

# .NET Eye For the CF Guy

Developing ASP.NET Applications Using C#

David Churvis  
Productivity Enhancement

Copyright © 2004 by Productivity Enhancement, Inc.

1

## ColdFusion Will Always Be #1

Make no mistake: we're not suggesting you abandon ColdFusion

Nor are we suggesting you switch to BlueDragon for .NET

We *are* suggesting that you learn what is possible with similar technologies, and that you bolster your employment opportunities by learning a highly sought-after skill...

*C# for ASP.NET*

Copyright © 2004 by Productivity Enhancement, Inc.

2

## In a Nutshell...

A tour of two identical applications

Differences in development

Differences in language

Things you will and won't miss about ColdFusion

Things you will and won't like about .NET

Integrating ColdFusion and .NET

Copyright © 2004 by Productivity Enhancement, Inc.

3

## A Tour of Two Identical Web Apps

Quick specifications on the simple application we want to build

Copyright © 2004 by Productivity Enhancement, Inc.

4

## Search

Search by first few characters in most cases

Use select menus to make exact match

User must enter a value in at least one field

When user searches, his criteria must be saved upon repeated trips to the list

Copyright © 2004 by Productivity Enhancement, Inc.

5

## List

Clicking on company name or employee SSN edits the record

Each record may be added or deleted

Two different styles of action links shown for Add and Delete links

Copyright © 2004 by Productivity Enhancement, Inc.

6

## Add

Add and edit is shared between one page

If the primary key is passed, the page is in edit mode; otherwise, the page is in add mode

Uses both client- and server-side validation

## Edit

Fields are populated from a database query

Employee edit uses same filtered select object as employee search

## Delete

Provides a detailed confirmation page before deleting the record

# Differences in Development

## Form & Action Pages versus a Single Web Form Object

ColdFusion uses pages

.NET uses objects

ColdFusion is request-driven, procedural code

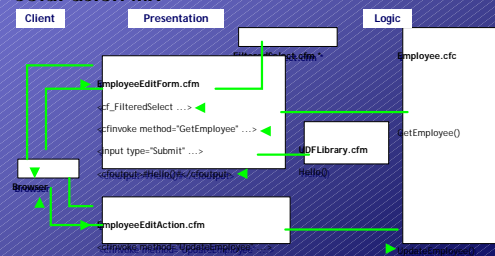
- Very straightforward
- Pages are directly requested and processed procedurally

.NET is event-driven, object-oriented code

- The browser requests a form
- The form's logic layer (codebehind) formats the form appropriately by setting properties on server controls
- Whenever the user does something on the form to trigger a server event, the form is posted, and ASP.NET fires the appropriate event

## The Request Cycle

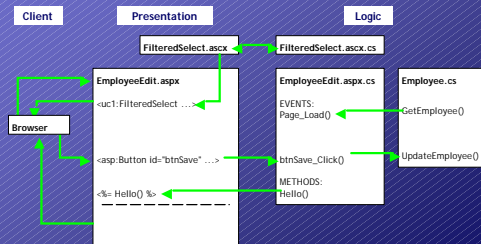
ColdFusion MX



\* A custom tag would typically call component methods and/or UDFs to separate presentation from logic. This example is not tied to a specific data object or method, so presentation and logic are combined in FilteredSelect.

## The Request Cycle

### ASP.NET



Copyright © 2004 by Productivity Enhancement, Inc.

13

## Integrated Display and Logic Versus Separated Display and Logic

### ColdFusion is output-oriented

- Effort is concentrated on generating the HTML to push to the browser
- Generated markup is mixed with the logic that defines the markup
- True separation of logic and display is rather messy

### .NET is logic-oriented

- Effort is concentrated on building the business logic separately from its presentation and having a minimal interaction between display and logic
- Server controls are put on a form and then bound to data in the form's codebehind module

Copyright © 2004 by Productivity Enhancement, Inc.

14

## Framework Building Blocks versus a True Pre-Built Framework

### ColdFusion offers:

- a few basic types
- a basic "framework"
- a tag-based scripting language
- Easy learning curve

### .NET offers:

- a vast library of objects
- a true web application framework
- your choice of object-oriented languages
- .NET has a steep learning curve, but is more powerful in some cases

Copyright © 2004 by Productivity Enhancement, Inc.

15

## Interpreted Script versus Compiled Code and Objects

### ColdFusion pages are dynamically compiled

### ASPX pages are also dynamically compiled

### Codebehind modules are pre-compiled

- All non-display code is put in codebehind files
- Compiled separately from the ASPX pages
- Many errors that would normally appear during runtime now appear during the compilation cycle
- You don't have to deploy most of your source code!

Copyright © 2004 by Productivity Enhancement, Inc.

16

## Differences In Language

## Untyped versus Typed

### ColdFusion's variables are untyped

- Variable type is discovered and cast during execution
- Very flexible, but...
- Can lead to difficulty maintaining code

### .NET variables are strongly typed

- Variable type is declared at compile time
- Incorrect variable usage is usually discovered before problems can occur
- Very difficult to get casting errors

Copyright © 2004 by Productivity Enhancement, Inc.

17

Copyright © 2004 by Productivity Enhancement, Inc.

18

## Procedural versus Object Oriented

### ColdFusion is procedural

- Even the advent of components does not change the basic nature of CF

### .NET is object-oriented

- Page objects respond to events raised by the framework and by user interaction
- Rather than generating HTML, you usually interact directly with objects on the server
- Manual markup generation is still available if necessary

Copyright © 2004 by Productivity Enhancement, Inc.

19

## Maintaining State: Session Scope versus Session Object

Very similar between ColdFusion and .NET

.NET has a Session.Abandon() method

.NET allows sessions to be replicated across servers using the .NET state service

Complex objects are automatically replicated during a failover event using .NET web farms

Copyright © 2004 by Productivity Enhancement, Inc.

20

## Custom Tags versus Custom Server Controls

### ColdFusion has custom tags

- Simple calling syntax
- Attributes become variables in the Attributes scope

### .NET has server controls

- User controls offer ease of development
- Custom controls offer power and flexibility
- Server controls are called with tags on the ASPX page
- Attributes in the tag call map to defined properties in the control

Copyright © 2004 by Productivity Enhancement, Inc.

21

## Security In .NET

Not enough time - covered in the one-day seminar and in training course

Copyright © 2004 by Productivity Enhancement, Inc.

22

## Structured Exception Handling

Not enough time - covered in the one-day seminar and in training course

Copyright © 2004 by Productivity Enhancement, Inc.

23

## Integrating ColdFusion and .NET

Copyright © 2004 by Productivity Enhancement, Inc.

24



## Simple Integration

### Posting from a CF page to an ASP.NET page

#### Web services

- You can define a web service in .NET and call it from ColdFusion, or vice-versa

## BlueDragon for .NET

### Run CFML on top of ASP.NET

- Familiar tag-based syntax, with the power of .NET running underneath
- Access .NET objects from ColdFusion
- Share variables between your CFML and your ASP.NET application
- Your CFML templates are part of the ASP.NET application
- You can even use .NET security from within your CFML code

## Deploying a .NET Application

## Deploying a .NET Application

### You don't have to deploy all your source code

- Because codebehind files are compiled, you do not have to deploy their source files.
- You do not have to deploy any .aspx.cs or .cs files, which contain most of your source code
- You must deploy .aspx, .ascx, .asax, and web.config files.

## Things You'll Miss About ColdFusion

## Pound Signs (the good)

### ColdFusion:

```
<cfoutput>
  
</cfoutput>
```

### .NET:

```
/logo.gif">
```

## CFOQUERY

```
<cfquery name="insertCompanyQuery"
datasource="#Application.dbDSN#">
INSERT INTO Company(
    CompanyName,
    Address,
    City,
    State,
    ZipCode,
    Phone
)
VALUES (
    '#Arguments.CompanyName#',
    '#Arguments.Address#',
    '#Arguments.City#',
    '#Arguments.State#',
    '#Arguments.ZipCode#',
    '#Arguments.Phone#'
)
</cfquery>
```

Yeah, yeah, I know...  
CFOQUERYADM and  
Trim(), but hey-- it's just an  
example, so give me  
a break! ☺

Copyright © 2004 by Productivity Enhancement, Inc.

31

## CFOQUERY

### .NET's version:

```
SqlConnection connection = new
SqlConnection(ConfigurationSettings.AppSettings["connectionString"]);
connection.Open();
SqlCommand command = new SqlCommand(
    @"
    INSERT INTO Company(
        CompanyName,
        Address,
        City,
        State,
        ZipCode,
        Phone
    )
    VALUES (
        '' + _companyName.Replace("'", '') + @'',
        '' + _address.Replace("'", '') + @'',
        '' + _city.Replace("'", '') + @'',
        '' + _state.Replace("'", '') + @'',
        '' + _zipCode.Replace("'", '') + @'',
        '' + _phone.Replace("'", '') + @''
    )
    ",
    connection
);
command.ExecuteNonQuery();
```

Copyright © 2004 by Productivity Enhancement, Inc.

32

## Rapid Application Development for Simple Applications

### Very easy to build simple output in CF

- It's also very easy to build a simple application
- You make a form, you post it to an action page, you redirect somewhere

### In .NET...

- you have to build an ASPX page to define a DataGrid, then bind it to a data source.
- For forms in .NET, you define your forms in a completely different syntax than your logic, and in a completely different place

Copyright © 2004 by Productivity Enhancement, Inc.

33

## Things You Won't Miss About ColdFusion

Copyright © 2004 by Productivity Enhancement, Inc.

34

## Pound Signs (the bad and the ugly)

### When do I use pound signs? Do I:

```
<cfoutput>
<cfif myVar LT myOtherVar>
    #myVar# is less than #myOtherVar#
</cfif>
</cfoutput>
```

### Or Do I:

```
<cfoutput>
<cfif #myVar# LT #myOtherVar#>
    #myVar# is less than #myOtherVar#
</cfif>
</cfoutput>
```

Copyright © 2004 by Productivity Enhancement, Inc.

35

## Pound Signs (the bad and the ugly)

### Or Do I Even:

```
<cfoutput>
<cfif #Evaluate("#myVar# LT #myOtherVar#")#>
    #Evaluate("myVar")# is less than #Evaluate("myOtherVar")#
</cfif>
</cfoutput>
```

### But in .NET I:

```
if(myVar < myOtherVar)
{
    Response.Write(myVar.ToString() + " is less than " +
myOtherVar.ToString());
}
```

No more syntax errors from misplaced pound signs

Copyright © 2004 by Productivity Enhancement, Inc.

36

## Type Checking Errors

No more "The value "" cannot be converted to a boolean" errors!

- Because all variables are strongly typed, you can be guaranteed that you are not making simple mistakes like using a string as a boolean - the compiler won't let you
- Type checking errors usually only happen when working with object inheritance and more advanced usage

Copyright © 2004 by Productivity Enhancement, Inc.

37

## Undefined Variable Errors

No more "Variable "myVar" is undefined" errors!

- Because all objects have defined properties, you cannot use a variable unless it exists
- Any local variables you use are guaranteed to exist - it is a compile-time error for them not to

Copyright © 2004 by Productivity Enhancement, Inc.

38

## Structures

Structures are unstructured! ☺

- Accessing a non-existent key throws a runtime error
- You must be very careful to write your ColdFusion code to take this into account

**.NET objects have defined structure**

- There is no way to access a nonexistent property, because your code won't even compile
- All properties of an object are defined at compile time
- You still have access to structure-like collections in .NET through Hashtable and its related classes

Copyright © 2004 by Productivity Enhancement, Inc.

39

## Arrays That Start at 1

ColdFusion's array indexes begin at 1

- This is handy for beginning programmers or those with a background in Visual Basic
- This is not handy for programmers who use arrays in JavaScript - or nearly any other language for that matter - and must get used to ColdFusion's array indexes

**C# array indexes begin at 0**

- Just like every other computer language on Earth
- No "array disco" programming

Copyright © 2004 by Productivity Enhancement, Inc.

40

## Things You'll Like About .NET

Copyright © 2004 by Productivity Enhancement, Inc.

41

## Visual Studio .NET

Takes some getting used to, but well worth it

**Unparalleled code editing - truly the best of all worlds**

- Customization and integrated scripting from Visual Interdev
- Intellisense and class browser from Visual C++ 6
- Visual editing from Visual Basic 6
- Even has a few features from emacs thrown in (such as incremental searching)

**Combines many different development styles into a wonderfully complete IDE**

Copyright © 2004 by Productivity Enhancement, Inc.

42

## The .NET Framework

### A huge library of supporting classes

#### Collection classes

- Many kinds of collections for many different uses

#### Data Access and XML classes

- Provides fine-grained control over database calls
- Excellent XML API

#### Web Forms classes

- Provides the server controls and databinding for ASP.NET applications

### And so much more!

Copyright © 2004 by Productivity Enhancement, Inc.

43

## Large Application Development is Simpler

Business Logic can be separated into an object library

Multiple views of the same object can easily be built without duplicating code

Data schemas can be bound directly to code

Copyright © 2004 by Productivity Enhancement, Inc.

44

## Stateful Server Controls

### Consider three lists with paging controls

- In ColdFusion, each control must take every other possible control into account when reposting a page
- In .NET, each control is completely independent
- .NET is much more flexible and foolproof

Copyright © 2004 by Productivity Enhancement, Inc.

45

## Better Debugging

Integrated debugging with ColdFusion is frustrating

In Visual Studio.NET, debugging is an integral part of the system

- Breakpoints can be defined for an individual line of code
- Breakpoints can be defined for a specific function
- Breakpoints can even be delayed until a specific number of calls have been made

#### Detailed object breakdown

- While stopped, you can drill down into all the properties and methods of all available objects

Copyright © 2004 by Productivity Enhancement, Inc.

46

## Tight Integration with Windows

OS

COM

File system

and more

Copyright © 2004 by Productivity Enhancement, Inc.

47

## Form Validation

### Validator objects

- Automatic client- and server-side validation
- One object handles both client-side and server-side validation
- Validation is very flexible

Copyright © 2004 by Productivity Enhancement, Inc.

48



## Automatic Code Documentation in XML

Every class, method, and property in your code can have an XML comment

- These comments are in XML format and describe every aspect of your code
- These comments can be used to generate a consolidated XML file, which can then be transformed into an HTML document

## Automatic TODO Lists for Uncompleted Code

Know what's left to do

- Placing a comment that begins with TODO: anywhere in your code adds an item to an automatically maintained to do list
- Define your own custom markers with varying levels of importance

## Thorough MSDN Documentation

Huge library at

<http://msdn.microsoft.com/library>

- The class reference alone makes a stack three feet high when printed out

Every available class is documented

- If you can see it in Visual Studio .NET, you can find information about it on MSDN

## Things You Won't Like About .NET

## The Learning Curve Is a Bit Steeper

It's a different development approach

New tools

You must think differently about application design

It can be confusing to know where to begin

## The Learning Curve Is a Bit Wider

Object-oriented concepts

The ASP.NET application framework

The .NET Framework Classes

## Lack of a Tight-Knit Community

The .NET community is much larger and more impersonal

- There is not a .NET equivalent of the CF-COMMUNITY list
- There are many different developer sites and lists with often conflicting information

## Where to Go Next

**Introduction to C# & ASP.NET**

- One-day seminar
- Monday, September 27, 2004 (future dates to come)

**C# & ASP.NET for ColdFusion Developers**

- Intensive five day hands-on training course
- November 8-12, 2004 (future dates to come)

**Both events in Atlanta**

Go to [www.ColdFusionTraining.com](http://www.ColdFusionTraining.com) for more details

## More Resources

VS.NET, <http://msdn.microsoft.com/vstudio>

MSDN, <http://msdn.microsoft.com>

CodeProject, <http://www.codeproject.com>

DotNet247, <http://www.dotnet247.com>

BlueDragon, <http://www.newatlanta.com>

# THANK YOU!



from the guys at  
**Productivity Enhancement**