

Klaus Löffelmann

## **Visual Basic 2008 – Neue Technologien – Crashkurs**



Klaus Löffelmann

# **Visual Basic 2008 Neue Technologien Crashkurs**

**Microsoft**<sup>®</sup>  
Press

Klaus Löffelmann: Visual Basic 2008 – Neue Technologien – Crashkurs  
Microsoft Press Deutschland, Konrad-Zuse-Str. 1, 85716 Unterschleißheim  
Copyright © 2008 by Microsoft Press Deutschland

Das in diesem Buch enthaltene Programmmaterial ist mit keiner Verpflichtung oder Garantie irgendeiner Art verbunden. Autor, Übersetzer und der Verlag übernehmen folglich keine Verantwortung und werden keine daraus folgende oder sonstige Haftung übernehmen, die auf irgendeine Art aus der Benutzung dieses Programmmaterials oder Teilen davon entsteht. Die in diesem Buch erwähnten Software- und Hardwarebezeichnungen sind in den meisten Fällen auch eingetragene Marken und unterliegen als solche den gesetzlichen Bestimmungen. Der Verlag richtet sich im Wesentlichen nach den Schreibweisen der Hersteller.

Das Werk, einschließlich aller Teile, ist urheberrechtlich geschützt. Jede Verwertung außerhalb der engen Grenzen des Urheberrechtsgesetzes ist ohne Zustimmung des Verlags unzulässig und strafbar. Das gilt insbesondere für Vervielfältigungen, Übersetzungen, Mikroverfilmungen und die Einspeicherung und Verarbeitung in elektronischen Systemen.

15 14 13 12 11 10 9 8 7 6 5 4 3 2 1  
10 09 08

ISBN 978-3-86645-518-4

© Microsoft Press Deutschland  
(ein Unternehmensbereich der Microsoft Deutschland GmbH)  
Konrad-Zuse-Str. 1, D-85716 Unterschleißheim  
Alle Rechte vorbehalten

Korrektorat: Kristin Grauthof, Lippstadt  
Fachlektorat: Walter Saumweber  
Satz: Silja Brands, ActiveDevelop, Lippstadt (<http://www.ActiveDevelop.de>)  
Umschlaggestaltung: Hommer Design GmbH, Haar ([www.HommerDesign.com](http://www.HommerDesign.com))  
Gesamtherstellung: Kösel, Krugzell ([www.KoeselBuch.de](http://www.KoeselBuch.de))

*Für*

*Ada & Zoe*

*Sarah & David*

*Ute & Ruprecht*



# Inhaltsverzeichnis

Ein nicht ganz so ernst gemeintes Vorwort .....	<b>XIII</b>
<a href="http://www.activedevelop.de">http://www.activedevelop.de</a> – Ein wenig Werbung in eigener Sache .....	<b>XV</b>
Codebeispiele .....	<b>XVI</b>
Support.....	<b>XVI</b>
Danksagungen .....	<b>XVII</b>
Teil A – Schnell zurechtfinden .....	<b>1</b>
1 Einführung .....	<b>3</b>
Die Geschichte von .NET .....	<b>4</b>
.NET Framework 3.0.....	<b>4</b>
.NET Framework 3.5.....	<b>5</b>
Welche Softwarevoraussetzungen benötigen Sie? .....	<b>6</b>
Wissenswertes zur Installation von Visual Studio 2008.....	<b>6</b>
Lauffähigkeit von Visual Studio und den erstellten Kompilaten unter den verschiedenen Betriebssystemen .....	<b>8</b>
Parallelinstallation verschiedener Visual Studio-Versionen .....	<b>9</b>
Wieso fehlen noch Features? – Visual Studio 2008 Service Pack 1 .....	<b>9</b>
Visual Studio 2008 und SQL Server 2005 bzw. SQL Server 2008.....	<b>10</b>
Die weiteren Kapitel in diesem Teil .....	<b>10</b>
2 Das ist neu in der IDE von Visual Studio 2008 .....	<b>13</b>
Der erste Start von Visual Studio .....	<b>14</b>
Zurücksetzen der IDE-Einstellungen .....	<b>15</b>
Verbesserungen an der integrierten Entwicklungsumgebung (IDE).....	<b>16</b>
Einfacheres Positionieren und Andocken von Toolfenstern.....	<b>16</b>
Navigieren durch Dateien und Toolfenster mit dem IDE-Navigator .....	<b>16</b>
Umgebungsschriftart definieren .....	<b>17</b>
Übernehmen von Visual Studio 2005-Einstellungen .....	<b>18</b>
Visual Studio 2005-Projekte nach Visual Studio 2008 migrieren .....	<b>19</b>
Designer für Windows Presentation Foundation-Projekte .....	<b>20</b>
IntelliSense-Verbesserungen.....	<b>22</b>
Syntax-Tooltips .....	<b>22</b>
Filtern beim Tippen .....	<b>23</b>
Inhalte von System- oder externen Typen in Vorschaufenstern fürs Debuggen konfigurieren	<b>24</b>
Zugriff auf den .NET Framework-Quellcode beim Debuggen .....	<b>26</b>
3 Das ist neu im Compiler von Visual Basic 2008 .....	<b>31</b>
First and definitely not least – die Performance des Visual Basic 2008-Compilers .....	<b>32</b>

.NET Framework Version Targeting (.NET Framework-Versionszielwahl) bei Projekten .....	<b>33</b>
Lokaler Typrückschluss.....	<b>36</b>
Typrückschluss für Typparameter bei Generics .....	<b>36</b>
Generelles Einstellen von Option Infer, Strict, Explicit und Compare .....	<b>36</b>
If-Operator vs. Iif-Funktion .....	<b>37</b>
Festlegen der Projekteinstellungen für die Benutzerkontensteuerung (Windows Vista) .....	<b>38</b>
Nullable-Typen.....	<b>39</b>
Besonderheiten bei Nullable beim Boxen .....	<b>41</b>
Anonyme Typen.....	<b>43</b>
Lambda-Ausdrücke .....	<b>44</b>
Abfrageausdrücke mit LINQ.....	<b>44</b>
Erweiterungsmethoden.....	<b>44</b>
Der eigentliche Zweck von Erweiterungsmethoden.....	<b>45</b>
<b>4 Klassen und Objekte .....</b>	<b>47</b>
Einführung.....	<b>48</b>
Miniadresso – die prozedurale Variante .....	<b>48</b>
Was ist eine Klasse?.....	<b>52</b>
Klassen mit New instanziiieren.....	<b>53</b>
New oder nicht New – wieso es sich bei Objekten um Verweistypen handelt.....	<b>54</b>
Nothing .....	<b>57</b>
Klassen anwenden.....	<b>57</b>
Klassencode .....	<b>59</b>
Eigenschaftenprozeduren (Property-Prozeduren) .....	<b>59</b>
Öffentliche Variablen oder Eigenschaften – eine Glaubensfrage? .....	<b>63</b>
Klassenmethoden mit Sub und Function.....	<b>66</b>
Zugriffsmodifizierer für Klassen, Methoden und Eigenschaften.....	<b>67</b>
Zugriffsmodifizierer bei Klassen .....	<b>67</b>
Zugriffsmodifizierer bei Prozeduren (Subs, Functions, Properties) .....	<b>67</b>
Zugriffsmodifizierer bei Variablen.....	<b>68</b>
Unterschiedliche Zugriffsmodifizierer für Eigenschaften-Accessors .....	<b>69</b>
Vererbung, Polymorphie, Abstrakte Klassen und Schnittstellen.....	<b>70</b>
Statische Komponenten .....	<b>71</b>
Statische Methoden .....	<b>71</b>
Statische Eigenschaften .....	<b>72</b>
Module in Visual Basic – automatisch statische Klassen erstellen.....	<b>73</b>
Delegaten und Lambda-Ausdrücke .....	<b>73</b>
Umgang mit Delegaten am Beispiel.....	<b>74</b>
Lambda-Ausdrücke .....	<b>76</b>
Lambda-Ausdrücke am Beispiel.....	<b>77</b>
<b>5 Arrays und Auflistungen .....</b>	<b>79</b>
Grundsätzliches zu Arrays.....	<b>80</b>
Änderung der Array-Dimensionen zur Laufzeit .....	<b>81</b>

Wertevorbelegung von Array-Elementen im Code .....	<b>83</b>
Mehrdimensionale und verschachtelte Arrays .....	<b>84</b>
Die wichtigsten Eigenschaften und Methoden von Arrays .....	<b>85</b>
Implementierung von Sort und BinarySearch für eigene Klassen .....	<b>87</b>
Enumeratoren .....	<b>95</b>
Benutzerdefinierte Enumeratoren durch Implementieren von IEnumerable .....	<b>96</b>
Grundsätzliches zu Auflistungen (Collections) .....	<b>98</b>
Wichtige Auflistungen der Base Class Library .....	<b>102</b>
ArrayList – universelle Ablage für Objekte .....	<b>102</b>
<b>6 Generics (Generika) und generische Auflistungen.....</b>	<b>109</b>
Einführung.....	<b>110</b>
Generics: Verwenden einer Codebasis für verschiedene Typen .....	<b>110</b>
Lösungsansätze .....	<b>111</b>
Typengeneralisierung durch den Einsatz generischer Datentypen .....	<b>113</b>
Beschränkungen (Constraints).....	<b>116</b>
Beschränkungen für generische Typen auf eine bestimmte Basisklasse .....	<b>116</b>
Beschränkungen auf Klassen, die bestimmte Schnittstellen implementieren.....	<b>121</b>
Beschränkungen auf Klassen, die über einen Standardkonstruktor verfügen .....	<b>124</b>
Beschränkungen auf Wertetypen.....	<b>124</b>
Kombinieren von Beschränkungen und Bestimmen mehrerer Typparameter .....	<b>125</b>
Generische Auflistungen (Generic Collections) .....	<b>126</b>
List(Of )-Auflistungen und Lambda-Ausdrücke .....	<b>128</b>
ForEach und die generische Action-Klasse.....	<b>129</b>
Sort und die generische Comparison-Klasse .....	<b>130</b>
Find und die generische Predicate-Klasse .....	<b>131</b>
<b>Teil B - Language Integrated Query (LINQ) .....</b>	<b>133</b>
<b>7 Einführung in LINQ .....</b>	<b>135</b>
Wie »geht« LINQ prinzipiell.....	<b>138</b>
Die Where-Methode .....	<b>141</b>
Die Select-Methode.....	<b>141</b>
Anonymen Typen .....	<b>142</b>
Typrückschluss für generische Typparameter.....	<b>143</b>
Die OrderBy-Methode.....	<b>145</b>
Sortieren nach Eigenschaften, die per Variable übergeben werden .....	<b>147</b>
Die GroupBy-Methode.....	<b>148</b>
Kombinieren von LINQ-Erweiterungsmethoden .....	<b>150</b>
Vereinfachte Anwendung von LINQ – Erweiterungsmethoden mit der LINQ- Abfragesyntax .....	<b>151</b>
<b>8 LINQ to Objects .....</b>	<b>153</b>
Einführung in LINQ to Objects .....	<b>154</b>
Verlieren Sie die Skalierbarkeit von LINQ nicht aus den Augen! .....	<b>154</b>

Der Aufbau einer LINQ-Abfrage.....	155
Kombinieren und verzögertes Ausführen von LINQ-Abfragen .....	161
Faustregeln für das Erstellen von LINQ-Abfragen .....	163
Kaskadierte Abfragen .....	163
Gezieltes Auslösen von Abfragen mit ToArray oder ToList .....	164
Verbinden mehrerer Auflistungen zu einer neuen.....	165
Implizite Verknüpfung von Auflistungen.....	166
Explizite Auflistungsverknüpfung mit Join .....	167
Gruppieren von Auflistungen .....	168
Gruppieren von Listen aus mehreren Auflistungen .....	170
Group Join .....	171
Aggregatfunktionen.....	172
Zurückliefern mehrerer verschiedener Aggregierungen .....	172
Kombinieren von gruppierten Abfragen und Aggregatfunktionen.....	173
9 LINQ to SQL .....	175
Einleitung.....	176
Voraussetzungen für die Beispiele dieses Kapitels .....	176
Beta-Warnung.....	177
Wie es bisher war – Visual Studio 2005 vs. Visual Studio 2008 .....	177
Die ersten Schritte.....	179
Kaskadierte Abfragen .....	184
Daten verändern und speichern .....	187
Concurrency-Check (Schreibkonfliktprüfung).....	188
Transaktionen.....	191
TransactionScope (Transaktionsgültigkeitsbereich).....	191
Verwenden der Transaktionssteuerung des DataContext.....	192
Was, wenn LINQ einmal nicht reicht.....	192
10 LINQ to XML .....	195
Einführung in LINQ to XML.....	196
Vergleich Visual Basic 8 und Visual Basic 9 .....	196
XML-Literale – XML direkt im Code ablegen.....	197
Erstellen von XML mithilfe von LINQ .....	198
Abfragen von XML-Dokumenten mit LINQ to XML.....	200
IntelliSense-Unterstützung für Linq to XML-Abfragen.....	201
Teil C - Windows Presentation Foundation (WPF).....	205
11 WPF – Einführung und Grundlagen .....	207
Einführung.....	208
Aller Anfang ist schwer .....	208
Architektur.....	209

Das .NET Framework ab Version 3.0 .....	<b>210</b>
WPF, .NET Framework 3.5 und Version Targeting .....	<b>211</b>
Vektorgrafik.....	<b>211</b>
Trennung von Design und Logik .....	<b>211</b>
XAML: Extensible Application Markup Language .....	<b>214</b>
Der WPF-Designer.....	<b>221</b>
Ereignisbehandlungsroutinen in WPF und Visual Basic.....	<b>222</b>
Logischer und visueller Baum .....	<b>223</b>
Die XAML-Syntax im Überblick .....	<b>225</b>
Ein eigenes XAMLPad.....	<b>228</b>
Zusammenfassung.....	<b>233</b>
<b>12</b> Steuerelemente .....	<b>235</b>
Einführung.....	<b>236</b>
Weitergeleitete Ereignisse (Routed Events) .....	<b>237</b>
Weitergeleitete Befehle (Routed Commands) .....	<b>246</b>
Eigenschaften der Abhängigkeiten.....	<b>250</b>
Eingaben .....	<b>253</b>
Schaltflächen .....	<b>255</b>
Bildlaufleisten und Schieberegler.....	<b>257</b>
Steuerelemente für die Texteingabe .....	<b>261</b>
Das Label-Element .....	<b>265</b>
Menüs.....	<b>266</b>
Werkzeugleisten (Toolbars) .....	<b>271</b>
Zusammenfassung.....	<b>274</b>
<b>13</b> Layout.....	<b>275</b>
Das StackPanel.....	<b>276</b>
Das DockPanel.....	<b>278</b>
Das Grid.....	<b>282</b>
Das GridSplitter-Element.....	<b>288</b>
Das UniformGrid .....	<b>291</b>
Das Canvas-Element.....	<b>292</b>
Das Viewbox-Element .....	<b>293</b>
Text-Layout .....	<b>295</b>
Das WrapPanel.....	<b>299</b>
Standard-Layout-Eigenschaften .....	<b>300</b>
Width- und Height-Eigenschaft .....	<b>300</b>
MinWidth-, MaxWidth-, MinHeight- und MaxHeight-Eigenschaft.....	<b>300</b>
HorizontalAlignment- und VerticalAlignment-Eigenschaft.....	<b>301</b>
Margin-Eigenschaft.....	<b>301</b>
Padding-Eigenschaft .....	<b>302</b>
Zusammenfassung.....	<b>303</b>

14 Grafische Grundelemente.....	<b>305</b>
Grundlagen.....	<b>306</b>
Die Grafik-Auflösung.....	<b>313</b>
Die grafischen Grundelemente.....	<b>315</b>
Rechteck und Ellipse.....	<b>316</b>
Einfache Transformationen.....	<b>318</b>
Die Linie.....	<b>320</b>
Die Polylinie.....	<b>320</b>
Das Path-Element.....	<b>325</b>
Hit-Testing mit dem Path-Element.....	<b>327</b>
Zusammenfassung.....	<b>329</b>
Stichwortverzeichnis.....	<b>331</b>

# Ein nicht ganz so ernst gemeintes Vorwort

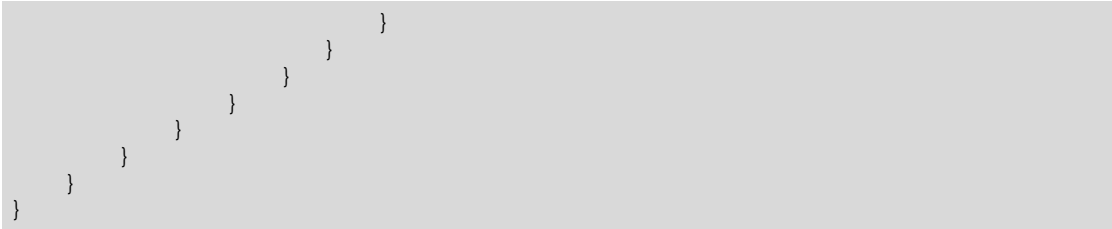
Modernste Technik macht frei. Ich gönnte mir vor einiger Zeit einen Windows Vista Tablet PC und eine UMTS-Flatrate. Und das bedeutet: Ich kann arbeiten und damit Vorworte von Büchern schreiben, wo und wann ich will. Wie in diesem Beispiel: Ich schreibe die Zeilen dieses Vorworts tatsächlich in einer Turnhalle irgendwo in Ratingen-Hösel auf drei Turnkästen liegend, während ich immer wieder zweien meiner vier Lieblingskinder beim Badminton zuschaue – in der freien Zeit zwischen zwei Fahren, in der ich (sehr, sehr gerne, übrigens) als ihr Fahrer fungiere.

Und während ich Sarah und David bewundernd zuschaue, und rätsle, wo sie ihre Reaktion hernehmen, und Bälle treffen, die ich schon mit bloßem Auge nicht mehr verfolgen kann, denke ich über eine kleine Rede ihres Papas nach. Das übrigens ist Ruprecht Dröge, SQL-Experte (für mich der Beste, für viele andere übrigens auch, selbst wenn er das immer abstreitet), ursprünglich Fachlektor meines Visual Basic Entwicklungsbuches und inzwischen sehr guter Freund. Diese Rede hielt er ein paar Tage nach der Basta 2007 in seiner Küche, kurz bevor wir einmal mehr das leckerste Sushi der Welt machten. In dieser Rede ging es mal wieder um *die* immerwährende Diskussion, bzw. die Abneigung gegen dieselbe, und seine schließlich gezogene Quintessenz kann ich uneingeschränkt unterschreiben.

Gleich ihm habe ich die ewigen Diskussionen über *dieses Thema* ja so satt. Mir steht es bis hier. Sie sind kindisch, sie führen zu nichts, und wenn die C#ler in der Überzahl sind, dann haben sie mit all ihren Argumenten einfach nur Unrecht – laut brüllen ändert leider nichts an den Fakten. Um was geht's? Ganz einfach: Um die unendliche Geschichte auf Kongressen, Conventions, auf Bastas und sonstigen Events, wo wir Entwickler uns immer treffen und über das leidige Thema: »Was ist besser – Visual Basic oder C#?« endlos lange philosophieren.

Liebe C#ler, die ihr dieses Buch wahrscheinlich gar nicht wahrnehmt, es sogar schmätzt und schneidet, und liebe Visual Basicler, die ihr euch nach 16 Jahren immer noch gegen die C#ler rechtfertigen müsst, weil ihr eure Anwendungen ja mit einer scheinbaren »Kinderprogrammiersprache« entwickelt, hier meine Argumentation für Visual Basic .NET. Endgültig – im Übrigen – denn ich bin unfassbar erfahren, habe die Weisheit in dieser Sache mit Löffeln verspeist und *darf* damit arrogant genug sein, dieser Diskussion ein für alle Mal ein Ende zu setzen: ;-)

1. Geschweifte Klammern lassen Programme *nicht* professioneller ausschauen. Was für ein fragwürdiges Argument soll das denn sein? (Und Leute, die mich kennen, nehmen »fragwürdig« hier als Platzhalter für andere Adjektive, die leider nichts in einem Computerbuch zu suchen haben!) Ganz im Gegenteil: Visual Basic hat den Vorteil, Strukturbefehle zur Verfügung zu stellen, bei denen man genau weiß, zu welchem Ende welcher Anfang gehört: End If gehört zu If; Loop gehört zu Do; Next zu For; End With zu With (ach ja, diese letzt genannte Schreibersparnisfunktion kennen die C#ler ja gar nicht – ups!). Bei C# hingegen sieht das Ende eines komplizierten Algorithmus folgendermaßen aus:



1:0 für Visual Basic.

2. Wenn die Fehlerliste von Visual Basic .NET leer ist, lässt sich ein VB-Programm starten. Klar – dann können natürlich noch logische Fehler darin sein; »anlaufen« wird das Programm aber erst einmal. Das ist in C# nicht so – jedenfalls nicht, ohne Zusatztools dazu kaufen zu müssen. In C# können sich zu diesem Zeitpunkt immer noch Referenzen auf Methoden, Eigenschaften, Feldvariablen oder sonstige Elemente befinden, die der Compiler nicht auflösen konnte, weil der Entwickler schlicht vergessen hat, sie zu deklarieren, zu implementieren oder er sich einfach nur vertippt hat. Das bedeutet dann: Programm kompilieren, feststellen, ob Fehler drin sind, Fehler raus machen, Programm wieder kompilieren, wieder schauen, ob noch Fehler drin sind, wieder Fehler raus machen, usw.

Im Übrigen ist das auch ein Vorteil von Visual Basic .NET gegenüber Visual Basic 6.0. Ein volles Kompilieren hat dort nur die erste nicht definierte Variable oder Methode gefunden; Sie mussten diese erst fixen, um anschließend wieder in die – so noch vorhanden – nächste nicht definierte Variable oder Methode zu rasseln. Deswegen sind die Turn-Around-Zeiten in Visual Basic 6.0 auch nur scheinbar kleiner als in Visual Basic .NET.

2:0 für Visual Basic.

3. C# ist, wie C++ und Java, mal definitiv nicht für die deutsche Tastatur gemacht. Die geschweiften Klammern, die man alle 7 Sekunden benötigt, sind ein Sehnenscheidenentzündungsgarant. Mehr Tipparbeit erfordert Visual Basic .NET dennoch nicht: Das Tippen eines If führt automatisch zur Einfügung von End If, das von Class zu End Class und das von Function zu End Funtion. Dieses Verfahren ist bei allen Strukturbefehlen dasselbe.

Also: 3:0 für Visual Basic .NET.

4. Nicht nur für Ein- und Umsteiger sind die vorhandenen Codeausschnitte eine enorme Erleichterung: Mit `[Strg][K]`, `[Strg][X]` gibt es in Visual Basic 2008 (und im übrigen auch schon in der 2005er Version) eine riesige Bibliothek mit kleinen Codesnippets, die wiederkehrende Coding-Aufgaben auf ein Minimum an Zeitaufwand reduzieren. Diese Snippets gibt es vom Grundsatz her auch in C# – allerdings ist hier nicht der Arbeitsaufwand sondern die Codeausschnittsbibliothek auf ein Minimum reduziert.

Und damit: 4:0 für Visual Basic.

Fairerweise gebe ich zu, dass es einige Argumente für Visual Basic .NET gibt, die gerne in solchen Diskussionen genannt werden, aber nicht wirklich pro Visual Basic .NET sind. Beispielsweise, dass es ein Haufen von »altem« Visual Basic 6.0-Code gibt, den man einfach per *Cut/Copy/Paste* in .NET-Projekten zum Laufen bekommt – leider nicht, denn VB6 und VB.NET sind zu verschieden, als dass das sicher klappen würde; das trifft eher nur in wenigen Ausnahmefällen zu. Und auch die leichtere Erlernbarkeit von Visual Basic .NET im Gegensatz zu C# trifft nur bedingt zu. Sicherlich »fühlen« sich Visual Basic 6.0-Programmierer anfangs besser aufgehoben, da ihnen grundsätzliche Sprachelemente bekannter vorkommen; doch ehrlicherweise

muss ich aus meinen Schulungserfahrungen sagen, dass das spätestens beim Thema OOP aufhört. Dort ist vielleicht sogar ein sauberer Schnitt eher brauchbar, da bekannte Strukturen nicht zum alten prozeduralen Programmieren wie in VB6 verleiten.

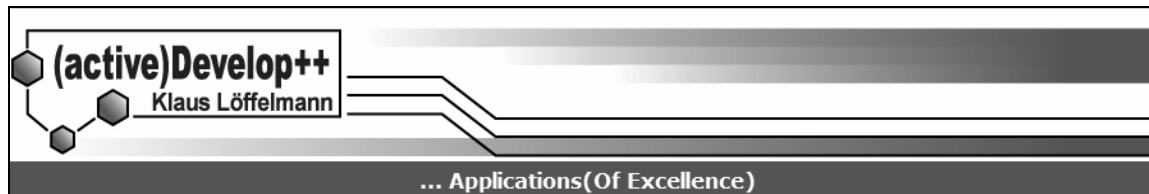
Aber dennoch. Sie sehen, dass Visual Basic .NET eindeutig die produktivere Programmiersprache ist. Und Sie sind vor allen Dingen gerüstet für die nächsten Diskussionen, aus der Sie als Sieger hervorgehen können! Und falls alles nichts hilft, dann schauen Sie Ihrem C#-Diskussionsgegner tief in die Augen, und sprechen Sie C# (eigentlich gesprochen: *βieh schahp*) einfach »zisch ab« aus.

Lassen Sie mich aber noch zum eigentlichen Thema kommen: Vor Ihnen liegt ein Buch, das den Anspruch hat, Ihnen auf möglichst kompakte Art und Weise die Neuheiten in Visual Basic 2008 vorzuführen. Natürlich erhebt es keinen Anspruch auf Vollständigkeit: Alleine das Thema *Windows Presentation Foundation* füllt locker eigene Bücher, sogar Buchreihen. Doch sind die meisten dieser Bücher (jedenfalls im Moment noch) in der Regel für die C#ler geschrieben. Dieses Buch möchte Ihnen deswegen einen Einstieg in alle neuen 2008er Themen ermöglichen, Sie also quasi in die jeweils richtige Richtung schicken, von der aus Sie sich alles weitere dann selber erarbeiten oder mit entsprechender weiterführender Literatur weitermachen können. Ein Tipp dazu: Die Online-Hilfe ist – trotz oftmals hapernder, da mit Übersetzungsfehlern zu kämpfender Texte – gar nicht so schlecht wie ihr Ruf. Gönnen Sie ihr ruhig mehrere Blicke, wenn Sie mit der Lektüre dieses Buches durch sind, oder auch schon währenddessen!

In diesem Sinne,

*Klaus Löffelmann am 20.01.2008*

## <http://www.activedevelop.de> – Ein wenig Werbung in eigener Sache



Man bekommt nicht oft die Möglichkeit, für sein eigenes Business ohne größeren finanziellen Aufwand werben zu können. Umso mehr mag ich natürlich die Gelegenheit am Schopf packen, um an dieser Stelle ein wenig Öffentlichkeitsarbeit für unser eigenes Dienstleistungsunternehmen zu betreiben. So denn: Falls Sie Bedarf an

- Beratungsdienstleistungen in Sachen .NET 1.1, 2.0, 3.5, LINQ und SQL Server 2005/2008,
- Schulungen (auch inhouse) mit den Schwerpunkten »Migration von VB6-Anwendungen«, SQL Server 2005/2008 und ADO.NET 2.0, C# 2005/2008 und Visual Basic 2005/2008 sowie Einführung in neue Technologien zu Visual Studio 2008,
- Dokumentationserstellung, Übersetzungen und Softwarelokalisationen oder
- Software Development Outsourcing

haben, informieren Sie sich einfach auf unserer Website unter <http://www.activedevelop.de>, und setzen Sie sich mit uns in Verbindung.

Mit einem immer größer werdenden Team von Entwicklern und Designern ist ActiveDevelop in der Lage, auch große Projekte zu stemmen – und viele erfolgreiche Bücher und Referenzen von abgeschlossenen Projekten namhafter Firmen unterstreichen unsere fachliche Kompetenz und unsere Zuverlässigkeit.

## Codebeispiele

Der herunterladbare Code umfasst für die meisten Kapitel Projekte, in denen Sie auch die Codefragmente und Beispiele der einzelnen Kapitel finden. Die gesamten im Buch behandelten Codebeispiele können Sie von der Website unter der folgenden Adresse herunterladen:

[www.microsoft-press.de/support.asp?s110=518](http://www.microsoft-press.de/support.asp?s110=518)

Ergänzende Änderungen zum Buch und eine Errata-Seite finden Sie ferner unter

[www.activedevelop.de](http://www.activedevelop.de)

## Support

Es wurden alle Anstrengungen unternommen, um die Korrektheit dieses Buchs und des Begleitinhalts sicherzustellen.

Microsoft Press stellt unter der folgenden Internetadresse eventuell notwendige Korrekturen zu veröffentlichten Büchern zur Verfügung:

<http://www.microsoft.com/germany/mspress/support>

Sollten Sie Anmerkungen, Fragen oder Ideen zu diesem Buch haben, senden Sie diese bitte an eine der folgenden Adressen von Microsoft Press:

Postanschrift:

Microsoft Press

Betrifft: Visual Basic 2008 – Neue Technologien – Crashkurs

Konrad-Zuse-Straße 1

85716 Unterschleißheim

E-Mail:

[presscd@microsoft.com](mailto:presscd@microsoft.com)

Beachten Sie bitte, dass unter der oben angegebenen Adresse kein Produktsupport geleistet wird. Supportinformationen zum .NET Framework, zu Visual Basic .NET oder Visual Studio finden Sie auf der Microsoft-Produktsupportseite unter:

<http://www.microsoft.com/germany/support>

## Danksagungen

Aus vielerlei Gründen blieben mir rund 6 Wochen, dieses Buch zu schreiben – und ehrlich gesagt: das war ein sehr sportlicher Plan. Alleine hätte ich das nicht geschafft. Auf keinen Fall. Immerhin hat dieses Buch nunmehr deutlich über 350 Seiten, obwohl schon die 288 Seiten, mit denen es einst geplant war, nicht zu schaffen gewesen wären, hätte es nicht so viele helfende Hände gegeben.

Und da möchte ich an erster Stelle Bernd Marquardt nennen und mich tausendmal bei ihm bedanken, denn ohne ihn hätte es das Kapitel über die Windows Presentation Foundation nicht gegeben. Ich durfte seinen auf C# ausgerichteten WPF-Crashkurs als Vorlage nehmen, und auch wenn die WPF keine wirklich neue Technologie in Visual Studio 2008 bzw. dem .NET Framework 3.5 ist, so haben wir mit dem Adaptieren der wichtigsten Themen seines Buches viele Leserwünsche erfüllt, die um mehr Visual Basic-Beispiele bei diesem Themenkomplex baten. Deswegen, Bernd, herzlichen Dank dafür! In diesem Zusammenhang möchte ich mich auch bei Jürgen Heckhuis bedanken, der Bernd's C#-Beispiele quasi nach Visual Basic 2008 übersetzt hat.

Herzlicher Dank gilt auch Andreas Belke, der sich mit großem Einsatz um das 9. und 10. Kapitel verdient gemacht hat. Wahrscheinlich war er einer der ersten Entwickler in Deutschland, der schon seit Monaten vor Erscheinen dieses Buches produktiv mit *LINQ to SQL*, *LINQ to Objects* und *LINQ to XML* in einem konkreten Projekt arbeiten durfte, und deswegen mehr als prädestiniert war, diese beiden Kapitel zu schreiben. Die Entscheidung, zu einem so frühen Zeitpunkt bereits eine neue Technologie zu verwenden, hat sich übrigens mehr als richtig herausgestellt, denn wie unsere Erfahrung zeigt: LINQ kann extrem viel Zeit sparen, und wenn das Entity-Framework mit dem SP1 zu Visual Studio 2008 erst einmal da ist, können wir Entwickler uns wirklich über ein richtig »cooles« Set an extrem arbeitserleichternden Werkzeugen freuen.

### Dann war da noch ... mein persönlicher LINQ to Entities!

Und dann gibt's da noch diese liebenswerte Gang in Ratingen-Hösel, an die ich unbedingt ein paar persönliche Worte richten muss: Was zwischen uns an Beziehung aus einem einst »einfachen« Fachlektorat entstand, kann ich gar nicht in Worte fassen, und ich denke, ich muss es auch gar nicht – es ist doch viel lustiger, sich die Vorstellung anderer Menschen zu diesem Thema anzuhören, findet ihr nicht? ;-)

Eines weiß ich jedoch sicher: Wenn ich an euch alle denke und mal gerade nicht sowieso bei euch 'rumlungere, vermisse ich euch voll, nämlich, wohl. Ich finde, ihr seid was ganz Besonderes, alle miteinander, und seit Wochen mache ich mir schon Gedanken, wie ich mich einmal für all die schönen Stunden, den Spaß, den ich mit euch beim Philosophieren hatte und habe, die Sushi-Sessions, die ich mit Euch halten durfte, und von denen es hoffentlich noch hunderte geben wird, die neuen An- und Einsichten, die ich durch euch erlangt habe, die Cabrio-Touren, die wir miteinander unternommen haben, und für Vieles, Vieles mehr bedanken könnte. Und jetzt hätte ich hier gerade *zufällig* ein Buch übrig, das danach schreit, gewidmet zu werden und deswegen widme ich es euch, Ada, Zoe, Sarah, David, Ute und Ruprecht. Ihr seid meine Lieblingsentitäten und – Achtung, schlimmer Kalauer – ich danke meinem Schicksal sehr für diesen ganz persönlichen *LINQ to Entities*!

