

A woman with long brown hair is looking out a window with a smile. The background is a warm, orange-toned interior, possibly a train or bus, with a window frame visible. The overall mood is positive and hopeful.

Christian Bleske

# Windows 8 Apps entwickeln

- Apps für die neue Oberfläche von Windows 8
- Entwickeln mit HTML5, JavaScript, XAML und C#
- Von den Grundlagen der Programmiersprachen bis zur fertigen App

Christian Bleske

# Windows 8 Apps entwickeln

Christian Bleske

# Windows 8 Apps entwickeln

- Apps für die neue Oberfläche von Windows 8
- Entwickeln mit HTML5, JavaScript, XAML und C#
- Von den Grundlagen der Programmiersprachen bis zur fertigen App

## Bibliografische Information der Deutschen Bibliothek

Die Deutsche Bibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte Daten sind im Internet über <http://dnb.ddb.de> abrufbar.

Alle Angaben in diesem Buch wurden vom Autor mit größter Sorgfalt erarbeitet bzw. zusammengestellt und unter Einschaltung wirksamer Kontrollmaßnahmen reproduziert. Trotzdem sind Fehler nicht ganz auszuschließen. Der Verlag und der Autor sehen sich deshalb gezwungen, darauf hinzuweisen, dass sie weder eine Garantie noch die juristische Verantwortung oder irgendeine Haftung für Folgen, die auf fehlerhafte Angaben zurückgehen, übernehmen können. Für die Mitteilung etwaiger Fehler sind Verlag und Autor jederzeit dankbar. Internetadressen oder Versionsnummern stellen den bei Redaktionsschluss verfügbaren Informationsstand dar. Verlag und Autor übernehmen keinerlei Verantwortung oder Haftung für Veränderungen, die sich aus nicht von ihnen zu vertretenden Umständen ergeben. Evtl. beigefügte oder zum Download angebotene Dateien und Informationen dienen ausschließlich der nicht gewerblichen Nutzung. Eine gewerbliche Nutzung ist nur mit Zustimmung des Lizenzinhabers möglich.

© 2013 Franzis Verlag GmbH, 85540 Haar bei München

Alle Rechte vorbehalten, auch die der fotomechanischen Wiedergabe und der Speicherung in elektronischen Medien. Das Erstellen und Verbreiten von Kopien auf Papier, auf Datenträgern oder im Internet, insbesondere als PDF, ist nur mit ausdrücklicher Genehmigung des Verlags gestattet und wird widrigenfalls strafrechtlich verfolgt.

Die meisten Produktbezeichnungen von Hard- und Software sowie Firmennamen und Firmenlogos, die in diesem Werk genannt werden, sind in der Regel gleichzeitig auch eingetragene Warenzeichen und sollten als solche betrachtet werden. Der Verlag folgt bei den Produktbezeichnungen im Wesentlichen den Schreibweisen der Hersteller.

**Satz:** DTP-Satz A. Kugge, München  
**art & design:** [www.ideehoch2.de](http://www.ideehoch2.de)  
**Druck:** C.H. Beck, Nördlingen  
Printed in Germany

ISBN 978-3-645-60200-6

# Vorwort

Liebe Leserinnen und Leser,

»Window 8, nicht Windows 8«. So oder ähnlich lauteten viele Aussagen kurz nach der Vorstellung der ersten Vorabversion. Ich musste über diese Aussage zuerst schmunzeln, doch dann hat sie mich zum Nachdenken angeregt. So radikal wie mit Windows 8 hat Microsoft die GUI von Windows noch nie verändert. Jetzt könnte man ja sagen, dass nur der Startknopf gegen einen neuen Startbildschirm ausgetauscht wurde, aber das wird den auch unter der Oberfläche vorhandenen Änderungen sicherlich nicht gerecht. Selbst Windows 1.0 (wurde im Jahr 1985 veröffentlicht) verwendete bereits die Fenstertechnik, um Anwendungen anzuzeigen. Diese Technik ist mit der neuen GUI (auch bekannt unter: Metro, ModernUI, Windows 8 UI) passé. Nicht nur die Oberfläche hat sich verändert. Auch die Art und Weise, wie Anwendungen geschrieben werden, wurde angepasst. Egal ob eine Anwendung (App) auf das Dateisystem zugreifen oder mit anderen Anwendungen Daten austauschen oder die eingebaute Kamera nutzen soll. Natürlich hat man immer noch die Möglichkeit, Anwendungen nur für den alten/klassischen Desktop zu schreiben. Somit gibt es eigentlich keinen Grund, den neuen Pfad zu beschreiten. Irgendwann – vielleicht schon mit Windows 9 – wird ModernUI wieder abgeschafft, oder? Tja, man könnte das fast glauben, wenn da nicht Google und Apple mit Android und iOS wären und Microsoft im Mobile-Bereich das Leben schwer machen würden. Mobile Endgeräte, also Smartphones und Tablet-Computer, sind für Microsoft zur Bedrohung geworden. Warum? Versuchen Sie doch einmal, mit Windows 7 auf einem Tablet-Computer zu arbeiten – ich wünsche viel Vergnügen. Der klassische Windows-Desktop ist für die Touch-Bedienung mit dem Finger nicht ausgelegt. Die neue UI wird ja nicht nur unter Windows 8 verwendet. Auch Windows Phone und die Xbox nutzen das Kachel-Design. Im Prinzip hat sich der ganze Konzern danach ausgerichtet. Wer also glaubt, dass die neue Oberfläche nur eine Modeerscheinung ist und bald wieder verschwindet, könnte falsch liegen.

Eine leichtgewichtige Oberfläche für leichtgewichtige Programme (Apps) verlangt auch vom Entwickler ein Umdenken, und das nicht nur bei der Bedienung von Windows 8, sondern natürlich auch bei der Entwicklung. Stichworte: Vollbild, kein Multitasking. Mit diesem Buch möchte ich Ihnen die Entwicklung solcher Anwendungen näher bringen. Fragen, Anregungen und Kritik stehe ich aufgeschlossen gegenüber. Wie üblich – wenn Sie bereits eines meiner Bücher besitzen, kennen Sie das – finden Sie im Nachwort des Buches meine (immer noch aktuelle) E-Mail-Adresse zur Kontaktaufnahme.

Ach ja, bevor ich es vergesse. Dieses Buch ist in gewisser Weise ein kleines Experiment. Wie Sie vielleicht bereits wissen, kann man (native) Apps für Windows 8 nicht nur mit der Kombination XAML/C++, Visual Basic und C# entwickeln, sondern auch mit HTML/JavaScript. In diesem Buch finden Sie (fast) alle Beispiele sowohl in XAML/C# als auch in HTML/JavaScript. Teilen Sie mir mit, ob Sie dieses Vorgehen als nützlich oder

eher als lästig empfinden oder ob es Ihnen egal ist. Übrigens, das Buch wäre inhaltlich nicht umfangreicher geworden, wenn ich auf eine Sprache verzichtet hätte.

An dieser Stelle möchte ich mich bei meinem Lektor Anton Schmid vom Franzis Verlag bedanken. Herzlichen Dank für die gute Zusammenarbeit! Mein Dank gilt auch den vielen anderen Mitarbeiterinnen und Mitarbeitern des Franzis-Verlags, die ich nicht kenne. Zuletzt möchte ich mich bei meiner Familie für die Geduld und Unterstützung bedanken.

Christian Bleske im November 2012

# Inhaltsverzeichnis

<b>1</b>	<b>Grundlagen.....</b>	<b>13</b>
1.1	Windows 8 – Neues Betriebssystem für neue Endgeräte.....	13
1.2	Für wen ist dieses Buch gedacht? .....	14
1.3	Die Beispielanwendungen .....	14
1.4	Was zum Starten benötigt wird ... .....	15
1.4.1	Windows 8 – Die Versionen .....	15
1.4.2	Visual Studio 2012 .....	20
1.4.3	Das Dev-Center für Windows Store Apps .....	29
<b>2</b>	<b>Entwicklung für Windows 8 in der Übersicht .....</b>	<b>31</b>
2.1	Zwei Desktops, ein Betriebssystem .....	31
2.2	Die Windows-8-(API-)Architektur .....	33
2.2.1	Windows-Store-Anwendungen .....	35
2.2.2	Desktop-Anwendungen.....	36
2.2.3	Web-Anwendungen.....	37
<b>3</b>	<b>JavaScript und C#: Grundlagen der Programmierung.....</b>	<b>39</b>
3.1	Variablen und Zuweisungen .....	39
3.2	Datentypen .....	41
3.2.1	Zeichen und Zeichenketten.....	42
3.2.2	Ganze Zahlen und Fließkommazahlen.....	43
3.2.3	Wahrheitswerte.....	45
3.2.4	Aufzählungen .....	45
3.2.5	Konstanten .....	46
3.3	Operatoren.....	46
3.3.1	Boolesche Operatoren .....	47
3.3.2	Arithmetische Operatoren .....	48
3.4	Kontrollstrukturen .....	49
3.4.1	Die Fallunterscheidung .....	49
3.4.2	Mehrfachauswahl .....	50
3.5	Schleifen.....	51
3.5.1	Kopfgesteuerte Schleifen ( <i>while</i> -Schleife) .....	52
3.5.2	Fußgesteuerte Schleifen .....	52
3.5.3	Zählschleifen .....	53
3.5.4	<i>For(In)</i> -/ <i>For(each)</i> -Schleife .....	54
3.6	Felder (Arrays) .....	55
3.6.1	Mehrdimensionale Felder .....	57
3.7	Einfache und Referenztypen .....	58

3.8	Typumwandlung.....	58
3.9	Fehlerbehandlung mit Exceptions.....	59
4	Objektorientierte Programmierung mit JavaScript und C# .....	63
4.1	Grundlagen der Objektorientierung .....	63
4.1.1	Was sind Objekte? .....	63
4.1.2	Klassen & Objekte?.....	64
4.1.3	Sichtbarkeit steuern in C#.....	67
4.1.4	Sichtbarkeit in JavaScript steuern .....	68
4.2	Methoden und Funktionen.....	69
4.2.1	Parameter von Funktionen/Methoden.....	70
4.2.2	Eigenschaften (Get- und Set-Methoden/Funktionen).....	71
4.2.3	Das Schlüsselwort <i>this</i> .....	74
4.2.4	Konstruktoren in JavaScript und C# .....	76
4.3	Vererbung in C#/JavaScript .....	77
4.3.1	Überschreiben in JavaScript/C# .....	80
4.3.2	Überladen von Methoden in C#.....	82
4.4	Schnittstellen (Interface) in C# .....	83
4.5	Namespace (Namensraum).....	84
5	Die Markup-Sprachen HTML & XAML.....	87
5.1	HTML.....	87
5.1.1	HTML & DOM.....	88
5.1.2	HTML-Page in einer ModernUI-App .....	89
5.1.3	Elemente in HTML-Page einfügen .....	90
5.2	XAML.....	91
5.2.1	Page .....	92
5.2.2	Steuerlemente in XAML-Page einfügen.....	93
5.3	Application-Page.....	94
6	Visual Studio 2012 .....	97
6.1	Microsoft Visual Studio 2012 – Überblick.....	97
6.2	Die IDE .....	98
6.2.1	Projektvorlagen für Windows-8-Apps .....	101
6.2.2	Solution Explorer .....	106
6.2.3	Properties-Fenster.....	108
6.2.4	Der HTML-Editor .....	109
6.2.5	Der XAML-Designer .....	110
6.2.6	XAML-Editor .....	111
6.2.7	Toolbox-Register .....	112
6.2.8	Device-Register .....	113
6.2.9	Der C#-Quellcode-Editor.....	114
6.2.10	Der JavaScript-Quellcode-Editor .....	116

<b>7</b>	<b>Projekte mit Visual Studio erstellen/verwalten/debuggen .....</b>	<b>117</b>
7.1	<b>Anlage von Projekten .....</b>	<b>117</b>
7.1.1	Bestandteile eines JavaScript-Projekts .....	118
7.1.2	Bestandteile eines C#-Projekts .....	120
7.1.3	Einfügen von Controls in eine Page .....	122
7.1.4	Schreiben von Code .....	125
7.1.5	Eigenschaften eines Projekts .....	131
7.1.6	Ausführen von Projekten .....	133
7.2	<b>Projekte in Visual Studio ergänzen/erweitern .....</b>	<b>137</b>
7.2.1	Hinzufügen neuer Pages .....	137
7.2.2	Neue Page mit JavaScript .....	137
7.2.3	Neue Page mit C# .....	138
7.2.4	Weitere Elemente hinzufügen .....	139
7.3	<b>Debugging mit Visual Studio .....</b>	<b>141</b>
7.3.1	Breakpoints und Debugging der App .....	141
7.3.2	Fenster und Debug-Optionen .....	143
<b>8</b>	<b>Blend for Visual Studio .....</b>	<b>147</b>
8.1	<b>Überblick .....</b>	<b>147</b>
8.1.1	Neues Projekt erstellen .....	150
8.1.2	Projects-Panel .....	151
8.1.3	Assets-Panel .....	152
8.1.4	States-Panel .....	153
8.1.5	Device-Panel .....	153
8.1.6	Objects and Timeline-Panel .....	154
8.1.7	Das Tools-Panel .....	157
8.1.8	Properties-Panel (nur XAML/C#) .....	159
8.1.9	Resources-Panel (nur XAML/C#) .....	162
8.1.10	HTML Attributes Panel & CSS Properties Panel .....	162
8.1.11	Effektives Arbeiten mit Blend .....	163
<b>9</b>	<b>Windows Store styled Apps entwickeln .....</b>	<b>165</b>
9.1	<b>Was macht eine Windows-Store-App aus? .....</b>	<b>165</b>
9.1.1	Layout einer Windows-Store-App .....	166
9.2	<b>Die AppBar (AppBar) .....</b>	<b>169</b>
9.2.1	AppBar mit HTML/JavaScript .....	170
9.2.2	AppBar mit XAML/C# .....	172
9.3	<b>Charms .....</b>	<b>175</b>
9.3.1	Settings-Flyout mit HTML/JavaScript .....	176
9.3.2	Settings-Flyout mit XAML/C# .....	178
9.3.3	Was ist ein Contract / eine Extension? .....	181
9.4	<b>Das LiveTile einer App .....</b>	<b>181</b>
9.4.1	Anzeige von Texten im LiveTile .....	185
9.4.2	BadgeNotification anzeigen .....	188

<b>10</b>	<b>Entwicklung einer App .....</b>	<b>193</b>
10.1	Beispiel Zahlenratespiel.....	193
10.2	Layout der App .....	194
10.3	Aufteilung der Page.....	195
10.3.1	HTML – Das MS-Grid-Steuerelement .....	195
10.3.2	XAML – Das Grid-Steuerelement .....	197
10.3.3	Das StackPanel-Steuerelement.....	199
10.4	Design der Oberfläche.....	203
10.4.1	Standard-App-Themes Light & Dark .....	211
10.5	Programmierung der (Spiel-)Funktionen .....	213
10.6	Portrait- & Landscape-Ansicht.....	218
10.7	Test von Auflösungen .....	219
10.8	Lebenszyklus einer App.....	220
10.9	Dialoge.....	225
<b>11</b>	<b>Navigation und Datenübergabe in Apps .....</b>	<b>229</b>
11.1	Windows 8 – Navigation Patterns .....	229
11.2	Pages und Navigation – Realisierung .....	231
11.3	Datenübergabe von Page zu Page.....	237
<b>12</b>	<b>Daten- und Dateizugriff.....</b>	<b>243</b>
12.1	Ordner und Daten.....	243
12.2	Anwendungseinstellungen .....	244
12.3	Daten in Dateien.....	249
12.4	Verträge und Rechte .....	254
<b>13</b>	<b>Sensoren .....</b>	<b>259</b>
13.1	Location Service.....	259
13.1.1	Bing Maps in Windows-8-Apps.....	264
13.2	Die Kamera verwenden.....	272
13.3	Accelerometer (Beschleunigungssensor).....	278
<b>14</b>	<b>Lokalisierung.....</b>	<b>283</b>
14.1	Projekt vorbereiten.....	283
14.2	Ressourcendateien hinzufügen .....	284
14.3	Texte auslagern.....	285
<b>15</b>	<b>Client &amp; Server mit WCF .....</b>	<b>291</b>
15.1	Grundlagen .....	291
15.2	Der WCF-Server/-Service .....	293
15.3	Der WCF-Client (Windows-8-App).....	297
<b>16</b>	<b>C#-Bibliotheken in JavaScript .....</b>	<b>303</b>
16.1	Windows Runtime Component.....	303
16.2	Klassenbibliothek in C# anlegen .....	305

16.3	Klassenbibliothek in JavaScript einbinden.....	305
16.4	GUI zum Testen .....	306
16.5	Aufruf der Bibliotheksfunktionen.....	307
16.6	Portable Class Library.....	309
16.7	Anlegen einer PCL .....	309
17	Testen der App.....	311
17.1	Testen mit Unit-Tests.....	311
17.2	Unit-Tests mit Visual Studio durchführen .....	312
17.3	Erweiterung mit Asserts.....	316
17.3.1	Vergleichstests .....	316
17.3.2	Statustests .....	317
17.3.3	Utility-Methoden .....	318
17.4	Überwachung mit Debug & Trace.....	318
18	App veröffentlichen .....	321
18.1	Registrierung als Developer im Windows Store.....	321
18.2	Vorbereitung der App .....	324
18.3	Test mit dem Windows App Certification Kit.....	325
18.4	App erstellen und verteilen.....	327
18.4.1	Windows 8 für Sideloadung konfigurieren.....	329
18.5	App in den Windows Store übertragen.....	332
18.5.1	App Name.....	335
18.5.2	Selling details.....	335
18.5.3	Advanced features.....	336
18.5.4	Ratings .....	337
18.5.5	Cryptography .....	338
18.5.6	Packages .....	338
18.5.7	Description .....	339
18.5.8	Notes to testers .....	339
A	Anhang – Controls .....	341
	A (Hyperlink) (HTML).....	341
	AppBar (HTML/XAML) .....	341
	Audio (HTML).....	341
	Border (XAML) .....	341
	Button (HTML/XAML) .....	342
	Canvas (XAML) .....	343
	CaptureElement (XAML).....	343
	CheckBox (HTML/XAML) .....	344
	ComboBox/DropDownBox (HTML/XAML) .....	345
	ContentControl (XAML) .....	346
	ContentPresenter (XAML).....	346
	DatePicker (HTML) .....	346
	Email Input Box (HTML) .....	347

Ellipse (XAML) .....	347
FlipView (HTML/XAML) .....	348
Flyout (HTML) .....	348
MS Grid/Grid (HTML/XAML) .....	351
GridView (XAML).....	351
HTML (HTML) .....	351
HyperlinkButton (XAML) .....	352
Image (XAML) .....	352
ListBox (XAML) .....	353
ListView (HTML).....	354
MediaElement (XAML) .....	355
Menu & MenuCommand (HTML) .....	357
PasswordBox (XAML).....	358
ProgressBar (HTML/XAML) .....	359
ProgressRing (HTML/XAML) .....	361
RadioButton (HTML/XAML) .....	362
ScrollViewer (XAML) .....	363
Slider/Range (HTML/XAML) .....	364
TextArea (HTML) .....	364
TextBlock (XAML).....	364
TextBox (XAML) .....	365
<b>Nachwort .....</b>	<b>367</b>
<b>Stichwortverzeichnis .....</b>	<b>369</b>

# 1 Grundlagen

Die Entwicklung von Anwendungen – pardon, man bezeichnet diese ja mittlerweile als Apps – für Windows 8 ist Ihr Ziel? Sie wissen aber noch nicht genau, ob Sie zur Programmierung HTML und JavaScript oder aber XAML und C# verwenden möchten? In diesem Fall sollten Sie weiterlesen. Beide Wege werden Ihnen in diesem Buch vorgestellt. Das hat den Vorteil, dass man ggf. auch Anforderungen umsetzen kann, die mit einer bestimmten Sprache nicht umsetzbar sind. In diesem Kapitel wird dafür das Fundament gegossen. So werden die einzelnen Versionen von Windows 8 und Visual Studio kurz vorgestellt, die Installation (von Visual Studio) besprochen, und natürlich darf auch ein »Hallo Welt«-Beispiel mit HTML/JavaScript und XAML/C# nicht fehlen.

## 1.1 Windows 8 – Neues Betriebssystem für neue Endgeräte

Microsoft bringt mit Windows 8 eine neue Version seines erfolgreichen Betriebssystems auf den Markt. Amüsant ist die Tatsache, dass auch der »neue« Kachel-Teil des Betriebssystems, auch als ModernUI bezeichnet, unter dem Namen »Windows« auftritt. Gerade diese neue Oberfläche, beziehungsweise die Anwendungen, welche unter dieser neuen Oberfläche ausgeführt werden, laufen fast ausschließlich (es gibt eine Ausnahme) im Vollbildmodus. Hier ist also wenig »ursprüngliches« Windows zu finden. Vielleicht wäre der Name Window passender? Allerdings ist dieser radikale Umbruch auch notwendig gewesen, da die Konkurrenz in Form von Android und iOS Microsoft zurzeit das Leben sehr schwer macht.

Gerade was den Markt der Tablet-Computer betrifft, ist Windows in der alten Form (egal ob XP, Vista oder 7) einfach nicht mehr zeitgemäß. Ein Betriebssystem dauerhaft mit einem Stift zu bedienen, kann heute auch nicht die Lösung sein, und die Bedienung der Windows-7-GUI von Hand ist eine Qual. Um also der neuen Form der Bedienung gerecht zu werden, wurde eine Oberfläche geschaffen, die sich sehr gut von Hand bedienen lässt. Die Vorlage für die neue Windows-GUI ist sicherlich Windows Phone gewesen.

Microsoft versucht nun also den Spagat zwischen Desktop-, den es ja auch noch gibt, und Touch-GUI. Ob das Konzept wirklich aufgeht, wird die Zeit zeigen. Noch eine Änderung gibt es mit Windows 8. Erstmals hat Microsoft ein Betriebssystem für alle Gerätetypen. Denn der Kern von Windows 8 wird auch im neuen Windows Phone 8 verwendet. Das bedeutet, angefangen vom Desktop-Computer über Tablets bis hin zu Smartphones gibt es dann eine gemeinsame Basis. Auch für Entwickler hat dieses Modell einen entscheidenden Vorteil: Egal für welchen Gerätetyp eine App entwickelt wird, die verwendete Technik (Programmiersprache, Entwicklungsumgebung) ist identisch. In fast allen Fällen kann man auf die Duos HTML/JavaScript (bei Windows Phone leider nicht)

oder XAML/C# zur Entwicklung zurückgreifen. Natürlich ist Visual Studio in der neuen Version auch immer mit von der Partie.

## 1.2 Für wen ist dieses Buch gedacht?

Als Autor möchte man am liebsten sagen, dass das eigene Buch allen Zielgruppen gerecht wird. Das ist natürlich bei keinem Buch der Fall. Aus diesem Grund beschreibe ich mit den folgenden Zeilen, welche Vorkenntnisse notwendig sind, damit das Lesen des Buches auch Spaß macht und kein Frust aufkommt.

Wenn Sie noch nie programmiert haben, dann können Sie den Einstieg in die Welt der Softwareentwicklung mit diesem Buch durchaus wagen. Warum? Mit JavaScript wird in diesem Buch eine Sprache zur Programmierung vorgestellt, die wirklich leicht zu erlernen ist. Aber auch wenn Sie sich für eine andere Sprache interessieren, vorzugsweise C#, können Sie weiterlesen, denn die Beispiele werden auch in C# vorgestellt.

Sollten Sie mit Begriffen wie Klasse, Objekt, Instanz, Eigenschaft oder Methode noch nicht viel anfangen können, dann lege ich Ihnen die Kapitel 3 und 4 ans Herz. Dort wird Basiswissen zu diesen Themen vermittelt. Zwar werden die gängigsten Sprachkonstrukte vorgestellt, Einsteiger in die Programmierung sollten aber noch zusätzliches Material heranziehen, um sich zurechtzufinden.

Außerdem gehe ich davon aus, dass Ihnen Grundbegriffe bezogen auf grafische Benutzeroberflächen bekannt sind. Das bedeutet, Sie sollten wissen, was ein Button, eine Text-Box oder eine ComboBox sind. Eine Übersicht mit einigen Codebeispielen zur Verwendung von Steuerelementen (Controls) finden Sie im Anhang des Buches.

Eine weitere Voraussetzung gibt es: HTML sollten Sie schon einmal gesehen bzw. damit gearbeitet haben. Zwar gehe ich an der einen oder anderen Stelle auf Besonderheiten ein, aber eine grundlegende Einführung zum Thema HTML finden Sie in diesem Buch nicht. Hier sollten Sie auf entsprechende Quellen zurückgreifen. Auch das Thema CSS wird in diesem Buch nicht grundlegend behandelt. Wenn Sie sich also dazu entscheiden, mit HTML und JavaScript zu arbeiten, um Windows-8-Apps zu entwickeln, dann sollten Sie auch, was das Thema CSS angeht, zusätzliche Informationen einholen.

### **Tipp**

Sie kennen sich mit HTML nicht aus? Eine gute Einführung zu diesem Thema und auch eine entsprechende Referenz finden Sie unter <http://de.selfhtml.org>. Auf dieser Website finden Sie auch eine Referenz zu JavaScript und Informationen zum Thema CSS.

## 1.3 Die Beispielanwendungen

In diesem Buch finden Sie zu einigen Themen Beispiele. Ich habe mich gegen ein durchgängiges Beispiel entschieden. Einfach weil mir keine App einfällt, in welcher alle Themen,

die vorgestellt werden sollen, vorkommen könnten. Sie finden also auf das Thema (z. B. Positionsbestimmung mit GPS) spezialisierte Beispiele. Alle Beispiele in diesem Buch wurden von mir auf ihre Lauffähigkeit hin getestet. Es sollten also keine Probleme bei der Ausführung dieser Beispiele auf Ihrem PC auftreten. Sollte etwas doch einmal nicht klappen, so können Sie mich über meine E-Mail-Adresse (cb.2000@hotmail.de) erreichen.

Natürlich müssen Sie den Quellcode der Beispiele nicht mühsam abtippen, wenn Sie nicht möchten. Der Code der Apps ist unter der URL [www.buch.cd](http://www.buch.cd) verfügbar. Sie sollten ihn auf jeden Fall herunterladen, da das Nachvollziehen der Beispiele manchmal hilfreich ist. Gerade beim Abtippen schleicht sich schnell einmal ein Fehler ein.

Bewusst sind die Beispiel-Projekte nicht allzu komplex gehalten, damit Ihnen die Einarbeitung leichter fällt.

#### Hinweis

Um den Quellcode herunterzuladen, benötigen Sie die ISBN-Nummer des Buches!

## 1.4 Was zum Starten benötigt wird ...

Neben etwas Geduld benötigen Sie natürlich auch noch etwas Hard- und Software. Der PC, der zur Entwicklung verwendet wird, sollte über eine Dual-Core-CPU mit mindestens 2 Gigahertz Taktfrequenz verfügen. Zwei Gigabyte Hauptspeicher sind unbedingt zu empfehlen – vier Gigabyte sind besser. Die Entwicklung von Windows-8-Apps kann nur unter Windows 8 erfolgen. Das bedeutet, Sie benötigen zwangsweise eine Kopie des neuen Betriebssystems. Es gibt allerdings Möglichkeiten, fürs erste auf den Kauf zu verzichten: Laden Sie sich einfach eine Testversion von Windows 8 Enterprise herunter und installieren Sie diese in einer virtuellen Maschine. Die Testversion kann 90 Tage lang genutzt werden. Im Abschnitt 1.4.1.1 finden Sie einige Tipps dazu.

Zur Installation von Windows 8 benötigen Sie mindestens 10–20 Gigabyte freien Speicherplatz auf der Festplatte. Gehen Sie davon aus, dass die Installation aller Entwicklungswerkzeuge auf der Festplatte einen Umfang von ca. 5–10 Gigabyte in Anspruch nimmt.

### 1.4.1 Windows 8 – Die Versionen

Wer kennt nicht das Versionschaos, welches bei Vista oder Windows 7 vorherrscht? Neben Home und Home Premium gibt es noch eine Starter-Edition. Professional gibt es auch, und Enterprise und Ultimate! Alles klar? In dieser Aufzählung fehlen natürlich noch die unterschiedlichen Server-Versionen.

Microsoft hat wohl selber erkannt, dass diese vielen Varianten von Windows nicht gerade die Übersichtlichkeit fördern. Mit der Einführung von Windows 8 gibt es »nur« noch drei Versionen für den Desktop-PC bzw. Tablets mit Intel- oder AMD-Prozessoren. Dies sind: Windows 8 (Standard), Windows 8 Professional und Windows 8 Enterprise. Die

Enterprise-Version von Windows 8 wird allerdings nicht im freien Handel erhältlich sein. Sie ist einzig für Großkunden verfügbar.

Eine vierte Version von Windows 8 ist auch nicht (direkt) auf dem Markt erhältlich. Es handelt sich hierbei um die Variante, welche für Endgeräte mit ARM-Prozessor bestimmt ist. Sie trägt auch nicht die 8 im Namen, sondern heißt Windows RT. Die Server-Version von Windows 8 (Windows Server 2012) wird hier nicht näher betrachtet. Zwar läuft auch auf dem Windows-Server ModernUI, aber die Entwicklung von Apps wird wohl eher auf einer der Desktop-Versionen stattfinden.

Zur Entwicklung von Apps ist es eigentlich unerheblich, welche Version man verwendet. Alle Varianten enthalten den neuen Desktop und egal, ob Windows 8 Standard, Professional oder Enterprise verwendet wird, der »alte« Windows-Desktop ist auch immer enthalten. Dieser Punkt ist wichtig, da Visual Studio diesen verwendet. Auch der Windows Store, über welchen die Apps später vertrieben werden können, ist Bestandteil aller Versionen bzw. lässt sich aufrufen. Alle anderen Funktionen werden für die Entwicklung von Apps nicht benötigt, spielen bei der Auswahl der Version also keine Rolle.

### **32 oder 64 Bit?**

Diese Frage ist wichtig. Bevor man aber pauschal die 32-Bit-Version verschmäht, sollte man die eine oder auch andere Überlegung einfließen lassen. Sofern der Entwicklungsrechner direkt mit Windows 8 aufgesetzt wird, macht es keinen Sinn, nicht zur 64-Bit-Variante zu greifen. Gerade wenn man mehr als 4 Gigabyte Hauptspeicher zur Verfügung hat, ist die 32-Bit-Version – Verschwendung. Bei der 64-Bit-Version gilt es ansonsten noch zu beachten, dass hier evtl. Probleme mit Hardwaretreibern auftreten können, weil diese in einer 64-Bit-Version nicht verfügbar sind.

Denkt man aber über eine virtuelle Maschine nach, die auf dem vorhandenen Betriebssystem (z. B. Vista oder Windows 7) aufgesetzt wird, so kann die 32-Bit-Version doch eine Überlegung wert sein, da sie weniger Hauptspeicher benötigt (mind. 1 Gigabyte im Vergleich zu 2 Gigabyte mit der 64-Bit-Version).

### **Windows in der Virtual Box**

Windows 8 virtuell betreiben und dann darauf noch entwickeln – geht das denn überhaupt? Die kurze und knappe Antwort lautet: Ja, sehr gut sogar. Allerdings setzt diese Vorgehensweise einen leistungsfähigen PC voraus. 8 Gigabyte RAM sollten es schon sein, und ein schneller Dual- oder Quad-Core-Prozessor ist für diesen Weg zwingend erforderlich. Mit einer solchen Ausstattung lässt sich die virtuelle Maschine fast so nutzen, als wäre das darin installierte Betriebssystem direkt auf der Hardware aufgesetzt worden. Man sollte aber darauf achten, dass die CPU eine entsprechende Virtualisierungstechnik unterstützt. Intel macht dies mit dem Kürzel VT-x kenntlich, bei AMD wird es durch den Begriff AMD-V deutlich gemacht.

Neben der Hardware benötigt man noch etwas Software, nämlich ein Programm, welches die Virtualisierungsschicht der Hardware für die Gast-Betriebssysteme bereitstellt. Hier kann man entweder auf kostenpflichtige Software wie z. B. VMWare Workstation zurückgreifen, oder aber man verwendet Virtual Box von Oracle (früher Sun). Es

handelt sich hierbei um eine vollwertige Lösung zur Virtualisierung. Im folgenden Abschnitt wird kurz erläutert, wie Windows 8 unter Virtual Box installiert werden kann.

### Hinweis

Der Download von Virtual Box erfolgt unter [www.virtualbox.org/wiki/Downloads](http://www.virtualbox.org/wiki/Downloads).

Der Download der Installationsdatei erfolgt von der Homepage [virtualbox.org](http://virtualbox.org) und ist ca. 80 Megabyte groß. Die Installation beginnt mit den obligatorischen Installationshinweisen und der Auswahl der zu installierenden Komponenten. Hier können Sie die Voreinstellungen beibehalten. Während der Installation werden Sie darauf hingewiesen, dass die Netzwerkverbindung kurzzeitig beendet wird. Diese Unterbrechung ist notwendig, um die benötigten Netzwerkkomponenten installieren zu können. Da im Anschluss alles auch wieder automatisch gestartet wird, kann man ruhig zustimmen. Die eigentliche Unterbrechung ist wirklich nur minimal (ca. 1 Sek.). Im Laufe der Installation werden Sie noch gebeten, der Installation eines Gerätetreibers von Oracle zuzustimmen.

Nach der Installation wird Virtual Box automatisch gestartet.



**Bild 1.1:** Virtual Box nach dem ersten Start

Zu diesem Zeitpunkt hat man nur die Möglichkeit, eine neue virtuelle Maschine über den Punkt *Neu* anzulegen. Aktiviert man diesen, wird ein Assistent zur Anlage gestartet.

Im ersten Schritt der Konfiguration muss man den Namen der virtuellen Maschine sowie die Version des Betriebssystems festlegen. Ein Tipp: Verwendet man als Namen Windows 8, so wird automatisch auch als Version Windows 8 selektiert. Anschließend kann der *Weiter*-Button betätigt werden. Im folgenden Dialog wird die Speicherkonfiguration

der virtuellen Maschine festgelegt. Voreingestellt sind 1024 Megabyte. Da auch Visual Studio installiert werden soll, kann es nicht schaden, den Speicher auf 2048 Megabyte zu erhöhen. Ein Hinweis: Die Konfiguration kann auch nachträglich geändert werden.

Als Nächstes muss eine virtuelle Festplatte angelegt werden. Im Dialog hat man die Möglichkeit, eine neue Festplatte anzulegen (Voreinstellung) oder eine vorhandene zu verwenden. Sie sollten eine neue anlegen. Die vorgeschlagene Größe von 25 Gigabyte genügt für Windows 8 und Visual Studio 2012. Einen Schritt weiter wird der Typ der virtuellen Festplatte festgelegt. Hier stehen unterschiedliche Dateitypen zur Auswahl. Man sollte auch hier die Voreinstellung beibehalten. Die anderen Dateitypen sind nur dann interessant, wenn auch noch mit anderen Virtualisierungssystemen gearbeitet wird. Wichtig ist die darauf folgende Entscheidung *dynamisch alloziert* oder *feste Größe*. Auch hier gilt: Voreinstellung beibehalten!

Eine *dynamisch allozierte* Festplattendatei belegt nicht sofort den kompletten Speicherplatz auf der realen Festplatte. Sie wird stückweise erweitert. Wird hingegen die Option *feste Größe* verwendet, sind direkt die voreingestellten 25 Gigabyte belegt. Warum die Unterscheidung? Nun, eine virtuelle Festplatte mit fester Größe ist etwas performanter. Die letzte Einstellung bzgl. der virtuellen Festplatte betrifft die Größe selbst. Wie gesagt, belassen Sie es (sofern möglich) bei der Voreinstellung. Im letzten Schritt der Anlage wird noch mal eine Zusammenfassung angezeigt. Betätigt man anschließend den Button *Erzeugen*, so wird die virtuelle Maschine angelegt. Im Programmfenster von Virtual Box ist nun ein entsprechender Eintrag zu finden.

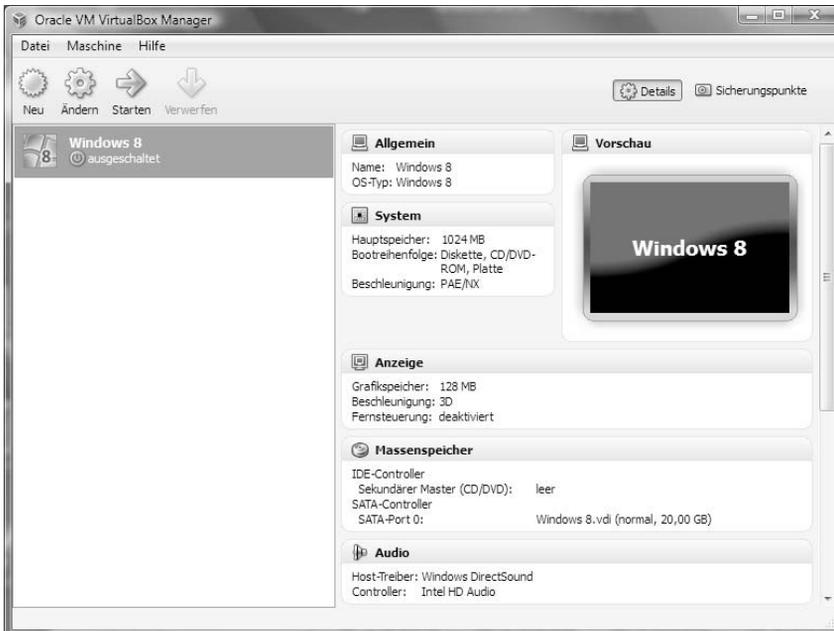


Bild 1.2: Eine virtuelle Windows-8-Maschine

Als Nächstes folgt die Installation von Windows 8. Hierzu benötigen Sie entweder eine DVD mit Windows 8 oder (besser, weil es schneller geht) ein ISO-Abbild dieser DVD.

Im Web finden Sie zahlreiche Programme, mit welchen sich ein solches ISO-Abbild erstellen lässt (z. B. UltraISO). Sie können auch einmal prüfen, ob Microsoft ISO-Abbilder von Windows 8 bereitstellt. Vielleicht sind ja die ISO-Abbilder der Beta noch verfügbar.

### Hinweis

Unter folgendem Link erhält man eine kostenlose Testversion von Windows 8 Enterprise, welche man für 90 Tage benutzen kann: <http://msdn.microsoft.com/en-us/evalcenter/jj554510.aspx>. Einzige Voraussetzung ist ein Microsoft-Account (früher Hotmail/Windows Live ID). Diesen Account kann man gleich mit anlegen.

Wenn man ein ISO-Abbild hat, dann muss dieses mit der virtuellen Maschine verknüpft werden. Hierzu selektiert man die Maschine in Virtual Box und betätigt die Schaltfläche *Ändern* im Hauptmenü. Anschließend wird ein Dialog *Windows 8 – Ändern* angezeigt.

### Hinweis

Über den Punkt *Ändern* kann die komplette Konfiguration der virtuellen Maschine verändert werden. Soll also beispielsweise der Hauptspeicher angepasst werden, so ist man hier am richtigen Ort.

Im linken Bereich finden Sie eine Liste der Konfigurationspunkte – siehe Bild 1.3.

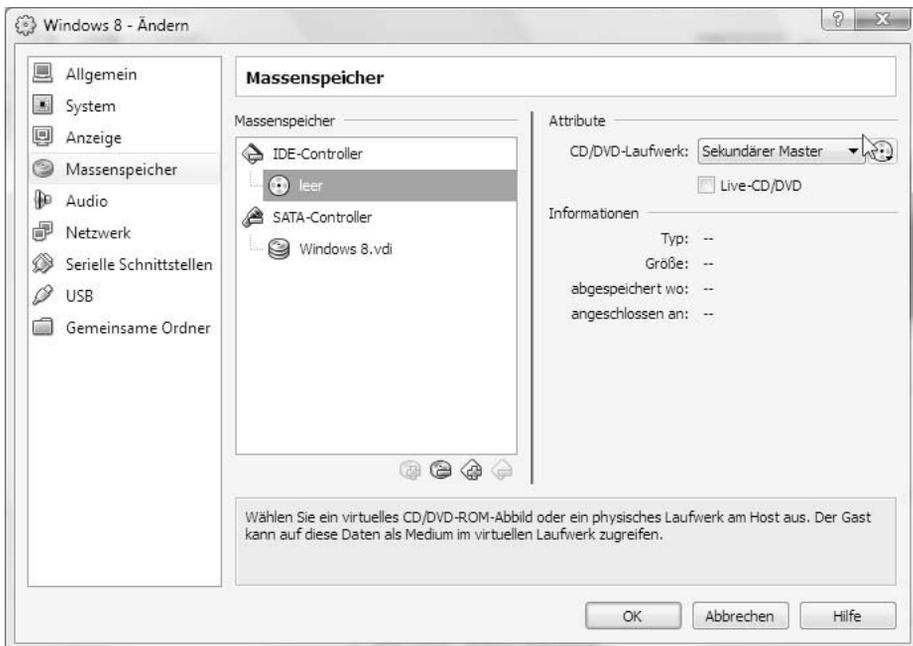


Bild 1.3: Anbindung eines Windows-8-ISO-Abbildes

Hier selektiert man den Eintrag *Massenspeicher*, wählt unterhalb des Eintrags *IDE-Controller* das CD-Symbol (leer) aus und klickt anschließend auf das CD-Symbol am rechten Rand unterhalb des Punkts *Attribute*. Es erscheint ein Datei-öffnen-Dialog, in welchem das ISO-Abbild ausgewählt werden muss. Der Dialog kann dann geschlossen werden. Nun befindet man sich wieder im Hauptmenü von Virtual Box. Zum Start der Installation muss jetzt nur noch der Punkt *Starten* (grüner Pfeil) aktiviert werden. Die virtuelle Maschine wird nun gebootet und die Installation von Windows 8 startet automatisch. Die Installation von Windows 8 selbst ist wie bereits die von Windows 7 nicht kompliziert. Nach der Installation und dem ersten Start bleibt nur noch ein Punkt übrig. Die Gast-Erweiterungen sollten innerhalb der virtuellen Maschine installiert werden. Anschließend kann dann ein gemeinsames Verzeichnis mit dem Wirtssystem eingerichtet und auch USB-Geräte (z. B. Sticks) innerhalb der virtuellen Maschine verwendet werden. Zur Installation muss die virtuelle Maschine im Vordergrund aktiv sein. Im Hauptmenü von Virtual Box wird dann ein zusätzlicher Menüpunkt *Geräte* angezeigt. In diesem Menü findet man den Punkt *Gasterweiterung installieren...* Anschließend wird automatisch im Gastsystem ein ISO-Abbild mit dieser Erweiterung angefügt und die Software installiert.

#### **Hinweis**

Damit alles korrekt funktioniert (z. B. gemeinsam genutztes Verzeichnis von Host und Gast), muss man zusätzlich noch das VirtualBox Extension Pack herunterladen. Der Download ist auf derselben Seite wie Virtual Box verfügbar.

Die Virtualisierung ist (nicht nur) für die Entwicklung ein Segen. Leistungsfähige Hardware vorausgesetzt, kann man problemlos Dinge testen, und wenn es einmal schief geht, einfach die virtuelle Maschine zurücksetzen. Es geht nichts mehr verloren. Bei der Entwicklung von Apps für Smartphones ist die Verwendung einer virtuellen Maschine bereits Standard (egal ob Windows Phone, Android oder iPhone).

### **1.4.2 Visual Studio 2012**

Das Werkzeug zur Entwicklung von Windows-8-Apps ist Visual Studio in der neuesten Version. Zwar können Teile der Entwicklung auch mit Expression Blend erfolgen (hauptsächlich im GUI-Bereich), aber Visual Studio ist der Dreh- und Angelpunkt für den kompletten Entwicklungsweg. Es ist sogar möglich, auf Expression Blend zu verzichten und nur Visual Studio zu verwenden. Da Expression Blend hauptsächlich für Designer entwickelt wurde, wird dieser Weg wohl häufiger vorkommen, zumal Expression Blend nicht jedermanns Sache ist.

#### **Die Versionen**

Microsoft hat sich sehr bemüht, das Versionschaos mit Windows 8 einzudampfen. Bei der neuen Version von Visual Studio hat man sich daran leider kein Beispiel genommen. Das Chaos herrscht immer noch vor. So gibt es auch bei Visual Studio 2012 wieder zahlreiche Versionen, was die beliebte Frage aufwirft, welche Version es denn sein soll oder muss? Im Angebot gibt es Visual Studio 2012 Ultimate, Visual Studio 2012 Premium,

Visual Studio 2012 Professional, Visual Studio 2012 Test Professional, Visual Studio 2012 Team Foundation Server und Visual Studio 2012 Express (die einzige Variante, die nichts kostet). Um es kurz und simpel zu machen: Wenn Sie allein arbeiten, vergessen Sie alle Varianten mit Ausnahme von Professional und Express. Die anderen Versionen sind nur dann interessant, wenn im Team gearbeitet wird. Jetzt stellt sich nur noch die Frage, ob Sie Geld ausgeben möchten (wer möchte das schon?) oder ob Sie auch mit einer im Funktionsumfang eingeschränkten, dafür aber kostenlosen Version leben können.

Hauptsächlich unterscheiden sich Professional und Express dadurch, dass mit der Professional-Version zusätzliche Funktionen (z. B. Remote Debugging) und Tools sowie kostenfreie Lizenzen anderer Microsoft-Produkte (z. B. SQL Server) zur Entwicklung ausgeliefert werden. Auch verfügt die Professional-Version über mehr Vorlagen für die unterschiedlichsten Projekttypen. So können Sie neben Apps für die moderne Oberfläche auch Web-Anwendungen (hierfür gibt es aber ebenfalls eine kostenfreie Lösung) entwickeln oder Anwendungen, welche Silverlight oder auch die Windows Presentation Foundation als Oberfläche verwenden. Auch Erweiterungen (Add-ins) für Microsoft Office können nur mit der Professional-Edition geschrieben werden.

Allerdings benötigt man diese zusätzlichen Funktionen für die Entwicklung von Windows-8-Apps nicht. Das bedeutet im Umkehrschluss, dass alles, was man für die Entwicklung von Windows-8-Apps für die neue Oberfläche benötigt, in der Express-Version enthalten ist. Im weiteren Verlauf wird (mit der einen oder anderen Ausnahme, auf welche aber gesondert hingewiesen wird) nur die Express-Version besprochen werden. Sie können aber natürlich auch mit der Professional-Edition die Beispiele nachvollziehen.

#### Hinweis

Aus den beschriebenen Gründen wird in den Abbildungen und in Erläuterungen nur auf die englische Sprachfassung von Visual Studio Express 2012 für Windows 8 und der zusätzlichen Tools eingegangen.

### Die Installation von Visual Studio 2012 (Offline / Online)

Bevor Sie Visual Studio 2012 Express installieren können, müssen Sie es von der entsprechenden Microsoft-Webseite <http://www.microsoft.com/visualstudio/deu/downloads> herunterladen. Es gibt zwei Wege zur Installation. Der erste führt über den Download eines sogenannten Web-Installers. Wählen Sie diesen Weg, so wird zuerst nur die ein paar Megabyte große Setup-Datei aus dem Internet heruntergeladen. Diese führen Sie dann aus. Anschließend prüft der Installer, welche weiteren Komponenten für die Installation benötigt werden. Diese werden automatisch heruntergeladen und die Installation startet. Dieser Weg empfiehlt sich unter zwei Gesichtspunkten: Der erste ist, dass bereits einige Komponenten auf dem Rechner installiert sind, beispielsweise durch eine vorhandene Installation von Visual Studio 2011 Express for Web. Der notwendige Zeitraum zur Installation verkürzt sich so zum Teil drastisch. Der Nachteil an dieser Variante ist, dass auf jedem Computer, auf dem die IDE installiert werden soll, der Download erneut gestartet werden muss.

Wenn Sie also vorhaben, Visual Studio auf mehr als nur einem Computer zu installieren, dann empfiehlt es sich, das komplette ISO-Image herunterzuladen. Das ISO-Image können Sie entweder auf eine DVD brennen (z. B. mit dem kostenlosen CDBurnerXP) oder aber direkt einbinden (z. B. mit Demon-Tools Lite – ebenfalls kostenlos).

### Welche Sprachversion?

Eigentlich spielt es keine Rolle mehr, ob Sie die deutsche Variante oder eine andere Sprache bevorzugen. Visual Studio gibt es neben Englisch zurzeit auch in Deutsch, Französisch, Italienisch oder Spanisch. Ein Vorteil der englischen Version ist, dass Updates oder Patches etwas früher zur Verfügung stehen als für die lokalisierten Versionen. Auch ist es manchmal so, dass die eingedeutschte Version unter merkwürdigen Fehlern leidet. Insgesamt ist es besser, Visual Studio in der Sprache Englisch zu installieren. Ein weiterer Punkt sind die oft seltsamen Texte bei eingedeutschten Fehlermeldungen. Googeln Sie einmal nach solchen Fehlern ...

Nach dem Aufruf der Setup-Datei beginnt die Installation. Es wird zuerst der obligatorische Dialog für die Zustimmung zu den Lizenzbedingungen angezeigt. Im nächsten Schritt kann man sich entscheiden, ob direkt installiert oder das Setup vorher noch konfiguriert werden soll. Die Möglichkeiten zur Anpassung sind allerdings stark beschränkt. Lediglich den Ordner, in welchem Visual Studio installiert wird, kann man sich aussuchen.



**Bild 1.4:** Installation von Visual Studio 2012 Express

Je nach Umfang der zu installierenden Komponenten kann die Installation schon einige Minuten in Anspruch nehmen. Nach der Installation finden Sie einige neue LiveTiles im ModernUI-Desktop von Windows 8. Neben Visual Studio 2012 Express finden sich noch Einträge für: Blend for Visual Studio (Design-Werkzeug zum visuellen Gestalten von

HTML- und XAML-Seiten), der Windows Performance Analyzer (ein Programm zur Analyse des Laufzeitverhaltens einer Anwendung) und der Windows App Cert Kit Analyzer (eine Anwendung, welche bei der Zertifizierung einer App hilft).



**Bild 1.5:** Der neue Desktop mit installiertem Visual Studio 2012 Express

Nach dem ersten Start von Visual Studio 2012 Express werden Sie aufgefordert, eine Developer-Lizenz für Windows-8-Apps zu erstellen. Diese Lizenz wird benötigt, um Apps zu Testzwecken unter Windows 8 installieren zu können. Ohne eine entsprechende Lizenz kann Visual Studio zur Erstellung von Apps nicht verwendet werden. Allerdings kostet diese Lizenz kein Geld. Einzig ein Microsoft-Account (früher Windows Live ID / Hotmail) wird benötigt.

Sobald die *I Agree*-Schaltfläche betätigt wurde, wird der *Windows Security*-Dialog angezeigt. In diesem müssen die Anmeldedaten des Hotmail-Kontos (E-Mail-Adresse und Passwort) eingetragen werden. Hat man zu diesem Zeitpunkt noch keinen Account, so kann man über einen Link im Dialog ein entsprechendes Konto anlegen. Im folgenden Schritt muss die *Sign in*-Schaltfläche betätigt werden. Wenn alles geklappt hat, dann informiert der letzte Dialog darüber, dass eine Lizenz angelegt wurde.



**Bild 1.6:** Anlegen einer Developer-Lizenz für Windows 8

### Hinweis

Es kann vorkommen, dass man keine Entwicklerlizenz erhält. Eine mögliche Fehlermeldung ist Error Code 0x080004005. Wie auch immer, das Ergebnis ist dasselbe, eine Entwicklerlizenz wird nicht erteilt. Betätigen Sie in einem solchen Fall einfach mehrmals den *Retry*-(Wiederholen-)Button. Hin und wieder klappt das. Auch ein Reboot der Windows-8-Maschine hat beim Autor schon zum Erfolg geführt.

### Schnellstart mit Hallo Welt

In diesem Abschnitt soll allen Neulingen, die noch nie mit Visual Studio gearbeitet haben, die Möglichkeit gegeben werden, einen ersten Blick auf die Entwicklungsumgebung zu werfen. Keine Sorge, egal ob Sie mit JavaScript arbeiten möchten oder C# bevorzugen, beide Seiten kommen zum Zuge.

### Hinweis

Den folgenden Abschnitt können Sie überspringen, wenn Sie sich bereits mit Visual Studio auskennen.

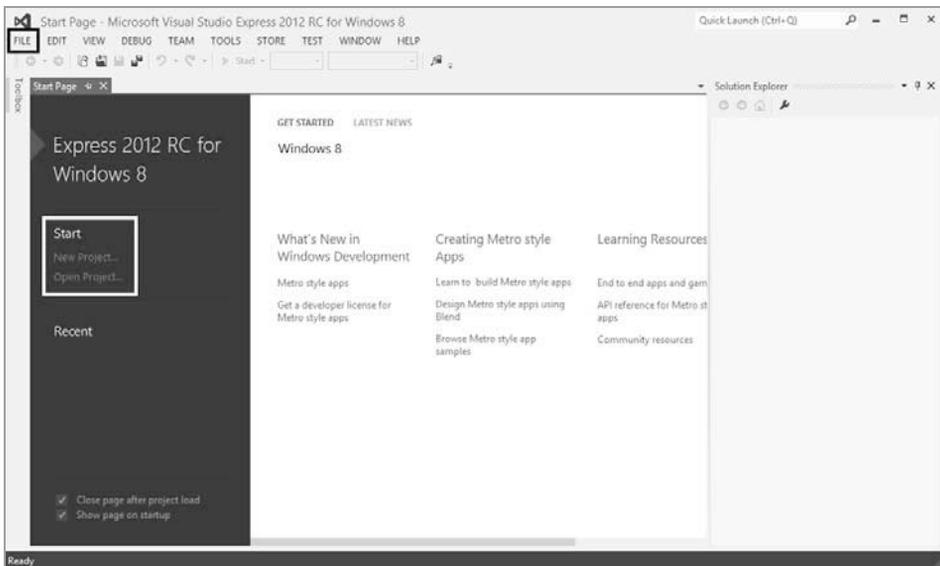
### JavaScript-/C#-Projekt anlegen

Nach der Installation möchte man sich sicherlich zuerst einmal in der Entwicklungsumgebung umsehen. Dieser Tatsache soll an dieser Stelle mit einem typischen HalloWelt-Projekt Rechnung getragen werden. In der weiteren Beschreibung wird davon ausgegangen, dass Visual Studio 2012 in der Express-Version verwendet wird.

Gewöhnungsbedürftig ist sicherlich die Tatsache, dass Visual Studio nun nicht mehr über das Startmenü, sondern über den neuen Startbildschirm von Windows 8 aktiviert wird. Ein Nachteil des neuen Startbildschirms ist sicherlich der Punkt, dass dieser bei vielen installierten Programmen mehr als unübersichtlich ist. Eine Möglichkeit, das gesuchte Programm schneller zu finden, ist die Verwendung der Lupenfunktion, welche in der rechten unteren Ecke zu finden ist. Betätigt man diesen Schalter, so wird die

Anzeige der LiveTiles verkleinert. Aber Achtung, die Lupe ist nur dann zu sehen, wenn man den Mauszeiger in die entsprechende Ecke bewegt. Ansonsten ist die Schaltfläche nicht zu sehen. Alternativ kann auch die rechte Maustaste betätigt werden, dann wird am unteren Bildschirmrand eine Leiste eingeblendet, die den Funktionsbutton *All Apps* enthält. Nach einem Mausklick auf diesen wird eine Version des ModernUI-Desktops angezeigt, welche kleinere LiveTiles bzw. Icons verwendet. In dieser Ansicht werden allerdings auch zusätzliche Tiles eingeblendet, die normalerweise nicht zu sehen sind. Auch die lokale Suche kann natürlich zum Auffinden von Programmen verwendet werden.

Nach dem Start der IDE kann ein neues Projekt entweder über das *File*-Menü (*Datei*-Menü) oder aber über den entsprechenden Punkt auf der Start-Page von Visual Studio erstellt werden:

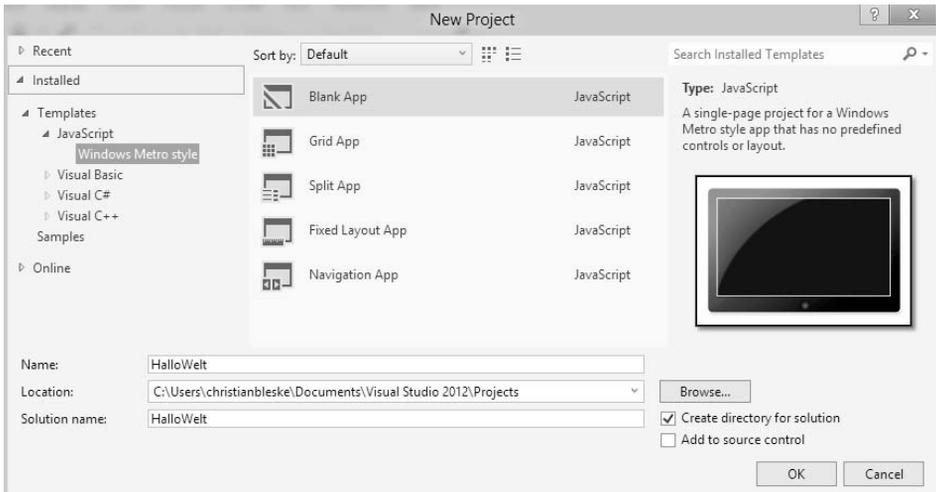


**Bild 1.7:** Neues Projekt in Visual Studio anlegen

Nach einem Mausklick wird der Dialog (*New Project / Neues Projekt*) mit den zur Verfügung stehenden Projektvorlagen angezeigt. Hier muss jetzt der Eintrag *Installed -> Templates* ausgewählt werden. Anschließend werden die zur Auswahl stehenden Sprachen angezeigt. Wenn Sie das erste Projekt mit JavaScript umsetzen möchten, so muss dieser Bereich erweitert werden. Analog verfahren Sie bitte, wenn Sie ein C#-Projekt erstellen möchten. Unterhalb der Sprache wird nur ein Punkt *Windows Metro Style* angezeigt. Selektiert man diesen, so werden die für eine Sprache vorhandenen Vorlagen angezeigt.

### Achtung

JavaScript und C# bieten zum Teil unterschiedliche Vorlagen an. An dieser Stelle spielt das jedoch keine Rolle. In beiden Fällen muss die Vorlage *Blank Application* ausgewählt werden.



**Bild 1.8:** Auswahl der Vorlage und Eingabe des Projektname

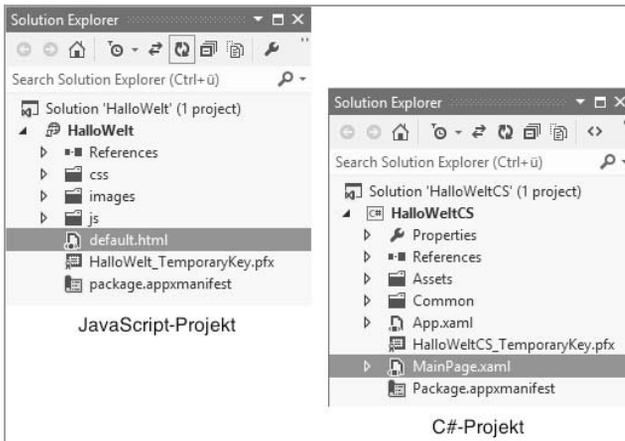
Im unteren Bereich des Dialogs befindet sich ein Eingabefeld für den Projektname. Dort kann auch der Speicherort des Projekts festgelegt werden. Als Projektname (Feld: *Name*) geben Sie bitte *HalloWelt* ein. Den Pfad können Sie frei wählen. Im Feld *Solution Name* können Sie ebenfalls *HalloWelt* eintragen. In Visual Studio ist es möglich, mehrere Projekte in einer Projektmappe zusammenzufassen. Die Solution ist eben diese Projektmappe. Anschließend muss die *OK*-Schaltfläche betätigt werden. Es dauert einen kleinen Moment, in welchem Visual Studio aus der Projektvorlage das Anwendungsgerüst generiert.

Die Entwicklungsumgebung zeigt nach der Projektanlage den Editor für die Erfassung von Quelltext an. Um die Zeichenkette »Hallo Welt« anzuzeigen, sobald das Programm gestartet wurde, muss der Text in das zugehörige Formular (HTML bei JavaScript und XAML bei C#) des Projekts eingefügt werden.

### Hinweis

Das Formular (HTML-/XAML-Datei) und der zugehörige Programmcode (JavaScript-/C#-Datei) bilden eine Einheit. Im Formular wird der Code eingefügt, welcher zur Darstellung des Formulars benötigt wird. Die Programmlogik wird in der Codedatei hinterlegt (.js oder .cs).

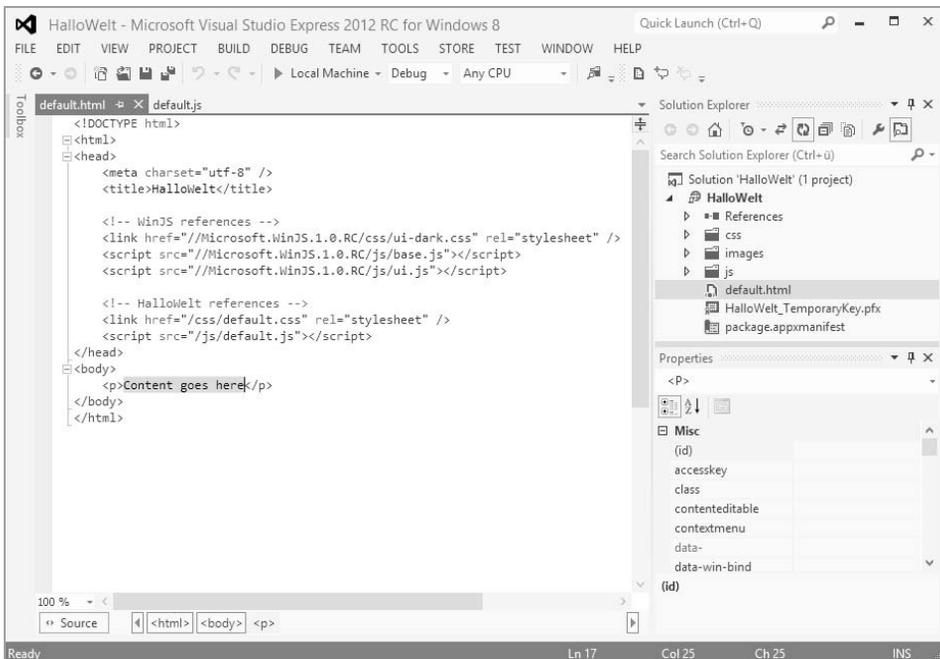
Die Bearbeitungsmöglichkeit des Formulars muss also jetzt erst einmal aktiviert werden. Im JavaScript-Projekt selektieren Sie im Solution Explorer die Datei *default.html*. In einem C#-Projekt heißt die zugehörige Datei *MainPage.xaml*. Aktivieren Sie nun die entsprechende Datei durch einen Doppelklick auf den Dateinamen im Solution Explorer. Die zugehörigen Quellcodedateien für JavaScript/C# tragen denselben Namen wie die Formulardateien, nur dass zusätzlich noch ein *.js* für JavaScript oder *.cs* für C# angehängt wird.



**Bild 1.9:** Aktivierung des zugehörigen Formulars im Solution Explorer

In einem JavaScript-Projekt sind die zugehörigen JavaScript-Dateien im Ordner `js` des Projekts zu finden. Bei einem C#-Projekt wird die zum Formular gehörige C#-Datei durch Betätigung des kleinen Dreiecks vor dem Formularnamen angezeigt.

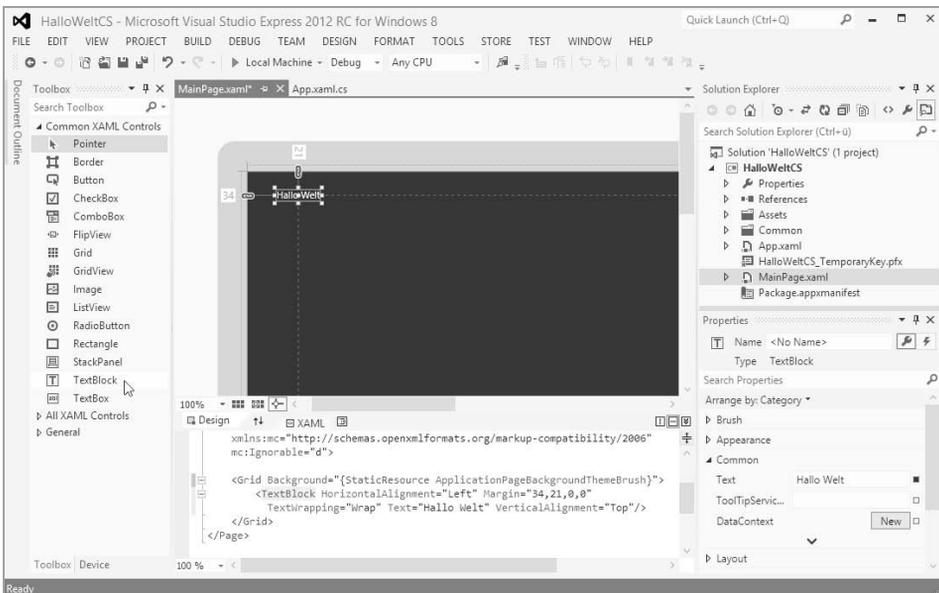
Nach dem Aufruf der Formularansicht kann die Zeichenkette eingegeben werden. Im JavaScript-Projekt suchen Sie hierzu im Editorfenster bitte die Zeile, in welcher der Text `Content goes here` zu finden ist, und tauschen diese gegen die neue Zeichenkette aus.



**Bild 1.10:** Zeichenkette »HaloWelt« im JavaScript-Projekt einfügen

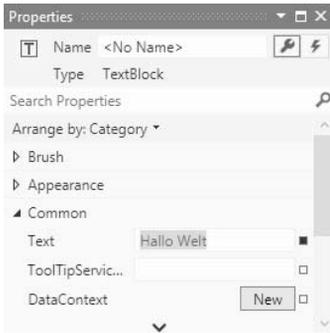
Im C#-Projekt ist der Aufwand für die erste Ausgabe etwas größer. Hier wird ein Steuerelement (Control) benötigt, um Text innerhalb des XAML-Formulars (XAML-Page) anzeigen zu können. Nach der Umschaltung in den XAML-Designer (das dauert je nach Rechner etwas) ist im oberen Bereich der IDE ein Ausschnitt der XAML-Page zu sehen. Unterhalb dieser Ansicht wird der XAML-Editor angezeigt, und im linken Bereich ist die *Toolbox* zu finden, welche die XAML-Steuerelemente enthält. Im Register *Common XAML Controls* finden Sie ein Control mit dem Namen *TextBlock*. Dieses kann entweder durch einen Doppelklick oder aber per Drag & Drop im XAML-Formular positioniert werden.

Sobald das *TextBlock*-Control innerhalb der XAML-Page eingefügt worden ist, muss nur noch der Text eingetragen werden. Hierzu gibt es zwei Möglichkeiten. Die erste führt durch Verwendung des XAML-Editors zum Ziel.



**Bild 1.11:** Einfügen der Zeichenkette »Hallo Welt« im XAML-Formular

Im XAML-Editor sucht man nach dem *TextBlock*-Control (Tag `<TextBlock/>`). Dieses Control hat eine Eigenschaft, welche die Bezeichnung *Text* trägt. Innerhalb der Anführungszeichen wird nun einfach die Zeichenkette »Hallo Welt« eingetragen. Eine zweite Möglichkeit besteht darin, das sogenannte Properties-Fenster zu verwenden. Im Properties-Fenster können die Eigenschaften eines im XAML-Designer markierten Controls bequem bearbeitet werden. Das Properties-Fenster befindet sich im rechten unteren Bereich der Entwicklungsumgebung. Sobald ein Control im XAML-Designer ausgewählt wurde, werden dessen Eigenschaften dort angezeigt.



**Bild 1.12:** Die Eigenschaften eines Controls werden im Properties-Fenster angezeigt

Die zahlreichen Eigenschaften eines Controls werden in Kategorien eingeteilt. Blättern Sie dieses durch, bis der Eintrag *Common* erscheint. Dieser muss erweitert werden (Mausklick auf das kleine Dreieck), im Abschnitt *Common* ist die Eigenschaft *Text* zu finden. Hier wird die Zeichenkette »Hallo Welt« eingegeben und dann die Return-Taste betätigt; anschließend ist die Änderung direkt im XAML-Designer zu sehen.

Ein letzter Schritt ist noch übrig: Das Programm muss nun aus der Entwicklungsumgebung heraus gestartet werden. Zum Start der App hat man mehrere Möglichkeiten. Entweder betätigt man die Taste F5 oder die Schaltfläche mit dem Dreieckssymbol und der Bezeichnung *Local Machine*. Anschließend wird das Programm gestartet und die berühmte Zeichenfolge wird angezeigt. Eine Projektvorlage auszuwählen und das Projekt anschließend zu starten sollte nun kein Problem mehr sein.

Wenn Sie Ihr Werk speichern möchten, dann können Sie den entsprechenden Menüpunkt entweder über das File-Menü finden (*Save All*) oder Sie betätigen den Button mit den zwei Disketten in der Menüleiste von Visual Studio (*Ctrl + Shift + S*).

### 1.4.3 Das Dev-Center für Windows Store Apps

Das Dev-Center ist eine Ressource für Programmierer, welche Windows-8-Apps entwickeln möchten. Hier finden Sie neben Tipps & Tricks sowie FAQs auch App-Beispiele und Foren, in denen Sie recherchieren und Fragen stellen können. Die Page erreichen Sie unter <http://msdn.microsoft.com/en-us/windows/apps/br229512.aspx>.

# Stichwortverzeichnis

## Symbole

.NET-Anwendungen 33  
 32 Bit 16  
 64 Bit 16

## A

abstract 79  
 Accelerometer 278, 282  
 Accelerometer-API 278  
 AccelerometerReading 282  
 AcceptsReturn 365  
 Active Document View 149  
 Actual Size 149  
 Add 99, 107  
   New Folder 149  
   New/Existing Item 149  
   New/Existing Project 149  
   Reference 149  
   Tab 112  
   Watch 145  
 Add-AppDevPackage 332  
 Adobe Photoshop 149  
 Adorners for Selected Rules 149  
 Advanced features 334, 336  
 Age rating and rating certificates 334  
 All XAML Controls 112  
 Altersfreigabe 338  
 Anchor-Element 341  
 Animationen 147, 154  
 Ansichten 218  
   Simulator 219  
 Anwendungseinstellungen 244  
 Anwendungsgerüst 26  
 App  
   Beschreibung 334, 339  
   Installation 332  
   Lebenszyklus 220  
   Name 334, 335  
   Package 328  
   Testen 311  
   Veröffentlichung 321  
 App.xaml 94, 121  
 App.xaml.cs 94, 121  
 AppBar 167, 169, 341  
   HTML/JavaScript 170  
   XAML/C# 172  
 AppBar-Tag 170, 173  
 Application 94  
 Application Manifest File 140  
 Application\_Closing 95  
 Application\_Launching 95  
 ApplicationData 244  
 Application-Objekt 94, 242  
 App-Manifest 118, 183, 218, 272, 283  
 App-Name 325  
 AppSettings 245  
 App-Themes 211  
 Appx-Package 332  
 Arbeitsbereich 147  
 Argumente 69  
 ArithmeticException 61  
 Arithmetische Operatoren 48  
 Arrays 55, 58  
   Initialisierung 56  
 AssemblyFileVersion 122  
 AssemblyInfo.cs 121  
 AssemblyTitle 122  
 AssemblyVersion 122  
 Asserts 316  
 Assets-Ordner 121  
 Assets-Panel 152  
 Assistenten 114  
 Asynchrone Programmierung 34  
 Attach to Process 141, 145  
 Audio 341  
   Tag 341

Auflösungen 165, 198, 219  
 Simulator 219  
 Aufzählungen 45  
 Aufzeichnungsmodus 156  
 Ausnahmen 59  
 Ausrichtung 118, 120  
 Auto Hide 110  
 automatische Codevervollständigung 114  
 AutoPlay 356

**B**

Background Tasks 255  
 BadgeNotification 188  
 BadgeUpdateManager 189  
 BasicPage 138, 236  
 Begrenzungslinien 149  
 Benutzeroberflächen 14  
 Berechtigung 255  
 Beschleunigungssensor 278  
 Beschreibungssprache 87, 91  
 Betriebssystem 15  
 Bewegungsablauf 155  
 Bibliothek 84, 106  
 Bing Maps 264  
 Anwendungstypen 266  
 Developer Account 264  
 Karte 269  
 BingMaps-Control 264  
 Blank App 102, 105, 154, 211, 231, 260, 272  
 Blank Page 138  
 Blend 147, 149, 150, 154, 155, 163  
 App-Layout 194  
 Blend for Visual Studio 97, 147  
 Boolean 45  
 Boolesche Operatoren 47, 48  
 Border 341  
 BorderBrush 160, 341  
 Border-Control 341  
 BorderThickness 160, 341  
 Breakpoint 141  
 Condition 144  
 Breakpoints-Fenster 144  
 Browse 112

Brush 202  
 Brush Transform 159  
 Build 107  
 Events 133  
 Project 149  
 Solution 100  
 Build-Register 133  
 Button 87, 92, 123, 159, 171, 174, 208, 234, 342  
 Button-Control 203, 207, 211  
 Button-Steuerelement 109, 342  
 Button-Tag 87  
 Byte 44

**C**

C# 24  
 Eigenschaften 71  
 Projektvorlagen 105  
 C#-Bibliotheken 303  
 C#-Ereignishandler 109  
 C#-Projekt 120  
 C#-Quellcode-Editor 114  
 C#-Ressourceneditor 290  
 Camera Settings 255  
 Canvas 159, 343  
 Canvas.Left 343  
 Canvas.Top 343  
 Canvas-Steuerelement 343  
 Capabilities 118, 254, 259, 266, 272, 276  
 Capture Screenshots 325  
 CaptureElement 343  
 Case-Anweisung 51  
 catch 60  
 Categories 153  
 char 42  
 Character 42, 43  
 Charms 175  
 Charms-Bar 168, 175  
 CheckBox 159, 344  
 CheckBox-Steuerelement 344  
 Choose Items 112  
 Choose Toolbox Items 112  
 class 64  
 Class 140

Class Library 106  
 Classic-Desktop 31  
 Clean Project 149  
 Clear All Data Tips 100  
 Click 109  
 ClickMode 161  
 Client & Server-Programmierung 291  
 Close 99  
   after project load 98  
   Document 148  
   Project 148  
   Solution 99  
 Cloud 248  
 Code Analysis 107  
 Code File 140  
 Code-Block 116  
 Code-Schnipsel 116  
 Codevervollständigung 111, 115  
 Codevervollständigungsfenster 112  
 ColumnDefinitions 197  
 Column-Objekte 200  
 ComboBox 345  
 ComboBox-Control 345  
 Common Properties 161  
 Common XAML Controls 112  
 Condition 144  
 Condition Tests 317  
 Configuration 133  
 Configuration Manager 131  
 Container-Steuerelement 363  
 Content before Chrome 165  
 ContentControl 346  
 ContentPresenter 346  
 ContentPresenter-Control 346  
 Context 353  
 Contracts 165, 181  
 Controls 122, 211, 341  
 Copy 99  
 Create  
   Storyboard Resource 154  
 Create App Package 332  
 Cryptography 334, 338  
 CSS 14  
 Cut 99

**D**

Dark 153  
 Dark-Theme 211  
 Dashboard 333  
 Data Store 243  
 DataContext 161  
 DataContract-Attribut 296  
 Daten übergeben 242  
 Datenbindung 161  
 Datentypen 41, 43  
 DatePicker 346  
 Debug 99, 141  
 Debug & Trace 318  
 Debuggen 143  
 Debug-Modus 325  
 Debug-Optionen 133, 145  
 Decimal 44  
 Dedicated Worker 139  
 Default language 283  
 default.html 26, 89, 102, 118  
 default.html-Page 232  
 default.js 102, 118, 267  
 Default-Konstruktor 76  
 Default-Sprache 283  
 default-Zweig 51  
 Deklaration 40  
 Delete 99  
   all Breakpoints 141  
 Deploy 107  
 Description 184, 334, 339  
 Designbereich 110  
 Designer 114  
 Desktop-Anwendungen 36  
 Dev-Center 29  
 Developer Account 321  
 Developer-Lizenz 23  
 Developer-Registrierung 321  
 Device-Panel 153  
 Device-Register 113, 213  
 Dialoge 225  
 Direct Selection 158  
 Disable Breakpoint 144  
 Display 219

Display Name 118, 120, 184  
 DLL 106  
 Document Object Model 88  
 DocumentLibrary 256  
 Documents Library Access 254  
 documentsLibrary 251  
 Dokumentation 115  
 Dokumentationsmöglichkeiten 115  
 DOM 88  
 Double 44  
 DropDownBox 345  
 Dynamik Liny Library 106

**E**

Edit 99  
   in Visual Studio 151  
   Menü 149  
 Editor 114  
 Eigenschaften 108  
 Einfache Typen 58  
 Elemente hinzufügen 139  
 Element-Schleife 54  
 Ellipse 153, 159, 347  
 Ellipse-Steuerelement 347  
 else-Zweig 49  
 Email Input Box 347  
 Entwicklungsumgebung 24, 110  
 Entwicklungswerkzeuge 15  
 Enum-Elemente 46  
 EOFException 61  
 Ereignishandler 223  
 Errors and warnings 133  
 Ersetzenfunktionen 99  
 Erweiterungen 165  
 EventListener 223  
 Exceptions 59, 61, 141  
 Exit 148  
 expando Properties 73  
 Explorer 106  
 Export Data Tips 100  
 Expression Blend 20, 151  
 Extensible Application Markup Language  
   91  
 Extensions 165, 181, 255  
 Eyedropper 158

**F**

F5 110  
 Fallunterscheidung 49  
 Fast and fluid 165  
 Fehlerbehandlung 59  
 Felder 55, 58  
   mehrdimensionale 57  
 Festplatte 15  
 File 99  
 FileOpenPickerContract 139, 140  
 finally-Klausel 62  
 Fit All 149  
 Fit Selection 149  
 Fixed Layout App 103  
 Flache Navigation 230, 231  
 Flat Navigation Pattern 231  
 Fließkommazahlen 43  
   Wertebereich 44  
 FlipView 348  
 FlipView-Steuerelement 348  
 Float 44  
 Flyout 175, 348  
 Font Manager 150  
 FontFamily 364  
 FontSize 364  
 font-size-Attribut 167  
 for(each)-Schleife 54  
 For(In)-Schleife 54  
 Foreground 160  
 Formular 122  
 for-Schleife 56  
 Foto-App 35  
 Frame 155  
 Füllfunktion 147  
 function 66  
 Funktionen 63, 69  
 Fußgesteuerte Schleife 52

**G**

Ganze Zahlen 43  
   Wertebereich 44  
 Gradient Tool 159  
 Geokoordinaten 261  
 geolocation-Objekt 261  
 Geolocator 263, 271

Geoposition 263  
 Gerade 159  
 Gestaltung von Benutzeroberflächen 147  
 Get-Methoden 71  
 getPosition 261  
 Gleichheits-Operator 47  
 Glyph 190  
 GPS-Empfänger 259  
 Grid 159, 195, 197, 199, 204, 207, 209, 351  
   Spalten 197  
   Zeilen 197  
 Grid App 102, 105, 231, 351  
 Grid-Steuerelement 197, 351  
 GridView 351  
 GridView-Control 351  
 GroupDetailPage 138, 139  
 GroupedItemsPage 138, 139  
 Gruppenrichtlinien 330  
 Gruppenrichtlinien-Editor 330

## H

HalloWelt-Projekt 24  
 Hardware 15  
 Hauptspeicher 15  
 Help 101  
 Hierarchical Navigation Pattern 230  
 Hierarchische Navigation 230  
 Hintergrundbild 201  
 Hochformat 118, 218  
   gedreht 218  
 HorizontalAlignment 364  
 HTML 14, 87  
   Attributes-Panel 162  
   File 140  
 HTML-Control 351  
 HTML-Editor 109  
 HTML-Page 122  
 Hyperlink 341  
 HyperlinkButton 352

## I

Icons 181  
 IDE 98  
 if-Anweisung 49, 50  
 Image 352

Image.Clip 352  
 Immediate 143  
 Import 148, 149  
 Import Data Tips 100  
 Initialisierung 41  
 Insert  
   Breakpoint 141  
 Installation 15  
 Installation einer App 332  
 Installed Templates 101  
 Int16 44  
 Int32 44  
 Int64 44  
 Integer 58  
 IntelliSense 115  
 Interface 83, 140  
 IOException 61  
 IsChecked 344, 362  
 ISO-Image 22  
 ItemDetailPage 138, 139  
 ItemsPage 138, 139

## J

JavaScript 14, 24  
   Eigenschaften 73  
   File 139, 140  
   Funktion 66  
   Objekte 66  
   Projektvorlagen 102  
   Vererbung 79  
 JavaScript-Projekt 118  
 JavaScript-Quellcode Editor 116

## K

Kamera 272  
 Keyframe 155  
 Klammern 69  
 Klassen 64  
   Deklaration 78  
 Klassenbibliothek 305  
   Bibliotheksfunktionen aufrufen 307  
 Komponenten 112  
 Konsolenanwendung 101  
 Konstanten 46  
 Konstruktor 76

Kontrollstrukturen 49  
 Kopfgesteuerte Schleife 52  
 Kryptografiefunktionen 338

**L**

Label-Control 279  
 Landscape 118, 153, 218  
 Landscape-flipped 218  
 Laufvariable 53  
 Laufzeitumgebungen 34  
 Layout 166, 194  
 LayoutAwarePage 236  
 Layout-Register 161  
 Lebenszyklus 220, 221  
 Light 153  
 Light-Theme 212  
 Line 159  
 ListBox 159, 353  
 ListBoxItem 353  
 ListBox-Steuerelement 353  
 ListView 354  
 ListView-Control 354  
 LiveTile 32, 167, 181, 218  
   BadgeNotification 188  
   Bilder 183  
   Texte 185  
   Vorlagentypen 182  
 LocalFolder 256  
 Local-Ordner 243  
 Locals-Fenster 144  
 Locals-Window 144  
 Location 153, 259, 266  
 Location Service 259  
 Location-Objekt 272  
 Lokalisierung 283  
   Ressourcendateien 284  
   Texte auslagern 285

**M**

MainPage 102  
 MainPage.xaml 26, 120  
 MainPage.xaml.cs 120  
 Make  
   Brush Resource 150  
   into Control 149

Manifest-Datei 120  
 MapLayer 271  
 Margin 164  
 Markup-Code 87  
 Maximum 361  
 MediaElement 355  
 Mehrdimensionale Felder 57  
 Mehrfachauswahl 50  
 Menu 357  
 Menü 99, 357  
   Debug 141  
   File 99  
 MenuCommand 357  
 MessageDialog 225  
 Methoden 69  
   Argumente 70  
   Deklaration 69  
   überladen 82  
 Methodenköpfe 84  
 Methodenname 69  
 Microphone 276  
 Minimum 361  
 Miscellaneous 161  
 ModernUI 165  
 MS Grid 351  
 MS Grid-Control 195, 204  
 MS Grid-Steuerelement 195

**N**

Name der App 118  
 Namensraum 84  
 Namespace 84, 85, 93  
 Navigate 239  
 Navigation 230, 231, 237  
 Navigation App 104  
 Navigation Patterns 229  
 navigator.js 233  
 Neues Projekt 25  
 Neues Projekt erstellen 150  
 New  
   Breakpoint 141  
   Item 148  
   Project 25, 98, 99, 117, 147, 148  
 NICHT-Operator 47  
 Note to testers 334, 339

NotRunning 222

## O

Oberfläche 203

Object-Menü 149

Objects and Timeline-Panel 154

Objekte 58, 63, 64

Objektorientierte Programmierung 63

Objekt-Schleife 54

ODER-Operator 47

On- und Offline-Hilfe 101

onActivated-Ereignis 202, 221

onLaunched 222

OOP 63

Opacity 160

Open

Folder in Windows Explorer 151

Project/Solution 148

Open Project 98, 99, 147

OperationContract 296

Operatoren 46

Options 100

Options and Settings 100, 141

Output 143

Override 82

## P

package.appxmanifest 118

Packages 334, 338

Page 92, 122, 166, 195, 229, 231

Aufteilung 195

Control 137

Datenübergabe 237

hinzufügen 137

Seitenrand 166

Setup 99

Page\_Loaded-Ereignis 157

PageControl 232

Page-Designer 110

Paint Bucket 158

Pan 158

Parameter 70

partial 66

Partielle Klassen 65

Password 358

PasswordBox 159, 358

PasswordChanged 358

PasswordChar 358

Paste 99

Pen 157

Pencil 159

Platzhalter 39

Playhead 155, 156

Play-Methode 355

Portable Class Library 309

Portrait 118, 153, 218

Portrait-flipped 218

Position 356

Positionsangabe 164

Positionsbestimmung 259, 262

Post-build event 133

Pre-build event 133

Preis 334, 335

Price tier 335

Print 99

private 67, 73

Programmcode 64

Programm-Code 87

Programmfluss 49

Programmierschnittstellen 33

Programmiersprachen 34

ProgressBar 359

ProgressBar-Steuerelement 359

ProgressRing 361

ProgressRing-Steuerelement 361

Project-Menü 149

Projects-Panel 151

Projekt

Ausführen 133

Eigenschaften 131

ergänzen 137

Projektmappe 99, 106, 151

Projektmappen-Explorer 107

Projektmappenverwaltung 98

Projektneuanlage 150

Projektstruktur 151

Projektvorlage 25, 26, 101, 102, 105, 150,  
293

Projektvorlagen für Windows-8-Apps 101

Properties 73, 108

Properties-Fenster 28, 108, 110  
 Properties-Ordner 121  
 Properties-Panel 159  
 protected 68  
 Prototypen-Funktion 79  
 Prozeduren 63  
 public 67, 73  
 PushPin 271  
 PushPin-Objekt 269

**Q**

Quellcode 15, 114  
 Quellcodeverwaltung 99  
 Quelltext-Editor 114  
 Querformat 118, 218  
   gedreht 218  
 QuickWatch 145

**R**

Radio Button 159  
 RadioButton 123 ff., 362  
 Range-Control 203, 206  
 Ratings 337  
 Rebuild Project 149  
 Recent Files 99  
 Recent Project and Solutions 99  
 Recent Projects 98, 148  
 Rechteck 159  
 Rectangle 153  
 Reference Paths 133  
 References-Ordner 120, 121  
 Referenztypen 58  
 Refresh-Button 107  
 Regionen 114  
 Register  
   Appearance 160  
   Application 132  
   Events 109  
   Transform 161  
 Release-Modus 324  
 Release-Version 133  
 Resource-Dictionary 140  
 ResourceLoader 289  
 Resources-File 139, 140  
 Resources-Panel 162

Ressourcen 162  
 Ressourcendateien 284  
 RoamingFolder 256  
 Roaming-Ordner 243  
 RoamingSettings 248  
 RowDefinitions 197  
 Row-Objekte 200  
 Run Project 149  
 Running 222

**S**

Save 148  
 Sbyte 44  
 Schaltfläche 87, 92  
 Schieberegler 203  
 Schleifen 51  
 Schlüsselwort 41, 65  
 Schnittstellen 83  
 Schnittstellendefinition 83  
 Schriften 166  
 Screenshots 325  
 ScrollBar 159  
 ScrollView 363  
 sealed 79  
 Search Contract 140  
 Search Target Contract 139  
 Search-Char 169  
 Seitenrand 164, 166  
 SelectedItem 354  
 Selection 158  
 SelectionMode 353  
 Selection-Pointer 158  
 Selling details 334, 335  
 Sensoren 259  
 ServiceContract 296  
 Set-Methoden 71  
 setting.html 176  
 Setting-Page 177  
 Settings-Char 175, 348  
 SettingsFlyout 348  
 Settings-Flyout 176  
   HTML/JavaScript 176  
   XAML/C# 178  
 Shapes 153  
 Share Target Contract 140

- Share-Charm 169
  - Short Cuts 176
  - Show
    - all Files 107
    - Headlines 149
    - on startup 98
  - Sichtbarkeit 67, 68
  - Sideloading 329
  - Silverlight-Grid 197
  - Simulator 134, 219
  - Slider 159
  - Slider/Range 364
  - Slider-/Range-Control 364
  - Slider-Control 210
  - Software 15
  - Softwareentwicklung 114
  - Solution 99
  - Solution Explorer 98, 106, 118
  - Source 352, 355
  - Source Control 99
  - Splash-Screen 147
  - Split App 103, 105
  - Split View Orientation 149
  - SplitPage 138, 139
  - Sprache 283
  - Stack-Panel 199
  - StackPanel-Steurelement 199
  - Standard-LiveTile 183
  - StandardStyles.xaml 121
  - Standard-Themes 211
  - Start
    - Debugging 100, 141
    - Page 98
    - without Debugging 141
  - Startbildschirm 31
  - Starten einer App 134
  - Startseite 98
  - Startup
    - Project 151
  - Startwert 53
  - States-Panel 153
  - Statustests 317
  - Step
    - Into 100, 141
    - Over 100, 141
  - Steurelement 110, 112, 122, 163, 341
    - Grid 197
    - MS Grid 195
  - Stoppwert 53
  - Storyboard 154
  - Stretch 352
  - String 42, 43, 58, 59
  - Style Sheet 140
  - Styles 152
  - Suchen 99
  - Supported rotations 218
  - Suspended 222
  - SuspensionManager 139
  - System.IO 84
- T**
- Tablet-Computer 259
  - Tag-Paar 87
  - Taktfrequenz 15
  - TargetName 352
  - Task-Manager 224
  - Templated Control 140
  - Templates 293
  - TemporaryFolder 256
  - Temporary-Ordner 243
  - Test Driven Development 315
  - Test Explorer 315
  - Testbericht 327
  - Testen 311
    - lokaler Test 325
  - Testfall 315
  - Testzertifikat 120, 121
  - Text 364
  - TextAlignment 364
  - TextArea 364
  - TextArea-Control 364
  - TextBlock 159, 364
  - TextBlock-Control 28, 203, 210
  - TextBlock-Steurelement 364
  - TextBox 159, 365
  - TextBox-Control 238
  - TextBox-Steurelement 365
  - Textdateien 255
  - Textfeld 123
  - TextFile 140

Textkörper 166  
 TextWrapping 364, 365  
 Theme 153, 213  
 this 74  
 TileNotification 188  
 TileTemplateType 182, 187, 188  
 tileUpdate 188  
 TileUpdateManager 185, 188  
 TileUpdater-Objekt 185  
 TimeSpan 356  
 Toggle Breakpoint 100, 141  
 Toolbox 94, 110, 112, 116, 124  
 Toolbox-Register 112  
 Tools-Menü 100  
 Tools-Panel 147, 157  
 toString 82  
 Trace 144  
 Treat warnings as errors 133  
 try 60  
 try...catch-Block 61  
 Typografie 166  
 Typumwandlung 58

**U**

Überschreiben 80  
 Überschriften 166  
 UInt16 44  
 UInt32 44  
 UInt64 44  
 UML 114  
 Umrechnungs-App  
   C# 130  
   JavaScript 127  
 Undo/Redo 99  
 UND-Operator 47  
 Undurchsichtigkeit 160  
 Ungleichheits-Operator 47  
 Ungleich-Operator 47  
 Unit Test Library 106  
 Unit-Tests 311  
 User Control 140, 178  
 User Experience 165

using 85  
 Utility-Methoden 318

**V**

Value 361  
 Variablen 39  
 Vererbung 77  
 Vergleichsoperator 47  
 Vergleichstests 316  
 Verteilte Anwendungen 291  
 VerticalAlignment 364  
 Verzeichnisstruktur 107  
 Video aufnehmen 276  
 View-Menü 99, 149  
 Virtual Box 16, 17  
 Virtuelle Maschine 18  
 Visibility 160  
 Visual Studio 20, 26, 93, 97, 98, 100, 101,  
   110, 111, 112, 133, 141, 147, 148, 150, 151  
   Installation 21  
   Neues Projekt 25  
   Projekt anlegen 117  
   Schnellstart 24  
   Sprachversion 22  
   Unit-Tests 312  
   Versionen 20  
 Visual Studio 2012 97, 303  
 Visual Studio 2012 Express 147  
 Visual Studio 2012 Express for Win8 97  
 Visual Studio Express 293  
 Visual Studio-Vorlage 167  
 visuelle Entwicklung 147  
 void 69  
 Vollbild 35  
 Vollbild-Apps 35  
 Vorlagen 167

**W**

Wahrheitswerte 45  
 Warnstufe 133  
 WCF 291  
 WCF Service Application 293  
 WCF Test Client 294  
 WCF-Client 297

WCF-Server 293  
 WCF-Service 293, 297  
 Web Services 292  
 Web-Anwendungen 37  
 Webcam 272  
 Web-Installer 21  
 Werkzeugleiste 112  
 Wetter-App 181  
 What's new 98  
 while-Schleife 52  
 Widget 32  
 Win32-Anwendungen 33  
 win-backbutton 234  
 Windows 7 101  
 Windows 8  
   (API-)Architektur 33  
   Versionen 15  
 Windows App Certification Kit 321, 325  
 Windows Communication Foundation  
   291  
 Windows Library for JavaScript 303  
 Windows Live ID 321  
 Windows Presentation Foundation 91  
 Windows RT 16, 34  
 Windows Runtime 34  
 Windows Runtime Component 106, 303  
 Windows Runtime-Bibliothek 303  
 Windows Store 321  
   App übertragen 332  
   Dashboard 333  
 Windows-Account 243  
 Windows-Store-Anwendungen 35  
 Windows-Store-App 165

WinJS 303  
 WinJS.UI.Menu 357  
 WinJS.UI.MenuCommand 357  
 WinRT 34, 303  
 Workspace 148, 150

## X

XAML 91, 92  
   Storyboard 154  
 XAML-Anweisungen  
   Page 92  
 XAML-Designer 106, 110, 111, 124  
 XAML-Editor 111, 124  
 XAML-Page 124  
 XML File 140, 141  
 XML Schema 140  
 XSLT File 140, 141

## Z

Zahlenratespiel 193  
 Zählschleifen 53  
 Zeichen 42, 43  
 Zeichenkette 42, 43, 59  
 Zeilennummer 144  
 Zeitleiste 155  
 Zoom 110, 147  
   In 149  
   Out 149  
 Zugriffsverweigerung 254  
 Zuweisung 39, 41  
 Zuweisungsoperator 46

Christian Bleske

# Windows 8 Apps entwickeln

Mit Windows 8 bringt Microsoft sein erfolgreiches Betriebssystem in Form für Tablets und die Touch-Bedienung, bedient dabei aber gleichzeitig noch die traditionellen Desktop-Computer. Wer Apps für die neue Modern-UI-Oberfläche und den Windows Store entwickeln will, erhält mit diesem Buch das nötige Know-how, egal ob er den Weg über HTML und JavaScript oder über die Microsoft-typischen Sprachen XAML und C# geht. Der ganze Weg der App-Entwicklung vom Layout bis zur Vermarktung im Windows Store wird gezeigt.

## HTML/JavaScript oder XAML/C#

Sie haben die Wahl: Parallel werden in diesem Buch alle Schritte der App-Entwicklung sowohl in der Kombination HTML und JavaScript als auch in der Kombination XAML und C# beschrieben. Sie können also die Sprache wählen, die Ihnen leichter erscheint bzw. in der Sie geübt sind. Als Entwicklungsumgebung wird in den Beschreibungen grundsätzlich Visual Studio 2012 zugrunde gelegt.

## Vom Entwurf bis in den Store

Es sind doch einige Schritte, die zur Erstellung einer erfolgreichen App notwendig sind, und hier werden sie gezeigt: Layout, Design der Oberfläche, Programmierung der Funktionen, Testen der App, Einstellen in den Store. Dieses Buch ist Ihr Begleiter auf dem gesamten Weg.

## Auf [www.buch.cd](http://www.buch.cd):

Der komplette Quellcode des Buches.  
als Basis für Ihre eigene App-Entwicklung.

## Aus dem Inhalt:

- Windows 8 – Neues Betriebssystem für neue Endgeräte
- Die Windows-8-(API-)Architektur
- JavaScript und C# – Grundlagen der Programmierung
- Objektorientierte Programmierung mit JavaScript und C#
- Die Markup-Sprachen HTML & XAML
- Visual Studio 2012
- Projektvorlagen für Windows-8-Apps
- Projekte mit Visual Studio erstellen/verwalten/debuggen
- Blend for Visual Studio
- Layout einer Windows-Store-App
- Die AppBar (AppBar)
- Das LiveTile einer App
- Entwicklung einer Beispiel-App: Layout der App, Design der Oberfläche, Programmierung der Funktionen, Test, Lebenszyklus einer App
- Navigation und Datenübergabe in Apps
- Daten- und Dateizugriff
- Sensoren

## Über den Autor:

Christian Bleske ist Autor, Trainer und Entwickler. Sein Arbeitsschwerpunkt liegt auf Client/Server-Technologien und mobilen Anwendungen. Seine Fachaufsätze erscheinen in vielen namhaften Entwicklerzeitschriften. Er lebt in Witten im Ruhrgebiet.



30,- EUR [D]  
ISBN 978-3-645-60200-6

Besuchen Sie  
unsere Website  
[www.franzis.de](http://www.franzis.de)

FRANZIS