



SimbaEngine SDK Technical White Paper

www.simba.com

Contents

Overview	1
SimbaEngine SDK Highlights	2
SimbaEngine SDK Solutions Architecture	2
Simba SQL Engine Internal Architecture	5
Specifications	7
Access Methods	7
How To Build a Driver/Provider....	7

This white paper provides a technical overview of SimbaEngine SDK – the most complete ODBC, JDBC, OLE DB Driver, or ADO.NET Data Provider development solution available.

Overview

SimbaEngine SDK is a software development kit (SDK) that enables you to build an ODBC, JDBC, or OLE DB Driver, or an ADO.NET Data Provider for data access to SQL and non-SQL data sources. SimbaEngine SDK is a set of components and tools for building a high-performance data connectivity solution that allows knowledge workers to access, use, distribute and act on any enterprise data using today's most popular front-end reporting and analysis tools.

Unlike other toolkits on the market, SimbaEngine SDK provides an open framework from which you can quickly develop a quality ODBC, JDBC, OLE DB Driver, or ADO.NET Data Provider. SimbaEngine SDK is the best choice, if performance, extensibility and compatibility are important to you. SimbaEngine SDK's SQL engine provides your non-SQL-enabled data store with full SQL capabilities. All you have to do is implement a few basic functions to open the whole world of open connectivity tools to your customers. In addition, SimbaODBC functionality provides your SQL-enabled data store with the gateway to the same connectivity tools.

Simba Technologies Inc.
938 West 8th Avenue
Vancouver, BC Canada
V5Z 1E5
Tel. +1 604 633 0008
Fax. +1 604 633 0004
solutions@simba.com

SimbaEngine SDK's architecture is proven and robust. More than 50 million ODBC Drivers in the marketplace today were built using Simba's ODBC technology.



SimbaEngine SDK Highlights

Here are just a few of the benefits of SimbaEngine SDK:

Extensibility – Build a simple Driver and then add features or advanced functionality whenever you want. You can extend your SimbaEngine SDK ODBC Driver solution after your initial development effort. For example, you can add JDBC or ADO.NET access, as well as advanced functionality, like indexes and pass-down filters and aggregations, or client/server functionality.

Performance – Our robust SQL engine and advanced functionality result in superior performance. A SimbaEngine SDK-built Data Driver or ADO.NET Data Provider excels. The combination of our robust SQL engine and advanced functionality, such as Collaborative Query Execution and other optimization, dramatically reduces the time it takes for data to arrive at a user's workstation for a given query.

Compatibility – Simba takes care of your Data Driver, or ADO.NET Data Provider's compatibility with the latest platforms, data access standards and desktop reporting and analysis tools. If you build your ODBC, JDBC or OLE DB Driver, or your ADO.NET Data Provider with SimbaEngine SDK, you do not have to worry about keeping up with ODBC, JDBC, or OLE DB conformance standards, platform changes, or new desktop tools. At Simba, we ensure the utmost compatibility of your data connectivity solution by issuing timely, regular upgrades and major releases.

32- and 64-bit Support – SimbaEngine SDK natively supports both 32-bit and 64-bit platforms. In addition, 32-bit and 64-bit platforms will interoperate across the client/server connection.

Template Development Approach – Get a prototype Data Driver, or ADO.NET Data Provider up and running quickly with our easy-to-use Quickstart Template. With SimbaEngine SDK, you can get a prototype Driver, or ADO.NET Data Provider

up and running within five days. You achieve this with our Quickstart Template, which contains nearly all of the code needed for a prototype. Alternatively, you may use the more sophisticated Codebase Example Driver included in the package. Once you know your Data Driver, or ADO.NET Data Provider is returning data, and you understand how your Data Driver or ADO.NET Data Provider works, you can prepare it for shipping, or set up the client/server environment according to our step-by-step instructions.

Comprehensive Documentation – Receive a full suite of documentation, including a developer's guide, technical reference materials and online help. SimbaEngine SDK comes with extensive documentation that ranges from tutorial-style instructions to full reference information. We also provide online help and PDF format documents for viewing online.

SimbaEngine SDK Solutions Architecture

SimbaEngine SDK is supported and tested on Windows, Linux and UNIX platforms, and is carefully designed with object oriented techniques.

SimbaEngine SDK contains a variety of components that can be mixed and matched to deliver the exact data access solution your customers need. You can use the SimbaODBC component alone to create a highly compatible ODBC data driver for a SQL-enabled data store, or you can use the Simba SQL Engine to create a high-performance ODBC data driver for a non-SQL data store. In both cases, you can use the Simba Client/Server component to create a remote access solution for your users. The following sections describe these options.

SimbaODBC Data Driver

A SimbaODBC data driver is the simplest application of the SimbaEngine SDK. The SimbaODBC component presents an ODBC 3.52 API to the user's application via the ODBC Driver Manager. All you develop is a Data Store Interface (DSI) API



Over half of the ODBC Drivers available today were built with Simba's technology.

implementation that connects the SimbaODBC components to your SQL-enabled data store. (see Figure 1)

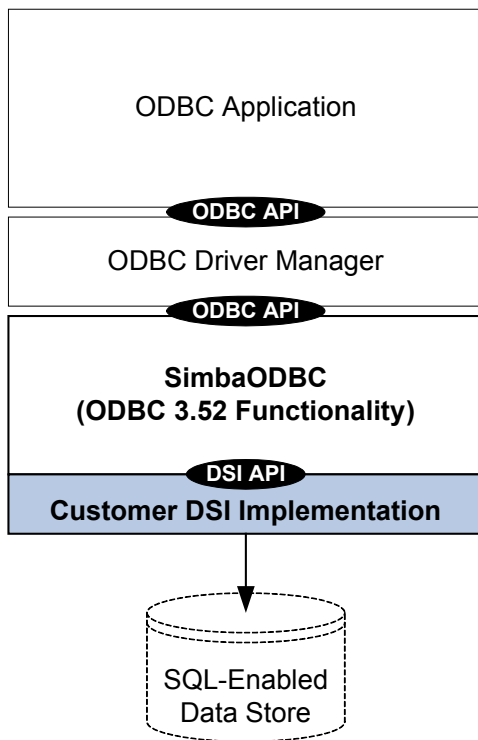


Figure 1. A simple ODBC driver created with the SimbaODBC component and your DSI implementation

When the user's application makes ODBC function calls into the ODBC interface of your new data driver, SimbaODBC makes DSI calls into your DSI implementation, which calls into your data store. The results are passed back to the user's application. Your DSI implementation is responsible for connecting to the data store, providing metadata, executing SQL statements and returned the results.

Simba SQL Engine Data Driver

If your data store cannot process SQL, then you can use the Simba SQL Engine component to create a data driver for your data store. You combine the same SimbaODBC component plus Simba SQL Engine to create an ODBC 3.52 data driver that can access your non-SQL, non-relational data store. Again, all you develop is a Data Store Interface (DSI) implementation that connects Simba SQL Engine to your non-SQL data store. (see Figure 2)

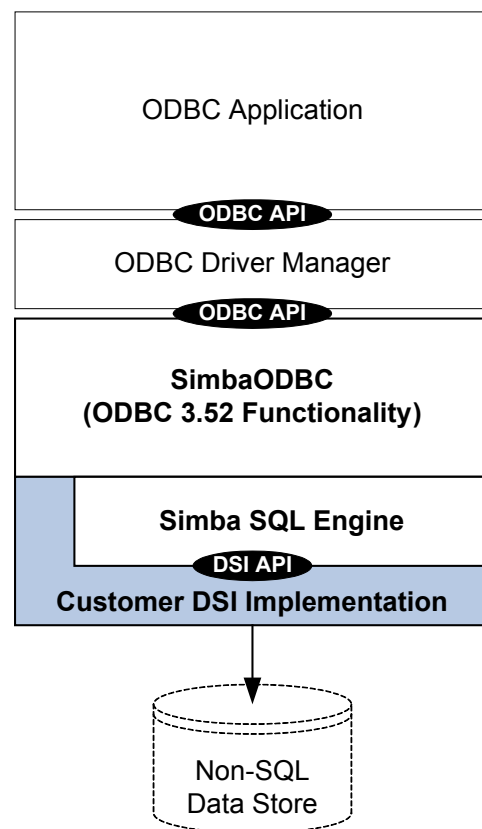


Figure 2. An ODBC driver created with both the SimbaODBC component and Simba SQL Engine along with your DSI implementation

Simba SQL Engine is a high-performance, optimizing SQL parser and execution engine that works in concert with SimbaODBC through the DSI API. When the user's application



makes ODBC function calls into the ODBC interface of your new data driver, SimbaODBC makes DSI calls into your DSI implementation and into Simba SQL Engine. The results are passed back to the user's application.

Your DSI implementation is responsible for connecting to the data store, providing metadata, opening data tables and fetching row and column data from the data tables. If your data store is not made up of tables and rows and columns, then your DSI is also responsible for providing a view of your data store that allows the Simba SQL Engine to retrieve data by asking for it as rows and columns. This can sometimes seem like a difficult transformation, but Simba's experience covers most data store architectures, and we can help with this important step.

Simba Client/Server

If your data store is on a powerful server that you want to use for SQL processing, or if you want to realize the performance benefits of having SQL processing close to the data store, you can use Simba Client/Server to move Simba SQL Engine to the server where end users can access it remotely. Simba Client/Server makes this easy. (see Figure 3)

SimbaServer is a high-performance server component tuned to deliver high-performance query results and use system resources sparingly. It communicates via the Simba Message Protocol – a generic data access wire protocol used by SimbaClient for ODBC and SimbaClient for JDBC. Both clients can remotely access the same SimbaServer.

SimbaServer links to the same DSI implementation that you created for the Simba SQL Engine data driver so you can develop in a local environment and deploy your DSI implementation remotely. A DSI implementation developed for a SimbaODBC data driver can be deployed remotely in the same way.

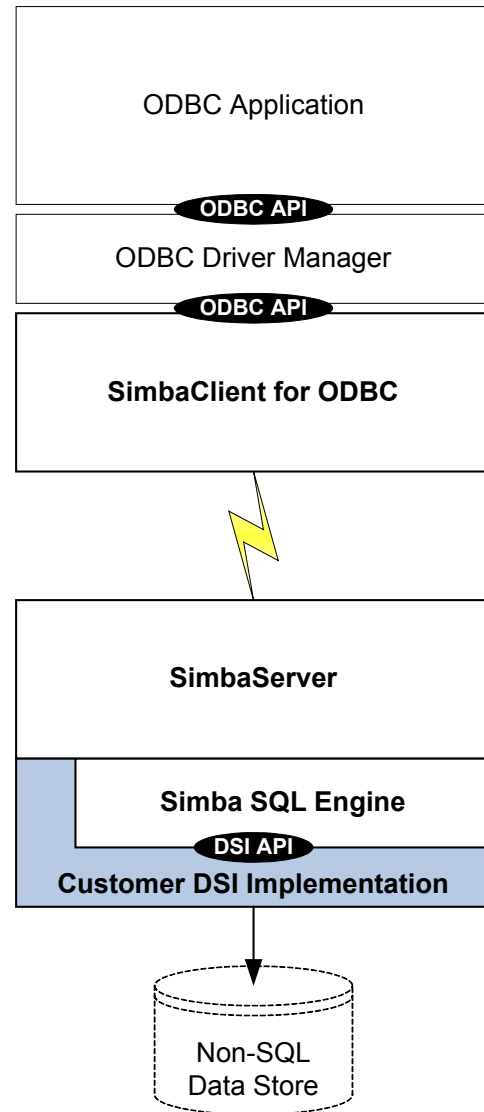


Figure 3. Simba Client/Server can provide remote access to Simba SQL Engine



Simba SQL Engine Internal Architecture

Simba SQL Engine is a high-performance SQL parsing and execution engine designed to connect to non-SQL, non-relational data stores. While the details of its implementation are complex, its design is remarkably simple. Two major components prepare and execute the SQL passed from the user's application. Both components call into your DSI implementation through the DSI API. They communicate via the Execution Tree – a structure created from the SQL statement that is designed to be optimized and easy to execute. (see Figure 4)

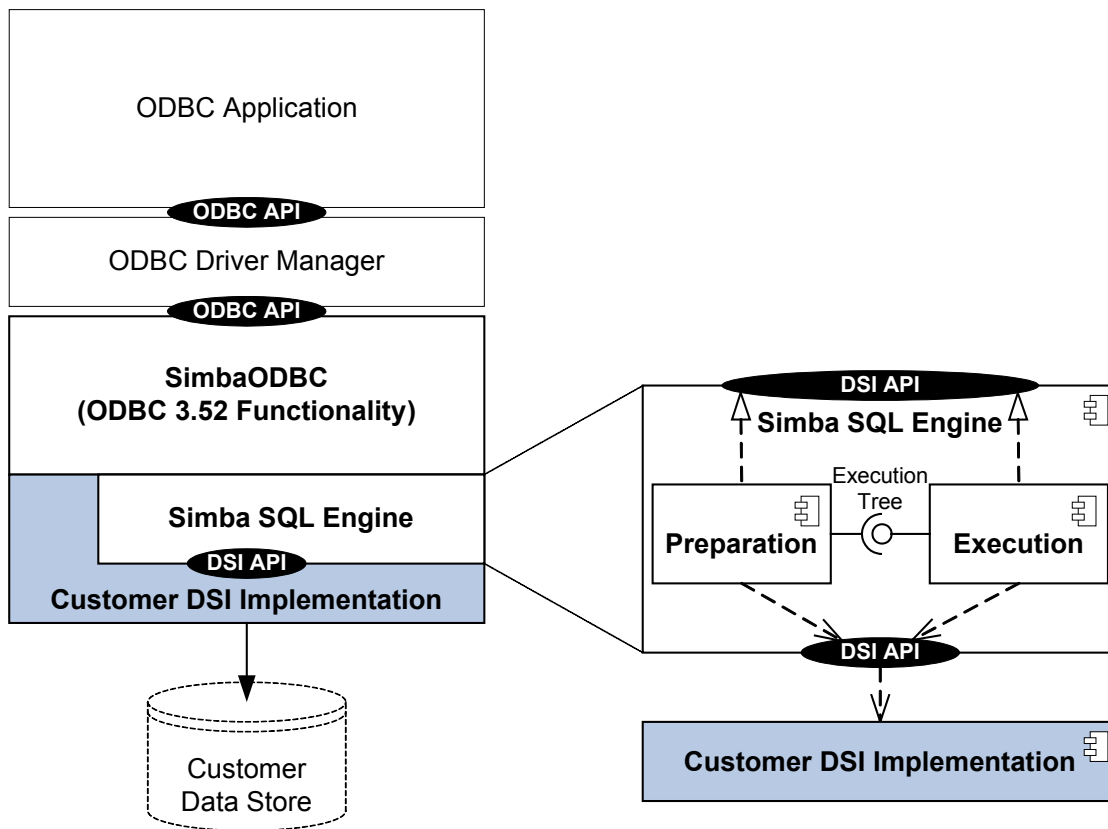


Figure 4. The two major components of Simba SQL Engine shown communicating via the Execution Tree

Preparation

In the Preparation component, the Parser decomposes the SQL statement into a parse tree that contains all of the SQL semantics. AEBUILDER transforms the parse tree, in a series of steps, into an Algebraic Expression Tree, or AE-Tree. This representation of the SQL statement is logically equivalent to the SQL statement, but much easier to analyze and transform. The optimizer works on the AE-Tree to optimize it for execution. The DSI implementation is also allowed to change the AE-Tree if it can execute parts of the AE-Tree faster than Simba SQL Engine. This is often the case for column-based and in-memory data stores. The ETGenerator finally transforms the optimized AE-Tree into an Execution Tree that is passed to the Execution engine. (see Figure 5)

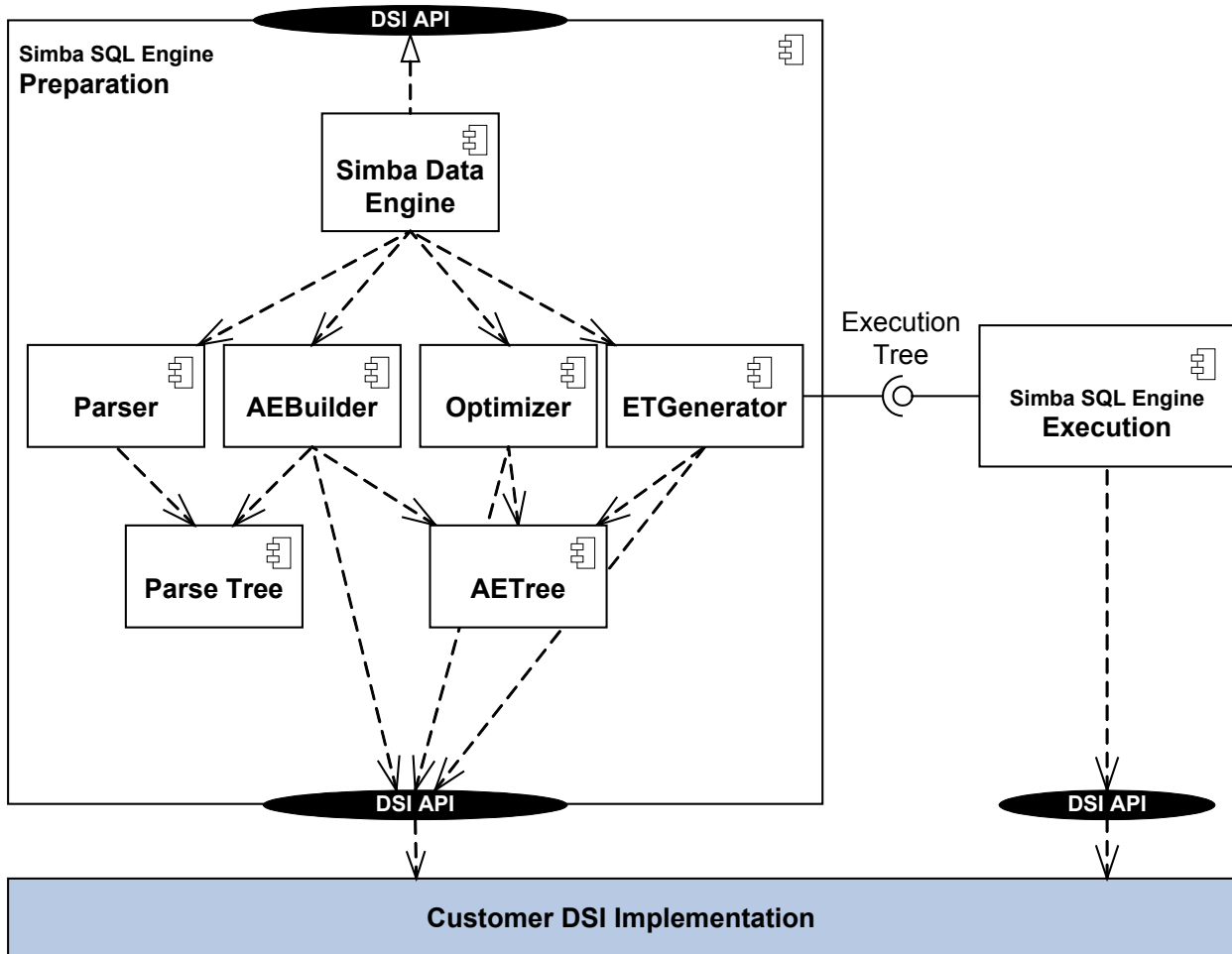


Figure 5. The Preparation component showing its major sub-components

Execution

The Execution engine is responsible for executing the Execution Tree by walking through it and making calls into your DSI implementation to retrieve row and column data. It filters the results of these calls, joins them, sorts them, and finally returns the result set back to the user's application.

If the DSI implementation has previously altered the AE-Tree so that it is executing parts of the query, this has been built into the Execution Tree and the Execution engine simply requests the results of the operation from the DSI implementation. This is Collaborative Query Execution.

Collaborative Query Processing

The Execution engine is capable of completely executing any SQL statement passed to it. However, if your Data Store is able to execute parts of the SQL statement – for example table filters, joins or aggregations – then the AE-Tree can be modified and responsibility for those functions passed to the DSI implementation. The Execution engine will request only the results of those passed-down functions from the DSI implementation, and will combine them seamlessly with the results of its own function execution. If your data store can deliver high-performance execution of specific functions, the result is high performance combined with broad compatibility.



Specifications

SimbaEngine SDK supports the following platforms and tools:

Stand-alone Data Driver and Server Platforms – All SimbaEngine SDK components support the 32- and 64-bit versions of Windows, UNIX and Linux.

Client Platforms – SimbaClient for ODBC supports the 32- and 64-bit versions of Windows, UNIX and Linux. SimbaClient for JDBC supports all client operating systems running a Java Virtual Machine (JVM).

Development Software – Windows: Microsoft Visual Studio.NET 2008, Microsoft Visual Studio.NET 2005. **UNIX:** For most platforms, the native compiler is supported. **Linux:** GNU Compiler environment. ODBC.NET Provider required for ADO.NET functionality. OLE DB / ODBC Bridge required for OLE DB functionality.

Desktop Tool Support – ODBC capable applications, including Microsoft Office and SAP Business Objects Crystal Reports.

Access Methods

The following access methods can be used with SimbaEngine SDK:

ODBC: Drivers written with SimbaEngine SDK can be accessed via the well established ODBC interface. Programs, such as Microsoft Office or SAP Business Objects Crystal Reports and others, that support ODBC will be able to access data stores whose data is exposed via an ODBC driver.

JDBC: SimbaClient for JDBC is a type 3 JDBC client that connects to SimbaServer.

OLE DB: The OLE DB interface is supported via the OLE DB to ODBC bridge. This provider can be obtained from Microsoft.

ADO.NET: ODBC drivers, developed using SimbaEngine, can be accessed via ADO.NET. The application will need to enable the inherent cursor library support from the ODBC driver manager for maximum compatibility.

How to Develop an ODBC, JDBC or OLE DB Driver, or an ADO.NET Data Provider

The following procedure describes how to build a read-write ODBC, JDBC or OLE DB Driver, or an ADO.NET Data Provider for Windows. With SimbaEngine SDK, you will use the following approach, depending on your objectives.

1. Build an ODBC Driver by creating a DSI implementation for your data store.
2. Should you wish to use a JDBC or OLE DB Driver, or an ADO.NET Data Provider, simply re-link your DSI implementation with SimbaServer.

ODBC Driver

To create your ODBC data driver, modify the SimbaEngine Quickstart example driver according to the requirements of your data store. You can build a prototype read-only ODBC Driver within five days by following the steps in "Getting Started with SimbaEngine SDK". Then you can continue to prepare your data driver for distribution, and, if you want, add advanced functionality.

Set-up a Networked Environment

To set up a client/server environment:

1. Develop and test your DSI implementation for your data store. Configure your ODBC Driver as a server-side data source.
2. Re-link your DSI implementation with SimbaServer.
3. Install and configure your new SimbaServer executable.
4. Install and configure SimbaClient for ODBC or SimbaClient for JDBC on your client machines.
5. Test and confirm that your Client/Server installation has the same capabilities as your stand-alone data driver.

Prepare Your Driver or ADO.NET Data Provider for Distribution

To prepare your data driver, or ADO.NET Data Provider for commercial distribution:



1. Customize your data driver to suit your data store by adding additional data types, controlling string comparisons and customizing error messages.
2. Add advanced functionality, such as bookmarks and pass down filters and aggregations. Improve your Driver's performance by adding support for indexes and making the code more efficient for your specific data store.
3. Improve your data driver's performance by making the code more efficient for your specific data store.
4. Make your Driver "ship quality" by ensuring that it is thread safe, locking your data in multi-user environments, building a configuration DLL and creating an installer.

JDBC Driver

Once you have built your ODBC Driver, you can create a JDBC Driver by employing our JDBC Client for JDBC with SimbaServer.

OLE DB Driver

Once you have built your ODBC driver, you can create an OLE DB Driver by employing an OLE DB / ODBC Bridge.

ADO.NET Data Provider

Once you have built your ODBC driver, you can create an ADO.NET Data Provider by employing an ODBC.NET Data Provider from Microsoft.

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, OLE DB for OLAP (ODBO) and XML for Analysis (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's firm commitment to delivering the highest customer value through innovative solutions and expert support has gained the company a reputation as the industry leader for data connectivity solutions.

Your competitive **advantage**
FOR DATA CONNECTIVITY SOLUTIONS

©2009 Simba Technologies Inc. All Rights Reserved.
Simba and the Simba logo are trademarks of Simba Technologies Inc.
All other trademarks or service marks are the property of their respective owners. Printed in Canada.