Honeywell

# Experion PKS
# Omni Interface Reference

EPDOC-XX77-en-431A
February 2015

**Release 431**

Honeywell

| Document | Release | Issue | Date |
|---|---|---|---|
| EPDOC-XX77-en-431A | 431 | 0 | February 2015 |

## Disclaimer

This document contains Honeywell proprietary information. Information contained herein is to be used solely for the purpose submitted, and no part of this document or its contents shall be reproduced, published, or disclosed to a third party without the express permission of Honeywell International Sàrl.

While this information is presented in good faith and believed to be accurate, Honeywell disclaims the implied warranties of merchantability and fitness for a purpose and makes no express warranties except as may be stated in its written agreement with and for its customer.

In no event is Honeywell liable to anyone for any direct, special, or consequential damages. The information and specifications in this document are subject to change without notice.

Copyright 2015 - Honeywell International Sàrl

# Contents

# Planning considerations for installing and configuring Omni Flow Computers

This reference describes how to set up, configure, and test Omni controller communications with the server.

### Revision history

| Revision | Date | Description |
|---|---|---|
| A | February 2015 | Initial release of document. |

### How to use this guide

The following steps show the order in which the controller interface should be configured. Complete each step before starting the next.

Steps for connecting and configuring an Omni controller.

| Steps | Go to |
|---|---|
| Connect the controller to the server | Hardware and software supported by Omni |
| Set the communication parameters | Network connections supported by Omni |
| Use Quick Builder to define channels | • Omni channel and controller reference<br>• *Quick Builder User's Guide* |
| Use Quick Builder to define controllers | • Omni channel and controller reference<br>• *Quick Builder User's Guide* |
| Download channel and controller definitions to the server | *Quick Builder User's Guide* |
| Test communications | Testing Omni communications with the server |
| Use Quick Builder to define points | Omni points reference |

### Related topics

"Hardware and software supported by Omni" on page 6
"Network connections supported by Omni" on page 7
"Data tables, addresses, and files supported by Omni" on page 8
"Function codes supported by Omni" on page 10
"About Omni utilities" on page 11
"Other documentation for Omni" on page 12
"Omni technical support" on page 13
"Omni communication module settings" on page 14
"Testing Omni communications with and the server" on page 32
"Omni points reference" on page 29

# Hardware and software supported by Omni

This section describes Omni hardware and software supported by Experion interface.

**Related topics**

“Omni hardware compatibility” on page 6
“Omni software compatibility” on page 6

## Omni hardware compatibility

The Omni interface supports Omni 3000™ and Omni 6000™ Flow Computers manufactured by Omni Flow Computers. It enables use of Omni RS-232/485 Serial I/O Modules (Model Number 68-6205) and/or Omni Serial/Ethernet-Modbus Mux Module (Model Number 68-6209).

For the current Omni documentation, see the Omni Flow Computer website.

**Related topics**

“Other documentation for Omni” on page 12

## Omni software compatibility

The Omni interface has no known compatibility issues. It has been qualified using the following:

- Omni 6000 version 24.74.17
- OmniCom for Windows version 1.23

**Attention**

The Omni interface supports the Modbus RTU and Modbus/TCP protocols. The Modbus ASCII protocol is not supported.

For serial and terminal server connections, the supported protocol is only Modbus RTU. For LANVendor connections, the supported protocols are Encapsulated Modbus RTU and Modbus/TCP. For the Modbus RTU protocol, the Cyclic Redundancy Check (CRC) is required.

# Network connections supported by Omni

This section describes network connections and redundancy.

**Related topics**

“Serial connections for Omni” on page 7
“Ethernet connections for Omni” on page 7
“Redundancy for Omni” on page 7

## Serial connections for Omni

Omni serial connections can include multi-drop connections of up to 12 flow computers on a single serial connection. You can also connect using a terminal server and/or modem.

For terminal server or modem connections, the Omni protocol must be configured as type 2, Modbus RTU (Modem).

For details about wiring serial connections to the Omni Flow Computer, see "Section 7.3: Connecting to a Personal Computer and Modem" in "Volume 1: System Architecture and Installation" of the *Omni 3000/6000 Flow Computer User Manual*.

**Related topics**

“Ethernet connections for Omni” on page 7
“Other documentation for Omni” on page 12

## Ethernet connections for Omni

For details about connecting the Omni Serial/Ethernet–Modbus Mux Module Model 68-6209, see the *Omni Technical Bulletin TB-020101G*.

**Related topics**

“Serial connections for Omni” on page 7
“Other documentation for Omni” on page 12

## Redundancy for Omni

Omni Flow Computers enable redundant Flow Computers using a peer-to-peer connection.

For redundant Flow Computers, you can use the redundant channel and controller port configuration to address the redundant Flow Computer.

For non-redundant Omni Flow Computers, you can use the channel and controller port configuration to address a separate communications port on the Omni.

# Data tables, addresses, and files supported by Omni

The Omni Flow Computer communicates using the Modbus protocol. The Omni implementation differs from some standard Modbus rules, requiring an Omni–specific interface. The following sections describe the protocol differences.

### Long data types

The Modbus standard supports 16-bit register values. It also supports longer data types to be transferred as 16-bit registers that are adjacent. The standard supports the transfer of a 32-bit value as two 16-bit registers. However, Omni supports one register per address for all data types. This means Omni returns a 32-bit value as one 32-bit register. This differs from both the register counter and the byte counter in the Modbus standard.

### Invalid addresses

The Modbus standard supports data packets of up to 256 bytes. Data acquisition systems use this provision to optimize data request packets.

The Omni also has some century boundary (*xxx00*) addresses that are not implemented. If these addresses occur within a requested data packet, zero values are returned for addresses above the *Spare* or *Reserved* address. This requires a modification of the packet optimization scheme to force separate packets to be requested around century boundaries.

### Non-numeric data

The Omni contains data stored as ASCII strings, report buffers, and archives that are not included in the Modbus standard.

These non-numeric data types are not available through the normal data acquisition process.

### Omni address map

The following table provides an overview of the Omni address space reference. For more details, see "Volume 4, Modbus Protocol Implementation" of the *Omni 3000/6000 User Manual*.

**Table 1: Omni address map**

| Address | Data Type | Description |
|---|---|---|
| 0000–0999 | Custom | Custom data packets and archive arrays.[1] |
| 1000–2999 | Boolean | General Boolean status and alarm flags and commands. |
| 3000–3999 | 16-bit integer | General configuration. |
| 4000–4999 | 8-character ASCII | General string data.[1] |
| 5000–5999 | 32-bit integer | Meter data. |
| 6000–8999 | 32-bit floating point | Meter data. |
| 9000–9999 | ASCII buffers | Reports.[1] |
| 10000–12999 | Undefined | Reserved. |
| 13000–13999 | 16-bit integer | Meter configuration. |
| 14000–14999 | 16-character ASCII | General string data.[1] |
| 15000–15999 | 32-bit integer | Meter data. |
| 16000–16999 | Undefined | Reserved. |
| 17000–17999 | 32-bit floating point | Meter configuration. |

[1] This is *not directly available* with the Omni interface

| Address | Data Type | Description |
|---|---|---|
| 18000–18999 | Undefined | Reserved. |

### Related topics

“Optimizing Omni scanning performance” on page 24
“Function codes supported by Omni” on page 10
“Address syntax for Omni controllers” on page 30

# Function codes supported by Omni

The following table lists the Modbus function codes used by Omni.

**Attention**

- All writes involve a single value per control. Writes to adjacent registers, or adjacent bit fields within an integer register, are processed separately even if the Modbus and/or Omni protocols would allow a single write command to modify multiple values.
- The Omni interface supports the Modbus RTU and Modbus/TCP protocols. The Modbus ASCII protocol is not supported. For serial and terminal server connections, the supported protocol is only Modbus RTU. For LANVendor connections, the supported protocols are Encapsulated Modbus RTU and Modbus/TCP. For the Modbus RTU protocol, the Cyclic Redundancy Check (CRC) is required.

**Table 2: Modbus function codes**

| Address | Data Type | Function codes |
|---|---|---|
| 1000–2999 | Boolean | Read: 1 = Read Boolean<br>Write: 5 = Write Boolean |
| 3000–3999 | 16-bit integer<br>(length = 2 bytes) | Read: 3 = Read multiple registers<br>Write: 16 = Write multiple registers<br>(Writing bits in an integer register involves a read/modify/write cycle) |
| 5000–5999 | 32-bit integer<br>(length = 4 bytes) | |
| 6000–8999 | 16-bit integer<br>(length = 4 bytes) | |
| 13000–13999 | 16-bit integer<br>(length= 2 bytes) | |
| 15000–15999 | 32-bit integer<br>(length = 4 bytes) | |
| 17000–17999 | 32-bit floating point<br>(length = 4 bytes) | |

### Related topics

“Data tables, addresses, and files supported by Omni” on page 8

# About Omni utilities

You can use the use the Omni configuration test utility, **omntst**, to perform the following tasks:

- Synchronizing the Omni Flow Computer time with the Experion server time.
- Testing the communications between the Omni Flow Computer and the Experion server.
- Communicating with password-protected Omni ports.

### Related topics

“Configuring the Omni password” on page 26
“Synchronizing the Omni Flow Computer time with the server time” on page 25
“Testing Omni communications with and the server” on page 32

# Other documentation for Omni

You can download the current version of the following Omni-related documents from the Omni Flow Computer website: http://www.omniflow.com.

- Volume 1 - System Architecture and Installation: *Omni 3000/6000 Flow Computer User Manual*.
- Omni Serial/Ethernet-Modbus Mux Module Model 68-6209: Setup & Installation - *Technical Bulletin TB-020101G*.
- Volume 4 - Modbus Protocol Implementation: *Omni 3000/6000 Flow Computer User Manual*.

### Related topics

“Omni communication module settings” on page 14
“Omni hardware compatibility” on page 6
“Serial connections for Omni” on page 7
“Ethernet connections for Omni” on page 7

# Omni technical support

- For technical support of the Omni Flow Computer, contact Omni Flow Computers, Inc.:
  - Web: http://www.omniflow.com/
  - Telephone: *1-281-240-6161*
- The following applications are available on a project basis:
  - *omni_asc*. For transfer of ASCII data between string point parameters and Omni string registers.
  - *omni_rpt*. For reading/writing Omni reports and report templates.

For information and support regarding these applications, contact your Honeywell representative.

# Omni communication module settings

The Omni interface supports connections using either the Omni RS-232 or the RS-485 Serial I/O Modules (Model Number 68-6205) and/or the Omni Serial/Ethernet-Modbus Mux Modules (Model 68-6209).

### Omni model number 68-6205 RS-232/485 Serial I/O Module

For details about communication module settings, see section 1.7.3, "Serial Communication Modules" in "Volume 1, System Architecture and Installation" of the *Omni 3000/6000 Flow Computer User Manual*.

### Omni model number 68-6209 Serial/Ethernet-Modbus Mux Module

For details about communication module settings, see the *Technical Bulletin, TB-020101G, Omni Serial/ Ethernet-Modbus Mux Module Model 68-6209: Setup & Installation.*

### Related topics

“Other documentation for Omni” on page 12

# Omni channel and controller reference

This section describes the configuration and addressing information specific to Omni channels and controllers.

In addition to the information contained in this reference, and for help to build channels and controllers, see the section titled "Building controllers or channels" in the *Quick Builder User's Guide*.

### Related topics


“Main properties for an Omni channel” on page 16

“Port properties for an Omni channel” on page 18

“Redundant port properties for an Omni channel” on page 21

“Main properties for an Omni controller” on page 22

“Optimizing Omni scanning performance” on page 24

“Synchronizing the Omni Flow Computer time with the server time” on page 25

“Configuring the Omni password” on page 26

# Main properties for an Omni channel

The Main tab defines the basic properties for an Omni channel.

For information about how to create a channel, see the topic titled "Building controllers and channels" in the *Quick Builder User's Guide*.

**Attention**
If you use both serial and Ethernet Omni controllers, you need to create a separate channel for each type of controller.

| Property | Description |
|---|---|
| Name | The unique name of the channel. A maximum of *10* alphanumeric characters (no spaces or double quotes). Note: In Station displays, underscore characters ( _ ) appear as spaces. |
| Description | (Optional) A description of the channel. A maximum of *132* alphanumeric characters, including spaces. |
| Marginal Alarm Limit | The communications alarm marginal limit at which the channel is declared to be marginal. When this limit is reached, a high priority alarm is generated. To change the priority of the alarm system wide, see the topic titled "Configuring system alarm priorities" in the *Server and Client Configuration Guide*. To change the priority of the alarm for one channel, see the topic titled "About configuring custom system alarm priorities for an individual channel or controller" in the *Server and Client Configuration Guide*.<br><br>A channel barometer monitors the total number of requests and the number of times the controller did not respond or response was incorrect. The barometer increments by two or more, depending on the error, and decrements for each good call.<br><br>To calculate an acceptable marginal alarm limit, use the formula: Square root of the number of controllers on the channel × Marginal Alarm Limit defined on those controllers (Normally, you specify the same value for all controllers on a channel).<br><br>For example, if there are 9 controllers on the channel and their Marginal Alarm Limit is set to *25*, the value would be *3* (which is the square root of 9) *x 25 = 75*. |
| Fail Alarm Limit | The communications alarm fail limit at which the channel is declared to have failed. When this barometer limit is reached, an urgent alarm is generated. To change the priority of the alarm system wide, see the topic titled "Configuring system alarm priorities" in the *Server and Client Configuration Guide*. To change the priority of the alarm for one channel, see the topic titled "About configuring custom system alarm priorities for an individual channel or controller" in the *Server and Client Configuration Guide*.<br><br>Set this to double the value specified for the channel Marginal Alarm Limit. |
| Diagnostic Scan Rate | The period, in seconds, between diagnostic scans that verify communications integrity with the controller. The default value is *60* seconds.<br><br>The diagnostic scans continue even if a controller is marked as failed, thus enabling the system to detect return-to-normal communications.<br><br>If there are multiple controllers configured on a channel, the diagnostic scan rate should be set to *60* seconds multiplied by the number of controllers on that channel. For example, if there are five controllers configured on the channel, the diagnostic scan rate should be set to *300* seconds. Diagnostic scans occur at the set scan rate per controller. Therefore, with five controllers and a default value of *60* seconds, the diagnostic scans will occur every *12* seconds. |

| Property | Description |
|---|---|
| Write Delay | If the channel is on a serial port, the length of time (in milliseconds) that the server waits before writing to any controller on the channel. The default value is *10* milliseconds.<br>A write delay is usually specified only if:<br>• The server communicates to the controller over a half-duplex radio link and the radio system requires time to key in each direction before the server or controller can send data.<br>• The radio system implements *RTS/CTS* handshaking.<br>If there is a communications problem and the controller does not respond to writes from the server, try changing this setting to *11* milliseconds or more. This should allow the controller enough time to become ready to receive data from the server. |
| Connect Timeout | The length of time that the server attempts to connect to the controller. The server will stop trying to connect to the controller once the timeout period passes. The default value *10* seconds.<br>Use the default value unless the communications line has a high error rate, or unless you are using modems. |
| Read Timeout | The length of time that the server will wait for a reply from the controller. The server will stop waiting once the timeout period passes. The default value is *2* seconds.<br>Use the default value unless the communications line has a high error rate, or unless you are using modems. |
| Item Type | The type of channel specified when this item was created. |
| Last Modified | The date and time the channel properties were modified. |
| Last Downloaded | The date and time the channel was last downloaded to the server. |
| Item Number | The unique item number currently assigned to this channel, in the format *CHNcc*, where *cc* is the channel number.<br>You can change the item number if you need to match your current server database configuration. The number must be between *01* and the maximum number of channels allowed for your system. For more information about setting the maximum value, see the topic titled "Adjusting sizing of non-licensed items" in the *Supplementary Installation Tasks Guide*. |

# Port properties for an Omni channel

The Port tab defines the communication-related properties for a channel. The **Port Type** for Omni connections can be:

- *Serial*. Only applicable to a directly-connected serial connection, such as RS-232, using a server COM port.
- *TerminalServer*. Only applicable to an architecture using network communications to a terminal server which is then connected to an Omni Flow Computer.
- *LANVendor*. Only applicable to a network connection to an Omni Ethernet-Modbus port.

### Serial port properties

**Attention**
Serial port properties are only applicable to serial Omni controllers.

| Property | Description |
|---|---|
| Serial Port Name | The device name of the serial port. |
| Protocol | **Attention**<br>• Do not use the *Modbus ASCII* option, as it is not supported. The only valid Omni Serial Port protocol is *Modbus RTU*.<br>• The Omni Flow Computer port must be configured for Modbus RTU [Modbus Type=0] or Modbus RTU (Modem) [Modbus Type=2] protocol. |
| Baud | The number of data bits per second.<br>The default is *9600*. |
| Number of Data Bits | The number of data bits used for transmission.<br>The default is *8*. |
| Stop Bits | The number of stop bits used for transmission<br>The default is *1*. |
| Parity | Defines parity verification of each character and must match configuration on the end device.<br>The default is *NONE*. |
| Checksum | The type of checksum error detection used for the port. Select the value that matches the setting on the communication device.<br>**Attention**<br>• Do not change the default value *NONE*. The Omni CRC is implemented internally by the Omni interface, and *not* through these Checksum options.<br>• The Omni Flow Computer port must be configured with CRC enabled. |
| XON/XOFF | The type of XON/XOFF software flow control used to stop a receiver from being overrun with messages from a sender. The types are:<br>• *Input* (use XON/XOFF to control the flow of data on the receive line)<br>• *None* (default)<br>• *Output* (use XON/XOFF to control the flow of data on the transmit line) |

| Property | Description |
|---|---|
| RS-232 | **Attention**<br>This is applicable only if the Omni Flow Computer is connected to the server using a modem connected to the RS-232 COM port.<br><br>These options are applicable to the RS-232 link:<br>• **Enable RTS/CTS flow control**. Select this if you want to use RTS/CTS for flow control to stop a receiver from being overrun with messages from a sender.<br>• **Detect DCD**. Select this if the Data Carrier Detect communication status line of the COM port requires monitoring (usually when using modem or microwave linking). When selected, the communications fails if the desired COM status line is not high—for example, on a dial-up link connection for a modem.<br>• **Detect DSR**. Select this if the Data Set Ready communication status line of the COM port requires monitoring (usually when using modem or microwave linking). When selected, the communications fails if the desired COM status is not achieved. |
| RS-485 | **Attention**<br>It is not recommended to use RS-485 connections for the Omni.<br><br>These options are applicable to the RS-485 link:<br>• **Enable Stallion RS-485 Half Duplex**. Select if RS-232 to RS-485 is done using a Stallion EasyConnection adapter.<br>• **Echo (Required for Stallion RS-485 ports)**. Select so that the server expects the messages it sends to the port on the transmit line to be echoed back on the receive line. Select for a Stallion EasyConnection adapter or a Black Box converter. |

### Terminal server port properties

**Attention**
Terminal Sever port properties are only applicable to an Omni Flow Computer that communicates with the server through a terminal server.

| Property | Description |
|---|---|
| Protocol | **Attention**<br>• Do not use the *Modbus ASCII* option, as it is not supported. The only valid Omni Terminal Server Port protocol is *Modbus RTU*.<br>• The Omni Flow Computer port must be configured for Modbus RTU (Modem) [Modbus Type=2] protocol. |
| Host Name | The host name or IP address of the terminal server to which the channel is connected.<br><br>Host names are defined in the *%SystemRoot%\system32\drivers\etc\hosts* file on each computer. |
| Terminal Server TCP Port No. | The TCP port number used for communication with the desired terminal server serial port.<br><br>See the documentation for the terminal server to configure this port number in the terminal server. |
| Idle Timeout | The time, in seconds, the channel waits for a successful connection to the server before closing the connection.<br><br>A value of *0* indicates that the connection is never closed. |

| Property | Description |
|---|---|
| Checksum | The type of checksum error detection used for the port.<br>**Attention**<br>• Do not change the default value *NONE*. The Omni CRC is implemented internally by the Omni interface, and *not* through these Checksum options.<br>• The Omni Flow Computer port must be configured with CRC enabled. |

### LANVendor port properties

**Attention**
LANVendor port properties are only applicable to an Omni Flow Computer connected using the Ethernet port of an Omni Serial/Ethernet-Modbus Mux Module. The port IP address(es) are configured in the Omni Controller.

| Property | Description |
|---|---|
| Protocol | Modbus Protocol to use when communicating with a device on this channel. Select one of the following:<br>• Encapsulated Modbus<br>• Modbus/TCP<br>**Attention**<br>Select Encapsulated Modbus or Modbus/TCP as Protocol for LANVendor connections and then configure the IP address and Port in the Omni controller's Main tab. |

### Related topics

“Redundant port properties for an Omni channel” on page 21

# Redundant port properties for an Omni channel

Redundant port properties are identical to the Omni channel's port properties.

**Attention**
Ensure that you select the same Port Type in the Redundant Port tab, as the Port tab.

### Related topics

“Port properties for an Omni channel” on page 18

# Main properties for an Omni controller

Use the Main tab to define the basic properties for an Omni controller.

For information about how to create a controller, see the topic titled "Building controllers and channels" in the *Quick Builder User's Guide*.

| Property | Description |
|---|---|
| Name | The unique name of the controller. A maximum of *10* alphanumeric characters (no spaces or double quotes). Note: In Station displays, underscore characters ( _ ) appear as spaces.<br>For LAN–connected controllers, the name is used to look up the IP address in the TCP/IP database if you do not specify an **IP Address** property. |
| Description | (Optional) A description of the controller. A maximum of *132* alphanumeric characters, including spaces. |
| Channel Name | The name of the channel on which the controller communicates with the server.<br>(You must have already defined a channel for it to appear in this list.) |
| IP Address 1 & IP Address 2 | **Attention**<br>**IP Address 1** and **IP Address 2** are only applicable if the **Port Type** of the associated channel is LANVendor.<br>**IP Address 2** is used only when the device has a redundant network adapter card and the Channel that the Controller has been built on has been defined as a redundant channel.<br>Network address(es) of the Omni Flow computer.<br>If the IP Address is left as 0.0.0.0, the Controller name is used as the hostname. For further information see the Name property. |
| Port No. | **Attention**<br>This field is only applicable if the **Port Type** of the associated channel is LANVendor. If the Port number is set to 0, it defaults to the Modbus/TCP default of 502.<br>The TCP Port number for communicating with the Ethernet port of the Omni Serial/ Ethernet-Modbus Mux Module. |
| Marginal Alarm Limit | The communications alarm marginal limit at which the controller is declared to be marginal. When this limit is reached, a high priority alarm is generated. To change the priority of the alarm system wide, see the topic titled "Configuring system alarm priorities" in the *Server and Client Configuration Guide*. To change the priority of the alarm for one controller, see the topic titled "About configuring custom system alarm priorities for an individual channel or controller" in the *Server and Client Configuration Guide*.<br>A controller barometer monitors the total number of requests and the number of times the controller did not respond or response was incorrect. The barometer increments by two or more, depending on the error, and decrements for each good call.<br>The default value is *25*. |
| Fail Alarm Limit | The communications alarm fail limit at which the controller is declared to have failed. When this barometer limit is reached, an urgent alarm is generated. To change the priority of the alarm system wide, see the topic titled "Configuring system alarm priorities" in the *Server and Client Configuration Guide*. To change the priority of the alarm for one controller, see the topic titled "About configuring custom system alarm priorities for an individual channel or controller" in the *Server and Client Configuration Guide*.<br>Set this to double the value specified for the controller Marginal Alarm Limit.<br>The default is *50*. |

| Property | Description |
|---|---|
| Dynamic Scanning<br>Fastest Scan Period | Select the **Dynamic Scanning** check box to enable dynamic scanning of all point parameters on this controller. The default setting for this check box is selected.<br>Define the fastest possible scan period (in seconds) that dynamic scanning will scan point parameters on this controller. The default is *15* seconds.<br>The dynamic scanning period does not affect the static scanning rate for a parameter. For example, if the scanning rate for a parameter is 10 seconds, and the dynamic scanning rate for the controller is 15 seconds, the parameter will still be scanned at a period of 10 seconds. |
| Protocol | Displays the protocol of the channel used by the controller. |
| Modbus ID | The Modbus ID of the device and port for communications. See the device documentation for details on configuring this parameter in the device. |
| Offset | There is a limit of 32,767 in the address range for each controller. If a range of addresses larger than this range is needed, additional controllers will be needed with different offsets. Current Omni devices do not use addresses above 18,999 so a single controller with offset of zero is sufficient.<br>Offset is less than the lowest register of the device that you want to read with this controller. Offset + 1 is the first valid register for this controller.<br>The default value is *0*. Therefore, by default, the first valid register is *00001*.<br>To use an address range between 32,767 and 65,535, you must specify a negative offset. You can derive the offset by using the formula:<br>*hw address = 65535 + offset + specified address*<br>The diagnostic address determined by the Offset value is overridden if the Diagnostic Address is specified. |
| Item Type | The type of controller specified when this item was created. |
| Last Modified | The date and time the controller properties were modified. |
| Last Downloaded | The date and time the controller was last downloaded to the server. |
| Item Number | The unique item number currently assigned to this controller, in the format *RTUnnnnn*.<br>You can change the item number if you need to match your current server database configuration. The number must be between *01* and the maximum number of controllers allowed for your system. For more information about setting the maximum value, see the topic titled "Adjusting sizing of non-licensed items" in the *Supplementary Installation Tasks Guide*. |

# Optimizing Omni scanning performance

The maximum amount of data that can be acquired from an Omni controller is influenced by the rate of sending scan packets to the controller. An understanding of Omni scan packets can help you configure points so that optimal data acquisition performance is achieved, by maximizing the amount of data acquired with each scan packet.

Communications with an Omni Flow Computer are restricted by real-time processing in the Omni. This limits the Omni to about five packets per second on each channel. If your system is approaching this limit, any disruption due to communication errors or time-outs will likely result in a channel overload condition and eventually to loss of scan packets.

The scan packets that have been built can be listed by using the utility **lisscn** (list scan). Listing scan packets helps verify the scanning strategy.

For more information about **lisscn**, see the section titled "Command Reference" in the *Server and Client Configuration Guide*.

### Related topics

“Data tables, addresses, and files supported by Omni” on page 8

# Synchronizing the Omni Flow Computer time with the server time

### To synchronize the Omni Flow Computer's time with the server time

1 On the primary server, open a Command Prompt window.

2 Type **omntst** to start the Omni configuration test utility.
The message **Enter the LRN or the device name of channel** appears.

3 Type the channel name or number.
For example, **CHN21** for *channel 21*.

The message **Ready to test Omni on channel** *nn*, **lrn** *nn* appears.

4 Type **delay** *ControllerNum*, *seconds*, where
*ControllerNum* is the server controller (RTU) number connected to the Omni.

*seconds* is the number of seconds past midnight (*1-86340*) each day to synchronize the clocks.

- A value of *-1* (minus one) indicates synchronize once and disable further synchronization.
- A value of *0* (zero) disables time synchronization.

For example, to set RTU 36 to synchronize its time with the server at 10 seconds past midnight, type:

```
delay 36, 10
```

The following messages appear:

**The sync delay for RTU** *ControllerNum* **was 0.00 seconds past midnight**

**The sync delay for RTU** *ControllerNum* **is now** *n*.**00 seconds past midnight**

5 Type **q** to quit **omntst**.

> **Attention**
> If a non-zero delay is configured, the time synchronization delay is checked on each diagnostic scan. If the synchronization is due and has not been executed today or if the controller is recovering from a communications failure, the times are synchronized by writing the server time and date to registers *3867-3872*. When the time synchronization occurs, an event is created to log the time change.

### Examples

The following examples demonstrate usage of the **delay** command.

| Command | Result |
|---|---|
| **delay 13,0** | Disables synchronization for controller **13**. |
| **delay 23,-1** | Synchronizes controller **23** once only and then disables further synchronization. |
| **delay 32,11400** | Enables synchronization for controller **32** daily at 3:10 AM. |
| **delay 5** | Prints the delay for controller **5**. |

### Related topics

"About Omni utilities" on page 11

# Configuring the Omni password

The Omni Password feature is required for Omni Flow Computer auditing of changes made through the SCADA interface. This provides an audit trail, which is required in some custody transfer applications. For more information about the Omni Flow Computer register 3800, see the Omni Flow Computer documentation.

If the Omni Flow Computer port is configured for auditing, the password will be required for data acquisition. If the password is incorrect or missing, any data request will fail and generate the following message:

```
Modbus Exception: 5: PASSWORD NEEDED
```

If the Omni Password is configured, it is tested with each diagnostic scan. If the diagnostic scan fails with the `PASSWORD NEEDED` exception code, the password is transmitted to the Omni to enable communication.

### To configure the Omni password

1. On the primary server, open a **Command Prompt** window.
2. Type **omntst** to start the Omni configuration utility.
   The message **Enter the LRN or the device name of channel** appears.
3. Type the channel name or number.
   For example, **CHN21** for *channel 21*.

   The message **Ready to test Omni on channel** *nn*, **lrn** *nn* appears.
4. Type **pass** *ControllerNum*, *password*, where
   *ControllerNum* is the server controller (RTU) number connected to the Omni.

   *password* is the Omni port password.

   For example, to set RTU 21 password to `OMNI`, type:

   **pass 21, OMNI**

   > **Attention**
   > The password can be up to eight characters in length and can contain spaces.
   >
   > A password with all spaces is equivalent to no password. Because the password can contain spaces, there is no space between the comma and the password.
5. Type **q** to quit **omntst**.

### To remove the Omni password

1. On the primary server, open a **Command Prompt** window.
2. Type **omntst** to start the Omni configuration utility.
   The message **Enter the LRN or the device name of channel** appears.
3. Type the channel name or number.
   For example, **CHN21** for *channel 21*.

   The message **Ready to test Omni on channel** *nn*, **lrn** *nn* appears.
4. Type **pass** *ControllerNum*, where
   *ControllerNum* is the server controller (RTU) number connected to the Omni.

   For example, to remove the password from RTU 21, type:

   **pass 21**
5. Type **q** to quit **omntst**.

**Related topics**

“About Omni utilities” on page 11

# Omni points reference

This section describes how to configure points for an Omni controller using Quick Builder.

In addition to the information contained in this reference, and for help to build points, see the section titled "Building and configuring points" in the *Quick Builder User's Guide*.

### Related topics

“Address syntax for Omni controllers” on page 30
“Planning considerations for installing and configuring Omni Flow Computers” on page 5

# Address syntax for Omni controllers

### Omni point address formats

Omni point addresses are entered with an address and format. The following table provides some address syntax examples.

**Table 3: Address syntax examples for Omni controllers**

| Data Type | Address example |
|---|---|
| Boolean | *ControllerName OmniAddress* **0** |
| 16-bit integer | *ControllerName OmniAddress* **0** |
| 16-bit integer bit field | *ControllerName OmniAddress* **bit width** |
| 32-bit integer | *ControllerName OmniAddress* **U32B** |
| Floating point | *ControllerName OmniAddress* **IEEEFP** |

### Related topics

“Data tables, addresses, and files supported by Omni” on page 8

## Omni scan packets

The Omni protocol is based on Modbus and retains the Modbus limitation of 255 bytes per data packet. This restriction limits the number of registers in a poll message as follows.

**Table 4: Maximum registers in a poll message**

| Address | Data Type | Maximum registers in a poll message |
|---|---|---|
| 1000–2999 | Boolean | 1,952 (122 × 16) |
| 3000–3999 | 16-bit integer | 122 |
| 5000–5999 | 32-bit integer | 61 |
| 6000–8999 | 32-bit floating point | 61 |
| 13000–13999 | 16-bit integer | 122 |
| 15000–15999 | 32-bit integer | 61 |
| 17000–17999 | 32-bit floating point | 61 |

The Omni also has some century boundary (*xxx00*) addresses that are not implemented. If these addresses occur within a requested data packet, zero values are returned for addresses above the *Spare* or *Reserved* address. This requires a modification of the packet optimization scheme to force separate packets to be requested around century boundaries.

The system also generates separate packets any time the scan rate changes. For example, requesting registers 7101–7104 (meter 1 flow rates) at 10 seconds and registers 7105–7106 (temperature and pressure) at 15 seconds, and registers 7118–7128 (batch in progress values) at 30 seconds generates three separate packets. Changing all of these to 10 seconds generates a single packet and reduces the loading on the channel.

# Troubleshooting Omni issues

This section describes troubleshooting tasks for Omni that you can perform either on the server or from any Station.

### Related topics

“Isolating Omni interface communication issues” on page 32
“Isolating Omni interface configuration issues” on page 34

# Isolating Omni interface communication issues

Use the following checklist to isolate and troubleshoot Omni communication issues.

| Task | Done |
|---|---|
| Check the system status display's Status pane for channel and controller statistics. | |
| Check the SCADA Controllers display to ensure that the channel status is Green. This indicates that the channel is enabled and is communicating with its associated device(s).<br>In addition, check that the Error statistics is under control. | |
| Check the SCADA Controllers display to ensure that the status of the controller is Green. This indicates that the device is communicating with the server.<br>In addition, check that the Error statistics is under control. | |
| Run the server diagnostic tools for detecting communication errors between the server and the device. | |
| Check the server log for communication errors. | |
| Check the event log for communication errors. | |

### Related topics

“Testing Omni communications with and the server” on page 32

## Testing Omni communications with and the server

You use the Omni test utility, **omntst**, to test communications between the server and the Omni controller after you have downloaded channel and controller definitions to the server database.

### Prerequisites

- Set up the controller.
- Connect all cables.
- Define the controller and channel in Quick Builder.
- Download the Quick Builder definitions to the server, without errors.
- Ensure the channel is out of service.

### To run the omntst test utility

1 On the primary server, open a Command Prompt window.

2 Type **omntst** to start the Omni configuration test utility.
The message **Enter the LRN or the device name of channel** appears.

3 Type the channel name or number.
For example, **CHN21** for *channel 21*.

The message **Ready to test Omni on channel** *nn*, **lrn** *nn* appears.

4 Type **read** *ControllerNum*, **5848**.
If the **omntst** is able to communicate with the Omni on the selected controller and channel, it returns the current Omni time in the *HHMMSS* format.

This indicates that the Omni Flow Computer is able to successfully communicate with the server. Otherwise, a message is printed indicating the error encountered.

### Related topics

“Planning considerations for installing and configuring Omni Flow Computers” on page 5

“About Omni utilities” on page 11

# Isolating Omni interface configuration issues

Use the following checklist to isolate Omni configuration issues.

| Task | Done |
|---|---|
| Check the server log for point configuration and scanning errors. | |
| Check the Event log for point configuration and scanning errors. | |
| Check the Quick Builder output file for point configuration errors. | |
| Check the Channel Scanning Statistics Display (Page 10 display) to view the health of the scanning subsystem. | |
| Use the **lisscn** diagnostic command to diagnose scanning issues. | |

# Notices

### Trademarks

Experion®, PlantScape®, SafeBrowse®, TotalPlant®, and TDC 3000® are registered trademarks of Honeywell International, Inc.

OneWireless™ is a trademark of Honeywell International, Inc.

### Other trademarks

Microsoft and SQL Server are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

Trademarks that appear in this document are used only to the benefit of the trademark owner, with no intention of trademark infringement.

### Third-party licenses

This product may contain or be derived from materials, including software, of third parties. The third party materials may be subject to licenses, notices, restrictions and obligations imposed by the licensor. The licenses, notices, restrictions and obligations, if any, may be found in the materials accompanying the product, in the documents or files accompanying such third party materials, in a file named third_party_licenses on the media containing the product, or at http://www.honeywell.com/ps/thirdpartylicenses.

# Documentation feedback

You can find the most up-to-date documents on the Honeywell Process Solutions support website at:

http://www.honeywellprocess.com/support

If you have comments about Honeywell Process Solutions documentation, send your feedback to:

hpsdocs@honeywell.com

Use this email address to provide feedback, or to report errors and omissions in the documentation. For immediate help with a technical problem, contact your local Honeywell Process Solutions Customer Contact Center (CCC) or Honeywell Technical Assistance Center (TAC) listed in the “Support and other contacts” section of this document.

# How to report a security vulnerability

For the purpose of submission, a security vulnerability is defined as a software defect or weakness that can be exploited to reduce the operational or security capabilities of the software.

Honeywell investigates all reports of security vulnerabilities affecting Honeywell products and services.

To report a potential security vulnerability against any Honeywell product, please follow the instructions at:

https://honeywell.com/pages/vulnerabilityreporting.aspx

Submit the requested information to Honeywell using one of the following methods:

- Send an email to security@honeywell.com.

  or

- Contact your local Honeywell Process Solutions Customer Contact Center (CCC) or Honeywell Technical Assistance Center (TAC) listed in the “Support and other contacts” section of this document.

# Support

For support, contact your local Honeywell Process Solutions Customer Contact Center (CCC). To find your local CCC visit the website, https://www.honeywellprocess.com/en-US/contact-us/customer-support-contacts/Pages/default.aspx.

# Training classes

Honeywell holds technical training classes on Experion PKS. These classes are taught by experts in the field of process control systems. For more information about these classes, contact your Honeywell representative, or see http://www.automationcollege.com.

# Index

## P



## R



## S



## T



## U



## W



## X