The Codebook Newsletter

August 1999
Volume 3, Issue 2

Contents

[Editors Note]

[Enhancing the CFOXTOOLS Class]

[Creating Useful Diagrams with Rational Rose]

[Download this month's .zip file]

NEW! As you've probably noticed, the Codebook Newsletter is now being published in a more web-friendly format. Please let us know what you think.

There are 2 articles in this issue. Our first article, **Enhancements to cFoxTools Class Definition**, describes the cFoxTools class definition in a new light. The second, **Creating Useful Class Diagrams with Rational Rose**, describes some tricks concerning how to use Rose packages to create more meaningful and descriptive class diagrams.

VFP 6.0's New Functionality. The initial purpose of the cFoxTools class definition was to provide a wrapper around FoxPro's FOXTOOLS.FLL library. Before the release of Visual FoxPro Version 6.0, FOXTOOLS.FLL provided some very handy file operation functions that did not exist natively in the VFP language. Fortunately, the Microsoft developers integrated some of these functions into VFP.

Encapsulation: Old Foxes, New Tricks and Old Man Time.
Encapsulation deals with (among other things) the concept of hiding the complexity of implementation within the confines of a common and unchanging interface. Let's start with a few facts...

Using Rational Rose to Present High Level Framework Components. When dealing with a new framework one of the first questions to ask is, "What are the components of this thing"? Rational Rose provides a means to answer this question through the use of a construct known as a package.

Flash Highlights

Read about the impact that the Flash methodology has on other companies.
CLICK FOR DETAILS

Please use our feedback form to submit questions and comments.

CONTENTS FEEDBACK CONTACT US SEARCH

Flash Creative Management, Inc.

Delivering Insight and Information

What's New

About Flash

Working with businesses to jointly map their strategies and build the supporting processes and technology to achieve their growth and profit objectives.

Cybrary

Careers

In The News:

November 1, 2000 - Flash has been acquired by GoAmerica, Inc.

To read the press release, click here. To read Yair Alan Griver's letter to Friends of Flash, click here.

Services

Codebook

CONTENTS FEEDBACK CONTACT US SEARCH

This page is best viewed with:

Flash Creative Management, Inc.

Home | What's New | About Flash | Services | Cybrary | Codebook

What's New

Upcoming Events

Here's what's going on at Flash:

December 26, 2000

"Use ClassNavigator to Work With Programmatic Class Files" by Senior Developer Michael Emmons appears in the January issue of FoxPro Advisor.

Senior Consultant Kamal Patel will speak at XML Connections in New Orleans, Louisiana April 1-4 2001. His topics will be "Programming for the Wireless World - Part 1" and "Programming for the Wireless World - Part 2". For details click on this logo:

TAKE IT TO THE LIMIT xml connections

November 16,2000

"What is XML and Why Should I Care?" by Sr. Developer Tony Stewart appears in the December issue of FoxPro Advisor. Another of Tony's articles, "Six Steps to a Successful XML Integration Project," will appear in a future issue of Dr. Dobb's Journal.

November 1, 2000

Flash has been acquired by GoAmerica, Inc.

To read the press release, click here. To read Yair Alan Griver's letter to Friends of Flash, click here.

October 25, 2000

- "Dynamic Queries in Distributed Internet Applications," a new article by Flash Sr. Developer Beth Massi appears in the Fall issue of Component Developer Magazine. To read the article, click here. To download the source code, click here.

October 20, 2000

- Flash Sr. Developer Michael Gellis has a new article published on the DevX portion of Inquiry.com called "Visual Basic.NET, Part I Class Structures and Interfaces." This is the first of a series of articles on Visual Basic 7.0; now known as Visual Basic.NET. To read the article, click here.
- Flash Sr. Developer Michael G. Emmons has a tip to "Extend the 'Save As Class' Feature" in Visual FoxPro appearing in the November issue of FoxPro Advisor.

October 16, 2000

- Flash CIO Yair Alan Griver will speak at the SD2000 conference in Washington, DC on November 2. His topics will be "Successfully Managing a Turnaround project - A Case Study," and "A Design Strategy for Using Windows DNA in Systems Development. For information, click on the button below.

SD 2000

October 11, 2000

- COMCodebook, Flash's development framework based on COM, is now available for free download. To go to the COMCodebook section, click here.

September 29, 2000

- COMING SOON! COMCodebook, Flash's development framework based on COM, will be available for free download on October 11.

September 29, 2000

- Senior Developer Kamal Patel presented "Programming for the Wireless World" at PADD meetings September 19 and 28. Seminar notes and samples are now available for download. To download, click here.

September 22, 2000

- The October issue of FoxPro Advisor features "Microsoft.NET Technology: Developing Intelligent Information Systems," by Flash CFO Yair Alan Griver, and "Stay In Synch with a Form Manager," by Flash developer Jenny Brown.

August 25, 2000

- NEWS FLASH! - Flash teams up with Microsoft to present a full day session at Visual FoxPro Devcon, with Matthew Arnheiter and Yair Alan Griver presenting.

August 19, 2000

- "Speed Up Application Maintenance with a DCOM Server" is Sr. Developer Michael G. Emmons' latest article in FoxPro Advisor. Save valuable bandwidth and still let users pack and reindex tables with this method.

August 11, 2000

- The third part of Sr. Developer Rick Hodders article "Zipping and Unzipping for Clients" appears in the September issue of FoxTalk.

August 2, 2000

- Sr. Consultant Edward Ziv's newest article, "Avoiding the Iceberg" appears in the Management Forum of the September issue of Software Development magazine. In his article Ed discusses five signs of imminent failure on a development project, and how to change course to avoid disaster. To visit Software Development's online edition, click here.

July 24, 2000

- Sr. Consultant Tony Stewart's newest article, "Six Steps to a A Successful XML Integration Project," tells how to avoid the "land mines" and improve the odds of success using XML. To read it, click here.

July 15, 2000

- Flash CIO Yair Alan Griver and Senior Developer Rick Hodder will speak at FoxPro Devcon 2000 in Miami Beach September 24-28. Yair will speak on Windows DNA and the Front End, and Windows DNA and the Middle Tier. Rick will speak on Patterns 1 & 2. For details click here.

- Senior Consultant Tony Stewart will speak at XML Connections in Scottsdale, AZ October 5-7. His topics will be "What is XML and Why Should I Care?" and "Building a Client-Side Interactive XML Application." For details click here.

July 1, 2000

- Jenny Brown's article "Build Multiple Row-per-Record Grids - Part II" appears in the July Issue of FoxPro Advisor Magazine.

June 24, 2000

- David Kinne's PowerPoint presentation from the ACT conference on Captiva Island, Florida gives you the essentials to design an workable website in an hour.

June 14, 2000

- Flash CIO Yair Alan Griver will appear on New Jersey News 12 "Small Business News" this Saturday and Sunday, June 17 & 18, discussing Extranets.

June 10, 2000

- Our latest book, "Visual Basic Developer's Guide to UML and Design Patterns," by Yair Alan Griver, Matthew Arnheiter, and Michael Gellis, is now available from Sybex. Click here to read more about it or to download the source code.

June 2, 2000

- Jenny Brown and Michael Emmons are both featured in the June issue of FoxPro Advisor Magazine. Jenny's article is "Build Multiple Row-per-Record Grids," and Michael's is "8 VFP-to-Oracle Integration Tips." Yair Alan Griver is featured in the Spring 2000 Code Component Developer Magazine with "Windows DNA Development, A Pattern Language," and Rick Hodder is featured in the June Foxtalk with "Zipping and Unzipping Files for Clients."

May 30, 2000

- Michael Kemp's new article "Listening Skills Save Time, Increase Effectiveness" is now available in the Business Articles section of our Cybrary.

May 11, 2000

- Five articles on Planning an Extranet Strategy, originally presented at our Executive technology Seminar in March, are now available in our Cybrary. Read "Overview" by Yair Alan Griver, "Rationale" by David Kinne, "Planning" by Eric Zimmerman, "Community" by Ed Ziv, and "Technology" by Tony Stewart.

April 17, 2000

- Ed Ziv's article on Bias Management in Facilitated Teams, originally published in the Professional Journal of the AFSM, is now available in the Business Articles section of our Cybrary.

March 27, 2000

- Senior Developer Kamal Patel's FlashParse presentation for PADD, documents, and samples are now available in our download section.

  FlashParse is a reporting COM component which can be used in any language which supports COM.
  It simplifies the process of generating complex reports allowing users to design reports in any HTML Editor. Samples for Visual FoxPro, Visual Basic, XML and ASP are included.

February 25, 2000

- Business Technology Consultant David Kinne will appear on Bloomberg Small Business with Erikson Blakney Saturday March 11 (USA Network, 6am EST). They discuss David's article "Employee Compensation: What Gets Rewarded Is What Gets Done," which has been published in several national publications. Learn how a values driven goals/rewards matrix can motivate employees to outstanding performance, and provide for fair and objective annual reviews.

January 28, 2000

- CIO Yair Alan Griver will address NJ PADD 2/24 on "Windows DNA (Distributed interNet Architecture) Part II. This session will review the theoretical underpinnings of a Windows DNA framework, and then will explore working samples using shared components with both a thick, VFP client and a thin, ASP-based web client.

  On 2/29 Yair will also speak at NYC PADD, delivering Part I of his DNA presentation. This was rescheduled from the 1/25 PADD meeting which was postponed due to snow.

January 26, 2000

- Sr. Developer Drew Georgopolus will speak on SQL Server 7 Data Management Objects (SQL DMO) at SQL Connections 2000 in New Orleans, May 2000

CIO Yair Alan Griver will speak on "DNA (Windows Distributed interNet Architecture) and the Front End" and on "DNA and the Data Tier" at VFP DevCon 2000 in New Orleans, May 2000.

- Sr. Developer Joe Lax's article on SQL 7.0 data security over the web appears in the February VBPJ (Visual Basic Programmer's Journal.)

December 29, 1999

- Sr. Developer Drew Georgopolus will speak on SQL Server 7 Data Management Objects (SQL DMO) at NYC PADD on February 29, 2000.

  The meeting is at the Microsoft Office in New York City (50th Street between 8th and 9th avenue)

December 27, 1999

- "**DCOM Security**," by Flash CIO Yair Alan Griver and Sr. Developer Matthew Arnheiter appears in the January 2000 issue of **Component Advisor** magazine. It discusses the technical issues of running DCOM across a firewall.

December 10, 1999

- Tony Stewart spoke at **XML '99**, December 5-9 in Philadelphia. Read his paper "**Building a Client-Side Interactive XML Application**."

  This paper provides a detailed case study describing how one can merge XSL with declarative descriptions of the desired runtime behavior, develop an XML-based activity language to describe that behavior, build an XSL processor to integrate styling with behavior, build a lightweight browser add-on that will host the XSL processor, and develop style sheets, action sheets and supplemental scripts that will create the Web front end for this project.

November 29, 1999

- In his most recent article, "**Visualization Works In Reaching Business Goals**," Flash Business Technology Consultant Michael Kemp discusses a few of the advanced techniques available for setting goals. One of the surprises with these techniques is that their diligent use can program the mind, effectively turning goals into accomplishments.

November 22, 1999

- Read the latest article by Flash Business Technology Consultant David W. Kinne, **"Employee Compensation: What Gets Rewarded Is What Gets Done,"** published in the February "Human Capital Strategies and News," and featured on TV on Bloomberg Reports Small Business News in February. Learn how a values driven goals/rewards matrix can motivate employees and provide for fair and objective annual reviews.

November 8, 1999

- The premiere edition of Exchange and Outlook magazine includes an article by Flash Software Developers Matthew Arnheiter and Michael Gellis on **"Programming Outlook and CDO."** Matt and Michael explain how, together, Outlook and Exchange can accomplish far more than just executing macros. Learn how to tap the power of this complete development platform.

October 22, 1999

- An article by CEO David Blumenthal and Senior Consultant Ray Stoddard appears in the October, 1999 issue of PM Network Magazine: The Professional Magazine of the Project Management Institute. The title of the article is **"Implementation Planning: The Critical Step."**

October 18, 1999

- Tony Stewart, Flash's newest employee, will be speaking at XML '99, December 5-9 in Philadelphia at the Pennsylvania Convention Center about building a client-side interactive XML application.

October 5, 1999

- The SQL Pro's ten minute solution at DEVX.COM this month, entitled "**Addressing User Comments About Performance Issues**," was written by Sr. Consultant Drew Georgopulos.

September 30, 1999

- CIO Alan Griver will be hosting November's NJ PADD meeting. Click here for details.

September 27, 1999

- If you're a FoxPro programmer, don't miss Sr. Consultant Kamal Patel's article, *Making Your Grids More Intelligent*.

September 14, 1999

- Sr. Consultant Kamal Patel will be hosting this month's NJ PADD meeting. Click here for details.

September 10, 1999

- The SQL Pro's ten minute solution at DEVX.COM this month, entitled "Create Your Own Identity," was written by Sr. Consultant Drew Georgopulos. It shows developers how to control the behavior of surrogate key generators without the headaches and overhead incurred with the identity type.

September 9, 1999

- Sr. Consultant Joe Lax's article, "SQL Service: Select the Right Join", appears in the October 1999 issue of the *Visual Basic Programmer's Journal*. Available now on newsstands.
- Our CEO, David Blumenthal, was quoted in the September 1999 edition of Beyond Computing in Ed Trapasso's article, "The Outsourcing Option".

August 31, 1999

- Notes and Downloads are available from Kamal Patel's Speed Issues of Application Development Using VFP.

August 24, 1999

- A new issue of **The Codebook Newsletter** has been posted.

August 23, 1999

- A new case study has been posted on our site. Find out how Flash has helped Remarketing Services of America, Inc. (RSA), the nation’s largest independent lease maturity management and vehicle remarketing company.
- Our business articles section has been updated to include a new article on Marketing Your Internet Strategy.
- A new Codebook Bug Tracking System has just been released. Developed internally by Sr. Consultant, Matthew Arnheiter. Written using Microsoft Exchange Server 5.5 and Collaboration Data Objects (CDO).

August 5, 1999

- Our business articles section has been updated to include three new articles on planning an Internet strategy.

July 19, 1999

- *PC Week* has quoted extensively one of Flash's experts -- on how to improve group dynamics. For details on what Senior Business Technology Consultant **Ed Ziv** has to say about improving group thinking, check out *Everyone But Me is Biased* in the July 19 edition of *PC Week Online*.

July 16, 1999

- Sr. Consultant Michael Gellis and CIO Yair Griver published a new book entitled "MCSD Test Success Visual Basic 6 Distributed Applications". Matthew Arnheiter, Sr. Consultant, was also a contributor.
- The Notes and Downloads page has been updated with the latest information.
- Drew Georgopulos, Sr. Consultant, has a new article about using SQL entitled "Managing Repetitive Jobs Without Cursors".

May 25, 1999

- Notes and Downloads are available from Kamal Patel's XML Presentation.

May 17, 1999

- Do you want to further develop yourself? Then visit the Flash Cybrary for an updated and improved listing of recommended business books.
- Updated Events Schedule

April 23, 1999

- A New Codebook Newsletter has recently been published and released and is now available for download, for free. It covers such topics as the Template Method and creating a One-to-Many Business Object in Codebook.

Please use our feedback form to submit questions and comments.

CONTENTS FEEDBACK CONTACT US SEARCH

About Flash

Flash has proven expertise in:

- Assisting in the development of strategic directions
- Aligning your business processes and information systems with your strategy
- Building COM-based solutions that support your strategic goals
- Applying object-oriented analysis and design techniques
- Delivering Year 2000 compliant mission-critical web and database applications
- Mentoring and training your staff to develop data warehousing systems
- Integrating seamless communications
- Developing systems for Executive Information Support and Real-Time Transaction Processing

Flash helps you by providing:

- Certified and talented resources to get the project done
- Objective opinions to insure that you are moving in the right direction
- Resolution to complex technological problems
- Guidance to improve and align processes
- Development and maintenance of mission critical systems
- Management of extremely large volumes of data

Flash Creative Management utilizes the latest business and software tools and techniques to create solutions. Our Codebook Development Framework is used throughout the United States and Europe to help corporations meet their strategic objectives.

Recognized Accomplishments and Awards:

- Microsoft Solutions Provider at the Partner Level
- Microsoft Authorized Technical Education Center
- Inc. 500 Company Fastest Growing Privately Held Companies - 1995
- Inc. 500 Company Fastest Growing Privately Held Companies - 1994
- Inc. 500 Company Fastest Growing Privately Held Companies - 1993
- Deloitte and Touche National Technology Fast 500 - 1996
- Deloitte and Touche National Technology Fast 500 - 1995
- Deloitte and Touche New Jersey Technology Fast 50 - 1996
- Deloitte and Touche New Jersey Technology Fast 50 - 1995
- Upside Magazine 140 "Emerging Stars" in Integrating Technology - 1994
- Flash professionals have passed more than fifteen (15) Microsoft Certification exams in database languages, network operating systems
  and communication tools including Visual Basic, Visual FoxPro, SQL Server,
  Windows NT, and Exchange
- Several Professionals with Microsoft's Most Valuable Professional Awards

Mission Our Approach Our People
Philosophy Client List Case Studies
Testimonials

Please use our feedback form to submit questions and comments.

CONTENTS FEEDBACK CONTACT US SEARCH

Flash Creative Management, Inc.

Home | What's New | About Flash | Services | Cybrary | Codebook

Services

Strategic Planning

Process Design

Net Strategy

Product Development

Project Management

Mentoring

Class Libraries

COM

Web Design

Code Reviews

RFP Design

Management Consulting Services

- Strategic Planning and Consulting
- Business Process Analysis, Design and Redesign
- Project Management and Implementation Planning
- New Product Development
- Internet / Extranet / Intranet Strategic Design
- Mentoring and Training

Technology Consulting Services

- COM Based Analysis, Design and Client/Server Development
- Web Site Design and Development
- Development of Class Libraries
- Software Code Reviews
- Request for Proposal Design and Software Search

Please use our feedback form to submit questions and comments.

CONTENTS | FEEDBACK | CONTACT US | SEARCH

Cybrary

Business Books

Business Articles

Technology Books

Technology Articles

Seminars

Notes & Downloads

Case Studies

Learn Your Way to Success

Welcome to the Flash Cybrary. This section provides links to learning materials
on Flash's site, as well as links to places we've found helpful at other sites on
the internet. In addition to a listing of (and links to) recommended management
consulting and technology books, articles, and white papers, we've also included
links to seminar notes from Flash Professionals.

Business Section

- Books
- Articles & White Papers

Technical Section

- Books
- Articles and White Papers
- The Codebook Newsletter
- Notes and Downloads from Prior Speaking Engagements

General Section

- Upcoming Events
- Notes & Downloads

Please use our feedback form to submit questions and comments.

CONTENTS FEEDBACK CONTACT US SEARCH

Codebook

Codebook World

Downloads

Partnering Programs

Newsletter

Bug Tracking System

COMCodebook, Flash's .COM- aware development framework, is now available for free download. To go to the COMCodebook section, click here.

What is the Codebook Methodology?

- A Framework for the development of strategic architectures
- A practical guide for 3-tier (3 layer), object oriented, and client-server development
- Part of Flash's technology consulting methodology
- A book and CD-ROM set

Please use our feedback form to submit questions and comments.

CONTENTS | FEEDBACK | CONTACT US | SEARCH

Careers

Overview

Do you want to:

- Work for a company that makes a difference?
- Be with a company that has a winning approach?
- Have an opportunity for great rewards?

Join us. We are looking for highly motivated individuals to do high level business analysis, object oriented development, training and business consulting. All of our positions afford an opportunity to work with the most prominent companies in America and Europe today.

Flash Creative Management professionals are lecturers at some of the leading business conferences, software conferences, and user groups; and are contributing editors to many industry trade publications. Flash invests in you so that you have opportunities to work with the latest tools and technologies. We create an atmosphere that allows you to excel professionally and a workplace which is challenging and fun.

Positions Available Now

- Senior Developer/Project Lead (2)

General Employment Inquiries

If the positions available do not match the type of position you are seeking, send us a brief statement about your professional qualifications and skills and submit it to our employment database. The next time a position becomes available, we will search our employment database and notify you of the opportunity.

All information will be kept in the strictest confidence. Flash Creative Management, Inc. is an equal opportunity employer.

Please use our feedback form to submit questions and comments.

The Codebook Newsletter

August 1999
Volume 3, Issue 2

Editor’s Comments

Charles T. Blankenship

Hello again. I know, I know. Last issue I promised to write an article describing how to create a non-GUI business object. Unfortunately, due to deadlines and slipped deadlines I have to postpone that one until next issue. Sorry.

The first article, **Enhancements to cFoxTools Class Definition**, describes the cFoxTools class definition in a new light. It also describes some new functionality built into the class that allows you to determine whether your applications instantiate the Foxtools object or not. Hopefully this enhancement addresses some of the concerns expressed on the forum. I also hope that this article broadens your perception for the use of the class definition from a simple wrapper to the FOXTOOLS.FLL into a handy toolbox for storing language enhancements for VFP that you use in your day to day development. Finally, it gives a description of one of the enhancements I use all the time.

The second article, **Creating Useful Class Diagrams with Rational Rose**, describes some tricks concerning how to use Rose packages to create more meaningful and descriptive class diagrams. This article documents what I learned after working on a large reverse engineering project where I created hundreds of class diagrams and reverse engineered a very large project. One of the applications that I used to help complete this project very quickly is Markus Eggar’s VFPMODEL.APP. This application allows for quick and easy reverse engineering of existing VFP class libraries which you can then use to create models of the application. The code for this application is included in the .ZIP file for this issue of The Codebook News. It is also available for download from Microsoft’s site. Finally, I also included the Rose model that I used to create the document in the .ZIP file as well.

See you next issue.

Flash Creative Management, Inc.

Home | What's New | About Flash | Services | Cybrary | Codebook | Careers

The Codebook Newsletter

August 1999
Volume 3, Issue 2

Enhancements to cFoxTools Class Definition

VFP 6.0’s New Functionality

Charles T. Blankenship, MCP

The initial purpose of the cFoxTools class definition was to provide a wrapper around FoxPro’s FOXTOOLS.FLL library. Before the release of Visual FoxPro Version 6.0, FOXTOOLS.FLL provided some very handy file operation functions that did not exist natively in the VFP language. Fortunately, the Microsoft developers integrated some of these functions into VFP. The result of their work manifested itself in the following enhancements to the language.

Foxtools functions	**Description**
The following functions have been added to Visual FoxPro 6.0 from Foxtools; they can now be used without executing SET LIBRARY TO FOXTOOLS.	Note that you must recompile any programs, class libraries, labels, or reports created in earlier versions of Visual FoxPro if they contain any of the following functions.
ADDBS( ) Function	Adds a backslash (if needed) to a path expression.
AGETFILEVERSION( ) Function	Creates an array containing information about files with Windows version resources such as .exe, .dll, and .fll files, or automation servers created in Visual FoxPro. Corresponds to the GetFileVersion( ) function in Foxtools.
DEFAULTEXT( ) Function	Returns a file name with a new extension if one doesn’t already exist.
DRIVETYPE( ) Function	Returns the type of the specified drive.
FORCEEXT( ) Function	Returns a string with the old file name extension replaced by a new extension.
FORCEPATH( ) Function	Returns a file name with a new path name substituted for the old one.
JUSTDRIVE( ) Function	Returns the drive letter from a complete path.
JUSTEXT( ) Function	Returns the three-letter extension from a complete path.
JUSTFNAME( ) Function	Returns the file name portion of a complete path and file name.
JUSTPATH( ) Function	Returns the path portion of a complete path and file name.
JUSTSTEM( ) Function	Returns the stem name (the file name before the extension) from a complete path and file name.

The following is a comparison between pre- and post-VFP 6.0 enhancements.

```
CFOXTOOLS::ISDIR()
```

Determines if a directory exists … replaced by the VFP **DIRECTORY()**function.

CFOXTOOLS::JUSTDRIVE()
Returns just the drive letter from a complete path … replaced by the VFP **JUSTDRIVE()** funtion.

CFOXTOOLS::JUSTFNAME()
Returns the fine name from a complete path … replaced by the VFP **JUSTFNAME()** function

CFOXTOOLS::JUSTPATH()
Returns the complete path from a fully qualified file name … replaced by the VFP **JUSTPATH()** function

CFOXTOOLS::JUSTPATHNODRIVE()
Returns the path, without the drive letter, from a fully qualified file name … no equivalent VFP function. However, this is really just a wrapper around the **CFOXTOOLS::JUSTPATH()** function followed with code that strips off the drive letter/colon. It finishes up by using the **CFOXTOOLS::ADDBS()** function to add a backslash if it is needed.

CFOXTOOLS::JUSTSTEM()
Returns the stem (first eight characters) of a file name from a fully qualified file name … replaced by the VFP **JUSTSTEM()** function.

CFOXTOOLS::ADDBS()
Adds a backslash to a file name, if needed … replaced by the VFP **ADDBS()** function.

CFOXTOOLS::CLEANDIR()
Returns a clean directly name (drive and path with no appended backslash) from a relative path … implemented using VFP's **SYS(2027)** function.

CFOXTOOLS::DRIVETYPE()
Returns a numeric value that identifies the type of drive whose letter was passed into the function via parameter … implemented using VFP's **DRIVETYPE()** function.

CFOXTOOLS::COPY()
Copies the highlighted text to the clipboard.

CFOXTOOLS::CUT()
Cuts the highlighted text to the clipboard.

CFOXTOOLS::PASTE()

Pastes the text that is in the clipboard.

The True Nature of CFOXTOOLS and IFOXTOOLS

Some important aspects of using software frameworks are 1) know where to put things and 2) know why you put them there. In this case, Codebook provides you with a class library that encapsulates language enhancements. This implies that the processes you continue to place into CFOXTOOLS/IFOXTOOLS should fit into one of two general categories. Either the native VFP language does not perform the required function at all or it performs the function but not particularly well and more robust processing is required.

Thinking Outside the Box

Do not think of the CFOXTOOLS class library as a wrapper to FOXTOOLS.FLL alone. You'll do yourself a disservice. Instead, think of it as your own personal toolbox that you use to enhance Visual FoxPro's native capabilities … TOOLS FOR FOXPRO if you will indulge me this one concept.

It is very important to realize that the methods used to accomplish these enhancements (the HOW of it all) are irrelevant. What is much more important and relevant is WHAT is being accomplished. For example, the fact that you needed to know if a directory exists has not changed over time. You needed to know if a directory existed when some of you gray-haired veterans were battling the first version of FoxBASE. The young whippersnappers of today still need to know if a directory exists when using Visual FoxPro 6.0. The only thing that has changed over time is HOW to accomplish this task using the language. With that, let's take a look at the concept known as encapsulation.

Encapsulation: Old Foxes, New Tricks and Old Man Time

Encapsulation deals with (among other things) the concept of hiding the complexity of implementation within the confines of a common and unchanging interface. Let's start with a few facts:

1. The ***need*** to determine if a directory exists has not changed in a very long time … we have always had this need.
2. The ***way*** this task is accomplished has changed from release to release
3. ***We already have an object that knows how to accomplish this task*** and most folks in the Codebook community already know how to use this object

The key point to understand is that the interface initially provided to Codebook users for determining whether or not a directory existed was initially surfaced using the **`CFOXTOOLS::ISDIR()`** method. The code used to solve this problem when Codebook shipped looked like this:

```
LPARAMETERS tcString
LOCAL llRetVal, lcTestString, laFiles[1]

lcTestString = ALLTRIM( This.AddBS( tcString )) - "*.*"

IF ADIR(laFiles, lcTestString, "DSH") > 0
  llRetVal = .T.
ELSE
  llRetVal = .F.
ENDIF

RETURN llRetVal
```

In Version 5.0 of VFP the Microsoft Developers graciously gave us the **`DIRECTORY()`** function. *This basically means that the process used to accomplish this task has changed while the need to accomplish this task remained unchanged.* This means the implementation of the **`IsDir()`** function in CFOXTOOLS.VCX ***can*** change from the above code to the following:

```
LPARAMETERS tcString

llRetVal = DIRECTORY( tcString )

RETURN llRetVal
```

In this particular situation, encapsulation hides the complexity and form of implementation behind a standardized and immutable public interface. This allows you to change the implementation code all you want … hopefully for the better … over time. In essence you can teach an old CFOXTOOLS class definition new tricks as time and progress march on while keeping your legacy code intact. But before you change an implementation you better be absolutely sure of one thing …

Assume that your boss just gave you permission to upgrade from VFP Version 3.0 to VFP Version 5.0 (hint: the **DIRECTORY()** function now exists.) You, being the astute programmer that you are, realize that the **DIRECTORY()** function is a much more efficient method of determining if a directory exists than the initial Codebook implementation and you decide to replace the original code with code that uses VFP's new functionality (you changed an implementation). Question … did this change break the following legacy code?

```
IF goApp.oFoxTools.IsDir( "C:\TEMP\" )
<do some processing>
ENDIF
```

The answer is POSSIBLY. This particular method, **IsDir()**, serves as a warning to those really anxious to change an implementation from release to release. Before you do so you must make sure that the new functionality performs the exact same job as before (assuming the original code was not broken that is) but in a more efficient manner (hopefully). Unfortunately the **DIRECTORY()** function does not detect hidden directories very well whereas the initial implementation worked just fine for this purpose. I do not advise replacing the original implementation with the **DIRECTORY()** function … it is not a logical equivalent and can break legacy code.

VFP/FOXTOOLS Equivalent Operations

Please examine the following code closely. It is Codebook's initial implementation of the AddBS() function.

```
*---------------- Location Section --------------------
*} Library: CUTILS.VCX
*} Class:   CfoxTools.VCX
*} Method:  AddBS()
*----------------------------------------------------
*) Description:
*)     Adds a backslash to the provided string
*----------------------------------------------------
LPARAMETERS tcString

LOCAL lcRetVal

lcRetVal = This.AddBS(JustDrive(tcString))
```

```
RETURN lcRetVal
```

Assume for a moment you use VFP5.0 and the FOXTOOLS library is loaded. This code uses the FOXTOOLS.FLL to perform the **ADDBS()** and **JUSTDRIVE()** functionality. Next assume that you upgraded to VFP6.0 and FOXTOOLS is not loaded. Question: Does this code have to change in any way? Answer: No! The reason is that VFP added both the **ADDBS()** and **JUSTDRIVE()** functions to VFP's native language without changing the syntax or command line. This code can remain the same from one version to the next. The only thing that changes is where VFP goes to perform the function.

Knowing this, it seemed like a good idea to load FOXTOOLS.FLL only if the developer is operating in the VFP 5.0 environment. So, we modified the LoadLib() method to check for the presence of VFP 5.0 and load FOXTOOLS.FLL only in that instance. Unfortunately, MS did not build all of the FOXTOOLS.FLL functions into VFP 6.0. This means we should have given the developer the option to load FOXTOOLS.FLL even in the VFP 6.0 environment. Knowing this, we made the following modifications to CFOXTOOLS class definition.

```
*--------------- Location Section --------------------
*} Library: CUTILS.VCX
*} Class:   CFoxTools
*} Method:  Init()
*----------------------------------------------------

THIS.SetLoadFoxToolsOption()
THIS.LoadLib()
RETURN .T. && This.LoadLib()
```

Two changes took place in the Init() method. The first is the addition of a custom method that teaches CFOXFOOLS how to determine whether or not to load FOXTOOLS.FLL. The second modification removed the return value's dependency upon the success or failure of the LoadLib() method.

The new method, SetLoadFoxToolsOption() follows. Note when and where either the environment or the developer controls the loading of FOXTOOLS.FLL. If you use VFP 5.0, FOXTOOLS.FLL must load into memory. However, if you use VFP 6.0 you have several choices. If you always want to load FOXTOOLS in all of the application you build with the Codebook framework, set the **IFoxTools.lLoadFoxTools** property to .T. If you want to be more selective, like on an application by application basis, set the **IFoxTools.lLoadFoxTools** property to .F. and set the *application's* **.lAlwaysLoadFoxTools** property to .T.

```
*--------------- Location Section --------------------
*} Library: CUTILS.VCX
*} Class:   CFoxTools
*} Method:  SetLoadFoxToolsOption()
*----------------------------------------------------
*) Description:
*)     Examines several environmental conditions to determine
*)     whether or not the developer desires to load the
*)     FOXTOOLS library into memory.
*)
*)     If developing in a VFP5.0 environment, they MUST
*)     load FOXTOOLS into memory for this functionality
*)     to work.  This fact takes priority over all
*)     property settings ... it is first in the CASE statement.
*)
```

```
*)     If developing in a VFP6.0 environment or later
*)     allow the developer to choose whether or not
*)     fox tools loads by setting properties at the
*)     application or class definition level.
*)
*)     CREATED - 05/29/1999 @ 10:55:00am - CTB
*--------------------------------------------------------
LOCAL lcVersion

***-----------------------------------------------------------
*** EGL: 8/28/98 - Changed this to work with VFP 6.0, since
***           it no longer uses FoxTools.FLL.
***-----------------------------------------------------------
lcVersion = VERSION()

***-----------------------------------------------------------
*** Check for VFP 5
*** EGL: 05/27/1999 - As it appears that there are some cases
***    where FoxTools is needed even in VFP 6, I’ve added an
***           option for always loading this library by setting a
***           property in goApp called ‘lAlwaysLoadFoxTools’.
***-----------------------------------------------------------
*!*    IF lcVersion = “Visual FoxPro 05.”
***-----------------------------------------------------------

DO CASE
CASE lcVersion = “Visual FoxPro 05.”
   *-----------------------------------------------------------
*-- For VFP to work in Version 5.0 FOXTOOLS must be loaded
   *-----------------------------------------------------------
THIS.lLoadFoxTools = .T.
CASE THIS.lLoadFoxTools
   *------------------------------------------------------
*-- Developer modified the class definition to always
*-- use FoxTools no matter what, honor that wish.
*-- (Default value for .lLoadFoxTools = .F. which
*-- allow the following conditions to determine
*-- if FoxTools loads)
   *------------------------------------------------------
THIS.lLoadFoxTools = .T.
CASE ((TYPE(“goApp.lAlwaysLoadFoxTools”) == “L”) AND goApp.lAlwaysLoadFoxTools)
   *--------------------------------------------------------
*-- If the developer wants to choose to load FOXTOOLS
*-- on an application by application basis, they should
*-- set the IFoxTools.lLoadFoxTools property to .T.
*-- and control the load from the application using
*-- the IApplication.lAlwaysLoadFoxTools property.
   *--------------------------------------------------------
THIS.lLoadFoxTools = .T.
OTHERWISE
   *--------------------------------------------------------
*-- If the developer is using VFP 6.0 and they did not
*-- specify that they wanted to use FOXTOOLS (in either
*-- one of two ways ... at the application or class
*-- definition level ) then do not load FOXTOOLS into
*-- memory
   *--------------------------------------------------------
THIs.lLoadFoxTools = .F.
ENDCASE
```

Some Changed Implementations, Cut, Copy and Paste

One of the proposed enhancements on the Codebook forum was the substitution of the **`SYS(1500)`** function for the **`_EDPASTE( _WONTOP() )`** type of code whose functionality is available only through the FOXTOOLS library. The documentation provided by VFP for **`SYS(1500)`** follows:

`Description`
`SYS(1500) activates a Visual FoxPro system menu item.`

Syntax
`SYS(1500, cSystemItemName, cMenuName)`

Returns
Character

Arguments
CSystemItemName - Specifies the name of the Visual FoxPro system menu item to activate.
cMenuName - Specifies the name of the Visual FoxPro system menu or submenu containing the menu item.

Remarks
See System Menu Names for a list of Visual FoxPro menu and menu item names. You can also use **`SYS(2013)`** - System Menu Name String to display a list of Visual FoxPro menu and menu item names.

User-defined menu items and disabled system menu items cannot be activated with SYS(1500).

`SYS(1500)` returns the empty string.

To make a long story short, **`SYS(1500)`** didn't work in all situations. It only works when the VFP menu system's EDIT pad is active. Note the information highlighted and bolded above. User-defined menu items and disabled system menu items cannot be activated with **`SYS(1500)`**. Bummer. What to do?

The answer is surprisingly simple. When FOXTOOLS library is loaded, use the old code … it's worked this long, it'll continue working for the future. When FOXTOOLS library is not loaded, use Codebook's menu system itself by KEYBOARDing the CTRL+X, CTRL+V and CTRL+C. The following lists the new implementation code for the Copy(), Paste() and Cut() methods.

```
*---------------- Location Section ---------------------
*} Library: CUTILS.VCX
*} Class:   CFoxTools
*} Method:  Paste()
*-------------------------------------------------------
*) Description:
*)     Either uses FOXTOOLS or the Codebook menu system
*)     functionality to effect a paste event into the
*)     active window.
*-------------------------------------------------------
```

```
*   Change Log:
*       CREATED Sunday, 11/05/95 11:40:39 - CTB:
************************************************************
DO CASE
CASE THIS.IsLibraryLoaded( "FOXTOOLS.FLL" )
  _EDPASTE( _WONTOP() )
CASE THIS.CodebookPasteBarExists()
  KEYBOARD '{CTRL+V}'
OTHERWISE
  SYS( 1500, "_MED_PASTE", "_MEDIT" )
ENDCASE

RETURN .T.



*---------------- Location Section ---------------------
*} Library: CUTILS.VCX
*} Class:   CFoxTools
*} Method:  Cut()
*-------------------------------------------------------
*) Description:
*)Either uses FOXTOOLS or the Codebook menu system
*)functionality to effect a cut event from the
*)active window.
*-------------------------------------------------------
*   Change Log:
*       CREATED Sunday, 11/05/95 11:40:39 - CTB:
************************************************************
DO CASE
CASE THIS.IsLibraryLoaded( "FOXTOOLS.FLL" )
   _EDCUT( _WONTOP() )
CASE THIS.CodebookCutBarExists()
   KEYBOARD '{CTRL+X}'
OTHERWISE
   SYS( 1500, "_MED_CUT", "_MEDIT" )
ENDCASE

RETURN .T.




*---------------- Location Section ---------------------
*} Library: CUTILS.VCX
*} Class:   CFoxTools
*} Method:  Copy()
*-------------------------------------------------------
*) Description:
*)Either uses FOXTOOLS or the Codebook menu system
*)functionality to effect a copy event from the
*)active window.
*-------------------------------------------------------
*   Change Log:
*       CREATED Sunday, 11/05/95 11:40:39 - CTB:
********************************************************
DO CASE
CASE THIS.IsLibraryLoaded( "FOXTOOLS.FLL" )
   _EDCOPY( _WONTOP() )
CASE THIS.CodebookCopyBarExists()
```

```
   KEYBOARD '{CTRL+C}'
OTHERWISE
   SYS( 1500, "_MED_COPY", "_MEDIT" )
ENDCASE


RETURN .T.
```

Some Enhancements

Now that you made it this far it is only fair to reward you with some tools from my FOXpro TOOLbox. Remember one of the categories into which FOXTOOLS class definition enhancements occurs … when the language accomplishes the specified task but additional processing is required to create a truly polished product? The following two methods document the processing required to generate a temporary file name. Place them into your I-Layer if you desire or complain loud enough on the forum and I'll place them in the C-Layer for all to benefit.

```
*---------------- Location Section --------------------
*} Library: IUTILS.VCX
*} Class:   IFOXTOOLS
*} Method:  XSYS2015()
*----------------------------------------------------
*-- Description:
*--           Returns a unique 10-character procedure name that
*--           begins with an underscore followed by a combination
*--           of letters and numbers.  SYS(2015) can be used
*--           to create unique procedure or function names.  The
*--           name that SYS(2015) returns is created from the
*--           system date and system time. Calling SYS(2015)
*--           more than once during the same millisecond
*--           interval will return a unique character string.
*--------------------------------------------------------
LOCAL lcUniqueString

DO WHILE .T.

   lcUniqueString = "XXX"+RIGHT(SYS(2015),7)

   IF THIS.cLastUniqueString == lcUniqueString
      *------------------------------------
      *-- Cycle around again until we get one
      *-- that is unique
      *------------------------------------
   ELSE
      *-- Store the newly generated string
      *-- for comparison next time and
      *-- EXIT the loop to return the value
      *------------------------------------
      THIS.cLastUniqueString = lcUniqueString
      EXIT
   ENDIF

ENDDO

RETURN lcUniqueString
```

```
*---------------- Location Section ---------------------
*} Library: IUTILS.VCX
*} Class:   IFOXTOOLS
*} Method:  CreateTemp()
*-------------------------------------------------------
*-- Description:
*--    Creates a temporary file name with the specified
*--    extension.  This method also makes sure that
*--    the generated file name is unique within
*-- the specified directory.
*-----------------------------------------------------------
LPARAMETERS tcExtension, ;
            tcCurrentDirectory, ;
            tcUniqueFileName

LOCAL lcExtension, ;
      lcDot, ;
      llRetVal, ;
      lcCurrentDirectory

lcDot    = "."
llRetVal = .F.

*===============================================*
*== VALIDATE AND LOCALIZE INCOMING PARAMETERS ==*
*===============================================*
IF TYPE('tcExtension') == "C" AND .NOT. EMPTY( tcExtension )
   lcExtension = tcExtension
ELSE
   lcExtension = ""
ENDIF

IF TYPE( 'tcCurrentDirectory' ) == "C" AND THIS.IsDir( tcCurrentDirectory )
   lcCurrentDirectory = THIS.AddBS( tcCurrentDirectory )
ELSE
   lcCurrentDirectory = THIS.AddBS( SYS(5) + SYS(2003) )
ENDIF

DO WHILE .T.

       *===================================*
       *== GENERATE THE UNIQUE FILE NAME ==*
       *===================================*
  DO CASE
  *--------------------
  CASE PARAMETERS() = 0
  *--------------------
       =MESSAGEBOX( "You must pass at least the return variable by reference to  " + ;
                           "IFOXTOOLS::GenerateUniqueFileName", 48, "Error")
       tcUniqueFileName  = ""
  *------------------------
  CASE EMPTY( lcExtension )
  *------------------------
       tcUniqueFileName = "_" + SUBSTR(THIS.XSYS2015(), 4, 10)
  *-----------------------------
  CASE .NOT. EMPTY( lcExtension)
  *-----------------------------
       IF LEFT( lcExtension, 1 ) == "."
```

```
          tcUniqueFileName = "_" + SUBSTR(THIS.XSYS2015(), 4, 10) + lcExtension
       ELSE
          tcUniqueFileName = "_" + SUBSTR(THIS.XSYS2015(), 4, 10) + lcDot + lcExtension
       ENDIF

  OTHERWISE
     *--------------------------------------------------------
     *-- You just got past a binary test ... why are you here?
     *--------------------------------------------------------
  ENDCASE

  *========================================================*
  *== TEST THE NEWLY GENERATED FILE NAME FOR UNIQUENESS ==*
  *========================================================*
  IF EMPTY( tcUniqueFileName )
     *------------------
     *-- Error condition
     *------------------
     llRetVal = .F.
     EXIT
  ELSE
     *-------------------------------------------------
     *-- Use the directory to determine if a file with
     *-- the generated name already exists therein.  If
     *-- so, keep executing the loop until a unique
     *-- file name comes up
     *-------------------------------------------------
     tcUniqueFileName = lcCurrentDirectory + tcUniqueFileName
     IF .NOT. FILE( tcUniqueFileName )
        llRetVal = .T.
        EXIT
     ELSE
        *---------------------------------------------------
        *-- Go around again until you get a unique file name
        *---------------------------------------------------
     ENDIF
  ENDIF

ENDDO

RETURN llRetVal
```

Using the **`CreateTemp()`** function is very simple. The signature for this method follows:

```
GoApp.oFoxTools.CreateTemp( [<cFileExtension>] , ;
                            [<cDestinationDirectory>],;
                           @<cUniqueFileName )
```

WHERE

`[<cFileExtension>]` - is the three letter DOS extension for the file. This parameter is optional
`[<cDestinationDirectory>]` - is the name of the directory into which the new file will be created. This parameter is optional … if left blank the current director is used.
`@<cUniqueFileName>` - is the return parameter and contains the name of the unique file name (if the function succeeded. Must be passed via refernce.

RETURNS

Logical .T. if the method succeeded, .F. if it did not successfully generate a unique file name.

Conclusion

Basically, FOXTOOLS.FLL is here to stay. Yes, the Microsoft Developers incorporated some of the functionality, previously available only through FOXTOOLS.FLL, but they left many of the functions out. The FOXTOOLS library is still very useful if you need one of the unincorporated functions it provides. However, if you do not need the additional functionality, there is no need to use up memory space. You now have the capability to load or ignore this library at your own discretion.

I also hope to have convinced you to expand your definition of the CFOXTOOLS class definition. It is not just for the encapsulation of FOXTOOLS.FLL but should be used as your little toolbox to enhance the functionality of FOXPRO within your own programming shop.

Appendix A: A List of FOXTOOLS Functions

On one of the lamentations expressed by Bob Archer on the Codebook forum was, “What I don’t understand is why they didn’t build ALL of the FOXTOOLS functions into 6.0”. For those of you who don’t know what those functions are, the following list enumerates them. Functions prefixed with a * came into being in VFP 5.0. Functions prefixed with a # came into being with VFP 6.0. Those functions with no prefix are only available via FOXTOOLS.FLL. Uh, they left a little out … didn’t they?

Further, this information makes James H. Hansen’s message on the Codebook Forum very clear. His complaint involved the use of the WORDNUM function … available through FOXTOOLS.FLL but not yet through VFP. Forcibly removing FOXTOOLS from loading into memory for VFP 6.0 (and greater) applications makes all of the uncommented functions below unavailable to developers.

Function	Address	
---------------------	-------------	
INITIALIZE	03471064	Call on load.
UNINITALIZE	03471078	Call on unload.
MSGBOX		
REGFN		
REGFN32		
CALLFN		
# DRIVETYPE		
* MKDIR		
* RMDIR		

GETPROSTRG

PUTPROSTRG

FOXTOUCH

RGBCOMP

# JUSTFNAME

# JUSTSTEM

# JUSTEXT

# JUSTPATH

# JUSTDRIVE

# FORCEPATH

# FORCEEXT

# DEFAULTEXT

ADDBS

VALIDPATH

CLEANPATH

REDUCE

STRFILTER

WORDS

WORDNUM

NEXTWORD

_EDSETPOS

_EDSKIPLIN

_EDGETPOS

_EDGETLPOS

_EDGETLNUM

_EDPOSINVI

_EDSTOPOS

_EDSTOSEL

_EDINSERT

_EDSENDKEY

_EDDELETE

_EDGETCHAR

_EDGETSTR

_EDINDENT

_EDCOMMENT

_EDSELECT

_EDCOPY

_EDCUT

_EDPASTE

_EDUNDO

_EDREDO

_EDUNDOON

_EDACTIVE

_EDLASTERR

_EDSETENV

_EDGETENV

_EDOPENFIL

_EDCLOSEFI

_EDREVERT

_EDPROPERTIES

_EDPROCLIST

GETCLIPDAT

GETCLIPFMT

ISCLIPFMT

ENUMCLIPFM

REGCLIPFMT

SETCLIPDAT

EMPTYCLIP

COUNTCLIPF

OPENCLIP

CLOSECLIP

MAINHWND

_GETWRECT

_WHTOHWND

_WFINDTITL

_WMAINWIND

_FINDWINDO

_FINDWINDP

_WONTOP

_WOPEN

_WSCROLL

_WOPENP

_WSCROLLP

_WCLOSE

_WHIDE

_WSHOW

_WZOOM

_WSELECT

_WSENDBEHI

_WGETPORT

_WSETPORT

_WMOVE

_WSIZE

_WMOVEP

_WSIZEP

_WTOP

_WBOTTOM

_WLEFT

_WRIGHT

_WHEIGHT

_WWIDTH

_WTOPP

_WBOTTOMP

_WLEFTP

_WRIGHTP

_WHEIGHTP

_WWIDTHP

_WCLEAR

_WCLEARREC

_WCLEARREP

_WPOSCURSO

_WPOSCURP

_WGETCURSO

_WGETCURP

_WATTR

_WSETATTR

_WPUTCHR

_WPUTSTR

_WSETTITLE

_WTITLE

_WFOOTER

GETFILEVERSION

FOXTOOLVER

Flash Creative Management, Inc.

Home | What's New | About Flash | Services | Cybrary | Codebook | Careers

The Codebook Newsletter

August 1999
Volume 3, Issue 2

Creating Useful Class Diagrams with Rational Rose

Class diagrams are a very useful tool for communicating the desired structure of an object to another individual. This article explains some tricks to use to make them even more useful.

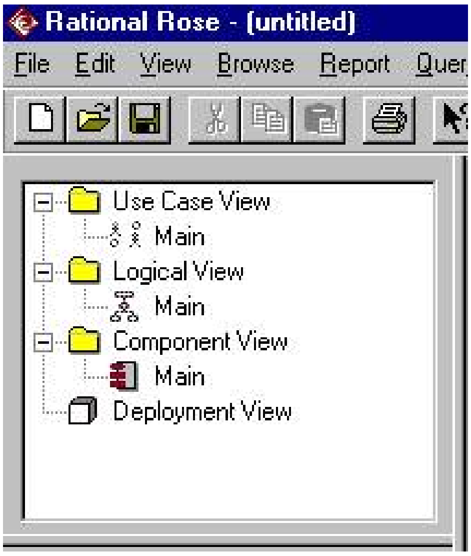
Charles T. Blankenship, MCP

Introduction

Using a framework to develop applications is very beneficial in many ways, but it presents some challenges as well. One of the largest challenges is categorizing the components of the framework for the purpose of presenting the framework to a new developer in a clear and logical manner. Some examples of the high level components in the Codebook framework include Collections, Controls, Forms, Business Objects, etc. Another problem when using a framework is communicating to a new user the physical manifestation of those high level components. For example, the physical files that make up the Control objects in Codebook are CONTRLS.VCX and ICONTRLS.VCX. The final challenge (that we'll address with this article) when working with a framework like Codebook is communicating to another developer the portions of the framework they should use when creating the objects that participate in the application being created. This article explains how to use Rational Rose in conjunction with a reverse engineering tool named VFPMODEL.APP (and its scaled down twin, Visual Modeler) to meet these challenges and conquer them.

Using Rational Rose to Present High Level Framework Components

When dealing with a new framework one of the first questions to ask is, "What are the components of this thing"? Rational Rose provides a means to answer this question through the use of a construct known as a package.



A Rational Rose Package, Described

A package is a collection of classes that are strongly related to one another, and each package represents a chunk of the logical architecture of the system.

Knowing this, it becomes obvious that an appropriate use of the package is the presentation of a high level framework components. Therefore, the first step in representing a framework to the user base is to create a package for each type of component in the framework. Some examples include Applications, Forms, Collections, Controls, Custom Controls, Menus, Template Menus, Events, Environments, Iterators, Logs, etc. Anything you can think of that represents a major component of your framework should be presented to the user of the model in a clear and concise way.

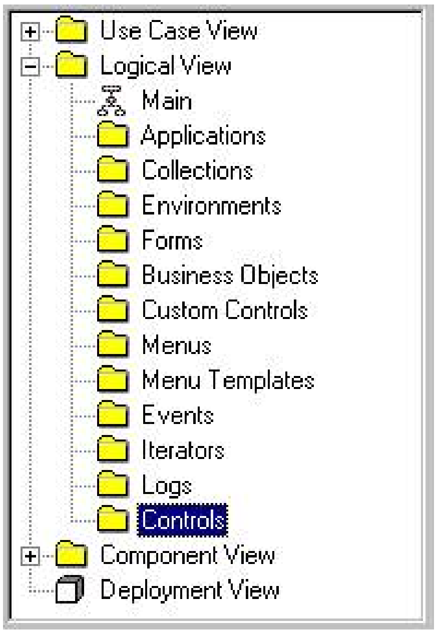
Please examine the diagram above. It illustrates how a brand new Rational Rose Model looks upon its creation. The place to present the high level components to the user of the framework in Rational Rose Model is in the Logical View package.

Creating High Level Component Packages

To create a high level component package in the Logical package of the new Rose Model follow these steps:

Ø Highlight the Logical package
Ø Right click on the Logical package
Ø Select New from the short cut menu, another short cut menu appears that gives you a list of new "things" you can create.
Ø Select Package from the short cut menu, a new package with the name "NewPackage" appears as a child of the Logical package
Ø Type in the name of the high level framework component, for example, "Controls".

The following figure illustrates a significant portion of Codebook's high level components.



The primary benefit of this approach is to communicate to a new user of the framework all of the high level components that comprise the Codebook framework. As soon as the developer expands the Logical View node of the Rational Rose model it becomes very clear that the Codebook framework consists of Applications, Collections, Environments, Forms, etc.

Illustrating the Makeup of High Level Components

Now that the user knows the framework contains Controls they should use to build applications, the next question is “Where are the Controls located”? Once again, specially named packages allow the framework provider to elegantly answer that question.

Identifying the Class Libraries that Make Up High Level Components

Just as the Controls package is a child of the Logical View it is possible to create another package that is a child of the Controls package.   These packages identify the class libraries that make up the high level framework components. You accomplish this in exactly the same way you created the Controls Package itself:

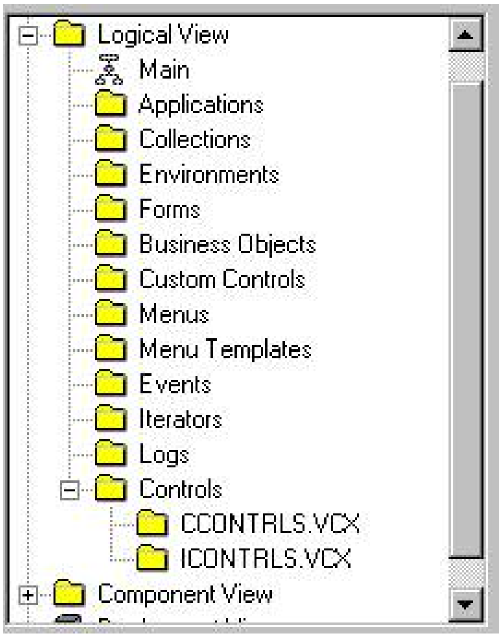
Ø   Highlight the Controls package

Ø   Right click on the Controls package

Ø   Choose New, another short cut menu appears

Ø   Select Package

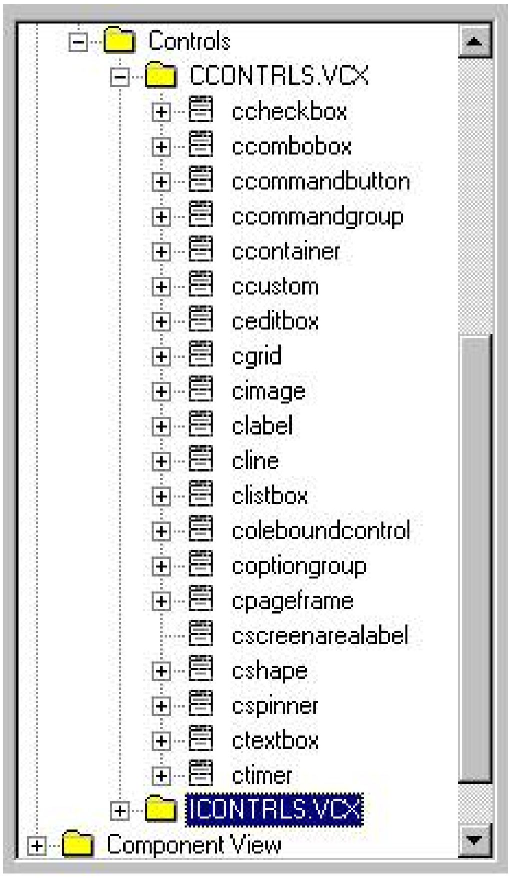
Ø   Name the package “CCONTRLS.VCX”. Do the same for another package, name it “ICONTROLS.VCX”. When complete, the Logical package for the model looks like this.



This is important information for a new application developer to the Codebook framework as it identifies the physical files within the framework that store the controls a developer should use to build applications. To reiterate, just by looking at the Logical view it is apparent that the Codebook framework contains controls and that those controls are contained in two physical files, CCONTRLS.VCX and ICONTRLS.VCX.

Reverse Engineering Visual FoxPro Class Libraries

The next step is to reverse engineer the CCONTRLS and ICONTRLS class libraries from their VFP components into the Rational Rose model. A nifty little application written by Markus Eggar, named VFPMODEL.APP accomplishes this task for us. The actual operation of VFPMODEL.APP is beyond the scope of this article; however, the code for it is downloadable from Microsoft’s Website. This code is also part of Codebook 6 distribution files. The instructions for operating this application extract with the application once it is downloaded and installed.



Specifically, VFPMODEL.APP allows you to examine the contents of a VFP class library and map the objects contained therein to a Rational Rose package. The diagram to the left illustrates the model after all of the classes contained in the CCONTRLS.VCX are imported into the CCONTRLS.VCX package and all of the controls in the ICONTRLS.VCX are imported into the ICONTRLS.VCX package (only the CCONTRLS.VCX) class library is expanded to show its contents.

Creating Class Diagrams

Now that all of the Codebook Controls are reverse engineered into the model it is possible to quickly make class diagrams from them. To create a class diagram of the Edit Box control perform the following steps.

Ø Select the Controls package

Ø Right click on it

Ø Select New from the shortcut menu

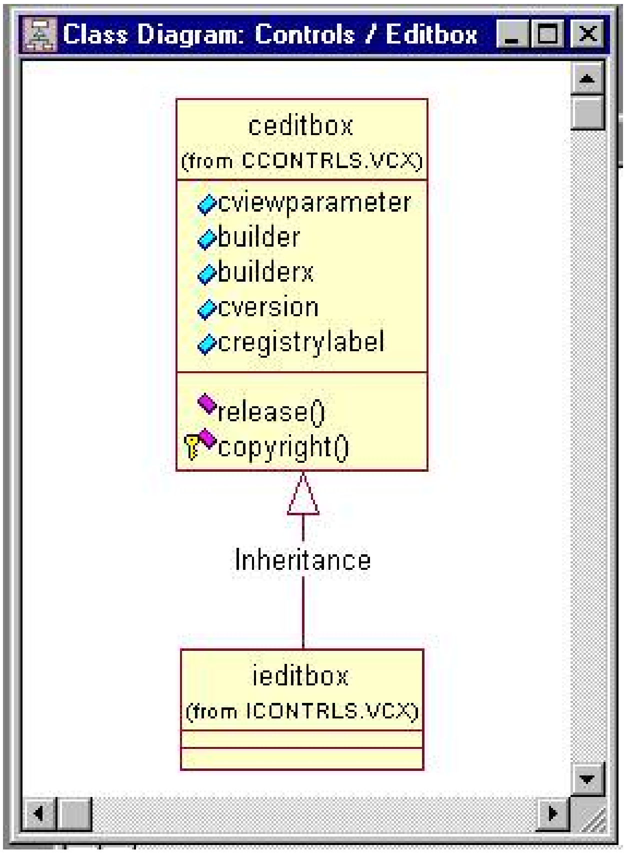
Ø Select Class Diagram from the next shortcut menu

Ø Name the class diagram “Editbox” and press Enter

Ø Double click on the new Editbox class diagram in the tree view and notice that the panel on the right displays a blank working surface for the Editbox class diagram.

Ø Drag the CEDITBOX class onto the class diagram

Ø Drag the IEDITBOX class diagram onto the class diagram and notice that a generalization line is automatically drawn from the IEDITBOX class to the CEDIT BOX class.

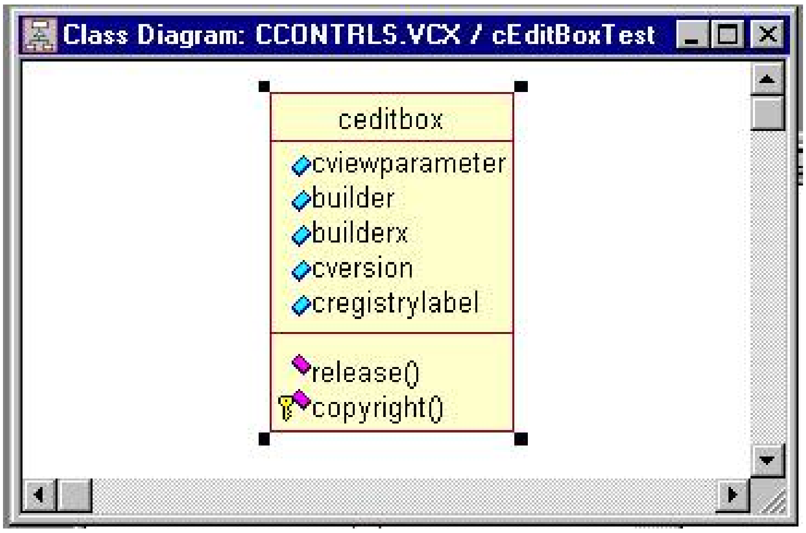


Your class diagram should look something like the diagram to the left. Please take the time to notice something special about this class diagram. Notice that Rational Rose identifies the package that contains the class CEDITBOX. Since we named the package CCONTRLS.VCX the class diagram not only illustrates the class hierarchy and composition of the Codebook Editbox control, it also identifies the physical location of the participating objects as well. In this case we know that CCONTROLS.VCX contains the CEDITBOX control and that ICONTRLS.VCX contains the IEDITBOX control.

These seemingly small facts are extremely important pieces of information to a developer new to the Codebook framework (or any framework for that matter).

The trick to getting Rational Rose to display this type of package information on the Class Diagram occurs due to the location of the class diagram itself within Rational

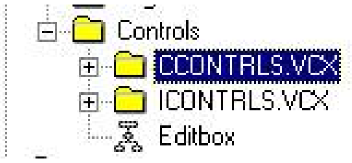
Rose. Notice that the class diagram is stored in the Controls package and not within either of the .VCX packages. When the components in the class diagram and the class diagram itself are stored in the same package, Rational Rose does not display the package information by default. The following class diagram illustrates the appearance of the CEDITBOX class definition when the class diagram is stored in the CCONTRLS.VCX.



Class Diagram: CCONTRLS.VCX / cEditBoxTest

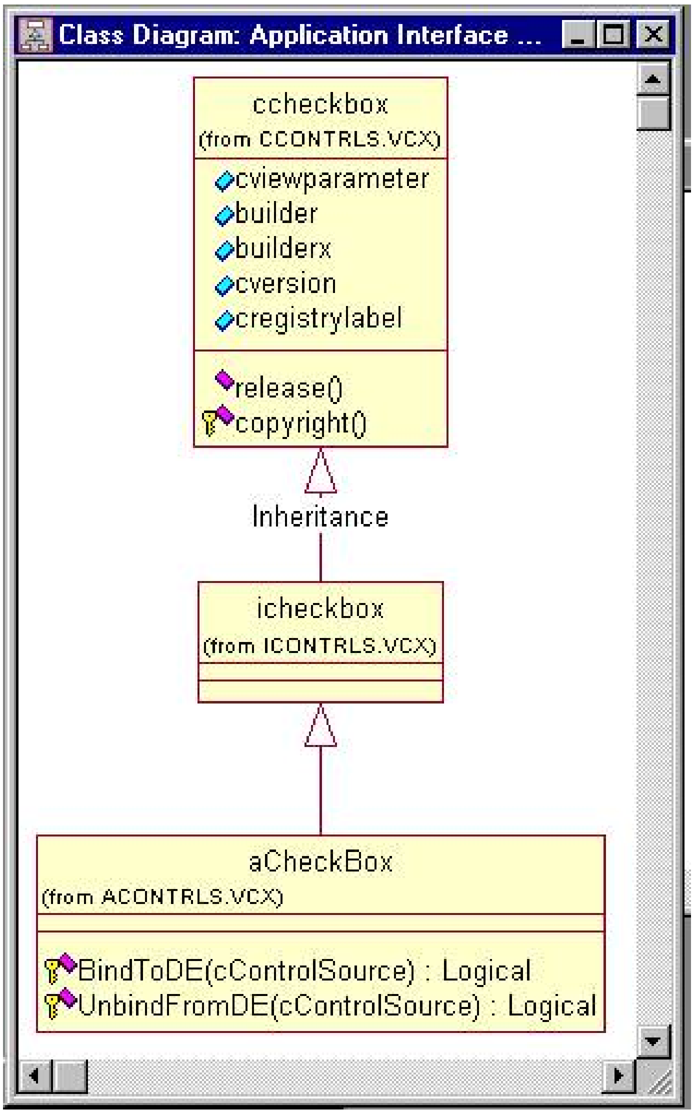
ceditbox
cviewparameter
builder
builderx
cversion
cregistrylabel
release()
copyright()

Therefore, if you are reading a Rational Rose class diagram and the class itself does not contain any package information, all you have to do to determine the package that contains the object is to look at the title of the class diagram. The first part of the Class Diagram title is the name of the package containing the diagram and the second part is the name of the class diagram itself. In this case the title for the class diagram is as follows: Class Diagram: CCONTRLS.VCX / cEditBoxTest.



Controls
CCONTRLS.VCX
ICONTRLS.VCX
Editbox

The tree view control in the Rational Rose model, now that the Editbox class diagram exists, looks like the figure to the left.



Using the Model to Communicate Design Decisions

Now that the framework information is captured in the model the task of communicating very specific design decisions to another developer becomes trivial. To illustrate this, consider the following example. Assume I want you to create a checkbox control from a framework component that must know how to bind itself to its data environment and unbind itself from its data environment. Consider the information that this requires. The programmer must first know from which control to inherit to create the new object. Next, they must know where to get the control. Then, they have to know what to call it. Finally, they must determine where to store it. If I don't have a very precise way of communicating this information to a developer the nature of the object I receive in return is anyone's guess. However, this model now gives me a very concise and precise method of communicating my design goals to a fellow programmer.

Simply by looking at the class diagram above it is possible to tell a fellow developer the following things:

Ø The new object's name must be ACHECKBOX
Ø It inherits from an object named ICHECKBOX which is located in ICONTRLS.VCX
Ø The new object must be located in the class library named ACONTRLS.VCX
Ø The name of the methods are UnbindFromDE() and BindToDE()
Ø Each method accepts one character type of parameter by the name cControlSource
Ø Each method must return a logical value

Conclusion

This article attempted to communicate the power available to developers who chose to use Rational Rose as a design tool. However, just like any powerful tool, tricks exist that make the tool even more useful and intuitive. I hope that this article showed you how to use Rose to present the components of a complex framework to a new developer in a logical and methodical way. A way that makes learning the framework easier as well as makes the task of communicating very specific design decisions to another, possibly remote developer, a very easy thing to do.

Case Studies

3M Health Information
AC Nielsen
BioMedicalInvest.com
Community Playthings
Corning Clinical
Reinauer
University of CA
Remarketing Services

Our collaborative efforts have resulted in breakthrough solutions that have directly impacted the "bottom lines" of our clients. These case studies will provide you with a brief overview of some of our accomplishments.

If your company has completed a project with Flash and would like to have a case study written, please contact Ray Stoddard.

1. 3M HIS
2. AC Nielsen
3. BioMedicalInvest.com
4. Community Playthings
5. Corning Clinical Labs(Quest Diagnostics)
6. Reinauer Transportation Companies
7. University of California's Education Abroad Program
8. Remarketing Services of America

Mission | Our Approach | Our People
Philosophy | Client List | Case Studies ◀
Testimonials

Please use our feedback form to submit questions and comments.

CONTENTS | FEEDBACK | CONTACT US | SEARCH

Feedback

Help us serve you better....

Strengthening our relationship with you and enhancing our services are our goals. We invite you to tell us what you think about our web site, company, products, and/or services. Please provide us with your contact information so that we will be able to continue to make each interaction with you more meaningful and valuable.

We welcome all of your comments and suggestions.

⬅ **Required Field**

Please provide the following contact information:

First name		⬅
Last name		⬅
Title		
Organization		
Street address		
Address (cont.)		
City		
State/Province	Select from list	⬅
Zip/Postal code		
Country	United States	
Work Phone		
FAX		
E-mail		⬅

URL □

Select any of the following options that apply:

- □ Career Opportunities
- □ Consulting Services
- □ Seminar Information
- □ Company Information
- □ General Requests
- □ Web Site

Please write any comments you have below:

□

[Submit Form] [Reset Form]

Please use our feedback form to submit questions and comments.

CONTENTS FEEDBACK CONTACT US SEARCH

Flash Creative Management, Inc.

Home | What's New | About Flash | Services | Cybrary | Codebook

Table of Contents

What's New

- Upcoming Events
- Downloadables
- Press Releases
- Top Stories

About Flash

- Case Studies
- Client List
- Mission
- Our Approach
- Our People
- Philosophy

Codebook

- Codebook Assistant
- Codebook World
- Newsletter
- Partnering Programs
- Testimonials

Cybrary

- Business Books
- Business Articles
- Seminars
- Technology Books
- Technology Articles
- Notes & Downloads

Management Services

- Strategic Planning
- Process Design
- Project Management
- Product Development
- Net Strategy
- Mentoring & Training

Technology Services

- Class Library Design
- Code Review
- COM
- RFP Design
- Web Design

Please use our feedback form to submit questions and comments.

CONTENTS | FEEDBACK | CONTACT US | SEARCH

Contact Us

Directions

Contact Information

Flash Creative Management, Inc.

433 Hackensack Avenue, 12th Floor

Hackensack, NJ 07601-6336

(Voice) 201-489-2500

(Fax) 201-489-6750

To contact us via e-mail, click here.

For your convenience we have provided you with driving directions from major nearby highways and public transportation information.

If you would like personalized driving directions, feel free to use our interactive map service powered by:

Maps by Expedia.com
www.expediamaps.com

Please use our feedback form to submit questions and comments.

CONTENTS FEEDBACK CONTACT US SEARCH

Flash Creative Management, Inc.

Search

Can FlashFinder Help You Find Something?

FlashFinder is a tool that allows you to search our web site for specific topics, phrases, concept, words, titles, authors, or dates. You may also use FlashFinder to peruse the Flash Archives for past issues of The Codebook Newsletter (CNL). After you find the article you want, you can print it out or save it on your hard drive or to a floppy disk.

To begin this service, type what you're looking for in the box provided and then click "Submit Search".

*****Please note that your results will appear on the bottom of this page.**

Search for: □

Submit Search | Clear

Please use our feedback form to submit questions and comments.