

Visual FoxPro Best Practices for the Next Ten Years

Conference Proceedings Outline

1. Best Practices for Development Environment Setup

Windows

Barbara Peisch
Nancy Folsom
Tamar Granor
Whil Hentzen

VFP Startup Routines

Barbara Peisch
Cathy Pountney
Doug Hennig
Marcia Akins
Nancy Folsom
Rick Borup
Rick Schummer
Tamar Granor
Whil Hentzen

Frequently Used Tools

Barbara Peisch
Cathy Pountney
Doug Hennig
Nancy Folsom
Rick Borup
Rick Schummer
Tamar Granor

Best Practices Summary

2. Best Practices for Error Handling

Introduction

Errors happen

Programmer errors
Predictable errors
Unpredictable errors

Real world stories

User response

Application response

- Address without notification
- Address with notification
- Try again
- Avoid
- Abandon

Error handling fundamentals

- Determining the cause of the error
- Notification
- Be assertive
- Error Log
- Clean shut down

Types of error handling in VFP

- Default error handling
- Global
- Object-level
- Structured

Error handling priorities

- Using one style
- Mixing two styles
- Mixing all three styles

Error handling strategy

Logging errors

- Who, when, where, why, and how
- Use VFP's LIST command
- AERROR function array contents
- Environment settings
- Program call stack
- Form data sessions
- config.fpw contents

Communicating errors and bugs

- Phone call
- Screen shots
- Recording a screencase

Send the error log files
Send a report
RSS feed

Tracking errors

Commercial framework approaches

Visual FoxExpress
The Global Error Handler
Use of Try/Catch
Handling of Expected Errors
Miscellaneous
Visual Extend (VFX)
Global Error Handler
TRY/CATCH
Error methods
Visual MaxFrame Professional (VMP)
Codemine
Object error event
ON ERROR handler
TRY...CATCH
The nErrorMode Property
Errors in Server Applications
Error Event Log and Text Log
Mere Mortals for VFP
Promatrix

Resources

Special thanks

Conclusion/Summary

About the Author

3. Best Practices for Class Design

Introduction

Where should the code go?

Stand-alone program file or instance
UDF/Procedure File
Visual Classes

Non-visual Classes

When should you define a class?

Will it be Re-Usable?

Will it help to Manage Complexity?

Is it Worth the Effort?

What sort of class do we want?

The component (abstract) model

The real world (concrete) model

What do you do when defining a class?

Must:

Could:

Should:

Managing the Hierarchies

The Class (or "Inheritance") hierarchy

How VFP implements inheritance

The "Object" (or "Containership") hierarchy

Composition and Aggregation

How do you build your Classes?

Class Library Structure

Look for Patterns

Code in Methods, not Events

Beware of putting functionality too high in the hierarchy

Methods should do one thing and one thing only

Use template methods for a consistent interface

Conclusion

About the Author

4. Best Practices for User Interface Design

Why do user interfaces matter?

Designing Interaction

Designing Good Interfaces

Basic principles

- Consistency
- Visibility
- Feedback
- Simplicity
- Error-tolerance
- Accessibility
- Standards and guidelines

Putting the principles to work

- Application-wide issues
- Use task terminology
- Use language well
- Use color wisely
- Use scalable fonts
- Make your application easy to use with both the keyboard and the mouse
- Remember so the user doesn't have to
- Provide Undo

Application Control and menus

- Organize the menu bar sensibly
- Provide hot keys and menu shortcuts
- Clue users in about menu items' behavior
- Manage users' access to menu items
- Use toolbars as mouse shortcuts
- Large and small toolbars offer users more control
- Shortcut menus provide another alternative

Forms

- Forms have title bars
- Distinguish between dialogs and documents
- Data entry forms are not modal
- Keep dialogs to a minimum during processing
- Dialogs are modal
- Facilitate the most common action
- Saving data

Finding data

Controls

- Use the right control for the job
- Use graphical buttons in toolbars
- Keep tooltips to tools
- Give the user access only to options that are available
- Make capitalization uniform
- Supply reasonable defaults
- Group related controls visually
- Set navigation order correctly

Better applications

Resources

5. Best Practices for Local Data Access

Some basics

- Should we use DBF files for primary data storage?
- Security
- Integrity
- What's good then?

Naming conventions

The Visual FoxPro DBC

Keys

- Primary Keys
- Candidate Keys
- Foreign Keys

Indexes

- Structural and Non-Structural Indexes
- Should you Index on DELETED()

Updating Records

Data Buffering

- What do we mean by 'buffering' anyway?
- Buffering strategies

Locking strategies
Buffering modes
What does all this mean when creating data-bound forms?
So how should we set up buffering in a form?
Using BufferModeOverride
Using CursorSetProp()
So what mode of buffering should I use in my forms?

Transactions

Restrictions and limitations
When to use a transaction?
What effect does rollback have on data?
Does that mean we can use transactions to enable SQL to see 'pending' changes?
How does a transaction impact on FoxPro's locking mechanisms?
Can multiple transactions exist simultaneously?

Optimization and Performance

Working with NULLS

Gotcha! Null Strings can fool you

Extending SQL in VFP

Sub queries
Derived Table
Computed Columns
Approaching complex queries
Example: What haven't they ordered recently?

The Last Word

About the Author

6. Best Practices for Remote Data Access

What do we mean by 'remote' data?

Remote Server Advantages/Disadvantages

Cost of Implementing a Dedicated Server

Getting connected

Connecting to a database server with ODBC
Connection Strings

- Using SQLStringConnect()
- Named Connections
- Using SQLConnect()
- How do I connect to a database using OLEDB
- Connecting to a database that is not installed locally
- Which is better, ODBC or OLEDB
- Which is better, DSNs or Connection Strings

Tools for accessing remote data

- FoxPro's SPT functions
- Connection management
- Connection properties
- Command execution
- Miscellaneous

Interacting with Remote Data

- Embedded quotations
- Empty and Null Values
- Solution: Format VFP Values using a function
- It's not just format
- Handling NULLs
- Fuzzy Queries
- Embedded Functions
- User-defined Functions

Transactions management

Server side code

Stored procedures

The Last Word

About the Author

7. Best Practices for Refactoring

Tune-up Your Code

- When a pig needs more than lipstick
- Example bad smell: Comments
- Example best practice: Good comments explain why, not what
- Example refactoring: Rename Method and Introduce Assertion

Best Practices while Refactoring

- Test after every change
- Make small changes
- Don't change unrelated code
- Know When To Stop
 - Stop if a bug appears
 - Stop if the system changed
 - Stop when you're uncertain
 - Stop when you're finished and the final test runs correctly
 - Stop if you're tired

Identifying Bad Smells

- Commented Code
 - Bad Comments Explain What, not How or Why
 - Comment States an Assumption
- Parameters
 - Many Parameters
 - Procedure Changes Value of a Parameter
- Conditionals
 - Conditional Structures and Behavior
 - Conditional Parameters
 - Business Logic in UI Controls
 - Overburdened Class
 - Data Clumps
 - Duplicate Code
 - Long or Procedural Code
- Change an Odor to an Aroma
 - General Purpose Refactoring
 - Replace Magic Number with Symbolic Content
 - Extract Method
 - Rename Method
 - Hide Delegate
 - Introduce Assertion
 - Refactor Conditional Structures
 - Decompose Conditional
 - Replace Nested Conditional with Guard Clauses
 - Replace Temp with Query
 - Remove Assignments to Parameters
- Refactor for N-tier Architecture
 - Self Encapsulate Field
 - Separate Domain from Presentation
 - Duplicate Observed Data
 - Convert Procedural Design to Objects

Conclusion/Summary

Bibliography

Biography

8. Best Practices for Reporting and Output

Overview

What does "Best Practices" mean?

Sample Files

- PRG files
- Form file
- VCX-based classes
- Reports

VFP Settings

- Setting REPORTBEHAVIOR
- Report's Private Datasession

Overview of an output system

- Use standard naming conventions and techniques
- Reduce the number of places that require changes
- Use an N-Tier Solution
 - The front end
 - Data retrieval
 - The output class

The Details

- The Data Class
- The Front End
 - Checkbox class
 - Container class
 - Naming convention
 - How things interact
- The Output Class
 - cReportMode
 - cProgressMess
 - ICancelProcessing
 - IDontConfirmOutput

- cDefaultOutputFile
- cOutput
- IDebug
- Process flow

Debugging

- Bugs that trigger an error
 - At development time
 - At runtime
- Bugs that give incorrect results
 - That doesn't add up!
 - Only your user knows for sure

Putting it all together

- Add a new report using existing output methods and PrtOut form
- Using your own form or program
- Adding a new output tool or format

And finally...

9. Best Practices for Project Management

Overview

In the Beginning - Creating the Vision and Scope Document

- Business Requirements
- Vision of the Solution
- Scope and Limitations
- Business Context

The Groundwork - Making some early decisions

- Choose a Methodology
 - Waterfall Method
 - Agile Development
 - Iterative and Incremental Development
- Choose the Tools
 - Project Management
 - Documents
 - Database
 - Langauge
 - Framework
 - 3rd Party Tools
 - Source Control

Bug Tracking/Enhancement Requests
Help

The Team - Assembling and managing the team

- Assembling the Team
 - In-house vs. Outsourcing
 - In-house
 - Consultants
 - Offshore Outsourcing
 - Hiring New Staff
 - Undesirable Characters
 - Desirable Characteristics
 - Provide Training
- Managing the Team
 - Set a Good Example
 - Be a Good Listener
 - Be Accountable
 - Be Respectful
 - Trust the Team
 - Get to Know the Individuals
 - Adjust Your Management Style
 - Understand What Drives Each Individual
 - Understand the Strengths and Weaknesses of Each Individual
 - Keep the Morale High
 - Morale Budget
 - Comp Time
 - Recognize a Good Job
 - Know What's Going On
 - Status Reports
 - Regular Meetings
 - Identify FYI Items vs. Action Items

The Details - Defining and clarifying the goals

- Cosmic Truths
- The Two Facets of Requirements
 - Requirements Development
 - Requirements Management
- The Documents
 - Use Case Document
 - Software Requirements Specification
- Preparing the Documents
 - How much details
 - Less
 - More
 - How to Gather Information
 - Chose Your Words Wisely

A Picture is Worth 1000 Words

The Schedule - Determining when it will be done

- Estimating
- Scheduling
- Managing Slippage

The Construction - Building the application

- Naming Conventions and Coding Standards
 - Naming Conventions
 - Forms, classes, methods, programs, and functions
 - Variables
 - Table names, cursors, and indexes
 - Field names
 - Coding Standards
 - Parameter Validation (valp)
 - Asserts (debugmsg)
 - Header Comment (hdr)
 - Change log (log)
 - Change Comment (chg)
 - Change Block (chgb)
 - To Do List (todo)
 - Code Reviews
 - Documentation

The Test - Making sure we did it right

- Create a Test Plan
- Types of Testing
 - Unit testing
 - System testing
- The Implementation
- The Documentation
- Build a Test Matrix
- Bug Bash
- Bug Fixes

The Implementation - Going live

- Installation
- Data Conversion
- Training
- Change Requests

The End - Wrapping up loose ends

In Conclusion - What have we learned?

10. Best Practices for Debugging

(tips from panel)

11. Best Practices for Vertical Application Development

Introduction

Application Licensing and Activation

- Application Licensing
- Application Activation
- Other Licensing Ideas

Error Reporting

- Diagnostic Logging
- Other Error Handling ideas

Application Maintenance Models

Subscription management

- Customer Purchase
- Software Activation
- Subscription Renewal
- Other Subscription Issues

Version Update Mechanisms

Support Policies

Summary

Biography

12. Best Practices for Middle Tier Development

Introduction

Defining the Middle Tier

General design considerations

Where does the middle tier live?

Accessing distributed components

Designing for COM+

Passing Data

Writing the Code

Installing and configuring in COM+

Configuring COM+ security

Conclusion

13. Best Practices for Deployment

Use modular design to facilitate different deployment types

Make the data independent of the app

Choose a setup authoring tool and learn it well

Use incremental refinement to fine tune your process

Document your process

Always use a checklist

Use version numbering for the primary EXE

uUse version control for MSI installs

Review what you must include

Review what you cannot include

Recompile all files

Remove printer environment information from report files

Copy all updated files to your gold directory

Build your setup from your gold directory

Manual builds

Semi-automatic builds

Fully automatic builds

Consider installing the VFP runtime files separately

Consider excluding reports from the EXE

Consider installing the VFP OLE DB provider and the ODBC driver

Test on a clean machine

Test as a regular user, not an administrator

Document what's new by date and version number

Maintain the historical record

Conclusion/Summary

* eof