



**DYNAMIC POSITIONING CONFERENCE**  
**October 17-18, 2006**

**Design & Control**

---

**Dynamic Positioning Data Using  
OPC Data Exchange Standard**

**David Sherman**

***Sherman & Associates, Inc. - Houston, Texas***

---

<b>DYNAMIC POSITIONING DATA USING OPC DATA EXCHANGE STANDARD.....</b>	<b>0</b>
<b>1 ABSTRACT.....</b>	<b>2</b>
<b>2 HISTORY.....</b>	<b>3</b>
<b>3 OPC FOUNDATION INTRODUCTION.....</b>	<b>3</b>
<b>4 OPC STANDARDS OR SPECIFICATIONS.....</b>	<b>3</b>
4.1 OPC DATA ACCESS.....	4
4.2 OPC ALARMS & EVENTS.....	4
4.3 OPC BATCH.....	4
4.4 OPC DATA EXCHANGE.....	4
4.5 OPC HISTORICAL DATA ACCESS.....	4
4.6 OPC SECURITY.....	4
4.7 OPC XML-DA.....	4
4.8 OPC COMPLEX DATA.....	5
4.9 OPC COMMANDS.....	5
<b>5 OPC FOUNDATION NEW DEVELOPMENTS.....</b>	<b>5</b>
5.1 UNIFIED ARCHITECTURE.....	5
5.2 THE FUTURE OF OPC.....	5
<b>6 HISTORICAL DATA ACCESS OVERVIEW.....</b>	<b>6</b>
<b>7 VENDOR DATA HISTORIANS.....</b>	<b>7</b>
7.1 OSI PI.....	7
7.2 GE FANUC.....	7
7.3 WONDERWARE.....	8
7.4 ROCKWELL.....	8
7.5 OTHER OPC LOGGERS.....	8
<b>8 COMPLIMENTING OPC SERVERS.....</b>	<b>8</b>
8.1 SNMP OPC SERVER.....	9
8.2 REAL TIME NETWORK ANALYSIS.....	9
8.3 POWER SUPPLY MANAGEMENT.....	9
8.4 UNMANAGED DEVICE MONITORING.....	9
8.5 DX OPC BRIDGE.....	9
<b>9 OPC REDUNDANCY.....</b>	<b>10</b>
<b>10 DATA DELIVERY WEB SERVICES.....</b>	<b>11</b>
10.1 WEB SERVICES.....	11
10.2 WEB SERVICES DATA SEQUENCE FIGURE.....	12
10.3 WEB SERVICES CONFIGURATION ACCESS FIGURE.....	12
10.4 OPC XML WEB-SERVICES - OPC DX - CONFIGURATION.....	13
<b>11 INFORMATION PORTAL OVERVIEW.....</b>	<b>13</b>
<b>12 MICROSOFT PORTAL SERVER PLATFORM.....</b>	<b>14</b>
12.1 MICROSOFT OFFICE SHAREPOINT SERVER 2007.....	14
12.2 MICROSOFT OFFICE FORMS SERVER 2007.....	14
12.3 MICROSOFT OFFICE EXCEL SERVER 2007.....	15
12.4 MICROSOFT OFFICE GROOVE SERVER 2007.....	15
12.5 MICROSOFT OFFICE PROJECT SERVER 2007.....	15
12.6 CONCLUSION.....	15

<b>13</b>	<b>DATA STANDARDS.....</b>	<b>15</b>
13.1	DATA STRUCTURE.....	16
13.2	XML AND THE DATA STRUCTURES.....	18
<b>14</b>	<b>OPC LOGGING.....</b>	<b>18</b>
14.1	OPC SE LOGGER GENERAL FEATURES.....	18
14.2	OPC LOGGER PREPARATION.....	19
14.3	OPC SERVER PROPERTIES.....	19
14.4	OPC GROUP PROPERTIES.....	20
14.5	OPC ITEMS PROPERTIES.....	20
14.6	OPC LOGGER USER INTERFACE.....	21
14.7	OPC LOGGER DATABASE LOGGING PROPERTIES.....	22
14.8	OPC LOGGER OPERATIONAL OBJECTS.....	22
14.9	OPC LOGGER OPTIONS: ARRANGING THE DATA.....	22
14.10	OPC LOGGER OPTIONS: SETTING THE LOGGING METHOD.....	22
14.11	OPC LOGGER ERROR HANDLING.....	23
14.12	OPC LOGGER OPTIONS: STARTUP.....	23
14.13	OPC LOGGER DATABASE SIDE NAMING RULES.....	23
14.14	OPC LOGGER GENERAL FAQ'S:.....	24
<b>15</b>	<b>CONCLUSION.....</b>	<b>25</b>
<b>16</b>	<b>DEFINITIONS.....</b>	<b>26</b>
16.1	DCOM.....	26
16.2	COM.....	26
16.3	OLE.....	26
16.4	XML.....	26
16.5	WEB SERVICES.....	26
16.6	SNMP.....	27
16.7	SQL.....	27
<b>17</b>	<b>REFERENCES.....</b>	<b>27</b>

## 1 Abstract

External data logging for dynamic positioning and vessel management systems has historically presented many problems. Many systems provide data in proprietary formats that require custom drivers to parse and store the data into a standard database software package. Some data points require high rates of logging causing problems due to large database storage files. Many data loggers are not “user- friendly” limiting the data extraction process. This problem is not limited to marine systems.

In recent years standards have been formulated to address the process of data exchange. For industrial systems the OPC standard has been developed. This paper will give an overview of the standard and its’ application to dynamic positioning and vessel management systems. Generic system configurations show how the standard is used to provide common data exchange interfaces. Guidelines are given for configuring database groups based on log frequency requirements in relation to storage file sizes. Typical dynamic positioning system data groups are configured for efficient database utilization. Backup software methods are presented for storing database files. Various hardware mass storage devices are compared for optimum performance.

Software tools for data analysis are configured within the generic system. These tools allow for data trending and enhanced analysis. Data transfer and reduction for presentation in common software packages are discussed. Data reduction for real time transmission is presented.

## 2 History

In the past dataloggers for marine systems as well as other industrial systems have normally been dedicated database software applications. In these types of systems data from the control processor is transmitted via a communication link to the logging processor where it is stored in a database. This process uses dedicated software driver programs to transfer the data. These drivers are known as Application Program Interfaces (API). They are unique to the hardware and software application so each type of interface is unique.

Each of the systems may have a dedicated logger with its' own communication link. In this arrangement each of the systems would normally have a proprietary software driver for transferring data. Each of the logging processors would store the data in a SQL database so that queries could be used to extract data for analysis. In some cases these databases are proprietary so standard software applications cannot be used to access the data. This created a situation where some data was not accessible for extraction to external systems.

Microsoft introduced the Windows Dynamic Data Exchange (DDE) interface for exchanging data. This was the first attempt to provide a simple and common interface to exchange data between applications. It had a number of drawbacks. It did not support network or multiple client support. It could not be used to exchange data of a distributed network. It was slow, required a lot of configuration and was not user-friendly.

The next development after DDE was Object Linking and Embedding (OLE). This was an improved method of exchanging data between applications. It was inefficient and lacked sufficient performance.

## 3 OPC Foundation Introduction

OLE for Process Control (OPC) was a major break-through for factory automation and data acquisition systems. It was developed by the OPC Foundation. It provided computer applications a standard interface to hardware and databases. It was originally based on Microsoft's OLE COM (component object model) and DCOM (distributed component object model) technologies, the specification defined a standard set of objects, interfaces and methods to access the underlying data in a hardware device.

Today OPC is a series of standards or specifications. The first, originally called the OPC Specification, is now called the Data Access Specification. There are currently nine standards or specifications completed and/or in development.

Vendors can include an OPC compliant interface in there application that allows other OPC compliant system exchange data. This type of interface allows systems using completely different software applications to exchange data using the common OPC interface.

## 4 OPC Standards or Specifications

The following list is the current set of standards from the OPC Foundation and their description statement for each.

#### **4.1 OPC Data Access**

Used to move real-time data from PLCs, DCSs, and other control devices to HMIs and other display clients. The Data Access 3 specification is now a Release Candidate. It leverages earlier versions while improving the browsing capabilities and incorporating XML-DA Schema.

#### **4.2 OPC Alarms & Events**

Provides alarm and event notifications on demand (in contrast to the continuous data flow of Data Access). These include process alarms, operator actions, informational messages, and tracking/auditing messages.

#### **4.3 OPC Batch**

This specification carries the OPC philosophy to the specialized needs of batch processes. It provides interfaces for the exchange of equipment capabilities (corresponding to the S88.01 Physical Model) and current operating conditions.

#### **4.4 OPC Data eXchange**

This specification takes us from client/server to server-to-server with communication across Ethernet fieldbus networks. This provides multi-vendor interoperability. It also adds remote configuration, diagnostic and monitoring/management services.

#### **4.5 OPC Historical Data Access**

Where OPC Data Access provides access to real-time, continually changing data, OPC Historical Data Access provides access to data already stored. From a simple serial data logging system to a complex SCADA system, historical archives can be retrieved in a uniform manner.

#### **4.6 OPC Security**

All the OPC servers provide information that is valuable to the enterprise and if improperly updated, could have significant consequences to plant processes. OPC Security specifies how to control client access to these servers in order to protect this sensitive information and to guard against unauthorized modification of process parameters.

#### **4.7 OPC XML-DA**

Provides flexible, consistent rules and formats for exposing plant floor data using XML, leveraging the work done by Microsoft and others on SOAP and Web Services.

## 4.8 OPC Complex Data

A companion specification to Data Access and XML-DA that allows servers to expose and describe more complicated data types such as binary structures and XML documents.

## 4.9 OPC Commands

A Working Group has been formed to develop a new set of interfaces that allow OPC clients and servers to identify, send and monitor control commands which execute on a device.

# 5 OPC Foundation New Developments

There have been a number of iterations in specifications. Many additional specifications were created since the original OPC DA specification. The following outlines the newest developments and future outlook.

## 5.1 Unified Architecture

The existing OPC COM based specifications have served the OPC Community well over the past 10 years, but as technology progresses so must the interoperability standards. Here are the factors that influenced the decision to create a new architecture:

Microsoft has deemphasized COM in favor of Web Services and SOA (Service Oriented Architecture)

OPC Vendors want a single set of services to expose the OPC data models (DA, A&E, HDA ...)

Other collaborating organizations need a reliable, efficient way to move higher level structured data.

## 5.2 The Future of OPC

Microsoft software architect Don Box said the company will not invest much more in Component Object Model (COM) and Distributed Component Object Model (DCOM). These are Microsoft's mechanisms for sharing objects between programs. Instead programs will use managed services based on the Extensible Markup Language (XML) to communicate with each other.

OPC-DA is based on DCOM and this foundation is becoming a legacy technology. Considering this, Microsoft will focus on web services. DCOM will be supported as COM & COM+ integration with the capabilities of the Windows Communication Foundation in WinFX or .Net 3.0. Web services. .Net offers many advantages and efficiency for future development and improvement.

With.NET, Web Services, XML, and the OPC Foundation OPC Unified Architecture it may be confusing and difficult to decide on what approach to take regarding OPC related projects. It is this author's opinion to stay with .NET. This new development provides services for building distributed systems all the way from simplistic cross-domain message passing and ORPC to cross-platform, cross-organization, vastly

distributed, service-oriented architectures providing reliable, secure, transactional, scalable and fast, online or offline, synchronous and asynchronous XML messaging.

This is not just a brand new distributed computing platform but similar to the OPC objective, Microsoft has built it on open standards and open technologies. This means that for the first time in decades there won't be artificial politics or induced divisions limiting a distributed computing technology to particular platforms or operating systems like CORBA, DCOM & Java RMI.

The next generation of DCOM interoperates with everyone regardless of the programming language or operating system used. What does this mean for the hundreds of existing OPC Servers?

From a developer standpoint one would use Microsoft's ability to help move existing application to the next level. Microsoft provides the ability to expose existing COM+ services as Windows Communication Foundation (WCF) services without requiring any code change in the COM(+) implementation on the server side. Running a tool and a corresponding 'invisible' WCF service is built for you. This service exposes the COM+ component as a WCF style web service.

It may appear that this paper is focused on historical data and access. It is important to outline some fundamentals and the importance of open standards and open technologies. This philosophy will also be the basis for the remainder of this paper.

## **6 Historical Data Access Overview**

The first step in determining the infrastructure required is to assure standards and interoperability as discussed above, and the data collection requirements.

There are two major avenues of historical data storage. The Process Historians with proprietary databases with specialties in compression, speed, and quantity. The other popular standard is SQL. A Process Historian is normally associated with industrial process data, and a SQL database is assumed to be used for business, or relational data. It is important to note that historians and SQL are not competing standards but complementary ones. In some cases industrial vendors utilize SQL for their process historians, some utilize SQL as a companion to their proprietary process historians, and others utilize a pure proprietary database for their historians.

OPC Historical Data Access (HDA) is used to exchange archived process data between an application and any data-archive in a consistent manner. In other words, OPC HDA client applications that implement trends, reports, or spreadsheets, can retrieve archived process data from process historians or databases in exactly the same manner. OPC HDA enables enterprise-wide interoperability because all applications that rely on historical data can rely on a single industrial standard that is supported by all key vendors. OPC HDA servers exist for most major process historian currently on the market. It is important that the application has an OPC HDA interface available.

The SQL style process datalogger is so widely used for business storage there is a multitude of tools, report writers, and integration components available.

## 7 Vendor Data Historians

It is important to review what is currently available on the market for data collection. Each of the following products would be chosen for some specific reasons ranging from technical requirements to budgetary constraints.

It is an important task to fully develop a document outlining technical requirements before moving to the step of choosing a historian. It is easy to choose a historian, select collect "ALL" data, and color the job complete. It will then be a different job when it comes time for analyzing specific data, maintenance, or integration of the process data to business rules. We will touch on this in more detail later on, for now here is an overview of some popular historians.

### 7.1 OSI PI

PI is high-end three-tiered software system that fully automates the collection, storage, and presentation of manufacturing data. PI provides applications capable of serving real time data to the desktop. There is also an HDA interface for the PI system allowing custom third party components access to its underlying data system.

The PI system is an example of the pure proprietary database style. The Compression method used by PI allows the historian to keep orders of magnitude more data on-line than conventional scanned systems. The compression method is called "swinging door compression." Swinging door compression discards values that fall on a line connecting values that are recorded in the Archive.

### 7.2 GE FANUC

Intellution iHistorian is a plant-wide multi-server data historian that collects, archives and distributes large amounts of real-time, plant floor process information at extremely high speed. iHistorian represents a leap in the performance, utility and affordability of plant-wide data historians. Scalable to 100,000 data points per server, iHistorian delivers high-volume data collection and retrieval, without sacrificing speed or performance.

The iHistorian system is also an example of the pure proprietary database style. The compression method used by PI allows the historian to keep orders of magnitude more data on-line than conventional scanned systems. Intellution is now owned by GE. It does not publish their compression methods.

The CIMPLICITY Database Logger provides a seamless way to analyze system processes and equipment performance by logging data to and reporting data from a wide variety of ODBC (Open Database Connectivity)-compliant databases. The CIMPLICITY system is an example of the SQL database style.

With this database style there are two methods the items can be logged. (Items by rows or Items by Fields) Items by rows will have tagname, timestamp, quality, value, and additional tagname field options. Items by fields will store a group of tags in one record with ID, TimeStamp, and Values in the Tag Fieldname column.

### 7.3 Wonderware

Wonderware IndustrialSQL Server is a high-performance real-time database for factory data. It combines the power and flexibility of a relational database with the speed and compression of a real-time system to integrate the office with the factory floor. IndustrialSQL Server is an extension to Microsoft SQL Server, acquiring plant data at dramatically increased speeds, reducing data storage volumes and integrating plant data with event, summary, production and configuration data.

The IndustrialSQL system is an example of the Proprietary/SQL companion database style.

### 7.4 Rockwell

RSSql™ tool designed to help you better manage manufacturing processes by integrating data from shop floor control systems with enterprise IT and other manufacturing applications. As part of the RSBizWare™ framework, RSSql provides an ideal platform to integrate and build the solutions for the enterprise.

Whether the solution requires a simple data-logging scenario or a bi-directional system in which your plant floor is controlled by the rules and/or the data in an enterprise databases, RSSql's flexible architecture can meet these needs. The RSSql™ system is an example of an advanced SQL database style.

Rockwell also has an advanced historian called RSBizWare™ Historian™. This is a proprietary database type historian. Using FactoryTalk® technologies shared by Rockwell Software products, RSBizWare Historian can connect to any control systems or Human-Machine Interface (HMI) software and collect data at high speed, in real-time, and at full resolution. Built-in web reporting capabilities and analysis tools access the data.

### 7.5 OTHER OPC Loggers

Many companies today use SQL Server, Oracle, or MySQL to store business data. They also might have a requirement to store and use process data from the plant floor, lab systems, etc. Rather than invest in a separate process historian, they can leverage their existing assets by simply adding the ability to directly collect and share process data from their existing database.

There is a variety of OPC data loggers on the market that can enable the above scenario. Some of these products are Matrikon ODBC Process Historian, Canary Labs Trend Historian, Inductive Automation FactorySQL. This author has a basic OPC Logger written in .NET that follows the SQL database model.

These products as with the larger scale products mentioned above have different features and limitations. Again it is important to outlining requirements and researching the features in a variety of products before a purchase.

## 8 Complimenting OPC Servers

The Data Access specification defines how both the client and the server application interface must be constructed. If the specification is followed properly, a client vendor knows that any OPC server that exists for an industrial device can provide the connectivity needed for data access. This means that not only industrial devices are available but commercial type devices can now utilize a standard to access

their tools to the industrial arena. Listed below are a few common elements of a system that have OPC interfaces.

### **8.1 SNMP OPC Server**

IndustrialSNMP (iSNMP) is an OPC 2.0 Server that combines a network management with analysis software developed specifically for the industrial controls marketplace. iSNMP Server seamlessly integrates monitoring and analyzing of managed and unmanaged Ethernet networks. This Server provides a real time view of Ethernet network devices, the overall Ethernet network traffic volume and overall Ethernet network status.

Application

### **8.2 Real Time Network Analysis**

Raw SNMP network data can be difficult to interpret. For the automobile user, a speedometer can convert the raw number of revolutions of the tires of a car into a more meaningful statistic: speed. An SNMP Network Analyst/Server provides a similar service to the industrial Ethernet network user. An SNMP Analyst/Server gathers raw SNMP data from the network devices and continually performs calculations to generate data that can be used, such as bandwidth utilization and network error rate statistics. Additionally, an SNMP Network Analyst/Server can offer the user a simple green light/ yellow light/ red light status tag for network devices. The HMI user can then easily summarize the health of their network devices with a simple inclusion of the device status tag.

### **8.3 Power Supply Management**

Uninterruptible power supplies have become necessary to ensure maximum uptime and protect powered network equipment from power surges. Most UPS's have embedded microprocessors and are capable of reporting their status and operating mode information via SNMP. Now when a battery needs replacement, or a UPS trips, you can have immediate notification in your HMI.

### **8.4 Unmanaged Device Monitoring**

Not all Ethernet network devices are SNMP managed. To help monitor any system from within your HMI, an SNMP Server auto-discovers both managed and unmanaged network devices. An SNMP Server can automatically generate OPC tags for each device that represent "heartbeat" and response time.

### **8.5 DX OPC Bridge**

It is important to elaborate on the OPCDX specification after introducing network monitoring servers. OPC Data Exchange (DX) was designed to address the needs of Device-to-Device and Bus-to-Bus interoperability by applying the same technologies currently used by desktop applications at the device level. By adding OPC DA Client support to an existing OPC DA server the result is a DX server that not only provides access to its data, but has the ability to gather data from other OPC DA enabled devices.

Historically there have been a number of high-level and low-level bus protocols, each one written to address the differing needs and uses of the devices they were intended to serve. Protocols such as PROFIBUS, DeviceNet, ControlNet, and many more, have been utilized in the past.

Recently there has been a growing movement towards the use of Ethernet as an industrial communication backbone. Initially each of these respective bus specifications was redeveloped for use over Ethernet. The

resulting PROFINet, Ethernet/IP, High-Speed Ethernet Fieldbus (HSE), and BACNET/IP provided a means to move smoothly from proprietary wiring schemes to Ethernet. However it did not address the key issue of true Device-to-Device interoperability or more importantly Bus-to-Bus interoperability. While these new Ethernet based protocols could coexist on the same wire, there was still no way to get data from a PROFINet device directly into an Ethernet/IP device.

OPC Data Exchange (DX) was designed to address the needs of Device-to-Device and Bus-to-Bus interoperability by applying the same technologies currently used by desktop applications at the device level. By adding OPC DA Client support to an existing OPC DA server we then have a DX server that not only provides access to its data, but has the ability to gather data from other OPC DA enabled devices. Once the DX Server gathers its data, it can share any portion of its dataset among devices, thus bridging the horizontal data flow between devices and achieving Server-to-Server communications. If the same method is applied to a control bus such as BACNET/IP, all of the I/O points on the bus could be available as fully accessible OPC items. An OPC DX server acting as a gateway to Ethernet/IP or other legacy protocols could access the BACNET/IP data even though it had no knowledge of BACNET protocol. OPC DX is the Machine-2-Machine interface for OPC.

## 9 OPC Redundancy

After connecting devices, bridging networks, monitoring network quality and logging data, another important issue needs to be considered, redundancy. OPC Redundancy is intended for OPC client applications that must use redundant hardware and/or software to achieve the highest degree of communication reliability. A Redundancy Module should provide the following features.

Transparent integration with any OPC compliant client/server configuration.

Automatic fail-over from a primary OPC server to a standby OPC server and, if desired, back to the primary server once it becomes available.

Hot, Cold, and Warm fail-over methods.

Watchdog tags that test for failure in multiple ways.

Ability for automatic failure notification (via log file, email, etc).

Extensive error tracking and diagnostic capabilities.

A typical network system will completely fail due to a single point of failure. While modern computers advertise reliability that exceeds 99.9%, this still translates to over three days of down time per year. If your operation can't afford this downtime, an alternative is to implement redundancy.

The general procedure for implementing redundant systems is to use "Primary" and "Standby" OPC servers. Such a system is illustrated in Figure 2. However, to properly use the redundancy architecture, the top-level application must:

- Be aware that a standby OPC server exists.
- Avoid using the standby OPC server until it is necessary to do so.
- Be able to detect when the Primary OPC server is no longer operational, and seamlessly fail-over to the standby OPC server.

## 10 Data Delivery Web Services

Real-time and historical data access is becoming an ever increasingly important aspect in today's business market. The ability to actively view current data shifts the focus of an operation from responding to alarms to a pro-active approach to operations.

Web services allow for seamless integration of real-time data and historical display tools into a common internet browser environment. This allows a user the access to data without downloading and installing additional software or risking security violations.

### *10.1 Web Services*

A Web Service is programmable application logically accessible using standard Internet protocols. Web Services combine the best aspects of component-based development and the Web. Like components, Web Services represent functionality that can be easily reused without knowing how the service is implemented. Unlike current component technologies which are accessed via proprietary protocols, Web Services are accessed via ubiquitous Web protocols (ex: HTTP) using universally-accepted data formats (ex: XML).

In practical business terms, Web Services has emerged as a powerful mechanism for integrating disparate IT systems and assets. They work using widely accepted technologies and are governed by commonly adopted standards. Web Services can be adopted incrementally with little risk and at low cost. Today's enterprises use Web Services for point-to-point application integration, to reuse existing IT assets and to securely connect business partners or customers. Independent Software Vendors (ISVs) embed Web Services functionality in their software products so they are easier to deploy.

Web Services are applicable to any type of Web environment: Internet, intranet, or extranet. Web Services can support business-to-consumer, business-to-business, department-to-department, or peer-to-peer interactions. A Web Service consumer can be a human user accessing the service through a desktop or wireless browser; it can be an application program; or it can be another Web Service. Web Services support existing security frameworks.

Basically a simple example would be creating an avenue to securely view or collect real-time rig data over the intranet through a web browser with no software to load on the Client PCs, data updates without page refreshes and the capability to Read and Write Data.

## 10.2 Web Services Data Sequence Figure

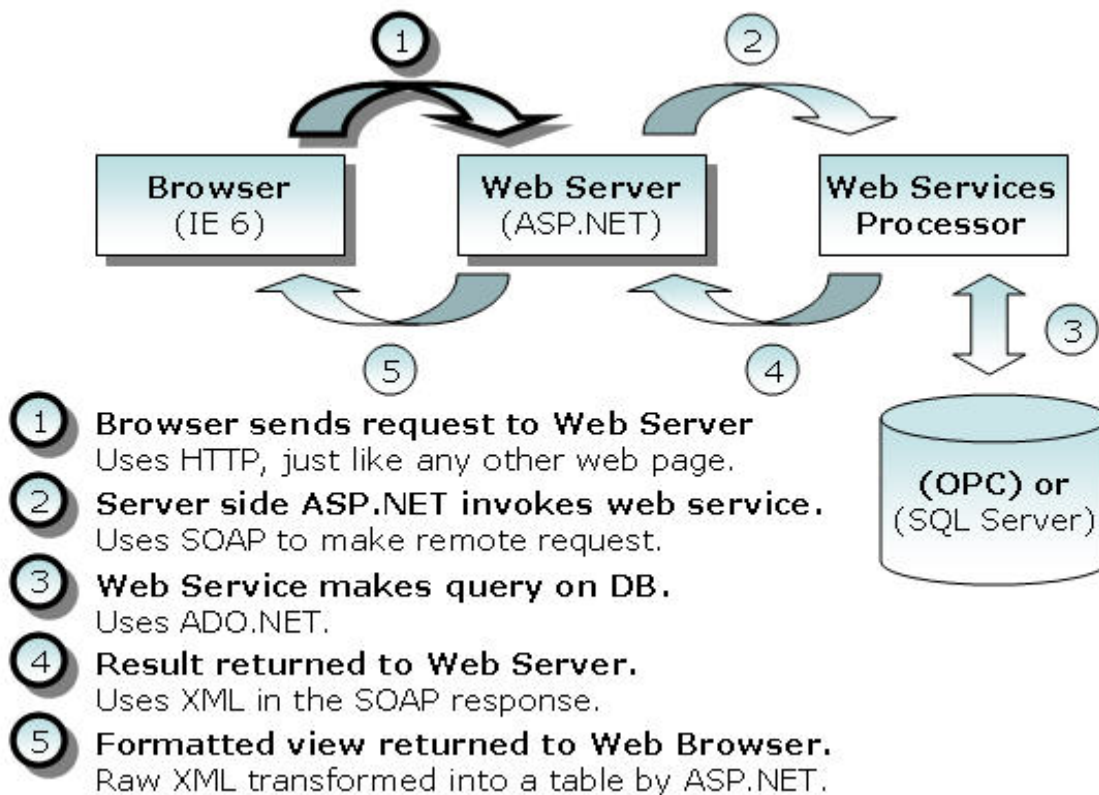
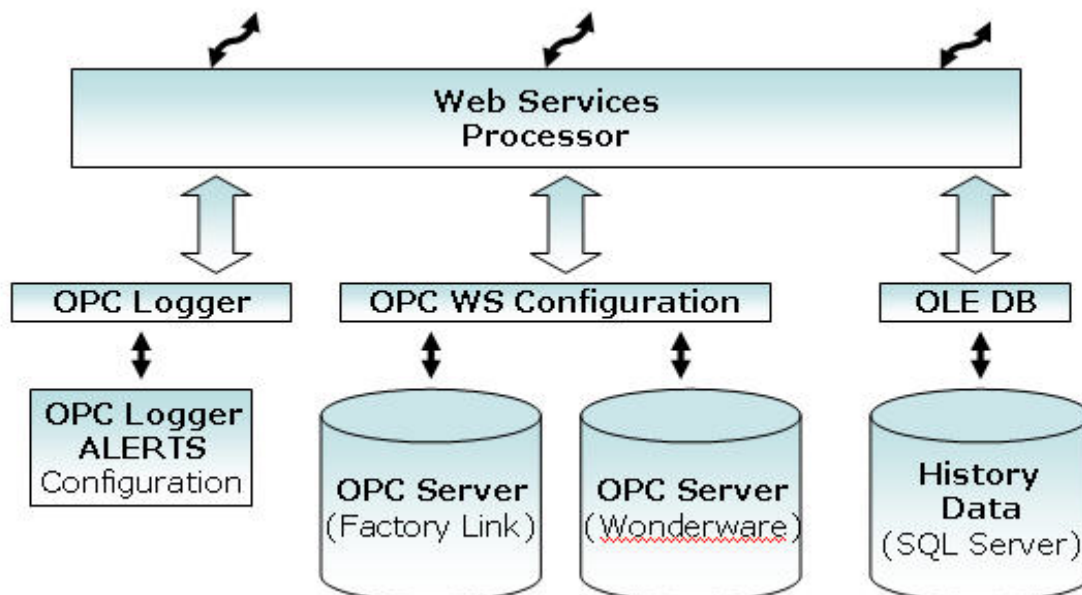


Figure 1: Web Service Data Sequence

## 10.3 Web Services Configuration Access Figure



## 10.4 OPC XML Web-services - OPC DX - Configuration

AN OPCDX web service should support flexible configuration and access to multiple OPC/XML servers. XML-DA gateway products that support a static configuration but cannot be changed at run-time should be avoided. Gateway solutions offer different configuration concepts.

Important gateway characteristics are:

- Support access to multiple
  - OPC DA servers
  - XML DA servers
- Customization module for local device access.
- Dynamic configuration. Multiple clients can modify the configuration during operation.
- Each item connection is configured individually
- Connections support override values and substitute values

## 11 Information Portal Overview

Since communication, collaboration and documentation are such important aspects of any project, it's important that there are good processes in place to facilitate these activities. After deciding on OPC for a standard data access component for the equipment then one must consider network monitoring and network bridge components. After reviewing a selection of data historians and loggers then outlined the avenues available to make the data available, Web Services, the question remains, how do we tie all these components together?

Generally in a corporate environment there are single points of access available to provide tools such as email, documents, reports, etc... the same philosophy for designs outlined above would apply. Asking someone what the ideal scenario would be to access data for the enterprise, comments similar to the following are common,

*"We want to query against multiple system, and vessels to analyze operational and maintenance discrepancies and abnormalities. We also want automated data streams to facilitate various reporting across vessels, systems, department, and employees. We want to control access to which data and applications are accessible to the users e.g. partners, employees, customers. In addition, we want it to be easy to set up, configure and use."*

In other words:

- A single point of access to all resources associated with a designated domain
- Federated access to all data sets and repositories, aggregated and categorized
- Collaboration technologies that bring people together
- Personalized secure interaction with the portal services
- Integration with Applications and workflow systems

Portal applications serve the needs of employees, business partners and customers. Users can sign-on to the portal and receive personalized pages providing access to the information, people, and applications they need. This personalized single point of access to all necessary resources reduces information overload, accelerates productivity, and increases operational awareness.

Keeping with standardization and the infrastructure OPC is based on, Microsoft's Office Sharepoint Server 2007 is another product that compliments the former. Other products in this area are available from vendors such as GE , OSI Soft, and Rockwell software however the underlying implementations, benefits, and infrastructure used would warrant its own paper.

Microsoft has decided to focus its efforts around a revamped SharePoint Portal server for its next generation Office product. Microsoft SharePoint Portal Server 2007 (MOSS) will act as a back-end for new client-based Office services. It also will incorporate a variety of workflow engines, designed to mesh with Windows Workflow Foundation, the next-generation Windows workflow technology that Microsoft is baking into Windows Vista, Longhorn Server and .NET 3.0.

## **12 Microsoft Portal Server Platform**

Included in MOSS 2007 will be the Office Forms Server 2007, an electronic-forms offering designed to complement InfoPath; Office Groove Server 2007, a collaboration server; and Office Project Portfolio Server 2007, a complement to Microsoft Project Server. But the core technology is Office SharePoint Portal Server 2007. This offering will combine Microsoft's current Content Management Server, SharePoint Portal Server and an Excel Server into a single product.

### **12.1 Microsoft Office SharePoint Server 2007**

Microsoft Office SharePoint Server 2007 allows sharing and collaborative editing of Office 2007 documents. It allows central storage of documents and management of Office documents, throughout the enterprise. These documents can be accessed either by the applications which created them, Microsoft Office Outlook 2007 or a web browser. Documents can also be managed through pre-defined policies that let users create and publish shared content, through a SharePoint site.

SharePoint server allows searching of all Office documents that it manages, centrally, thereby making data more accessible. It also provides access control for documents. Specialized server components can plug into the SharePoint server to extend the functionality of the server, such as Excel Services exposing data analysis services for Excel services. Data from other data sources can also be merged with Office data.

SharePoint also lets users personalize the SharePoint site or filter content of interest. SharePoint documents can also be locally cached by clients for offline editing; the changes are later merged.

### **12.2 Microsoft Office Forms Server 2007**

New to Office 2007, this server helps enterprises centralizes control and management of electronic forms via a common Web interface. This server solution is a scalable, security-enhanced, standards-based data-gathering that can be used to enhance business processes. It extends the reach of data and enables work flow to be accessed and filled out using any browser, including mobile phone browsers. Forms Server 2007 also supports using a database or other data sources as the back-end for the form such as OPC data sources. Additionally, it allows centralized deployment and management of forms.

### **12.3 Microsoft Office Excel server 2007**

New to Office 2007, this server helps enterprises centralize control and management of electronic forms via a common Web interface. The server will provide performance, scalability, and security capabilities that aren't possible in client-side solutions, and make it easier to tie Excel spreadsheets into back-end data sources. The Excel client will be able to access these server-side spreadsheets via Web services, or retrieve entire XML-based workbooks. You will also be able to perform most Excel functions on server-side spreadsheets via a Web browser.

### **12.4 Microsoft Office Groove Server 2007**

New to Office 2007, this is a collaboration platform that enables teams to work together dynamically inside collaborative workspaces. Teams often lack context because not everyone is in the same place and connected to the same data and systems. Groove workspaces add context by putting all people, tools, and data in one place and making them accessible to all team members' right on their own computers. Groove automatically "pushes" just the changes you make in a workspace to your team members' computers in the most efficient manner. No more attaching, detaching, uploading, or downloading.

### **12.5 Microsoft Office Project Server 2007**

Microsoft Office Project Server 2007 allows one to centrally manage and coordinate projects. It allows budget and resource tracking, and activity plan management. The project data and reports can also be further analyzed using Cube Building Service and of course all the project management data can be accessed from a web browser.

### **12.6 Conclusion**

This may appear to be more complex than required or needed in this document however, this system allows all five of the bullet points above. You would use its content management feature so it can be changed anytime over the network with Sharepoint. An OPCDataViewer webpart could be utilized by simply drag a webpart over to the web page and configuring with domain, devicename, group, and item list. The webpart would then display the data to the user as graphics, summary, spreadsheet, or real-time display. For SQL Server type Historians there is a multitude of similar tools available to acquire the historical data.

## **13 Data Standards**

Standards for the lower level communication objects, the data logging objects, and the methods available to retrieve data have been presented. This is the core requirements for data retrieval, storage, and access. Moving back to the basics, this paper represents a focus on DP and VMS systems and as such a standard or definition should be outlined specifically for the data in these systems. In short the standards discussed above give significant advantages to the developer from a technology point of view but not from a user's point of view. The same problems exist concerning retrieving data, even with the infrastructure in place, if there is not a standard data structure available to show the users what data to select for particular needs.

A good example of such a standard is WITSML, The Wellsite Information Transfer Standard Markup Language. This is a standard for sending well site information in an XML document format between business partners. XML schemas are used to define the content of an XML document.

Extensible Markup Language (XML) is a simple, very flexible text format. Originally designed to meet the challenges of large-scale electronic publishing, XML is also playing an increasingly important role in the exchange of a wide variety of data on the Web and elsewhere. XML is a standard and was created to structure, store and to send information.

Structured information contains both content (words, pictures, etc.) and some indication of what role that content plays (for example, content in a section heading has a different meaning from content in a footnote, which means something different than content in a figure caption or content in a database table, etc.). Almost all documents have some structure.

### 13.1 Data Structure

As documents have structure so does data. An easy example of a data structure is an OPC data structure. This is the structure that an OPC Client would use to retrieve data from an OPC Server. The most important thing to learn from this example is groups. (See Figure 1)

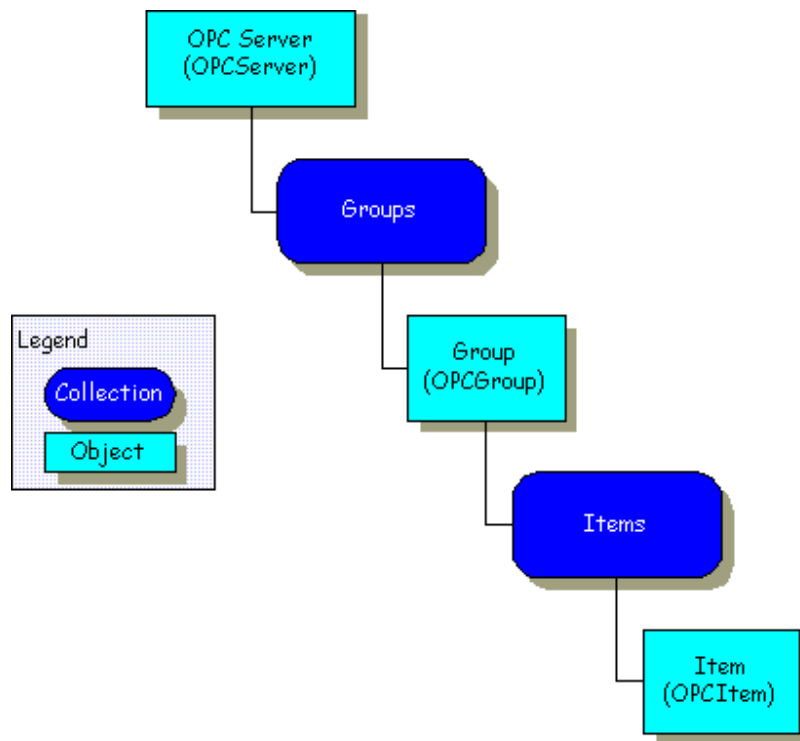


Figure 1: OPC DA Structure

The OPC object structure contains an OPC Server with a collection of OPC Groups that in turn contain collections of OPC Items (Tags). This structure is used to implement a client/server model where the OPC Server provides a standard interface to the OPC objects (OPCGroups and OPCItems), allowing client applications to control devices and manage device data in a standardized manner.

Considering preceding the outlined structure, it is helpful to reference the WITSML standard and the Petrotechnical Open Standards Consortium, Inc (POSC). POSC has defined a set of groups that will hold datasets considered to be standard information required for a well site. (See Figure 2) Each group also holds a standard set of data points as well as the ability to add custom data points. Keep in mind that a data point can also be a data type array which holds additional data points.

<b>Group Schemas</b>	<b>Data Schemas</b>
grp_bhaRun.xsd	obj_cementJob.xsd
grp_cementJob.xsd	grp_cementJob.xsd
grp_convCore.xsd	cs_wbGeometry.xsd
grp_dtsInstalledSystem.xsd	grp_wbGeometry.xsd
grp_dtsMeasurement.xsd	cs_wbGeometrySection.xsd
grp_fluidsReport.xsd	cs_cementStage.xsd
grp_formationMarker.xsd	cs_cementingFluid.xsd
grp_log.xsd	cs_cementPumpSchedule.xsd
grp_message.xsd	cs_cementAdditive.xsd
grp_mudLog.xsd	cs_cementTest.xsd
grp_opsReport.xsd	cs_commonData.xsd
grp_rig.xsd	cs_customData.xsd
grp_risk.xsd	
grp_sidewallCore.xsd	
grp_surveyProgram.xsd	
grp_target.xsd	
grp_trajectory.xsd	
grp_tubular.xsd	
grp_wbGeometry.xsd	
grp_wellbore.xsd	
grp_well.xsd	
grp_wellLog.xsd	
cs_commonData.xsd	
cs_customData.xsd	
attgrp_uid.xsd	

Figure 2: POSC Well site Group Structure

## 13.2 XML and the Data Structures

XML-based capabilities have been one of the main selling points for Office 2003. By saving documents as XML files, the new Office will allow back-end computing systems such as corporate databases to retrieve and reuse data from documents. XML support also allows Office to become a client for viewing and manipulating data from Web services. Web Services were briefly covered in several earlier sections of this paper. In the real time information portal section, Sharepoint is simply a container that implements many web services to create a website. Ideally some of the data historians listed earlier utilize this method to display information over the internet but further research is required to ensure proper implementations in the product to ensure continuity of standards across an enterprise. The conveyance here is showing the integration avenues for both configuration and retrieval if the data required is structured as well.

## 14 OPC Logging

To better understand the formulation of the data structures outlined above, it is necessary to understand the basic configuration of a datalogger. To simplify this discussion I have chose to outline the configuration of the simple OPC SQL Datalogger Sherman & Associates developed. This was developed to include all the core requirements. These core requirements are the underlying basis for almost all historians. The SE OPC Logger Client supports both local and remote OPC server connections. Remote connections are handled through the operating system's DCOM interface.

### *14.1 OPC SE Logger General Features*

#### **OPC support**

Collect data from any OPC Data Access 1.0 or 2.0 compliant Server

Connect from 1 to N OPC servers

Works with local or remote OPC Servers - DCOM timeout feature helps with remote connections

#### **Flexible**

Flexible formatting of data storage tables/records

Flexible Logging & Rollover Options

Trigger logging start/stop based on system event

Log all values or only log on change of value

User configurable logging rate on each group

Import Tags from OPC servers for fast configuration - saves engineering time

#### **Powerful**

32-bit multi-threaded .NET architecture

Diagnostic log for start/stop and errors

## 14.2 OPC Logger Preparation

Begin by creating a server connection. A server connection connects to an OPC server. Once this connection is made, a group can be added to the connection with varying properties such as update rate, dead band, and storage rate. Finally, items can be added to the individual groups. These items have properties such as an initial active state, data type and access path. Active items will automatically receive data change notifications and will display its current value and quality in the item view.

## 14.3 OPC Server Properties

A server connection provides a link between an OPC server and this client. Through this connection, groups will be added.

Select New Server... from the Edit menu or click the toolbar button to create a new server connection using the following property sheet:

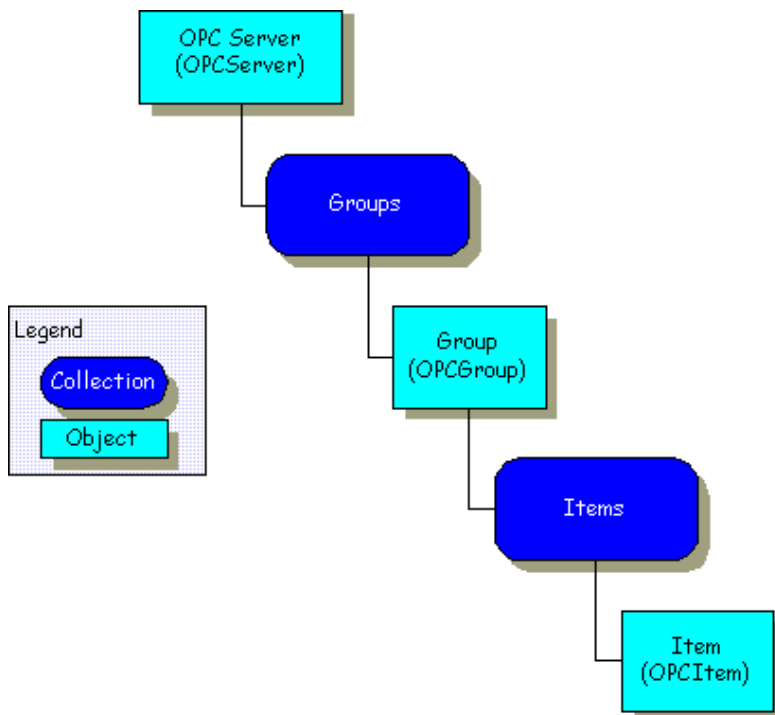
Specify the Prog ID of the OPC Server the client should connect to. This allows browsing for registered servers by expanding any of the following branches:

OPC General - All Servers should add a special OPC key in the system registry to be identified as an OPC server. This method allows for clients and servers to be used (the preferred method is to use component categories). Servers that are registered with this key will be listed.

Double-clicking on any registered server will automatically add the Server.

Remote Machine Name

Specify the name of the machine the OPC server (specified by the Prog ID) resides on. If the server is located on the same machine as the client, leave this field blank.



## ***14.4 OPC Group Properties***

### **Name**

Enter a name to identify the group. If no name is entered, it is the responsibility of the OPC Server to generate a unique name for you.

### **Update Rate**

Specifies how often, in milliseconds, the OPC server should provide updates for changing data.

### **Time Bias**

Specifies the time difference in minutes between the client/server and the actual device. The information allows the client to convert the timestamp received by the server back to the local time of the device.

### **Percent Dead band**

Specifies the percent change in data required to notify the client of a data change.

### **Language ID**

Specifies the language to be used by the server when returning values as text for operations on this group.

### **Active State**

Specifies the active state of the group. When the group is active, the client will receive data change updates for all active items at the specified update rate.

### **Update Notification**

Specifies the method to use when receiving data change notifications, or initiating asynchronous requests. Valid selections are OPC 1.0 (No Timestamp), OPC 1.0 (With Timestamp) and OPC 2.0. Data Access 2.0 servers automatically supply timestamps with all data transactions.

When the group has been successfully added to the project, invoke the group properties by selecting Properties... from the Edit menu to modify group properties and view the supported group interfaces.

## ***14.5 OPC Items Properties***

Items represent data that may be accessed via an OPC server. An item specifies the following properties: item ID, access path, requested data type, and active state.

Select New Item... from the Edit menu or click on the toolbar button to define an item using the item editor dialog box:

### **Access Path**

Required by some OPC servers to complete an item definition. Refer to your OPC server's documentation to determine whether you need to specify an access path.

**Item ID**

The OPC server item used to reference the data. Refer to the OPC server's documentation to determine valid item names for the location in question. If the server supports tag browsing then you can select an item by using the browse controls.

**Data Type**

The requested data type should be used when communicating between the OPC server and the OPC Quick Client. This should be specified to agree with the size and type of the register or memory location that is being addressed. Supported data types are defined as follows:

Native	Default as determined by the OPC server
Boolean	Single bit
Char	Signed 8 bit value
Byte	Unsigned 8 bit value
Short	Signed 16 bit value
Word	Unsigned 16 bit value
Long	Signed 32 bit value
DWord	Unsigned 32 bit value
Real	Single precision floating point value. (32 bits)
Double	Double precision floating point value (64 bits)
String	Zero terminated character array.

Check the Array box if the data type is an array type.

**Active State**

Specifies the initial active state for the item when adding the item. The state can be changed later as described later on in this topic.

**Browsing**

If the server supports browsing, the browsing controls will be available for use. Browse the branch list to search for selected items. Apply appropriate filtering to simplify the search.

***14.6 OPC Logger User Interface***

Visual/Operational Changes:

When the OPC Quick Client adds, activates or inactivates an item there may be a change in the appearance of the item. The image that appears to the left of each item will change to one of the following types:

The item is active.

The item is inactive.

The client could not add the specified item to the group.

## ***14.7 OPC Logger Database Logging Properties***

For each Group select a data source for the items to be logged. Select the method the items will be logged. (Items by rows or Items by Fields) Items by rows will have tagname, timestamp, quality, and value fields. Items by fields will store all tags in one record with ID, TimeStamp, and Values in the Tag Fieldnames.

For each group allow the database and tables to be created automatically from the OPC, Group, and Tags configured in the client.

For each Group allow logging to be enabled or disabled.

## ***14.8 OPC Logger Operational Objects***

Logger Service

The actual object that creates the database and logs the data

Logger Administrator

The graphical user interface that allows configuration of the data to be logged

## ***14.9 OPC Logger Options: Arranging the Data***

The OPC Data Logger provides two different formats for storing the data: Table and Detail format. Your decision about which to use is a function of the nature of your data and more importantly what you plan to do with the data after you log it. You should consider your ultimate use for the data and store it in a format that supports your reporting or analysis objectives.

- Table Format - one record per time sampling period with the fields being all the sampled OPC items
- Detail Format - one record per OPC item with a variety of data stored for each item

Using Detail Format will optionally allow additional data storage about each item since each sample is one record in the database.

When choosing a format, it has an effect on log data. The OPC Data Logger on the Log Method tab indicates which log methods work with which formats and helps you choose valid combinations.

## ***14.10 OPC Logger Options: Setting the Logging Method***

Setting Logging Method is very closely related to the decision about the logging format since the two decisions drive each other. There are two major Logging Methods:

- Exception Based - only logs data when a value changes. You have two choices:
  - Logging individual tags as they change - requires use of the Detail Mode of storage
  - Logging all tags when any one changes - requires use of the Table Mode of storage
- Timer Based - logs all tags based on a time interval that you set. The time interval can log tags two different ways:

- Log all tags when the time interval has passed - requires use of the Detail Mode of storage
- Log individual tags on a timer - requires use of the Table Mode of storage

For both methods, optionally tags can be logged whenever they change and on the timer regardless if values change or not.

Start or Stop on a trigger that is tied to a discrete or analog process tag - Logging can be set to begin when a user specified OPC item changes from True to False or False to True. Logging can also be set to begin with a user specified non-discrete tag meets a specified value or other conditions as shown in the screens below. Separate tags may be used for starting and stopping.

### ***14.11 OPC Logger Error Handling***

The OPC Data Logger provides several options for handling errors that may occur in the logging process. These options are shown below

Logging has Started  
 Logging has Stopped  
 OPC Server is not available  
 OPC Tag can not be subscribed  
 Severe Error, Logging Stopped

### ***14.12 OPC Logger Options: Startup***

Once you have your OPC Data Logger solution setup then options for starting can be selected. Since the OPC Logger is a Service there are several options.

- Automatically start service on boot up.
- Start Service manually

Both of these methods will start logging the last loaded configuration file.

### ***14.13 OPC Logger Database Side Naming Rules***

Rules for Regular Identifiers for SQL Server

=====

The rules for the format of regular identifiers are dependent on the database compatibility level, which can be set with sp\_dbcmptlevel. For more information, see sp\_dbcmptlevel.

When the compatibility level is 80, the rules are:

The first character must be one of the following:

A letter as defined by the Unicode Standard 2.0. The Unicode definition of letters includes Latin characters from a through z and from A through Z, in addition to letter characters from other languages. The underscore (\_), "at" sign (@), or number sign (#).

Certain symbols at the beginning of an identifier have special meaning in SQL Server. An identifier beginning with the "at" sign denotes a local variable or parameter. An identifier beginning with a number sign denotes a temporary table or procedure. An identifier beginning with double number signs (##) denotes a global temporary object.

Some Transact-SQL functions have names that start with double at signs (@@). To avoid confusion with these functions, it is recommended that you do not use names that start with @@.

Subsequent characters can be:

Letters as defined in the Unicode Standard 2.0.

Decimal numbers from either Basic Latin or other national scripts.

The "at" sign, dollar sign (\$), number sign, or underscore.

The identifier must not be a Transact-SQL reserved word. SQL Server reserves both the uppercase and lowercase versions of reserved words.

Embedded spaces or special characters are not allowed.

General rules:

1. Any space or dot(.) will be replaced with underscore
2. Any special character will be removed

Server name will be the database name so the ProgID type such as "OCSTK.DA.Sim.32" will be created as "OCSTK\_DA\_Sim\_32"

Remote Server "\\RemoteServer\OCSTK.DA.Sim.32" will be created as "OCSTK\_DA\_Sim\_32@RemoteServer"

Group name will be restricted to valid character when entered from UI

Any non-integer numeric compatible types will be converted to System.Double (sql float)

Any integer-compatible types will be converted to System.Int64 (SQL bigint) VT\_BOOL to

System.Boolean (database bit)

Any String type will be (SQL varchar(255))

Any array types will be filtered out for selection automatically (not supported at this time)

#### ***14.14 OPC Logger General FAQ's:***

Can the same server be added more than once? "NO"

Can the same item be added more than once in same group? "NO"

Can the same item be added more than once in different group? "YES"

Currently a server will be added to the tree view only if it's available (can be connected) and the current setting is saved into a configuration file. How is the situation handled if that server is not available?

"Displays Error in error log and disables logging. It will continue to try to connect to the server on interval. If the server becomes available it enables logging."

What if the connected server suddenly becomes unavailable? It Logs the error information into event log. It will continue to try to connect to the server on interval. If the server becomes available it enables logging." If the AlertMe Module is installed you may configure it to Email an alert to an operator."

Can I add another two items in this group when the logging is in progress? "NO, that group is not allowed to change if the logging is in progress, you have to stop the logging for that group (there is different icon for the group to show its state) to add/remove items."

## **15 Conclusion**

This paper has outlined all the core components necessary to implement an extensive real-time and historical data collection and access scenario. If each section of this paper is utilized then a powerful, user friendly, scalable remote historical data access and reporting system can be achieved.

## 16 Definitions

The following acronym definitions were derived from webopedia.com, an online dictionary and search engine for computer and Internet technology definitions.

### 16.1 DCOM

Abbreviation of *Distributed Component Object Model*, an extension of the Component Object Model (COM) that allows COM components to communicate across network boundaries. Traditional COM components can only perform interprocess communication across process boundaries on the same machine. DCOM uses the RPC mechanism to transparently send and receive information between COM components (i.e., clients and servers) on the same network. DCOM was first made available in 1995 with the initial release of Windows NT 4.

### 16.2 COM

Abbreviation of *Component Object Model* A software architecture developed by Microsoft to build component-based applications. COM objects are discrete components, each with a unique identity, which expose interfaces that allow applications and other components to access their features. COM objects are more versatile than Win32 DLLs because they are completely language-independent, have built-in interprocess communications capability, and easily fit into an object-oriented program design. COM was first released in 1993 with OLE2, largely to replace the interprocess communication mechanism DDE used by the initial release of OLE. ActiveX also is based on COM.

### 16.3 OLE

Abbreviation of *Object Linking and Embedding*, OLE is a compound document *standard* developed by Microsoft Corporation. It enables you to create objects with one application and then link or embed them in a second application. Embedded objects retain their original format and links to the application that created them.

### 16.4 XML

Short for Extensible Markup Language, a specification developed by the W3C. XML is a pared-down version of SGML, designed especially for Web documents. It allows designers to create their own customized tags, enabling the definition, transmission, validation, and interpretation of data between applications and between organizations.

### 16.5 Web Services

The term Web services describes a standardized way of integrating Web-based applications using the XML, SOAP, WSDL and UDDI open standards over an Internet protocol backbone. XML is used to tag the data, SOAP is used to transfer the data, WSDL is used for describing the services available and UDDI is used for listing what services are available. Used primarily as a means for businesses to communicate

with each other and with clients, Web services allow organizations to communicate data without intimate knowledge of each other's IT systems behind the firewall.

## 16.6 SNMP

Short for Simple Network Management Protocol, a set of protocols for managing complex networks. The first versions of SNMP were developed in the early 80s. SNMP works by sending messages, called protocol data units (PDUs), to different parts of a network. SNMP-compliant devices, called agents, store data about themselves in Management Information Bases (MIBs) and return this data to the SNMP requesters.

## 16.7 SQL

Abbreviation of structured query language, and pronounced either see-kwell or as separate letters. SQL is a standardized query language for requesting information from a database. The original version called SEQUEL (structured English query language) was designed by an IBM research center in 1974 and 1975. SQL was first introduced as a commercial database system in 1979 by Oracle Corporation.

# 17 References

Microsoft Corporation

<http://www.microsoft.com/>

OPC Foundation

<http://www.opcfoundation.org/>