

THE EXPERT'S VOICE® IN .NET

Pro ASP.NET 4 CMS

Advanced Techniques for C# Developers
Using the .NET 4 Framework

*Learn the latest features of .NET 4 to build
powerful ASP.NET 4 web applications*

Alan Harris

Apress®

Contents

■ Contents at a Glance	iv
■ Contents	v
■ About the Author	xii
■ About the Technical Reviewer.....	xiii
■ Acknowledgments.....	xiv
■ Introduction.....	xv
■ Chapter 1: Visual Studio 2010 and .NET 4.....	1
Who This Book Is For	1
Who This Book Is Not For (or “Buy Me Now, Read Me Later”).....	2
What’s New in .NET 4.....	2
C# Optional and Named Parameters	3
C#’s dynamic Keyword.....	5
Dynamic and Functional Language Support	10
Parallel Processing	10
Managed Extensibility Framework (MEF).....	13
Distributed Caching with Velocity	13
ASP.NET MVC	16
A Tour of Visual Studio 2010.....	18
Windows Presentation Foundation	18
Historical Debugging	19
Improved JavaScript IntelliSense.....	21
jQuery Support	22

Building a CMS.....	24
CMS Functional Requirements.....	24
Creating the Application Framework	25
Summary.....	28
■ Chapter 2: CMS Architecture and Development.....	29
Motivations for Building a CMS.....	29
Motivations for Using .NET.....	30
Application Architecture.....	30
The CMS Application Tiers	32
CommonLibrary: The Data Transfer Objects	33
GlobalModule: The HttpModule	35
Components of a CMS Page.....	37
Buckets	37
Embeddable Objects	39
Embeddable Permissions.....	41
Handling CMS Content	43
The Content Table	43
The ContentVersion Table	44
Assembling Content on Demand.....	44
How Embeddable Objects Handle Versions	46
Summary.....	46
■ Chapter 3: Parallelization.....	47
What Is Parallelization?.....	47
Good Parallelization Candidates.....	47
Differences from Multithreading.....	48
Parallel Pitfalls	48
Deadlocks	48
Race Conditions	51
Thread Starvation.....	54

Amdahl's Law	55
.NET 4 Parallelization Concepts	56
Task vs. Data Parallelism.....	56
Task Parallel Library	56
Task.Wait()	57
Parallel.For() and Parallel.ForEach().....	59
Parallel LINQ (aka PLINQ)	59
.AsParallel()	60
CMS Parallelization Opportunities.....	61
Creating a Data Mining Embeddable.....	62
Expanding the Data Mining Tasks.....	66
Tagging	70
Tagging on the Client	73
Fleshing Out the Tagging Embeddable	75
What's in a Name?	76
Handling Tag Input.....	79
Tag Processing in the Business Tier	82
POST Problems	87
Finalizing Tag Storage	89
Inserting Tags	92
Content Tags	96
Summary.....	102
■ Chapter 4: Managed Extensibility Framework and the Dynamic Language Runtime.....	103
Managed Extensibility Framework.....	103
The Manual Way.....	103
The MEF Way	105
Working from Usage to Implementation	106
Exposing Libraries via MEF	106
A Simple Plug-in Contract.....	107

Implementing the Plug-In.....	107
Using the Plug-In.....	108
Catalogs and Containers	112
Supporting Multiple Parts	113
Dynamic Language Runtime	117
The dynamic Keyword.....	118
Benefits of the dynamic Keyword	119
CMS Plug-Ins	120
IEmbeddable	121
Server Controls as Embeddables	122
Displaying Embeddables.....	124
PageAssembler	125
Additional Methodology Benefits	128
Missing DLLs.....	128
Exceptions in Embeddables	129
A More Complex Emeddable	130
Breadcrumbs.....	130
Navigating the CMS Tree	131
Summary.....	133
■ Chapter 5: jQuery and Ajax in the Presentation Tier	135
An Introduction to jQuery	135
The \$() Factory	136
Implicit Iteration	138
Ajax via jQuery	139
Caching and Ajax	140
Avoiding Caching on GET Requests	141
Lightweight Content Delivery with Ajax and HTTP Handlers.....	142
Handling Asynchronous Errors.....	147
Handling DOM Modifications.....	150
Improving the CMS Admin with jQuery	154

Creating Collapsible Panels	156
Expanding with jQuery	158
Displaying the JavaScript Disabled Message	160
Poka-Yoke Devices	161
Summary.....	164
■ Chapter 6: Distributed Caching via Memcached	165
What Is a Distributed Cache, and Why Is it Important?	165
Memcached	166
Acquiring a Memcached Client Library	167
Getting Started with Memcached	168
Complex Object Types.....	178
Protocol Considerations	181
Memcached Internals and Monitoring	183
Building a Cache-Friendly Site Tree.....	185
Visualizing the Tree.....	186
Defining a Node.....	186
Defining the Tree.....	187
Finding Nodes	188
Inserting Nodes	190
Serializing/Deserializing the Tree for Memcached Storage.....	191
Memcached Configuration Considerations	194
Summary.....	196
■ Chapter 7: Scripting via IronPython	197
How Does the CMS Benefit from Scripting Capabilities?	197
Easier Debugging	197
Rapid Prototyping.....	198
An Introduction to IronPython and Its Syntax.....	198
What Is IronPython?	198
Installing IronPython	199

The IronPython Type System	200
Creating Classes and Controlling Scope	203
Constructors as Magic Methods	206
self	207
Exception Handling	211
Conditional Logic, Iterators, and Collections.....	214
Accessors and Mutators	216
Assembly Compilation.....	217
Compiling IronPython Code to a DLL.....	218
Compiling IronPython Code to an Executable	219
Building Scripting Capabilities into the CMS.....	220
Handling Script Files Between Tiers	223
Calling Scripts for a CMS Page	224
A Simple Scripting Example.....	226
Summary.....	228
■ Chapter 8: Performance Tuning, Configuration, and Debugging.....	229
The CMS Definition of Performance	229
Latency.....	229
Throughput.....	230
Establishing Baselines	230
Component vs. System Baselines	230
The Web Capacity Analysis Tool	231
Installing WCAT	231
WCAT Concepts.....	231
Configurations.....	232
Scenarios	233
Running a WCAT Test Against the CMS	235
Interpreting Performance Results	236
Improving CMS Performance with Caching	237
HTTP.sys and the OutputCache.....	237

Benchmarking CMS Performance	238
Configuration Considerations.....	239
Enable Release Mode for Production	239
Removing the Server, X-Powered-By, and X-AspNet-Version Headers	241
Debugging Concepts.....	244
White-Box vs. Black-Box Debuggers	244
User Mode vs. Kernel Mode	245
Historical Debugging via IntelliTrace	246
Collaborative Debugging.....	250
Importing and Exporting Breakpoints	250
DataTip Pinning and Annotation.....	253
Summary.....	256
■ Chapter 9: Search Engine Optimization and Accessibility	257
An Introduction to Search Engine Optimization	257
Science or Dark Art?	258
General Guidelines	258
Friendly URLs	265
Data Requirements for Friendly URLs	265
Stored Procedures for Friendly URLs	268
Exploiting the Page Life Cycle.....	269
Request Exclusion.....	271
Retrieving Friendly URLs	273
Retrieving Alias URLs and Response.RedirectPermanent().....	279
Summary.....	283
■ Index	285