OLE for Process Control (OPC) Overview

This paper provides an overview of the OPC standard and discusses the benefits of OPC for vendors and end users.

Contents

Figures

Introduction

OLE for Process Control (OPC), a new industry standard will benefit vendors and end users in the Process Control community. OPC defines a standard interface that allows applications to access data from a variety of process control devices. The applications must implement one OPC compliant driver to access data from any OPC compliant server.

The History of OPC

The OPC specification development effort has its roots in a Microsoft industry focus group known as WinSEM (Windows for Science, Engineering, and Manufacturing). This is a diverse group of companies with a common interest in developing products using Microsoft technology. Slow progress in developing a standard indicated that this was not the most efficient forum for standards development. Five companies decided to take the initiative to develop a draft of an open standard for submission to the Process Control industry. By keeping the group small and specifying limited functionality in the initial release, it was thought that a standard could be developed in a relatively short span of time. This turned out to be a valid approach. The standard was developed in one year with input from Process Control Industry leaders. This is interesting but doesn't have anything to do with OPC.

OPC Technology

The OPC standard specifies an interface between process data client applications and servers. The standard was purposely limited to the reading and writing of process values. Alarm handling, process events, security, batch structures, and historical data access were all deferred to subsequent releases. An OPC compliant client can read and write data to any OPC compliant server.

OPC is based on the OLE/COM standard from Microsoft. OLE/COM was designed by Microsoft to be extensible by others. This allowed OPC to be developed on top of an existing technology, rather than inventing a completely new technology. It also provided a large number of clients for the OPC server data in the applications that are already OLE aware.

OPC provides for a high degree of interoperability between client and server applications supplied by different vendors. A good analogy to demonstrate this concept is the PC bus hardware standard. The PC bus allowed hardware from a variety of vendors to inter-operate because it all used the same interface standard. In the same way, OPC allows software modules from a variety of vendors to inter-operate through a standard interface. This concept is shown in Figure 1.

Figure 1. The Software Backplane

The OPC standard defines two interfaces for applications. One interface is designed for high volume/high throughput applications. This is an OLE custom interface. This interface is used by applications developed in C++. A second interface is provided for ease of access to the data. This interface is used by Visual Basic™ (VB) and Visual Basic for Applications (VBA) users. This interface lowers the programming knowledge level required to access data from an OPC server. Engineers familiar with Visual Basic, or macro writing in Excel or Word, have access through this interface. Both interfaces are shown in Figure 2. OPC OLE Interfaces .

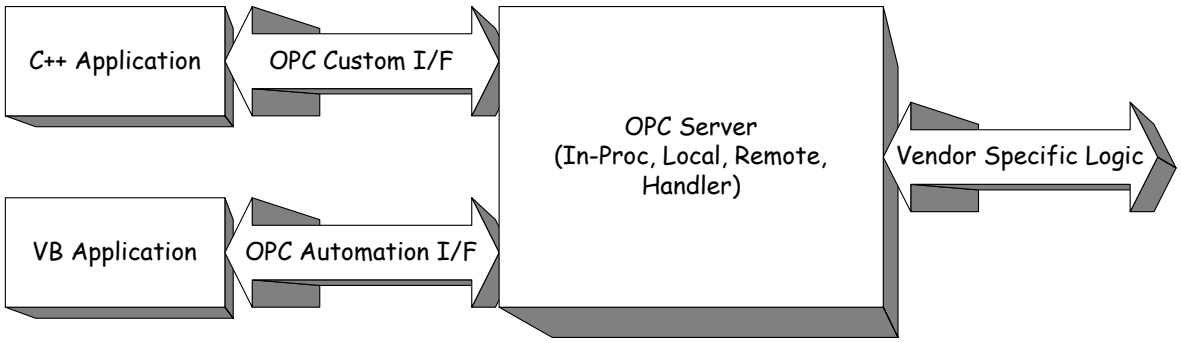

Figure 2. OPC OLE Interfaces

The OPC Standard defines several objects with associated properties and methods. These objects are used to expose process data from a process control device in a standard manner. Every OPC server will make data available to clients using this model.

The first OPC object is the OPC server. The OPC server is responsible for acquiring the data requested by a client from the process control device. Each server contains OPC groups, which are logical collections of data to be gathered. The groups in a server may be shared by many clients or be private to a single client. The relationship between servers and groups is shown in Figure 3. Relationship between servers and groups .

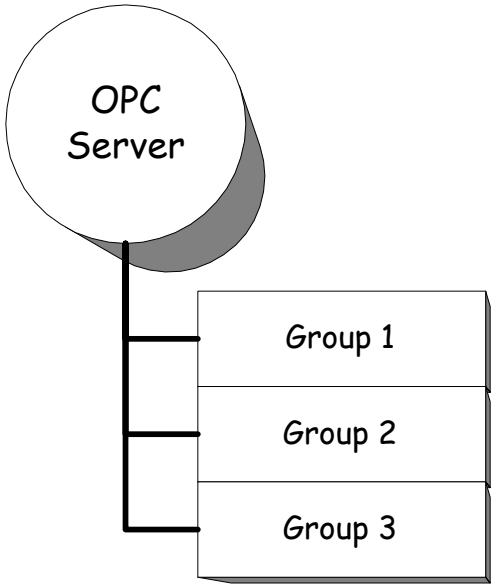

Figure 3. Relationship between servers and groups

The OPC group object holds the OPC items that contain the values retrieved from the process control device. The client adds items to groups. The client may want to group all process temperature values into a group. It might make sense for the client to have all the data from a process unit be organized into a group. The client can specify to the server the update rate for the group that defines the rate at which the server will check to see if the exception limit for a value has changed. The client also specifies the exception limit used for items in the group to determine if the value should be reported back to the client. The relationship between groups and items is shown in Figure 4. Relationship between groups and items .

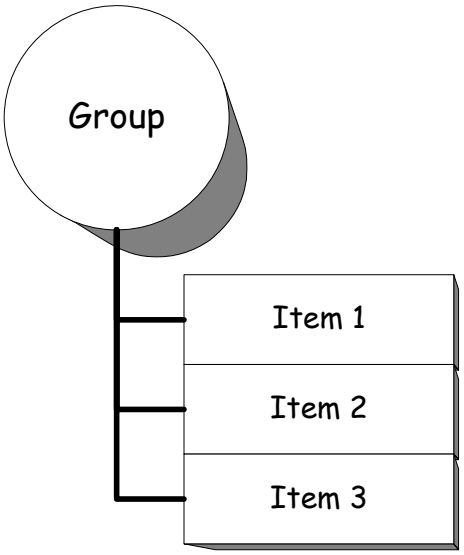

Figure 4. Relationship between groups and items

Items contain the actual value received from the process control device. The client specifies the item to the server through the item ID. The item ID is server specific. It uniquely identifies to the server how to locate the value in the process control device. For a DCS system this might resemble TIC-200.PV. For a PLC system the item ID might be a register number. The items typically return a single value. The server may optionally return a timestamp and quality flag with the value.

Long-term Direction

The OPC Foundation™, a nonprofit organization, is currently working on future releases of the OPC specification. This organization is chartered with the expansion of the standard into those areas deferred from the initial release. These areas are: alarm handling, event handling, security, batch structures, and historical data access. Membership in this organization is open to all interested parties. Members are forming into subcommittees to address areas in their area of expertise or where they have specific interests. Parallel development of different areas of the standard should help in keeping the OPC specification a rapidly expanding standard.

Vendor Benefits

In the past, client application vendors had to develop a different driver for each control device. The OPC standard provides the greatest benefit to client vendors in that they now only have to develop one driver to access data from a process control device. Both of these situations are diagrammed in Figure 5.

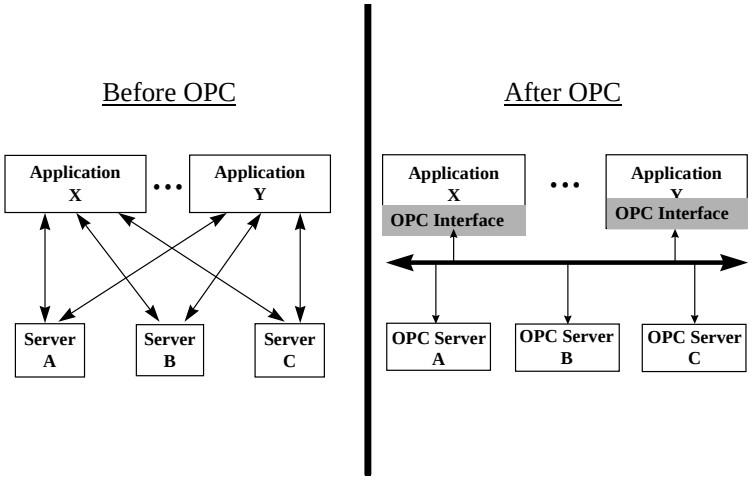

Figure 5. Before and after OPC

If the control device vendor modified the interface to the device, the client vendor would also have to modify the client's driver. OPC isolates the client software from the details of the various systems below the OPC interface. The device vendor is able to modify the functionality under the OPC server interface without affecting the client software. Client vendors can now expend resources on true value added activities for their products in place of the effort required to maintain a library of device drivers.

End User Benefits

Using the OPC specification, end users can chose the client application that best meets their needs. In the past, a user had to use specific client software that provided an interface for a particular control device. With OPC, any OPC compliant client application can interface to a control device with an OPC compliant server. In this way, the user gets the best solution for a particular task.

Another benefit comes from lower integration costs and risks. With plug and play OPC compliant components, available from a variety of vendors, the system integrator can spend more time on the final integration goal and less time developing custom drivers. Since the solution is based on standard OPC components rather than custom drivers, the project risk is lower.

The OPC OLE interface to Visual Basic and VBA provides access to process data for the engineer who is skilled in these tools. To quickly put together a diagnostic application in VB to view specific process data requires using a form builder and only a few lines of basic code. A spreadsheet to derive some report information from process data is easily accomplished through the use of a VBA macro.

Frequently Asked Questions and Answers

What is OPC?

OPC stands for "OLE for Process Control". OPC is a communication standard based on Microsoft's OLE technology that fosters greater interoperability between automation/control applications, field systems/devices,

and business/office applications. OPC defines standard objects, methods, and properties built on OLE technologies for servers of real-time information like DCS, PLCs, historian, and other software applications, and communicates the information they contain to standard OLE-enabled clients.

How will this effort benefit the user/vendor community in the process control and manufacturing world?

For software application users, OPC has the potential to make “plug and play” software a reality. OPC will allow different applications written in different languages running on different platforms to integrate. OPC compliant applications will also integrate with the Microsoft suite of Office and BackOffice products.

Hardware users will also benefit. Process device hardware vendors will develop OPC compliant interfaces, much like PC printer vendors provide printer drivers that are able to work with multiple Windows applications. Users will be able to choose best-in-class products for a given application and easily integrate information from these devices with software applications across the enterprise.

Vendors will benefit by reducing development costs associated with developing multiple drivers to automating systems. Instead, vendors will be able to focus on the value-added product functionality.

Why are extensions to OLE needed for process control applications?

Extensions are needed to ensure interoperability to automation and control equipment users. It is quite likely that two applications claiming OLE/COM-awareness might actually support very different OLE aspects and, therefore, not be interoperable.

OLE for Process Control (OPC) defines standard objects, methods, and properties for meeting the interoperability requirements of real time process automation applications. These requirements include a standard technique for addressing information contained in process control devices/systems, efficient transfer of data from a process device to an application, the ability for a client to use several servers simultaneously, and server-specific configuration support.

By providing extensions to OLE in the form of structure definition, interfaces, and techniques for more efficient data transfer, OPC leverages the advantages of OLE while augmenting OLE for process control and automation applications.

How does OPC relate to fieldbus standards efforts?

OPC compliments fieldbus standardization efforts. Most fieldbus efforts focus on providing secure communications among field devices, while the focus of OPC is on providing communications between applications, and between field devices and applications.

An examination of the work of the Fieldbus Foundation presents an excellent example of how OPC compliments fieldbus efforts. The Fieldbus Foundation is focused primarily on providing a standard that provides secure communications among field devices. In addition, the Fieldbus Foundation is providing Fieldbus Messaging Services (FMS), which are used to integrate field devices to three device-support applications: System Management, Device Definition, and Function Block Service applications. These three fieldbus applications use FMS to communicate up and down the H1 or H2 fieldbus network. However, there is currently no standardized Application Programming Interface (API) above the Function Block Services defined by the Fieldbus Foundation.

The OPC Task Force uses OLE/COM technology fill in this void by specifying objects, methods, and properties allowing many applications to integrate with each other, as well as, integrating many applications to field devices.

How "fast" is the OPC standard (i.e. what "speed" of data transfer does it provide)?

The answer is, IT DEPENDS! OPC is a protocol standard, not a specific implementation scheme. OPC allows a wide variety of implementation schemes including "in-process" / "out-of-process" servers and "remote" / "local" servers. The type of server implemented, as well as the hardware platform used, network environment, etc. can ALL impact the "speed" or performance of the resulting implementation. It is technically possible to implement OPC with a wide range of speed and throughput capacities.

Will OPC run on ANY operating system or computer?

No, OPC is based on Microsoft's OLE environment. Consequently, OPC requires the OLE "COM" technology to be present in the operating system. The OLE "COM" technology is currently available on Microsoft's NT operating system, and Windows 95 operating system. Efforts are currently underway to port OLE "COM" technology to UNIX and other operating systems.

Compare / Contrast DDE (Dynamic Data Exchange) with OPC.

The following table will clarify the differences between DDE and OPC:

DDE:	OPC
passes 1 value per request	can pass multiple values per request
no time stamp	time stamp standard part of response
no “quality” flag or indication on data	quality indication standard part of response
no underlying structure similar to OLE	based on OLE / COM structure
does not support OLE Automation	supports OLE Automation

NOTE: Since DDE does not support OLE Automation, this means the user cannot take advantage of the OLE Automation interface on standard MS applications like Excel.

What was the OPC Task Force?

The OPC Task Force was a small group of companies committed to rapidly developing and promoting the use of a communication standard based on OLE, to foster greater interoperability between automation software applications. The Task Force completed Version 1 of the OPC specification and transferred ownership of the standard to the OPC Foundation.

The OPC Task Force members were Emerson Process Management Systems Division (Austin, Texas), Intellution (Norwood, Mass.), Intuitive Technology (Marlboro, Mass), Opto 22 (Temecula, Calif.), and Rockwell Software (West Allis, Wisc.), Emerson Process Management, was the master editor of the standard.

Were other companies involved in the review of Version1 draft of OPC?

The OPC Review Process was open to any firm or individual who had the desire to participate. Over 100 firms expressed support for OPC efforts and requested the opportunity to review the Version 1 draft, including:

AEG Schneider	ASA Modicon
Elsag Bailey	Measurex
Cambridge Soft Corporation	Control Technology
Digital	Hewlett Packard
Foxboro	Gensym
Industrial Systems, Inc.	PID
Honeywell	Iconics
Wonderware	Labtech
MCM Enterprise, Ltd.	MAX Control Systems
National Instrument	Kenonic Controls
Oil Systems, Inc.	Optimas Corporation
Prediktor AS	Setpoint
Star Enterprises	Applicom International
Siemens	Square D
Technologies Integration	US Data
Wonderware	Yamatake Honeywell
Zintech Corporation	

What was the schedule for producing a usable standard?

The OPC Task Force released a draft of the standard at the end of 1995 and made prototypes and sample implementation code publicly available in the weeks following. It was not the intent of the draft standard to encompass all requirements but rather to serve as a basis for ongoing input from the user/vendor community. The version 1 draft focused on defining the components required for the real-time transfer of data between an OPC server and an OLE compliant application. An OLE Automation and an OLE custom interface were specified. This allows both Visual Basic and C++ applications to access OPC servers. The final version of Release 1.0 was published in August, 1996.

How did the OPC Task Force succeed in delivering results quickly when so many standards efforts take longer than originally foreseen?

Microsoft and members of the OPC Task Force felt that the best way to quickly develop and introduce a prototype OPC standard was to create a small, focused team from various segments within the industry itself. The OPC Task Force approach:
1) allowed rapid development of a draft standard that addressed key issues, while allowing extensions to be added later as needed
2) ensured that the standard is truly open and supportive of different industry needs and interests
3) allowed for review and comment by all other interested parties once a draft of the proposed standard was created
4) prevented gridlock that is often associated with standards efforts composed of large industry consortia with disparate interests.

Who is on the OPC Foundation?

There are over 170 members of the OPC Foundation (as of March 1999), including the following:

Emerson Process Management	Intellution
Intuitive Technology	Opto 22
Rockwell Software	Wonderware Corp.
ABB Industrial Systems Inc.	Applicom International
Biles & Associates	Canary Labs, Inc.
FactorySoft, Inc.	Hardy Software Systems, Inc.
ICONICS, Inc.	Institut fuer Automation und Kommunikation
Johnson Yokogawa Corp.	National Instruments
OMNX	PID
Process Automation Systems, Inc.	ProMicro Ltd.
RDI Software Technologies, Inc.	S-S Technologies, Inc.
Star Enterprise T	A Engineering Co., Inc.
The Foxboro Company	The Software Studio
Toshiba Corporation	CI Technologies, Inc.

For a complete list of the current members, go to to the OPC Foundation site (www.opcfoundation.org) .

Can any other companies join the OPC efforts?

The OPC Foundation is a not-for-profit corporation chartered to extend and evolve the OPC Specification. Suppliers and end-users are encouraged to join. The first OPC General Assembly meeting was held at the ISA show in October 1996.

What is Microsoft's involvement in the OPC effort?

Microsoft was instrumental in forming the OPC Task Force. Microsoft technical resources worked closely with the OPC Task Force on the OPC standard. Microsoft will also act as an on-going advisor to the OPC Foundation.

Who owns the standard?

The standard is in the public domain and available to anyone who wishes to use it. A copy of the standard is posted on the OPC Foundation web site so that it is readily accessible to the user/vendor community.