

Einstieg in WPF 4.5

Grundlagen und Praxis. Wie Sie die Windows Presentation Foundation richtig einsetzen

Bearbeitet von
Thomas Theis

erweitert 2012. Taschenbuch. 525 S. Paperback

ISBN 978 3 8362 1967 9

Format (B x L): 17,2 x 23 cm

[Weitere Fachgebiete > EDV, Informatik > Programmiersprachen: Methoden > Microsoft Programmierung](#)

schnell und portofrei erhältlich bei

The logo for beck-shop.de features the text 'beck-shop.de' in a bold, red, sans-serif font. Above the 'i' in 'shop' are three red dots of increasing size. Below the main text, the words 'DIE FACHBUCHHANDLUNG' are written in a smaller, red, all-caps, sans-serif font.

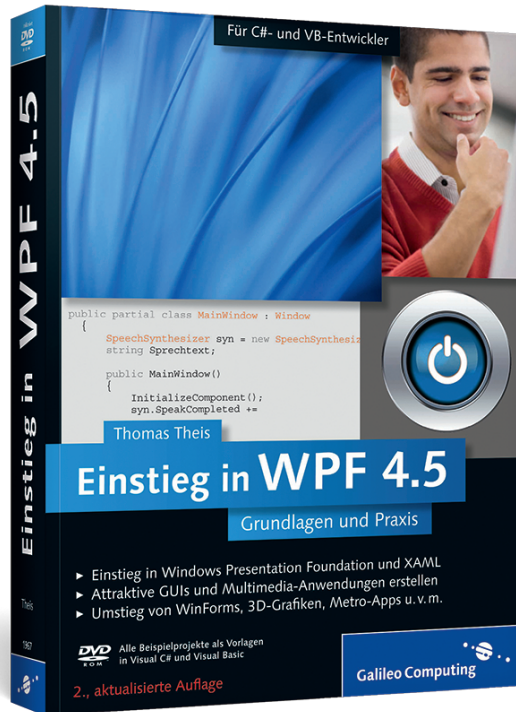
beck-shop.de
DIE FACHBUCHHANDLUNG

Die Online-Fachbuchhandlung beck-shop.de ist spezialisiert auf Fachbücher, insbesondere Recht, Steuern und Wirtschaft. Im Sortiment finden Sie alle Medien (Bücher, Zeitschriften, CDs, eBooks, etc.) aller Verlage. Ergänzt wird das Programm durch Services wie Neuerscheinungsdienst oder Zusammenstellungen von Büchern zu Sonderpreisen. Der Shop führt mehr als 8 Millionen Produkte.

Thomas Theis

Einstieg in WPF 4.5

Grundlagen und Praxis



Auf einen Blick

1	Einführung	15
2	XAML und WPF	21
3	Layout	31
4	Steuerelemente	55
5	Ereignisse und Kommandos	139
6	Anwendungen	161
7	Vorlagen	201
8	Daten	225
9	2D-Grafik	251
10	3D-Grafik	301
11	Animation	347
12	Audio und Video	387
13	Dokumente und Drucken	415
14	Interoperabilität	449
15	Windows Store-Apps für Windows 8	463

Inhalt

1	Einführung	15
1.1	Vorteile der WPF	15
1.1.1	Grafik in der WPF	16
1.2	Aufbau des Buchs	16
1.3	Visual Studio 2012	17
1.3.1	Ein neues Projekt	18
1.4	Kaxaml	19
1.5	XAML und C# bzw. VB	19
1.6	Danksagung	19
2	XAML und WPF	21
2.1	Dokumentstruktur	21
2.2	Property Elements	23
2.3	Dependency Properties	24
2.4	Attached Properties	25
2.5	Markup Extensions	26
2.6	Routed Events	27
2.7	Attached Events	29
3	Layout	31
3.1	Canvas	32
3.1.1	Positionierung	32
3.1.2	Elemente neu erzeugen	34
3.1.3	Layout-Hierarchie	35

3.2 StackPanel	37
3.2.1 Elemente neu erzeugen	38
3.3 WrapPanel	39
3.3.1 Elemente neu erzeugen	41
3.3.2 Vertikale Orientierung	41
3.4 DockPanel	42
3.4.1 DockPanel in Hierarchie	43
3.4.2 Elemente neu erzeugen	44
3.5 Grid	45
3.5.1 Elemente neu erzeugen	46
3.5.2 Elemente über mehrere Zellen	48
3.5.3 Größe der Zellen festlegen	49
3.5.4 Die Größe der Zellen flexibel gestalten	51
3.6 Layout-Kombination	53

4 Steuerelemente 55

4.1 Allgemeiner Aufbau	55
4.1.1 Größe, Schrift, Farbe, Bedienung per Tastatur	56
4.1.2 Sichtbarkeit, Bedienbarkeit	58
4.1.3 Elemente mit EventHandler neu erzeugen, Elemente löschen	60
4.1.4 Padding, Innenabstand	61
4.1.5 Margin, Außenabstand	63
4.1.6 Alignment, Ausrichtung	65
4.2 Schalter	66
4.2.1 Button	66
4.2.2 RepeatButton	68
4.2.3 ToggleButton und CheckBox	69
4.2.4 RadioButton	71
4.2.5 Auswahl einstellen	74
4.3 Text und Beschriftung	74
4.3.1 Label	74
4.3.2 TextBlock	76
4.3.3 ToolTip	79
4.3.4 TextBox	81

4.3.5	PasswordBox	83
4.3.6	RichTextBox	84
4.4	Auswahl	85
4.4.1	ListBox, Einzel-Auswahl	85
4.4.2	ListBox, Mehrfach-Auswahl	89
4.4.3	ComboBox	92
4.4.4	TreeView	94
4.5	Zahlenwerte	101
4.5.1	ProgressBar	101
4.5.2	Slider	105
4.5.3	ScrollBar	108
4.6	Container	109
4.6.1	Border	109
4.6.2	GroupBox	112
4.6.3	Expander	113
4.6.4	TabControl	116
4.7	Menüs und Leisten	118
4.7.1	Hauptmenü	118
4.7.2	Kontextmenü	121
4.7.3	Symbolleiste	123
4.7.4	Statusleiste	126
4.8	Datum	128
4.8.1	Calendar	128
4.8.2	DatePicker	133
4.9	Weitere Elemente	134
4.9.1	Image	134
4.9.2	WebBrowser	136

5 Ereignisse und Kommandos 139

5.1	Tastatur	139
5.1.1	Anzeige der Tastaturinformationen	139
5.1.2	Steuerung durch Tasten	141
5.2	Maus	142
5.2.1	Anzeige der Mausinformationen	142

5.3 Eingabestift	145
5.4 Touchscreen	147
5.5 Kommandos	152
5.5.1 Eingebaute Kommandos	152
5.5.2 Kommandos mit Eingabegesten verbinden	155
5.5.3 Eigene Kommandos	157

6 Anwendungen 161

6.1 Allgemeiner Aufbau	161
6.1.1 Einfache Anwendung	161
6.1.2 Anwendung mit Steuerelement	163
6.1.3 Reihenfolge der Ereignisse	164
6.1.4 Aufruf von der Kommandozeile	167
6.2 Ressourcen	170
6.2.1 Physische Ressourcen	170
6.2.2 Logische Ressourcen	173
6.3 Fenster	176
6.3.1 Eigenschaften und Ereignisse von Fenstern	176
6.3.2 Eigene Dialogfelder	180
6.4 Navigation mit Seiten	183
6.4.1 Eine Reihe von Seiten	183
6.4.2 Frame mit Unterseiten	187
6.5 Gadgets	190
6.6 Browseranwendung	192
6.7 Ribbonanwendung	194

7 Vorlagen 201

7.1 Styles	201
7.1.1 Benannte Styles	202
7.1.2 Typ-Styles	205
7.1.3 Vererbung benannter Styles	206

7.1.4	Vererbung von Typ-Styles	207
7.1.5	Verwandte Steuerelement-Typen	208
7.1.6	EventSetter	210
7.2	Property Trigger	212
7.2.1	Einfache Property Trigger	212
7.2.2	Multi-Trigger	213
7.3	Control Templates	215
7.3.1	Ein erstes Control Template	215
7.3.2	Control Template mit Trigger	216
7.3.3	Control Template mit Bindung	218
7.3.4	Control Template in Typ-Style	219
7.4	Skins	221
8	Daten	225
8.1	Datenbindung	225
8.1.1	Setzen und Lösen einer Bindung	225
8.1.2	Richtung und Zeitpunkt einer Bindung	227
8.2	Validierung	230
8.3	Datenquellen	232
8.3.1	Ein Objekt als Datenquelle	232
8.3.2	Kontext einer Datenbindung	234
8.3.3	Auflistung von Objekten	235
8.3.4	Object Data Provider	238
8.3.5	Datenbank	239
8.4	DataGrid	241
8.4.1	Einfacher Aufbau	241
8.4.2	Standard-Einstellungen	243
8.4.3	Weitere Spaltentypen	244
8.5	DataTemplates	246
8.6	DataTrigger	248

9	2D-Grafik	251
9.1	Shapes	251
9.1.1	Rechtecke und Ellipsen	252
9.1.2	Linie	254
9.1.3	Polygon und Polylinie	256
9.1.4	Linienende	257
9.2	Geometrien	259
9.2.1	Einfache geometrische Formen	259
9.2.2	Kombinierte Geometrien	262
9.2.3	Pfadgeometrien für komplexe Formen	264
9.2.4	Pfadgeometrie in Pfadmarkupsyntax	268
9.2.5	Geometriegruppe	268
9.3	Drawings	270
9.4	Pinsel	273
9.4.1	SolidColorBrush	273
9.4.2	LinearGradientBrush	274
9.4.3	RadialGradientBrush	277
9.4.4	ImageBrush	279
9.5	Transformationen	282
9.5.1	RotateTransform mit RenderTransform	283
9.5.2	RotateTransform mit LayoutTransform	285
9.5.3	ScaleTransform	286
9.5.4	SkewTransform	288
9.5.5	TranslateTransform	290
9.5.6	TransformGroup	291
9.6	Transparenz	292
9.6.1	Transparenz mit Opacity und Background	292
9.6.2	Maskierung mit OpacityMask	294
9.6.3	Ausstanzung mit Clip	295
9.7	Effekte	296
9.8	Verzierungen	298

10 3D-Grafik 301

10.1 Allgemeiner Aufbau	301
10.1.1 Koordinatensystem	301
10.1.2 Kamera, Licht und Material	303
10.1.3 Dreieck in XAML	303
10.1.4 Ein Dreieck in Programmcode erzeugen	306
10.1.5 Würfel	308
10.1.6 Gemeinsame Punkte	311
10.2 Kamera	311
10.2.1 Perspektivische Kamera	312
10.2.2 Lage der Kamera	313
10.3 Licht	314
10.4 Modelle	319
10.4.1 Gruppe von 3D-Körpern	319
10.4.2 3D-Körper mit Ereignissen	321
10.4.3 Gruppe von 3D-Körpern mit Ereignissen	323
10.4.4 3D-Körper mit Oberflächengestaltung	324
10.5 Material und Textur	327
10.5.1 Material	327
10.5.2 Textur	329
10.6 Transformationen	332
10.6.1 ScaleTransform3D	333
10.6.2 TranslateTransform3D	334
10.6.3 RotateTransform3D	336
10.6.4 Transform3DGroup	339
10.6.5 Transform3DGroup aus Rotationen	341
10.7 Eine 3D-Landschaft	343

11 Animation 347

11.1 Allgemeiner Aufbau	348
11.1.1 Einfache DoubleAnimation	348
11.1.2 DoubleAnimation, weitere Eigenschaften	351
11.1.3 PointAnimation	354

11.2 Storyboard	355
11.2.1 Storyboard als Ressource	356
11.2.2 Storyboard per Programmcode	358
11.2.3 Storyboard steuern	360
11.2.4 Animierte Transformation	362
11.2.5 ColorAnimation	364
11.3 Event Trigger	365
11.3.1 Event Trigger in Element	365
11.3.2 Event Trigger und Ressourcen	366
11.3.3 Event Trigger in Style	367
11.3.4 Event Trigger zur Steuerung	369
11.4 Animierte 3D-Rotation	371
11.5 Keyframes	374
11.5.1 Keyframes für Double	374
11.5.2 Keyframes für Color	377
11.5.3 KeyFrames für String	378
11.6 Easing Functions	379
11.7 Pfadanimationen	383

12 Audio und Video 387

12.1 Audio	387
12.1.1 SoundPlayer in Programmcode	387
12.1.2 SystemSound	390
12.1.3 SoundPlayer in XAML	391
12.1.4 MediaPlayer für Audio	391
12.1.5 MediaElement für Audio	395
12.2 Video	396
12.2.1 MediaElement für Video	396
12.3 Sprachausgabe	398
12.3.1 Text ausgeben	398
12.3.2 Text zusammensetzen	403

12.4 Spracheingabe	407
12.4.1 Externe Spracherkennung	408
12.4.2 Interne Spracherkennung	410
12.4.3 Steuerung per Spracherkennung	411

13 Dokumente und Drucken 415

13.1 FlowDocument	415
13.1.1 FlowDocumentReader	416
13.1.2 Block-Typ Absatz	418
13.1.3 Block-Typ Abschnitt	420
13.1.4 Block-Typ Liste	422
13.1.5 Block-Typ Tabelle	425
13.1.6 Block-Typ Steuerelement-Container	429
13.1.7 Inlines	431
13.1.8 Inline-Typ Figure	436
13.1.9 FlowDocumentScrollViewer	438
13.1.10 FlowDocumentPageViewer	439
13.1.11 RichTextBox	440
13.2 FixedDocument	443
13.3 Drucken	445

14 Interoperabilität 449

14.1 Windows Forms in WPF	449
14.1.1 Windows Forms-Steuerelemente in WPF	449
14.1.2 Windows Forms-Standard-Dialogfelder in WPF	451
14.2 WPF in Windows Forms	454
14.2.1 WPF-Steuerelemente in Windows Forms	454
14.3 MS Office in WPF	457
14.3.1 Excel-Mappe	458
14.3.2 Word-Dokument	460

15 Windows Store-Apps für Windows 8	463
15.1 Projektvorlagen für Windows Store Apps	463
15.2 Projektvorlage Blank	465
15.3 Steuerelemente	467
15.4 Seitenvorlagen für Windows Store Apps	469
15.5 Eine Reihe von Seiten	470
15.6 Eine geteilte Seite	474
15.7 Seitenvorlage Standardseite	475
15.8 Projektvorlage Grid	478
15.9 Projektvorlage Split	481
15.10 Prüfen einer App	483
Der Autor	485
Index	487

Kapitel 1

Einführung

In diesem ersten Kapitel werden einige grundlegende Begriffe der WPF, der Aufbau des Buchs und die Arbeit mit dem Visual Studio von Microsoft erläutert.

WPF steht für *Windows Presentation Foundation*. Es handelt sich dabei um eine 2006 gänzlich neu eingeführte Bibliothek von Klassen, die zur Gestaltung von Oberflächen und zur Integration von Multimedia-Komponenten und Animationen dient. Sie vereint die Vorteile von DirectX, Windows Forms, Adobe Flash, HTML und CSS.

1.1 Vorteile der WPF

Der Umstieg auf diese neue Technologie geschieht nur langsam. In der Praxis setzen Entwickler häufig noch den Vorgänger der WPF, Windows Forms, ein. In diesem Abschnitt werden einige Eigenschaften und Vorteile der WPF dargestellt.

Die WPF ermöglicht eine verbesserte Gestaltung von Oberflächen. Layout, 3D-Grafiken, Sprachintegration, Animation, Datenzugriff und vieles mehr basieren auf einer einheitlichen Technik. Der Benutzer kann außerdem die Bedienung dieser Oberflächen schnell und intuitiv erlernen.

Einzelne Elemente oder ganze Oberflächen sind schneller anpassbar und austauschbar. Die Aufgabenbereiche des Designers (Gestaltung der Oberfläche) und des Entwicklers (Codierung der Abläufe) sind klarer getrennt. So kann die Erstellung einer Anwendung in parallelen Schritten erfolgen.

Die WPF wurde gänzlich neu entwickelt; es musste keine Rücksicht auf alte Techniken genommen werden. Desktop-Anwendungen können ohne großen Aufwand auch für die Nutzung im Web umgestellt werden.

WPF-Anwendungen können außer auf die klassischen Medien Maus, Tastatur und Bildschirm auch auf Touchscreen und Digitalisierbrett zugreifen. Sie können über Sprache gesteuert werden und Sprachausgaben erzeugen.

Windows Forms ist länger auf dem Markt als die WPF. Daher besitzt es einige Elemente, die in der WPF noch nicht vorliegen. Diese Elemente werden aber in naher Zukunft hinzugefügt. Außerdem haben Sie die Möglichkeit, beide Techniken zu vereinen. Sie können Elemente aus Windows Forms in einer WPF-Anwendung unterbringen und umgekehrt. So können Sie die Vorzüge aus beiden Welten nutzen.

1.1.1 Grafik in der WPF

Die WPF nutzt intern DirectX statt des veralteten GDI+, wie es bei Windows Forms der Fall ist. Damit wird die Darstellung hardwarebeschleunigt. 2D- und 3D-Grafiken haben mehr Möglichkeiten und sind schneller.

Es wird Vektorgrafik statt Pixelgrafik verwendet. Damit ist eine Anwendung besser skalierbar. Sie wird unabhängig von der Auflösung und passt für viele verschiedene Ausgabemedien. Dies wird aufgrund des mittlerweile fließenden Übergangs von Smartphone über Pad, Netbook, Laptop, Desktop bis hin zu Großbildschirmen immer wichtiger.

Die Möglichkeiten der Grafik-Hardware beim Benutzer können besser genutzt werden. Die Grafik-Hardware wurde mit den Jahren immer besser und billiger, und damit stiegen auch die Erwartungen der Benutzer weiter an. Falls beim Benutzer permanent oder temporär keine geeignete Grafik-Hardware vorhanden sein sollte, so besitzt die WPF Fallback-Mechanismen. Dies beeinflusst die Entwicklung nicht, nutzt aber die Möglichkeiten optimal aus.

1.2 Aufbau des Buchs

In jedem Abschnitt wird die Thematik anhand eines vollständigen Projekts erläutert. Sie sehen jeweils einen Screenshot und die wichtigen Teile des Codes. Ich empfehle Ihnen, das jeweilige Projekt auf Ihren PC zu kopieren und es auf Ihrem Rechner aufzurufen, parallel zum Lesen des Buchs. Viele Zusammenhänge werden durch die Bedienung der Anwendung noch deutlicher.

In diesem Kapitel 1 werden einige grundlegende Begriffe erläutert. Die Besonderheiten und Erweiterungen von XAML gegenüber XML und der WPF im Vergleich zu einer herkömmlichen Klassenbibliothek folgen in Kapitel 2.

Im Kapitel 3 lernen Sie verschiedene Layout-Möglichkeiten zur Anordnung der Elemente kennen. Die WPF bietet zahlreiche Steuerelemente, diese folgen, in Gruppen unterteilt, in Kapitel 4.

Mithilfe der WPF können Sie auf alte und neue Eingabemedien zugreifen. Diese werden, zusammen mit dem Prinzip der Kommandos, in Kapitel 5 erläutert. Kapitel 6 beschreibt die verschiedenen Anwendungstypen und das Prinzip der Ressourcen.

Vorlagen sorgen für einheitliches, aber individuelles Aussehen – siehe Kapitel 7. In Kapitel 8 wird erläutert, wie Sie eine Verbindung zwischen der Oberfläche und den Anwendungsdaten herstellen können.

Die besondere Stärke der WPF liegt in der Grafik. Der Aufbau von 2D-Grafiken und 3D-Grafiken wird in den Kapiteln 9 und 10 besprochen. Das Ganze gerät mithilfe von Animationen in Bewegung, die Thema von Kapitel 11 sind.

Multimediateilkomponenten aus dem Bereich Audio und Video können Sie mithilfe der WPF in Ihre Anwendungen integrieren. Dies ist Thema von Kapitel 12.

In Kapitel 13 lernen Sie, wie Sie verschiedene Formen von Dokumenten erstellen, benutzen und ausdrucken. In Kapitel 14 folgt das Zusammenspiel der WPF mit Windows Forms und MS Office. Kapitel 15 bietet eine Einführung in die Windows Store Apps unter Windows 8.

1.3 Visual Studio 2012

Die Entwicklungsumgebung Visual Studio 2012 von Microsoft ist selber mithilfe der WPF entwickelt worden. Die frei verfügbaren Express-Ausgaben ermöglichen einen schnellen Einstieg in die Programmierung mit WPF. Für die Beispiele dieses Buchs können Sie »Visual Studio Express 2012 for Desktop« unter Windows 7 mit Service Pack 1 oder Windows 8 verwenden. Nur die Windows Store Apps in Kapitel 15 müssen mit »Visual Studio Express 2012 for Windows 8« unter Windows 8 entwickelt werden.

Die Oberfläche einer Anwendung wird mithilfe von XAML entworfen. XAML steht für *eXtensible Application Markup Language*. Es handelt sich dabei um eine XML-basierte Markierungssprache, die nicht nur in der WPF zum Einsatz kommt.

Innerhalb des Visual Studio können Sie die Oberfläche gleichzeitig in zwei Ansichten sehen: im grafischen Entwurf und im XAML-Code. Eine Änderung in einer der beiden Ansichten wirkt sich unmittelbar auf die jeweils andere Ansicht aus.

Während der Codierung werden Sie sowohl in XAML als auch im Programmiercode von der kontextsensitiven Hilfe *IntelliSense* unterstützt. Dank *IntelliSense* werden unter anderem nützliche Listen eingeblendet, zum Beispiel nach einem Punkt in der Objektschreibweise. Diese Listen enthalten nur die Elemente, in denen die bereits eingegebene Buchstabenkombination vorkommt (siehe Abbildung 1.1).

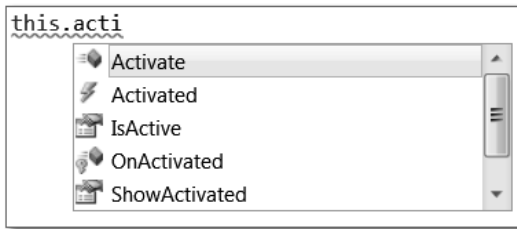


Abbildung 1.1 Diese Liste enthält nur Elemente mit »acti«.

Falls Sie einen Begriff markieren und die Taste **[F1]** betätigen, wird auch der Kontext beachtet und das passende Ziel erkannt. Dies ist dann besonders nützlich, falls der markierte Begriff zum Beispiel gleichzeitig eine Klasse und eine Eigenschaft bezeichnet.

1.3.1 Ein neues Projekt

Das Visual Studio bietet die Standardelemente einer Entwicklungsumgebung: Projektmappenexplorer, Code- und Designfenster, Eigenschaftfenster inklusive einer Liste der Ereignisse und vieles mehr. Ein neues Projekt entwerfen Sie wie folgt:

1. Rufen Sie Menü **DATEI • NEUES PROJEKT** auf.
2. Wählen Sie in der Kategorie **INSTALLIERT • VORLAGEN • VISUAL C#** die Vorlage **WPF-ANWENDUNG** aus, und vergeben Sie einen Namen.
3. Entwerfen Sie die Oberfläche im Designer, inklusive des XAML-Codes.
4. Ordnen Sie die Ereignisse den Ereignismethoden zu, entweder innerhalb des XAML-Codes oder im **EIGENSCHAFTENFENSTER**, Reiter **EREIGNISSE**.
5. Codieren Sie die Abläufe im Codefenster.
6. Nicht vergessen: Menü **DATEI • ALLE SPEICHERN**; selbst ein bereits erfolgreich gestartetes Projekt könnte ansonsten verloren gehen!

Sollten Sie versehentlich einzelne Fenster geschlossen haben: Im Menü **ANSICHT** können Sie den Projektmappenexplorer und das Eigenschaftfenster wieder einblenden. Das Designfenster blenden Sie anschließend über einen Doppelklick auf die Datei *MainWindow.xaml* im Projektmappenexplorer ein, das Codefenster über die Datei *MainWindow.xaml.cs*. Zur normalen Anordnung der Fenster gelangen Sie über das Menü **FENSTER • FENSTERLAYOUT ZURÜCKSETZEN**.

1.4 Kaxaml

Bei Kaxaml handelt es sich um einen frei verfügbaren, ressourcensparenden XAML-Editor. Er stammt von einem Entwickler, der auch im Team der WPF tätig war: Robby Ingebretsen. Kaxaml bietet einige nützliche Hilfen, um den ersten Entwurf einer Oberfläche vorzunehmen. Sie finden Kaxaml auf dem Datenträger zum Buch oder über <http://www.kaxaml.com>.

1.5 XAML und C# bzw. VB

Eine Anwendung kann ausschließlich aus XAML-Code oder ausschließlich aus Code in einer der Programmiersprachen bestehen, zum Beispiel Visual Basic oder Visual C#. Meist wird allerdings gemischt: Die Oberfläche wird in XAML entworfen, die Abläufe werden in einer Programmiersprache codiert. Jedoch sind die Übergänge fließend; es herrscht keine strenge Trennung wie in Windows Forms.

In vielen Projekten dieses Buchs werden Elemente sowohl mit XAML als auch per Programmcode erzeugt. Dies macht den hierarchischen Aufbau der Anwendung und das Zusammenspiel der einzelnen Elemente noch deutlicher.

Die Entscheidung, welche Sprache Sie verwenden, hängt von Ihren persönlichen Vorlieben und Erfahrungen ab. Es wird auf die gleiche Klassenbibliothek zugegriffen, und es stehen vergleichbare sprachliche Mittel zur Verfügung. Alle Beispielprojekte dieses Buchs liegen in zwei Versionen vor: im Buch in Visual C#, auf dem Datenträger zum Buch in beiden Sprachen. Die Erklärungen im Buch können ebenfalls für beide Sprachen genutzt werden, da dieselben WPF-Typen zugrunde liegen.

1.6 Danksagung

An dieser Stelle möchte ich mich bei Anne Scheibe, Christine Siedle, Matthias Geirhos, Friederike Daenecke und dem ganzen Team von Galileo Press für die Unterstützung und die hilfreiche Kritik bei der Erstellung dieses Buchs bedanken.

Kapitel 3

Layout

Die WPF stellt vielfältige Alternativen zur Anordnung der Elemente bereit. Diese Layout-Möglichkeiten erleichtern die Trennung von grafischer Gestaltung und Programmierung und damit eine Aufgabenteilung zwischen Designer und Entwickler.

Sie bestimmen die Anordnung der Steuerelemente in Ihrer Anwendung über das Layout. Damit sorgen Sie für ein ansprechendes Aussehen und eine gute Bedienbarkeit der Oberfläche. Sie soll stufenlos in der Größe skalierbar sein und unterschiedlichen Umgebungen angepasst werden können. Die früher übliche Vergabe fester Positionen sollten Sie daher möglichst vermeiden.

Der Inhalt des Client-Bereichs eines Anwendungsfensters ist genau ein Element. Im Allgemeinen ist dies ein Layout-Element. Dieses Layout-Element kann der Ursprung einer Hierarchie von Layouts sein. Layouts können also ineinander verschachtelt sein (siehe auch Abschnitt 3.1.3, »Layout-Hierarchie«).

Die gemeinsame Basisklasse der verschiedenen Layout-Klassen ist die Klasse `Panel`. Sie stellt viele gemeinsame Member zur Verfügung. Im Projekt *PanelAlle* (siehe Abbildung 3.1) sehen Sie fünf mögliche Layouts: links oben *Canvas*, rechts oben *StackPanel*, links unten *WrapPanel*, rechts unten *DockPanel*. Alle zusammen sind innerhalb eines *Grid* angeordnet.

Sie können auch das Innere eines Elements mithilfe von Layouts frei gestalten: Im Projekt *PanelAlle* sehen Sie einen Button, der Text und eine *CheckBox* (hier: CB) beinhaltet. Ein anderer Button enthält Text und zwei *RadioButtons* (hier: RB). Sie sehen: Die Grenze zwischen Layout und Steuerelement ist fließend. Auch in diesem Punkt zeigt sich die Vielseitigkeit der WPF.

Eine Anmerkung: Als Beispiel für die Steuerelemente, die mithilfe von Layouts positioniert werden, verwende ich häufig Buttons, unter anderem wegen ihrer guten Erkennbarkeit.

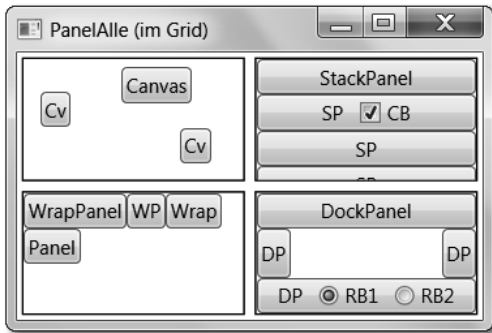


Abbildung 3.1 Alle fünf Layouts

3.1 Canvas

In früheren Anwendungen wurden Steuerelemente häufig fest positioniert. Dies sollten Sie, wie oben erläutert, möglichst vermeiden. Dennoch gibt es Situationen, in denen dies für einen Teil der Oberfläche oder die gesamte Oberfläche unumgänglich ist. Dann verwenden Sie einen `Canvas`.

Zur Positionierung in x-Richtung verwenden Sie dabei die *Attached Properties* `Canvas.Left` und `Canvas.Right`. Für die Positionierung in y-Richtung nehmen Sie `Canvas.Top` und `Canvas.Bottom`. Die Werte für diese Eigenschaften beziehen sich auf das logisch übergeordnete Element. Die Lage der Elemente in z-Richtung können Sie mithilfe der *Attached Property* `Panel.ZIndex` beeinflussen. Elemente mit unterschiedlichen Werten für `ZIndex` liegen vom Betrachter aus hintereinander bezüglich der Bildebene.

3.1.1 Positionierung

Im nachfolgenden Projekt *CanvasPositionen* wird eine Reihe von Buttons sowohl mithilfe von XAML als auch mithilfe von Programmcode positioniert (siehe Abbildung 3.2).



Abbildung 3.2 Positionierte Steuerelemente

Zunächst der Aufbau in XAML:

```
<Window ... Height="150" Width="320">
  <Canvas x:Name="cv">
    <Button x:Name="b1" Click="b1_Click">
      1: ohne Left und Top</Button>
    <Button Canvas.Left="30" Canvas.Top="18" x:Name="b2"
      Click="b2_Click">2: Left 30, Top 18</Button>
    <Button Canvas.Left="5" Canvas.Bottom="5">
      3: Left 5, Bottom 5</Button>
    <Button Canvas.Right="5" Canvas.Bottom="22" Panel.ZIndex="1">
      4: Right 5, Bottom 22, ZIndex 1</Button>
    <Button Canvas.Right="5" Canvas.Bottom="39">
      5: Right 5, Bottom 39</Button>
    <Button Canvas.Right="5" Canvas.Bottom="5" Panel.ZIndex="1"
      Click="b6_Click">6: Right 5, Bottom 5, ZIndex 1</Button>
  </Canvas>
</Window>
```

Die Steuerelemente stehen innerhalb des Canvas-Containers. Dieser füllt, als einziges Element, den gesamten Client-Bereich des Fensters aus. Er bekommt hier einen eindeutigen Namen, weil ihm später ein Steuerelement per Programmcode hinzugefügt wird. Die Steuerelemente sind dem Canvas untergeordnet.

Bei Button 1 gibt es keine Positionsangaben, daher liegt er ganz links oben. Bei Button 2 ist der Abstand vom linken und vom oberen Rand, bei Button 3 der Abstand vom linken und vom unteren Rand des Canvas festgelegt. Bei den restlichen Buttons 4 bis 6 wird mit unterschiedlichen Abständen vom rechten und vom unteren Rand gearbeitet.

Sollten sich einzelne Steuerelemente überlappen, so liegt das später erzeugte Element in z-Richtung über dem früher erzeugten Element. Dies sehen Sie bei den Buttons 1 und 2. Mit der *Attached Property* `Panel.ZIndex` können Sie darauf Einfluss nehmen. Ohne Angabe gilt `Panel.ZIndex = 0`. Ein positiver Wert »hebt« das Steuerelement in Richtung Betrachter, ein negativer Wert »versenkt« das Steuerelement in der Oberfläche. Daher überlappt Button 4 den Button 5, wird aber von Button 6 überlappt.

Die Lage können Sie auch per Programmcode beeinflussen, wie dies für die Buttons 1 und 2 durchgeführt wird:

```
private void b1_Click(object sender, RoutedEventArgs e)
{
    b1.SetValue(Canvas.LeftProperty, 10.0);
}
```

```

        b1.SetValue(Canvas.TopProperty, 10.0);
        b1.SetValue(Panel.ZIndexProperty, 1);
    }

    private void b2_Click(object sender, RoutedEventArgs e)
    {
        double left, top;
        left = (double)b2.GetValue(Canvas.LeftProperty);
        top = (double)b2.GetValue(Canvas.TopProperty);

        b2.SetValue(Canvas.LeftProperty, left + 10);
        b2.SetValue(Canvas.TopProperty, top + 10);
        b2.Content = "2: verschoben";
    }

```

Button 1 wird absolut verschoben, und zwar auf Position 10,10. Gleichzeitig wird er dem Betrachter entgegengehoben. Daher überlappt er nun Button 2. Die Methode `SetValue()` dient zum Verändern der Werte von *Dependency Properties*. `Canvas.LeftProperty` steht für die *Attached Property* `Canvas.Left`, und bei den anderen verhält es sich entsprechend. Die Werte für `Left` und `Top` müssen vom Typ `double` sein, der Wert für `ZIndex` vom Typ `int`.

Button 2 wird bei jedem Click relativ verschoben: um den Wert 10 nach rechts und um den Wert 10 nach unten. Die Methode `GetValue()` dient zum Ermitteln des aktuellen Werts der *Dependency Property*. Diese ist vom Typ `object`. Für die spätere Weiterverwendung ist daher eine explizite Typkonvertierung notwendig. Hier ist es wichtig, den richtigen Typ zu wählen. Die Eigenschaft `Content` steht für den Inhalt des Elements, also für die Aufschrift des Buttons.

Hinweis: Die ursprüngliche Position des Buttons 1 können Sie nicht über die Methode `GetValue()` ermitteln, da ihm die Eigenschaften `Left` und `Top` nicht per XAML zugewiesen wurden. Die Methode liefert in diesem Falle den Wert »nicht definiert«. Abhilfe: Setzen Sie `Left` und `Top` in XAML auf 0.

3.1.2 Elemente neu erzeugen

Im Projekt *CanvasPositionen* dient Button 6 zur Erstellung von weiteren Elementen per Programmcode:

```
private void b6_Click(object sender, RoutedEventArgs e)
{
    Button nb = new Button();
    nb.Content = "Neu";
    nb.SetValue(Canvas.RightProperty, 5.0);
    nb.SetValue(Canvas.BottomProperty, 80.0);
    cv.Children.Add(nb);
}
```

Mit dieser Technik können Sie Steuerelemente in allen Panel-Typen (Canvas, StackPanel ...) neu erzeugen. Zunächst wird eine neue Instanz des Steuerelements angelegt. Diese bekommt Eigenschaften, wie Aufschrift und Lage. Hier ist es wichtig, *double*-Werte zu wählen.

Anschließend wird sie der Auflistung *Children* des jeweiligen Panels mithilfe der Methode *Add()* hinzugefügt. Diese Auflistung verweist auf die untergeordneten Elemente eines Panels, hier also des Canvas.

3.1.3 Layout-Hierarchie

In einer Hierarchie von Layouts lassen sich mehrere Layouts, auch unterschiedlichen Typs, miteinander kombinieren. Dies wird im nachfolgenden Projekt *CanvasInCanvas* anhand von Canvas-Layout-Elementen gezeigt. Angaben wie *Canvas.Left* beziehen sich dabei immer auf das direkt übergeordnete Layout-Element.

Innerhalb eines Canvas, der den Client-Bereich des Fensters einnimmt, werden zwei untergeordnete Canvas positioniert. Diese beinhalten wiederum Buttons (siehe Abbildung 3.3).

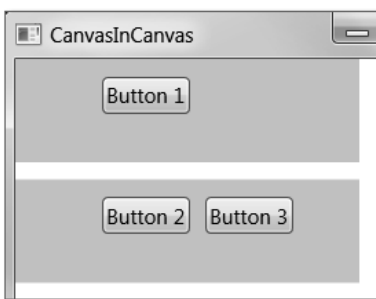


Abbildung 3.3 Untergeordnete Elemente

Zunächst der Aufbau in XAML:

```
<Window ...>
  <Canvas>
    <Canvas Width="200" Height="60" x:Name="cv1"
      Background="LightGray">
      <Button Canvas.Top="10" Canvas.Left="50">Button 1</Button>
    </Canvas>
    <Canvas Width="200" Height="60" x:Name="cv2" Canvas.Top="70"
      Background="LightGray">
      <Button Canvas.Top="10" Canvas.Left="50">Button 2</Button>
      <Button Canvas.Top="10" Canvas.Left="110" x:Name="b3"
        Click="b3_Click">Button 3</Button>
    </Canvas>
  </Canvas>
</Window>
```

Die beiden inneren Canvas `cv1` und `cv2` sind dem äußeren Canvas untergeordnet. Die Angabe `Canvas.Top` des unteren Canvas bezieht sich auf den äußeren Canvas. Die beiden Buttons 1 und 2 sind gleichartig positioniert. Ihre Angaben `Canvas.Top` und `Canvas.Left` beziehen sich allerdings einmal auf den ersten, einmal auf den zweiten inneren Canvas.

Hinweis: Die Eigenschaft `Background` für die Hintergrundfarbe ist vom Typ `Brush` (dt. *Pinzel*) und nicht vom Typ `Color`. In XAML werden häufig *Type Converter* genutzt, die eine passende Umwandlung vornehmen können. Mehr zum Typ `Brush` finden Sie in Abschnitt 9.4, »Pinzel«.

Die Unterordnung bezüglich der beiden Canvas können Sie auch per Programmcode verändern. Betätigt der Benutzer den dritten Button, so wechselt der Button vom unteren zum oberen Canvas:

```
private void b3_Click(object sender, RoutedEventArgs e)
{
    if (b3.Parent == cv2)
    {
        cv2.Children.Remove(b3);
        cv1.Children.Add(b3);
    }
}
```

Die Eigenschaft `Parent` liefert einen Verweis auf das übergeordnete Element. Falls es sich in diesem Fall um den unteren Canvas handelt, so wird die Unterordnung mithilfe

der Methode `Remove()` aufgelöst und eine neue Unterordnung zum oberen Canvas erstellt: Der Button wechselt nach oben (siehe Abbildung 3.4).

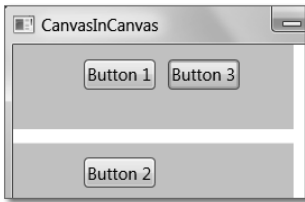


Abbildung 3.4 Hier wurde die Unterordnung geändert.

3.2 StackPanel

Ein `StackPanel` »stapelt« wortwörtlich die Steuerelemente: Diese werden einfach in einer Reihe untereinander oder nebeneinander angeordnet. Im Projekt *StackPanel-Anordnung* werden einige Möglichkeiten dargestellt.

Den Standard-Fall mit vertikaler Orientierung sehen Sie in Abbildung 3.5 links. In Abbildung 3.5 Mitte sind die Steuerelemente nebeneinander angeordnet. Zusätzlich wurde die Richtung der Reihe geändert: Die Steuerelemente werden von rechts nach links gestapelt. Sollte die Umgebung zu wenig Platz bieten, dann sind möglicherweise einige Elemente nicht erreichbar. Zur Abhilfe können Sie das `StackPanel` in ein Steuerelement vom Typ `ScrollViewer` einbetten (siehe Abbildung 3.5 rechts).

Falls die Steuerelemente innerhalb eines vertikal orientierten `StackPanel` keine eigene Breite haben, so nehmen sie die maximal verfügbare Breite in Anspruch. Entsprechendes gilt für die Höhe in einem horizontal orientierten `StackPanel`.

Alle Layouts können Sie in einer Hierarchie anordnen. Die drei beschriebenen `StackPanel` sind insgesamt wiederum in einem übergeordneten `StackPanel` mit horizontaler Orientierung eingebettet, diesmal in der Standard-Ablaufrichtung »von links nach rechts«.

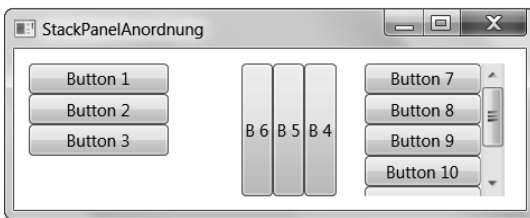


Abbildung 3.5 Verschiedene `StackPanel`

Der Aufbau in XAML:

```
<Window ...>
  <StackPanel Orientation="Horizontal">
    <StackPanel Width="100" Margin="10">
      <Button Click="neu_Click">Button 1</Button>
      <Button>Button 2</Button>
      <Button>Button 3</Button>
    </StackPanel>
    <StackPanel Orientation="Horizontal"
      FlowDirection="RightToLeft" ...> ... </StackPanel>
  <ScrollView VerticalScrollBarVisibility="Auto"...>
    <StackPanel> ... </StackPanel>
  </ScrollView>
</StackPanel>
</Window>
```

Die Orientierung wird mit der Eigenschaft `Orientation` festgelegt. Es gibt die Werte `Vertical` (übereinander) und `Horizontal` (nebeneinander). Die Eigenschaft `FlowDirection` bestimmt die Richtung der Reihe. Erlaubte Werte sind `LeftToRight` und `RightToLeft`. Der letztgenannte Wert macht nur Sinn, falls `Orientation` den Wert `Horizontal` hat.

Die Eigenschaft `VerticalScrollBarVisibility` des Elements vom Typ `ScrollView` können Sie auf den Wert `Auto` stellen. Dann wird sie nur eingeblendet, wenn sie benötigt wird, also wenn es »zu viele« Elemente gibt. Testen Sie dies im vorhandenen Projekt, indem Sie einfach die Fensterhöhe mit der Maus verändern.

Zur besseren Darstellung wurden die Breite sowie der Außenabstand der untergeordneten `StackPanel` festgelegt, und zwar über die Eigenschaften `Width` und `Margin`. Mehr zu diesen Eigenschaften erfahren Sie in Kapitel 4, »Steuerelemente«.

3.2.1 Elemente neu erzeugen

Im Projekt *StackPanelAnordnung* können Sie mithilfe der Buttons 1, 4 und 7 weitere Buttons per Programmcode erstellen:

```
private void neu_Click(object sender, RoutedEventArgs e)
{
    Button nb = new Button();
    nb.Content = "Neu";
}
```

```

Panel p = (sender as Button).Parent as Panel;
p.Children.Add(nb);
}

```

Es wird ein neuer Button erzeugt und beschriftet. Dann wird das jeweils übergeordnete Panel des geklickten Buttons (1, 4 oder 7) mithilfe der Eigenschaft `Parent` ermittelt. Der neu erzeugte Button wird der Auflistung `Children` dieses Panels mithilfe der Methode `Add()` als neues, untergeordnetes Element hinzugefügt.

Sie können feststellen, dass `StackPanels` mit vertikaler Orientierung nach unten erweitert werden. `StackPanels` mit horizontaler Orientierung werden nach rechts erweitert, unabhängig von der Richtung der Reihe.

3.3 WrapPanel

Ein `WrapPanel` ist, etwas vereinfacht ausgedrückt, ein `StackPanel` mit automatischem Zeilenumbruch. Die Steuerelemente werden der Reihe nach angeordnet. Falls es nicht mehr genügend Platz gibt, dann wird eine weitere Reihe aufgemacht. Es kommt hinzu, dass die Steuerelemente nur noch den notwendigen Platz einnehmen, nicht mehr den maximal verfügbaren Platz. Im Projekt *WrapPanelAnordnung* sehen Sie einige Möglichkeiten.

Das oberste `WrapPanel` in Abbildung 3.6 beinhaltet sieben Steuerelemente. Sie sind jeweils nur so breit wie nötig. Das letzte Element passte nicht mehr in die Reihe, daher wurde eine neue Reihe eröffnet. Die Richtung der Reihe weist im Standardfall von links nach rechts.

Beim zweiten `WrapPanel` in Abbildung 3.6 sind die fünf Steuerelemente von rechts nach links angeordnet. Es gibt auch hier eine zweite Reihe.

Im nächsten `WrapPanel` in Abbildung 3.6 wurde die Höhe einzelner Steuerelemente geändert. Dies hat Auswirkungen auf die anderen Steuerelemente, die sich aktuell in der gleichen Reihe befinden. Sie nehmen die gleiche Höhe an, falls sie keine eigene Höhe haben.

Sie können aber auch eine einheitliche Breite beziehungsweise Höhe für alle Steuerelemente festlegen. Dies sehen Sie am letzten `WrapPanel` in Abbildung 3.6.



Abbildung 3.6 Verschiedene WrapPanel

Der Aufbau in XAML:

```
<Window ...>
  <StackPanel>
    <WrapPanel Margin="5">
      <Button Click="neu_Click">Button 1</Button>
      <Button>B 2</Button>
      ...
      <Button>Button 7</Button>
    </WrapPanel>
    <WrapPanel FlowDirection="RightToLeft" ...> ... </WrapPanel>
    <WrapPanel ...>
      ...
      <Button Height="40">3: Height 40</Button>
      <Button Height="18">4: Height 18</Button>
      ...
    </WrapPanel>
    <WrapPanel ItemWidth="70" ItemHeight="30" ...> ...
  </WrapPanel>
</StackPanel>
</Window>
```

Wiederum wird über die Eigenschaft `FlowDirection` die Richtung der Reihe festgelegt. In einer Reihe haben alle Steuerelemente dieselbe Höhe. Sobald eines der Steuerelemente

seine Höhe ändert (Eigenschaft `Height`), ändert sich auch die Höhe der anderen Steuerelemente in der gleichen Reihe. Falls eine einheitliche Breite beziehungsweise Höhe gewünscht wird, können Sie dies über die Eigenschaften `ItemWidth` und `ItemHeight` des Panels bestimmen.

3.3.1 Elemente neu erzeugen

Mithilfe der verschiedenen Buttons mit der Aufschrift `Button 1` können Sie weitere Buttons im jeweiligen Panel per Programmcode erstellen. Die Methode `neu_Click()` aus dem Abschnitt 3.2.1, »Elemente neu erzeugen«, können Sie hier unverändert anwenden, da alle Layoutklassen von der gemeinsamen Basisklasse `Panel` abgeleitet sind.

3.3.2 Vertikale Orientierung

Auch in einem vertikal angeordneten `WrapPanel` gibt es zwei Möglichkeiten für die Richtung der Reihe. Dies sehen Sie im Projekt *WrapPanelOrientierung* in Abbildung 3.7.

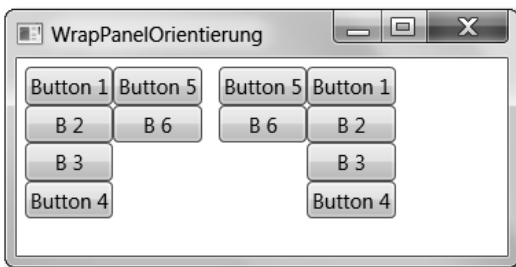


Abbildung 3.7 Zwei vertikale `WrapPanel`

Beide vertikalen Reihen bieten nicht genügend Platz, daher wird jeweils eine weitere vertikale Reihe eröffnet. Im zweiten Fall weist die Richtung der Reihe von rechts nach links. Der Aufbau in XAML:

```
<Window ...>
  <StackPanel Orientation="Horizontal">
    <WrapPanel Orientation="Vertical" ...> ... </WrapPanel>
    <WrapPanel Orientation="Vertical"
      FlowDirection="RightToLeft" Margin="5">
      ...
    </WrapPanel>
  </StackPanel>
</Window>
```

Auch in diesem Projekt können Sie über die beiden Buttons mit der Aufschrift **BUTTON 1** weitere Buttons erstellen.

3.4 DockPanel

In einem `DockPanel` ordnen Sie die Steuerelemente so an, wie Sie es aus vielen Anwendungen kennen: Oben wird zum Beispiel das Hauptmenü angedockt, unten eine Statuszeile, links und rechts gibt es weitere Bedienmöglichkeiten. In der verbleibenden Mitte wird der zu bearbeitende Inhalt dargestellt. Ein Beispiel sehen Sie im Projekt *DockPanelTBRL* (siehe Abbildung 3.8).



Abbildung 3.8 Reihenfolge »Top«, »Bottom«, »Left«, »Right«

Der Aufbau in XAML:

```
<Window ...>
  <DockPanel>
    <Button DockPanel.Dock="Top">1: Top</Button>
    <Button DockPanel.Dock="Bottom">2: Bottom</Button>
    <Button DockPanel.Dock="Left">3: L</Button>
    <Button DockPanel.Dock="Left">4: L</Button>
    <Button DockPanel.Dock="Left">5: L</Button>
    <Button DockPanel.Dock="Right">6: Right</Button>
    <TextBlock Margin="10">Inhalt</TextBlock>
  </DockPanel>
</Window>
```

Den Steuerelementen innerhalb eines `DockPanel`s wird die *Attached Property* `DockPanel.Dock` zugeordnet. Die Werte für die Eigenschaft stammen aus der Enumeration `Dock: Top, Bottom, Left und Right`.

Wichtig ist die Reihenfolge: Der erste Button wird oben angeordnet und erstreckt sich über die gesamte Breite. Beim zweiten Button gilt das Gleiche für unten. Sie können mehrere Steuerelemente im gleichen Bereich andocken: Die Buttons 3 bis 5 werden nebeneinander links dargestellt. Diese Buttons können sich allerdings nicht mehr bis ganz oben oder ganz unten erstrecken, da dieser Platz bereits durch die Buttons 1 und 2 belegt ist. Beim Button 6 gilt das Gleiche für rechts.

Das letzte Element, hier ein `TextBlock`, wird gar nicht andocked. Daher füllt es den verbleibenden Platz. Mehr zum Element `TextBlock` folgt in Abschnitt 4.3.2.

3.4.1 DockPanel in Hierarchie

Im nachfolgenden Projekt *DockPanelLRT* sehen Sie Elemente, die jeweils wiederum andere Elemente enthalten (siehe Abbildung 3.9).

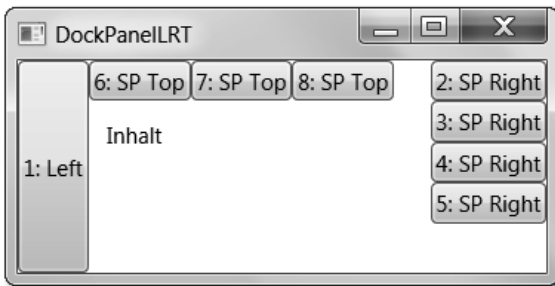


Abbildung 3.9 DockPanel und StackPanel

Die Elemente wurden in der Reihenfolge links, rechts und oben erzeugt. Daher steht für das obere Element nicht mehr die gesamte Breite zur Verfügung. Das rechte und das obere Element ist jeweils ein `StackPanel`, das weitere Elemente beinhaltet. Der Aufbau in XAML:

```
<Window ...>
  <DockPanel>
    <Button DockPanel.Dock="Left">1: Left</Button>
    <StackPanel DockPanel.Dock="Right">
      <Button>2: SP Right</Button>
      ...
    </StackPanel>
    <StackPanel DockPanel.Dock="Top" Orientation="Horizontal">
      <Button>6: SP Top</Button>
      ...
    </StackPanel>
    <TextBlock>Inhalt</TextBlock>
  </DockPanel>
</Window>
```

```

    </StackPanel>
    <TextBlock Margin="10">Inhalt</TextBlock>
  </DockPanel>
</Window>

```

Es gibt drei Elemente, die mit der *Attached Property* `DockPanel` festgelegt wurden: ein `Button` und zwei `StackPanels`.

3.4.2 Elemente neu erzeugen

Im nachfolgenden Projekt *DockPanelLastChild* wird erläutert, wie Sie einem `DockPanel` neue Elemente an der gewünschten Stelle hinzufügen (siehe Abbildung 3.10).



Abbildung 3.10 Elemente neu erzeugen

Die Elemente wurden in der Reihenfolge links, rechts, oben, unten erzeugt. Es wurden zwei weitere Elemente mit den Werten `Top` und `Bottom` für die Eigenschaft `DockPanel.Dock` hinzugefügt.

Zunächst der Aufbau in XAML:

```

<Window ...>
  <DockPanel LastChildFill="False"> ...
    <Button DockPanel.Dock="Top" Click="neu_Click">
      3: Top</Button>
    <Button DockPanel.Dock="Bottom" Click="neu_Click">
      4: Bottom</Button>
  </DockPanel>
</Window>

```

Mithilfe der Eigenschaft `LastChildFill` können Sie bestimmen, ob das letzte Element den verbleibenden Platz im Fenster füllt (Standardwert = `True`) oder nicht (Wert = `False`). Der

Aufbau der Methode `neu_Click()` ist etwas aufwendiger als bei den anderen Panels, da Sie zunächst die Position ermitteln müssen:

```
private void neu_Click(object sender, RoutedEventArgs e)
{
    Button sb = sender as Button;
    Object dp = sb.GetValue(DockPanel.DockProperty);

    Button nb = new Button();
    nb.Content = "Neu";
    nb.SetValue(DockPanel.DockProperty, dp);

    Panel p = sb.Parent as Panel;
    p.Children.Add(nb);
}
```

Mithilfe der Methode `GetValue()` wird der Wert der *Dependency Property* `DockPanel.DockProperty` ermittelt, die die Position des auslösenden Buttons angibt. Dies wird die Position des neuen Buttons, die mithilfe von `SetValue()` festgelegt wird.

3.5 Grid

Ein Grid dient zur regelmäßigen, übersichtlichen Anordnung der Elemente in einem Raster. Sie legen zunächst die Anzahl der Zeilen und Spalten fest. Den einzelnen Steuerelementen ordnen Sie anschließend die Koordinaten ihrer Zelle im Grid zu, die aus der Nummer der Zeile und der Nummer der Spalte bestehen. Ein erstes Beispiel folgt im Projekt *GridAnordnung* (siehe Abbildung 3.11).



Abbildung 3.11 Ein Grid mit drei Zeilen und zwei Spalten

Der Aufbau in XAML:

```
<Window ...>
  <Grid x:Name="gr">
    <Grid.RowDefinitions>
      <RowDefinition />
      <RowDefinition />
      <RowDefinition />
    </Grid.RowDefinitions>
    <Grid.ColumnDefinitions>
      <ColumnDefinition />
      <ColumnDefinition />
    </Grid.ColumnDefinitions>
    <Button Grid.Row="0" Grid.Column="0" Click="b1_Click">
      1: 0, 0</Button>
    <Button Grid.Row="0" Grid.Column="1" Click="b2_Click">
      2: 0, 1</Button>
    <Button Grid.Row="1" Grid.Column="1">3: 1, 1</Button>
    <Button Grid.Row="2" Grid.Column="0" Click="b4_Click">
      4: 2, 0</Button>
  </Grid>
</Window>
```

Die Gestaltung der einzelnen Zeilen und damit auch ihre Anzahl wird in der Auflistung `Grid.RowDefinitions` festgelegt. Das Gleiche gilt für die Spalten in der Auflistung `Grid.ColumnDefinitions`. Die Zuordnung zu den einzelnen Zellen des Grids geschieht mithilfe der *Attached Properties* `Grid.Row` und `Grid.Column`. Die Zählung beginnt bei 0. Dies ist auch der Standardwert. Bei Button 1 hätten Sie also die Zuordnung weglassen können.

3.5.1 Elemente neu erzeugen

Im Projekt *GridAnordnung* dient BUTTON 1 zur Erzeugung eines neuen Elements innerhalb einer vorhandenen Zelle. BUTTON 2 fügt eine neue Spalte mit einem weiteren Element hinzu. BUTTON 4 fügt eine neue Zeile hinzu, ebenfalls mit einem weiteren Element. Damit kann der Benutzer das Grid verändern, sodass es zum Beispiel wie in Abbildung 3.12 aussieht.

Der zugehörige Programmcode:

```
private void b1_Click(...)
{
    Button nb = new Button();
```

```

nb.Content = "Neu";
nb.SetValue(Grid.RowProperty, 2);
nb.SetValue(Grid.ColumnProperty, 1);
gr.Children.Add(nb);
}

private void b2_Click(...)
{ ...
    nb.SetValue(Grid.RowProperty, 0);
    gr.ColumnDefinitions.Add(new ColumnDefinition());
    nb.SetValue(Grid.ColumnProperty,
        gr.ColumnDefinitions.Count - 1);
    gr.Children.Add(nb);
}

private void b4_Click(...)
{ ...
    gr.RowDefinitions.Add(new RowDefinition());
    nb.SetValue(Grid.RowProperty, gr.RowDefinitions.Count - 1);
    nb.SetValue(Grid.ColumnProperty, 0);
    gr.Children.Add(nb);
}

```



Abbildung 3.12 Ein Grid mit vier Zeilen und vier Spalten

Mithilfe der Methode `SetValue()` setzen Sie die *Dependency Properties* `Grid.RowProperty` und `Grid.ColumnProperty` auf die gewünschten Werte. Zur Erzeugung einer neuen Spalte wird der Auflistung `ColumnDefinitions` mithilfe der Methode `Add()` ein neues Element hinzugefügt. Die Eigenschaft `ColumnDefinitions.Count` liefert die aktuelle Anzahl der Spalten. Damit können Sie die Position für das neue Element in der neuen Spalte bestimmen. Entsprechendes gilt für die Auflistung `Grid.RowDefinitions`.

3.5.2 Elemente über mehrere Zellen

Elemente können sich über mehrere Zellen erstrecken. Dies wird ähnlich wie in HTML-Tabellen gelöst. Ein Beispiel sehen Sie im Projekt *GridSpannweite* (siehe Abbildung 3.13).



Abbildung 3.13 Elemente über mehrere Zellen

Der Aufbau in XAML:

```
<Window ...>
  <Grid>
    <Grid.RowDefinitions>
      <RowDefinition />
      <RowDefinition />
      <RowDefinition />
    </Grid.RowDefinitions>
    <Grid.ColumnDefinitions>
      <ColumnDefinition />
      <ColumnDefinition />
    </Grid.ColumnDefinitions>
    <Button Grid.Row="0" Grid.Column="0" Grid.ColumnSpan="2">
      1: 0, 0, ColumnSpan 2</Button>
    <Button Grid.Row="1" Grid.Column="0">2: 1, 0</Button>
    <Button Grid.Row="1" Grid.Column="1" Grid.RowSpan="2">
      3: 1, 1, RowSpan 2</Button>
    <Button Grid.Row="2" Grid.Column="0">4: 2, 0</Button>
  </Grid>
</Window>
```

Zunächst wird das Grundgerüst aufgebaut, das aus drei Zeilen und zwei Spalten besteht. Anschließend ordnen Sie die Elemente wie gewohnt über die *Attached Properties* `Grid.Row` und `Grid.Column` zu. Die *Attached Properties* `Grid.ColumnSpan` und `Grid.RowSpan` dienen dazu, die Elemente von der angegebenen Zelle aus über die entsprechende Anzahl

an Zellen zu »spannen«. Der Inhalt der Zelle 0,1 stammt somit aus der Zelle 0,0. Der Inhalt der Zelle 2,1 stammt entsprechend aus der Zelle 1,1.

3.5.3 Größe der Zellen festlegen

Bisher waren alle Zellen gleich groß. Natürlich haben Sie auch die Möglichkeit, die Größe der Zellen selbst zu bestimmen. Die Zellgröße kann sich nach dem Inhalt richten, sie kann einen bestimmten Wert annehmen oder in einem festen Verhältnis zur Größe der anderen Zellen stehen.

Im nachfolgenden Projekt *GridAuto* wird die Höhe beziehungsweise die Breite bestimmter Zellen nach dem Inhalt ausgerichtet (siehe Abbildung 3.14).



Abbildung 3.14 Höhe beziehungsweise Breite automatisch

Der Aufbau in XAML:

```
<Window ...>
  <Grid>
    <Grid.RowDefinitions>
      <RowDefinition Height="Auto" />
      <RowDefinition />
      <RowDefinition />
    </Grid.RowDefinitions>
    <Grid.ColumnDefinitions>
      <ColumnDefinition />
      <ColumnDefinition Width="Auto" />
    </Grid.ColumnDefinitions>
    ...
  </Grid>
</Window>
```

Dank des Wertes `Auto` für die Eigenschaft `Height` richtet sich die Höhe der ersten Zeile nach der Höhe des Textes auf dem Element. Das Entsprechende gilt für die Breite (`Width`) der zweiten Spalte, die sich nach der Breite des Textes richtet. Die gilt unabhängig von den Einstellungen für `Grid.RowSpan` und `Grid.ColumnSpan`.

Im nachfolgenden Projekt *GridWert* stehen die Höhen der Zeilen in einem bestimmten Verhältnis zueinander. Dagegen richtet sich die Breite bestimmter Spalten nach einem Wert (siehe Abbildung 3.15).

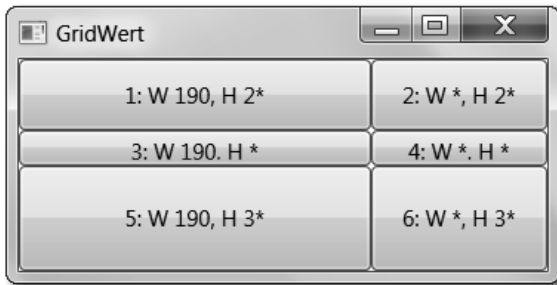


Abbildung 3.15 Höhe im Verhältnis, Breite mit Wert

Der Aufbau in XAML:

```
<Window ...>
  <Grid>
    <Grid.RowDefinitions>
      <RowDefinition Height="2*" />
      <RowDefinition Height="*" />
      <RowDefinition Height="3*" />
    </Grid.RowDefinitions>
    <Grid.ColumnDefinitions>
      <ColumnDefinition Width="190" />
      <ColumnDefinition Width="*" />
    </Grid.ColumnDefinitions>
    ...
  </Window>
```

Die Höhen der Zeilen stehen im Verhältnis 2 zu 1 zu 3. Ein einfacher `*` steht bei einer Verhältnisangabe für 1*. Die Breite der linken Spalte wurde mit dem Wert 190 festgelegt, für die rechte Spalte verbleibt der Rest. Auch hier sehen Sie wieder die Parallelen zu HTML.

3.5.4 Die Größe der Zellen flexibel gestalten

Sie möchten dem Benutzer Ihrer Anwendung die Möglichkeit geben, Zeilenhöhe und Spaltenbreite zu verändern? Dies ermöglichen Ihnen Elemente vom Typ `GridSplitter`. Im nachfolgenden Projekt *GridVerschieben* wird Ihnen ein Beispiel gezeigt (siehe Abbildung 3.16).



Abbildung 3.16 Verschobenenes Grid

Ursprünglich waren die Zellen mit den Buttons gleich groß. Der Benutzer hat aber bereits die beiden schwarz hervorgehobenen `GridSplitter` genutzt, um Höhe und Breite zu verstellen. Der Aufbau in XAML:

```
<Window ...>
  <Grid>
    <Grid.RowDefinitions>
      <RowDefinition />
      <RowDefinition MinHeight="10" />
      <RowDefinition Height="Auto" />
      <RowDefinition MinHeight="10" />
    </Grid.RowDefinitions>
    <Grid.ColumnDefinitions>
      <ColumnDefinition MinWidth="10" />
      <ColumnDefinition Width="Auto" />
      <ColumnDefinition MinWidth="10" />
    </Grid.ColumnDefinitions>
    <Button Grid.Row="0" Grid.Column="0">
      1: 0, 0, MinW 10</Button>
    <GridSplitter Grid.Row="0" Grid.Column="1" Grid.RowSpan="4"
      ResizeBehavior="PreviousAndNext" Width="3"
      Background="Black" />
    <Button Grid.Row="0" Grid.Column="2">
```

```

        2: 0, 2, MinW 10</Button>
<Button Grid.Row="1" Grid.Column="0">
    3: 1, 0, MinH 10</Button>
<Button Grid.Row="1" Grid.Column="2">4: 1, 2</Button>
<GridSplitter Grid.Row="2" Grid.Column="0"
    Grid.ColumnSpan="3" ResizeBehavior="PreviousAndNext"
    HorizontalAlignment="Stretch" Height="3"
    Background="Black" />
<Button Grid.Row="3" Grid.Column="0">
    5: 3, 0, MinH 10</Button>
<Button Grid.Row="3" Grid.Column="2">6: 3, 2</Button>
</Grid>
</Window>

```

Das Grid umfasst vier Zeilen und drei Spalten. Eine Zeile und eine Spalte werden jeweils von einem GridSplitter eingenommen. Für die beiden Nachbarzeilen (1 und 3) der Zeile mit dem GridSplitter (2) ist es sinnvoll, eine Minimalhöhe zu vereinbaren. Ansonsten würden sie bei einer extremen Verschiebung des GridSplitters gänzlich verschwinden. Entsprechend haben die Nachbarspalten (0 und 2) der Spalte mit dem GridSplitter (1) eine Minimalbreite.

Der Spalten-GridSplitter in Zelle 0,1 geht über die gesamte Spalte (`Grid.RowSpan=4`). Zur besseren Bedienung hat er eine Breite von 3 und ist schwarz.

Der Zeilen-GridSplitter in Zelle 2,0 geht über die gesamte Zeile (`Grid.ColumnSpan=3`). Zur besseren Bedienung hat er eine Höhe von 3, ist schwarz und dehnt sich über die gesamte Breite aus (`HorizontalAlignment = Stretch`).

Bei beiden GridSplitters ist die Eigenschaft `ResizeBehavior` mit dem gleichen Wert festgelegt. Damit legen Sie fest, welche Zeilen beziehungsweise Spalten ihre Größe verändern. Der Wert stammt aus der Enumeration `GridResizeBehavior`. Erlaubt sind:

- ▶ `PreviousAndNext`: Zeile über und unter dem GridSplitter beziehungsweise Spalte links und rechts vom GridSplitter (gilt hier)
- ▶ `CurrentAndNext`: Zeile beziehungsweise Spalte des GridSplitters und Zeile darunter beziehungsweise Spalte rechts
- ▶ `PreviousAndCurrent`: Zeile beziehungsweise Spalte des GridSplitters und Zeile darüber beziehungsweise Spalte links
- ▶ `BasedOnAlignment`: Die Größenänderung richtet sich nach den Alignment-Eigenschaften.

3.6 Layout-Kombination

Es folgt ein Beispiel für die Kombination verschiedener Layouts (Projekt *PanelKombi*). Bei einigen Steuerelementen wurde auch das Innere mithilfe eines Layouts gestaltet. Zunächst sehen Sie in Abbildung 3.17 die Anwendung in Originalgröße nach dem Start.

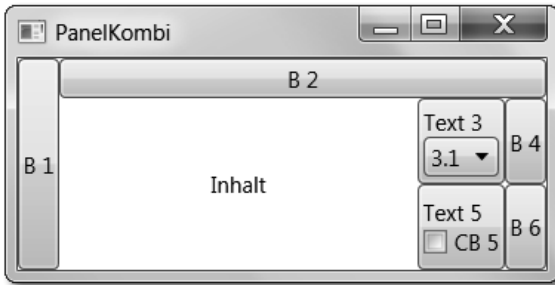


Abbildung 3.17 Nach dem Start

Innerhalb der Button-Steuerelemente 3 und 5 sind unter dem Text eine ComboBox beziehungsweise eine CheckBox angeordnet. Nach einer Verkleinerung durch den Benutzer kann die Anwendung aber auch so aussehen wie in Abbildung 3.18.

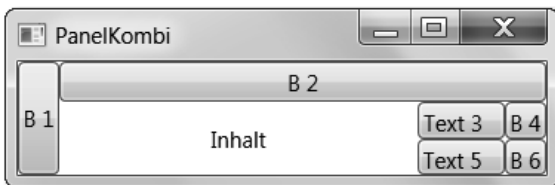


Abbildung 3.18 Nach der Verkleinerung

Der Aufbau in XAML:

```
<Window ...>
  <DockPanel>
    <Button DockPanel.Dock="Left">B 1</Button>
    <Button DockPanel.Dock="Top">B 2</Button>
    <Grid DockPanel.Dock="Right">
      <Grid.RowDefinitions>
        <RowDefinition />
        <RowDefinition />
      </Grid.RowDefinitions>
      <Grid.ColumnDefinitions>
```

```

        <ColumnDefinition />
        <ColumnDefinition />
    </Grid.ColumnDefinitions>
    <Button Grid.Row="0" Grid.Column="0">
        <StackPanel>
            <TextBlock>Text 3</TextBlock>
            <ComboBox>
                <ComboBoxItem IsSelected="True">3.1</ComboBoxItem>
                <ComboBoxItem>3.2</ComboBoxItem>
                <ComboBoxItem>3.3</ComboBoxItem>
            </ComboBox>
        </StackPanel>
    </Button>
    <Button Grid.Row="0" Grid.Column="1">B 4</Button>
    <Button Grid.Row="1" Grid.Column="0">
        <StackPanel>
            <TextBlock>Text 5</TextBlock>
            <CheckBox>CB 5</CheckBox>
        </StackPanel>
    </Button>
    <Button Grid.Row="1" Grid.Column="1">B 6</Button>
</Grid>
<TextBlock HorizontalAlignment="Center"
    VerticalAlignment="Center">Inhalt</TextBlock>
</DockPanel>
</Window>

```

Das Hauptelement des Layouts ist ein DockPanel. Es beinhaltet drei gedockte Elemente (zwei Buttons und ein Grid) sowie einen TextBlock. Das Grid hat zwei Zeilen und zwei Spalten. Die Elemente der ersten Spalte sind Buttons, die mithilfe eines StackPanels gestaltet wurden. Diese StackPanels beinhalten je einen TextBlock und ein weiteres Element. Mehr zu Aufbau und Eigenschaften der Steuerelemente erfahren Sie im folgenden Kapitel.

Index

>	69	Absatz (Forts.)	
<	69	<i>erster/letzter</i>	420
* Größenangabe	427	<i>formatieren</i>	441
< anzeigen	69	<i>hinzufügen</i>	420
> anzeigen	69	<i>Inhalt</i>	431
2D-Grafik	251	<i>zusammenhalten</i>	419
<i>für 3D-Oberfläche</i>	324	Abschnitt	416, 420
3D-Grafik	301	<i>ändern</i>	422
<i>aus externen Daten</i>	306	<i>formatieren</i>	420
<i>Drehung</i>	336	<i>hinzufügen</i>	421
<i>Form des Objekts</i>	305	AccelerationRatio	
<i>gemeinsame Punkte</i>	311	<i>AnimationTimeline</i>	352
<i>Grundelement</i>	302	AcceptsReturn	
<i>Kamera</i>	304, 311	<i>TextBox</i>	82
<i>Kameralage</i>	308, 313	Add()	
<i>Landschaft</i>	343	<i>AdornerLayer</i>	299
<i>Leinwand</i>	304	<i>Blocks</i>	420
<i>Licht</i>	305, 309, 314	<i>Children</i>	35
<i>Material</i>	305, 327	<i>ColumnDefinitions</i>	47
<i>mehrere Körper</i>	319	<i>eigene Auflistung</i>	237
<i>mehrere Körper, mit Ereignis</i>	323	<i>Inlines</i>	78
<i>mit 2D-Oberfläche</i>	324	<i>InputGestureCollection</i>	159
<i>mit Ereignis</i>	321	<i>Int32Collection</i>	308
<i>mit Steuerelementen</i>	324	<i>Items</i>	88, 99
<i>Modell</i>	319	<i>PathSegmentCollection</i>	267
<i>Rotationsgruppe</i>	341	<i>Point3DCollection</i>	308
<i>Rückseite</i>	310	<i>PointCollection</i>	257
<i>Skalierung</i>	333	<i>ValidationRules</i>	231
<i>Textur</i>	326, 329	AddChild()	
<i>Transformation</i>	332	<i>IAddChild</i>	445
<i>Transformationsgruppe</i>	339	AddedLength	
<i>Verschiebung</i>	334	<i>TextChange</i>	82
<i>Vorderseite</i>	306	Adobe Flash	15
3D-Würfel	308	AdornedElement	
A		<i>Adorner</i>	300
Abhängigkeitseigenschaft	24	<i>RenderSize</i>	300
Absatz	85, 416, 418	Adorner	299
<i>ändern</i>	420	<i>AdornedElement</i>	300
<i>ausrichten</i>	419	<i>OnRender()</i>	300
<i>Beginn</i>	443	AdornerLayer	
<i>einrücken</i>	419	<i>Add()</i>	299
<i>Ende</i>	443	<i>GetAdornerLayer()</i>	299
		AlignCenter	
		<i>EditingCommands</i>	441

- AlignLeft
 - EditingCommands* 441
- AlignRight
 - EditingCommands* 441
- AllowsTransparency
 - Window 191
- Alphakanal 274, 292
- Alt
 - ModifierKeys* 157
- AmbientLight 305, 314
- Angehängte Eigenschaft 25
- Angehängtes Ereignis 29
- Angle
 - AxisAngleRotation3D* 336
 - RotateTransform* 283, 284, 286
- AngleVelocity
 - ManipulationVelocities* 151
- AngleX
 - SkewTransform* 288
- AngleY
 - SkewTransform* 288
- Animation 347
 - anhalten* 361
 - beenden* 361
 - Beginn* 348
 - Beschleunigung* 352
 - der 3D-Rotation* 371
 - der Bewegung* 348
 - der Farbe* 364, 377
 - der Größe* 358
 - der Transformation* 362
 - der Transparenz* 357
 - der Zeichenkette* 378
 - Ende* 348
 - entlang Pfad* 383
 - federt* 383
 - fortsetzen* 361
 - gemeinsame Eigenschaften* 356
 - Geschwindigkeit* 349
 - Geschwindigkeit ändern* 361, 371
 - mit Event Trigger steuern* 369
 - nach Funktion* 379
 - nach Sinus* 383
 - nach Spline* 375
 - ohne Programmcode* 365
 - per Programmcode* 358
 - Rückkehr* 352
 - springen* 371
- Animation (Forts.)
 - sprunghafte* 375, 383
 - starten* 349, 357, 361, 366
 - Startverzögerung* 352
 - steuern* 360
 - unterteilen* 374
 - variable Geschwindigkeit* 374
 - Veränderung* 348
 - Wiederholung* 352
 - Zeitdauer* 349, 351
 - Zieleigenschaft* 357, 359
 - Zielelement* 358, 359
 - zu Punkt springen* 361
 - zum Ende springen* 361
 - zuordnen* 349, 357, 361, 371
- AnimationTimeline 347, 348, 351
- Anwendung
 - Anzahl Aufrufparameter* 169
 - Aufbau* 161
 - Aufruf von Kommandozeile* 167
 - Aufrufkommando* 169
 - Aufrufparameter* 166, 169
 - Installation* 171
 - Minimal-Aufbau* 161
 - Ressource* 170
 - Rückgabeparameter* 166, 170
 - schließen* 153
 - Startdatei* 166
 - starten* 163
 - wird beendet* 166, 167
 - wird gestartet* 166, 167
- App.xaml 165, 168, 173
- App.xaml.cs 166, 168
- AppendBreak()
 - PromptBuilder* 404, 406
- AppendText()
 - PromptBuilder* 404
- AppendTextWithHint()
 - PromptBuilder* 404, 406
- Application 163
 - Current.Resources* 223
 - Exit* 166, 167
 - Resources* 175, 215
 - Run()* 163
 - Startup* 166, 167
 - StartupUri* 166
- ApplicationCommands 152
 - Close* 153

ApplicationExitCode		
ExitEventArgs	170	
ArcSegment	266, 358	
IsLargeArc	267	
IsStroked	266	
Point	266	
RotationAngle	267	
Size	266	
SweepDirection	267	
Args		
Count()	169	
StartupEventArgs	169	
Arrange()	448	
Asterisk		
SystemSounds	391	
Attached Event	29, 229	
Attached Property	25	
Audio-Ausgabe	387, 391	
Aufklapplement	113	
Auflistung	85	
Anzahl Einträge	88	
eigener Typ	236	
Eintrag	85	
Eintrag einfügen	88	
Eintrag hinzufügen	88	
Eintrag löschen	89, 92	
leeren	89, 101	
Nummer eines Eintrags	99	
Auflistungstyp	85	
Aufrufparameter	166	
Aufzählung		
in Dokument	416, 422	
Ausrichtung	419	
horizontal	52, 66	
vertikal	66	
Ausrichtung des Inhalts		
horizontal	66	
vertikal	66	
Ausstattung	295, 396	
Auswahl		
einstellen	74	
Auswahlelement	71, 85	
AutoGenerateColumns		
DataGrid	245	
AutoReverse		
AnimationTimeline	352	
AutoToolTipPlacement		
BottomRight	107	
AutoToolTipPlacement (Forts.)		
Slider	107	
TopLeft	107	
Axis		
AxisAngleRotation3D	336	
AxisAngleRotation		
animieren	373	
AxisAngleRotation3D	336	
B		
BackEase	382	
Background	36, 57, 292	
BackgroundWorker	103	
DoWork()	104	
IsBusy	104	
ProgressChanged	104	
ReportProgress()	104	
RunWorkerAsync()	104	
WorkerReportsProgress	103	
BackMaterial		
GeometryModel3D	310	
Balance		
MediaPlayer	392	
BasedOn		
Style	207	
BasedOnAlignment		
GridResizeBehavior	52	
BaselineAlignment		
Inline	433	
Batch-Datei	168	
Baumstruktur	21	
Bézier-Kurve	264	
Bedienbarkeit	58	
Beep		
SystemSounds	391	
Begin()		
Storyboard	357, 361	
BeginAnimation()	349	
BeginStoryboard	366, 371	
BeginStoryboardName	371	
BeginTime		
AnimationTimeline	352	
TimeSeekOrigin	361	
Benannter Style	202	
vererben	206	
Benutzeroberfläche		
wechseln	221	

Berühren		
<i>beenden</i>	148	
<i>beginnen</i>	148	
Berührung		
Bildschirm	147	
Punkt	148	
Berührungsempfindlich	145	
Beschriftung	74	
<i>einfach</i>	74	
<i>formatierte</i>	76	
Bevel		
<i>PenLineJoin</i>	258	
BezierSegment	264	
Bild	134	
<i>als Farbpinsel</i>	279	
<i>Datenquelle</i>	134	
<i>Dehnung</i>	134	
<i>Größe</i>	134	
<i>in Dokument</i>	432, 436	
<i>neu laden</i>	173	
Bilddatei		
<i>in Geometrie</i>	271	
Bildlaufleiste	108	
Binding	226	
<i>DataTrigger</i>	249	
<i>ElementName</i>	226	
<i>GetBinding()</i>	230	
<i>Mode</i>	227	
<i>Path</i>	226	
<i>Source</i>	234	
<i>UpdateSourceTrigger</i>	228	
<i>ValidationRules</i>	230	
BindingExpression		
<i>UpdateSource()</i>	228	
BindingOperations	227	
<i>ClearAllBindings()</i>	227	
<i>ClearBinding()</i>	227	
<i>SetBinding()</i>	227	
Bindung		
<i>an Geste</i>	156	
<i>an Maus</i>	156	
<i>an Taste</i>	156	
Bitmapeffekt	296	
BitmapImage	173	
BitmapSource	173	
BlackoutDates		
<i>Calendar</i>	132	
Blickrichtung	305	
Block	85, 416	
<i>alle löschen</i>	428	
<i>BreakPageBefore</i>	421	
<i>gruppieren</i>	420	
<i>TextAlignment</i>	419	
<i>Verweis auf Element</i>	420	
BlockCollection	416	
Blocks	416	
<i>Add()</i>	420	
<i>Clear()</i>	428	
<i>ElementAt()</i>	420	
<i>FirstBlock</i>	420	
<i>LastBlock</i>	420	
Blocksatz	419	
BlockUIContainer	416, 429	
<i>ändern</i>	430	
<i>hinzufügen</i>	430	
BlurEffect	296	
<i>Radius</i>	298	
BlurRadius		
<i>DropShadowEffect</i>	298	
Bogensegment	266	
<i>Bogen sichtbar</i>	266	
<i>Drehrichtung</i>	267	
<i>Drehwinkel</i>	267	
<i>mehr als 180 Grad</i>	267	
<i>Radius</i>	266	
<i>Zielpunkt</i>	266	
Bold	77	
<i>FontWeights</i>	58	
<i>Inline-Typ</i>	431	
bool?	70	
BooleanAnimationUsingKeyFrames	374	
Border	109	
<i>BorderBrush</i>	109	
<i>BorderThickness</i>	109	
<i>CornerRadius</i>	109	
BorderBrush		
<i>Border</i>	109	
<i>Tabelle</i>	427	
BorderThickness		
<i>Border</i>	109	
<i>Tabelle</i>	427	
Both		
<i>TickPlacement</i>	105	
Bottom		
<i>Canvas</i>	32	

BottomRight			
<i>AutoToolTipPlacement</i>	107	Calendar (Forts.)	
<i>TickPlacement</i>	105	<i>SelectedDate</i>	131
BounceEase	383	<i>SelectedDates</i>	130
<i>Bounces</i>	383	<i>SelectedDatesChanged</i>	130
<i>Bounciness</i>	383	<i>SelectionMode</i>	128
Bounces		CalendarBlackoutDatesCollection	132
<i>BounceEase</i>	383	CalendarDateRange	132
Bounciness		CalendarSelectionMode	128
<i>BounceEase</i>	383	Camera	304
Box		Cancel	
<i>TextMarkerStyle</i>	423	<i>CancelEventArgs</i>	167
BreakPageBefore		CancelEventArgs	167
<i>Block</i>	421	<i>Cancel</i>	167
Browser	136	CanExecute	
<i>angezeigte Seite</i>	136	<i>CommandBinding</i>	153
<i>darf sich in History bewegen</i>	138	CanGoBack	
<i>in History bewegen</i>	138	<i>NavigationService</i>	187
<i>navigieren zu HTML-Code</i>	136	CanGoBack()	
<i>navigieren zu URI</i>	136	<i>WebBrowser</i>	138
<i>Seite ganz geladen</i>	136	CanGoForward	
<i>Seite gewechselt</i>	136	<i>NavigationService</i>	187
Browseranwendung	192	CanGoForward()	
Brush	57, 109, 273	<i>WebBrowser</i>	138
<i>DiffuseMaterial</i>	305, 330	CanMinimize	
<i>GeometryDrawing</i>	272	<i>SizeMode</i>	178
Brushes		CanResize	
<i>Transparent</i>	109	<i>SizeMode</i>	178
Button	66	CanResizeWithGrip	
<i>Basistyp</i>	209	<i>SizeMode</i>	178
<i>dauernd betätigen</i>	68	CanUserAddRows	
ButtonBase	209	<i>DataGrid</i>	244
ButtonState		CanUserDeleteRows	
<i>MouseButtonEventArgs</i>	144	<i>DataGrid</i>	244
By		CanUserReorderColumns	
<i>AnimationTimeline</i>	348	<i>DataGrid</i>	244
Byte	274	CanUserResizeColumns	
ByteAnimation	347	<i>DataGrid</i>	244
ByteAnimationUsingKeyFrames	374	CanUserResizeRows	
		<i>DataGrid</i>	244
		CanUserSortColumns	
		<i>DataGrid</i>	244
C		Canvas	32
Calendar	128	<i>Bottom</i>	32
<i>BlackoutDates</i>	132	<i>Left</i>	32
<i>DisplayDate</i>	130	<i>LeftProperty</i>	34
<i>DisplayDateEnd</i>	132	<i>Right</i>	32
<i>DisplayDateStart</i>	132	<i>Top</i>	32
<i>FirstDayOfWeek</i>	132	<i>TopProperty</i>	34

Cascading Style Sheets	202	Clear()	
Cell		<i>Blocks</i>	428
<i>DataGridSelectionUnit</i>	244	<i>Items</i>	89, 101
CellOrRowHeader		ClearAllBindings()	
<i>DataGridSelectionUnit</i>	244	<i>BindingOperations</i>	227
Cells		ClearBinding()	
<i>TableRow</i>	425	<i>BindingOperations</i>	227
Center		ClearContent()	
<i>EllipseGeometry</i>	260	<i>PromptBuilder</i>	403
<i>RadialGradientBrush</i>	277	ClickCount	
CenterOwner		<i>MouseButtonEventArgs</i>	144
<i>WindowStartupLocation</i>	177, 182	Client-Bereich	31
CenterScreen		Clip	295
<i>WindowStartupLocation</i>	177	<i>MediaElement</i>	396
CenterX/Y/Z		ClipboardCopyMode	
<i>RotateTransform3D</i>	336	<i>DataGrid</i>	244
<i>ScaleTransform3D</i>	333	Close	
ChangedButton		<i>ApplicationCommands</i>	153
<i>MouseButtonEventArgs</i>	144	Closed	
Changes		<i>ContextMenu</i>	122
<i>TextChangedEventArgs</i>	82	<i>Window</i>	166
CheckBox	69	Closing	
<i>Basistyp</i>	209	<i>Window</i>	166
<i>Checked</i>	70	CLR-Property	24
<i>IsChecked</i>	70	cm	
<i>IsThreeState</i>	70	<i>Größe</i>	419
<i>Unchecked</i>	70	Code	
<i>Zustand</i>	70	<i>mehrfach verwenden</i>	173
Checked		Collapsed	
<i>CheckBox</i>	70	<i>Expander</i>	113
<i>RadioButton</i>	71	<i>TreeViewItem</i>	96
<i>ToggleButton</i>	70	<i>Visibility</i>	59
Child	21	Color	58
<i>BlockUIContainer</i>	429	<i>DirectionalLight</i>	310
<i>ElementHost</i>	457	<i>DropShadowEffect</i>	298
<i>PageContent</i>	443	<i>EmissiveMaterial</i>	329
<i>WindowsFormsHost</i>	449	<i>FromArgb()</i>	274, 453
Children	21	<i>GradientStop</i>	275
<i>Add()</i>	35	<i>PointLight</i>	317
<i>FixedPage</i>	443	<i>SolidColorBrush</i>	274
<i>Remove()</i>	37, 61	<i>SpecularMaterial</i>	329
<i>TransformGroup</i>	292	ColorAnimation	347, 364
<i>Viewport3D</i>	305	ColorAnimationUsingKeyFrames	374, 377
Choices	413	ColorDialog	453
Circle		Colors	58
<i>TextMarkerStyle</i>	423	Column	
CircleEase	383	<i>Grid</i>	46

ColumnDefinitions			
<i>Count</i>	47	Conditions	
<i>Grid</i>	46	<i>MultiTrigger</i>	214
ColumnProperty		ContainerUIElement3D	323
<i>Grid</i>	47	Content	56
Columns		<i>ContentControl</i>	219
<i>DataGrid</i>	246	<i>Label</i>	74
<i>Table</i>	425	ContentControl	219
ColumnSpan		<i>Content</i>	219
<i>Grid</i>	48	ContentEnd	
<i>TableCell</i>	427	<i>FlowDocument</i>	442
CombinedGeometry	262	<i>Paragraph</i>	443
<i>Geometry1</i>	263	ContentStart	
<i>Geometry2</i>	263	<i>FlowDocument</i>	442
<i>GeometryCombineMode</i>	263	<i>Paragraph</i>	443
ComboBox	92	ContextMenu	121
<i>Auswahl gewechselt</i>	94	<i>Closed</i>	122
<i>editierbar</i>	92	<i>HorizontalOffset</i>	121
<i>Eintrag</i>	92	<i>IsOpen</i>	123
<i>IsEditable</i>	92	<i>Opened</i>	122
<i>SelectionChanged</i>	94	<i>VerticalOffset</i>	121
<i>StaysOpenOnEdit</i>	92	Control	173
<i>Text</i>	92	<i>ModifierKeys</i>	157
<i>Text der Auswahl</i>	92	Control Template	215
<i>Vorauswahl</i>	94	<i>Definition</i>	215
ComboBoxItem	92	<i>in Typ-Style</i>	219
ComboBoxStyleKey		<i>mit Property Trigger</i>	216
<i>ToolBar</i>	126	<i>Triggers</i>	217
COM-Komponente	457	<i>Zieleigenschaft</i>	218
Command	441	<i>Zuordnung</i>	216
<i>CommandBinding</i>	153	ControlTemplate	
<i>KeyBinding</i>	156	<i>TargetType</i>	216
CommandBinding	153	<i>Zieltyp</i>	216
<i>CanExecute</i>	153	Copy	81
<i>Command</i>	153	CopyToOutputDirectory	171
<i>Executed</i>	153	CornerRadius	
CommandBindings	153	<i>Border</i>	109
CommandLine		Count	
<i>Environment</i>	169	<i>RemovedItems</i>	87
Commands	152	<i>SelectedDates</i>	131
CommandTarget	441	Count()	
<i>KeyBinding</i>	156	<i>Args</i>	169
ComponentCommands	152	CSS	77, 202, 419
Condition		CubicEase	382
<i>Property</i>	215	CurrentAndNext	
<i>Value</i>	215	<i>GridResizeBehavior</i>	52
		Cut	81

D

Data	
<i>Path</i>	260, 268
DataContext	235, 238, 241
DataFormats	442
DataGrid	241
<i>ausgewählte Inhalte</i>	246
<i>Auswahl gewechselt</i>	245
<i>AutoGenerateColumns</i>	245
<i>CanDeleteAddRows</i>	244
<i>CanUserAddRows</i>	244
<i>CanUserReorderColumns</i>	244
<i>CanUserResizeColumns</i>	244
<i>CanUserResizeRows</i>	244
<i>CanUserSortColumns</i>	244
<i>CheckBox</i>	246
<i>ClipboardCopyMode</i>	244
<i>Columns</i>	246
<i>ComboBox</i>	246
<i>Daten ändern</i>	244
<i>Daten hinzufügen</i>	244
<i>Daten löschen</i>	244
<i>Datenquelle</i>	242
<i>Datensätze auswählen</i>	244
<i>eigenes Feld</i>	246
<i>Felder übernehmen</i>	245
<i>GridLinesVisibility</i>	244
<i>HeadersVisibility</i>	244
<i>Hyperlink</i>	246
<i>IsReadOnly</i>	244
<i>ItemsSource</i>	242
<i>Köpfe sichtbar</i>	244
<i>Kopiermodus</i>	244
<i>Linien sichtbar</i>	244
<i>SelectedItems</i>	246
<i>SelectionChanged</i>	245
<i>SelectionMode</i>	244
<i>SelectionUnit</i>	244
<i>Spalte sortieren</i>	244
<i>Spalten</i>	246
<i>Spalten tauschen</i>	244
<i>Spaltenbreite</i>	244
<i>Spaltentypen</i>	244
<i>Zeilenhöhe</i>	244
<i>Zelle auswählen</i>	244
DataGridCheckBoxColumn	246
DataGridClipboardCopyMode	244
DataGridComboBoxColumn	246
DataGridGridLinesVisibility	244
DataGridHeadersVisibility	244
DataGridHyperlinkColumn	246
DataGridSelectionMode	244
DataGridSelectionUnit	244
DataGridTemplateColumn	246
DataGridTextColumn	246
DataRowView	246
<i>Row</i>	246
DataTable	241
DataTemplate	246
<i>DataType</i>	248
<i>Triggers</i>	249
DataTrigger	248
<i>Bedingung</i>	249
<i>Binding</i>	249
<i>Value</i>	249
DataType	
<i>DataTemplate</i>	248
Datei öffnen	
<i>Dialogfeld</i>	452
Dateiauswahl	194
Daten	
<i>externe</i>	225
Datenbank	239
<i>Adapter</i>	241
<i>aktualisieren</i>	243
<i>im DataGrid</i>	241
<i>Provider</i>	241
<i>SQL-Befehle</i>	243
<i>Tabelle</i>	241
<i>Verbindung</i>	241
Datenbindung	225
<i>aktualisieren</i>	228
<i>alle auflösen</i>	227
<i>an Auflistung von Objekten</i>	235
<i>an Datenbank</i>	239, 241
<i>an ObjectDataProvider</i>	238
<i>an Objekt</i>	232
<i>eine auflösen</i>	227
<i>ermitteln</i>	230
<i>gebundene Eigenschaft</i>	226
<i>gebundenes Element</i>	226
<i>Kontext</i>	234
<i>Label</i>	75
<i>ListBox</i>	237
<i>mit Data Template</i>	248

- Datenbindung (Forts.)
 - Quelle* 232, 234
 - Richtung* 227
 - setzen* 227
 - Validierung* 230
 - Zeitpunkt* 228
- Datenformat 442
- Datenquelle
 - Auflistung von Objekten* 235
 - Datenbank* 239, 241
 - Object Data Provider* 238
 - Objekt* 232
- DatePicker 133
 - SelectedDate* 134
 - SelectedDateChanged* 133
 - SelectedDateFormat* 133
- DatePickerFormat 133
- DateTime 128, 130, 133
- DateTime? 131
 - auf Wert prüfen* 134
 - HasValue* 134
- Datum 128
- Datumsbereiche 128
- Datumswähler 133
 - ausgewähltes Datum* 134
 - Format* 133
 - Wechsel der Auswahl* 133
- Datumswerte 128
- DayOfWeek 132
- Debug-Modus 483
- DecelerationRatio
 - AnimationTimeline* 352
- DecimalAnimation 347
- DecimalAnimationUsingKeyFrames 374
- Dehnung 134
- Delegate 164
- delete
 - SQL* 243
- Delta
 - MouseWheelEventArgs* 145
- DeltaManipulation
 - ManipulationDeltaEventArgs* 150
- Dependency Property 24
 - Wert ermitteln* 34
 - Wert setzen* 34
- Desktop 463
- Dialogfeld
 - eigenes* 180
- DialogResult 452
 - Window* 182
- Dicke 63, 65
- DictationGrammar 409
- DiffuseMaterial 305
 - Brush* 305, 330
- Digitalisiertablett 145
- Direction
 - DirectionalLight* 309
 - DropShadowEffect* 298
 - SpotLight* 316
- DirectionalLight 309, 314
 - Color* 310
 - Direction* 309
- DirectX 15
- Disc
 - TextMarkerStyle* 423
- DiscreteDoubleKeyFrame 375
- DisplayDate
 - Calendar* 130
- DisplayDateEnd
 - Calendar* 132
- DisplayDateStart
 - Calendar* 132
- DisplayMemberBinding
 - GridViewColumn* 240
- Dock
 - DockPanel* 42
 - TabStripPlacement* 116
- DockPanel 42, 118, 124, 126
 - Dock* 42
 - DockProperty* 45
 - LastChildFill* 44
- DockProperty
 - DockPanel* 45
- Document
 - DocumentViewer* 443
- DocumentViewer 443
- Dokument
 - Absatz* 418
 - ändern* 415, 440
 - anzeigen* 415
 - Beginn* 442
 - blättern* 417, 439
 - dynamisches* 415
 - Ende* 442
 - formatieren* 419, 440
 - mit Element* 429

Dokument (Forts.)		
<i>Position</i>	442	
<i>scrollen</i>	417, 438	
<i>Seite</i>	443	
<i>statisches</i>	415	
<i>zoomen</i>	417	
<i>zum Drucken</i>	443	
double?	349	
<i>HasValue</i>	351	
DoubleAnimation	347, 348	
DoubleAnimationUsingKeyFrames	374	
DoubleAnimationUsingPath	383	
DoubleCollection	107	
DoWork()		
<i>BackgroundWorker</i>	104	
Drag		
<i>SystemGesture</i>	147	
Drag&Drop	171	
<i>ermöglichen</i>	192	
DragMove()		
<i>Window</i>	192	
DrawEllipse		
<i>DrawingContext</i>	300	
Drawing	270	
DrawingContext	300	
<i>DrawEllipse</i>	300	
DrawingGroup	271	
DrawingImage	272	
Drehbuch	355	
Drehung		
<i>2D-Grafik</i>	283	
<i>3D-Grafik</i>	336	
<i>Touchscreen</i>	148	
Dreidimensionale Grafik	301	
Dreidimensionale Landschaft	343	
Dreieck		
<i>in 3D-Grafik</i>	302	
DropShadowEffect	297	
Druck		
<i>vorbereiten</i>	445	
<i>Warteschlange</i>	446	
Drucken		
<i>visuelles Objekt</i>	446	
Drucker		
<i>angeschaltet</i>	447	
<i>Anzahl Jobs</i>	447	
<i>Dokumentgröße</i>	447	
<i>Druckbereich</i>	448	
Drucker (Forts.)		
<i>Name</i>	447	
Duration		
<i>AnimationTimeline</i>	351	
<i>TimeSeekOrigin</i>	361	
Durchsichtigkeit	191, 292	
DynamicResource	175	
Dynamische Ressource	174, 175	
E		
EaselIn		
<i>EasingMode</i>	380	
EaselInOut		
<i>EasingMode</i>	380	
EaseOut		
<i>EasingMode</i>	380	
Easing Function	379	
<i>Modus</i>	380	
EasingDoubleKeyFrame	375	
EasingFunctionBase	380	
EasingMode		
<i>in Easing Function</i>	380	
echo off	168	
Ecke abrunden	109	
EditingCommands	152, 440	
<i>ToggleBold</i>	154	
Effect	296	
Eigenschaft		
<i>Abhängigkeits-</i>	24	
<i>zentral definieren</i>	202	
Eigenschaftselement	23	
Eigenschaftswert		
<i>überschreiben</i>	204	
Eingabegeste	155	
<i>hinzufügen</i>	159	
<i>Sammlung</i>	159	
Eingabestift	145	
ElasticEase	383	
<i>Oscillations</i>	383	
<i>Springiness</i>	383	
Element		
<i>allgemeine Eigenschaften</i>	55	
<i>andocken</i>	42	
<i>Ausrichtung</i>	65	
<i>Außenabstand</i>	38, 63	
<i>ausstanzen</i>	295	
<i>bedienbares</i>	58	

Element (Forts.)

<i>binden</i>	227
<i>Breite</i>	38, 57
<i>Datenkontext</i>	235, 238
<i>drehen</i>	283
<i>Eigenschaft</i>	22
<i>einrahmen</i>	109
<i>entfernen</i>	61
<i>Fokus setzen</i>	83
<i>gestalten</i>	31, 215
<i>gruppieren</i>	112
<i>hervorheben</i>	109
<i>Hintergrund</i>	292
<i>Hintergrundfarbe</i>	36, 57
<i>Höhe</i>	57
<i>im Layout</i>	58
<i>in Dokument eingebettet</i>	429
<i>in Dokument verankern</i>	436
<i>in Raster anordnen</i>	45
<i>in Text eingebettet</i>	431
<i>Inhalt</i>	56
<i>Innenabstand</i>	61
<i>ist geladen</i>	166
<i>ist initialisiert</i>	166
<i>Kontext-Info</i>	79
<i>letztes zum Füllen</i>	44
<i>mit Bild</i>	66
<i>mit Schatten</i>	297
<i>Name</i>	21
<i>neigen</i>	288
<i>neu erzeugen</i>	34
<i>positionieren</i>	32
<i>Schriftart</i>	57
<i>Schriftdehnung</i>	57
<i>Schriftgewicht</i>	57
<i>Schriftgröße</i>	57
<i>Schriftstil</i>	57
<i>sichtbares</i>	58
<i>skalieren</i>	286
<i>stapeln</i>	37
<i>stapeln mit Umbruch</i>	39
<i>Stil</i>	201
<i>Tastatursteuerung</i>	66
<i>Template</i>	216
<i>Transparenz</i>	292
<i>übereinander</i>	284
<i>übergeordnetes</i>	27, 36, 98
<i>verschieben</i>	290

Element (Forts.)

<i>verwischen</i>	296
<i>Vordergrundfarbe</i>	57
<i>Zuordnung lösen</i>	37
<i>zusammenfassen</i>	109
ElementAt()	
<i>Blocks</i>	420
<i>Inlines</i>	79
ElementHost	454
ElementName	76
<i>Binding</i>	226
Ellipse	252, 259
<i>Größe</i>	260
<i>Ort</i>	260
EllipseGeometry	259
<i>Center</i>	260
<i>RadiusX</i>	260
EmissiveMaterial	327
<i>Color</i>	329
EndPoint	
<i>LinearGradientBrush</i>	275
<i>LineGeometry</i>	261
EndStyle()	
<i>PromptBuilder</i>	403
Environment	
<i>CommandLine</i>	169
Ereignis	
<i>angehängtes</i>	25, 29
<i>Auslöser</i>	29
<i>Bearbeitungsreihenfolge</i>	210
<i>Behandlung abbrechen</i>	212
<i>geroutetes</i>	28, 30
<i>in 3D-Grafik</i>	321
<i>Registrierer</i>	29
<i>Style für Reaktion</i>	210
<i>wiederholen</i>	68
Ereignishandler	28
Ereignismethode	211
Ereignisreihenfolge	164
errorlevel	168
EvenOdd	
<i>FillRule</i>	270
Event	
<i>EventSetter</i>	211
Event Trigger	365
<i>als Ressource</i>	366
<i>in Style</i>	367
<i>steuert Animation</i>	369

Event Trigger (Forts.)		
zuordnen	370	
EventArgs	29	
Event-Bubbling	28	
EventHandler		
neu erzeugen	60	
EventSetter		
Bearbeitungsreihenfolge	210	
Event	211	
Handler	211	
Style	210	
EventTrigger		
RoutedEvent	366	
SourceName	370	
Event-Tunneling	28	
Exclamation		
SystemSounds	391	
Exclude		
GeometryCombineMode	263	
Executed		
CommandBinding	153	
Exit		
Application	166, 167	
ExitEventArgs	166	
ApplicationExitCode	170	
ExpandDirection		
Expander	113	
Expanded		
Expander	113	
TreeViewItem	96	
Expander		
Aufklapprichtung	113	
Beschriftung	113	
Collapsed	113	
ExpandDirection	113	
Expanded	113	
Header	113	
in Windows Forms	454	
IsExpanded	113	
ist aufgeklappt	113	
klappt auf	113	
klappt zu	113	
ExpandSubtree()		
TreeViewItem	100	
ExpansionVelocity		
ManipulationVelocities	151	
Explicit		
UpdateSourceTrigger	228	
Expliziter Style	202	
Exponent		
ExponentialEase	383	
ExponentialEase	383	
Exponent	383	
Extended		
DataGridSelectionMode	244	
SelectionMode	89	
eXtensible Application Markup Language	17	
F		
Farbe	58	
animieren	364, 377	
Komponente	274, 292	
konvertieren	453	
linearer Verlauf	274	
radialer Verlauf	277	
Farbe auswählen		
Dialogfeld	453	
Farbverlauf		
Übergangspunkt	275	
Fenster	176	
Änderung der Größe	177	
Besitzer	181	
darf transparent sein	191	
eigenes Unterfenster	180	
Eigenschaft	176	
Ereignis	176	
Größe	176	
Größe anpassen	179	
Größe geändert	127	
Größe vorher, nachher	178	
Größe wurde geändert	178	
immer oben	178	
in Taskbar anzeigen	177	
ist entladen	166	
ist geladen	73, 127, 166	
ist geschlossen	166	
ist initialisiert	166, 222	
Layout-Aktualisierung	448	
modal anzeigen	181	
Navigation	183	
nicht-modal anzeigen	181	
Position wurde geändert	178	
Rahmenart	191	
Rückgabewert	182	
Schließen abbrechen	167	

- Fenster (Forts.)
- StartPosition* 177
 - Status* 137
 - Status wurde geändert* 178
 - Titel* 176
 - Unterfenster erzeugen* 181
 - versehentlich geschlossen* 18
 - wird geschlossen* 166, 243
- Fettschrift 58, 77, 154, 431
- FieldOfView
- PerspectiveCamera* 312
- Figure
- Inline-Typ* 432, 436
- FigureHorizontalAnchor 436
- Figures
- PathGeometry* 266
- FigureVerticalAnchor 436
- FilePrompt 401
- FileStream 194, 223
- Fill
- Path* 260
 - Shape* 253
 - Stretch* 134
- Fill()
- OleDbAdapter* 241
- FillRule
- GeometryGroup* 270
- FindResource() 176, 205
- Finger
- Touch* 147
- FirstBlock
- Blocks* 420
- FirstDayOfWeek
- Calendar* 132
- FixedDocument 443
- FixedPage 443
- Flat
- PenLineCap* 258
- Flick
- SystemGesture* 147
- FlipX
- TileMode* 281
- FlipXY
- TileMode* 281
- FlipY
- TileMode* 281
- FlowDirection
- StackPanel* 38
- FlowDirection (Forts.)
- WrapPanel* 40, 115
- FlowDocument 84, 415
- FlowDocumentPageViewer 415, 439
- FlowDocumentReader 415, 416
- ViewingMode* 417
- FlowDocumentReaderViewingMode 417
- FlowDocumentScrollViewer 415, 438
- Focus() 83
- FolderBrowserDialog 453
- FontDialog 454
- FontFamily 57
- TextElement* 419
- FontSize 57, 419
- FontStretch 57
- FontStyle 57
- FontWeight 57
- Foreground 57
- Forever
- Duration* 351
 - RepeatBehavior* 352
- Formatierung 419
- kaskadierende* 77
- Fortschritt
- ist eingetreten* 105
 - Prozentsatz* 105
- Fortschrittsbalken 101
- Frame 187
- Aufbau* 189
 - Datei* 189
 - erste* 189
 - Navigate()* 472
 - Navigationsziel* 189
 - Page* 472, 475
 - Source* 189
 - Titel* 189
- From
- AnimationTimeline* 348
- FromArgb()
- Color* 274, 453
- FullRow
- DataGridSelectionUnit* 244
- ## G
- Gadget 190
- Geometrie 259
- Füllregel* 270

Geometrie (Forts.)		
<i>gruppieren</i>	268	
<i>kombinierte</i>	262	
<i>kompakte Schreibweise</i>	268	
<i>komplexe</i>	264	
<i>Mengenlehre</i>	263	
<i>mit Bilddatei</i>	271	
<i>mit Schriftartzeichen</i>	271	
<i>mit Videodatei</i>	271	
<i>Teilfigur</i>	264	
<i>Teilsegment</i>	264	
Geometry	259, 295	
<i>GeometryDrawing</i>	272	
<i>GeometryModel3D</i>	305	
<i>Viewport2DVisual3D</i>	326	
Geometry1		
<i>CombinedGeometry</i>	263	
Geometry2		
<i>CombinedGeometry</i>	263	
GeometryCombineMode		
<i>CombinedGeometry</i>	263	
GeometryDrawing	272	
GeometryGroup	268	
<i>FillRule</i>	270	
GeometryModel3D	305, 319	
<i>BackMaterial</i>	310	
<i>Transform</i>	334	
Gerichtetes Licht	309, 314	
Geste	155	
<i>Bindung</i>	156	
<i>hinzufügen</i>	159	
<i>Sammlung</i>	159	
GetAdornerLayer()		
<i>AdornerLayer</i>	299	
GetBinding()		
<i>Binding</i>	230	
GetInstalledVoices		
<i>SpeechSynthesizer</i>	399	
GetIntermediateTouchPoints()		
<i>TouchEventArgs</i>	148	
GetPosition()		
<i>MouseButtonEventArgs</i>	144	
<i>MouseEventArgs</i>	128, 144	
<i>MouseWheelEventArgs</i>	145	
GetTouchPoint()		
<i>TouchEventArgs</i>	148	
GetType()	29	
GetValue()	24, 34, 142	
Glänzendes Material	327	
GlyphRunDrawing	271	
GoBack()		
<i>NavigationService</i>	187	
<i>WebBrowser</i>	138	
GoForward()		
<i>NavigationService</i>	187	
<i>WebBrowser</i>	138	
GradientOrigin		
<i>RadialGradientBrush</i>	277	
GradientStop	275	
<i>Color</i>	275	
<i>Offset</i>	275	
GradientStopCollection	277	
GradientStops		
<i>LinearGradientBrush</i>	277	
Grafik	251, 301	
Grafik-Hardware	16	
GrammarBuilder	411, 413	
Grammatik		
<i>eigene</i>	411	
<i>laden</i>	409, 413	
Grid	45	
<i>Anzahl der Spalten</i>	47	
<i>Anzahl der Zeilen</i>	47	
<i>Column</i>	46	
<i>ColumnDefinitions</i>	46	
<i>ColumnProperty</i>	47	
<i>ColumnSpan</i>	48	
<i>Row</i>	46	
<i>RowDefinitions</i>	46	
<i>RowProperty</i>	47	
<i>RowSpan</i>	48	
<i>Spalte hinzufügen</i>	47	
<i>Spalten</i>	46	
<i>Zeile hinzufügen</i>	47	
<i>Zeilen</i>	46	
<i>Zellen aufspannen</i>	49	
<i>Zellgröße</i>	49	
<i>Zellgröße flexibel</i>	51	
GridLength	427	
GridLinesVisibility		
<i>DataGrid</i>	244	
GridResizeBehavior	52	
GridSplitter	51	
<i>HorizontalAlignment</i>	52	
<i>ResizeBehavior</i>	52	
GridUnitType	427	

GridView	240	Height (Forts.)	
<i>Datenbindung</i>	240	<i>SizeToContent</i>	179
<i>Spaltenbeschriftung</i>	240	Window	176
GridViewColumn	240	Hidden	
<i>DisplayMemberBinding</i>	240	<i>Visibility</i>	59
Header	240	Hierarchie	21, 27
Größenänderung		<i>darstellen</i>	94
2D-Grafik	286	<i>von Layouts</i>	35
3D-Grafik	333	Hilfestellung	17
Faktor	286	Hintergrund	
Verhalten	52	<i>durchsichtiger</i>	292
GroupBox	112	Hintergrundvorgang	103
<i>Beschriftung</i>	113	<i>berichtet</i>	104
Header	113	<i>darf berichten</i>	103
GroupName		<i>ist tätig</i>	104
RadioButton	72	<i>starten</i>	104
Guid		<i>Zustand geändert</i>	104
StylusButton	147	History	
Gültigkeitsbereich	173	<i>Navigation</i>	184
Style	206, 208	HoldEnter	
		<i>SystemGesture</i>	147
H		Horizontal	
Hand		<i>Orientation</i>	38
SystemSounds	391	HorizontalAlignment	66
Handled		<i>GridSplitter</i>	52
RoutedEventArgs	212	Stretch	52
Handler		HorizontalAnchor	
EventSetter	211	Figure	436
HasValue		HorizontalContentAlignment	66
DateTime?	134	HorizontalOffset	
double?	351	ContextMenu	121
Hauptelement	21	HoverEnter	
Hauptmenü	118	SystemGesture	147
Header		HoverLeave	
Expander	113	SystemGesture	147
GridViewColumn	240	Hyperlink	185
GroupBox	113	Inline-Typ	431
MenuItem	119	NavigateUri	186, 190
RibbonMenuItem	197	TargetName	190
TabItem	116	Ziel	186, 190
TreeViewItem	96	Zielframe	190
HeadersVisibility			
DataGrid	244	I	
Height	57	IAddChild	445
Image	134	Icon	
ListBox	86	MenuItem	119
RowDefinition	50		

Image	68, 134	insert	
<i>Height</i>	134	<i>SQL</i>	243
<i>Source</i>	68, 134, 173, 272	Insert()	
<i>Stretch</i>	134	<i>Items</i>	88, 99
<i>Width</i>	134	InsertAfter()	
ImageBrush	279	<i>Inlines</i>	79
<i>für 3D-Grafik</i>	332	InsertBefore()	
<i>ImageSource</i>	279	<i>Inlines</i>	79
<i>TileMode</i>	279	Installation	171
<i>Viewbox</i>	279	Int16Animation	347
<i>ViewboxUnits</i>	281	Int16AnimationUsingKeyFrames	374
<i>Viewport</i>	279	Int32Animation	347
<i>ViewportUnits</i>	281	Int32AnimationUsingKeyFrames	374
ImageDrawing	271	Int32Collection	305
ImageSource	134	<i>Add()</i>	308
<i>ImageBrush</i>	279	Int64Animation	347
<i>RibbonMenuItem</i>	197	Int64AnimationUsingKeyFrames	374
Impliziter Style	205	IntelliSense	17
in		Interoperabilität	449
<i>Größe</i>	419	Intersect	
InAir		<i>GeometryCombineMode</i>	263
<i>StylusEventArgs</i>	146	Inverted	
Inch	419	<i>StylusEventArgs</i>	146
IndexOf()		IsBusy	
<i>Items</i>	99	<i>BackgroundWorker</i>	104
Ingebretsen	19	IsCancel	67
Initialized		IsCheckable	
<i>Element</i>	166	<i>MenuItem</i>	119
<i>Window</i>	166, 222	IsChecked	
InitialVelocities		<i>CheckBox</i>	70
<i>ManipulationInertiaStartingEventArgs</i> ...	151	<i>MenuItem</i>	121
Inline	416	<i>RadioButton</i>	72
<i>BaselineAlignment</i>	433	<i>ToggleButton</i>	70
<i>TextDecorations</i>	433	IsClosed	
InlineCollection	76, 416, 431	<i>PathFigure</i>	266
Inlines	76	IsDefault	67
<i>Add()</i>	78	IsEditable	
<i>ElementAt()</i>	79	<i>ComboBox</i>	92
<i>InsertAfter()</i>	79	IsEnabled	59
<i>InsertBefore()</i>	79	<i>SpellCheck</i>	82
<i>Paragraph</i>	416, 431	IsEnabledProperty	
InlineUIContainer		<i>SpellCheck</i>	82
<i>Inline-Typ</i>	431	IsExpanded	
InnerConeAngle		<i>Expander</i>	113
<i>SpotLight</i>	316	<i>TreeViewItem</i>	96
InputBindings	156	IsFilled	
InputGestureCollection	159	<i>PathFigure</i>	266
<i>Add()</i>	159		

- IsIndeterminate
 ProgressBar 103
 IsLargeArc
 ArcSegment 267
 IsLoaded
 Window 73
 IsLocked
 ToolBarTray 124
 IsManipulationEnabled 148
 IsMoveToPointEnabled
 Slider 107
 IsMuted
 MediaPlayer 392
 IsOpen
 ContextMenu 123
 IsPageViewEnabled
 FlowDocumentReader 417
 IsReadOnly
 DataGrid 244
 IsRepeat
 KeyEventArgs 140
 IsScrollViewEnabled
 FlowDocumentReader 417
 IsSelected
 Selector 86, 91, 94, 117
 TreeViewItem 96
 IsSelectionRangeEnabled
 Slider 107
 IsSnapToTickEnabled
 Slider 107
 IsStroked
 ArcSegment 266
 LineSegment 266
 IsThreeState
 CheckBox 70
 ToggleButton 70
 IsTwoPageViewEnabled
 FlowDocumentReader 417
 IsVisualHostMaterial
 Viewport2DVisual3D 326
 Italic 77
 FontStyles 58
 Inline-Typ 431
 ItemCollection 85
 ItemHeight
 WrapPanel 41
 Items 85
 Add() 88, 99
 Items (Forts.)
 Clear() 89, 101
 Count 88
 IndexOf() 99
 Insert() 88, 99
 Remove() 89, 92, 100
 ItemsSource
 DataGrid 242
 ListView 240
 ItemWidth
 WrapPanel 41
- ## J
- Justify
 TextAlignment 419
- ## K
- Kachel 279, 463
 Kalender 128
 Anzahl ausgewählte Daten 131
 Anzeigedatum 130
 ausgeschlossene Daten 132
 ausgewählte Daten 130
 ausgewähltes Datum 131
 Auswahlmodus 128
 Datumsbereich 132
 erster Wochentag 132
 Grenzwerte 132
 Wechsel der Auswahl 130
 Kamera
 Blickrichtung 305
 für 3D-Grafik 304, 311
 perspektivische 312
 Position 305
 Sichtfeld 305
 Kameralage
 für 3D-Grafik 308, 313
 Kaskadierend 77
 Kaxaml 19
 KeepTogether
 Paragraph 419
 KeepWithNext
 Paragraph 419
 Key 139
 KeyBinding 157

KeyBinding	156	Label (Forts.)	
<i>Command</i>	156	<i>Inhalt</i>	74
<i>CommandTarget</i>	156	<i>Target</i>	75
<i>Key</i>	157	Landschaft	
<i>Modifiers</i>	157	<i>in 3D-Grafik</i>	343
KeyDown	139	LargeChange	
KeyEventArgs	139	<i>ScrollBar</i>	108
<i>IsRepeat</i>	140	<i>Slider</i>	105
<i>RoutedEvent</i>	140	LargeImageSource	
Keyframes	347, 374	<i>Ribbon</i>	197
KeyGesture	159	LastBlock	
KeySpline		<i>Blocks</i>	420
<i>SplineDoubleKeyFrame</i>	377	LastChildFill	
KeyTime		<i>DockPanel</i>	44
<i>für KeyFrame</i>	375	Laufleiste	37
<i>Paced</i>	375	Laufzeit	174
<i>Uniform</i>	375	Lautstärke	
KeyUp	139	<i>Mediendatei</i>	392
Kombinierte Geometrie	262	<i>Sprachausgabe</i>	399
Kommando	152, 441	<i>Sprache</i>	406
<i>ausführen</i>	153	Layer	
<i>Bindung</i>	153	<i>für Verzierung</i>	298
<i>darf ausgeführt werden</i>	153	Layout	31
<i>geroutetes</i>	158	<i>Basisklasse</i>	31
<i>Sondertaste</i>	157	<i>Hierarchie</i>	35
<i>Tastenbindung</i>	156	<i>kombinieren</i>	53
<i>Ziel</i>	441	LayoutTransform	283, 285
<i>Zielelement</i>	156	Left	
Kommandozeile	167	<i>Canvas</i>	32
Komplexe Geometrie	264	LeftClick	
Konfigurations-Manager	483	<i>MouseAction</i>	157
Kontextmenü	121	LeftDoubleClick	
<i>Eintrag</i>	121	<i>MouseAction</i>	157
<i>ist offen</i>	123	LeftProperty	
<i>öffnet sich</i>	122	<i>Canvas</i>	34
<i>Platzierung</i>	121	LeftToRight	
<i>schließt sich</i>	122	<i>FlowDirection</i>	38
<i>synchron halten</i>	122	Leinwand	
Koordinatensystem	301	<i>für 3D-Grafik</i>	304
Kopieren		Leiste	118
<i>in Ausgabeverzeichnis</i>	171	Lernprogramm	
Kursivschrift	58, 77, 431	<i>für Spracherkennung</i>	408
L			
Label	74	Licht	305
<i>Content</i>	74	<i>für 3D-Grafik</i>	309, 314
<i>Datenbindung</i>	75	Lichtkegel	315
		Line	254
		<i>Koordinaten</i>	255
		LinearDoubleKeyFrame	375

Linearer Farbverlauf	274	ListBoxItem (Forts.)	
LinearGradientBrush	274	<i>Unselected</i>	86
<i>EndPoint</i>	275	Liste	85
<i>für 3D-Grafik</i>	332	<i>ändern</i>	424
<i>GradientStops</i>	277	<i>hinzufügen</i>	424
<i>StartPoint</i>	275	<i>in Dokument</i>	422
LinearVelocity		<i>markieren</i>	423
<i>ManipulationVelocities</i>	151	<i>Markierungsabstand</i>	423
LineBreak	68, 77	ListItem	423
<i>Inline-Typ</i>	431	ListItemCollection	423
LineGeometry	259	ListView	239
<i>EndPoint</i>	261	<i>Darstellung</i>	240
<i>StartPoint</i>	261	<i>Datenquelle</i>	240
LineSegment	266	<i>füllen</i>	241
<i>IsStroked</i>	266	<i>ItemsSource</i>	240
<i>Point</i>	266	<i>View</i>	240
Linie	254, 259	Load()	
<i>Endpunkt</i>	261	<i>SoundPlayer</i>	388
<i>Startpunkt</i>	261	<i>TextRange</i>	443
Linienende	257	<i>XamlReader</i>	223
Liniensegment	266	LoadAsync()	
<i>Linie sichtbar</i>	266	<i>SoundPlayer</i>	388
<i>Zielpunkt</i>	266	LoadCompleted	
List		<i>WebBrowser</i>	136
<i>Block-Typ</i>	416, 422	Loaded	
<i>MarkerOffset</i>	423	<i>Element</i>	166
<i>MarkerStyle</i>	423	<i>Window</i>	127, 166
<i>StartIndex</i>	423	LoadGrammar()	409
ListBox	85	<i>SpeechRecognitionEngine</i>	413
<i>ausgewählte Einträge</i>	87, 91	LocationChanged	
<i>Auswahl gewechselt</i>	86	<i>Window</i>	178
<i>Datenbindung</i>	237	Logische Ressource	173
<i>Eintrag</i>	85	Long	
<i>Eintrag auswählen</i>	88	<i>DatePickerFormat</i>	133
<i>Eintrag sichtbar machen</i>	88	LookDirection	
<i>Mehrfachauswahl</i>	89	<i>OrthographicCamera</i>	305
<i>nicht mehr ausgewählter Eintrag</i>	87	<i>PerspectiveCamera</i>	312
<i>Nummer des ausgewählten Eintrags</i>	87	LostFocus	
<i>raumsparende Variante</i>	92	<i>UpdateSourceTrigger</i>	228
<i>ScrollIntoView()</i>	88	LowerLatin	
<i>SelectedIndex</i>	87	<i>TextMarkerStyle</i>	423
<i>SelectedItem</i>	87	LowerRoman	
<i>SelectedItems</i>	91	<i>TextMarkerStyle</i>	423
<i>Selection_Changed</i>	86		
<i>SelectionMode</i>	89		
<i>Vorauswahl</i>	86, 91		
ListBoxItem	85		
<i>Selected</i>	86		

M

-
- Magere Schrift 58
 - Main()
 - Window 163
 - MainPage 466, 476
 - MainWindow.xaml 173
 - Manipulation
 - Ereignis 147, 148
 - erlaubt 148
 - Trägheit 149
 - ManipulationBoundaryFeedback 149
 - ManipulationCompleted 149, 151
 - ManipulationCompletedEventArgs 151
 - ManipulationDelta 148, 150
 - ManipulationDeltaEventArgs 150
 - ManipulationInertiaStarting 149, 151
 - ManipulationInertiaStartingEventArgs 151
 - ManipulationOrigin 150
 - ManipulationStarted 148, 150
 - ManipulationStarting 148, 150
 - ManipulationVelocities 151
 - Manual
 - SizeToContent 179
 - WindowStartupLocation 177
 - Margin 38, 63
 - MarkerOffset
 - List 423
 - MarkerStyle
 - List 423
 - Markup Extension 26, 226
 - Maske 294
 - Material 305
 - für 3D-Grafik 327
 - GeometryModel3D 305
 - MaterialGroup 327
 - MatrixAnimationUsingKeyFrames 374
 - MatrixAnimationUsingPath 383
 - Maus 142
 - Anzahl Clicks 144
 - bewegt 127
 - Bindung 156
 - Buttonstatus 144
 - Click-Arten 157
 - Ereignis 144
 - erweiterte Taste 144
 - Position 128, 144, 145
 - welcher Button 144
 - Mausaktion 155, 159
 - Mausrad
 - Änderung 145
 - Mausrad (Forts.)
 - Click 157
 - Ereignis 145
 - Info über 142
 - Maustaste
 - Ereignis 144
 - Info über 142
 - Maximized
 - WindowState 137, 178
 - Maximum
 - ProgressBar 102
 - ScrollBar 108
 - Slider 105
 - MaxLength
 - TextBox 84
 - Measure() 448
 - MediaCommands 152
 - MediaElement 395, 396
 - MediaFailed
 - MediaElement 396
 - MediaPlayer 391
 - MediaTimeline 396
 - Mediendatei
 - abspielen 391, 395, 396
 - Fehler anzeigen 396
 - Menu 118
 - Eintrag 118
 - Menü 118
 - Platzierung 118
 - Menüband 194
 - Menultem 118, 121
 - Beschriftung 119
 - Bild 119
 - Header 119
 - Icon 119
 - IsCheckable 119
 - IsChecked 121
 - markierbares 119
 - markiertes 121
 - MeshGeometry3D 305
 - TextureCoordinates 326
 - Microsoft Excel 14.0 Object Library 457
 - Microsoft Ribbon for WPF.msi 195
 - Microsoft Word 14.0 Object Library 457
 - Microsoft.ACE.OLEDB.12.0 241

Microsoft.Office.Interop	457	MouseGesture	159
Microsoft.Windows.Controls.Ribbon	197	MouseLeave	144
MiddleClick		<i>ModelUIElement3D</i>	322
<i>MouseAction</i>	157	MouseMove	144
MiddleDoubleClick		<i>Window</i>	127
<i>MouseAction</i>	157	MouseUp	144
Mikro	408	MouseWheel	144
<i>setzen</i>	411	MouseWheelEventArgs	142
Minimized		<i>Delta</i>	145
<i>WindowState</i>	178	<i>GetPosition()</i>	145
Minimum		<i>RoutedEvent</i>	145
<i>ProgressBar</i>	102	MP3-Datei	
<i>ScrollBar</i>	108	<i>abspielen</i>	392, 395
<i>Slider</i>	105	MPG-Datei	
Miter		<i>abspielen</i>	396
<i>PenLineJoin</i>	258	MS Access	239
Modales Fenster	181	MS Excel	
Mode		<i>Mappe erzeugen</i>	458
<i>Binding</i>	227	MS Excel 2010	457
Model3DCollection	321	MS Office	
Model3DGroup	319	<i>in WPF</i>	457
Modell		MS Word	
<i>für 3D-Grafik</i>	319	<i>Dokument erzeugen</i>	460
ModelUIElement3D	321	MS Word 2010	457
<i>MouseEnter</i>	322	MS.Internal.NamedObject	246
<i>MouseLeave</i>	322	Multiple	
ModelVisual3D	305	<i>SelectionMode</i>	89
ModifierKeys	157	MultipleRange	
Modifiers		<i>CalendarSelectionMode</i>	128
<i>KeyBinding</i>	157	Multitouch	
MouseAction		<i>Ereignis</i>	147
<i>MouseBinding</i>	157	Multi-Trigger	213
MouseBinding	156	MultiTrigger	214
<i>MouseAction</i>	157	<i>Conditions</i>	214
MouseButton	144		
MouseButtonEventArgs	142	N	
<i>ButtonState</i>	144	Name	21
<i>ChangedButton</i>	144	<i>StylusButton</i>	147
<i>ClickCount</i>	144	Namespace	
<i>GetPosition()</i>	144	<i>einbinden</i>	22, 159
<i>RoutedEvent</i>	144	<i>lokaler</i>	159, 234
MouseButtonState	144	Navigate()	
MouseDown	29, 144	<i>Frame</i>	472
MouseEnter	144	<i>NavigationService</i>	186
<i>ModelUIElement3D</i>	322	<i>WebBrowser</i>	136
MouseEventArgs	128, 142	Navigated	
<i>GetPosition()</i>	128, 144	<i>WebBrowser</i>	136
<i>RoutedEvent</i>	144		

NavigateToString()	
WebBrowser	136
NavigateUri	
Hyperlink	186, 190
Navigation	183
History	184
Vorwärts, Rückwärts	184
NavigationCommands	152
Navigationsdienst	186
NavigationService	193
CanGoBack	187
CanGoForward	187
GoBack()	187
GoForward()	187
Navigate()	186
Page	186
Navigationshost	192
NavigationWindow	183, 184, 185, 189
Source	185, 189
Title	189
Neigung	
2D-Grafik	288
Winkel	288
new	35
NewSize	
SizeChangedEventArgs	178
Nicht definiert	70, 71
None	
Stretch	134
NonZero	
FillRule	270
NoResize	
ResizeMode	178
Normal	
FontStyles	58
FontWeights	58
WindowState	178
NoWrap	
TextWrapping	77
null	71, 131, 349
O	
Oberfläche	
gestalten	31
object	74, 85, 119
ObjectAnimationUsingKeyFrames	374
ObjectDataProvider	238
ObjectInstance	
ObjectDataProvider	239
Objekt	
Ereignisauslöser	30
Oblique	
FontStyles	58
ObservableCollection	236
Offset	
GradientStop	275
SeekStoryboard	371
TextChange	82
OffsetX/Y/Z	
TranslateTransform3D	335
OldValue	
RoutedPropertyChangedEventArgs ...	97, 108
OleDbAdapter	241
Fill()	241
Update()	243
OleDbCommandBuilder	243
OleDbConnection	241
OneTime	
Mode	228
OneWay	
Mode	228
OneWayToSource	
Mode	228
OnRender()	
Adorner	300
Opacity	292
DropShadowEffect	298
OpacityMask	294
Opazität	292
Open()	
MediaPlayer	392
Opened	
ContextMenu	122
OpenFileDialog	452
Orientation	
ProgressBar	101
ScrollBar	108
Slider	105
StackPanel	38
ToolBarTray	124
WrapPanel	41
OrthographicCamera	304
UpDirection	308, 313
Orthographische Kamera	304

Oscillations	
<i>ElasticEase</i>	383
OuterConeAngle	
<i>SpotLight</i>	316
Owner	
Window	181

P

Paced	
<i>KeyTime</i>	375
Padding	61
Page	183, 185, 466
<i>FlowDocumentReaderViewingMode</i>	417
<i>Frame</i>	472, 475
<i>NavigationService</i>	186
<i>WindowTitle</i>	185
PageContent	443
PageContentCollection	443
Pages	
<i>FixedDocument</i>	443
Panel	31
<i>ZIndex</i>	32, 284
<i>ZIndexProperty</i>	34
Paragraph	85, 416, 418
<i>KeepTogether</i>	419
<i>KeepWithNext</i>	419
<i>TextIndent</i>	419
Parent	36, 98
Parse()	
<i>TimeSpan</i>	353
PasswordBox	
<i>PasswordChar</i>	83
PasswordChar	
<i>PasswordBox</i>	83
Paste	81
Path	260
<i>Binding</i>	226
<i>Data</i>	260, 268
<i>Fill</i>	260
<i>Stroke</i>	260
<i>StrokeThickness</i>	260
PathFigure	264
<i>IsClosed</i>	266
<i>IsFilled</i>	266
<i>StartPoint</i>	266
PathFigureCollection	264
PathGeometry	264, 358
<i>Figures</i>	266
<i>PointAnimationUsingPath</i>	385
<i>zur Animation</i>	383
PathSegmentCollection	264
<i>Add()</i>	267
Pause()	
<i>MediaPlayer</i>	392
<i>SpeechSynthesizer</i>	398
<i>Storyboard</i>	361
PauseStoryboard	371
Pen	
<i>GeometryDrawing</i>	272
PenLineCap	258
PenLineJoin	258
PerspectiveCamera	312
Perspektivische Kamera	312
Pfadanimation	383
Pfadgeometrie	264, 295
Pfadmarkupsyntax	268, 295
Physische Ressource	170
Pinzel	273
<i>einheitliche Farbe</i>	273
<i>linearer Farbverlauf</i>	274
<i>radialer Farbverlauf</i>	277
Pinseleyp	58
Pixel	419
<i>GridUnitType</i>	427
Play()	
<i>MediaPlayer</i>	392
<i>SoundPlayer</i>	173, 387
<i>SystemSound</i>	391
PlayLooping()	
<i>SoundPlayer</i>	387
PlaySync()	
<i>SoundPlayer</i>	387
Point	257, 260
<i>ArcSegment</i>	266
<i>LineSegment</i>	266
<i>relative Koordinaten</i>	274
Point3D	305
Point3DCollection	305
<i>Add()</i>	308
PointAnimation	347, 354
PointAnimationUsingKeyFrames	374
PointAnimationUsingPath	383
<i>PathGeometry</i>	385

PointCollection	256, 326	ProgressBar (Forts.)	
<i>Add()</i>	257	<i>Maximum</i>	102
PointLight	315	<i>Minimum</i>	102
Points		<i>Orientation</i>	101
<i>Polygon</i>	256	<i>undefinierter Zustand</i>	103
PolyBezierSegment	264	<i>Value</i>	102
Polygon	256	<i>Werte</i>	102
<i>Points</i>	256	ProgressChanged	
Polyline	256	<i>BackgroundWorker</i>	104
PolyLineSegment	264, 269	ProgressChangedEventArgs	105
Polylinie	256	<i>ProgressPercentage</i>	105
PolyQuadraticBezierSegment	264	ProgressPercentage	
Position		<i>ProgressChangedEventArgs</i>	105
<i>MediaPlayer</i>	392	Projekt	
<i>MeshGeometry3D</i>	305	<i>Datenbank hinzufügen</i>	239, 241, 245
<i>OrthographicCamera</i>	305	<i>Element hinzufügen</i>	171
<i>PerspectiveCamera</i>	312	<i>Fenster hinzufügen</i>	180
<i>PointLight</i>	317	<i>neu erzeugen</i>	18
<i>SpotLight</i>	316	<i>neues</i>	464
<i>TouchPoint</i>	148	<i>Ressource</i>	170
Positionierung		<i>Ressourcen-Wörterbuch hinzufügen</i>	222
<i>fest</i>	32	<i>Seite hinzufügen</i>	185
Power		<i>speichern</i>	18
<i>PowerEase</i>	382	<i>WAV-Datei hinzufügen</i>	388
PowerEase	382	Projektmappenexplorer	18, 171
<i>Power</i>	382	Projektvorlage	464
PresentationCore	454	<i>Blank</i>	465
PresentationFramework	454	<i>Grid</i>	478
Pressed		<i>Split</i>	481
<i>MouseButtonState</i>	144	PromptBreak	406
Preview-Ereignishandler	28	PromptBuilder	403
PreviewMouseDown	29	<i>AppendBreak()</i>	406
PreviousAndCurrent		<i>AppendTextWithHint()</i>	406
<i>GridResizeBehavior</i>	52	<i>ToXml()</i>	406
PreviousAndNext		PromptRate	406
<i>GridResizeBehavior</i>	52	PromptStyle	403
PreviousSize		<i>StartStyle()</i>	406
<i>SizeChangedEventArgs</i>	178	PromptVolume	406
Primitives	71	Property	24
PrintDialog	445	<i>Condition</i>	215
PrintQueue		<i>Setter</i>	203
<i>PrintDialog</i>	446	<i>Trigger</i>	213
PrintVisual()		Property Element	23
<i>PrintDialog</i>	446	Property Trigger	212, 248
ProgressBar	101	<i>in Control Template</i>	216
<i>Grenzwerte</i>	102	<i>mehrere Bedingungen</i>	213
<i>IsIndeterminate</i>	103	PropertyChanged	
<i>Lage</i>	101	<i>UpdateSourceTrigger</i>	228

PropertyPath	227	Rahmen	109
Provider	241	<i>Dicke</i>	109
pt		<i>durchsichtig</i>	109
<i>Größe</i>	419	<i>Eckenradius</i>	109
Punkt		<i>Farbe</i>	109
<i>Größe</i>	419	Rate	
<i>im 3D-Raum</i>	305	<i>PromptRate</i>	406
Punktlicht	315	Rechteck	252, 259
px		<i>Eckenabrundung</i>	260
<i>Größe</i>	419	<i>Ort und Größe</i>	260
Q			
QuadraticBezierSegment	264	Rechtschreibung	
QuadraticEase	382	<i>prüfen</i>	82
qualified double	419	RecognitionResult	409
QuarticEase	382	RecognizeAsync()	
QuaternionAnimation	347	<i>SpeechRecognitionEngine</i>	411
QuaternionAnimationUsingKeyFrames	374	RecognizeAsyncCancel()	
QuaternionRotation3D	336	<i>SpeechRecognitionEngine</i>	411
Question		RecognizeMode	411
<i>SystemSounds</i>	391	Rect	260
QuickInfo	79	<i>RectangleGeometry</i>	260
QuinticEase	382	Rectangle	252
R			
Radialer Farbverlauf	277	RectangleGeometry	259
RadialGradientBrush	277	<i>RadiusX</i>	260
<i>Center</i>	277	<i>RadiusY</i>	260
<i>für 3D-Grafik</i>	332	<i>Rect</i>	260
<i>GradientOrigin</i>	277	RectAnimation	347
RadioButton	71	RectAnimationUsingKeyFrames	374
<i>Basistyp</i>	209	Registerkarte	116
<i>Checked</i>	71	Reiter	
<i>GroupName</i>	72	<i>Registerkarte</i>	116
<i>gruppieren</i>	72, 112	Relative	
<i>IsChecked</i>	72	<i>UriKind</i>	173, 186
<i>Unchecked</i>	71	Released	
<i>Zustand</i>	71	<i>MouseButtonState</i>	144
Radius		Release-Modus	483
<i>BlurEffect</i>	298	Remove()	
RadiusX		<i>Children</i>	37, 61
<i>EllipseGeometry</i>	260	<i>Items</i>	89, 92, 100
<i>RectangleGeometry</i>	260	RemovedItems	
RadiusY		<i>Count</i>	87
<i>RectangleGeometry</i>	260	<i>SelectionChangedEventArgs</i>	87
		RemovedLength	
		<i>TextChange</i>	82
		RenderSize	
		<i>AdornedElement</i>	300
		RenderTransform	283
		<i>RenderTransformOrigin</i>	283

RenderTransformOrigin		
<i>RenderTransform</i>	283	
RepeatBehavior		
<i>AnimationTimeline</i>	352	
RepeatButton	68, 108	
ReportProgress()		
<i>BackgroundWorker</i>	104	
ResizeBehavior		
<i>GridSplitter</i>	52	
ResizeMode		
<i>Window</i>	177	
Resource Dictionaries	176	
Resources		
<i>Application</i>	175, 215	
<i>Window</i>	175	
Ressource	170	
<i>aktuell</i>	223	
<i>Auflistung von Objekten</i>	237	
<i>dynamische</i>	174, 175	
<i>Event Trigger</i>	366	
<i>für gesamte Anwendung</i>	173, 215	
<i>logische</i>	173	
<i>nur für Fenster</i>	173	
<i>Objekt</i>	234	
<i>physische</i>	170	
<i>Schlüssel</i>	173	
<i>statische</i>	174, 175	
<i>Storyboard</i>	356	
<i>Style</i>	203	
<i>suchen</i>	176, 205	
<i>Wörterbuch</i>	176	
<i>Wörterbuch hinzufügen</i>	222	
<i>zur Laufzeit tauschen</i>	174	
Result		
<i>SpeechRecognizedEventArgs</i>	409	
Resume()		
<i>SpeechSynthesizer</i>	398	
<i>Storyboard</i>	361	
ResumeStoryboard	371	
RGB-Komponente	274, 292	
Ribbon	195	
<i>Anwendungsmenü</i>	195	
<i>Registerkarte</i>	195	
Ribbonanwendung	194	
RibbonApplicationMenu	195	
RibbonApplicationMenuItem	197	
RibbonButton	198	
RibbonCheckBox	198	
RibbonComboBox	198	
RibbonControlsLibrary	195	
RibbonGallery	198	
RibbonGalleryCategory	198	
RibbonGalleryItem	198	
RibbonGroup	195	
RibbonMenuButton	198	
RibbonMenuItem	197, 198	
RibbonRadioButton	198	
RibbonTab	195	
RibbonToggleButton	198	
RibbonWindow	195, 197	
RichTextBox	84, 415, 440	
Richtung		
<i>im 3D-Raum</i>	305	
Right		
<i>Canvas</i>	32	
RightClick		
<i>MouseAction</i>	157	
RightDoubleClick		
<i>MouseAction</i>	157	
RightDrag		
<i>SystemGesture</i>	147	
RightTap		
<i>SystemGesture</i>	147	
RightToLeft		
<i>FlowDirection</i>	38	
RotateTransform	283	
<i>Angle</i>	283, 284, 286	
RotateTransform3D	336	
Rotation		
<i>2D-Grafik</i>	283	
<i>3D-Grafik</i>	336	
<i>Drehpunkt</i>	283	
<i>Drehwinkel</i>	283	
<i>in 3D-Grafik animieren</i>	371	
<i>ManipulationDelta</i>	150	
Rotation3DAnimation	347	
Rotation3DAnimationUsingKeyFrames	374	
RotationAngle		
<i>ArcSegment</i>	267	
Rotationsgruppe		
<i>3D-Grafik</i>	341	
Round		
<i>PenLineCap</i>	258	
<i>PenLineJoin</i>	258	
Routed Events	28	
RoutedCommand	158	

RoutedEvent	29	ScaleTransform (Forts.)	
<i>EventTrigger</i>	366	<i>ScaleY</i>	286
<i>KeyEventArgs</i>	140	ScaleTransform3D	333
<i>MouseButtonEventArgs</i>	144	ScaleX/Y	
<i>MouseEventArgs</i>	144	<i>ScaleTransform</i>	286
<i>MouseWheelEventArgs</i>	145	ScaleX/Y/Z	
RoutedEventArgs	30	<i>ScaleTransform3D</i>	333
<i>Handled</i>	212	Schaltfläche	66
<i>Source</i>	118	Schieber	
RoutedEventHandler		<i>Slider</i>	105
<i>neu erzeugen</i>	61, 163	Schiene	
RoutedPropertyChangedEventArgs	97, 108	<i>Slider</i>	105
<i>OldValue</i>	97, 108	Schlagschatten	297
Row		Schlüssel	
<i>DataRowView</i>	246	<i>Control Template</i>	216
<i>Grid</i>	46	<i>Ressource</i>	173
RowDefinitions		<i>Style</i>	203
<i>Count</i>	47	Schrift	
<i>Grid</i>	46	<i>formatieren</i>	441
RowGroups		<i>hochgestellt</i>	433
<i>Table</i>	425	<i>konvertieren</i>	454
RowProperty		<i>tiefgestellt</i>	433
<i>Grid</i>	47	Schrift auswählen	
Rows		<i>Dialogfeld</i>	451, 454
<i>TableRowGroup</i>	425	Schriftart	419
RowSpan		Schriftartzeichen	
<i>Grid</i>	48	<i>in Geometrie</i>	271
<i>TableCell</i>	427	Schriftgröße	419
Rückgabeparameter	166, 170	Schriftposition	
Rückwärts		<i>vertikal</i>	433
<i>Navigation</i>	184	Schriftverzierung	433
Run	77, 420	Scroll	
<i>Inline-Typ</i>	431	<i>FlowDocumentReaderViewingMode</i>	417
Run()		Scrollbalken	
<i>Application</i>	163	<i>ListBox</i>	86
RunWorkerAsync()		ScrollBar	108
<i>BackgroundWorker</i>	104	<i>Grenzwerte</i>	108
		<i>große Änderung</i>	108
		<i>kleine Änderung</i>	108
		<i>Lage</i>	108
		<i>LargeChange</i>	108
		<i>Maximum</i>	108
		<i>Minimum</i>	108
		<i>Orientation</i>	108
		<i>SmallChange</i>	108
		<i>ValueChanged</i>	108
		<i>Wert geändert</i>	108
S			
SampleDataSource.cs	480		
Save()			
<i>TextRange</i>	443		
SayAs	406		
Scale			
<i>ManipulationDelta</i>	150		
ScaleTransform	286		
<i>ScaleX</i>	286		

ScrollIntoView()		SelectedIndex	
<i>ListBox</i>	88	<i>ListBox</i>	87
ScrollView	37	<i>TabControl</i>	116
<i>Sichtbarkeit</i>	38	SelectedItem	
Section	416, 420	<i>ListBox</i>	87
SecurityException	194	SelectedItemChanged	
Seek()		<i>TreeView</i>	96
<i>Storyboard</i>	361	SelectedItems	
SeekStoryboard	371	<i>DataGrid</i>	246
<i>Offset</i>	371	<i>ListBox</i>	91
SeekToFill()		SelectedText	
<i>Storyboard</i>	361	<i>TextBox</i>	83
Seite	185	SelectedValue	
<i>darf in History bewegen</i>	187	<i>RibbonGallery</i>	198
<i>Daten übermitteln</i>	186	SelectedValuePath	
<i>der Reihe nach</i>	183	<i>RibbonGallery</i>	198
<i>erste</i>	185	Selection_Changed	
<i>in Frames</i>	187	<i>ListBox</i>	86
<i>in History bewegen</i>	187	<i>TabControl</i>	116
<i>Navigation</i>	183	SelectionChanged	
<i>wechseln</i>	186	<i>ComboBox</i>	94
Seitenumbruch	421	<i>DataGrid</i>	245
Seitenvorlage	469	<i>RibbonGallery</i>	198
<i>Elementdetails</i>	478	SelectionChangedEventArgs	87
<i>Elemente</i>	482	<i>RemovedItems</i>	87
<i>Geteilte Seite</i>	482	SelectionEnd	
<i>Gruppendetails</i>	478	<i>Slider</i>	107
<i>Gruppierte Elemente</i>	478	SelectionLength	
<i>Standardseite</i>	475	<i>TextBox</i>	83
SelectAll()		SelectionMode	
<i>TextBox</i>	83	<i>Calendar</i>	128
Selected		<i>DataGrid</i>	244
<i>ListBoxItem</i>	86	<i>ListBox</i>	89
<i>TreeViewItem</i>	96	SelectionStart	
SelectedDate		<i>Slider</i>	107
<i>Calendar</i>	131	<i>TextBox</i>	83
<i>DatePicker</i>	134	SelectionUnit	
SelectedDateChanged		<i>DataGrid</i>	244
<i>DatePicker</i>	133	Selector	
SelectedDateFormat		<i>IsSelected</i>	86, 91, 94, 117
<i>DatePicker</i>	133	SelectVoice()	
SelectedDates		<i>SpeechSynthesizer</i>	399
<i>Calendar</i>	130	sender	29
<i>Count</i>	131	Separator	64, 118
SelectedDatesChanged		SetBinding()	227
<i>Calendar</i>	130	<i>BindingOperations</i>	227
SelectedDatesCollection	130	SetInputToDefaultAudioDevice()	
		<i>SpeechRecognitionEngine</i>	411

SetOutputToDefaultAudioDevice()		Single (Forts.)	
<i>SpeechSynthesizer</i>	399	<i>SelectionMode</i>	89
SetOutputToWaveFile()		SingleAnimation	347
<i>SpeechSynthesizer</i>	399	SingleAnimationUsingKeyFrames	374
SetSpeedRatio()		SingleBorderWindow	
<i>Storyboard</i>	361	<i>WindowStyle</i>	191
SetStoryboardSpeedRatio	371	SingleDate	
<i>SpeedRatio</i>	371	<i>CalendarSelectionMode</i>	128
SetTargetName()		SingleRange	
<i>Storyboard</i>	359	<i>CalendarSelectionMode</i>	128
SetTargetProperty()		Single-Threaded Apartment Thread	162
<i>Storyboard</i>	359	Size	266
Setter	203	<i>ArcSegment</i>	266
<i>Property</i>	203	SizeAnimation	347, 358
<i>TargetName</i>	217	SizeAnimationUsingKeyFrames	374
<i>Value</i>	203	SizeChanged	
SetValue()	24, 34, 142	<i>Window</i>	127, 178
ShadowDepth		SizeChangedEventArgs	178
<i>DropShadowEffect</i>	298	SizeToContent	
Shape	251	<i>Window</i>	179
<i>Fill</i>	253	Skalierung	
<i>Füllfarbe</i>	253	2D-Grafik	286
<i>Linien dicke</i>	253	3D-Grafik	333
<i>Linien ende</i>	257	<i>Touchscreen</i>	148
<i>Linienfarbe</i>	253	SkewTransform	288
<i>Stroke</i>	253	<i>AngleX</i>	288
<i>StrokeEndLineCap</i>	257	<i>AngleY</i>	288
<i>StrokeLineJoin</i>	258	Skin	221
<i>StrokeStartLineCap</i>	257	SkipStoryboardToFill	371
<i>StrokeThickness</i>	253	Sleep()	
Shift		<i>Thread</i>	103
<i>ModifierKeys</i>	157	Slider	105
Short		<i>AutoToolTipPlacement</i>	107
<i>DatePickerFormat</i>	133	<i>Bereich markiert</i>	107
Show()		<i>Grenzwerte</i>	105
<i>Window</i>	181	<i>große Änderung</i>	105
ShowDialog()		<i>IsMoveToPointEnabled</i>	107
<i>PrintDialog</i>	445	<i>IsSelectionModeEnabled</i>	107
<i>Standard-Dialogfeld</i>	452	<i>IsSnapToTickEnabled</i>	107
<i>Window</i>	181	<i>Lage</i>	105
ShowInTaskbar		<i>LargeChange</i>	105
<i>Window</i>	177	<i>Markierungsgrenze</i>	107
Sicherheitseinschränkung	193	<i>Maximum</i>	105
Sichtbarkeit	58, 113, 126, 128	<i>Minimum</i>	105
Sichtfeld	305	<i>Orientation</i>	105
SineEase	383	<i>SelectionEnd</i>	107
Single		<i>SelectionStart</i>	107
<i>DataGridSelectionMode</i>	244	<i>Skala</i>	107

- Slider 105 (Forts.)
 - Skala, Platzierung* 105
 - Skala, Strichdichte* 105
 - springt nur zu Skalenstrich* 107
 - springt zu Mausposition* 107
 - TickFrequency* 105
 - TickPlacement* 105
 - Ticks* 107
 - ToolTip, Platzierung* 107
 - Value* 105
 - ValueChanged* 105
 - Wert* 105
 - Wert geändert* 105
- SmallChange 108
- ScrollBar 108
- SmallImageSource 197
 - Ribbon* 197
- SolidColorBrush 58, 273
 - Color* 274
- Sondertaste 157
 - Kommando* 157
- Sonderzeichen 69
 - anzeigen* 69
- SoundLocation 173, 388
 - SoundPlayer* 173, 388
- SoundPlayer 173, 387
 - Play()* 173
 - SoundLocation* 173
- SoundPlayerAction 391
- Source 234
 - Binding* 234
 - Frame* 189
 - Image* 68, 134, 173, 272
 - MediaElement* 396
 - MediaTimeline* 396
 - NavigationWindow* 185, 189
 - RoutedEventArgs* 118
 - SoundPlayerAction* 391
 - WebBrowser* 136
- SourceName 370
 - EventTrigger* 370
- Span 431
 - Inline-Type* 431
- Speak() 398
 - SpeechSynthesizer* 398
- SpeakAsync() 398
 - SpeechSynthesizer* 398
- SpeakCompleted 403
 - EventHandler* 403
 - SpeechSynthesizer* 398
- SpecularMaterial 327
 - Color* 329
 - SpecularPower* 329
- SpecularPower 329
 - SpecularMaterial* 329
- Speech Recognition Grammar 411
 - Specification* 411
- Speech Synthesis Markup Language .. 401, 403
- SpeechRecognitionEngine 410
 - RecognizeAsync()* 411
 - RecognizeAsyncCancel()* 411
 - SetInputToDefaultAudioDevice()* 411
- SpeechRecognized 409
 - SpeechRecognizer* 409
- SpeechRecognizedEventArgs 409
- SpeechRecognizer 409
- SpeechSynthesizer 398
- SpeedRatio 392
 - MediaPlayer* 392
 - SetStoryboardSpeedRatio* 371
- SpellCheck 82
 - IsEnabled* 82
 - IsEnabledProperty* 82
- Spielerei 190
- SplashScreen 465
- SplineDoubleKeyFrame 375
 - KeySpline* 377
- SpotLight 315
- Sprache 406
 - Art der Ausgabe* 406
 - aus Datei* 401
 - Ausgabe gemäß W3C* 403
 - ausgeben* 398
 - Eingabe gemäß W3C* 411
 - Eingabegerät* 411
 - eingeben* 407
 - erkennen* 407
 - Pause* 406
 - speichern in SSML* 404, 406
 - speichern in WAV* 399
 - steuert Anwendung* 411
 - Zuhören beenden* 411
 - Zuhören starten* 411
 - zusammensetzen* 398, 403

Spracherkennung			
<i>einschalten</i>	408	StaysOpenOnEdit	
<i>integrierte</i>	410	<i>ComboBox</i>	92
<i>von Windows</i>	408	Stern	
Sprachgeschwindigkeit	406	<i>Größenangabe</i>	427
Sprachlautstärke	406	Steuerelement	
Sprachstil	406	<i>anordnen</i>	31
Springiness		<i>Gruppen</i>	55
<i>ElasticEase</i>	383	Stift	
SQL-Befehl	243	<i>berührt</i>	146
Square		<i>schwebt</i>	146
<i>PenLineCap</i>	258	Stop()	
<i>TextMarkerStyle</i>	423	<i>MediaPlayer</i>	392
SRGS	411	<i>SoundPlayer</i>	387
SSML	403	<i>Storyboard</i>	361
Ssml		StopStoryboard	371
<i>SynthesisMediaType</i>	401	Storyboard	355
StackPanel	37	<i>als Ressource</i>	356
<i>Orientierung</i>	38	<i>Begin()</i>	357, 361
<i>Richtung</i>	38	<i>für Mediendatei</i>	396
Standard-Dialogfeld	193, 451	<i>Pause()</i>	361
<i>Rückgabewert</i>	452	<i>Resume()</i>	361
Star		<i>Seek()</i>	361
<i>GridUnitType</i>	427	<i>SeekToFill()</i>	361
Startbildschirm	463	<i>SetSpeedRatio()</i>	361
StartIndex		<i>SetTargetName()</i>	359
<i>List</i>	423	<i>SetTargetProperty()</i>	359
StartPoint		<i>Stop()</i>	361
<i>LinearGradientBrush</i>	275	<i>TargetName</i>	358
<i>LineGeometry</i>	261	<i>TargetProperty</i>	357
<i>PathFigure</i>	266	Strahlendes Material	327
StartStyle()		Stretch	134
<i>PromptBuilder</i>	403	<i>HorizontalAlignment</i>	52, 66
<i>PromptStyle</i>	406	<i>Image</i>	134
Startup		<i>VerticalAlignment</i>	66
<i>Application</i>	166, 167	StringAnimationUsingKeyFrames	374, 378
StartupEventArgs	166	Stroke	
<i>Args</i>	169	<i>Path</i>	260
StartupUri		<i>Shape</i>	253
<i>Application</i>	166	StrokeEndLineCap	
StateChanged		<i>Shape</i>	257
<i>Window</i>	178	StrokeLineJoin	
STAThread	162	<i>Shape</i>	258
StaticResource	175	StrokeStartLineCap	
Statische Ressource	174, 175	<i>Shape</i>	257
StatusBar	126	StrokeThickness	
Statusleiste	126	<i>Path</i>	260
<i>Platzierung</i>	126	<i>Shape</i>	253

Style	201, 419	StylusMove	146
<i>abgeleiteter Style</i>	207	StylusOutOfRange	146
<i>als Ressource</i>	203	StylusSystemGesture	146
<i>BasedOn</i>	207	StylusSystemGestureEventArgs	
<i>Basis-Style</i>	207	<i>SystemGesture</i>	147
<i>benannter</i>	202	StylusUp	146
<i>Definition</i>	203	Subscript	
<i>Eigenschaft</i>	203	<i>BaselineAlignment</i>	433
<i>Eigenschaftswert</i>	203	Superscript	
<i>EventSetter</i>	210	<i>BaselineAlignment</i>	433
<i>expliziter</i>	202	SweepDirection	
<i>für Typ</i>	205	<i>ArcSegment</i>	267
<i>für verwandte Typen</i>	208	Symbolleiste	123
<i>Gültigkeitsbereich</i>	206, 208	<i>Platzierung</i>	124
<i>impliziter</i>	205	<i>Styles</i>	126
<i>mit Event Trigger</i>	367	Symbolleistencontainer	123
<i>Sammlung</i>	221	<i>ist gesperrt</i>	124
<i>Schlüssel</i>	203	<i>Lage</i>	124
<i>TargetType</i>	205	SynthesisMediaType	401
<i>Triggers</i>	213	System	130, 162
<i>vererben</i>	206, 207	System.Collections.ObjectModel	236
<i>Ziel</i>	205	System.Component.Model	103, 167
<i>Zuordnung</i>	204, 205	System.Data	241
Stylus	145	System.Data.OleDb	241
<i>Anzahl der Tipps</i>	147	System.Drawing	451
<i>Geste</i>	147	System.Drawing.Color	453
<i>Schaltfläche</i>	146, 147	System.Drawing.Font	454
StylusButton		System.IO	194, 223, 394
<i>Guid</i>	147	System.Media	172, 173, 401
<i>Name</i>	147	System.Printing	446
<i>StylusButtonState</i>	147	System.Speech	407
StylusButtonDown	146	System.Speech.Recognition	407
StylusButtonState		System.Speech.Synthesis	398
<i>StylusButton</i>	147	System.Threading	103
StylusButtonDownUp	146	System.Windows	162
StylusDevice		System.Windows.Controls	164, 232
<i>StylusEventArgs</i>	146	System.Windows.Controls.Primitives	71
StylusDown	146	System.Windows.Forms	193, 449
StylusDownEventArgs		System.Windows.Input	158
<i>TapCount</i>	147	System.Windows.Markup	223, 444
StylusEnter	146	System.Windows.Media.Animation	347
StylusEventArgs		System.Windows.Media.Media3D	301
<i>InAir</i>	146	System.Xaml	454
<i>Inverted</i>	146	SystemGesture	
<i>StylusDevice</i>	146	<i>StylusSystemGestureEventArgs</i>	147
StylusInAirMove	146	SystemSound	390
StylusInRange	146	<i>Play()</i>	391
StylusLeave	146	SystemSounds	391

Systemton
abspielen 390

T

TabControl 116
Auswahl gewechselt 116
Nummer der ausgewählten Karte 116
Platzierung 116
SelectedIndex 116
Selection_Changed 116
TabStripPlacement 116
Tabelle 416, 425
ändern 428
hinzufügen 428
Rahmen 427
Spalte 425
Zeile 425
Zeilengruppe 425
Zelle 425
Zellen überspannen 427
TableItem 116
Beschriftung 116
Header 116
Table 416, 425
Columns 425
RowGroups 425
TableCell 425
ColumnSpan 427
RowSpan 427
TableCellCollection 425
TableColumnCollection 425
TableRow 425
Cells 425
TableRowCollection 425
TableRowGroup 425
Rows 425
TableRowGroupCollection 425
Tablet-PC 463
TabStripPlacement
Dock 116
TabControl 116
Tap
SystemGesture 147
TapCount
StylusDownEventArgs 147
Target
Label 75

TargetName
Hyperlink 190
Setter 217
Storyboard 358
TargetProperty
Storyboard 357
TargetType
ControlTemplate 216
Style 205
Tastatur 139
Tastatursteuerung 57
Taste
Alt 57
bedienen 139
Bindung 156
Enter 66
Ereignis 140
ESC 66
F1 18
Info über 139
Return 66
wiederholt gedrückt 140
Tastenkombination 155, 159
Template 215, 246
TemplateBinding 218
Text
Änderung prüfen 82
anhängen 78
ComboBox 92
Ein- und Ausgabe 74
einfügen 79
eingeben 81
ganz markieren 83
geschützter 83
Länge begrenzen 84
markierter Teil 83
Position 79
Spracherkennung 409
SynthesisMediaType 401
teilweise markieren 83
TextBlock 76
TextAlignment
Block 419
Textbereich 442
TextBlock 76
in FixedDocument 445
Inhalt 76
mit Inlines 431

TextBlock (Forts.)		
<i>Text</i>	76	
<i>TextWrapping</i>	77	
TextBox	81	
<i>AcceptsReturn</i>	82	
<i>MaxLength</i>	84	
<i>mehrzeilig</i>	81	
<i>mit Scrollbalken</i>	82	
<i>SelectAll()</i>	83	
<i>SelectedText</i>	83	
<i>SelectionLength</i>	83	
<i>SelectionStart</i>	83	
<i>TextChanged</i>	81	
<i>TextWrapping</i>	82	
<i>VerticalScrollBarVisibility</i>	82	
TextChange	82	
TextChanged		
<i>TextBox</i>	81	
TextChangedEventArgs	82	
<i>Changes</i>	82	
TextDecorations		
<i>Inline</i>	433	
Text-Editor	84	
Text-Eingabe	84	
TextElement		
<i>FontFamily</i>	419	
TextIndent		
<i>Paragraph</i>	419	
TextMarkerStyle	423	
TextPointer	442	
TextRange	442, 443	
Textur		
<i>für 3D-Grafik</i>	326, 329	
TextureCoordinates		
<i>MeshGeometry3D</i>	326	
TextWrapping		
<i>TextBlock</i>	77	
<i>TextBox</i>	82	
Thickness	63, 65, 109	
ThicknessAnimation	347	
ThicknessAnimationUsingKeyFrames	374	
Thin		
<i>FontWeights</i>	58	
Thread		
<i>Sleep()</i>	103	
ThreeDBorderWindow		
<i>WindowStyle</i>	191	
Thumb	105, 108	
TickFrequency		
<i>Slider</i>	105	
TickPlacement		
<i>Both</i>	105	
<i>BottomRight</i>	105	
<i>Slider</i>	105	
<i>TopLeft</i>	105	
Ticks		
<i>Slider</i>	107	
Tile	279	
<i>TileMode</i>	281	
TileMode		
<i>ImageBrush</i>	279	
TimeSeekOrigin	361	
TimeSpan	351	
<i>Parse()</i>	353	
Title		
<i>NavigationWindow</i>	189	
<i>Window</i>	176	
To		
<i>AnimationTimeline</i>	348	
ToggleBold		
<i>EditingCommands</i>	154, 441	
ToggleButton	69	
<i>Checked</i>	70	
<i>IsChecked</i>	70	
<i>IsThreeState</i>	70	
<i>Unchecked</i>	70	
<i>Zustand</i>	70	
ToggleUnderline		
<i>EditingCommands</i>	441	
ToLongDateString()		
<i>DateTime</i>	133	
ToolBar	123	
<i>ComboBoxStyleKey</i>	126	
ToolBarTray	123	
<i>IsLocked</i>	124	
<i>Orientation</i>	124	
ToolTip	79	
ToolWindow		
<i>WindowStyle</i>	191	
Top		
<i>Canvas</i>	32	
TopLeft		
<i>AutoToolTipPlacement</i>	107	
<i>TickPlacement</i>	105	
TopMost		
<i>Window</i>	178	

TopProperty		
<i>Canvas</i>	34	
ToShortDateString()		
<i>DateTime</i>	133	
TotalManipulation		
<i>ManipulationCompletedEventArgs</i>	151	
Touch		
<i>Ereignis</i>	147, 148	
TouchDevice	147	
TouchDown	148, 150	
TouchEnter	148	
TouchEventArgs	148	
TouchLeave	148	
TouchMove	148, 150	
TouchPoint	148	
TouchPointCollection	148	
Touchscreen	147, 463	
TouchUp	148, 150	
ToXml()		
<i>PromptBuilder</i>	404, 406	
Track	105	
Trägheit		
<i>bei Manipulation</i>	149	
Transform	282	
<i>GeometryModel3D</i>	334	
Transform3D	332	
Transform3DGroup	339	
Transformation	282	
<i>3D-Grafik</i>	332	
<i>animieren</i>	362	
<i>mit Verschiebung</i>	283	
<i>ohne Verschiebung</i>	283	
<i>Ursprung</i>	283	
Transformationsgruppe		
<i>2D-Grafik</i>	291	
<i>3D-Grafik</i>	339	
TransformGroup	291	
<i>Children</i>	292	
TranslateTransform	290	
<i>X</i>	290	
<i>Y</i>	290	
TranslateTransform3D	334	
Translation		
<i>ManipulationDelta</i>	150	
Transparenz	191, 274, 292	
<i>animieren</i>	357	
<i>Effekt</i>	294	
<i>gleitende</i>	294	
Transparenz (Forts.)		
<i>Maske</i>	294	
TreeView	94	
<i>alle übergeordneten Elemente</i>	98	
<i>alle untergeordneten Elemente</i>	98	
<i>ausgewählter Eintrag</i>	96	
<i>Auswahl gewechselt</i>	97	
<i>Eintrag</i>	94	
<i>Eintrag anhängen</i>	99	
<i>Eintrag einfügen</i>	99	
<i>SelectedItemChanged</i>	96	
<i>vorher ausgewählter Eintrag</i>	97	
TreeViewItem	94	
<i>Beschriftung</i>	96	
<i>Collapsed</i>	96	
<i>Expanded</i>	96	
<i>ExpandSubtree()</i>	100	
<i>Header</i>	96	
<i>IsExpanded</i>	96	
<i>IsSelected</i>	96	
<i>ist aufgeklappt</i>	96	
<i>ist ausgewählt</i>	96	
<i>klappt auf</i>	96	
<i>klappt zu</i>	96	
<i>Selected</i>	96	
<i>Unselected</i>	96	
<i>Untereinträge aufklappen</i>	100	
<i>wurde abgewählt</i>	96	
<i>wurde ausgewählt</i>	96	
Triangle		
<i>PenLineCap</i>	258	
TriangleIndices		
<i>MeshGeometry3D</i>	305	
Trigger	212, 366	
<i>Bedingung</i>	213, 215	
<i>Control Template</i>	217	
<i>DataTemplate</i>	249	
<i>für Daten</i>	248	
<i>für Eigenschaft</i>	212	
<i>für Ereignis</i>	365	
<i>in Style</i>	368	
<i>Property</i>	213	
<i>Style</i>	213	
<i>Value</i>	213	
<i>Zielelement</i>	217	
TriggerCollection	366	
TwoFingerTap		
<i>SystemGesture</i>	147	

TwoPage	
<i>FlowDocumentReaderViewingMode</i>	417
TwoWay	
Mode	228
Typ	
ermitteln	29
Umwandlung	21
Type Converter	21
typeof()	472
Typ-Style	205
mit Control Template	219
vererben	207

U

UIElement	429
Umgebungslicht	305, 314
Umschalter	69, 71
einstellen	74
Unchecked	
CheckBox	70
RadioButton	71
ToggleButton	70
Underline	
Inline-Typ	431
Undurchsichtigkeit	292
Uniform	
KeyTime	375
Stretch	134
UniformToFill	
Stretch	134
Union	
GeometryCombineMode	263
Unloaded	
Window	166
Unselected	
ListBoxItem	86
TreeViewItem	96
Unterelement	21
Unterstreich	431
Art	433
Unterstrich	
Tastatursteuerung	57, 119
Unterteilung	64, 118
update	
SQL	243
Update()	
OleDbAdapter	243

UpdateSource()	
BindingExpression	228
UpdateSourceTrigger	
Binding	228
UpDirection	
OrthographicCamera	308, 313
UpperLatin	
TextMarkerStyle	423
UpperRoman	
TextMarkerStyle	423
Uri	173, 186
UriKind	186
Relative	173

V

Validate()	
ValidationRule	232
ValidationResult	232
ValidationRules	232
Add()	231
Binding	230
Validate()	232
Value	
Condition	215
DataTrigger	249
ProgressBar	102
Setter	203
Slider	105
Trigger	213
ValueChanged	
ScrollBar	108
Slider	105
Vector3D	305
Vector3DAnimation	347
Vector3DAnimationUsingKeyFrames	374
VectorAnimation	347
VectorAnimationUsingKeyFrames	374
Vektorgrafik	16
Verschiebung	
2D-Grafik	290
3D-Grafik	334
Touchscreen	148
Wert	290
Vertical	
Orientation	38
VerticalAlignment	66
VerticalAnchor	436

VerticalContentAlignment	66
VerticalOffset	
ContextMenu	121
VerticalScrollBarVisibility	
ScrollViewer	38
TextBox	82
Verzeichnis auswählen	
Dialogfeld	453
Verzierung	298
Video-Ausgabe	396
Videodatei	
in Geometrie	271
VideoDrawing	271
View	
ListView	240
Viewbox	
ImageBrush	279
ViewboxUnits	
ImageBrush	281
ViewingMode	
FlowDocumentReader	417
Viewport	
ImageBrush	279
Viewport2DVisual3D	324
Geometry	326
IsVisualHostMaterial	326
Visual	326
Viewport3D	304
Children	305
ViewportUnits	
ImageBrush	281
Visibility	59, 113, 126, 128
Visible	
Visibility	59
Visual	
Viewport2DVisual3D	326
Visual Basic	19
Visual C#	19
Visual Studio	17
Volume	
MediaPlayer	392
PromptVolume	406
SpeechSynthesizer	399
Vorlage	201
WPF-Anwendung	18
WPF-Browseranwendung	193
Vorwärts	
Navigation	184

W

WAV-Datei	173
abspielen	173, 387, 391
Dateiname	173
laden	388
WaveAudio	
SynthesisMediaType	401
WebBrowser	136
CanGoBack()	138
CanGoForward()	138
GoBack()	138
GoForward()	138
LoadCompleted	136
Navigate()	136
Navigated	136
NavigateToString()	136
Source	136
Weichzeichner	297
WheelClick	
MouseAction	157
Width	38, 57
ColumnDefinition	50
Image	134
ListBox	86
OrthographicCamera	305
SizeToContent	179
Tabellenzelle	427
Window	176
WidthAndHeight	
SizeToContent	179
Window	22, 176
AllowsTransparency	191
Closed	166
Closing	166
DialogResult	182
DragMove()	192
Height	176
Initialized	166, 222
IsLoaded	73
Loaded	127, 166
LocationChanged	178
Main()	163
MouseMove	127
Owner	181
ResizeMode	177
Resources	175
Show()	181

Window (Forts.)		
<i>ShowDialog()</i>	181	
<i>ShowInTaskbar</i>	177	
<i>SizeChanged</i>	127, 178	
<i>SizeToContent</i>	179	
<i>StateChanged</i>	178	
<i>Title</i>	176	
<i>Topmost</i>	178	
<i>Unloaded</i>	166	
<i>Width</i>	176	
<i>Window_Closing()</i>	243	
<i>WindowStartupLocation</i>	177	
<i>WindowState</i>	137	
<i>WindowStyle</i>	191	
Window_Closing()		
<i>Window</i>	243	
Windows		
<i>Druckdialog</i>	445	
<i>ModifierKeys</i>	157	
<i>Spracherkennung</i>	408	
Windows 8	463	
Windows App Cert Kit	483	
Windows Forms		
<i>in WPF-Anwendung</i>	449	
<i>mit WPF-Element</i>	454	
<i>Standard-Dialogfeld in WPF</i>	451	
Windows Phone	463	
Windows Presentation Foundation	15	
Windows Presentation Foundation-Host ...	192	
Windows Store-App		
<i>beenden</i>	465	
<i>Navigation</i>	470, 474	
<i>prüfen</i>	483	
<i>Styles</i>	467	
Windows Store-Apps	463	
WindowsBase	454	
WindowsFormsHost	449	
WindowsFormsIntegration	449, 454	
Windows-Spracherkennung		
<i>integrierte</i>	410	
WindowStartupLocation		
<i>CenterOwner</i>	182	
<i>Window</i>	177	
WindowState		
<i>Window</i>	137	
WindowStyle		
<i>Window</i>	191	
WindowTitle		
<i>Page</i>	185	
Wochentag	132	
WorkerReportsProgress		
<i>BackgroundWorker</i>	103	
WPF	15	
<i>Eigenschaften</i>	15	
<i>Vorteile</i>	15	
WPF-Anwendung		
<i>Minimal-Aufbau</i>	161	
WPF-Browseranwendung	192	
WPF-Interoperabilität		
<i>Toolbox-Kategorie</i>	455	
Wrap		
<i>TextWrapping</i>	77	
WrapPanel	39	
<i>einheitliche Größe</i>	41	
<i>FlowDirection</i>	115	
<i>Orientierung</i>	41	
<i>Richtung</i>	40, 115	
WrapWithOverflow		
<i>TextWrapping</i>	77	
Würfel		
<i>in 3D-Grafik</i>	308	
X		
X		
<i>Point</i>	257	
<i>TranslateTransform</i>	290	
x:Class	22	
x:Key	203, 216	
x:Name	21	
x:Null	70	
x:Type	205, 248	
X1		
<i>Line</i>	255	
X2		
<i>Line</i>	255	
XAML	17	
<i>Attribut</i>	21	
<i>Editor</i>	19	
<i>Erweiterung</i>	26	
<i>mit Programmiercode</i>	19	
<i>Paketdatenformat</i>	442	
XAML Browser Application	192	
XamlPackage		
<i>DataFormats</i>	442	

XamlReader	223
<i>Load()</i>	223
XBAP	192
XML	
<i>Knoten</i>	23
XML-Datei	
<i>mit SSML</i>	404, 406
xmlns	22
xmlns:x	22
Xor	
<i>GeometryCombineMode</i>	263

Y

Y	
<i>Point</i>	257
<i>TranslateTransform</i>	290
Y1	
<i>Line</i>	255
Y2	
<i>Line</i>	255

Z

z-Achse	301
Zahlenbereich	
<i>darstellen</i>	101
<i>Wert darstellen</i>	105
Zahlenwerte	
<i>darstellen</i>	101
Zeichenkette	
<i>animieren</i>	378
Zeilenumbruch	68, 77, 431
<i>steuern</i>	77, 82
Zeitspanne	353
Zeitverzögerung	103
Zentimeter	419
ZIndex	
<i>Panel</i>	32, 284
ZIndexProperty	
<i>Canvas</i>	34
Zoom	
<i>FlowDocumentReader</i>	417
ZoomIncrement	417
Zweidimensionale Grafik	251
Zwischenablage	81