Introduction to OmniServer

Overview
Table of Contents

INTRODUCTION 3

OMNISERVER FACTS 4

OMNISERVER CONCEPTS 6

Configuration
  Devices 6
  Topics 7
  Protocols 8
  Clients 8

Diagnostics 9
Introduction

This document is meant to provide general OmniServer information, covering both factual details and the concepts behind how OmniServer works and some general pointers for using OmniServer with a device protocol. This document will briefly explain the purpose of the different configuration elements present in OmniServer.
OmniServer Facts

- Began development in 1994 on the Windows 3.1 platform

- The name “OmniServer” means “All Server” or “Everywhere Server” (Omni = All, Everywhere, as in “Omnipresent”).

- Was developed to address the fact that many PLCs did not have device drivers, and companies were spending countless hours and thousands of dollars to develop custom drivers.

- OmniServer was unique in that a customer could configure OmniServer to talk to a variety of devices without having to write a single line of code. All configuration could be done through dialog boxes.

- Originally targeted towards waste treatment facilities using Wonderware’s InTouch communications. It was expanded later to add SuiteLink, PackedDDE, and AdvancedDDE communications used by HMI Vendors. (This was all prior to the introduction of “OPC”.

- First OmniServer was sold in October 1995.

- Added FIX DMACS support in 1996 (i-FIX PDB Client Interface).

- Added Ethernet (Winsock) support in 1997.

- Was one of the first servers to offer OPC support in 1998.

- Released Version 2.0 in January 2002, which completely re-designed the user interface and added configurable Error Detection Codes. Also, OmniServer Client Wedges were added.

- Was purchased by Software Toolbox in March 2005 from Descartes System Sciences, Inc.

- Released Version 2.5 in July 2005.


- OmniServer is a configurable OPC server.

- The term configurable is used here to describe the main function of OmniServer – that a user can alter how OmniServer talks to a device through the GUI (via a protocol).
• This differs from servers like TOP Server, in which the basic protocol cannot be changed by the user. With OmniServer, all protocols can be changed as desired or needed.

• This is similar to the UCON driver in TOP Server, but the way UCON defines protocols is radically different from OmniServer.
OmniServer Concepts

There are two parts to the OmniServer program: Configuration and Diagnostics. Each of these parts can be accessed either through the menu or through the left-hand sidebar.

Configuration

There are four configuration tools available to the user.

1. Devices
2. Topics
3. Protocols
4. Clients

Devices

An OmniServer “Device” tells OmniServer what physical port is available and how it is to be configured. There is no checking to see if these ports actually exist on the PC (Serial or Parallel), or if the port is reachable (Winsock), so be careful when adding OmniServer devices to only add devices that actually exist.

Troubleshooting tip: OmniServer relies upon Windows for all device communications – there are no “device drivers” built-in to OmniServer for the actual physical connection. If someone is having problems communicating with a device, Hyperterminal can be used to test a connection to the device as a cross-check. If Hyperterminal works, OmniServer will work. If HyperTerminal does not work, OmniServer will not either. The only exception to this rule is when a device needs to have a delay between transmitted characters. HyperTerminal will work because, when you type in messages, you naturally will have a delay between characters. OmniServer will send out characters as fast as possible – without delay – and, therefore, will appear not to work when HyperTerminal will. This is very rare, but does occur with some protocols from time to time.
The three device types are:

1. **Serial** – This is your basic Serial port, with configuration for Baud Rates, Data Bits, Parity, etc.

2. **Parallel** – This is a write-only device driver that sends data out the LPT ports. **Please Note:** There was a bug in V2.5 and prior which does not output data to the LPT port. Anyone experiencing this issue will need to upgrade to V2.6

3. **Winsock** – This is your basic Ethernet port. OmniServer can communicate via Ethernet using TCP, UDP, or TELNET, for both solicited and unsolicited messages. There is also a PING button available to see if the device associated with the entered IP Address is available. This is useful only for determining the device is there, not for determining it’s readiness for communications.

**Troubleshooting tip:** Look at the OmniServer Logger (see Diagnostic section) for errors pertaining to devices in use. Most serial errors can indicated an OmniServer device not configured correctly (framing errors, parity errors, etc.), a missing physical COM port (COM port cannot be found), or the port is in use by another program (Access Denied). For Winsock ports, you will most likely get “WINSOCK 100xx” errors. These are errors generated by Windows and passed to OmniServer. For a complete list of Winsock errors, see: [http://msdn.microsoft.com/en-us/library/ms740668.aspx](http://msdn.microsoft.com/en-us/library/ms740668.aspx).

**Topics**

An OmniServer “Topic” associates a name with a Protocol / Device combination. For example, if you wanted to use the “MODBUSR” sample protocol with Device “COM1”, then you would create a topic that associates “COM1” with the protocol “MODBUSR”. From that point on, any client program who wants to access the Modbus RTU device connected to the PC’s COM1 serial port will use this defined topic name.

Here are some important points about Topics:

- You can only assign one device per topic. **For Your Information**, OmniServer V1.x did allow for multiple devices per topic. This functionality was dropped because it was complex internally and increased the potential for data from different devices to be misrepresented to the client application.

- Multiple Topics can use the same Protocol. This means if you have 100 physical devices out there using the same protocol format, in OmniServer, you would create 100 Devices and 100 Topics, but each Topic would use the same Protocol that had been created. Internally, OmniServer will “copy” the protocol in order to keep the data for each device separate for the data for the other devices.
• Multiple Topics can point to the same Device. This is useful in “multi-drop” situations. For example, you have 10 Modbus RTU devices multi-dropped on the same RS-485 line connected to a single PC serial port. The Device in OmniServer will be the same serial COM device for each of these 10 RTUs, as well as the Protocol being the same. It would be necessary here to create 10 Topics, each configured to use the same Device and Topic. In cases like this, any data received from a device will contain some sort of “DeviceID” in the data stream. This “DeviceID” would be accounted for in the Protocol for these devices using a Topic Variable. When a Protocol uses a Topic Variable, the value of that Topic Variable is configured in the Topic. This variable will match the “DeviceID” in the data stream received from the device. When OmniServer receives a data stream from a device, it will look at that “DeviceID” and use it to determine to which Topic the data belongs.

Protocols

The heart of OmniServer is the Protocol. A Protocol is defined as a description of how a data stream is to be constructed so that the receiving device can understand it, as well as how a data stream coming from a device is to be interpreted by OmniServer. For a comparison, each TOP Server device driver contains within it a protocol definition that defines how TOP Server is to communicate with that device and interpret any actual data bytes received from that device.

OmniServer does the same thing, essentially, in the Protocol, which is a group of patterns that OmniServer either uses to build a sequence of bytes that the device will understand or that OmniServer uses to compare the sequence of bytes in a device response data string to determine how to process the data (which bytes should be placed in items, which bytes are a checksum, etc.).

The difference is that you cannot change a protocol for an off-the-shelf solution like TOP Server without rewriting the driver at the code level. With OmniServer, any user can change the protocol on the fly through the dialog boxes available for protocol configuration in the OmniServer Configuration interface. This is what makes OmniServer (and similarly, TOP Server UCON) configurable.

Clients

With the release of OmniServer V2.0, client programs were added to OmniServer. There are basically two types of clients:

1. **Informational Clients** - These client icons, such as DDE, Suitelink and OPC, are information only (no configuration). If you hover the mouse pointer over these icons, they will show you the information a client program would need to make that particular type of connection to OmniServer.
2. **OmniServer Wedges** - Available in OmniServer Professional only (with the exception of the Keyboard Wedge, which is the only client available in the OmniWedge version). These internal clients make connections to the OmniServer polling engine and perform client-type operations. Currently, there are four client wedges available:

- **Keyboard Wedge** – Takes OmniServer item data and inserts that data into Windows applications as keystrokes (picture your item values being “typed” into MS Word by OmniServer).

- **File Wedge** – Takes OmniServer item data and creates or appends formatted data to a file. This can be in either Binary or ASCII format – this is completely configurable in the File Wedge configuration.

- **Database Wedge** – Takes OmniServer item data and inserts or replaces data into a pre-defined ODBC database. Items are mapped to columns in an ODBC database, and the Wedge defines the database (Must be configured in Windows Database Source Administrator) and table to use.

- **E-Mail Wedge** – New in OmniServer Professional V2.6. Monitors OmniServer item data for a configured trigger condition to occur. It then sends an e-mail notification to any recipients specified in the configuration, with the e-mail text also specified in the wedge configuration. The e-mail text is fully configurable to include any text or item values desired.

**Diagnostics**

OmniServer comes with a wide range of diagnostic tools to help you troubleshoot the protocols. Its diagnostic screens are very detailed and informative (though OPC Diagnostics are not available internal to OmniServer for troubleshooting OPC connectivity issues). You can determine the cause of just about any problem with these screens – if you know how to use them.

**Troubleshooting tip:** OmniServer is a “pure” server, in the true sense of a client/server architecture. That being said, nothing will happen (no ports opened) unless a client program is asking for updates for at least one item in the protocol associated with the Topic. Why is this? It pertains to resource conservation. The beauty of a client/server architecture is that a server is only asking the device for the data the ultimate client application is requesting. There is no need to request any other data, if it is not needed, because this would be inefficient. If a client is not connected, there will be no activity in any of the diagnostic screens in OmniServer. The same is true of the majority of OPC Servers.
There are four diagnostic screens in OmniServer. They are listed below in order of importance/usefulness:

1. **Item Values** – This screen shows all the data being requested by clients. By default, you will see each configured Topic defined here as a folder. Under this folder, all items from this topic that are being requested from any client application will be listed. You will always see a reserved item called “status”. Additionally, if the protocol associated with a Topic is using a Topic Variable, there will also be a reserved item for that Topic Variable listed here. These items are always available to any client for a particular topic and their presence does not indicate that a client is connected.

   **Troubleshooting tip:** If “Status” has a value of True (1), then the last operation by OmniServer was a success. If it is False, then the last operation was not a success. At that point, look in the OmniServer Logger diagnostic window for more information.

   **How do you use the Item Values Window?** Since this window shows any items requested by a client, you use this window to tell if a client is connected. If the Topics listed in this window have no items other than “status” or Topic Variables, then no client application is connected and OmniServer is doing nothing, as should be expected. **No client connection is one of the most common problems with OmniServer.** If there are other items listed here, then that confirms there is a client program somewhere requesting item updates from OmniServer, eliminating client connectivity as a potential problem.

2. **Logger** – This diagnostic window, serves as the “event log” for OmniServer, displaying a log of any errors, events, sends or receives of data that occur in OmniServer. By default, OmniServer is configured to display Errors and Events that may occur, but will not show data sends or receives between OmniServer and devices. These logging options can be enabled/disabled via the View -> Logging menu or by clicking their icons in the OmniServer toolbar for the Logger window.

   **How do you use the Logger Window?** Look here for any errors that occur with device communications. If the Item Value window shows items being requested and the client program is not receiving updates, look at the Logger to determine what error might have occurred.

   **Troubleshooting tip:** Most errors are hardware related – the only OmniServer specific error that could be reported is “Timeout waiting for response” – which might be a protocol problem, but could also be a problem with the device. All other problems are hardware or device configuration related.
**Troubleshooting tip:** The Logger has a finite number of lines it will hold before losing entries. Starting with V2.6, this size is configurable (View Menu -> General Options -> Logger) from 64 KB up to 999 KB. It is still best to use the I/O Monitor for viewing Sends and Receives rather than filling the Logger with them.

3. **I/O Monitor** – This diagnostic window is essential for troubleshooting any OmniServer protocol. It allows you to see data bytes that are sent to a device by OmniServer and data bytes that are received from a device by OmniServer. The data is color coded to make it easy to determine which data is sent data and which data is received data and any problems there might be with received data.

   - **Black Data** – This is always going to be data that OmniServer has sent to the device. This would be the Request portion of the Host Messages or the Response portion of an Unsolicited Message in the Protocol.

   - **Blue Data** – This is received data that has not been processed. This normally means that the Response portion of a Host Message or Received portion of an Unsolicited Message in the Protocol does not match the pattern of data received by OmniServer. It will always turn into one of the colors below (unless no more data is received from the device).

   - **Green Data** – This is received data that has been successfully processed and matched to a message in the Protocol, allowing the data to be parsed into any configured Items.

   - **Magenta Data** – This is received data that has been processed but did not match any message in the Protocol. This means that there is at least one byte in the Response or Received sections of the messages in the Protocol that does not exactly match the pattern of the data being received.

   - **Red Data** – This is received data that was blue or magenta but also failed an Error Detection Code (EDC) check. This would only ever be possible if the device uses Error Detection Codes such as a checksum.

In addition to the color coding, the I/O Monitor also displays both Hexadecimal and ASCII representations of the same data bytes, making it easy to interpret what has been sent and received. It is perfect for comparing the data received from the device to the messages configured in the Protocol, if the data is not being processed as it should be.
Troubleshooting tip: The I/O Monitor gives you the option of selecting the Topic or the Device for viewing sent and received data. For the most part, you will look at Devices (not Topics). The reason for this is that the data the I/O Monitor displays for the Topic is only good data for that Topic, which is not particularly useful if you are having problems with your protocol (there will be nothing displayed, even if you are getting blue, magenta or red data responses from the device). Therefore, it is always best to select the Device of interest in the I/O Monitor since it will display all data, good or bad.

How do you use the I/O Monitor Window? This window is used to show the raw data being processed by OmniServer. It is especially good for determining if a protocol has been configured correctly. Receiving no Magenta, Red, Blue or Green Data at all? Then the Black Data being sent to the device (the commands) are not being interpreted correctly by the device. Receiving only Magenta, Blue or Red? That means the device is responding, but the message pattern for that response in the Protocol is not correct. Receiving all Green data? Then the Protocol is correct!