



SimbaProvider SDK 4.6

Quick Start Guide

October 6, 2016

Simba Technologies Inc.



Copyright ©2016 Simba Technologies Inc. All Rights Reserved.

Information in this document is subject to change without notice. Companies, names and data used in examples herein are fictitious unless otherwise noted. No part of this publication, or the software it describes, may be reproduced, transmitted, transcribed, stored in a retrieval system, decompiled, disassembled, reverse-engineered, or translated into any language in any form by any means for any purpose without the express written permission of Simba Technologies Inc.

Trademarks

Simba, the Simba logo, SimbaEngine, and Simba Technologies are registered trademarks of Simba Technologies Inc. in Canada, United States and/or other countries. All other trademarks and/or servicemarks are the property of their respective owners.

Simba Technologies Inc.

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

Tel. +1 (604) 633-0008
Fax. +1 (604) 633-0004

www.simba.com

Printed in Canada

Table of Contents

- Introduction 1
 - Who Should Read this Guide 1
 - The SimbaProvider OLAP Packages 1
 - Application Compatibility..... 4
- Installing the SDK and Sample Providers6
 - Hardware and Software Requirements..... 6
 - Configuring your Windows Development Environment..... 6
 - Installing the Windows Base Kit 8
 - Configuring your Linux Development Environment..... 9
 - Installing the Linux Base Kit 11
- Building the Sample Providers 12
 - Building the ODBO Sample Provider 12
 - Building the XMLA Sample Providers..... 12
- Testing the Sample Providers 16
 - Testing and Debugging the ODBO Provider..... 16
 - Testing and Debugging the .NET XMLA Provider..... 17
 - Testing and Debugging the Java XMLA Provider for Windows..... 19
- Resources..... 21
 - How to Contact Us..... 21
 - Getting Additional Information on XMLA and OLE DB 22

Table of Figures

- Figure 1 - JAMS.....3
- Figure 2 - Dataset Viewer3
- Figure 3 - Rowset Viewer.....4

Introduction

Welcome to the SimbaProvider for OLAP SDK™ 4.6 Installation Guide. This document contains all the steps needed to install and prepare the SDK for development and testing.

Who Should Read this Guide

This guide is for both new customers and those already familiar with the SimbaProvider for OLAP SDK™ and includes the steps necessary to install, build, and test the sample OLAP providers.

This guide assumes you have a working understanding of modern database principles and terminology, the C++ language, your development environment (compiler, linker, libraries), and web application principles and terminology. If you will be developing in Java, you should also be familiar with the Java language, the J2EE specification, and your particular web server (this document covers the usage of JBoss).

After you have installed and built the providers using this guide, refer to the SimbaProvider Development Guide for information on developing your own provider using the sample provider.

The SimbaProvider OLAP Packages

The SimbaProvider for OLAP SDK™ is a collection of 32 and 64-bit tools, code, and technologies for building an online analytical processing (OLAP) provider which can be used for business intelligence and other multi-dimensional analysis. This enables front-end consumer applications to access your OLAP data sources in a generic and consistent way.

OLAP providers allow consumer applications to run Multidimensional Expressions (MDX) queries to perform operations against an OLAP data source. MDX is the query language used for accessing data in OLAP data sources.

This release of SimbaProvider allows you to build two types of providers for accessing OLAP data sources:

- **OLE DB for OLAP (ODBO):** ODBO is a Microsoft technology that extends OLE DB to access an OLAP data source. ODBO providers are specific to Windows and are built as COM objects for local use on a Windows system. Therefore, the SimbaProvider OLAP SDK™ allows you to build an ODBO provider in native Win32 C++ (for both 32-bit and 64-bit targets).
- **XML for Analysis (XMLA):** an XML-based data access standard built on XML, SOAP, and HTTP, and the successor to ODBO. XMLA providers are always built as web services which provide their data over a network, thus allowing them to be built for different platforms while providing platform agnostic access to the data. The SimbaProvider OLAP SDK™ allows XMLA providers to be built in 64-bit binaries using Java or .NET. Both targets however, use an ISO/ISM implementation written in native C++.

The SimbaProvider for OLAP SDK™ contains components that fulfill the API requirements for both ODBO and XMLA interfaces. These components take care of most of the details of these APIs, maximizing interoperability with front-end clients and freeing you to focus on the provider's core services (e.g., session management, metadata retrieval and caching, and MDX statement semantic implementation, etc).

The SDK also includes a sample data source implemented using .csv (text) files. This allows you to get up and running quickly with the sample since no additional drivers or software (e.g. servers) are necessary. Note that prior to SimbaProvider for OLAP SDK™ 4.5, the sample data required SQL Server, Microsoft Analysis Services, and other components to be installed. Existing users who have followed these steps in the past will therefore need to review the new sample code to see how access to the new text-based sample data is being performed.

SimbaProvider for OLAP is distributed in "kits" as described in the following sub sections. The kits are available in both *evaluation* and *release* versions. The *evaluation* version is a free version provided by Simba that includes a license enabling functionality for a limited time period. The *release* version, which must be purchased from Simba, removes this restriction and also includes the source code implementation for the sample provider's MDX functions.

Windows Base Kit

The Base Kit contains the source code and SimbaProvider binaries required to develop an ODBO and XMLA provider on Windows using Java, native Win32 C++, and .NET. The kit includes sample provider implementations for all three platforms that you can use as a guide or starting point in developing your own provider.

The kit also includes the following tools in both 32-bit and 64-bit versions that are helpful for testing MDX commands against the sample provider. The steps described in this document however will use Microsoft Excel.

- **JAMS** ("just another MDX sample"): a Simba tool that shows an OLAP catalog hierarchy in a convenient tree view format and allows ad-hoc MDX commands to be executed against an ODBO data source.

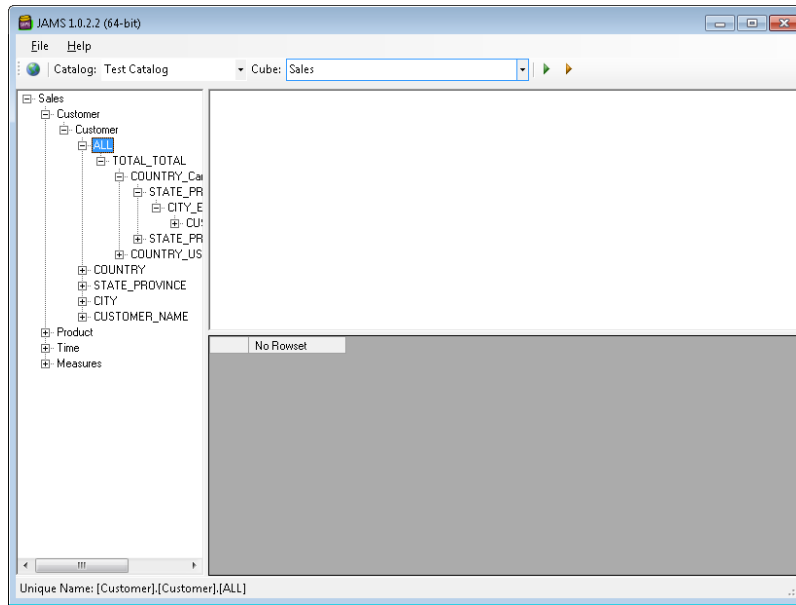


Figure 1 - JAMS

Note: JAMS does not support querying of the schema metadata. Use the following two tools instead to view that information.

- **Dataset Viewer:** a Microsoft tool included as part of the Microsoft Data Access SDK (MSDASDK) prior to version 2.6. Dataset Viewer allows data from an ODBO source to be viewed. The version of Dataset Viewer included in the SimbaProvider for OLAP SDK™ has been modified from the original source to fix several known bugs.

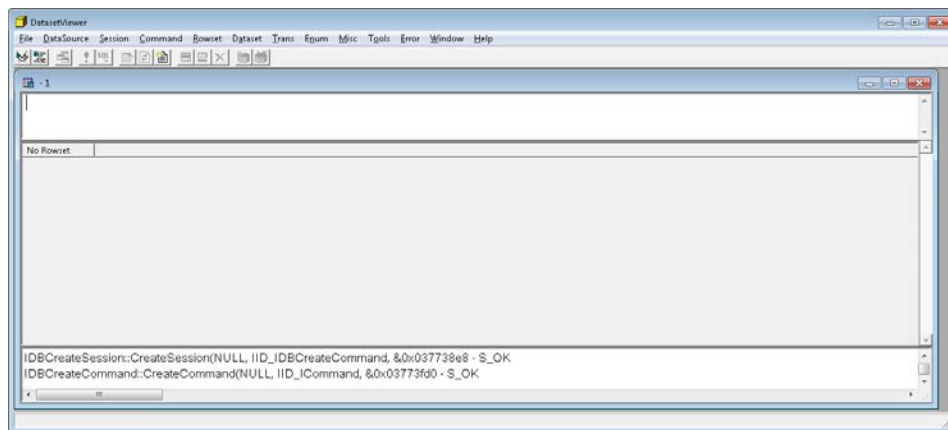


Figure 2 - Dataset Viewer

- **Rowset Viewer:** another Microsoft tool for viewing data from an ODBO source. This tool offers the same functionality as Dataset Viewer but originates from the Microsoft Platform SDK.

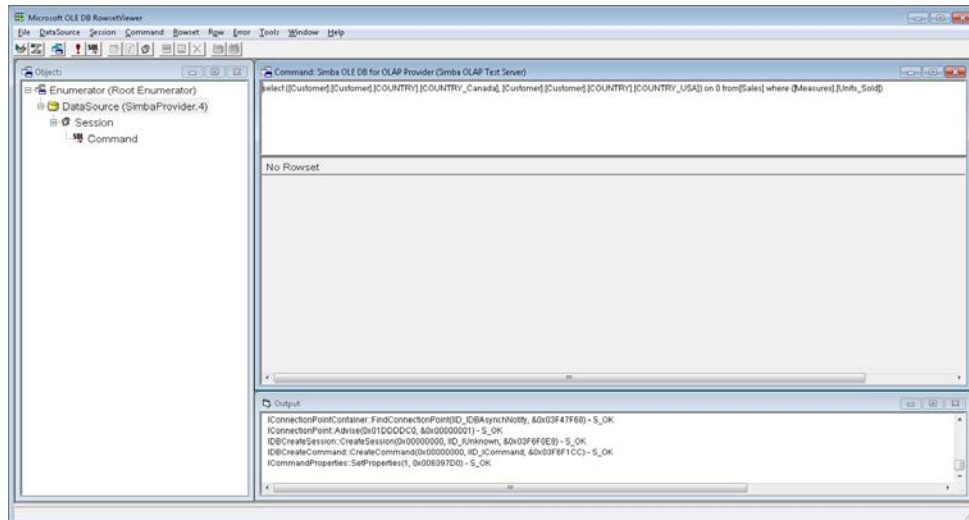


Figure 3 - Rowset Viewer

All three of the tools mentioned above only work with ODBO OLAP data providers. To test against an XMLA provider, you need to use SimbaO2X (described below).

Linux Kit

The Linux Kit contains the source code and SimbaProvider binaries to enable the development of an XMLA provider on Linux in Java. The kit includes a sample provider implementation that you can use as a guide or starting point in developing your own provider.

SimbaO2X

SimbaO2X is a binary component that bridges ODBO-to-XMLA. This enables you to connect an ODBO-based reporting tool like Excel to any XMLA server such as an XMLA provider that you might develop using the SimbaProvider for OLAP SDK™. SimbaO2X is available in 32-bit and 64-bit versions to support 32-bit or 64-bit ODBO applications. SimbaO2X is available free when you purchase the release version of the SimbaProvider for OLAP Base Kit.

Application Compatibility

SimbaProvider for OLAP SDK™ 4.6 has been tested with and works with the following applications:

- Microsoft Excel 2010, 2013, and 2016
- Microsoft Reporting Services 2014
- MicroStrategy 10.2
- XLCubed 9
- Tableau 10
- ADOMD.NET

To inquire about compatibility with Cognos, SAP, Infor, arcPlan or other applications, contact Simba at support@simba.com.

Installing the SDK and Sample Providers

The following subsections outline the hardware and software requirements; how to install the required tools; and, how to configure your development environment.

Hardware and Software Requirements

Hardware and OS

- 1 GHz Pentium processor or better
- 2 GB RAM minimum, 4GB RAM recommended
- Windows Server 2008 R2(64-bit), Windows Server 2012 (64-bit), Windows 7, Windows 8, Windows 10
Note: The XMLA providers are only supported in 64-bit versions and not 32-bit.
- SUSE Linux ES11 SP3 (x64)
- 2 GB of free disk space (depending on build configurations; not including disk space requirements for third-party software)

Development Software

- Microsoft Visual Studio 2015
- Microsoft Excel 2010, 2013, and/or 2016 (for testing purposes)

If you plan to develop in Java, you will also need:

- JDK 1.8 SDK (64-bit)
- Eclipse 4.4 or higher
- JBoss EAP 6.4 (or higher) or Tomcat 7 (or higher)
- Ant 1.9

If you plan to develop in Java on Linux, you will also need:

- GCC 4.7

Configuring your Windows Development Environment

Administrator privileges, or root on Linux, are required to complete the following procedure. Ensure that you are logged into your machine using an account that has administrator privileges.

Note: In Windows, you may need to elevate your privileges due to User Account Control (UAC) and will need to run some programs (e.g. Visual Studio) under administrative mode.

To install the development tools from the SimbaProvider for OLAP SDK™:

1. Install Microsoft Visual Studio 2015 as required for your development needs.
 2. Install the following Java development tools, if you are planning to develop a provider in Java:
 - **Java Platform, Standard Edition 8 (Java SE 8) SDK (64-bit):** this can be downloaded from <http://www.oracle.com/technetwork/java/javase/downloads/index.html>. Be sure to install the full JDK and not just the JRE.
 - **Eclipse 4.4 or higher:** this is available at <http://www.eclipse.org/downloads/>. Simba has chosen to support Eclipse but you are free to migrate/convert this to your desired IDE.
 - **JBoss EAP 6.4 or higher:** this is available at <http://developers.redhat.com/products/eap/download/>. We recommend that you install JBoss in a directory with a short pathname and without spaces (e.g. c:\jboss-eap-6.4) as longer paths might make deployment and troubleshooting more difficult.
- OR
- **Tomcat 7 or higher** (if not running JBoss): this is available at <http://tomcat.apache.org/whichversion.html>.
 - **Apache Ant v1.9 or higher:** this is available at: <http://ant.apache.org/bindownload.cgi>.
3. Configure your Java environment as follows:
 - **JAVA_HOME:** create an environment variable called JAVA_HOME and point it to where your JDK is installed (e.g. c:\Java\jdk8). Check to make sure that the Java binaries in your PATH are the same Java binaries that are part of the JDK above. NOTE: build the Java XMLA provider in 64 bit and ensure that JAVA_HOME to point to the 64 bit JDK. This variable will be used when you startup JBoss.
 - **Ant binary:** include the Ant binary directory in your PATH (e.g. c:\Ant\bin).
 - **JBOSS_HOME:** create an environment variable called JBOSS_HOME that points to where you have installed your JBoss application server (e.g. c:\jboss-eap-6.4).

Installing the Windows Base Kit

To install the Windows Base kit:

1. Run one of the following the SimbaProvider for OLAP SDK™ Base Kit installers to install the Base Kit:

Evaluation versions: if you are evaluating the SimbaProvider OLAP SDK™, select the following installer:

- SimbaProvider_Windows_Eval_BaseKit_VS2015_4.6.x.x.zip

Release versions: if you are using the release version of the Simba OLAP SDK™, select the following installer:

- SimbaProvider_Windows_Release_BaseKit_VS2015_4.6.x.x.zip

2. Setup your license files if you are running the evaluation version of the Windows Base Kit as follows:

32-Bit Windows

- Copy *libeay32.dll* and *SimbaProviderSDK.lic* from
`<install folder>\SimbaProvider for OLAP SDK Eval
 4.6\Simba\Common\ThirdParty\openssl\VS_32\bin`
 to
`C:\Windows\System32`

64-Bit Windows

- If you are building a 32-bit provider:
 Copy *libeay32.dll* and *SimbaProviderSDK.lic* from:
`<install folder>\SimbaProvider for OLAP SDK Eval
 4.6\Simba\Common\ThirdParty\openssl\VS_32\bin`
 to
`C:\Windows\SysWOW64\`
- If you are building a 64-Bit provider:
 Copy *libeay32.dll* and *SimbaProviderSDK.lic* from
`<install folder>\SimbaProvider for OLAP SDK Eval
 4.6\Simba\Common\ThirdParty\openssl\VS_64\bin`

to

`C:\Windows\System32`

After installing the SimbaProvider for OLAP SDK™ Base Kit for Windows, your installation directory will contain the following subdirectories.

- `<InstallPath>/SimbaProvider for OLAP SDK 4.6/Docs`: contains the SimbaProvider OLAP Installation Guide (this document), the SimbaProvider Developers guide and the MDX Engine Developer's Guide.
- `<InstallPath>/SimbaProvider for OLAP SDK 4.6/SampleProvider/Win32`: contains the source code and Microsoft Visual Studio project files to build the sample provider.
- `<InstallPath>/SimbaProvider for OLAP SDK 4.6/SampleProvider/Win32/DataStore/`: contains a sample datastore that provides access to data stored in text (.csv) files. The .csv files are under a sub folder called *TestData*.
- `<InstallPath>/SimbaProvider for OLAP SDK 4.6/Simba`: contains the header files and Simba components that you will need to build the sample provider.
- `<InstallPath>/SimbaProvider for OLAP SDK 4.6/Tools`: contains JAMS, Dataset Viewer, and Rowset Viewer tools for testing while developing your provider.

The following environment variable will be created:

- **SIMBAHOME**: set to `<InstallPath>\SimbaProvider for OLAP SDK 4.6`

The following registry settings will be created. The following shows a sample; the actual path will depend on your installation folder:

32-Bit Windows:

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Simba]
"InstallPath" = "<InstallPath>\SimbaProvider for OLAP SDK 4.6\"
```

64 bit windows:

If you are building 32 bit provider:

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Simba]
"InstallPath" = "<InstallPath>\SimbaProvider for OLAP SDK 4.6\"
```

If you are building 64 bit provider:

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Simba]
"InstallPath" = "<InstallPath>\SimbaProvider for OLAP SDK 4.6\"
```

Configuring your Linux Development Environment

To configure your Linux Development Environment:

1. Install the following Java development tools, if you are planning to develop a provider in Java:
 - **Java Platform, Standard Edition 8 (Java SE 8) SDK (64-bit)**: this can be downloaded from <http://www.oracle.com/technetwork/java/javase/downloads/index.html>. Be sure to install the full JDK and not just the JRE.
 - **Eclipse 4.4 or higher**: this is available at <http://www.eclipse.org/downloads/>. Simba has chosen to support Eclipse but you are free to migrate/convert this to your desired IDE.
 - **JBoss EAP 6.4 or higher**: this is available at <http://developers.redhat.com/products/eap/download/>. We recommend that you install JBoss in a directory with a short pathname and without spaces (e.g. c:\jboss-eap-6.4) as longer paths might make deployment and troubleshooting harder.

OR

 - **Tomcat 7 or higher** (if not running JBoss): this is available at: <http://tomcat.apache.org/whichversion.html>.
 - **Apache Ant v1.9 or higher**: this is available at <http://ant.apache.org/bindownload.cgi>.
2. Setup the following environment variables:
 - **JBOSS_HOME** (if using JBoss): set to the JBoss installation folder (e.g. /home/<your name>/jboss-eap-6.4).
 - **CATALINA_HOME** (if using Tomcat): set to the Tomcat installation folder (CATALINA_HOME=/home/<your name>/TomcatStandalone7).
 - **JAVA_HOME**: set to the JDK installation folder (e.g. /usr/java/jdk1.8.0_91).
 - **SIMBAPROVIDER_DATAFOLDER**: set to the location of the sample data files included with SimbaProvider for OLAP SDK™ Base Kit (e.g. set to /home/<your name>/SimbaProvider-4.6.x.x/SampleProvider/Win32/DataStore/TestData/).
 - change **LD_LIBRARY_PATH** (for Evaluation version only): add the following path to LD_LIBRARY_PATH:
 <InstallPath>/SimbaProvider-4.6.x.x/Simba/Linux/lib/x64/release

OR

 - <InstallPath>/SimbaProvider-4.6.x.x/Simba/Linux/lib/x64/debug
3. Set the log location in *tracesettings.conf*:

The sample `tracesettings.conf` is under `<InstallPath>/SimbaProvider-4.6.x.x/SampleProvider/Win32/Provider/LogImpl/Configuration/LinuxConfiguration`. Make sure that the path to write the log file to is writable (the path is specified by "LogEx.LogWriter.LogFile.Path" in `tracesettings.conf`).

Installing the Linux Base Kit

To install the Linux Base Kit:

1. Decompress the SimbaProvider for OLAP SDK™ Linux Base Kit package to your home directory:
 - `> Tar xzvf vm_SimbaProvider_Linux_EvalKit_4.6.x.x.tar.gz` (for the evaluation kit)
 - `> Tar xzvf vm_SimbaProvider_Linux_ReleaseKit_4.6.x.x.tar.gz` (for the release kit)
2. Configure the license file if you have installed the evaluation version of the Linux Kit:
 - Create a folder in `/etc` called `MdxProvider`.
 - Copy the license file `SimbaProviderSDK.lic` that was provided by Simba, into `/etc/MdxProvider`.

After installing SimbaProvider for OLAP SDK™ Base Kit for Linux, your installation directory will contain the following subdirectories.

- `<InstallPath>/SimbaProvider-4.6.x.x/SampleProvider/Linux`: contains the makefiles to build the sample provider
- `<InstallPath>/SimbaProvider-4.6.x.x/SampleProvider/Win32`: contains all of the code required to build the sample provider.
- `<InstallPath>/SimbaProvider-4.6.x.x/SampleProvider/Docs`: contains the SimbaProvider for OLAP SDK™ Installation Guide (this document), the SimbaProvider Developers guide and the MDX Engine Developer's Guide (included in the release kit only).
- `<InstallPath>/SimbaProvider-4.6.x.x/SampleProvider/Win32/DataStore`: contains a sample datastore that provides access to data stored in text (.csv) files. The .csv files are under a sub folder called `TestData`.
- `<InstallPath>/SimbaProvider-4.6.x.x/Simba`: contains the header files and Simba components that you will need to build the sample provider.

Note: The Linux Base Kit does not include test tools.

After installing the Base Kit for Linux, you will need to create one environment variable:

- `SIMBAHOME`: set to `<InstallPath>/SimbaProvider-4.6.x.x`

Building the Sample Providers

Building the ODBO Sample Provider

To build the ODBO sample provider:

1. Start Microsoft Visual Studio 2015.
2. Select *File | Open Solution*.
3. Open the *<Install Path>/SimbaProvider for OLAP SDK 4.6/SampleProvider/Win32/Provider/Main/SimbaProvider VS2015.sln* solution.

Note: in the evaluation versions of the kit, the solution file is *SimbaProvider_Eval VS2015.sln*.

4. Select the *Debug* configuration and platform (Win32 / x64) from the toolbar or by selecting *Build | Configuration Manager*.
5. Select *Build | Build Solution*.

The following files are generated in *<Install Path>/SimbaProvider for OLAP SDK 4.6/SampleProvider/Win32/ Bin/Win32/<configuration>/*:

- **SimbaProvider.dll:** the Win32 COM DLL containing the ODBO provider that can be used with ODBO applications like Excel.
- **SimbaProviderEr.dll:** a Win32 COM DLL containing the OLE DB error lookup utilities as required by the ODBO specification (specifically this implements the OLD DB IErrorLookup interface).

Building the XMLA Sample Providers

Building the .NET XMLA Sample Provider

To build the .NET XMLA sample provider:

1. Start Microsoft Visual Studio 2015.
2. Select *File | Open Solution*.
3. Open the following *<Install Path>\SimbaProvider for OLAP SDK 4.6\SampleProvider\dotNET\XmlaProvider VS2015.sln* solution.

Note: in the evaluation versions of the kit, the solution file is *XmlaProvider_Eval VS2015.sln*.

4. Select the *Debug* configuration and 64-bit platform from the toolbar or by selecting *Build | Configuration Manager* from the menu bar.
5. Build the solution (Select *Build | Build Solution* from the menu bar).

The following assemblies are generated in *<Install Path>/SimbaProvider for OLAP SDK 4.6/SampleProvider/ dotNET\XMLA VS2015\bin*:

- **Customer.ManagedIso.dll**: the assembly containing the XMLA provider implementation.
- **Customer.XmlaWebServiceEval.dll**: the ASP.NET WebService dll that exposes the provider as a web service for use in IIS.
- **Simba.Olap.Common.dll**: a .NET assembly provided by Simba.
- **Simba.Olap.dll**: a .NET assembly provided by Simba.
- **Simba.Xmla.dll**: a .NET assembly provided by Simba.

Building the Java XMLA Sample Provider for Windows

The Java Sample Provider uses an ISO implementation written in C++ that bridges to Java and included as part of the sample Java XMLA provider. This C++ implementation must first be built using Visual Studio to generate a .DLL file.

All of the necessary JAR files are already built, and are located in the *<Install Path>/SimbaProvider for OLAP SDK 4.6/Simba/Java/jars* directory. An Ant script is provided to build a WAR file and deploy to Jboss or Tomcat. After the C++ implementation has been built, the Java XMLA Sample Provider is built using Ant or Eclipse as a web service that links to the C++ .DLL file.

The following subsections describe how to build these two projects.

To build the C++ ISO implementation:

1. Start Microsoft Visual Studio 2015.
2. Select *File | Open Solution*.
3. Open one of the following solutions depending on which version of the Windows Base Kit you have installed:
 - *<Install Path>/SimbaProvider for OLAP SDK 4.6/SampleProvider/Win32/CustomerJavaXMLA VS2015.sln*
 - *<Install Path>/SimbaProvider for OLAP SDK Eval 4.6/SampleProvider/Win32/CustomerJavaXMLA_Eval VS2015.sln*
4. Select the *Debug* or *Release* configuration and 64-bit platform from the toolbar or by selecting *Build | Configuration Manager* from the menu bar.
5. Build the solution (Select *Build | Build Solution* from the menu bar).

To build the Java XMLA Sample Provider:

1. Start Eclipse.

2. If you are not prompted for a workspace, select *File | Switch Workspace...*
3. Enter *<Install Path>/SimbaProvider for OLAP SDK 4.6/SampleProvider/Java/Eclipse Workspace* as the workspace and click *OK*.
4. Select *Window | Show View | Package Explorer* if the *Package Explorer* pane is not visible. Right click on the *Package Explorer* and select *Import...*
5. In the *Import* dialog select *Existing Projects into Workspace* and click *Next >*.
6. Enter *<Install Path>/SimbaProvider for OLAP SDK 4.6/SampleProvider/Java/Eclipse Workspace* as the root directory.
7. Ensure that the *CustomerXMLA* project is checked.
8. Click *Finish*. At this point, disregard any errors you may have in the *Problems* pane.
9. Perform the following steps to ensure that you have installed the required JRE:
 - a. Right click on the *CustomerXMLA* project.
 - b. Select *Properties*.
 - c. Select *Java Build Path* and click on the *Libraries* tab.
 - d. The version of the JRE system library should be indicated as Java 8. If not, do the following to change the JRE:
 - i. Select the *JRE System Library* entry and click the *Edit...* button.
 - ii. Select the *Alternate JRE* radio button and click the *Installed JREs...* button.
 - iii. Click the *Add...* button on the *Installed JREs* window.
 - iv. Click the *Browse...* button on the new window.
 - v. Navigate to the directory where your JRE is installed and click *OK*.
 - vi. Select the JRE that you want to add on the *Installed JREs* window and click *OK*.
 - vii. Select *Alternate JRE* on the *Edit Library* window and click *Finish*.
 - viii. Click *OK* to close to *Properties* window.

To deploy the Java XMLA Sample Provider web service:

1. Select *Show View | Ant* from the *Window* menu. This will bring up the *Ant View*.
2. Press the *Add Buildfiles* button in the *Ant View*,
3. Select the *build.xml* file in the *Buildfile Selection* dialog, under the *CustomerXMLA* folder. Hit *Ok*.
4. Select one of the following targets in the *Ant view* and press *Run the Selected Target* to package your web application into a war file and deploy your provider:
 - **deploy_release**: this will package a release version of your web application into a war file and deploy your war file to the JBOSS server and copy it to jboss home.

- **deploy_tomcat_release**: this will package a release version of your web application into a war file and deploy your war file to Tomcat home.
- **deploy_debug**: this will package a debug version of your web application into a war file and deploy your war file to the JBOSS server and copy it to jboss home.
- **deploy_tomcat_debug**: this will package a debug version of your web application into a war file and deploy your war file to Tomcat home.

Building the Java XMLA Sample Provider for Linux

This section provides the steps for building the Sample Java Provider on the Linux command line.

1. Navigate to `~/SimbaProvider-4.6.x.x/SampleProvider/Linux>`
2. Enter one of the following command lines to build the respective target:
 - Release:
> `make TARGET=release BITS=x64`
 - Debug:
> `make TARGET=debug BITS=x64`
3. Navigate to `~/SimbaProvider-4.6.x.x/SampleProvider/Java>`
4. Follow the steps from the previous section for using Eclipse to deploy the WAR file, or enter one of the following command lines to deploy the chosen target to either JBoss or Tomcat:
 - Release jboss:
> `ant deploy_release`
 - Debug jboss:
> `ant deploy_debug`
 - Release tomcat:
> `ant deploy_tomcat_release`
 - Debug tomcat:
> `ant deploy_tomcat_debug`

Testing the Sample Providers

The following sub sections describe how to test and debug both the XMLA and ODBO sample providers for Windows that were built in the previous sections.

Note: Testing and debugging for Linux is performed in a similar manner to the process described in *Testing and Debugging the Java XMLA Provider for Windows* on page 19.

The following subsections describe how to perform testing using Microsoft Excel as the consumer application. Excel can work directly with the ODBO sample provider, while SimbaO2X is required for Excel to work with an XMLA (web service) provider.

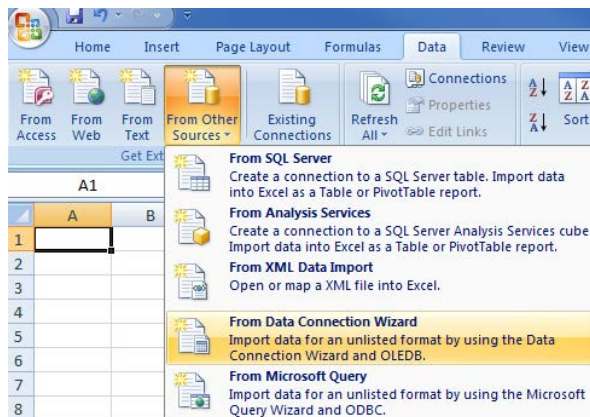
Testing and Debugging the ODBO Provider

To attach Visual Studio to Excel, set a breakpoint in the ODBO sample provider, and verify that you can access data:

1. Run Microsoft Excel.
2. Run Visual Studio and open the `<Install Path>/SimbaProvider for OLAP SDK 4.6/SampleProvider/Win32/Provider/Main/SimbaProvider VS2015.sln` solution.

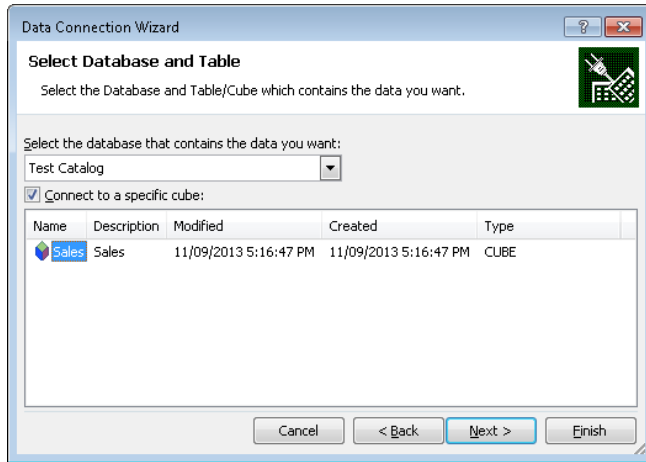
Note: In the evaluation version of the kit, the solution file is `SimbaProvider_Eval VS2015.sln`.

3. Place a breakpoint in `DataStoreText::ConnectToServer()` (located in `DataStoreText.cpp` in the `TextOlap` project).
4. Switch to Excel, navigate to the *Data* tab, locate the *From Other Sources* dropdown, and select *From Data Connection Wizard*:

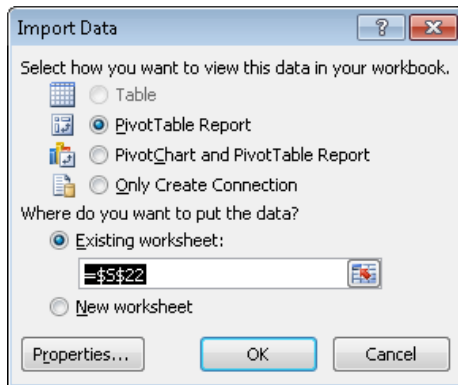


5. Select *Other/Advanced* in the wizard and click *Next*.
6. Select *Simba OLE DB for OLAP Provider* in the *Data Link Properties* dialog and click *Next*.
7. Ignore the credentials, and click *Test Connection*. This should trigger the breakpoint set in `DataStoreText::ConnectToServer()`. Remove the breakpoint and continue execution. A dialog should be displayed indicating a successful connection. Click *OK* to continue.

8. Select *Test Catalog* from the dropdown below the *Test Connection* button and click OK. The *Data Connection Wizard* will be displayed listing the *Sales Cube*:



9. Click *Finish* to close the dialog. A confirmation dialog will appear asking you to replace an existing .odc file. Click *Yes* to continue.
10. Click *OK* when the *Import Data* dialog appears:



11. Select the fields for which to display data in the *PivotTable Field List* that appears on the right hand side of Excel. A pivot table containing the data will appear in Excel's grid.

Testing and Debugging the .NET XMLA Provider

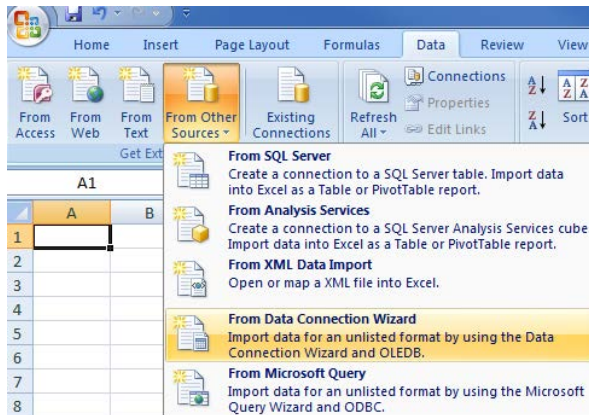
Excel is only capable of communicating with an ODBO provider. The *SimbaO2X* utility establishes a bridge for connecting Excel to an XMLA-based sample provider.

To run and debug the .NET XMLA sample provider:

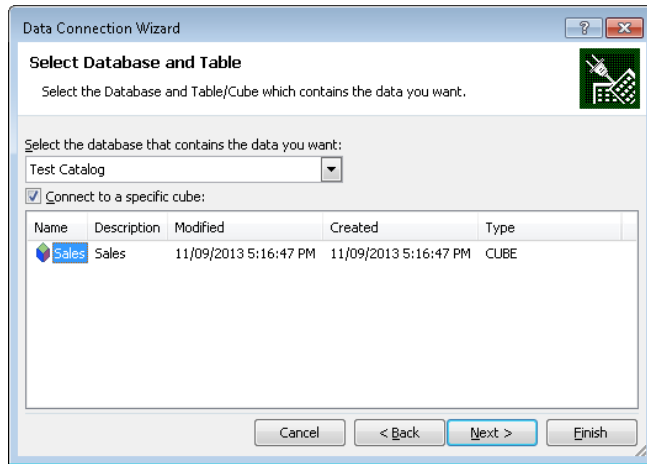
1. Run IIS and ensure that your XMLA .NET web service has been installed and is running on IIS.
2. Run Microsoft Excel.
3. Run Visual Studio and open the `<Install Path>/SimbaProvider for OLAP SDK 4.6/Sample Provider/dotNET/XmlaProvider VS2015.sln` solution.

Note: in the evaluation version of the kit, the solution file is *XmlaProvider_Eval VS2015.sln*.

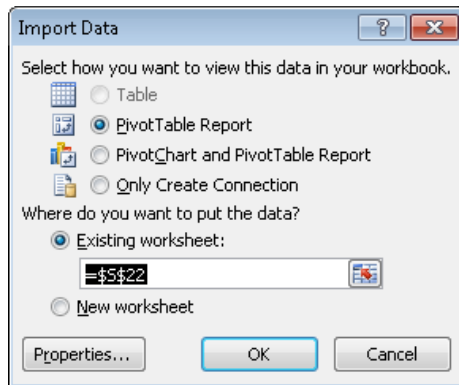
4. Place a breakpoint in *DataStoreText::ConnectToServer()* (located in *DataStoreText.cpp* in the *TextOlap* project).
5. Navigate to *Debug|Attach to Process*.
6. Enable the *Show processes in all sessions* checkbox below the list of processes.
7. Select *w3wp.exe* in the *Available Processes* list and click *Attach*.
8. Switch to Excel, navigate to the *Data* tab, locate the *From Other Sources* dropdown, and select *From Data Connection Wizard*;



9. Select *Other/Advanced* in the wizard and click *Next*.
10. Select *SimbaO2X* in the *Data Link Properties* dialog and click *Next*.
11. Specify the URL of the XMLA host and web service in the *Specify the XMLA connection URL* (for example, a web service being run on the local PC would have a URL similar to *http://localhost/customerxmlwebservice/XmlaWebService.asmx* or *http://localhost/customerxmlwebservice/*).
12. Select *Test Catalog* from the dropdown below the *Test Connection* button. This should hit the breakpoint set earlier. Remove the breakpoint and continue execution.
13. Click *OK* on the *Data Link Properties* dialog in Excel. The *Data Connection Wizard* will be displayed listing the *Sale Cube*:



14. Click *Finish* to close the dialog. A confirmation dialog will appear asking you to replace an existing .odc file. Click *Yes* to continue.
15. Click *OK* when the *Import Data* dialog appears:



16. Select the fields for which to display data for in the *PivotTable Field List* that appears on the right hand side of Excel. A pivot table containing the data will appear in Excel's grid.

Testing and Debugging the Java XMLA Provider for Windows

Excel is only capable of communicating with an ODBO provider. The *SimbaO2X* utility establishes a bridge for connecting Excel to an XMLA-based sample provider.

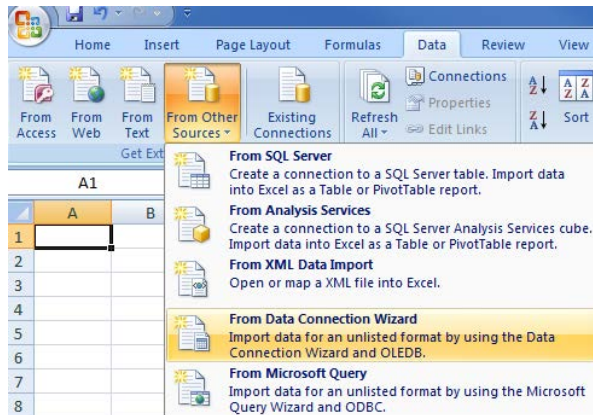
Note: The following steps are similar to the ones described in "Testing and Debugging the .NET XMLA Provider" on page 17.

To run and debug the Java XMLA sample provider web service in JBoss:

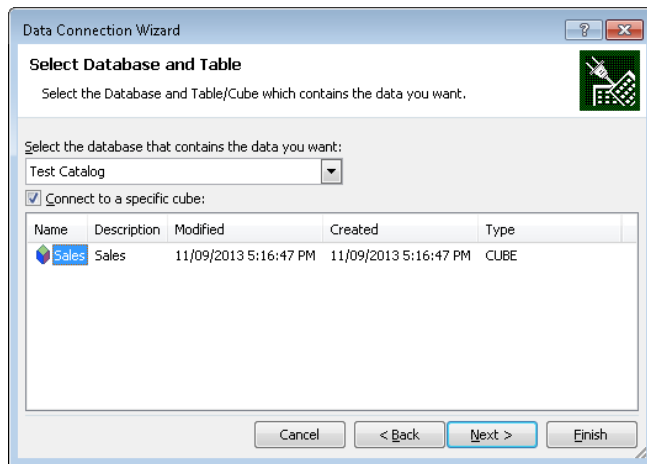
1. Run JBoss by navigating to *jboss/bin* and double clicking on *standalone.bat*.
2. Run Microsoft Excel.
3. Run Visual Studio and open the *<Install Path>/SimbaProvider for OLAP SDK 4.6/Sample Provider/Win32/CustomJavaXMLA VS2015.sln* solution.

Note: in the evaluation version of the kit, the solution file is *CustomerJavaXMLA_Eval VS2015.sln*.

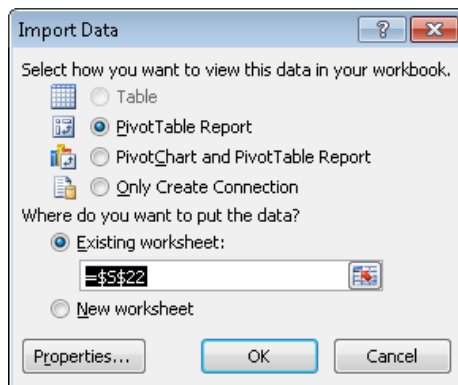
4. Place a breakpoint in *DataStoreText::ConnectToServer()* (located in *DataStoreText.cpp* in the *TextOlap* project).
5. Navigate to *Debug|Attach to Process*.
6. Enable the *Show processes in all sessions* checkbox below the list of processes.
7. Select *java.exe* in the *Available Processes* list and click *Attach*.
8. Switch to Excel, navigate to the *Data* tab, locate the *From Other Sources* dropdown, and select *From Data Connection Wizard*:



9. Select *Other/Advanced* in the wizard and click *Next*.
10. Select *SimbaO2X* in the *Data Link Properties* dialog and click *Next*.
11. Specify the URL of the XMLA host and web service in the *Specify the XMLA connection URL* (for example, a web service being run on the local PC would have a URL similar to: <http://localhost:8080/XmlaWebService/>).
12. Select *Test Catalog* from the dropdown below the *Test Connection* button. This should hit the breakpoint that was set earlier. Remove the breakpoint and continue execution.
13. Click *OK* on the *Data Link Properties* dialog in Excel. The *Data Connection Wizard* will appear, listing the *Sale Cube*:



14. Click *Finish* to close the dialog. A confirmation dialog will appear asking you to replace an existing .odc file. Click *Yes* to continue.
15. Click *OK* when the *Import Data* dialog appears:



16. Select the fields for which to display data for in the *PivotTable Field List* that appears on the right hand side of Excel. A pivot table containing the data will appear in Excel's grid.

Resources

How to Contact Us

If you have difficulty using the SimbaProvider for OLAP SDK™, contact our Technical Support staff. We welcome your questions, comments, and feature requests.

Technical Support is available Monday to Friday from 8 a.m. to 6 p.m. Pacific Time.

Please have a detailed summary of the environment (operating system, version, patch level, Visual Studio patch-level etc.) for your machine ready before you call or write us. Telling us this information will help us help you more effectively.

You can contact Technical Support via:

- Email: support@simba.com
- Website: www.simba.com
- Telephone: (604) 633-0008 ext. 3
- Fax: (604) 633-0004

You can also follow us on Twitter @SimbaTech

Getting Additional Information on XMLA and OLE DB

The XML for Analysis 1.1 specification is available for download at <http://news.xmlforanalysis.com/docs/xmla1.1.doc>.

For complete information on the OLE DB 2.7 specification, see the OLE DB Programmer's Reference (Microsoft). You can view this at the Microsoft Developer Network web site at <http://msdn.microsoft.com/library>.