Oracle Plant Automation Interface
White Paper

Multi-Vendor Industry Standards Based Open Communications technology protecting long term investment
Abstract

In the past, communication with plant automation required custom interfaces using drivers proprietary to the control vendor. This approach locked the customer's investment into a single vendor's proprietary technology using unique interface solutions with no hope of software reuse when interfacing to other devices. Today, market pressures are driving the adoption of open standards. This white paper discusses a specific application developed for a specialty gas supplier using JAVA technology to connect an Oracle database with the plant automation system using such open standards, specifically, the OPC industry standards for access to Process Control. This solution provides a model based on open technologies and standards that may be applied in any setting where connectivity between the production and management networks is desired.

Problem Definition

Custom vendor interfaces, while different for each vendor, attempted to solve the same problem. The general needs included a way to read and write data to and from the control system in real time (transfer rates meeting the dynamics of the process under control) reliably. The need for a generic method to exchange information in a well defined and open way is driving the control industry. No one can afford to write monitoring applications for each specific device, there are too many different devices and the unit volumes are not large
OLE for Process Control

OPC (OLE for Process Control) is an industry standard collaboratively developed by worldwide leading automation hardware / software vendors working in cooperation with Microsoft (www.opcfoundation.org).

The history of OPC for Process control has its origin in 1992 and began with a group calling itself WinSEM (Windows in Science, Engineering and Manufacturing). This group was composed of members from industrial control and data acquisition companies, with Microsoft acting as a catalyst. Like many other OLE (Object Linking and Embedding) based standards for other industries such as insurance companies and the medical industry, OPC is based on input from concerned parties. OPC for Process Control was conceived as a way to bridge the differences in data acquisition between proprietary devices, in this case, process control sensors and machinery, by specifying a standard interface that would easily allow a compliant client to access the sensor or machinery data through the device’s OPC server, regardless of its manufacturer.

The OPC task force began by public announcement in 1995 at the ISA (Instrument Society of America) show in New Orleans.

The founding companies were:

- Fisher-Rosemount
- Intellution
- Intuitive Technology
- OPTO 22
- Rockwell Software
- Microsoft (in a supportive and consultative role)

The OPC foundation launched OPC server demonstrations at the 1996 ISA show in Chicago. Various companies demonstrated servers running at the Microsoft booth. Several collaborative demonstrations followed at major trade shows around the world.

The foundation has broad industry support with over 220 member companies. This includes nearly all of the world’s major providers of control systems, instrumentation, and process control systems. Compliant software applications display the following logo from the OPC foundation:

[OPC Foundation Logo]

1 www.opc.dial.pipex.com
Brief Java History

The Java programming language was developed by Sun Microsystems as a result of a research project to develop advanced software for network devices and embedded systems. As their research reached a certain stage of development, they came to realize that the existing programming languages could not meet all their requirements, so a decision was made to create a new language platform. The platform design was drawn from a variety of languages including Eiffel, SmallTalk, Objective C, LISP and Cedar/Mesa.

The design requirements for the Java language were clearly defined. The language had to be simple, object-oriented, familiar, robust and secure, architecture neutral, portable, have a high performance, and be interpreted, threaded, and dynamic. This gives the language its strengths and is designed from the ground up with networking in mind.

- The beginning of Java starts in 1991
- Originated by Sun Microsystems’ James Gosling et.al.
- Project code-named ‘Green’ to produce products for the consumer electronics industry.
- Products had to be very small and would be based on many computer architectures
- Started with C++, but found it to be unsatisfactory
- Designed their own language called ‘Oak,’ which later became Java.
- At the same time the World-Wide Web was growing fast.
- Gosling et. al. realized that they could write a Cool Java Browser in Java
- HotJava was shown at Sun World ’95 on May 23, 1995
- In the fall of 1995, Netscape decided to make Netscape 2.0 Java-enabled, and released it in January 1996
- The Java craze started….

Oracle Corporation Java Thrust

Oracle 8i uses Java in addition to PL/SQL as a way to internally program the database. Java is strategic technology to Oracle, and its programs may be tightly integrated with the database. It was chosen for the application because of its ability to provide e-commerce

http://Niagara.river.edu
solutions fueled by the world’s leading relational data plant manufacturing automation system database technology.  

**JNI to OPC Interface**

A goal of the gravimetric fill system was to minimize the number of programming tools required to develop the GPower application. Java technology in conjunction with the Oracle 8i database provided the best long term solution for the new application. No other language provides the e-commerce advantages of the Java language. JNI (Java Native Interface) is the Java standard to interface Java to external software.

The BOC communications between the web browser and the Siemens PLC is accomplished by using a small C++ (InfoTech developed DLL (Dynamic Link Library)) program as the go between JNI and a 3rd party DLL licensed from Softing Inc. This DLL incorporates all of the OPC functionality of the Data Access 2.0 OPC standard. This approach allowed the use of a pure Java interface to OPC software running on a Windows NT server. The interface to the utility uses another Java standard called RMI (Remote Method Invocation). This design is scalable as well as flexible, since the components may be separated on different servers if the need for increased performance arises.

**Automation Model for GPower**

---

3 See Attached Oracle White Paper “Delivering the Promise of Internet Computing”
The above model represents the communications system that provides reliable and accurate communications to support plant automation. The color codes indicate elements which can be distributed across as many server systems as required to support system scalability. Presently, all application code resides on the Oracle server. True thin client (the only requirement for clients is a web browser) multi-user connectivity using Internet technologies is achieved.

**Five Primary elements** The heart of the system consists of:

- *Siemens PLC Wonderware HMI served by the Siemens OPC server. This constitutes the plant automation and control system*

Using OPC as the vehicle to communicate to the automation system insulates the information engineering system from the HMI vendor and the automation technology. This allows for change out of the technology without requiring ANY changes to the plant manufacturing automation system.

- *Client workstation. This browser based computer requires no software other than an Internet browser. With proper security, this terminal can be any place on the planet with access to the Internet. It is used to access and control recipe specifications and blend recipes.*

The system is designed to facilitate the separation of recipe specification (what the customer needs) from the batch recipe (the know how to produce the product to meet the recipe specification to satisfy the customer requirements)

- *The Oracle 8i database is the center piece of the system. It provides the repository to contain the information to fuel the running of the business. It contains all relevant information about schedules, orders, recipe specification, batch recipes, and actual production data on batch produced.*
The next element is the Java / OLE for Process Control interface. It is the mechanism used to communicate between the plant manufacturing automation system and the plant Controls. This open standards base interface allows the plant manufacturing automation system to send commands to the production control system to execute actions to produce customer products.

This component provides the mechanism with which communication between the web server / Oracle 8i database and the process control system is achieved. OPC provides reliable connections to transfer time-stamped values which have associated quality indicators. It is robust and can sustain high data throughput.

The web server is the last element of the system which provides multi user connections to the Oracle 8i database. Client connections from the browser based computers interface to the system using web pages. The Java programs run on the server and handle transactions to the database. The Java Server programs also exchange information between the user, database, and control system.

**Description of the interactions among elements**

The following steps describe how the elements of the system interact to exchange information.

1. Browser based client logs into the system using username and password.
2. HTML (Hypertext Markup Language) forms are used to navigate the system.
3. All interactions to specify recipe specifications, batch recipes, enter orders, schedule production and execute batches are interactions between the web server and the Oracle 8i database.

4. HTML screens designed to sequence the production require the web server to communicate with the control system.

5. When communications with the control system are desired, the web server calls a method (function). Java knows that the method (function) is not local to the executing Java thread. It locates the OPC facilities by contacting the Java Registry and initiating a network request.

6. The Java Registry looks up the method required and establishes an executing Java thread to service the incoming request.

7. JNI (Java Native Interface) knows how to communicate to the outside world (OPC) and asks the Softing DLL to locate the OPC server.

8. OPC uses DCOM (Distributed Component Object Model) to communicate over the network to locate the named OPC server. Subject to NT security, the OPC server(s) establish the connection.

9. All connections are established and persist throughout the communications session until terminated by the web server.

10. Information is exchanged between the web server and the OPC interface to accomplish all information exchange.

11. A simple transfer protocol is used to hand shake the data to and from the automation system (see attached flow charts).

12. The web server exchanges data between the client and Oracle 8i database as required.

The same facilities can be used to communicate between the plant’s manufacturing automation system and any of the multitudes of available OPC compliant servers, such as accounting and management databases.