Kapitel 13

# Windows Forms-Anwendungen erstellen

## In diesem Kapitel lernen Sie

- wie Sie eine Windows Forms-Anwendung erstellen
- wie Sie Steuerelemente in ein Form einfügen
- was Ereignisse sind und wie Sie auf Benutzereingaben reagieren, die ein Steuerelement erhält
- wie Sie Benutzereinstellungen in einer Konfigurationsdatei abspeichern und diese wieder einlesen
- wie Sie Daten von Ihrer Anwendung aus in die Zwischenablage kopieren

## 13.1 Projekt für Windows Forms-Anwendungen erstellen

In den meisten Beispielen dieses Buches haben wir die grundlegenden Sprachkonstrukte von Visual Basic 2010 anhand von Konsolenanwendungen vorgestellt, damit Sie sich nicht gleichzeitig mit den Fragen der Programmierung von Windows Forms-Anwendungen, die eine grafische Benutzeroberfläche besitzen, auseinandersetzen müssen. Konsolenanwendungen sind sehr nützlich, wenn Sie Programme erstellen, die auch ohne Benutzereingaben ihre Aufgaben erledigen können. Die meisten Windows Forms-Anwendungen leben jedoch durch eine grafische Oberfläche, über die der Benutzer mit Maus, Tastatur und anderen Eingabegeräten Befehle an das Programm senden kann und auf die das Programm dann reagiert.

In diesem Kapitel werden Sie eine Windows Forms-Anwendung erstellen, die folgende Features enthält:

- In dem Fenster der Anwendung können Sie mit drei Schiebereglern jeweils den Rot-, Grün- und Blauanteil einer Farbe festlegen
- Die jeweils ausgewählte Farbkombination wird in einem kleinen Kasten angezeigt
- Der Hexadezimalwert dieser Farbkombination kann durch das Anklicken einer Schaltfläche in die Windows-Zwischenablage kopiert werden
- Das Programm merkt sich außerdem die zuletzt eingestellte Farbe in einer Konfigurationsdatei; dieser Wert wird beim Programmstart eingelesen und in dem Fenster der Anwendung angezeigt

Wenn Sie dieses Kapitel durchgearbeitet haben, sieht die Anwendung so aus, wie es Abbildung 13.1 zeigt.


**Abbildung 13.1:** Das Fenster der Anwendung, die Sie in diesem Kapitel erstellen werden

1. Starten Sie Visual Basic 2010 Express, wenn es noch nicht gestartet ist. Wählen Sie den Menübefehl *Datei/Neues Projekt,* damit das Dialogfeld *Neues Projekt* angezeigt wird.
2. Markieren Sie in der Vorlagenliste den Eintrag *Windows Forms-Anwendung.* Geben Sie in das Textfeld *Name* **Farbregler** ein und klicken Sie auf *OK.*

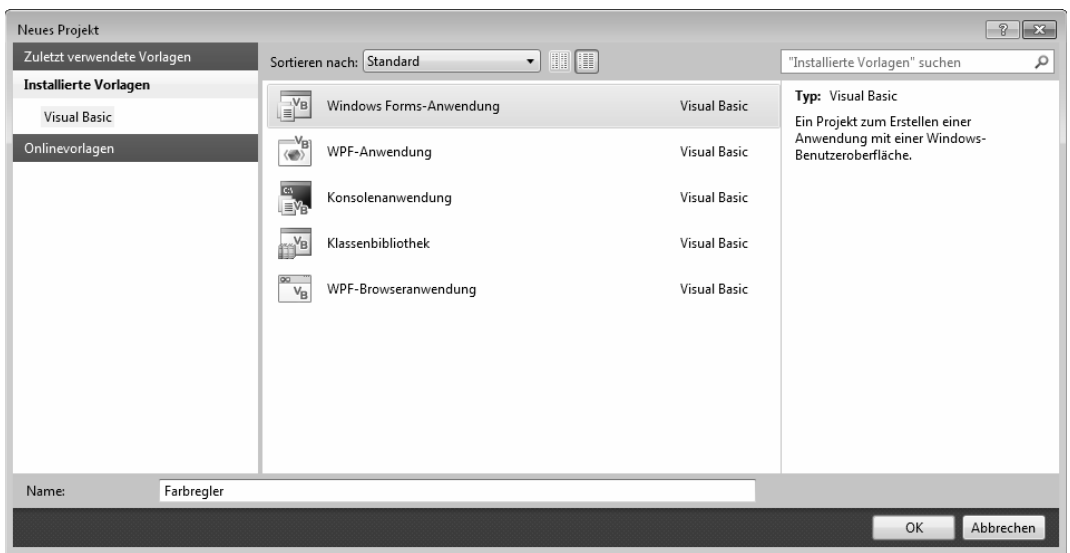



**Abbildung 13.2:** Wenn Sie eine Anwendung erstellen wollen, die in einem Fenster dargestellt wird, wählen Sie im Dialogfeld *Neues Projekt* die Vorlage *Windows Forms-Anwendung* aus

Nach dem Klick auf *OK* erstellt Visual Basic 2010 Express das Grundgerüst Ihrer Anwendung. Im Unterschied zu den Konsolenanwendungen sehen Sie jedoch zuerst keinen Programmcode. Stattdessen wird das Form der Anwendung im sogenannten Formular-Designer angezeigt, wie es Abbildung 13.3 zeigt. Die Form hat den Namen *Form1.*

3. Richten Sie das Fenster von Visual Basic 2010 Express so ein, wie es Abbildung 13.3 zeigt:

- Damit die Toolbox an der linken Seite des Fensters eingeblendet wird, wählen Sie den Befehl *Ansicht/Toolbox*. Über den kleinen Pfeil der Titelleiste der Toolbox können Sie ein Kontextmenü öffnen und dort festlegen, dass die Toolbox automatisch in den Hintergrund gebracht wird, wenn Sie sie nicht benötigen. Die Toolbox wird dann als Registerkarte am linken Fensterrand angezeigt. Um die Toolbox dann wieder zu öffnen, klicken Sie die Registerkarte an.
- Wählen Sie *Ansicht/Projektmappen-Explorer*, falls die Projektmappe und deren Inhalt derzeit nicht sichtbar sind.
- Unterhalb des Projektmappen-Explorers sehen Sie in Abbildung 13.3 das Eigenschaftenfenster. Falls das Eigenschaftenfenster nicht sichtbar ist, wählen Sie den Befehl *Ansicht/Eigenschaftenfenster* aus. Im Eigenschaftenfenster können Sie die Eigenschaften des derzeit im Formular-Designer ausgewählten Elements ansehen und verändern.

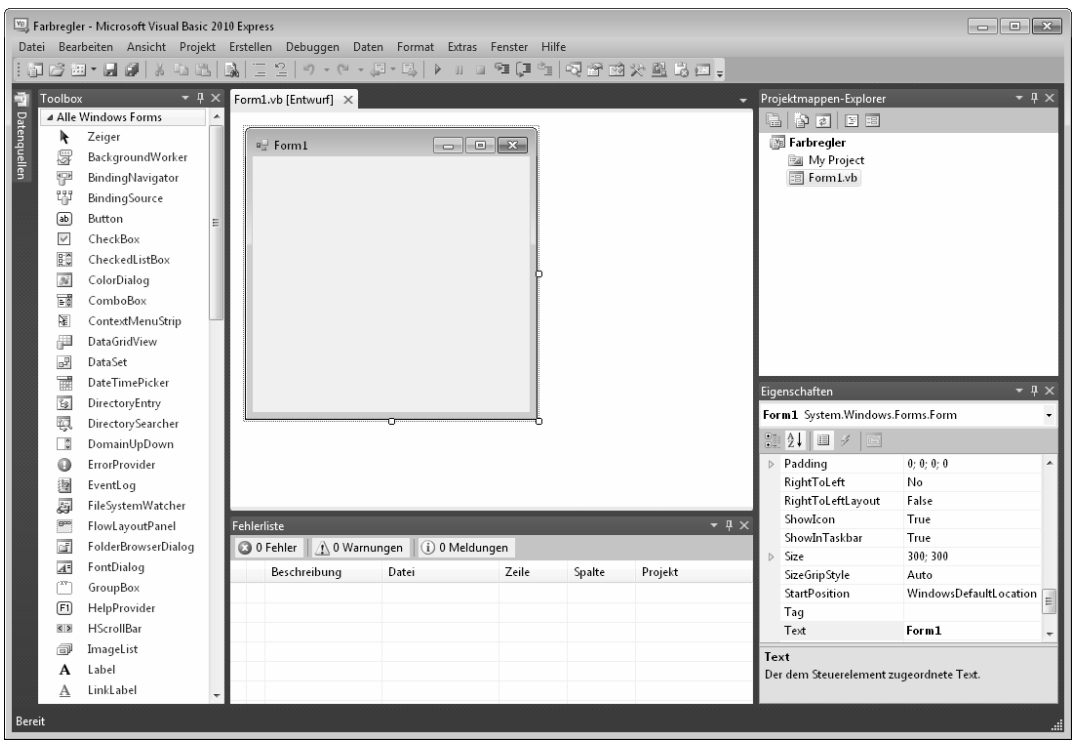



**Abbildung 13.3:** Im Fenster von Visual Basic 2010 Express sind die Toolbox, das noch leere Formular der Anwendung und der Projektmappen-Explorer sowie das Eigenschaftenfenster sichtbar

4. Das erste Form einer neuen Windows Forms-Anwendung, die Sie mit der Vorlage erstellen, besitzt immer den Namen `Form1`. Dieser Name ist nicht sehr aussagekräftig. Besser ist es, vor allem, wenn Sie eine Anwendung mit mehreren Forms erstellen (z.B. wenn Sie unterschiedliche Fenster und/oder Dialogfelder verwenden), die Forms mit einem eindeutigen Namen zu versehen. Klicken Sie im Projektmappen-Explorer den Eintrag *Form1.vb* mit der rechten Maustaste an, wählen Sie *Umbenennen* und geben Sie als Namen **ColorForm.vb** ein. Drücken Sie die Eingabetaste.

5. Klicken Sie das Form *ColorForm* im Formular-Designer an. Im Eigenschaftenfenster werden dann alle Eigenschaften dieses Forms angezeigt. Suchen Sie nach der Eigenschaft *Name*. Dort steht als Name des Forms bereits *ColorForm*. Die integrierte Entwicklungsumgebung hat den Namen des Forms automatisch angepasst, als Sie den Dateinamen geändert haben. Die Eigenschaft *Name* eines Forms oder eines Steuerelements ist wichtig, da Sie über den Namen von Ihrem Code aus auf das Form bzw. Steuerelement zugreifen. Mehr dazu später.
6. Suchen Sie im Eigenschaftenfenster nach der Eigenschaft *Text*. Die Eigenschaft *Text* eines Forms ist der Text, der in der Titelleiste des Fensters angezeigt wird. Klicken Sie in die Spalte neben *Text* und legen Sie als Eigenschaft **Farbauswahl** fest. Die Änderung ist im Form sofort sichtbar.
7. Suchen Sie im Eigenschaftenfenster nach der Eigenschaft *MaximizeBox*. Diese Eigenschaft legt fest, ob in der Titelleiste die *Maximieren*-Schaltfläche angezeigt werden soll. Ändern Sie diesen Eigenschaftswert auf *False,* da dieses Programm keine *Maximieren*-Schaltfläche benötigt. Im Formular-Designer wird diese Schaltfläche nun deaktiviert dargestellt. Während der Programmausführung selbst wird sie nicht mehr sichtbar sein.
8. Suchen Sie nach der Eigenschaft *StartPosition*. Ändern Sie diesen Eigenschaftswert auf *CenterScreen*. Hierdurch erreichen Sie, dass das Fenster der Anwendung nach dem Starten mittig auf dem Bildschirm positioniert wird.

Hiermit sind die Änderungen an den Eigenschaften des Forms abgeschlossen. Im weiteren Verlauf dieses Buches werden wir Änderungen, die wir an den Eigenschaften eines Forms oder eines Steuerelements vornehmen müssen, in Tabellenform darstellen. Gehen Sie dann wie oben beschrieben vor, um die Eigenschaft zu ändern, also:

1. Gewünschtes Element im Formular-Designer anklicken.
2. Im Eigenschaftenfenster nach der Eigenschaft suchen und
3. den Wert der Eigenschaft ändern.

**Tipp: Sortierreihenfolge der Eigenschaften ändern**

Mit den ersten beiden Schaltflächen oben am Eigenschaftenfenster können Sie die Sortierreihenfolge der Eigenschaften ändern. Die erste Schaltfläche zeigt die Eigenschaften nach Kategorien an, die zweite alphabetisch. Verwenden Sie die alphabetische Sortierung, um schnell die Eigenschaft zu finden, die Sie ändern möchten.

# 13.2 Steuerelemente einfügen

In diesem Abschnitt werden Sie die Steuerelemente einfügen, mit denen der Benutzer die Farbe einstellen kann, mit denen der ausgewählte Farbwert angezeigt und dessen Hex-Wert ausgegeben wird, sowie die Schaltfläche einfügen, mit der der Farbwert in die Zwischenablage eingefügt werden kann.

1. Öffnen Sie die Toolbox, falls diese nicht sichtbar ist.
2. Klicken Sie, falls erforderlich, auf das Dreieck vor der Gruppe *Alle Windows Forms*, um alle Einträge dieser Gruppe anzeigen zu lassen.

### Tipp: Toolbox automatisch im Hintergrund

In der Titelleiste der Toolbox finden Sie die Schaltfläche *Automatisch im Hintergrund*. Wenn diese Schaltfläche aktiv ist (der Pin zeigt nach links, wie Sie es in der nachfolgenden Abbildung sehen), verschwindet die Toolbox automatisch wieder vom Bildschirm, nachdem Sie ein Steuerelement in ein Form eingefügt haben.


Sie müssen dann mit der Maus auf die Registerkarte *Toolbox* zeigen, um die Toolbox wieder zu öffnen. Wenn die Option ausgeschaltet ist (der Pin zeigt dann nach unten), ist die Toolbox dauerhaft sichtbar.

3. Klicken Sie auf den Eintrag *GroupBox*, ziehen Sie eine *GroupBox* auf das Form und legen Sie sie dort ab. Eine GroupBox dient dazu, mehrere Steuerelemente mit einem Rahmen zu gruppieren. Optional kann ein Text festgelegt werden, der auf dem Rahmen angezeigt werden soll.
4. Achten Sie darauf, dass das Steuerelement *GroupBox* markiert ist, und ändern Sie im Eigenschaftenfenster folgende Eigenschaften auf die in der Tabelle angegebenen Werte:

| Eigenschaft | Wert |
|---|---|
| Location | x=15, y=12 |
| Size | Width = 260, Height = 152 |
| Text | Farbregler |

Manche Eigenschaften, wie hier *Location* und *Size*, bestehen aus mehreren Werten. Sie können diese entweder direkt hinter dem Namen der Eigenschaft durch ein Semikolon getrennt eingeben oder Sie klicken auf das Dreieck vor dem Eigenschaftennamen, um den Knoten zu öffnen und geben dann die Werte getrennt ein.

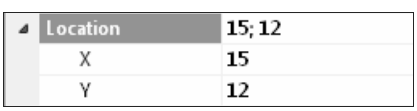



**Abbildung 13.4:** Sie können die einzelnen Werte einer Eigenschaft, die aus mehreren Werten besteht, eingeben, indem Sie auf das Dreieck vor dem Namen der Eigenschaft klicken

### Hinweis: Koordinatenangaben

Die Koordinatenangaben, die Sie bei der Eigenschaft *Location* angeben, beziehen sich relativ auf die linke obere Ecke des Containers, in dem sich das Steuerelement befindet. Der Container der GroupBox ist das Form selbst. Bei den Label-Steuerelementen, die Sie im nächsten Schritt einfügen, ist der Container die GroupBox. Daher beziehen sich die Koordinatenangaben dort auf deren obere linke Ecke.

5. Klicken Sie das Form an, um es zu markieren, damit die nachfolgend eingefügten Steuerelemente das Form selbst als Container verwenden und damit sich die Positionsangaben, die Sie weiter hinten festlegen, auf das übergeordnete Steuerelement beziehen.

6. Doppelklicken Sie dreimal auf den Eintrag *Label* in der Gruppe *Allgemeine Steuerelemente* in der Toolbox. Wenn Sie auf einen Eintrag in der Toolbox doppelklicken, wird dieses Steuerelement an der nächsten Standardposition dem Form eingefügt. Im Form sind nun drei Label-Steuerelemente mit den Beschriftungen *Label1*, *Label2* und *Label3* sichtbar.
7. Label-Steuerelemente dienen dazu, Text wie beispielsweise Beschriftungen auszugeben. Ändern Sie die Eigenschaften dieser Steuerelemente so ab, wie es die folgende Tabelle zeigt:

| Aktueller Name auf dem Form | Eigenschaft | Wert |
|---|---|---|
| Label1 | (Name) | lblRed |
| | Location | x= 12, y= 16 |
| | Text | Rot: |
| Label2 | (Name) | lblGreen |
| | Location | x= 12, y= 61 |
| | Text | Grün: |
| Label3 | (Name) | lblBlue |
| | Location | x= 12, y= 106 |
| | Text | Blau: |

**Tipp: Steuerelemente mithilfe der Ausrichtungslinien positionieren**

Wenn Sie eigene Forms erstellen und auf den Forms Steuerelemente ablegen, werden Sie selten, wie hier gezeigt, die Positionsangaben eintippen, sondern sie mit der Maus verschieben. Um die Steuerelemente mit der Maus einfach rechts- oder linksbündig positionieren zu können und um gleichmäßige Abstände zu erhalten, können Sie die Ausrichtungslinien des Formular-Designers verwenden. Wenn Sie ein Steuerelement mit der Maus ziehen, sehen Sie links oder rechts neben dem Steuerelement blaue vertikale Linien, mit denen Sie mehrere Steuerelemente links- bzw. rechtsbündig ausrichten können.


Wenn Sie vor einem Steuerelement einen kleinen horizontalen Strich sehen, zeigt dieser den Mindestabstand zwischen zwei Steuerelementen oder zwischen einem Steuerelement und seinem Container an.

8. Öffnen Sie in der Toolbox die Gruppe *Alle Windows Forms* und klicken Sie den Eintrag *TrackBar* an.
9. Ziehen Sie auf dem Form rechts neben dem Label *Rot* ein Rechteck auf, das ungefähr die Größe hat wie der oberste Schieberegler in Abbildung 13.1. Setzen Sie die Eigenschaften dieses Steuerelements auf die Werte, die Sie in der folgenden Tabelle finden:

| Aktueller Name auf dem Form | Eigenschaft | Wert |
|---|---|---|
| TrackBar1 | (Name) | trkRed |
| | Location | x=45, y=16 |
| | Size | Width=175; Height=45 |
| | Minimum | 0 |
| | Maximum | 255 |
| | TickFrequency | 20 |

Eine RGB-Farbe setzt sich aus den drei Farben Rot, Grün und Blau zusammen. Jede dieser drei Farben kann einen Wert im Bereich zwischen 0 und 255 besitzen. Damit der Anwender mit dem Schieberegler nur einen Wert in diesem Bereich einstellen kann, wurde die Eigenschaft *Minimum* auf 0 und die Eigenschaft *Maximum* auf 255 gesetzt. Mit der Eigenschaft *TickFrequency* wird die Anzahl der Positionen zwischen den Teilstrichen festgelegt, die im unteren Bereich des Steuerelements sichtbar sind.

10. Das Steuerelement `trkRed` besitzt nun bis auf den Namen und die Position schon alle Eigenschaftswerte, die auch für den Schieberegler für Grün und für den für Blau benötigt werden. Um nicht alle Eigenschaften bei den noch fehlenden Steuerelementen neu setzen zu müssen, ist es am einfachsten, das Steuerelement `trkRed` zu kopieren. Klicken Sie das Steuerelement `trkRed` an, um es auszuwählen. Drücken Sie die Taste Strg und ziehen Sie mit gedrückter Maustaste eine Kopie dieses Steuerelements neben den Text *Grün*. Wiederholen Sie dies beim Schieberegler für den Blauanteil.

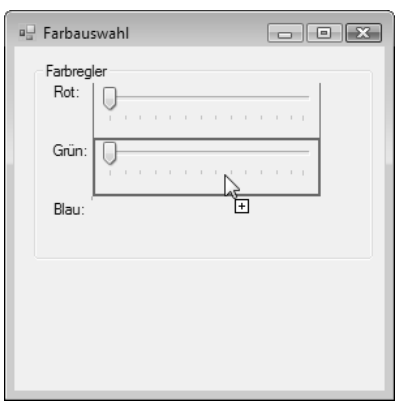



**Abbildung 13.5:** Bei gedrückter Strg-Taste können Sie mit Ziehen & Ablegen eine Kopie eines Steuerelements erstellen

11. Ändern Sie nun die Namen der kopierten Steuerelemente und prüfen Sie, ob sie sich an der richtigen Position befinden.

| Aktueller Name auf dem Form | Eigenschaft | Wert |
|---|---|---|
| TrackBar1 | (Name) | trkGreen |
| | Location | x= 45, y=61 |
| TrackBar2 | (Name) | trkBlue |
| | Location | x= 45, y=106 |

12. Fügen Sie nun die noch fehlenden Steuerelemente in das Form ein und gehen Sie so vor, wie Sie es in den bisherigen Schritten gelernt haben. Die nachfolgende Tabelle enthält alle Informationen, die Sie benötigen.

| Steuerelementtyp | Eigenschaft | Wert |
|---|---|---|
| Panel | (Name) | pnlColorSample |
| | BorderStyle | FixedSingle |
| | Location | x=15, y=171 |
| | Size | Width=260, Height=50 |
| Label | Location | x=15, y=231 |
| | Text | Hexadezimalwert: |
| TextBox | (Name) | txtHexValue |
| | ReadOnly | True |
| | Location | x=115, y=227 |
| | Size | Width=67, Height=20 |
| Button | (Name) | btnCopy |
| | Location | x=200, y=226 |
| | Size | Width=75, Height=23 |
| | Text | Kopieren |

Damit ist die Benutzeroberfläche der Anwendung fertiggestellt. Die Optik steht, aber das Programm besitzt noch keine wirkliche Funktionalität. Diese werden Sie in den folgenden Abschnitten ergänzen.

# 13.3 Anwendungseinstellungen verwenden

Die Liste der in Abschnitt 13.1 aufgeführten Features sagt, dass die zuletzt eingestellte Farbkombination beim Programmende gespeichert und beim Programmstart wieder eingelesen werden soll. Für dieses Feature, das die Anwendung benutzerfreundlicher macht, brauchen Sie keine einzige Zeile Code zu schreiben. Zuerst müssen Sie festlegen, welche Einstellungen Sie in Ihrer Anwendung verwenden wollen, und anschließend diese Einstellungen mit den Steuerelementen verknüpfen, die sie verwenden.

## Anwendungseinstellungen definieren

In diesem Abschnitt werden Sie die Namen festlegen, unter denen die Einstellungen gespeichert werden sollen, und die Standardwerte der Einstellungen definieren.

1. Klicken Sie im Projektmappen-Explorer das Projekt *Farbregler* mit der rechten Maustaste an und wählen Sie *Eigenschaften*. Sie können alternativ im Menü *Projekt* den Befehl *Farbregler-Eigenschaften* wählen. Im Arbeitsbereich von Visual Basic 2010 Express wird ein Fenster mit mehreren Registerkarten, die sich an der linken Seite befinden, angezeigt. Dieses Fenster wird Projekt-Designer genannt.

2. Klicken Sie auf die Registerkarte *Einstellungen.*

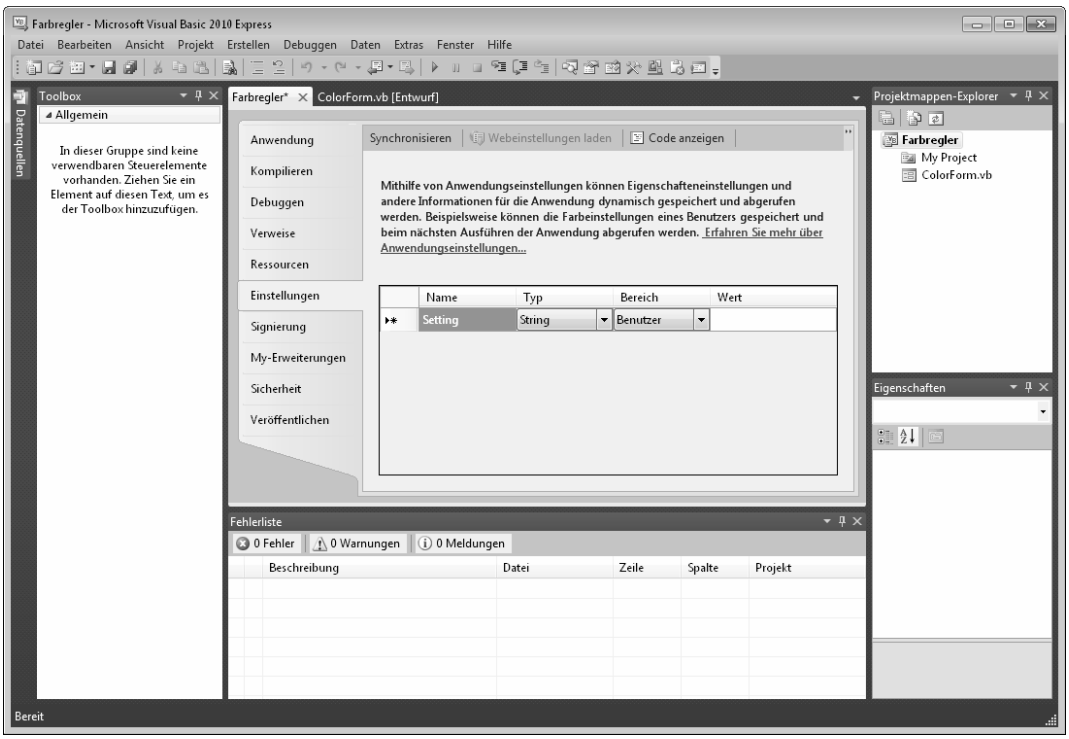



**Abbildung 13.6:** Im Projekt-Designer können Sie u.a. die Anwendungseinstellungen für Ihr Projekt definieren

Die Seite mit den Anwendungseinstellungen wird geöffnet. Hier können Sie die Namen, den Datentyp und den anfänglichen Wert der Einstellungen festlegen, die in einer Konfigurationsdatei gespeichert werden sollen.

**Tipp: Wenn viele Quellcode- und Entwurfsfenster geöffnet sind**

Am rechten Rand der Leiste mit den Registerkarten der geöffneten Quellcodedateien sehen Sie eine kleine Schaltfläche mit einem nach unten zeigenden Pfeil. Wenn Sie den Pfeil anklicken, werden in einer Liste alle aktiven Dateien angezeigt, und zwar auch diejenigen, für die in der Leiste aus Platzgründen keine Registerkarte sichtbar ist. Klicken Sie in der Liste den Namen der Datei an, die im Bearbeitungsfenster angezeigt werden soll.

3. Klicken Sie in das Feld unterhalb von *Name,* geben Sie dort **red** ein, wählen Sie in der Liste *Typ* den Eintrag *Integer* aus und geben Sie in das Feld *Wert* **128** ein.
4. Drücken Sie die Tabulatortaste, um zur zweiten Zeile zu gelangen.
5. Erstellen Sie einen Eintrag mit dem Namen **green**, ebenfalls vom Typ *Integer* und mit dem Wert **0**. Erstellen Sie einen dritten Eintrag mit dem Namen **blue** und legen Sie als Startwert **128** fest.
6. Klicken Sie in der Navigationsleiste oberhalb des Code-Editors die *Schließen*-Schaltfläche an, um den Projekt-Designer zu schließen. Sie sehen das folgende Dialogfeld:

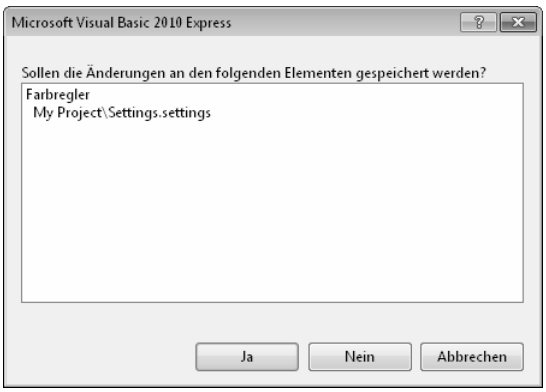



**Abbildung 13.7:** Hier werden Sie gefragt, ob die Änderungen an der Projektmappendatei gespeichert werden sollen

7. Klicken Sie auf *Ja.* (Im Kasten auf Seite 292 finden Sie Informationen darüber, welche Änderungen an der Projektmappe durch die Definition der Einstellungen vorgenommen wurden.)
8. Starten Sie das Projekt erneut und prüfen Sie, ob nach dem Programmstart noch alle drei Schieberegler auf der mittleren Position stehen. Dies muss so sein, da das Programm noch keinen Code enthält, mit dem die Anwendungseinstellungen eingelesen werden.

## Die TrackBars mit den Anwendungseinstellungen verknüpfen

Lassen Sie uns nun die Aktionen durchführen, die nötig sind, um drei Schieberegler mit den im vorigen Abschnitt erstellten Anwendungseinstellungen zu verknüpfen.

1. Öffnen Sie das Form *ColorForm* im Formular-Designer.
2. Klicken Sie das Steuerelement `trkRed` an und wechseln Sie zum Eigenschaftenfenster.
3. Klicken Sie auf das Dreieck vor *(ApplicationSettings).*
4. Klicken Sie die Schaltfläche mit den drei Auslassungzseichen an, die rechts neben *(PropertyBinding)* sichtbar wird. Das Dialogfeld *Anwendungseinstellungen für "trkRed"* wird geöffnet.

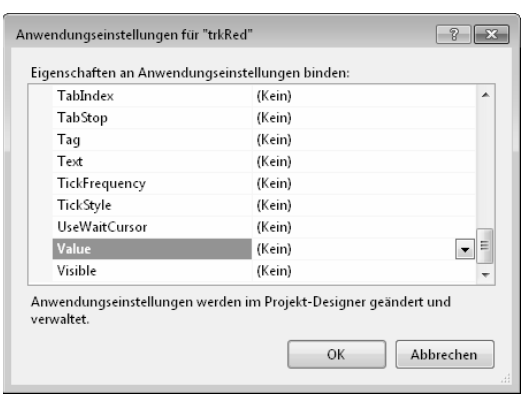



**Abbildung 13.8:** Auf diesem Dialogfeld können Sie eine Anwendungseinstellung mit einer Eigenschaft eines Steuerelements verknüpfen

5. Klicken Sie auf den Pfeil neben *Value,* damit sich das Auswahlfenster öffnet, das Sie in Abbildung 13.9 sehen und in dem alle definierten Anwendungseinstellungen angezeigt werden.

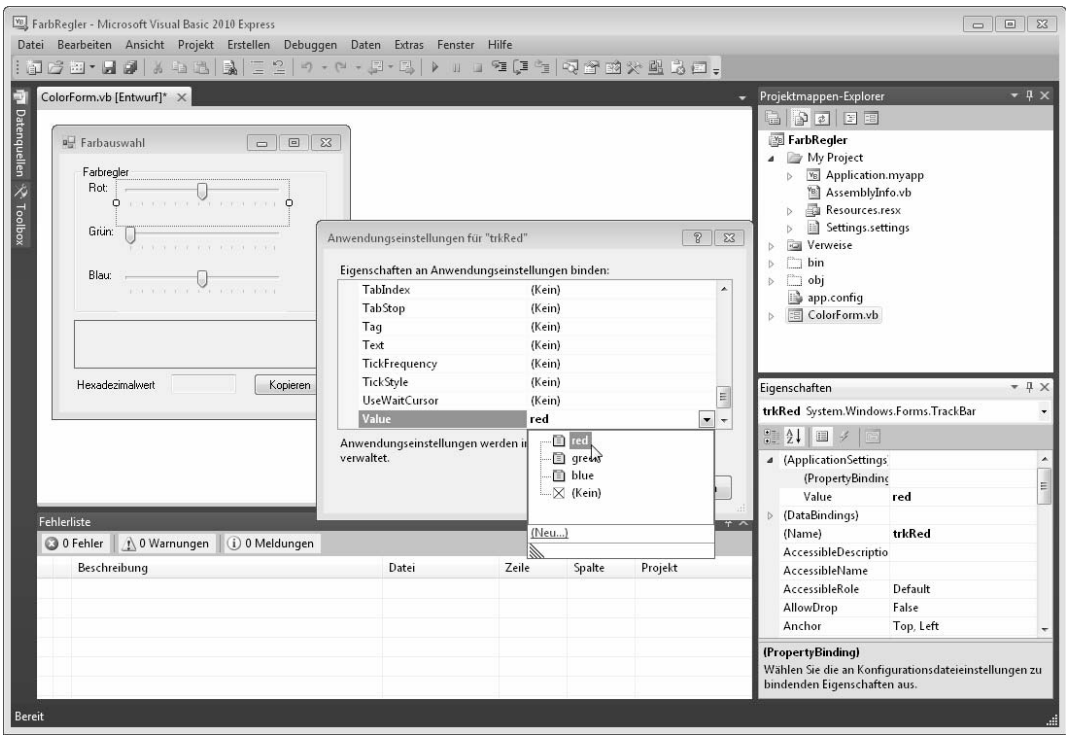



**Abbildung 13.9:** Über das Eigenschaftenfenster können Sie eine Anwendungseinstellung mit allen Eigenschaften des im Formular-Designer markierten Steuerelements verknüpfen

6. Klicken Sie auf *red.* Beachten Sie, dass der Schieberegler für *Rot* im Formular-Designer nun an der Position steht, die Sie in den Anwendungseinstellungen definiert haben, nämlich 128.
7. Wiederholen Sie die Schritte 2 bis 6 für die Steuerelemente `trkGreen` und `trkBlue` und verknüpfen Sie die Eigenschaft *Value* mit den Anwendungseinstellungen *green* bzw. *blue.*
8. Wechseln Sie zum Projektmappen-Explorer, klicken Sie das Projekt *Farbregler* mit der rechten Maustaste an und wählen Sie *Eigenschaften.* Wechseln Sie zur Registerkarte *Anwendung* und prüfen Sie, ob das Kontrollkästchen *My.Settings beim Herunterfahren speichern* eingeschaltet ist. Ist dies der Fall, werden die drei Einstellungen automatisch beim Programmende gespeichert.

### Hinweis: Die Anwendungseinstellungen im Code verwenden

Sie können über den Namespace `My` auch von Ihrem Code aus auf die Anwendungseinstellungen zugreifen. Verwenden Sie hierfür die Syntax `My.Settings.`*`NameDerEinstellung`*. Um die Einstellungen für den Rotanteil zu verwenden, würden Sie den Ausdruck `My.Settings.Red` benutzen. Wenn Sie die Einstellungen selbst verwalten, müssen Sie diese beim Programmende mit `My.Settings.Save` in der Einstellungsdatei sichern.

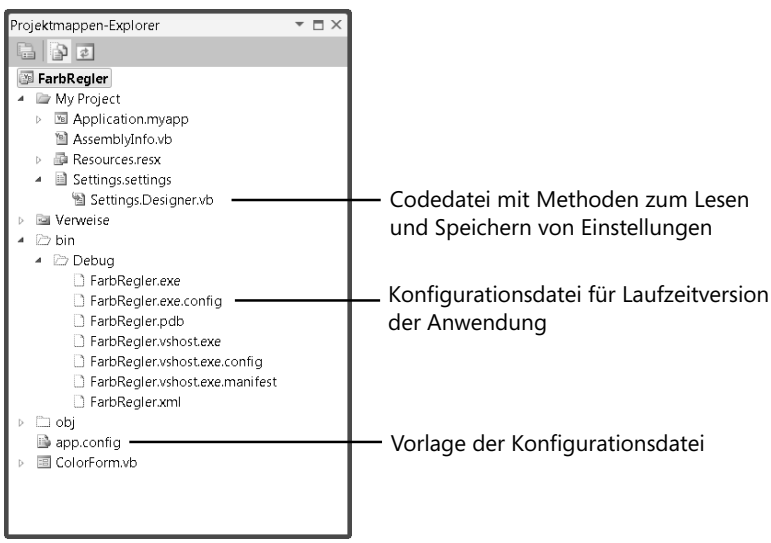
Weitere Informationen über den Namespace `My` finden Sie in Kapitel 16.

**Background: Neue Dateien im Projekt nach der Definition von Anwendungseinstellungen**

Lassen Sie uns einen Blick in den Projektmappen-Explorer werfen, um zu sehen, welche Änderungen Visual Basic 2010 Express am Projekt vorgenommen hat, nachdem Sie die Anwendungseinstellungen definiert haben. Um die folgenden Erklärungen an Ihrem Projekt nachvollziehen zu können, klicken Sie im Projektmappen-Explorer die Schaltfläche *Alle Dateien anzeigen* an.

Das Projekt enthält nun die neuen Dateien *Settings.settings* und *Settings.Designer.vb*. Diese Dateien enthalten die Anwendungseinstellungen sowie eine Klasse mit dem Namen *Settings*, die Methoden und Eigenschaften zur Verfügung stellt, über die Sie zur Laufzeit auf die Anwendungseinstellungen zugreifen können. Diese Methoden werden auch beim Programmende automatisch aufgerufen, wenn Sie im Projekt-Designer das Kontrollkästchen *Einstellungen beim Herunterfahren automatisch speichern* eingeschaltet haben. Dieser Quellcode wird von Visual Basic 2010 Express erstellt.

Im Hauptordner sehen Sie die Datei *app.config*. *app.config* ist eine Datei im XML-Format. Dort werden die Namen und die Standardwerte der Einstellungen gespeichert, wie Sie sie im Projekt-Designer festgelegt haben.




Zur Laufzeit des Programms wird in den Ordner, in dem sich auch die ausführbare Datei der Anwendung befindet, eine Kopie von *app.config* angelegt. Der Dateiname leitet sich vom Namen der Anwendung ab und wird um die Erweiterung *.config* ergänzt. In unserem Fall ist der Dateiname *Farbregler.exe.config*.

# 13.4 Auf Ereignisse reagieren

Wenn Sie eine Windows Forms-Anwendung erstellen, müssen Sie zwei verschiedene Teilaufgaben erledigen. Zum einen erstellen Sie die Benutzeroberfläche der Anwendung, das Form, mit den Steuerelementen, die Sie benötigen, damit der Anwender so wie gewollt mit der Anwendung interagieren kann. Diesen Schritt haben Sie im vorhergehenden Abschnitt erledigt.

Der zweite Schritt besteht darin, auf Ereignisse zu reagieren. Der Code einer Windows Forms-Anwendung wird nie nacheinander ausgeführt, da diese Art von Programmen ereignisgesteuert ist. Das bedeutet, der Ausführungsverlauf der Anwendung wird von externen Vorkommnissen bestimmt, die Ereignisse genannt werden. Ereignisse sind Signale, mit denen die Anwendung darüber informiert wird, dass etwas geschehen ist, worauf die Anwendung reagieren sollte.

Nehmen Sie als Beispiel das Projekt, das wir in diesem Kapitel erstellen. In der Beispielanwendung soll im Panel `pnlColorSample` immer die Farbe angezeigt werden, die in den Schiebereglern eingestellt ist. Ihr Code muss also informiert werden, wenn der Anwender die Position eines der Farbregler verändert. Diese Information kommt zuerst beim Steuerelement an. Das Steuerelement löst dann ein Ereignis aus. Sie können dieses Ereignis abonnieren und werden dann automatisch über das Ereignis informiert, indem in Ihrem Programmcode eine Prozedur aufgerufen wird, die Sie vorher erstellt haben. Solch eine Prozedur wird Ereignisbearbeitungsmethode, oder kurz Ereignishandler, genannt.

In diesem Abschnitt werden Sie zunächst sehen, was Sie tun müssen, um ein Ereignis zu abonnieren, und dann, wie Sie den Code des Ereignishandlers schreiben.

## Ereignishandler für Steuerelemente erstellen

In diesem Abschnitt werden Sie den Code erstellen, der auf das Verschieben der Farbregler reagiert.

1. Lassen Sie das Form *ColorForm* im Formular-Designer anzeigen.
2. Doppelklicken Sie auf den obersten Schieberegler. Der Code für das Form wird im Editor geöffnet. Durch den Doppelklick wurde automatisch der Rumpf eines Ereignishandlers erstellt. Dieser hat den Namen `trkRed_Scroll`; ist also aus dem Namen des Steuerelements (»trkRed«) und dem Namen des Ereignisses, hier »Scroll«, zusammengesetzt.

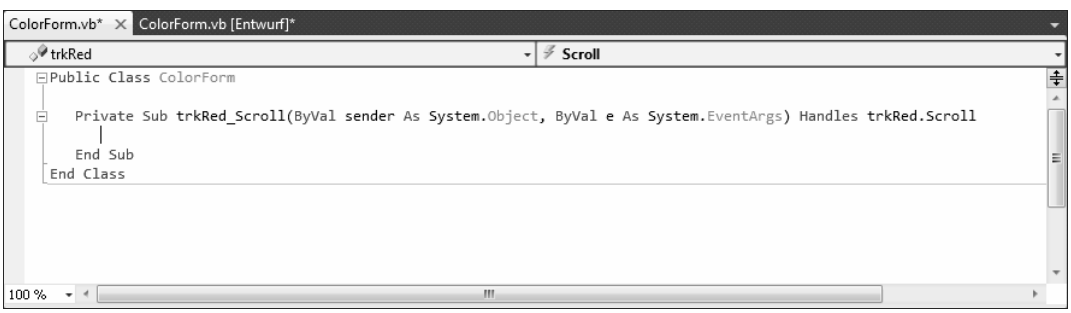



**Abbildung 13.10:** Wenn Sie auf ein Steuerelement doppelklicken, erstellt Visual Basic 2010 Express einen leeren Ereignishandler für das Standardereignis des Steuerelements. Bei einer TrackBar ist das Standardereignis Scrollen, bei einer Schaltfläche (was Sie ein wenig später sehen werden) ist es das Click-Ereignis.

Der Ereignishandler erhält zwei Parameter: Der erste hat den Namen `sender`, ist vom Datentyp `System.Object` und enthält das Objekt (in unserem Beispiel also die Instanz des Trackbar-Steuerelements), das das Ereignis ausgelöst hat. Der zweite Parameter lautet `e` und besitzt den Datentyp `System.EventArgs`; über diesen Parameter können an den Ereignishandler weitere Informationen über das Ereignis übermittelt werden. Beide Parameter werden im aktuellen Projekt nicht benötigt; wir kommen hierauf in Kapitel 14 zurück, in dem Sie lernen, wie Sie eigene Steuerelemente erstellen können.

Ganz am Ende der Zeile stehen die Anweisungen, die aus dieser Prozedur einen Ereignishandler machen. Das Schlüsselwort `Handles` legt fest, dass es sich bei dieser Sub-Prozedur um einen Ereignishandler handelt, und zwar für das `Scroll`-Ereignis des Steuerelements mit dem Namen `trkRed`. Der Name des Steuerelements und der Name des Ereignisses werden nach dem Schlüsselwort `Handles` angegeben und durch einen Punkt verbunden.

3. Fügen Sie in den Ereignishandler einen Aufruf der noch nicht vorhandenen Prozedur `showColor` ein. `showColor` wird sich darum kümmern, dass im `Panel`-Steuerelement die derzeit ausgewählte Farbe angezeigt wird.

```
Private Sub trkRed_Scroll(ByVal sender As System.Object, ByVal e As System.EventArgs) _
Handles trkRed.Scroll
   showColor()
End Sub
```

4. Kehren Sie zu `ColorForm` im Formular-Designer zurück und doppelklicken Sie auf den Schieberegler für den Grünwert.

5. Fügen Sie auch in den Ereignishandler für das Scroll-Ereignis des Schiebereglers für den Grünanteil den Aufruf von `showColor` ein:

```
Private Sub trkGreen_Scroll(ByVal sender As System.Object, ByVal e As System.EventArgs) _
Handles trkGreen.Scroll
   showColor()
End Sub
```

6. Wiederholen Sie die Schritte 4 und 5 für das Scroll-Ereignis des Schiebereglers für den Blauanteil:

```
Private Sub trkBlue_Scroll(ByVal sender As System.Object, ByVal e As System.EventArgs) _
Handles trkBlue.Scroll
   showColor()
End Sub
```

7. Jetzt fehlt uns noch die Prozedur `showColor`. Das *Panel*-Steuerelement (wie fast alle sichtbaren Steuerelemente auch) besitzt die Eigenschaft *BackColor*, die wir einfach nur zuweisen müssen, um die Hintergrundfarbe zu ändern. Die Eigenschaft *BackColor* besitzt den Datentyp *Color*. Wir müssen also zuerst eine Instanz des Datentyps *Color* erstellen und dann dieser Instanz die RGB-Werte zuweisen. Die Variable des Datentyps *Color*, die wir dann erhalten, können wir im letzten Schritt der Eigenschaft *BackColor* zuweisen. Erstellen Sie eine neue Sub-Prozedur mit dem Namen `showColor` und fügen Sie dort folgenden Code ein:

```
Private Sub showColor()
   ' neue Instanz des Datentyps Color erstellen
   Dim currentColor As New Color()

   ' Farbe festlegen
   currentColor = Color.FromArgb(trkRed.Value, trkGreen.Value, trkBlue.Value)

   ' aktuelle Farbe dem Panel-Steuerelement zuweisen
   pnlColorSample.BackColor = currentColor
End Sub
```

Wählen Sie *Debuggen/Debugging starten,* um die Anwendung zu starten, und schauen Sie sich den aktuellen Zwischenstand des Programms an. Neben den Programmfeatures, die noch nicht implementiert sind, gibt es ein kleines Problem. Nach dem Programmstart befinden sich zwar die Schieberegler an der richtigen Position, aber die Farbe im Panel-Steuerelement wird noch falsch angezeigt.

## Ereignishandler für das Form erstellen

Dieses Problem ist schnell gelöst, indem wir eine weitere Ereignisbearbeitungsroutine erstellen, diesmal jedoch nicht für eines der Steuerelemente, sondern eine für das Form selbst.

1. Kehren Sie zum Form *ColorForm* im Formular-Designer zurück und markieren Sie das Trackbar-Steuerelement für die rote Farbe.
2. Öffnen Sie das Eigenschaftenfenster, wenn es nicht geöffnet ist (Shortcut: F4).
3. Klicken Sie im Eigenschaftenfenster auf die Schaltfläche *Ereignisse* (die mit dem gelben Blitz).

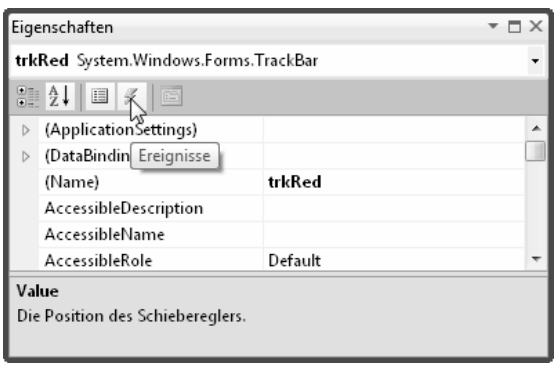



**Abbildung 13.11:** Im Eigenschaftenfenster können außer den Eigenschaften auch die Ereignisse der ausgewählten Komponente angezeigt werden. Hierzu klicken Sie die Schaltfläche *Ereignisse* an.

Die Tabelle im unteren Bereich des Eigenschaftenfensters zeigt in der linken Spalte alle Ereignisse, die für das derzeit im Formular-Designer markierte Element vorhanden sind. Wenn für eines der Ereignisse eine Ereignisbearbeitungsmethode vorhanden ist, finden Sie deren Namen in Fettschrift in der rechten Spalte.

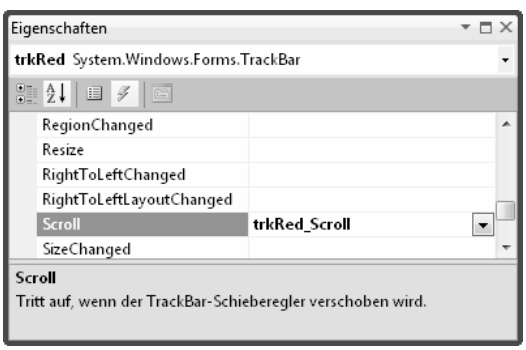



**Abbildung 13.12:** Im Eigenschaftenfenster werden die für das markierte Element vorhandenen Ereignisse angezeigt. Wurde für ein Ereignis ein Ereignishandler definiert, steht der Name der Methode in Fettschrift neben dem Namen des Ereignisses.

In der Dropdownliste im oberen Bereich des Eigenschaftenfensters werden der Name und der Datentyp des derzeit markierten Elements ausgegeben. Unten im Eigenschaftenfenster sehen Sie außerdem noch einen erklärenden Text, der beschreibt, wann das markierte Ereignis ausgelöst wird.

**Tipp: Schnell den Code eines Ereignishandlers anzeigen lassen**

Doppelklicken Sie im Eigenschaftenfenster auf den Namen einer Ereignisbearbeitungsmethode, um den zugehörigen Quellcode im Code-Editor anzeigen zu lassen.

4. Markieren Sie im Formular-Designer das Form `ColorForm`. Achten Sie darauf, dass in der Dropdownliste im Eigenschaftenfenster *ColorForm* angezeigt wird.
5. Suchen Sie nach dem Ereignis *Load* und doppelklicken Sie auf den Namen. Das *Load*-Ereignis wird ausgelöst, nachdem das Form geladen ist. Im Code-Editor wird die leere Methode mit dem Namen `ColorForm_Load` angezeigt.
6. Fügen Sie in den Ereignishandler den Aufruf von `showColor` ein, damit das `Panel`-Steuerelement beim Laden des Forms den aktuellen Farbwert anzeigt:

```
Private Sub ColorForm_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) _
Handles MyBase.Load
   showColor()
End Sub
```

7. Starten Sie diese Version des Programms, um zu sehen, ob alles funktioniert.

## RGB-Farbwert in Hexadezimaldarstellung umwandeln

In diesem Abschnitt werden Sie eine Methode erstellen, die die aktuelle RGB-Farbe in einen String im Hexadezimalformat konvertiert. Anschließend werden Sie die Click-Ereignisbearbeitungsmethode für die Schaltfläche des Forms programmieren und dort diese Hexadezimaldarstellung in die Windows-Zwischenablage kopieren.

1. Erstellen Sie eine neue Function-Prozedur mit dem Namen `convertRGBToHex`, die einen String zurückgibt.

```
Private Function convertRGBToHex() As String

End Function
```

2. Fügen Sie in die Methode den folgenden Code ein:

```
Private Function convertRGBToHex() As String
   ' die einzelnen Farbwerte in Hex-Strings konvertieren
   Dim redValueHex As String = trkRed.Value.ToString("X")
   Dim greenValueHex As String = trkGreen.Value.ToString("X")
   Dim blueValueHex As String = trkBlue.Value.ToString("X")

   ' den String zusammenbauen
   Dim hexColor As String
   hexColor = "#" _
       + redValueHex.PadLeft(2, "0") _
       + greenValueHex.PadLeft(2, "0") _
       + blueValueHex.PadLeft(2, "0")
   Return hexColor
End Function
```

In den ersten drei Anweisungen wird für die aktuellen Werte der Eigenschaft `Value` der drei `TrackBar`-Steuerelemente die Methode `ToString` aufgerufen. Durch die Angabe des Formatstrings, hier des

Parameters "X", wird ToString angewiesen, aus dem Farbwert, der eine Zahl darstellt, eine Zeichenkette im Hexadezimalformat zu erstellen. Diese drei Strings werden drei Variablen zugewiesen.

In der nächsten Anweisung werden die drei Zeichenketten zusammengebaut. Hier wird PadLeft verwendet, um zu erreichen, dass, falls erforderlich, im Rückgabewert von PadLeft zwei (das ist das erste Argument an PadLeft) Nullen (das ist das zweite Argument an PadLeft) voranstehen. Die Zeichenkette wird mit dem Zeichen # eingeleitet, um anzugeben, dass es sich hierbei um eine Hexadezimalzahl handelt.

3. Gehen Sie zum Quellcode von showColor und fügen Sie am Ende der Prozedur folgende Anweisung ein, mit der der Rückgabewert von convertRGBToHex im Label-Steuerelement angezeigt wird:

```
Private Sub showColor()
   ' neue Instanz des Datentyps Color erstellen
   Dim currentColor As New Color()

   ' Farbe festlegen
   currentColor = Color.FromArgb(trkRed.Value, trkGreen.Value, trkBlue.Value)

   ' aktuelle Farbe dem Panel-Steuerelement zuweisen
   pnlColorSample.BackColor = currentColor

      ' Farbwert in Hexadezimaleinstellung im Label txtHexValue anzeigen
      txtHexValue.Text = convertRGBToHex()
End Sub
```

4. Starten Sie das Programm, um zu sehen, ob der neue Code funktioniert.
5. Lassen Sie das Form *ColorForm* im Formular-Designer anzeigen und klicken Sie dazu auf die Registerkarte *ColorForm.vb [Entwurf]*.
6. Doppelklicken Sie auf die Schaltfläche mit dem Text *Kopieren.* Es wird ein leerer Ereignishandler für das Click-Ereignis dieser Schaltfläche im Code-Editor eingefügt.

```
Private Sub btnCopy_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) _
Handles btnCopy.Click

End Sub
```

7. Fügen Sie in diese Methode folgende Codezeile ein:

```
Private Sub btnCopy_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) _
Handles btnCopy.Click
   My.Computer.Clipboard.SetText (txtHexValue.Text)
End Sub
```

So einfach geht es, etwas in die Zwischenablage zu kopieren. Die Klasse Clipboard aus dem Namespace My.Computer stellt Prozeduren bereit, um Daten in der Windows-Zwischenablage abzulegen und von dort abzurufen. Die Prozedur SetText erhält als Parameter den Wert, den wir aus der Eigenschaft Text des TextBox-Steuerelements txtHexValue abrufen. (Mehr über My.Compuer.Clipboard finden Sie in Kapitel 16.)

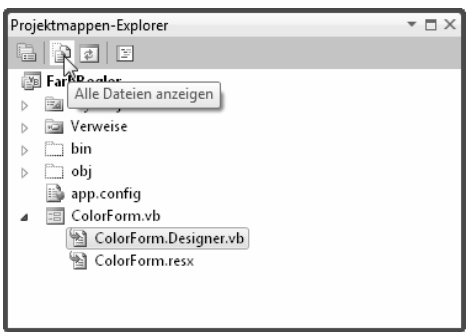
8. Starten Sie das Programm, um zu sehen, ob der neue Code funktioniert. Kopieren Sie den Hexadezimalwert in die Zwischenablage. Starten Sie eine andere Windows-Anwendung, wie beispielsweise den Editor, und fügen Sie dann dort die Zeichenkette ein.

Damit haben Sie Ihre erste Windows Forms-Anwendung fertiggestellt.

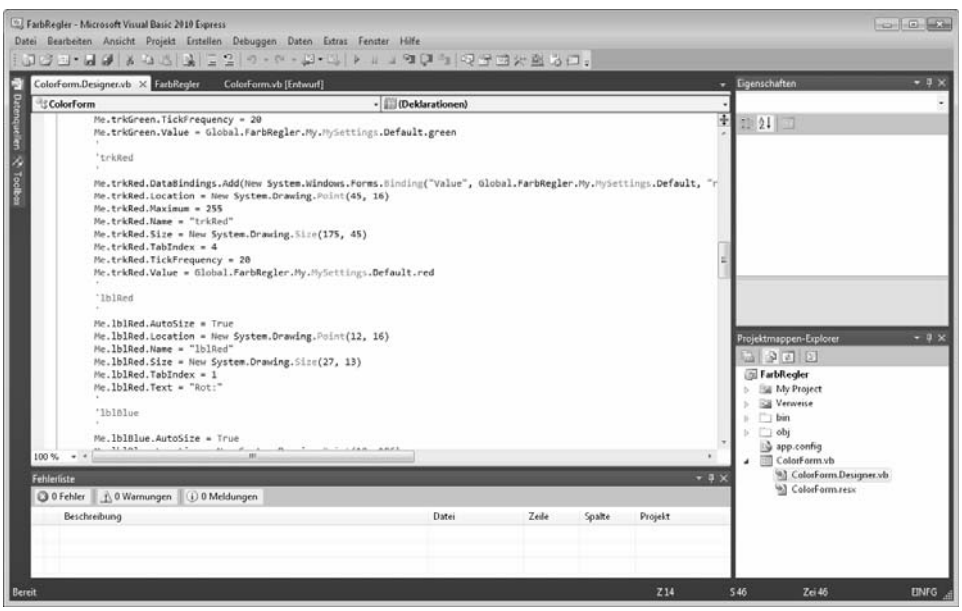
## Background: Die Designer-Datei

Wenn Sie ein Windows Form erstellen, erstellen und bearbeiten Sie Ihren eigenen Code, wie beispielsweise die Ereignishandler, in einer Datei, die den gleichen Namen besitzt wie das Form. Visual Basic 2010 Express erstellt im Hintergrund eine weitere Datei, die sogenannte Designer-Datei, die aber normalerweise vor Ihren Augen verborgen wird.

Der Projektmappen-Explorer besitzt eine Schaltfläche *Alle Dateien anzeigen*. Wenn Sie diese anklicken, werden im Projektmappen-Explorer auch die Dateien angezeigt, die normalerweise ausgeblendet sind. Wenn Sie dann noch das Dreieck vor *ColorForm.vb* anklicken, wird als untergeordnetes Element die Designer-Datei für das Form sichtbar; sie hat den Namen *ColorForm.Designer.vb*.




Klicken Sie *ColorForm.Designer.vb* mit der rechten Maustaste an und wählen Sie *Code anzeigen*, um die Datei zu öffnen.



In der Datei *ColorForm.Designer.vb* befinden sich die Informationen über das Form und die Steuerelemente, die Sie im vorherigen Abschnitt eingefügt haben, und auch alle Informationen über die Eigenschaften, die Sie gesetzt haben. Die Abbildung auf der vorigen Seite zeigt einen Ausschnitt aus der Datei. Im oberen Bereich sehen Sie den Code, mit dem das Steuerelement `trkRed`, also einer der Schieberegler, initialisiert wird, im unteren Bereich sehen Sie die Zuweisungen der Eigenschaften an das Steuerelement `lblRed`. Wenn Sie ganz ans Ende der Datei scrollen, finden Sie einen Anweisungsblock, in dem die Zeilen mit `Friend WithEvents` anfangen. Das sind die Zeilen, über die die Verknüpfung der Ereignisse mit den Steuerelementen und deren Ereignishandlern erst möglich wird. Klicken Sie in der Navigationsleiste auf die Schaltfläche *Schließen,* um die Datei *ColorForm.Designer.vb* wieder zu schließen.

In den meisten Fällen brauchen und sollten Sie den Code dieser Datei nicht ändern. Trotzdem ist es gut zu wissen, dass vieles von dem, was wie Magie erscheint, doch keine Zauberei ist, sondern ein ausgeklügeltes Konzept, bei dem letztendlich auch nur normaler Visual Basic-Code die Arbeit macht.

Wenn Sie Änderungen an der Optik eines Forms vornehmen oder Eigenschaften ändern wollen, dann machen Sie dies im Formular-Designer. Änderungen, die Sie manuell an dem Code dieser Datei vornehmen, werden überschrieben, wenn Sie zu einem späteren Zeitpunkt Änderungen im Formular-Designer vornehmen und Visual Basic 2010 Express die Änderungen in diese Datei schreibt.

Wenn Sie möchten, können Sie die Schaltfläche *Alle Dateien anzeigen* noch einmal anklicken, damit die Dateien wieder aus dem Projektmappen-Explorer verschwinden. Oder Sie lassen die Dateien sichtbar, die nun nicht mehr rätselhaft für Sie sind.

# 13.5 Übungen zu diesem Kapitel

In diesem Abschnitt finden Sie einige Übungen zu diesem Kapitel. Die richtigen Antworten, Lösungen und den Programmcode finden Sie wie immer auf der Website *www.vsxpress.de.* Außerdem finden Sie im Unterordner *Übungen\Kap13* des Ordners, in den Sie die Beispieldateien installiert haben, das Projekt *Farbregler2,* in dem alle Codeänderungen aus den nachfolgenden Übungen implementiert sind.

## Übung 13.1: Label-Steuerelemente ergänzen

Ergänzen Sie das Form `ColorForm` um drei Label-Steuerelemente. Positionieren Sie diese jeweils rechts neben den drei Schiebereglern. Nennen Sie die Label-Steuerelemente **lblRedValue, lblGreenValue** und **lblBlueValue**.

Passen Sie anschließend die drei Ereignishandler `trkRed_Scroll`, `trkGreen_Scroll` und `trkBlue_Scroll` so an, dass im zugehörenden Label-Steuerelement der aktuelle Farbwert als Dezimalzahl ausgegeben wird.

Modifizieren Sie außerdem den Code für das Load-Ereignis des Forms, damit beim Anzeigen des Forms die aktuellen Farbwerte in den Label-Steuerelementen angezeigt werden.

## Übung 13.2: Weitere Anwendungseinstellungen definieren

In dieser Übung sollen Sie eine weitere Anwendungseinstellung erstellen, aus der beim Programmstart die letzte Position des Fensters eingelesen wird und in der die aktuelle Position des Fensters beim Programmende gespeichert wird.

Nennen Sie diese Anwendungseinstellung **FormLocation.** Öffnen Sie im Projekt-Designer auf der Registerkarte *Einstellungen* die Liste *Typ* und wählen Sie `System.Drawing.Point` aus. In diesem Datentyp können Sie eine Position speichern, die aus einer X-und einer Y-Koordinate besteht. Initialisieren Sie diese Einstellung mit den Werten 0;0.

Nachdem Sie die Eigenschaft definiert haben, gehen Sie so vor wie bei den TrackBar-Steuerelementen gezeigt. Verknüpfen Sie dieses Mal jedoch die Eigenschaft *Location* des Forms mit der neuen Anwendungseinstellung.

# 13.6 Zusammenfassung

In diesem Kapitel haben Sie Ihre erste Windows Forms-Anwendung erstellt und gesehen, wie Ihnen Visual Basic 2010 Express mit seinen ausgefeilten Features hierbei viel Arbeit abnimmt.

- Sie können nun ein neues Projekt des Typs *Windows Forms-Anwendung* erstellen und haben gesehen, wie einfach sich der Name der Quelldatei des Forms und damit auch der Name des Forms ändern lässt.
- Sie haben die Toolbox verwendet, um einem Form Steuerelemente hinzuzufügen, und dabei die verschiedenen Verfahren (Ziehen und Ablegen, Doppelklick, Kopieren eines bereits vorhandenen Steuerelements), um so das wiederholte Setzen von gleichbleibenden Eigenschaften zu vermeiden, im Praxiseinsatz kennengelernt.
- Sie haben gesehen, wie Sie Anwendungseinstellungen definieren und diese mit der Eigenschaft eines Steuerelements verknüpfen.
- Sie können Ereignisbearbeitungsroutinen für Steuerelemente und das Form selbst erstellen und so auf Eingaben des Benutzers reagieren.

Das nächste Kapitel wendet sich dem Thema Steuerelemente aus einer anderen Perspektive zu. Sie werden dort lernen, wie einfach es ist, ein eigenes Steuerelement zu erstellen, da Sie dabei nicht bei null anfangen müssen, sondern auf die Funktionalität bereits vorhandener Steuerelemente zurückgreifen können.