

About “Easy Data Export”

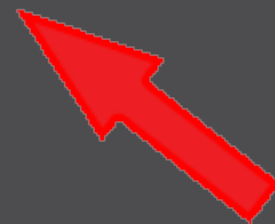
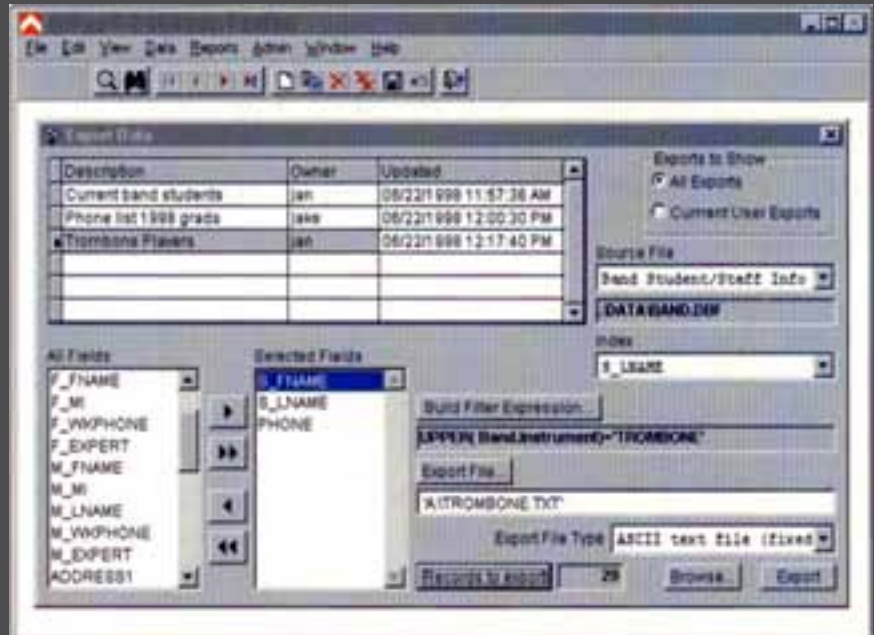
Microsoft FoxPro was the hottest product on the market for developing desktop database applications, back in the 1990’s.

I wrote this article for FoxPro Advisor Magazine after completing a project for a client. This module was a very small piece, and I had their full permission to share it publicly.

My co-author was the man I always met with when I went to the client’s office. Although he did not write a word of the article, he designed the whole process for exporting data. I hoped that including his name as a co-author would earn him respect in the eyes of his bosses and co-workers.

~ Jan

By the way: I added the red arrows (such as the one at right) in a couple of places in the article.



FOXPRO®

ADVISOR®

The Developer's Guide to Microsoft Visual FoxPro and FoxPro 2.x

DECEMBER 1999

WWW.ADVISOR.COM

Automate Your Testing

Designing for Users

6

Good applications serve users without forcing them to know what's going on under the hood.

Reliable Backups

14

ADVISOR LABS: Take the human element out of back ups with Xceed Backup Control.

String Comparisons and Better Performance

20

A review of some programming basics and a look at speed alternatives for designing faster applications.

Code Reuse With Business Objects

24

Once you've created a business object, access its functionality in other applications as well.

Easy Data Export

36

Let users export the data they want in the format they need.

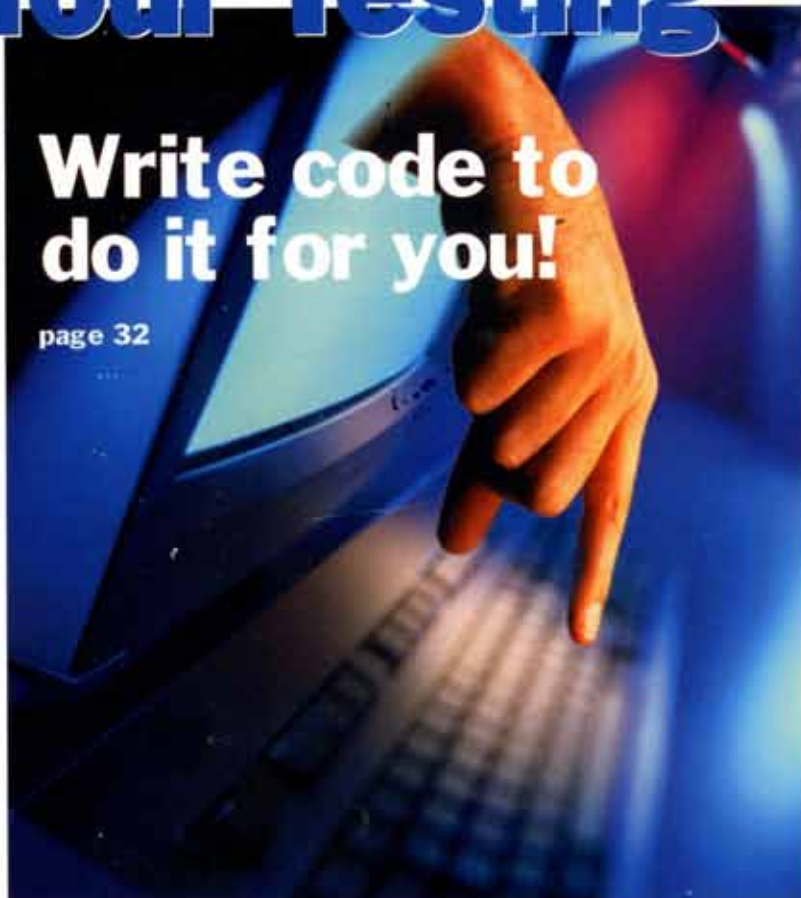
ADVISOR ANSWERS

- Look For an Index by Key
- Show Deleted Records in Views
- Share Connections for Views

10 ADVISOR TIPS

50

- Calculate Elapsed Times Trap
- Change Files to Read-Write
- Replace the NOEXT() Function
- Create a Header for a Self-Running Procedure File
- Avoid the Anomalous Prompt to Save Changes
- Run VFP 5.0 Applications in VFP 6.0
- Aid for Dropping Controls on Reports
- Follow-Up on Debugging SQL Code
- Prevent Multiple Application Instances
- Get an Arrow MousePointer for Grids Follow-up
- Calculate a Specific Previous Day of the Week



Write code to do it for you!

page 32

VOLUME 7 NUMBER 12

US\$4.99 Cdn\$7.25 UK£2.50



ADVISOR MEDIA, INC.

ADVISOR®—Guiding IT ProfessionalsSM Since 1983

ACTIVEX CONTROL
SHOWCASE **ADVISOR LABS**

14 Control Your Backups

By Andrew Ross MacNeill

If you want to make sure something gets done, automate it! The Xceed Backup Control makes it easy to ensure necessary-but-often-neglected backups get done.



FUNCTIONAL FOXPRO

20 String Comparisons and Speed

By Malcolm C. Rubel, Contributing Editor

Visual FoxPro's strengths include giving the developer more than one way to go about writing code, but it often pays (in application performance) to evaluate your options.

- Visual FoxPro 6.0/5.0/3.0
- FoxPro 2.x

OBJECT-ORIENTED ANALYSIS
AND DESIGN

24 Create Business Objects—Then Put 'em to Work!

By Kevin McNeish

Don't reinvent the wheel! After you create a business object in Visual FoxPro, you can access its functionality in other applications as well. This article takes a close look at the anatomy of business objects and the mechanics of constructing them in Visual FoxPro.

- Visual FoxPro 6.0/5.0/3.0

UTILITIES

36 Easy Data Export

By Jan Smith and Jonathan Mead

Get out of the loop when your users need data! This handy utility lets users export the data they want in the format they need.

- Visual FoxPro 6.0/5.0/3.0



50 ADVISOR TIPS

By Steve Sawyer, Contributing Editor

Steve and readers share useful tips on FoxPro development. This issue, find out how to:

- Calculate Elapsed Times Trap
- Change Files to Read-Write
- Replace the NOEXT() Function
- Create a Header for a Self-Running Procedure File
- Avoid the Anomalous Prompt to Save Changes
- Run VFP 5.0 Applications in VFP 6.0
- Aid for Dropping Controls on Reports
- Debugging SQL Code Follow-up Tip
- Prevent Multiple Application Instances
- Getting an Arrow MousePointer for Grids Follow-up Tip
- Calculate a Specific Previous Day of the Week

- Visual FoxPro 6.0/5.0/3.0
- FoxPro 2.x



FoxPro Advisor Professional Resource CD
Some of the extras on this month's CD:

- ✓ **ADVISOR ANSWERS**
The FindTag function that returns the name of the tag corresponding to a particular index key. See page 10.
- ✓ **Control Your Backups**
The Xceed Backup Library for adding on-demand or scheduled backups to Windows applications. See page 14.
- ✓ **Automate Your Testing**
An example class including the testcode method, the author's CEE macros for setting up TestCode, and the COB Editor Extensions. See page 32.
- ✓ **Easy Data Export**
Two versions of the export form, one using VFP base classes and one customized for the Visual ProMatrix framework. See page 36.
- ✓ **ADVISOR TIPS**
A class to use an arrow as a mouse pointer when over a grid. See page 50.

READER SERVICES

- 8** ADVISOR Reader Services
- 44** Business Directory
- 49** Advertiser and Product Index

Look for these icons:



You'll find article-related and other valuable files on this issue's PROFESSIONAL RESOURCE CD and on www.Advisor.com.

Easy Data Export

Let users export the data they want in the format they need.

By Jan Smith and Jonathan Meed

Don't you wish you had 10 dollars for every time someone said, "I need a copy of such-and-such data, so I can put it into my spreadsheet and play with it." You thought you were done tweaking that application. Sadly, you realize, it's a never-ending process.

Or is it? Here's an easy way to get out of the loop. Just add this export utility to your applications, and let users make their own copies of whatever data they want, in the format of their choice. (You'll find the export form and supporting code at <http://www.Advisor.com/>.)

What does it do?

Here's the idea: the user selects a source table (the table the data comes from) from a drop-down list, and then uses a two-column mover to select fields from the table. Next, he enters any desired filter and order criteria, selects the type of output (text file, spreadsheet, etc.), and specifies the name and location of the output file. That's all there is to it. A Browse button previews the resulting data set, and an Export button sends a copy of the data to the specified output file.

What's even better is that each person using the export form can create, save, and re-run as many of these export configurations as needed. Users become the owners of their configurations, and you get out of the loop!

As shown in figure 1, the grid at the top-left corner of the form shows a description of

these export configurations. The Owner column shows the user ID—from the security system—of the person who created the configuration. The Updated column shows when the record was created or last updated.

When the form first opens, the grid shows all previously defined export configurations. However, by clicking on the Current User Exports option button, the list shows only configurations for the current user. Users can see and even run each other's exports, but they can only edit their own.

But what about those times when someone else's configuration would be just what's needed, except for a minor change or two? No problem. It takes one mouse-click to make a copy. The owner of the copy can then make changes to the copy.

In our real-world system, the configuration table, CONFIG.DBF, also stores configurations

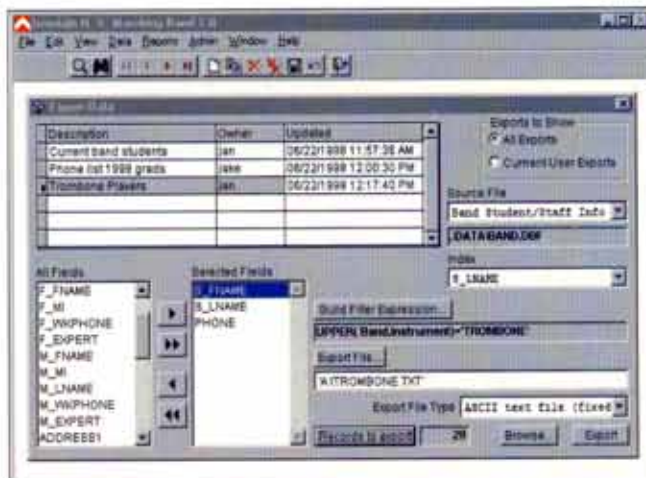


Figure 1: **Export form**—Used to create, save, and maintain export configurations, and to copy data to external files.

HIGHLIGHTS

Visual FoxPro 6.0/5.0

Jan Smith is president of PC Data Systems, Inc., located in New Brighton, Minnesota. He is a former Design Engineer who develops Visual FoxPro applications serving a variety of business needs. 651-636-2993; jsmith6@uswest.net; CompuServe: 70324,2034.

Jonathan Meed is a design engineer for Independent Technologies, located in New London, Minnesota. wesrock@midstate.tds.net.

for filters and for graphs. Because of this multiple-use requirement, we put most of the variables for each configuration—except for description, owner (user ID), type (export, filter, or graph), and date/time—into a single memo field. This gave lots of flexibility during the design and coding of the form. If you need another variable, simply write it into the code, without having to modify the structure of the CONFIG.DBF table (table 1).

Table 1: Structure of CONFIG.DBF—Stress export configurations, also configurations for filters and graphs.

Field Name	Type	Width
DESCRIPT	C	25
OWNER	C	10
TYPE	C	10
UPDATED	T	8
DATA	M	4

The flip side is that it takes a bit of processing to save and restore the variables from the memo field. In the old days, you had to use public variables and hope you didn't have any name collisions. In these days of object-oriented programming, we can use a combination of local variables and properties of the form. Besides, when a program restores variables from a memo field, even variables that were originally public come back as private. Local variables, existing only within the current method, get restored as local.

A word about the framework

We use the Visual ProMatrix framework for our applications. ProMatrix has an application object named oApp and a security object named oSec. When a user logs into the program, property oSec.cCurrentUserID is set to the ID of the current user. The oApp object is an application-level object that is instantiated when the program runs. Object oApp.oToolbar refers to the toolbar for the active form. Depending on who the current user is, and who the owner of the current record is, various controls on the export form are enabled or disabled. To make this form framework-independent, all such references are inside IF/ENDIF statements that test to see if the objects exist. If they don't exist, the form leaves everything enabled.

Likewise, all code for adding, copying, saving, or deleting records from the configuration table is in the framework's form classes. This issue's PROFESSIONAL RESOURCE CD contains the generic export form (also available at <http://www.Advisor.com/>) and a special version customized for ProMatrix.

Our form selects lists of tables and fields from data dictionary files. ProMatrix users won't have to change anything. Otherwise, just create TABLES.DBF and FIELDS.DBF, with the structures shown in tables 2 and 3.

Add your own file and field information. In the Fields table (table 3), the Order field is for entering a number indicating the order in which the fields are to be listed. If you have your data tables in a \DATA directory below your program directory, you can leave the Location field empty in the Tables file.

REVVED & READY

With VISUAL MAXFRAME PROFESSIONAL 4.0

Take your Visual FoxPro® 5.0 and 6.0 application development efforts to the next level with **Visual MaxFrame Professional (VMP) 4.0!**

VMP 4.0 is loaded with new features abstracted into the object oriented framework that your applications can implement immediately via inheritance.

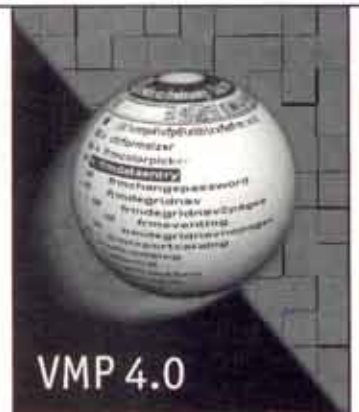
Major enhancements over VMP 3.0 include:

- client/server services
- optional security system
- revised data-entry form "action" management
- new and enhanced developer tools
- improved performance
- new classes and routines
- more examples

VMP 4.0 continues the tradition of outstanding flexibility, reliability, and ease-of-use. Continue to develop in your own style, including any combination of .SCX- or .VCX-based forms. VMP is written exclusively in Visual FoxPro® and includes complete source code, extensive documentation, and an example application. Custom on-site training and project mentoring is available by experienced VMP developers, including VMP architect Drew Speedie.

For information on how to order the latest downloads, VMP job opportunities, and much more, point your browser to www.maxlink.com/vmpmain.htm

Visual FoxPro® is a registered trademark of Microsoft Corporation.



VMP 4.0

META-MOR
INDUSTRY SOLUTIONS

14170 Newbrook Drive, Suite 100
Chantilly, VA 20151
1-888-662-9832
1-703-631-6520 fax

Table 2: **Structure of TABLES.DBF**—For non-ProMatrix users; holds information about tables from which to export data.

Field Name	Type	Width
FILE_NAME	C	254
LOCATION	M	4
DESC	C	30

Table 3: **Structure of FIELDS.DBF**—For non-ProMatrix users; holds information about fields from which to export data.

Field Name	Type	Width
FILE_NAME	C	254
FIELD_NAME	C	128
ORDER	N	30

If you aren't using long file names, you can shorten the widths of the File_name fields.

One last framework-related topic is the variable naming convention. All variables that are part of the framework begin with the letter "s," for "system." The authors of ProMatrix suggest developers prefix their variables with the letter "d," for "developer." For example, dlcFileName indicates developer, local, character, filename.

Custom form properties

The code uses several custom properties of the form. They're listed in table 4.

Managing configurations

Moving data between the memo field, Config.Data, and the form properties is a key part of the process. Two custom methods are used. The Presave method executes near the beginning of the save process, prior to updating the table. It declares the variables as LOCAL, copies values to them from properties of the form, then uses SAVE TO MEMO to put the local variables into the memo field. The properties are, of course, available throughout the form, but they can't be saved directly into the memo field.

At this point, the presave routine returns control to the save routine, to finish saving the data.

The procedure for reversing the process—restoring the data from the memo field to the variables, then back to the form properties—is equally simple. The method is named RestoreCfg. The code for both is quite straightforward; you'll find it in the form available at <http://www.Advisor.com/>.

The supporting cast

Before describing the export process, let's take a quick look at the supporting features, such as choosing a source file or setting the record order. The code for these isn't shown here, but is, like PreSave and RestoreCFG, included in the form on the Web site and PROFESSIONAL RESOURCE CD.

Continued

object-oriented²

The Mere Mortals Framework™

- Flexible architecture based on solid object-oriented design
- Guides you in creating well-designed, object-oriented applications
- View UML diagrams of Framework components in Rational Rose or Visual Modeler
- Fully object-oriented and FAST!

Oak Leaf Enterprises
The Object-Oriented Specialists
804-979-2417
www.oakleafsd.com

Easy Data Export

Table 4: **Custom properties**—These properties were added to the export form.

Property	Description
ltsOwner	Set to True if current user is owner of export configuration.
cSourceFile	Name of the file from which to export.
cExportFile	Name of the file to hold the exported data.
cOrderExpr	Order expression for the record currently being edited.
nRecCount	Number of records to export.
nCurrentRec	Current record number in CONFIG.DBF; used to set the dynamic background color in the grid.
cFilterExpr	Filter expression.
aExportFileType	Array of export file types.
aTableList	Array of table names.
aTagList	Array of index tags and descriptions for the source files.
cCurrentUserID	ID of the current user.

In a network environment, many users may have created export configurations. The Exports to Show options at the top right of figure 1 let you show all export configurations in the grid, or just those for the current user.

Choose data source

Before you export data, choose the data to export. The Source File combo box provides a list of table or view names to choose from. When you choose a file, the relative path and file name appears in the text box below the combo box.

When you choose a file from the Source File combo box list, the InteractiveChange method runs a SQL command to create an array of field names from the data dictionary files. It also calls another method to create an array of index tags, again from a data dictionary file. These arrays fill the All Fields mover list and the Index combo box.

Select order

Next, use the Index combo box to specify an order for the exported data. The choices are limited to index tags, because the Export form works by setting an order using an index tag, setting a filter, then copying records in the specified format.

Filtering the data

That brings us to the subject of filter expressions. The Build Filter Expression button opens the FoxPro expression builder, letting you create a filter. These expressions must be appropriate for a SET FILTER command.

Selecting fields

When you first choose a source file, the All Fields list contains a list of all fields in the table. By clicking on the buttons between the All Fields list and the Selected Fields list, you can choose the fields to include in the export file.

Continued

VFP 2000

Accounting Software

Written in MS Visual FoxPro 5.0/6.0

VFP 2000 Accounting Software Includes

General Ledger	Sales Quotes
Accounts Receivables	Sales Orders
Accounts Payables	Sales Invoicing
Inventory Control	Point of Sale
Lot Tracking	Freight Forwarder
Bill of Materials	Warehouse Control
Purchase Orders	Messaging System
RFQ	Internet Enabled
EDI	HTML on-line Help

Complete system prices starting at \$895
Source Code Included!

Download a FREE DEMO

www.fermen.com

Fermen Corporation

3785 NW 82 Ave. Suite 208, Miami, FL 33166

Phone: 305-477-1744 Fax: 305-477-2014 email: info@fermen.com

Double Impact Workshop Tour

Hole up with renowned experts **Rick Strahl** and **Markus Egger** for two days of intensive training in Visual FoxPro and Visual Studio technologies. Find out about the latest in Internet and COM technologies, XML, Object Modeling and much more!

Blasting into a city near you,
check out www.eps-software.com
for complete tour dates, times
and locations.

**It's a direct hit for your advanced
development education needs.**

Y2K FOX™

The Instant Y2K
Fix for Legacy
xBase applications!

It's not too late to fix the Y2K bug in FoxPro, SBT, Clipper or dBase!

Y2KFOX safely and instantly fixes the Y2K bug by automatically adjusting all 2-digit years to the correct century, as data is entered.

Y2KFOX also minimizes data entry errors, by translating 2-digit years, even if you have "Set Century On".

Call for a demo, and find out why FoxPro Advisor calls Y2KFOX "an elegant tool that offers magic for solving Y2K problems."

- Recognizes 2/29/00.
- Fixes the Y2K bug in minutes, with no source code changes.
- Works with CTOD and during data entry.
- For FoxPro, Clipper or dBase.
- Ask about our Y2K utility for SBT!
- Expert Y2K Consulting Services available.

★ Call 800.480.1604 or 203.656.0448

www.y2kfox.com

From Communication Horizons, established 1986

NetLib®

Stop placing your
client's data at risk!

Security Encryption Component

Protect confidential or proprietary data from
use outside the authorized application

- Data stays encrypted on disk *at all times!*
- No need to reveal the password to the user.
- Report-writer friendly.

NetLib 5.0 SEC protects Visual FoxPro, ABRA, and other 32-bit database applications.

NetLib 3.0 SEC protects FoxPro Windows and other 16-bit database applications.

NetLib 2.5 protects FoxPro DOS databases.

★ Call 800.480.1604 or 203.656.0448

www.netlib.com

Easy Data Export

Where should I put it?

Now that you've chosen a source table, set the record order, created a filter, and selected the fields, where do you want the export file to go? The Export File button and the text box below it help you answer that question. The button opens the FoxPro file selector dialog for you to choose the drive, path, and file name. Or, simply type in the information.

Spreadsheet or text file?

Next, choose the type of output for your export file. The Export File Type combo box lets you do that. The list includes all file types supported by the FoxPro COPY TO command. We suspect some of these are a bit obsolete (quick! when was the last time you used a MOD, DIF, or SYLK file format?), but we put them all in anyway. Please feel free to take out the ones you don't plan to use.

In the LOAD method of the form, fill the array property aExportFileType with information about each export data type. The first column has the description. The middle column has the type as used in the FoxPro COPY TO syntax, and the third column has the actual extension for the type of file. Here's the code to load the first row of the array. The remaining rows of the array contain references to the other data types listed in Help for the COPY TO command.

```
DIMENSION Thisform.aExportFileType[12,3]
```

```
Thisform.aExportFileType[01,1] = "Database (dbf)"
```

```
Thisform.aExportFileType[01,2] = "DBF"
```

```
Thisform.aExportFileType[01,3] = "DBF"
```

Reality check

Just before exporting, it's nice to make sure the export configuration will give you the expected results. Wondering how many records will be exported? Just click on the Records to Export button. Would you like to see the data before exporting? Not only will the Browse button show you the records you chose, it will show only the fields you chose, in the proper order.

The main event

The export process takes place when the user clicks on the Export button. The click method calls a form-level method, DoExport. There, the code checks to make sure there are output fields selected, there is an output file specified, the output directory exists, and so forth. Once those details are verified, the program constructs a string of field names from the Selected Fields list, copies values from form properties to local variables, asks if the user wants to overwrite an existing file (if there is one), then exports the data (listing 1).

Listing 1: Putting it all together—This code runs when the user clicks on the Export button.

```
*****  
* FRMEXPORT.DOEXPORT  
* Author: Jan Smith  
* Created: 03/18/1998  
* Purpose: Export the data.  
*****  
*  
* Key Variables:  
* dlcExportFile - export file name  
* dlcFieldList - field list  
* dlcFileType - export file type  
* dlcFilterExpr - filter expression  
* dlcOrderExpr - order expression
```

```
LOCAL dlnCurrArea, dlcCurrValue, dlcExportFile, ;  
      dlcFieldList, dlcFileType, dlcFilterExpr, ;
```


Easy Data Export

```

dlnOrderExpr, dlnListCount, dllContinue

dlnCurrArea = SELECT(0)
dllContinue = .T.

* Check to make sure there are fields selected.
IF THISFORM.lstDestination.LISTCOUNT = 0
  * No fields selected.
  =MESSAGEBOX("No fields are listed in the " ;
    + "'Selected Fields' list. " ;
    + "There is nothing to export.", ;
    MB_OK+MB_ICONEXCLAMATION, MB_CAPTION)

  RETURN
ENDIF

* Check for empty destination file.
IF ALLTRIM(THISFORM.txtExportFile.VALUE) == ""
  * No export file name entered.
  =MESSAGEBOX("There is no export file name " ;
    + "entered.", MB_OK+MB_ICONEXCLAMATION, ;
    MB_CAPTION)

  RETURN
ENDIF

* Check the export drive.
DO WHILE .T.
  dlnSpace = DISKSPACE(JUSTDRIVE( ;
    THISFORM.txtExportFile.VALUE))
  IF dlnSpace = -1
    *
    * There's an error; ask the user if he wants
    * to try again.
    *
    IF MESSAGEBOX("The drive is not available. " ;
      + "Try again?", MB_YESNO+MB_ICONQUESTION, ;
      MB_CAPTION) = IDNO

      RETURN
    ENDIF
  ELSE
    EXIT
  ENDIF
ENDDO

* Check the export path.
dlnPath = JUSTPATH(THISFORM.txtExportFile.VALUE)
IF NOT DIRECTORY(dlnPath)
  *
  * The directory doesn't exist. Ask if they want
  * to create it.
  *
  IF MESSAGEBOX("Directory doesn't exist. " ;
    + "Create it?", MB_ICONQUESTION+MB_YESNO, ;
    MB_CAPTION) = IDYES

    * In case this is a 'long' path name (with
    * embedded spaces), we must put quotes around
    * the drive\path string.
    *
    dlnPath = "" + dlnPath + ""
    MKDIR &dlnPath
  ELSE
    RETURN
  ENDIF
ENDIF

dlnFileType = THISFORM.aExportFileType[ ;
  ThisForm.cboFileType.ListIndex,2]
dlnFilterExpr = THISFORM.cFilterExpr
dlnOrderExpr = ALLTRIM(THISFORM.cOrderExpr)

* Construct a field list from the
* 'Selected Fields' list object.
*
dlnListCount = THISFORM.lstDestination.LISTCOUNT
dlnFieldList = ""
FOR dlnCt = 1 TO dlnListCount
  dlnFieldList = dlnFieldList ;
    +ALLTRIM(THISFORM.lstDestination.LIST[dlnCt]);
    +IIF(dlnCt < dlnListCount, ",","")
NEXTFOR

```

Continued

GREAT EDITORIAL COVERAGE
IS LIKE A BEAUTIFUL SUNSET,
YOU WANT TO SEE IT AGAIN & AGAIN



When your company, product, or service receives coverage in ADVISOR's magazines, capitalize on the exposure by purchasing reprints of those articles and

reviews. Use reprints as marketing or promotion tools through direct mail, trade shows, press kits, sales kits and educational sessions.

CALL TODAY AND BASK IN THE GLOW OF A GREAT REPRINT!

800-336-6060 858-278-5600
adsales@Advisor.com
www.Advisor.com

ADVISOR
MEDIA

Take off!

You've spent your time surfing the Web - now it's time to take off and scale new heights in Web development!

- Use visual design tools and the Interactive IDE to build Web applications.
- Break down complex logic into logical, reusable components.
- Use pre-built toolbox objects to drag and drop functionality into Web requests.
- Full FoxPro code support in code snippets, HTML scripts, classes and reports.
- Extend tools like Web Connection, Active Server Pages, FoxISAPI and any other FoxPro or COM compliant tool with Web functionality.

visual
WebBuilder

www.visual-webbuilder.com

Designed by Markus Egger and Rick Strahl, an EPS Software and West Wind Tools production.

```

* Got the field list; let's do it!
*
dlcSourceFile = ALLTRIM(THISFORM.cSourceFile)
dlcAlias = JUSTSTEM(dlcSourceFile)

IF USED(dlcAlias)
  SELECT (dlcAlias)
  dlcSourceWasOpen = .T.
ELSE
  SELECT 0
  USE (dlcSourceFile) SHARED
  dlcSourceWasOpen = .F.
ENDIF

IF NOT dlcFilterExpr == ""
  SET FILTER TO &dlcFilterExpr AND NOT DELETED()
ELSE
  SET FILTER TO NOT DELETED()
ENDIF

IF dlcOrderExpr <> "(No Order)"
  SET ORDER TO &dlcOrderExpr
ENDIF

GOTO TOP

dllSafetyOn = (SET("SAFETY") = "ON")

* Get the row number (dlnRow) of the file type,
* in the 'aExportFileType' array.
*
dlnElementNum = ;
ASCAN(THISFORM.aExportFileType, dlcFileType)
dlnRow = ASUBSCRIPT(THISFORM.aExportFileType, ;
dlnElementNum, 1)

* Check the file name to see if it has an
* extension. We need to do this in case the user
* entered a file name without an extension. In that
* case, the 'IF FILE(&dlcExportFile)' expression
* below might not detect an existing file. That's
* because, although the 'copy to...' command knows
* (from the file type) what extension to use, the
* 'if file...' command doesn't.

* Make sure it's the proper file extension.
*
dlcExportFile = ForceExt(THISFORM.cExportFile, ;
THISFORM.aExportFileType[dlnRow, 3] )

* Now put single quotes around the file name,
* in case this is a long file name.
*
dlcExportFile = "'" + dlcExportFile + "'"

IF FILE(&dlcExportFile) AND dllSafetyOn
  dllOverwrite = MESSAGEBOX(dlcExportFile ;
+ ' already exists. Overwrite it?', ;
MB_ICONQUESTION+MB_YESNO+MB_DEFBUTTON2, ;
MB_CAPTION) = IDYES

  IF dllOverwrite
    SET SAFETY OFF
  ELSE
    WAIT WINDOW "Export Canceled!" TIMEOUT 1.5
  ENDIF
ENDIF

ELSE
  dllOverwrite = .T.
ENDIF

IF dllOverwrite
  WAIT WINDOW "Exporting data..." NOWAIT

  IF dlcFileType = "DBF"
    * DBF files must not have the 'type' clause.
    COPY TO &dlcExportFile ;
    FIELDS &dlcFieldList
  ELSE
    COPY TO &dlcExportFile ;
    FIELDS &dlcFieldList ;
    TYPE &dlcFileType
  ENDIF
  WAIT WINDOW "Done exporting data!" TIMEOUT 1.5

```

```

ENDIF
IF dllSafetyOn
  SET SAFETY ON
ENDIF

IF NOT dllSourceWasOpen
  USE IN (dlcAlias)
ENDIF
SELECT Config

```

The code shown is for VFP 6. In VFP 5, load the FoxTools library early on.

Exporting yourself

In this article, we've given you most of the highlights and many of the details. As noted above, the PROFESSIONAL RESOURCE CD has two versions of the form. One, EXPORT.SCX/SCT, is based only on the FoxPro base classes. The other, EXPORTPM.SCX/SCT, uses the ProMatrix foundation. If you don't have ProMatrix, modify the base-class form to use your class libraries.

Thanks to Shar Feldheim and Jason Landry for testing the two versions of the form.

Fellow programmers, freedom is at hand! Escape now from the never-ending task of servicing users' insatiable appetites for data to slice and dice. Get them into the export business, and life will be better for all of you. **ADVISOR**

West Wind
TECHNOLOGY
powerfully powerful. That's reality.

Web Connection
Making Waves on the Web

Making Waves on the Web with Visual Fox Pro

Web Connection includes everything you need to build flexible and scalable Web applications now.

- Concentrate on building your app, not figuring out the Web.
- Quickly generate HTML from your data.
- Use scripting to mix HTML and Fox code.
- Render Visual FoxPro forms as Dynamic HTML.
- Use easy tools for XML conversions, PDF output from reports and distributed HTTP applications.

Best of all... it's easy to get started with quick set-up, extensive documentation, examples and a quick start tutorial.

For a free demo or more information, contact: www.west-wind.com

Where do you want to surf today?