

# Inhaltsverzeichnis

<b>Vorwort</b>	<b>11</b>
<b>1 CLR und Sprachsyntax (C#/Visual Basic)</b>	<b>13</b>
1.1 .NET Framework 4.5.1 und 4.5.2 erkennen	13
1.2 .NET Framework 4.6 erkennen	15
1.3 Den Large Object Heap komprimieren	16
1.4 C# 6.0 und Visual Basic 14 in älteren .NET-Projekten nutzen	18
1.5 Einsatz der dynamischen Typisierung in C#	19
1.6 Kovarianz (Covariance) in C#	22
1.7 Kontravarianz (Contravariance) in C#	24
1.8 Ko- und Kontravarianz in Visual Basic .NET	26
1.9 Null-conditional Operator in C# 6.0 und Visual Basic 14	26
1.9.1 Motivation	26
1.9.2 Der neue Null-conditional Operator	28
1.9.3 Null-conditional Operator und Ereignisse	28
1.9.4 Null-conditional Operator und Indexer	29
1.10 String-Interpolation in C# 6.0 und Visual Basic 14	29
1.11 Operator „nameof“ in C# 6.0 und Visual Basic 14	30
1.12 Exception-Filter in C# 6.0	32
<b>2 .NET Framework Class Library (FCL)</b>	<b>35</b>
2.1 ExpandoObject	35
2.2 Prüfung auf 64 Bit	35
2.3 BigInteger	36
2.4 Standortermittlung	37
2.5 Interprozesskommunikation mit Memory-mapped Files	38
2.6 Auf Textdateien mittels LINQ zugreifen	39
2.7 Erweiterungsmethode „String.Truncate()“	40

2.8	Erweiterungsmethoden „String.ToDateTime()“, „ToInt32()“, „ToDecimal()“	41
2.9	Eine einfache Objektausgabefunktion für alle .NET-Objekte	45
2.10	Zugriff auf COM-Bibliotheken ohne Primary Interop Assemblies (NoPIA)	50
2.11	Übertragen von Daten zwischen Streams	50
2.12	Enums und Bitmasken	51
2.13	Caching mit „System.Runtime.Caching“	52
2.14	Caching-Datenmenge begrenzen	53
2.15	Caching ganz einfach per Cachemanager	55
2.16	Verzögertes Instanziiieren mit Lazy	63
2.17	Tuples	65
2.18	SortedSet	67
2.19	Observer	68
2.20	API-basierte Konfiguration in MEF 2	71
2.21	„InnerException“-Ausgabe ohne Stacktrace	75
2.22	Dateien mit ZIP komprimieren	77
2.23	Den angemeldeten Benutzer ermitteln	80
2.24	Eigenschaften eines Benutzerkontos ändern	82
2.25	Benutzerinformationen auslesen	85
2.26	Ein neues Benutzerkonto anlegen	89
<b>3</b>	<b>ADO.NET und Entity Framework</b>	<b>93</b>
3.1	Abgebrochene Datenverbindung automatisch neu aufbauen	93
3.2	Ladeoptimierung durch Abfragen ohne Änderungsverfolgung (No-Tracking Queries)	97
3.3	Objekte löschen, ohne sie vorher zu laden	100
3.4	Setzen des Concurrency Mode für alle Spalten in der EDMX-Datei	101
3.5	Setzen des Concurrency Mode für alle Spalten bei Code-based Modeling	103
3.6	Entity Framework Logging	108

3.7	Entity Framework Profiling	113
3.8	Speicherooperationen optimieren	117
3.9	Massenoperationen mit Entity Framework	121
3.10	UPDATE und DELETE per Lambdalausdruck	122
<b>4</b>	<b>Windows Management Instrumentation (WMI)</b>	<b>125</b>
4.1	Liste der verfügbaren Laufwerke	125
4.2	Füllstand der Laufwerke auflisten	127
4.3	Computer neustarten	128
4.4	Laufwerksname ändern	130
4.5	Computer umbenennen	131
4.6	Hardware auflisten	132
<b>5</b>	<b>ASP.NET Web Forms und MVC</b>	<b>135</b>
5.1	C# 6 und Visual Basic 14 in ASP.NET-Webseiten verwenden	135
5.2	Tipps zur Leistungssteigerung in ASP.NET Web Forms	136
5.3	Leistungssteigerung durch Seiten-Caching	137
5.3.1	Caching-Profile in der „web.config“-Datei	141
5.3.2	Caching einzelner Steuerelemente	141
5.4	Asynchrone Controller in ASP.NET MVC	143
5.5	Minification und Bundling	144
5.6	In ASP.NET 4.x wie in ASP.NET 3.5 rendern	146
5.7	HTML Encoded Code Expressions in ASP.NET 4.0	147
5.8	Vorlagen für Felder und Models in ASP.NET MVC	149
5.9	Razor Helper für Views ASP.NET MVC	151
5.10	Views für mobile Anwendungen in ASP.NET MVC	152
5.11	ASP.NET-MVC-Modelle mit jQuery Validate validieren	154
5.12	Sprach- und Ländereinstellungen für ASP.NET MVC festlegen	157
5.13	Bei ASP.NET MVC 4 Seiten über Google, Facebook, Twitter und Co. anmelden	159
5.14	Pipelinemodule für Querschnittsfunktionen in ASP.NET SignalR	162

<b>6</b>	<b>Windows Communication Foundation (WCF)</b>	<b>165</b>
6.1	Kerberos vs. NTLM	165
6.2	Antwortformat bei REST-Services dynamisch festlegen	166
6.3	Bandbreite mit „EmitDefaultValue=false“ sparen	167
6.4	Lebensdauer von Sessions beeinflussen	169
6.5	Fehlerdetails bei WCF Services anzeigen	170
6.6	Hilfeseite für REST-Services	172
6.7	Anpassung der Serialisierung von String-Listen mit „CollectionDataContractAttribute“	172
6.8	Leistungsindikatoren für WCF-Services	174
6.9	Einfluss auf den Mengentyp im Proxy	175
6.10	Port Sharing bei TCP-basierten Services	176
6.11	UDP Binding und Multicasts in WCF 4.5	178
6.12	Erweiterbare Datenverträge	180
6.13	Programmatische Impersonation	181
6.14	Kompression bei binärer Kodierung in WCF 4.5	181
6.15	Deklarative Impersonation	182
6.16	Impersonation für alle Operationen festlegen	183
6.17	Vereinfachte Codekonfiguration in WCF 4.5	183
6.18	Unterstützung für mehrere Authentifizierungsarten pro Endpunkt in WCF 4.5	186
<b>7</b>	<b>WCF Data Services</b>	<b>189</b>
7.1	Schnell erstellte CRUD Web Services mit WCF Data Services	189
7.2	Fehlermeldungen aktivieren	193
7.3	Zugriffsrechte einschränken	194
7.4	Datenmengenbeschränkungen aktivieren	197
7.5	Serverseitiges Paging	198
7.6	Zeilen zählen	200
7.7	Individuelle Dienstoperationen	201

7.8	Individuelle generische Dienstoperationen	204
7.9	Hosting eines WCF Data Service in eigenen Anwendungen	206
<b>8</b>	<b>ASP.NET Web API</b>	<b>207</b>
8.1	ASP.NET Web API ohne IIS verwenden	207
8.2	JSON-Serialisierung bei ASP.NET Web API anpassen	208
8.3	Zirkuläre Referenzen mit ASP.NET Web API serialisieren	210
8.4	Clientseitige Proxies für Web APIs generieren	213
8.5	Web APIs mit Swagger dokumentieren	215
8.6	Swashbuckle zur Generierung von Dokumentationen für Web APIs konfigurieren	217
8.7	ASP.NET Web API: Fortschritt ermitteln	220
8.8	Tracing in ASP.NET Web API	221
8.9	Controllerbasierte Konfiguration in ASP.NET Web API	224
8.10	Routenbasierte Konfiguration	225
8.11	SSL mit ASP.NET Web API ohne IIS nutzen	226
8.12	OData mit ASP.NET Web API	227
<b>9</b>	<b>AngularJS</b>	<b>229</b>
9.1	Unterstützung für ECMAScript 6 Promises in AngularJS ab 1.3	229
9.2	Verschachtelte Formulare mit AngularJS validieren	230
9.3	Mit AngularJS auf unsichere Eingaben reagieren	233
9.4	Validierungsfehler komfortabel mit AngularJS und ngMessages anzeigen	236
<b>10</b>	<b>Windows Presentation Foundation (WPF)</b>	<b>239</b>
10.1	Wartecursor anzeigen	239
10.2	Eigene Cursorgrafiken	242
10.2.1	Cursorgrafiken erstellen	242
10.2.2	Cursordateien zuweisen	243
10.2.3	Multi-Image-Cursordateien für High-DPI-Displays	243

10.3	Fenster via ViewModel öffnen und manipulieren	245
10.4	WPF-View an beliebige Methoden in ViewModel binden	247
10.5	Windows-7-Integration bei WPF	250
10.5.1	Vorschaufenster	250
10.5.2	Schaltflächen im Vorschaufenster (Thumbnail-Buttons)	251
10.5.3	Symbole und Fortschrittsanzeige in der Taskleiste	252
<b>11</b>	<b>Visual Studio und andere Werkzeuge</b>	<b>255</b>
11.1	Codewiederverwendung mit Portable Class Libraries (PCLs)	255
11.2	Den Überblick bei langen Fehlerlisten behalten	259
11.3	Fehlerschlangenlinien im Projektmappen-Explorer	260
11.4	Webseitenprobleme mit dem Page Inspector analysieren	261
11.5	Copy-and-Paste-Entwicklung entlarven	263
11.6	Metadaten mit Code Lens	265
11.7	Rückgabewerte im Visual-Studio-Debugger analysieren	265
11.8	Schneller Webseiten in vielen Browsern mit Browser Link überprüfen	266
11.9	Festen HTTP-Port für den ASP.NET Development Server vergeben	269
11.10	HTTP-Port für den IIS Express ändern	270
11.11	Visual Studio Power Productivity Tools	273
11.12	Verbesserter Solution Explorer	274
11.13	Verbesserte Registerkartenverwaltung mit Visual Studio 2010 Productivity Power Tools	275
11.14	Mit Fiddler sehen, wie Browser und Server kommunizieren	276
11.15	Mit „Fiddler“ den Localhost abhören	279
	<b>Über die Autoren</b>	<b>281</b>
	<b>Stichwortverzeichnis</b>	<b>283</b>