



Contents Summary

Typographic Conventions	i
Acronyms	v
Glossary	ix
Introduction	xi
Software & Hardware	xv
Contents Summary	xix
List of Figures	xxiii
Photographic Credits	xxvii

VI. More Functional Programming

■ Functional Programming?!	3
<i>What is Functional Programming?</i>	3
<i>Error Management Prelude</i>	8
■ Function State & Composition	15
<i>Abends and Bypassing Function Evaluation</i>	16
<i>The Technicalities of Function Composition</i>	19
<i>State - Normalising Function Inputs and Outputs</i>	23
<i>Who, What, Where and When am I?</i>	25
<i>Valued & Valueless State</i>	36
■ Function Evaluation Context	41
<i>In a Perfect World...</i>	41
<i>The Lazy Coding Style</i>	46
<i>Incorporating Option Binding in our Evaluation</i>	49
■ Summary	53
<i>Recommended Entertainment!</i>	53

VII. Event Tracing for Windows

■ Semantic, Structured Logging	59
<i>Semantic Logging</i>	60
<i>What makes Semantic logs Structured?</i>	61
■ F# as an ETW Event Source	69
<i>Overview of Expected Header Usage</i>	70
<i>Quantifying Event Emission Permutations</i>	74
<i>An F# EventSource Class</i>	85
■ The Event Payload	105
<i>Some Small Code Changes</i>	105
<i>Serialisation</i>	107
<i>Implementing the ETW Payload</i>	112
■ Emitting ETW Events	125
<i>An ETW Event Mailbox and Agent</i>	125
<i>User Instigation of an ETW Event</i>	130
<i>Testing ETW Event Emission</i>	133
■ Using an ETW Event Sink	139
<i>The SLAB Event Sink</i>	139
<i>ETW PerfView</i>	146

VIII. WMI Performance Metrics

■ Performance Monitoring	153
<i>Instrumentation & Performance Counters</i>	154
<i>Changes to the ETW Code</i>	159
<i>Implementing WMI Performance Counters</i>	162
<i>Updating Performance Counters</i>	169

<i>Running the Code</i>	172	<i>Configuring the SHFB Project</i>	319
■ <i>The Perils of Asynchronicity</i>	179	<i>Building an SHFB Project</i>	329
<i>Mutability</i>	180	■ <i>Debugging the Common Library</i>	337
<i>Missing Events</i>	184	<i>Friendly Assemblies</i>	337
■ <i>Creating a NuGet Package</i>	195	<i>Testing Referenced Internal</i>	
<i>Determining Distribution</i>		<i>Types</i>	338
<i>Dependencies</i>	195	■ <i>Coding the XSettings Type Provider</i>	
<i>The NuGet Package Builder GUI</i>	197	343
<i>Using your local NuGet</i>		<i>Implementing XSettings</i>	345
<i>Repository</i>	201	<i>Instantiation of a Type Provider</i>	
<i>Testing the Instrumentation NuGet</i>		<i>Implementation</i>	351
<i>Package</i>	201	■ <i>Debugging a Type Provider</i>	371
IX. Type Providers		<i>Setting Project Debug</i>	
■ <i>Preamble</i>	211	<i>Properties</i>	372
<i>Goals</i>	212	<i>An Overview of the Microsoft</i>	
■ <i>Static Provision of Data</i>	215	<i>Process Explorer</i>	373
<i>Static versus Dynamic</i>	215	<i>A Script for Type Initialisation</i>	
<i>A Simple, Sample Type Provider</i>	217	<i>Debugging</i>	374
<i>Design Phases & Implementation</i>		<i>Debugging an Instantiated Type</i>	
<i>Boundaries</i>	221	<i>Provider</i>	380
■ <i>Assembly Configuration Settings</i>	227	■ <i>A Manifest Resources Type Provider</i>	
<i>Overview</i>	228	383
<i>Xml Schemas</i>	232	<i>Core Functionality for Manifest</i>	
<i>Testing the Schema</i>	250	<i>Resources</i>	384
<i>Schema Location</i>	251	<i>Extended Functionality for</i>	
■ <i>XSettings Underlying Data Access</i>		<i>Manifest Resources</i>	428
.	253	<i>Implementing the Manifest Type</i>	
<i>Planning for Data Provision</i>	254	<i>Provider</i>	445
<i>Coding Core Functionality</i>	257	■ <i>An Extended Resources Provider</i>	
■ <i>The Type Provider Common Library</i>		459
.	281	<i>Schema Simple Type Enumeration</i>	
<i>Project Layout & Structure</i>		<i>Facets</i>	460
<i>Overview</i>	282	<i>Schemes and Schemas</i>	464
<i>XPath for Attribute Values</i>	286	<i>Modelling the Type Provider</i>	
<i>Enhancing Uri File Referencing</i>	287	<i>Nesting Levels</i>	499
<i>Functions supporting the XSettings</i>		<i>XResources Extended</i>	
<i>Type Provider</i>	305	<i>Functionality</i>	501
■ <i>Generating an API Reference</i>	313	<i>Core Functionality</i>	503
<i>An Overview of Xml Comments</i>	314	<i>Implementing the XResources Type</i>	
<i>Implementing SHFB in your</i>		<i>Provider</i>	519
<i>Solution</i>	317	<i>Using the XResources Type</i>	
		<i>Provider</i>	535

