

Inhalt

Vorwort	9
1 Einführung	11
1.1 Windows-Forms-Glossar.....	11
1.2 Einführungsbeispiel	13
1.3 Weitergabe von Windows-Forms-Programmen	20
2 Standardsteuerelemente	21
2.1 Überblick	21
2.2 Eigenschaften und Methoden.....	25
2.3 Ereignisse.....	35
2.4 Buttons	40
2.5 Beschriftungsfelder (Label, LinkLabel)	43
2.6 Textfeld (TextBox)	45
2.7 Formatiertes Textfeld (RichTextBox)	48
2.8 Maskiertes Textfeld (MaskedTextBox)	52
2.9 Bildfeld (PictureBox).....	53
2.10 Bitmap-Container (ImageList).....	54
2.11 Listenfeld (ListBox)	55
2.12 Listenfeld mit Auswahlkästchen (CheckedListBox).....	62
2.13 Kombinationslistenfeld (ComboBox)	63
2.14 Mehrspaltiges Listenfeld (ListView).....	65
2.15 Hierarchisches Listenfeld (TreeView)	85
2.16 Datumsauswahl (MonthCalendar)	94
2.17 Zeitauswahl (DateTimePicker)	96
2.18 Schiebe- und Zustandsbalken, Drehfelder.....	97
2.19 Datei- und Verzeichnisauswahl (OpenFile-, SaveFile- und FolderBrowserDialog).....	102
2.20 HTML-Seiten anzeigen (WebBrowser).....	104

3	Gestaltung von Benutzeroberflächen	107
3.1	Steuerelemente verankern und andocken	107
3.2	Steuerelemente gruppieren (GroupBox, Panel)	111
3.3	Steuerelemente automatisch anordnen (FlowLayoutPanel, TableLayoutPanel).....	114
3.4	Mehrblättrige Dialoge (TabControl)	118
3.5	Fenster teilen (SplitContainer).....	120
3.6	Fenstergestaltung und Spezialeffekte.....	121
3.7	Standarddialoge	125
3.8	Menüs (MenuStrip).....	127
3.9	Symbolleiste (ToolStrip)	134
3.10	Statusleiste (StatusStrip).....	139
3.11	Steuerelemente beschriften (ToolTip).....	141
3.12	Eigene Online-Hilfe (HelpProvider).....	143
3.13	Icons im Infobereich der Taskbar (NotifyIcon)	146
4	Verwaltung mehrerer Fenster	149
4.1	Grundlagen	149
4.2	My.Forms	155
4.3	Modale Dialoge.....	156
4.4	MDI-Anwendungen.....	160
4.5	Splash-Fenster.....	165
5	Tastatur, Maus, Zwischenablage	167
5.1	Tastatur.....	167
5.2	Maus.....	172
5.3	Zwischenablage	178
5.4	Drag&Drop	181
6	Programmiertechniken	189
6.1	Windows-Forms-Eigenschaften automatisch speichern (My.Settings)	189
6.2	Projektressourcen verwalten (My.Resources)	191
6.3	Lokalisierung von Windows-Anwendungen.....	193
6.4	Validierung von Eingaben (ErrorProvider).....	199
6.5	Owner-Drawn-Steuerelemente	204
6.6	Steuerelemente dynamisch verwalten.....	210
6.7	Neue Steuerelemente programmieren	217
6.8	Steuerelemente zusammensetzen (UserControl).....	220

7	Interna	229
7.1	Code des Windows-Forms-Designers	229
7.2	Anwendungs-Framework	235
7.3	Fenster dynamisch erzeugen	240
7.4	Formulare vererben.....	242
7.5	Betriebssysteminformationen ermitteln	244
7.6	Bildschirmauflösung (DPI)	246
8	Multithreading.....	251
8.1	Nachrichtenschleife (DoEvents)	252
8.2	Timer-Steuerelement.....	254
8.3	Multithreading in Windows-Anwendungen.....	255
8.4	Verzeichniseigenschaften in einem eigenen Thread ermitteln.....	261
8.5	Hintergrundprozesse (BackgroundWorker).....	266
8.6	Apfelmännchengrafik im Hintergrund berechnen.....	270
9	Grafikprogrammierung (GDI+)	277
9.1	Einführung	277
9.2	Dispose für Grafikobjekte	282
9.3	Linien, Rechtecke, Vielecke, Ellipsen, Kurven (Graphics-Klasse)	284
9.4	Farben (Color-Struktur).....	296
9.5	Linienformen (Pen-Klasse).....	298
9.6	Füllmuster (Brush-Klassen)	302
9.7	Koordinatensysteme und -transformationen	306
9.8	Text ausgeben (Font-Klassen).....	310
9.9	Schriftgröße einstellen	317
9.10	Schriftattribute, Textformatierung	326
9.11	Bitmaps nutzen.....	332
9.12	Bitmaps manipulieren	338
9.13	Icons	345
9.14	Metafile-Dateien	346
10	Grafik-Programmiertechniken	349
10.1	Zeichen- und Textqualität	349
10.2	Rotierter Text	352
10.3	Grafikobjekte zusammensetzen (GraphicsPath)	356
10.4	Durchsichtige Bitmaps.....	357
10.5	Umgang mit Regionen und Clipping	361

10.6	Internationale Paint- und Resize-Ereignisse.....	364
10.7	Rechteck-Auswahl mit der Maus (Rubberbox).....	370
10.8	Bitmap-Grafik zwischenspeichern (AutoRedraw).....	374
10.9	Flimmerfreie Grafik (Double-Buffer-Technik).....	377
10.10	Scrollbare Grafik.....	380
10.11	Einfache Animationseffekte.....	381
10.12	Bitmap über Byte-Feld adressieren.....	384
11	Drucken.....	389
11.1	Überblick.....	390
11.2	PrintDocument-Komponente.....	391
11.3	PrintDialog- und PageSetupDialog-Steuerelement.....	395
11.4	PrintPreviewDialog-Steuerelement.....	399
11.5	Druckereigenschaften und Seitenlayout.....	401
11.6	Beispiel – Mehrseitiger Druck.....	406
11.7	Beispiel – Inhalt eines Textfelds ausdrucken.....	410
11.8	Drucken ohne Steuerelemente.....	419
11.9	Bitmap mit Fensterinhalt erzeugen und ausdrucken.....	428
A	Von VB6 zu VB2008.....	431
	Neu in VB2008 im Vergleich zu VB6.....	432
	Der Migrationsassistent.....	446



Vorwort

Die Botschaft von Microsoft ist überdeutlich: Windows Forms ist alte, »tote« Technologie, die man wohl oder übel noch eine Weile warten muss, aber nicht mehr weiterentwickelt. Die Zukunft heißt Windows Presentation Foundation (WPF). Microsoft möchte ganz offensichtlich, dass alle Entwickler möglichst rasch auf den WPF-Zug aufspringen.

Ganz so schnell wird es aber voraussichtlich nicht gehen: Es gibt Tausende von Windows-Forms-Programmen, die noch über viele Jahre weiter gepflegt werden müssen. Windows Forms sieht im Vergleich zur WPF vielleicht »alt« aus, ist dafür aber ausgereift und stabil. Bei der Programmentwicklung unter Visual Studio hilft ein Designer, der wesentlich besser und effizienter funktioniert als sein WPF-Gegenstück, das mehr eine Design-Studie als ein Designer ist.

Das Angebot an WPF-Steuerelementen ist in vielen Bereichen noch sehr bescheiden und kann mit Windows Forms nicht einmal ansatzweise mithalten. Für Datenbank Anwendungen ist die WPF noch weitgehend ungeeignet: Es fehlen nicht nur wichtige Datenbanksteuer-elemente, sondern auch elementare Entwicklungswerkzeuge (die Visual Data Tools in VS2008 sind nur für Windows Forms konzipiert) sowie eine bessere Unterstützung von der WPF-*data-binding* für ADO.NET.

Kurz und gut: So tot, wie Microsoft Windows Forms gerne erklären möchte, ist diese Technologie noch lange nicht. Wer Wert auf Kompatibilität zu Windows-Versionen vor XP legt, wer momentan keine Zeit hat, um sich in eine neue Technologie einzuarbeiten, oder wer ganz einfach einen großen Bogen um Version-1.0-Software macht (und nichts anderes ist die WPF), wird selbst heute noch Neuentwicklungen mit Windows Forms starten.

Für mein VB2008-Buch musste ich eine Entscheidung treffen: Ich konnte entweder wie in den zwei Voraufgaben nochmals Windows Forms beschreiben oder auf die WPF setzen. Beide Techniken ausführlich in gedruckter Form zu beschreiben war leider unmöglich: Entweder hätte das Buch im Dünndruck erscheinen müssen (schlechte Lesequalität) oder als zweibändiges Werk (teuer). Da ich der Meinung bin, dass es zu Windows Forms bereits genug Dokumentation gibt und dass das Interesse meiner Leser eher in Richtung von zukunftsweisenden, neuen Technologien geht, habe ich mich für die WPF entschieden.

Gleichzeitig ist mir aber klar, dass nicht jeder sofort auf die WPF umsteigen wird. Selbst Entwickler, die sich mit der WPF bereits anfreunden, werden wahrscheinlich noch über viele Jahre weiter Windows-Forms-Code warten, adaptieren und erweitern müssen. Kurzum, dieses E-Book ist für alle gedacht, die weiterhin mit Windows Forms arbeiten. Den Text habe ich weitgehend aus der vorigen Auflage meines VB-Buchs übernommen. Lediglich die Strukturierung hat sich ein wenig geändert: Damit kann dieses »Buch im Buch« als ein eigenständiges, klar gegliedertes Werk benutzt werden. Selbstverständlich habe ich sämtliche Programme mit VB2008 unter Windows Vista getestet und bei dieser Gelegenheit neue Screenshots erstellt. Alle Beispielpprogramme befinden sich auf der CD zum Buch.

Michael Kofler, April 2008

<http://www.kofler.cc>

* * * TIPP

Einige Windows-Forms-Themen habe ich im gedruckten Teil des Buchs belassen:

- ▶ ***Multithreading-Grundlagen:** Die Grundlagen gelten sowohl für die WPF als auch für Windows Forms und sind in Abschnitt 21.4 im gedruckten Teil des Buchs zusammengefasst. Windows-Forms-spezifische Informationen und Beispiele finden Sie hier im E-Book in Kapitel 8.*
- ▶ ***Weitergabe von Windows-Programmen:** Aus Entwicklersicht ist es nahezu gleichgültig, ob Sie WPF- oder Windows-Forms-Programme weitergeben möchten. Der einzige Unterschied ist die .NET-Version: Für Windows-Forms-Programme reicht .NET 2.0, WPF-Programme erfordern zumindest .NET 3.0. Tipps zu Setup-Projekten und zur Click-Once-Weitergabe finden Sie in Kapitel 22 des gedruckten Teil des Buchs.*
- ▶ ***Datenbankprogrammierung:** Die WPF ist aus meiner Sicht noch kein geeignetes Fundament für ADO.NET-Datenbankanwendungen. Deswegen basieren auch in der Neuauflage die meisten ADO.NET-Beispielprogramme zur auf Windows Forms. Die entsprechenden Kapitel – speziell 28, 29 und 31 – befinden sich wegen Ihrer anhaltenden Wichtigkeit im gedruckten Teil des Buchs.*

! ! ! ACHTUNG

Die Weitergabe dieses E-Books – egal, in welcher Form – ist nicht zulässig!