



SimbaEngine ODBC SDK SCADA Technical White Paper

By Simba Technologies Inc.

This white paper provides an overview of solutions to the challenges faced when accessing data from real-time data sources and historical archives. These types of data stores are common in industries like manufacturing, finance, government, energy and many others. For this paper, we will use data created using a SCADA application as an example.

Overview

Organizations making use of computers for automated production often incorporate SCADA systems into their manufacturing processes. SCADA is an acronym for Computer-based Supervisory Control and Data Acquisition. A SCADA system might be used in a manufacturing or processing facility to record and control the process. SCADA data consists of automatic measurements taken many times at many points in the process, combined with the time of the measurement and the identity of the measurement point. The measurement time is a timestamp and the measurement point identity is called a tag. The basic goals of a SCADA system is to display, automate, supervise, monitor and control the process.

Problem

As well as automated monitoring of a real-time process, data storage is a critical component of a SCADA system. For performance reasons, the information generated by systems of this type is often archived in non-standard data formats that provide the required performance. The stored data consists of measurement, or values, from specific endpoints or tags. The design emphasis for data storage is on speed and data compression. This often means a SCADA system is based upon requirements that do not include the generation of custom reports and ad hoc analysis.

Most SCADA Independent Software Vendors (ISVs) have excellent reporting and analysis tools, which work well for most of their users. However, there can be a problem when users want to use standards-based tools, such as Microsoft Excel and Access, SAP Business Objects Crystal Reports and

IBM Cognos applications, to access the data contained in the SCADA application, or in the real-time or historical data stores. When giving these reporting tools access to a SCADA data source, using industry standard data access makes sense.

The tools provided with a SCADA system will extract data from the database, but SCADA data is often arranged in the database in a way that makes it hard to analyze directly. In a SCADA database, the data looks something like Table 1, below.

TIME	TAG	VALUE
12:01:01	T101	4.5
12:01:01	T102	6.8
12:01:01	T103	3.1
12:01:02	T101	4.6
12:01:02	T102	6.7
12:01:02	T103	3.0
etc.		

Table 1: Data stored in the native format in a SCADA data store.

However, for a time series analysis, the data needs to look more like Table 2, below.

TIME	T101	T102	T103
12:01:01	4.5	6.8	3.1
12:01:02	4.6	6.7	3.0
Etc.			

Table 2: The way data should be stored for easy time-series analyses.

Data in the form in Table 2 is easily used by common business tools like Microsoft Excel to create graphs and perform analyses. This kind of transformation can be programmed into



the vendor's reporting tools, but an easier and more general way to solve the problem is to add SQL reporting directly from the SCADA archive database using SimbaEngine ODBC SDK.

With SimbaEngine ODBC SDK, the data in the SCADA database can be transformed from the native format in Table 1 to the form in Table 2 with a simple self-join SQL query. In addition, the SQL query can optionally filter the data to return only a time period of interest. The following SQL statement will extract the required data:

```
SELECT T1.TIME, T1.VALUE AS T101, T2.VALUE AS T102,
T3.VALUE AS T103
FROM
ARCHIVE T1, ARCHIVE T2, ARCHIVE T3
WHERE
T1.TIME Between '12:00:00' And '12:02:59'
And T2.TIME Between '12:00:00' And '12:02:59'
And T3.TIME Between '12:00:00' And '12:02:59'
And T1.TAG='T101'
And T2.TAG='T102'
And T2.TAG='T103'
And T1.TIME = T2.TIME
And T1.TIME = T3.TIME
ORDER BY T1.TIME
```

We are assuming that the data is stored in a table called 'ARCHIVE' with the format shown in Table 1. This SQL query is a self-join of the ARCHIVE table on the TIME column. It selects the VALUES for the desired TAGs and filters them by TIME. Normally, self-joins can be expensive. However, with the filter on T1 in the WHERE clause and the additional filters placed on the tables T2 and T3, Simba SQL Engine can pass down these filters to the SCADA system. If the SCADA system can filter efficiently on the TIME column, then the query performance is increased.

SimbaEngine ODBC SDK with SCADA

Using SimbaEngine ODBC SDK will enable you to write an ODBC driver for accessing your SCADA data source within a few weeks. SimbaEngine ODBC SDK already contains 99% of the code needed to develop your driver. All you have to do is connect Simba SQL Engine to your data source API, following our documented examples.

The Data Store Interface (DSI) API is how Simba SQL Engine communicates with your SCADA data store. You write the code to implement the DSI that links the DSI API to your data store and Simba SQL Engine will do the rest. The DSI implementation is the link that provides report writer applications with access to your SCADA data source, or any other non-SQL data source.

The Data Store Interface (DSI) Implementation

The DataStoreInterface (DSI) implementation is the component of the SimbaEngine ODBC driver that interacts directly with your data store. It is the unique bridge between the Simba SQL Engine and your data store. The DSI implementation is a comprehensive set of functions that can be implemented iteratively, providing basic access or optimized access that leverages the capabilities of the underlying data store. Examples of the types of functions are:

- Connect to and Disconnect from the data store;
- Deliver custom connection information from the reporting application;
- Open and Retrieve information about tables;
- Execute SQL queries and retrieve the results.

The DSI API is very powerful and flexible, and it's capable of accessing any SCADA data source.

Included with SimbaEngine ODBC SDK is an easy-to-use Quickstart Template. This template includes all of the code needed to quickly and easily create a working prototype data driver. Following our method, we guarantee that you will be able to build a prototype ODBC and JDBC driver accessing your data in five days.

Figure 1 shows two examples of deployments for SCADA data stores.

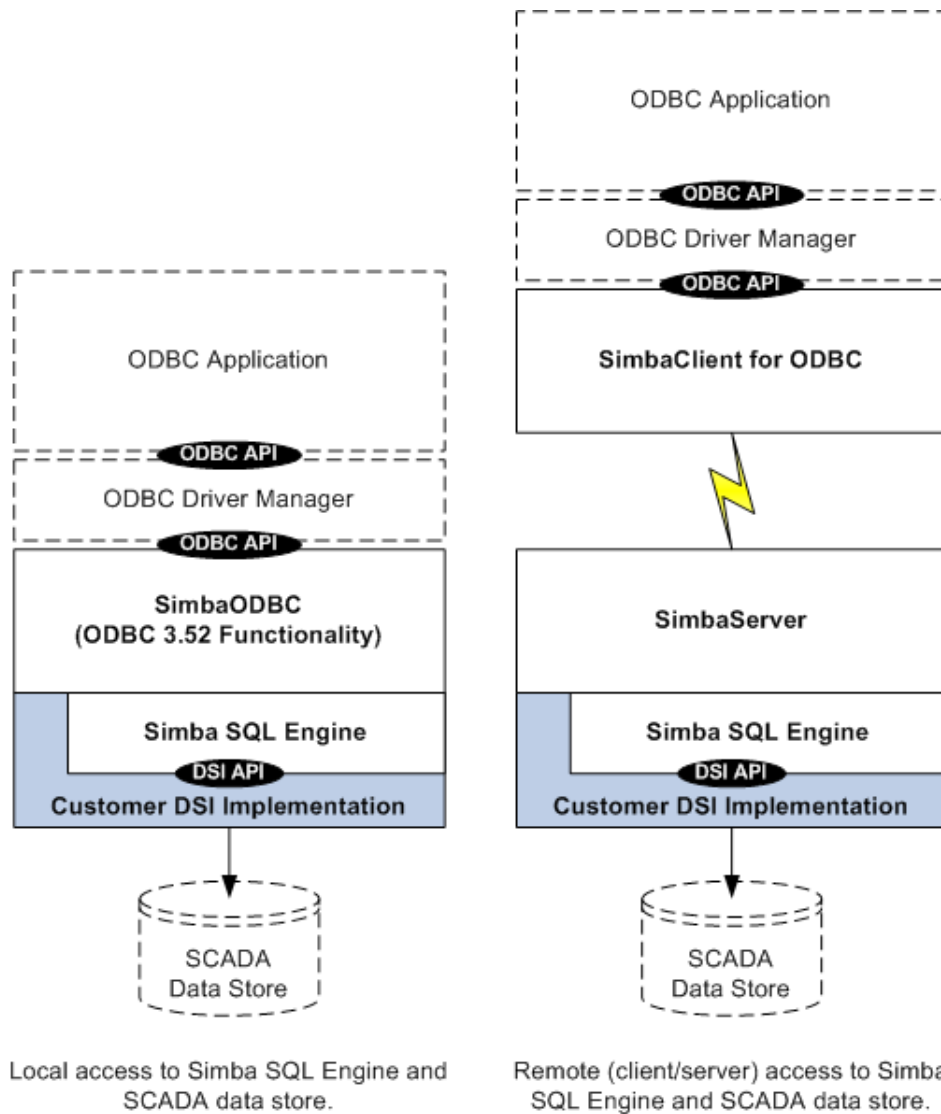


Figure 1: Two examples of a Simba SQL Engine data access solution for a SCADA data store.

Collaborative Query Execution

Simba SQL Engine's Collaborative Query Execution functionality allows your data store to execute those parts of each SQL query for which it has been optimized. You can selectively use your native data store functionality to accelerate query elements such as table filters, joins and aggregations. This feature enhances the performance of complex queries from large data stores and allows you to pass on the performance of your data store to your customers.

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.

©2010 Simba Technologies Inc. All Rights Reserved.
Printed in Canada.

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

Simba and the Simba logo are trademarks of Simba Technologies Inc. All other trademarks or service marks are the property of their respective owners.