Building a Web Services ODBC Driver

Using .NET, C# and Microsoft® HealthVault® as an Example

By Simba Technologies Inc.

Prerequisites

This white paper assumes that you are familiar with ODBC and C/C++ development under Microsoft® Visual Studio®, as well as familiar with .NET development in C# and web services. Experience with Microsoft HealthVault® is a bonus.

You will need:

- Microsoft Visual Studio 2008.
- The HealthVault SDK: available from the HealthVault Developer Center at http://msdn.microsoft.com/en-us/healthvault/.
- SimbaEngine SDK: available from the Simba web site at http://www.simba.com/odbc-sdk.htm. Click on Free Download and follow the instructions.

Introduction

This white paper describes the steps required to build an ODBC driver with .NET and C# using SimbaEngine SDK. For this example, we use the Microsoft HealthVault web service as a convenient and universally-available data source that can be accessed via its web service. The sample code demonstrates one way to use .NET and C# to easily connect SimbaEngine to the Microsoft HealthVault web service using the tools available in Microsoft Visual Studio.

HealthVault is Microsoft's platform to store and maintain health and fitness information. It provides a robust and substantial web service along with tools to help the developer access the data. You can find out more about HealthVault at http://msdn.microsoft.com/en-us/healthvault/. The sample ODBC driver described here allows ODBC client applications, such as Microsoft Excel®, to retrieve data directly from HealthVault, as if it is a local relational database like Microsoft Access®. The same strategy can be used to access other data sources that are available through web services.

SimbaEngine SDK provides a C++ ODBC interface and SQL engine that require only a data source code layer, the Data Record Manager or DRM, to connect to a data source. The DRM code maps the data source API to the SQL engine. This is illustrated in Figure 1. DRM code has been written that maps a wide range of data types and storage schemes to the SQL engine DRM API.

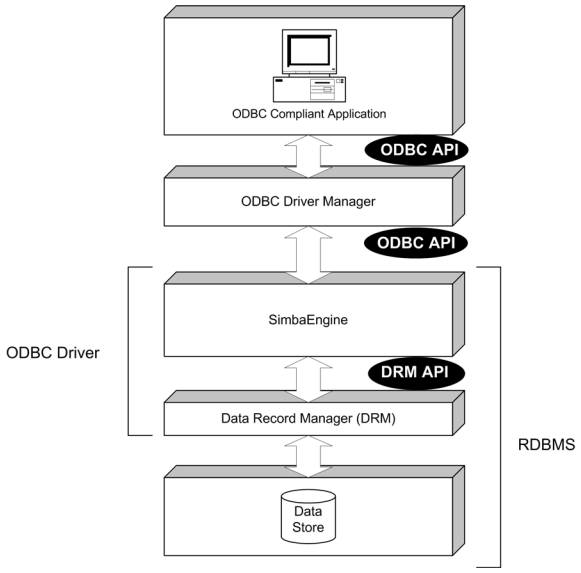

Fig. 1: A typical SimbaEngine ODBC driver architecture using a local data source.

Once a DRM is written for a data source, the data is available to the wide range of ODBC-enabled applications. The data source does not have to be relational, or even arranged in tables and columns. In this example, the HealthVault data is notionally stored as tables and columns, but it is made available through the web service in XML documents. The sample DRM described here maps the XML documents to the SQL engine.

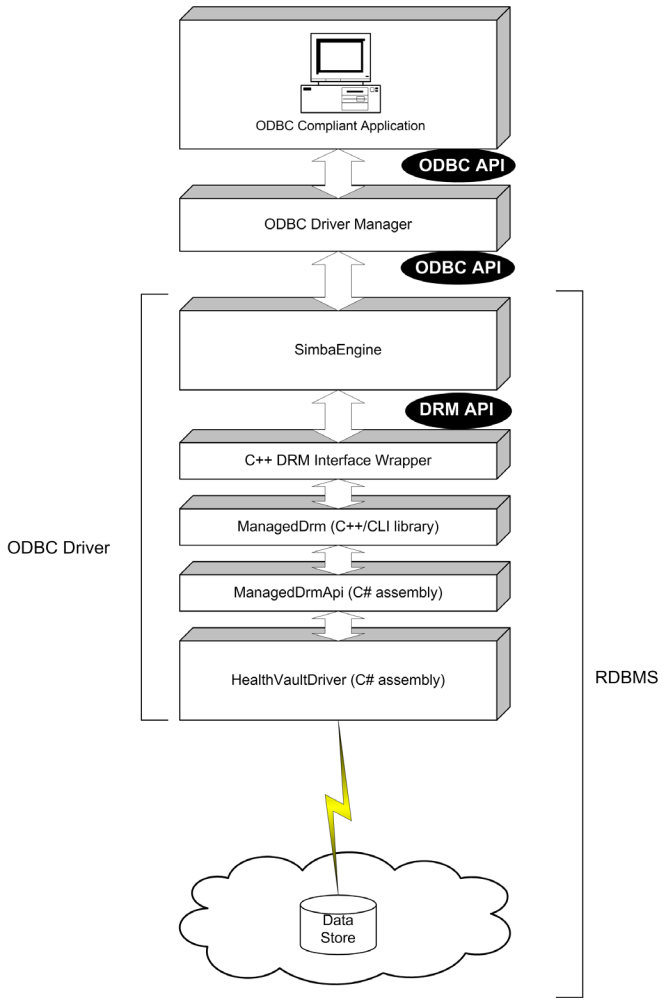

Fig. 2: The SimbaEngine HealthVault ODBC driver architecture showing various layers.

The custom DRM that connects to HealthVault is written in C# and communicates with HealthVault via the HealthVault SDK. SimbaEngine interacts with the .NET DRM via a bridging library written in C++/CLI.[1] Note that while this example uses C# to implement the .NET DRM, it could have been written in any .NET language, such as IronPython or IronRuby.

The sample files for this C# .NET DRM are in the *<Install Dir>\SourceCode\Engine\DataRecordManager\DotNetSample* directory installed with SimbaEngine SDK. The directory contains a VisualStudio 2008 solution called DotNetSample that contains the following projects. Figure 2 shows how the project output files are arranged in the driver.

ManagedDrm

This project creates a C++/CLI library that fulfills the role of a traditional DRM code by linking directly to the SQL engine. It loads the managed C# driver assembly and acts as a bridge between SimbaEngine and the .NET part of the DRM code by calling the .NET DRM API code found in the ManagedDrmApi project.

ManagedDrmApi

This is a C# assembly that defines the DRM API for the .NET code. The C# .NET code must implement the interfaces provided by this assembly. This assembly also contains a number of type conversion helper classes to make the API implementation easier.

HealthVaultDriver

This is a C# assembly that implements the .NET DRM interface specified in the ManagedDrmApi project. It actually makes the connection to the web services using the C# HealthVault SDK and provides read-only access to the HealthVault data store.

HealthVaultDriverTest

This is a C# test application that uses the ADO.NET *OdbcConnection* classes to load and test parts of the HealthVault driver via ODBC. If you are transforming this DRM code to use a web service other than HealthVault, this test application is useful for incremental code testing. It reduces test complexity by eliminating SimbaEngine from the test system.

HealthVaultDriverAuth

Before any HealthVault application can access a user's data, that user must grant the application permission to do so. At the time of writing, this can be done simply via a web application called HealthVaultDriverAuth. This is a C# ASP.NET application that you start in VisualStudio, which guides you through the process of creating a HealthVault account and granting the ODBC driver access to the HealthVault data using this web application. The use of this application may change over time.

1 For more information on Microsoft's C++/CLI, please refer to http://msdn.microsoft.com/en-us/library/ms379617(VS.80).aspx.

Building the .NET C# HealthVault ODBC Driver

Building the .NET C# HealthVault ODBC driver is easy once you have all the SDKs and tools properly installed on your computer. Follow these steps to set up and build the complete driver.

1. The solution files in this example are compatible with Microsoft Visual Studio 2008, the HealthVault SDK and SimbaEngine SDK. The HealthVault SDK can be downloaded from the HealthVault Developer Center at http://msdn.microsoft.com/en-us/healthvault/. SimbaEngine SDK can be obtained from the Simba web site at http://www.simba.com/odbc-sdk.htm; click on *Free Download* and follow the instructions.

2. Install the HealthVault SDK and SimbaEngine SDK and verify that they are working properly.

3. You will need to add the HealthVault SDK assemblies to the Windows Global Assembly Cache.[2] There is a script in SimbaEngine SDK to automate this. Look for the batch file *'Install_HealthVault.bat'* in *<Install Dir>\SourceCode\Engine\DataRecordManager\DotNetSample\HealthVaultDriver* and run it.

4. Once the HealthVault and SimbaEngine SDKs are installed, navigate to *<Install Dir>\SourceCode\Engine\DataRecordManager\DotNetSample* in SimbaEngine SDK installation. Open the solution file *'DotNetSample.sln'* in Visual Studio 2008 and build both the debug and release configurations. This will build the actual C# DRM and ODBC driver that will connect to the HealthVault web service.

Preparing and Using the HealthVault ODBC Driver

Because the HealthVault database contains health and medical information, there are safeguards built into the system to prevent abuse. These involve encryption and authentication using certificates that must be properly set up on your computer before you can access the HealthVault web service. This is the case no matter how you choose to access HealthVault. Follow these steps to register and authenticate yourself with HealthVault and to test your new ODBC driver.

1. You will need to install the application certificate that identifies the ODBC driver to HealthVault. To do this, click on *'Start'* and navigate to the *'Microsoft HealthVault\SDK'* menu. Run the *'HealthVault Application Manager.'* When this is running, click on the *'Import pfx'* button. Select the *'HealthVaultDriver.pfx'* file located in *<Install Dir>\SourceCode\Engine\DataRecordManager\DotNetSample\HealthVaultDriver*. When the certificate has been successfully imported, you should see the *'HealthVaultDriver'* certificate in the certificate list. Once the certificate has been successfully imported, you can close the *'HealthVault Application Manager'*.

2. Before you can connect to the HealthVault web service and access data, you must create a HealthVault account and grant your new ODBC driver permission to access your data. You can do this as follows by using the HealthVaultDriverAuth ASP.NET application. From within Visual Studio 2008, right click on the HealthVaultDriverAuth project and select *'View in browser'*. The HealthVaultDriverAuth application will open in your default browser and guide you through the HealthVault account creation and authorization process. When the authorization is complete, you will be given the opportunity to add test data to your account. Data in your HealthVault account is required to run the ODBC test application, HealthVaultDriverTest, and it is much more interesting to see real data come back to Excel from HealthVault. Alternatively, you can connect to the HealthVault development environment. The driver will do this by default.

3. Once the ODBC driver has been authorized and test data has been added to your HealthVault account, you should be able to run the HealthVaultDriverTest test application and have all the tests pass.

2 For more information on the Global Assembly Cache please refer to http://msdn.microsoft.com/en-us/library/yf1d93sz(VS.71).aspx.

4. In order to use the driver with other ODBC client applications like Microsoft Excel, you will need to install the driver. There is a script in SimbaEngine SDK to automate this. Navigate to *<install dir>\SourceCode\Engine\DataRecordManager\DotNetSample\HealthVaultDriver*. Look for the batch file *'Install_Release.bat'* and run it. This script will create a Health Vault ODBC DSN on your computer, and it will install the driver and its .NET dependencies into the .NET Global Assembly Cache.

You should now be able to run Excel and retrieve your test data directly from HealthVault, as if it is a local relational database. Run Excel and click on the *Data tab* (for Excel 2007). In the *Get External Data* group, choose *From Other Sources* and click on *From Microsoft Query*. Select the *'HealthVault Release DSN'* in the *Choose Data Source* dialog box and click *OK*. Pick all of the tables from the *Query Wizard – Choose Columns* dialog box and click *Next*. Click *Next* for *Filter Data and Sort Order*, and click on *Finish* in the *Finish* dialog box. The *Import Data* dialog box will appear asking you where to put the data. Click *OK*. After a moment, your HealthVault data should populate your spreadsheet.

Congratulations! You have retrieved database data from a web service directly into Excel using SimbaEngine SDK!

Conclusion

The SimbaEngine SDK .NET C# DRM can be used as is to retrieve data from HealthVault, but it can also be used as a base from which you can build an ODBC driver for your own web service. Web services are becoming the portal of choice for serving up information on the Internet, and the .NET platform makes it easy to access those web services. The SimbaEngine SDK .NET C# DRM makes it easy to allow popular ODBC-enabled applications access web services-based information using the power of the .NET platform.

About Simba Technologies Inc.

Simba Technologies Inc. is the recognized world leader in standards-based data access products and solutions. Simba works with the world's leading software companies to deliver first class data connectivity solutions.

Simba is a pioneer in ODBC, MDX, ODBO and XMLA. Since 1991, Simba has developed advanced data access solutions for thousands of end users. Today, more than half of all MDX providers have been built with Simba technology, and through a partnership with Microsoft, Simba's SQL technology has been installed on more than 30 million desktops worldwide.

Simba Technologies Incorporated
938 West 8th Avenue
Vancouver, BC Canada
V5Z 1E5

Tel. +1.604.633.0008
Fax. +1.604.633.0004
Email. solutions (at) simba.com
www.simba.com