

Kapitel 2

World Wide Wings – Das mehrschichtige .NET 4.0- Fallbeispiel in diesem Buch

In diesem Kapitel:

Einleitung	80
Download	80
Szenario	80
Wichtige Hinweise	83
Datenmodell	84
Projekte und Projektmappen	89
Softwarevoraussetzungen für das Fallbeispiel	94
Installation und Konfiguration	95
Mögliche Fehler	95
Fragen zu diesem Fallbeispiel	95

Einleitung

Die meisten Computerbücher verwenden einzelne, isolierte Beispiele auf »Hallo Welt«-Niveau. Dieses Buch geht in Teilen sehr bewusst einen anderen Weg: Viele Programmcodebeispiele basieren auf einem durchgängigen Fallbeispiel, der fiktiven Fluggesellschaft World Wide Wings (WWW), die mehrere verschiedenartige Clients (Web, Desktop, Office) besitzt. In der Datenbank der Fluggesellschaft werden Passagiere, Flüge, Flugbuchungen und Mitarbeiter verwaltet.

Die in diesem Buch hauptsächlich verwendete Version von World Wide Wings trägt die Versionsnummer 0.6.0.

WICHTIG Dieses Kapitel ist eine Kurzfassung der bei dem Download mitgelieferten *Lies Mich*-Datei. Allen Lesern ist angeraten, die *Lies-Mich*-Datei (*readme.pdf*, ca. 50 Seiten) zu lesen, um Schwierigkeiten bei der Nutzung des Beispiels zu vermeiden.

Download

Das Beispiel wird ständig aktualisiert und erweitert. Sie können die jeweils aktuelle Fassung auf folgender Website kostenlos beziehen:

<http://www.world-wide-wings.de>

Szenario

Die Fluggesellschaft World Wide Wings (abgekürzt WWWings oder WWW) ist eine Charter-Fluggesellschaft. Sie bietet nationale, europäische und interkontinentale Flüge an. Die verschiedenen WWWings-Anwendungen verwalten Flüge, Passagiere, Flugbuchungen und Mitarbeiter der Fluggesellschaft.



Abbildung 2.1 Logo der Fluggesellschaft World Wide Wings

Um das Beispiel nicht zu komplex und unübersichtlich zu machen, wurden folgende Annahmen getroffen:

- Es gibt keinen Flugplan, sondern nur individuelle Flüge, die einmalig zu einem bestimmten Termin stattfinden
- Bei Buchungen einer Route aus mehreren Flügen werden die Datumsangaben nicht berücksichtigt
- Es gibt nur eine Buchungsklasse
- Alle Plätze auf einem Flug haben denselben Einheitspreis
- Jedes Flugzeug wird nur von einem Piloten geflogen
- Es gibt keine Flugbegleiter

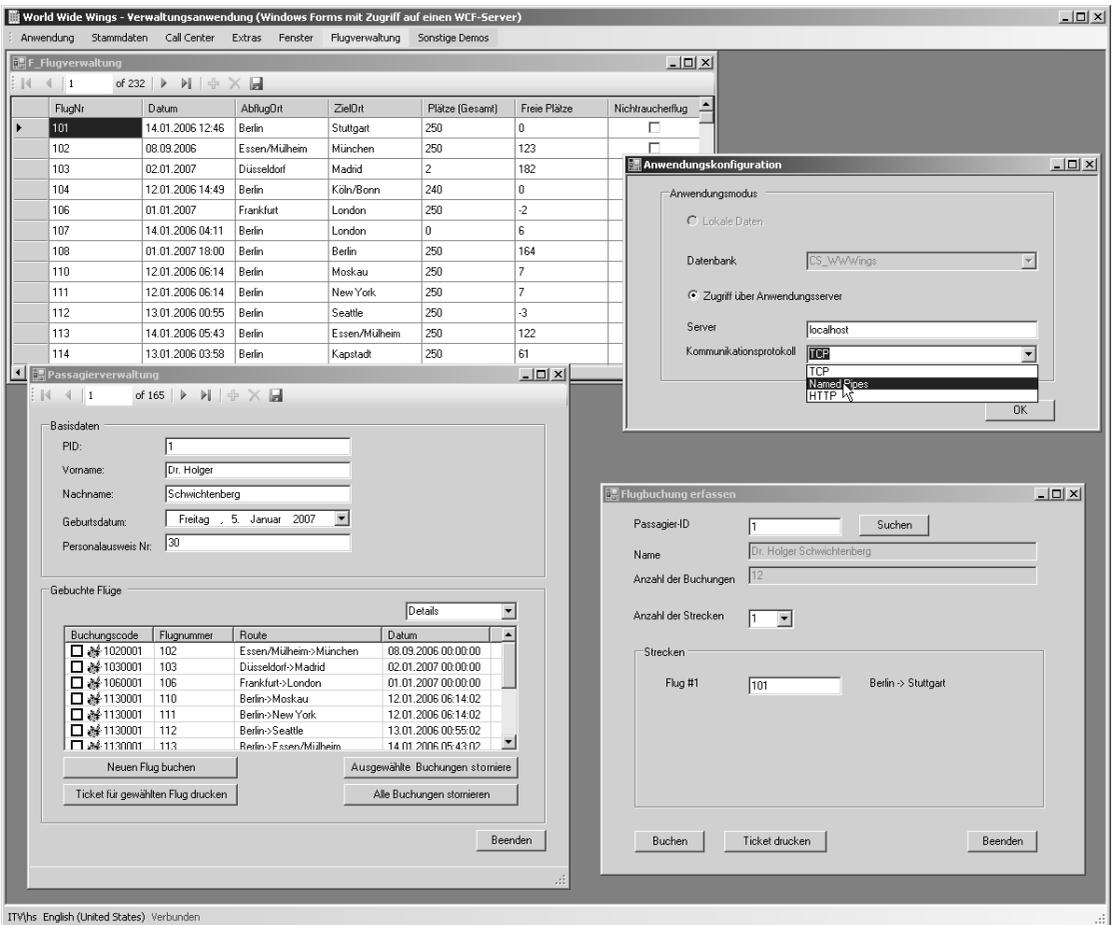


Abbildung 2.2 Beispiel aus der Windows Forms-basierten Desktop-Anwendung



Abbildung 2.3 Beispiel aus der WPF-basierten Desktop-Anwendung

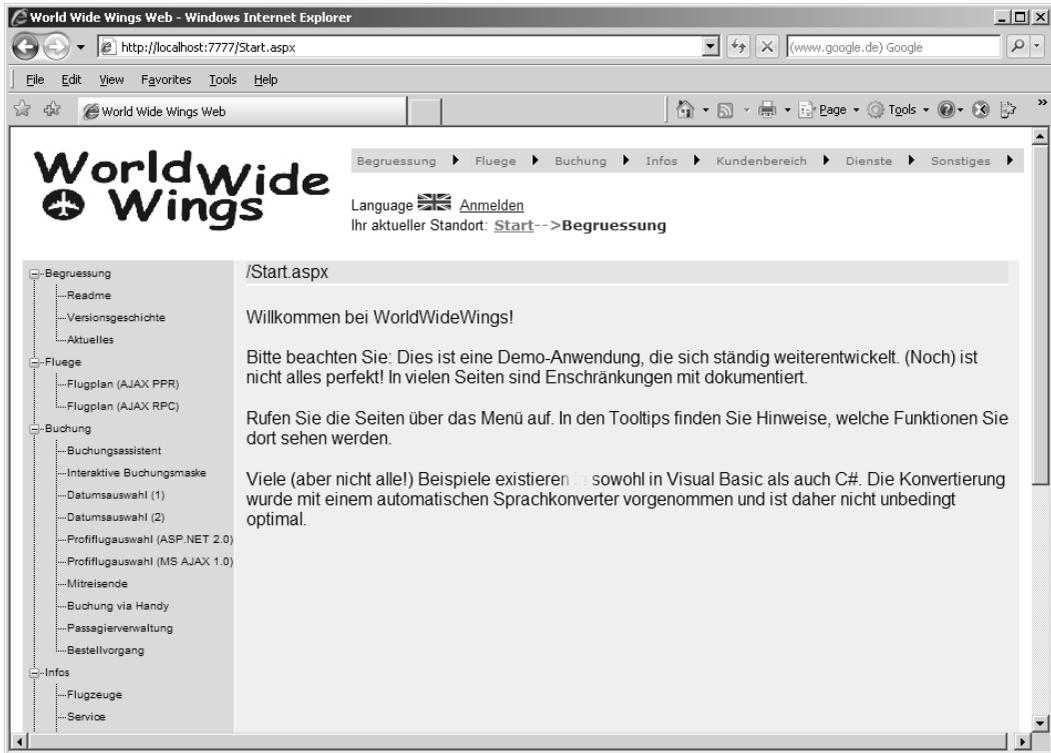


Abbildung 2.4 Beispiel aus der ASP.NET-basierten Webanwendung



Abbildung 2.5 Beispiel aus der Silverlight-Anwendung mit Kopplung an Microsoft Excel (wahlweise im Browser oder außerhalb des Browsers)

Wichtige Hinweise

Bitte beachten Sie zu dem World Wide Wings-Fallbeispiel die folgenden Punkte:

- Die Implementierung des Fallsbeispiels ist nur ein *Prototyp* mit starken Vereinfachungen an verschiedenen Stellen. Erwarten Sie bitte keine vollständige Fluggesellschaftsverwaltungssoftware. Dies wäre wirtschaftlich als ehrenamtliches Nebenprodukt eines Fachbuchs nicht zu leisten. Das Fallbeispiel lässt Ihnen noch genug Raum zum Üben.
- Einige Funktionen (z. B. Pflege der Flugtabelle und Buchen eines Flugs) sind redundant, d. h. mehrfach auf verschiedene Weise (z. B. in einer Seite oder mit Hilfe eines Assistenten) implementiert, um die unterschiedlichen Vorgehensweisen aufzeigen und vergleichen zu können. Bitte wundern Sie sich also nicht, wenn Sie in unteren Schichten des Projekts funktionsidentische Implementierungen von Aufgaben finden.
- Die Webanwendung vermischt unterschiedliche Architekturmodelle in verschiedenen Bereichen in einer Anwendung. Alle Versuche, eine Vielzahl von verschiedenen Ansätzen in verschiedenen Projekten zu pflegen, haben zu einer unhandlich großen Anzahl von Projekten geführt. Daher wurden die Projekte so weit wie möglich konsolidiert.

- Die Implementierung des Fallbeispiels kann in diesem Buch aufgrund der Seitenrestriktionen nur in kleinen Ausschnitten gezeigt werden. Im Buch sind daher nur kurze Ausschnitte des Programmcodes abgedruckt. In den Listing-Unterschriften ist jedoch in eckigen Klammern ein Verweis auf den Standort der Implementierung enthalten. Im Download-Paket zu diesem Buch finden Sie eine vollständigere Implementierung.
- Nicht alle Mechanismen von .NET können sinnvoll an diesem Fallbeispiel gezeigt werden, ohne das Szenario unüberschaubar aufzublähen. Soweit sinnvoll, stützen sich alle Beispiele irgendwie auf das Szenario.
- An einigen Stellen sind aus Vereinfachungsgründen ganz bewusst Servernamen und Pfade in den Programmcode hineincodiert worden. Dies ist kein Anschauungsbeispiel für Softwarearchitektur, sondern die didaktische Notwendigkeit, um in einem Buch die Beispiele kurz und überschaubar zu halten.
- Einige Teile des Fallbeispiels liegen sowohl in C# als auch in Visual Basic vor. Die Übersetzung ist zum Teil durch Sprachkonverter erfolgt und bietet daher nicht immer optimalen Quellcode.
- Die gesamte Anwendung ist im Auslieferungszustand darauf ausgelegt, auf einem System zu arbeiten. Wenn Sie die Anwendung verteilen wollen, müssen Sie Konfigurationen (Verbindungszeichenfolgen, Webservice- und Remoting-URLs) anpassen. Im Standard steht der Rechnername auf ».« bzw. *localhost*, was jeweils den lokalen Rechner bezeichnet.

Datenmodell

Es gibt zwei Datenmodelle, beide für Microsoft SQL Server 2008.

Das ältere Datenmodell (Datenbankschema-Version 4)

Die nachstehende Abbildung zeigt das ältere Datenmodell der WWW-Datenbank, die als Microsoft SQL Server 2008-Datenbank (*WWWings4.mdf*) enthalten ist. Die meisten Teile der Beispielanwendung verwenden dieses Datenmodell.

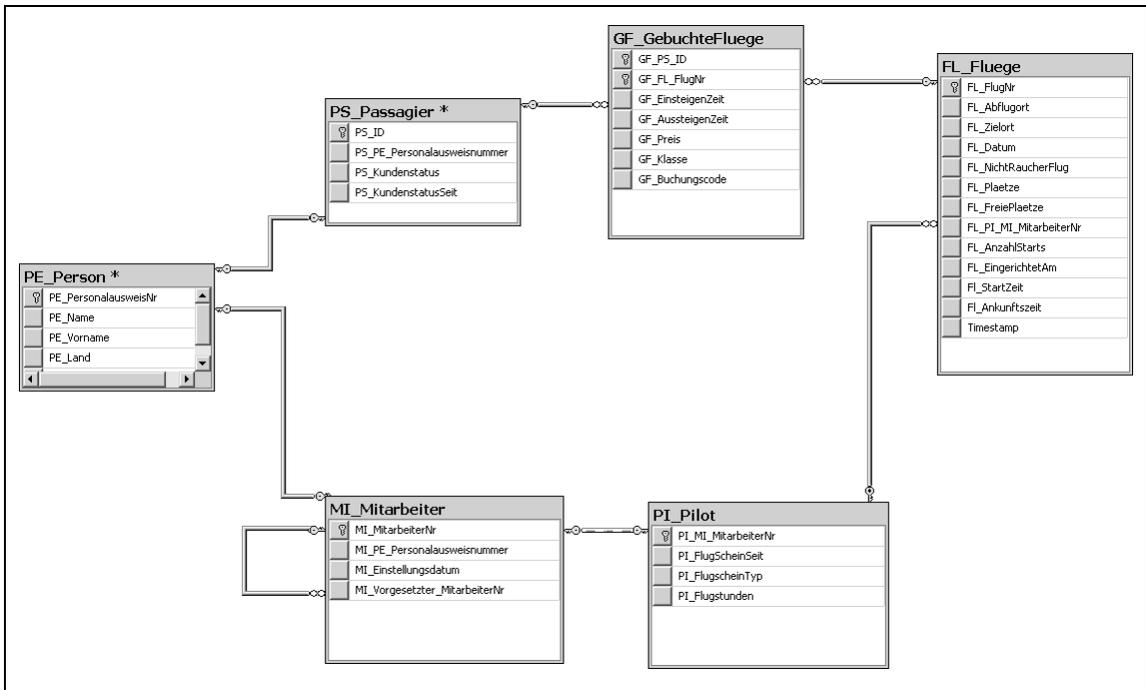


Abbildung 2.6 Älteres Datenmodell des Fallbeispiels (Version 4)

Das neuere Datenmodell (Datenbankschema-Version 6)

Die nachstehende Abbildung zeigt das neuere Datenmodell der WW-Wings-Datenbank, die als Microsoft SQL Server 2008-Datenbank (*WWWings6.mdf*) enthalten ist. Neuere Teile der Beispielanwendung verwenden schon dieses Datenmodell.

Folgende Änderungen wurden gegenüber Schema-Version 4 vorgenommen (insbesondere für das ADO.NET Entity Framework):

- Es wird nun für die Buchungen eine reine M:N-Zwischentabelle (hier: *Flug_Passagier*) verwendet
- Es wurde ein einheitlicher Primärschlüssel für die Tabellen *Person*, *Mitarbeiter* und *Pilot* gewählt
- Auf die Tabellenkürzel wurde verzichtet
- Die Tabellennamen wurden alle in den Singular gesetzt
- Es wurden weiteren Spalten mit bisher nicht verwendeten Datentypen ergänzt (z. B. *Foto*, *Memo*)
- Bisher ungenutzte Spalten (z. B. *EingerichtetAm*, *AnzahlStarts*) wurden entfernt

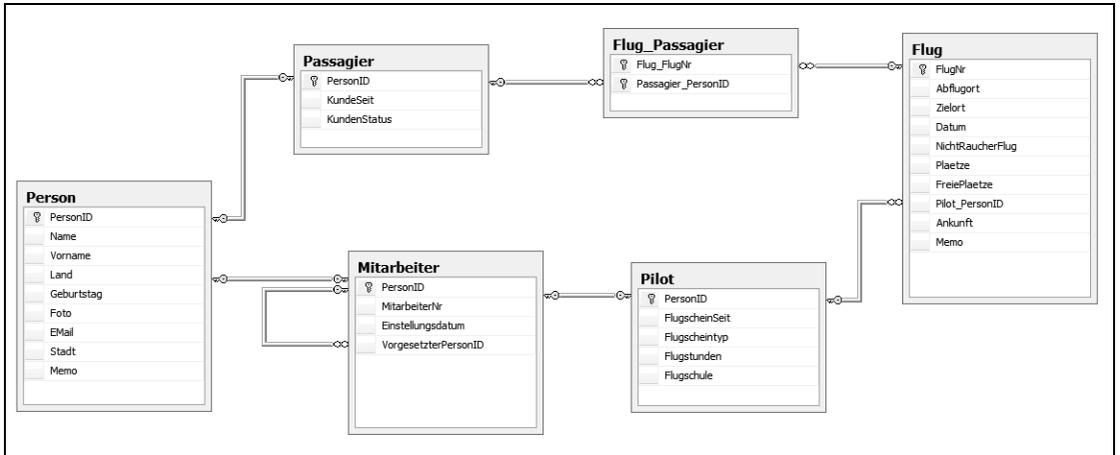


Abbildung 2.7 Neueres Datenmodell des Fallbeispiels (Version 6)

HINWEIS

Die mitgelieferte Microsoft Access-Datenbank ist ein Export der Daten. Sie wird in den Beispielen aber nicht verwendet. Sie können die Beispiele aber entsprechend umbauen.

Eine Weitergabe von Löschoperationen (kaskadierendes Löschen/Löschweitergabe) ist im Datenmodell nur für die 1:N-Beziehung zwischen *Flug* und *Flug_Passagier* (den Buchungen) vorgesehen. Eine Weitergabe bei Änderungen an Primärschlüsseln ist nicht vorgesehen, da die Primärschlüssel in diesem Fallbeispiel als unveränderlich gelten.

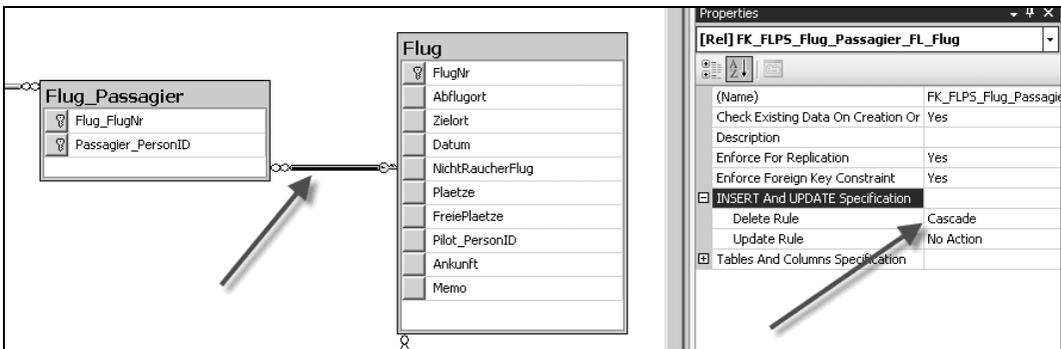


Abbildung 2.8 Löschweitergabe

Mehrschichtarchitektur

Das Fallbeispiel ist in vielen Teilen mehrschichtig aufgebaut: Benutzerschnittstellen sind von Geschäftslogik und Datenzugriff getrennt. Die Benutzerschnittstellensteuerung greift wahlweise direkt auf die Geschäftslogik zu (d.h. die Geschäftslogik wird im Prozess des Clients ausgeführt) oder aber die Benutzerschnittstellensteuerung greift auf einen dedizierten Anwendungsserver zu. Als Kommunikationstechnik stehen drei alternative Lösungen zur Verfügung:

- WCF (gehostet im IIS oder einem eigenen Anwendungsserver, der als Systemdienst oder Konsolenanwendung läuft)
- ASMX (gehostet im IIS)
- .NET Remoting (gehostet in einem eigenen Anwendungsserver, der als Systemdienst oder Konsolenanwendung läuft)

Im Fall von WCF und ASMX gibt es eine serviceorientierte Fassade. .NET Remoting hingegen setzt direkt auf dem domänenspezifischen Objektmodell auf. Die Datenzugriffsschicht läuft immer im gleichen Prozess wie die Geschäftslogik. Eine Verteilung ist hier noch nicht vorgesehen.

WICHTIG An einigen Stellen wird im Fallbeispiel bewusst auf Mehrschichtigkeit ganz verzichtet. Mit den Designern und Assistenten von Visual Studio können sehr schnell Anwendungen »zusammengeklickt« werden, die den Datenzugriffcode mit der Benutzerschnittstelle vermengen. Da am Markt eine Nachfrage nach solchen Rapid Application Development (RAD)-Werkzeugen besteht und Visual Studio diese Szenarien ja auch unterstützt, sind diese Werkzeuge auch in diesem Fallbeispiel enthalten – und sei es nur, um Ihnen die »Gefahren« Ihrer Anwendung zu zeigen.

Das erste Bild zeigt die »alte« Architektur, die in allen Beispielen zum Einsatz kommt, die vor .NET 3.5 entstanden sind. Durchgezogene Pfeile sind prozessinterne Assembly-Referenzen. Gestrichelte Pfeile sind prozessübergreifende Kommunikationsbeziehungen. Die hellgrauen Kästen stellen die Prozessgrenzen dar.

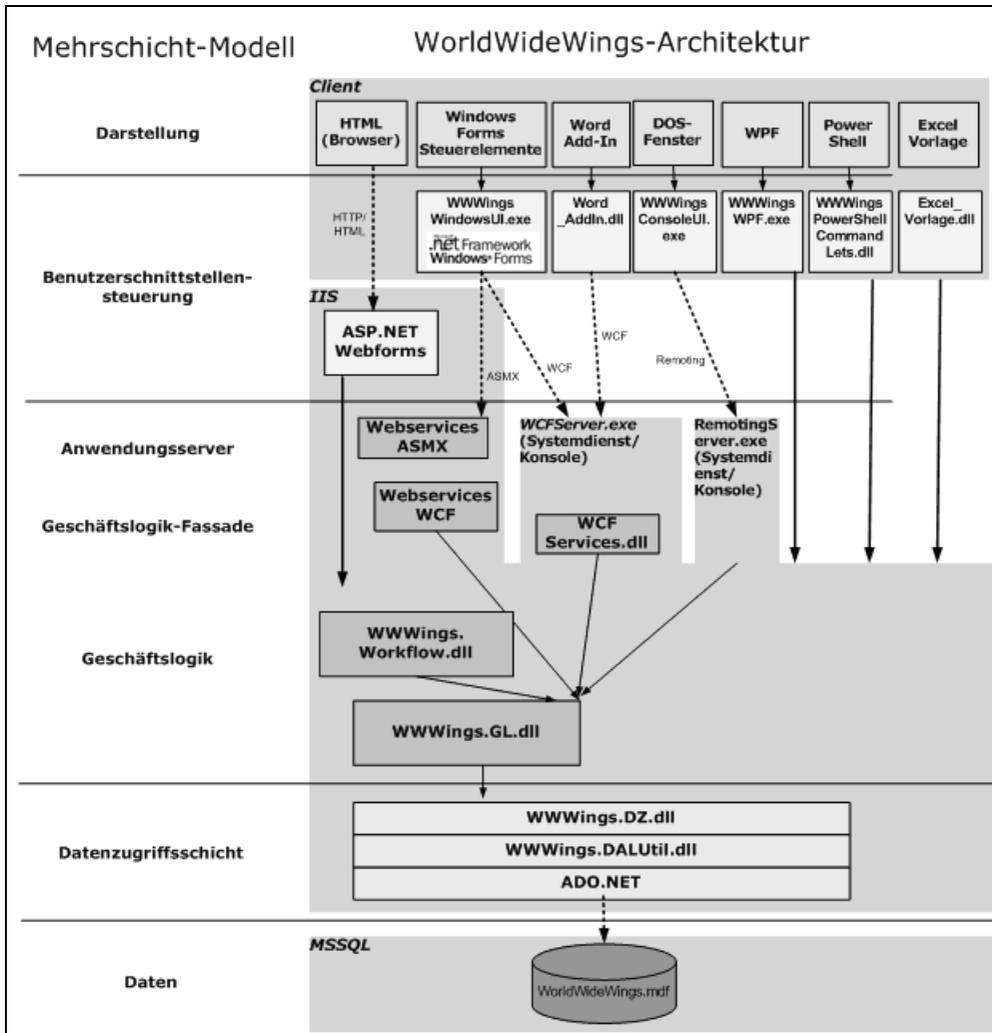


Abbildung 2.9 Gesamtarchitektur des Fallbeispiels (ältere Beispiele mit DB-Schemaversion 4)

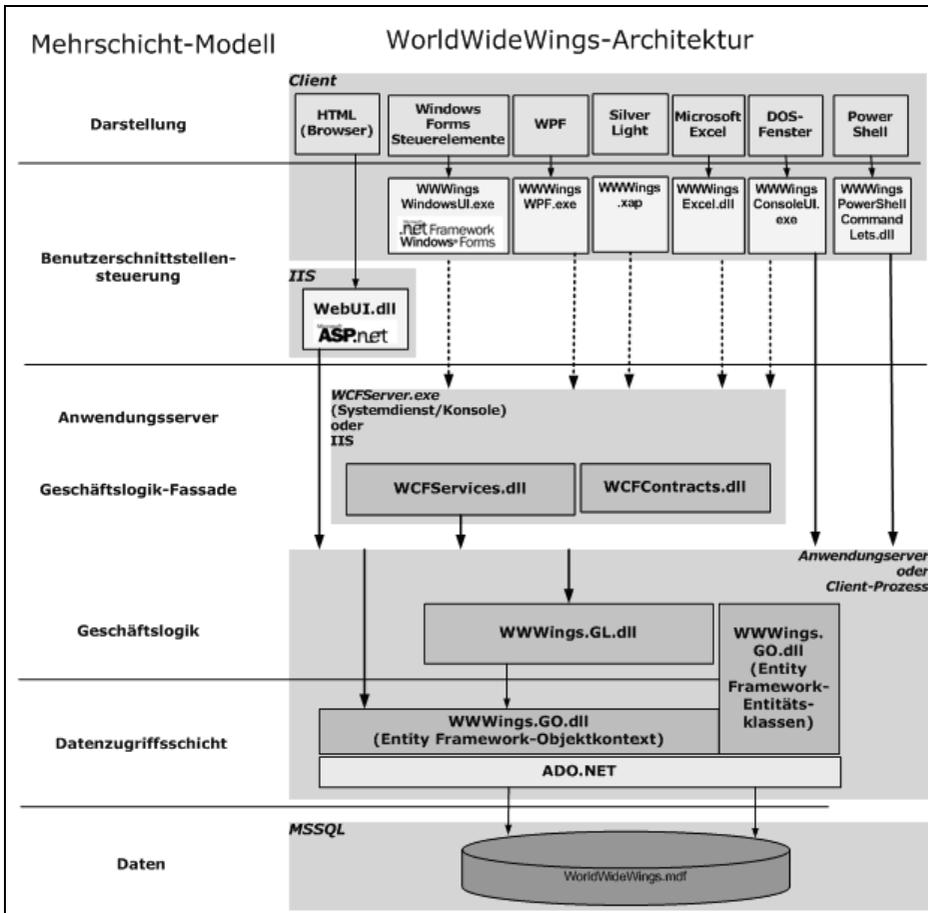


Abbildung 2.10 Gesamtarchitektur des Fallbeispiels (neuere Beispiele mit DB-Schemaversion 6)

Projekte und Projektmappen

Die Anwendung ist in zahlreiche Projekte zerlegt, von denen einige sowohl in Visual Basic als auch in C# existieren. Durchweg ist die C#-Implementierung aber weiter fortgeschritten.

Baustein	Anwendungsart	DB-Schema	Visual Basic-Projekt	C#-Projekt	Erläuterung
DALUtil	DLL	4	ITVisions_DALUtil.vbproj	ITVisions_DALUtil.csproj	Hilfsbibliothek zur vereinfachten Nutzung von ADO.NET (inkl. Provider-Fabriken)
Datenzugriffsschicht (.NET 2.0/3.0-Stil ohne Entity Framework)	DLL	4	WWWings_DZS_ohneEF_VB.vbproj	WWWings_DZS_ohneEF_CS.csproj	Demonstriert den Datenzugriff mit ADO.NET und der DALUtil-Bibliothek

Baustein	Anwendungsart	DB-Schema	Visual Basic-Projekt	C#-Projekt	Erläuterung
Geschäftslogik (.NET 2.0/3.0-Stil ohne Entity Framework)	DLL	4	<i>WWWings_GL_ohneEF_VB.vbproj</i>	<i>WWWings_GL_ohneEF_CS.csproj</i>	Demonstriert die Implementierung eines domänenspezifischen Objektmodells (inkl. Einsatz von Generics) sowie Serviceorientierung auf Basis der ADO.NET-basierten Datenzugriffsschicht
Geschäftslogik (mit ADO.NET Entity Framework)	DLL	6	Nicht verfügbar	<i>WWWings_GL.csproj</i>	Demonstriert die Implementierung eines domänenspezifischen Objektmodells (inkl. Einsatz von Generics) sowie Serviceorientierung auf Basis der Entity Framework-Modelle
Geschäftslogik-objekte (mit ADO.NET Entity Framework-Standardvorlage)	DLL	6	Nicht verfügbar	<i>WWWings_GO_Standard.csproj</i>	Entity Framework-Modell mit Standardvorlage inkl. Modell-erweiterungen
Geschäftslogik-objekte (mit ADO.NET Entity Framework-SelfTrackingEntities-Vorlage)	DLL	6	Nicht verfügbar	<i>WWWings_GO_STE.csproj</i>	Entity Framework-Modell mit STE-Vorlage inkl. Modell-erweiterungen

Tabelle 2.1 Visual Studio-Projekte für untere Schichten

Baustein	Anwendungsart	DB-Schema	Visual Basic-Projekt	C#-Projekt	Erläuterung
Anwendungsserver (.NET Remoting)	EXE	4	<i>WWWings_RemotingServer</i>	Nicht verfügbar	.NET Remoting Server für Geschäftslogik, programmiert als Windows-Systemdienst, der aber auch von der Konsole aus gestartet werden kann
Dienst-Verträge	DLL	4+6	Nicht verfügbar	<i>WWWings_WCF_Contracts</i>	Schnittstellendefinition für WCF-Dienste ▶

Baustein	Anwendungsart	DB-Schema	Visual Basic-Projekt	C#-Projekt	Erläuterung
WCF-Fassade	DLL	4 +6	Nicht verfügbar	<i>WWWings_WCFServices</i>	WCF-Dienste für Zugang zu der Geschäftslogik, zum Hosting in IIS und eigenem WCF-Anwendungsserver
Anwendungsserver (WCF)	EXE	4 + 6	Nicht verfügbar	<i>WWWings_WCFServer</i>	WCF-Server für Geschäftslogik, programmiert als Windows-Systemdienst, der aber auch von der Konsole aus gestartet werden kann
Client Proxies	DLL	4+ 6	Nicht verfügbar	<i>WWWings_WCF_ClientProxies</i>	Client Proxies für WCF-Dienste. Werden eingebunden in Konsolen- und Desktop-Projekte.

Tabelle 2.2 Visual Studio-Projekte für Anwendungsserver

Baustein	Anwendungsart	DB-Schema	Visual Basic-Projekt	C#-Projekt	Erläuterung
Konsolen-Client	EXE	4	<i>WWWings_ConsoleUI_VB.vbproj</i>	<i>WWWings_ConsoleUI_CS.csproj</i>	Einfache Testroutinen für Geschäftslogik und Datenzugriff (lokal und z.T. via .NET Remoting)
Windows-Client mit Windows Forms	EXE	4	Nicht verfügbar	<i>WWWings_WindowsUI_CS.csproj</i>	Windows-Anwendung, erstellt mit Windows Forms
Windows-Steuer-elemente für Windows Forms	DLL	4	<i>WWWings_Steuer-elemente.vbproj</i>	<i>WWWings_Steuer-elemente.csproj</i>	Steuerelementbibliothek für den Windows-Client
Windows-Client mit WPF	EXE	4	Nicht verfügbar	<i>WWWings_WPF.cs</i>	Windows-Anwendung, erstellt mit WPF ▶

Baustein	Anwendungsart	DB-Schema	Visual Basic-Projekt	C#-Projekt	Erläuterung
Sammlung von kleineren Beispielen ohne Zusammenhang zur Gesamtanwendung	DLL	4	<i>VerschiedeneDemos_VB.vbproj</i>	<i>VerschiedeneDemos_CS.csproj</i>	Von der Gesamtanwendung losgelöste Beispiele für die Verwendung der FCL (z. B. ADO.NET, System.Net) und die Syntax der jeweiligen Programmiersprache (z. B. Datentypen). Aufruf durch den Konsolen-Client oder den Web-Client.
Web-Client mit ASP.NET Webforms und ASP.NET AJAX	Mehrere DLL	4 und 6	<i>/Web/Webforms</i>		Web-Client, in dem gemischt die Sprachen VB und C# verwendet werden unter Einsatz von ASP.NET Webforms und ASP.NET AJAX. Gleichzeitig Host für einige ASMX- und WCF-Dienste.
Webserversteuerelemente	DLL	4	<i>WWWings_WebSteuerelemente.vbproj</i>	<i>WWWings_WebSteuerelemente.csproj</i>	Steuerelementbibliothek für den Web-Client
Web-Client mit ASP.NET Dynamic Data	Mehrere DLL	6	Nicht verfügbar	<i>Web_DynData_EF und Web_DynData_LTS</i>	Stammdatenpflege-Website auf Basis von ASP.NET Dynamic Data mit LINQ to SQL (LTS) oder ADO.NET Entity Framework (WF) im Hintergrund.
SilverlightShared Library	DLL	6	Nicht verfügbar	<i>SilverlightSharedLibrary.csproj</i>	Web- & Desktop-Client mit Microsoft Silverlight 4.0: Von Client und Server gemeinsam genutzte Bibliothek
SilverlightServices	DLL	6	Nicht verfügbar	<i>SilverlightServices.csproj</i>	Web- & Desktop-Client mit Microsoft Silverlight 4.0: Dienste für Silverlight ▶

Baustein	Anwendungsart	DB-Schema	Visual Basic-Projekt	C#-Projekt	Erläuterung
SilverlightServer	DLL	6	Nicht verfügbar	<i>SilverlightServer.csproj</i>	Web- & Desktop-Client mit Microsoft Silverlight 4.0: Webserver für Silverlight
Silverlight-ClientProxies	DLL	6	Nicht verfügbar	<i>SilverlightClientProxies.csproj</i>	Web- & Desktop-Client mit Microsoft Silverlight 4.0: Client Proxies für Silverlight
SilverlightClient	DLL	6	Nicht verfügbar	<i>SilverlightClient.csproj</i>	Web- & Desktop-Client mit Microsoft Silverlight 4.0: Silverlight-Client
Client für Microsoft Excel	XSL/DLL	4	<i>WWWings_Office_Client.vbproj</i>	Nicht verfügbar	VSTO-Dokument für Microsoft Excel, das Flugdaten aus dem World Wide Wings-Beispiel ausliest
Clients für Microsoft Word	DOC/DLL	4	Nicht verfügbar	<i>OfficeWord_AddIn und Office-Word_Vorlage</i>	VSTO-Add-In bzw. Vorlage für Microsoft Word, das Flugdaten aus dem World Wide Wings-Beispiel ausliest und in das aktuelle Dokument einfügt
PowerShell Commandlets	DLL	4	<i>WWWings_PowerShell_Commandlets</i>	Nicht verfügbar	Commandlets für die Windows-PowerShell für die WWWings-Geschäftslogik: Get-Flug, Get-Passagier, New-Buchung etc.

Tabelle 2.3 Visual Studio-Projekte für Client/Oberflächen

Baustein	Anwendungsart	DB-Schema	Visual Basic-Projekt	C#-Projekt	Erläuterung
Unit Tests	DLL	4 und 6	Nicht verfügbar	<i>WebUI_Tests.csproj</i>	VSTS-Testprojekt
MVC Unit Tests	DLL	4 und 6	Nicht verfügbar	<i>WWWings_UnitTests.csproj</i>	VSTS-Testprojekt für MVC-Client

Tabelle 2.4 Test-Projekte

Baustein	Anwendungsart	DB-Schema	Visual Basic-Projekt	C#-Projekt	Erläuterung
VisualStudioInstallerSetup	DLL	n/a	Nicht verfügbar	VisualStudioInstallerSetup.vdproj	Setup-Projekt für WPF-Anwendung
Silverlight-ClickOnce	DLL	n/a	Nicht verfügbar	WWWings_UnitTests.csproj	Click Once-Installations-Routine für Silverlight-Anwendung

Tabelle 2.5 Setup-Projekte

Die Projekte sind in mehreren Projektmappen unterschiedlich zusammengestellt für verschiedene Anwendungsfälle, zum Beispiel enthält *WWWings_Web.sln* nur die Projekte, die für den Web-Client benötigt werden. Die Projektmappe *WWWings_AlleProjekte.sln* enthält alle Projekte – die Übersetzung dauert daher hier am längsten.

Softwarevoraussetzungen für das Fallbeispiel

Absolut notwendige Installationsvoraussetzungen für alle Teile der Anwendung sind:

- .NET Framework 4.0.30319.1
- Visual Studio 2010 (4.0.30319.1) Professional Edition
- SQL Server 2008 R2 Express

Einige Teile der Anwendung erfordern außerdem folgende Installationspakete:

- MSMQ-Dienst in Windows (für WCF-Anwendungsserver)
- Visual Studio 2010 Ultimate
- SQL Server 2008 R2 Standard oder Enterprise Edition (nur für ADO.NET Query Notifications)
- Microsoft Word 2007/2010 und Excel 2007/2010 (nur für Office-Clients)
- Visual Studio Tools for Microsoft Office 2010 (nur für Office-Clients)
- Microsoft Silverlight SDK 4.0 (nur für Silverlight-Client)
- Windows PowerShell 2.0
- MSMQ-Dienst (installiert von der Windows-CD) – für WCF-Server

Einige Projekte im Ordner `_SONSTIGEBEISPIELE` erfordern zudem:

- Telerik/Vanatec Open Access 4.0
- NHibernate 2.1

Bitte beachten Sie, dass mit anderen Versionen Probleme bei der Übersetzung und beim Betrieb auftreten können.

WICHTIG Wenn Sie die optionalen Softwarevoraussetzungen nicht erfüllen können, werden einige Projekte nicht laden, nicht kompilieren oder nicht korrekt arbeiten. Schließen Sie diese Projekte aus der jeweiligen Projektmappe aus.

Installation und Konfiguration

Bitte lesen Sie dazu das *Lies Mich*-Dokument (*readme.pdf*).

Mögliche Fehler

Das Fallbeispiel wird vor jeder Auslieferung fehlerfrei übersetzt (siehe Beweis im *readme.pdf*), aber aufgrund begrenzter Ressourcen in diesem ehrenamtlichen Projekt nur in den Kernelementen funktionell getestet. Laufzeitfehler können nicht ausgeschlossen werden; die Übersetzbarkeit ist jedoch getestet worden.

HINWEIS Warnungen sind keine Fehler: Die Projekte erzeugen zum Teil bewusst Warnungen, weil z. B. im Rahmen eines Geschwindigkeitsvergleichs zwischen frühem und spätem Binden einige Variablen nicht typisiert wurden.

Fragen zu diesem Fallbeispiel

Wenn Sie Fragen zu diesem Beispiel haben, nutzen Sie bitte die webbasierten Foren auf der Homepage des Autors:

<http://www.IT-Visions.de/Foren>

Der Autor versichert Ihnen, dass die Beispiele bei ihm korrekt kompiliert wurden. Wenn die Beispiele bei Ihnen nicht kompilieren, prüfen Sie bitte die Softwarekonfiguration. Wenn die Beispiele nicht korrekt ausgeführt werden, prüfen Sie bitte, ob in dem betreffenden Code alle Parameter auf Ihre Umgebung angepasst sind.

