



A Guide to FoxPro CursorAdapter Classes

Version: 1.6
Author: Stamati Crook
redware research ltd
<http://www.redware.com>
Date: 29 January 2005
Document: cursorAdapter01.doc

Contents

Contents	2
Copyright 2004, 2005	3
Introduction	4
CursorAdapter Class	5
CursorAdapter	5
Setup Table Buffering.....	6
Use Parameters to Filter Data.....	6
Fill a CursorAdapter using a Stored Procedure	7
Update Data	8
Save a CursorAdapter as a Visual Class.....	9
Use the CursorAdapter Event Model for Data Validation	10
Overcome the Property Length Problem in a Visual Class.....	10
Attach an Existing Cursor.....	11
Create a Cursor with the CursorAdapter Builder	11
Build a CursorAdapter with cabuilder.prg	14
Conclusion	16

Copyright 2004, 2005

The document is brought to you by **redware research ltd** and is copyright and published commercially. This means that you cannot distribute the document freely and should instead refer colleagues to our web site or to <http://www.amazon.com> where they may download their own copy for a small fee. Although every precaution has been taken in the preparation of this document, the publisher and author assume no responsibility for errors or omissions. Neither is any liability assumed for damages resulting from the use of the information contained herein. For information, please contact:

redware research ltd, 104 Tamworth Road, Hove BN3 5FH, England.
<http://www.redware.com>

This document is updated regularly and we would be pleased to send you an update if you register on our website. You will need this password: **adapter**. We also have other documents covering database development with Visual FoxPro and SQL Server and also Open Source web database applications available from our website and [amazon.com](http://www.amazon.com).

Introduction

Visual FoxPro has excellent facilities for database application development utilising a client-server database management system to store data as a replacement for the native FoxPro table files. The advantages of this approach are too numerous to mention here and there are various FoxPro features available to the developer requiring access to client-server data:

- Remote Views created in a database container to reference a client server database query representing retrieved data as a cursor.
- Parameterised Remote Views that limit the amount of data retrieved from the server.
- SQL pass-through queries that allow commands to be executed directly on the database server with the results returned as a cursor.
- Table buffering and commit and rollback commands to control the timing of updates sent to the database server.
- Connection control allowing performance optimisation, asynchronous queries, and so forth.

These features have been available since version 3.0 of Visual FoxPro and are now supplemented with the `CursorAdapter` class (first available with FoxPro 8.0) that allows an object-oriented software object that accesses client-server data in a flexible manner. Some notable features include:

- Transparent definition of native FoxPro or client-server database offering the best way to migrate an existing application to client-server.
- Seamless integration with ADO and XML.
- Manipulation of an extensive range of properties compatible with existing remote views and connection properties.
- Integration into the data environment of a form.
- Event driven model allowing sophisticated client-side validation and processing.

This document complements the redware Visual FoxPro Client-Server Handbook which describes in detail the techniques and optimisations to access client-server databases using all the available features of Visual FoxPro through to version 7.0. The use of the `CursorAdapter` object is discussed in detail with regards to accessing client-server data (ADO and XML interfaces are not covered here).

If you are not familiar with making connections to client-server databases and the use of table buffering with Visual FoxPro you will need further study or purchase of the redware FoxPro SQL Server Handbook (available from www.redware.com). Optimisation features are also not covered in this document which will be integrated into the next edition of the Handbook.

CursorAdapter Class

The CursorAdapter class is new for Visual FoxPro 8.0 and provides an object-oriented base class for creating cursors that control access to data.

CursorAdapters work well with various different data sources:

- Data stored in native FoxPro format on your local disk drive or local area network servers.
- Client-Server data stored in a central database management system accessed using ODBC drivers.
- ADO recordsets created on the workstation or by a middle tier business component server.
- XML documents.

CursorAdapters can be created programmatically or using the CursorAdapter builder that forms part of the visual design tools for the DataEnvironment of a Form. Third party tools such as the `cabuilder` from www.mctweedle.com build CursorAdapter class libraries directly from existing local or remote databases.

There are several advantages of using CursorAdapter classes:

- Object oriented inheritance allows a base class to be defined with application specific properties and methods.
- CursorAdapters are suitable for implementing systems that need to operate with either local or remote data according to the installation.
- The event model allows for powerful validation and trigger functionality to be implemented on the client-side of the application (perhaps to support triggers without needing to program for different database management systems).
- CursorAdapter definitions can be created and stored in a Visual Class Library and can be added into a DataEnvironment for a Form in a similar manner to local tables or views defined in a database container.

Some disadvantages include:

- CursorAdapters are objects and the object variable needs to remain in scope for the data to be available.
- Visual Class Libraries have a limit of 255 characters for properties and these properties often need to be defined in a method of the class.
- Views in the Database Container have additional properties for the individual fields defined for the cursor.
- A cursor created by a CursorAdapter class is instantiated in program code instead of with the `USE` command.

CursorAdapters combine many of the best qualities of views defined in a database container with the flexibility of programmatic control and object oriented inheritance. Storing CursorAdapter definitions within a Visual Class Library in combinations with the builder tool allows visual design and persistence of the definitions (although this needs some improvement). The main benefit however is a single object oriented technique for accessing data from a variety of data sources comprising local tables, client-server tables, ADO recordset objects and XML documents.

Note: This chapter does not discuss interfacing a CursorAdapter object with XML or ADO.

CursorAdapter

A CursorAdapter uses a Connection to communicate with a client server database when accessing data through with ODBC. A `SQLCONNECT` or `SQLSTRINGCONNECT` command is issued to get a Connection handle and the `DATASOURCETYPE` and assigned to the `DATASOURCE` property defined as shown below.

The `SELECTCMD` property is given a command that is executed on the server to create a local cursor from server data when the `CURSORFILL` method is called.

```
lnHandle = SQLSTRINGCONNECT( ;
    'DRIVER=SQL Server;SERVER=(local);UID=sa;PWD=;DATABASE=pubs' )
loPubs = CREATEOBJECT('cursoradapter')
loPubs.DATASOURCETYPE = 'ODBC'
loPubs.DATASOURCE = lnHandle
loPubs.ALIAS = 'caAuthors'
loPubs.SELECTCMD=[select * from authors]
IF loPubs.CURSORFILL()
    BROWSE
ELSE
    ? 'Error'
ENDIF
```

Note: This cursor will be closed as the program ends and the variable holding a reference to the CursorAdapter goes out of scope.

An identical procedure is followed for local FoxPro data where the `DATASOURCE` is blank and the `DATASOURCE` type is set to `'NATIVE'`.

```
lcTable = HOME()+'samples\northwind\customers'
IF SPACE(1) $ lcTable
    lcTable = '' + lcTable + ''
ENDIF
loCustomer = CREATEOBJECT('cursoradapter')
loCustomer.DATASOURCETYPE = 'NATIVE'
loCustomer.ALIAS = 'caCustomer'
loCustomer.SELECTCMD=[select * from ] + lcTable
IF loCustomer.CURSORFILL()
    BROWSE
ELSE
    ? 'Error'
ENDIF
```

Note: To avoid confusion it is best to use a different alias name to the name of the table as FoxPro also opens the original table in the data session.

Setup Table Buffering

The `BUFFERMODEOVERRIDE` property of the cursorAdapter object can be used to set up table or row level buffering to be used in conjunction with the `TABLEUPDATE` and `TABLEREVERT` commands.

Remember also to:

```
SET MULTILOCKS ON
```

Use Parameters to Filter Data

Accessing client server data without specifying a filter is expensive as all the data must be retrieved from the server into a cursor on the local machine. A parameterised query may be specified in the `SELECTCMD` property to specify a selection of data.

The following example shows a CursorAdapter object retrieving an empty cursor by specifying the `NODATA` property when filling the cursor with the `CURSORFILL` command. This adds a `WHERE 1=2` clause onto the `SELECT` command and returns an empty cursor with zero records.

```
RELEASE loPubs
PUBLIC loPubs
```