

MERTECH DATA SYSTEMS, INC.
18503 Pines Boulevard, Suite 312 | Pembroke Pines, Florida 33029 | USA
Tel: (954)585-9016 | Fax: (866)228-1213
www.mertechdata.com

Flex2SQL™

Mertech's ISAM to SQL Database Connectivity
(ISDBC) Drivers For DataFlex®



Contents

- Overview
- Product Fact Sheet
- Interaction With The DataFlex Runtime
- Handling Key DataFlex Features
- Supported Platforms

Copyright © 2000 - 2008 Mertech Data Systems, Inc. All rights reserved. This document is for informational purposes only. Mertech makes no warranties, expressed or implied, in this document.

Oracle is a trademark of Oracle Corporation.
SQL Server is a trademark of Microsoft Corporation.
Btrieve® and Pervasive.SQL® are trademarks of Pervasive Software.
Magic® is a trademark of Magic Software Enterprises.
Other trademarks and trade names mentioned herein are the property of their respective owners.

Overview

The Internet revolution has underscored the importance of making data available reliably and at a high speed to an ever-growing user base. Corporations continue to consolidate their data into single, enterprise-wide databases so that information about customers, products, and the market can be easily extracted and manipulated. These databases also allow customers and their mobile work force to update and retrieve information at the same time, from any location. The new, data-intensive demands of today's businesses requires a database server that is robust, scalable, gives excellent response time in extracting and manipulating data, has great disaster recovery features and above all provides excellent security features and guarantees 24X7 availability.

Relational or Structured Query Language (SQL) based databases from companies like Oracle, Microsoft, IBM and open-source databases such as MySQL and PostgreSQL meet those criteria. These products cover about 90% of the database market and every new major application supports one or more of these databases. As you can see, relational databases are at the heart of any reasonable-size IT infrastructure.

Mertech's ISAM to SQL Database Connectivity (ISDBC) drivers allow DataFlex developers to offer SQL database servers with their applications *without changing source code or compromising on performance*. The seamless integration of non-DataFlex databases with a DataFlex application has been made possible largely by a change in the DataFlex runtime architecture. This change was introduced with DataFlex 3.1c and Visual DataFlex 4, which allow multiple databases to work with DataFlex applications without rewriting or recompiling any existing data (as long as these applications run on the versions described above or later versions).

The Flex2SQL product bundle is comprised of a GUI migration tool that takes your existing DataFlex tables and indexes and migrates them to your target backend, creating the table structures, index structures and recnum values for all the files in the filelist. The second component of the Flex2SQL bundle is the database driver DLL or shared-object, which works with the DataFlex runtime and handles all client/server connection and database operations.

Flex2SQL Product Fact Sheet

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 table structure.
- Support for DataFlex restructure commands
- Alias file handling
- Support for DataFlex tools like DBBuilder, DBExplorer, DFQuery (the version shipped with DF 3.2 not earlier versions)

Product Specific Features:

- Comprehensive database migration tool that eases the job of migrating and managing tables and data
- Access views and synonyms as DataFlex files
- Record locking instead of file locking
- Seamless integration with existing TABLES and VIEWS by automatic generation of FD and TAG files from existing tables and views
- Detailed migration report
- SQL Script generation for creating tables and indexes independently
- Table Restructuring Utility for easier maintenance of migrated tables
- Visual Query builder for visual creating and testing your queries
- Export of Visual Query Builder output as DataFlex code using embedded SQL
- Index Maintenance Utility to manage foreign and primary keys after migration.
- Extensive macro commands to help optimize data fetches without changing your DataFlex code
- Support for calling stored procedures and functions
- Comprehensive SQL command set for embedding SQL statements
- High-speed data migration for all supported databases
- Read data back from table into DataFlex files

Interaction with the DataFlex Runtime

The DataFlex (DF) runtime engine in Visual DataFlex (VDF) and DataFlex 3.1c and above allows external database drivers to be loaded as DLLs and work as a part of the runtime. The DataFlex database is also handled as a driver and is the default driver.

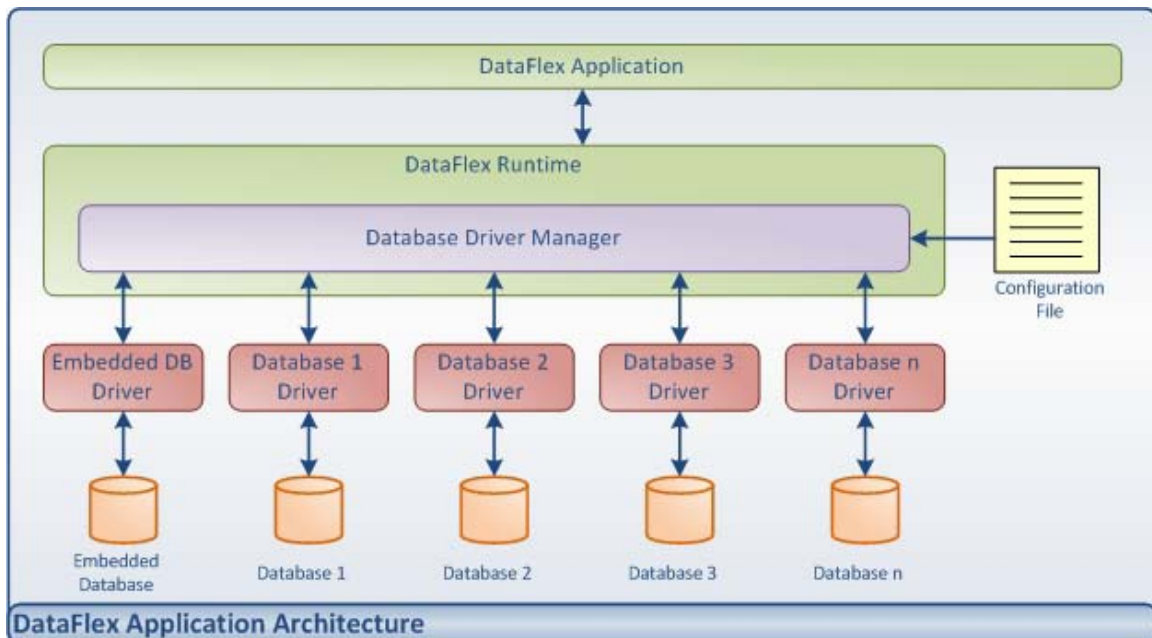


Figure 1: DataFlex runtime architecture and driver interaction

These changes in the runtime enable a driver to work with existing DataFlex programs without changing a single line of code. All that is needed 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 the reader an idea on how these pieces work together, let us work with a simple DataFlex program that opens a file.

```
Open MYFILE
```

The runtime engine calls the DataFlex API function call, `dfFileOpen()`, 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\schemata*tableName`. The “.INT” extension means that there is an intermediate file. There is more information on this in the next section. The DataFlex API sees the .INT extension (or the other format) 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 `FileOpen` code and passes the control to the driver.

The driver, on its part, does whatever is necessary to “open” the file. One function that the driver code performs is searching the system catalogs to find out if the table name (MYFILE) is a valid table name or not. If it is, it performs certain operations that are needed to complete the opening of the file. Once it is finished it passes the control back to the API, which returns control back to the runtime engine.

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

The Intermediate File

The intermediate file is a text file generated by Flex2SQL, Mertech’s database migration utility, for each DataFlex file that is converted over to the target database. The intermediate file (.INT) contains information used by the DataFlex API and the driver.

When a command in DataFlex is given to open the file with a root name containing an .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. If there is an error in opening the intermediate file, an appropriate error message is generated.

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” of the file.

Handling Key DataFlex Features

Recnum

Mertech's drivers support Primary Index (or Record Identity Index). The user has an option of creating an auto-increment field during the migration of the table. The primary key on the table is the first index of the DataFlex data file, not the recnum field. However, any index can be chosen as a primary index and it does not have to be a unique numeric key. In the absence of an intermediate file, the first unique numeric index on the table is selected. If a unique numeric index is not present, then the first unique index is used as RECNUM. If not, an error will be generated.

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

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.

Transactions

Most relational dbms 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. See the *Programmer's Guide for Mertech Drivers* for more details on these new SQL commands.

Overlaps

All Mertech drivers support overlap fields, even partial overlaps. Overlap fields are very specific to DataFlex and are extensively used. The 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 (this means that if your DataFlex database contains overlap fields, you will always have to use the intermediate file to access the tables on the server). 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.

Relationships

Relationships are handled in the same way as overlaps, that is, through intermediate files, and work the same way as a DataFlex relationship does. The current relationship implementation of the DataFlex runtime is not able to exploit joins on the server.

Restructuring

Mertech's products provide complete support for DataFlex restructuring code as well as support for DBBuilder. However, the drivers do not work with DFFile utility, as DFFile is not compliant with the 3.1x runtime architecture.

The driver only supports those functions that are supported by the native database. For example, Oracle does not allow you to drop a column from a table. The operation will report the appropriate error.

Alias File Handling

If you are using alias files and have OPEN AS commands, you will have to change how the non-DataFlex files are handled after migration. Instead of using

```
OPEN "XYZ" As "FileName" // open DataFlex file
```

you will now have to use

```
OPEN "driverName:XYZ" as "FileName // or  
OPEN "XYZ.INT" as FileName
```

for recently migrated tables. The driver prefix or the .INT file suffix both tell the DataFlex runtime that the alias file is a non-DataFlex file.

Support for Embedded SQL

SQL databases like Oracle and MS SQL Server offer a robust and significantly faster database manager than DataFlex does. Mertech's drivers handle all of the intricacies of using DataFlex with relational databases, a set oriented database engine. However, there are some areas where, with a little tuning of the code, you can significantly speed up a batch process by utilizing the set nature of the relational databases. The advantages of using embedded SQL are that the power of SQL generates more complex results, gives you more control over your results and reduces network traffic.

The most important area where using these macro extensions will benefit is in batch processes. Embedded SQL statements or stored procedure calls can be used for procedures that manipulate a large number of tables and data. By using stored procedures, most of the processing is done on fast servers and very little data transfer takes place between the server and the client.

Keeping the above objectives in mind, Mertech has developed embedded SQL commands that allow you to optimize your programs for faster data access.

Supported Platforms

	<i>DBMS</i>	Oracle	DB2	MS SQL	MySQL	PostgreSQL
OS Platforms						
HP Unix		X				
IBM AIX		X	X			
Linux		X	X		X	X
SCO Open V					X	
SCO Unixware		X				
Sun Solaris		X				
TRU64 Unix		X				
Windows		X	X	X	X	X

X= Supported

Supported Oracle Version:

Oracle version 7.3x and above

Supported DB2 Versions

DB2 version 7.x and above

Supported MS SQL Server Version

MS SQL Server 6.5 and above

Supported MySQL Server Version

MySQL Server v**4.1** and above

Supported PostgreSQL Server Version

PostgreSQL Server Version 7.x and above (we recommend 7.3.x and above)

Evaluate Our Product

You can evaluate our product free of charge for 30 days at <http://www.mertechdata.com/>. After 30 days, all you need is a registration code to continue.

All of our product documentation and white papers are also available on our website. Case studies are posted on the website as well.

If you have any questions regarding our products and your environment, please feel free to contact us. Send an email to our tech support at techsupport@mertechdata.com or sales at sales@mertechdata.com or visit us at <http://www.mertechdata.com/>. Pre-sales consultations are free!