

Inhalt

Vorwort	21
Hinweise zum Buch	25

TEIL I WPF-GRUNDLAGEN UND KONZEPTE

1	Einführung in die WPF	39
1.1	Die WPF und das .NET Framework	39
1.1.1	Die WPF im .NET Framework 3.0	39
1.1.2	Die WPF und das .NET Framework 3.5	40
1.1.3	Die WPF und das .NET Framework 4.0	41
1.1.4	Die WPF als zukünftiges Programmiermodell	43
1.1.5	Stärken und Eigenschaften der WPF	45
1.1.6	Auf Wiedersehen GDI+	47
1.2	Von Windows 1.0 zur Windows Presentation Foundation	48
1.2.1	Die ersten Wrapper um die Windows-API	48
1.2.2	Windows Forms und GDI+	48
1.2.3	Die Windows Presentation Foundation	49
1.3	Die Architektur der WPF	52
1.3.1	MilCore – die »Display Engine«	53
1.3.2	WindowsBase	56
1.3.3	PresentationCore	56
1.3.4	PresentationFramework	57
1.3.5	Vorteile und Stärken der WPF-Architektur	57
1.4	Konzepte	59
1.4.1	XAML	60
1.4.2	Dependency Properties	62
1.4.3	Routed Events	66
1.4.4	Commands	69
1.4.5	Styles und Templates	72
1.4.6	3D	73
1.5	Zusammenfassung	75
2	Das Programmiermodell	77
2.1	Einführung	77
2.2	Grundlagen der WPF	78
2.2.1	Namespaces	78

2.2.2	Assemblies	79
2.2.3	Die Klassenhierarchie	79
2.3	Projektvorlagen in Visual Studio 2010	87
2.3.1	WPF-Anwendung (Windows)	88
2.3.2	WPF-Browseranwendung (Web)	89
2.3.3	WPF-Benutzersteuerelementbibliothek	89
2.3.4	Benutzerdefinierte WPF-Steuerelementbibliothek	90
2.4	Windows-Projekte mit Visual Studio 2010	91
2.4.1	Ein Windows-Projekt mit XAML und C#	92
2.4.2	Eine reine Codeanwendung (C#)	105
2.4.3	Eine reine, kompilierte XAML-Anwendung	107
2.4.4	Best Practice	110
2.5	Application, Dispatcher und Window	110
2.5.1	Die Klasse Application	110
2.5.2	Die Klasse Dispatcher	118
2.5.3	Fenster mit der Klasse Window	124
2.6	Zusammenfassung	138

3	XAML	141
3.1	Einführung	141
3.2	<Warum>XAML?</Warum>	141
3.3	Elemente und Attribute	144
3.4	Namespaces	145
3.4.1	Der XML-Namespace der WPF	146
3.4.2	Der XML-Namespace für XAML	149
3.4.3	Über Namespace-Alias	150
3.4.4	XAML mit eigenen CLR-Namespaces erweitern	151
3.5	Properties in XAML setzen	154
3.5.1	Die Attribut-Syntax	155
3.5.2	Die Property-Element-Syntax	156
3.5.3	Die Content-Property (Default-Property)	157
3.5.4	Die Attached-Property-Syntax	159
3.6	Type-Converter	159
3.6.1	Vordefinierte Type-Converter	160
3.6.2	Eigene Type-Converter implementieren	162
3.6.3	Type-Converter in C# verwenden	166
3.7	Markup-Extensions	168
3.7.1	Verwenden von Markup-Extensions in XAML und C#	168
3.7.2	XAML-Markup-Extensions	171
3.7.3	Markup-Extensions der WPF	172

3.8	XAML-Spracherweiterungen	173
3.9	Collections in XAML	178
3.9.1	Collections, die IList implementieren	179
3.9.2	Collections, die IDictionary implementieren	180
3.10	XamlReader und XamlWriter	183
3.10.1	XAML mit XamlReader dynamisch laden	183
3.10.2	Objekte mit XamlWriter in XAML serialisieren	185
3.11	Zusammenfassung	187

4 Der Logical und der Visual Tree 189

4.1	Einleitung	189
4.2	Zur Veranschaulichung verwendete Komponenten	192
4.2.1	Der InfoDialog von FriendStorage	192
4.2.2	Die Anwendung XAMLPadExtensionClone	193
4.3	Der Logical Tree	195
4.3.1	Der Logical Tree des InfoDialogs	195
4.3.2	Für den Logical Tree verantwortliche Klassen	199
4.3.3	Die Klasse LogicalTreeHelper	203
4.3.4	NameScopes, FindName und FindLogicalNode	208
4.4	Der Visual Tree	215
4.4.1	Der Visual Tree des InfoDialogs	216
4.4.2	Eigene Klassen im Visual Tree	219
4.4.3	Die Klasse VisualTreeHelper	222
4.4.4	Der Visual Tree und das Rendering	225
4.5	Zusammenfassung	230

5 Controls 233

5.1	Einleitung	233
5.2	Die Klasse Control	236
5.3	ContentControls	238
5.3.1	Buttons	241
5.3.2	Labels	248
5.3.3	ToolTips anzeigen	250
5.3.4	Scrollen mit ScrollViewer	253
5.3.5	WPF- und HTML-Inhalte mit Frame darstellen	256
5.3.6	ContentControls mit Header	258
5.4	ItemsControls	261
5.4.1	ItemsControls mit Header	264
5.4.2	Baumansicht mit der TreeView	267

5.4.3	Navigation über Menüs	272
5.4.4	Elemente mit einem Selector auswählen	277
5.4.5	Eine StatusBar mit Informationen	284
5.4.6	»Alternating Rows« mit AlternationCount	285
5.5	Controls zur Textdarstellung und -bearbeitung	286
5.5.1	TextBox zum Editieren von Text	287
5.5.2	RichTextBox für formatierten Text	289
5.5.3	PasswordBox für maskierten Text	289
5.5.4	TextBlock zur Anzeige von Text	290
5.5.5	Zeichnen mit dem InkCanvas	290
5.6	Datum-Controls	293
5.6.1	Calendar	293
5.6.2	DatePicker	296
5.7	Range-Controls	297
5.7.1	Bereich mit Slider auswählen	298
5.7.2	ProgressBar zur Statusanzeige	299
5.7.3	Scrollen mit der ScrollBar	300
5.8	Sonstige, einfachere Controls	300
5.8.1	Decorator zum Ausschmücken	300
5.8.2	Bilder mit der Image-Klasse darstellen	302
5.8.3	Einfaches Popup anzeigen	303
5.9	Zusammenfassung	306

6 Layout 309

6.1	Einleitung	309
6.2	Der Layoutprozess	310
6.2.1	Die zwei Schritte des Layoutprozesses	310
6.2.2	MeasureOverride und ArrangeOverride	312
6.2.3	Ein eigenes Layout-Panel (DiagonalPanel)	313
6.2.4	Zusammenfassung des Layoutprozesses	318
6.3	Layoutfunktionalität von Elementen	319
6.3.1	Width und Height	319
6.3.2	Margin und Padding	320
6.3.3	Alignments	322
6.3.4	Die Visibility-Property	324
6.3.5	Die UseLayoutRounding-Property	325
6.3.6	Transformationen	327
6.4	Panels	338
6.4.1	Die Klasse Panel	339
6.4.2	Canvas	341

6.4.3	StackPanel	343
6.4.4	WrapPanel	345
6.4.5	DockPanel	346
6.4.6	Grid	348
6.4.7	Primitive Panels	361
6.4.8	Übersicht der Alignments in den verschiedenen Panels	363
6.4.9	Wenn der Platz im Panel nicht ausreicht	365
6.5	Das Layout von FriendStorage	367
6.5.1	Das Hauptfenster aus Benutzersicht	367
6.5.2	Das Hauptfenster aus Entwicklersicht	369
6.5.3	Animation des Freunde-Explorers	378
6.6	Zusammenfassung	382

7 Dependency Properties 385

7.1	Einleitung	385
7.2	Die Keyplayer	386
7.2.1	DependencyObject und DependencyProperty	386
7.2.2	Was ist die Property Engine?	388
7.3	Dependency Properties	389
7.3.1	Eine Dependency Property implementieren	390
7.3.2	Metadaten einer Dependency Property	392
7.3.3	Validieren einer Dependency Property	399
7.3.4	Die FontSize-Property als Ziel eines Data Bindings	402
7.3.5	Existierende Dependency Properties verwenden	404
7.3.6	Read-only-Dependency-Properties implementieren	406
7.3.7	Ermittlung des Wertes einer Dependency Property	409
7.3.8	Lokal gesetzte Werte löschen	410
7.3.9	Überblick der Quellen mit DependencyPropertyHelper	412
7.3.10	Auf Änderungen in existierenden Klassen lauschen	412
7.4	Attached Properties	413
7.4.1	Eine Attached Property implementieren	414
7.4.2	Ein einfaches Panel mit Attached Properties	417
7.4.3	Bekannte Vertreter	421
7.5	Zusammenfassung	423

8 Routed Events 427

8.1	Einleitung	427
8.2	Die Keyplayer	428
8.2.1	Die Klassen RoutedEvent und EventManager	428

8.2.2	Die Routing-Strategie	429
8.2.3	Das Interface <code>IInputElement</code>	432
8.2.4	Die Klasse <code>RoutedEventArgs</code>	433
8.2.5	Das Event System	434
8.3	Eigene Routed Events	435
8.3.1	Ein Routed Event implementieren	435
8.3.2	Das Routed Event als Attached Event verwenden	440
8.3.3	Existierende Routed Events in eigenen Klassen nutzen	442
8.3.4	Instanz- und Klassenbehandlung	443
8.4	Die <code>RoutedEventArgs</code> im Detail	448
8.4.1	Sender vs. Source und <code>OriginalSource</code>	449
8.4.2	Die <code>Handled-Property</code>	451
8.5	Routed Events der WPF	453
8.5.1	Tastatur-Events	454
8.5.2	Maus-Events	458
8.5.3	Stylus-Events (Stift)	462
8.5.4	Multitouch-Events	463
8.5.5	Die statischen Mitglieder eines <code>FrameworkElements</code>	466
8.6	Zusammenfassung	467

9 Commands 469

9.1	Einleitung	469
9.2	Die Keyplayer	470
9.2.1	Das Interface <code>ICommand</code>	470
9.2.2	Das Interface <code>ICommandSource</code>	471
9.3	Eigene Commands mit <code>ICommand</code>	472
9.3.1	Ein Command implementieren	472
9.3.2	Das Command verwenden	473
9.3.3	Das Command von der Logik entkoppeln	474
9.4	Die »wahren« Keyplayer	477
9.4.1	Die Klassen <code>RoutedCommand</code> / <code>RoutedUICommand</code>	478
9.4.2	Der <code>CommandManager</code>	480
9.4.3	Die Klasse <code>CommandBinding</code>	481
9.4.4	Elemente mit einer <code>CommandBindings-Property</code>	482
9.4.5	Das Zusammenspiel der Keyplayer	483
9.5	Eigene Commands mit der Klasse <code>RoutedUICommand</code>	486
9.5.1	Die eigenen Commands in <code>FriendStorage</code>	487
9.5.2	Commands mit <code>InputGestures</code> versehen	488
9.5.3	<code>CommandBindings</code> zum Window-Objekt hinzufügen	490
9.5.4	Die Commands im Menü und in der <code>ToolBar</code> verwenden	492

9.6	Built-in-Commands der WPF	497
9.6.1	Built-in-Commands in FriendStorage	498
9.6.2	Bestehende Commands mit InputBindings auslösen	500
9.6.3	Controls mit integrierten CommandBindings	502
9.7	Das Model-View-ViewModel-Pattern (MVVM)	505
9.7.1	Die Idee des Model-View-Controller-Patterns (MVC)	506
9.7.2	Die Idee des Model-View-ViewModel-Patterns (MVVM)	507
9.7.3	Ein MVVM-Beispiel	509
9.8	Zusammenfassung	514

TEIL II FORTGESCHRITTENE TECHNIKEN

10 Ressourcen 519

10.1	Logische Ressourcen	519
10.1.1	Logische Ressourcen definieren und verwenden	520
10.1.2	Die Suche nach Ressourcen im Detail	523
10.1.3	Elemente als Ressourcen verwenden	527
10.1.4	Statische Ressourcen	530
10.1.5	Dynamische Ressourcen	532
10.1.6	Ressourcen in separate Dateien auslagern	536
10.1.7	Logische Ressourcen in FriendStorage	539
10.2	Binäre Ressourcen	541
10.2.1	Binäre Ressourcen im .NET Framework	542
10.2.2	Binäre Ressourcen bei der WPF	545
10.2.3	Die Pack-URI-Syntax	548
10.2.4	Auf Dateien im Anwendungsverzeichnis zugreifen	549
10.2.5	In C# auf binäre Ressourcen zugreifen	550
10.2.6	Lokalisierung von WPF-Anwendungen	553
10.2.7	Eine binäre Ressource als Splashscreen	563
10.3	Zusammenfassung	566

11 Styles, Trigger und Templates 569

11.1	Einleitung	569
11.2	Styles	570
11.2.1	Grundlagen und Keyplayer	570
11.2.2	Styles als logische Ressourcen definieren	573
11.2.3	Einen Style für verschiedene Typen verwenden	577
11.2.4	Bestehende Styles erweitern	579

11.2.5	Setter und EventSetter	581
11.2.6	Styles und Trigger	584
11.3	Trigger	584
11.3.1	Property-Trigger	585
11.3.2	DataTrigger	590
11.3.3	EventTrigger	592
11.3.4	Komplexe Bedingungen mit Triggern	597
11.4	Templates	599
11.4.1	Arten von Templates	599
11.4.2	Layout mit ItemsPanelTemplate	601
11.4.3	Daten mit DataTemplates visualisieren	602
11.4.4	Das Aussehen von Controls mit ControlTemplates anpassen	606
11.4.5	Das Default-ControlTemplate eines Controls	609
11.4.6	Verbindung zwischen Control und Template	612
11.4.7	Two-Way-Contract zwischen Control und Template	616
11.4.8	VisualStateManager statt Trigger verwenden	620
11.4.9	Templates in C#	631
11.5	Styles, Trigger & Templates in FriendStorage	632
11.5.1	Der Next-Button	633
11.5.2	Die Image-Objekte der Toolbar-Buttons	635
11.5.3	Die DataGridRows des Freunde-Explorers	637
11.6	Zusammenfassung	639

12 Daten 643

12.1	Einleitung	643
12.2	Data Binding	644
12.2.1	Data Binding in XAML	644
12.2.2	Data Binding in C#	645
12.2.3	Die Binding-Klasse im Detail	646
12.2.4	Der DataContext	649
12.2.5	Die Path-Property im Detail	650
12.2.6	Die Richtung des Bindings	652
12.2.7	Der UpdateSourceTrigger	654
12.2.8	Die BindingExpression	655
12.2.9	Bindings entfernen	657
12.2.10	Debugging von Data Bindings	658
12.3	Datenquellen eines Data Bindings	660
12.3.1	Binding an die Dependency Properties eines Elements	660
12.3.2	Binding an einfache .NET Properties	661
12.3.3	Binding an logische Ressourcen	663

12.3.4	Binding an Quellen unterschiedlichen Typs	664
12.3.5	Binding an relative Quellen mit RelativeSource	667
12.3.6	Binding der Target-Property an mehrere Quellen	669
12.3.7	DataSourceProvider für Objekte und XML	674
12.3.8	Binding an XLinQ	681
12.4	Data Binding an Collections	682
12.4.1	Der Fallback-Mechanismus	682
12.4.2	Die CollectionViews der WPF	686
12.4.3	Die DefaultView	690
12.4.4	Daten filtern, sortieren und gruppieren	691
12.4.5	Hinzufügen und Löschen von Daten	697
12.4.6	Mehrere Collections als Datenquelle verwenden	698
12.4.7	Binding an ADO.NET DataSet	699
12.5	Benutzereingaben validieren	702
12.5.1	Validieren mit ExceptionValidationRule	704
12.5.2	Validieren mit eigener ValidationRule	706
12.5.3	Validieren mit DataErrorValidationRule	707
12.5.4	Die Validation-Klasse	709
12.5.5	Validieren mehrerer Bindings mit BindingGroup	711
12.6	Das DataGrid	720
12.6.1	Die verwendeten Testdaten	721
12.6.2	Autogenerieren von Columns	723
12.6.3	Unterschiedliche Column-Typen	725
12.6.4	Columns manuell zum DataGrid hinzufügen	728
12.6.5	Die Breite einer Column	730
12.6.6	Columns mit der DataGridTemplateColumn	731
12.6.7	RowDetails anzeigen	733
12.6.8	Daten gruppieren	735
12.6.9	Die Auswahlmöglichkeiten festlegen	737
12.6.10	Auf ausgewählte Daten zugreifen	737
12.6.11	Bearbeiten von Daten	739
12.6.12	Daten im DataGrid validieren	740
12.6.13	Sonstige Eigenschaften des DataGrids	744
12.7	Daten mit DataTemplates visualisieren	745
12.7.1	Auswahl mit DataTemplateSelector	745
12.7.2	Hierarchische DataTemplates	747
12.8	Drag & Drop	750
12.9	Daten in FriendStorage	754
12.9.1	Die Entitäten Friend, Address und FriendCollection	755
12.9.2	Daten im MainWindow	755

12.9.3	Daten im NewFriendDialog	765
12.9.4	Speichern in gezippter .friends-Datei	767
12.10	Zusammenfassung	769

TEIL III REICHHALTIGE MEDIEN UND EIGENE CONTROLS

13	2D-Grafik	773
13.1	Einleitung	773
13.2	Shapes	774
13.2.1	Das Rectangle	775
13.2.2	Die Ellipse	776
13.2.3	Linien mit Line und Polyline	776
13.2.4	Spezielle Formen mit Polygon	778
13.2.5	Ein Außerirdischer aus Shapes	781
13.2.6	Die StrokeXXX-Properties der Shape-Klasse	782
13.2.7	Komplexe Shapes mit Path	785
13.3	Geometries	786
13.3.1	RectangleGeometry und EllipseGeometry	787
13.3.2	LineGeometry	788
13.3.3	Mehrere Geometry-Objekte gruppieren	789
13.3.4	Geometries kombinieren	789
13.3.5	Komplexe Formen mit PathGeometry	791
13.3.6	Die Klasse StreamGeometry	794
13.3.7	Die Path-Markup-Syntax	795
13.3.8	Clipping mit Geometry-Objekten	797
13.4	Drawings	797
13.4.1	GeometryDrawing und DrawingGroup	798
13.4.2	ImageDrawing und VideoDrawing	800
13.4.3	Ein Außerirdischer aus Geometries und Drawings	801
13.5	Programmierung des Visual Layers	803
13.5.1	Die Klasse DrawingContext	804
13.5.2	DrawingVisual einsetzen	805
13.5.3	Visual-Hit-Testing	807
13.6	Brushes	808
13.6.1	Der SolidColorBrush und die Color-Struktur	809
13.6.2	Farbverläufe mit GradientBrushes	811
13.6.3	TileBrushes	815
13.7	Cached Compositions	821
13.7.1	BitmapCache für ein Element aktivieren	822

13.7.2	Nebeneffekte des Cachings	823
13.7.3	Element mit BitmapCacheBrush zeichnen	826
13.8	Effekte	828
13.8.1	Die Effect-Klassen	828
13.8.2	Blur und DropShadow verwenden	828
13.8.3	Properties von BlurEffect und DropShadowEffect	830
13.8.4	Effekte mit eigenen Pixelshadern	831
13.8.5	Pixelshader mit weiteren Konstanten	840
13.9	Bitmaps	843
13.9.1	BitmapSources – Bildquellen	843
13.9.2	Bitmap-Operationen	844
13.9.3	Bitmap-Operationen in FriendStorage	845
13.10	Zusammenfassung	846

14 3D-Grafik 849

14.1	Einleitung	849
14.2	3D im Überblick	850
14.2.1	Inhalte einer 3D-Szene	850
14.2.2	2D und 3D im Vergleich	851
14.3	Die Objekte einer 3D-Szene im Detail	853
14.3.1	Das 3D-Koordinatensystem	853
14.3.2	Der Viewport3D als Fernseher	854
14.3.3	Die richtige Kamera	855
14.3.4	Visual3D-Objekte	859
14.3.5	Model3D-Objekte	860
14.3.6	GeometryModel3D aufbauen	861
14.3.7	Licht ins Dunkel bringen	867
14.3.8	Transformationen	870
14.3.9	Verschiedene Materialien	873
14.3.10	Texturen	874
14.3.11	Normalen	878
14.4	Benutzerinteraktion mit 3D-Objekten	882
14.4.1	Interaktivität in WPF 3.0 mit Visual-Hit-Testing	882
14.4.2	Interaktivität in WPF 3.5 mit UIElement3D	883
14.4.3	Interaktive 2D-Elemente auf 3D-Objekten in WPF 3.5	885
14.5	Komplexe 3D-Objekte	887
14.5.1	Landschaft im Code generieren	887
14.5.2	Kugel erstellen	889
14.5.3	Komplexe 3D-Objekte mit Third-Party-Tools erstellen	891
14.6	Zusammenfassung	892

15 Animationen	895
15.1 Einleitung	895
15.2 Animationsgrundlagen	896
15.2.1 Voraussetzungen für Animationen	896
15.2.2 Übersicht der Animationsarten und -klassen	897
15.2.3 Timelines und Clocks	900
15.2.4 Das Interface IAnimatable	903
15.3 Basis-Animationen in C#	904
15.3.1 Start- und Zielwert mit From, To und By	905
15.3.2 Dauer, Startzeit und Geschwindigkeit	908
15.3.3 Rückwärts und Wiederholen	911
15.3.4 Die Gesamtlänge einer Timeline	912
15.3.5 Wiederholen mit neuen Werten	912
15.3.6 Beschleunigen und Abbremsen	914
15.3.7 Das Füllverhalten einer Animation	915
15.3.8 Animation mit AnimationClock steuern	917
15.3.9 Animationen in FriendStorage	921
15.4 Basis-Animationen in XAML	922
15.4.1 Einfache Animation in XAML	925
15.4.2 Das Storyboard als Timeline-Container	928
15.4.3 Animation mit ControllableStoryboard steuern	931
15.5 Keyframe-Animationen	933
15.5.1 Lineare Keyframe-Animationen	934
15.5.2 SplineKeyframe-Animationen	937
15.5.3 Animationen mit diskreten Keyframes	938
15.6 Pfad-Animationen	942
15.7 Easing Functions	944
15.7.1 Grundlagen der Easing Functions	944
15.7.2 Easing Functions in Basis-Animationen	948
15.7.3 Easing Functions in Keyframe-Animationen	950
15.7.4 Eigene Easing Functions erstellen	952
15.8 Low-Level-Animationen	954
15.9 Zusammenfassung	958
16 Audio und Video	959
16.1 Einleitung	959
16.2 Audio (.wav) mit SoundPlayerAction und SoundPlayer	959
16.2.1 Audio mit SoundPlayerAction (XAML)	960
16.2.2 Audio mit SoundPlayer (C#)	961

16.3	Audio und Video mit MediaPlayer (C#)	963
16.3.1	Einfaches Abspielen	964
16.3.2	Steuerung mit MediaClock und MediaTimeline	967
16.4	Audio und Video mit MediaElement (XAML)	970
16.4.1	Einfaches Abspielen	971
16.4.2	Steuerung mit Methoden (unabhängiger Modus)	972
16.4.3	Steuerung mit MediaTimeline (Clock-Modus)	972
16.4.4	Storyboard mit MediaTimeline und AnimationTimeline	975
16.4.5	Snapshots von Videos	977
16.5	Zusammenfassung	980

17 Eigene Controls 983

17.1	Einleitung	983
17.2	Custom Control	984
17.2.1	Die Struktur eines Custom Controls	985
17.2.2	Der zu erstellende VideoPlayer	988
17.2.3	Klassenname anpassen	988
17.2.4	Template-Parts definieren	990
17.2.5	Dependency Properties erstellen	993
17.2.6	Routed Events implementieren	997
17.2.7	Commands unterstützen	998
17.2.8	Das Aussehen des lookless Controls festlegen	1001
17.2.9	Das Control testen	1006
17.2.10	Optional weitere Theme-Styles anlegen	1007
17.2.11	Templates auf Windows-Ebene definieren	1010
17.3	Custom Control mit Visual States	1014
17.3.1	Visual States im Code implementieren	1014
17.3.2	States für andere sichtbar machen	1017
17.3.3	States im Default-ControlTemplate unterstützen	1018
17.3.4	Den VideoPlayer mit Visual States testen	1020
17.4	User Control	1021
17.4.1	Die Struktur eines User Controls	1021
17.4.2	Das zu erstellende PrintableFriend-Control	1023
17.4.3	UI des Controls definieren	1024
17.4.4	Properties in der Codebehind-Datei erstellen	1025
17.4.5	Die Content-Property festlegen	1027
17.5	Alternativen zu Custom Control und User Control	1028
17.5.1	Wann sollte man die OnRender-Methode überschreiben?	1028
17.5.2	Adorner erstellen und Elemente damit ausschmücken	1028
17.6	Zusammenfassung	1034

18	Text und Dokumente	1037
18.1	Einleitung	1037
18.2	Text	1038
18.2.1	FrameworkContentElement als Basis für Text	1039
18.2.2	Formatierung mit Spans	1040
18.2.3	Formatierung mit den Properties aus TextElement	1042
18.2.4	Elemente im Text mit InlineUIContainer	1045
18.2.5	Fonts und Typefaces	1045
18.2.6	Typographie	1047
18.2.7	Die FormattedText-Klasse	1048
18.2.8	Texteffekte	1049
18.2.9	Nützliche Eigenschaften der TextBlock-Klasse	1051
18.3	Das Text-Rendering beeinflussen	1053
18.3.1	Kleine Zeichen sind schlecht lesbar	1054
18.3.2	Die Schrift führt beim Animieren zu Performance-Problemen	1055
18.3.3	Der Algorithmus für das Anti-Aliasing lässt sich nicht festlegen	1055
18.3.4	Der ClearType-Algorithmus greift nicht immer	1056
18.4	Flow-Dokumente	1058
18.4.1	Die Klasse FlowDocument	1058
18.4.2	Die fünf Block-Arten	1060
18.4.3	Die AnchoredBlocks Figure und Floater	1064
18.4.4	Controls zum Betrachten	1067
18.5	Annotationen	1069
18.6	XPS-Dokumente (Fixed-Dokumente)	1073
18.6.1	FlowDocument als XPS speichern	1074
18.6.2	XPS-Dokument laden und anzeigen	1078
18.6.3	Die Inhalte eines XPS-Dokuments	1079
18.6.4	XPS in C# mit FixedDocument & Co. erstellen	1083
18.7	Drucken	1085
18.7.1	Einfaches Ausdrucken	1085
18.7.2	Drucken mit PrintQueue	1087
18.7.3	Festlegen von Druckeigenschaften mit PrintTicket	1088
18.7.4	Drucken mit PrintDialog	1088
18.8	Dokumente in FriendStorage	1090
18.8.1	Hilfe mit Flow-Dokument	1090
18.8.2	Export der Freundesliste als XPS	1091
18.8.3	Drucken der Freundesliste	1096
18.9	Zusammenfassung	1096

TEIL IV WPF-ANWENDUNGEN UND INTEROPERABILITÄT

19	Windows, Navigation und XBAP	1101
19.1	Einleitung	1101
19.2	Windows-Anwendungen	1102
19.2.1	»Built-in-Dialoge	1102
19.2.2	Anwendungen mit UI Automation automatisieren	1104
19.2.3	Deployment	1120
19.3	Windows-Anwendungen und die Windows 7-Taskbar	1121
19.3.1	Übersicht der Möglichkeiten	1122
19.3.2	Thumb-Buttons im Vorschaufenster	1124
19.3.3	Ein Overlay-Bild auf dem Taskbar-Button	1127
19.3.4	Eine Fortschrittsanzeige auf dem Taskbar-Button	1128
19.3.5	Den Ausschnitt im Thumbnail festlegen	1130
19.3.6	Eine JumpList mit JumpTasks	1131
19.3.7	JumpList mit JumpTasks und JumpPaths	1132
19.3.8	JumpList mit letzten und häufigen Elementen	1134
19.4	Navigationsanwendungen	1136
19.4.1	Container für eine Page	1137
19.4.2	Navigation zu einer Seite/Page	1141
19.4.3	Navigation-Events	1146
19.4.4	Daten übergeben	1149
19.4.5	Daten mittels PageFunction zurückgeben	1153
19.5	XBAP-Anwendungen	1157
19.5.1	FriendViewer als XBAP erstellen	1157
19.5.2	Generierte Dateien	1160
19.5.3	XBAP vs. Loose XAML	1161
19.5.4	XBAP vs. Silverlight	1161
19.6	Zusammenfassung	1162
20	Interoperabilität	1163
20.1	Einleitung	1163
20.2	Unterstützte Szenarien und Grenzen	1164
20.2.1	Mögliche Interoperabilitätsszenarien	1164
20.2.2	Grenzen und Einschränkungen	1165
20.3	Windows Forms	1166
20.3.1	Windows Forms in WPF	1167
20.3.2	WPF in Windows Forms	1175
20.3.3	Dialoge	1176

20.4	ActiveX in WPF	1179
20.5	Win32	1181
20.5.1	Win32 in WPF	1182
20.5.2	WPF in Win32	1192
20.5.3	Dialoge	1196
20.5.4	Win32-Nachrichten in WPF abfangen	1201
20.6	Direct3D in WPF	1204
20.6.1	Voraussetzungen und Konfiguration	1204
20.6.2	Die Direct3D-Oberfläche integrieren	1206
20.7	Zusammenfassung	1210
Index	1211