Introduction

Current architectural trend is for an enterprise to wrapper applications with Web Services in what is called the Service-Oriented Architecture (SOA) in hopes of creating an enterprise architecture that is flexible to the needs of various information consumers. Similar trend is in play for ASP and eCommerce sites like SalesForce, eBay, and Amazon. They want to enable customization and interoperability with their platforms through Web Services.

This means the functionality contained within these applications is now accessible to applications that can support Web Services requests. But most of the desktop applications used by business users run as Windows or Unix applications that do not have built in support for accessing services through SOAP. There is a need to migrate to SOA but at the same time there is a demand from the users to maintain compatibility with the 100s of applications they use for reporting and analysis.

The use of OpenAccess™ to implement a custom ODBC or JDBC driver with full SQL capability is the best solution to bridging the current applications and the Web Services worlds. Most of today’s reporting, analysis, and database applications have the ability to interface to data sources through ODBC or JDBC. With OpenAccess you can leverage this widespread feature to enable your Web Services compliant application to be accessed without requiring clients to change their applications.

OpenAccess Solution

OpenAccess SDK provides the framework and pre-built components to quickly allow the implementation of a custom JDBC and ODBC driver for any data source that has a Web Services interface.

For example, to ODBC enable a Web Services like the ones exposed by SalesForce.com or Amazon.com, the steps are: 1) obtain the WSDL, 2) define how this service will be exposed in terms of virtual tables, 3) use a tool like Visual Studio.NET to generate wrapper classes to access the Web Services methods, 4) and implement glue
code (<500 lines of code) to tie in this class to the OpenAccess SQL engine to support schema and data access requests. Assuming you are exposing a virtual table ACCOUNTS, a user can perform a query like:

```sql
select * from ACCOUNTS where balance > 3000
```

would be handled through the use of the custom IP code to make a WS method calls to return XML formatted data for the ACCOUNTS and then mapping the XML formatted data into rows.

**Your Development Effort**

1. Design and code the adapter code in either C, C++, Java, or .NET (5 days)
2. Do your QA (2 days)
3. Package up for distribution (1 day)

Expected time of completion: **8 man days**

---

**Conclusion**

The ability to expose a Web Service as a virtual SQL database with ODBC and JDBC allows 100s of Windows and Unix applications to leverage the investment being made in SOA. OpenAccess allows the implementation of an enterprise quality ODBC or JDBC driver by leveraging the OpenAccess platform that includes 99% of what is needed to implement a driver. The remaining 1% is specific to the Web Service and can be coded in Java or .NET in a very short time because both of these environments provide rich support for Web Service access and XML processing.