, if you are developing a process on a computer different from the one that will be running the process) or if you do not want to poll any PLC connected to the card.

If you deselect the **Card exists in this computer** check box, you are disabling communications using this interface card with all PLCs connected to it.

**Table 3-1.** Allen-Bradley DH+ Interface Memory Addresses

Memory Address	AB 1784-KT Exclude	AB 1784-KTx 1784-PCMk Exclude	SS-5136-SD Exclude*	Recommendation
A000	A300-A3FF	A000-A0FF	A000-A3FF	Typically used by VGA drivers. Use if no other option is available.
A400	A700-A7FF	A400-A4FF	A400-A7FF	
A800	AB00-ABFF	A800-A8FF	A800-ABFF	
AC00	AF00-AFFF	AC00-ACFF	AC00-AFFF	
B000	B300-B3FF	B000-B0FF	B000-B3FF	Used by MDA & CGA drivers. Use if no Dxxx option is available.
B400	B700-B7FF	B400-B4FF	B400-B7FF	
B800	BB00-BBFF	B800-B8FF	B800-BBFF	
BC00	BF00-BFFF	BC00-BCFF	BC00-BFFF	
C000	C300-C3FF	C000-C0FF	C000-C3FF	Typically used by BIOS. Use if no Dxxx option is available.
C400	C700-C7FF	C400-C4FF	C400-C7FF	
C800	CB00-CBFF	C800-C8FF	C800-CBFF	
CC00	CF00-CFFF	CC00-CCFF	CC00-CFFF	
D000	D300-D3FF	D000-D0FF	D000-D3FF	Normally available. Try to use one of these first.
D400	D700-D7FF	D400-D4FF	D400-D7FF	
D800	DB00-DBFF	D800-D8FF	D800-DBFF	
DC00	DF00-DFFF	DC00-DCFF	DC00-DFFF	
* The 5136-SD memory exclusions recommended here are based on 16K memory mapping. Because you are using the SS card to emulate the KT card, there is no advantage to using its 32K memory access capability.				

**Baud rate** (1784-KTx, 1784-PCMk, 5136-SD only) selects the baud rate of the DH+ network. The default is 57.6k baud. Before selecting a higher baud rate, be aware of that only a few PLCs (such as the SLC5/04) support higher baud rates; that every node on a DH+ network must support the baud rate used on that network; that the maximum network cable length is smaller for higher baud rates; and that the correct values for the termination resistors at the ends of the network cable are different for higher baud rates. Consult the manuals that came with your hardware for more detailed information.

IRQ (all cards) identifies the interrupt setting of all DH+ interface cards installed in the computer. This selection should match the IRQ settings on *all* of the interface cards.

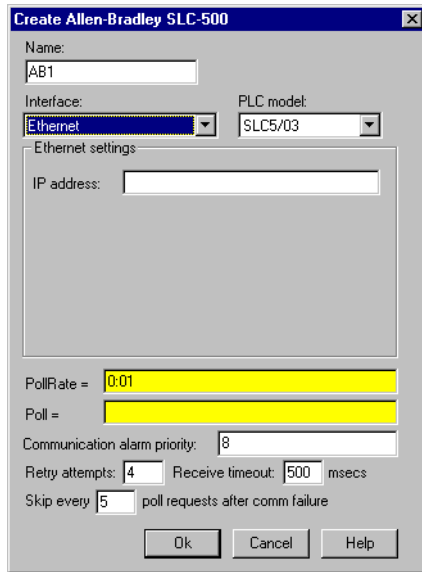
Assigning an interrupt to the interface card(s) improves overall computer performance somewhat. Any time one of the cards receives an input, it generates an interrupt recognized by LookoutDirect.



**Caution** Be sure to verify that no other drivers or cards are mapped to the selected memory address or use the same interrupt.

## Allen-Bradley Ethernet Interface Parameters

The Ethernet **Interface** selection enables direct communication between your computer and a PLC using a standard Ethernet network. The following diagram shows an AB\_PLC5 configured for Ethernet communications.



**IP address** specifies the Internet protocol address of the PLC. An Internet protocol address consists of four numbers, separated by periods. Each number ranges from zero to 255 decimal. Thus, a typical Internet address might be 128.7.9.231. Ensure that the **IP address** you enter matches the Internet protocol address of the PLC your object represents. You can also enter the IP address by name.



## Using the 5136-SD card from S-S Technologies, Inc.

To use the 5136-SD card, select the S-S 5136-SD interface in the **Create Object** dialog box. It is not necessary to run the `sdinst.exe` program that ships with the card because *LookoutDirect* downloads the KT-emulation module automatically as part of the initialization process. It is, however, necessary to tell *LookoutDirect* the port address specified by the switch settings on the card.

**Create Allen-Bradley PLC-5**

Name:  PLC Address (0 - 77):  octal

Interface: **S-S 5136-SD** PLC model: **PLC5/15**

1784-KT card settings

Card number:  1  2  3  4  5  6  7  8

Memory address:  Port:  Node address:  octal

Card exists in this computer Baud rate:

IRQ:  3  4  5  7  9  
(all cards)  10  11  12  15  none

PollRate =

Poll =

Communication alarm priority:

Retry attempts:  Receive timeout:  msec

Skip every  poll requests after comm failure

## Allen-Bradley Register Addressing

Lookout*Direct* has adopted a sequential, flat addressing scheme for the Allen-Bradley PLCs. The addresses are sequential by data type, having nothing to do with the actual slot number. For example, consider the following slot configuration:

PLC slot Location	Type of Module in PLC Slot	Channel Address in Lookout <i>Direct</i>
slot 1	16 channel analog input	I:0-I:16
slot 2	16 channel analog output	O:0-O:16
slot 3	8 channel input, 8 channel output	I:17-I:24, O:17-O:24
slot 4	16 channel analog input	I:25-I:32

You can see from the example, the slot number of the module is irrelevant to Lookout*Direct*, and the input channels follow consecutive numbers as the output channels follow their own consecutive numbers.

## Allen-Bradley Data Members

Each AB object contains a great deal of data. All readable and writable members (inputs/outputs) are bundled with the object. As soon as you create an AB object you immediately have access to all the object data members.

The AB object classes automatically generate an efficient read/write blocking scheme based on the inputs and outputs you are using in your process file. You are not required to build your own I/O blocking table. However, you can ensure peak performance by organizing your PLC data into contiguous groups.

**Table 3-2.** AB\_PLC2 Data Members

Data Member	Type	Read	Write	Description
0 - 7777	numeric	yes	yes	16-bit signed binary word ranging from -32,768 to +32,767
0_0 - 7777_17	logical	yes	no	One bit within a 16-bit binary word
CommFail	logical	yes	no	Object-generated signal that is ON if, for any reason, Lookout <i>Direct</i> cannot communicate with the PLC.

**Table 3-2.** AB\_PLC2 Data Members (Continued)

<b>Data Member</b>	<b>Type</b>	<b>Read</b>	<b>Write</b>	<b>Description</b>
OffHook	logical	no	yes	When true, instructs the PLC to retain exclusive use of its assigned communication port. This prevents <i>LookoutDirect</i> from hanging up between polls, saving the redial overhead. This also prevents other blocks from communicating over the same channel.
Poll	logical	no	yes	When transitioned from false to true, the <i>LookoutDirect</i> object polls the PLC device
PollRate	numeric	no	yes	Specifies the frequency at which the <i>LookoutDirect</i> object polls the PLC device
Update	logical	yes	no	Object-generated signal that pulses each time the object polls the device

**Table 3-3.** AB\_SLC500 Data Members

<b>Data Member</b>	<b>Type</b>	<b>Read</b>	<b>Write</b>	<b>Description</b>
B:0 - B255:255	numeric	yes	yes	16-bit signed binary word ranging from -32,768 to +32,767
B:0_0-B255:255_15	logical	yes	yes	One bit within a 16-bit binary word
B_0 - B255_4095	logical	yes	yes	One bit within the specified datafile. For example, B3_32 specifies datafile 3, word 2, bit 1.
C:0.ACC - C255:255.ACC	numeric	yes	yes	Counter accumulated value. Two-byte signed word ranging from -32,768 to +32,767
C:0.PRE - C255:255.PRE	numeric	yes	yes	Preset counter value. Two-byte signed word ranging from -32,768 to +32,767
C:0_CU - C255:255_CU	logical	yes	yes	Counter up-enable bit.
C:0_DN - C:255:255_DN	logical	yes	yes	Counter done bit.

**Table 3-3.** AB\_SLC500 Data Members (Continued)

Data Member	Type	Read	Write	Description
C:0_OV - C255:255_OV	logical	yes	yes	Counter overflow bit.
C:0_UA - C255:255_UA	logical	yes	yes	Counter update accumulation bit (HSC in fixed controller only)
C:0_UN - C255:255_UN	logical	yes	yes	Counter underflow bit.
C:0_CD - C255:255_CD	logical	yes	yes	Counter down-enable bit.
CommFail	logical	yes	no	Object-generated signal that is ON if, for any reason, <i>LookoutDirect</i> cannot communicate with the SLC.
F:0-F255:255	numeric	yes	yes	Floating point value
I:0 - I:30	numeric	yes	yes	Unsigned 16-bit input value ranging from 0 to 65,535
I:0_0 - I:30_15	logical	yes	yes	One bit within a 16-bit input word
N:0 - N255:255	numeric	yes	yes	16-bit signed integer value ranging from -32,768 to +32,767.
N:0_0 - N255:255_15	logical	yes	yes	One bit within a 16-bit signed integer word
O:0 - O:30	numeric	yes	yes	Unsigned 16-bit output value ranging from 0 to 65,535
O:0_0 - O:30_15	logical	yes	yes	One bit within a 16-bit output word
OffHook	logical	no	yes	When true, instructs the PLC to retain exclusive use of its assigned communication port. This prevents <i>LookoutDirect</i> from hanging up between polls, saving the redial overhead. This also prevents other blocks from communicating over the same channel.
Poll	logical	no	yes	When transitioned from false to true, the <i>LookoutDirect</i> object polls the SLC device

**Table 3-3.** AB\_SLC500 Data Members (Continued)

<b>Data Member</b>	<b>Type</b>	<b>Read</b>	<b>Write</b>	<b>Description</b>
PollRate	numeric	no	yes	Specifies the frequency at which the LookoutDirect object polls the SLC device
R:0_FD - R255:255_FD	logical	yes	yes	Control “found” single-bit logical indicator
R:0.LEN - R255:255.LEN	numeric	yes	yes	Control “length” signed integer ranging from –32,768 to +32,767
R:0.POS - R255:255.POS	numeric	yes	yes	Control “position” signed integer ranging from –32,768 to +32,767
R:0_DN - R255:255_DN	logical	yes	yes	Control “done” single-bit logical indicator
R:0_EM - R255:255_EM	logical	yes	yes	Control “empty” single-bit logical indicator
R:0_EN -R255:255_EN	logical	yes	yes	Control “enable” single-bit logical indicator
R:0_ER - R255:255.ER	logical	yes	yes	Control “error” single-bit logical indicator
R:0_EU - R255:255_EU	logical	yes	yes	Control “enable unloading” single-bit
R:0_IN - R255:255.IN	logical	yes	yes	Control “inhibit comparison” flag single-bit logical indicator
R:0_UL - R255:255_UL	logical	yes	yes	Control “unload” single-bit logical indicator
S:0 - S:96	numeric	yes	yes	SLC status file containing a signed integer ranging from –32,768 to +32,767 (see Allen-Bradley documentation)
S:0_0 -S:96_15	logical	yes	yes	Individual SLC status bits (see Allen-Bradley documentation)
ST9:0 - ST255:255	text	yes	yes	String, limited to 83 characters in length. See the note on Allen-Bradley string data members at the end of this table.

**Table 3-3.** AB\_SLC500 Data Members (Continued)

Data Member	Type	Read	Write	Description
T:0.ACC - T255:255.ACC	numeric	yes	yes	Accumulated timer value ranging from -32,768 to +32,767
T:0.PRE - T255:255.PRE	numeric	yes	yes	Preset timer value ranging from -32,768 to +32,767
T:0_DN - T255:255_DN	logical	yes	yes	Timer “done” single-bit logical indicator
T:0_EN - T255:255_EN	logical	yes	yes	Timer “enabled” single-bit logical indicator
T:0_TT - T255:255_TT	logical	yes	yes	Timer “timing” single-bit logical indicator
Update	logical	yes	no	Object-generated signal that pulses each time the object polls the device



**Note** There is no Allen-Bradley default file type associated with strings. You must configure your Allen-Bradley device to have a string file. See your Allen-Bradley documentation for details on this configuration procedure.

The string data member only works with the SLC 500 Enhanced series of PLCs, including the SLC 5/03; OS301 and SLC 5/04., and with the PLC 5 Enhanced series, PLC 5/11, PLC 5/20, PLC 5/30, PLC 5/40, PLC 5/60, PLC 5/80.

**Table 3-4.** AB\_PLCS Data Members

Data Member	Type	Read	Write	Description
B:0 - B999:999	numeric	yes	yes	16-bit signed binary word ranging from -32,768 to +32,767
B:0_0-B999:999_15	logical	yes	yes	One bit within a 16-bit binary word.
B_0 - For B999_15999	logical	yes	yes	One bit within the specified datafile; for example, B3_32 specifies datafile 3, word 2, bit 1.
C:0.ACC - C999:999.ACC	numeric	yes	yes	Counter accumulated value. Two-byte signed word ranging from -32,768 to +32,767
C:0.PRE - C999:999.PRE	numeric	yes	yes	Preset counter value ranging from -32,768 to +32,767.

**Table 3-4.** AB\_PLC5 Data Members (Continued)

<b>Data Member</b>	<b>Type</b>	<b>Read</b>	<b>Write</b>	<b>Description</b>
C:0_CD - C999:999_CD	logical	yes	yes	Counter down-enable bit.
C:0_CU - C999:999_CU	logical	yes	yes	Counter up-enable bit.
C:0_DN - C:999:999_DN	logical	yes	yes	Counter done bit.
C:0_OV - C999:999_OV	logical	yes	yes	Counter overflow bit.
C:0_UA - C999:999_UA	logical	yes	yes	Counter update accumulation bit (HSC in fixed controller only).
C:0_UN - C999:999_UN	logical	yes	yes	Counter underflow bit.
CommFail	logical	yes	no	Object-generated signal that is ON if, for whatever reason, <i>LookoutDirect</i> cannot communicate with the PLC.
F:0-F999:999	numeric	yes	yes	Floating point value.
I:0 - I:277	numeric	yes	yes	Unsigned 16-bit input value ranging from 0 to 65,535.
I:0_0 - I:277_17	logical	yes	yes	One bit within a 16-bit input word (octal).
N:0 - N999:999	numeric	yes	yes	16-bit signed integer value ranging from -32,768 to +32,767.
N:0_0 - N999:999_15	logical	yes	yes	One bit within a 16-bit signed integer word.
O:0 - O:277	numeric	yes	yes	Unsigned 16-bit output value ranging from 0 to 65,535.
O:0_0 - O:277_17	logical	yes	yes	One bit within a 16-bit output word (octal).

**Table 3-4.** AB\_PLC5 Data Members (Continued)

<b>Data Member</b>	<b>Type</b>	<b>Read</b>	<b>Write</b>	<b>Description</b>
OffHook	logical	no	yes	When true, instructs the PLC to retain exclusive use of its assigned communication port. This prevents LookoutDirect from hanging up between polls, saving the redial overhead. This also prevents other blocks from communicating over the same channel.
Poll	logical	no	yes	When transitioned from false to true, the LookoutDirect object polls the PLC device.
PollRate	numeric	no	yes	Specifies the frequency at which the LookoutDirect object polls the PLC device.
R:0.LEN - R999:999.LEN	numeric	yes	yes	Control “length” signed integer ranging from -32,768 to +32,767.
R:0.POS - R999:999.POS	numeric	yes	yes	Control “position” signed integer ranging from -32,768 to +32,767.
R:0_DN - R999:999_DN	logical	yes	yes	Control “done” single-bit logical indicator.
R:0_EM - R999:999_EM	logical	yes	yes	Control “empty” single-bit logical indicator.
R:0_EN - R999:999_EN	logical	yes	yes	Control “enable” single-bit logical indicator.
R:0_ER - R999:999.ER	logical	yes	yes	Control “error” single-bit logical indicator.
R:0_EU - R999:999_EU	logical	yes	yes	Control “enable unloading” single-bit logical indicator.
R:0_FD - R999:999_FD	logical	yes	yes	Control “found” single-bit logical indicator.
R:0_IN - R999:999.IN	logical	yes	yes	Control “inhibit comparison” flag logical indicator.



**Table 3-4.** AB\_PLC5 Data Members (Continued)

Data Member	Type	Read	Write	Description
R:0_UL - R999:999_UL	logical	yes	yes	Control “unload” single-bit logical indicator.
S:0 - S:127	numeric	yes	yes	PLC status file containing a signed integer ranging from –32,768 to +32,767 (see Allen-Bradley documentation).
S:0_0 - S:127_15	logical	yes	yes	Individual PLC status bits (see Allen-Bradley documentation).
ST9:0 - ST255:255	text	yes	yes	String, limited to 83 characters in length. See the note on Allen-Bradley string data members at the end of this table.
T:0.ACC - T999:999.ACC	numeric	yes	yes	Accumulated timer value ranging from –32,768 to +32,767.
T:0.PRE - T999:999.PRE	numeric	yes	yes	Preset timer value ranging from –32,768 to +32,767.
T:0_DN - T999:999_DN	logical	yes	yes	Timer “done” single-bit logical indicator.
T:0_EN - T999:999_EN	logical	yes	yes	Timer “enabled” single-bit logical indicator.
T:0_TT - T999:999_TT	logical	yes	yes	Timer “timing” single-bit logical indicator.
Update	logical	yes	no	Object-generated signal that pulses each time the object polls the PLC device.



**Note** There is no Allen-Bradley default file type associated with strings. You must configure your Allen-Bradley device to have a string file. See your Allen-Bradley documentation for details on this configuration procedure.

The string data member only works with the SLC 500 Enhanced series of PLCs, including the SLC 5/03; OS301 and SLC 5/04., and with the PLC 5 Enhanced series, PLC 5/11, PLC 5/20, PLC 5/30, PLC 5/40, PLC 5/60, PLC 5/80.

## Allen-Bradley Error Messages

AB objects report the statuses of commands they issue to AB devices. When *LookoutDirect* receives a response from an AB device, it reads the status (STS) byte and, if necessary, the extended status (EXT STS) byte to verify the device executed the *LookoutDirect* command properly. If the command was not executed properly, *LookoutDirect* reports the failure as an alarm containing the status code and its meaning. The following is an example of such an alarm:

```
EXT STS = 0F: not enough levels in address
```

AB object classes can also generate alarms internally. The following is a list of AB alarms generated by *LookoutDirect*, their descriptions, and possible responses. In the messages, *KT* is used to refer to any of the DH+ interface cards (1784-KT, 1784-KTx, 1784-PCMK, or 5136-SD) and *SS* is used to refer to the 5136-SD card.

### **Cannot resolve ip address: *address***

The AB object failed to find any node on the network that corresponds to the given IP address. Confirm that the IP address entered in the **Modify Object** dialog box is correct.

### **Cannot get session id from plc**

The AB object sent a message to the PLC requesting a TCP/IP session and failed to receive a satisfactory response.

### **Cannot communicate with device (code=dd)**

The AB object timed out while waiting for a response (via TCP/IP) from the PLC. If the code is 0, the object timed out while trying to establish the TCP/IP connection; if the code is 1, the object timed out while waiting for a session id from the PLC; if the code is 2, the object timed out while waiting for a response to a poll request. Confirm that the IP address of the PLC has been entered correctly and that the PLC is reachable over the TCP/IP network.

### **Download of file sdipds.ss1 failed**

*LookoutDirect* was unable to write the KT-emulation program file to the 5136-SD card physical memory. This could be due to either an invalid port or memory address or to a faulty or improperly seated card.

### **Invalid memory address for card: 0xAAAA**

The memory address specified for the card (for example, D400) is not valid for this model of card. The address must be a multiple of 0x0100 and lie in the range 0xA000 to 0xDF00. Moreover, the KT, KTx, PCMK, and

5136-SD cards support different sets of valid memory addresses. See the documentation that shipped with the card for details.

**Invalid node address for card: xx**

Node addresses must be between 0 and 63 decimal.

**Invalid port number for SS card: 0xPPP**

The port number specified for the 5136-SD card is invalid. See the documentation that shipped with the card for the list of valid port addresses.

**Invalid port or memory address**

LookoutDirect was unable to write to the 5136-SD card physical memory. This could be due to either an invalid port or memory address or to a faulty or improperly seated card.

**KT card failed to find resources**

**KT card receive mailbox is in an invalid state**

**KT card send mailbox is in an invalid state**

You will probably never see one of these alarms. If you do, call National Instruments and ask for technical support.

**KT card dual-ported memory test failed at location xxxx**

The interface card failed a memory test when it was first powered on. The memory test reads, writes and rereads the dual-ported memory to ensure memory access by the card. Verify that the card is configured for the memory address that you specified. Verify that your memory manager (like EMM386) excludes the appropriate portion of memory. Verify that your card is not trying to use the same memory location as another card. You may need to restart the card by calling up the AB object definition dialog box and selecting **OK**. If that does not work try rebooting the computer. Other causes can include memory conflicts, a bad interface card, or a misbehaving driver.

**KT card CTC test timed out**

**KT card CTC test failed with status code xx**

The interface card failed the Counter Timer Circuit test when it was first powered on. This test verifies proper functionality of the card timer and counter modes over all CTC channels. You may need to restart, reseal, or replace the interface card.

**KT card timed out while loading protocol code**

**KT card failed with status code *xx* while loading protocol**

Lookout*Direct* was not able to transfer a loader file to the card and subsequently download the card protocol firmware. Try to restart the interface card by calling up the AB object definition dialog box and selecting **OK**. If that does not work try rebooting the computer.

**KT card is no longer responding**

The Lookout*Direct* AB object did not receive the interface card heartbeat within the last second. Normally, the card generates a heartbeat any time it receives the DH+ network token. If the alarm does not deactivate after 30 seconds, try to restart the card by calling up the AB object definition dialog box and selecting **OK**. If that does not work try restarting Lookout*Direct* or rebooting the computer.

**KT card memory address conflicts with card *n***

Lookout*Direct* found another interface card with the same memory address. Be sure that the memory address on each interface card is different and that the corresponding **Memory address** in the Lookout*Direct* object matches the card address.

**KT card not present in this computer**

The **Card exists in this computer** check box is deselected (this is the default setting). Select **Object»Modify** to retrieve the PLC definition parameters dialog box and select the **Card exists in this computer** check box and **OK** to initialize the card.

**KT card RAM test timed out**

**KT card RAM test failed with status code *xx***

The interface card failed a memory test when it was first powered on. The memory test writes a pattern to on-board RAM and reads its content to verify the card memory is working. Confirm that the memory address specified in the **Object»Modify** dialog box is correct. You may need to restart, reseal, or replace the card.

**KT card signature test failed**

The AB object does not recognize the card. Make sure that the interface card is actually installed in the PC and that you indicated the correct memory address in Lookout*Direct*. You may need to reseal the card, or the card may require repair. Also ensure that you did not identify more **Card Numbers** than actual physical cards.

**KT card SI0 test timed out****KT card SI0 test failed with status code xx**

The interface card failed the Serial Input Output test when it was first powered on. You may need to restart, reseal, or replace the interface card.

**KT card send mailbox timed out**

The AB object timed out while waiting for the KT card to signal that it is ready to be given a new message to send. This is most likely due to a communication problem between the computer and the PLC. Confirm that the network cable is properly installed and that the PLC is turned on.

**NAK response received**

The AB object received a NAK (not acknowledged) response when it polled the device. The device received a command from *LookoutDirect* but it did not accept the message. The command that the device received may be incomplete or contain irregularities due to poor network performance. If your Serial Port is configured for radio, you may need to adjust the **RTS delay off** setting. Also consider increasing the number of **Retry attempts**.

**No response — no ACK for our transmission**

*LookoutDirect* is not getting any response from the device. This could be caused by just about anything. Verify that the **Data rate**, **Parity**, and **Error detection** settings are the same as the settings on the device. Make sure you are using the proper **Serial port** on your computer. Verify that the device interface module and other network equipment is connected and working properly. If you are using a modem, verify that your object **Phone number** and the serial port **Dial-up** settings are correct. This may also be caused by low level noise or reflections on the highway, or marginal circuitry on a card.

### **No response within timeout period**

#### **No response received after receiving ENQ**

The AB object received an acknowledgment of its poll from the device. The device accepted the command from *LookoutDirect*. However, the device did not appear to send anything else back in response. You may have to increase **Receive timeout** to make sure *LookoutDirect* allows enough time to receive the message.

### **EOT response received**

The AB object received an EOT (end of transmission) response when it polled the device indicating that the device did not have a message ready to give in response to the *LookoutDirect* poll request. It is unlikely that you will ever see this error message.

### **Received TSN does not match**

#### **Response message garbled -- bad CRC or bad BCC**

#### **Response message garbled -- no DLE EXT**

#### **Response message garbled -- bad DLE follower**

The AB object is receiving messages from the device. However, the messages may be failing the selected data integrity test. Verify that the object **Error detection** setting is the same as the settings on the device. Another cause may be that the last part of the message is actually getting clipped off before it is completed. You may have to increase the **Receive gap** Serial Port setting to ensure *LookoutDirect* is receiving the entire message. If your Serial Port is configured for radio this could be caused by an audible squelch tail occurring at the end of a radio transmission. You may need to adjust the **RTS delay off** or the **CTS timeout** settings. Also consider increasing the number of **Retry attempts**.

### **Socket communications error *dd: msg***

The AB object has encountered a problem while attempting to communicate using TCP/IP. The error number *dd* and corresponding error message *msg* give further information. Confirm that the IP address of the PLC has been entered correctly and that the PLC is reachable over the TCP/IP network.

### **SS card failed**

This message is suffixed with an error message read from the card itself. You may need to contact the vendor of your card for technical support.

### **SS card failed while performing diagnostics**

*LookoutDirect* successfully wrote the KT-emulation program to the 5136-SD card, but the program failed to terminate. Try running the

`sdinst.exe` program that ships with the 5136-SD card, using the `CHK` option to confirm that the card is working properly.

#### **Unable to access physical memory at segment 0xAAAA**

*LookoutDirect* was unable to access the memory at the given segment address. The memory may already be in use by the operating system or by another application. Either change the object memory address (which may involve changing switch settings on the card itself) or, if you are using a memory manager, make sure that it is excluding the correct portion of memory.

#### **Unable to open port 0xPPP for SS card**

*LookoutDirect* was unable to open the port number specified for the 5136-SD card. Make sure that you have specified the port that is selected by the jumper settings on the card. Make sure that port and the following two ports (for example, 250, 251, and 252) are not in use by any other devices in your computer.

#### **Unexpected data response length**

#### **Unexpected response**

#### **Unhandled error**

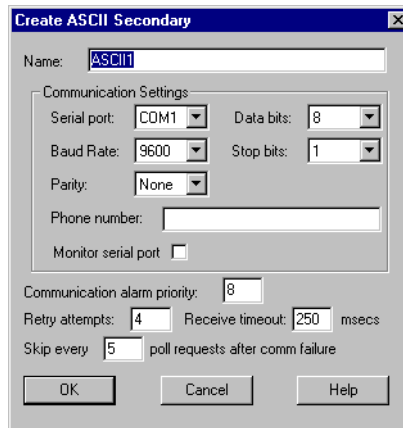
You will probably never see one of these alarms. If you do, call National Instruments and ask for technical support.

# ASCII

---

ASCII is a protocol driver class Lookout*Direct* uses to communicate with any serial device that accepts ASCII characters. This object is only available with 32-bit versions of Lookout*Direct*.

An ASCII object contains no predefined data points. When you create an ASCII object, you must define your data request strings as well as the template Lookout*Direct* uses to parse the response frame.



**Serial port** specifies which COM port the object uses for communicating to the external device. This does not specify the communication type. Communication type is determined by the **Options»Serial Ports...** command.

**Baud rate** indicates the rate that Lookout*Direct* uses to communicate with the hardware device.

**Data bits** indicates the number of data bits that Lookout*Direct* uses to communicate with the hardware. This setting should match the selection made on the physical device.

**Stop bits** indicates the number of stop bits that Lookout*Direct* uses to communicate with the hardware device. This setting should match the selection made on the physical device.

**Parity** indicates the parity that Lookout*Direct* uses to communicate with the hardware device. This setting should match the selection made on the physical device.



**Phone number** specifies the number to be dialed if the selected serial port is configured for dial-up. This number only applies to the individual protocol object.

**Monitor Serial Port** specifies whether you can receive unsolicited frames.

**Communication alarm priority** determines the priority level of alarms generated by the ASCII object. Such alarms are typically related to communications with the physical device.

**Retry attempts** specifies the number of times LookoutDirect attempts to establish communications with a device if it is not getting a valid response. After it tries the number of **Retry attempts** specified, the ASCII object generates an alarm and releases the communication port back to the communications service which then moves on to the next device in the polling queue (if any). Refer to Chapter 3, *Serial Communications*, in the *LookoutDirect Developer's Manual* for more information.

**Receive timeout** is the amount of time LookoutDirect waits for a response from a device before retrying the request.

The **Skip every N poll requests after comm failure** setting instructs LookoutDirect not to poll a device it has lost communication with on every scheduled poll. Instead, LookoutDirect polls the device only once in the specified number of poll cycles. Once communication has been reestablished, the device is polled on its regular cycle.

## ASCII Data Members

**Table 3-5.** ASCII Data Members

Data Member	Type	Read	Write	Description
CommFail	logical	yes	no	Object-generated signal that is on if LookoutDirect cannot communicate with the device(s).
OffHook	logical	no	yes	Keeps the driver from releasing the serial port.
Request	text	yes	no	Exact request frame sent.
RequestFormat	text	no	yes	Format used to create request frame.
Response	text	yes	no	Exact response frame received.
ResponseFormat	text	no	yes	Format used to parse response frame.

**Table 3-5.** ASCII Data Members (Continued)

Data Member	Type	Read	Write	Description
RQSum:1:1 - RQSum255:255	numeric	yes	no	Request byte sum
RQV1, RQV512	numeric	no	yes	Variable list used to populate request frame with numeric values.
RQV1.logical, RQV512.logical	logical	no	yes	Variable list used to populate request frame with logical values.
RQV1.txt, RQV512.txt	text	no	yes	Variable list used to populate request frame with text values.
RSFilter	text	no	yes	All characters in this string will be filtered out of the incoming response before processing.
RSSum1:1 - RSSum255:255	numeric	yes	no	Response byte sum
RSV1, RSV512	numeric	yes	no	Variable list used to store values retrieved from response frame.
RSV1.logical, RSV512.logical	logical	yes	no	Variable list used to store values retrieved from response frame.
RSV1.txt, RSV512.txt	text	yes	no	Variable list used to store values retrieved from response frame.
Send	logical	no	yes	Sends request frame.
Update	logical	yes	no	Object-generated signal that pulses low each time it polls the device.
RSVn, RSVn.txt and RSVn.logical all represent the same value in different forms RQVn, RQVn.txt and RQVn.logical all represent the same value in different forms				

## Request and Response Format Strings

The request and response format strings consist of static characters and markers that control how the request and response frames respectively are formatted or decoded. The request format string is used to *create* the request frame, which is sent to the device, while the response format string is used to *decode* the response frame, which comes from the device.

Static characters in the format strings are reproduced exactly in the request or response frame. Markers specify the location within the frame and type

of data which should be found there, such as five characters read as an unsigned integer, for example. The ASCII object constructs a request frame by processing the sequence of static characters and markers in the request format string, and including data from RQV data members.

The response format string decodes a response frame using an analogous process, storing the results in RSV data members.

To construct a request frame, the ASCII object parses the request format string character by character. Static characters are copied directly to the request frame. When a marker is encountered the ASCII object reads a value from the appropriate RQV variable and places it into the request frame.

There are 512 RQV and RSV values provided for in the ASCII object data member collection. The first marker in a format string uses the value from RQV1 (or RQV1.txt or RQV1.logical), the next marker uses the value RQV2, and so on. Values taken from Response strings are stored in RSV data members in the same way.

Keep in mind that writing into RQV1 changes the value both for RQV1.txt and RQV1.logical. Their only difference is the format in which they are represented. The same principle applies to the RSV data members.



**Note** There is no precedence to the order in which multiple objects connected to the same variable number initialize upon opening the process file. Consider, for example, the case in which a Pot object is connected to RQV1 while a TextEntry object is connected to RQV1.txt. You should take care to initialize such variables to the proper value after opening a process file.

To decode a response frame, the ASCII object compares the response frame to the response format string character by character. The static characters in the response frame must match those in the response format string or the decoding process terminates. Static characters are, in effect, discarded by the ASCII object as they are matched between the response format string and the response frame.

When the ASCII object encounters a marker, it places the data indicated by the marker into the appropriate RSV data member.

The conversion of a portion of the response frame to a data type specified by a marker in the response format string must be valid, or the process will terminate.

If nothing halts the process, decoding terminates when the end of the response frame string is reached.

There are examples of both request frames and response frames at the end of this section, but for the examples to make sense, you must first understand the ASCII object markers.

## ASCII Object Markers

The general format for a marker is:

%[width][type]

Each field in the marker format is a single character or a number signifying a particular format option.

The % sign denotes the beginning of the marker. For example, to specify that a percent-sign character is a static character part of the frame, use %%.

**Width** is a positive decimal integer specifying the number of characters that particular value occupies in the frame. By default ASCII pads the value with blank spaces if the value takes up fewer characters than the value specified by width. Including a 0 before the width value forces the ASCII object to pad with zeroes instead of blank spaces.

**Type** determines whether the field is interpreted as a character, a string, or a number.

**Table 3-6.** Data Types Allowed by ASCII

Character	Data Type
c	Character
d	Decimal integer
O, o	Octal
x, X	Hexadecimal integer
u	Unsigned decimal integer
e, f	Floating-point
s	String

**Table 3-6.** Data Types Allowed by ASCII

b*	Byte (binary)
*For the %b data type: <ul style="list-style-type: none"> <li>– Number of bytes can be specified, for example %3b, %2b.</li> <li>– Response format can read as either signed or unsigned, for example %^b and %3^b are signed and %b is unsigned.</li> <li>– Endian order can be specified, for example %3~b is big endian and %3b is little endian. %5~^b and %2^~b forms are also valid.</li> </ul>	

The simplest format specification contains only the percent sign and a type character (for example, %s). That would place the value in the response frame in the RSV1.txt data member.

Request Format String	RQV1	Request Frame
>%5d	34	> 34
>%05d	34	>00034

The request format string also has a precision value in the form **%[width].[precision][type]**. This specifies the number of digits to the right of the decimal point, if any, in the request frame. If you use a float (%f) and do not specify a precision value, the ASCII object assumes a default of 6.

Characters are converted and stored in RSV data members from response frames in the order they are encountered in the response format. However, fewer than **[width]** characters may be read if a white-space character (space, tab, or newline) or a character that cannot be converted according to the given format occurs before **[width]** is reached.

Values needed for request frames come from the RQV data members, and are also used in the order in which they occur in the request format.

To read strings not delimited by space characters, or that contain spaces, you can substitute a set of characters in brackets ([ ] s (string) type

character. The corresponding input field is read up to the first character that does not appear in the bracketed character set. Using a caret (^) as the first character in the set reverses this effect: the ASCII object reads input field up to the first character that does appear in the rest of the character set.

Response Format String	RSV1.txt	Response Frame
<code> %[A-Z,a-z, ]\$</code>	Natl Inst	<code> \$Natl Inst\$</code>
<code> &gt;%^[^,s]</code>	days	<code> &gt;day</code>

Notice that  `%[a-z]` and  `%[z-a]` are interpreted as equivalent to  `%[abcde...z]`, and that the character set is case sensitive. Valid control characters accepted include  `\a, \A, \b, \B, \f, \F, \n, \N, \r, \R, \t, \T, \v, \V, \l, \', \", and \?`.

Any ASCII character can be specified in the format string using  `\xbb` or  `\nnn` masks.  `\xbb` is a hexadecimal byte with each  `b` representing a valid hex character in the range (0...9, a...F), for example  `\xff` or  `\x1a`.

`\nnn` is an octal byte with each  `n` being a valid octal character in the range 0 to 7, for example  `\123` or  `\347`. This value may not exceed 255 as it is meant to represent a single byte of data. These two features can be used to create and compare static characters. These can also be specified in the regular expression, for example  `%[\xff, \123, \a, \b]` is a valid frame.



**Note** The brackets only work in response format strings. They have no effect in the request format string.

The ASCII object scans each field in the response frame character by character. It may stop reading a particular field before it reaches a character for a variety of reasons:

- The specified width has been reached.
- The next character cannot be converted as specified.
- The next character conflicts with a character in the response format string that it is supposed to match.
- The next character fails to appear in a given character set.

No matter what the reason, when the ASCII object stops reading a field, the next field is considered to begin at the first unread character. The conflicting character, if there is one, is considered unread and is the first character of the next field.

## Entering ASCII Object Format String

For a static connection to one of the format data members, enter your format string in the yellow field box in the Edit Connections dialog box. Remember to begin and end the format strings with quotation marks so that LookoutDirect accepts the string input.

You can also connect any valid text data member, such as a text entry object, to the format data members.

## Request Frame Construction Examples

Request Format String	RQV	Request Frame
<01%4u%s	RQV1=1234 RQV2.txt=Steph	<011234Steph
<01%04u%s	RQV1=34 RQV2.txt=Steph	<010034Steph
<01% 4u%s	RQV1=34 RQV2.txt=Steph	<01 34Steph

A zero in front of the four pads with zeroes; a space pads with spaces.

## Response Format Examples

Response Frame	Response Format String	RSV
*(16.38:	*(%52f:	RSV1=16.38



**Note** The decimal point counts as a character when decoding floats (%f). Also, decimal points denoting precision are not allowed when decoding a float in the response frame.

Response Frame	Response Format String	RSV
>>Test Text<<	>>%s<<	RSV1.txt=Test

The space between the words terminates the conversion. See the preceding bracketed character example in order to span a space or other special characters.

Response Frame	Response Format String	RSV
>>Test Text<<	>>%s%s<<	RSV1.txt=Test RSV2.txt=Text
>>DogCat<<	>>%3s%3s<<	RSV1.txt=Dog RSV2.txt=Cat

The response format uses a space as a delimiter.

## Using Sum Data Members

The ASCII object includes summing data members you can use to calculate checksum characters. This can be a checksum you want to write into an outgoing request frame or a checksum you want to verify in an incoming response frame.

For example, if you want to calculate a checksum for the request A00B, you would use an RQSum (request sum) data member. In the case of A00B, you would use RQSum1 : 4, which would give you a sum of the ASCII byte values of characters 1 through 4. Once you have this sum, you can manipulate it mathematically any way necessary for the checksum value you need. You can then insert this value at the end of your frame as a byte (%b) or a series of bytes.

The same technique works in reverse for RSSum (response sum) data members.

For example, consider the response z00A@. You know that you are expecting 4 bytes plus a checksum. Assuming that this checksum calculation involves the first four characters, use RSSum1 : 4 to get the byte sum of characters 1 through 4. After performing the appropriate mathematical manipulation, you can compare this value with the actual byte read from the frame, and determine when there is a checksum failure.



**Note** There are many different methods for calculating checksums, and these data members cannot support all of them. Before attempting to use them for checksum calculation, make sure your checksum can be calculated from a simple byte sum of characters in the frame.



## ASCII Error Messages

### **No response from device within timeout period**

Lookout*Direct* received no response from the device within the **Receive timeout** period. The ASCII object was able to establish a socket, but when it sent a message to the device, the device did not respond—as if it were not there. You may have to significantly increase **Receive timeout** (and **Poll Rate**) to ensure Lookout*Direct* is allowing enough time to receive the expected response. Also, verify your cable connections, power, configuration settings, and IP settings.

### **Not enough data to send a valid frame**

This means that the ASCII object has not received enough data to fill in all the variables in the Request Format frame. This could mean that you do not have connections made to all of the RQVs that the ASCII object is expecting.

### **Frame Error (garbled)**

ASCII got a response frame, but static characters in the response did not match up to the response format string.

### **Data type or length does not match format string**

ASCII got a response frame, but certain characters in the response were not in the format stated by the markers in the response format string.

### **Illegal control character**

The request frame had a % modifier followed by an invalid control code.

**Related Objects** *IPASCII*



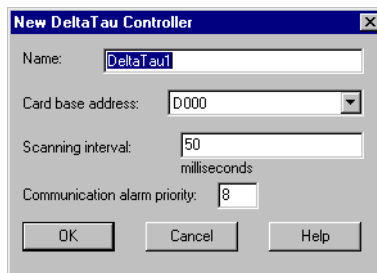
# DeltaTau

---

DeltaTau is a protocol driver object class Lookout*Direct* uses to communicate with Delta Tau Data Systems PMAC Motion Controllers. Create a DeltaTau object for each card installed in the computer.

This object class communicates with Delta Tau PMAC cards through dual-ported memory, so be sure that your PMAC hardware includes the dual-ported RAM option.

The following figure shows a Delta Tau card configured to use PC memory beginning at address D000.



**Card base address** specifies the beginning memory location of the dual ported RAM address. It should match card settings.

**Scanning Interval** identifies the frequency that the DeltaTau object in Lookout*Direct* polls the PMAC Motion Controller. Intervals can range from 10 ms to 1,000 ms.

**Communication alarm priority** specifies the priority level of alarms generated by the object.

## DeltaTau Data members

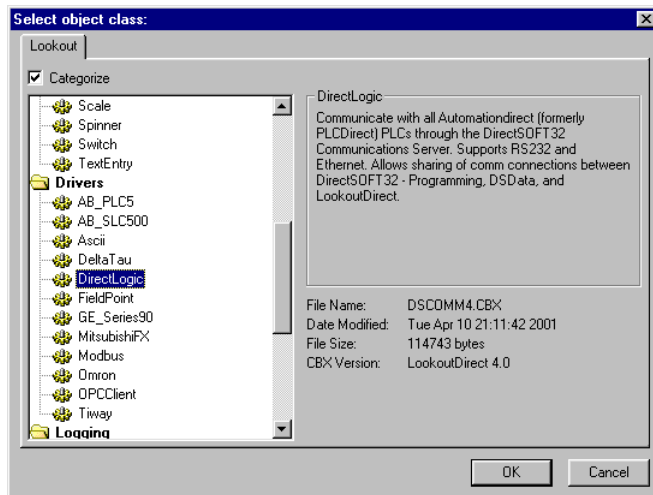
Like other protocol driver objects, DeltaTau objects contain a great deal of data. All readable and writable members (inputs/outputs), polling instructions, and so on, are bundled with the object. Therefore, as soon as you create a DeltaTau object you immediately have access to all the object data members. The following table lists data members for the DeltaTau object.

**Table 3-7.** DeltaTau Data Members

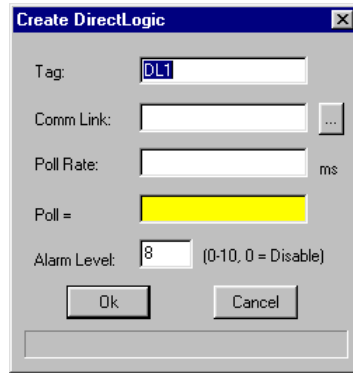
<b>Data Member</b>	<b>Type</b>	<b>Read</b>	<b>Write</b>	<b>Description</b>
DS0 – DS8190	numeric	yes	yes	Signed 32-bit word ranging from –2,147,483,648 to 2,147,483,647
DW0 – DW8190	numeric	yes	yes	32-Bit double-precision word ranging from 0 to 4,294,967,295
F0 – F8188	numeric	yes	yes	32-bit IEEE floating point word
S0 – S8190	numeric	yes	yes	Signed 16-bit word ranging from –32,768 to 32,767
Update	logical	yes	no	Driver-generated signal that pulses each time Lookout <i>Direct</i> scans the PMAC Motion Controller card
W0 – W8190	numeric	yes	yes	16-Bit word ranging from 0 to 65,535
W0.0 – W8190.15	logical	yes	yes	1 Bit in a 16-Bit word

# DirectLogic

The DirectLogic protocol driver object class is used to communicate to PLC Direct by Koyo 105, 205, 305, and 405 PLCs. This driver object supports additional Koyo private labeled PLCs under GE Series One, Texas Instruments Series 305, 405 and Siemens Simatic TI305, TI405 names. Please see the Protocol Reference Chart within the LookoutDirect Learning Guide to verify the part numbers of these additional Koyo PLCs. LookoutDirect supports both point-to-point, multidrop and Ethernet configurations. You can connect LookoutDirect to a fixed PLC Programming port, a networking DirectNET/CCM port, PLC DCM (data communications module), or to an Ethernet (ECOM) communications module.



After the DirectLogic driver class is selected the following dialog appears.



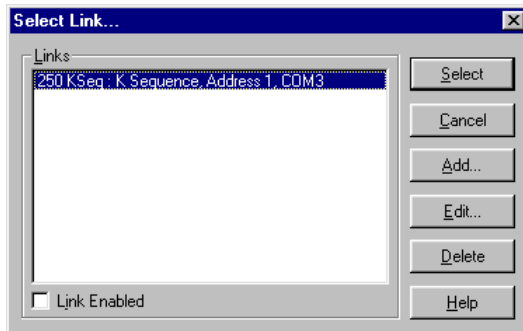
The selections are as follows:

**Tag:** User defined object name that can be referenced for connecting other objects.

**Comm Link:** The associated link chosen for the configured driver object. When the “...” button is pressed the **Select Link** dialog will appear.

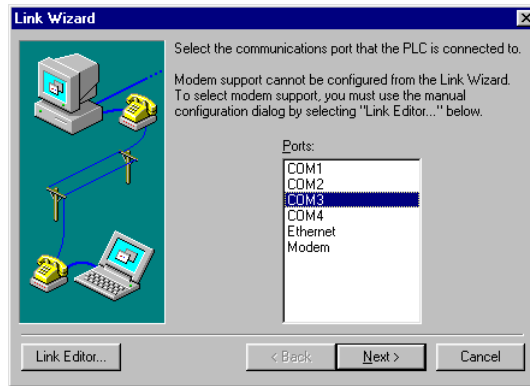
**Poll Rate=** is a numerical expression that determines how often to poll the device in milliseconds.

**Poll** is a logical expression. When the expression changes from false to true, LookoutDirect polls the device one time. You can use a simple expression like the signal from a pushbutton, or a complex algorithm.

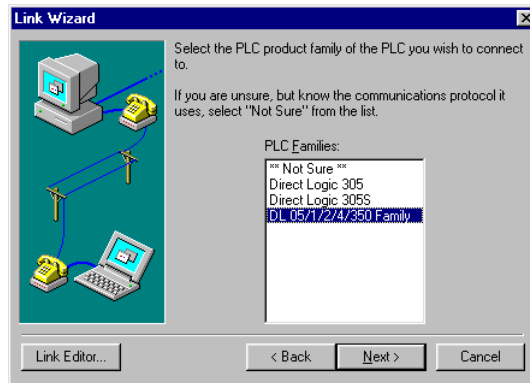


If a link is not available, you will have to add one. Click the **Add...** button to create a new communications link then select the correct Serial or Ethernet port that you have the cable attached from your PC to the PLC.

**Note** NOTE: If you are creating a serial Link that will connect through a modem, see the Modem Setup Guide Appendix C in the LookoutDirect Learning Guide. If you are creating an Ethernet Link, refer to the DirectSoft Communications Server online help.

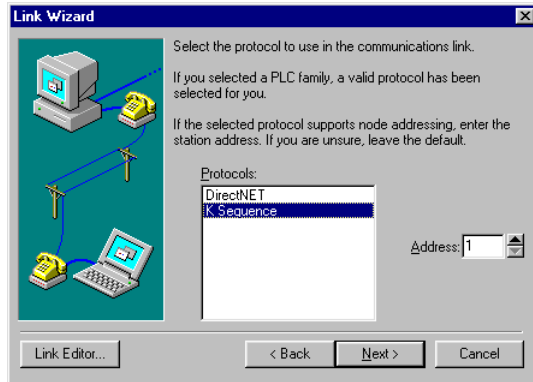


After making your choice click on the **Next** button.



Select the PLC family by single clicking on the appropriate choice. If you are unsure of the PLC family but know which communications protocol to use, you can select the “\*\*Not Sure\*\*” choice. If you are using a PLC Direct compatible PLC the LinkWizard can try to detect the model automatically.

Click on the Next> button when you are finished with your selection.



Select either the DirectNET or K-Sequence protocol. If during the previous step you selected one of the families listed, the highlight bar will be on a valid protocol for that family of PLC. The choice of protocol to use will depend on two factors:

- Whether or not the PLC you are connected to supports the protocol on the port where the cable is connected. For a listing of protocols available on each port of DirectLogic compatible PLCs see the Protocol Reference Chart in the Creating and Managing Communication Links chapter within the LookoutDirect Learning Guide. This chart is also available in the DirectSoft Communications Server Help that can be accessed by clicking on the Link Editor button or clicking on the Help button.
- If you need to perform write operations to individual Discrete I/O points or control relays, then you must select K-sequence protocol. DirectNET protocol cannot write to individual bit locations.

If the PLC has been configured to a node Address other than 1, enter that address now. Click on the **Next** button when you are finished.

The DirectLogic driver object will now attempt to establish a communication Link with the PLC using the node address and protocol you have selected. It will try the combination of 9600 Baud, and Odd Parity first. If this combination is unsuccessful, an 'auto-bauding' sequence will be used to try and determine the correct baud rate and parity combination. If these attempts are unsuccessful, the following dialog is displayed. You can click the Link Editor button and manually attempt to adjust the port



configuration, or you can consult the Appendix B Communications Troubleshooting Guide in the LookoutDirect Learning Guide.

## DirectLogic Protocol Status

The DirectLogic driver object monitors protocol statistics and the data can be viewed by selecting Options >> DirectSoft Link Status. This menu selection will only be visible in the Options menu if a DirectLogic driver object was previously created in your lookout process. Select the correct object from the list that appears, and then click **Select**. The **Link Info** dialog box appears.

The **Link Info** dialog box is useful for troubleshooting communication errors. Each error is time stamped and has an error name and an extended error description. At anytime you can press the edit button to manually change a setting on your link or to press Auto then Accept to clear the error log. The Link Enable box allows you to turn the communications link On and Off.

## 105/205/350/405 DirectLogic PLC Family Data Members

**Table 3-8.** 105/205/350/405 DirectLogic PLC Data Members

Data Member	Type	Read	Write	Description
Activated	Logical	Yes	No	Object-generated signal when TRUE, this flag signifies an active5 communication connection between the process file and the PLC.
C0 – C377	Logical	Yes	Yes	Control Relays – addressed in octal and mapped to V40600 – V40777.
CT0 – CT377	Logical	Yes	Yes	Counter status (done) bits – addressed in octal and mapped to V41140 – V41157
CTA0:B – CTA377:B	Numeric	Yes	Yes	Counter current value words BCD – addressed in octal and mapped to V01000 – V01377

**Table 3-8.** 105/205/350/405 DirectLogic PLC Data Members

CTA0:DB – CTA376:DB	Numeric	Yes	Yes	Counter current value double words (two adjacent addresses; 32-Bit) BCD – addressed in octal and mapped to V01000 – V01377
Failed	Logical	Yes	No	Object-generated signal when TRUE, this flag signifies the process file is no longer communicating with the PLC.
GX0 – GX3777	Logical	Yes	Yes	Remote I/O Inputs – addressed in octal and mapped to V40000 – V40177
GY0– GY3777	Logical	Yes	Yes	Remote I/O Outputs – addressed in octal and mapped to V40200 – V40277
Paused	Logical	Yes	No	Object-generated signal when TRUE, this flag signifies that the communication connection has paused but is still active and will go FALSE when its paused condition is satisfied. Usually caused by modifying the Link while it is in use.
S0 – S1777	Logical	Yes	Yes	Stage status (active) bits – addressed in octal and mapped to V41000 – V41077
SP0 – SP777	Logical	Yes	No	Special Relays (system status bits) – addressed in octal and mapped to V41200 – V41237
T0 – T0377	Logical	Yes	Yes	Timer status (done) bits – addressed in octal and mapped to V41100 – V41117
TA0:B – TA377:B	Numeric	Yes	Yes	Timer current value words BCD – addressed in octal and mapped to V00000 – V00377
TA0:DB – TA376:DB	Numeric	Yes	Yes	Timer current value double words (two adjacent addresses; 32-Bit) BCD – addressed in octal and mapped to V00000 – V00377
V0 – V41237	Numeric	Yes	Yes	Single (16-Bit) V-memory registers decimal

**Table 3-8.** 105/205/350/405 DirectLogic PLC Data Members

V0:D – V41237	Numeric	Yes	Yes	Double (two adjacent registers; 32-Bit) V-memory registers decimal
V0:B – V41237:B	Numeric	Yes	Yes	Single (16-Bit) V-memory registers BCD 0- 9999
V0:DB – V41236:DB	Numeric	Yes	Yes	Double (two adjacent registers; 32-Bit) V-memory registers BCD 0- 99999999
V0:R - V41236:R	Numeric	Yes	Yes	Double word V-memory registers signed real (IEEE 32-Bit Floating Point)
V0:S - V41237:S	Numeric	Yes	Yes	Single (16-Bit) V-memory registers signed decimal ranging from -32768 to 32767
VC0 – VC3760	Numeric	Yes	Yes	Single (16-Bit) V-memory word registers decimal Aliases for mapped Control Relays C0 – C3777
VC0:B – VC3760:B	Numeric	Yes	Yes	Single (16-Bit) V-memory word registers BCD Aliases for mapped Control Relays C0 – C3777
VCT0 – VCT360	Numeric	Yes	Yes	Single (16-Bit) V-memory word registers decimal Aliases for mapped Counter Status (done) bits CT0 – CT377
VCT0:B – VCT360:B	Numeric	Yes	Yes	Single (16-Bit) V-memory word registers BCD Aliases for mapped Counter Status (done) bits CT0 – CT377
VGX0 – VGX3760	Numeric	Yes	Yes	Single (16-Bit) V-memory word registers decimal Aliases for mapped Remote I/O Inputs GX0 – GX3777
VGX0:B – VGX3760:B	Numeric	Yes	Yes	Single (16-Bit) V-memory word registers BCD Aliases for mapped Remote I/O Inputs GX0 – GX3777
VGY0 – VGY3760	Numeric	Yes	Yes	Single (16-Bit) V-memory word registers decimal Aliases for mapped Remote I/O Outputs GY0 – GY3777

**Table 3-8.** 105/205/350/405 DirectLogic PLC Data Members

VG0:B – VG376:B	Numeric	Yes	Yes	Single (16-Bit) V-memory word registers BCD Aliases for mapped Remote I/O Outputs GY0 – GY377
VS0 – VS1760	Numeric	Yes	Yes	Single (16-Bit) V-memory word registers decimal Aliases for mapped Stage status (active) bits S0 – S177
VS0:B – VS1760:B	Numeric	Yes	Yes	Single (16-Bit) V-memory word registers BCD Aliases for mapped Stage status (active) bits S0 – S177
VSP0 – VSP760	Numeric	Yes	Yes	Single (16-Bit) V-memory word registers decimal Aliases for mapped Special Relays (system status bits) SP0 – SP77
VSP0:B – VSP760:B	Numeric	Yes	Yes	Single (16-Bit) V-memory word registers BCD Aliases for mapped Special Relays (system status bits) SP0 – SP77
VT0 – VT360	Numeric	Yes	Yes	Single (16-Bit) V-memory word registers decimal Aliases for mapped Timer Status (done) bits T0 – T37
VCT0:B – VCT360:B	Numeric	Yes	Yes	Single (16-Bit) V-memory word registers BCD Aliases for mapped Timer Status (done) bits T0 – T37
VX0 – VX1760	Numeric	Yes	Yes	Single (16-Bit) V-memory word registers decimal Aliases for mapped Inputs X0 – X177
VX0:B – VX1760:B	Numeric	Yes	Yes	Single (16-Bit) V-memory word registers BCD Aliases for mapped Inputs X0 – X177
VY0 – VY1760	Numeric	Yes	Yes	Single (16-Bit) V-memory word registers decimal Aliases for mapped Outputs Y0 – Y177
VY0:B – VY1760:B	Numeric	Yes	Yes	Single (16-Bit) V-memory word registers BCD Aliases for mapped Outputs Y0 – Y177

**Table 3-8.** 105/205/350/405 DirectLogic PLC Data Members

X0 – X1777	Logical	Yes	Yes	Inputs – addressed in octal and mapped to V40400 – V40477
Y0 – Y1777	Logical	Yes	Yes	Outputs – addressed in octal and mapped to V40500 – V40577

## 305/305S Direct Logic PLC Family Data Members

**Table 3-9.** 305/305S DirectLogic Data Members

Data Member	Type	Read	Write	Description
Activated	Logical	Yes	No	Object-generated signal when TRUE, this flag signifies an active communication connection between the process file and the PLC.
C160 – C373 C1000 – C1067	Logical	Yes	No	Control Relays – addressed in octal
C374 – C377 C770 – C777 C1070 – C1077	Logical	Yes	No	Special Relays – addressed in octal
Failed	Logical	Yes	No	Object-generated signal when TRUE, this flag signifies the process file is no longer communicating with the PLC.
IO0 – IO157 IO700 – IO767	Logical	Yes	No	Inputs and Outputs – addressed in octal
Paused	Logical	Yes	No	Object-generated signal when TRUE, this flag signifies that the communication connection has paused but is still active and will go FALSE when its paused condition is satisfied. Usually caused by modifying the Link while it is in use.
R0 – R777	Numeric	Yes	Yes	Single (8-Bit) byte registers decimal

**Table 3-9.** 305/305S DirectLogic Data Members

<b>Data Member</b>	<b>Type</b>	<b>Read</b>	<b>Write</b>	<b>Description</b>
R0:W – R776:W	Numeric	Yes	Yes	Single (16-Bit) word registers decimal
R0:WB – R776:WB	Numeric	Yes	Yes	Single (16-Bit) word registers BCD
RC160 – RC370 RC760RC1000 – RC1070	Numeric	Yes	Yes	Single (8-Bit) byte Control Relay and Special Relay registers decimal Aliases.
RC160:W – RC360:W RC760:W RC1000:W – RC1060:W	Numeric	Yes	Yes	Single (16-Bit) word Control Relay and Special Relay registers decimal Aliases.
RC160:WB – RC360:WB RC760:WB RC1000:WB – RC1060:WB	Numeric	Yes	Yes	Single (16-Bit) word Control Relay and Special Relay registers BCD Aliases.
RIO0 – RIO150 RIO700 – RIO760	Numeric	Yes	Yes	Single (8-Bit) byte registers decimal Aliases for mapped Inputs and Outputs IO0 – IO157 and IO700 – IO767
RIO0:W – RIO140:W RIO700:W – RIO750:W	Numeric	Yes	Yes	Single (16-Bit) word registers decimal Aliases for mapped Inputs and Outputs IO0 – IO157 and IO700 – IO767
RIO0:WB – RIO140:WB RIO700:WB – RIO750:WB	Numeric	Yes	Yes	Single (16-Bit) word registers BCD Aliases for mapped Inputs and Outputs IO0 – IO157 and IO700 – IO767
RS400 – RS570	Numeric	Yes	Yes	Single (8-Bit) byte Shift Registers decimal

**Table 3-9.** 305/305S DirectLogic Data Members

<b>Data Member</b>	<b>Type</b>	<b>Read</b>	<b>Write</b>	<b>Description</b>
RS400:W – RS560:W	Numeric	Yes	Yes	Single (16-Bit) word Shift Registers decimal
RS400:WB – RS560:WB	Numeric	Yes	Yes	Single (16-Bit) word Shift Registers BCD
SR400 – SR577	Logical	Yes	No	Shift Register Status Bits – addressed in octal
T600 – T677	Logical	Yes	No	Timer/Counter status (done) bits – addressed in octal
TCA600:WB – TCA677W:WB	Numeric	Yes	Yes	Timer/Counter current value words BCD – addressed in octal and mapped to R600 – R677

# GE\_Series90

GE\_Series90 is a protocol driver class LookoutDirect uses to communicate with GE Series 90-30 and GE Series 90-70 programmable logic controllers (PLCs) using SNPX, a Series Ninety Protocol.

**PLC Address** is a slave address and refers to the PLC address setting as configured on the device. The address can be up to eight ASCII characters.

**Model** chooses either 90-30 or 90-70.

**Interface** selects the protocol. You can choose between SNPX and Ethernet.

**Serial port** specifies which port the object uses for communication to the external device. This does not specify the communication type. Communication type is determined by the **Options»Serial Ports...** command.

**Data rate, Parity, Data bits, and Stop bits** reference the settings on the hardware device.

**Phone number** specifies the number to be dialed if the serial port setting is configured for dial-up. This number only applies to the individual protocol object.



**PollRate** is a numeric expression that determines how often to poll the device. GE\_Series90 then polls the device at the specified time interval. Normally, this is a simple time constant such as 0:01 (one second). See *Numeric Data Members* in Chapter 2, *How LookoutDirect Works*, of the *Getting Started with LookoutDirect* manual for information on entering time constants.

**Poll** is a logical expression. When this expression changes from FALSE to TRUE, LookoutDirect polls the device. You can use a simple expression like the signal from a pushbutton, or a complex algorithm.

**Communication alarm priority** determines the priority level of object-generated alarms (0 – 10).

**Retry attempts** specifies the consecutive number of times LookoutDirect attempts to establish communications with a device if it is not getting a valid response. After it tries the number of **Retry attempts** specified, the GE\_Series90 object generates an alarm and releases the communication port back to the communications service which then moves on to the next device in the polling queue (if any). Refer to Chapter 3, *Serial Communications*, in the *LookoutDirect Developer's Manual* for more information.

**Receive timeout** is the time delay LookoutDirect uses in waiting for a response from a device before retrying the request.

The **Skip every...** setting instructs LookoutDirect to not poll a device it has lost communication with on every scheduled poll. Instead, LookoutDirect skips the device in the polling cycle accordingly. Once communications have been reestablished, the device is polled on its regular cycle.

## GE\_Series90 Data Members

This protocol driver object contains a great deal of data. All readable and writable members (inputs/outputs), polling instructions, read/write blocking, serial port usage, and so on, are bundled with the object. Therefore, as soon as you create a GE\_Series90 object you immediately have access to all the object data members (see data member list in Table 3-10).



**Note** LookoutDirect protocol driver objects automatically generate an efficient read/write blocking scheme based on the inputs and outputs being used in your process file. You are not required to build your own I/O blocking table.

**Table 3-10.** GE\_Series90 Data Members

<b>Data Member</b>	<b>Type</b>	<b>Read</b>	<b>Write</b>	<b>Description</b>
AI1 – AI64	numeric	yes	yes	16-bit analog inputs encoded as unsigned binary integers ranging from 0 to 65535
AID1 – AID64	numeric	yes	yes	32-bit analog inputs encoded as unsigned binary integers ranging from 0 to 65535
AQ1 – AQ64	numeric	yes	yes	16-bit analog outputs encoded as unsigned binary integers ranging from 0 to 65535
AQD1 – AQD64	numeric	yes	yes	32-bit analog outputs encoded as unsigned binary integers ranging from 0 to 65535
CommFail	logical	yes	no	Object-generated signal that is on if, for whatever reason, <i>LookoutDirect</i> cannot communicate with the PLC.
I1 – I512	logical	yes	no	Single bit discrete inputs
M1 – M4096	logical	yes	yes	Single bit discrete (Internal coil)
OffHook	logical	no	yes	When TRUE, this flag instructs the GE object to retain exclusive use of its assigned communication port.
Poll	logical	no	yes	When this value transitions from FALSE to TRUE, <i>LookoutDirect</i> polls the device.
PollRate	numeric	no	yes	<i>LookoutDirect</i> expression that determines the device polling frequency.
Q1 – Q512	logical	yes	yes	Single bit discrete outputs
R1 – R9999	numeric	yes	yes	16-bit holding registers encoded as unsigned binary integers ranging from 0 to 65535

**Table 3-10.** GE\_Series90 Data Members (Continued)

<b>Data Member</b>	<b>Type</b>	<b>Read</b>	<b>Write</b>	<b>Description</b>
RD1 – RD9998	numeric	yes	yes	32-bit holding registers encoded as unsigned binary integers ranging from 0 to 65535
S1 – S32	logical	yes	no	System fault
SA1 – SA32	logical	yes	no	Special Contacts A
SB1 – SB32	logical	yes	no	Special Contacts B
SC1 – SC32	logical	yes	no	Special Contacts C
Update	logical	yes	no	Object-generated signal that pulses each time the driver polls the device

## GE\_Series90 Status Messages

### No response within timeout period

Lookout*Direct* did not received the expected response within the **Receive timeout** period. The object sent an inquiry and received an acknowledgment, but the device did not send an expected response to the request. This might happen if the response was interrupted. You may have to increase **Receive timeout**.

### No return inquiry response from secondary unit

Lookout*Direct* received no response from the device within the **Receive timeout** period. The driver object is able to use the COM port, but when it polls the device, it does not respond—as if it is not even there. You may have to increase **Receive timeout** to ensure Lookout*Direct* is allowing enough time to receive the expected response. Also, verify your baud rate settings, cable connections, power, configuration settings, COM port settings, and polling addresses.

### Bad LRC or BCC

The object is receiving a poll response from the device, but it could not decipher the response because it is garbled. Verify that all devices connected to the COM port have unique addresses. The last part of the message may actually be getting clipped off before it is completed. Consider increasing the number of **Retry attempts**. You may have to increase the **Receive gap** Serial Port setting to ensure Lookout*Direct* is receiving the entire message. If your Serial Port is configured for radio, this

could be caused by an audible squelch tail occurring at the end of a radio transmission. Try adjusting **RTS delay off** and **CTS timeout**.

**No attach response within timeout period**

An attempt was made to establish communications with the PLC without any response. Check your cabling and COM port selections, power, configuration settings, and polling addresses.

**Invalid response [x]**

An error in the structure of a response frame was detected. You may have two PLCs with the same address.

**Incorrect response length [x]**

A response was received with an unexpected length. You may have to increase the **Receive gap** Serial Port setting to ensure Lookout*Direct* is receiving the entire message.

**Incorrect response Address**

A response was received with an address not matching the objects address. You may have two master devices on the network.

**SNPX ERROR—Major code: x Minor code: x**

The response message contained an SNPX error code. Refer to your GE documentation for the meaning of this particular error.

# Modbus

---

Modbus and ModbusMOSCAD are protocol driver classes Lookout*Direct* uses to communicate with equipment such as programmable logic controllers (PLCs), remote terminal units (RTUs), or any other piece of equipment using Modbus Serial (ASCII or RTU) or Modbus Plus communication protocol.

The Modbus object class has general-purpose addresses, such as holding register 40001, and is suitable for communicating with nearly all Modbus devices, including the Control Microsystems TeleSAFE RTU.

The ModbusMOSCAD object class works with Motorola MOSCAD PLCs and RTUs. It also uses the Modbus Serial (ASCII or RTU) or Modbus Plus communication protocol, but its data members reflect the address of Motorola MOSCAD devices.

You can limit the number of channels Lookout*Direct* uses on the SA-85 card by creating a `modbus.ini` file in the Lookout*Direct* directory, as shown in the following example.

```
[ALL]
MaxChannels=channel
```

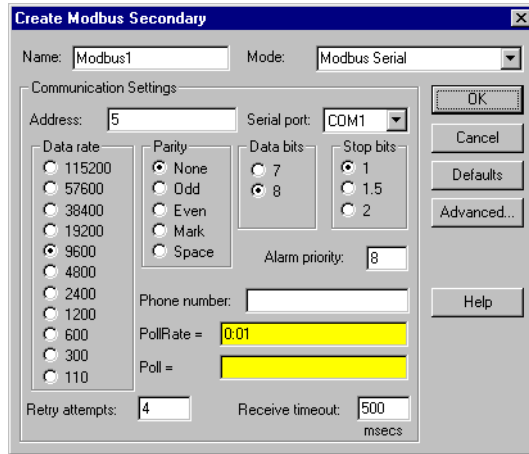
where *channel* is a number between 1 and 8, inclusive (default = 8).

These protocol driver objects contain a great deal of data. All readable and writable members (inputs/outputs), polling instructions, read/write blocking, serial port usage, and so on are bundled with the object. Therefore, as soon as you create a Modbus or ModbusMOSCAD object you immediately have access to all the object data members (see data member list in Table 3-12).



**Note** Lookout*Direct* protocol driver objects automatically generate an efficient read/write blocking scheme based on the inputs and outputs being used in your process file. You are not required to build your own I/O blocking table.

As protocol drivers, both object classes conform to the specifications in the Modicon Modbus Protocol Reference Guide PI-MBUS-300 Rev. C. The drivers support ASCII and RTU transmission modes, as well as Modbus Plus.



In this example, *LookoutDirect* is connected to a Modbus-speaking PLC with an address of 5 using serial port 1 (which was previously configured for hardwired communications), and polling the device every second.

**Modbus Serial** indicates that the slave device talks either Modbus ASCII or Modbus RTU. When you select this option, *LookoutDirect* first tries to communicate using the RTU format. If unsuccessful, it then tries the ASCII format (a little slower). If your network is susceptible to repeated communication problems, and if these problems slow scanning considerably, you may want to disable *LookoutDirect* from retrying both formats. This can speed communication retries by *LookoutDirect*; however, it will not fix your communication problems. Call National Instruments technical support to for information on how to prohibit *LookoutDirect* from trying to communicate using both formats.

**Modbus Plus Network** indicates that the slave device is connected to the *LookoutDirect* computer via a Modbus Plus network card.



**Note** NetBIOS-based networking software typically uses software interrupt 5C. This is also the default software interrupt used by the Modbus Plus Network card driver. Change the Modbus Plus software interrupt from 5C to 5D, 5E, or 5F. Refer to your Modicon documentation for instructions on changing the software interrupt setting.

**Modbus Ethernet** indicates you are communicating with the Modbus slave device using an Ethernet connection.

If you select **Modbus Serial**, you must specify **Address**, **Serial Port**, **Data Rate**, **Parity**, **Data Bits**, and **Stop Bits**. And if you are using a Dial-up

modem connected to your communication port, you must also specify a **Phone Number**.

If you select **Modbus Plus Network**, you need only specify the remote device **Address**.

If you select **Modbus Ethernet** you must specify the **IP address** in addition to **Alarm Priority**, **Poll Rate**, **Poll Retry attempts**, and **Receive timeout**.

**Address** is a slave address and refers to the PLC or RTU address setting as set on the device dip switches. If devices share a common line, they require unique addresses (1 to 255).

**IP address** is the Internet Protocol address for the Modbus slave object you are communicating with.

**Serial port** specifies which port the object uses for communication to the external device. This does not specify the communication type. Communication type is determined by the **Options»Serial Ports...** command.

**Data rate**, **Parity**, **Data bits**, and **Stop bits** reference the settings on the hardware device.

The **Defaults** button replaces the current settings with default values.

**Alarm priority** determines the priority level of Modbus-generated alarms.

**Phone number** specifies the number to be dialed if the serial port setting is configured for dial-up. This number only applies to the individual protocol object.

**PollRate** is a numeric expression that determines how often to poll the device. Modbus then polls the device at the specified time interval. Normally, this is a simple time constant such as 0:01 (one second). See *Numeric Data Members* in Chapter 2, *How LookoutDirect Works*, of the *Getting Started with LookoutDirect* manual for information on entering time constants.

**Poll** is a logical expression. When this expression changes from FALSE to TRUE, *LookoutDirect* polls the device. You can use a simple expression like the signal from a pushbutton, or a complex algorithm.

**Retry attempts** specifies the consecutive number of times *LookoutDirect* attempts to establish communications with a device if it is not getting a

valid response. After it tries the number of **Retry attempts** specified, the Modbus object generates an alarm and releases the communication port back to the communications subsystem which then moves on to the next device in the polling queue (if any). Refer to Chapter 3, *Serial Communications*, in the *LookoutDirect Developer's Manual* for more information.

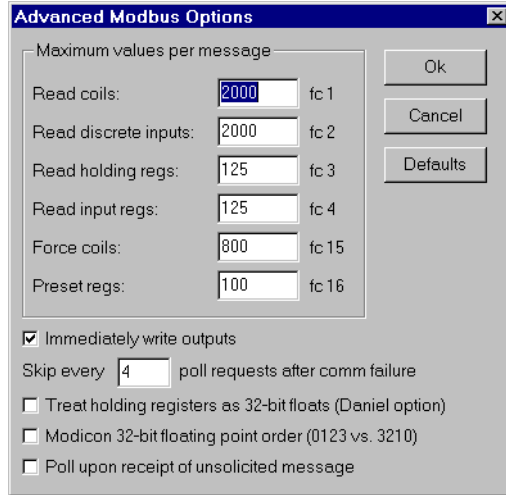
**Receive timeout** is the time delay *LookoutDirect* uses in waiting for a response from a device before retrying the request.

## Advanced Modbus Parameters

The Modbus driver attempts to block the reads and writes of coils, input registers and holding registers into groups to maximize communication efficiency. Through the **Advanced Modbus Options** dialog box, you can control the maximum block sizes that the driver uses. In fact, if your device does not support the default block sizes, you may have to specify smaller blocks.

The **Advanced...** button invokes the **Advanced Modbus Options** dialog box you can use to customize specific options within the Modbus protocol.





The Modbus object class uses Modbus Function Codes 01, 02, 03, 04, 05, 06, 15, and 16; and expects the remote I/O device to support these codes as specified by Modbus. The driver can communicate with up to 247 Modbus slave devices on each serial port.

The **Maximum values per message** settings specify the maximum number of elements *LookoutDirect* attempts to read (fc 1 – fc 4), or write (fc 15 and fc 16), in a single Modbus message. The default values represent the maximum number of elements that the protocol can transmit in a single message, and provides optimal speed. However, some devices are not capable of handling the maximum number of elements, so you should set the values according to the documentation for those devices.

If the **Immediately write outputs** option is ON, *LookoutDirect* immediately polls the device any time a value changes that is being written out to the device. If it is OFF, *LookoutDirect* waits until the next scheduled poll to write out changed values.

The **Skip every...** setting instructs *LookoutDirect* to not poll a device it has lost communication with on every scheduled poll. Instead, *LookoutDirect* skips the device in the polling cycle accordingly. Once communications have been reestablished, the device is polled on its regular cycle.

The **Daniel option** is device-dependent and instructs *LookoutDirect* to treat holding registers as 32-bit IEEE floating values instead of 16-bit values. If you set this flag, you must also set your hardware device to treat holding registers as 32-bit floats—most devices do not support this option, but Bristol-Babcock RTUs and Daniel flow meters do.



**Note** Activating the Daniel option deactivates all Modbus holding register members (on that device) except for 40001 – 49999 and 4000001 – 465000. If you attempt to read D40001, for example, the returned value is 0, and Lookout*Direct* will not attempt to write D40001 to the RTU. Of course, in devices that do not support this option, you can still read and write two adjacent holding registers as a floating point value with the Modbus data members F40001 – F4999. In fact, this is a more general purpose solution than the Daniel option, because you can still read bits and word values out of the holding registers, too.

Some Modbus PLCs reverse byte order in a floating point data member. If your process is returning garbled or senseless floating point values, select the **Modicon 32-bit floating point order (0123 vs. 3210)** option in the **Advanced Modbus Options** dialog box. This option chooses whether the characters within the floating point registers (data members F40001 – F49999 and F400001 – F465000) are in little endian or big endian format.

On occasion a Modbus PLC will *direct* a message to your process. To poll for all data member values following receipt of such a poll, select the **Poll on receipt of unsolicited message** option in the **Advanced Modbus Options** dialog box. Using this option is a way to force a poll of a Modbus device based on an event, without reference to the configured poll intervals and somewhat outside the usual Lookout*Direct* event-driven action.



**Caution** Notice that the D and F data members read two adjacent registers as single, 32-bit numbers. One consequence of this is that if you connect to two adjacent registers, such as **D40010** and **D40011**, you will get incorrect values because of overlapping. Make sure that you do not connect to adjacent registers with the D or F data members.

## Modbus Protocol Statistics

The driver monitors Modbus Protocol Statistics. This data is held within readable data members of the Modbus object and you can see them in the Modbus Protocol Statistics dialog box. To view the dialog box, select **Options»Modbus...** and click on **Statistics...**



**Note** The **Options»Modbus...** option is only visible in the **Options** menu if a Modbus object was previously created in your Lookout*Direct* application.

	count	%
Valid response frames:	0	0.0
Total errors:	0	0.0
Bad CRC or LRC:	0	0.0
Response too short:	0	0.0
No response:	0	0.0
Garbled:	0	0.0
Exceptions:	0	0.0
Since last reset:	11/18 16:00:05	

The **Count** column contains the accumulated number of messages received from the selected **Device** that fall into each respective category since the last time the Reset button was pressed. The percent column (%) indicates the percentage of messages received that fall into each respective category since the last time the Reset button was pressed.

When you depress the **Reset** button, the **ResetCounts** data member is set TRUE, setting all statistical values to zero. *LookoutDirect* records the date and time that the reset was last performed in the **Since last reset** data field.

## Modbus Data Members

The Modbus object class supports both 5-digit and 6-digit addressing. When you use a 6 digit address, the left-most digit represents the address *type* as follows:

**Table 3-11.** 6-Digit Address Coding

First Digit	Address Type
0	Single-bit coils
1	Discrete inputs
3	Input registers
4	Holding registers

The remaining 5 digits represent the actual address of the coil, input or holding register.



**Note** When you reference address 000001 and address 1, you are referring to the same point, but 40001 and 040001 do not refer to the same point. Because zero is the left-most digit in the 6-digit address, 040001 points to the 40001<sup>st</sup> single-bit coil and 40001 refers to the first holding register.

**Table 3-12.** Modbus Data Members

Data Member	Type	Read	Write	Description
000001 – 065000	logical	yes	yes	6-digit addresses of single-bit coils
1 – 9999	logical	yes	yes	Single-bit coils
100001 – 165000	logical	yes	no	6-digit addresses of single-bit discrete inputs
10001 – 19999	logical	yes	no	Single bit discrete inputs
300001 – 365000	numeric	yes	no	6-digit addresses of 16-bit input registers encoded as unsigned binary integers ranging from 0 to 65535
30001 – 39999	numeric	yes	no	16-bit input registers encoded as unsigned binary integers ranging from 0 to 65535

**Table 3-12.** Modbus Data Members (Continued)

<b>Data Member</b>	<b>Type</b>	<b>Read</b>	<b>Write</b>	<b>Description</b>
400001 – 465000	numeric	yes	yes	6-digit addresses of 16-bit input registers encoded as unsigned binary integers ranging from 0 to 65535
400001.1 – 465000.16	logical	yes	yes	6-digit address used to access individual bits out of holding registers and read them as logical ON/OFF values. The least significant bit is 1; the most significant, 16.
40001 – 49999	numeric	yes	yes	16-bit holding registers encoded as unsigned binary integers ranging from 0 to 65535
40001.1 – 49999.16	logical	yes	yes	Access individual bits out of holding registers and read them as logical ON/OFF values. The least significant bit is 1; the most significant, 16.
BadCRC	numeric	yes	no	Number of responses from device whose message failed the cyclic redundancy check (CRC) or the longitudinal redundancy check (LRC)
BCD300001 – BCD365000	numeric	yes	no	6-digit addresses of 16-bit input registers encoded as binary-coded decimal integers ranging from 0 to 9999
BCD30001 – BCD39999	numeric	yes	no	16-bit input registers encoded as binary-coded decimal integers ranging from 0 to 9999
BCD400001 – BCD465000	numeric	yes	yes	6-digit addresses of 16-bit holding registers encoded as binary-coded decimal integers ranging from 0 to 9999
BCD40001 – BCD49999	numeric	yes	yes	16-bit holding registers encoded as binary-coded decimal integers ranging from 0 to 9999

**Table 3-12.** Modbus Data Members (Continued)

<b>Data Member</b>	<b>Type</b>	<b>Read</b>	<b>Write</b>	<b>Description</b>
CommFail	logical	yes	no	Driver-generated signal that is ON if LookoutDirect cannot communicate with the device for whatever reason
D400001 – D465000	numeric	yes	yes	6-digit addresses of 32-bit unsigned holding register—reads two adjacent holding registers as a single 32-bit number ranging from 0 to 4,294,967,296.
D40001 – D49999	numeric	yes	yes	32-bit unsigned holding register—reads two adjacent holding registers as a single 32-bit number ranging from 0 to 4,294,967,296.
Exceptions	numeric	yes	no	Number of responses from device whose message was understandable to the driver but included an error code indication from the device
F400001 – F465000	numeric	yes	yes	6-digit addresses of 32-bit IEEE floating point register—reads two adjacent holding registers as a single 32-bit floating point value
F40001 – F49999	numeric	yes	yes	32-bit IEEE floating point register—reads two adjacent holding registers as a single 32-bit floating point value
Garbled	numeric	yes	no	Number of responses from device whose message was unintelligible to the driver
NoResponse	numeric	yes	no	Number of polls generated by driver not responded to by device
OffHook	logical	no	yes	When TRUE, this flag instructs the Modbus object to retain exclusive use of its assigned communication port
Poll	logical	no	yes	When this expression transitions from FALSE to TRUE, LookoutDirect polls the device.

**Table 3-12.** Modbus Data Members (Continued)

<b>Data Member</b>	<b>Type</b>	<b>Read</b>	<b>Write</b>	<b>Description</b>
PollRate	numeric	no	yes	Lookout <i>Direct</i> expression that determines the device polling frequency.
ProtocolErrors	numeric	yes	no	Total number of bad messages received from polled device
ResetCounts	logical	no	yes	Resets number to zero in the following data members: ValidFrame, NoResponse, TooShort, BadCRC, Garbled, Exceptions, & ProtocolErrors
S400001 – S465000	numeric	yes	yes	6-digit addresses of 16-bit holding registers encoded as signed binary integers ranging from –32767 to +32768.
S40001 – S49999	numeric	yes	yes	16-bit holding registers encoded as signed binary integers ranging from –32767 to +32768.
TooShort	numeric	yes	no	Number of responses from device whose message length was too short
Update	logical	yes	no	Driver-generated signal that pulses each time the driver polls the device
ValidFrame	numeric	yes	no	Number of good messages received from polled device

**Comments** You can use the OffHook data member to enhance communications when using the Modbus object class with dial-up modems. When OffHook is TRUE and the serial port is connected to a dial-up modem, the Modbus object does not hang up the modem when the poll is complete. Rather, it keeps the phone off the hook, retaining exclusive use of the serial port. As long as OffHook is TRUE, the Modbus object continues to poll the same PLC without hanging up the modem.

As soon as OffHook goes FALSE, the object releases the serial port to the communications subsystem, which goes to the next poll request in the queue, if any. The object also releases the port if data communications are lost for any reason—such as if the PLC modem breaks the connection.

When using OffHook, consider defining the driver object **PollRate** to poll fast when OffHook is TRUE, and poll at its normal rate when OffHook is FALSE. You might tie a Switch object to the OffHook writable data member for this very purpose.

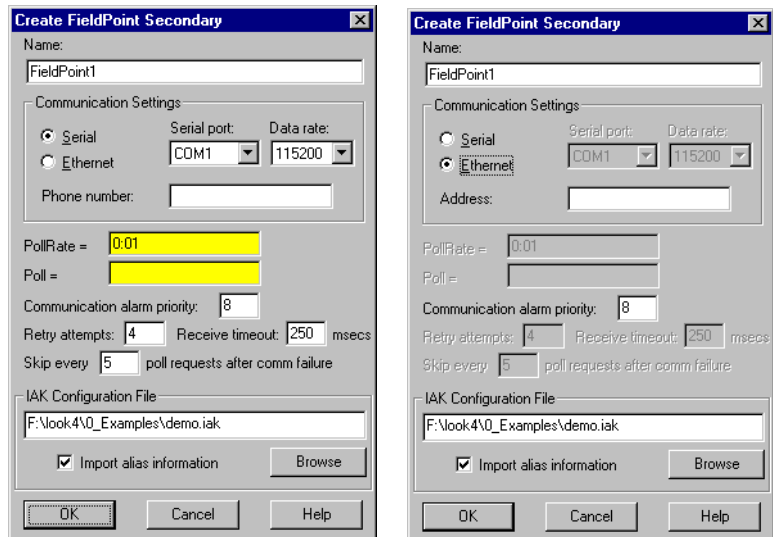


# National Instruments FieldPoint

FieldPoint is a protocol driver class LookoutDirect uses to communicate with FieldPoint devices using an enhanced version of the Optomux communication protocol. This object works with the FieldPoint models FP-1000, FP-1001, FP-AI-110, FP-AO-200, FP-DI-330, and FP-DO-400.

This protocol uses no parity, eight data bits and one stop bit. In LookoutDirect, a single FieldPoint object represents all devices connected to the same COM port.

The LookoutDirect FieldPoint object can read and write to all predefined data points allowed by the particular FieldPoint module. When you create a FieldPoint object, you have immediate access to all the object data members. See the *FieldPoint Data Members* section for more information on object data members.



When you select the **Ethernet** option, the serial port configuration options are disabled.

Enter the IP address in the **Address** field. You could also enter the FieldPoint network name instead.



**Note** If you replace a FieldPoint FP-1000 module (serial communications) with a FieldPoint FP-1600 module (Ethernet communications) or replace an FP-1600 module with an FP-1000 module, you must re-import your .IAK file. First, you must use

FieldPoint Explorer to update your .IAK file. If necessary, consult your FieldPoint online help for detailed instructions on using FieldPoint Explorer. To re-import the .IAK file, unselect the **Import alias information** checkbox and click on **OK**. Reopen the FieldPoint object dialog box and select the **Import alias information** checkbox again. When you click on **OK**, LookoutDirect re-imports the new .IAK file information.

**Serial port** specifies which COM port the object uses for communicating to the external device. This does not specify the communication type. Communication type is determined by the **Options»Serial Ports...** menu command.

**Data rate** indicates the baud rate that LookoutDirect uses to communicate with the hardware device. This setting should match the selection made on each of your network modules.

**Phone number** specifies the number to be dialed if the selected serial port is configured for dial-up. This number only applies to the individual protocol object.

**PollRate** is a numeric expression that determines how often to poll the device. The object then polls the device at the specified time interval. Ordinarily, this is a simple time constant such as 0:01 (one second). See *Numeric Data Members* in Chapter 2, *How LookoutDirect Works*, of the *Getting Started with LookoutDirect* manual for information on entering time constants.

**Poll** is a logical expression. When transitioned from FALSE to TRUE, LookoutDirect polls the device. This can be a simple expression, like the signal from a pushbutton, or it can be a complex algorithm.

**Communication alarm priority** determines the priority level of alarms generated by the FieldPoint object. Such alarms are typically related to communications with the physical device.

**Retry attempts** specifies the consecutive number of times LookoutDirect attempts to establish communications with a device if it is not getting a valid response. After it tries the number of **Retry attempts** specified, the FieldPoint object generates an alarm and releases the COM port. Refer to Chapter 3, *Serial Communications*, in the *LookoutDirect Developer's Manual* for more information.

**Receive timeout** is the time delay LookoutDirect uses in waiting for a response from a device before retrying the request.

The **Skip every n** setting instructs *LookoutDirect* to not poll a device it has lost communication with on every scheduled poll. Instead, *LookoutDirect* skips the device in the polling cycle. Once communications have been reestablished, the device is polled on its regular cycle.

**IAK configuration file** is a dialog for selecting an IAK configuration file. The IAK file contains alias and scaling information which is extracted for use in *LookoutDirect*. Choose the configuration file you want to use by entering the path directly, or use the **Browse** button. Check **Import alias information** if you want to extract information from the selected file.

## FieldPoint Data Members

As with all *LookoutDirect* drivers, you can access I/O points and other data through data members. The following is a table of data members currently supported by the FieldPoint object class.

**Table 3-13.** National Instruments FieldPoint Data Members

Data Member	Type	Read	Write	Description
AI000.00 - AI255.15	numeric	yes	no	Analog input channels.
AO000.00 - AO255.15	numeric	yes	yes	Analog output channels.
CA000.00 - CA255.15	numeric	yes	no	Use to connect to count inputs on a FieldPoint counter module.
CD000.00 - CD255.15	logical	yes	no	Use to connect to digital inputs on a FieldPoint counter module.
CI000.00 - CI255.15	logical	no	yes	Counter increment.
CO000.00 - CO255.15	logical	yes	yes	Use to connect to digital outputs in a FieldPoint Counter module.
CommFail	logical	yes	no	Goes high if <i>LookoutDirect</i> cannot communicate with the device.
CR000.00 - CR255.15	logical	no	yes	Counter reset.
DI000.00 - DI255.15	logical	yes	no	Discrete input channels.
DO000.00 - DO255.15	logical	yes	yes	Discrete output channels.
PF000.00 - PF255.15	numeric	yes	yes	Pulse width modulator period setting. Period is the entire on/off time of the pulse in milliseconds.

**Table 3-13.** National Instruments FieldPoint Data Members (Continued)

Data Member	Type	Read	Write	Description
Poll	logical	no	yes	When transitioned from low to high, <i>LookoutDirect</i> begins a poll cycle on the device.
PollRate	numeric	no	yes	Specifies the frequency at which <i>LookoutDirect</i> polls the device.
PW000.00 - PW255.15	numeric	yes	yes	Sets the FieldPoint pulse module duty cycle on/off ratio. A setting of 10.00 means the pulse is on for 10% of the pulse period and off for 90%.
RL000.00 - RL255.15	logical	yes	yes	Use to connect to digital output channels on a FieldPoint relay module.
RT000.00 - RT255.15	numeric	yes	no	RTD temperature measurement module analog input; returns a value in degrees Centigrade.
TC000.00 - TC255.15	numeric	yes	no	Use to connect to analog input channels on a FieldPoint thermocouple module.
Update	logical	yes	no	Goes high when <i>LookoutDirect</i> begins a poll cycle on the device.



**Note** When you use the CA, CD, CI, CO, CR, PW, RL, RT, and TC data members with FieldPoint serial modules, they are synonyms for analog or digital inputs or outputs as follows:

CI, CD—Digital Input

CR, CA, RL—Digital Output

PF, PP, RT, TC—Analog Input

PW—Analog Output

With FieldPoint Ethernet, these data members represent different FieldPoint modules.



**Note** The first two characters of the I/O data members represent the kind of module being accessed. The next three digits represent the device address of the module. This is the address of the I/O module itself, not the network module that governs it. Following the period are two digits representing the channel number within the module.

Not all of these data members are valid for every FieldPoint module. For all the device

types you are able to select the full range of device addresses and channels. So if you select DO123.03, you need to be certain that the device at address 123 is in fact a discrete output module.

For a more complete definition of the function of these data members, see FieldPoint documentation.



**Note** In the event of a power cycle to the FieldPoint device during use, the configuration of the device reverts to some default state, which is configurable. You should keep in mind that if the ranges you configured into the IAK file differ from those in the power-up configuration, the scaling information imported from the IAK file and used as a Lookout*Direct* alias might become outdated and incorrect after a power loss. To avoid this, make certain your power-up configuration ranges and your IAK configuration ranges are identical.

## FieldPoint Multiple Discrete Data Members

**Table 3-14.** Multiple Discrete Data Members

Data Member	Type	Read	Write	Description
MDI000.0000 - MDI255.FFFF	numeric	yes	no	Multiple discrete input channels
MDO000.0000 - MDO255.FFFF	numeric	yes	yes	Multiple discrete output channels

These special purpose data members are for reading or writing a numeric integer value to a set of discrete channels.

For instance, when you are configuring your modules with FieldPoint Explorer, you have the option of selecting more than one discrete channel for a data item that you are defining. If you do this and import the resulting .IAK file into Lookout*Direct* for use as aliases, the aliases created will correspond to this set of data members. You can then read and write to all the discrete channels with a single numeric data member. The data member names are in the form `MTTAAA.CCCC`, where:

`M`Indicates multiple as opposed to single

`TT`Two characters specifying module type (Discrete Out, Discrete In)

`AAA`Three numeric characters specifying module address

`CCCC`Four hexadecimal characters specifying which of the 16 channels are included in this data member



**Note** These data members will not enumerate. You may use them either by importing configurations from FieldPoint Explorer or by entering the data member name explicitly.

## FieldPoint Error Messages

### No response within timeout period

Lookout*Direct* received no response from a device within the **Receive timeout** period. The FieldPoint object is able to use the COM port, but when it polls the device, the device does not respond. If you have daisy-chained several devices, you have introduced an inherent delay. You may have to significantly increase **Receive timeout** (and **Poll Rate**) to ensure Lookout*Direct* is allowing enough time to receive the expected response. This increase has nothing to do with the processing capabilities of Lookout*Direct*. Rather it is based solely on **Data rate** and the number of devices on the chain. Also, verify your baud rate settings, cable connections, power, configuration settings, COM port settings, and polling addresses.

### Module returning ?? checksum

This means that the frame sent from the PLC in response to the command sent by Lookout*Direct* out returned ?? instead of a valid checksum. Check FieldPoint configuration.

### Message Garbled - Bad CRC

This means the checksum (CRC in this case) failed in a frame received by Lookout*Direct*. Check cabling or for two or more devices with the same address.

### Unexpected data response length

The frame received was of an unexpected length. Check the Lookout*Direct* **receive gap** setting.

### Error loading IAK configuration file

Lookout*Direct* was not able to successfully extract data from the .IAK configuration file. Try running the FieldPoint Explorer again and reconfigure your hardware.

### FP error: Power-up clear expected

A command other than power-up clear was attempted after power-up or power failure. The command sent is ignored and normal operations should resume.

**FP error: Undefined command**

The addressed module does not support this command. (for example, trying to write to an input module) Check to see if you are sending a command appropriate to the module.

**FP error: Checksum error**

This means the checksum (CRC in this case) failed in a frame sent by *LookoutDirect*. Check the *LookoutDirect* **receive gap** setting.

**FP error: Input buffer overrun**

The command sent to the FieldPoint module was too long. Check the *LookoutDirect* **receive gap** setting.

**FP error: Non-printable ASCII character received**

Only characters from ASCII value 33 to 127 are permitted in FieldPoint commands. The command is ignored.

**FP error: Data field error**

An insufficient or incorrect number of characters were received by the FieldPoint module for the specified command. Check the *LookoutDirect* **receive gap** setting.

**FP error: Communications link network watchdog timed out**

There has been no network traffic in the amount of time specified by your watchdog configuration settings, and the system has reverted to its watchdog defaults.

**FP error: Specified limits invalid for the command**

This includes the case where an invalid digit (hex or decimal) was received. Check the *LookoutDirect* **receive gap** setting.

**FP error: ASCII to binary conversion error**

One or more ASCII characters could not be converted to binary on the FieldPoint module. Check the *LookoutDirect* **receive gap** setting.

**FP error: Invalid device address**

The command is valid, but the addressed module does not support the command received. Check to see if you are sending a command appropriate to the module.

**FP error: Serial framing error**

An improperly framed command was received by the FieldPoint module. Check the *LookoutDirect* **receive gap** setting.

**FP error: Addressed module does not exist**

Make sure that you are addressing a valid module address.

**FP error: Invalid channel**

One or more channels specified in the command either do not exist or do not support the operation specified.

**FP error: Invalid range setting**

Check to see that the range information on the module has not changed, possibly due to a loss of power.

**FP error: Invalid operation for the module**

One or more module-specific operations specified in the command either do not exist or do not support the operation specified. Make sure that you are not requesting a discrete operation for an analog module, and vice versa.

**FP error: Module has been hotswapped since last command**

The alarm should deactivate immediately after it appears. Its appearance is only to acknowledge that a hot swap has occurred.

**FP error: Irrecoverable hardware fault**

A malfunction in the FieldPoint firmware or hardware has made communications from *LookoutDirect* impossible.

**Channel specific error: dev:##,ch:##,err:##**

These are error codes returned from the FieldPoint I/O modules. The alarm message specifies a device address, channel number, and error code. See FieldPoint documentation for a description of the error condition.

---

## National Instruments Lookout*Direct* OPC Client

---

Lookout*Direct* uses the National Instruments Lookout*Direct* OPC client to read data from and write data to any OPC server. It supports numeric, logical (boolean), and text I/O. The Lookout*Direct* 4 OPC client normally reads from the cache. You can connect a switch or pushbutton to the `PollDevice` data member to trigger a device update.

For your convenience, you can create `OPCFieldPoint` and `OPCNIDAQ` versions of the `OPCClient` object class by selecting these two special cases from the **Select Object Class** dialog box.





**Note** If you are using the National Instruments OPC IATest server, there is no device to poll. Triggering the PollDevice data member returns a zero (0). Simulated data will be restored shortly after such a poll.

The **Server Name** box enumerates all of the OPC servers registered on the local computer. Select the appropriate server.



**Note** The Server Name listbox contains only local servers. It has no effect on what you find when you browse remote servers.

You can then select one of the **In-Process Server**, **Local Server**, and **Remote Server** options. These options specify the type of server the OPC client will attempt to launch. If it cannot successfully connect to the selected server, the OPC client generates an alarm.

Select **Remote Server** to enable **Computer Name**, which specifies the name of the computer on which the remote server is to be launched. If you know the computer name you want, enter it preceded with two backslashes, as in \\PUMMEL.

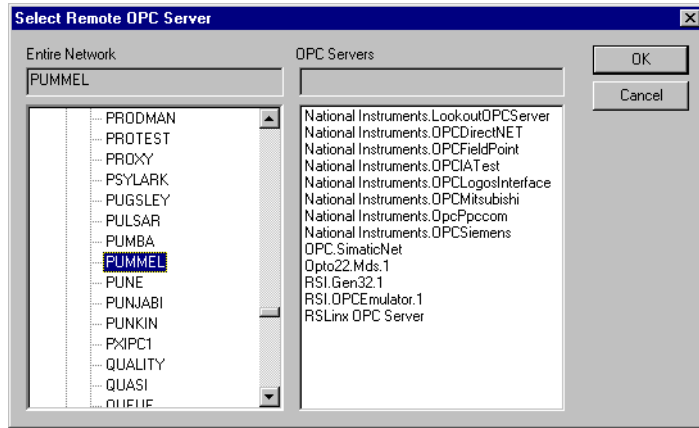


**Caution** In-process servers should only be used if LookoutDirect is the only program communicating with the server. If other programs address the same server, use the Local Server option.

The **Browsing** options include **Disabled**, **Flat**, and **Hierarchical**. If your OPC servers permit browsing, select either **Flat** or **Hierarchical**. If you

enable flat browsing all data members appear in any LookoutDirect windows displaying OPC client data members. If you select **Hierarchical**, the data members are arranged in hierarchical folders.

To browse remote servers, click on the browsing button next to the **Remote Server** field. The following dialog box appears.



Browse for the computer you want, and select the OPC server running on the computer that you want to access.



**Note** You must have the Microsoft Remote Registry service installed for remote OPC browsing to work on a computer running Windows 98/95. To install, select **Start»Settings»Control Panel** and then select **Network**. Check the Configuration tab to see if Microsoft Remote Registry is installed. If it is not, click on the **Add** button, and select **Service** in the **Select Network Component Type** dialog box. Choose Microsoft as the manufacturer. If the Remote Registry service is not visible, you will need your Windows 98/95 CD-ROM. You may have to browse through the disk to find the correct service. Once you have installed this service, you can successfully use remote OPC browsing in the LookoutDirect OPCClient object.

Select **Use Asynchronous I/O** if you want to use asynchronous communications with your OPC server. This is the preferred communications mode, and should be used when possible.

The **Force Refresh after Write** option is only available if you select the **Use Asynchronous I/O** option. This option forces the OPC server to return the current status of all data members every time you write to one.



**Note** Selecting **Force Refresh after Write** can sometimes impair performance of your LookoutDirect process, depending on the number of data members in your OPC server and

other system variables, including communications speed and what other tasks are being performed at any particular moment.

**Update Rate** is a numeric expression that determines how often the OPC server updates the client. Enter this time interval in milliseconds.

**Deadband** sets the percentage change that must take place in a data member before the OPC server reports a change in value to LookoutDirect.

**Poll Device** is a logical expression. When this expression changes from FALSE to TRUE, LookoutDirect polls the OPC server for all values. You can use a simple expression like the signal from a pushbutton, or a complex algorithm. In most applications there is no need to do device reads, so this parameter is often left blank.

Enter a **Default Access Path** if you want to simplify entering paths to various data members.

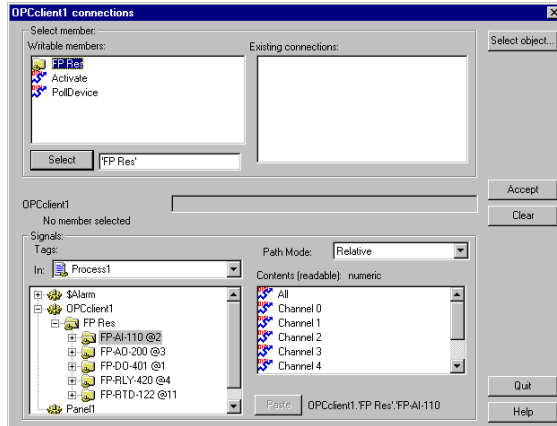
**Communication Alarm Priority** specifies the priority of alarms generated by this OPC client object.

## OPC Client Data Members

As with all LookoutDirect drivers, you can access I/O points and other data through data members.

Unlike other LookoutDirect driver objects, however, the OPCClient data member set changes depending on the OPC servers you have running on your computer.

If you selected hierarchical browsing when you created the OPCClient, the data members for the OPC server you have connected your OPC client to appear as data members inside a folder in your OPCClient list of data members in the LookoutDirect Object Explorer, connection editor, and other windows as shown in the following illustration.



If you selected flat browsing, you will see a long list of available data members without hierarchical organization into folders.

If you did not enable browsing, or your computer cannot browse certain OPC servers, you will only see the data members built into the LookoutDirect OPCClient.

The LookoutDirect OPCClient object class currently contains the built-in data members contained in the following table.

**Table 3-15.** OPCClient Data Members

Data Member	Type	Read	Write	Description
Activate	logical	no	yes	When TRUE, the OPC client is active and receives data. When FALSE, the OPC client does not update.
CommFail	logical	yes	no	Goes high if LookoutDirect cannot communicate with the server.
DataError	logical	yes	no	Goes high if the object cannot properly process the data returned by the server.

**Table 3-15.** OPCClient Data Members (Continued)

Data Member	Type	Read	Write	Description
PollDevice	logical	no	yes	When transitioned from FALSE to TRUE, the object polls the server.
Update	logical	yes	no	Pulses high and low when the OPC server updates LookoutDirect or LookoutDirect successfully polls the server.

The data members for the OPC Client depend on your OPC server. To use one of the data members from your server, enter the data member directly into an expression or URL field.

If the OPC server permits browsing, you can use data members as you would with any other LookoutDirect driver object. If you cannot browse your OPC server, enter the item manually in the form acceptable to your OPC server, as in the following example:

```
OPCclient1.'device\folder\itemName'
```



**Note** Notice the use of single quotes. Using a single quote around a component of a path allows the use of any character in that path, not just the normally allowed characters.

If all your items use the same access path, use the **Default Access Path** option. If you need to specify a different access path, use a period and tilde (.~) to denote the first element of the path, as in the following example:

```
OPCclient1.'device\itemN'.'~accesspath'
```

Notice that the access path must be a separate component. Single quotes denote a string as being a single component.

If an item name or access path already contains a tilde, you must enter one additional tilde to act as an escape character for that tilde, and a second additional tilde to denote an access path (for example, if the access path is ~COM1, enter ~~~COM1).

## Examples

```
OPCclient1.'4:0'.'~Modbus Demo Box'
```

The access path is Modbus Demo Box and the OPC item ID is 4:0.

```
OPCclient1.'4:0'
```

The access path is Null and the OPC item ID is 4 : 0.



**Note** You cannot browse access paths for an item. You must enter access paths manually.

# Omron

Omron is a protocol driver class *LookoutDirect* uses to communicate with Omron devices using the Host Link serial communication protocol.

An Omron object contains a great deal of data. It supports reading and writing of all predefined data points. When you create an Omron object, you have immediate access to all the data members for that object (see *Omron Data Members* list in Table 3-16).

**Serial port** specifies which COM port the object uses for communicating to the external device. This does not specify the communication type. Communication type is determined by the **Options»Serial Ports...** command.

**Data rate** indicates the baud rate that *LookoutDirect* uses to communicate with the hardware device. This **Data rate** setting should match the selection made on the physical device.

**Data bits** indicates the number of data bits that *LookoutDirect* uses to communicate with the hardware device. This **Data bits** setting should match the selection made on the physical device.

**Stop bits** indicates the number of stop bits that *LookoutDirect* uses to communicate with the hardware device. This **Stop bits** setting should match the selection made on the physical device.

**Parity** indicates the parity that *LookoutDirect* uses to communicate with the hardware device. This **Parity** setting should match the selection made on the physical device.

**Phone number** specifies the number to be dialed if the selected serial port is configured for dial-up. This **Phone number** only applies to the individual protocol object.

**PollRate** is a numeric expression that determines how often to poll the device. The object then polls the device at the specified time interval. Normally, this is a simple time constant such as 0:01 (one second). See *Numeric Data Members* in Chapter 2, *How LookoutDirect Works*, of the *Getting Started with LookoutDirect* manual for more information on entering time constants.

**Poll** is a logical expression. When this expression changes from FALSE to TRUE, *LookoutDirect* polls the device. You can use a simple expression like the signal from a pushbutton, or a complex algorithm.

**Communication alarm priority** determines the priority level of alarms generated by the Omron object. Such alarms are typically related to communications with the physical device.

**Retry attempts** specifies the consecutive number of times *LookoutDirect* attempts to establish communications with a device when it is not getting a valid response. After it tries the number of **Retry attempts** specified, the Omron object generates an alarm and releases the communication port back to the communications subsystem. The subsystem then moves on to the next device in the polling queue (if any). See Chapter 3, *Serial Communications*, in the *LookoutDirect Developer's Manual* for more information.

**Receive timeout** is the time delay *LookoutDirect* uses in waiting for a response from a device before retrying the request.

The **Skip every...** setting instructs *LookoutDirect* not to poll a device it has lost communication with on every scheduled poll. Instead, *LookoutDirect* skips the device in the polling cycle. Once communications have been reestablished, the device is polled on its regular cycle.

## Omron Data Members

As with all *LookoutDirect* drivers, you can access I/O points and other data through data members. The following is a table of data members currently supported by the Omron object class.



Table 3-16. Omron Data Members

Data Member	Type	Read	Write	Description
AR0–AR27	numeric	yes	yes	Auxiliary relay area, read as 16-bit word.
AR0.0–AR27.15	logical	yes	yes	Auxiliary relay area, read as 1-bit discrete.
DM0–DM9999	numeric	yes	yes	Data memory area, 16-bit word.
CommFail	logical	yes	no	Object-generated signal that is on if, for any reason, Lookout <i>Direct</i> cannot communicate with the device(s).
HR0–HR99	numeric	yes	yes	Holding relay area, read as 16-bit word.
HR0.0–HR99.15	logical	yes	yes	Holding relay area, read as 1-bit discrete.
IR0–IR511	numeric	yes	yes	I/O area, read as 16-bit word.
IR0.0–IR511.15	logical	yes	yes	I/O area, read as 1-bit discrete.
LR0–LR63	numeric	yes	yes	Link relay area, read as 16-bit word of information.
LR0.0–LR63.15	logical	yes	yes	Link relay area, read as 1-bit discrete.
TC0–TC999	numeric	yes	no	Timer/Counter, read as 16-bit word.
Update	logical	yes	no	Object-generated signal that pulses low each time it polls the device.



**Note** The Omron requires a special cable configuration in order to work properly. See your Omron hardware documentation for the correct configuration.

## Omron Status Messages

### No response within timeout period

Lookout*Direct* received no response from a device within the **Receive timeout** period. The Omron object is able to use the COM port, but when it polls the device, it does not respond—as if it is not even there.

**Cannot set PLC to MONITOR mode**

The Omron object is trying to set the PLC in MONITOR mode in order to communicate with the PLC correctly, but cannot perform the operation.

**Incorrect address in response**

The frame received had an incorrect source address. Check for two or more devices with the same address.

**Incorrect command in response**

The frame received had an incorrect command. Check for two or more devices with the same address.

**Incorrect data type in response**

The frame received had an incorrect data type marker.

**Incorrect frame check sum (FCS)**

The frame received had an incorrect check sum.

**Omron errors reported in the response**

These errors are reported by the Omron device, and are in turn reported to you in text form.

## Omron Models Supported

C20, C200, C500, C1000, C2000, CQM, CPM1

# Tiway

Tiway is a protocol driver object Lookout uses to communicate with series 5xx PLCs manufactured by Siemens, formerly made by Texas Instruments.

Protocol driver objects contain a great deal of data. All readable and writable members (inputs/outputs), polling instructions, read/write blocking, serial port usage, and so on are bundled with the object. As soon as you create a Tiway object you immediately have access to all the object data members (see data member list in this section).



**Note** Lookout protocol driver objects automatically generate an efficient read/write blocking scheme based on the inputs and outputs being used in your process file. You are not required to build your own I/O blocking table.

**PLC Model** specifies the PLC model number for the requested device.

**PollRate** is a numeric expression that determines how often to poll the device. Tiway then polls the device at the specified time interval. Normally, this is a simple time constant such as 0:01 (one second). See *Numeric Data Members* in Chapter 2, *How Lookout Works*, of the *Getting Started with Lookout* manual for more information on entering time constants.

**Poll** is a logical expression. When this expression changes from FALSE to TRUE, Lookout polls the device. You can use a simple expression like the signal from a pushbutton, or a complex algorithm.

**Receive timeout** is the time delay Lookout uses in waiting for a response from a device before retrying the request.

**Retry attempts** specifies the consecutive number of times Lookout attempts to establish communications with a device it does not get a valid response from. After Retry attempts times, Tiway generates a communication alarm and Lookout moves on to the next device in the polling queue (if any).

**Alarm priority** determines the priority level of Tiway generated alarms.

**Phone number** specifies the number to be dialed if the serial port setting is configured for dial-up. This number only applies to the individual protocol object.

The **Skip every \_\_\_ polls** setting instructs Lookout not to poll a device it has lost communication with on every scheduled poll. Instead, Lookout skips the device in the polling cycle accordingly. Once communications have been reestablished, the device is polled on its regular cycle.

## Update Write Settings

The Lookout default for the Tiway driver object is for the object to perform an update write to the PLC registers every 100 polls. Notice that this can be problematic if the PLC has been changing its own register values. To change this default operation, you must create an entry in the `Lookout.INI` file. Create a Tiway group with the key `UpdateOutputs`.

Setting the `UpdateOutputs` key equal to 0 means the Tiway object will not perform update writes. Setting the key equal to some positive integer N will set the Tiway object to perform update writes once every N polls.

See Appendix C, *The Lookout .INI File*, in the *Lookout Developer's Manual* for more information on using the `Lookout.INI` file. You can also refer to the `Lookout.INI` file topic in Lookout Help for additional information.

## Communication Techniques

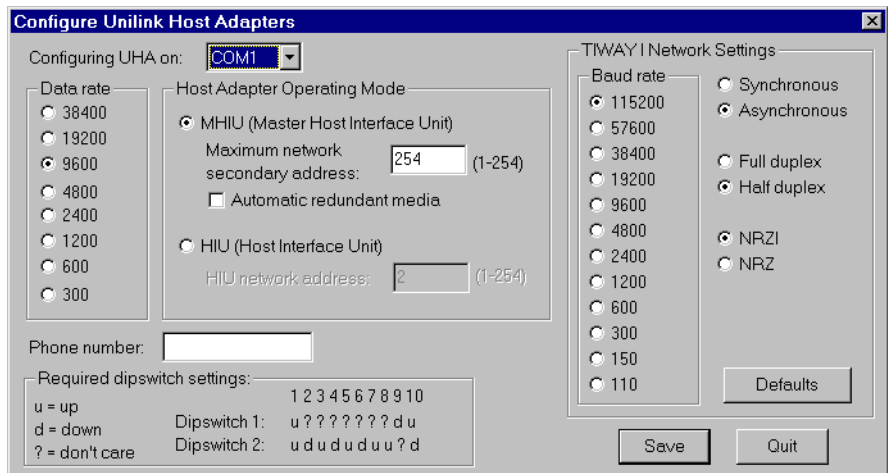
Lookout communicates with Siemens PLCs in several ways: direct serial connection to the **Local port**, serial connection to an **external Unilink Host Adapter**, through an internal **Unilink PC Adapter** card, or through an internal **CTI TCP/IP** card.

## Local Port

The **Local port** settings determine the **serial port**, **data rate**, and **phone number** (if any) to be used in a direct connect setup. Because the **Local port** protocol does not include address information, this option is limited to only one (1) PLC per serial port.

## Unilink Host Adapter

If **Unilink Host Adapter** is selected, you must specify the **Serial port** to be used and the **NIM** (Network Interface Module) **address** as set at the PLC. You also should configure several settings on the Unilink Host Adapter by selecting the **Configure UHA...** button.



The settings in this dialog box are globally applied to all PLCs on the specified TIWAY network (each network requires a separate serial port). Therefore, it is only necessary to configure each Unilink Host Adapter one time—you need not repeat this step every time you create a new Tiway object.

**Data rate** specifies the communication speed between the computer and the Unilink Host Adapter. It also determines the required dip switch settings on the UHA for the selected baud rate.

The **Host Adapter Operating Mode** determines if the Unilink Host Adapter is the network manager (**Master Host Interface Unit**) or just another network secondary (**Host Interface Unit**). There must be exactly one MHIU per TIWAY network.

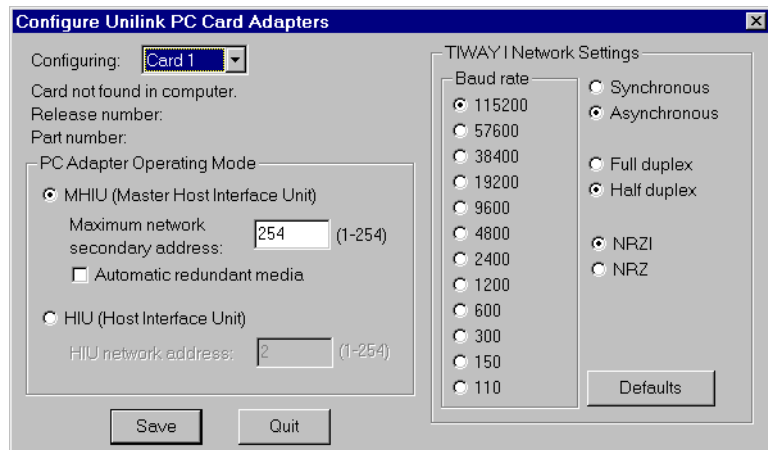
Enabling **Automatic redundant media** instructs the Unilink Host Adapter to attempt communications over a redundant TIWAY network to any secondary it loses communications with.

The **TIWAY I Network Settings** configure the communication parameters for the TIWAY network. This network runs between the Unilink Host Adapter and its secondaries. The Lookout default network settings correspond to the default NIM settings as shipped from Siemens. See your TIWAY documentation to modify any of these parameters.

## Unilink PC Adapter

Because the **Unilink PC Adapter** is an internal card, it eliminates the 19200 baud serial bottleneck and replaces it with the 8 MHz PC ISA bus speed. Therefore, the performance gains over the **Unilink Host Adapter** and **Local port** settings can be substantial.

If **Unilink PC Adapter** is selected, you must specify the **Card** to be used and the **NIM (Network Interface Module) address** as set at the PLC. You should also configure several settings on the Unilink PC Adapter by selecting the **Configure PCA...** button.



The settings in this dialog box are globally applied to all PLCs on the specified TIWAY network (each network requires a separate card). Therefore, it is only necessary to configure each Unilink PC Adapter one time—this step need not be repeated every time a new Tiway object is created.

The **PC Adapter Operating Mode** determines if the Unilink PC Adapter is the network manager (**Master Host Interface Unit**) or just another network secondary (**Host Interface Unit**). There must be exactly one MHIU per TIWAY network.

Enabling **Automatic redundant media** has no effect with the PC Adapter card because it has only one port. If Siemens adds a second port, Lookout automatically supports this option.

The **TIWAY I Network Settings** configure the communication parameters for the TIWAY network. This network runs between the Unilink PC Adapter card and its secondaries. Lookout default network settings correspond to the default NIM settings as shipped from Siemens. See your TIWAY documentation to modify any of these parameters.

## CTI TCP/IP

Lookout supports the Control Technology Incorporated (CTI) Ethernet TCP/IP adapter cards that can be installed in SIMATIC TI545 PLCs. In order to work with such cards, your PC must be equipped with an Ethernet network card and a Windows Sockets-Compliant TCP/IP software package. Such packages are available from Microsoft, FTP Software, and NetManage, Inc.

The Lookout **CTI TCP/IP** protocol option is Windows Sockets Compliant. It uses connectionless UDP sockets in software, an industry standard for TCP/IP protocols. In this protocol, a FIFO (first-in, first-out) stack is used to temporarily store communication messages if the data highway is busy or if multiple poll request are generated by several Tiway objects.

Because **CTI TCP/IP** utilizes sockets to momentarily store poll requests, this protocol eliminates bottlenecks imposed by multiple Tiway objects trying to access the data highway at the same time. Performance gains over **Local port, Unilink Host Adapter** and **Unilink PC Adapter** settings can be substantial when you are configuring a system that has several PLCs on the same network.

If **CTI TCP/IP** is selected, you need to specify the **IP address** (Internet protocol address) of the PLC. An Internet protocol address consists of four numbers, separated by periods. Each number ranges from zero to 255 decimal. Thus, a typical Internet address might be 128.7.9.231. Ensure that the **IP address** you enter matches the Internet protocol address of the PLC as specified in its EEPROM or as programmed using PCL.

You can add a secondary IP address to the **CTI TCP/IP** parameter. Lookout now toggles between the primary and secondary IP address after a COM failure (assuming a secondary address exists). Enter the secondary ID after the first, preceded by a space or a comma. For example:

207.68.156.61, 1.2.3.4

## Tiway Data Members

**Table 3-17.** Tiway Data Members

<b>Data Members</b>	<b>Type</b>	<b>Read</b>	<b>Write</b>	<b>Description</b>
AERR1 – AERR32000	numeric	yes	yes	(Analog Alarm) Error
AHA1 – AHA32000	numeric	yes	yes	(Analog Alarm) High alarm limit
AHHA1 – AHHA32000	numeric	yes	yes	(Analog Alarm) High high alarm limit
ALA1 – ALA32000	numeric	yes	yes	(Analog Alarm) Low alarm limit
ALLA1 – ALLA32000	numeric	yes	yes	(Analog Alarm) Low low alarm limit
AODA1 – AODA32000	numeric	yes	yes	(Analog Alarm) Orange deviation limit
APVH1 – APVH32000	numeric	yes	yes	(Analog Alarm) Process variable high limit
APVL1 – APVL32000	numeric	yes	yes	(Analog Alarm) Process variable low limit
ARCA1 – ARCA32000	numeric	yes	yes	(Analog Alarm) Rate of change limit
ASP1 – ASP32000	numeric	yes	yes	(Analog Alarm) Setpoint
ASPH1 – ASPH128	numeric	yes	yes	(Analog Alarm) Setpoint high limit
ASPL1 – ASPL128	numeric	yes	yes	(Analog Alarm) Setpoint low limit
ATS1 – ATS32000	numeric	yes	yes	(Analog Alarm) Sample rate
AVF1 – AVF128	numeric	yes	yes	(Analog Alarm) Alarm flags
C1 – C32000	logical	yes	yes	Control Registers
CommFail	logical	yes	no	Driver-generated signal that is ON if Lookout cannot communicate with the device for whatever reason



**Table 3-17.** Tiway Data Members (Continued)

<b>Data Members</b>	<b>Type</b>	<b>Read</b>	<b>Write</b>	<b>Description</b>
K1 – K32000	numeric	yes	yes	K-memory unsigned 16-bit integer value ranging from 0 to 65535
K1. – K32000.	numeric	yes	yes	K-memory 32-bit IEEE floating point value
K1D – K32000D	numeric	yes	yes	K-memory 32-bit unsigned integer value
K1S – K32000S	numeric	yes	yes	K-memory signed 16-bit integer value ranging from –32768 to 32767
LADB1 – LADB64	numeric	yes	yes	(Analog Alarm) Deadband
LADB1 – LADB64	numeric	yes	yes	(Loop) Deadband
LER1 – LER64	numeric	yes	no	(Loop) Error
LHA1 – LHA64	numeric	yes	yes	(Loop) High alarm limit
LHHA1 – LHHA64	numeric	yes	yes	(Loop) High high alarm limit
LKC1 – LKC64	numeric	yes	yes	(Loop) Gain
LKD1 – LKD64	numeric	yes	yes	(Loop) Derivative gain
LLA1 – LLA64	numeric	yes	yes	(Loop) Low alarm limit
LLLA1 – LLLA64	numeric	yes	yes	(Loop) Low low alarm limit
LMN1 – LMN64	numeric	yes	yes	(Loop) Output
LMX1 – LMX64	numeric	yes	yes	(Loop) Bias
LODA1 – LODA64	numeric	yes	yes	(Loop) Orange deviation limit
LPV1 – LPV64	numeric	yes	yes	(Loop) Process variable
LPVH1 – LPVH64	numeric	yes	yes	(Loop) Process variable high limit
LPVL1 – LPVL64	numeric	yes	yes	(Loop) Process variable low limit
LRCA1 – LRCA64	numeric	yes	yes	(Loop) Rate of change limit
LSP1 – LSP64	numeric	yes	yes	(Loop) Setpoint
LSPH1 – LSPH64	numeric	yes	yes	(Loop) Setpoint high limit
LSPL1 – LSPL64	numeric	yes	yes	(Loop) Setpoint low limit

**Table 3-17.** Tiway Data Members (Continued)

<b>Data Members</b>	<b>Type</b>	<b>Read</b>	<b>Write</b>	<b>Description</b>
LTD1 – LTD64	numeric	yes	yes	(Loop) Rate
LTI1 – LTI64	numeric	yes	yes	(Loop) Reset
LTS1 – LTS64	numeric	yes	yes	(Loop) Sample rate
LYDA1 – LYDA64	numeric	yes	yes	(Analog Alarm) Yellow deviation limit
LYDA1 – LYDA64	numeric	yes	yes	(Loop) Yellow deviation limit
Poll	logical	no	yes	When this value transitions from FALSE to TRUE, Lookout polls the device
PollRate	numeric	no	yes	Specifies the frequency at which the Lookout object polls the device
STW1 – STW32000	numeric	yes	no	Status Words
TCC1 – TCC32000	numeric	yes	yes	(Analog Alarm) Timer/counter current
TCP1 – TCP32000	numeric	yes	yes	(Analog Alarm) Timer/counter preset
Update	logical	yes	no	Driver-generated signal that pulses each time the driver polls the device
V1 – V32000	numeric	yes	yes	V-memory unsigned 16-bit integer value ranging from 0 to 65535
V1. – V32000.	numeric	yes	yes	V-memory 32-bit IEEE floating point value
V1B1 – V32000B16	logical	yes	yes	One bit of a word written out as a whole word
V1D – V32000D	numeric	yes	yes	V-memory 32-bit unsigned integer value
V1S – V32000S	numeric	yes	yes	V-memory signed 16-bit integer value ranging from -32768 to 32767
V1T – V32000T	text	yes	yes	Two characters of text

Table 3-17. Tiway Data Members (Continued)

Data Members	Type	Read	Write	Description
WX1 – WX32000	numeric	yes	no	Word Image Inputs—16-bit values that typically range from 6400 – 32000 for 4 – 20 mA signals, and 0 – 32000 for 0 – 5V signals.
WY1 – WY32000	numeric	yes	yes	Word Image Outputs—16-bit values that typically range from 6400 – 32000 for 4 –20 mA signals, and 0 – 32000 for 0 – 5V signals.
X1 – X32000	logical	yes	no	Discrete Inputs—unassigned Xs may be used as control registers
Y1 – Y32000	logical	yes	yes	Discrete Outputs—same memory space as Discrete Inputs, so X37 references the same point as Y37. Unassigned Ys may be used as control registers

## Importing APT Name Files

After you have created at least one Tiway object, the Tiway class adds a menu selection to the Lookout **Options** menu you can use to import an APT name file database for each Tiway object created. You can re-import name files as your APT programs are modified, and Lookout readjusts the aliased name names automatically, in real time.

