Flex2SQL

Connecting DataFlex Applications to Modern SQL Databases

Introduction

The Internet revolution has underscored the importance of making data available reliably, at high speeds, to an ever-growing user base. Corporations continue consolidating their data into single, enterprise-wide databases so they can easily extract and manipulate information about their customers, products, and market. These databases also allow customers and staff to update and retrieve information at the same time, from any location. The new, data-intensive demands of today's businesses require a database server that's robust, scalable, provides excellent response time, has great disaster recovery features, and above all, provides excellent security features and guaranteed 24/7 availability.

Relational, Structured Query Language (SQL) databases, from companies like Oracle, Microsoft, and IBM (as well as open-source alternatives), meet those criteria. These products cover about 90% of the database market, and every major new application supports one or more of them. Relational databases are now the heart of any reasonably sized IT infrastructure.

For companies that use or develop applications written using DataFlex, moving to SQL can present significant challenges. Typically, these companies must either rewrite their entire application to support relational databases or create a new application from scratch. Both solutions are costly and time-intensive, preventing many companies from making the leap.

Mertech's Solution

Mertech offers a flexible, cost-efficient third option. Our Flex2SQL product allows DataFlex developers to deploy their applications alongside a SQL backend, without rewriting their source code or compromising performance. Using Flex2SQL's easy migration tools, you can quickly convert your DataFlex application to store and retrieve data using a SQL database. Join the over 9,000 companies on five continents that have saved millions of dollars using Mertech's data migration tools and data connectivity drivers.

Flex2SQL Product Fact Sheet

Flex2SQL includes both a GUI migration tool, used to migrate your existing DataFlex data to your target backend, and a Dynamic Link Library (DLL) that handles database operations, client/server connections, and the conversion of DataFlex calls into optimized SQL queries.

Additionally, Flex2SQL includes the following features:

Supported DataFlex Features

- Complete support for all DataFlex code and database structures
- Support for overlap/partial overlap fields
- Complete support for case-sensitive indexes
- Complete support for mixed ascending and descending index segments
- Preservation of RECNUM values during data migration
- Complete support for transaction with rollback
- Exact replication of DataFlex's table structure
- Support for DataFlex file relationships
- Alias file handling
- Support for DataFlex tools like Database Builder and Visual DataFlex Studio

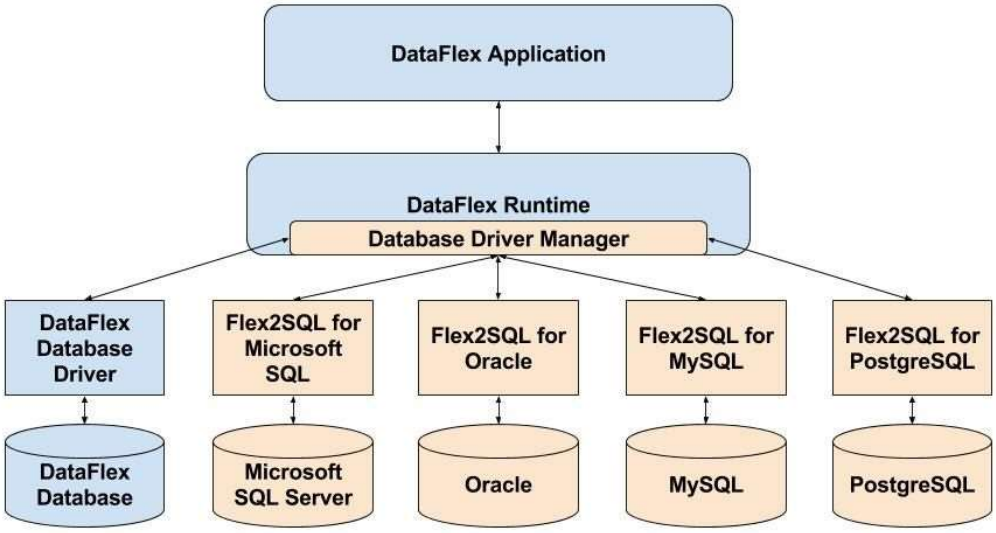
Product-Specific Features

- Comprehensive database migration tool that helps you migrate and manage tables and data
- Access to views and synonyms as DataFlex files
- Record locking instead of file locking
- Seamless integration of non-DataFlex tables into DataFlex applications
- Access to existing data stored in non-DataFlex database systems
- SQL script generation for creating tables and indexes independently
- File creation and restructuring tool that allows you to commit changes directly to the backend or to generate DataFlex restructure code for distribution to remote sites
- Full index maintenance tool that allows addition and deletion of indexes and verification of indexes on the backend
- Extensive macro commands to help optimize data fetches without changing your DataFlex code
- Unified stored procedure interface that works across all drivers in the same fashion

- Comprehensive SQL command set for embedding SQL statements
- High-speed data migration using native tools like SQL*Loader, Bulk Copy, and DB2Load
- Use of native database transaction logging and rollback capabilities for disaster recovery
- Complete migration report that shows you at a glance the creation of tables, indexes, and the number of records migrated
- Storage analysis tool that allows you to evaluate storage availability before migration

Interaction with the DataFlex Runtime

The DataFlex runtime engine in Visual DataFlex and DataFlex 3.1c and above allows external database drivers to be loaded as DLLs and work as part of the runtime, alongside your DataFlex database.



These runtime changes allow the Flex2SQL drivers to work alongside your existing DataFlex programs without modifying your code. All you need is a login to the database (if the database requires it). The login can be handled as a separate dialog which “chain waits” to the main program. The login remains effective for all other programs and “chain waits”.

To give you an idea on how these pieces work together, let’s look at a simple DataFlex program that opens a file, called *Open MYFILE*.

Making a DataFlex API Call with Flex2SQL

The runtime engine calls the DataFlex API function call, dFileOpen(), which searches the filelist entries for a file named MYFILE. It should be mentioned that if MYFILE is on a server, the entry in the filelist will be MYFILE.INT or of the type ORA_DRV:\\servername\schema*tableName. The .INT extension means there is an intermediate file (as described in the next section). The DataFlex API sees the .INT extension (or the embedded server name) and opens the .INT file.

From this .INT file, the API looks for the driver name entry. If the driver was previously loaded, it searches for the driver from a driver table. If it was not previously loaded, then it loads the driver and adds it to the driver table. Once the driver has been loaded, the API calls the driver's FileOpen code and passes control to the driver.

The driver searches the system catalogs to find out if the table name (MYFILE) is a valid table name. If it is, the driver performs the operations necessary to open the file. Once the driver is finished, it passes control back to the API, which then returns control to the runtime engine.

In a similar fashion, the driver handles all major functions, such as find, save, edit, delete, transactions, and locking. The API calls each of these functions for each specific driver.

The Intermediate File

Flex2SQL generates an intermediate (.INT) file for each DataFlex file converted to the target database. The intermediate file contains information used by the DataFlex API and the database driver.

When a command in DataFlex is given to open the file with a root name containing a .INT extension, it is an indication to the API that the file is a non-DataFlex file. Instead of opening the DataFlex file, the API opens the intermediate file.

The API searches the .INT file for the DRIVER_NAME token. This tells the API which driver to load to handle the file being opened. The API then loads the appropriate driver and passes control to the driver to handle "opening" the file.

Handling Key DataFlex Features

RECNUM or RowID Table Style

You can choose to use RECNUM or RowID table style during the migration process. Choosing RECNUM emulates existing DataFlex record processing where a unique numeric identifier is used to identify each record. You can elect to preserve existing DataFlex RECNUM values or let the SQL server assign RECNUM values starting with one.

As long as a DataFlex program accesses a table with the Mertech driver, if the RECNUM field exists, it is automatically incremented. If another program wants to access this table, the Flex2SQL migration utility provides an option to create a trigger on the table to handle the auto-increment feature.

Alternately, you can choose RowID table style to take advantage of new RowID VDF commands. This is new applications' preferred method.

Locking

Mertech's drivers support record-level locking. Since DataFlex locking is based on files, this is an added benefit for using the client/server solution in medium to large multi-user environments. If a DataFlex program contains reread, the driver re-finds the record in the record buffer and locks the current record until the transaction is completed. Any record not explicitly locked can be updated by other users. This provides a great boost in concurrency over the native DataFlex database.

Transactions

Most relational database management systems are configured to automatically start a transaction when save and update operations take place. In DataFlex terms, reread/lock/start_transaction commands all start transactions and unlock/end_transaction commit transactions. Transactions can be aborted or rolled back only by the abort_transaction command. Since most DataFlex programs use reread/unlock combination, there is no automatic provision to rollback transactions in case of an error. However, transaction rollback can be implemented either by using the DataFlex transaction commands or by using commands implemented by Mertech.

Overlaps

All Mertech drivers support overlap fields, even partial overlaps. An overlap field can be thought of as a virtual field which is comprised of one or more actual fields.

Mertech's drivers handle overlaps through the intermediate file. Flex2SQL generates all the necessary descriptions of the overlap fields so the programmer does not have to worry about how the overlap is generated. No change in the code is required to work with overlaps.

Mertech provides triggers that can be enabled to ensure overlaps are taken care of if data is updated using non-DataFlex tools.

Relationships

An important part of good database design is reducing data redundancy and improving efficiency. To achieve this goal, you define each table so it stores only one kind of data and then define relationships between tables. Flex2SQL provides a utility to synchronize relationships between DataFlex tables and the SQL Server backend. Enforcing relationships on the SQL Server helps ensure data consistency.

Relationships are handled in the same way as overlaps, through intermediate files, and work the same way a DataFlex relationship does.

Restructuring

You can restructure and manage your database in a variety of ways. One way is to use Flex2SQL's restructuring dialogs and your own restructuring code. Another option is to make your SQL server the main repository for your database structure and use available SQL modeling tools to manage your database. Then, you can run Flex2SQL's Generate .INT file from Table/View/Synonym command to bring your changes back into DataFlex.

Mertech also supports restructuring and table management using VDF Studio or Database Builder. You can use DataFlex as the master repository for your data, and the Mertech drivers and DataFlex will ensure all updates are reflected in the SQL backend, .INT files, and any FD files.

Embedded SQL

You can achieve significant performance improvements by rewriting some batch processes in SQL. This can be either as embedded SQL statements, or as

a stored procedure or function for execution directly on the server.

A simple example is a business process that computes the outstanding balance for all customers. In a record-oriented scenario, all the records have to be brought over to the client in a loop that computes a running total. This can instead be turned into a short snippet of SQL code that executes fully on the server, with minimal network traffic and interaction.

Mertech has developed a comprehensive SQL command set for embedding SQL statements that optimize your programs for faster data access. In addition, Mertech has developed a class layer that allows you to use the ESQL classes available in the Data Access sql.pkg. Mertech's class layer implements a unified ESQL interface across all our drivers, allowing for applications to migrate to database servers DAW does not support, such as MySQL and PostgreSQL.

Other Features

Dynamic Primary Keys

This powerful feature addresses a shortcoming in VDF when it comes to using server-side generated Universally Unique Identifiers (UUID), sometimes called Globally Unique Identifiers (GUID), as primary keys in database operations. This new feature gives developers the ability to create UUID columns with server-side UUID generation and use them seamlessly with VDF applications.

NULL Handling

DataFlex applications have no concept of a SQL NULL value. A NULL value is always mapped to an empty value in a DataFlex application. For example, if you have a DF_ASCII field that is assigned to a VARCHAR table column, and the column is NULL, choose whether you want a non-index field to be NULLable when creating a new table or restructuring an existing table. If NOT NULL is selected, you can choose an appropriate default value for the field.

Microsoft Azure

Microsoft Azure is an application platform that allows you to deploy in the Cloud. Mertech drivers provide transparent support to run your Microsoft Azure SQL Database on Azure, with no Flex2SQL-related code changes required. We do not currently support Azure migration.

Failover Support

Microsoft introduced database mirroring to increase database availability. When mirroring is configured and a failure is detected in a principal server, a standby or mirror server can take over with no loss of committed data. To add support for SQL Server Mirroring, Mertech added the SET_MIRROR_SERVER command to the SQLFlex command set. To use the SQL Server Mirroring feature, developers add the SET_MIRROR_SERVER command to their login module. This identifies the mirrored server to the driver, which in turn negotiates communication with the server. If the principal server fails, the driver automatically connects to the mirror server and continues operation. No additional code is required in DataFlex to check for a lost connection.

Auto-reconnect

Mertech's driver code that handles the reconnect after a mirroring failover is the same code used to handle the auto-reconnect after other disconnects. Previously, if a server timed out a client connection due to lack of activity, a DataFlex program encountered unexpected errors. With Mertech's auto-reconnect feature, the driver automatically reconnects to the server and continues from the same point where the disconnect occurred.

Supported Platforms

Flex2SQL supports operating systems in which DataFlex and one of the target backends listed below are available. The following table shows Flex2SQL's availability for various platforms.

Operating System (with DataFlex installed)	Oracle - 10g or later	MS SQL Server - V2005 and above (including Azure and LocalDB)	MySQL - V5.0 and above	PostgreSQL - V8.4 and above
Linux	X		X	X
Windows	X	X	X	X

Evaluate Our Product

You can download and evaluate our product, free of charge, at mertechdata.com. The Flex2SQL download includes a 30-day, full-use license. After 30 days, all you need is a registration code to continue. Contact sales@mertechdata.com to purchase a registration code after your trial has ended.

Contact Information

If you'd like to know more about Mertech's products, please visit our website, mertechdata.com, or contact us at:

Corporate Head Office

Mertech Data Systems, Inc.
18503 Pines Blvd. Suite 312
Pembroke Pines, FL 33029 USA
Tel: +1.954.585.9016
Fax: +1.866.228.1213
Email: sales@mertechdata.com

California Office

Mertech Data Systems, Inc.
7621 N. Del Mar Ave., Suite 101
Fresno, CA 93711 USA

Copyrights and Trademarks