Integrate IEDs With OPC Technology

Pankaj Bhatt and Roger P. Baldevia, Jr.
Schweitzer Engineering Laboratories, Inc.

Presented at the
5th Annual Clemson University Power Systems Conference
Clemson, South Carolina
March 14–17, 2006

Originally presented at the
7th Annual Western Power Delivery Automation Conference, May 2005

INTEGRATE IEDS WITH OPC TECHNOLOGY

Pankaj Bhatt and Roger P. Baldevia, Jr.
Schweitzer Engineering Laboratories, Inc.
Pullman, WA USA

INTRODUCTION

As technology spawns more intelligent electronic devices (IEDs), the communication protocols that support them also increase. Today, engineers and technicians are looking for ways to integrate hardware and protocols on a common platform that will meet the needs of data consumers in the utility or plant.

Most IEDs specialize in retrieving pertinent data and making it available to various data consumers. Operators focusing on traditional Supervisory Control and Data Acquisition (SCADA) data, system planning engineers requiring maintenance information, and managers gathering economic statistics are some of the data consumers who benefit from tight integration of IEDs and protocols.

DeviceNet, MODBUS®, DNP3, Profibus, Foundation Fieldbus, IEC 60870, and IEC 61850 are a few of the protocols available in the utility or industrial field. When selecting the protocol to best meet integration needs, consider these factors:

- Corporate or personal preference
- Compatibility with hardware and protocols already in place
- Economic feasibility
- Efficiency
- Ease of use
- Functionality of protocol
- Ability of protocol to retrieve data of interest
- Maturity, standardization, and documentation of protocol

OPC, Object Linking and Embedding (OLE) for Process Control, defines a rule set for data exchange between software applications, much the same way that protocols provide the rules for data exchange between devices. These software applications include protocol drivers, databases, and programmable logic, as well as distributed and centralized monitoring, control, and SCADA systems. The data sources to support these applications are often IEDs, such as meters, relays, programmable logic controllers (PLCs), Remote Terminal Units (RTUs), or in-service databases.

The goal of the project documented in this paper was to design and test a system using several protocol drivers that also act as OPC servers for several different protocols within the same system. This demonstrates that different protocols can coexist and their common OPC interface will support multiple applications on various hardware platforms representing real-world applications. We used readily available OPC servers that could easily be used off the shelf, without software customization.

Connecting devices using various protocols that support OPC demonstrates a method that can be used for making information from disparate protocols available on one common Human Machine Interface (HMI). We also show how OPC provides a common platform to meet protocol-handling and hardware-integration situations for other installations.

PROTOCOLS

Protocols are necessary for proper communications and data exchange between IEDs. A protocol is defined as "a set of conventions governing the treatment and especially the formatting of data in an electronic communications system." In other words, protocols provide the translation or common language for IEDs to communicate and exchange data with one another. There are numerous protocols that one can use in integration applications, ranging from vendor-specific (proprietary) to standard protocols. The standard protocols in the following list are readily available:

- MODBUS Serial and MODBUS TCP (Administered by a vendor-independent standards group [MODBUS-IDA] and considered a market standard because of its simplicity and popularity.)
- Distributed Network Protocol, DNP3 and DNP/IP (Administered by a vendor-independent standards group [DNP3 User Group].)
- DeviceNet (uses Common Industrial Protocol, CIP) (Administered by the Open DeviceNet Vendor Association, ODVA, an international organization of automation companies.)

MODBUS Serial and MODBUS TCP

All MODBUS data requests consist of a slave device ID or address, a function code, data, and a checksum. MODBUS TCP/IP uses TCP/IP and Ethernet to carry the MODBUS messaging structure. The MODBUS TCP/IP specification is "open," meaning that it is designed so that the MODBUS protocol works on networks built from off-the-shelf and existing industrial Ethernet components and does not require hardware from a specific vendor.

The network at the top of Figure 1 shows a traditional PLC (MODBUS Master) communicating with MODBUS Slave devices (IEDs) and, in turn, passing the collected data to a personal computer (PC) supporting the HMI. The network at the bottom of Figure 1 illustrates another example where the PC directly polls each of the IEDs, performs the HMI function, and supports a software application that performs the programmable logic that is executed in the PLC in the network pictured at the top of the figure.

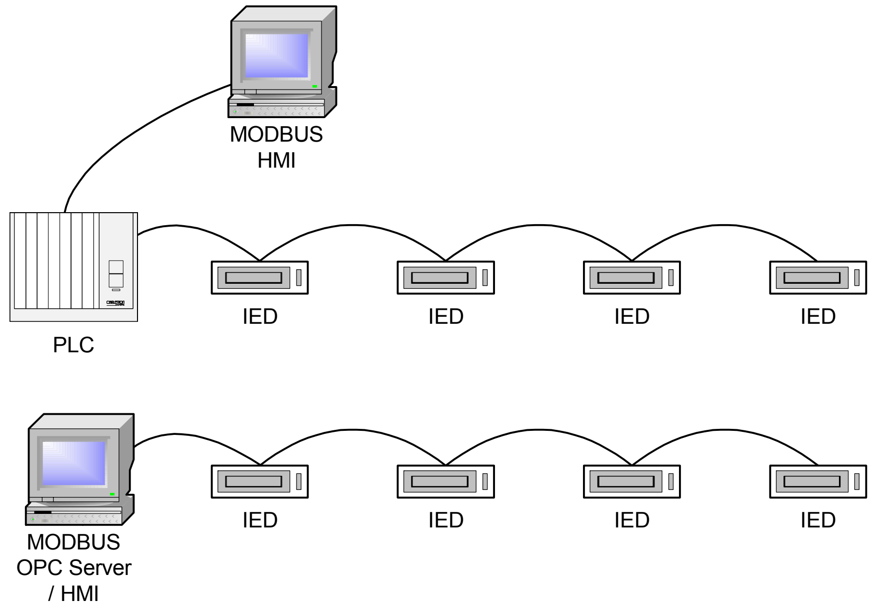


Figure 1 Two MODBUS Network Examples

DNP

Distributed Network Protocol (DNP) is also known as DNP3. DNP3 is a nonproprietary protocol maintained by a users group of vendors and end users, rather than a proprietary protocol maintained by just one vendor. DNP3 was designed specifically to support typical SCADA communications over low-bandwidth communications channels. Its support of several data-acquisition methods, including the report-by-exception method, which transfers only the data that has changed since the last poll, makes DNP very efficient. In addition, all data are referred to by labels that represent objects, such as analog value, status value, and alarm value objects, rather than by labels that represent memory locations, as in MODBUS. Users can request each class of objects (analog, status, alarm, etc.) separately without needing to know how or where the data is stored in the source IED. Unlike MODBUS, DNP3 provides event-time-stamping of the data, so that time-tagged change-of-state and analog rate-of-change data from the IED can be transferred with the appropriate time stamp.

DNP3 supports many communications media and architectures, including both serial and Ethernet connections. The DNP3 protocol continues to evolve with Ethernet applications via DNP over IP.

DeviceNet

DeviceNet is an open, low-level network that connects industrial IEDs to higher-level devices such as PLCs and computers. The DeviceNet network uses Common Industrial Protocol (CIP) for control, configuration, and data acquisition. DeviceNet interoperates with multiple vendor IEDs.

ODVA manages DeviceNet technology built on CIP, and develops DeviceNet specifications. This protocol is not intended to be vendor specific and is considered nonproprietary. However, because it relies on specific technology to create the networks, it is not an open network. ODVA promotes the worldwide adoption of DeviceNet products in industrial automation.

OPC

As mentioned previously, OPC defines a rule set for data exchange between software applications. Leading worldwide automation suppliers collaborated with the Microsoft® Corporation to develop the standards specifications known as OPC. Their goal was to provide data to a variety of end users from assorted data sources. OPC is the common link between applications acting as data sources and applications acting as data consumers; examples of both source and consumer include HMIs and information storage/retrieval systems. OPC was designed to work with a single computer or between applications on multiple computers networked together. New technologies, such as wide area networks (WANs) and faster network connections, make OPC very effective over distributed networks.

Before OPC technology, the distinction between hardware developers and software developers was clearly defined. Software developers developed their own applications and drivers (software interface) to various hardware devices. For example, an HMI developer not only created the HMI package but also provided drivers to a PLC or some other IED acting as the data source. The development of OLE/component object model (COM) brought the software and hardware realms together. Today, software developers can create applications without worrying about compatibility issues with hardware devices. Now, hardware developers need develop only one set of component software for a device that supports applications from many software developers. A COM enhancement, distributed component object model (DCOM) provides access for client applications to remote OPC servers. Figure 2 and Figure 3 illustrate the concept of OPC technology. Figure 4 illustrates the physical layout of implementing OPC technology. This figure shows data from field devices being converted to OPC data via the OPC server and made available to OPC applications via OPC tags.

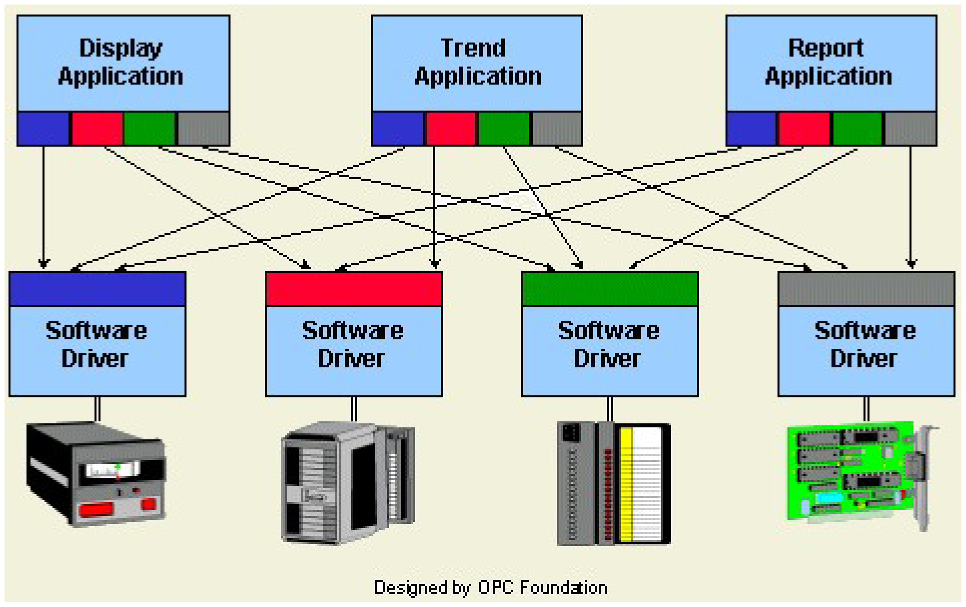


Figure 2 Data Exchange via Unique Protocols on Direct Connections for Each Device

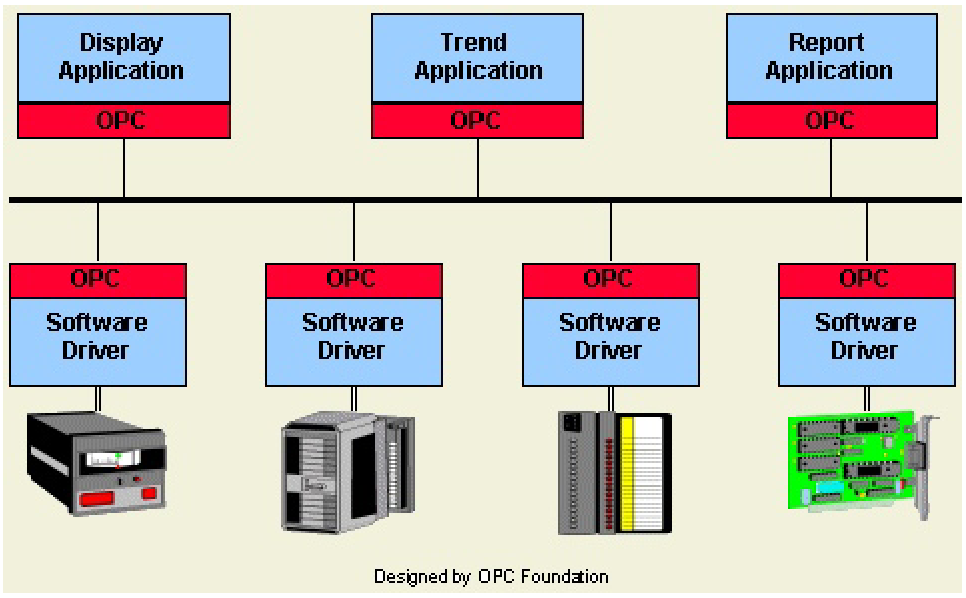


Figure 3 Data Exchange via Distributed OPC Applications on Shared Logical Network

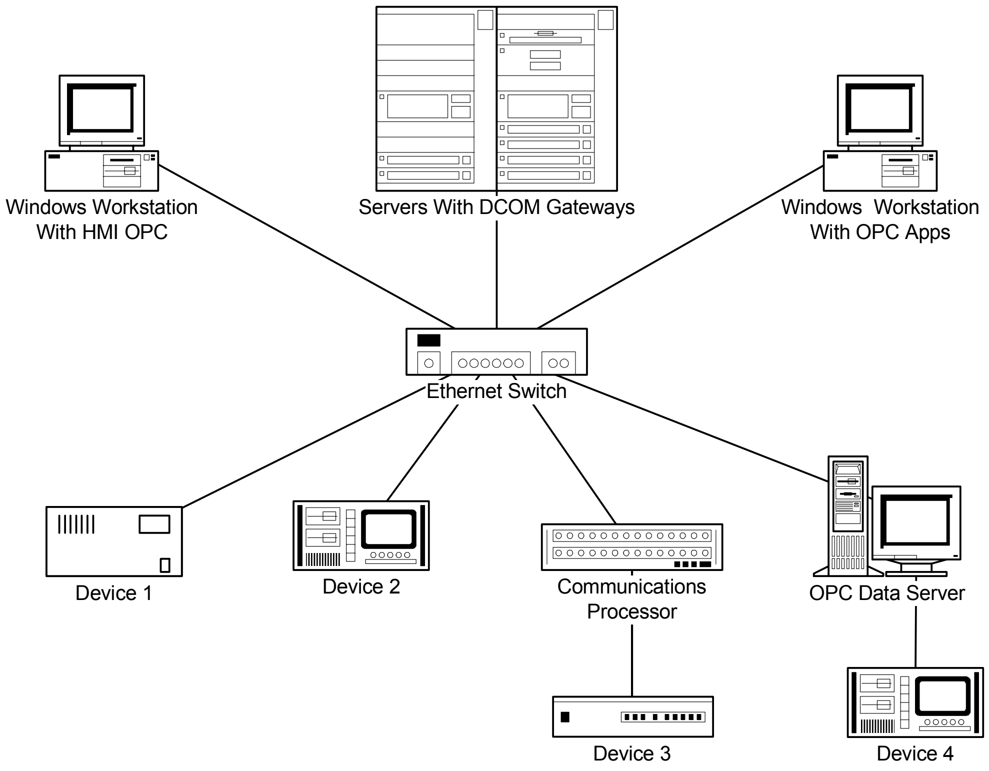

Windows Workstation With HMI OPC
Servers With DCOM Gateways
Windows Workstation With OPC Apps
Ethernet Switch
Device 1
Device 2
Communications Processor
OPC Data Server
Device 3
Device 4

Figure 4 Data Exchange via Distributed OPC Applications on Shared Network; Other Ethernet Traffic Is Not Affected by OPC Traffic

OPC is widely used because its open standards support open connectivity. The OPC Foundation promotes interoperability through creation and maintenance of open-standard specifications, and by adapting and creating standards to meet evolving industry needs.

At present, there are nine OPC standards existing or under development:

OPC Data Access	Moves real-time data from PLCs, Digital Control Systems (DCSs), and other control devices to HMIs and other display clients.
OPC Alarms & Events	Provides alarm and event notifications on demand (in contrast to the continuous data flow of OPC Data Access). These notifications include process alarms, operator actions, informational messages, and tracking/auditing messages.
OPC Batch	Carries the OPC technology to the specialized needs of batch processes. This standard specifies interfaces for exchanging equipment capabilities (corresponding to the S88.01 Physical Model) and present operating conditions.
OPC Data eXchange	Transports data from client/server to server-to-server with communication across Ethernet networks. This standard provides multivendor interoperability, remote configuration, and diagnostic and monitoring/management services.
OPC Historical Data Access	Provides access to data that is already stored (in contrast to OPC Data Access, which provides access to real-time, continually changing data). From a simple serial data logging system to a complex SCADA system, historical archives can be retrieved in a uniform manner.
OPC Security	Specifies client control access to plant process servers in order to protect this sensitive information and to guard against unauthorized modification of process parameters. All the OPC servers provide information that is valuable to the enterprise and, if improperly updated, could have significant consequences to plant processes.
OPC XML-DA	Provides flexible, consistent rules and formats for exposing plant floor data using eXtensible Markup Language (XML), leveraging the work done by Microsoft and others on XML Simple Object Access Protocol (SOAP) and Web Services.
OPC Complex Data	Allows servers to expose and describe more complicated data types such as binary structures and XML documents. This standard specifies a companion specification to Data Access and XML-DA.
OPC Commands	Specifies a new set of interfaces that allow OPC clients and servers to identify, send, and monitor device control commands.

SOFTWARE AND HARDWARE REQUIREMENTS FOR OPC

One of the advantages of OPC is flexibility in implementation. This section gives an overview of software and hardware requirements.

Software

OPC technology is based on Microsoft COM/DCOM, which is an open, published protocol that links software applications. These components are readily available in Microsoft operating systems from Windows® 95 to Windows XP. Any OPC server that is COM/DCOM compliant can be installed in these Microsoft operating systems. The choice of an operating system depends on OPC package support and security issues.

Hardware

There are many vendors providing OPC server and client packages; each package has a set of minimum requirements that guides hardware selection. Hardware selection also depends on the environment in which the OPC software operates. Three widely used hardware types are home/office personal computers, industrial computers, and rugged computers.

Home/Office Personal Computers: Home/office PCs range from personal desktop computers to laptops that are being used in an office or at home. These types of computing machines are readily available from computer retail stores or mail order and generally run Microsoft Windows operating systems. These computers are ideal for testing automation applications in well-controlled environments.

Industrial Computers: Unlike the home/office PC, industrial computers are built to withstand moderately harsh environments that may include vibration, wide temperature range, and large air contaminants. Industrial computers provide better reliability than PCs. These computers have a design similar to home/office PCs with the addition of components that perform better in harsh environments: redundant power supplies, additional fans, air filters, and shock-mounted drive cages. Industrial computers often support Microsoft Windows, embedded Microsoft, or embedded Linux operating systems. Embedded operating systems have smaller, specific feature sets, can lock out unwanted features, and need less memory, which leads to more compact and more power-efficient applications.

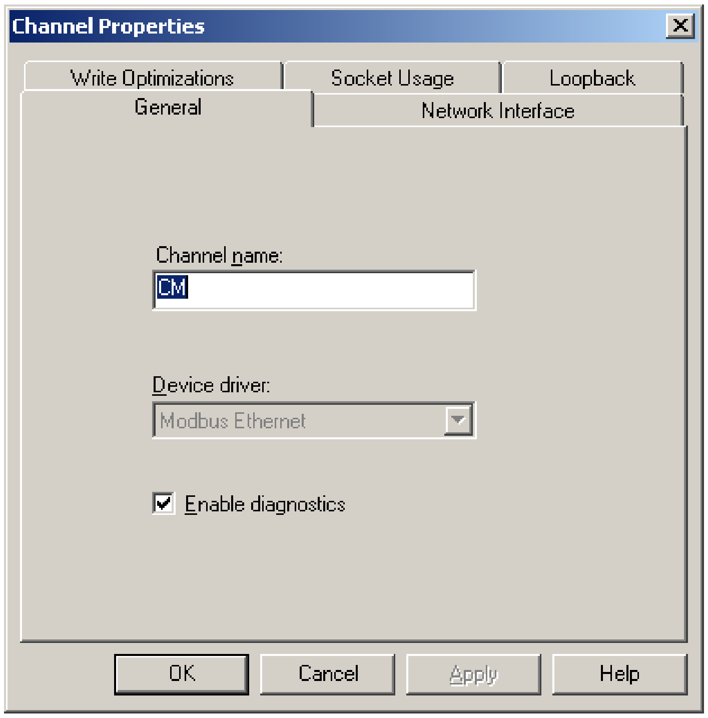
Rugged Computers: Unlike PCs and industrial computers, rugged computers are built to withstand extremely harsh environments such as excessive vibration, extreme temperature ranges, moisture, and small particulate air contaminants. Rugged computers are used for mission-critical applications where reliability and low failure rates are essential. These computers have a very different design from PCs and industrial computers. They are designed for extremely harsh environments with special purpose power supplies, protection and isolation on power and communications connections, redundant serial, USB, and network connections, diagnostic LEDs, no moving parts, and no vent holes. In addition, components can be selected and placed to avoid adverse reaction to vibration. Rugged computers support Microsoft Windows, embedded Microsoft, or embedded Linux operating systems.

Collecting Data From OPC Servers

OPC servers are software applications that run in a Windows or embedded operating system, transmit a protocol through a communications connection, and translate the data received into OPC. These applications also accept OPC data from other applications and translate it into appropriate protocol messages.

Some common OPC servers communicate with IEDs that support DNP, MODBUS, and DeviceNet, and then create tag databases. Tag is a term used to describe an element in a software application that represents all the data associated with a data point, including present and past values, time stamps, and attributes of the point and the data.

Once the MODBUS server is launched, the channel is selected. The channel represents the device connected on the Ethernet port of the PC. In this case, the channel is assigned the name CM, and the MODBUS Ethernet driver is also selected (see Figure 5).



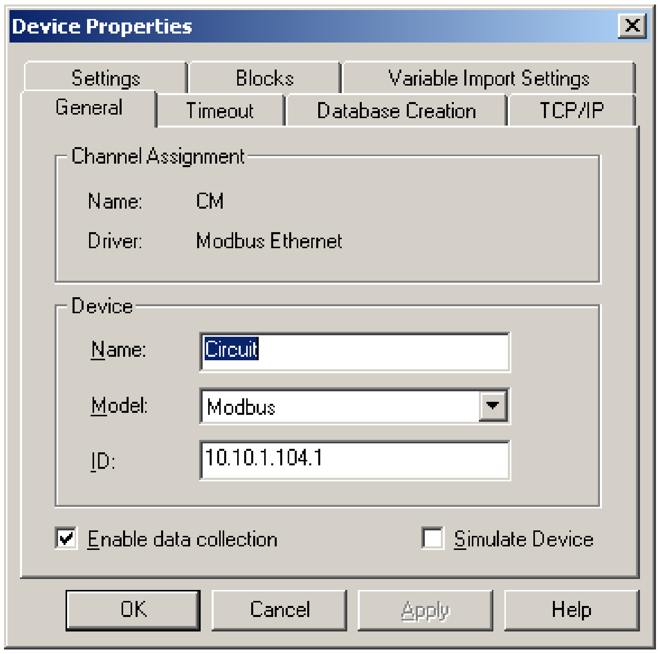
Channel name: CM

Device driver: Modbus Ethernet

[✓] Enable diagnostics

Figure 5 Selecting the Channel Name and Driver

After channel selection, the device is selected and assigned a name, in this case, Circuit (see Figure 6). The device is also assigned an IP address, along with the “model” (driver). The TCP/IP port assigned is 502.



Channel Assignment

Name: CM

Driver: Modbus Ethernet

Device

Name: Circuit

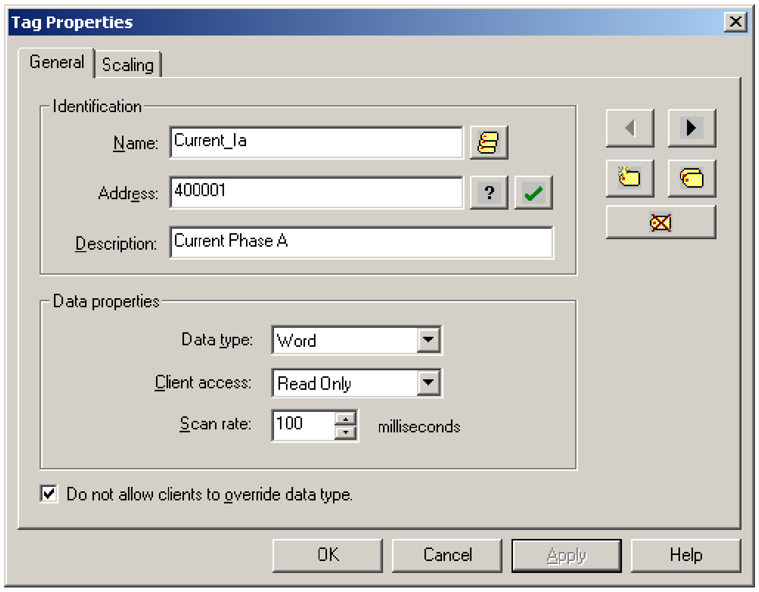
Model: Modbus

ID: 10.10.1.104.1

☑ Enable data collection ☐ Simulate Device

Figure 6 Selecting the Device Name and Address

After device selection, various tags are created. Figure 7 shows the window for defining the tag name, (Current_Ia).



Identification

Name: Current_Ia

Address: 400001

Description: Current Phase A

Data properties

Data type: Word

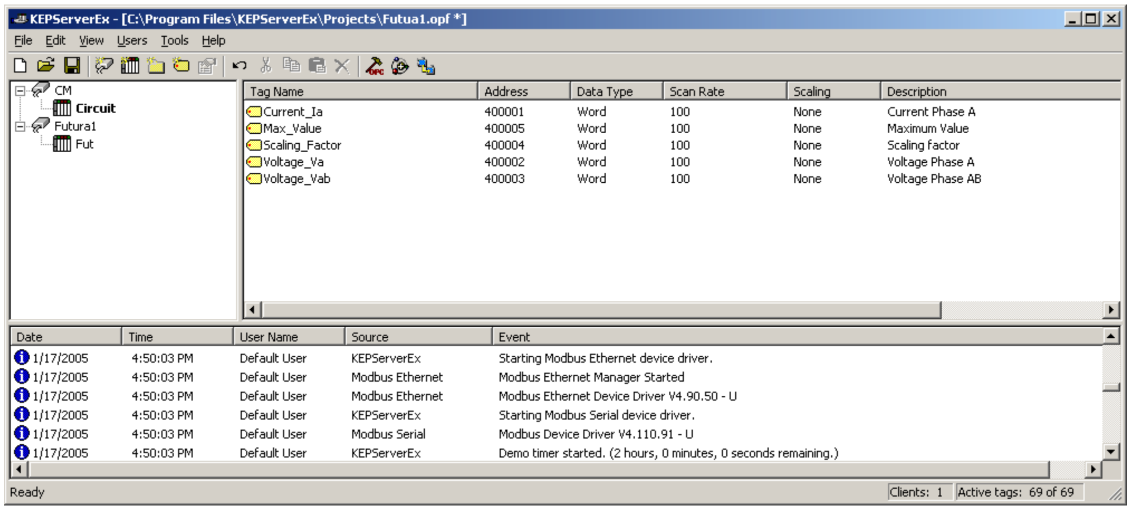
Client access: Read Only

Scan rate: 100 milliseconds

☑ Do not allow clients to override data type.

Figure 7 Selecting the Tag Name, Address, and Other Properties

Figure 8 illustrates a screen display of communications settings used by the OPC Server so that it can communicate with the IED. Each device has a MODBUS address or index number, which is obtained from the device manual. The other communications settings for each IED include Address Locations Being Polled, Data Type, and Scan Rate.


KEPServerEx - [C:\Program Files\KEPServerEx\Projects\Futua1.opf *]
File Edit View Users Tools Help
CM
Circuit
Futura1
Fut
Ready
Clients: 1
Active tags: 69 of 69

Tag Name	Address	Data Type	Scan Rate	Scaling	Description
Current_Ia	400001	Word	100	None	Current Phase A
Max_Value	400005	Word	100	None	Maximum Value
Scaling_Factor	400004	Word	100	None	Scaling factor
Voltage_Va	400002	Word	100	None	Voltage Phase A
Voltage_Vab	400003	Word	100	None	Voltage Phase AB

```
Date        Time        User Name     Source            Event
1/17/2005   4:50:03 PM  Default User  KEPServerEx       Starting Modbus Ethernet device driver.
1/17/2005   4:50:03 PM  Default User  Modbus Ethernet   Modbus Ethernet Manager Started
1/17/2005   4:50:03 PM  Default User  Modbus Ethernet   Modbus Ethernet Device Driver V4.90.50 - U
1/17/2005   4:50:03 PM  Default User  KEPServerEx       Starting Modbus Serial device driver.
1/17/2005   4:50:03 PM  Default User  Modbus Serial     Modbus Device Driver V4.110.91 - U
1/17/2005   4:50:03 PM  Default User  KEPServerEx       Demo timer started. (2 hours, 0 minutes, 0 seconds remaining.)
```

Figure 8 MODBUS OPC Server Configuration Screen Illustrating Address Locations Being Polled, Data Type, and Scan Rate

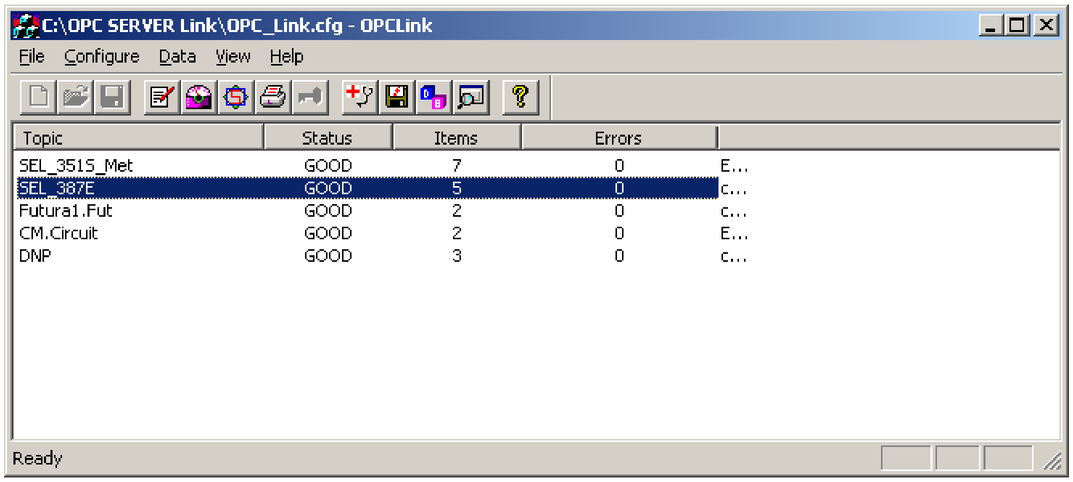
Creating tags in an OPC server varies from one vendor to another, however, there are basic similarities. A tag is composed of Tag Identification and Data Properties. Tag Identification provides information about the tag: tag name, address of the source from which the tag derives its data, and description. Data Properties provides the data type (such as word, float, Boolean, etc.), the access type (either read/write or read only), and the scaling properties of the value of the tag.

Once configured in the project, each protocol OPC server collects data from the respective devices, and then makes these data available as OPC tags to the OPC HMI client software. The HMI software presents the data as visual information. The link between OPC Servers and HMI is the OPC Link, an application that falls under the category of protocol gateway or converter. In our application it serves as a bridge between the OPC server and the Wonderware (HMI) Suitelink protocol. However, because it is accessing an OPC server, it also functions as an OPC client. OPC Link provides information on how to access the OPC server whether the server resides on a local or a remote computer. This information is known as the Topic Definition.

Figure 9 illustrates the topic definitions that were created to access the various OPC servers. This figure also shows the status of the communications channel between the HMI client and the OPC Server via the OPC Link, the number of tags (items) being accessed on the OPC server, and whether there are any errors during the operation of OPC Link.

The serial port expander converts the USB port on the computer into four serial ports. This device enumerates itself as standard communications ports. The maximum baud rate on these ports is 115 kbps.

Two scenarios were developed for this paper. The first one involved installing the OPC Client (HMI) and the OPC Server on a single computer. The second option, discussed later, involved having the OPC Client installed on a separate computer.


C:\OPC SERVER Link\OPC_Link.cfg - OPCLink
File Configure Data View Help
Ready

Topic	Status	Items	Errors	
SEL_351S_Met	GOOD	7	0	E...
SEL_387E	GOOD	5	0	c...
Futura1.Fut	GOOD	2	0	c...
CM.Circuit	GOOD	2	0	E...
DNP	GOOD	3	0	c...

Figure 9 OPC Link Communication Diagnostics

Figure 10 illustrates a stand-alone OPC Server with a built-in HMI. This application is considered stand-alone because it performs both functions. The screen capture in Figure 11 shows the single HMI for various devices with different protocols.

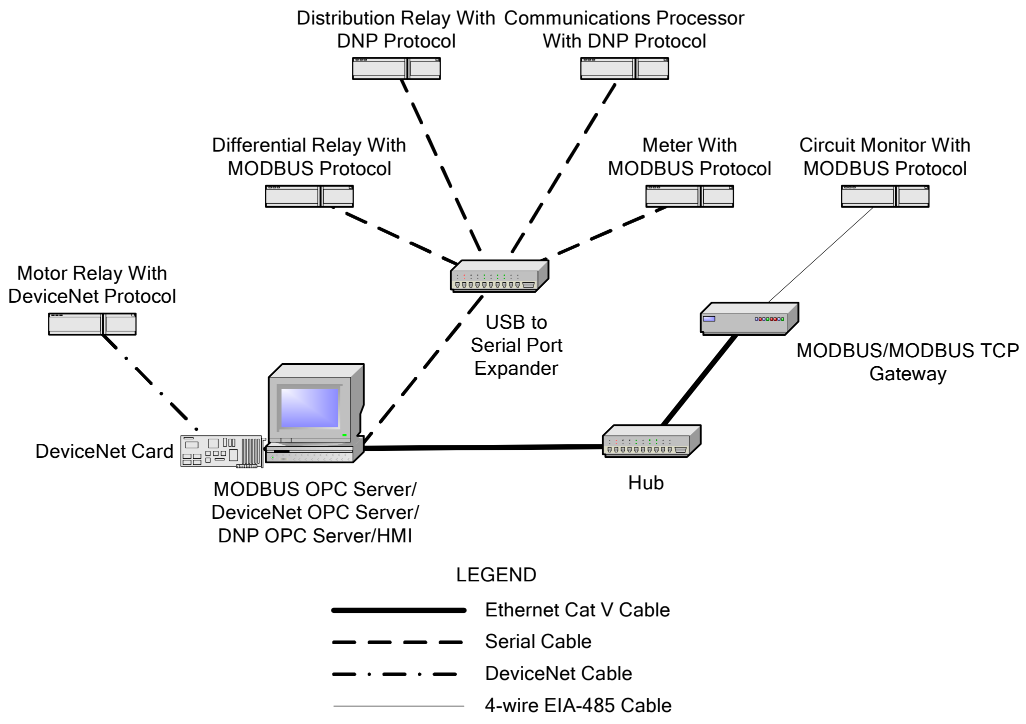
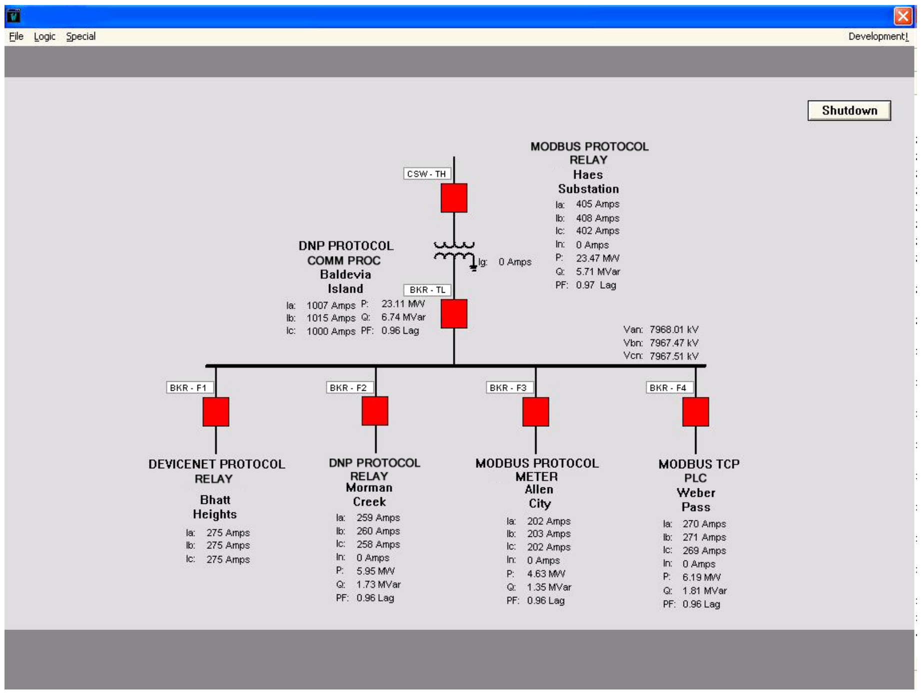

Distribution Relay With DNP Protocol
Communications Processor With DNP Protocol
Differential Relay With MODBUS Protocol
Meter With MODBUS Protocol
Circuit Monitor With MODBUS Protocol
Motor Relay With DeviceNet Protocol
USB to Serial Port Expander
MODBUS/MODBUS TCP Gateway
DeviceNet Card
MODBUS OPC Server/ DeviceNet OPC Server/ DNP OPC Server/HMI
Hub
LEGEND
Ethernet Cat V Cable
Serial Cable
DeviceNet Cable
4-wire EIA-485 Cable

Figure 10 OPC Server and HMI on the Same Computer



```
MODBUS PROTOCOL
RELAY
Haes
Substation
Ia:  405 Amps
Ib:  408 Amps
Ic:  402 Amps
In:  0 Amps
P:   23.47 MW
Q:   5.71 MVar
PF:  0.97 Lag

Ig:  0 Amps

DNP PROTOCOL
COMM PROC
Baldevia
Island
Ia:  1007 Amps  P:   23.11 MW
Ib:  1015 Amps  Q:   6.74 MVar
Ic:  1000 Amps  PF:  0.96 Lag

Van:  7968.01 kV
Vbn:  7967.47 kV
Vcn:  7967.51 kV

DEVICENET PROTOCOL
RELAY
Bhatt
Heights
Ia:  275 Amps
Ib:  275 Amps
Ic:  275 Amps

DNP PROTOCOL
RELAY
Morman
Creek
Ia:  259 Amps
Ib:  260 Amps
Ic:  258 Amps
In:  0 Amps
P:   5.95 MW
Q:   1.73 MVar
PF:  0.96 Lag

MODBUS PROTOCOL
METER
Allen
City
Ia:  202 Amps
Ib:  203 Amps
Ic:  202 Amps
In:  0 Amps
P:   4.63 MW
Q:   1.35 MVar
PF:  0.96 Lag

MODBUS TCP
PLC
Weber
Pass
Ia:  270 Amps
Ib:  271 Amps
Ic:  269 Amps
In:  0 Amps
P:   6.19 MW
Q:   1.81 MVar
PF:  0.96 Lag
```

Figure 11 Devices with Disparate Protocol Connected to a Single HMI

NETWORKING OPC

OPC becomes very powerful in a networking environment, which allows installation of the OPC server on one computer and an OPC client, such as an HMI, on another. The first implementation in this paper illustrated a stand-alone OPC system where server and client were on the same computer. This type of installation is useful in substation automation applications or in any application in which an HMI (OPC client) is needed near the field devices (IEDs). However, there are many applications where field devices are scattered across a plant or where field device environments are too harsh for HMI computers. In these applications, Distributed Component Object Model (DCOM), mentioned previously, becomes important.

DCOM provides the means of connecting OPC clients to remote OPC servers by configuring the DCOM security settings for each component, both client and server. DCOM is based on providing permissions for a user or group of users to access/launch a particular component that resides on a remote computer. Using DCOM, which communicates through an Ethernet network, may require the assistance of the network administrator, especially for networks with security devices, such as firewalls. Figure 12 is a simplified illustration of how DCOM works.

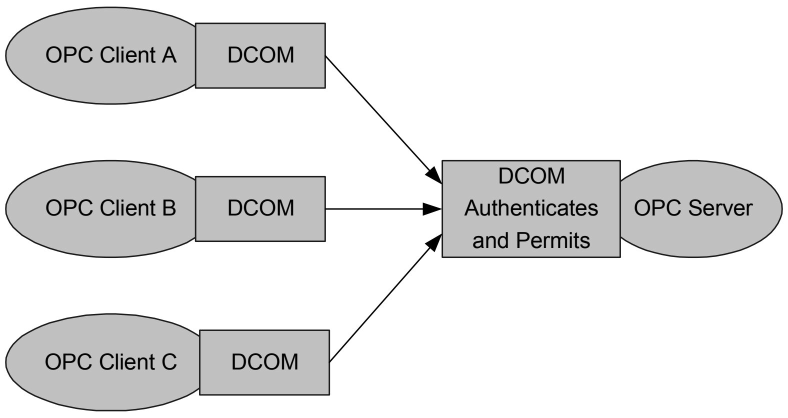


Figure 12 DCOM Security

When configuring an OPC server for remote access, follow the specific vendor's recommended DCOM security settings. In some cases, the server software will actually ask, during installation, whether you would like to modify the DCOM settings so that it can be accessed across a network. The following section discusses general settings in DCOM and accessing DCOM. We also share our experiences with DCOM.

DCOM Configuration on OPC Server PC and OPC Client PC

To configure DCOM for general settings in Windows NT/2000 and XP, first launch the DCOM application called "dcomcnfg.exe" The following Microsoft Knowledge Base articles provide detailed information about launching this particular application for other Windows operating systems.

176799—INFO: Using DCOM Config (dcomcnfg.exe) on Windows NT

182248—How To Use DCOM Config (dcomcnfg.exe) with Windows 95/98/Me

After launching the DCOM application you should have a screen similar to Figure 13.

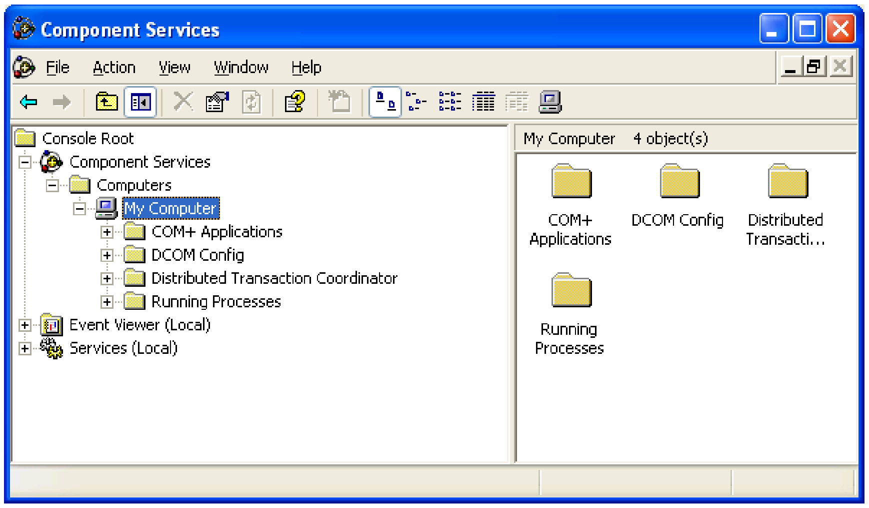


Figure 13 Component Services Screen

The settings can be configured globally, which changes the access settings for all components, or on a component level. Configuring the security on a component level allows particular servers to be accessed remotely. The security properties for each component (OPC server) fall into two categories: Access Permissions and Launch Permissions. With these categories, you can make the OPC server available to everyone or to a particular user or group of users.

The security settings, Authentication and Impersonation, allow remote clients to access the computer where the OPC server resides before the application/component (a DNP OPC server or MODBUS OPC server). As indicated in the following tables, combinations of these two settings can result in no security checking or every access/request checking.

These tables summarize the settings used to configure the DCOM of the OPC server.

Computer Properties

Authentication	Impersonation	Comments
None	Anonymous	No security checking
Connect	Identify	Verification only on the initial connection
Call	Identify	Security check on every call during connection

Application Properties

Security Launch	Security Access	Comments
Customize	Customize	Specify user or groups

DCOM uses Microsoft Windows networking technology of usernames and passwords, so consider whether logging in as a local user account or a domain user account will provide the most appropriate access privileges.

Migrating Stand-Alone OPC to Networked OPC

The OPC installation described earlier in this paper was a stand-alone application in which the OPC client and server resided on the same computer. To convert this stand-alone application to a network application, we transferred the OPC client application (HMI) to a remote computer and connected them via an Ethernet network. This new application demonstrated remote OPC connectivity by connecting two separate remote field offices; one in Belleville, IL and one in Franklin, TN.

Both computers belong to the same domain and are authenticated by a domain controller. We used the same username (belonging to a User or Power User group) and password on both computers, which ensures that the applications that are installed and running on both machines (OPC server and OPC client) have similar permissions and access. This system restricts these applications so that they can only be launched and accessed by this particular user. Authentication is already completed because both computers are on the same domain.

Next, we configured the client computer to refer to a remote node (remote OPC server) instead of a local host (i.e., stand-alone application). Figure 14 illustrates the configuration of the OPC client for a remote application.

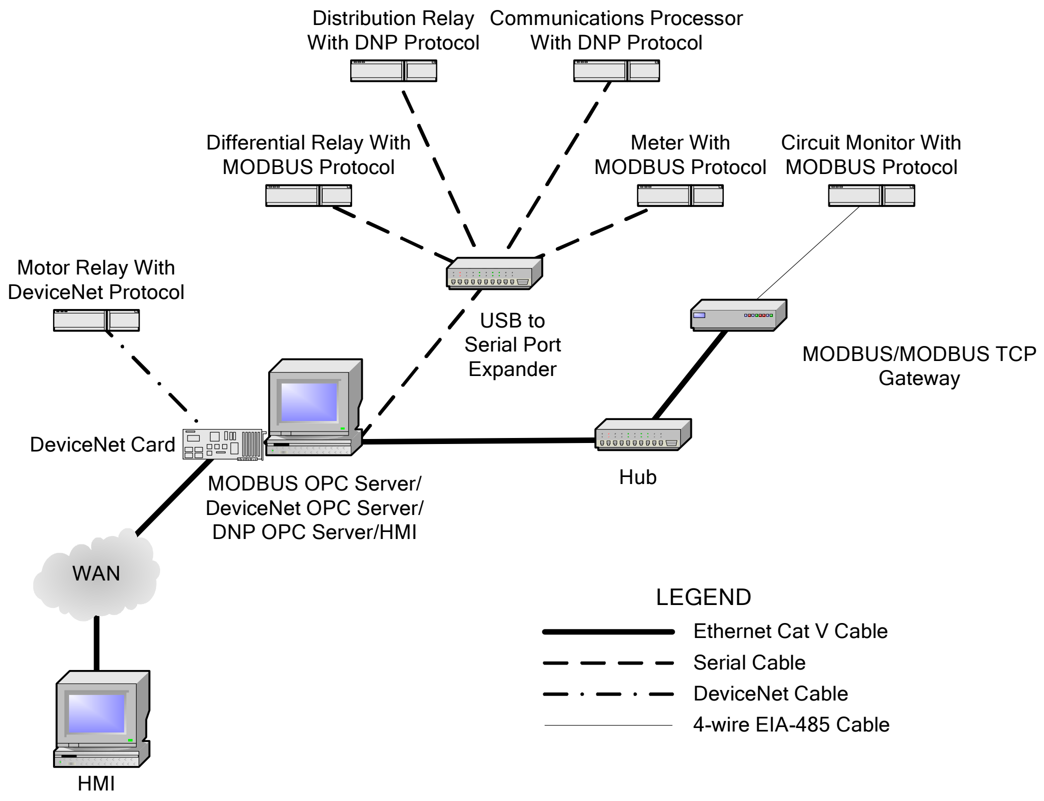

Distribution Relay With DNP Protocol
Communications Processor With DNP Protocol
Differential Relay With MODBUS Protocol
Meter With MODBUS Protocol
Circuit Monitor With MODBUS Protocol
Motor Relay With DeviceNet Protocol
USB to Serial Port Expander
MODBUS/MODBUS TCP Gateway
DeviceNet Card
MODBUS OPC Server/ DeviceNet OPC Server/ DNP OPC Server/HMI
Hub
WAN
HMI
LEGEND
Ethernet Cat V Cable
Serial Cable
DeviceNet Cable
4-wire EIA-485 Cable

Figure 14 OPC Server and HMI on Different Computers Connected Across WAN

Connecting to remote OPC servers can be an easy process or a very technical one, depending upon your application. The implementation involves mainly determining the type of network to use, the type of security, and the network structure. Although DCOM is readily available in Microsoft Windows operating systems, there are third-party applications that can make this remote connectivity simple and there are network practices (tunneling, VPN) that can assist in strengthening your security issues.

CONCLUSIONS

1) OPC technology is a powerful tool for integrating a variety of devices into one common platform.
2) OPC technology is easy to implement and configure, and has endless options. It is possible to integrate several protocols into one device and use OPC to provide information to assorted data consumers.
3) OPC provides the means of transporting field data to computing systems through various communication infrastructures via Ethernet LAN (local area networks) or WAN (wide area networks).

4) OPC communicating over a WAN requires an assessment of network and security issues. Following vendors' requirements may or may not provide remote access.

5) OPC servers vary from vendor to vendor but, in general, all provide the same result. Use vendor demonstrations to determine the server that meets your specifications: ease of use, protocol compatibility, etc.

6) First time OPC users should refer to the OPC Foundation website (www.opcfoundation.org) for a listing of OPC products, vendors, and various documentation.

SOURCES

Ken Curtis, DNP Users Group, "A DNP3 Protocol Primer," http://www.dnp.org/files/dnp3_primer.pdf

"Introduction to OPC and other OPC Documentation," OPC Task Force, OPC Foundation, "OPC Overview," http://www.opcfoundation.org

"What is OPC? Basic Principles and Advantages," OPC Foundation, http://www.merz-sw.com

"DCOM for OPC," Data Layers Limited, http://www.opcware.com/DCOM.html

"DNP IO Server Instruction Manual for DNP IO Server," http://www.ioserver.com

"DeviceNet Technical Overview," OpenDeviceNet Vendor Association (ODVA), http://www.odva.org/10_2/05_tech/PUB00026R1.pdf

"Kepware OPC Instruction Manual for KepServerEX," Kepware Technologies, http://www.kepware.com/Support_Center/helpfiles.htm

"Designing and Configuring a DeviceNet Network Using RSNetWorx for DeviceNet Software," Rockwell Automation Student Manual, Rockwell Automation, 2004. www.rockwell.com

"SST DeviceNet PCMCIA Card Instruction Manual for SST DeviceNet PCMCIA Card," Woodhead Industries, http://www.mysst.com/tech/content.asp

"Square D Instruction Manual for PowerLogic Series 4000 Circuit Monitor (CM4000) and Ethernet Gateway (EGX400)," Square D, http://www.squared.com

"Instruction Manual for the Futura Meter," Electro Industries/Gauge Tech, http://www.electroind.com

"InTouch 8.0 Application and Documentation," Invensys Systems, http://www.wonderware.com

Modbus-IDA organization. http://www.modbus.org

Figures 4 and 5 reprinted with permission from OPC Foundation 2004.

Biographies

Pankaj Bhatt has a B.S. in Electrical Engineering from M.S. University of Baroda in India in 1984. He earned his M.E. in Electrical Engineering from City College of City University of New York in 1987 and his M.S. in Electric Power Engineering from Rensselaer Polytechnic Institute in Troy, New York in 1989.

Mr. Bhatt is an international integration application engineer with Schweitzer Engineering Laboratories. Before joining SEL in 2003, Mr. Bhatt designed, configured, and installed control systems for fossil power plants and gas turbines using Distributed Control System (DCS) at Siemens Westinghouse.

Prior to that he was involved with system integration including load shedding, demand limiting, and power monitoring projects for SquareD PowerLogic. As an employee of New York State Electric and Gas, he developed engineering designs and coordinated start-up of DCS, programmable logic controllers, 5 kV-class switchgear, and motor control centers for various power plant operations. He obtained his Professional Engineering license in New York in 1994. He is a member of both IEEE and ISA.

Rogelio P. Baldevia, Jr. earned his BS in Electrical Engineering from Seattle University in 1993. That same year, he started working for Guam Power Authority's engineering department as a systems planning engineer and later transferred into operations as a SCADA engineer. His experience encompasses communication technologies, systems planning studies, and energy management systems.

Mr. Baldevia joined SEL in 2003 as an integration application engineer. His responsibilities include providing technical support, as well as assisting and training SEL customers on their integration and automation applications. Mr. Baldevia obtained his Professional Engineering license in 1997 from the U.S. Territory of Guam.