

Visual Studio Tools for Office (VSTO)



Olaf Lischke: Visual Studio Tools for Office – www.lischke-edv.de

Inhalt

- Visual Studio Tools for Office
 - Überblick
 - Host Controls
 - Add-Ins
- Entwicklung mit VSTO
 - Anwendungs- vs. Dokumentkontext
 - Add-Ins vs. Vorlagen vs. Dokumente
 - OpenXML Dateiformat
 - Deployment
- Die Office-Objektmodelle
 - Excel
 - Word
 - Outlook
 - Powerpoint
- Office GUI
 - Ribbon Bars
 - Action/Task Panes
 - SmartTags



Olaf Lischke: Visual Studio Tools for Office – www.lischke-edv.de

VSTO Möglichkeiten

Anwendung	VSTO
Word 2007/2010	Anpassung für Dokumente/Vorlagen Dokumenterweiterungen mit ActionPanes u. SmartTags Anpassbarkeit der RibbonBar Ereignisbehandlung Datenbindung von Steuerelementen Add-Ins auf Anwendungsebene
Excel 2007/2010	Anpassung für Dokumente/Vorlagen Dokumenterweiterungen mit ActionPanes u. SmartTags Anpassbarkeit der RibbonBar Ereignisbehandlung Datenbindung von Steuerelementen Add-Ins auf Anwendungsebene
Outlook 2007/2010	Add-Ins auf Anwendungsebene Anpassbarkeit der RibbonBar Formular-Designer



Olaf Lischke: Visual Studio Tools for Office – www.lischke-edv.de

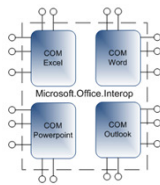
Dokumentebene vs. Anwendungsebene

- **Dokumentebene:**
 - Funktionen stehen nur innerhalb eines Dokuments/einer Vorlage zur Verfügung
 - Dokument/Vorlage muss vom Benutzer verwendet werden, um Zugriff auf die Funktionen zu erhalten
 - **Einsatzbereiche:** Vorlagen und Dokumente für besondere Zwecke, z.B. Auswertungen, Sonderschreiben, Verträge...
- **Anwendungsebene:**
 - Funktionen stehen der gesamten Anwendung jederzeit zur Verfügung
 - Add-In-Konzept
 - **Einsatzbereiche:** Spezielle Funktionen für einzelne Abteilungen/gesamte Organisation, z.B. Dokument-Handling, Corporate Identity,...



Olaf Lischke: Visual Studio Tools for Office – www.lischke-edv.de

Microsoft.Office.Interop



- Wrapperklassen für Office-Bibliotheken
- Viele aus VBA bekannt(?)
- Teils eigene Objekttypen:
 - z.B. Excel: `NamedRange`, `ListObject`
- Zugriff über Namespaces:
 - z.B. `Microsoft.Office.Interop.Excel`
- Zusätzlich:
 - `Microsoft.Office.Tools` für den Zugriff auf allgemeine Office-Elemente (Ribbons, Action Panes, SmartTags etc.)
 - `Microsoft.Office.Core` enthält z.B. Enumerationen



Olaf Lischke: Visual Studio Tools for Office – www.lischke-edv.de

Excel-Objektmodell

- `Microsoft.Office.Interop.Excel.Application`
 - Excel-Anwendung
 - Globale Events
 - Datei-Handling
- `Microsoft.Office.Interop.Excel.Workbook`
 - Excel-Arbeitsmappe
 - Mappen-Events
 - Blatt-Handling
- `Microsoft.Office.Interop.Excel.Worksheet`
 - Excel-Tabelle
 - Tabellen-Events
- `Microsoft.Office.Interop.Excel.Range`
 - Excel-Zelle(n)



Olaf Lischke: Visual Studio Tools for Office – www.lischke-edv.de

Excel Technologien

Technologie	Programmierung auf Dokumentebene	Programmierung auf Anwendungsebene	Benutzerdef. Tabellenblatt-funktionen
VSTO	X	X	
VBA	X	X	X
„klassisches“ Add-In (*.xla)		X	X



Olaf Lischke: Visual Studio Tools for Office – www.lischke-edv.de

Word-Objektmodell

- `Microsoft.Office.Interop.Word.Application`
 - Word-Anwendung
 - Globale Events
 - Dokument-Handling
- `Microsoft.Office.Interop.Word.Document`
 - Word-Dokument
 - Dokument-Events
- `Microsoft.Office.Interop.Word.Selection`
 - Auswahl (blinkender Cursor/Markierung)
- `Microsoft.Office.Interop.Word.Range`
 - Beliebiger Dokumentbereich, nicht sichtbar (i. Ggs. zur Auswahl)
- `Microsoft.Office.Interop.Word.Bookmark`
 - Word-Textmarke



Olaf Lischke: Visual Studio Tools for Office – www.lischke-edv.de

Outlook-Objektmodell

- `Microsoft.Office.Interop.Outlook.Application`
 - Outlook-Anwendung
 - Globale Events
 - CreateItem
- `Microsoft.Office.Interop.Outlook.Explorer`
 - Listenfenster in Outlook (Posteingang, Kalender, Kontakte...)
- `Microsoft.Office.Interop.Outlook.Inspector`
 - Fensterdarstellung eines einzelnen Elements (Mail, Termin...)
- `Microsoft.Office.Interop.Outlook.MAPIFolder`
 - Ordner des Postfachs/Öffentliche Ordner
- `Microsoft.Office.Interop.Outlook.[...]Item`
 - Outlook-Elemente: MailItem, AppointmentItem, ContactItem, TaskItem



Olaf Lischke: Visual Studio Tools for Office – www.lischke-edv.de

Powerpoint-Objektmodell

- `Microsoft.Office.Interop.Powerpoint.Application`
 - Powerpoint-Anwendung
 - Globale Events
 - Dokument-Handling
- `Microsoft.Office.Interop.Powerpoint.Presentation`
 - Powerpoint-Präsentation
 - Präsentations-Events
- `Microsoft.Office.Interop.Powerpoint.Slide`
 - -Powerpoint-Folie
- `Microsoft.Office.Interop.Powerpoint.Shape`
 - Powerpoint-Objekt (Titel, Text, Grafik...)



Olaf Lischke: Visual Studio Tools for Office – www.lischke-edv.de

SmartTags .NET 3.5

- SmartTag kreieren:


```
SmartTag = New SmartTag(URN, Caption)
```
- Schlüsselwörter festlegen:


```
SmartTag.Expressions.Add(...)
```

 - Sehr nützlich: Reguläre Ausdrücke!
- Aktionen festlegen:


```
Aktion = new Action(...);
SmartTag.Actions = new Action[] { Aktion };
```
- SmartTag hinzufügen: `this.VstoSmartTags.Add(SmartTag)`
- EventHandler anfügen:


```
Aktion.Click += new
    ActionClickEventHandler(Aktion_Click)
```
- EventHandler füllen



Olaf Lischke: Visual Studio Tools for Office – www.lischke-edv.de

SmartTags .NET 4.0

- SmartTag kreieren:


```
SmartTag = Globals.Factory.CreateSmartTag(URN, Caption)
```
- Schlüsselwörter festlegen:


```
SmartTag.Expressions.Add(...)
```

 - Sehr nützlich: Reguläre Ausdrücke!
- Aktionen festlegen:


```
Aktion = Globals.Factory.CreateAction(...);
SmartTag.Actions = new Action[] { Aktion };
```
- SmartTag hinzufügen: `this.VstoSmartTags.Add(SmartTag)`
- EventHandler anfügen:


```
Aktion.Click += new
    ActionClickEventHandler(Aktion_Click)
```
- EventHandler füllen



Olaf Lischke: Visual Studio Tools for Office – www.lischke-edv.de

OpenXML - Grundidee

- XML-basiertes Dokumentformat
 - Komplette dokumentiert
 - Strukturierter Aufbau
 - Kompaktes Format
- [Jede Menge weiterer Buzzwords...]
- **Für Entwickler:**
Dokumenterstellung/-zugriff ohne die dazugehörige Anwendung möglich!



Olaf Lischke: Visual Studio Tools for Office – www.lischke-edv.de

OpenXML – Grundlegende Struktur .docX

- **Root-Ebene:** [Content_Types].xml
 - Zuordnung Dateitypen zu Dateiendungen
 - Zulässigkeit von Komponenten
- **Ordner _rels:** .rels
 - Definiert die Beziehungen zur Hauptkomponente
- **Ordner word:** document.xml
 - Die Hauptkomponente
- **Ordner docProps:** app.xml, core.xml
 - app.xml: Einstellungen der Anwendung
 - core.xml: Dateieigenschaften
- **Ordner customXml:**
 - Benutzerdefinierte Anpassungen (z.B. Ribbons)
- **Ggf. weitere Ordner** für
 - Bilder und andere Medien
 - Anwendungsabhängige Elemente



Olaf Lischke: Visual Studio Tools for Office – www.lischke-edv.de

OpenXML – Grundlegende Struktur .xlsX

- **Root-Ebene:** [Content_Types].xml
 - Zuordnung Dateitypen zu Dateiendungen
 - Zulässigkeit von Komponenten
- **Ordner _rels:** .rels
 - Definiert die Beziehungen zur Hauptkomponente
- **Ordner xl:** workbook.xml
 - Die Hauptkomponente
- **Unterordner worksheets:** sheetN.xml
 - Die einzelnen Blätter der Mappe, bei 1(!) beginnend
- **Ordner docProps:** app.xml, core.xml
 - app.xml: Einstellungen der Anwendung
 - core.xml: Dateieigenschaften
- **Ggf. weitere Ordner** für
 - Bilder und andere Medien
 - Anwendungsabhängige Elemente



Olaf Lischke: Visual Studio Tools for Office – www.lischke-edv.de

OpenXML – document.xml

<w:document>	Dokument
<w:body>	Body
<w:p>	Paragraph (Absatz)
<w:r>	Run (Lauf)
<w:t>	Text
Hello World!	
</w:t>	
</w:r>	
</w:p>	
</w:body>	
</w:document>	



Olaf Lischke: Visual Studio Tools for Office – www.lischke-edv.de

OpenXML workbook.xml, sheetN.xml

workbook.xml	
<x:workbook>	Mappe
<x:sheets>	Blätter
<x:sheet name="Blattname" ... />	Einzelnes Blatt
...	
</x:sheets>	
</x:workbook>	
sheetN.xml	
<x:worksheet>	Blatt
<x:sheetData />	Daten
</x:worksheet>	



Olaf Lischke: Visual Studio Tools for Office – www.lischke-edv.de

WordProcessingML erzeugt .docX

- Dokument erzeugen:
wdDokument = WordprocessingDocument.Create(Pfad, Typ)
- Hauptteil einbauen:
ptHauptteil = wdDokument.AddMainDocumentPart()
- Dokument-Element:
ptHauptteil.Document = New Document()
- Body-Element:
bdBody = ptHauptteil.Document.AppendChild(New Body())
- Paragraph:
prPara = bdBody.AppendChild(New Paragraph())
- Lauf:
rnLauf = prPara.AppendChild(New Run())
- Text:
rnLauf.AppendChild(New Text(„Hallo Welt!“))



Olaf Lischke: Visual Studio Tools for Office – www.lischke-edv.de

SpreadsheetML erzeugt .xlsX

- Mappe erzeugen:
`xlMappe = spreadsheetDocument.Create(Pfad, Typ)`
- Hauptteil einbauen:
`ptHauptteil = xlMappe.AddWorkbookPart()`
- Workbook-Element:
`ptHauptteil.Workbook = New Workbook()`
- Worksheet-Part:
`ptSheet = ptHauptteil.AddNewPart(Of WorksheetPart)()`
- Sheet erzeugen:
`ptSheet.Workbook.AppendChild(Of Sheets)(New Sheets())`



Olaf Lischke: Visual Studio Tools for Office – www.lischke-edv.de

Nützliche Tools von Microsoft

- **VSTO Power Tools**
 - VSTO_PT.exe – Die Kern-PowerTools (z.B. OpenXML Editor)
 - VSTO_PExtLibs.exe – Extension Libraries (nicht nötig, wenn VSTO 2008/2010 vorhanden)
 - VSTO_PTRibbonIDs.exe – Ribbon-Symbole für englisches(!) VS
- **2007OfficeControlIDsExcel2007.exe**
Office2010ControlIDs.exe
 - Selbstentpackendes Archiv mit Exceltabellen über IDs, Namen und Struktur der Office-Ribbon-Elemente (für Excel, Word, Outlook und Powerpoint)
- **OpenXML SDK**
 - Nützliche Tools, um OpenXML-Dokumente zu generieren, ohne Office-Anwendungen starten zu müssen



Olaf Lischke: Visual Studio Tools for Office – www.lischke-edv.de

OpenXML SDK 2.0

- **OpenXMLSDKv2.msi** (4 Mb)
 - Liefert DocumentFormat.OpenXml.dll
Streng getypte Bibliothek für den Zugriff auf OpenXML-DOM
- **OpenXMLSDKTools.msi** (100 Mb)
 - Enthält das OpenXmlSdkTool
 - Validator
 - FileDiff
 - DocumentReflector /



Olaf Lischke: Visual Studio Tools for Office – www.lischke-edv.de

Deployment von VSTO-Lösungen

- ClickOnce
 - Automatische Zusammenstellung:
 - Setup.exe für die VSTO-Runtime
 - Bereitstellungsmanifest *.vsto zur Installation der Lösung
 - Anwendungsmanifest
- Setup-Projekt
 - *.msi-Paket zur Installation
 - Manuelle Zusammenstellung der Dateien u. Registry-Einträge
- Manuelle Kopie



Olaf Lischke: Visual Studio Tools for Office – www.lischke-edv.de

ClickOnce-Bereitstellung für VSTO

- 2 Wege:
 - Erstellen/Veröffentlichen... *Build/Publish...*
 - Assistentengeführt
 - Projekt-Eigenschaften, Register „Veröffentlichen“
 - benutzt die Einstellungen des Registers
- Verzeichnisse:
 - Veröffentlichungsort: „Upload“-Verzeichnis
 - Installationsverzeichnis: Verzeichnis aus Sicht des Benutzers
- Ergebnis:
 - Setup.exe – Bootstrap, prüft Existenz von VSTO et al.
 - *.vsto – Bereitstellungsmanifest (Doppelklick durch Benutzer)
 - Ordner „Application Files“ – Anwendung + Manifest etc.
 - Registry-Einträge sind angelegt



Olaf Lischke: Visual Studio Tools for Office – www.lischke-edv.de

ClickOnce-Bereitstellung für Vorlagen

- ClickOnce-Bereitstellung „wie üblich“
- Zusätzliches Ergebnis:
 - Vorlage-Datei liegt im Bereitstellungsverzeichnis
- Falls Vorlage woanders liegen soll:
 - Dokumenteigenschaften/Erweiterte Eigenschaften:
 - Eigenschaft *_AssemblyLocation* muss den Pfad des Bereitstellungsverzeichnisses vorangestellt bekommen (Achtung: Anhang „[GUID]“ nicht ändern/löschen!)



Olaf Lischke: Visual Studio Tools for Office – www.lischke-edv.de

Setup-Projekt für VSTO-Lösungen

- Eigenständiges(!) Projekt (Setup-Projekt)
- Anwendungsordner:
 - Assembly, Bereitstellungs-, Anwendungsmanifest
- Weitere Abhängigkeiten nicht notwendig
 - Ggf. gefundene löschen...
- Registry-Einträge:
 - HKCU\Software\Microsoft\Office\[App]\AddIns\[AddInName]
 - Einträge:
 - Description, FriendlyName: [AddInName] o.ä.
 - LoadBehavior (DWORD): Meistens 3
 - Manifest: [TARGETDIR][AddInName].vsto|vstolocal



Olaf Lischke: Visual Studio Tools for Office – www.lischke-edv.de

Zertifikate

- Certmgr.msc
 - Management-Konsole für Zertifikate
 - Import von Zertifikaten
 - Ändern der Vertrauensstellung
 - **Nicht möglich:** Erstellen neuer Zertifikate
- Selfcert.exe
 - Erstellen eigener Zertifikate
 - Startmenü:
 - Microsoft Office Tools/Digitales Zertifikat für VBA-Projekte



Olaf Lischke: Visual Studio Tools for Office – www.lischke-edv.de

Vielen Dank für Ihre/Eure
Aufmerksamkeit!

Olaf Lischke

Anwendungsentwicklung – Beratung – Schulung

olaf.lischke@lischke-edv.de

+49 (234) 976 698 80



Olaf Lischke: Visual Studio Tools for Office – www.lischke-edv.de