

*Die umfassende Software –
Entwicklungsumgebung zur
Migration von VFX für Visual
FoxPro Anwendungen nach
Silverlight!*

Visual Extend 1.0 für Silverlight



Entwicklerhandbuch

Copyright

Visual Extend ist ein Produkt der ISYS GmbH. Jede Vervielfältigung von VFX-bezogenem Material ist nur nach schriftlicher Genehmigung durch die ISYS GmbH gestattet und in allen VFX-Veröffentlichungen muss die ISYS GmbH als Urheber von VFX ausdrücklich erwähnt werden.

Inhaltsverzeichnis

Einleitung.....	4
Entwicklungswerkzeuge	5
Visual FoxPro 9 und Visual Extend 13	5
Visual Studio 2010.....	5
Silverlight 4 Tools.....	5
Silverlight 4 Toolkit	5
Programmiersprachen	5
Installation.....	6
Startseite.....	7
Anlegen eines neuen Projekts	9
VFX – Silverlight Wizard	10
Was macht dieser Assistent?	10
Auswahl der Silverlight Projektmappe.....	11
Datenzugriff	11
Auswahl von Formularen	12
Auswahl der Projekte für Formulare	13
Auswahl von Formulareigenschaften	13
Auswahl von Berichten	14
Start der VFX für Silverlight Anwendung	15
Weiterbearbeitung.....	16
Architektur	17
Serverprojekte	17
Clientprojekte.....	17
Klassenhierarchie	17
Anwendungsobjekt.....	18
Eigenschaften	18
Formulare	22
Eigenschaften	22
Ausführung von VFP Code.....	23
GUINEU	23
Action Button.....	24
Lokalisierung	25
Lokalisierung von Formularen	26
Verwendung von Bilddateien.....	27
Datei-Manager	27
Optimierung des Datenzugriffs	28
Weitere Informationen	29
Forum, Newsgroup.....	29
Technische Dokumentation.....	29

Einleitung

Visual Extend für Silverlight ist die umfassende Entwicklungsumgebung für die Entwicklung von Geschäftsanwendungen mit Microsoft Silverlight.

In Zusammenhang mit Visual Extend für VFP wird dem VFP Entwickler die einfache Möglichkeit gegeben, bestehende VFX für VFP Projekte nach VFX für Silverlight zu migrieren. Die migrierten Projekte sind in der Bedienung ähnlich zu den vorhandenen VFX für VFP Anwendungen. VFP Entwicklern eröffnen sich damit völlig neue Perspektiven.

Die migrierten Silverlight Anwendungen sind Internetanwendungen und laufen mit VFP Datenbanken und SQL Server Datenbanken, auch wenn die bisherige VFX für VFP Anwendung nicht mit SQL Server Datenbank ausgeführt werden konnte.

Silverlight Anwendungen laufen auf Windows und Mac. Es ist zu erwarten, dass künftig eine größere Anzahl von Plattformen Silverlight unterstützen werden.

In VFX für Silverlight Anwendungen ist die Ausführung von VFP Code auf der Clientseite und auf der Serverseite möglich.

Entwicklungswerkzeuge

Zur Verwendung von VFX für Silverlight sind einige Entwicklungswerkzeuge erforderlich.

Visual FoxPro 9 und Visual Extend 13

Die Migration von bestehenden VFX für VFP Projekten wird mit dem VFX – Silverlight Wizard durchgeführt. Um diesen Assistenten ausführen zu können, sind Installationen von VFP 9 und von VFX 13 erforderlich. Der VFX – Silverlight Wizard ist nur in der Vollversion von VFX 13 enthalten, nicht jedoch in der Trial Version.

VFX 13 kann von der Visual Extend Webseite heruntergeladen werden:

<http://www.visualextend.de/>

Visual Studio 2010

VFX für Silverlight Anwendungen werden mit Visual Studio 2010 und Silverlight 4 entwickelt.

Zur Entwicklung von VFX für Silverlight Projekten kann eine beliebige Version von Visual Studio 2010 verwendet werden, Es kann die kostenlos erhältliche Version „Visual Web Developer 2010 Express“ verwendet werden. In diesem Handbuch wird davon ausgegangen, dass Visual Web Developer 2010 Express verwendet wird. Als Bezeichnung wird allgemein „Visual Studio“ verwendet. Andere Versionen von Visual Studio haben erweiterte Möglichkeiten, die für die Entwicklung von VFX für Silverlight Anwendungen jedoch nicht erforderlich sind.

Visual Web Developer 2010 Express kann von der Microsoft Webseite heruntergeladen werden:

<http://www.microsoft.com/express/Downloads/>

Silverlight 4 Tools

Nach der Installation von Visual Studio müssen die Silverlight 4 Tools installiert werden.

Visual Web Developer 2010 Express kann von der Microsoft Webseite heruntergeladen werden:

Silverlight 4 Tools können von der Microsoft Silverlight Webseite heruntergeladen werden:

<http://www.silverlight.net/getstarted/>

Silverlight 4 Toolkit

Der Silverlight 4 Toolkit kann von der Codeplex Webseite heruntergeladen werden:

<http://silverlight.codeplex.com/>

Im Silverlight 4 Toolkit sind Steuerelemente enthalten, die die auf den Silverlight 4 Basisklassen basieren und eine erweiterte Funktionalität bieten.

Programmiersprachen

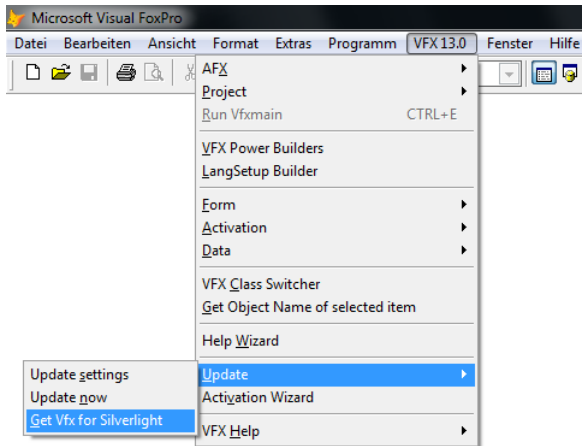
Die Entwicklung von VFX für Silverlight Anwendungen erfolgt mit der Programmiersprache C#. Durch die Verwendung von GUINEU ist auch die Programmierung mit VFP möglich. Außerdem kann VFP Code in COM Servern ausgeführt werden.

Die Benutzeroberfläche von Silverlight Anwendungen wird mit der Sprache XAML beschrieben.

Installation

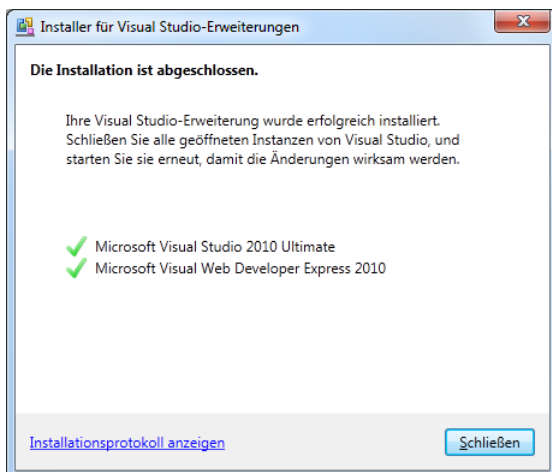
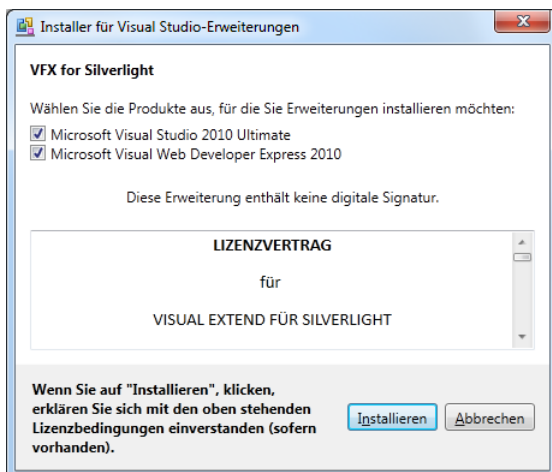
VFX für Silverlight ist eine Erweiterung von Visual Studio und wird mit einem Visual Studio Erweiterungs-Installer installiert. Das Installationsprogramm hat den Namen VfxForSilverlight.vsix.

VFX für Silverlight kann aus dem Menü von VFX für VFP über den Menüpunkt „Update“, „Get VFX for Silverlight“ heruntergeladen werden.



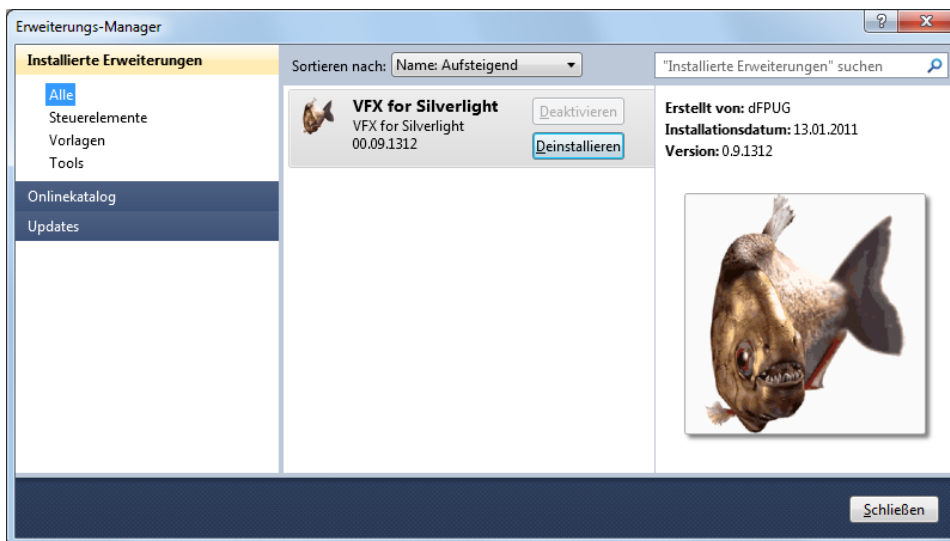
Während der Installation müssen alle Instanzen von Visual Studio geschlossen sein.

Das Installationsprogramm erscheint in der Sprache der verwendeten Windows-Version und erkennt alle installierten Versionen von Visual Studio. Es kann ausgewählt werden, zu welcher Version von Visual Studio die Erweiterung VFX for Silverlight installiert werden soll.



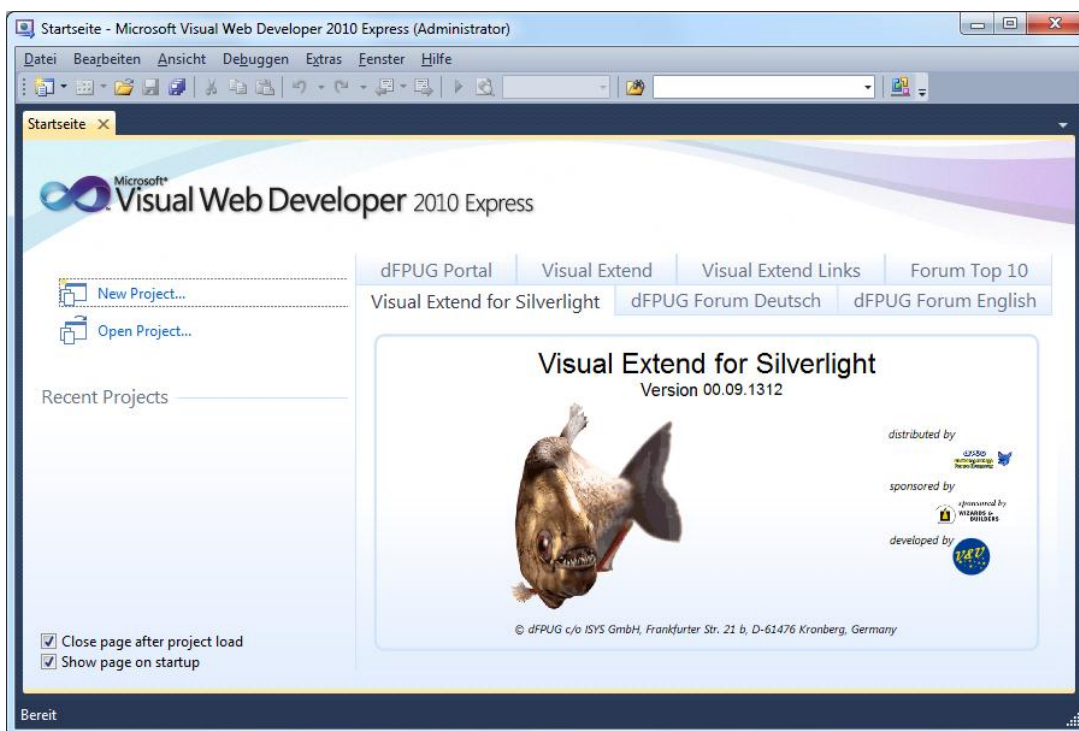
Nach erfolgreicher Installation erscheint im Dialog „Neues Projekt“ von Visual Studio unter C# die Vorlage „VFX for Silverlight“. Basierend auf diesem Vorlageprojekt werden neue VFX für Silverlight Anwendungen erstellt.

Falls VFX für Silverlight deinstalliert werden soll, muss Visual Studio mit Administratorrechten gestartet werden. Über den Menüpunkt „Extra“, „Erweiterungs-Manager...“ kann die Deinstallation durchgeführt werden.



Startseite

VFX für Silverlight enthält eine Startseite für Visual Studio. Um diese Startseite zu verwenden, muss in Visual Studio im Dialog Extras, Optionen die Option „Alle Einstellungen anzeigen“ aktiviert werden. Anschließend kann auf der Seite Allgemein, Start unter „Startseite anpassen“ die Seite „VFX für Silverlight“ eingestellt werden.



Auf der Startseite von VFX für Silverlight stehen die gleichen Optionen zur Verwaltung von Projekten wie auf der Startseite von Visual Studio zur Verfügung. Es können neue Projekte angelegt und bestehende Projekte geöffnet werden. Am linken Rand befindet sich eine Liste mit zuletzt verwendeten Projekten.

Zusätzlich stehen auf der VFX für Silverlight Startseite sieben Seiten mit Informationen zur Verfügung. Die erste Seite wird beim Start angezeigt und zeigt insbesondere die Versionsnummer von VFX für Silverlight. Die Informationen auf den weiteren Seiten werden nur angezeigt, wenn eine Internetverbindung vorhanden ist.

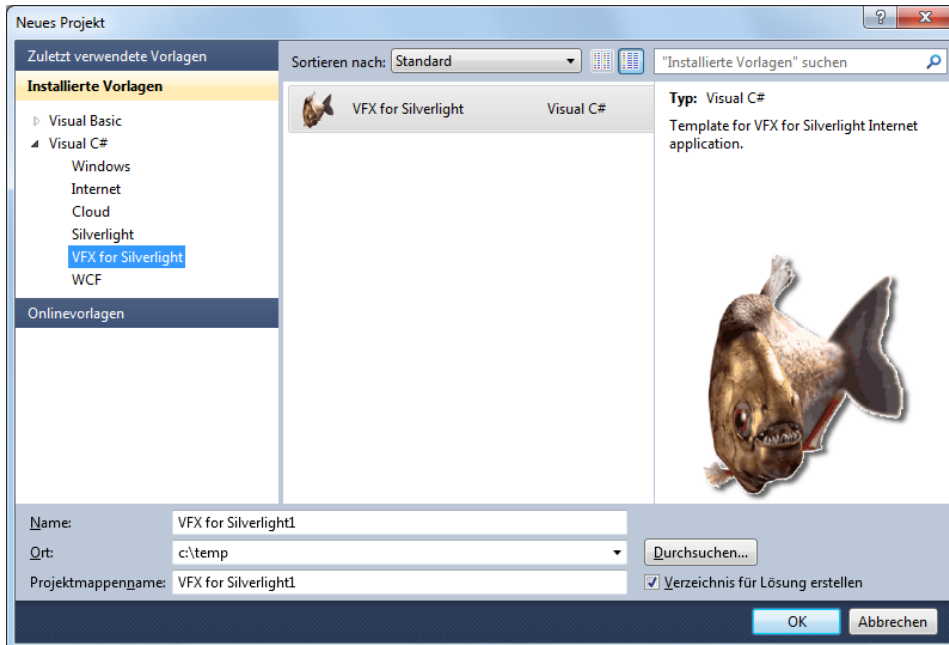
Auf der zweiten und dritten Seite werden die letzten 50 Nachrichten aus dem deutschsprachigen und dem englischsprachigen Forum angezeigt. Es folgt die Portalseite der dFPUG. Aus dem Portal können zahlreiche Dateien heruntergeladen werden. Die nächste Seite zeigt die aktuelle Webseite von Visual Extend. Es folgt eine

Seite mit Links rund um Visual Extend. Die letzte Seite schließlich zeigt die Rangliste der aktivsten Schreiber im Forum.

Anlegen eines neuen Projekts

Um ein neues Projekt, basierend auf VFX für Silverlight, zu erstellen, wird in Visual Studio der Dialog „Neues Projekt“ über die Startseite, den Menüpunkt „Datei“, „Neues Projekt...“ oder mit einem Klick auf die Schaltfläche „Neues Projekt“ in der Symbolleiste geöffnet.

In diesem Dialog wird „VFX for Silverlight“ unter „Visual C#“ „VFX for Silverlight“ ausgewählt. Im mittleren Teil des Dialogs wird dann das Vorlageprojekt „VFX for Silverlight“ ausgewählt.



Das neue Projekt sollte den Namen der zu migrierenden Anwendung erhalten. Außerdem ist ein Ordner anzugeben, in dem das neue Projekt gespeichert werden soll. Dieser Ordner ist später im VFX – Silverlight Wizard anzugeben.

Das Projekt enthält einen mit VFP erstellten COM Server. Das Projekt ist zur weiteren Bearbeitung mit dem VFX – Silverlight Wizard vorgesehen. Die direkte Ausführung ist nicht vorgesehen, jedoch möglich, wenn der VFP COM Server registriert ist.

VFX – Silverlight Wizard

Der nächste Entwicklungsschritt ist die Migration einer vorhandenen VFX für VFP Anwendung nach VFX für Silverlight. Das zu migrierende Projekt muss mit VFP geöffnet werden.

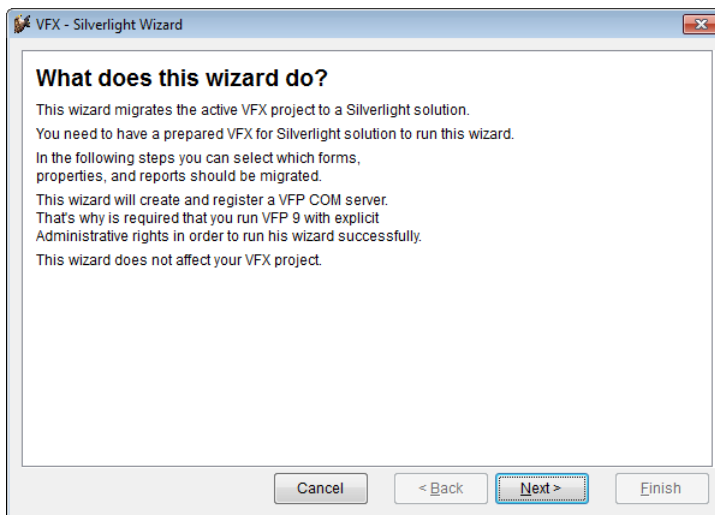
Wenn die Projektmappe in Visual Studio während der Ausführung des VFX – Silverlight Wizard geöffnet bleiben soll, müssen in Visual Studio alle Dateien gespeichert werden, bevor der VFX – Silverlight Wizard gestartet wird.

In Visual Extend für VFP ist der VFX – Silverlight Wizard integriert und kann über den Menüpunkt „VFX 13.0“, „Projekt“, „Silverlight Wizard“ gestartet werden. Dieser Assistent ermöglicht die Migration von Formularen und Berichten in ein VFX für Silverlight Projekt. Dieser Assistent führt den Entwickler durch wenige Schritte.

Der Assistent erstellt und registriert einen VFP COM Server. Auf Windows Vista und neueren Windows Versionen muss VFP 9 explizit mit Administratorrechten ausgeführt werden, damit dieser Assistent erfolgreich ausgeführt werden kann.

Die migrierte VFX für Silverlight Anwendung greift in der Entwicklungsumgebung auf die gleiche Datenbank zu, wie die VFX für VFP Anwendung. Jede Tabelle der Datenbank muss über einen Primärschlüssel verfügen.

Was macht dieser Assistent?

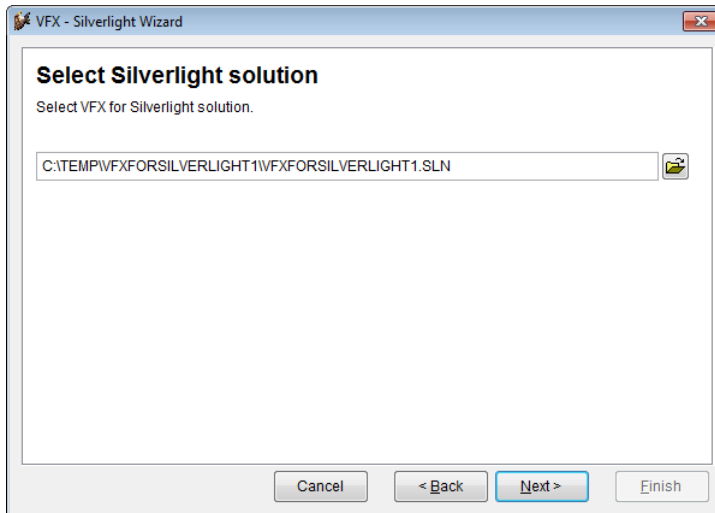


Im ersten Schritt wird die Funktionsweise des Assistenten erläutert. Der Assistent migriert das aktive VFX Projekt in eine Silverlight Projektmappe. Es muss eine vorbereitete VFX für Silverlight Projektmappe vorhanden sein, um diesen Assistenten ausführen zu können. In den folgenden Schritten des Assistenten kann ausgewählt werden, welche Formulare, Eigenschaften und Berichte migriert werden sollen.

Der Assistent erstellt und registriert einen VFP COM Server. VFP 9 muss daher explizit mit Administratorrechten ausgeführt werden, damit dieser Assistent erfolgreich ausgeführt werden kann.

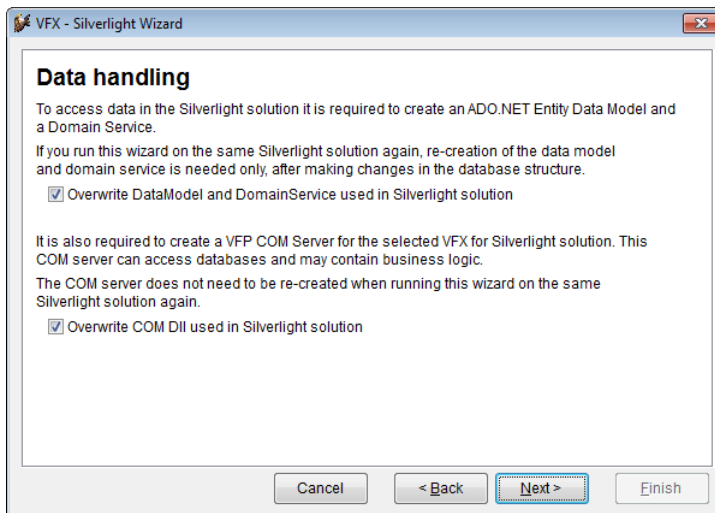
Der Assistent verändert nicht das VFX für VFP Projekt.

Auswahl der Silverlight Projektmappe



In diesem Schritt wird eine zuvor vorbereitete VFX für Silverlight Projektmappe ausgewählt. Die vom VFX – Silverlight Wizard erstellten Formulare und Berichte werden der hier ausgewählten Projektmappe hinzugefügt.

Datenzugriff



In diesem Schritt werden der VFX für Silverlight Projektmappe Komponenten für den Datenzugriff hinzugefügt.

Der Datenzugriff der Silverlight Anwendung erfolgt über ein ADO.NET Entity Data Model und einen Domain Service. Beide Komponenten werden vom Assistenten generiert.

Die Generierung eines ADO.NET Entity Data Model und eines Domain Service ist bei wiederholter Anwendung des Assistenten auf dem gleichen Projekt nur dann erforderlich, wenn zwischenzeitlich Änderungen an der Datenstruktur vorgenommen wurden. Bei der ersten Anwendung des Assistenten auf einem VFX für Projekt müssen ADO.NET Entity Data Model und Domain Service auf jedem Fall generiert werden.

Bei der ersten Anwendung des Assistenten muss ein VFP COM Server erstellt werden. Über den COM Server kann der Zugriff auf Datenbanken erfolgen. Wenn VFP Datenbanken verwendet werden sollen, muss der Datenzugriff über den COM Server erfolgen. Im COM Server kann außerdem mit VFP erstellte Geschäftslogik abgelegt werden.

Der VFX – Silverlight Wizard sucht nach der Datei Config.vfx im Ordner des VFX für VFP Projekts. Wenn diese Datei gefunden wird, erstellt der Assistent eine neue Datei Config.vfx im Ordner VfxDataLayer.Web\VFPCOMDomainService der VFX für Silverlight Projektmappe. In dieser Datei Config.vfx wird mit vollständigen Pfadnamen auf die VFP Datenbank verwiesen, die im VFX für VFP Projekt verwendet wird. Dem Namen des Eintrags für die Datenbank in der Datei Config.vfx wird der Präfix „VFP_“ vorangestellt.

Die Entwicklungsumgebungen des VFX für VFP Projekts und der VFX für Silverlight Projektmappe verwenden somit die gleiche Datenbank.

In der Datei Config.vfx vorhandene Einträge, die auf SQL Server Datenbanken verweisen, die die VFX Systemtabellen enthalten, werden nicht in die Datei Config.vfx der VFX für Silverlight Projektmappe übernommen. Stattdessen werden der Datei Web.config Verbindungszeichenfolgen hinzugefügt.

In der Datei Config.vfx vorhandene Einträge, die auf SQL Server Datenbanken verweisen, die die VFX Systemtabellen nicht enthalten, werden in die Datei Config.vfx der VFX für Silverlight Projektmappe mit vollständigen Pfadnamen übernommen.

Wenn keine Datei Config.vfx vorhanden ist, wird die in dem VFX für VFP Projekt enthaltene Datenbank verwendet.

Für die einwandfreie Funktion der VFX für Silverlight Anwendung ist es unbedingt erforderlich, dass alle Tabellen der Datenbank über einen Primärindex verfügen. Von der Anwendung benötigte Relationen müssen in der Datenbank eingetragen sein.

In der VFX für Silverlight Projektmappe werden ein Datenmodell und ein Domain Service angelegt.

Das Datenmodell wird basierend auf der Struktur der ersten SQL Server Datenbank in der Datei Config.vfx generiert. Falls in der Datei Config.vfx kein Eintrag auf eine SQL Server Datenbank befindet, wird der erste Eintrag genommen, der dann auf eine VFP Datenbank verweist.

Das Datenmodell wird mit dem Namen <Name der Datenbank>Model.cs im Ordner VfxDataLayer.Web\Models angelegt. Der Domain Service wird mit dem Namen <Name der Datenbank>Service.cs im Ordner VfxDataLayer.Web\Services angelegt.

Das für den Zugriff auf VFX Systemtabellen erforderliche Datenmodell und der hierfür erforderliche Domain Service sind in dem VFX für Silverlight Vorlageprojekt enthalten.

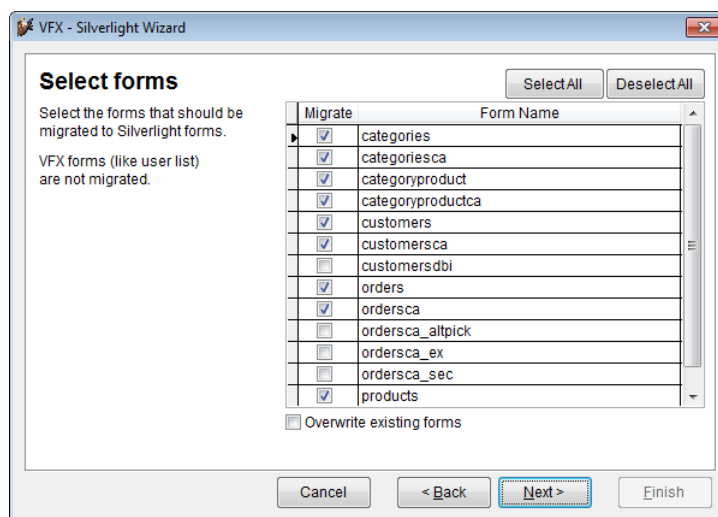
Im Datenmodell wird für jede Tabelle der Datenbank eine Klasse mit dem Namen der Tabelle angelegt.

Im Ordner VfxDataLayer.Web\VFPComDomainService der VFX für Silverlight Projektmappe wird ein VFP COM Server angelegt. Dieser COM Server wird erstellt, registriert und in die Projektmappe eingetragen.

Wenn der VFX für Silverlight Wizard auf Windows Vista oder einer neueren Windows Version ausgeführt wird und die Benutzerkontensteuerung aktiviert ist, muss VFP 9 explizit mit Administratorrechten gestartet werden, damit ein COM Server registriert werden kann.

Der Name des COM Servers wird <VFX für VFP Projektname>COMDomainService. Es wird also für jede VFX für Silverlight Projektmappe ein eigener VFP COM Server erstellt.

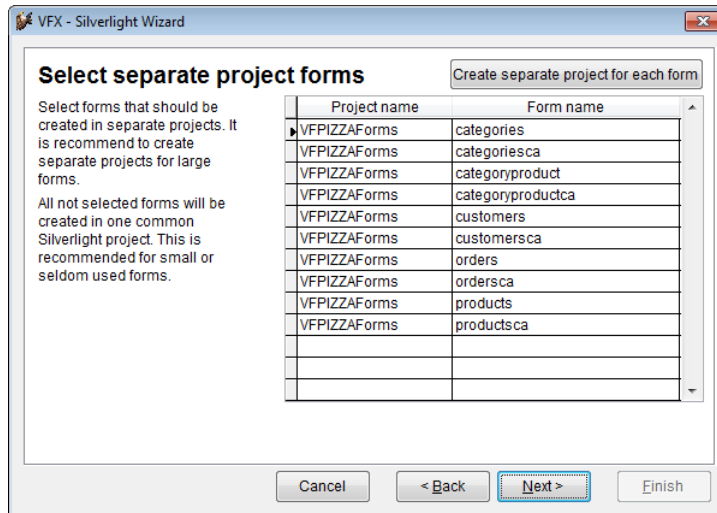
Auswahl von Formularen



In diesem Schritt erscheint eine Liste mit Formularen, die im VFX für VFP Projekt enthalten sind. Standardmäßig wird vorgeschlagen die Formulare nach Silverlight zu migrieren, die in das Projekt eingeschlossen sind. VFX Formulare, wie zum Beispiel die Benutzerverwaltung, werden nicht nach Silverlight migriert, weil diese Formulare bereits im Vorlageprojekt von VFX für Silverlight enthalten sind.

Falls in der Datenumgebung eines Formulars eine Datenquelle verwendet wird, die nicht über einen Primärschlüssel verfügt, erscheint im Protokoll des Assistenten eine Warnung. Das Formular kann nach Silverlight migriert werden. Das migrierte Formular kann ausgeführt werden, jedoch ist die Bearbeitung von Daten nicht möglich, solange der Datenquelle kein Primärschlüssel hinzugefügt wird.

Auswahl der Projekte für Formulare



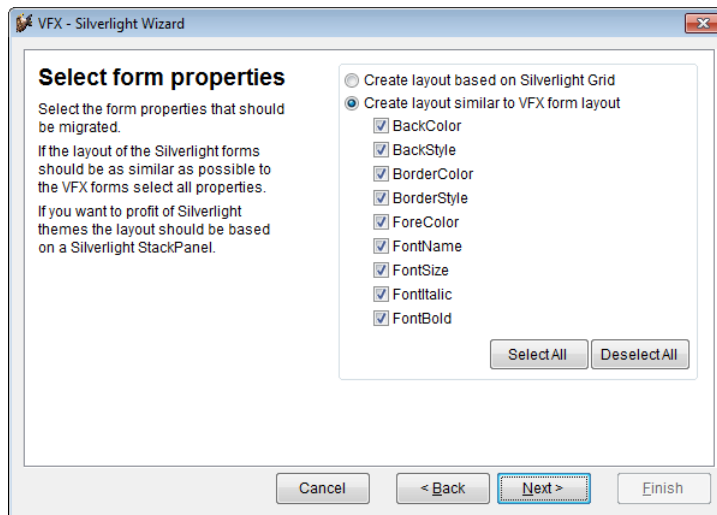
In diesem Schritt werden die Namen der Projekte angegeben, denen die Silverlight Formulare hinzugefügt werden sollen. Standardmäßig werden alle Silverlight Formulare dem Projekt <Projektname>Forms hinzugefügt. Es ist möglich eigene Projektnamen einzutragen. Es können mehrere oder alle Formulare einem Projekt hinzugefügt werden. Es kann aber auch für jedes Formular ein eigenes Projekt erstellt werden.

Bei der Erstellung der Silverlight Client-Anwendung wird aus jedem Projekt eine XAP-Datei generiert, die zur Laufzeit an den Client im Internet übertragen werden muss. Durch die Aufteilung in mehrere Projekte kann die Größe der XAP Dateien verkleinert werden. Die Ladezeit wird so verringert.

Formulare können beliebig in Projekte gruppiert werden. Es mag sinnvoll sein, umfangreiche Formulare in eigene Projekte zu platzieren. Mehrere kleine Formulare können in einem Projekt zusammengefasst werden.

Mit einem Klick auf die Schaltfläche „Create separate project for each form“ wird für jedes Formular ein Projektname generiert.

Auswahl von Formulareigenschaften



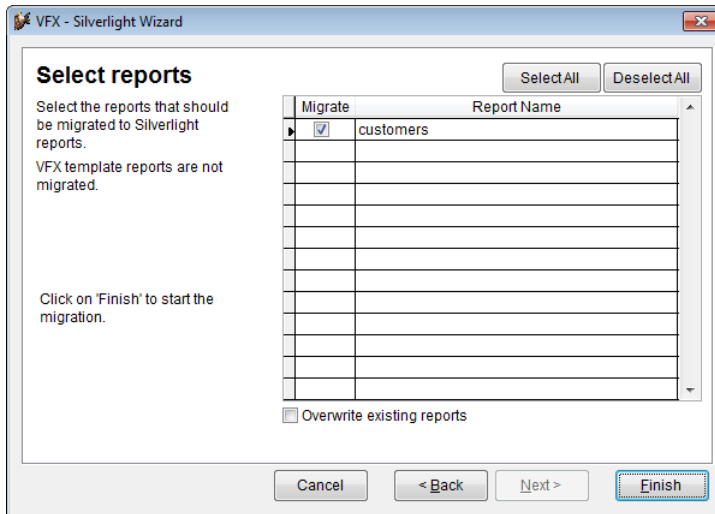
Standardmäßig werden Silverlight Formulare mit einem Layout generiert, das weitgehend dem Layout des VFX Formulars entspricht. Zu diesem Zweck werden alle möglichen Layouteigenschaften nach Silverlight migriert.

Wenn der Anwender die Möglichkeit erhalten soll Styles zu verwenden, sollten nicht alle Layouteigenschaften nach Silverlight migriert werden. Alle Eigenschaften, die hard-codiert in Silverlight Formularen enthalten sind, können zur Laufzeit nicht durch die Anwendung von Styles geändert werden.

Durch die Option „Create layout based on Silverlight StackPanel“ wird die Position von Steuerelementen nicht nach Silverlight migriert. Vielmehr werden die Steuerelemente im Silverlight-Formular in einem Silverlight Grid angeordnet. Ein Silverlight Grid ist vergleichbar mit einer Tabelle. In dieser Tabelle werden typischerweise in einer Spalte Bezeichnungen und in der nächsten Spalte Textboxen und andere Steuerelemente zur Dateneingabe

platziert. Der Vorteil dieser Art der Darstellung ist, dass Formulare gut skalierbar sind. Formulare lassen sich so zum Beispiel auch auf Mobilgeräten mit sehr kleinen Bildschirmen gut anzeigen.

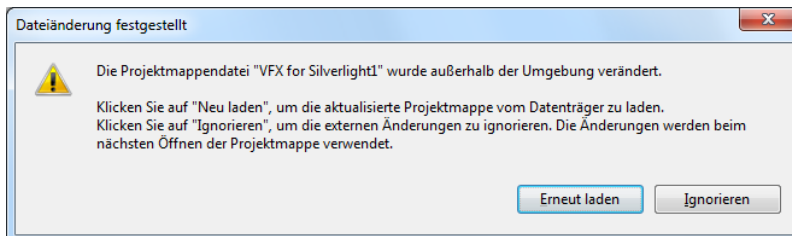
Auswahl von Berichten



In diesem Schritt werden alle im VFX für VFP Projekt enthaltenen Berichtsdateien angezeigt. Standardmäßig werden alle Berichte zu Silverlight Berichten migriert, ausgenommen VFX Berichte, wie zum Beispiel die Vorlagedateien für Grid-Berichte.

Damit VFP Berichte nach Silverlight migriert werden können, dürfen diese nicht über Cursor in der Datenumgebung verfügen. Berichte müssen auf den Datenquellen basieren, die das aufrufende Formular bereitstellt.

Wenn die Projektmappe in Visual Studio geöffnet ist, während der VFX – Silverlight Wizard ausgeführt wird, erscheint in Visual Studio die Frage, ob die Projektmappe neu geladen werden soll, weil eine andere Anwendung Änderungen an der Projektmappe vorgenommen hat. In diesem Fall muss die Projektmappe neu geladen werden.



Start der VFX für Silverlight Anwendung

Die migrierte VFX für Silverlight Anwendung kann in der Entwicklungsumgebung von Visual Studio durch einen Klick auf den grünen Pfeil in der Symbolleiste oder mit der Funktionstaste F5 ausgeführt werden.

Die Anwendung startet mit einem Begrüßungsbildschirm. Wenn die Anwendung zur Verwendung von mehr als einer Datenbank konfiguriert ist, erscheint der Mandantenauswahldialog. Anschließend erscheint der Anmeldedialog. Nach der erfolgreichen Benutzeranmeldung läuft die Anmeldung. Im oberen Bereich erscheint eine Symbolleiste und im linken Bereich erscheint der Öffnen-Dialog.

Die weitere Bedienung ist ähnlich zu Office Anwendungen, bzw. zu VFX für VFP Anwendungen.

Weiterbearbeitung

Die VFX für Silverlight Anwendung kann in Visual Studio weiterbearbeitet werden.

Es ist aber auch möglich in der VFX für VFP Anwendung Änderungen an Formularen vorzunehmen und diese Formulare anschließend mit dem VFX – Silverlight Wizard erneut zu migrieren.

Architektur

Silverlight Internet-Anwendungen bestehen aus mindestens zwei Projekten: einem Serverprojekt und einem Clientprojekt. Das Serverprojekt wird auf dem Server ausgeführt und enthält den Datenzugriff und Geschäftslogik.

Das Clientprojekt ist das eigentliche Silverlight-Projekt und wird auf dem Client-Rechner ausgeführt. Aus Optimierungsgründen ist das Clientprojekt in viele, kleine Teilprojekte aufgeteilt.

Serverprojekte

VFX für Silverlight Anwendungen haben zwei Serverprojekte.

Das Startprojekt der Projektmappe hat den Namen VFXforSilverlight.Web. Das Startprojekt wird auf dem Internet Server vom Internet Information Server ausgeführt. Dieses Projekt enthält die Startseite default.htm.

Die Internetadresse der Startseite wird vom Internetbenutzer in der Adresszeile des Browsers eingegeben, um die Anwendung zu starten. Die Startseite default.htm prüft, ob im Browser das Silverlight Plug-In installiert ist und lädt die Silverlight Startanwendung auf der Clientseite.

Das zweite Serverprojekt hat den Namen VfxDataLayer.Web. Dieses Projekt enthält die Datenmodell, über die der Datenzugriff erfolgt sowie die Domain Services, mit denen der Datenaustausch zwischen Clientprojekten und Serverprojekten erfolgt.

Der Datenzugriff erfolgt in VFX für Silverlight Anwendungen mit ADO.NET Entity Data Modellen. Diese Modelle können mit Visual Studio erstellt werden, wenn für die zu verwendende Datenbank ein ADO.NET Entity Data Model-Treiber zur Verfügung steht. Für Microsoft SQL Server wird so ein Treiber mit Visual Studio installiert.

Bei der Migration von VFX für VFP Projekten wird ein ADO.NET Entity Data Model für den Zugriff auf VFP Datenbanken vom VFX – Silverlight Wizard erstellt. Der Zugriff auf die Datenbank erfolgt hierbei mit einem VFP COM Server, der ebenfalls vom VFX – Silverlight Wizard generiert wird. Ein ADO.NET Entity Data Model-Treiber für VFP Datenbanken ist daher nicht erforderlich.

Ein Domain Service kann mit dem Assistenten von Visual Studio generiert werden. Bei der Migration einer VFX für VFP Anwendung wird ein Domain Service vom VFX – Silverlight Wizard erstellt.

Clientprojekte

In einer Silverlight Anwendung wird ein Silverlight Projekt zu einer Datei mit der Namensendung .xap kompiliert. Genauer gesagt, werden die DLLs, die das Kompilat enthalten, in eine Archivdatei verpackt. Diese Archivdatei enthält die Namensendung .xap. XAP-Dateien können mit dem Windows-Explorer in ZIP Dateien umbenannt werden und man kann so die enthaltenen DLL-Dateien sehen.

Beim Laden einer Silverlight Anwendung über das Internet wird eine XAP-Datei von der Serverseite an den Client übertragen und dort vom Silverlight Plug-In ausgeführt.

Um die Wartezeit beim Laden einer Silverlight Anwendung zu verkürzen, sollte eine XAP-Datei klein sein.

VFX für Silverlight wird dem durch ein Modularisierungskonzept gerecht. Beim Start der Anwendung wird VfxLoader.xap geladen.

Klassenhierarchie

Die in VFX für Silverlight implementierte Klassenhierarchie ist ähnlich zu der Hierarchie, die in VFX für VFP implementiert wurde. Die Funktionalität von VFX für Silverlight ist in Klassendateien mit dem Namenszusatz "Base" enthalten. Von diesen Klassendateien gibt es 1:1 Ableitungen in Klassendateien ohne den Namenszusatz "Base". Diesen Klassendateien können eigene Erweiterungen hinzugefügt werden.

Anwendungsobjekt

Das Anwendungsobjekt ist im Projekt VfxApplicationBase in der Programmdatei Controls/VfxAppObjectBase.cs definiert. Eine 1:1 Ableitung, in der individuelle Anpassungen für das jeweilige Projekt gemacht werden können, befindet sich im Projekt VfxApplication in der Programmdatei Controls/VfxAppObjectBase.cs.

Das Anwendungsobjekt stellt Eigenschaften und Methoden global zur Verfügung. In den Eigenschaften des Anwendungsobjekts können Einstellungen gemacht werden, die das Verhalten der Anwendung betreffen.

Eigenschaften

```
AllowRelogon = true;
```

Mit dieser Eigenschaft kann eingestellt werden, ob das Symbol „Anmelden“ in der Symbolleiste angezeigt werden soll. Der Wert dieser Eigenschaft wird nur berücksichtigt, wenn der Wert der Eigenschaft LoginBehavior auf LoginAtStartup oder auf OptionalLogin steht.

```
PathToApplicationLogo = "/VfxLoader;component/Icons/Icon128.png";
```

In dieser Eigenschaft wird die Datei angegeben, in der sich das Logo der Anwendung befindet. Standardmäßig befindet sich das Logo im Projekt VfxLoader in der Datei Icons/Icon128.xaml. Zur Laufzeit der Anwendung wird das Logo am linken Rand der Symbolleiste angezeigt.

```
VfxMenuStyle = VfxMenuStyle.XPStyle;
```

Der Wert dieser Eigenschaft beschreibt den Stil, in dem die Bedienung der Anwendung erfolgt. Zurzeit steht nur der Stil XPStyle zur Verfügung. Hierbei befindet sich am oberen Rand der Anwendung eine Symbolleiste zur Steuerung von Formularen. Am linken Rand befindet sich ein Öffnen-Dialog, über den Formulare geöffnet werden können.

```
InsertDateFieldName = "ins_date";
```

Name des Feldes in einer Tabelle, in dem das Datum der Neuanlage eines Datensatzes gespeichert wird. Wenn sich in einer Tabelle ein Feld mit dem angegebenen Namen befindet, wird der Wert beim ersten Speichern automatisch gefüllt. Der Typ des Feldes kann Date oder DateTime sein. Wenn der Typ des Feldes DateTime ist, wird beim Speichern ein Zeitstempel eingefügt.

```
EditDateFieldName = "edt_date";
```

Name des Feldes in einer Tabelle, in dem das Datum der letzten Änderung eines Datensatzes gespeichert wird. Wenn sich in einer Tabelle ein Feld mit dem angegebenen Namen befindet, wird der Wert bei jedem Speichervorgang aktualisiert. Der Typ des Feldes kann Date oder DateTime sein. Wenn der Typ des Feldes DateTime ist, wird beim Speichern ein Zeitstempel eingefügt.

```
InsertUserFieldName = "ins_usr";
```

Name des Feldes in einer Tabelle, in dem der Name des Anwenders gespeichert wird, der den Datensatz neu angelegt hat. Wenn sich in einer Tabelle ein Feld mit dem angegebenen Namen befindet, wird der Wert beim ersten Speichern automatisch gefüllt. Das Feld muss vom Typ Zeichen mit der Länge 32 sein.

```
EditUserFieldName = "edt_usr";
```

Name des Feldes in einer Tabelle, in dem der Name des Anwenders gespeichert wird, der den Datensatz zuletzt bearbeitet hat. Wenn sich in einer Tabelle ein Feld mit dem angegebenen Namen befindet, wird der Wert bei jedem Speichern automatisch gefüllt. Das Feld muss vom Typ Zeichen mit der Länge 32 sein.

```
InsertTimeFieldName = "ins_time";
```

Name des Feldes in einer Tabelle, in dem die Uhrzeit der Neuanlage eines Datensatzes gespeichert wird. Wenn sich in einer Tabelle ein Feld mit dem angegebenen Namen befindet, wird der Wert beim ersten Speichern automatisch gefüllt. Der Typ des Feldes muss C(8) sein.

```
EditTimeFieldName = "edt_time";
```

Name des Feldes in einer Tabelle, in dem die Uhrzeit der letzten Änderung eines Datensatzes gespeichert wird. Wenn sich in einer Tabelle ein Feld mit dem angegebenen Namen befindet, wird der Wert beim jedem Speichern automatisch gefüllt. Der Typ des Feldes muss C(8) sein.

```
LoginBehavior = LoginBehavior.OptionalLogin;
```

Mit dieser Eigenschaft wird angegeben, ob eine Benutzeranmeldung erforderlich ist, optional möglich oder zwingend erforderlich ist. Der Wert dieser Eigenschaft bestimmt insbesondere das Verhalten beim Start der Anwendung.

OptionalLogin (Standardwert)

Die Anwendung startet ohne Benutzeranmeldung. Es wird die Standarddatenbank verwendet. Dies ist die erste Datenbank, die in der Datei Web.config gefunden wird. Wenn in der Datei Web.config keine Datenbank eingetragen ist, wird die erste Datenbank verwendet, die in der Datei Config.vfx eingetragen ist. Während die Anwendung ausgeführt wird, ist eine Benutzeranmeldung durch einen Klick auf die Schaltfläche „Anmelden“ in der Symbolleiste möglich. Diese Einstellung eignet sich für Anwendungen, in denen ein Teil der Funktionalität öffentlich zugänglich sein soll, während andere Teile der Anwendung durch eine Benutzeranmeldung geschützt sind. So sind zum Beispiel Internetanwendungen möglich, auf die ohne Benutzeranmeldung nur lesend zugegriffen werden kann, während ein Schreibzugriff eine Benutzeranmeldung erfordert.

LoginAtStartup

Die Anwendung startet mit Benutzeranmeldung. Dieses Verhalten entspricht dem Verhalten von VFX für VFP Anwendungen. Wenn in den Dateien Web.config und Config.vfx mehr als eine Datenbank gefunden wird, erscheint vor der Benutzeranmeldung der Mandantenauswahldialog. Ohne Benutzeranmeldung kann die Anwendung nicht ausgeführt werden. Diese Einstellung empfiehlt sich für Internet- und Intranetanwendungen, in denen keine Funktionalität öffentlich, bzw. ohne Anmeldung, verfügbar sein soll.

NoLogin

Die Anwendung startet ohne Benutzeranmeldung. Es wird die Standarddatenbank verwendet. Dies ist die erste Datenbank, die in der Datei Web.config gefunden wird. Wenn in der Datei Web.config keine Datenbank eingetragen ist, wird die erste Datenbank verwendet, die in der Datei Config.vfx eingetragen ist. Eine Benutzeranmeldung ist nicht möglich. Diese Einstellung ist empfehlenswert, wenn eine Internetanwendung öffentlich verfügbar sein soll oder in nicht öffentlich zugänglichen Intranetanwendungen, in denen keine Benutzeranmeldung verwendet werden soll.

`ApplicationName = "VFXforSilverlight";`

In dieser Eigenschaft wird der Name der Anwendung angegeben. Der Name erscheint in der Symbolleiste rechts vom Icon der Anwendung.

`LangID = "ENG";`

Wenn nicht mit Lokalisierung zur Laufzeit gearbeitet wird, wird in dieser Eigenschaft das Kürzel der verwendeten Sprache angegeben. Wenn mit Lokalisierung zur Laufzeit gearbeitet wird, wird in dieser Eigenschaft das Kürzel der Sprache angegeben, mit der die Anwendung gestartet wird. Der Standardwert ist Englisch (ENG).

`AllowUserCustomization = true;`

Wenn der Wert dieser Eigenschaft auf true gestellt ist, können angemeldete Benutzer individuelle Einstellungen zur Bedienung der Anwendung vornehmen. Wenn kein Benutzer angemeldet ist, können keine individuellen Einstellungen vorgenommen werden. Wenn der Wert dieser Eigenschaft auf false gestellt ist, können auch angemeldete Benutzer keine individuellen Einstellungen vornehmen.

`AutoEdit = GlobalSettings.AlwaysEnabled;`

Mit dieser Eigenschaft kann eingestellt werden, ob Steuerelemente auf allen Formularen standardmäßig enabled oder disabled sind.

AlwaysEnabled

Mit dieser Einstellung sind Steuerelemente auf allen Formularen standardmäßig enabled. Der Benutzer kann den Fokus auf ein Steuerelement setzen und sofort mit der Bearbeitung beginnen. Bei der ersten Änderung wird das Formular vom Ansichtsmodus automatisch in den Bearbeitungsmodus umgeschaltet.

AlwaysDisabled

Mit dieser Einstellung sind Steuerelemente auf allen Formularen standardmäßig disabled. Eine Bearbeitung ist nur möglich, nachdem der Benutzer in der Symbolleiste auf „Bearbeiten“ geklickt hat und so das Formular vom Ansichtsmodus in den Bearbeitungsmodus umschaltet hat.

UseObjectSettings

Mit dieser Einstellung gilt die Einstellung des jeweiligen Formulars.

```
MainForm = "MainPage";
```

Das in dieser Eigenschaft angegebene Formular wird beim Start der Anwendung automatisch geöffnet. Wenn eine Benutzeranmeldung erforderlich ist, wird das Formular nach der Benutzeranmeldung geöffnet.

```
RuntimeLocalization = true;
```

Wenn der Wert dieser Eigenschaft auf true gestellt ist, wird die Anwendung mit Lokalisierung zur Laufzeit ausgeführt. Dabei werden alle Anzeigetexte zur Laufzeit aus der Tabelle Vfxmsg gelesen und die der ausgewählten Sprache angezeigt. Wenn der Wert dieser Eigenschaft auf false gestellt ist, werden zur Laufzeit keine Texte aus der Tabelle Vfxmsg gelesen.

```
ShowIntroForm = true;
```

Mit dieser Eigenschaft wird eingestellt, ob der Begrüßungsbildschirm angezeigt werden soll.

```
DisableFormResize = false;
```

Wenn der Wert dieser Eigenschaft auf true gestellt ist, können Benutzer die Größe von Formularen zur Laufzeit nicht ändern. Wenn der Wert false ist, kann die Größe von Formularen zur Laufzeit geändert werden. Die vom Benutzer eingestellte Größe wird beim Schließen des Formulars in der Tabelle VfxResources gespeichert und beim nächsten Öffnen wiederhergestellt.

```
Century = false;
```

Mit dieser Eigenschaft kann eingestellt werden, ob in Datumsfeldern das Jahrhundert angezeigt werden soll. Wenn der Wert dieser Eigenschaft false ist, wird das Jahrhundert nicht angezeigt, die Jahreszahl wird zweistellig angezeigt. Wenn der Wert dieser Eigenschaft true ist, wird das Jahrhundert angezeigt, die Jahreszahl wird vierstellig angezeigt.

```
RequiredFieldInitStyle = "Required";
```

Mit dieser Eigenschaft wird der Stil eingestellt, mit dem Pflichtfelder formatiert werden. Der Standardwert ist „Required“. Es können weitere, eigene Stile implementiert werden.

```
RequiredFieldFailureStyle = "RequiredInvalid";
```

Mit dieser Eigenschaft wird der Stil eingestellt, mit dem Pflichtfelder formatiert werden, wenn beim Speichern noch keine Eingabe erfolgte. Der Standardwert ist „RequiredInvalid“. Es können weitere, eigene Stile implementiert werden.

```
HideWhenEmpty = GlobalSettings.UseObjectSettings;
```

Mit dieser Eigenschaft kann eingestellt werden, ob Steuerelemente versteckt werden soll, wenn in einer Tabelle keine Datensätze gespeichert sind. In diesem Fall wird auf dem Formular ein Hinweis angezeigt. Mit einem Klick auf diesen Hinweis kann der Benutzer das Formular in den Neuanlagemodus umschalten.

```
PasswordValidityDays = 0;
```

Mit dieser Eigenschaft wird die Dauer der Gültigkeit von Kennwörtern eingestellt. Wenn die Gültigkeit eines Kennworts abgelaufen ist, wird der Benutzer bei der Anmeldung aufgefordert, ein neues Kennwort einzugeben. Ohne die Eingabe eines neuen Kennworts ist die Anmeldung nicht mehr möglich. Wenn der Wert dieser Eigenschaft 0 ist, bleiben Kennwörter unbefristet gültig.

```
PasswordLength = 0;
```

Mit dieser Eigenschaft wird die minimale Länge von Kennwörtern angegeben. Wenn der Wert dieser Eigenschaft 0 ist, ist die Eingabe eines Kennworts nicht erforderlich.

```
PasswordStrengthLevel = PasswordStrengthLevel.Weak;
```

Mit dieser Eigenschaft wird die für Kennwörter erforderliche Sicherheit eingestellt.

```
EnforcePasswordHistoryCount = 0;
```

Mit dieser Eigenschaft kann eingestellt werden, wie viele Kennwörter historisch gespeichert werden sollen. Kennwörter, die sich in der Historie befinden, können nicht als neues Kennwort verwendet werden. Der Standardwert ist 0.

```
ShowFilterActivatedInFormCaption = GlobalSettings.UseObjectSettings;
```

Mit dieser Eigenschaft kann eingestellt werden, ob in der Titelzeile eines Formulars angezeigt werden soll, ob ein Filter gesetzt ist.

Formulare

Formulare haben Eigenschaften, mit denen das Verhalten des Formulars gesteuert werden kann.

Eigenschaften

Eigenschaften von Formularen sind in der Datei mit der Namenserweiterung `.xaml.cs` des Formulars angegeben.

```
AskToSave = true;
```

Wenn der Wert dieser Eigenschaft auf `true` gestellt ist, wird der Anwender gefragt ob Änderungen gespeichert werden sollen, wenn versucht wird das Formular zu schließen und nicht gespeicherte Änderungen vorhanden sind. Wenn der Wert dieser Eigenschaft auf `true` gestellt ist, werden Änderungen ohne Rückfrage gespeichert.

```
AutoEdit = true;
```

Wenn der Wert dieser Eigenschaft auf `true` gestellt ist, sind die Steuerelemente eines Formulars enabled. Die Bearbeitung der Daten kann unmittelbar begonnen werden.

Wenn der Wert dieser Eigenschaft auf `false` gestellt ist, sind die Steuerelemente eines Formulars disabled. Wenn der Anwender die Daten bearbeiten will, muss er zunächst auf die Schaltfläche „Bearbeiten“ klicken um das Formular in den Bearbeitungsmodus umzuschalten. Dabei wird der Status der Steuerelemente auf `enabled` gesetzt.

```
AutoResizeControl = true;
```

Wenn der Wert dieser Eigenschaft auf `true` gestellt ist, kann die Größe des Formulars zur Laufzeit vom Anwender geändert werden.

```
InitialHeight = 295;
```

In dieser Eigenschaft wird die Höhe des Formulars angegeben.

```
InitialWidth = 538;
```

In dieser Eigenschaft wird die Breite des Formulars angegeben.

```
MinButton = true;
```

Wenn der Wert dieser Eigenschaft auf `true` gestellt ist, wird in der Titelzeile des Formulars eine Schaltfläche zum Minimieren des Formulars angezeigt.

```
MultiInstance = true;
```

Wenn der Wert dieser Eigenschaft auf `true` gestellt ist, kann das Formular mehrmals gestartet werden.

```
ReportName = "CustomersTemplate.xaml";
```

In dieser Eigenschaft kann der Name einer Berichtsvorlage angegeben werden. Wenn der Wert dieser Eigenschaft leer ist, wird zur Laufzeit ein Bericht basierend auf den Spalten des DataGrid des Formulars generiert.

Ausführung von VFP Code

GUINEU

In allen Silverlight Client Projekten kann VFP Code mithilfe von GUINEU ausgeführt werden. GUINEU ist eine Laufzeitumgebung zur Ausführung von mit VFP kompilierten Programmdateien

Die GUINEU Laufzeitumgebung befindet sich in der Datei vfx.guineu.runtime.dll. Diese DLL ist in das Projekt VfxGuineuRuntime. In diesem Projekt befindet sich außerdem die Klasse VfxGuineu.cs mit deren Hilfe die Funktionalität von GUINEU verwendet wird.

Der Programmcode muss mit der Entwicklungsumgebung von VFP erstellt werden. Die kompilierte FXP Datei muss in das jeweilige Silverlight Projekt eingeschlossen werden.

Es ist empfehlenswert auch die PRG Datei dem Silverlight Projekt hinzuzufügen. Durch einen Doppelklick auf die PRG Datei im Projektmappen-Explorer kann VFP gestartet und die PRG Datei bearbeitet werden.

Um GUINEU in einer C# Klasse verwenden zu können, muss ein Feld vom Typ VfxGuineu hinzugefügt werden. Der Instanziierung wird als Parameter der Name der kompilierten VFP Programmdatei mitgegeben.

```
private readonly VfxGuineu _fox = new VfxGuineu("Orders.FXP");
```

Der Aufruf von Funktionen in der VFP Programmdatei ist mit einer Zeile C# Code möglich.

```
<Rückgabewert> = _fox.Do("<Name der Funktion>", <optionale Parameter>, ...);
```

Die Auswertung des Rückgabewerts ist optional. Es können so viele Parameter übergeben werden, wie als Parameter von der VFP Funktion akzeptiert werden. Parameter können von beliebigem Typ sein. Es ist auch möglich Objekte aus der Silverlight Benutzeroberfläche als Parameter zu übergeben. Die Eigenschaften von Objekten stehen in der VFP Funktion zur Verfügung und können gelesen und geschrieben werden.

```
private void xpgfPageFramePage1txtShiptoname_GotFocus(object sender,
System.Windows.RoutedEventArgs e)
{
    _fox.Do("ShipToName_GotFocus", txtTooltip);
}
```

```
private void xpgfPageFramePage1txtShiptoname_TextChanged(object sender,
System.Windows.Controls.TextChangedEventArgs e)
{
    OnEdit();
    _fox.Do("Validate", xpgfPageFramePage1txtShiptoname,
xpgfPageFramePage1edtShiptoaddress, chkSpeichern);
}
```

```
private void xpgfPageFramePage1txtOrderdate_LostFocus(object sender,
System.Windows.RoutedEventArgs e)
{
    OnEdit();
    var orderDate =
((orders)ViewModel.WorkAliasCollection.CurrentItem).orderdate;
    txtDelivery.Text = _fox.Do("Lieferdatum", orderDate);
}
```

Action Button

In der Silverlight Benutzeroberfläche steht die Steuerelementklasse VfxActionButton zur Verfügung. Diese Klasse ist von der Silverlight Basisklasse Button abgeleitet und verfügt über zusätzliche Eigenschaften, mit deren Hilfe eine Methode in einem VFP COM Server auf der Serverseite ausgeführt werden kann.

```
<VfxControl:VfxActionButton
    Content="Proper"
    Height="23"
    HorizontalAlignment="Left"
    Margin="443,209,0,0"
    Name="btnAction"
    ProcedureName="proper"
    ProcedureParameters="{Binding Path=Customers.CurrentItem.customername,
Mode=OneWay}"
    TooltipService.ToolTip="Use VFP function PROPER() to format customer name"
    OperationResultLoaded="btnAction_OperationResultLoaded"
    VerticalAlignment="Top"
    Width="75" />
```

ProcedureName

Der Eigenschaft ProcedureName wird der Name der Methode des COM Servers zugewiesen. Bei einem Klick auf den Button wird die hier angegebene Methode ausgeführt.

Es kann sich bei dem hier angegebenen Namen um eine VFP Funktion, wie hier im Beispiel PROPER(), den Namen einer Methode des COM Servers oder um den Namen einer Programmdatei handeln, die vom COM Server zur Laufzeit geladen wird.

Der Vorteil in der Verwendung von Programmdateien liegt darin, dass Änderungen im Programmcode gemacht werden können, ohne dass der COM Server aktualisiert oder neu registriert werden muss. Durch die Verwendung von externen Programmdateien können auch neue VFP Funktionen aus einer Silverlight Anwendung genutzt werden, ohne dass der COM Server neu erstellt und neu registriert werden muss.

ProcedureParameters

In dieser Eigenschaft können optionale Parameter der Methode angegeben werden.

OperationResultLoaded

Der Aufruf des COM Servers erfolgt asynchron. In der Eigenschaft OperationResultLoaded wird der Name der Methode der aktuellen C# Klasse angegeben, die aufgerufen wird, wenn die Ausführung der COM Server Methode beendet ist.

Lokalisierung

Der VFX – Silverlight Wizard übernimmt die Lokalisierungseinstellungen von der migrierten VFX für VFP Anwendung.

Wenn nicht mit Lokalisierung zur Laufzeit gearbeitet wird, werden dennoch alle von VFX verwendeten Texte, zum Beispiel Bezeichnungen in Dialogen und Tooltips, zur Laufzeit aus der Tabelle vfxmsg gelesen. Im Gegensatz zu VFX für VFP werden in VFX für Silverlight keine Includedateien mit Konstanten verwendet.

Die Lokalisierungseinstellungen werden mit Eigenschaften des Anwendungsobjekts in der Datei VfxApplication – Controls\ VfxAppObject.cs gemacht.

```
RuntimeLocalization = true;
```

Wenn der Wert dieser Eigenschaft auf true gesetzt ist, läuft die Anwendung mit Lokalisierung zur Laufzeit.

```
LangID = "GER";
```

Wenn eine Anwendung ohne Lokalisierung zur Laufzeit arbeitet, wird mit dieser Eigenschaft die verwendete Sprache eingestellt. Wenn der Wert der Eigenschaft RuntimeLocalization auf true gesetzt ist, gibt der Wert dieser Eigenschaft die Sprache an, mit der die Anwendung gestartet wird.

Der Wert der Eigenschaft LangID enthält die Abkürzung der verwendeten Sprache. Für jede verfügbare Sprache gibt es eine Spalte in der Tabelle Vfxmsg.

Sprache	Abkürzung
Albanisch	ALB
Bulgarisch	BUL
Chinesisch (traditionell)	CHT
Chinesisch (vereinfacht)	CHS
Deutsch (Deutschland)	GER
Deutsch (Schweiz)	CHD
Englisch (UK)	ENG
Estnisch	EST
Finnisch	FIN
Französisch	FRE
Griechisch	GRE
Italienisch	ITA
Japanisch	JPN
Niederländisch	NL
Polnisch	PL
Portugiesisch	POR
Rumänisch	RO
Russisch	RU
Slovakisch	SVK
Spanisch	ESP
Tschechisch	CZE
Türkisch	TR

Es stehen nicht alle Texte in allen Sprachen zur Verfügung.

Wenn mit Lokalisierung zur Laufzeit gearbeitet wird, kann die Liste der verfügbaren Sprachen in der Tabelle VfxLanguage eingestellt werden. In dieser Tabelle gibt es für jede Sprache einen Datensatz. Wenn in der Spalte IsActive True steht, steht diese Sprache in der Anwendung zur Verfügung.

Die Auswahl der Sprache kann in einer Combobox im Anmeldedialog oder in einer Combobox in der Symbolleiste vorgenommen werden. Die eingestellte Sprache wird je Anwender gespeichert und bei der Benutzeranmeldung nach der Eingabe des Benutzernamens wiederhergestellt.

Beim Wechsel der Sprache werden auch die Regionaleinstellungen gesetzt. Dazu gehören Dezimaltrennzeichen, Datumsformat und andere Einstellungen.

Wenn eine Anwendung ohne Lokalisierung zur Laufzeit arbeitet, sind die Comboboxen zur Auswahl der Sprache nicht sichtbar.

Lokalisierung von Formularen

Alle Formulare implementieren das Interface IVfxLocalization. Wenn das Formularereignis Loaded ausgeführt wird, wird die Methode InitializeLocalizationManager(string language) aufgerufen. Mit dieser Methode wird das Objekt VfxLocalizationManager instanziiert. Dieses Objekt durchsucht das Formular nach lokalisierbaren Objekten. Lokalisierbare Objekte verfügen über die Eigenschaft MessageId oder ToolTipMessageID.

Verwendung von Bilddateien

In VFX für Silverlight Anwendungen können Bilddateien der Formate PNG und JPG verwendet werden.

Datei-Manager

VFX für Silverlight enthält Funktionen, die den Upload und den Download von Dateien ermöglichen.

Optimierung des Datenzugriffs

Im Datenzugriff steckt ein erhebliches Optimierungspotenzial. Während in Netzwerkanwendungen die vom Server zum Client übertragene Datenmenge nur von untergeordneter Bedeutung ist, ist in Internetanwendungen die Übertragung möglichst kleiner Datenmengen von großer Wichtigkeit. Die Übertragungsgeschwindigkeit ist im Internet in der Regel um ein Vielfaches langsamer, als in Netzwerkanwendungen.

Während in Netzwerkanwendungen die Benutzung einer Tabelle und die Präsentation aller Datensätze in einem Formular für den Anwender in vielen Fällen möglich ist, ist dies in Internetanwendungen nur dann möglich, wenn sich in einer Tabelle sicher nur wenige Datensätze befinden und die Strukturgröße eines Datensatzes nicht besonders groß ist.

Für Internetanwendungen ist es in der Regel erforderlich, dass Formulare zunächst ohne Daten geladen werden und der Benutzer möglichst genau eingeben muss, welche Daten er sehen will.

Weitere Informationen

Aktuelle Informationen zu VFX für Silverlight sind auf der Webseite <http://www.visualextend.de> zu finden.

Forum, Newsgroup

Unterstützung zu VFX für Silverlight ist in der Newsgroup <news://news.dfpug.net> erhältlich. Auf die Newsgroup kann auch über die Startseite von VFX für Silverlight zugegriffen werden.

Technische Dokumentation

In VFX für Silverlight ist eine technische Dokumentation enthalten. Aus der technischen Dokumentation ist die Vererbung zwischen den einzelnen Klassen ersichtlich. Es sind alle Klassen, Eigenschaften und Methoden beschrieben.