WHITE PAPER

Advantage Database Server

Migrating From Microsoft Access Jet Database Engine

Bill Todd, The Database Group, Inc.

SYBASE®
iAnywhere®

TABLE OF CONTENTS

INTRODUCTION

In the early to mid 1990's client/server technology was still expensive and complex so many business applications were written using desktop databases such as the Microsoft Access Jet database engine. The Microsoft Access Jet engine seemed like a good choice at the time because:

- It is relational
- It is multi-user
- It is free with Microsoft's development tools
- It is easy to deploy
- End users can view data or create ad hoc reports using Microsoft Access which is widely available as part of Microsoft Office

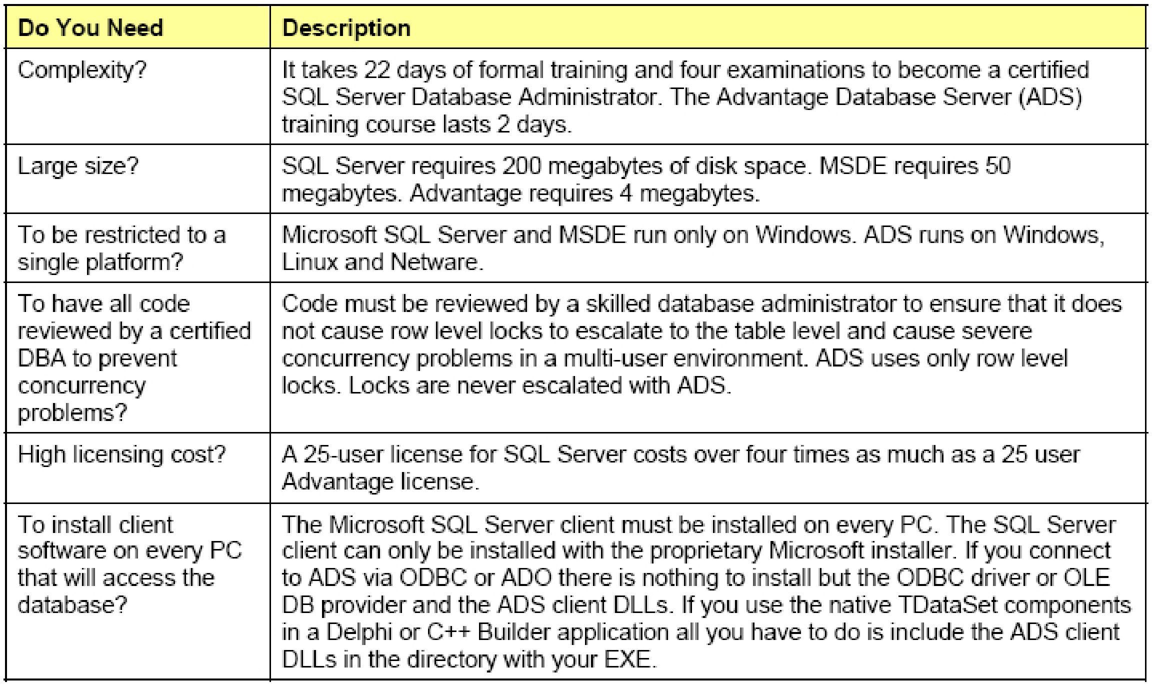
However, over time limitations began to emerge. The applications are not scalable. Performance deteriorates to an unacceptable level with more than 10 to 15 users. The total size of the database cannot exceed two gigabytes. Like all file server databases, the Microsoft Access database is vulnerable to corruption because every workstation writes directly to the database file. If any workstation crashes while writing to the database the database can become corrupt. The consequences of corruption are more severe with a Microsoft Access database than with most other desktop databases because all of the tables and indices are contained in a single file. If a corrupt file cannot be repaired the entire database is lost. Application development is more difficult than with modern databases that support stored procedures and triggers. Even if you have not encountered these problems you should be looking for a replacement because Microsoft has deprecated the Jet database engine.

NOTE: To download the complete source code for the application that accompanies this paper, please visit the Advantage Developer Zone (http://devzone.advantagedatabase.com), in the Documentation/Whitepapers section.

MICROSOFT'S SOLUTION

Microsoft's answer to these limitations is to move your applications to Microsoft SQL Server or to the Microsoft Desktop Engine (MSDE). Microsoft SQL server is a powerful, complex, enterprise database server. MSDE is SQL Server with a governor that causes performance to degrade rapidly with more than five users.

There is one significant problem with Microsoft's solution. Applications written for a simple desktop database like the Microsoft Access Jet engine do not need all of the complex features of an enterprise database server. The question is not whether your converted application will run on SQL Server. It will run and run well. The important question is, do you need everything that comes with SQL Server? To answer that question, examine the following table.



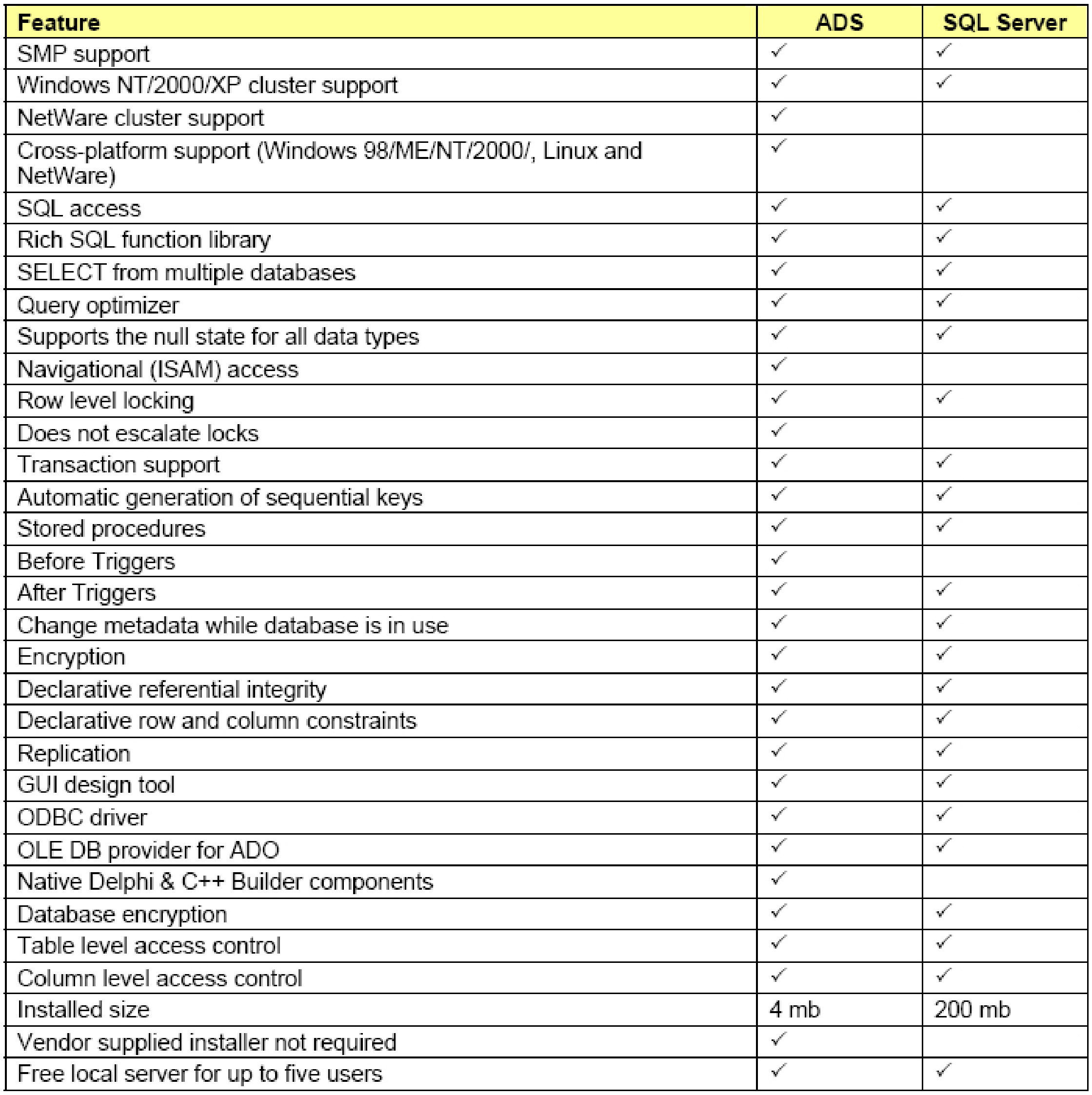
Do You Need	Description
Complexity?	It takes 22 days of formal training and four examinations to become a certified SQL Server Database Administrator. The Advantage Database Server (ADS) training course lasts 2 days.
Large size?	SQL Server requires 200 megabytes of disk space. MSDE requires 50 megabytes. Advantage requires 4 megabytes.
To be restricted to a single platform?	Microsoft SQL Server and MSDE run only on Windows. ADS runs on Windows, Linux and Netware.
To have all code reviewed by a certified DBA to prevent concurrency problems?	Code must be reviewed by a skilled database administrator to ensure that it does not cause row level locks to escalate to the table level and cause severe concurrency problems in a multi-user environment. ADS uses only row level locks. Locks are never escalated with ADS.
High licensing cost?	A 25-user license for SQL Server costs over four times as much as a 25 user Advantage license.
To install client software on every PC that will access the database?	The Microsoft SQL Server client must be installed on every PC. The SQL Server client can only be installed with the proprietary Microsoft installer. If you connect to ADS via ODBC or ADO there is nothing to install but the ODBC driver or OLE DB provider and the ADS client DLLs. If you use the native TDataSet components in a Delphi or C++ Builder application all you have to do is include the ADS client DLLs in the directory with your EXE.

The simple truth is that using a complex enterprise database management system for an application that does not need it results in unnecessary licensing, training, development, deployment, administration and support costs.

IS ADVANTAGE DATABASE SERVER A BETTER CHOICE?

If you need a database server that supports over 500 concurrent users, large datasets (over 100 gigabytes), requires no administration, is easy to learn and easy to deploy then Advantage Database Server (ADS) is right for you. ADS was designed for embedded and distributed applications where high performance, reliability and zero administration are the key requirements. ADS combines these qualities with a rich feature set that gives you all the tools you need to tackle any database project.

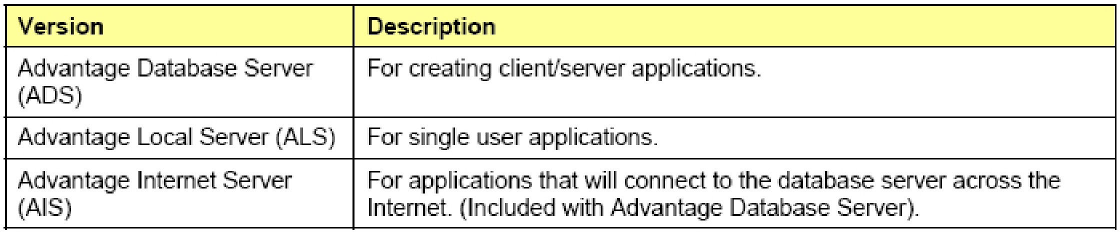
The following table compares features offered by ADS and SQL Server. This list is not exhaustive. It includes the features that are important for embedded, distributed and workgroup database applications and ignores features that are of interest only at the enterprise level.



Feature	ADS	SQL Server
SMP support	✓	✓
Windows NT/2000/XP cluster support	✓	✓
NetWare cluster support	✓	
Cross-platform support (Windows 98/ME/NT/2000/, Linux and NetWare)	✓	
SQL access	✓	✓
Rich SQL function library	✓	✓
SELECT from multiple databases	✓	✓
Query optimizer	✓	✓
Supports the null state for all data types	✓	✓
Navigational (ISAM) access	✓	
Row level locking	✓	✓
Does not escalate locks	✓	
Transaction support	✓	✓
Automatic generation of sequential keys	✓	✓
Stored procedures	✓	✓
Before Triggers	✓	
After Triggers	✓	✓
Change metadata while database is in use	✓	✓
Encryption	✓	✓
Declarative referential integrity	✓	✓
Declarative row and column constraints	✓	✓
Replication	✓	✓
GUI design tool	✓	✓
ODBC driver	✓	✓
OLE DB provider for ADO	✓	✓
Native Delphi & C++ Builder components	✓	
Database encryption	✓	✓
Table level access control	✓	✓
Column level access control	✓	✓
Installed size	4 mb	200 mb
Vendor supplied installer not required	✓	
Free local server for up to five users	✓	✓

Choosing the Right Version of Advantage

Advantage comes in three versions. The information in the following table will help you pick the one that is right for your application.



Version	Description
Advantage Database Server (ADS)	For creating client/server applications.
Advantage Local Server (ALS)	For single user applications.
Advantage Internet Server (AIS)	For applications that will connect to the database server across the Internet. (Included with Advantage Database Server).

You can choose which version or versions of Advantage your application will connect to by setting a property and using the correct syntax to specify the path to the database. You will see examples in the sections of this paper that describe converting each sample application.

CONVERTING YOUR APPLICATIONS

Although converting from the Microsoft Access Jet engine to ADS is remarkably easy there are several steps you must perform. Some of these steps are common to all development environments while others are specific to a particular development tool. There are three tasks common to all development environments.

- Resolving SQL dialect differences
- Adapting to differences in database features
- Converting your data

After discussing these common tasks this paper takes a detailed look at converting three types of applications.

- A Visual Basic 6.0 application that uses ADO to connect to an Access database
- A Delphi application using ADO to connect to an Access database
- A Delphi application using the BDE native Microsoft Access driver to connect to an Access database

There is also a fourth conversion option; moving your Visual Basic or Delphi application to Microsoft's .NET environment.

Moving to .NET

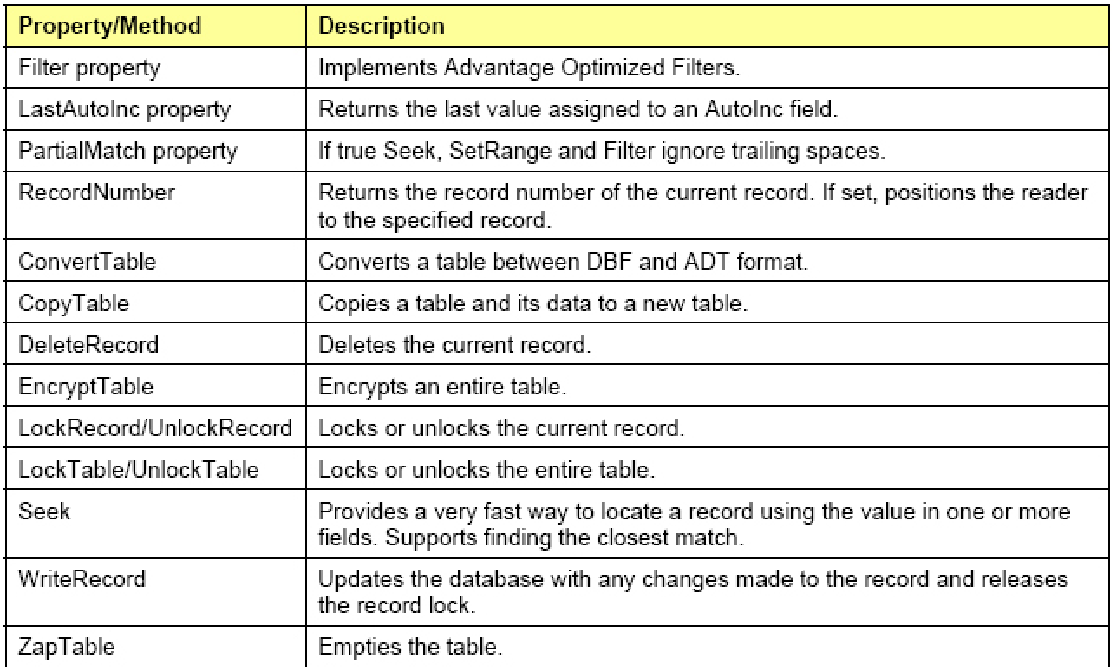
If your application is written in Visual Basic moving to Visual Basic for .NET requires major changes to your code. Alternatively, moving a Delphi application to Delphi for .NET is much easier because most of the Visual Component Library has been ported to .NET. A detailed look at these conversions is beyond the scope of this paper but, in both cases, moving from the Microsoft Access Jet database to Advantage is easy thanks to the Advantage ADO.NET data provider. The Advantage Data Provider includes the following components.

- Advantage Connection
- Advantage DataAdapter
- Advantage DataReader
- Advantage ExtendedDataReader
- Borland Data Provider (BDP) data source

The Advantage Connection, Advantage DataAdapter and Advantage DataReader provide the same functions as the corresponding components in other .NET data providers. The connection string for an Advantage database can include the following Advantage specific options.

- Server Type
- Table Type
- Encryption Password
- TrimTrailingSpaces
- Compression
- CharType
- LockMode
- SecurityMode

The Advantage ExtendedDataReader provides features unique to Advantage that are not found in any other .NET data provider. The following table describes some of these features. The ExtendedDataReader provides traditional ISAM access to a table and makes it much easier to migrate applications that use ISAM databases to .NET.



Property/Method	Description
Filter property	Implements Advantage Optimized Filters.
LastAutoInc property	Returns the last value assigned to an AutoInc field.
PartialMatch property	If true Seek, SetRange and Filter ignore trailing spaces.
RecordNumber	Returns the record number of the current record. If set, positions the reader to the specified record.
ConvertTable	Converts a table between DBF and ADT format.
CopyTable	Copies a table and its data to a new table.
DeleteRecord	Deletes the current record.
EncryptTable	Encrypts an entire table.
LockRecord/UnlockRecord	Locks or unlocks the current record.
LockTable/UnlockTable	Locks or unlocks the entire table.
Seek	Provides a very fast way to locate a record using the value in one or more fields. Supports finding the closest match.
WriteRecord	Updates the database with any changes made to the record and releases the record lock.
ZapTable	Empties the table.

In C# Builder and Delphi 8 for .NET Borland introduced the Borland Data Provider (BDP) architecture. BDP varies from Microsoft's .NET data provider architecture in that it lets you easily change databases without changing the components or code in your application. The Advantage Data Provider includes an Advantage provider for BDP. If you are using C# Builder or Delphi 8 for .NET you have the option to install the BDP provider when you install the Advantage ADO.NET Data Provider.

Conversion Issues Common to All Environments

There are several differences between the Microsoft Access Jet database and Advantage that you may encounter regardless of the language your application is written in and regardless of the data access layer that you use. This paper includes detailed instructions for converting applications that use three different combinations of programming language and data access technology. After you finish the steps for your specific environment you must also examine the issues described in the following sections to see if they apply to your application.

SQL Dialect Differences

There are several differences between the Microsoft Jet SQL dialect and Advantage SQL. Most of these differences exist because Microsoft Jet SQL does not follow the ANSI SQL standard. This means that you would encounter the same problems converting from the Microsoft Access Jet engine to any standard SQL database.

Microsoft Access uses * and ? for Wildcards

When using the SQL LIKE operator, Microsoft Jet SQL uses the asterisk (*) wildcard to represent a string of zero or more characters. Advantage uses the ANSI standard % wildcard. Microsoft Jet SQL uses the question mark (?) wildcard to represent any single character while Advantage uses the ANSI standard underscore (_). For example, using Microsoft Jet SQL you might write:

```
SELECT * FROM CUSTOMER
WHERE COMPANYNAME LIKE 'E*'
```

to get a list of the companies whose name starts with the letter E. In Advantage SQL the statement would be:

```
SELECT * FROM CUSTOMER
WHERE COMPANYNAME LIKE ‘E%’
```

To find all of the customers where the ContactTitle column contains a value that is five characters long with Microsoft Jet SQL you might use:

```
SELECT * FROM CUSTOMER
WHERE CONTACTTITLE LIKE ‘?????’
```

but with Advantage SQL the query would be

```
SELECT * FROM CUSTOMER
WHERE CONTACTTITLE LIKE ‘_____’
```

Advantage also offers a faster more powerful alternative to LIKE through its server side full text search engine. With a full text search index you can use the CONTAINS function to specify complex search conditions using the AND, OR, NOT and NEAR operators. For a detailed description see Full Text Search in the Advantage on-line help file.

Microsoft Access uses DISTINCTROW to Eliminate Duplicate Rows

To list the unique countries in which you have customers, using Microsoft Jet SQL, you would use:

```
SELECT DISTINCTROW COUNTRY FROM CUSTOMER
```

To get the same result with Advantage SQL use the ANSI standard DISTINCT operator:

```
SELECT DISTINCT COUNTRY FROM CUSTOMER
```

Microsoft Access Allows both Single and Double Quotation Marks to Delimit String Literals

Using Microsoft Jet SQL both of these statements will produce the same result.

```
SELECT * FROM CUSTOMER
WHERE [CUSTOMER ID] = ‘VICTE’

SELECT * FROM CUSTOMER
WHERE [CUSTOMER ID] = “VICTE”
```

Microsoft Jet SQL allows you to use either single or double quotation marks to delimit string literals. Microsoft Jet SQL uses square brackets to enclose delimited identifiers, such as table or column names that include spaces.

Advantage SQL follows the ANSI standard and uses single quotation marks to enclose string literals and double quotation marks to enclose delimited identifiers. However, Advantage SQL also allows you to use square brackets to enclose delimited identifiers so the first query above will run in Advantage without change. To make the second query run you must replace the double quotation marks giving:

```
SELECT * FROM CUSTOMER
WHERE [CUSTOMER ID] = ‘VICTE’
```

Microsoft Jet SQL Includes the VAR and STDEV Aggregate Function

Microsoft Jet SQL includes the VAR aggregate to calculate variance of a column and the STDEV aggregate to calculate standard deviation of a column. These functions are not available in Advantage SQL.

Microsoft Jet SQL Supports Full and Right Joins

Microsoft Jet SQL includes support for both FULL and RIGHT outer joins. Advantage SQL supports LEFT outer joins and INNER joins only, however, it is not difficult to rewrite a query that uses a right or full outer join in Advantage SQL.

Queries that use right outer joins are particularly easy. Consider the following query that returns all customers and the ship date of all orders. Customers with no orders are included in the result.

```
SELECT Customers.CustomerID, Customers.CompanyName, Orders.ShippedDate
FROM Orders RIGHT OUTER JOIN Customers ON Customers.CustomerID = Orders.CustomerID;
```

To get the same result in Advantage SQL just reverse the order of the tables in the FROM clause then change RIGHT OUTER JOIN to LEFT OUTER JOIN as shown below.

```
SELECT Customers.CustomerID, Customers.CompanyName, Orders.ShippedDate
FROM Customers LEFT OUTER JOIN Orders ON Customers.CustomerID = Orders.CustomerID;
```

Converting a FULL OUTER JOIN query such as

```
SELECT Customers.CustomerID, Customers.CompanyName, Orders.ShippedDate
FROM Customers FULL OUTER JOIN Orders ON Customers.CustomerID = Orders.CustomerID;
```

is only slightly more complex. The following statement does the job using UNION ALL.

```
SELECT Customers.CustomerID, Customers.CompanyName, Orders.ShippedDate
FROM Customers LEFT OUTER JOIN Orders ON Customers.CustomerID = Orders.CustomerID UNION ALL
SELECT Customers.CustomerID, Customers.CompanyName, Orders.ShippedDate
FROM Orders LEFT OUTER JOIN Customers ON Customers.CustomerID = Orders.CustomerID WHERE Customers.CustomerID IS NULL
```

This query combines the result sets from two SELECT statements. The first SELECT returns a row if there are matching records in both the Customers and Orders tables or if there is a row in Customers with no matching records in Orders. A full outer join would also return a row if there is a row in Orders with no matching row in Customers and this is exactly what the second SELECT does.

Advantage LEFT OUTER JOIN queries produce static result sets so you cannot edit the result set directly. The situations that cause static queries are outlined in the Advantage Help file. Some of the operations that cause static result sets are: Left Outer Joins, expressions in the select list (i.e., LastName + FirstName as FullName), DISTINCT queries and GROUP BY or HAVING clauses.

Database Feature Differences

By default Advantage ignores filters when counting records to improve performance. This means that SELECT statements that include a WHERE clause may not return accurate record counts. There are two ways to get an accurate record count with Advantage. The first is to tell Advantage to count the records in the filtered dataset. If you are using ADO add FilterOptions=RESPECT_WHEN_COUNTING to your connection string. If you are using Delphi, Kylix or C++ Builder and the TAdsQuery component click the TAdsQuery component to select it. In the Object Inspector, click the plus sign to expand the AdsTableOptions property. Find the AdsFilterOptons property and set it to RESPECT_WHEN_COUNTING. While this method will provide an accurate record count it may cause poor performance on large tables.

The second solution is to create an index on all of the fields in the WHERE clause. Advantage will use the index to determine an accurate record count regardless of the FilterOptions setting.

In Access Jet database engine applications it is common to edit a query’s result set. Advantage SELECT statements that include a left outer join, an expression in the SELECT list, the DISTINCT keyword, a GROUP BY clause or a HAVING clause return a static result set that cannot be edited. To modify the result set, a separate INSERT or UPDATE statement should be executed against the underlying table(s).

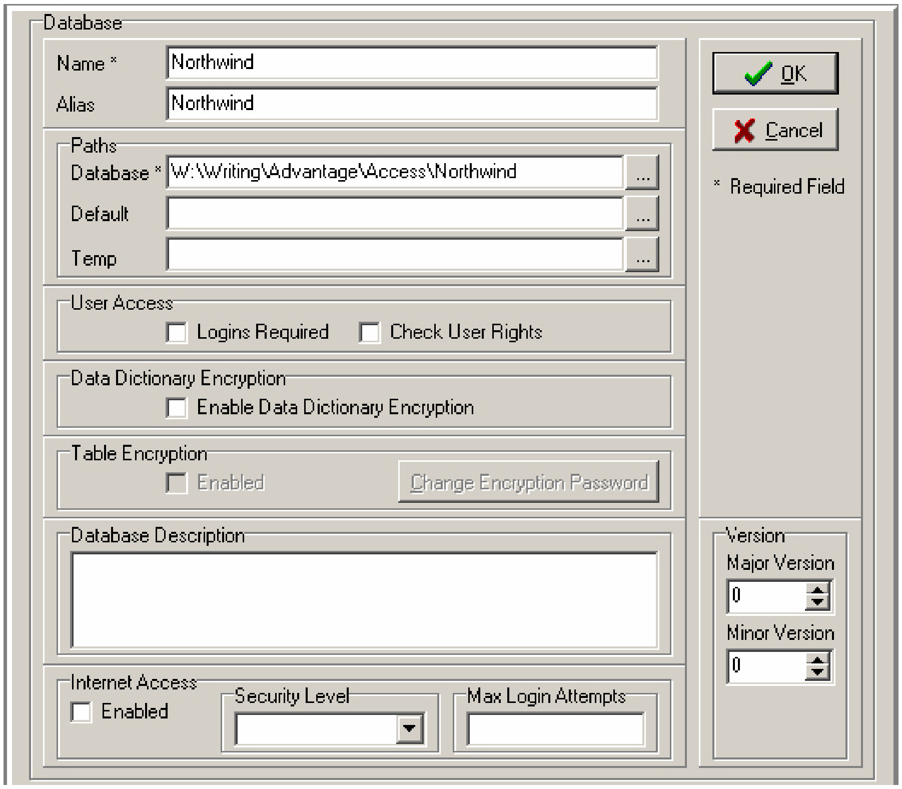
When using ADO with a server-side cursor Advantage ignores the Cursor Type setting and returns a cursor based on the query. If the user requests a Dynamic or Keyset cursor and the query results in a static cursor then that is what is returned by ADS. One exception to this rule is the adOpenForwardOnly cursor type. Support for this cursor type was added with version 6.2. Client-side cursors run entirely on the client and the cursor type you specify is the type you will get.

Access Text fields are case insensitive while the Advantage CHARACTER data type is case sensitive. If you import your data using the import utility in Advantage Data Architect click the option to import strings as case insensitive, as described in the next section. If you create your Advantage tables first then import the data using a custom program use the CICHARACTER data type for all string fields that you want to be case insensitive.

CONVERTING YOUR DATA

The first step in converting your application to ADS is to convert your database so you will have data to test with. In this example you will use the Advantage Data Architect to convert the sample Northwind database that comes with Microsoft Access.

1. Open Advantage Data Architect.
2. Choose Database | New Database from the menu to display the dialog shown below.



Database
Name * Northwind
Alias Northwind
Paths
Database * W:\Writing\Advantage\Access\Northwind
Default
Temp
User Access
☐ Logins Required ☐ Check User Rights
Data Dictionary Encryption
☐ Enable Data Dictionary Encryption
Table Encryption
☐ Enabled
Database Description
Internet Access
☐ Enabled
Security Level
Max Login Attempts
Version
Major Version 0
Minor Version 0
* Required Field

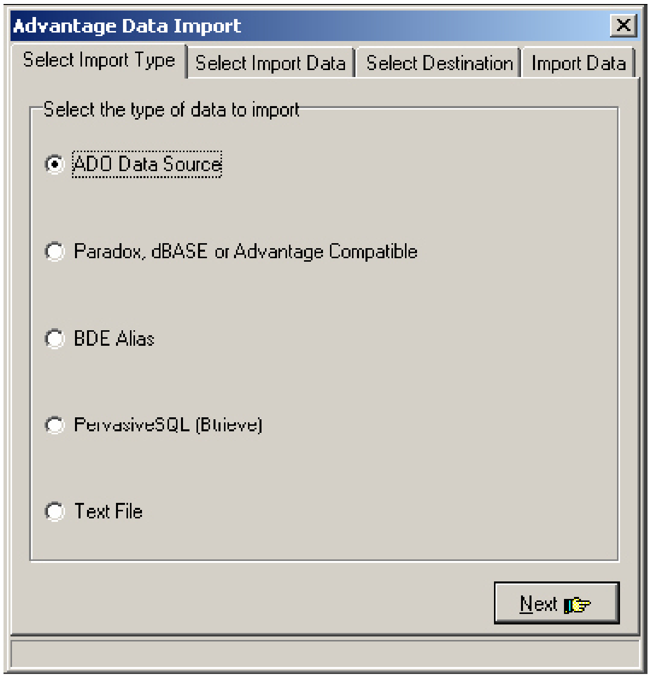
If you have questions while you are setting the properties for your database press F1 for help.

3. Enter a Name and Alias for the database. The Name value will be the name for the Advantage Data Dictionary file. The name of the data dictionary is the name of the database.
4. Enter an Alias for the database. The Alias will be used to create an Alias=Path entry in the ADS.INI file. Using an alias in your application to refer to the database lets you change the database location by changing the entry in ADS.INI.

5. Check the Logins Required check box to require that each user provide a valid user name and password to connect to the database.
6. Check the User Rights check box if you want ADS to check that the user has access rights to objects in the database such as tables and procedures.
7. Check Enable Data Dictionary Encryption if you want to encrypt your data dictionary for security. You cannot change this setting after the data dictionary has been created. Encrypting the data dictionary does not encrypt the data in the tables. If you want to encrypt your table data you must check the Table Encryption Enabled check box. Note that the Table Encryption Enabled check box is disabled now. You can only set table encryption after the data dictionary has been created. If you want to enable table encryption after you have created your data dictionary right click the data dictionary name in the tree view and choose Properties.
8. Set the database version number. The version number provides a convenient way to track changes in the database structure.
9. Add an optional Database Description if you wish.
10. If your database will be accessed across the Internet using Advantage Internet Server check the Internet Access Enabled check box.
11. If Internet Access is enabled set the Security Level to 0, 1, or 2.
    - Level 0 does not require user login and does not encrypt traffic between the client and server.
    - Level 1 requires users to login but does not encrypt network traffic.
    - Level 2 requires login and encrypts network traffic.
12. Set the maximum number of times a user may attempt to login.
13. Click the OK button to create the data dictionary.
14. You will be notified that the database properties have been saved then you will be prompted for a password for the administrative user, ADSSYS. The password for ADSSYS in the sample database is adssys. After you have entered the password Advantage Data Architect will create the data dictionary.

The next step is to import your existing Microsoft Access database.

1. Choose Tools | Import from the Advantage Data Architect menu to open the Advantage Data Import dialog.



Select the type of data to import

● ADO Data Source

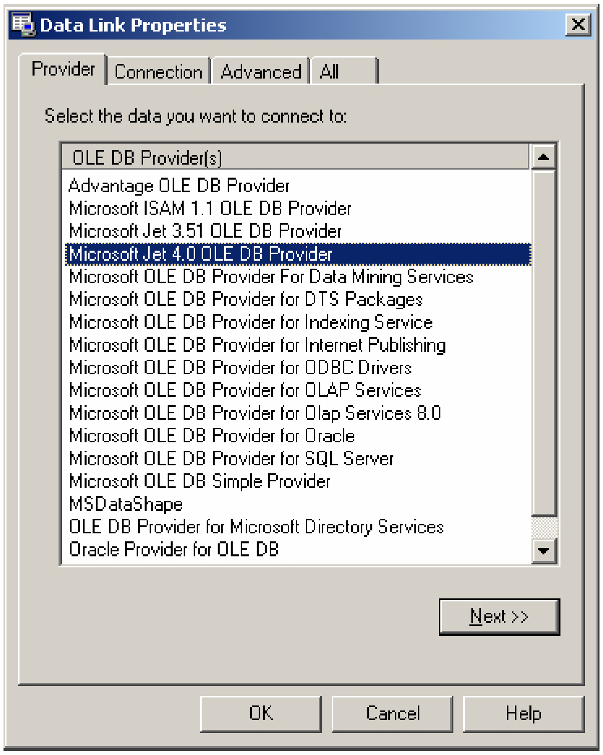
○ Paradox, dBASE or Advantage Compatible

○ BDE Alias

○ PervasiveSQL (Btrieve)

○ Text File

2. Click the ADO Data Source radio button.
3. Click Next button to move to the Select Import Data tab.
4. Enter your ADO connection string or click the Build button to open the Microsoft Data Link Properties dialog..



Select the data you want to connect to:

OLE DB Provider(s)
Advantage OLE DB Provider
Microsoft ISAM 1.1 OLE DB Provider
Microsoft Jet 3.51 OLE DB Provider
Microsoft Jet 4.0 OLE DB Provider
Microsoft OLE DB Provider For Data Mining Services
Microsoft OLE DB Provider for DTS Packages
Microsoft OLE DB Provider for Indexing Service
Microsoft OLE DB Provider for Internet Publishing
Microsoft OLE DB Provider for ODBC Drivers
Microsoft OLE DB Provider for OLAP Services
Microsoft OLE DB Provider for Olap Services 8.0
Microsoft OLE DB Provider for Oracle
Microsoft OLE DB Provider for SQL Server
Microsoft OLE DB Simple Provider
MSDataShape
OLE DB Provider for Microsoft Directory Services
Oracle Provider for OLE DB

5. On the Provider tab choose the Microsoft Jet 4.0 OLE DB Provider.

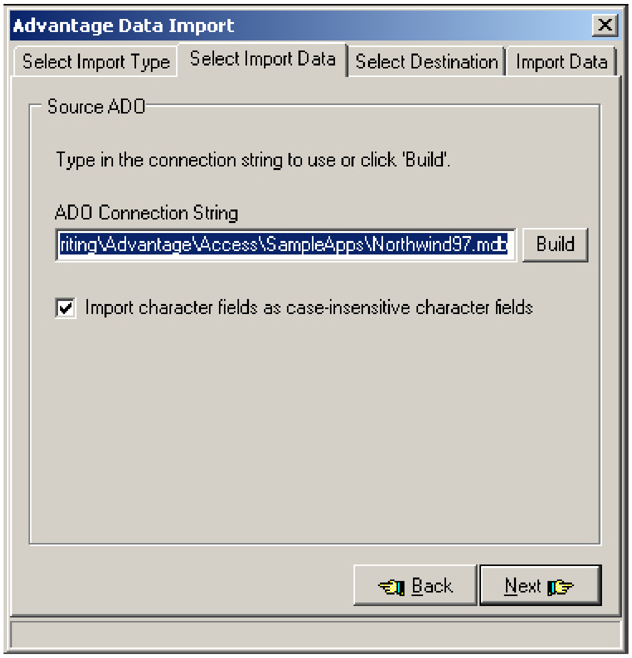
6. Click the Next button then enter or select the path to your Microsoft Access database.

7. If your database requires a password uncheck the Blank Password check box and enter the password. If you wish, you can check the Allow Saving Password check box to include the password in the connection string, however, be aware that the password is not encrypted.

8. Click the Test Connection button to ensure that you can connect to your database then click the OK button.

9. Click the Next button on the Advantage Data Import wizard and enter the path to the directory that holds your Advantage database.

10. Character fields are case insensitive in Access. If you want your character fields to be case insensitive in Advantage check the Import Character Fields as Case Insensitive Character Fields check box as shown below.



Type in the connection string to use or click 'Build'.

ADO Connection String
riting\Advantage\Access\SampleApps\Northwind97.mdb

☑ Import character fields as case-insensitive character fields

11. Click the Next button then click the Import button.

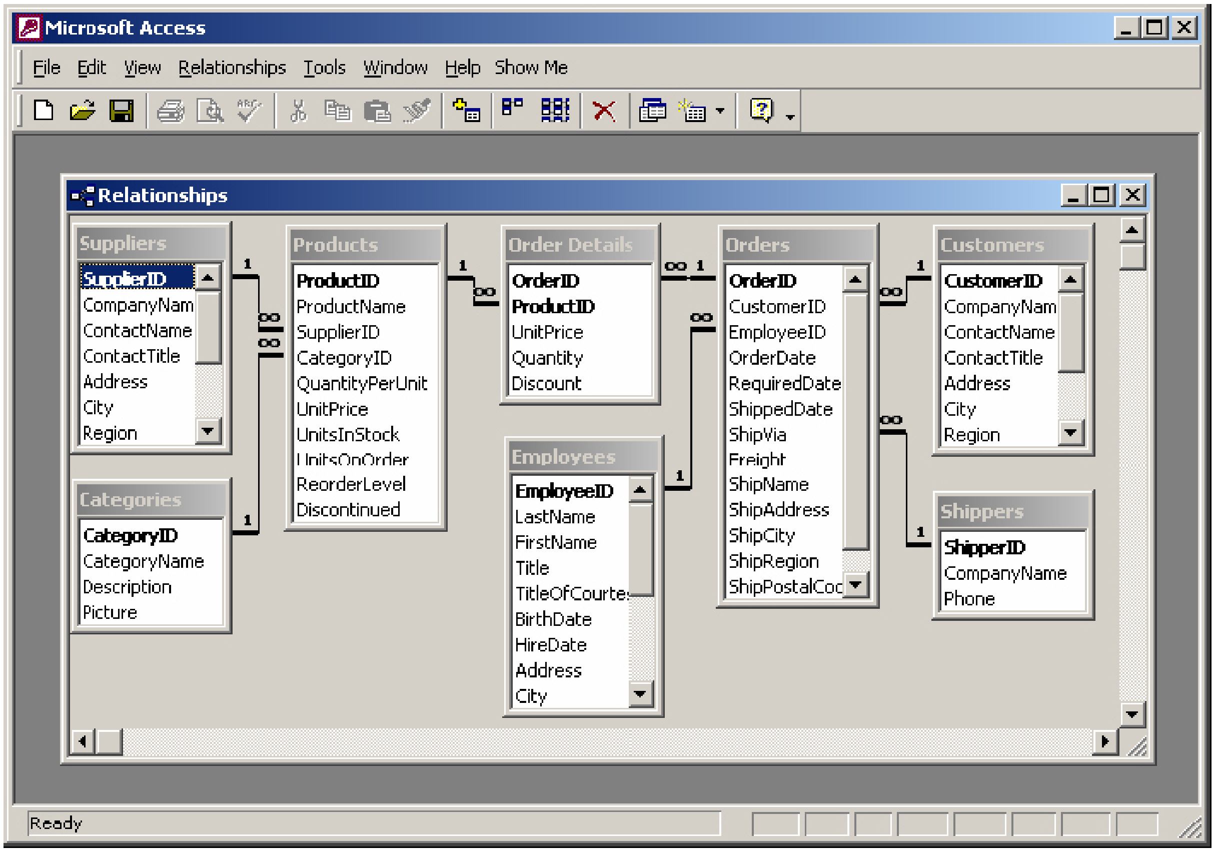
As the tables and indexes are imported progress messages are displayed in the memo control. When importing has finished you can print the progress log for review. The import operation creates Advantage tables that are not part of any data dictionary. These are called free tables. The next step is to add the free tables to the data dictionary you created earlier.

1. Your new database should still be open in the Advantage Data Architect. If it is not, open it by choosing Database | Open Database from the menu.
2. Right click the Tables node in the tree view and choose Add Existing Table(s).
3. Select all of the tables and click the Open button.

Although the Advantage Data Architect imports all of the tables and indexes it cannot determine which index is the primary index. To select the primary index for the Employees table follow these steps.

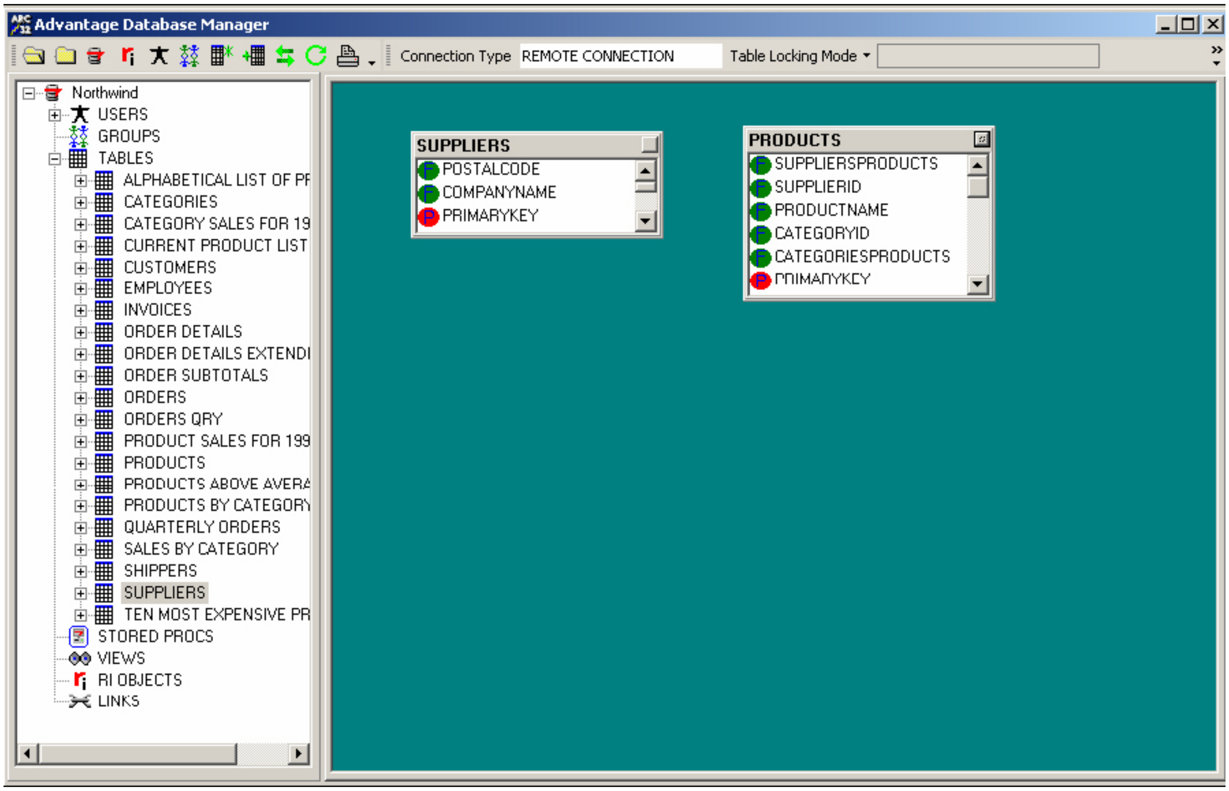
1. Click the plus sign next to Tables in the tree view to expand the list of tables.
2. Click the plus sign next to EMPLOYEES.
3. Click the plus sign next to INDEX FILES.
4. Click the plus sign next to EMPLOYEES.adi.
5. Right click the index you want to use as the primary key. In this example use the index named PRIMARYKEY.
6. Choose Make Primary Index from the context menu.
7. Repeat steps 2 through 6 for all of the tables in the data dictionary that need a primary key.

The next step is to recreate the referential integrity relationships because they cannot be imported. If you have a copy of Access available, open the database and click the Relationships button on the toolbar to show the relationships form below.



Perform the following steps in Advantage Data Architect to create the referential integrity relationship between the Suppliers table and the Products table.

1. Drag the Suppliers and Products tables onto the Advantage Data Architect workspace so that the workspace looks like this.

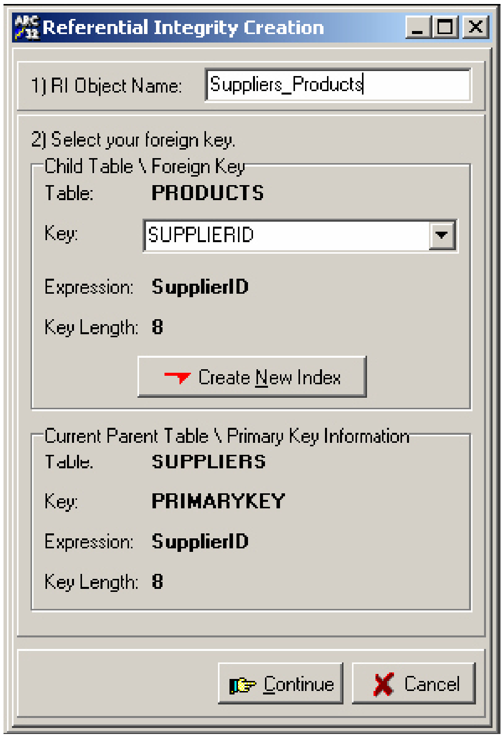


Advantage Database Manager

Connection Type REMOTE CONNECTION

Table Locking Mode

2. Drag any field from the child table (Products) to the parent table (Suppliers) to open the Referential Integrity Creation dialog shown below.



Referential Integrity Creation

1) RI Object Name: Suppliers_Products

2) Select your foreign key.

Child Table \ Foreign Key

Table: **PRODUCTS**

Key: SUPPLIERID

Expression: **SupplierID**

Key Length: **8**

Current Parent Table \ Primary Key Information

Table: **SUPPLIERS**

Key: **PRIMARYKEY**

Expression: **SupplierID**

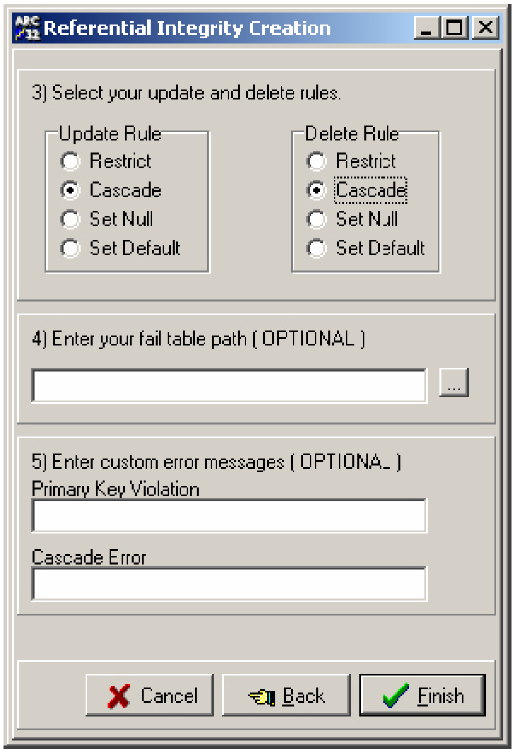
Key Length: **8**

3. Enter a name for the referential integrity object.

4. If the correct foreign key field for the Products table is not shown in the drop-down box choose the correct field.

5. The foreign key field should have an index. If it does not click the Create New Index button to create one.

6. Click Continue to display the second page of the dialog as shown below.



Referential Integrity Creation

3) Select your update and delete rules.

Update Rule
○ Restrict
● Cascade
○ Set Null
○ Set Default

Delete Rule
○ Restrict
● Cascade
○ Set Null
○ Set Default

4) Enter your fail table path (OPTIONAL)

5) Enter custom error messages (OPTIONAL)

Primary Key Violation

Cascade Error

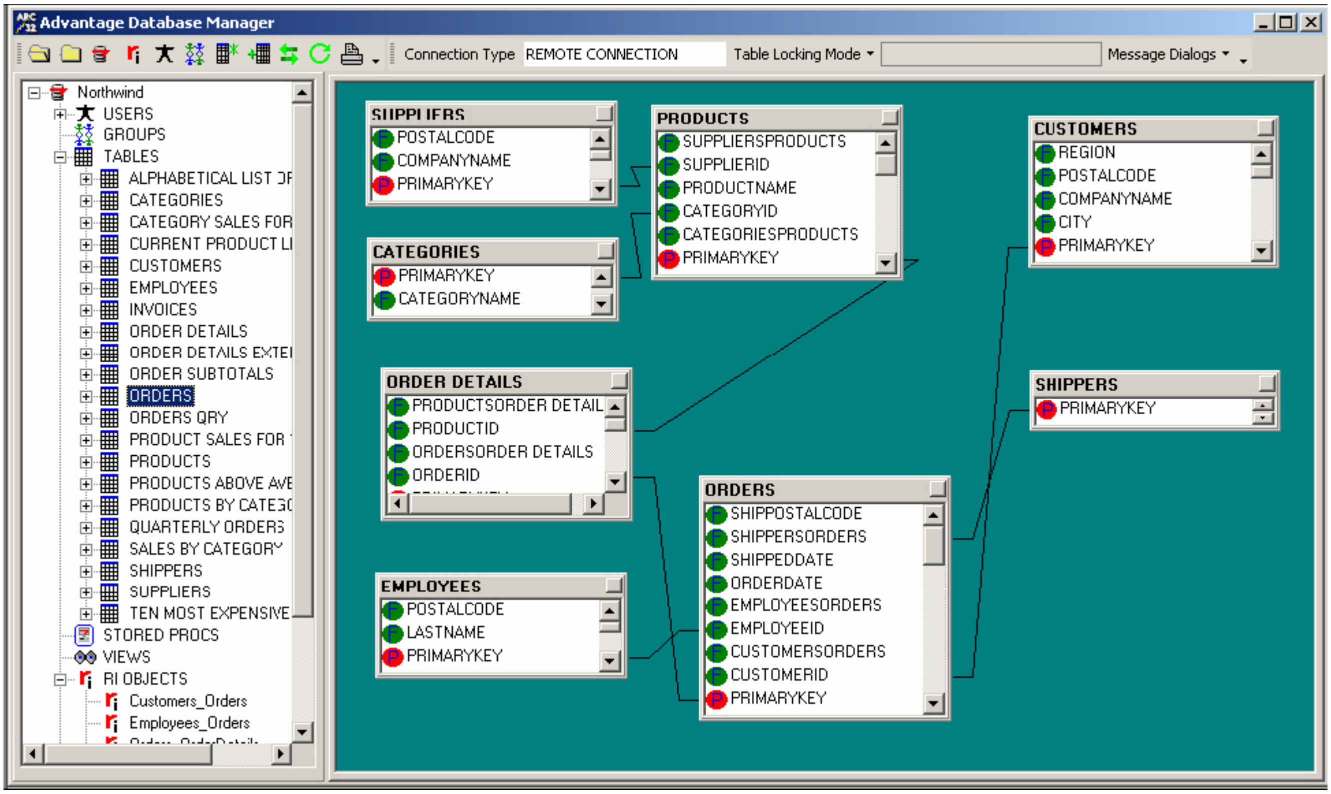
7. Select Cascade for both the update rule and the delete rule.

8. If you want records that fail the referential integrity constraint placed in a table enter the path. For this example leave the path blank.

9. If you want customer error messages, enter them.

10. Click Finish.

11. Repeat these steps for each referential integrity relationship in the database. When you are finished the entity relationship diagram in Advantage Data Architect should look something like this.



Your new database has only one user named ADSSYS. ADSSYS is the administrative user, the only user that, by default, has the ability to create, drop or alter database objects. You can grant administrative rights to other users but only ADSSYS has administrative rights. Because ADSSYS is all-powerful, you may not want your application to login to the database as ADSSYS.

To create a new user for your application, follow these steps.

1. Right click Users in the tree view and choose Add User from the context menu.
2. Enter a user name and password then click Continue. The sample application uses App for the user name and app for the password.
3. Click the Table Rights button and give the user full rights to all tables.
4. Click the Finish button.

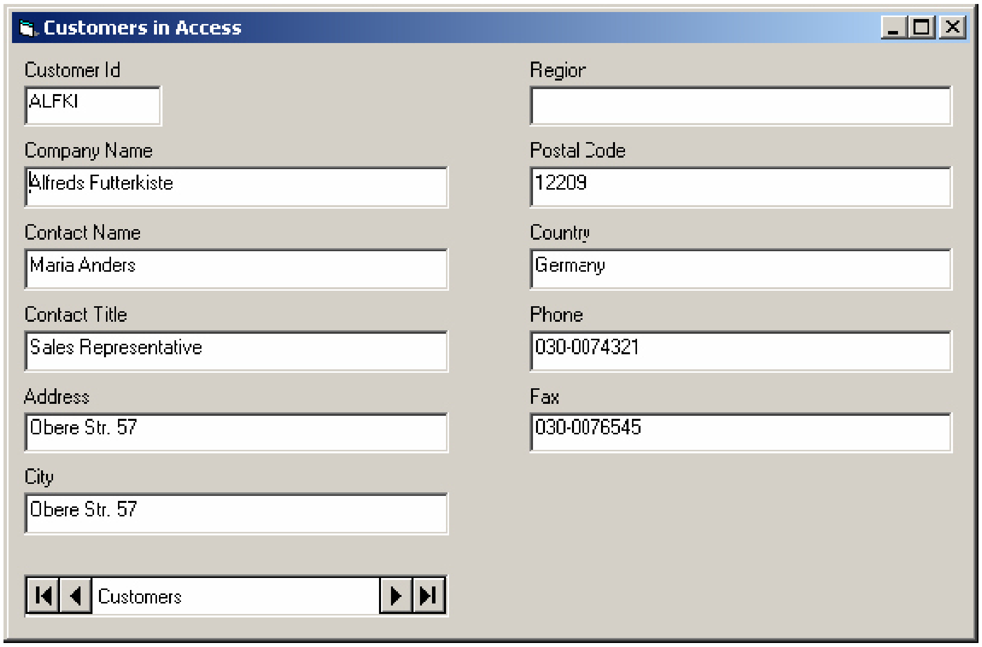
Your database has now been imported into Advantage and you are ready to convert your application.

CONVERTING A VISUAL BASIC APPLICATION

Visual Basic applications that use the Microsoft Access Jet database fall into two groups. Applications written in the last few years use ADO to connect to the database and can be converted easily by changing the OLE DB provider and connection string.

Older Visual Basic applications connect using DAO and an ODBC driver. These applications can be converted to ADS using the Advantage ODBC driver. However, Microsoft no longer supports or develops DAO. If you are converting an application that uses DAO you should seriously consider converting to ADO at the same time.

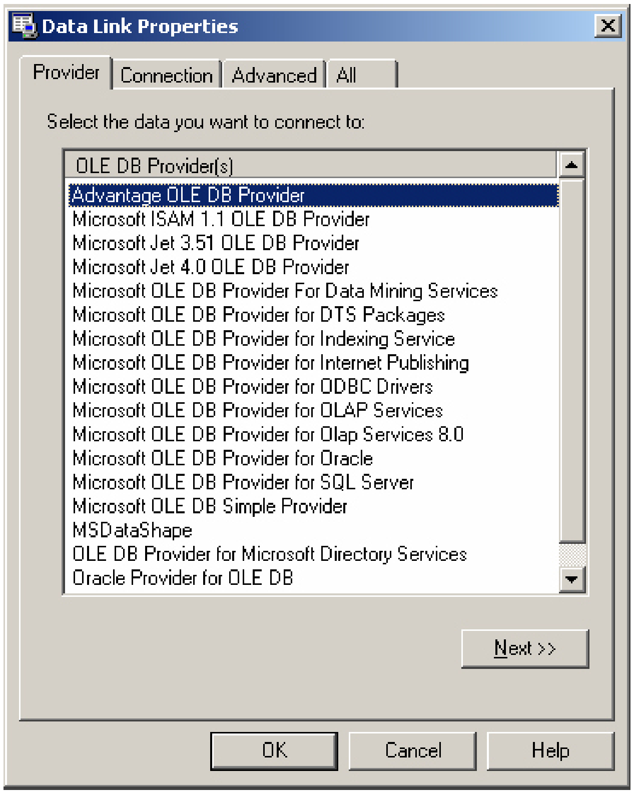
Converting a Visual Basic 6 application that connects to a Microsoft Access Jet database using ADO is very easy. The figure below shows the sample application that accompanies this paper, displaying customers from the Northwind database. This is a pretty simple application but it demonstrates everything you need to know to convert from a Microsoft Access database to Advantage Database Server.



Customer Id: ALFKI
Company Name: Alfreds Futterkiste
Contact Name: Maria Anders
Contact Title: Sales Representative
Address: Obere Str. 57
City: Obere Str. 57
Regior:
Postal Code: 12209
Country: Germany
Phone: 030-0074321
Fax: 030-0076545

To switch this application from Microsoft Access to Advantage just follow these steps:

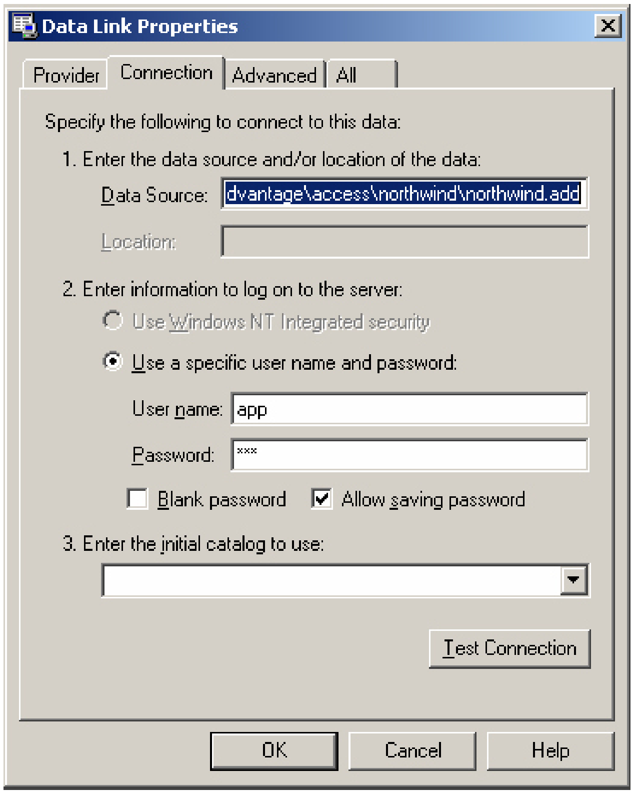
1. Download and install the Advantage OLE DB provider.
2. Open the project in Visual Basic.
3. Right click the ADO Data Control at the bottom of the form and choose ADODC Properties from the context menu.
4. Click the Build button next to the connection string to open the Data Link Properties dialog.



Select the data you want to connect to:

OLE DB Provider(s)
Advantage OLE DB Provider
Microsoft ISAM 1.1 OLE DB Provider
Microsoft Jet 3.51 OLE DB Provider
Microsoft Jet 4.0 OLE DB Provider
Microsoft OLE DB Provider For Data Mining Services
Microsoft OLE DB Provider for DTS Packages
Microsoft OLE DB Provider for Indexing Service
Microsoft OLE DB Provider for Internet Publishing
Microsoft OLE DB Provider for ODBC Drivers
Microsoft OLE DB Provider for OLAP Services
Microsoft OLE DB Provider for Olap Services 8.0
Microsoft OLE DB Provider for Oracle
Microsoft OLE DB Provider for SQL Server
Microsoft OLE DB Simple Provider
MSDataShape
OLE DB Provider for Microsoft Directory Services
Oracle Provider for OLE DB

5. Select the Advantage OLE DB Provider then click Next to display the Connection tab.



Specify the following to connect to this data:

1. Enter the data source and/or location of the data:

Data Source: dvantage\access\northwind\northwind.add

Location:

2. Enter information to log on to the server:

○ Use Windows NT Integrated security

● Use a specific user name and password:

User name: app

Password: ***

□ Blank password ☑ Allow saving password

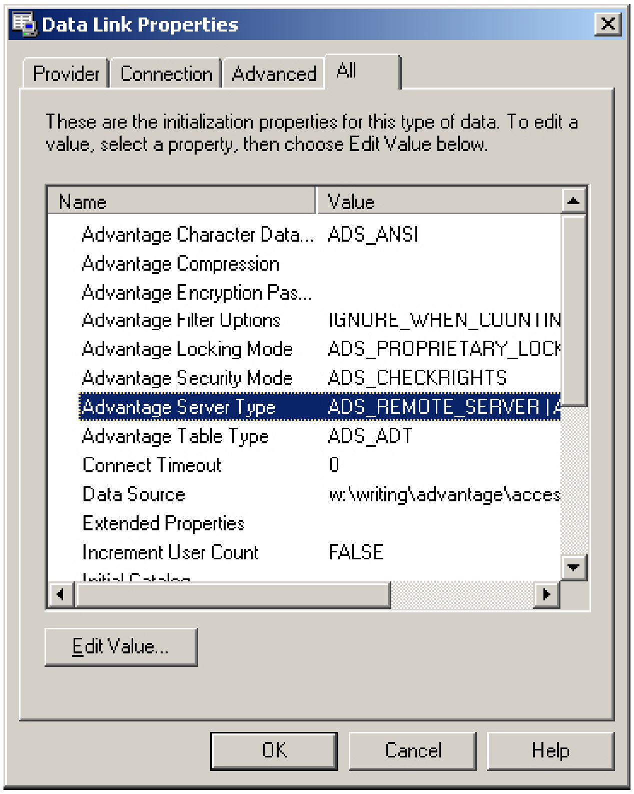
3. Enter the initial catalog to use:

6. Enter the path to your Advantage database directory in the Data Source edit box.
   If you are using Advantage Local Server or if you are using Advantage Database Server and it is running on your local machine enter the path to the data dictionary file, in this example NORTHWIND.ADD. If you are using Advantage Database Server and it is running on another machine on the network enter the UNC path to the data dictionary file. To connect across the Internet see Specifying the Advantage Internet Server Location in the on-line help.

   To select which Advantage servers your application will try to connect to click the All tab and scroll down to the Advantage Server Type property shown below. Edit the property and enter the server types in the order you want your application to try to connect to them. For example:

   ADS_REMOTE_SERVER | ADS_LOCAL_SERVER | ADS_AIS_SERVER

   tells your application to try to connect to a remote server first, then the local server and finally the Internet server.



These are the initialization properties for this type of data. To edit a value, select a property, then choose Edit Value below.

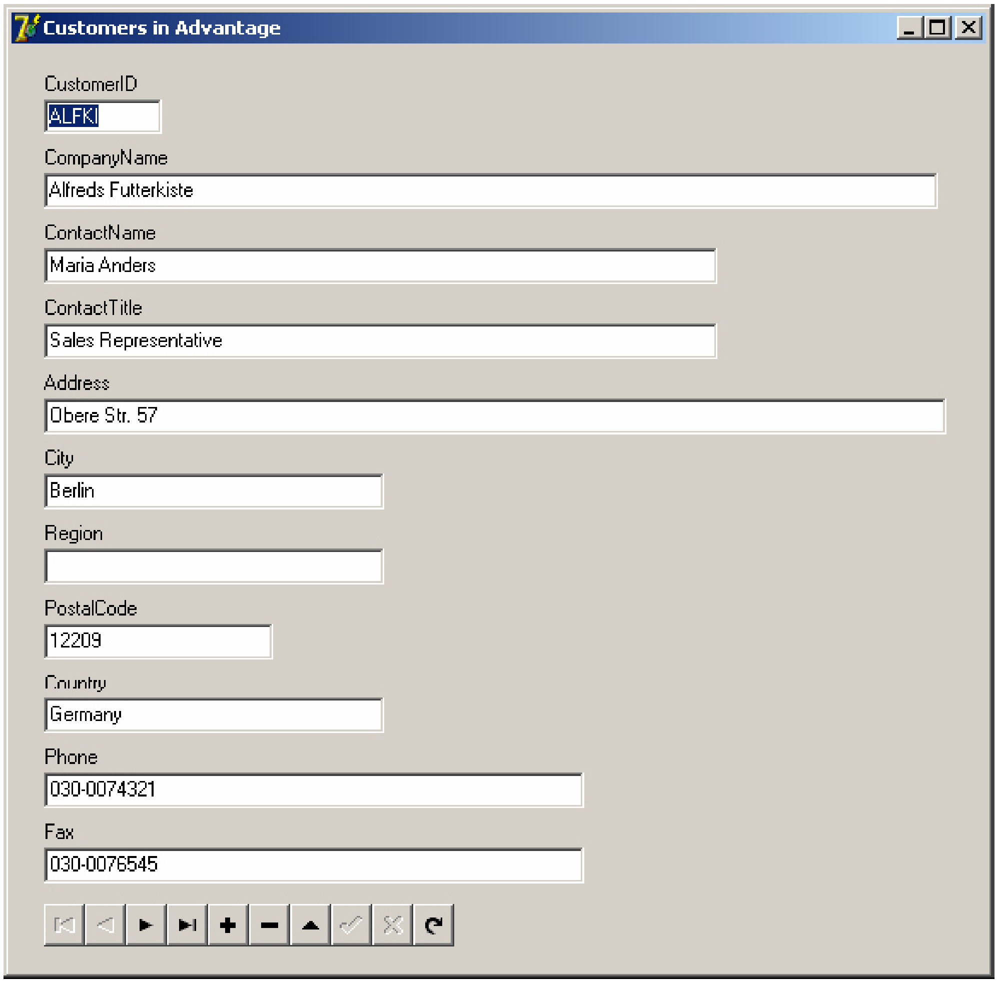
Name	Value
Advantage Character Data...	ADS_ANSI
Advantage Compression	
Advantage Encryption Pas...	
Advantage Filter Options	IGNORE_WHEN_COUNTIN
Advantage Locking Mode	ADS_PROPRIETARY_LOCK
Advantage Security Mode	ADS_CHECKRIGHTS
Advantage Server Type	ADS_REMOTE_SERVER |A
Advantage Table Type	ADS_ADT
Connect Timeout	0
Data Source	w:\writing\advantage\acces
Extended Properties	
Increment User Count	FALSE
Initial Catalog	

7. Enter the user name and password that your application will use.
8. Click the Test Connection button.
9. Click OK to close the Data Link Properties dialog.
10. Click OK to close the ADO Data Control Properties dialog.
11. Save and compile your application.

This example assumes that you use a connection string in your application. If you connect using a Data Link file you do not need to change your application at all. Just double click your Data Link file to open the Data Link Properties dialog then follow steps 5 through 9 above. Finally, do not forget to review the Conversion Issues Common to all Environments section to see if any of the differences described there apply to your application.

CONVERTING A DELPHI ADO APPLICATION

Converting a Delphi application that connects to a Microsoft Access Jet database using ADO is easy. The screen below shows the sample application that accompanies this paper, displaying customers from the Northwind database. This is a pretty simple application but it demonstrates everything you need to know to convert a Delphi ADO application from Microsoft Access to ADS.


Customers in Advantage

CustomerID
ALFKI

CompanyName
Alfreds Futterkiste

ContactName
Maria Anders

ContactTitle
Sales Representative

Address
Obere Str. 57

City
Berlin

Region

PostalCode
12209

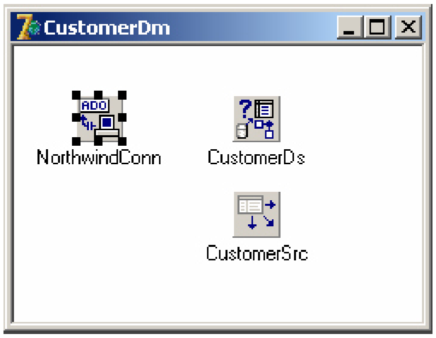
Country
Germany

Phone
030-0074321

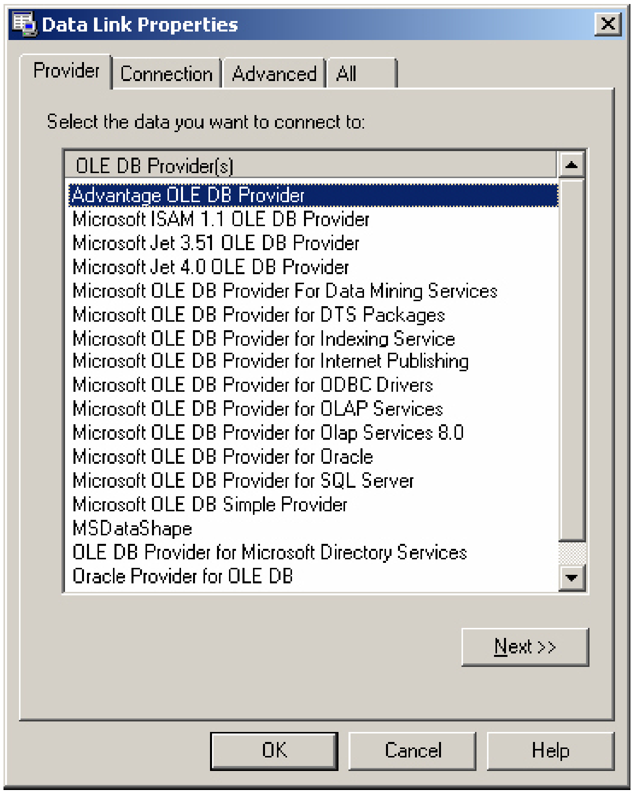
Fax
030-0076545

To switch this application from Microsoft Access to Advantage just follow these steps.

1. Download and install the Advantage OLE DB provider.
2. Open the project in Delphi.
3. Open the CustomerDm data module.



4. Select the ADOConnection component.
5. Select the ConnectionString property in the Object Inspector and click the ellipsis button.
6. Click the Build button in the ConnectionString dialog to open the Data Link Properties dialog.

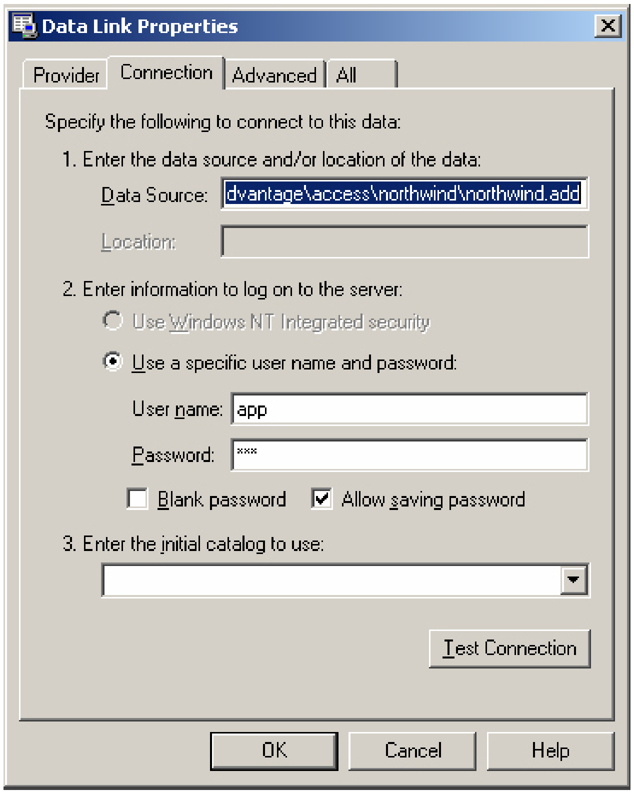


Select the data you want to connect to:

OLE DB Provider(s)

- Advantage OLE DB Provider
- Microsoft ISAM 1.1 OLE DB Provider
- Microsoft Jet 3.51 OLE DB Provider
- Microsoft Jet 4.0 OLE DB Provider
- Microsoft OLE DB Provider For Data Mining Services
- Microsoft OLE DB Provider for DTS Packages
- Microsoft OLE DB Provider for Indexing Service
- Microsoft OLE DB Provider for Internet Publishing
- Microsoft OLE DB Provider for ODBC Drivers
- Microsoft OLE DB Provider for OLAP Services
- Microsoft OLE DB Provider for Olap Services 8.0
- Microsoft OLE DB Provider for Oracle
- Microsoft OLE DB Provider for SQL Server
- Microsoft OLE DB Simple Provider
- MSDataShape
- OLE DB Provider for Microsoft Directory Services
- Oracle Provider for OLE DB

7. Select the Advantage OLE DB Provider then click Next to display the Connection tab.



Specify the following to connect to this data:

1. Enter the data source and/or location of the data:

Data Source: dvantage\access\northwind\northwind.add

Location:

2. Enter information to log on to the server:

○ Use Windows NT Integrated security

● Use a specific user name and password:

User name: app

Password: ***

□ Blank password ☑ Allow saving password

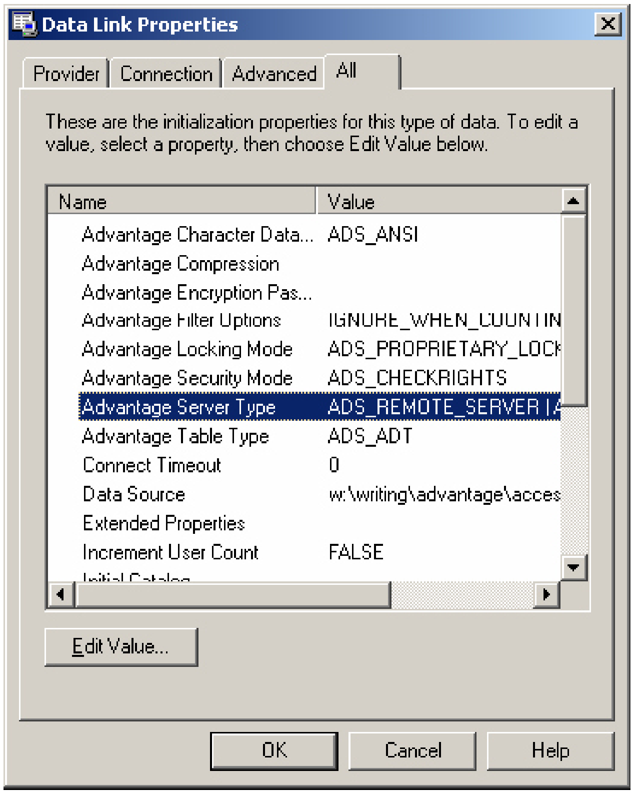
3. Enter the initial catalog to use:

8. Enter the path to your Advantage database directory in the Data Source edit box.
   If you are using Advantage Local Server or if you are using Advantage Database Server and it is running on your local machine enter the path to the data dictionary file, in this example NORTHWIND.ADD. If you are using Advantage Database Server and it is running on another machine on the network enter the UNC path to the data dictionary file. To connect across the Internet see Specifying the Advantage Internet Server Location in the on-line help.

   To select which Advantage servers your application will try to connect to click the All tab and scroll down to the Advantage Server Type property shown below. Edit the property and enter the server types in the order you want your application to try to connect to them. For example:

   ADS_REMOTE_SERVER | ADS_LOCAL_SERVER | ADS_AIS_SERVER

   tells your application to try to connect to a remote server first, then the local server and finally the Internet server.



These are the initialization properties for this type of data. To edit a value, select a property, then choose Edit Value below.

Name	Value
Advantage Character Data...	ADS_ANSI
Advantage Compression	
Advantage Encryption Pas...	
Advantage Filter Options	IGNORE_WHEN_COUNTIN
Advantage Locking Mode	ADS_PROPRIETARY_LOCK
Advantage Security Mode	ADS_CHECKRIGHTS
Advantage Server Type	ADS_REMOTE_SERVER | A
Advantage Table Type	ADS_ADT
Connect Timeout	0
Data Source	w:\writing\advantage\acces
Extended Properties	
Increment User Count	FALSE
Initial Catalog	

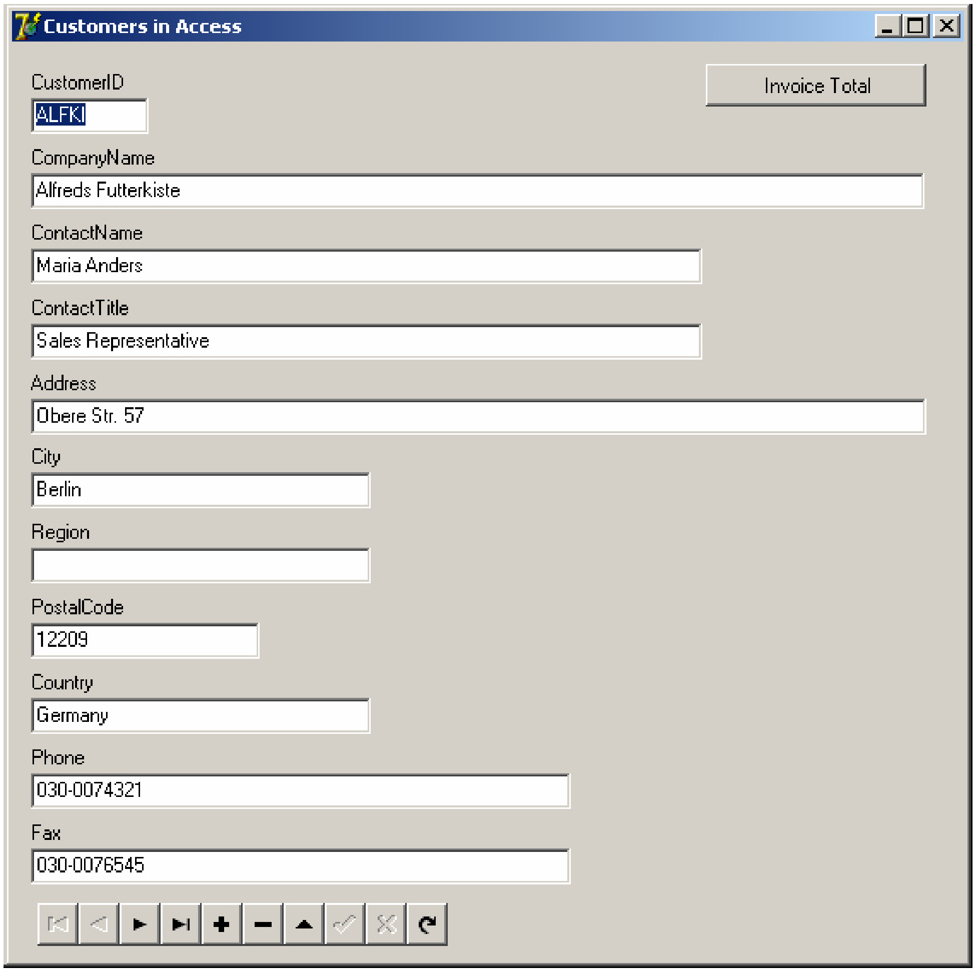
9. Enter the user name and password that your application will use.
10. Click the Test Connection button.
11. Click OK to close the Data Link Properties dialog.
12. Click OK to close the ConnectionString dialog.
13. Save and compile your application.

This example assumes that you use a connection string in your application. If you connect using a Data Link file you do not need to change your application at all. Just double click your Data Link file to open the Data Link Properties dialog then follow steps 7 through 11 above. Finally, remember to review the Conversion Issues Common to all Environments section to see if any of the differences described there apply to your application.

CONVERTING A DELPHI BDE APPLICATION

There are two ways to convert a Delphi application that uses the BDE MSACCESS driver to connect to a Microsoft Access database. The easiest method is to use the BDE to connect to Advantage using the Advantage ODBC driver. However, this is not the best choice. Borland has stopped development of the BDE and is encouraging developers to convert their applications to another data access technology. Moving your database from Microsoft Access to ADS gives you the opportunity to abandon the BDE in favor of the Advantage TDataSet components with only a bit more effort. That is the option described here.

Once again the sample application is a simple one that displays data from the Customers table in the Northwind database. This form also contains a button that runs a query to display the total amount for all of the invoices for the current customer.



CustomerID
ALFKI

CompanyName
Alfreds Futterkiste

ContactName
Maria Anders

ContactTitle
Sales Representative

Address
Obere Str. 57

City
Berlin

Region

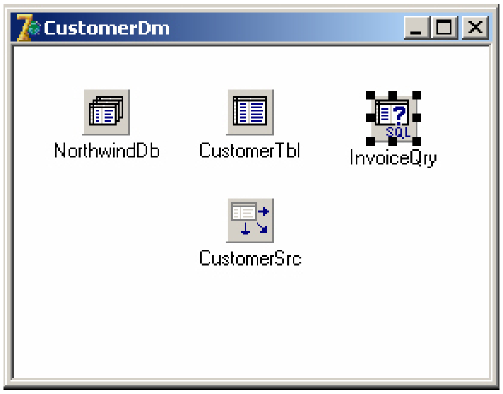
PostalCode
12209

Country
Germany

Phone
030-0074321

Fax
030-0076545

The data module contains a TDatabase, a TTable, a TQuery and a TDataSource. It also contains a custom method, GetInvoiceTotal, that runs the query and returns the total amount for the customer's invoices. Here are the steps to convert this application to use ADS and the Advantage TDataSet components.



1. Open the project in Delphi.
2. Open the data module named CustomerDm.
3. Make sure the CustomerForm form is not open in the Delphi IDE.
4. Right click data module and choose View as Text from the context menu.
5. Press Ctrl+R to open the Replace Text dialog.
6. Enter TDatabase for the Text to Find.
7. Enter TAdsConnection in the Replace With edit box.
8. Click Replace All.
9. In the same way, replace all occurrences of TTable with TAdsTable.
10. Replace all occurrences of TQuery with TAdsQuery.
11. Right click anywhere in the editor window and choose View as Form.
12. An error dialog will appear telling you that the Connected property does not exist. Click the Ignore All button. These errors occur because the TAdsConnection component does not have some of the properties that the TDatabase component has. Choosing Ignore All removes these properties.
13. Replace all occurrences of TDatabase in the data module's unit file with TAdsConnection.
14. Replace all occurrences of TTable with TAdsTable.
15. Replace all occurrences of TQuery with TAdsQuery.
16. Click the TAdsConnection component to select it.
17. In the Object Inspector, enter App for the UserName property.
18. Enter app for the Password.

There are two ways to specify the Advantage database that you want to connect to. The first is by entering the path to the database in the ConnectPath property of the TAdsConnection component. The disadvantage of this method is that you must recompile your application if you need to change the location of the database.

The second method is to create an ADS.INI file in your project directory and define an alias for your database. An alias is more flexible because, if you need to move the database, all you have to change is one line in the ADS.INI file.

1. Choose File | New | Other from the Delphi menu.
2. Double click the Text icon on the New tab of the Object Repository.
3. Enter the following two lines in the new file. Change the path as necessary to point to the Advantage database on your system.

```
[Databases] Northwind=W:\Writing\Advantage\Access\Northwind\Northwind.add;D
```

4. Choose File | Save As from the menu and name the file ADS.INI .
5. Compile and run your application.

To connect to a database on a remote server set the alias path to a UNC path, for example:

```
[Databases] Northwind=\\TestServer\D\Northwind\Northwind.add;D
```

To connect across the Internet see Specifying the Advantage Internet Server Location in the on-line help. To control which server types your application will try to connect to use the TAdsConnection component's ServerTypes property. Finally, do not forget to review the Conversion Issues Common to all Environments section to see if any of the differences described there apply to your application.

CONCLUSION

Whether you need to replace the Microsoft Access Jet database engine because Microsoft has deprecated it or because you have encountered one or more of its limitations, moving your Visual Basic 6 or Delphi application to Advantage Database Server is remarkably easy. When you move to Advantage Database Server you will be using true client server architecture with a database server that scales to hundreds of users yet is small, easy to deploy, reliable, requires no administration, has an exceptionally low total cost of ownership and provides all the features you need for the most demanding application.

Contact Us

iAnywhere Solutions, Inc.
Worldwide Headquarters
United States
One Sybase Drive
Dublin, CA 9
4568-7902 U.S.A.

For General Information:
AdvantageInfo@iAnywhere.com
North America T 1-800-801-2069 1-519-883-6898

For Specific Regional Product Information:
(800)-235-7576 ext. 5030 (208)-322-7575 ext. 5030
Benelux +31(0)73 6235359
Germany +49 (0)7032/ 798-200
United Kingdom +44 (0)117-333 9000

www.AdvantageDatabase.com

SYBASE®
iAnywhere®