



# Using MS-SQL Server with Visual DataFlex



**March, 2009**

## Target Audience

It is assumed that the reader of this white paper has general knowledge of the Visual DataFlex development environment. No specific Microsoft SQL Server knowledge is assumed.

## Introduction

This whitepaper describes the concepts behind and the tasks involved with using Visual DataFlex with Microsoft SQL Server. Specifically, we will be focusing on Microsoft SQL Server Express. The beginning of the paper focuses on the architecture behind the database connectivity features of Visual DataFlex while the latter part of this paper focuses on implementation steps.

All screen shots and references were made using Visual DataFlex 14.1. While the concepts apply to earlier versions of the product, 14.1 made major modifications to database handling in the Visual DataFlex Studio. Thus, if not using 14.1, you may find the specific references will not apply to your earlier versions of Visual DataFlex. It is suggested that you download and install 14.1 if you will be performing a test migration using this white paper as a guide.

We will begin with a discussion comparing and contrasting the native DataFlex database and the Microsoft SQL server database.

## Native DataFlex databases in Visual DataFlex

In essence, the application opens each individual table directly. The *open* command accesses the *filelist.cfg* to determine the physical location of the file and that file (the *.dat*, *.k\**, etc) is opened directly. As updates are made, data is written directly into the opened file. Often, DataFlex native database tables reside on a file server. Thus, the application is often opening a file on a different machine than it is running on, but it is still opening the files directly. Figure 1 displays a diagram showing the architecture of native DataFlex database connectivity within Visual DataFlex.

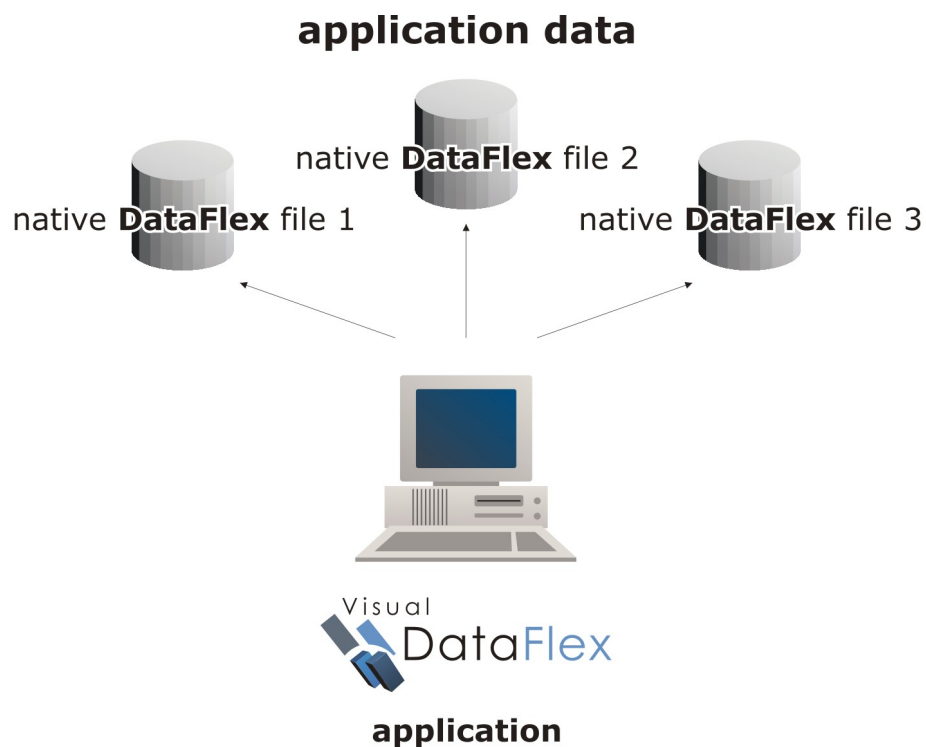


Figure 1

## Advantages of the native DataFlex database

- Extremely easy to manage as there is no database server involved.
- Database access is normally extremely fast especially for find and save operations.
- Simple deployment as you can just copy files from your development environment to your deployment environment.

## Disadvantages of the native DataFlex database

- All locks are file locks (as opposed to record locks). In an environment with many users, performance may degrade during heavy activity as locking is not granular.
- There is no transaction logging or journaling. If the file server hosting the data files crashes, and any damage is done to the data files, you may lose important data.
- If there is any type of network interruption, data corruption can occur.
- There is no security built into DataFlex native databases. No passwords or access controls are part of the native DataFlex database architecture.
- A DataFlex native database is somewhat of a proprietary data format. Other applications with database connectivity will often be unable to access DataFlex native databases.
- A DataFlex native database is limited to 16.6 million records.

## SQL Server databases in Visual DataFlex

Each *open* command access an intermediary file called a **.INT** file (intermediate file). The **.INT** file contains the information on where the non DataFlex native database table can be found, information on indexes, hidden columns, table type, etc. In the case of Microsoft SQL Server, it would contain the server name, login ID, password, database and table name. Thus, when Visual DataFlex opens a Microsoft SQL Server table, the **.INT** file is referenced and the information passed on to the Microsoft SQL Server Connectivity Kit driver. The driver then translates the open call into SQL statements to open the table on the target SQL Server. Figure 2 is a diagram showing the architecture behind how Visual DataFlex accesses Microsoft SQL Server tables.

The beauty of this architecture is that the underlying Visual DataFlex application does not need to be modified. The translation takes place at the driver level and not at the application level. This means that in most cases little or no code modifications would be needed to allow a Visual DataFlex application that currently uses the native DataFlex database to use Microsoft SQL Server databases.

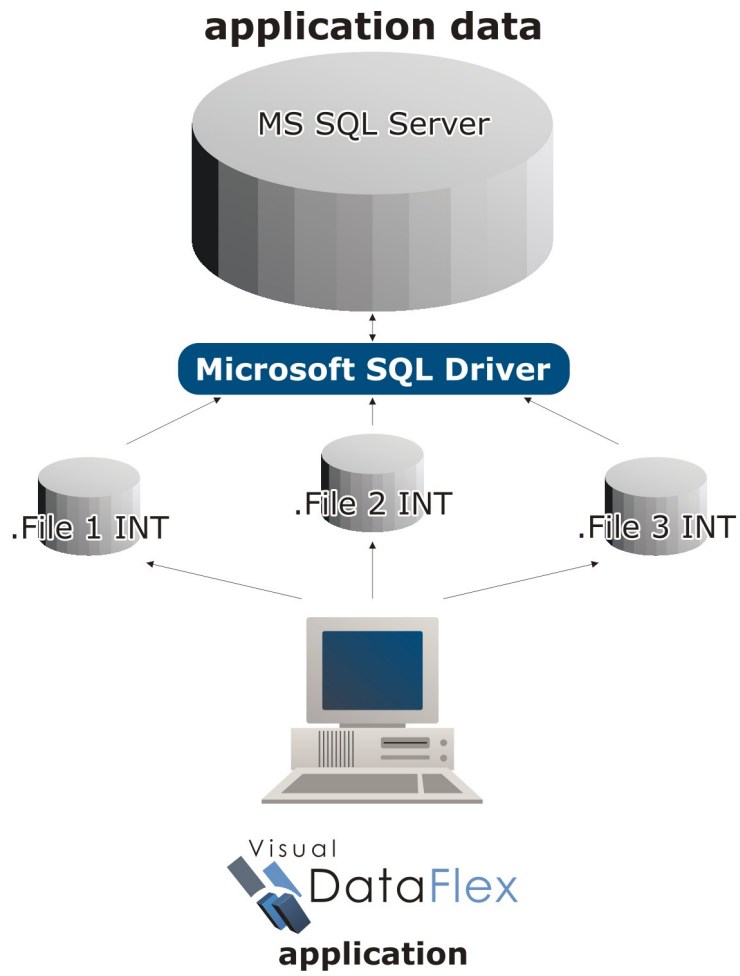


Figure 2

## Advantages of Microsoft SQL databases

- Microsoft SQL Server has built in data integrity/recovery facilities.
- Microsoft SQL Server has user and group level security.
- Record locking is supported allowing a more linear performance curve when dealing with large numbers of concurrent users.
- It is an industry standard database meaning almost any database aware application will be able to access your data.
- It is a known database backend that can lend credibility to your application offering.

## Disadvantages of Microsoft SQL databases

- Requires a level of knowledge of the Microsoft SQL Server environment.
- Deployment and migration is a bit more involved as it's not as simple as just copying files.
- Cost – depending on the edition right for you, SQL Server will need to be purchased in addition to Visual DataFlex.

## Migration Steps

The basic sequence of steps to follow to migrate a Visual DataFlex application that uses native DataFlex databases to Microsoft SQL Server is as follows:

1. Download and install the Microsoft SQL server;
2. Install the Microsoft SQL Server Connectivity Kit (a Data Access Worldwide product) Note, the Visual DataFlex Studio comes with a developers version of the MSSQL driver for development purposes;
3. Open your application in the Visual DataFlex Studio;
4. Load the Microsoft SQL Server Connectivity Kit (MSSQL) driver;
5. Convert your tables to Microsoft SQL Server;
6. Test your application.

In this white paper, we will be covering:

- Downloading and installing the Microsoft SQL Server (express edition).
- Accessing an existing workspace using DataFlex native databases.
- Loading the Microsoft SQL Server Connectivity Kit (MSSQL) driver.
- Converting the tables to Microsoft SQL Server.

## 1. Downloading and Installing the Microsoft SQL Server

For purposes of this document, we will be using Microsoft SQL Server Express 2008. The express version of Microsoft SQL Server is free, which makes it ideal for migration testing. The express version is limited to 1 CPU, 1 GB of memory, and up to 4GB of data area for tables. For many applications, this limitation may be acceptable. For other applications, especially applications with large numbers of concurrent users or large data stores, a different edition of SQL Server 2008 may be required. Of course, other editions will require a product purchase from Microsoft. You may check all the editions available and their limitations at <http://www.microsoft.com/sqlserver/2008/en/us/editions.aspx>.

To download Microsoft SQL Server Express 2008, follow this line <http://www.microsoft.com/express/sql/download/>.

There are multiple editions available for download. The one we will be using in this white paper is SQL Server 2008 Express with Tools. For details on how to install SQL Server 2008 Express, refer to **Appendix A**.

## 2. Convert your tables to Microsoft SQL Server

Now that Microsoft SQL Server is installed, we can perform the conversion. Start by opening Visual DataFlex Studio and loading the workspace that you want to convert (Important! You should make a copy of the workspace folder prior to this step so that you experiment with the conversion and still have the original workspace intact).

For our example, we will use a copy of the Order Entry sample.

The first step is to run Database Builder. Click on the Tools pull down and select Database Builder (see figure 3).

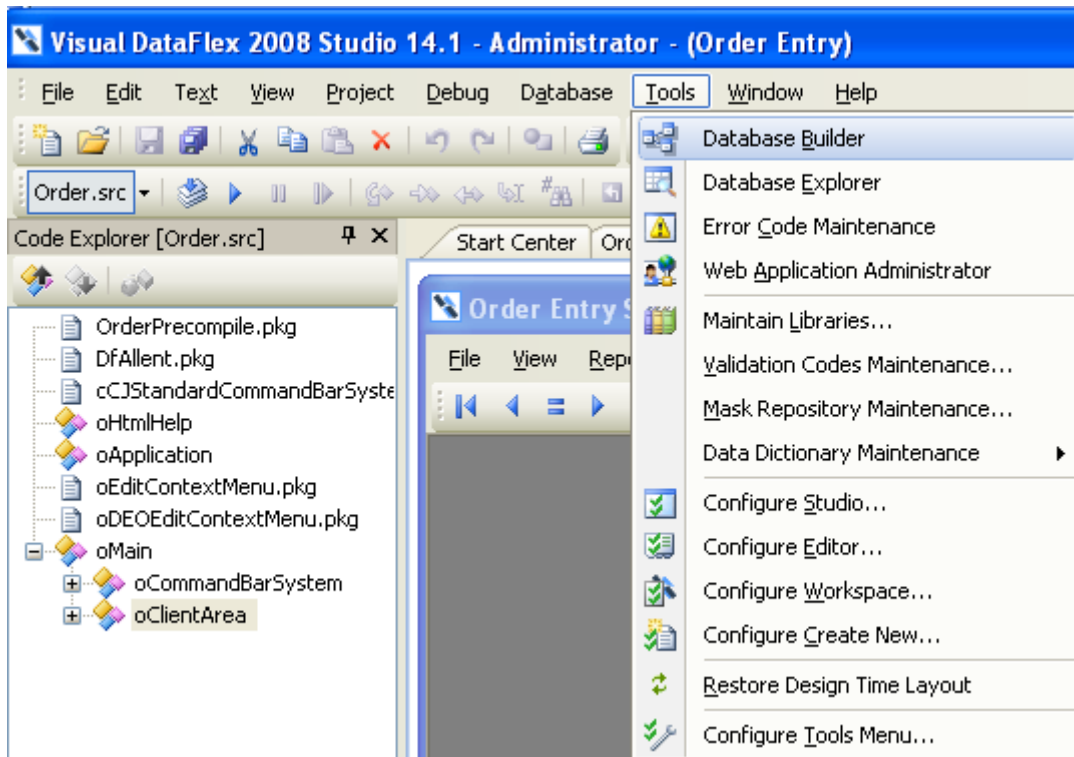


Figure 3

In Database Builder, click on the Workspace pull down and choose Configure the Database Connection (figure 4).

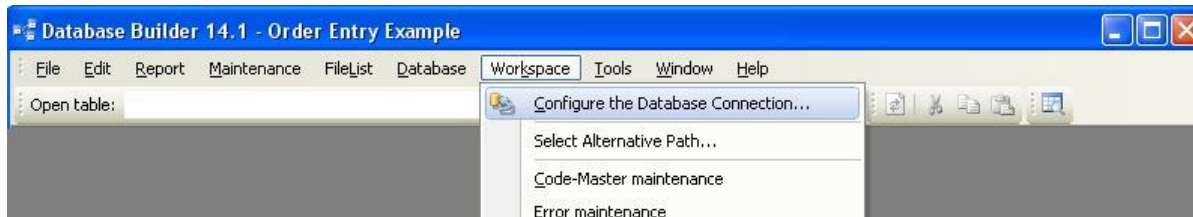


Figure 4

This will launch the Configure Database Connection Wizard. Click the *Driver Administrator* button so that you can select the MSSQL driver (figure 5).



Figure 5

We then select the MSSQL driver on the list, and then click the *Load Now* button to load our MSSQL driver (figure 6).



Figure 6

Then, click the OK button, and in the “Select a database type” pull down, select the MSSQL driver (figure 7).

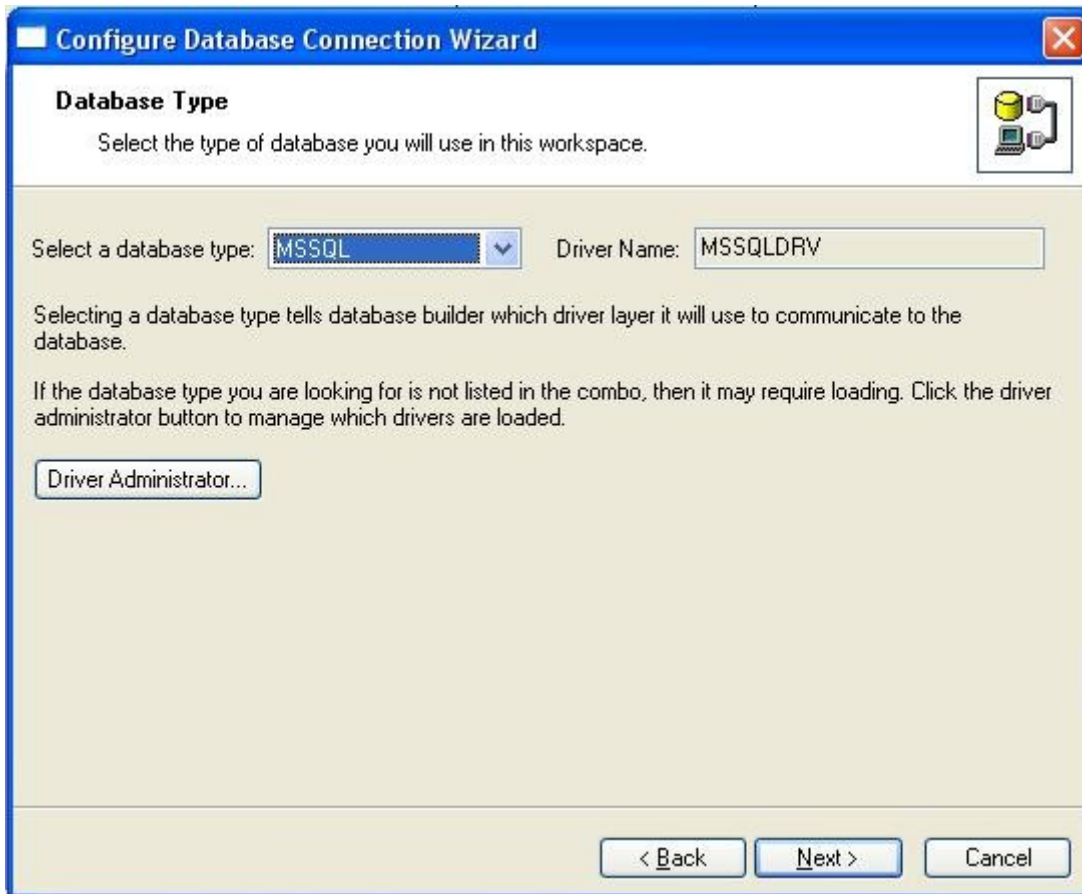


Figure 7

Next, click the *Next* button and fill out the server name, login name and password (figure 8). Note, in this white paper, we are using the default “sa” login and the default “master” database. In a live environment, we would recommend that you create a new database for your Visual DataFlex application and also create one or more SQL users. Information on how to do this is beyond the scope of this white paper but can be easily found in the Microsoft SQL Server documentation.



**Configure Database Connection Wizard**

**MSSQL Database Connection**

Enter the connection settings to your MSSQL database

Select or enter a Server Name:  
 DOUG3200\SQLEXPRESSVDF

Enter user name and password to log on to the server:  
 User Name: sa  
 Password: \*\*\*\*\*  Store Password

Use NT Authentication

Select or enter a Database Name:  
 master

Login to server successful!

< Back    Next >    Cancel

Figure 8

Our workspace is now configured to use the Microsoft SQL Server driver by default, and to use our server as the default connection.

Next, we click on the Database pull down and see an SQL Server option. Select that and click “Convert to MS SQL Server...” (figure 9).

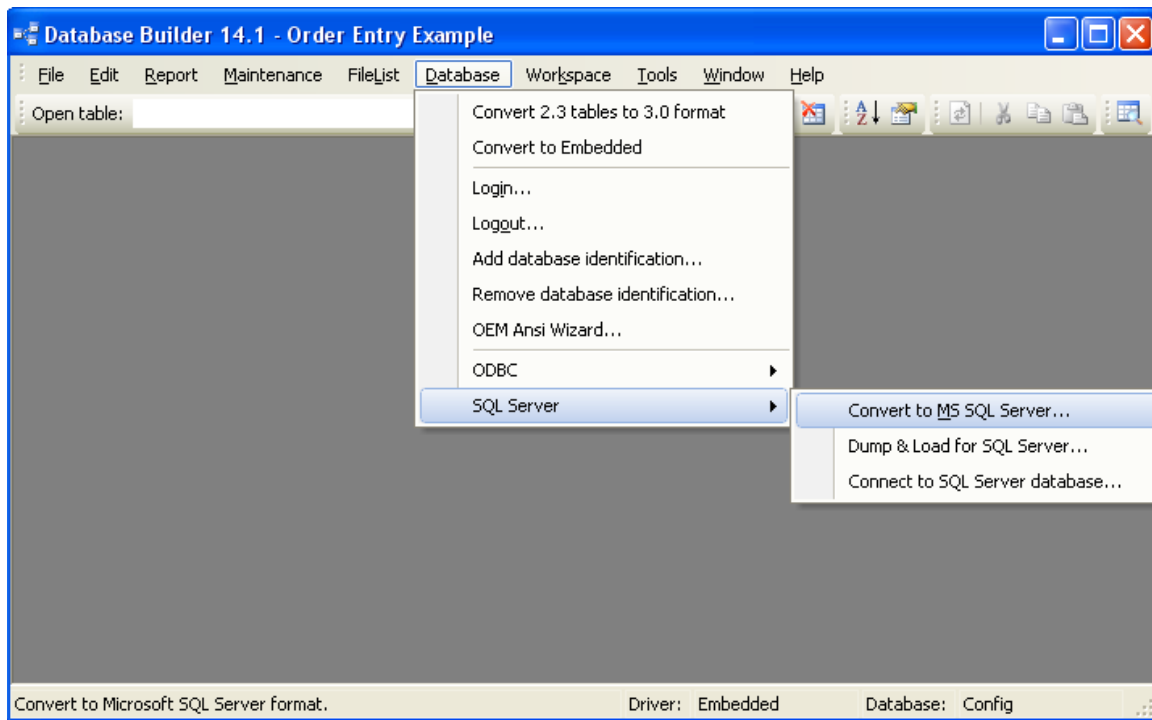


Figure 9

The conversion wizard will start and show us a list of all of our tables. Select the tables to convert (in our example, we select all of them – Figure 10).

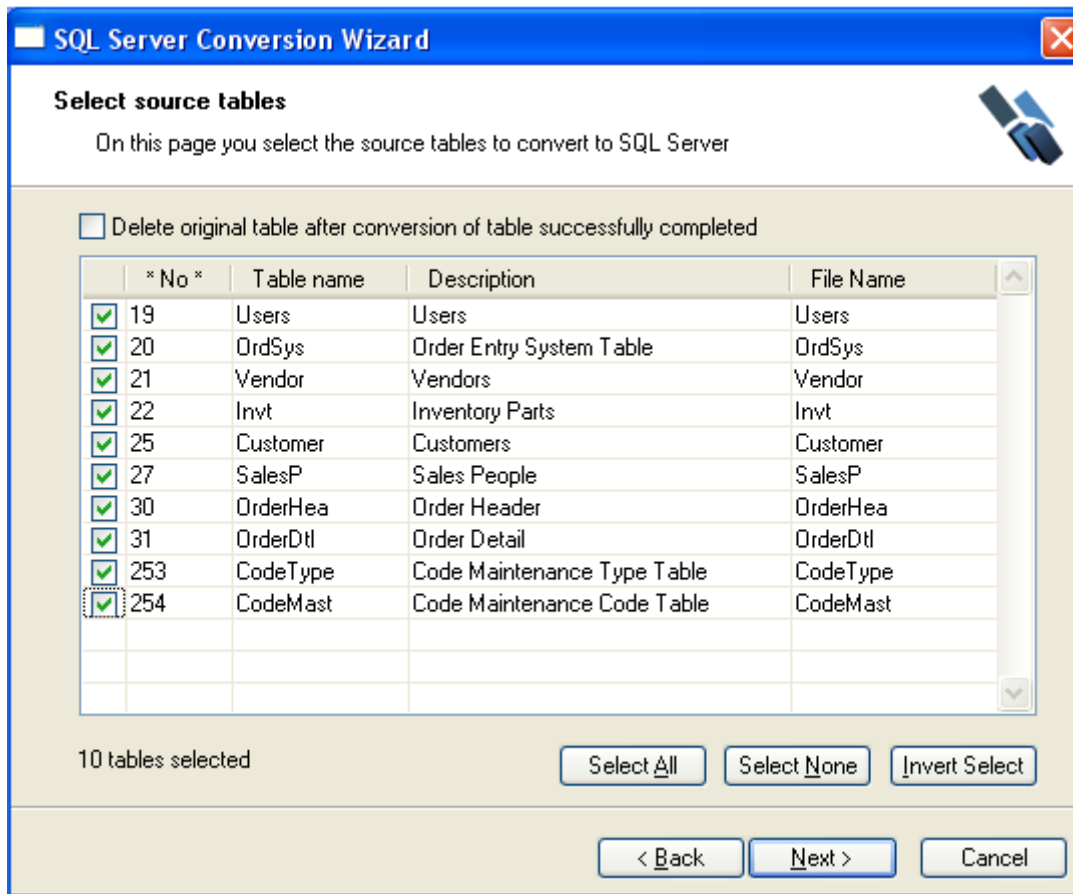


Figure 10

Notice the checkbox at the top *Delete original table after conversion of table successfully completed*. If checked, the original DataFlex native database files will be deleted when the conversion completes. Leaving the checkbox unchecked keeps the original files (although they will no longer be used by Visual DataFlex).

Now, click Next to continue the conversion wizard.

The wizard will scan the table structures and give you a status screen (figure 11).

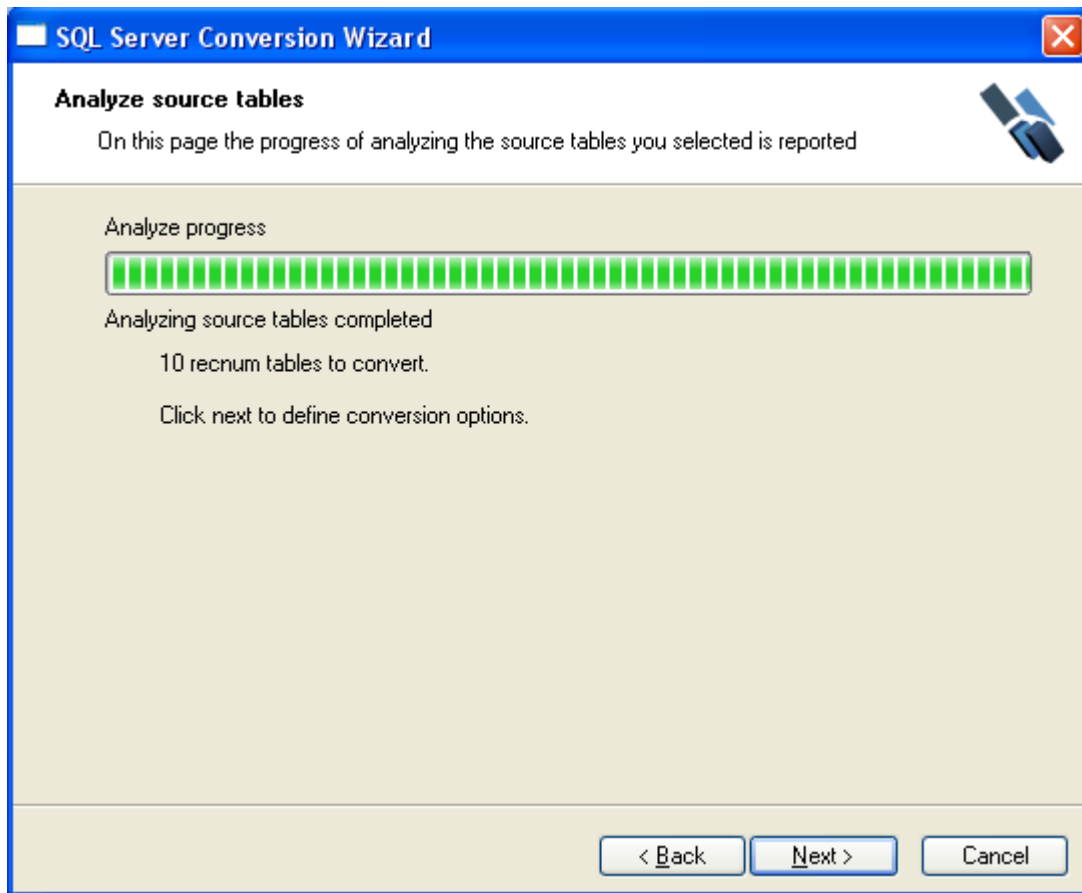


Figure 11

Notice it shows us there are 10 recnum tables to convert. Our SQL Server driver supports two types of SQL tables: recnum and standard. A recnum table is created for maximum compatibility and actually creates a new field to store a unique numeric identity for each record (like the RECNUM field in DataFlex native tables). This is probably what you will need to use if you are moving legacy applications that use recnum in any fashion. The standard table type doesn't create the recnum field for you, but instead relies on you specifying one or more fields that when combined create a unique record identity. Standard tables use the RowId concepts in Visual DataFlex instead of the older RECNUM concept.

The next screen (figure 12) shows you how you can specify which of the tables (if any) should be converted to standard tables as opposed to recnum tables. This will then allow you to specify a unique identifier rather than having a recnum field added to your table.

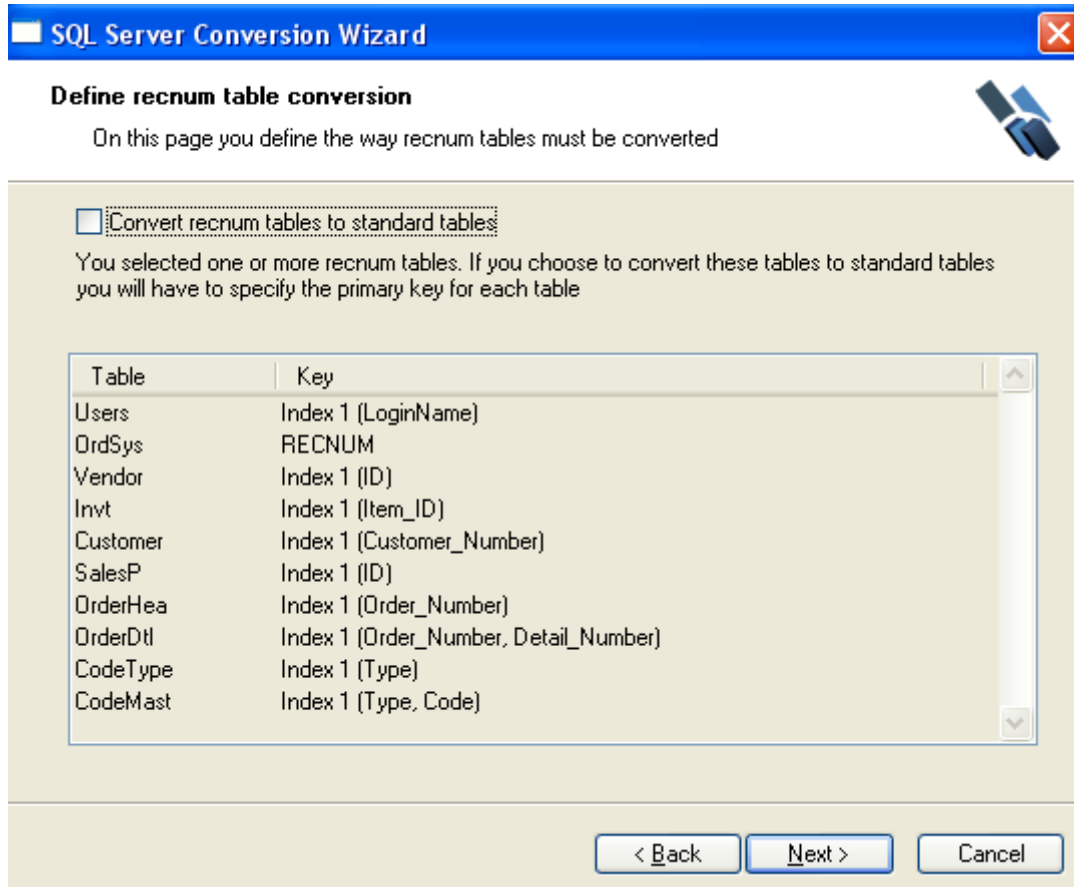


Figure 12

In this example, we will keep the tables as Recnum tables.

Next, we are asked for some information: the name of the SQL Server, the database to use (an SQL Server can have multiple named databases – collections of tables), the user to log in as, the login password, and the SQL Server user who will be assigned as owner of the newly created tables (figure 13). Notice how the wizard automatically defaulted to our server and user, using the information we setup when we configured the workspace to use the Microsoft SQL Server by default.

**SQL Server Conversion Wizard**

**Define Conversion Options**

On this page the conversion options for SQLServer are defined

Server: DOUG3200\SQLEXPRESSVDF

Database: master

User: sa

Password: \*\*\*\*\*

Owner: |

Use NT Authentication

Store data in Ansi format

Convert definition only

< Back   Next >   Cancel

Figure 13

Since we are using a login name and password instead of network login credentials, we leave *Use NT Authentication* unchecked. We also want to convert the data and the table definition, so we leave the *Convert definition only* unchecked.

Click Next and you will see a status page (figure 14).

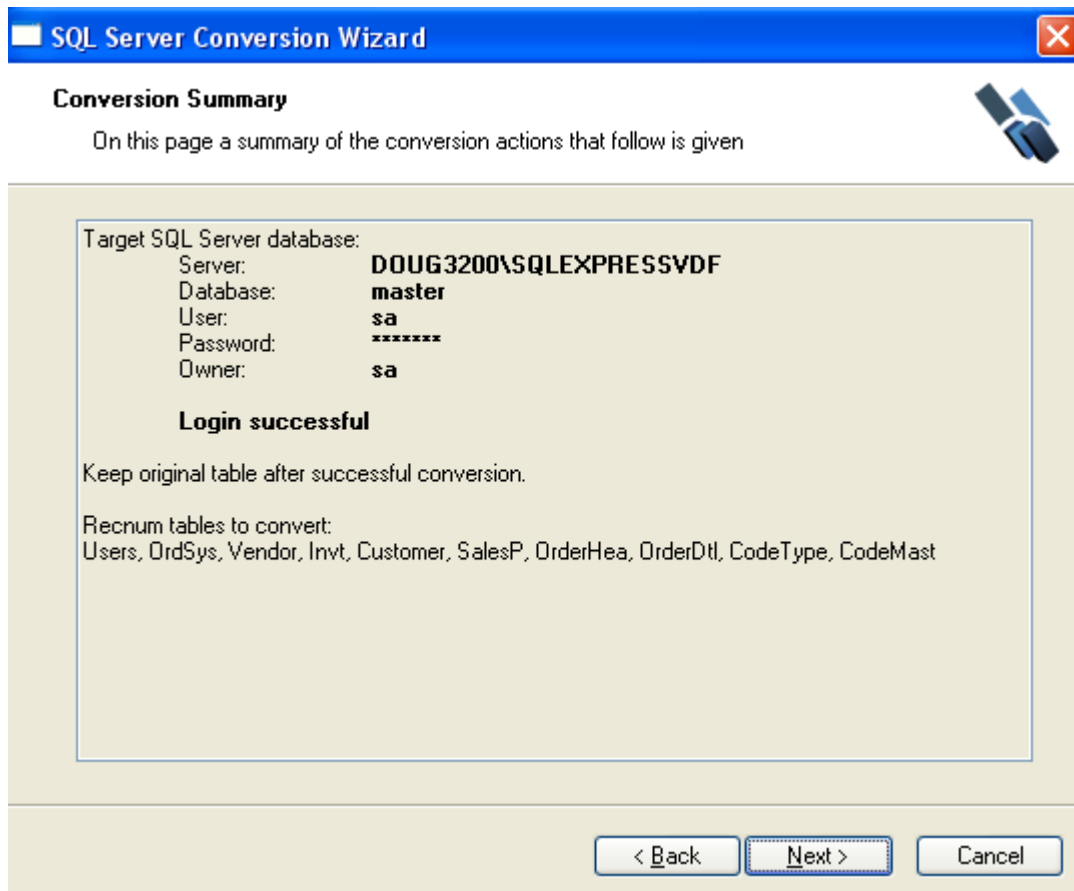


Figure 14

Click *Next* and the process will continue. If your database does not have ARITHABORT set, the wizard asks if you would like to have that SQL option enabled (ARITHABORT) to allow for case insensitive indexes. We click yes to allow this feature (figure 15).

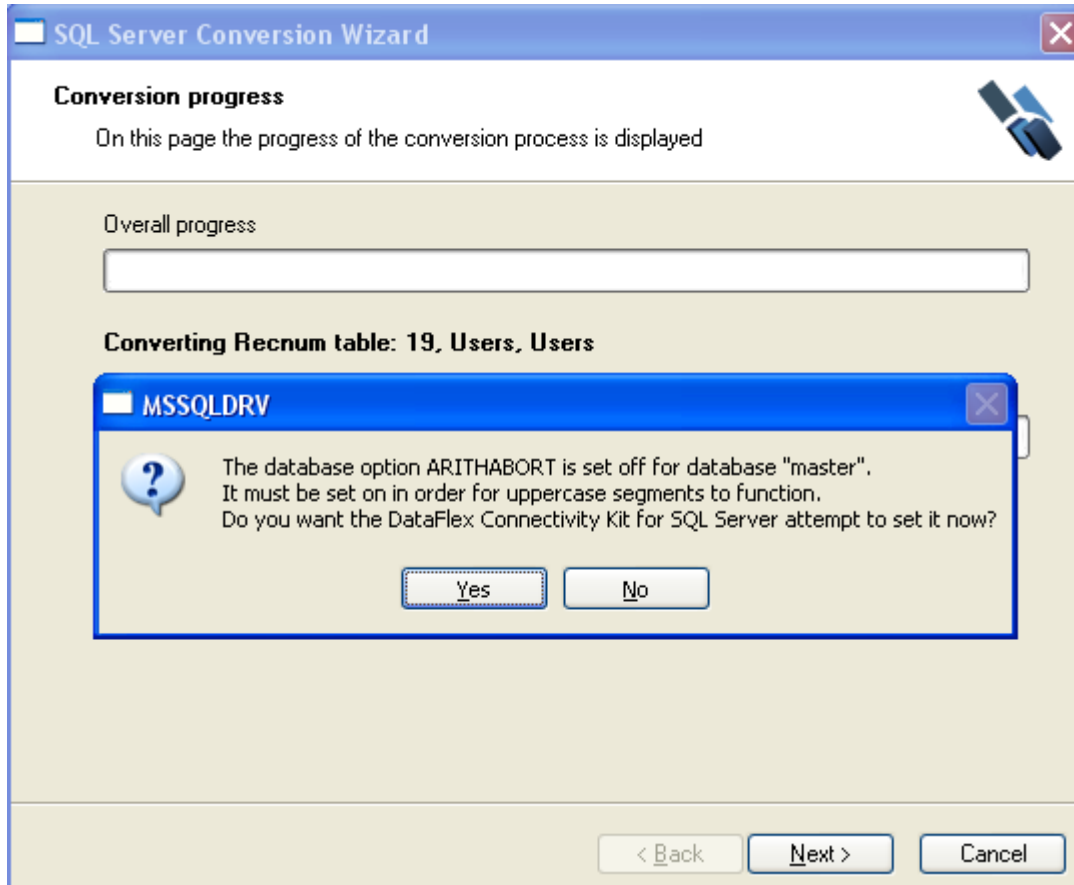


Figure 15

Click *Yes* to allow case insensitive indexes, then click *Next*. The conversion will then process. At the end, you will see a status page listing a summary. When the conversion completes, your tables have now been converted to Microsoft SQL Server and the data copied to the server. If any errors occurred during the conversion, you can open the file *Mssqldrv.Log* to review a log of the conversion process, including any errors that occurred.

To see what Database Builder did, click the File pull down and choose the Open Table option (figure 16).

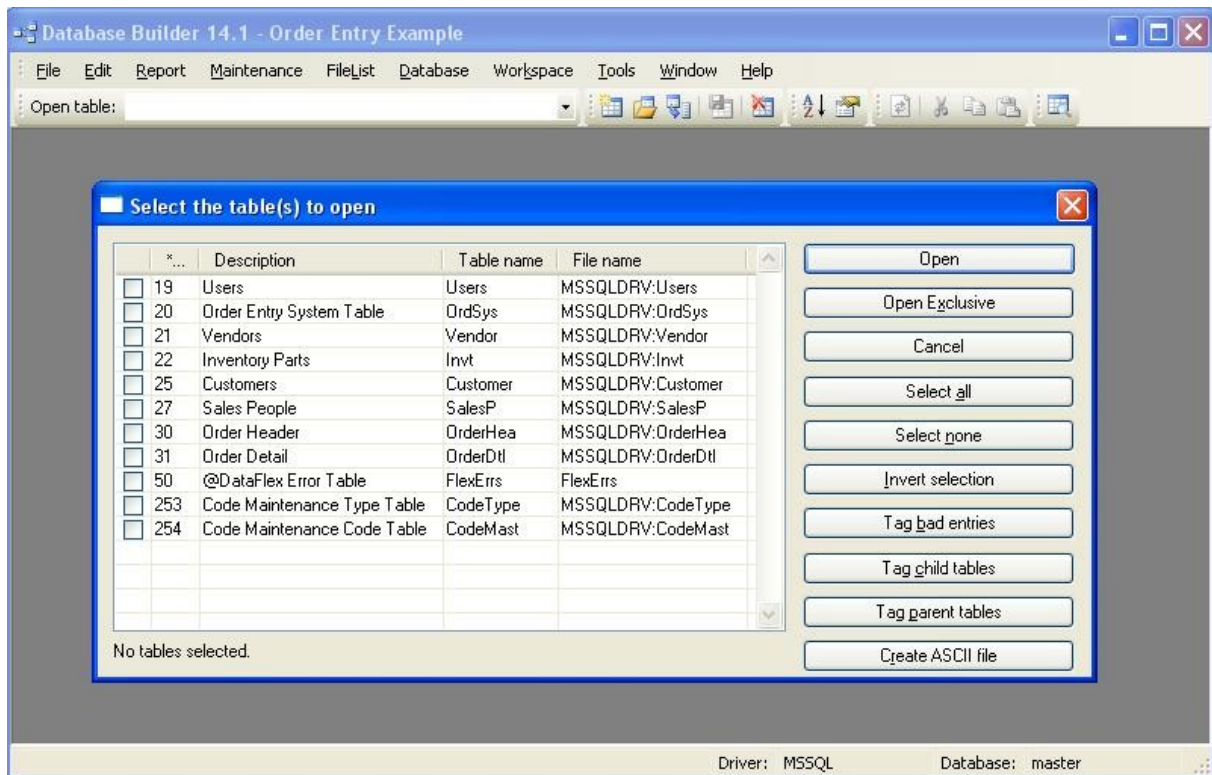
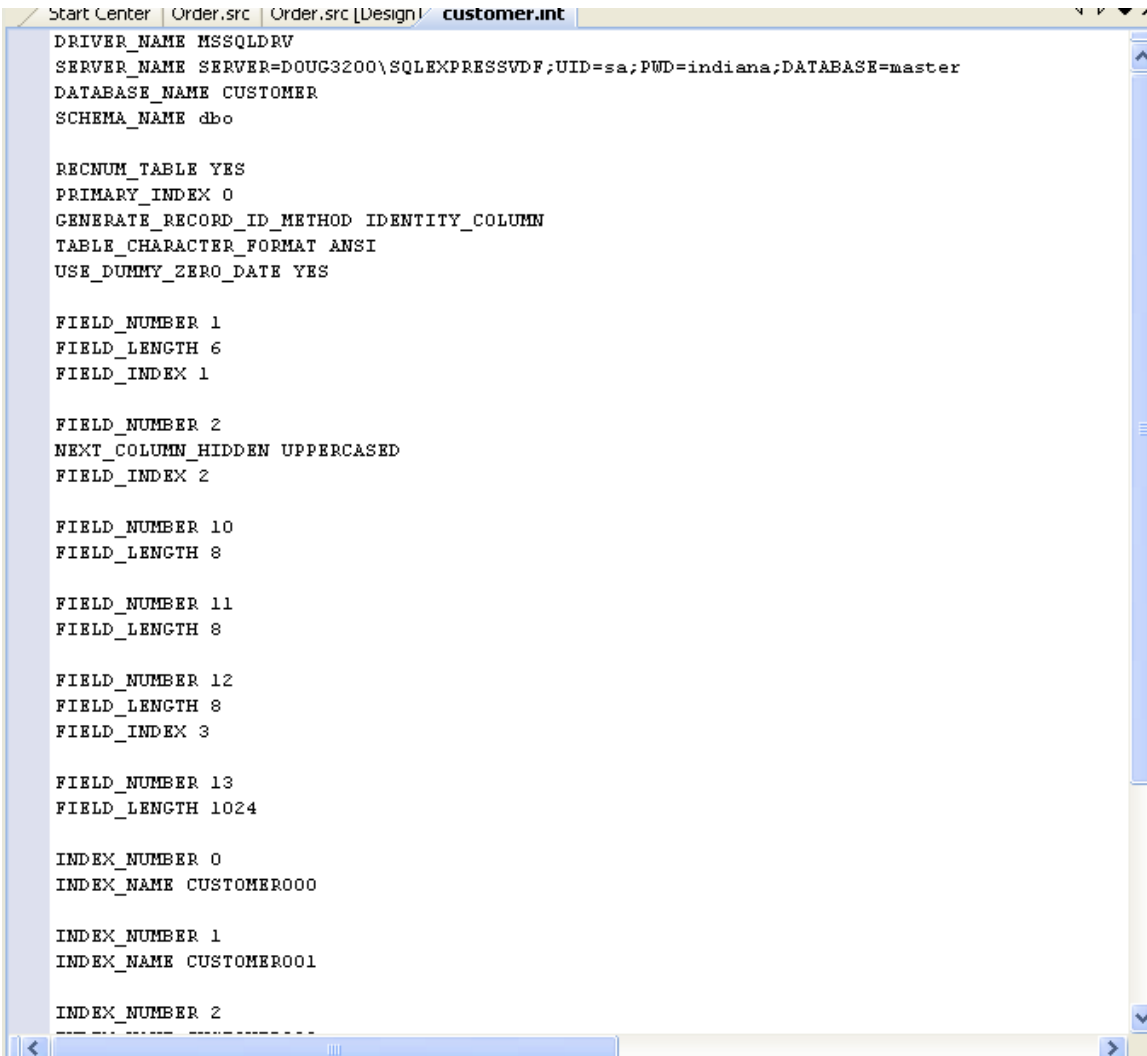


Figure 16

We can now close Database Builder and return to the Visual DataFlex Studio. If we now look in our data folder, we will see .INT files for each of our newly converted tables. If we open one (figure 17), we can see the conversion wizard filled in the information required to allow Visual DataFlex to connect to and open the new SQL tables. You will also see .CCH files (one per table). CCH files are created to allow the Visual DataFlex engine to quickly open non native DataFlex databases efficiently. They contain all of the table information the Visual DataFlex engine needs to properly open, read/write, and update data in those tables. CCH files are automatically maintained by Visual DataFlex as you make changes to tables via the Studio and Database Builder.



```

Start Center | Order.src | Order.src [Design] | customer.int
DRIVER_NAME MSSQLDRV
SERVER_NAME SERVER=DOUG3200\SQLEXPRESSVDF;UID=sa;PWD=indiana;DATABASE=master
DATABASE_NAME CUSTOMER
SCHEMA_NAME dbo

RECNUM_TABLE YES
PRIMARY_INDEX 0
GENERATE_RECORD_ID_METHOD IDENTITY_COLUMN
TABLE_CHARACTER_FORMAT ANSI
USE_DUMMY_ZERO_DATE YES

FIELD_NUMBER 1
FIELD_LENGTH 6
FIELD_INDEX 1

FIELD_NUMBER 2
NEXT_COLUMN_HIDDEN UPPERCASED
FIELD_INDEX 2

FIELD_NUMBER 10
FIELD_LENGTH 8

FIELD_NUMBER 11
FIELD_LENGTH 8

FIELD_NUMBER 12
FIELD_LENGTH 8
FIELD_INDEX 3

FIELD_NUMBER 13
FIELD_LENGTH 1024

INDEX_NUMBER 0
INDEX_NAME CUSTOMER000

INDEX_NUMBER 1
INDEX_NAME CUSTOMER001

INDEX_NUMBER 2
    
```

Figure 17

If we run the SQL Server Management Studio (installed in the Microsoft SQL Server 2008 folder), we can connect to our server and view its tables (figure 18).

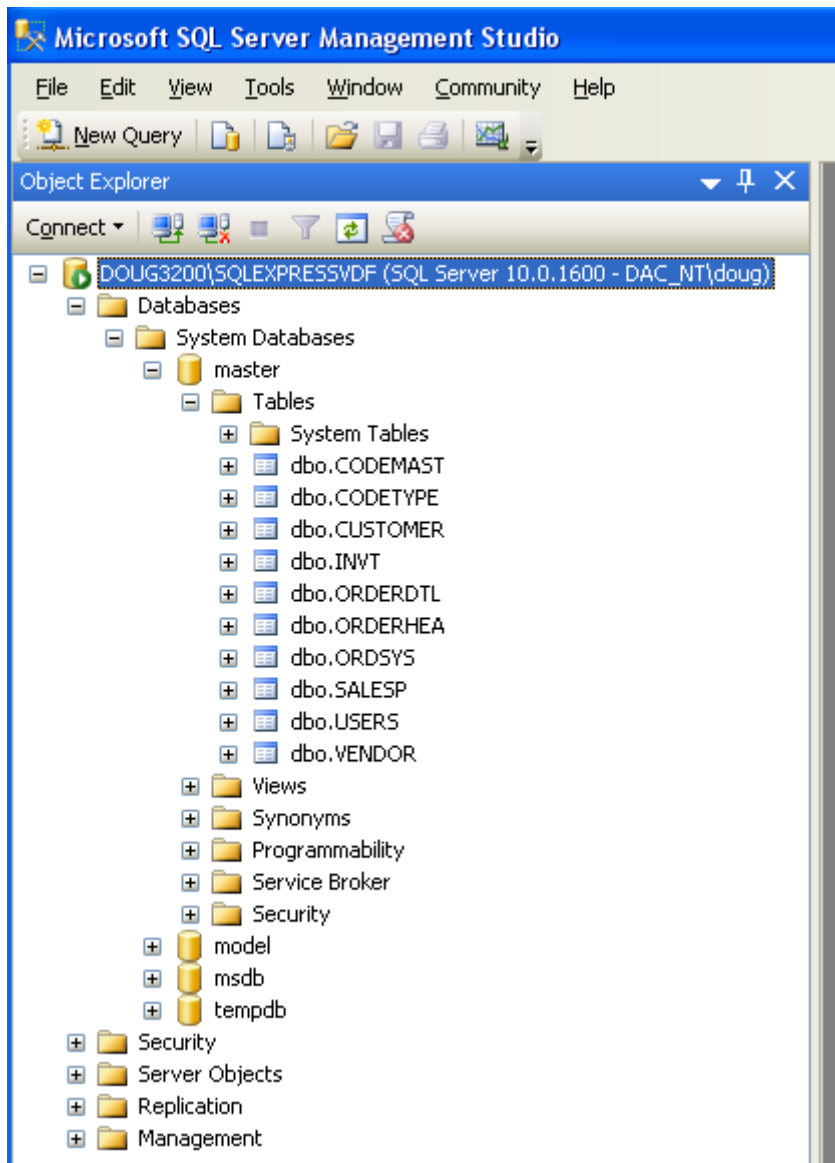


Figure 18

If we then expand the Customer table and expand Columns, we can see the SQL table mimics our original native DataFlex database table (figure 19).

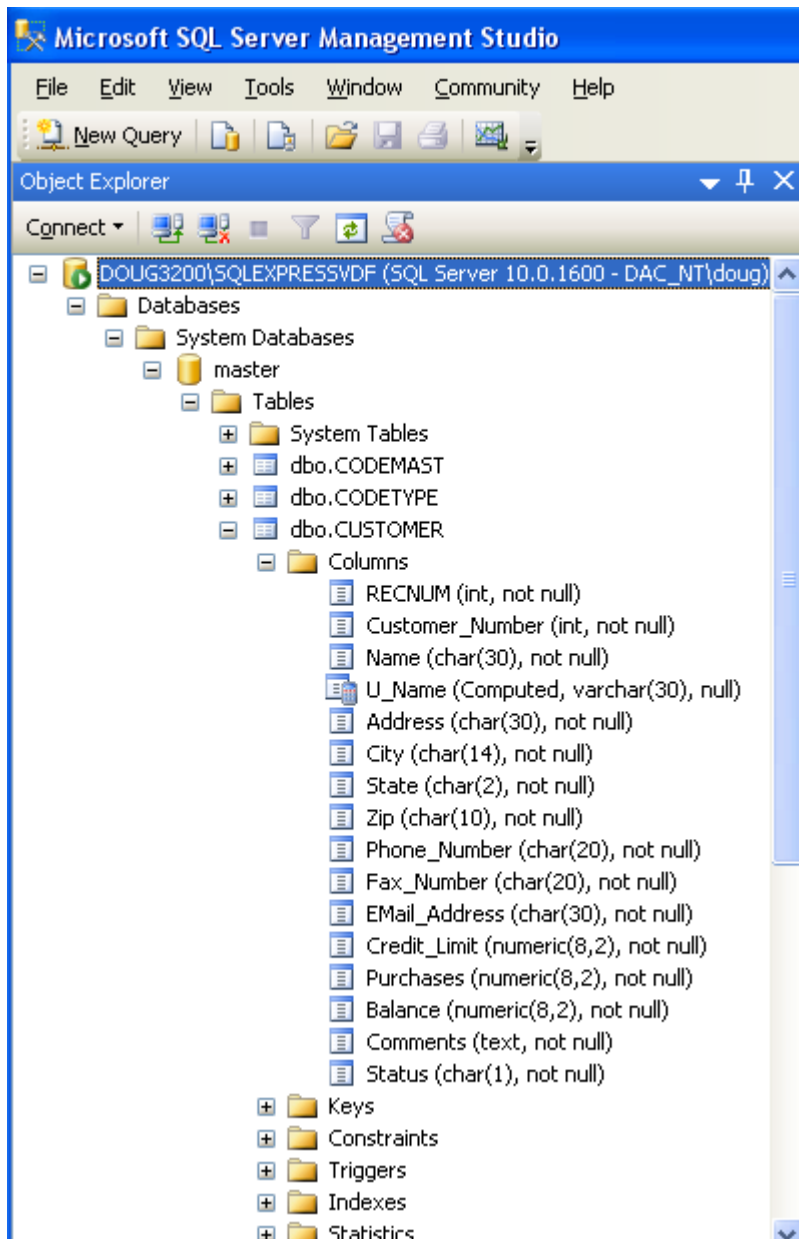


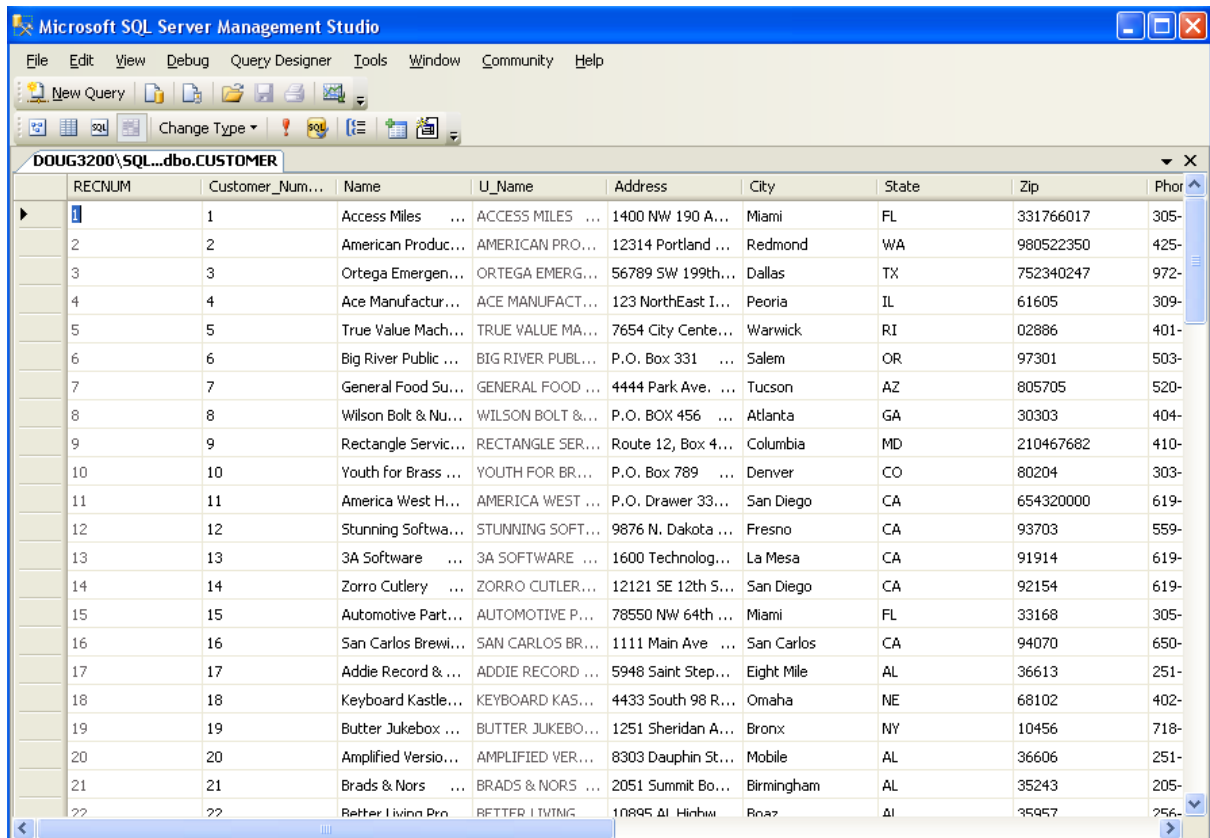
Figure 19

The data types from the DataFlex native database have been converted into Microsoft SQL Server data types.

DataFlex field type	Microsoft SQL Server field type
ASCII	Char
Numeric (with decimals)	Numeric
Numeric (no decimals)	Int
ASCII fields marked as case insensitive	Varchar
Date	DateTime

We also see the RECNUM field that it created for us. Also, there is a special field called U\_Name that it created to support the case insensitive indexing we had selected for the customer name field.

If we right click the customer table (dbo.customer) and choose Edit top 200 rows, we can see that the native SQL tool (SQL server Management Studio) can see our data just fine as will any other tool which can read native Microsoft SQL Server tables (see figure 20).



RECNUM	Customer_Num...	Name	U_Name	Address	City	State	Zip	Phor
1	1	Access Miles ...	ACCESS MILES ...	1400 NW 190 A...	Miami	FL	331766017	305-
2	2	American Produc...	AMERICAN PRO...	12314 Portland ...	Redmond	WA	980522350	425-
3	3	Ortega Emergen...	ORTEGA EMERG...	56789 SW 199th...	Dallas	TX	752340247	972-
4	4	Ace Manufactur...	ACE MANUFACT...	123 NorthEast I...	Peoria	IL	61605	309-
5	5	True Value Mach...	TRUE VALUE MA...	7654 City Cente...	Warwick	RI	02886	401-
6	6	Big River Public ...	BIG RIVER PUBL...	P.O. Box 331 ...	Salem	OR	97301	503-
7	7	General Food Su...	GENERAL FOOD ...	4444 Park Ave. ...	Tucson	AZ	805705	520-
8	8	Wilson Bolt & Nu...	WILSON BOLT &...	P.O. BOX 456 ...	Atlanta	GA	30303	404-
9	9	Rectangle Servic...	RECTANGLE SER...	Route 12, Box 4...	Columbia	MD	210467682	410-
10	10	Youth for Brass ...	YOUTH FOR BR...	P.O. Box 789 ...	Denver	CO	80204	303-
11	11	America West H...	AMERICA WEST ...	P.O. Drawer 33...	San Diego	CA	654320000	619-
12	12	Stunning Softwa...	STUNNING SOFT...	9876 N. Dakota ...	Fresno	CA	93703	559-
13	13	3A Software ...	3A SOFTWARE ...	1600 Technolog...	La Mesa	CA	91914	619-
14	14	Zorro Cutlery ...	ZORRO CUTLER...	12121 SE 12th S...	San Diego	CA	92154	619-
15	15	Automotive Part...	AUTOMOTIVE P...	78550 NW 64th ...	Miami	FL	33168	305-
16	16	San Carlos Brewi...	SAN CARLOS BR...	1111 Main Ave ...	San Carlos	CA	94070	650-
17	17	Addie Record & ...	ADDIE RECORD ...	5948 Saint Step...	Eight Mile	AL	36613	251-
18	18	Keyboard Kastle...	KEYBOARD KAS...	4433 South 98 R...	Omaha	NE	68102	402-
19	19	Butter Jukebox ...	BUTTER JUKEBO...	1251 Sheridan A...	Bronx	NY	10456	718-
20	20	Amplified Versio...	AMPLIFIED VER...	8303 Dauphin St...	Mobile	AL	36606	251-
21	21	Brads & Nors ...	BRADS & NORS ...	2051 Summit Bo...	Birmingham	AL	35243	205-
22	22	Better Living Pro	BETTER LIVING	10895 Al Highw...	Pratz	AL	35957	256-

Figure 20

Now we can run our application and see that it continues to work identically. That is the ultimate goal of the connectivity built into Visual DataFlex – the ability to seamlessly run your existing native database applications against SQL Server.

## Maintaining a migrated application

Now that our application data has been moved to Microsoft SQL server, there will be little or any difference in how we would maintain this application vs. one that uses the native DataFlex database. We can continue to use the Studio, Database Builder, Database Explorer, etc. These components are all able to work with our newly converted data files – even to the extent of adding/deleting fields and/or indexes.

The one area that you should explore is backup and recovery. There are built in facilities in Microsoft SQL Server to allow for transaction logging and data recovery. It is beyond the scope of this white paper to explore that topic, but one you should definitely invest time in. There are numerous backup products that inherently support backing up SQL servers on online databases, when users are using them. Your existing backup strategies will need to be updated to reflect the difference in how your data is now being stores.

## Conclusion

Moving to Microsoft SQL Server as you have read is not a daunting task. The tools and wizards built into Visual DataFlex make this task easy to accomplish. The net result of such a move is a more main stream database backend, ability for third party applications to work with your data, and an increase in data integrity and data restoration capabilities. You may find your application now has more perceived value due to a more industry standard implementation.

Keep in mind that you do not need to move all applications or even all customers to Microsoft SQL server. You can continue to maintain DataFlex native database versions of all of your applications and offer the SQL Server versions as needed (perhaps with increased cost as there is added value...) Think of Microsoft SQL Server as another tool on your tool belt, another way to enhance your Visual DataFlex applications.

For more information on Microsoft SQL Server connectivity and on connectivity in general, open up your Visual DataFlex help, click on the Contents tab, open the *Developing Database Applications* section. You will see a *Data Connectivity* section that has different articles available for a wide range of connectivity topics.

## Appendix A: Installing Microsoft SQL Server Express Edition

Figure 21 shows the first installation screen you will see. Select your language and which version of SQL Server Express 2008 to install (we will use the first option for this white paper).



Figure 21

When installing SQL Server, you may be asked to install a Windows Hotfix. If required, it would generally require a reboot as part of the Hotfix installation process.

You may also be asked to install the Windows PowerShell Software update as part of the Hotfix installation (figure 22).

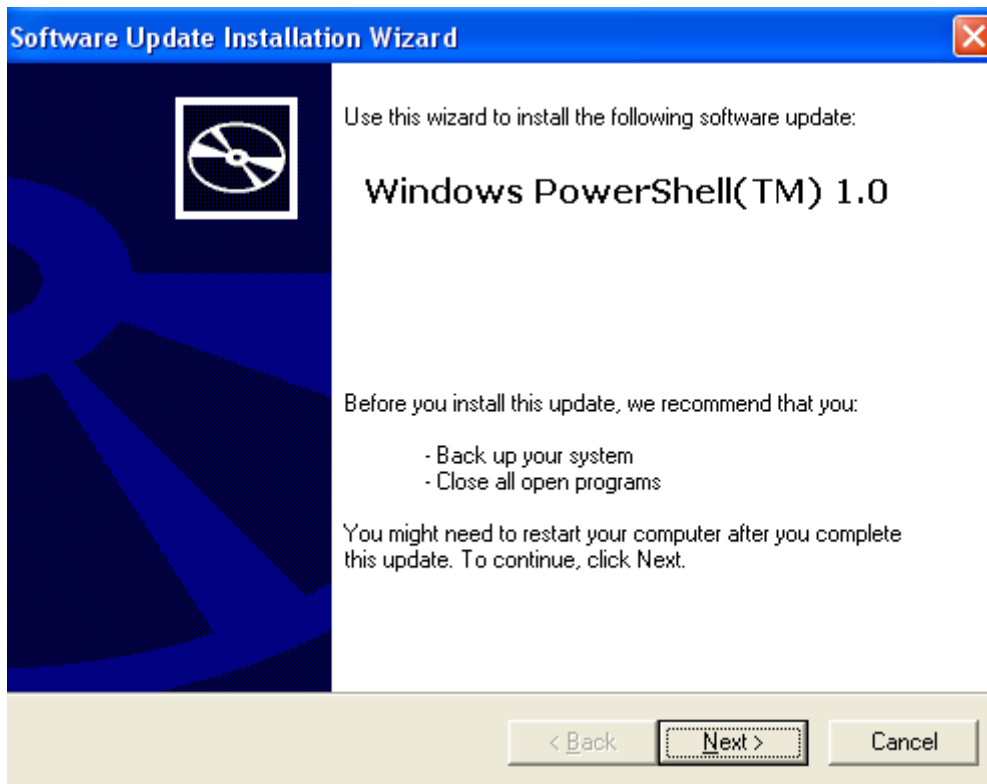


Figure 22

The following screen then appears, showing no installation issues to address (figure 23).

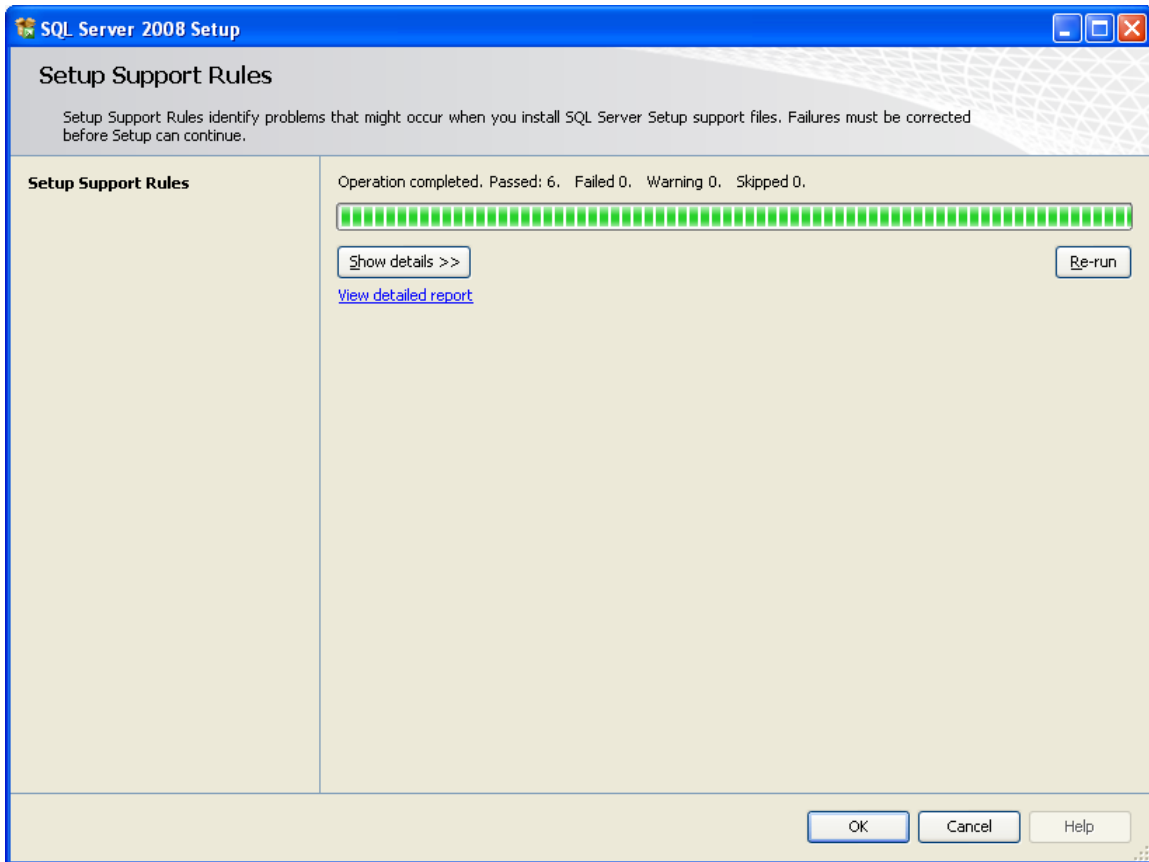


Figure 23

Next, a screen is displayed where you can enter an installation key, presuming you were installing an evaluation of a paid edition. If installing the Express Edition, there is no option to enter a key. (figure 24).

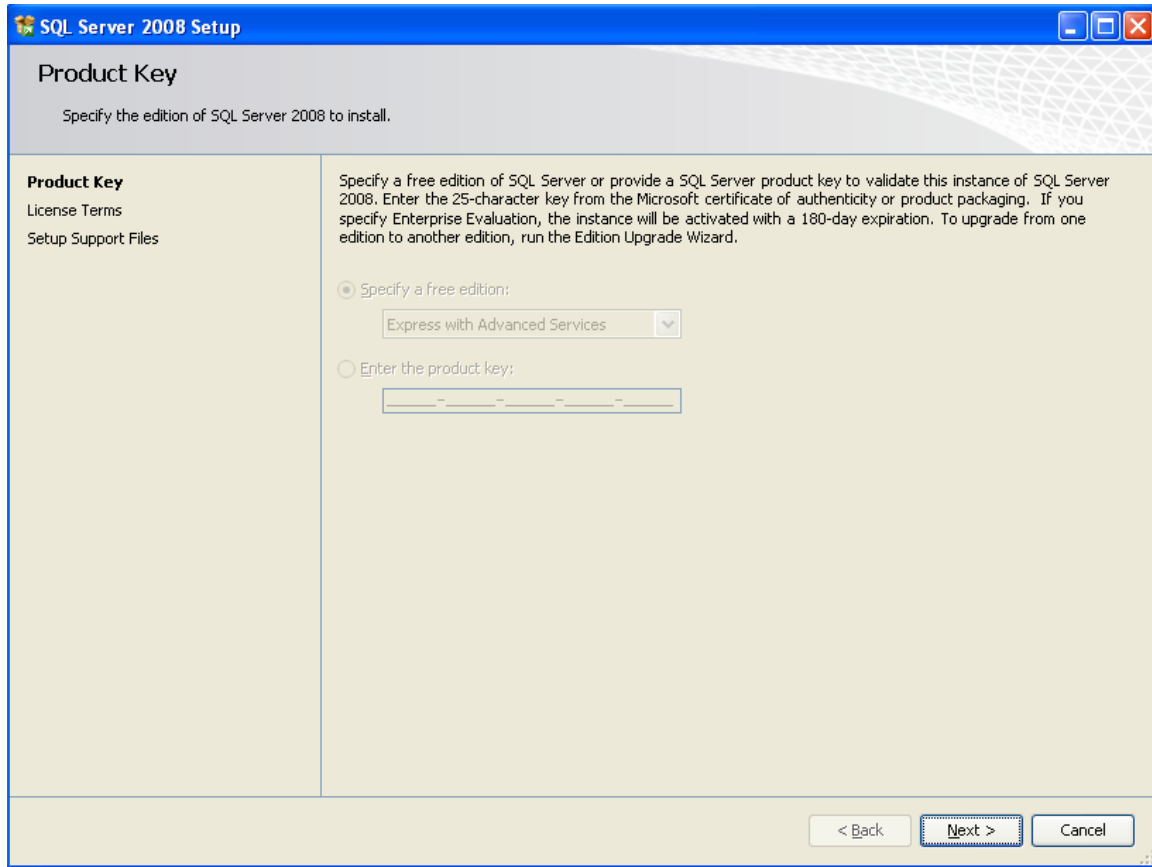


Figure 24

It will then ask you to accept the License Terms and Conditions.

The installation then continues with the support files (figure 25).

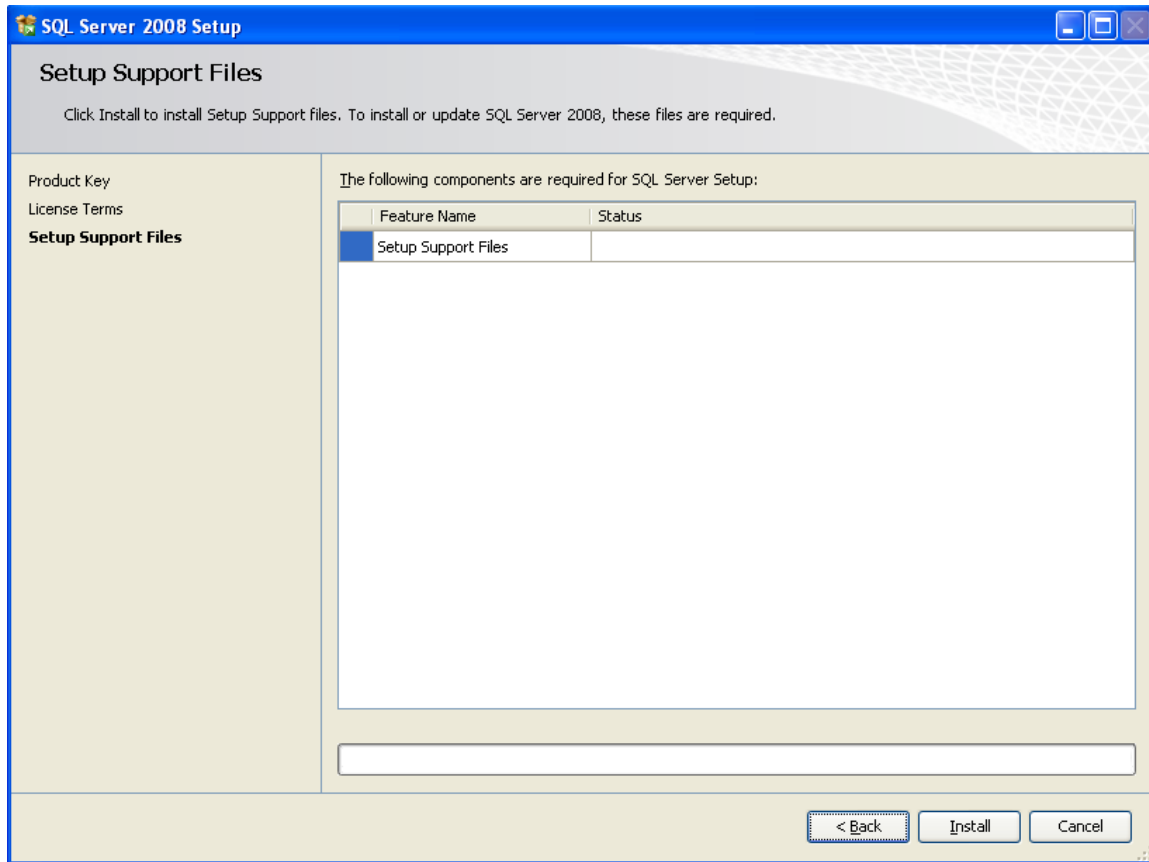


Figure 25

Another page then appears showing there are no problems to correct prior to continuation of the installation (figure 26).

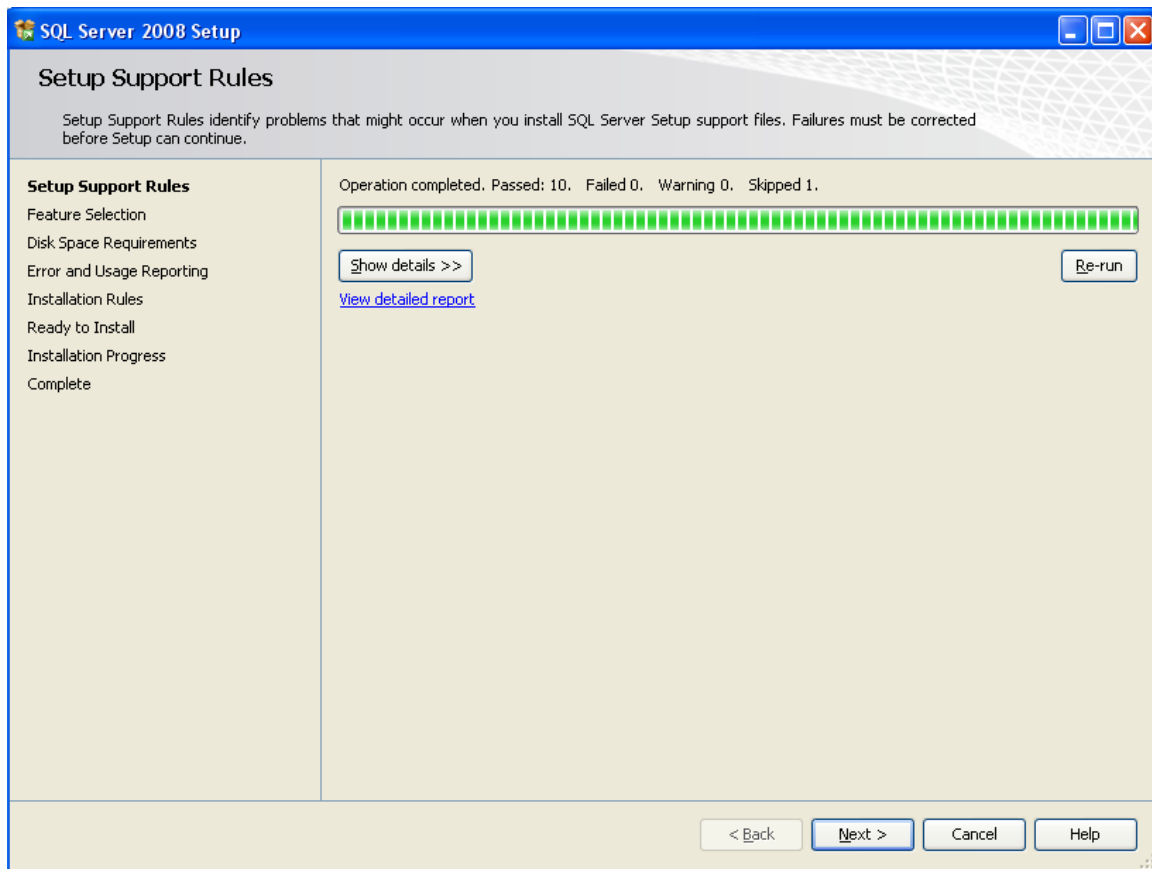


Figure 26

Next, it will ask you to select which features to install. We can just check them all to avoid ever having to run the installation again! (figure 27).

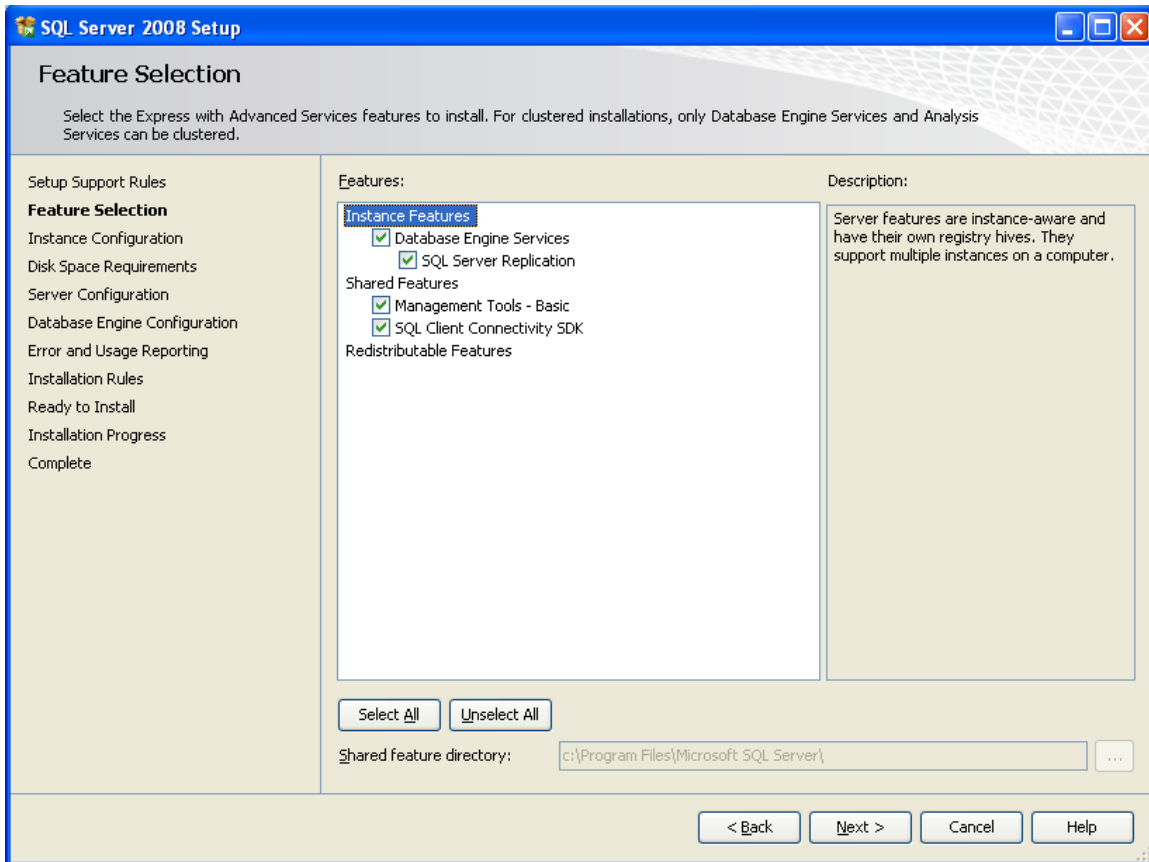


Figure 27

Next, it will ask you to fill in the instance name and choose a directory to install into (figure 28).

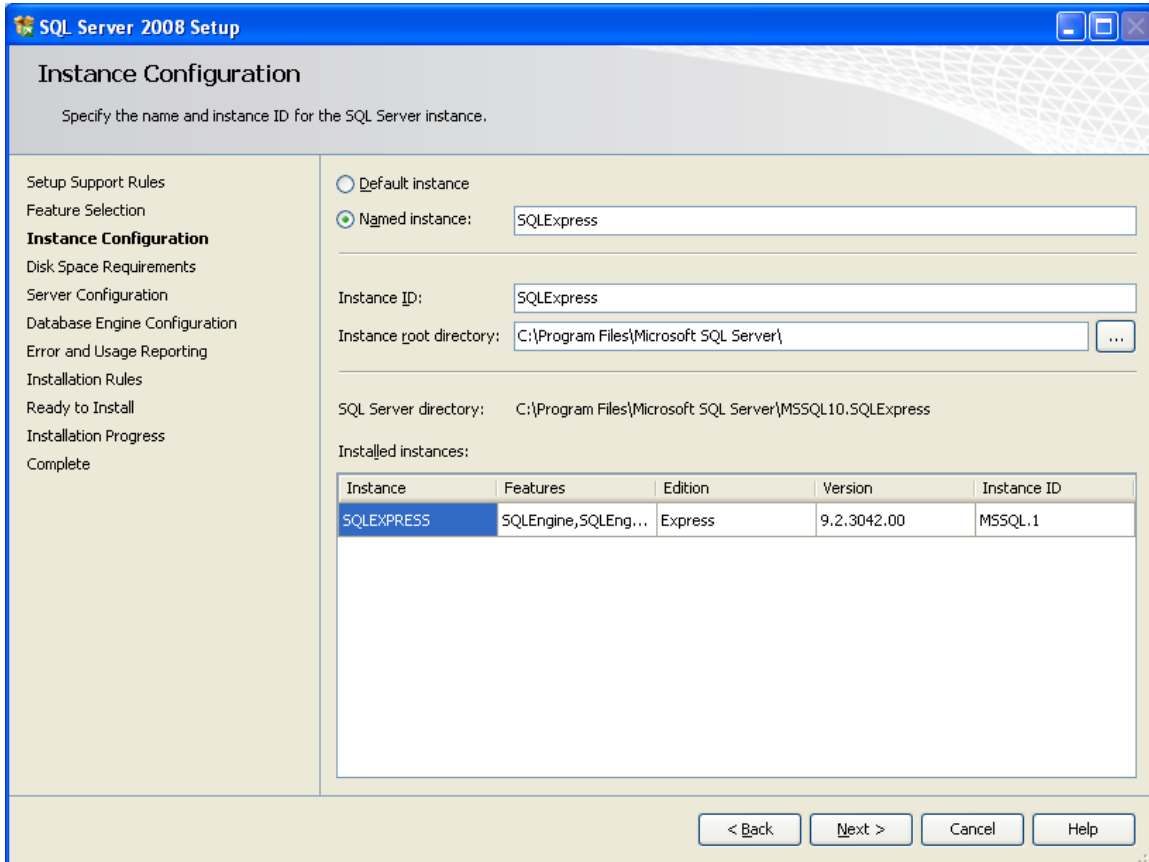


Figure 28

If you already have a previous installation of SQLExpress, you will need to change the instance name. For example, to SQLExpressVDF (figure 29).

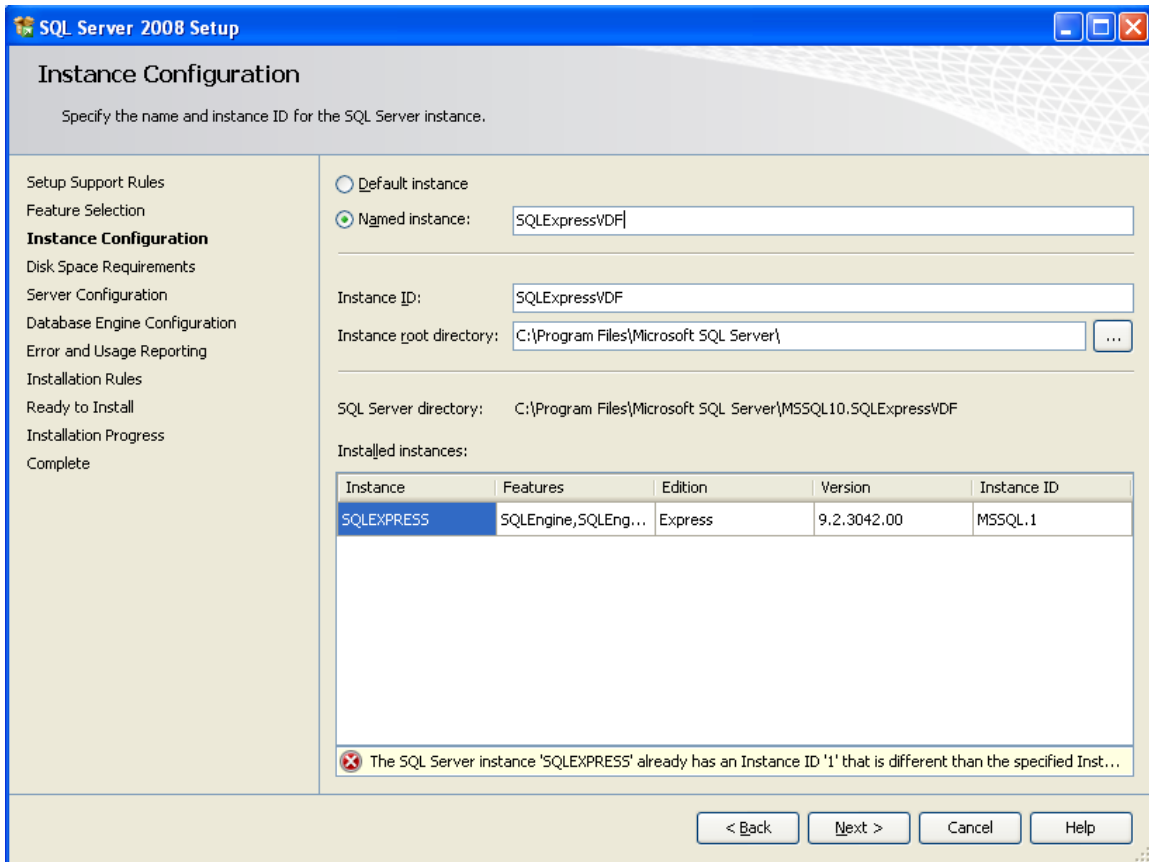


Figure 29

It then verifies there is enough disk space.

Next it asks you to choose the account name and password to use for the various SQL services (figure 30).

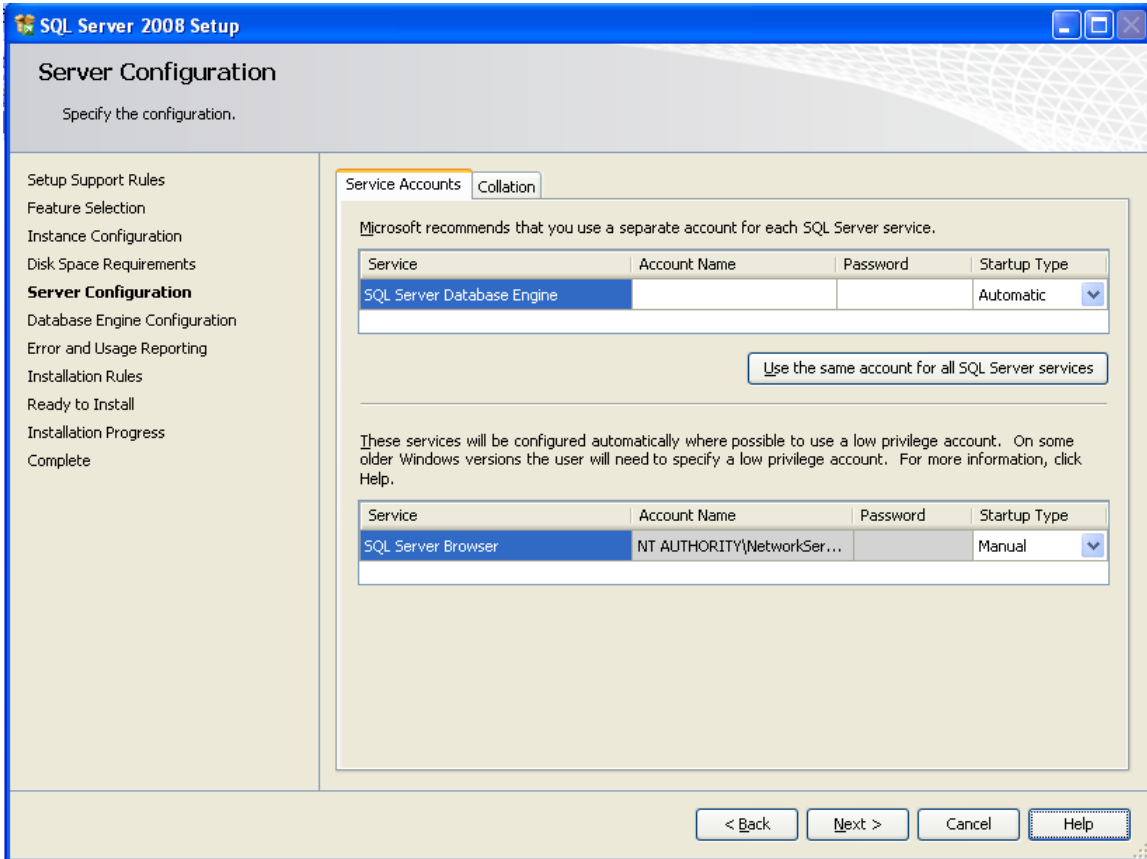


Figure 30

Next, it asks you to setup authentication information. If you choose mixed mode, you will be able to pass a username and password to the server to login as opposed to assigning permissions based on user login credentials. This actually works out well since you can assign the name and password to use when you convert your DataFlex native databases to Microsoft SQL Server (see figure 31).

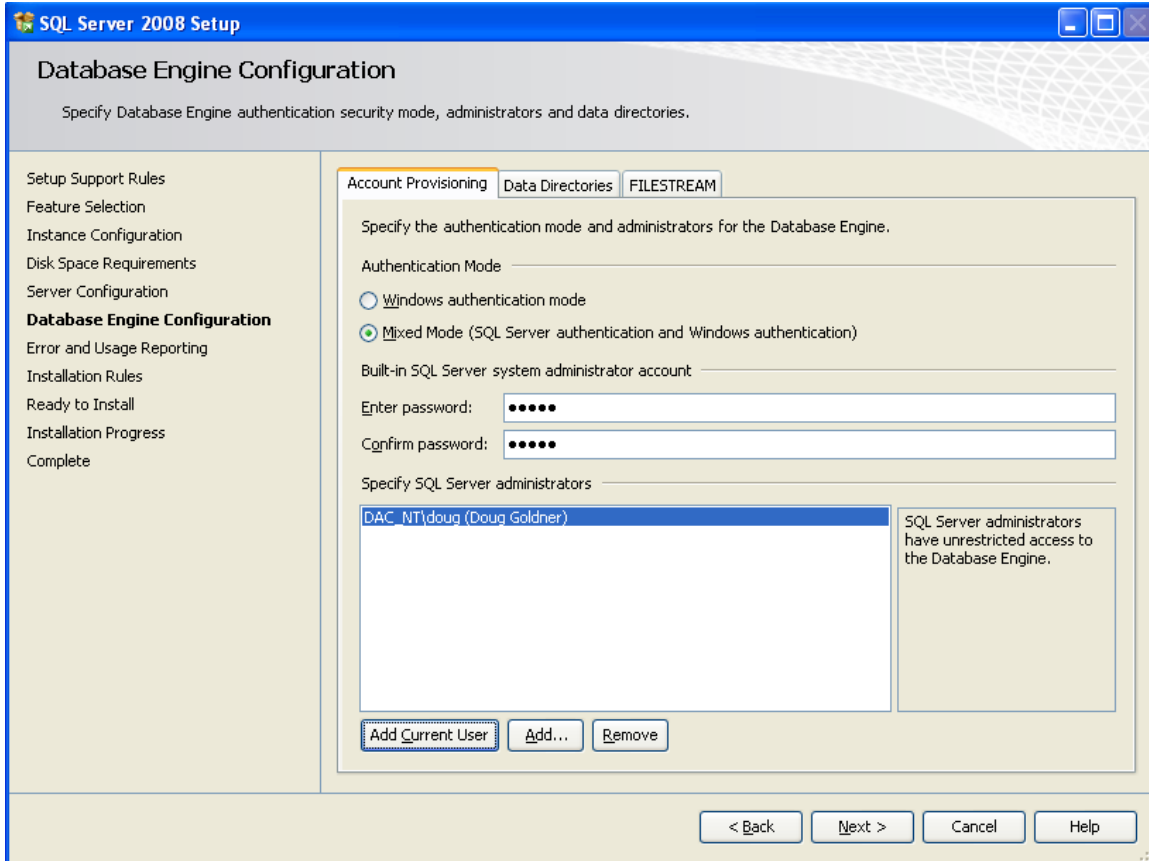


Figure 31

It then asks what reports to send to Microsoft to help them enhance the product. You can choose to not to have it automatically send any reports.

Then, it will again confirm no errors to handle and continued the installation.

One last screen shows all of the install options, and the installation began.

When finished, it will display a status screen showing that everything went OK.

If you now go into your control panel, then administrative tools, and click on services, you will see the new SQL Server service running (figure 32).

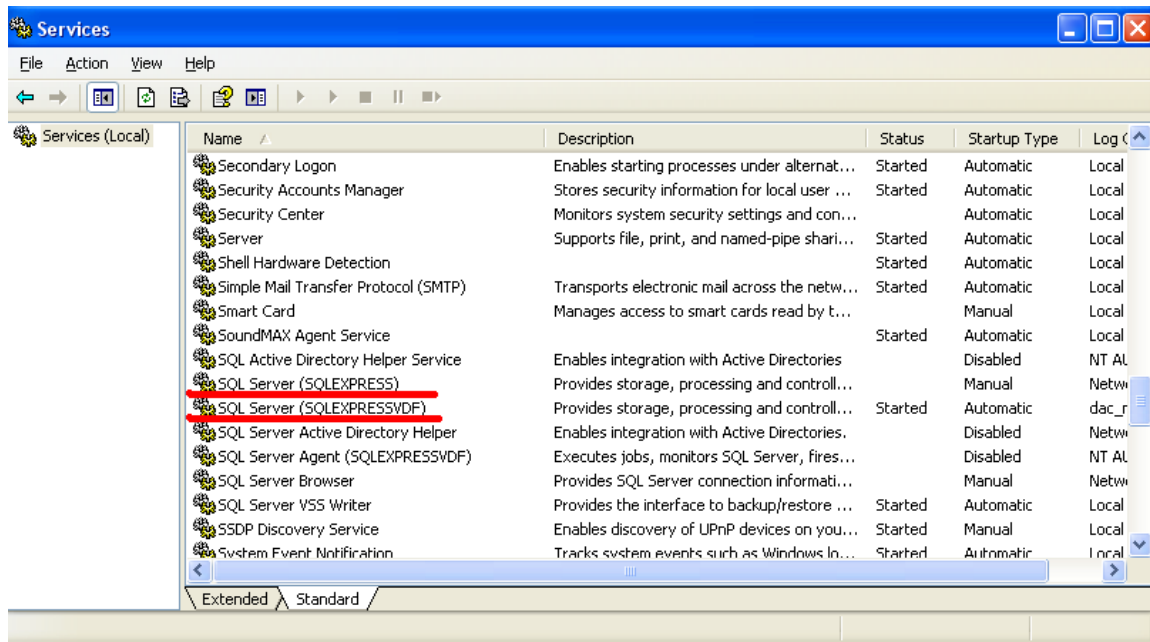


Figure 32

## Contact Us

### Data Access North America

14000 Southwest 119th Avenue  
Miami, Florida 33186  
U.S.A.

**Telephone:** +1-305-238-0012

**Fax:** +1-305-238-0017

**Toll Free:** +1-800-451-3539

**Email:** [sales@dataaccess.com](mailto:sales@dataaccess.com)

**Web:** [www.DataAccess.com](http://www.DataAccess.com)

### Data Access Asia Pacific

Shop 9, 246 Dorset Road,  
PO Box 873  
Boronia VIC 3155  
Australia

**Telephone:** +61 3 9761 3644

**Fax:** +61 3 9761 3655

**Email:** [CustomerService@DataAccess.com.au](mailto:CustomerService@DataAccess.com.au)

**Web:** [www.DataAccess.com.au](http://www.DataAccess.com.au)

### Data Access Brazil

Av.Paulista, 1776, 21st Floor  
Bela Vista  
01310-921 São Paulo  
Brazil

**Telephone:** (55) 11-3262-2000 tel

**Fax:** (55) 11-3284-1579 fax

**Email:** [Info@DataAccess.com.br](mailto:Info@DataAccess.com.br)

**Web:** [www.DataAccess.com.br](http://www.DataAccess.com.br) (Portuguese)

**Technical Website:** [www.4developers.com.br](http://www.4developers.com.br) (Portuguese)

### Data Access Europe

Lansinkesweg 4  
Hengelo (o) 7553 AE  
The Netherlands

**Telephone:** (31) 74-25-55-609 tel

**Fax:** (31) 74-25-03-466 fax

**Email:** [Info@DataAccess.nl](mailto:Info@DataAccess.nl)

**Web:** [www.DataAccess.nl](http://www.DataAccess.nl)

**Official Product Sites:**

Windows and Web applications

**VISUAL DATAFLEX**

[www.VisualDataFlex.com](http://www.VisualDataFlex.com)

Content Management System

**ELECTOS**

[www.Electos.com](http://www.Electos.com)

Business Intelligence

**DYNAMIC AI**

[www.DataAccess.com/Dynamic](http://www.DataAccess.com/Dynamic)