

# Visual Studio Tools voor Office

## BUSINESSFUNCTIONALITEIT INTEGREREN IN OFFICE-DOCUMENTEN

Bij Visual Studio 2005 is Visual Studio Tools for Office geïntegreerd. Met Visual Studio Tools for Office 2005 is het maken van applicaties die geïntegreerd zijn in Office eenvoudiger geworden en zijn er nieuwe mogelijkheden bij gekomen. Dit artikel geeft een indruk van de nieuwe features, die worden gedemonstreerd aan de hand van een demoapplicatie.

Visual Studio Tools voor Office 2005 (VSTO) is een onderdeel van Visual Studio 2005 en wordt gebruikt om Office-applicaties te ontwikkelen. Ook in Visual Studio 2003 was al VSTO beschikbaar. De 2005-versie is uitgebreid en maakt het eenvoudiger om Smart Documents te ontwikkelen. In dit artikel besteden we speciale aandacht aan de nieuwe mogelijkheden die VSTO 2005 biedt. Eerst worden de nieuwe features besproken en vervolgens worden deze toegepast in een demoapplicatie.

Een bekende voorloper van VSTO is Visual Basic for Applications (VBA). Het gebruik van .NET in plaats van VBA heeft de volgende voordelen:

- De .NET-assembly met de business logic kan eenvoudig in meer documenten worden gebruikt zonder dat onderhoud ook op meer plaatsen hoeft plaats te vinden.
- De kracht van .NET kan gebruikt worden in een Office-document.

### Integratie in de IDE

VSTO-oplossingen worden in Visual Studio ontwikkeld, waarvoor in Visual Studio 2005 een speciale projecttemplate aanwezig is. Dit .NET-project bevat een Office-document (de Word- of Excel-template), en voor het toevoegen van functionaliteit is een grafische interface aanwezig. Hiermee kun je visuele elementen koppelen aan events op een document. Een voorbeeld hiervan is code die wordt uitgevoerd wanneer een gebruiker de inhoud van een bookmark heeft aangepast. Ook het opmaken van het Office-document vindt plaats in Visual Studio. In afbeelding 1 wordt hiervan een voorbeeld gegeven. Ook .NET-controls kunnen eenvoudig in het document worden geplaatst. Dit kun je gebruiken voor keuzelijsten, maar bijvoorbeeld ook voor kalender-controls in formulieren.

### Eigenschappen van een VSTO-document

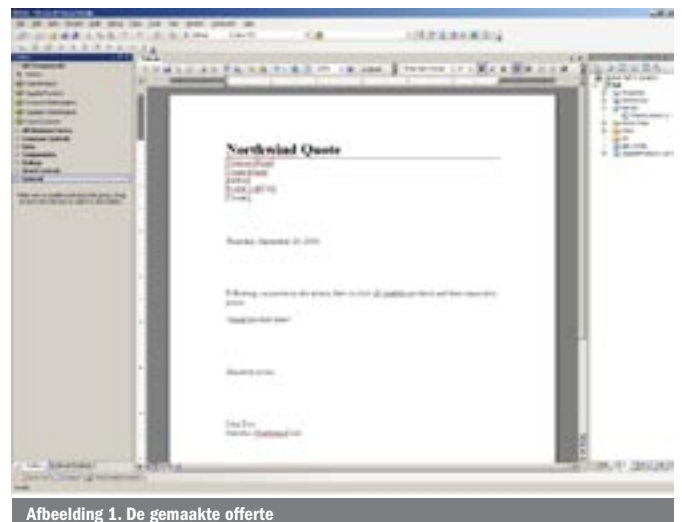
Een Office VSTO-document wordt gekenmerkt door dat het twee custom-properties op het document heeft: `_AssemblyName` en `_AssemblyLocation`. In VSTO 2003 stonden in deze properties respectievelijk de naam van de main assembly waar de .NET-code en de locatie waar de main assembly zich bevinden. Op basis van deze functies kon de VSTO 2003 class-loader de juiste assembly laden. In VSTO 2005 bestaan de custom-properties nog steeds, maar hebben zij een andere invulling. De `_AssemblyName` is een flag geworden die aangeeft of het een VSTO-document is. `AssemblyName = "*"`  betekent dat er een code-extension aanwezig is. De `AssemblyLocation` bevat de GUID van het Runtime Storage Control. Dit is een control die als ActiveX-control aan een document wordt toegevoegd en die de VSTO-diensten voor het document verzorgt. De locatie van de assembly staat in een zogenaamd manifest.

### Manifest-based deployment

Het application-manifest bevat referenties naar de assemblies. Bij deployment kan verwezen worden naar een manifest op de server. VSTO gebruikt dezelfde techniek als .NET ClickOnce voor het bepalen of een nieuwe versie moet worden gedownload. Hiermee is het mogelijk om alleen op de server een nieuwe versie van assemblies en manifest uit te rollen, waarna het Office-document de nieuwe versies automatisch downloadt.

### Gescheiden data en view

Met VSTO 2005 zijn gegevens en de objecten die de gegevens gebruiken gescheiden. De gegevens worden opgeslagen in een XML data-island dat gekoppeld is aan het document. De opslag van de data in het document wordt door de Runtime Storage Control verzorgd. De representatie van gegevens gebeurt via zogenaamde view-controls. View-controls zijn documentspecifieke controls met een eigen representatie, properties en events. View-controls kunnen worden gekoppeld, via databinding, aan een dataset of een kolom in een datatable in een data-island. Een voorbeeld van een view-control in Excel is een lijst die gebonden is aan een tabel in een dataset. In Word is een view-control bijvoorbeeld een bookmark. Een bijkomend voordeel van het data-island is dat de gegevens bij het document worden bewaard en gebruikt kunnen worden om zonder netwerkconnectie (in een offline scenario) de gegevens in het document te gebruiken en/of te manipuleren.



Afbeelding 1. De gemaakte offerte

## Server-programming

Omdat de gegevens worden opgeslagen in een data-island, is het mogelijk om deze data los van het document te benaderen. Via de ServerDocument-class, die onderdeel is van .NET Framework 2.0, kan dit zelfs vanaf een server zonder dat Office op de server geïnstalleerd is. Via de ServerDocument-class kunnen de data op de server worden uitgelezen, als een document wordt ontvangen, of kunnen gegevens worden toegevoegd voordat het document naar de eindgebruiker wordt verstuurd.

## Actions pane

Net als in VSTO 2003 kan ook in VSTO 2005 de actions pane worden gebruikt voor custom actions bij een document. In VSTO 2005 is deze mogelijkheid verder uitgebreid. Het is mogelijk om gebruik te maken van .NET-usercontrols, waardoor het veel makkelijker is om actions panes te ontwikkelen.

## Beveiliging

De beveiliging van de een VSTO 2005-document wordt gerealiseerd door .NET code access security. Omdat de assemblies via COM-interop met Office communiceren moet de code onder full trust draaien. In VSTO 2005 is er een extra beveiliging bijgekomen. Als een document op een shared location staat, zoals een netwerk-share of een SharePoint-documentbibliotheek, moet aan deze locatie ook full trust worden toegekend.

## Schema-based programming

In Microsoft Office Professional 2003 kunnen XML-schema's gebruikt worden om de data in het document te structureren. Als een XML-schema aan een document wordt gekoppeld in Visual Studio, dan genereert VSTO 2005 voor het gebruikte schema een objectmodel. Hiermee kan de data in het document op 'strong type based' manier worden benaderd en kunnen events aan de data-structuur worden gekoppeld.

## Voorbeeldscenario

In de volgende paragrafen zullen we in meer detail op de nieuwe features van VSTO 2005 ingaan, aan de hand van een case study. De code van deze case study is te vinden op [www.microsoft.nl/netmagazine11](http://www.microsoft.nl/netmagazine11). Het scenario van de case study is het maken van een eenvoudige offerte in Word. Bij de offerte kan een klant gezocht worden en de gegevens van deze klant kunnen aan het document worden toegevoegd. Ook kan een product worden geselecteerd, waarna de gegevens aan een productenlijst in het document kunnen worden toegevoegd. Bij het maken van de applicatie wordt het gebruik van de actions pane, Office Word View Controls, het werken met data, offline-gebruik en serverside programming gedemonstreerd.

## Structuur offerte

De offerte die is gemaakt, is te zien in afbeelding 1. Om de naam en adresgegevens van de klant in het document te kunnen plaatsen, wordt gebruik gemaakt van bookmarks. Voor ieder veld is er een eigen bookmark. De CompanyName-bookmark is uitgebreid met een functie die het mogelijk maakt om met een rechterklik op de tekst een zoekactie uit te voeren; zie afbeelding 2. Het document bevat een eigen actions pane. Deze wordt getoond in afbeelding 3. De actions pane bevat twee usercontrols. Een voor het zoeken van een klant en een voor het selecteren van een product.

## De actions pane

Een actions pane in VSTO 2005 wordt samengesteld met speciale usercontrols, genaamd actions pane-controls. Deze controls kunnen uit de new item-lijst gekozen worden als een nieuw item aan de solution wordt toegevoegd. Het ontwikkelen van de actions pane-control kan op dezelfde wijze worden uitgevoerd als normale usercontrols. Voor de offerte zijn twee actions pane-usercontrols gemaakt: InsertCustomer en InsertProduct. Deze actions pane-usercontrols kunnen op eenvoudige wijze aan het document worden toegevoegd. De actions pane-controls worden geïnstantieerd



Afbeelding 2. Een voorbeeld van een zoekactie

waarna ze aan de document actions pane worden toegevoegd. De volgorde van toevoeging bepaalt hoe de actions pane-controls in de actions pane getoond worden. In codevoorbeeld 1 is te zien hoe de toevoeging kan worden uitgevoerd. De eerste actions pane-control is InsertCustomer. Deze bevat een text-box, een button en een listbox. Hiermee wordt vanuit de Northwind-database een selectie van customers opgehaald en in de listbox getoond. Dit is standaard .NET-werk en wordt in dit artikel niet verder beschreven. Als een klant uit de listbox wordt gekozen, worden de detailgegevens in de offerte toegevoegd. Dit gebeurt door de bookmarks met de juiste gegevens te vullen in de SelectedIndexChanged-eventhandler van de listbox. Dit is te zien in codevoorbeeld 2. Via een special globaal object *Globals* kunnen de bookmarks via *Globals.ThisDocument.bookmarkname* benaderd worden en kan de waarde worden gezet. De tweede actions pane-control is de InsertProduct. Deze control bevat een dropdownlist en een listbox. De dropdownlist bevat alle suppliers en de listbox de producten die bij een supplier horen. Ook deze controls worden gevuld vanuit de Northwind-database. Als een supplier en product zijn gekozen, kunnen deze met een button Insert aan het document worden toegevoegd. In codevoorbeeld 3 wordt de code getoond, waarbij er een tabel-header wordt toegevoegd aan het document op de locatie waar de cursor staat. In codevoorbeeld 4 staat de code om een rij toe te voegen met de supplier- en productgegevens.



Afbeelding 3. De actions pane

```
// Declare the taskpane controls
private bid.Action_Pane.InsertCustomer insertCustomerCtrl;
private bid.Action_Pane.InsertProduct insertProductCtrl;

// Instantiate taskpane controls
insertCustomerCtrl = new bid.Action_Pane.InsertCustomer();
insertProductCtrl = new bid.Action_Pane.InsertProduct(dsSupplierProducts);

// Add the taskpane controls to the document's ActionPane
this.ActionsPane.Controls.Add(insertCustomerCtrl);
this.ActionsPane.Controls.Add(insertProductCtrl);
```

Codevoorbeeld 1.

```
Globals.ThisDocument.showCompanyName.Text = row["CompanyName"].ToString();
Globals.ThisDocument.showContactName.Text = row["ContactName"].ToString();
Globals.ThisDocument.showAddress.Text = row["Address"].ToString();
Globals.ThisDocument.showPostalCode.Text = row["PostalCode"].ToString();
Globals.ThisDocument.showCity.Text = row["City"].ToString();
Globals.ThisDocument.showCountry.Text = row["Country"].ToString();
```

Codevoorbeeld 2.

```
// Insert a table at the cursor
Word.Table tbl = Globals.ThisDocument.Tables.Add
(Globals.ThisDocument.Application.Selection.Range, 1, 4,
ref missing, ref missing);

// Insert table headings
SetHeadings(tbl.Cell(1, 1), "Supplier Name");
SetHeadings(tbl.Cell(1, 2), "Product Name");
SetHeadings(tbl.Cell(1, 3), "Quantity Per Unit");
SetHeadings(tbl.Cell(1, 4), "Unit Price");
```

Codevoorbeeld 3.

```
//Make a new row at the table in the document
Word.Row newRow = Globals.ThisDocument.Tables[1].Rows.Add(ref missing);

//Fill the cells of the new row
newRow.Cells[1].Range.Text = companyName;
newRow.Cells[2].Range.Text = productName;
newRow.Cells[3].Range.Text = quantityPerUnit;
newRow.Cells[4].Range.Text = unitPrice;
```

Codevoorbeeld 4.

```
private void ThisDocument_Startup(object sender, System.EventArgs e)
{
    ...
    // add a popupmenu to the document
    this.AddPopupMenu();

    // add eventhandler to the bookmark to show popup when right-clicked
    this.showCompanyName.BeforeRightClick += new
        Microsoft.Office.Tools.Word.ClickEventHandler
        (showCompanyName_BeforeRightClick);
}

private void showCompanyName_BeforeRightClick(object sender,
Microsoft.Office.Tools.Word.ClickEventArgs e)
{
    ...
    //The selected bookmark is stored to retrieve its
    //content in the search event.
    selectedBookmark = (Microsoft.Office.Tools.Word.Bookmark)sender;
    cbPopup.ShowPopup(missing, missing);
    e.Cancel = false;
}
}
```

Codevoorbeeld 5.

```
public partial class ThisDocument
{
    ...
    // dataset needs to be public to be cached
    [Cached]
    public SupplierProducts dsSupplierProducts;

    private void ThisDocument_Startup(object sender, System.EventArgs e)
    {
        //create the products dataset, which is automatically cached
        if (dsSupplierProducts == null)
        {
            //Create and Fill dsSupplierProducts
            ...
        }
        else
        {
            //use cached data
            //dsSupplierProducts is automatically instantiated by VSTO
        }
    }
}
```

Codevoorbeeld 6.

## Het search popup-menu bij een bookmark

Naast de actions pane is er in de offerte nog een andere wijze ontwikkeld om customers te zoeken. De bookmark *CompanyName* is voorzien van een popup-menu dat verschijnt als er rechts geklikt wordt op de bookmark. In afbeelding 2 wordt de GUI getoond. Na de rechtermuisklik wordt de inhoud van de bookmark gebruikt om in de database te zoeken. De resultaten worden op dezelfde wijze verwerkt als bij het zoeken via de actions pane. In codevoorbeeld 5 wordt getoond hoe dit is gerealiseerd. In het Startup-event wordt eerst een popup-menu gemaakt. Hiervoor wordt de standaard Office CommandBar gebruikt. Vervolgens wordt gebruik gemaakt van de mogelijkheid in VSTO 2005 om events aan een view-control te koppelen. Aan de bookmark *CompanyName* wordt het *BeforeRightClick*-event gekoppeld. In de eventhandler hiervan wordt de geselecteerde bookmark bewaard en de popup getoond. De bookmark wordt bewaard zodat de inhoud door de search-functie gebruikt kan worden. Aan het eind wordt de *Cancel*-flag van het event op *True* gezet. Hiermee wordt voorkomen dat andere eventhandlers (bijvoorbeeld het standaard popupmenu) worden aangeroepen.

## Offline

In de offerte wordt een dataset gebruikt die product- en supplier-gegevens bevat. Deze dataset wordt in het document gecached, zodat deze beschikbaar is bij offline gebruik. Naast een dataset kunnen alle objecten die serialiseerbaar zijn door VSTO worden gecached. Het cacheten van een object in een VSTO-project is eenvoudig. Codevoorbeeld 6 toont hoe dit moet. Eerst wordt de typed dataset *dsSupplierProducts* gedeclareerd. Deze class-variabele moet de scope 'public' hebben om gecached te kunnen worden. Met het plaatsen van het attribuut *[Cached]* wordt aangegeven dat VSTO deze class-variabele moet cacheten. In het startup-event wordt gecontroleerd of de class-variabele *dsSupplierProducts* null is. Indien dit het geval is wordt deze geïnstantieerd en automatisch door VSTO gecached. Als het document wordt bewaard en opnieuw opgestart, zal VSTO de class-variabele *dsSupplierProducts* automatisch instantiëren en vullen met de gegevens uit de cache.

## Businessfunctionaliteit voor eindgebruikers

VSTO 2005 levert veel nieuwe functionaliteit. Hierdoor wordt het steeds eenvoudiger om applicaties geïntegreerd in Office te ontwikkelen en te beheren. Daarmee wordt het mogelijk om businessfunctionaliteit aan eindgebruikers te leveren in een omgeving die voor hen bekend is.

**Sander van den Hoven** is Information Worker Consultant bij Microsoft Service. Zijn mailadres is [sander.vandenhoven@microsoft.com](mailto:sander.vandenhoven@microsoft.com). Dit artikel is geschreven in samenwerking met **Frédéric Lapierre** ([frederic.lapierre@microsoft.com](mailto:frederic.lapierre@microsoft.com)) en **Ingeborg Struijk** ([ingebors@microsoft.com](mailto:ingebors@microsoft.com)), beiden Information Worker consultants bij Microsoft Services.

### Nuttige internetadressen

Demo applicatie bij dit artikel: [www.microsoft.nl/netmagazine11](http://www.microsoft.nl/netmagazine11)  
Office development centre: <http://msdn.microsoft.com/office/>  
MSDN-documentatie: [http://msdn2.microsoft.com/en-us/library/d2b7z6d\(en-US,VS.80\).aspx](http://msdn2.microsoft.com/en-us/library/d2b7z6d(en-US,VS.80).aspx)  
Word Labs: [http://msdn.microsoft.com/library/en-us/odc\\_vsto2005\\_tr/html/OfficeVSTO2005WordLabs.asp](http://msdn.microsoft.com/library/en-us/odc_vsto2005_tr/html/OfficeVSTO2005WordLabs.asp)  
Excel Labs: [http://msdn.microsoft.com/library/en-us/odc\\_vsto2005\\_tr/html/OfficeVSTO2005ExcelLabs.asp](http://msdn.microsoft.com/library/en-us/odc_vsto2005_tr/html/OfficeVSTO2005ExcelLabs.asp)  
Code snippets: [http://msdn.microsoft.com/library/en-us/odc\\_vsto2005\\_ta/html/officeVSTOCodeSnippets.asp](http://msdn.microsoft.com/library/en-us/odc_vsto2005_ta/html/officeVSTOCodeSnippets.asp)  
MSDN-artikel: [http://msdn.microsoft.com/library/en-us/odc\\_vsto2005\\_ta/html/OfficeVSTOServerCapabilities.asp](http://msdn.microsoft.com/library/en-us/odc_vsto2005_ta/html/OfficeVSTOServerCapabilities.asp)  
Runtime Storage Control: <http://msdn2.microsoft.com/en-us/library/7ydwehbf>  
Caching data (voor offline gebruik van VSTO-documenten): [http://msdn2.microsoft.com/en-us/library/75akte27\(en-US,VS.80\).aspx](http://msdn2.microsoft.com/en-us/library/75akte27(en-US,VS.80).aspx)  
ClickOnce: <http://msdn.microsoft.com/msdnmag/issues/04/05/clickonce/>  
Appl. & deployment manifests: <http://msdn2.microsoft.com/en-us/library/a3swb498>  
XML Schemas and data: [http://msdn2.microsoft.com/en-us/library/y36t3e16\(en-US,VS.80\).aspx](http://msdn2.microsoft.com/en-us/library/y36t3e16(en-US,VS.80).aspx)