

LightSwitch

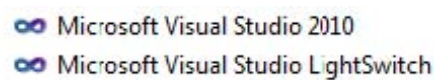
Thomas Reinwart

Zielgruppe sind Entwickler und Power User, die mit Lightswitch einfach und schnell eine Multi-Tier Applikation erstellen können. Alle jene, die zuvor MS Access Anwendungen erstellt haben bietet Lightswitch eine simple Möglichkeit eine Businessanwendung zu erstellen. Der Vorteil bei Lightswitch ist, dass hier das .net Framework (Eigene Lightswitch Namespaces) und die Möglichkeit der Sprachauswahl C#, VB.net verwendet wird. Die erstellten Businessanwendungen bieten also neben der Sprache eine aktuelle Technik wie Desktop- und Web-Anwendung (Silverlight), was mit Access nicht möglich ist.

Erstellte Lightswitch-Anwendungen bauen auf einer klassischen *three tier* Architektur auf, also *presentation*, *logic* und *storage*. Der *presentation tier* ist für die Darstellung der Daten und der Interaktion mit dem Anwender zuständig. Der *logic tier* für das Datenhandling, also Daten prüfen, lesen und updates. Und *storage* für die Datenablage.

LightSwitch Installation

Wenn Visual Studio 2010 (SP1) am System verfügbar ist, wird hier das Lightswitch Projekt Template hinzugefügt.



Weiters wird ein Visual Studio Lightswitch installiert, mit dem ausschließlich Lightswitch Projekte implementiert werden können. HotFix (bei BETA 2) nicht vergessen zu installieren.

Shell / Screens

Die Applikation Shell basiert auf dem MVVM Pattern. Die Screens basieren ebenfalls auf dem MVVM-Pattern und bestehen aus dem three tier Architektur Screen object, screen layout und visual tree.

Wo ist XAML zu finden?

Entwickler von Lightswitch Anwendungen müssen sich mit XAML beim Erstellen der Screens nicht auseinandersetzen. Lightswitch speichert das Layout und erzeugt dynamisch visual tree basierende Silverlight Controls.

Datenbank Entities

Folgende Beziehungen zwischen Tabellen werden unterstützt:

- many-to-one,
- many-to-zero-or-one
- one-to-zero-or-one

one-to-one und many-to-many werden nicht unterstützt, dies kann man aber umgehen in dem man eigene join-Identitäten verwendet.

Datenbank Connectivity

Diese Datenbank Provider werden derzeit unterstützt:

- ADO.NET Entity Framework für SQL Server (2005, 2008, 2008 R2, auch Express) und SQL Azure
- WCF Data Services für SharePoint 2010 mittels OData protocol
- WCF RIA DomainService



	VISUAL STUDIO LIGHTSWITCH	VISUAL STUDIO PROFESSIONAL
LightSwitch Runtime	✓	✓ ¹
Visual Studio Project System	✓	✓
IntelliSense	✓	✓
Team Explorer (Team Foundation Server Integration)	✓ ²	✓ ³
LIGHTSWITCH APPLICATION DEVELOPMENT		
Predefined Screen Templates	✓	✓ ¹
Application Skinning and Theming	✓	✓ ¹
Data Entity Designer	✓	✓ ¹
Business-oriented Data Types (e.g., Email Address, etc.)	✓	✓ ¹
Automatic Data Input Validation	✓	✓ ¹
Windows Azure Deployment	✓	✓ ¹
SQL Azure Support	✓	✓
LANGUAGES, EDITORS & COMPILERS		
Visual Basic	✓	✓
Visual C#	✓	✓
Visual C++	✓	✓
Visual F#	✓	✓
HTML/JavaScript	✓	✓
Silverlight/XAML Editor	✓	✓
PROJECT TYPES		
LightSwitch Application	✓	✓ ¹
ASP.NET	✓	✓
ASP.NET AJAX	✓	✓
ASP.NET MVC	✓	✓
Console Application	✓	✓
Database Projects	✓	✓
Office Applications & Add-ins	✓	✓
Setup Projects	✓	✓
SharePoint Applications & Web Parts	✓	✓
Test Projects	✓	✓
Visual Studio Add-ins	✓	✓
Windows Forms	✓	✓
Windows Phone	✓	✓
WCF	✓	✓
WPF	✓	✓
XNA Games	✓	✓

1. When installed with Visual Studio LightSwitch, but only available for LightSwitch applications.
 2. Team Explorer installed separately, but integrates with LightSwitch.
 3. Requires a Team Foundation Server Client Access License (CAL).

Bild 1: Technischer Vergleich zu Visual Studio Lightswitch / Professional

Bild 2: Auswahl des Applikation Types

Praxis

Anhand einer Beispiel Datenbank (SQL Express 2008 R2) mit den Tabellen *Customer*, *Product*, *Sold*, einer einfachen Verlaufsdatenbank, möchte ich nun die Funktionsweise von Lightswitch zeigen:

Designer

Nach der Auswahl der Darstellungsform (Bild 2) wurden der/die Detailseiten für die Darstellung generiert.

Anschließend füge ich ein Paar Datensätze für die Datenerfassung, Navigation und Suche ein (Bild 5) und starte das Projekt.

Mit der Generierung wurden die Seiten in das Standard Navigationsframework eingefügt, in dem die Navigation, Daten speichern und GUI Designer bereits enthalten sind.

Weitere Anpassungen im Layout kann man nun auch über die Funktion „Bildschirm entwerfen“ während der Laufzeit durchführen (Bild 6).

Individuellen Code ergänzen (Bild 7)

Zu den vordefinierten Events kann man individuellen Code ergänzen, C# oder VB.net, je nachdem was bei der Projektanlage gewählt wurde.

Namespaces

```
using Microsoft.LightSwitch;
using Microsoft.LightSwitch.Framework.Client;
using Microsoft.LightSwitch.Presentation;
using Microsoft.LightSwitch.Presentation.Extensions;
```

Beispielcode

```
namespace LightSwitchApplication
{
    public partial class SearchProducts
    {
        partial void SearchProducts_Activated()
        {
            // Write your code here.
            this.ShowMessageBox(
                "Eingabe Produktbezeichnung",
                "Suche ...",
                MessageBoxOption.Ok);
        }
    }
}
```

Resumee

Der Vorteil ist sicher darin zu sehen, dass die Voraussetzung der technischen Kenntnisse reduzierter sind als bei der herkömmlichen Entwicklung einer Business Anwendung mit asp.net oder WPF im Visual Studio. Man kommt also mit vielen Dingen nicht „in Berührung“, da es wie schon bei Access durch Wizards Dialoge abgedeckt sind. Das spart für den Laien natürlich viel Zeit, die Lernkurve steigt schneller, eine Anwendung ist relativ rasch zusammengestellt und ist unter der Haube mit aktueller Technik versehen.

Von Nachteil kann es sein, wenn ich doch mehr will, als Lightswitch bietet, etwa an Datenbank Connectivity, Zugriff auf Stored Procedures etc.

Damit ist es nun relativ einfach möglich, seine Access Anwendungen endlich los zu werden.

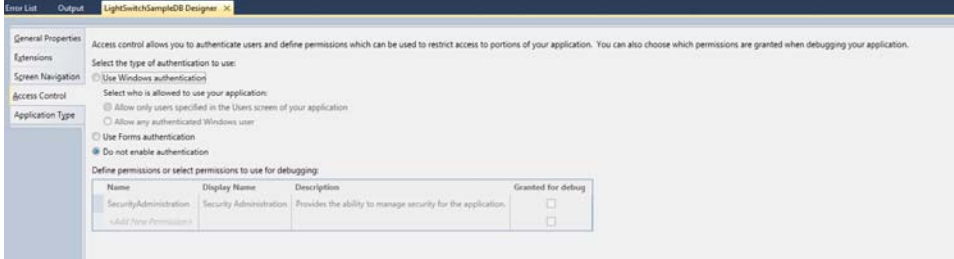


Bild 3: Security ist bereits integriert

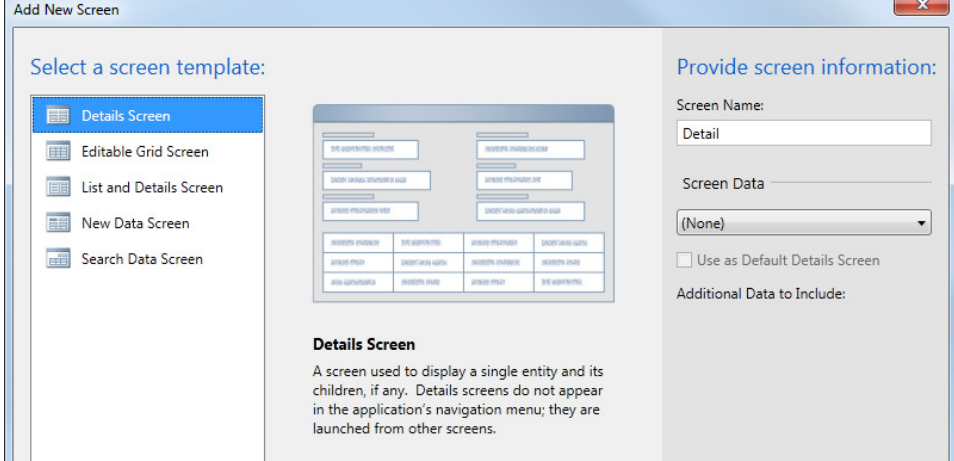


Bild 4: Auswahl der gewünschten Darstellungsform

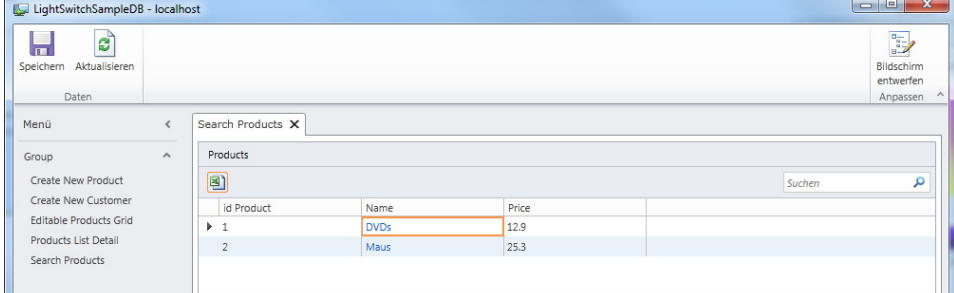


Bild 5: Einige Datensätze eingeben

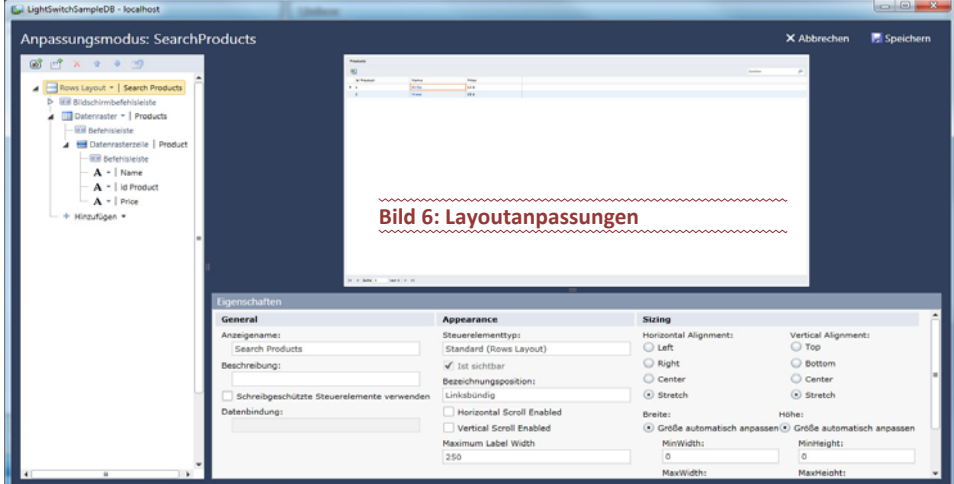


Bild 6: Layoutanpassungen

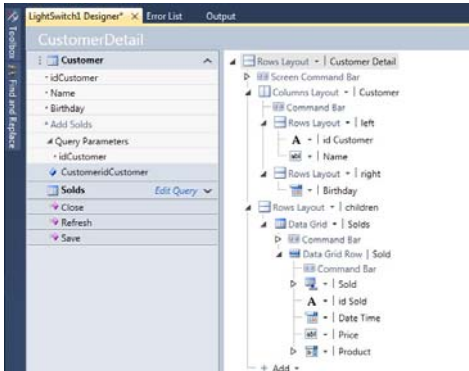


Bild 4: Designer



Bild 7: Code ergänzen