Users Guide

Shop online at

omega.com®

ΩΞOMEGA®

www.omega.com
e-mail: info@omega.com

OMEGASOFT®

OMWT Series
Wireless Transmitters and Receivers

Software:
Receiver Software
DDE Software
OPC Software
Versions:
OMWT-SOFT
version 2.1
OMWT-SOFT-DDE
version 2.2c
OMWT-SOFT-OPC
version 2.2c

Manuals:
M-4064
M-4065
M-4066

Product
Documentation

ΩΞ OMEGA
ENGINEERING, INC.
www.omega.com
800-DAS-IEEE (800-327-4333)
© COPYRIGHT 2001 OMEGA ENGINEERING, INC.

OWMT-SOFT-OPC

Version 2.2
Copyright ©2004, Omega Engineering, Inc.

M-4066

OMEGAnet® Online Service
www.omega.com

Internet e-mail
info@omega.com

Servicing North America:

USA: One Omega Drive, Box 4047
ISO 9001 Certified   Stamford CT 06907-0047
Tel: (203) 359-1660   FAX: (203) 359-7700
e-mail: info@omega.com

Canada: 976 Bergar
Laval (Quebec) H7L 5A1, Canada
Tel: (514) 856-6928   FAX: (514) 856-6886
e-mail: info@omega.ca

For immediate technical or application assistance:

USA and Canada: Sales Service: 1-800-826-6342 / 1-800-TC-OMEGA®
Customer Service: 1-800-622-2378 / 1-800-622-BEST®
Engineering Service: 1-800-872-9436 / 1-800-USA-WHEN®
TELEX: 996404   EASYLINK: 62968934   CABLE: OMEGA

Mexico: En Espan˜ol: (001) 203-359-7803   e-mail: espanol@omega.com
FAX: (001) 203-359-7807   info@omega.com.mx

Servicing Europe:

Benelux: Postbus 8034, 1180 LA Amstelveen, The Netherlands
Tel: +31 (0)20 3472121   FAX: +31 (0)20 6434643
Toll Free in Benelux: 0800 0993344
e-mail: sales@omegaeng.nl

Czech Republic: Frystatska 184, 733 01 Karvina´, Czech Republic
Tel: +420 (0)59 6311899   FAX: +420 (0)59 6311114
Toll Free: 0800-1-66342   e-mail: info@omegashop.cz

France: 11, rue Jacques Cartier, 78280 Guyancourt, France
Tel: +33 (0)1 61 37 2900   FAX: +33 (0)1 30 57 5427
Toll Free in France: 0800 466 342
e-mail: sales@omega.fr

Germany/Austria: Daimlerstrasse 26, D-75392 Deckenpfronn, Germany
Tel: +49 (0)7056 9398-0   FAX: +49 (0)7056 9398-29
Toll Free in Germany: 0800 639 7678
e-mail: info@omega.de

United Kingdom: One Omega Drive, River Bend Technology Centre
ISO 9002 Certified   Northbank, Irlam, Manchester
M44 5BD United Kingdom
Tel: +44 (0)161 777 6611   FAX: +44 (0)161 777 6622
Toll Free in United Kingdom: 0800-488-488
e-mail: sales@omega.co.uk

It is the policy of OMEGA to comply with all worldwide safety and EMC/EMI regulations that apply.
OMEGA is constantly pursuing certification of its products to the European New Approach Directives.
OMEGA will add the CE mark to every appropriate device upon certification.
The information contained in this document is believed to be correct, but OMEGA Engineering, Inc. accepts
no liability for any errors it contains, and reserves the right to alter specifications without notice.
WARNING: These products are not designed for use in, and should not be used for, human applications.

Contents

Overview

Quick Start

When OMWT-SOFT-OPC is run for the first time it displays this help screen. You may view this help screen again by selecting "Help" from OMWT-SOFT-OPC's main window.

OMWT-SOFT-OPC is designed to gather data from sensors. OMWT-SOFT-OPC makes available the data it receives from each sensor through OPC tags.

The main OMWT-SOFT-OPC Server window will show the total number of devices in the startup list and how many of those that are online. The offline devices will be shown in the list box at the bottom of the window.

OMWT-SOFT-OPC can be a stand-alone data logger. When data logging is enabled, OMWT-SOFT-OPC will store collected data in an ASCII file at a programmable interval. See section "OMWT-SOFT-OPC Server Data Logging".

This Help and all other Help topics may be accessed from the main window by selecting Help or pressing F1.

To Get Started:

- Plug the reciever into your serial port

- Start OMWT-SOFT-OPC.

- If necessary, tell OMWT-SOFT-OPC what kind of Receiver you are using.

- When the "Add Wireless Device" screen appears, press the service mode button on your transmitter to transmit a packet that OMWT-SOFT-OPC can see.

- When a device appears in the window, you may click to select it and then press the 'Edit' button to change its label name and transmit time.

- Press OK to add these sensors to the OMWT-SOFT-OPC list

- OMWT-SOFT-OPC will create OPC tag objects for each device and start polling and gathering data from each device.

If there is not an **OMWT-SOFT-OPC.ini** file or if [Server]NumberDevices=0 and when OMWT-SOFT-OPC starts, it will automatically search for all devices on the network and add them to its list of devices to process.

Description

OMWT-SOFT-OPC is a data acquisition OLE for Process Control (OPC) server that acquires data from devices and passes this data using OPC to a client application. OMWT-SOFT-OPC waits for the device to transmit and then processes the transmission.

 Any Windows® application that can use OPC can obtain this data from the OMWT-SOFT-OPCServer. Examples of such applications include Wonderware® Intouch, National Instruments® LookOut, Capital Equipment Corp.® TestPoint, and general development applications like Microsoft® Visual Basic and Borland® Delphi.

OMWT-SOFT-OPC can be a stand-alone data logger. When data logging is enabled, OMWT-SOFT-OPC will also store collected data in an ASCII file at a programmable interval. Another program such as a spreadsheet or a database manager can import the data.   See section "OMWT-SOFT-OPC Server Data Logging".

OMWT-SOFT-OPC Server and Wireless

OMWT-SOFT-OPC Server can receive transmissions from wireless devices such as the wireless temperature transmitter. Other devices include the wireless humidity sensor and wireless analog input. OMWT-SOFT-OPC uses a wireless receiver such as the OMWT-REC232-600, OMWT-XREC-SER and OMWT-RECIP to receive transmissions from wireless sensors. OMWT-SOFT-OPC can also use multiple OMWT-RECIP Receivers to receive data through a TCP/IP connection.

When setting up a wireless network, OMWT-SOFT-OPC passively waits for transmissions instead of actively polling for devices. This can make the acquisition of sensors take a longer time than for a wired network as OMWT-SOFT-OPC must wait for the device to transmit. You can speed up this process by pressing the service button on the device so that it transmits while you are in the OMWT-SOFT-OPC Wireless Setup screen.

Each device has an entry in the device's INI file section. The INI entry and OPC tag **"polltime"** for each device has a different meaning when working with wireless transmitters. OMWT-SOFT-OPC cannot poll the wireless sensors. It must wait for a transmission.   Therefore, OMWT-SOFT-OPC uses polltime to estimate how frequently a transmission should come from a device. It also uses a multiple of this time to determine if a device is offline.   This multiple is configurable by the user.

Wireless Setup

If you have OMWT-SOFT-OPC running with a wireless receiver such as an OMWT-REC232-600 or OMWT-XREC-SER, OMWT-SOFT-OPC will display the Setup screen when it starts for the first time. The setup screen is the method by which OMWT-SOFT-OPC acquires wireless devices.   The default on the setup screen is to accept only packets that are sent in service mode from the device (the packet is sent by pressing the service button on the device.)   To change this default, uncheck the "Service Mode Only" checkbox. OMWT-SOFT-OPC will then display all the packets it receives. When OMWT-SOFT-OPC receives a packet from a device new to it, it names the device following the naming conventions detailed below, and it displays the device's name and serial number in the window.

You have several options after the device has been displayed. The "Clear New" button will clear all the devices that have just been received. The "Delete" button will delete a single device. Click on a device to select it and click on "Delete" to

delete it.   To edit a device click on the device to select it and click on "Edit" to edit the device. There is also a "Stop" button. If you click that button, OMWT-SOFT-OPC will stop listing devices in the window. When you click on the "Stop" button it becomes a "Receive" button. If you click on that, OMWT-SOFT-OPC will start receiving packets and listing devices again.

When you are ready, hit "OK", or "Cancel" to exit.

Edit Device

The Edit screen for a new wireless device shows the device's serial number, which the user cannot change, the device's current Label, which can be changed by the user, and the Transmit Rate for the device. The Label cannot be the same as the label used by any current device. The Transmit Rate is in seconds. The default value is 60.   OMWT-SOFT-OPC uses the Transmit Rate to determine if a device has gone offline.

Receivers

The following is a list of supported receivers:

- OMWT-REC232-600
- OMWT-XREC-SER (at 19200 baud)
- OMWT-RECIP
- OMWT-REPEAT

Device List

OWMT-SOFT-OPC Server interfaces to a large number of devices. New devices are continually being created for wireless network. Check with your distributor for the latest.

The following is a list of devices supported by OWMT-SOFT-OPC Server:

OMWT-TEMP15	Temperature
OMWT-TEMPRH	Temperature/Humidity
OMWT-AIN5V	Analog Input 5V
OMWT-AIN10V	Analog Input 10V
OMWT-AIN20MA	Analog Input 20 milliamp
OMWT-VIB	Vibration
OMWT-MOT	Motion
OMWT-IRCNT	IR Counter
OMWT-THERM	Thermisitor
OMWT-DOORCNT	Door Counter with Temperature
OMWT-PULSECNT	Pulsse Counter with Temperature
OMWT-TEMPALRM	Temperature with Alarm
OMWT-ACCESSREADER	Access/Control Reader
OMWT-DIRCNT	Directional Counter
OMWT-DUALOUT	Dual Discrete Output
OMWT-ANGOUT	Analog Output

Polling

For a wireless device the polltime has a slightly different meaning. OWMT-SOFT-OPC does not actively poll wireless devices. Rather, it waits to receive a packet from the devices. Therefore, the polltime is how often OWMT-SOFT-OPC expects to receive a packet from the wireless devices.

The polltime for a wireless device is configurable during device setup. OWMT-SOFT-OPC will use this time to determine when to mark a device offline. OWMT-SOFT-OPC has a INI file parameter called "Tries" that sets how long a device has to transmit before it is considered offline. For example, if the device has a polltime of 1 minute, and OWMT-SOFT-OPC has "tries" set to 3, then OWMT-SOFT-OPC will consider the device offline if it does not receive a packet from that device in 3 minutes ("tries" times "polltime").

Identification

OWMT-SOFT-OPC identifies each device with a unique label name. A client application uses this label (a OPC topic) to obtain data from the server.

Filtering

For both analog and temperature devices, OMWT-SOFT-OPC allows the user to define a deadband to control the amount of change that must occur before the server updates a Client with new data. Deadband is useful to control the amount of information that is passed through OPC to the client application. For analog devices, the deadband is entered as the percentage of full scale of Engineering units. For temperature devices, the deadband is entered in degrees C.

Also for both analog and temperature devices, OMWT-SOFT-OPC provides the means to filter the data with either an Average, Median or combination Median Average filter before either logging data or delivering data through OPC. In the OMWT-SOFT-OPC INI file, you specify the filter type and the number of samples to filter.

What is OPC?

OLE for Process Control (OPC) is a specific specification of Microsoft's Object Linking and EmbeOPCding (OLE) and Component Object Model (COM) that is administered by the OPC Foundation (150 member companies including Microsoft Corporation). OPC provides a common interface for devices and applications to communicate. Devices that gather or originate data become OPC servers, providing data to OPC client applications in a consistent fashion. At the heart of OPC is the client/server model in which the OPC server provides and interfaces to the OPC object, allowing client applications to control devices and manage device data in a generic fashion. OPC servers can be accessed through most HMI software and through a variety of languages, including C++, Visual Basic, and Delphi. The OPC server collects data from physical devices for distribution to OPC client applications and also has the responsibility for updating device data when an OPC client issues a write.

An OPC client connects to and communicates with an OPC server through one of two interfaces defined in the OPC specification. The OLE automation interface allows users to access data through the use of a common, easily understood scripting language. The COM interface is more complex, but provides finer levels of control and flexibility for developers and users proficient in languages such as C or C++. OPC client applications can access OPC servers locally within a computer or from a networked server.

Data Logging

OMWT-SOFT-OPC Server can log data to an ASCII file at a programmable interval. The default name of the file is OMWT-SOFT-OPC.LOG.

A client application can control logging through OPC variables. The client can start/stop logging or just log a single record.

For diagnostic purposes, OMWT-SOFT-OPC can log errors that have occurred while waiting or process device data.

For more information see the section "OMWT-SOFT-OPC Server Data Logging".

Diagnostics

With OMWT-SOFT-OPC, you can monitor the health of your wireless network and the devices attached. OMWT-SOFT-OPC reports, through OPC, errors that it encounters while polling devices. These errors can be noted in an event error log. See the section called "Server Item Names" for more information.

OMWT-SOFT-OPC Main Window

Main Window

OMWT-SOFT-OPC displays the number of devices online and offline. It also lists the devices that are currently offline displaying the name and serial number of the device.   If OMWT-SOFT-OPC is running with TCP/IP receivers, then for each offline device it will display the node address, name and serial number of the device.

Setup

OWMT-SOFT-OPC simplifies setup by automatically finding, identifying and beginning to poll devices in a network. OWMT-SOFT-OPC assumes default initial values. You can easily change these values by editing the OWMT-SOFT-OPC.INI file with an ASCII text editor. (The installation program for OWMT-SOFT-OPC Server created a short cut to the OWMT-SOFT-OPC.INI file. Double clicking will automatically start WordPad or Notepad.) See the section "OWMT-SOFT-OPC Server .INI File Format".

Change Port

The Change Port submenu allows you to switch to another wireless receiver. When the Change Port submenu is selected, OWMT-SOFT-OPC displays the Select Communication Port window. Choose one of the following:

> **AUTO** - instructs OWMT-SOFT-OPC to search Port Numbers 1 to 4 for the receiver.
>
> **OMWT-REC232-600** – receives RF (wireless) packets
>
> **OMWT-XREC-SER** – receives RF (wireless) packets (set at 19200 baud)
>
> **OMWT-RECIP** – uses a receiver that uses the TCP/IP protocol to receive sensor packet data.

When OK is clicked, OWMT-SOFT-OPC tries to initialize the port. For the OMWT-RECIP, OWMT-SOFT-OPC confirms that the receiver is present. For the OMWT-REC232-600 and OMWT-XREC-SER, OWMT-SOFT-OPC assumes the receiver is present. OMWT-XREC-SER must be set to 19200 baud. If OWMT-SOFT-OPC fails, OWMT-SOFT-OPC displays the OWMT-SOFT-OPC Initializing Error Window allowing you to retry, select another port or exit OWMT-SOFT-OPC. If OWMT-SOFT-OPC succeeds, OWMT-SOFT-OPC starts listening for sensor packets and processes the packet according to the device list in the INI file.

Add/Delete IP Connection

(OMWT-RECIP Receiver only)

OWMT-SOFT-OPC will display the "Add/Delete IP Connection" screen. IP addresses already setup will be displayed. The following is a list of the columns:

Node – Node number for the IP address. OWMT-SOFT-OPC uses this number to identify this IP connection. OWMT-SOFT-OPC uses this number to update the OPC item "nodaladdress".

IP Address – the IP address that is used to address the receiver.

Port – the port number that is used along with the IP address to address the receiver.

Conn – identifies who initiated the connection. Outgoing: OWMT-SOFT-OPC initiated the connection with the receiver. Incoming: the receiver initiated the connection with OWMT-SOFT-OPC.

Password – initial password used to gain access to the receiver.

Description – a general description field associated with the IP connection.

The following is list of buttons in the "Add/Delete IP Connection" screen:

Test – OWMT-SOFT-OPC will try to make a connection to this IP address and report back success or failure.

Add New – OWMT-SOFT-OPC displays the "New IP Address" screen. The following is list of the fields to enter to add a new IP connection:

IP Address – the IP address that is used to address the receiver

Port – the port number that is used along with the IP address to address the receiver.

Description– a general description field associated with the IP connection.

Password – initial password used to gain access to the receiver. If the receiver does not have a password set or does not require a password then leave this field blank.

Connection – identifies whether OWMT-SOFT-OPC will initiate the connection or whether the receiver will initiate the connection. Outgoing: OWMT-SOFT-OPC initiates the connection. Incoming: OWMT-SOFT-OPC waits for the receiver to initiate the connection.

Edit – OWMT-SOFT-OPC displays the "Edit" screen showing the selected IP connection. The "Edit" screen has the same fields as the "New IP Address" screen.

Delete – delete the selected IP connection.

View IP Connection Status

(OMWT-RECIP Receiver only)

OWMT-SOFT-OPC will display the "IP Status" screen. OWMT-SOFT-OPC shows the current state of the all the connections. The following describes the columns:

Node – Node number for the IP address. OWMT-SOFT-OPC uses this number to identify this IP connection. OWMT-SOFT-OPC uses this number to update the OPC item "nodaladdress".

IP Address – the IP address that is used to address the receiver.

Port – the port number that is used along with the IP address to address the receiver.

Conn – identifies who initiated the connection. Outgoing: OWMT-SOFT-OPC initiated the connection with the receiver. Incoming: the receiver initiated the connection with OWMT-SOFT-OPC.

Description – a general description field associated with the IP connection.

Status – the current status of the connection. The following is a list of the possible status messages:

Never Connected – OWMT-SOFT-OPC never attempted or received a connection to the receiver

IP Connected – OWMT-SOFT-OPC is currently connected to the receiver but has not logged in.

Connected – OWMT-SOFT-OPC is currently connected to the receiver is waiting to receive sensor packets.

Disconnected – OWMT-SOFT-OPC was connected but is now disconnected from the receiver. If the connection was initiated by OWMT-SOFT-OPC, OWMT-SOFT-OPC will continually try to reestablish the connection.

TCP/IP Error – a TCP/IP error was encounter and OWMT-SOFT-OPC is not connected to the receiver.

No IP Connection – OWMT-SOFT-OPC tried to establish a connection but failed.

Wrong Password – The receiver requires a login password and the incorrect password was used by OWMT-SOFT-OPC.

Outputs

Outputs are only send packets. The Output devices do not send packets. Therefore the setup for an Output sensor must be done manually. Some Outputs need to be "trained" to the packets that it will receive for from the Server.

New and Edit

Create or edit an Output object to manage an Output sensor. The following describe the parameters necessary to set up an Output sensor.

Type – "Dual Discrete Outputs" or "12 bit Analog Output". Select the type of Output sensor. Note changeable only for the Edit Output window.

Serial Number – Some Outputs sensors can "learn" a serial number. Others have the serial number preassigned and must be entered in the field. These sensors have the serial number labeled. For those sensors that must learn their serial number, OWMT-SOFT-OPC creates a default serial number. Press the "Generate New Serial Number" button to generate a new serial number.

Label – name assigned for this Output sensor. The label name is used for the OPC topic name. This name must be unique. For the Edit Window, the Label field contains a list box. Select the Output to modify from the list. You can also change the label for the current Output.

Node – This field applies only when using the OMWT-RECIP Receiver. The Node List box contains a list of the OMWT-RECIP Receivers to send the output packet to. For the"Send All" selection, OWMT-SOFT-OPC sends the packet to all the OMWT-RECIP Receivers.

Dual Discrete Outputs

Send when output B is written – If checked, OWMT-SOFT-OPC will only send the output packet only when the "outputB" item is written to (OPC poke). Output A is first written to and the output B. If unchecked, OWMT-SOFT-OPC will send the output packet when either the "outputA" or "outputB" items are written to (OPC poke). The state of both outputs is sent in the same packet.

12 bit Analog Output

Scale – the scale to apply to the raw reading (12 bits: 0 to 4095).

Offset – the offset to apply to the reading after the scale is applied. Engeering Value = Scale * raw + Offset.

Units – units label for the analog output.

Train

The Train dialog is used in the processing of training the Output device to output packets (as produced by OWMT-SOFT-OPC). Follow these steps to train an Output device:

1. Select the desired Output from the list of Outputs.

2. Press and hold the button on the Output device (label as "train").

3. Click the button on the Train window. Repeat until the LED on the Output device is on.

The Output device is now ready to receive Output packets from OWMT-SOFT-OPC.

Search and Add Devices

OWMT-SOFT-OPC will display the "Add Wireless Device" screen. Devices already in the OWMT-SOFT-OPC.ini file are displayed in the top portion of the screen. Follow the setup procedure to add more devices.

The setup screen is the method by which OWMT-SOFT-OPC acquires wireless devices. The default on the setup screen is to accept only packets that are sent in service mode from the device (the packet is sent by pressing the service button on the device.) To change this default, uncheck the "Service Mode Only" checkbox. OWMT-SOFT-OPC will then display all the packets it receives. When OWMT-SOFT-OPC receives a packet from a new device, it names the device following the naming conventions detailed below, and it displays the device's name and serial number in the window.

You have several options after the device has been displayed. The "Clear New" button will clear all the devices that have just been received. The "Delete" button will delete a single device. Click on a device to select it and click on "Delete" to delete it. To edit a device, click on the device to select it and click on "Edit" to edit the device. There is also a "Stop" button. If you click that button, OWMT-SOFT-OPC will stop listing devices in the window. When you click on the "Stop" button, it becomes a "Receive" button. If you click on that, OWMT-SOFT-OPC will start receiving packets and listing devices again.

When you are ready, hit "OK", or "Cancel" to exit.

Edit Device

The Edit screen for a new RF device shows the device's serial number, which the user cannot change, the device's current Label, which can be changed by the user, and the Transmit Rate for the device. The Label cannot be the same as the label used by any current device. The Transmit Rate is in seconds. The default value is 60. OWMT-SOFT-OPC uses the Transmit Rate to determine if a device has gone offline.

ReConfigure Net

OWMT-SOFT-OPC clears device information from the OWMT-SOFT-OPC.ini startup file and completes a SearchAdd to build a new device list as if starting for the first time. **Use caution here**; *all changes that you have made manually to the device information contained in the OWMT-SOFT-OPC.ini file will be deleted and overwritten*. OWMT-SOFT-OPC does not modify the global configuration parameters contained in the sections Server and Logging of the OWMT-SOFT-OPC.ini file.

Devices

Clicking this menu option shows the View Device List window. OMWT-SOFT-OPC shows a list of devices and their serial numbers.

To see the help for a device, click on the device and then click the help button. OMWT-SOFT-OPC will display the help for the selected device.

Errors

OMWT-SOFT-OPC displays the Communication Errors window. This window shows the last error that occurred.

Tries Count: Number of tries that have occurred before a device is considered offline.

Major Error Count: Number of errors that (after tries) took the device offline.

Last Error Code: Number error code of the last error

Last Error Device: The name of the device that had the last error.

Last Error Time: The time when the last error occurred.

Last Error Message: A description of the last error.

Click the Reset button clear the errors and reset the counts.

Quit

When you select this menu option, OMWT-SOFT-OPC terminates.

Help

You select the the Help file's contents or index window or you can display the OMWT-SOFT-OPC About window.

OMWT-SOFT-OPC Server .INI File Format

OMWT-SOFT-OPC.ini

The OMWT-SOFT-OPC.INI file contains the configuration parameters for the server and the devices being polled. When OMWT-SOFT-OPC searches and finds a new device it adds the device to the INI file and assigns default initial parameters. These parameters can be changed by using a text editor and editing the INI file, or by using the included OMWT-SOFT-OPC.ini Editing Utility, which will be discussed later in this Help. (The installation program for OMWT-SOFT-OPC Server created a short cut to the OMWT-SOFT-OPC.INI file.) Double clicking will automatically start WordPad or Notepad. The following rules must be kept when editing the INI file.

1) All device label names must be unique.

2) The NumberDevices in the [Server] section must equal the largest device section [Device*n*].

3) The must be no missing device sections [Device*n*] in the list.

Note: OMWT-SOFT-OPC looks at the ini file only at startup or after adding a new device. Changes made to the ini file will not be reflected in OMWT-SOFT-OPC until OMWT-SOFT-OPC is closed and restarted.

The following is a list of the OMWT-SOFT-OPC.INI parameters.

SERVER

[Server] - Section name.

PortType - Communication medium (AUTO, OMWT-232-600, etc.).

PortNo - Port number.

NumberDevices - Number of devices to be polled by the server.

OfflinePollTime - How often to poll for offline devices. Milliseconds

Tries - No. of tries during polling before a device is marked offline.

HostAdapterCheck - Enables/disables the detection of receiver. (0 or 1)

BaudRate –options are: 115,200; 38,400; 19,200; or 1200. The default is 19200. Use this parameter for receivers that use a different baud rate than the default. Make sure the value in the ini file matches the value set in the hardware.

DefaultTempUnits - specifies the starting units of Temperature devices

DefaultPressUnits - specifies the starting units of Pressure devices

ConfigureEnable – 0 – disables the Setup Menu option on the main menu. The user then cannot make any changes to the configuration. 1 – (default) – enable the Setup Menu option on the main menu.

SmallINI – suppresses the listing of minor variables in order to make the ini file smaller (useful for large networks since under Windows, an ini file may be a maximum of 64K).

DefaultThermistorCorrID – overrides the default thermistor Correction ID (normally 128). All thermistors will be assigned this Correction ID unless the ID is overriOPCn in the individual device section.

ForwardAll – species if all packets are forwarded or only packets associated with OMWT-SOFT-OPC's device list get forwarded across a TCP/IP connection. A TCP/IP client must initiate the connection to the port specified with the ForwardPort in the [IP Addresses] section. Each packet has the following format: *sensor packet,node number, node description, sensor label*.

Logging

[Logging] - section name

> **LogRate** - 0: no logging; 1-100000 logging interval in seconds. 5 sec or greater OMWT-SOFT-OPC opens and closes the log file at every logging interval. Less than 5 OMWT-SOFT-OPC leaves the file open until OMWT-SOFT-OPC is terminated or "Server.Logging.LogRate" is set to 0 or 5 and greater.

> **LogFile** - full pathname for the log file - default "OMWT-SOFT-OPC.LOG"

> **OfflineIndicator** - character or string to indicate offline - NULL: last value

> **LogErrors** - 1 - log runtime network errors to OMWT-SOFT-OPC.ERR. 0 - do not log errors

IP Addresses

[IP Addresses] – section name (for OMWT-RECIP receivers)

> **IPWaitPort** – port number that OWMT-SOFT-OPC will wait for connections initiated by an IP receiver. (default is 1060)

> **ForwardPort** – port number that OWMT-SOFT-OPC will wait for connections to forward received packets. (default is 0 which means disabled) OWMT-SOFT-OPC will forward received packets through this connection. Each packet has the following format: *sensor packet,node number, node description, sensor label*. The INI parameter FowardAll in the [Server] section controls what packets get sent.

DIGITAL I/O

[Device*n*] - Section name.

> **Address** – serial number of the sensor.

> **DeviceType** - Device type. (DeviceType=DIGITAL)

> **PollTime** - How often to poll the device. (ms)

> **Label** - Name used for the device (and Topic).

> **Log** - Enable/disable logging of device.

> **OutputOnDemand** - Output is initiated on demand or during poll.

> **DescriptionA** - General purpose description.

> **DescriptionB** - General purpose description.

TEMPERATURE Setup

(OMWT-TEMP15 and OMWT-THERM)

[Device*n*] - Section name.

> **Address** - – serial number of the sensor.
>
> **DeviceType** - Device type. (DeviceType=TEMP)
>
> **PollTime** - How often to poll the device. (ms)
>
> **Label** - Name used for the device (and Topic).
>
> **Log** - Enable/disable logging of device.
>
> **LogDecimalPlaces** - Number of decimals to create when logging.
>
> **Units** - See OPC tags list under input.units for description.
>
> **Cal1Raw** - Two point calibration point 1.
>
> **Cal2Raw** - Two point calibration point 2.
>
> **Cal1Engr** - Two point calibration engineering units for point 1.
>
> **Cal2Engr** - Two point calibration engineering units for point 2.
>
> **DeadBand** - Dead band filtering value in ºC. (Default 0.01)
>
> **FilterElements** - running macro filter of the sampled data. Syntax: [*no. of elements*, *type*] where *no. of elements* is number of samples to filter; *type* – "AVERAGE", "MEDIAN" and "AVERAGEMEDIAN". Default is "=3,MEDIAN". Example: FilterElements1=6,AVERAGE
>
> **Description** - General purpose description.
>
> **CorrID** – What CorrectionID to use with this device. Must be found in the "Correction.INI" file that you write and keep in the same directory as the OMWT-SOFT-OPC.ini file. 0 is the default and means no correction; 127 is the maximum. If the device type is a thermistor, the default CorrID is found in the [Server] section "DefaultThermistorCorrID", which defaults to 128.

HUMIDITY

(OMWT-TEMPRH)

[Device*n*] - Section name.

Address - – serial number of the sensor.

DeviceType - Device type. (DeviceType=HUMIDITY)

PollTime - How often to poll the device. (ms)

Label - Name used for the device (and Topic).

Log - Enable/disable logging of device.

LogDecimalPlaces - Number of decimals to create when logging.

PollSamples - No. of samples per poll (must be odd no.) If >0 Median Filter if <0 then Average Filter is used.

Temperature - Indicates if sensor needs temperature compensation. (0 or 1).

DwellTime - Time allowed to charge the probe. (ms)

PostDwell - Time after reading to allow bus to recover. (ms)

OverSampleDwell - Time before each over sample to charge probe. (ms)

DeadBand - Dead band filtering value in % of full scale.

Description - General purpose description.

Cal1Raw - Two point calibration point 1.

Cal2Raw - Two point calibration point 2.

Cal1Engr - Two point calibration engineering units for point 1.

Cal2Engr - Two point calibration engineering units for point 2.

Units - % Relative Humidity

TempCoeff - PPM temperature coefficient of the sensor.

TempCalib - Temperature at which the calibration was done.

TempAddress – serial number of the temperature sensor in the probe.

MacroFilterElements - running macro filter of the sampled data. Syntax: [*no. of elements*, *type*] where *no. of elements* is number of samples to filter; *type* – "AVERAGE", "MEDIAN" and "AVERAGEMEDIAN". Default is "=3,MEDIAN". Example: FilterElements1=6,AVERAGE

CorrID – What CorrectionID to use with this device. Must be found in the "Correction.INI" file that you write and keep in the same directory as the OMWT-SOFT-OPC.ini file. 0 is the default and means no correction; 127 is the maximum.

DIGCOUNTER and Directional Counter

(OMWT-IRCNT, OMWT-DIRCNT)

[Device*n*] - Section name.

Address - – serial number of the sensor.

DeviceType - Device type. (DeviceType=DIGCOUNTER or DIRECTCNT)

PollTime - How often to poll the device. (ms)

Label - Name used for the device (and Topic) (typically DIGCOUNTER1 or DIRECTCNT1).

Log - Enable/disable logging of device.

LogDecimalPlaces - Number of decimals to create when logging.

ScaleA - Multiplier for Counter A

ScaleB - Multiplier for Counter B

UnitsA - Generic label for units - no function

UnitsB - Generic label for units - no function

DescriptionA - Generic description field

DescriptionB - Generic description field

EnableCounterB - Enable the gather of Counter B

Description – General purpose description field

CNTTEMP

(OMWT-DOORCNT)

[**Device***n*] - Section name.

Address – serial number of the sensor.

DeviceType - Device type. (DeviceType=CNTTEMP)

PollTime - How often to poll the device. (ms)

Label - Name used for the device (and Topic) (typically CNTTEMP1).

Log - Enable/disable logging of device.

LogDecimalPlaces - Number of decimals to create when logging.

Scale - Multiplier for Counter A

Units - Generic label for units - no function

Description - Generic description field

temp.Units - See OPC tag item temp.input.units for description.

temp.Cal1Raw - Two point calibration point 1.

temp.Cal2Raw - Two point calibration point 2.

temp.Cal1Engr - Two point calibration engineering units for point 1.

temp.Cal2Engr - Two point calibration engineering units for point 2.

temp.DeadBand - Dead band filtering value in ºC. (Default 0.00)

temp.FilterElements - running macro filter of the sampled data. Syntax: [*no. of elements*, *type*] where *no. of elements* is number of samples to filter; *type* – "AVERAGE", "MEDIAN" and "AVERAGEMEDIAN". Default is "=0". Example: FilterElements1=6,AVERAGE

temp.Description - General purpose description.

temp.CorrID – What CorrectionID to use with this device. Must be found in the "Correction.INI" file that you write and keep in the same directory as the OMWT-SOFT-OPC.ini file. 0 is the default and means no correction; 127 is the maximum

FASTCNTTEMP

(OMWT-PULSECNT)

[Device*n*] - Section name.

> **Address** – serial number of the sensor.
>
> **DeviceType** - Device type. (DeviceType=FASTCNTTEMP)
>
> **PollTime** - How often to poll the device. (ms)
>
> **Label** - Name used for the device (and Topic) (typically FASTCNTTEMP1).
>
> **Log** - Enable/disable logging of device.
>
> **LogDecimalPlaces** - Number of decimals to create when logging.
>
> **Scale** - Multiplier for Counter A
>
> **Units** - Generic label for units - no function
>
> **Description** - Generic description field
>
> **temp.Units** - See OPC item temp.input.units for description.
>
> **temp.Cal1Raw** - Two point calibration point 1.
>
> **temp.Cal2Raw** - Two point calibration point 2.
>
> **temp.Cal1Engr** - Two point calibration engineering units for point 1.
>
> **temp.Cal2Engr** - Two point calibration engineering units for point 2.
>
> **temp.DeadBand** - Dead band filtering value in ºC. (Default 0.00)
>
> **temp.FilterElements** - running macro filter of the sampled data. Syntax: [*no. of elements*, *type*] where *no. of elements* is number of samples to filter; *type* – "AVERAGE", "MEDIAN" and "AVERAGEMEDIAN". Default is "=0". Example: FilterElements1=6,AVERAGE
>
> **temp.Description** - General purpose description.
>
> **temp.CorrID** – What CorrectionID to use with this device. Must be found in the "Correction.INI" file that you write and keep in the same directory as the OMWT-SOFT-OPC.ini file. 0 is the default and means no correction; 127 is the maximum

ALARMTEMP

(OMWT-TEMPALRM)

[Device*n*] - Section name.

Address - serial number of the sensor.

DeviceType - Device type. (DeviceType=TEMP)

PollTime - How often to poll the device. (ms)

Label - Name used for the device (and Topic).

Log - Enable/disable logging of device.

LogDecimalPlaces - Number of decimals to create when logging.

Units - See OPC item temp.input.units for description.

Cal1Raw - Two point calibration point 1.

Cal2Raw - Two point calibration point 2.

Cal1Engr - Two point calibration engineering units for point 1.

Cal2Engr - Two point calibration engineering units for point 2.

DeadBand - Dead band filtering value in ºC. (Default 0.01)

FilterElements - running macro filter of the sampled data. Syntax: [*no. of elements*, *type*] where *no. of elements* is number of samples to filter; *type* – "AVERAGE", "MEDIAN" and "AVERAGEMEDIAN". Default is "=3,MEDIAN". Example: FilterElements1=6,AVERAGE

Description - General purpose description.

CorrID – What CorrectionID to use with this device. Must be found in the "Correction.INI" file that you write and keep in the same directory as the OMWT-SOFT-OPC.ini file. 0 is the default and means no correction; 127 is the maximum

Analog Input

(OMWT-AIN5V, OMWT-AIN10V, OMWT-AIN20MA, OMWT-VIB, OMWT-MOT)

[Devicen] - Section name

    Address – serial number of the sensor.

    DeviceType - DeviceType=AI

    PollTime - How often to poll the device (ms).

    Label - Name used for the device (and Topic) (typically AI1)

    Log - Enable/disable logging of device.

    LogDecimalPlaces - Number of decimals to create when logging.

    Cal1Raw - Two point calibration point 1 (default 0).

    Cal2Raw - Two point calibration point 2 (default 4095).

    Cal1Engr - Two point calibration engineering units for point 1 (default 0).

    Cal2Engr - Two point calibration engineering units for point 2 (default 10).

    DeadBand - Dead band filtering value in % of full scale (Cal2Engr).

    FilterElements – running macro filter of the sampled data. Syntax: [*no. of elements*, *type*] where *no. of elements* is number of samples to filter; *type* – "AVERAGE", "MEDIAN" and "AVERAGEMEDIAN". Default is "=1,MEDIAN" – no filtering. Example: FilterElements=3,AVERAGE

    Units - units label (default "%").

    Description - Generic description field

    CorrID – What CorrectionID to use with this device. Must be found in the "Correction.INI" file that you write and keep in the same directory as the OMWT-SOFT-OPC.ini file. 0 is the default and means no correction; 127 is the maximum.

Access/Control Reader

(OMWT-ACCESSREADER)

[Device*n*] - Section name.

> **Address** - serial number of the sensor.
>
> **DeviceType** - Device type. (DeviceType=IDR)
>
> **PollTime** - How often to poll the device or wait for packet. (ms)
>
> **Label** - Name used for the device (and Topic).
>
> **Log** - Enable/disable logging of device.
>
> **LogDecimalPlaces** - Number of decimals to create when logging.
>
> **Description** - General purpose description.

Repeater

(OMWT-REPEAT)

[Device*n*] - Section name.

> **Address** - serial number of the sensor.
>
> **DeviceType** - Device type. (DeviceType=REPEATER)
>
> **PollTime** - How often to poll the device or wait for packet. (ms)
>
> **Label** - Name used for the device (and Topic) (typically REPEATER1).
>
> **Log** - Enable/disable logging of device.
>
> **LogDecimalPlaces** - Number of decimals to create when logging.
>
> **Description** - General purpose description.

Dual Discrete Output

(OMWT-DUALOUT)

[Device*n*] - Section name.

> **Address** - serial number of the sensor.
>
> **NodalAddress** – node where the output will be sent. Defaults to 0 for the OMWT-REC323-600 Receiver. For OMWT-RECIP, the node number represents each receiver starting at 1. If set to 0, OMWT-SOFT-OPC will send the output packet to all nodes.
>
> **DeviceType** - Device type. (DeviceType= OUTPUTDISC)
>
> **PollTime** - How often to poll the device or wait for packet. (ms) Defaults to 0 for outputs.
>
> **Label** - Name used for the device (and Topic) (typically OUTPUTDISC1).
>
> **Log** - Enable/disable logging of device.
>
> **Description** - General purpose description.
>
> **TriggerOnB** – 0 – OMWT-SOFT-OPC will send the output packet when either items "outputa" or "outputb" is written to; 1 – OMWT-SOFT-OPC will send the outputpacket when only item "outputb" is written to. Set "outputa" before writing the value to "outputb". Both outputs A and B are set through one packet.
>
> **OutTries** – number of times to send the output packet. (default 10)
>
> **OutInterval** – (in milliseconds) – the interval of time between sending output packets. (default 1000).

Wireless Analog Output

(OMWT-ANGOUT)

[Device*n*] - Section name.

Address - serial number of the sensor.

NodalAddress – node where the output will be sent. Defaults to 0 for the OMWT-REC323-600 Receiver.   For OMWT-RECIP, the node number represents each receiver starting at 1. If set to 0, OMWT-SOFT-OPC will send the output packet to all nodes.

DeviceType - Device type. (DeviceType= OUTPUTANALOG)

PollTime - How often to poll the device or wait for packet. (ms) Defaults to 0 for outputs.

Label - Name used for the device (and Topic) (typically OUTPUTANALOG1).

Log - Enable/disable logging of device.

LogDecimalPlaces - Number of decimals to create when logging.

Description - General purpose description.

Cal1Raw - Two point calibration point 1 (default 0).

Cal2Raw - Two point calibration point 2 (default 4095).

Cal1Engr - Two point calibration engineering units for point 1 (default 0).

Cal2Engr - Two point calibration engineering units for point 2 (default 100).

Units - units label (default "%").

Description - Generic description field

OutTries – number of times to send the output packet. (default 10)

OutInterval – (in milliseconds) – the interval of time between sending output packets. (default 1000).

OMWT-SOFT-OPC INI Backup

Any time OMWT-SOFT-OPC Server modifies the INI file, OMWT-SOFT-OPC will create a copy of the previous INI file. OMWT-SOFT-OPC Server names this backup file the same file name as the INI file but names the file extension as ".Bnn" where n is from 01 to 20. OMWT-SOFT-OPC will create up to 20 backups. If OMWT-SOFT-OPC needs to create more, it will overwrite the oldest backup which will be typically starting at ".B01".

If you need to revert to a backup just rename the original OMWT-SOFT-OPC.INI file and then rename the backup to OMWT-SOFT-OPC.INI.

OMWT-SOFT-OPC Server Tags

Using OPC Tags

OMWT-SOFT-OPC uses the Device Name for the first part of the OPC tag name. The individual tag names from Device follow. OMWT-SOFT-OPC uses a hieratchial tree from name with various degrees of levels. The following is two examples of access two tags from a temperature device:

> TEMP1.input.value
>
> TEMP2.input.units

Device Names

OMWT-SOFT-OPC uses the device name as the first part of the OPC tag name. A device contains a nodal address. When "Search and Add Devices" is selected from setup, OMWT-SOFT-OPC Server will automatically assign label names that will be used as part of the OPC tag name for each new device found. The user can then change the name if they choose.

Default names are:

DIGITAL*n*	(Digital I/O)
TEMP*n*	(OMWT-TEMP15 and OMWT_THERM)
HUMIDITY*n*	(OMWT-TEMPRH)
THUMIDITY*n*	(Temperature device built-in to a OMWT-TEMPRH)
Ai*n*	(OMWT-AIN5V/AIN10V/AIN20MA/VIB/MOT)
DIGCOUNTER*n*	(OMWT-IRCNT)
CNTTEMP*n*	(OMWT-DOORCNT)
FASTCNTTEMP*n*	(OMWT-PULSECNT)
ALARMTEMP*n*	(OMWT-TEMPALRM)
IDR*n*	(OMWT-ACCESSREADER)
DIRECTCNT*n*	(OMWT-DIRCNT)
REPEATER*n*	(OMWT-REPEAT)
OUTPUTDISC*n*	(OMWT-DUALOUT)
OUTPUTANALOG*n*	(OMWT-ANGOUT)

Where n is a number starting at 1. OMWT-SOFT-OPC Server guarantees that each label assigned will be unique (No Duplicates Allowed).

Device Tag Names

The Server uses device tag names to allow access to the I/O data and specific operational information. Tag names are dependent on the type of devices used.

Temperature

(OMWT-TEMP15, OMWT-THERM): TEMP*n* or 'T' Prefixed Topic Names

online *(R)* - indicates if the device is communicating (0 or 1)

nodaladdress *(R)* – (TCP/IP Receiver). The node address starts at 1. For TCP/IP Receivers, the node address corresponds to the node number for the TCP/IP connection.

repeateraddress *(R)* – (Wireless mode) contains the location identifier of the OMWT-REPEAT repeater. If the sensor did not pass through a repeater then the value is 0.

polltime *(R/W)* - how often to poll the device (in milliseconds)

serialno *(R)* - registration number of the temperature sensor

input *(R)* - current temperature in engineering units

input.raw *(R)* - current temperature in ºC

input.units *(R/W)* -engineering units of the input tag (ºC, ºF, or ºK). When this tag is written, it changes the temperature scale in the **input.value** tag. **input.units** must be one of the following: "Celsius",”C”, "Fahrenheit",”F”, or "Kelvin",”K”.

input.deadband *(R/W)* - filtering deadband in ºC

input.conv *(R/W)* - calibration and unit conversion parameters string. (Cal1Raw,Cal1Engr,Cal2Raw,Cal2Engr)

input.desc *(R)* - General identification description field.

 (R) – indicates that the tag is readable.

(R/W) – indicates that the tag is both readable and writeable. OPC Write operations will either change an I/O point at the device, or an operation will be performed.

Digital I/O: DIGITAL*n*

online *(R)* - indicates if the device is communicating (0 or 1)

nodaladdress *(R)* – (TCP/IP Receiver). The node address starts at 1. For TCP/IP Receivers, the node address corresponds to the node number for the TCP/IP connection.

repeateraddress *(R)* – (Wireless mode) contains the location identifier of the OMWT-REPEAT repeater. If the sensor did not pass through a repeater then the value is 0.

polltime *(R/W)* - how often to poll the device (in milliseconds)

serialno *(R)* - registration number of the temperature sensor

inputA *(R)* - level of PIOA (0 or 1)

inputB *(R)* - level of PIOB (0 or 1)

counterA*(R)* - counter of the activity latch of PIOA

counterB *(R)* - counter of the activity latch of PIOB

counterAB *(R)* - counter A minus counter B

outputA.value *(R/W)* - current value of the output of PIOA (0 or 1)

outputB.value *(R/W)* - current value of the output of PIOB (0 or 1)

outputA.ondemand *(R/W)* - 1 - output on demand; 0 - on next poll (0 or 1)

outputB.ondemand *(R/W)* - 1 - output on demand; 0 - on next poll (0 or 1)

input.descA *(R)* - General identification description field for channel A.

input.descB *(R)* - General identification description field for channel B.

(R) – indicates that the tag is readable.

(R/W) – indicates that the tag is both readable and writeable. OPC Write operations will either change an I/O point at the device, or an operation will be performed.

Temperature/Humidity: HUMIDITY*n*

(OMWT-TEMPRH)

online *(R)* - indicates if the device is communicating (0 or 1)

polltime *(R/W)* - how often to poll the device (in milliseconds)

nodaladdress *(R)* – (TCP/IP Receiver). The node address starts at 1. For TCP/IP Receivers, the node address corresponds to the node number for the TCP/IP connection.

repeateraddress *(R)* – (Wireless mode) contains the location identifier of the OMWT-REPEAT repeater. If the sensor did not pass through a repeater then the value is 0.

serialno *(R)* - registration number of the temperature sensor

input.value *(R)* - current value of the pressure sensor in engineering units

input.raw *(R)* - current value in inches of Hg.

input.binary *(R)* - current value in binary units from the pressure probe.

input.conv *(R/W)* - calibration and unit conversion parameters string. (Cal1Raw,Cal1Engr,Cal2Raw,Cal2Engr,TempCoeff,TempCalib)

input.units *(R/W)* - units of the "%RH".

input.deadband *(R/W)* - deadband filtering in % of engineering Units

desc *(R)* - General identification description field.

tempflag *(R)* - Indicates if a temperature sensor is present in probe.

(R) – indicates that the tag is readable.

(R/W) – indicates that the tag is both readable and writeable. OPC Write operations will either change an I/O point at the device, or an operation will be performed.

DIGCOUNTER and Directional Counter: DIGCOUNTER*n* and DIRECTCNT*n*

(OMWT-IRCNT, OWT-DIRCNT)

online*(R)* - indicates if the device is communicating (0 or 1)

polltime*(R/W)* - how often to poll the device (in milliseconds)

nodaladdress *(R)* – (TCP/IP Receiver). The node address starts at 1. For TCP/IP Receivers, the node address corresponds to the node number for the TCP/IP connection.

repeateraddress *(R)* – (Wireless mode) contains the location identifier of the OMWT-REPEAT repeater. If the sensor did not pass through a repeater then the value is 0.

serialno*(R)* - registration number of the temperature sensor

dio.inputa*(R)* – state of the inputa, 1 or 0

dio.inputb*(R)* – state of the inputb, 1 or 0

cnt.inputa*(R)* - current count of dio.inputa

cnt.inputa.diff *(R/W)*- current count from the time the OMWT-SOFT-OPC was started or when the variable was reset to 0.

cnt.inputa.time*(R)* - time in milliseconds of last update (uses Win API GetTickCount())

cnt.inputa.scale*(R/W)* - value multiplied against the counts

cnt.inputa.units *(R/W)*- units description field

cnt.inputa.desc*(R/W)* - general purpose description field

cnt.inputb*(R)* - current count of dio.inputb

cnt.inputb.diff*(R)* - current count from the time the OMWT-SOFT-OPC was started or when the variable was reset to 0.

cnt.inputb.time*(R)* - time in milliseconds of last update (uses Win API GetTickCount())

cnt.inputb.scale*(R/W)* - value multiplied against the counts

cnt.inputb.units*(R/W)* - units description field

cnt.inputb.desc*(R/W)* - general purpose description field

(R) – indicates that the tag is readable.

(R/W) – indicates that the tag is both readable and writeable. OPC Write operations will either change an I/O point at the device, or an operation will be performed.

CNTTEMP: CNTTEMP*n*

(OMWT-DOORCNT)

online*(R)* - indicates if the device is communicating (0 or 1)

polltime*(R/W)* - how often to poll the device (in milliseconds)

nodaladdress *(R)* – (TCP/IP Receiver). The node address starts at 1. For TCP/IP Receivers, the node address corresponds to the node number for the TCP/IP connection.

repeateraddress *(R)* – (Wireless mode) contains the location identifier of the OMWT-REPEAT repeater. If the sensor did not pass through a repeater then the value is 0.

serialno*(R)* - registration number of the temperature sensor

dio.inputa*(R)* – state of the input, 1 or 0

dio.inputb*(R)* – state of the input, 1 or 0

cnt.input*(R)* - current count from the senosr

cnt.input.diff *(R/W)*- current count from the time the OMWT-SOFT-OPC was started or when the variable was reset to 0.

cnt.input.time*(R)* - time in milliseconds of last update (uses Win API GetTickCount())

cnt.input.scale*(R/W)* - value multiplied against the counts

cnt.input.units *(R/W)*- units description field

cnt.input.desc*(R/W)* - general purpose description field

temp.input *(R)* - current temperature in engineering units

temp.input.raw *(R)* - current temperature in ºC

temp.input.units *(R/W)* - engineering units of the input tag (ºC, ºF, or ºK). When this tag is written, it changes the temperature scale in the **input.value** tag. **input.units** must be one of the following: "Celsius",”C”, "Fahrenheit",”F”, or "Kelvin",”K”.

temp.input.deadband *(R/W)* - filtering deadband in ºC

temp.input.conv *(R/W)* - calibration and unit conversion parameters string. (Cal1Raw,Cal1Engr,Cal2Raw,Cal2Engr)

temp.input.desc*(R/W)* - General identification description field.

(R) – indicates that the tag is readable.

(R/W) – indicates that the tag is both readable and writeable. OPC Write operations will either change an I/O point at the device, or an operation will be performed.

FASTCNTTEMP: FASTCNTTEMP*n*

(OMWT-PULSECNT)

online*(R)* - indicates if the device is communicating (0 or 1)

polltime*(R/W)* - how often to poll the device (in milliseconds)

nodaladdress *(R)* – (TCP/IP Receiver). The node address starts at 1. For TCP/IP Receivers, the node address corresponds to the node number for the TCP/IP connection.

repeateraddress *(R)* – (Wireless mode) contains the location identifier of the OMWT-REPEAT repeater. If the sensor did not pass through a repeater then the value is 0.

serialno*(R)* - registration number of the temperature sensor

cnt.inputa*(R)* - current count from the sensor

cnt.inputa.diff *(R/W)*- current count from the time the OMWT-SOFT-OPC was started or when the variable was reset to 0.

cnt.inputa.time*(R)* - time in milliseconds of last update (uses Win API GetTickCount())

cnt.inputa.scale*(R/W)* - value multiplied against the counts

cnt.inputa.units *(R/W)*- units description field

cnt.inputa.desc*(R/W)* - general purpose description field

cnt.inputb*(R)* – 8 bit time in seconds of when the sensor captured the last count

cnt.inputb.units*(R/W)* - units description field

cnt.inputb.desc*(R/W)* - general purpose description field

temp.input *(R)* - current temperature in engineering units

temp.input.raw *(R)* - current temperature in ºC

temp.input.units *(R/W)* - engineering units of the input tag (ºC, ºF, or ºK). When this tag is written, it changes the temperature scale in the **input.value** tag. **input.units** must be one of the following: "Celsius",”C”, "Fahrenheit",”F”, or "Kelvin",”K”.

temp.input.deadband *(R/W)* - filtering deadband in ºC

temp.input.conv *(R/W)* - calibration and unit conversion parameters string. (Cal1Raw,Cal1Engr,Cal2Raw,Cal2Engr)

temp.input.desc*(R/W)* - General identification description field.

(R) – indicates that the tag is readable.

(R/W) – indicates that the tag is both readable and writeable. OPC Write operations will either change an I/O point at the device, or an operation will be performed.

ALARMTEMP: ALARMTEMP*n*

(OMWT-TEMPALRM)

online *(R)* - indicates if the device is communicating (0 or 1)

nodaladdress *(R)* – (TCP/IP Receiver). The node address starts at 1. For TCP/IP Receivers, the node address corresponds to the node number for the TCP/IP connection.

repeateraddress *(R)* – (Wireless mode) contains the location identifier of the OMWT-REPEAT repeater. If the sensor did not pass through a repeater then the value is 0.

polltime *(R/W)* - how often to poll the device (in milliseconds)

serialno *(R)* - registration number of the temperature sensor

input *(R)* - current temperature in engineering units

input.raw *(R)* - current temperature in ºC

input.units *(R/W)* -engineering units of the input tag (ºC, ºF, or ºK). When this tag is written, it changes the temperature scale in the **input.value** tag. **input.units** must be one of the following: "Celsius",”C”, "Fahrenheit",”F”, or "Kelvin",”K”.

input.deadband *(R/W)* - filtering deadband in ºC

input.conv *(R/W)* - calibration and unit conversion parameters string. (Cal1Raw,Cal1Engr,Cal2Raw,Cal2Engr)

input.desc *(R)* - General identification description field.

input.isalarm *(R)* – is the temperature above the set point (0 or 1).

input.isalarmtime *(R)* – has the temperature been above the set point for more than the alarm time (0 or 1).

input.marker *(R)* – general purpose 8 bit identifier from the sensor

input.alarmtemp *(R)* – set point for the high temperature alarm in engineering units

input.alarmtime *(R)* – set point for the amount of time that must pass for the input to be above the input.alarmtemp to trigger the input.alarmtime alarm

(R) – indicates that the tag is readable.

(R/W) – indicates that the tag is both readable and writeable. OPC Write operations will either change an I/O point at the device, or an operation will be performed.

AnalogInput: AI*n*

(OMWT-AIN5V, OMWT-AIN10V, OMWT-AIN20MA, OMWT-VIB, OMWT-MOT)

online*(R)* - indicates if the device is communicating (0 or 1)

polltime*(R/W)* - how often to poll the device (in milliseconds)

nodaladdress *(R)* – (TCP/IP Receiver). The node address starts at 1. For TCP/IP Receivers, the node address corresponds to the node number for the TCP/IP connection.

repeateraddress *(R)* – (Wireless mode) contains the location identifier of the OMWT-REPEAT repeater. If the sensor did not pass through a repeater then the value is 0.

serialno*(R)* - registration number of the sensor

input*(R)* - current value of the sensor as percent of full scale

input.conv*(R/W)* - unit conversion parameters string. (Cal1Raw,Cal1Engr,Cal2Raw,Cal2Engr)

input.units*(R/W)* – default is '%'.

input.desc*(R/W)* - General identification description field.

(R) – indicates that the tag is readable.

(R/W) – indicates that the tag is both readable and writeable. OPC Write operations will either change an I/O point at the device, or an operation will be performed.

Access/Control Reader: IDR*n*

(OMWT-ACCESSREADER)

online *(R)* - indicates if the device is communicating (0 or 1)

polltime *(R/W)* - how often to poll the device (in milliseconds)

nodaladdress *(R)* – (TCP/IP Receiver). The node address starts at 1. For TCP/IP Receivers, the node address corresponds to the node number for the TCP/IP connection.

repeateraddress *(R)* – (Wireless mode) contains the location identifier of the OMWT-REPEAT repeater. If the sensor did not pass through a repeater then the value is 0.

serialno *(R)* - registration number of the access/control reader

updatecount *(R/W)* – value is incremented whenever the sensor is updated.

access *(W)* – 1- access granted; 0 – access denied. When the client writes a value, the Server takes the contents of the id, serialno, unlocked, striketime and shunttime items and creates a packet that is sent to the reader.

batterylevel *(R)* – last voltage measurement of the battery.

batterylow *(R)* – indicates that the battery is low (0 – okay; 1 – low battery)

striketime *(W)* – the amount of time in seconds to engage the door lock. (0 to 255)

shunttime *(W)* – the amount of time in seconds that the door can remain open before a "Door Held Open" alarm event will be generated. (0 to 255)

doorheldopen *(R)* – indicates that the door is being held open (0 – normal door operation; 1 – held open)

dooropen *(R)* – indicates the current state of the door (0 – door is closed; 1 – door is open)

id *(R/W)* – 8 digit identifier of the last read card. A client can poke an id value to send with the output packet when the access item is poked. The id is represented as SSSIIIII where SSS is the 3 digit site code and IIIII is the 5 digit card number.

info *(R)* – an tag of information from the reader (0 to 255).

idtype – type of ID media. 0 – proximity card, 1 – IR fob.

unlocked *(W)* – – property to lock or unlock the door (0 – locked; 1 – unlocked).

tamper *(R)* – indicates if the read has detected tampering (0 – reader okay; 1 – tampered)

Note: the Server will send an output packet when the **access** item is written to respond to a reader request to open the door. The Server uses the contents of the **serialno**, **id**, **access**, **unlocked**, **striketime** and **shunttime** items to generate the output packet. These fields (mainly **id**, **unlocked**, **striketime** and **shunttime**) must be set before the **access** item is written by the client.

(R) – indicates that the tag is readable.

(W) – indicates that the tag is writeable only.

(R/W) – indicates that the tag is both readable and writeable. OPC Write operations will either change an I/O point at the device, or an operation will be performed.

Repeater: REPEATER*n*

(OMWT-REPEAT)

online *(R)* - indicates if the device is communicating (0 or 1)

polltime *(R/W)* - how often to poll the device (in milliseconds)

nodaladdress *(R)* – (TCP/IP Receiver). The node address starts at 1. For TCP/IP Receivers, the node address corresponds to the node number for the TCP/IP connection.

repeateraddress *(R)* – (Wireless mode) contains the location identifier of the OMWT-REPEAT Repeater. If the sensor did not pass through a repeater then the value is 0.

serialno *(R)* - registration number of the access/control reader

updatecount *(R/W)* – value is incremented whenever the sensor is updated.

battery *(R)* - current battery voltage (in volts)

locatorid *(R)* – location id of the repeater. This is the id that is attached to sensor packets and is represented by the OPC item "repeateraddress". (starts with "a")

cnt418 *(R)* – count of packets received through the 418 Mhz receiver.

cnt900 *(R)* – count of packets received through the 900 Mhz transiever.

radionetwork *(R)* – describes the set 900 Mhz network class of the repeater

(R) – indicates that the tag is readable.

(W) – indicates that the tag is writeable only.

(R/W) – indicates that the tag is both readable and writeable. OPC Write operations will either change an I/O point at the device, or an operation will be performed.

Dual Discrete Output: OUTPUTDISC*n*

(OMWT-DUALOUT)

online *(R)* - indicates if the device is communicating (0 or 1)

polltime *(R/W)* - how often to poll the device (in milliseconds)

nodaladdress *(R/W)* – (TCP/IP Receiver). The node address starts at 1. For TCP/IP Receivers, the node address corresponds to the node number for the TCP/IP connection. You can write to this parameter to determine which node to send the output packet. (write 0 and OMWT-SOFT-OPC will send to all nodes)

repeateraddress *(R)* – (Wireless mode) contains the location identifier of the OMWT-REPEAT Repeater. If the sensor did not pass through a repeater then the value is 0.

serialno *(R)* - registration number of the access/control reader

updatecount *(R/W)* – value is incremented whenever the sensor is updated.

outputa *(R/W)* – set to 0 or 1. Packet will be sent if TriggerOnB=0 option is set in the INI file.

outputb *(R/W)* – set to 0 or 1. Packet will be sent when written.

out.status *(R)* – 1 – OMWT-SOFT-OPC is in the process of sending output packets based on tries and interval; 0 – OMWT-SOFT-OPC has finished sending the output packet.

out.interval *(R/W)* – time in milliseconds between OMWT-SOFT-OPC sending the output packet.

out.tries *(R/W)* – the number of times OMWT-SOFT-OPC will send the output packet. "-1" – never stops sending packets.

out.service *(W)* – When written to, OMWT-SOFT-OPC sends output packets to the output device for training.

Note: when either **outputa** or **outputb** is written to, OMWT-SOFT-OPC will send out output packets based on **outtries** and **outinterval**. It first sets **outservice** to 1, and then to 0 when finished.

(R) – indicates that the tag is readable.

(W) – indicates that the tag is writeable only.

(R/W) – indicates that the tag is both readable and writeable. OPC Write operations will either change an I/O point at the device, or an operation will be performed.

Wireless Analog Output: OutputAnalog*n*

(OMWT-ANGOUT)

online *(R)* - indicates if the device is communicating (0 or 1)

polltime *(R/W)* - how often to poll the device (in milliseconds)

nodaladdress *(R/W)* – (TCP/IP Receiver). The node address starts at 1. For TCP/IP Receivers, the node address corresponds to the node number for the TCP/IP connection. You can write to this parameter to determine which node to send the output packet. (write 0 and OMWT-SOFT-OPC will send to all nodes)

repeateraddress *(R)* – (Wireless mode) contains the location identifier of the OMWT-REPEAT Repeater. If the sensor did not pass through a repeater then the value is 0.

serialno *(R)* - registration number of the access/control reader

updatecount *(R/W)* – value is incremented whenever the sensor is updated.

output.value *(R/W)* – set variable to the desired Engineering value

output.raw *(R/W)* – set the variable to the binary value (0 to 4095).

output.nits *(R/W)* - units label assigned to the analog output.

output.conv *(R/W)* - unit conversion parameters string. (Cal1Raw,Cal1Engr,Cal2Raw,Cal2Engr)

out.status *(R)* – 1 – OMWT-SOFT-OPC is in the process of sending output packets based on tries and interval; 0 – OMWT-SOFT-OPC has finished sending the output packet.

out.interval *(R/W)* – time in milliseconds between OMWT-SOFT-OPC sending the output packet.

out.tries *(R/W)* – the number of times OMWT-SOFT-OPC will send the output packet. "-1" – never stops sending packets.

out.service *(W)* – When written to, OMWT-SOFT-OPC sends output packets to the output device for training.

Note: when either **output.value** or **output.raw** is written to, OMWT-SOFT-OPC will send out output packets based on **outtries** and **outinterval**. It first sets **outservice** to 1, and then to 0 when finished.

(R) – indicates that the tag is readable.

(W) – indicates that the tag is writeable only.

(R/W) – indicates that the tag is both readable and writeable. OPC Write operations will either change an I/O point at the device, or an operation will be performed.

Server Item Names

Server Items are made available when the "Server" Topic is defined. These items give information about the Server and the current operational status of the Server.

PortType *(R)* - communication medium: "COM", "LPT", etc.

PortNo *(R)* - port number.

PollTries *(R)* - No. of missed polls before a device is considered offline.

NumberDevices *(R/W)* - number of devices in the system

OfflinePolltime *(R/W)* - how often to poll for off-line devices Milliseconds

NoDevicesOffline *(R)* - how many devices currently offline

Online *(R)* - indicates if communicating with any devices (0 or 1)

Ready *(R)* - 0 indicates that OMWT-SOFT-OPC is not receiving (initializing or do a restart function); 1-OMWT-SOFT-OPC is ready to receive.

CntNode – the number of nodes OMWT-SOFT-OPC is maintaining. For TCP/IP Receiver, this is the number of connections to receivers. Use this number to access the Node*n* items.

Node*n*.online *(R)* – indicates if the node is online or offline (0 – offline; 1 – online). *n* is the node number starting at 1.

Node*n*.address *(R)* – address of the nodeFor TCP/IP Receiver, the address is the IP address. *n* is the node number starting at 1.

Node*n*.description *(R)* – the description string for the node. *n* is the node number starting at 1.

NetErrors.Last - the last error that occurred (string); see below.

NetErrors.LastNo - the last error that occurred (enumerated); see below.

NetErrors.LastDevice - the topic name of the device that had the error

NetErrors.Count - number of errors that have occurred since Server started

NetErrors.Com - number of CRC errors

NetErrors.NoHostAdapter - number of errors from not detecting the Adapter

DeviceCount.DIGITAL *(R)* - number of digital devices

DeviceCount.TEMP *(R)* - number of temperature devices

DeviceCount.HUMIDITY *(R)* - number of humidity devices

DeviceCount.DIGCOUNTER *(R)* – number of digital counter devices

DeviceCount.**AI** *(R)*- number of Analog Input devices

DeviceCount.CNTTEMP – number of Counter Temperature

DeviceCount.FASTCNTTEMP – number of Fast Counter Temperature

DeviceCount.ALARMTEMP – number of Alarm Temperature

DeviceCount.IDR – number of Access/Control Readers

DeviceCount.DIRECTCNT – number of Directional Counters

DeviceCount.REPEATER – number of Repeaters

DeviceCount.OUTPUTDESC – number of Dual Discrete Outputs

DeviceCount.OUTPUTANALOG – number of Wireless Analog Outputs

Logging.Rate *(R/W)* **-** indicates the rate of logging in seconds. 0 - no logging. Less than 1 is accepted. Less than 5, OMWT-SOFT-OPC leaves the file open until terminated or set to 0. When 5 or greater, OMWT-SOFT-OPC opens, then logs, and then closes the file at every logging interval. When -1, OMWT-SOFT-OPC logs once immediately; OMWT-SOFT-OPC sets to 0 when completed.

Logging.LogFile *(R)* **-** full pathname of the log file.

Desc *(R)* - extra server tag clients can read/write for any purpose.

Ready *(R)* - 0 OMWT-SOFT-OPC is initializing; 1 OMWT-SOFT-OPC is ready and polling.

Version *(R)* - version number of OMWT-SOFT-OPC.

SearchAdd *(R/W)* - Client must write any number to start SearchAdd function. When completed OMWT-SOFT-OPC Server will set back to zero.

Quit *(R/W)* - When Client sets value greater than 0, OMWT-SOFT-OPC Server terminates and unloads.

Restart *(R/W)* - Client must set as follows.

1 to 999 OMWT-SOFT-OPC Server deletes the **OMWT-SOFT-OPC.ini** file device information, removes all device objects from memory, performs a SearchAdd function, creates a new device list, and rewrites this information to the **OMWT-SOFT-OPC.ini** file.

1000+ OMWT-SOFT-OPC Server deletes the **OMWT-SOFT-OPC.ini** file device information and then restarts the program.

<0 OMWT-SOFT-OPC Server restarts the program.

After successfully completing these operations, the **restart** tag is set back to zero.

* All error counters have the following format <u>0.000</u> where 0.001 is aOPCd to the count for any device error that causes a retry and a 1.000 is aOPCd to the counter if the error causes the device to go offline.

 (R) – indicates that the tag is readable.

(R/W) – indicates that the tag is both readable and writeable. OPC Write operations will either change an I/O point at the device, or an operation will be performed.

Description of NetErrors.LastNo and NetErrors.Last

-44	Communication Error: CRC16 or Time Out
-46	Cannot Find Receiver.
-39	OMWT-SOFT-OPC Server Internal Error.
-33	Invalid calibration entered.
-25	Communication error with Receiver.

OMWT-SOFT-OPC Server Data Logging

Features

OMWT-SOFT-OPC Server can log data to an ASCII file at a programmable interval. The default name of the file is OMWT-SOFT-OPC.LOG. The name can be changed by changing the INI file. The log rate (programmable interval) can be changed in the INI file (*[Logging] LogRate=*) or by a client application through OPC (SERVER!LogRate). By default, all devices listed in the INI file are logged. You can disable logging for each device (*[Devicen] Log=*). The number of decimal places logged for each device can be changed as well (*[Devicen] LogDecimalPlaces=*).   When a device is offline, OMWT-SOFT-OPC uses the last value gathered when online for logging. If the device has never been online, then -999999.0 is logged. You can have OMWT-SOFT-OPC place a character such as '*' or a string such as 'offline' in the logging string instead of the last value(*[Logging] OfflineIndicator=*). All these parameters can be changed in the INI file.

A client application can control logging through OPC variables. The client can start/stop logging or just log a single record. You can have your client application synchronize logging by keying off of one of the device's OPC variable changing values and then have OMWT-SOFT-OPC log one record. The client can also control the log rate. The full path name of the log file is also available to a client application through OPC. See the section "OPC Server Items" for more information.

For diagnostic purposes, OMWT-SOFT-OPC can log errors that have occurred while waiting and processing data from the sensors. This feature is enabled in the INI file (*[Logging] LogErrors=*). OMWT-SOFT-OPC places the errors in a file called OMWT-SOFT-OPC.ERR.   See the section "OMWT-SOFT-OPC.INI: Logging" for more information.

File Management

If the log rate is greater than or equal to 5 seconds, OMWT-SOFT-OPC opens and then closes the log file each time data is logged. If the log rate is less than 5 seconds, OMWT-SOFT-OPC leaves the file open and closes the file when OMWT-SOFT-OPC is terminated, the log rate is set to zero, or the log rate is set to 5 or greater.

The log file can be viewed while OMWT-SOFT-OPC is logging. The other program must not write to the file, lock the file, set the file as read only, or set the file's sharing parameters so that OMWT-SOFT-OPC cannot log.

When processing data with another program, the best approach is to either rename the file if the log rate is 5 or greater (OMWT-SOFT-OPC will automatically create a new file on the next log) or make a copy of the log file. When renaming, you may get an access error. Just try again.

The Error log file is opened and then closed whenever OMWT-SOFT-OPC writes an error record.

File Format - Data Log

The log record format is as follows:

*hh:mm:ss,mm/dd/yyyy,data1,data2,data3, ... , data**n***

Where:

hh:mm:ss is the time (ex. 23:40:50)

mm/dd/yyyy is the date (ex. 02/03/99)

*data**n*** - is the data for each device that has logging enabled.

Most devices have one value for the data. However 4 devices have different formats.

MANALOG: value1,value2,...,valuen, where n is the number of analog channels enabled.

DIGITAL: inputa, inputb, outputa, outpub, countera, counterb,

COUNTER: countera, counterb - if both channels are enabled

PROBE3: value, temp (only for Probe3 that has temperature compensation)

AIDO: ai1,ai2,ai3,ai4,do1,do2,do3,do4

DIGCOUNTER: dio.inputa, dio.inputb,cnt.inputa,cnt.inputb

CNTTEMP: dio.inputa, dio.inputb,cnt.input, temp.input

FASTCNTTEMP: cnt.inputa,cnt.inputb,temp.input

ALARMTEMP: value,isalarm,isalarmtime,alarmtemp,alarmtime

IDR: cardid,access,islocked,isbattery,doorheldopen,tamper, irtype, batterylevel,into,strike,shunt,dooropen

DIRECTCNT: dio.inputa, dio.inputb,cnt.inputa,cnt.inputb

REPEATER: batteryvoltage, locatorid, 418cnt, 900cnt, netclass

OUTPUTDISC: outputa,outputb

OUTPUTANALOG: outputvalue

Example log of a Pressure probe with temperature compensation and a Temperature probe.

10:33:45,03/30/1998,21.28,21.15,29.71

10:34:05,03/30/1998,21.29,21.15,29.71

10:34:25,03/30/1998,21.29,21.15,29.71

10:34:45,03/30/1998,21.33,21.15,29.71

File Format - Error Log

The log record format is as follows:

hh:mm:ss,mm/dd/yyyy,Error number,Error Msg,Online Flag,DeviceLabel

Where:

hh:mm:ss is the time (ex. 23:40:50)

mm/dd/yyyy is the date (ex. 02/03/99)

Error number -- Error number as docummented for OPC variable SERVER!ErrorNo

Error Msg - Error message string

Online Flag - Is device considered Online?

DeviceLabel - Name of the device

Example error log:

10:26:30,04/16/1998,-48,CRC16 or Time Out,1,TEMP8

10:26:30,04/16/1998,-48,CRC16 or Time Out1,TEMP9

OMWT-RECIP Receivers

Overview of OMWT-RECIP Receivers

OWMT-SOFT-OPC can connect to OMWT-RECIP receivers and receive and process sensors packets. As of this writing, the OWMT-RECIP (operating in "Pass-Thru" mode) is the only supported TCP/IP receiver. OWMT-SOFT-OPC makes a TCP/IP connection to a receive and then waits for sensor packets. OWMT-SOFT-OPC then processes the sensor packet as if it came from a receiver (like OWMT-REC232). OWMT-SOFT-OPC can initiate the connection or it can receive connections from a receiver. OWMT-SOFT-OPC can maintain multiple connections simultaneously (only limited by the resources of the host computer).

Setting up to use OMWT-RECIP Receivers

The OMWT-RECIP Receiver must be assigned a valid IP address and IP port number. If the OMWT-RECIP Receiver is going to initiate the connection to OWMT-SOFT-OPC then the OMWT-RECIP Receiver will need to be set up with the IP address and port number of the computer that is running OWMT-SOFT-OPC. See the manual for the specific receiver you want to use for information about setting the IP addresses.

To prepare OWMT-SOFT-OPC to use a OMWT-RECIP Receiver, make sure the Port type is "OMWT-RECIP Receiver". If you are using OWMT-SOFT-OPC for the first time, select "OMWT-RECIP Receiver" from the "Select Communication Port" option when it shows a window displaying "Cannot find Receiver". OWMT-SOFT-OPC will then display the "All Current IP Addresses" window. You can select this window from the Setup menu. Click the "Add New" button to add a new IP connection. Click the "Test" button to test the selected IP Address. Click the "Edit" button to change the properties of a connection. When you click the OK button, OWMT-SOFT-OPC will attempt to the list of specified IP addresses. You can view the status of the connections by selection the Setup Menu and then select "View IP Connection Status" to display the "IP Status" window.

OMWT-RECIP Receiver Notes

If OWMT-SOFT-OPC initiates the connection then OWMT-SOFT-OPC will continually try to establish the connection when the connection is disrupted or never accepted. If the receiver initiates the connection, OWMT-SOFT-OPC relies on the receiver to re-establish the connection if the connection is disrupted.

Each TCP/IP connection is assigned a "node number". This "node number" gets associated with a sensor's data. The "nodaladdress" OPC item is updated with the "node number".

Information about the individual TCP/IP connection is made available through the Server OPC parameters (the "Node*x*" OPC item). See the Server OPC topic for more information.

OWMT-SOFT-OPC can wait for connections initiated by the receiver. OWMT-SOFT-OPC will automatically add the TCP/IP connection to its list. OWMT-SOFT-OPC waits on port 1060 by default. You can change this port number through the OWMT-SOFT-OPC INI file. In the "IP Addresses" section change the "IPWaitPort=" line.

Error Messages

Communication Errors

OMWT-SOFT-OPC Server communication initialization errors will cause the 'OMWT-SOFT-OPC Initializing Error!" window to be displayed to help you determine what the cause of failure may be and to allow you to retry or abort the program.

Error In Finding/Initializing Port.

You have specified a port that does not exist on your computer. Use **Setup** to change to a port you have, or select **Auto** and let OMWT-SOFT-OPC Server search for you.

Cannot Open Communication Port; Already In Use.

Close application that is using the communication port or provide OMWT-SOFT-OPC Server with another RS-232 communications port.

Cannot Find Receiver!

This indicates OMWT-SOFT-OPC Server cannot find the receiver. If the receiver is connected with a serial cable, try another serial cable. The receiver uses the REC232 handshake lines for power. Is the power LED on?

Try another port.

Cannot find and load lowlevel driver!

OMWT-SOFT-OPC Server cannot find one of its DLL drivers. Either the driver is missing from the Windows\System folder or the registry entry got erased. Try reinstalling the program. If problem continues, please consult your dealer.

Runtime Errors

Communication Error: CRC16 or Time Out

Sensor transmission was not received within the polltime or the transmission got corrupted.

Cannot Find Receiver.

Receiver has become disconnected from the computer.

OMWT-SOFT-OPC Server Internal Error.

Contact your dealer.

Corrections

Why Use Corrections

Some devices, like a thermistor, need their data corrected in order to be meaningful. A thermistor uses resistance to measure temperature. The relation between the resistance and the temperature is non linear. A thermistor uses a set of correction parameters to convert resistance to temperature. Other devices may need to correct data values in a non linear way. OMWT-SOFT-OPC provides a way for the user to correct the data from a sensor using a set of correction parameters or a lookup table.

Correction IDs

Tell OMWT-SOFT-OPC that you want a device to use a correction by specifying the Correction ID in the OMWT-SOFT-OPC.INI file. Within the device section add a line saying "CorrID=" and the number of the correction that you want OMWT-SOFT-OPC to use. The Corrections are found in the Correction.INI file that you write.

Pre-defined Thermistor

The thermistor made by the vendor Thermometrics has the part number RL0503-5820-97-MS . This thermistor's parameters are predefined in OMWT-SOFT-OPC and given the Correction ID 128. (In the device section of the OMWT-SOFT-OPC.INI file, write "CorrID=128" to use this type.)

The parameters for the predefined thermistor are as follows:

NTC_DIVIDERES = 10000

NTC_NORMALRES = 10000

Fitting constants:

A = 3.3539264E-3

B = 2.5609446E-4

C = 1.9621987E-6

D = 4.6045930E-8

How to use Corrections

Correction.INI file

After adding the Correction ID to the device section of the OMWT-SOFT-OPC.INI file, you must set up the Correction used by that Correction ID. If you're using a correction that is not predefined, you must write a Correction.INI file. This file should be stored in the same directory as the OMWT-SOFT-OPC.INI file. Name your file "Correction.INI" and include all the correction sections you like, from 1 to 127. Include a CorrType, either 1 (for correction parameters) or 2 (for a lookup table).

Lookup Table

A lookup table is a CorrType of 2. If you do a lookup table, you must observe a couple of rules. You must specify Elements, which is the number of values in the list. The RawValues list and the CorrectedValues list must be of the same length. The RawValues list must go from smallest to largest (numbers must be in increasing order). When writing your lookup values, do not put a space after the comma. There is a functional limit to the number of elements in the lists; roughly, 500 integer values or 300 real values, for the RawValues and CorrectedValues taken individually.

Here is an example of a lookup correction section:

> [Correction5]
>
> CorrType=2
>
> Elements=5
>
> RawValues=400,450,500,550,600
>
> CorrectedValues=1400,1450,1500,1550,1600

Thermistor NTC Parameters

You may choose to correct your sensor's data with a group of parameters. Use CorrType 1. If you decide to use parameters, specify the parameters as in the following example:

> [Correction2]
>
> CorrType=1
>
> DivideRes=10000
>
> NormalRes=10000
>
> A=3.3539264E-3
>
> B=2.5609446E-4
>
> C=1.9621987E-6
>
> D=4.6045930E-8

Predefined Corrections

The device type "Thermistor" will default to a correction ID of 128. To override this default you can add a CorrID parameter to the device section of the OMWT-SOFT-OPC.INI file, or you can set a new default for all thermistors in the Server section of the OMWT-SOFT-OPC.INI file.   To change the default CorrectionID for all thermistors add the parameter "DefaultThermistorCorrID=" to the Server section, and set it to whatever CorrectionID you wish.

Predefined Corrections use the following equations. The first equation normalizes the passed value, and the second produces the conversion to degrees Celsius.

Parameters:

> NTC_DIVIDERES = 10000
>
> NTC_NORMALRES = 10000
>
> Fitting constants:
>
> A = 3.3539264E-3
>
> B = 2.5609446E-4
>
> C = 1.9621987E-6
>
> D = 4.6045930E-8

R_{normal} = NTC_DIVIDERES / (32767 / value) - 1   /   NTC_NORMALRES

Value in degrees Celsuis = 1 /   (A + $B*R_{normal}$ + $C*R_{normal}^{2}$ + $D*R_{normal}^{3}$)

WARRANTY/DISCLAIMER

OMEGA ENGINEERING, INC. warrants this unit to be free of defects in materials and workmanship for a period of **13 months** from date of purchase. OMEGA's Warranty adds an additional one (1) month grace period to the normal **one (1) year product warranty** to cover handling and shipping time. This ensures that OMEGA's customers receive maximum coverage on each product.

If the unit malfunctions, it must be returned to the factory for evaluation. OMEGA's Customer Service Department will issue an Authorized Return (AR) number immediately upon phone or written request. Upon examination by OMEGA, if the unit is found to be defective, it will be repaired or replaced at no charge. OMEGA's WARRANTY does not apply to defects resulting from any action of the purchaser, including but not limited to mishandling, improper interfacing, operation outside of design limits, improper repair, or unauthorized modification. This WARRANTY is VOID if the unit shows evidence of having been tampered with or shows evidence of having been damaged as a result of excessive corrosion; or current, heat, moisture or vibration; improper specification; misapplication; misuse or other operating conditions outside of OMEGA's control. Components which wear are not warranted, including but not limited to contact points, fuses, and triacs.

OMEGA is pleased to offer suggestions on the use of its various products. However, OMEGA neither assumes responsibility for any omissions or errors nor assumes liability for any damages that result from the use of its products in accordance with information provided by OMEGA, either verbal or written. OMEGA warrants only that the parts manufactured by it will be as specified and free of defects. OMEGA MAKES NO OTHER WARRANTIES OR REPRESENTATIONS OF ANY KIND WHATSOEVER, EXPRESS OR IMPLIED, EXCEPT THAT OF TITLE, AND ALL IMPLIED WARRANTIES INCLUDING ANY WARRANTY OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE HEREBY DISCLAIMED. LIMITATION OF LIABILITY: The remedies of purchaser set forth herein are exclusive, and the total liability of OMEGA with respect to this order, whether based on contract, warranty, negligence, indemnification, strict liability or otherwise, shall not exceed the purchase price of the component upon which liability is based. In no event shall OMEGA be liable for consequential, incidental or special damages.

CONDITIONS: Equipment sold by OMEGA is not intended to be used, nor shall it be used: (1) as a "Basic Component" under 10 CFR 21 (NRC), used in or with any nuclear installation or activity; or (2) in medical applications or used on humans. Should any Product(s) be used in or with any nuclear installation or activity, medical application, used on humans, or misused in any way, OMEGA assumes no responsibility as set forth in our basic WARRANTY / DISCLAIMER language, and, additionally, purchaser will indemnify OMEGA and hold OMEGA harmless from any liability or damage whatsoever arising out of the use of the Product(s) in such a manner.

RETURN REQUESTS/INQUIRIES

Direct all warranty and repair requests/inquiries to the OMEGA Customer Service Department. BEFORE RETURNING ANY PRODUCT(S) TO OMEGA, PURCHASER MUST OBTAIN AN AUTHORIZED RETURN (AR) NUMBER FROM OMEGA'S CUSTOMER SERVICE DEPARTMENT (IN ORDER TO AVOID PROCESSING DELAYS). The assigned AR number should then be marked on the outside of the return package and on any correspondence.

The purchaser is responsible for shipping charges, freight, insurance and proper packaging to prevent breakage in transit.

FOR **WARRANTY** RETURNS, please have the following information available BEFORE contacting OMEGA: 1. Purchase Order number under which the product was PURCHASED, 2. Model and serial number of the product under warranty, and 3. Repair instructions and/or specific problems relative to the product.	FOR **NON-WARRANTY** REPAIRS, consult OMEGA for current repair charges. Have the following information available BEFORE contacting OMEGA: 1. Purchase Order number to cover the COST of the repair, 2. Model and serial number of the product, and 3. Repair instructions and/or specific problems relative to the product.

Where Do I Find Everything I Need for Process Measurement and Control?
OMEGA…Of Course!
Shop online at www.omega.com

TEMPERATURE
Thermocouple, RTD & Thermistor Probes, Connectors, Panels & Assemblies
Wire: Thermocouple, RTD & Thermistor
Calibrators & Ice Point References
Recorders, Controllers & Process Monitors
Infrared Pyrometers

PRESSURE, STRAIN AND FORCE
Transducers & Strain Gages
Load Cells & Pressure Gages
Displacement Transducers
Instrumentation & Accessories

FLOW/LEVEL
Rotameters, Gas Mass Flowmeters & Flow Computers
Air Velocity Indicators
Turbine/Paddlewheel Systems
Totalizers & Batch Controllers

pH/CONDUCTIVITY
pH Electrodes, Testers & Accessories
Benchtop/Laboratory Meters
Controllers, Calibrators, Simulators & Pumps
Industrial pH & Conductivity Equipment

DATA ACQUISITION
Data Acquisition & Engineering Software
Communications-Based Acquisition Systems
Plug-in Cards for Apple, IBM & Compatibles
Datalogging Systems
Recorders, Printers & Plotters

HEATERS
Heating Cable
Cartridge & Strip Heaters
Immersion & Band Heaters
Flexible Heaters
Laboratory Heaters

ENVIRONMENTAL
MONITORING AND CONTROL
Metering & Control Instrumentation
Refractometers
Pumps & Tubing
Air, Soil & Water Monitors
Industrial Water & Wastewater Treatment
pH, Conductivity & Dissolved Oxygen Instruments

M-4066/0604

Free Manuals Download Website

http://myh66.com

http://usermanuals.us

http://www.somanuals.com

http://www.4manuals.cc

http://www.manual-lib.com

http://www.404manual.com

http://www.luxmanual.com

http://aubethermostatmanual.com

Golf course search by state

http://golfingnear.com

Email search by domain

http://emailbydomain.com

Auto manuals search

http://auto.somanuals.com

TV manuals search

http://tv.somanuals.com