

Kapitel 3

Ihre ersten drei .NET 4.0-Anwendungen

In diesem Kapitel:

Motivation	98
Erstellung einer Konsolenanwendung	98
Erstellung einer Webanwendung mit Datenbankzugriff	99
Erstellung einer Windows-Anwendung mit Webservice-Zugriff	104

Motivation

Für ungeduldige Leser bietet dieses Kapitel drei Schritt-für-Schritt-Anleitungen für die Erstellung von drei einfachen .NET-Anwendungen, wobei auf die ausführliche Begründung der Schritte hier zugunsten der schnellen Zielerreichung verzichtet wird. Wenn Sie es nicht so eilig haben, Ihre ersten .NET-Anwendungen auf dem Bildschirm zu sehen, können Sie auch direkt mit dem Kapitel 4 »Grundkonzepte des .NET Framework 4.0« des Buchs fortfahren, um zunächst die Hintergründe zu verstehen.

HINWEIS Die ersten beiden in diesem Kapitel besprochenen .NET-Anwendungen laufen nicht nur unter .NET 4.0 und Visual Studio 2010, sondern auch unter .NET 2.0/3.x und Visual Studio 2005 bzw. 2008. Aus didaktischen Gründen wird bei diesen Einstiegsbeispielen auf die komplexeren Funktionen von .NET 3.x/4.0 verzichtet. Das dritte Beispiel läuft in der dokumentierten Form jedoch nur auf .NET ab 3.5 und Visual Studio ab 2008.

Erstellung einer Konsolenanwendung

Gebiet	Technik
Programmiersprache:	C# 2010
Anwendungstyp:	Konsolenanwendung
Eingesetzte Klassen:	System.DateTime System.Console
Eingesetzte Werkzeuge:	beliebiger Texteditor <i>csc.exe</i>
Installationsvoraussetzungen:	.NET Framework Redistributable 2.0/3.0/3.5/4.0

Die Anwendung fragt nach Ihrem Namen und begrüßt Sie je nach Tageszeit mit einer anderen Grußformel.

Ziel



Abbildung 3.1 Ablauf der ersten .NET-Anwendung

Lösung

Gehen Sie folgendermaßen vor, um Ihre erste einfache C#-Anwendung mithilfe des Kommandozeilen-Compilers *csc.exe* zu erstellen:

1. Öffnen Sie einen Texteditor, z. B. den bei Windows mitgelieferten Editor *Notepad.exe*.
2. Geben Sie den nachstehenden Quellcode ein (oder laden Sie die Beispiele von der Leser-Website herunter).

```

namespace de.WWWings.Konsolenanwendung
{
    class Programm
    {
        static void Main(string[] args)
        {
            string name, begruessungstext;
            System.Console.Title = "WorldWideWings CheckIn";
            System.Console.WriteLine("Ihr Name?");
            System.Console.BackgroundColor = System.ConsoleColor.White;
            System.Console.ForegroundColor = System.ConsoleColor.Red;
            name = System.Console.ReadLine();
            System.Console.ResetColor();
            if (System.DateTime.Now.Hour <= 12)
            { begruessungstext = "Guten Morgen"; }
            else
            { begruessungstext = "Guten Tag"; }
            System.Console.WriteLine("{0}, {1}, willkommen an Bord!", begruessungstext, name);
            System.Console.Beep();
            System.Console.ResetColor();
            System.Console.WriteLine("Drücken Sie eine Taste, um die Anwendung zu verlassen!");
            System.Console.ReadKey();
        }
    }
}

```

Listing 3.1 Quellcode für Beispiel 1

3. Speichern Sie das Programm unter *Konsolenanwendung.cs*.
4. Öffnen Sie ein DOS-Fenster und setzen Sie das aktuelle Verzeichnis auf das Verzeichnis, in dem Sie den Quellcode abgelegt haben.
5. Kompilieren Sie die Anwendung durch den nachstehenden Befehl:

```
c:\WINDOWS\Microsoft.NET\Framework\v4.0.30319\csc.exe Konsolenanwendung.cs
```

HINWEIS Der Pfad kann auf Ihrem System in Abhängigkeit von der verwendeten Windows-Version, dem Systemlaufwerk und der installierten .NET-Version abweichen.

Als Ergebnis der Kompilierung sollte es nun eine Datei *Konsolenanwendung.exe* geben. Starten Sie diese Anwendung von der Kommandozeile aus.

Erstellung einer Webanwendung mit Datenbankzugriff

Gebiet	Technik
Programmiersprache:	C# oder Visual Basic
Anwendungstyp:	Serverseitige Webanwendung
Eingesetzte Klassen:	System.Web.* (ASP.NET) System.Data.* (ADO.NET)
Eingesetzte Werkzeuge:	Visual Studio 2010 oder Visual Web Developer Express 2010
Installationsvoraussetzungen:	Visual Studio 2010 oder Visual Web Developer Express 2010

Ziel

Ziel dieser Aufgabe ist es, eine Webseite zum Anzeige und Editieren von Daten aus einer Datenbank zu entwickeln.

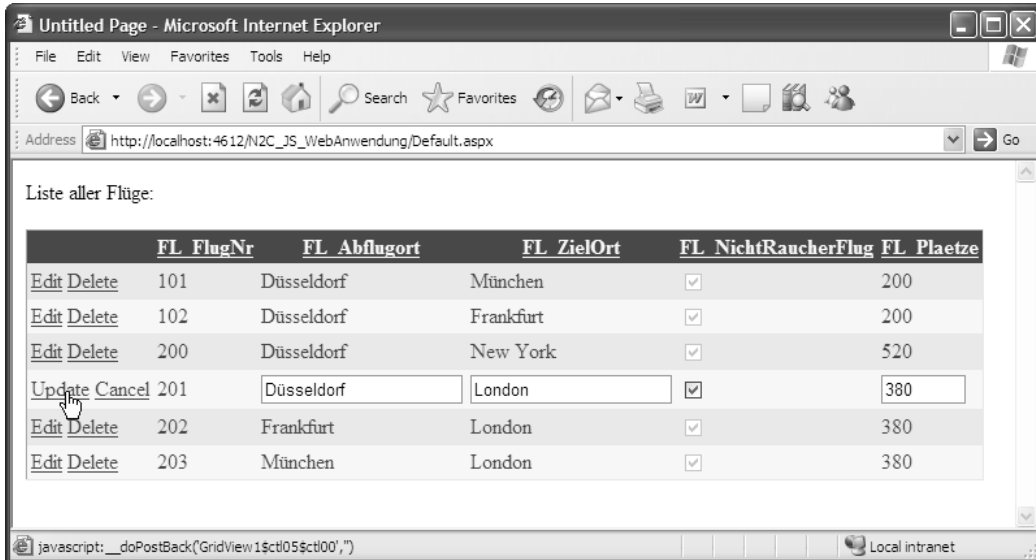


Abbildung 3.2 Webanwendung

Lösung

1. Starten Sie Visual Studio oder Visual Web Developer Express.
2. Wählen Sie das Menü *Datei/Neu/Website* (*File/New/Web Site*).
3. Wählen Sie unter *Vorlagen* (*Templates*) »ASP.NET Website«. Wählen Sie unter *Sprache* (*Language*) eine beliebige Sprache. Wählen Sie unter *Speicherort* (*Location*) »Dateisystem/File System« und einen Ordner in Ihrem lokalen Dateisystem. Klicken Sie auf OK.
4. Visual Studio erzeugt ein neues Projekt mit einer leeren Webseite *default.aspx*. Kopieren Sie die in den Downloads zu diesem Buch enthaltene Microsoft Access-Datenbank *WorldWideWings.mdb* in das Verzeichnis *App_Data* des Projekts.
5. Wählen Sie die leere Webseite aus und wechseln Sie in die Design-Ansicht (*Ansicht/Designer* bzw. *View/Designer*).
6. Geben Sie folgenden Text auf der Seite ein: **Liste aller Flüge.**
7. Öffnen Sie den Server-Explorer (*Ansicht/Server-Explorer* bzw. *View/Server Explorer*). Wenn Sie Visual Web Developer Express verwenden, heißt der Eintrag *Datenbankexplorer* (*Database Explorer*).
8. Wählen Sie auf dem Ast *Datenverbindungen* (*Data Connections*) den Kontextmenüeintrag *Verbindung hinzufügen* (*Add Connection*).

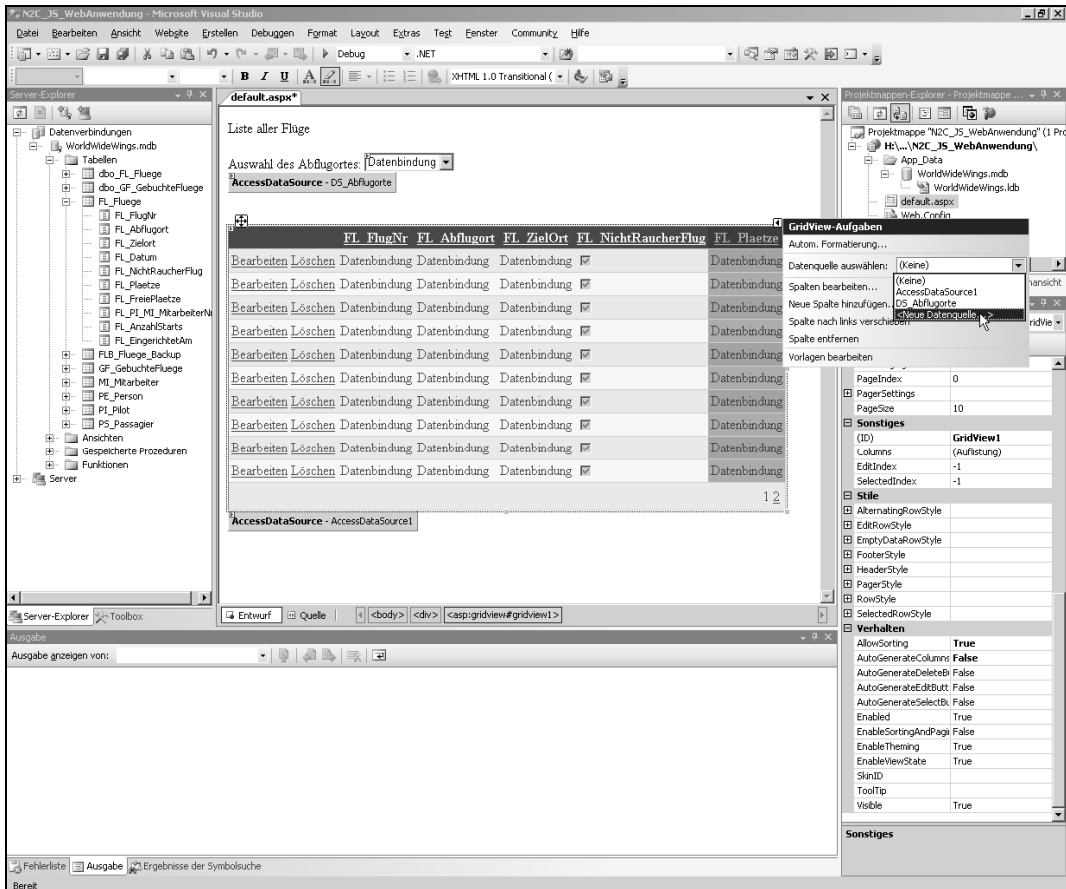


Abbildung 3.3 Konfiguration des Datensteuerelements

9. Erzeugen Sie mit dem Assistenten eine Verbindung zur Datenbank *WorldWideWings.mdf* (entweder mit dem Provider *SQL Server* oder *SQL Server Database File*, wenn Sie eine Microsoft SQL Server Express-Variante verwenden wollen).
10. Die Datenbank erscheint im Server Explorer. Öffnen Sie den Zweig im Server Explorer.
11. Ziehen Sie mit der Maus den Eintrag *Tabellen/Flug* auf die Webseite unter den eingegebenen Text. Visual Studio erzeugt dann automatisch ein GridView-Steuerelement und ein AccessDataSource-Steuerelement für die Verbindung zur Datenbank.
12. Wählen Sie das Smarttag an der oberen rechten Kante des GridView-Steuerelements (*GridView-Aufgaben* bzw. *GridView Tasks*).
13. Wählen Sie im Smarttag-Menü *Automat. Formatierung* (*Auto Format*) und dort *Schiefer/Slade* aus.
14. Aktivieren Sie im Kontextmenü die Häkchen für *Sortieren* (*Sorting*), *Blättern* (*Paging*), *Editieren* (*Editing*) und *Löschen* (*Deleting*).
15. Starten Sie die Anwendung durch Drücken von **[F5]**. Visual Studio startet nun den integrierten ASP.NET Development Server (*WebDev.WebServer.exe*) und die von Ihnen entwickelte Anwendung.

HINWEIS Sie haben für diese Anwendungen keinen Quellcode eingeben müssen, und Sie werden feststellen, dass auch Visual Studio keinen Programmcode generiert hat; Visual Studio hat lediglich XML-Elemente erzeugt. Diese XML-Elemente konfigurieren Web-Steuerelemente, die das gesamte Verhalten dieser Anwendung bereits implementieren. Im Kapitel »ASP.NET« (dieses Zusatzkapitel können Sie als PDF auf dem Leser-Portal herunterladen) werden Sie erfahren, dass Sie viele Standardszenarien ohne eigenen Programmcode realisieren können und dennoch an jeder Stelle die Option haben, durch eigenen Programmcode das vordefinierte Standardverhalten abzuändern.

TIPP Um die Ordnung der Spalten, die Spaltenüberschriften oder die Formatierung zu modifizieren, können Sie *Spalten bearbeiten* (*Edit Columns*) aus dem Smarttag-Menü des GridView-Steuerelements wählen.

Erweiterung des Beispiels

Die Anwendung soll nun um ein Auswahlménü erweitert werden, um nur die Flüge für einzelne Abflugorte anzuzeigen.

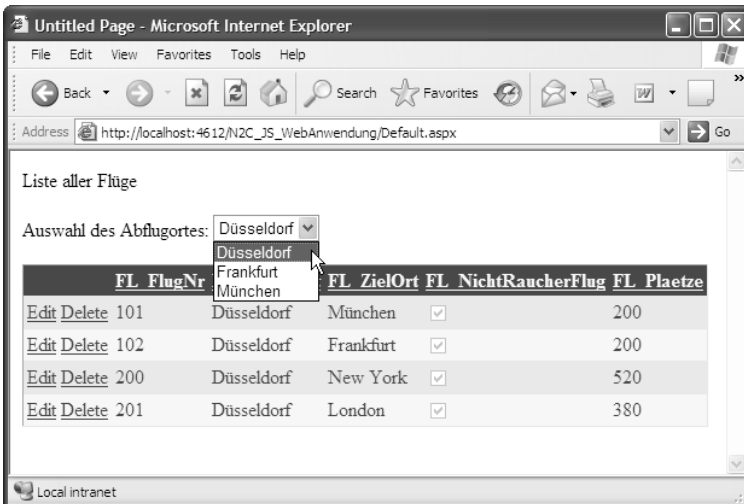


Abbildung 3.4 Erweiterung der Flugverwaltung um ein Auswahlménü für die Abflughäfen

Schritte für die Erweiterung

1. Tippen Sie zwischen Überschrift und Tabelle den Text **Auswahl des Abflugortes** ein.
2. Öffnen Sie die Toolbox (*Ansicht/Toolbox* bzw. *View/Toolbox*).
3. Fügen Sie ein Steuerelement vom Typ *DropDownList* hinter den eingegebenen Text ein.
4. Stellen Sie in den Eigenschaften (*Ansicht/Eigenschaftenfenster* bzw. *View/Properties*) die Eigenschaft *AutoPostBack* auf *True*.
5. Wählen Sie im Smarttag-Menü den Punkt *Datenquelle auswählen* (*Choose Data Source*).
6. Wählen Sie im folgenden Fenster *Neue Datenquelle* (*New Data Source*).
7. Wählen Sie *Access Database* und geben Sie als ID **DS_Abflugorte** ein.

8. Wählen Sie die Datenbankdatei *App_Data/WorldWideWings.mdb* aus.
9. Wählen Sie als Tabelle *Flug* aus. Wählen Sie als einzige Spalte *Abflugort* an. Klicken Sie das Häkchen *Nur eindeutige Zeilen zurückgeben (Return only unique rows)* an.
10. Klicken Sie im nächsten Bildschirm auf *Testabfrage (Test Query)*, um zu sehen, ob die gewünschte Liste der Abflughäfen erscheint.
11. Klicken Sie auf *Fertig stellen (Finish)*.
12. Wenn Sie nun die Anwendung starten, können Sie bereits die Flughäfen auswählen; jedoch reagiert die Tabelle noch nicht auf die Auswahl. Dafür müssen Sie die Datenquelle der Tabelle anpassen.
13. Wählen Sie im Designer im Smarttag-Menü des GridView-Steuerelements den Punkt *Datenquelle konfigurieren (Configure Data Source)*.
14. Überspringen Sie *Datenbank auswählen (Choose a database)*.
15. Auf der Seite *Die Select-Anwendung konfigurieren (Configure Select Statement)* wählen Sie *Tabelle/Name = Flug* und *Spalten = **.
16. Klicken Sie auf die *WHERE*-Schaltfläche.
17. Stellen Sie das Dialogfeld *WHERE-Klausel hinzufügen (Add WHERE Clause)* so ein, wie im folgenden Screenshot dargestellt.

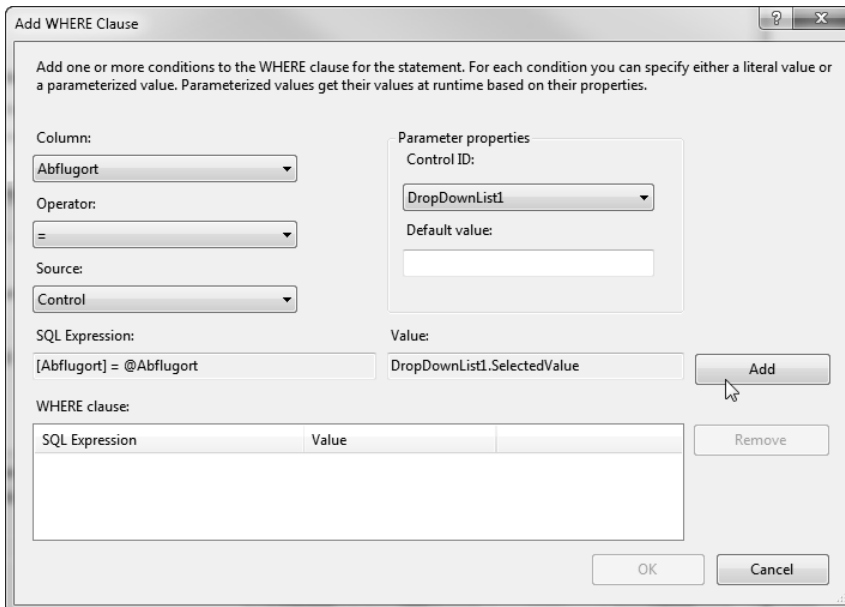


Abbildung 3.5 Bindung der Tabelle an das Auswahlfeld

18. Klicken Sie auf *Hinzufügen (Add)* und *OK*.
19. Wählen Sie in dem übergeordneten Formular nun *Weiter (Next)* und *Fertig stellen (Finish)*.
20. Starten Sie dann die Webanwendung.

Erstellung einer Windows-Anwendung mit Webservice-Zugriff

Gebiet	Technik
Programmiersprache:	Visual Basic 2010
Anwendungstyp:	Windows-Anwendung
Eingesetzte Klassen:	System.Windows.Forms.* (Windows Forms) System.ServiceModel.* (Windows Communication Foundation)
Eingesetzte Werkzeuge:	Visual Studio 2010 (mit VB 2010) oder Visual Basic 2010 Express
Installationsvoraussetzungen:	Visual Studio 2010 (mit VB 2010) oder Visual Basic 2010 Express

Ziel

Ziel dieser Aufgabe ist es, eine Windows-Anwendung zur Übermittlung von Daten an einen XML-Webservice zu entwickeln. Als Webservice wird ein vorhandener Dienst (<http://www.it-visions.de/webservices/leser.asmx>) verwendet, der die Benutzer-Registrierung für die Leser-Website zu diesem Buch ermöglicht. Ausnahmsweise hat dieses Beispiel nichts mit den »WorldWideWings« zu tun, sondern hat für Sie als Leser einen praktischen Nutzen, um die Dienstleistungen für Leser in Anspruch nehmen zu können.

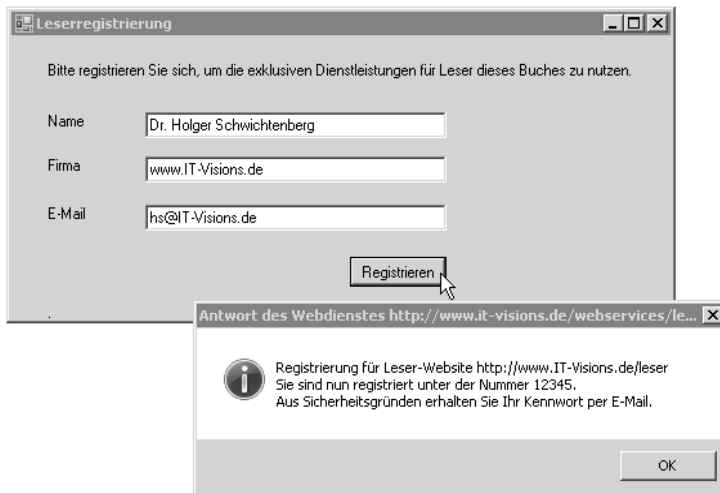


Abbildung 3.6 Webservice-basierte Registrierungsanwendung für Buchleser

WICHTIG Für dieses Beispiel benötigen Sie einen Internetzugang.

Lösung

1. Starten Sie Visual Studio 2010 oder Visual Basic 2010 Express.
2. Wählen Sie das Menü *Datei/Neu/Projekt* (*File/New/Project*).
3. Wählen Sie unter *Projekttypen* (*Project Types*) den Eintrag *Visual Basic/Windows* und *Windows-Anwendung* (*Windows Application*) unter *Vorlagen* (*Templates*) aus. Geben Sie als Namen **N2C_CB_WindowsAnwendung** ein. Nach dem Klick auf die OK-Schaltfläche erzeugt Visual Studio ein Projekt mit einem leeren Fenster (*form1.vb*).
4. Wählen Sie auf der leeren Formularfläche das Kontextmenü *Eigenschaften* (*Properties*) und geben Sie unter der Eigenschaft *Text* das Wort **Leserregistrierung** ein.
5. Vergrößern Sie das Fenster so, dass die komplette Fensterüberschrift lesbar ist.
6. Erstellen Sie das nachstehend abgedruckte Formular, bestehend aus vier Label-Steuerelementen, drei Textbox-Steuerelementen und einem Button-Steuerelement. Nutzen Sie zur Positionierung die blauen Positionierungslinien in Visual Studio, die automatisch beim Bewegen eines Steuerelements erscheinen.

Abbildung 3.7 Positionierungshilfslinien für Windows Forms-Steuerelemente in Visual Studio

7. Vergeben Sie im *Eigenschaften*-Fenster folgende Namen für die Steuerelemente: **C_Name**, **C_Firma**, **C_Email** und **C_Registrieren**.
8. Wählen Sie im Kontextmenü des Projekts im Projektmappen-Explorer den Eintrag *Add Service Reference*.
9. Es öffnet sich das Dialogfeld zum Hinzufügen eines Verweises auf einen Webservice. Geben Sie in die Zeile *URL* ein: **http://www.it-visions.de/webservices/leser.asmx**. Wählen Sie dann die *Gehe zu* (*Go*)-Schaltfläche neben die URL, sodass Sie die Informationsseite wie in der nachstehenden Abbildung sehen.
10. Geben Sie als Namensraum ein: **de.ITVisions.Leser**.
11. Klicken Sie auf *OK*. Im Projektmappen-Explorer erscheint dann unter *Service References* ein neuer Eintrag. Dieser Eintrag repräsentiert eine WCF-Proxy-Klasse, die Visual Studio für den Webservice angelegt hat, sodass der Entwickler diesen aus dem Code wie eine lokale Klasse nutzen kann.
12. Führen Sie einen Doppelklick auf der Schaltfläche *Registrieren* aus. Sie sollten nun das Editor-Fenster mit einer leeren Ereignisbehandlungsroutine **C_Registrieren_Click()** sehen.

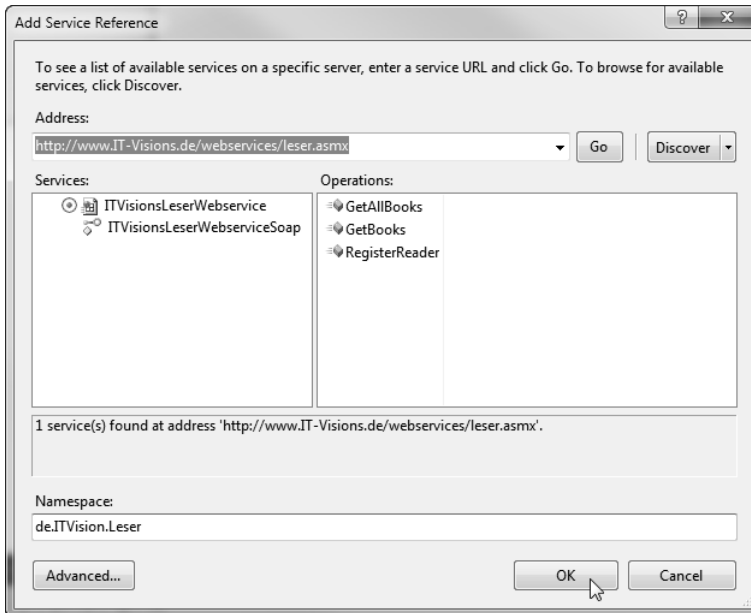


Abbildung 3.8 Erstellen eines WCF-basierten Proxies für einen XML-Webservice

13. Geben Sie dort nachstehende Zeilen ein, um die Methode `RegisterReader()` in der Proxy-Klasse aufzurufen und das Ergebnis in einem Dialogfeld anzuzeigen.

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

    ' Proxy erzeugen
    Dim ws As New de.ITVisions.Leser.ITVisionsLeserWebserviceSoapClient()

    ' Methode aufrufen
    Dim Ausgabe As String = _
        ws.RegisterReader("N35", Me.C_Name.Text,
                        Me.C_Firma.Text, Me.C_eMail.Text)

    ' Proxy schließen
    ws.Close()

    ' Ausgabe
    MsgBox(Ausgabe, MsgBoxStyle.Information, "Antwort des Webdienstes " &
        ws.Endpoint.Address.ToString())

End Sub
```

14. Starten Sie die Anwendung.

HINWEIS Bitte geben Sie Ihren Namen, Ihre Firma und Ihre E-Mail-Adresse an, wenn Sie tatsächlich als Leser für dieses Buch registriert werden und Zugang zu den exklusiven Dienstleistungen für Leser erhalten möchten. Wenn Sie keinen Zugang wünschen, lassen Sie bitte die Felder leer. Sie werden dennoch eine Rückmeldung von dem Webdienst erhalten. Sie können die Registrierung ohne Bedenken mehrfach aufrufen: Der Webdienst erkennt, wenn Sie bereits registriert sind.