



WhiteCap Applications, Inc
Technical Bulletin

Real-time Integration of the Stratus VOS file system Within an Oracle Database Using Periscope

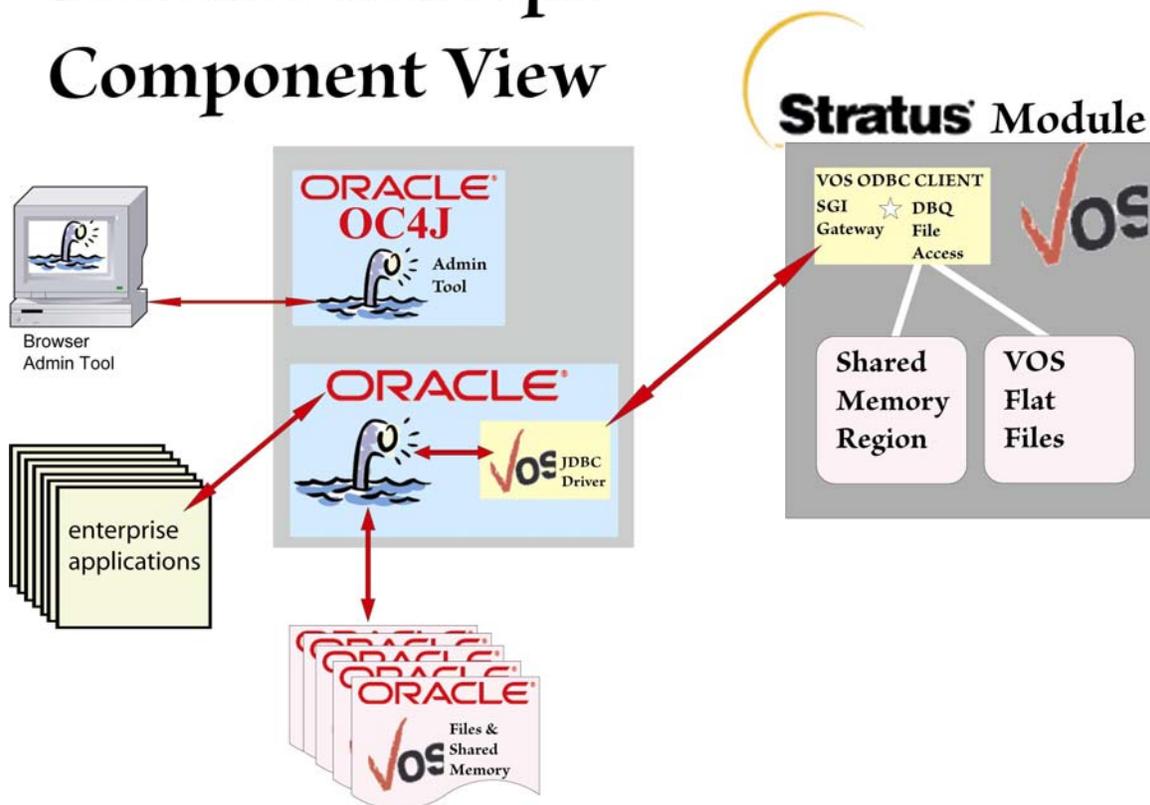
Goal

Real-time seamless access of the Stratus VOS file system from an Oracle database

Benefits

- Stratus VOS data becomes another real-time database source within Oracle
- Stratus VOS data can now be accessed and updated by any application that can connect to the Oracle database
- Stratus VOS data can be “joined” with other business data, and made available thru Oracle tables
- Stratus VOS data does not have to be replicated to Oracle, but can be accessed real-time out of the Oracle database.
- Stratus VOS data is transformed into an Oracle view, so that Oracle applications can work with consistent data types.
- Legacy applications can continue to maintain their Stratus VOS data, while enterprise applications have full access from an Oracle standard
- Reports can be generated out of Oracle that contain a real-time snapshot of all data across the enterprise

Stratus Periscope Component View



There are three major components to the Stratus Periscope connection:

- 1. Stratus VOS Client**
- 2. Periscope Administration**
- 3. Periscope Universal Database Access**

Stratus VOS Client

The Periscope VOS Client is comprised of three packages, implemented as a complete solution:

- 1. DBQ File Access**

Implements file access to both VOS flat files as well as VOS Shared Memory using verbs that are similar to the ODBC standard

- 2. SGI Gateway**

Acts as the TCP/IP gateway for SQL*NET requests from Oracle

- 3. VOS JDBC driver**

Maps Oracle JDBC verbs to VOS ODBC requests

The DBQ component must be configured to identify the VOS files that will be accessed, and their associated file schemas. The configuration is accomplished with the typical VOS Table & Tin method. Shared Memory access is configured via a User Exit routine that must be programmed to understand the Shared Memory layout and access indexing method.

The SGI Gateway implements the TCP/IP connection between DBQ and Oracle.

The VOS JDBC driver was created to map JDBC verbs to VOS IO functions. The VOS JDBC driver communicates with the SGI Gateway via SQL*NET. The driver can be installed into the Oracle database, or into Oracle OC4J if the web services method is being used.

The combination of all components provides full Oracle access to all VOS files. Oracle access is full function (Read, Write, Update, Delete), and real-time. All VOS data is transformed back to an Oracle view and is transparent to applications accessing VOS data via Oracle tables.

Periscope Administration

The Periscope Administration package runs within the Oracle OC4J container as a web service. Periscope Admin is responsible for the definition of all database connection gateways, and for all Oracle code generation necessary to implement virtual database access to a remote database.

Periscope Admin is active only for system configuration. Once a remote database has been “virtualized”, the Periscope Admin tool is not required to be active and can be terminated.

Configuration of virtual database access requires no programming and typically takes less than a couple of minutes for a table.

Periscope Universal Database Access

Periscope implements Universal Database Access to any disparate data source that a driver has been configured for. Periscope is installed into the Oracle database. Drivers for the disparate data source can be ODBC, JDBC, or of a custom type. JDBC drivers can be loaded directly into Oracle if the JDBC driver JAVA version matches the JVM version within the Oracle database. JDBC drivers can also be loaded into OC4J and accessed from Oracle as a web service. We use this method if the JDBC driver JAVA version does not match the Oracle JVM version.

Once Periscope has been installed into Oracle, we use the Periscope Admin tool to “virtualize” a disparate database into virtual Oracle tables. These Periscope driven tables appear as local Oracle tables, but are actually virtual connections into the disparate database source.

Once a disparate database has been virtualized within Oracle, it can be accessed thru normal Oracle functions with few restrictions.

When an application selects data from the Periscope/Oracle table, Oracle invokes Periscope. Periscope maps the IO request thru the appropriate driver back to the disparate data source to be executed. The result of the IO is returned to Periscope, which then transforms the data back into a normal Oracle view. All Periscope driven data accessed thru Oracle will appear as normal Oracle tables and will be completely transparent to the application.

While Periscope is complex, the actual overhead of Periscope against a virtualized table ranges anywhere from 10ms to 150ms for the actual SQL request. Since database access is typically performed against larger subsets of the database, actual overhead per record retrieved is typically insignificant.

Technical Limitations

There are no data type or file type limitations for VOS files represented within an Oracle database. Not all Oracle data types can be found within VOS, so inserting Oracle data into a VOS file might require converting the data to a form that can be stored within the VOS file system.

VOS does not support two phased commits & rollback, so this Oracle feature is not supported when working with VOS files.

For additional information contact:

James J. McFadden
WhiteCap Applications, Inc.
(402) 968-3674
Jim.McFadden@WhiteCapApplications.com