

Advantage Database Server and Visual FoxPro

A Getting Started Guide

SYBASE ANYWHERE

TABLE OF CONTENTS

- 1 Introduction
- 1 Integrating Advantage with *Visual FoxPro*
- 1 DBC Conversion Utility
- 1 Data Access Mechanisms
- 2 Remote Views
 - 2 Creating a Remote View
 - 2 Making a Remote View Editable
- 3 Cursor Adapters
- 7 SQL Pass Through
- 7 Resources

KEY BENEFITS

- Fully scalable from local, to peer-to-peer, to client/server environments — with one set of source code
- Share DBF tables with existing FoxPro applications, easier conversion to client/server
- Encryption
- Data hiding
- ODBC access to VFP 9 tables
- Internet access to DBF tables
- Direct table access in ADO.NET
- Full text search

INTRODUCTION

Advantage Database Server has a long history of supporting FoxPro tables and indexes. In the early days, one Advantage client was a 16-bit CA-Clipper RDD (replaceable data driver) that supported an early FoxPro file format. Over the years, Advantage has matured into a feature-rich product with support for many different development environments. With the release of Advantage 9, much of the focus has been to add features that will benefit Visual FoxPro developers. The Advantage 9 release added support for more DBF field types such as DateTime, Varbinary, Varchar, Autoinc, and Currency. It also includes support for NULL field values, candidate keys, and international collation support compatible with Visual FoxPro. With this and other new functionality added to Advantage Database Server, it creates a powerful client/server solution for native DBF file support with transactions, full text search, referential integrity, replication, encryption, support for DBFs over 4GB, and other great features.

All of this comes with a major benefit: It is not necessary to convert your DBF data to another format in order to use Advantage. In fact, it is possible to continue using existing Visual FoxPro applications to access the data while concurrently accessing the same data with Advantage-enabled applications.

The numerous Advantage client options open up many opportunities. In addition to being able to add client/server data access to your Visual FoxPro application and access the same tables concurrently with Visual FoxPro, you can use other Advantage clients to access your data in completely new ways. The Advantage ODBC driver gives the ability to access your Visual FoxPro 9 tables from many different applications. You can create a web application using the Advantage .NET Data Provider using your existing data. Or use the Advantage JDBC driver to extend your application onto new platforms. Available clients include an OLE DB provider, an ODBC driver, a JDBC driver, a PHP driver, a DBD driver for Perl, a TDataSet descendant for Delphi, a Visual Objects RDD, the Advantage Client Engine API for powerful low level data access, and the general purpose Advantage Data Architect for table and database management.

In addition, Advantage provides a natural migration path for Visual FoxPro developers writing new applications using Visual Studio. Advantage provides direct table access previously not available to ADO.NET developers. Advantage Database Server allows developers the flexibility to combine powerful SQL statements and relational data access methods with the performance and control of navigational operations, such as direct table and index access and direct table movement (seeks, skips, etc.). Combining these two powerful data access methods allows the result set returned from an SQL query or from a table opened directly to be navigated via the highly optimized Advantage ISAM database engine. The result is an easy-to-use interface that supports SQL commands and direct-result-set navigation.

INTEGRATING ADVANTAGE WITH VISUAL FOXPRO

Visual FoxPro provides several different data access mechanisms that can be utilized to integrate Advantage into your existing application. Aside from Visual FoxPro's native data access, some common methods to access external data are Remote Views, Cursor Adapters, and SQL Pass Through. A brief introduction to these methods follows.

Because Advantage can share your DBF tables with existing Visual FoxPro applications, it is possible to change applications in piecemeal fashion rather than needing to accomplish a full conversion in a single step. You can also add new Advantage-enabled applications on top of existing applications.

DBC CONVERSION UTILITY

Advantage does not support direct access to the database container (DBC) file. Therefore, if you are using a DBC it may be desirable to run DBCConvert.prg against the DBC. This utility is installed with the Advantage OLE DB provider and the Advantage ODBC driver. This utility will create an Advantage Data Dictionary based on the contents of a DBC. It does not modify the DBC or any data. The utility will create an ADD that references all of the tables referenced in the DBC, create long field names, convert table and field validation rules, default field values, and convert referential integrity rules.

DATA ACCESS MECHANISMS

One of the great features of Visual FoxPro is the way that it integrates the data access mechanisms into a similar programming paradigm. Once the data has been retrieved (e.g., through a remote view or a Cursor Adapter), the subsequent usage of the data is the same as if it had been retrieved through a USE statement. This reduces the

AVAILABLE CLIENTS

- ODBC Driver
- .NET Data Provider
- JDBC Driver
- OLE DB Provider
- PHP Driver
- DBD Driver for Pearl
- TDataSet Descendant for Delphi
- Visual Objects RDD
- Advantage Client Engine for API
- Advantage Data Architect

amount of code that you need to change in order to use Advantage to handle your data.

REMOTE VIEWS

Remote views use ODBC to access data that is external to Visual FoxPro. If you are already using local views in Visual FoxPro, switching to a remote view to access the same data through Advantage may be the simplest option. Note that in order to use a remote view, you must have a database open in your project. Visual FoxPro stores information about the remote view in the DBC.

When creating a remote view, it may be desirable to limit the rows that are returned from the table through a WHERE clause (the Filter tab in the View Designer provides access to this functionality). When opening a view, the default behavior is to read all records from the view. If the table is large, this can be an expensive operation.

In order to use a remote view with Advantage, you must have the Advantage ODBC driver installed. A simple way to use a remote view is to use an ODBC data source to specify connection information. You can set up an ODBC data source through the Windows Control Panel. There is usually an icon named "Data Sources (ODBC)" that provides access to the ODBC administrator. You can also start it from a command prompt with the command ODBCAD32.EXE.

If you plan to share DBF tables with Visual FoxPro's native data access and Advantage at the same time, you should choose the "Compatible" option for the "Advantage Locking" choice in the Advantage ODBC driver setup dialog. If you choose "Proprietary" and are using Advantage Database Server (remote server), then Visual FoxPro will not be able to access the tables, because Advantage opens tables in a mode that allows other applications to only open the table in a read-only mode.

The ODBC setup dialog defaults to a table type setting of "Advantage". This table type is used to access Advantage proprietary format tables (ADT tables). Change this value to "Visual FoxPro".

If you do not have Advantage Database Server (remote server) available, you should make sure that only "Local Server (ALS)" is selected in the Advantage ODBC setup dialog. This will speed up the connection to Advantage. If both the remote and local option are chosen, then the ODBC driver will attempt a remote server connection that must first fail before it scales back to the local option.

Creating a Remote View

1. One way to create a remote view is through the Project Manager in Visual FoxPro. Choose the Data tab, expand the Databases tree, click on the Remote Views tree and click the New button. This will show a dialog that prompts for a connection or data source. If you do not have any connections defined, you can select an ODBC data source by choosing the "Available data sources" option. Once you have selected a connection or data source, click OK and the view designer will be shown. At this point, you can choose which columns the view should contain.
2. Another way to create a remote view is with the CREATE SQL VIEW command in Visual FoxPro. For example, the following command will create a remote view named MyView that retrieves all columns from a table named test. It uses an ODBC data source named ADS.

```
CREATE SQL VIEW MyView REMOTE CONNECTION ADS as select * from test
```

Once the view is created, you can modify it with the view designer the same as if you created it in the project manager.

3. After you have created a remote view, you can use it much like you would a normal table. For example, the commands "use MyView" and "Browse" will open the view and display the results in the Browse window.

Making a Remote View Editable

The steps described above will create a remote view that can be used to bring data into Visual FoxPro from Advantage. In order for Visual FoxPro to be able to save any edits back to the physical table, it is necessary to provide the information that allows Visual FoxPro to generate the SQL statements to perform the updates. A simple way to do this is through the View Designer.

1. Open the view designer for the remote view. For example, you can use the command "MODIFY VIEW MyView".
2. Click on the "Update Criteria" tab in the View Designer.

3. This tab shows the list of available fields. Choose the key fields that uniquely identify a record by clicking in the left-most column of the list (under the key icon).
4. Select which fields should be updatable by clicking in the second column in the field list or simply click the "Update All" button to make all fields updatable.
5. Select the "Send SQL updates" check box. If this is not selected, Visual FoxPro will not send the update statements to Advantage and edits will not be saved.

CURSOR ADAPTERS

The CursorAdapter in Visual FoxPro is an extensible class that uses ADO (e.g., through OLE DB) to provide access to external data. The following steps show one way to access data through a CursorAdapter. While it is possible to use an ODBC driver through ADO, the best way to use a CursorAdapter with Advantage is to go through the Advantage OLE DB provider.

1. Verify that the Advantage OLE DB Provider is installed.
2. Open the Data Explorer in Visual FoxPro. You can do this by using the command "DO dataexplorer.app". The dataexplorer.app file should exist in your Visual FoxPro installation directory.
3. Add a connection in the Data Explorer. Choose the "Add Connection" button and then select "ADO Connection".
4. Click on the "Use connection string" option to use the Advantage OLE DB provider. You can use the Build button to create the connection string, or you can type it into the edit field. An example connection string to a directory containing free tables might be "Provider=Advantage.OLEDB.1;Data Source=x:\data\freetables;ServerType=ADS_REMOTE_SERVER;TableType=ADS_VFP_TABLE". This connection string specifies that Advantage Database Server (remote server) is to be used and that the table type is VFP (Visual FoxPro). An example connection string to a data dictionary using Advantage Local Server is: "Provider=Advantage.OLEDB.1;User ID=adssys;Data Source=x:\data\dd\test.add;ServerType=ADS_LOCAL_SERVER".
5. If you want Visual FoxPro to be able to access data natively and share it concurrently with Advantage Database Server, you need to choose compatibility locking. For example, add "LockMode=ADS_COMPATIBLE_LOCKING" to the connection string. The default is to use proprietary locking.
6. After creating the connection, expand the connection in Data Explorer and then expand the Tables tree. Drag (with the mouse) one of the tables onto a code window. Visual FoxPro will generate the code to create a CursorAdapter instance for that table.
7. You can use the generated code as needed. For example, add a Browse command at the bottom of the generated code and run the code to view the table. Note that you can browse tables directly from the Data Explorer too.

SQL PASS THROUGH

Like remote views, SQL Pass Through utilizes ODBC to perform data access. It provides a more direct access to the ODBC interface. It gives excellent low level control with the ability to specify transactions, execute stored procedures, run specific updates, etc. However, it can involve writing more code especially if you want updates to be sent to Advantage automatically.

The following example code runs a simple SELECT statement to retrieve the contents of a table named test. It stores the results in a Visual FoxPro cursor named MyResult. It assumes that an ODBC data source named "ads" exists.

```
#include "foxpro.h"
hConn = SQLCONNECT("ads")
IF hConn > 0
    SQLEXEC(hConn, "select * from test", "MyResult")
    BROWSE
    USE
    SQLDISCONNECT(hConn)
ELSE
    ? "Failed to connect to the data source"
ENDIF
```

In the above example, if you wanted to perform updates to the data, one possibility would be to execute SQL statements to do the updates. For example, `SQLEXEC(hConn, "insert into test (empid) values (111);"`).

If you want updates to be automatically sent to Advantage, however, some additional code must be written. In particular, several cursor properties must be set in order for the update statements to be automatically generated. The following code is a short example that makes some of the fields in the view editable.

```
#include "foxpro.h"
hConn = SQLCONNECT( "ads" )
IF hConn > 0
    ** retrieve the cursor
    SQLEXEC( hConn, "select * from test", "MyResult" )

    ** Set the properties necessary for automatic updates

    ** specify the table name
    CURSORSETPROP("Tables", "test", "MyResult" )
    ** indicate which field is the primary key
    CURSORSETPROP("KeyFieldList", "empid", "MyResult" )
    ** specify which fields can be updated
    CURSORSETPROP("UpdatableFieldList", "deptnum, lastname", "MyResult" )
    ** provide a mapping of view names to table names for the update
    CURSORSETPROP("UpdateNameList", ;
        "deptnum test.deptnum, ;
        empid test.empid, ;
        lastname test.lastname", "MyResult" )
    ** indicate that the WHERE clause will use only the primary key
    CURSORSETPROP("WhereType", DB_KEY, "MyResult" )
    ** Specify that updates should be sent to the backend
    CURSORSETPROP("SendUpdates", .t., "MyResult")

    BROWSE
    USE
    SQLDISCONNECT( hConn )
ELSE
    ? "Failed to connect to the data source"
ENDIF
```

RESOURCES

The Advantage.FoxPro newsgroup can be used to ask questions and request peer feedback. Advantage newsgroup information can be found at <http://devzone.advantagedatabase.com/dz/content.aspx?key=7>.

Additional questions can be directed to the Advantage Technical Services team at Advantage@iAnywhere.com.

The Advantage Developer's Zone (devzone.advantagedatabase.com) can be used to search the newsgroups, search knowledge base items, view online help, etc.

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
Inquiries**
1-800-801-2069
(208)-322-7575 ext. 5030
Germany
+49 (0)7032/ 798-200
United Kingdom
+44 (0)117 315 3900