

Peter Himschoot en Wim Uyttersprot

Peter Himschoot is werkzaam als softwarearchitect en trainer bij U2U, en is MSDN Regional Director voor België en Luxemburg. Hij is bereikbaar via peter@u2u.net. Wim is director van U2U en bereikbaar via wim@u2u.net. U2U is een Microsoft CTEC, gevestigd in Brussel en gespecialiseerd in software-ontwikkeling voor .NET, BizTalk 2004, SharePoint 2003 en MS Office 2003.

Ontwikkelen Smart Documents met Microsoft Office System

AFWEZIGHEIDRAPPORT.DOC

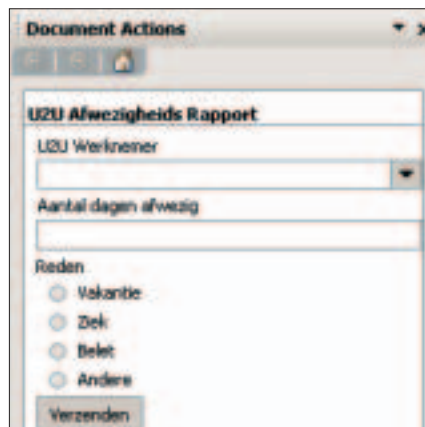
'Smart Documents'-technologie is Microsoft's laatste innovatie op het gebied van integratie van bestaande applicaties met Office-toepassingen zoals Microsoft Word en Excel. In dit artikel bouwen we een voorbeeld van een Smart Document.

Een Smart Document is een Office-document waarbij je de gebruiker helpt met het 'op een slimme manier' invullen van een document. Dit gebeurt door de gebruiker de informatie die hij of zij nodig heeft rechtstreeks aan te bieden. In een volledig uitgewerkt Smart Document hoeft de gebruiker niet handmatig verscheidene applicaties te openen en gegevens over te brengen van de ene naar de andere applicatie. Met een beetje coderen, maakt de Smart Documents-technologie het mogelijk dat gegevens automatisch worden uitgewisseld tussen de Office-documenten. Smart Documents kunnen bovendien ingevoerde gegevens valideren en doorsturen naar andere applicaties. In het voorbeeld dat we hier bouwen, helpen we de gebruiker met het invullen van een zogenaamd afwezigheidsrapport. De bedoeling is dat de gebruikers hun eigen naam kunnen kiezen uit een lijst van bestaande gebruikers, en dat ze de reden van afwezigheid aanduiden door op een radiobutton te klikken.

Het programmeermodel

Wie al eens een 'Office Smart Tag' heeft geschreven in Office XP, zal de nieuwe manier van programmeren direct herken-

nen. We moeten namelijk een object dat door Office wordt opgeroepen implementeren via een interface. We zullen de verschillende methodes moeten schrijven die worden gebruikt voor het aanmaken van controls in de 'Document Actions' taskpane. Afbeelding 1 toont een voorbeeld van de 'Document Actions'.

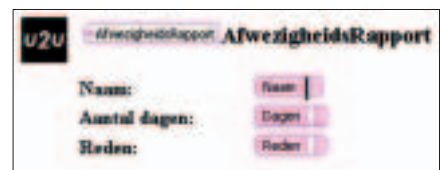


Afbeelding 1. Voorbeeld van de 'Document Actions'

Een eerste Smart Document

In het voorbeeld van het afwezigheidsrapport openen we het 'Afwezigheidsrapport.doc'; zie afbeelding 2. Dit document bevat een aantal velden dat de gebruiker dient in te vullen. Ook is het document gekoppeld aan een XML-sche-

ma, om deze velden gemakkelijk te kunnen herkennen. Het gebruikte schema bevindt zich in een apart bestand: Afwezigheidsrapport.xsd (zie downloads). Er zijn drie velden: de naam van de afwezige, het aantal dagen dat hij afwezig is, en de reden van afwezigheid. We kunnen deze velden zien door de XML-tags zichtbaar te maken zoals in afbeelding 2. Het is de bedoeling deze XML-elementen te binden aan controls in de taskpane.



Afbeelding 2. Afwezigheidsrapport.doc

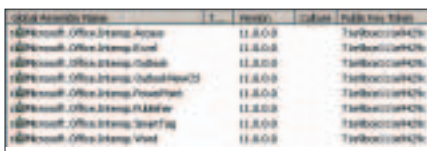
Smart Documents Type Names

Wanneer op één van die velden in het document wordt geklikt, zal Office op zoek gaan naar de desbetreffende Smart Document-extensies. Dit gebeurt op basis van een unieke naam, de Smart Document Type Name (SDTN), wat een string is met het formaat 'namespace#element'. De namespace en het element hierin komen overeen met deze van het XML-schema. In het

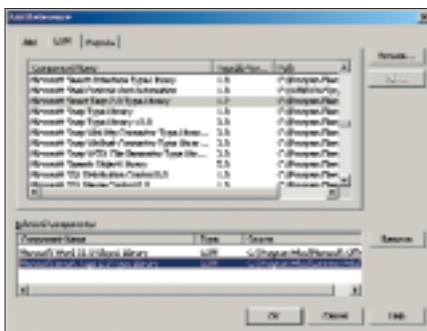
voorbeeld koppelen we Smart Documents-code aan het 'AfwezigheidRapport'-element uit de AfwezigheidRapport-namespace: de te gebruiken SDTN ziet er dan als volgt uit: 'AfwezigheidRapport# AfwezigheidRapport'.

Beginnen met coderen

Het voorbeeld van dit artikel werkt enkel als Microsoft Office 2003 is geïnstalleerd inclusief de .NET-integratie. Controleer hiervoor in de GAC (via Windows Explorer: <INSTALL DRIVE>\Windows\assembly), waar zich de Microsoft.Office.Interop-assemblies moeten vinden (zie afbeelding 3), met in het bijzonder de SmartTag- en Word-assemblies. Na het starten van Microsoft Visual Studio 2003, maken we een nieuw VB.NET library-project dat we de naam 'Afwezigheidsrapport' geven. We geven de class die hierbij automatisch wordt aangemaakt een nieuwe naam: 'AfwezigheidsRapportSmartDoc'. Ook voegen we referenties (via de COM-tab) toe naar 'Microsoft Smart Tags 2.0 Type Library' en naar 'Microsoft Word 11.0 Object Library'; zie afbeelding 4.



Afbeelding 3. Microsoft.Office.Interop-assemblies



Afbeelding 4. Wat te selecteren

We voegen ook een Imports-statement toe naar de volgende namespaces Microsoft.Office.Interop.SmartTags en Microsoft.Office.Interop.Word. Onze class moet ook de ISmartDocument-interface implementeren. MS Office, in dit geval Word, zal een instantie van onze class aanmaken en de gedefinieerde methodes via deze interface aanroepen. Eerst wordt de SmartDocInitialize-methode aangeroepen. Deze methode kunnen we gebruiken

```
Imports Microsoft.Office.Interop.SmartTag
Imports Microsoft.Office.Interop.Word
Imports System.Diagnostics

Public Class AfwezigheidsRapportSmartDoc
    Implements ISmartDocument

    ' Initialisatie methode die als eerste wordt aangeroepen.
    Public Sub SmartDocInitialize(ByVal ApplicationName As String, _
        ByVal Document As Object, ByVal SolutionPath As String, _
        ByVal SolutionRegKeyRoot As String) _
        Implements ISmartDocument.SmartDocInitialize
    End Sub
```

Codevoorbeeld 1. Methode is leeg

```
Public ReadOnly Property SmartDocXmlTypeCount() _
    As Integer Implements ISmartDocument.SmartDocXmlTypeCount
    Get
        Return aantalSmartDocs
    End Get
End Property

Public ReadOnly Property SmartDocXmlTypeName(_
    ByVal XMLTypeID As Integer) As String _
    Implements ISmartDocument.SmartDocXmlTypeName
    Get
        Return afwezigheidElement
    End Get
End Property

Public ReadOnly Property SmartDocXmlTypeCaption(_
    ByVal XMLTypeID As Integer, _
    ByVal LocaleID As Integer) As String _
    Implements ISmartDocument.SmartDocXmlTypeCaption
    Get
        Return "Afwezigheids Rapport"
    End Get
End Property
```

Codevoorbeeld 2. Geen gebruik van de XMLTypeID-parameter

om onze eigen initialisaties te definiëren die we door Word wensen te laten voeren; in ons voorbeeld (zie codevoorbeeld 1) laten we deze methode echter leeg.

Methodes voor de Smart Documents Types

Na het aanroepen van de SmartDocInitialize-methode, vraagt Microsoft Word de SmartDocXmlTypeCount-property op. Hierin geven we het aantal SDTN's terug die we ondersteunen, wat in ons voorbeeld juist één is. Microsoft Office zal de waarde van deze property gebruiken om de SmartDocXmlTypeName- en de

SmartDocXmlCaption-properties op te vragen voor elk van deze SDTN's. In onze code weten we welke SDTN wordt opgevraagd, doordat Office een index tussen 1 en de count in de XMLTypeID-parameter opgeeft. De SmartDocXmlTypeName-property moet de naam van de SDTN terug moet geven, geven we in ons voorbeeld de string 'AfwezigheidsRapport# AfwezigheidsRapport' terug. De SmartDocXmlCaption wordt gebruikt als de titel voor de Document Actions-taskpane. In ons voorbeeld wordt dit de string 'AfwezigheidsRapport'. In het codevoorbeeld 2 gebruikten we de XMLTypeID-parameter

```

Public ReadOnly Property ControlCount(ByVal XMLTypeName As _
String) As Integer Implements _
Microsoft.Office.Interop.SmartTag.ISmartDocument.ControlCount
Get
    Return 4
End Get
End Property

Public ReadOnly Property ControlID(ByVal XMLTypeName As String, _
ByVal ControlIndex As Integer) As Integer Implements _
Microsoft.Office.Interop.SmartTag.ISmartDocument.ControlID
Get
    Return 100 + ControlIndex
End Get
End Property

Public ReadOnly Property ControlNameFromID(ByVal ControlID As _
Integer) As String Implements _
Microsoft.Office.Interop.SmartTag.ISmartDocument.ControlNameFromID
Get
    Return "Afwezigheid#" & ControlID.ToString()
End Get
End Property

Public ReadOnly Property ControlTypeFromID(ByVal ControlID As _
Integer, ByVal ApplicationName As String, ByVal LocaleID As _
Integer) As Microsoft.Office.Interop.SmartTag.C_TYPE Implements _
Microsoft.Office.Interop.SmartTag.ISmartDocument.ControlTypeFromID
Get
    Select Case ControlID
        Case afwezigheidNaamControlID
            Return C_TYPE.C_TYPE_COMBO
        Case afwezigheidDagenControlID
            Return C_TYPE.C_TYPE_TEXTBOX
        Case afwezigheidRedenControlID
            Return C_TYPE.C_TYPE_RADIOGROUP
        Case afwezigheidSendControlID
            Return C_TYPE.C_TYPE_BUTTON
    End Select
End Get
End Property

```

Codevoorbeeld 3. Implementatie van de ControlNameFromID en ControlTypeFromID

niet, omdat er maar 1 SDTN is. In afbeelding 1 zie je die titel trouwens.

Methodes voor de Smart Documents-controls

MS Word stelt ons voor elke SDTN een aantal vragen. De eerste vraag is het aantal controls dat nodig is voor dit type; we willen vier controls te gebruiken, dus 'ControlCount' moet 4 zijn. Office zal voor elke control een controlID vragen, een getal dat elke control op unieke wijze iden-

tificeert in het hele Smart Document. We gebruiken hiervoor een truck: voor elk element gebruiken we een offset. Als ons document twee SDTN's bevat, wordt dit 100 als offset voor het eerste, en 200 voor het tweede type. Aangezien we maar één SDTN gebruiken wordt de ControlID-property geïmplementeerd met als Return 100 + ControlIndex. Deze controlID wordt daarna gebruikt om het type control op te vragen, de naam van de control en de caption. De ID's worden trouwens later

nog gebruikt. Implementatie van de ControlNameFromID en ControlTypeFromID vind je in codevoorbeeld 3.

Soorten controls

In de Document Actions taskpane kunnen allerlei controls worden getoond; zie afbeelding 5. In het 'Afwezigheidsrapport.doc' document gebruiken we een combobox, een textbox, een radiogroup en een button. Om voor elke controlID het juiste type terug te geven, implementeren we deze methode als een grote Select Case-statement.



Afbeelding 5. Lijst van controls

Initialisatie van controls

Telkens als een element in het document wordt gekozen, zal Office vragen hoeveel controls we willen, en die aanmaken in de taskpane. Hierna moeten die controls nog worden geïnitieerd. Dit gebeurt met de PopulateXXX-methodes. Een combobox vullen we met elementen met behulp van de PopulateListOrComboContent-methode (codevoorbeeld 4) en een opgave van de List- en Count-argumenten - die ByRef zijn. In het voorbeeld is dit een nieuwe string-array gevuld met namen. In een real-world scenario zou een applicatie de lijst kunnen creëren. Op een analoge manier initialiseren we een radiobutton-group via de PopulateRadioGroup-methode en de argumenten List, Count- en InitialSelected. Om onze radiobutton-group de redenen van afwezigheid te laten weergeven, vullen een array met 'Vakantie', 'Ziek', 'Belet' en 'Andere'. Om geen default te selecteren krijgt InitialSelected de waarde 0. De code vind je in codevoorbeeld 4.

Events van controls opvangen

Wanneer eenmaal de controls zijn aangemaakt en geïnitieerd, mogen we de events verwachten. Deze moeten we in onze code niet als normale .NET-events implementeren, maar als metho-

```
Public Sub PopulateListOrComboContent(ByVal ControlID As Integer, ByVal ApplicationName As String, _
    ByVal LocaleID As Integer, ByVal Text As String, ByVal Xml As String, ByVal Target As Object, _
    ByVal Props As Microsoft.Office.Interop.SmartTag.ISmartDocProperties, ByRef List As System.Array, _
    ByRef Count As Integer, ByRef InitialSelected As Integer)
    Implements Microsoft.Office.Interop.SmartTag.ISmartDocument.PopulateListOrComboContent
        Dim namen As String() = New String() {"Jurgen", "Patrick", "Peter", "Wim"}
        List = namen
        Count = namen.Length
End Sub

Public Sub PopulateRadioGroup(ByVal ControlID As Integer, ByVal ApplicationName As String, _
    ByVal LocaleID As Integer, ByVal Text As String, ByVal Xml As String, ByVal Target As Object, _
    ByVal Props As Microsoft.Office.Interop.SmartTag.ISmartDocProperties, ByRef List As System.Array, _
    ByRef Count As Integer, ByRef InitialSelected As Integer)
    Implements Microsoft.Office.Interop.SmartTag.ISmartDocument.PopulateRadioGroup
        Dim redenen As String() = New String() {"Vakantie", "Ziek", "Belet", "Andere"}
        List = redenen
        Count = redenen.Length
        InitialSelected = 0
End Sub
```

Codevoorbeeld 4. PopulateListOrComboContent-methode

```
Public Sub OnListOrComboSelectChange(ByVal ControlID As Integer, ByVal Target As Object, _
    ByVal Selected As Integer, ByVal Value As String) _
    Implements Microsoft.Office.Interop.SmartTag.ISmartDocument.OnListOrComboSelectChange
        Dim namen As String() = New String() {"Jurgen", "Patrick", "Peter", "Wim"}
        Dim r As Range = CType(Target, Range)
        r = r.XMLNodes(2).Range
        r.Text = Value
End Sub

Public Sub OnTextboxContentChange(ByVal ControlID As Integer, ByVal Target As Object, _
    ByVal Value As String) Implements Microsoft.Office.Interop.SmartTag.ISmartDocument.OnTextboxContentChange
        Dim r As Range = CType(Target, Range)
        r = r.XMLNodes(3).Range
        r.Text = Value
End Sub

Public Sub OnRadioGroupSelectChange(ByVal ControlID As Integer, ByVal Target As Object, _
    ByVal Selected As Integer, ByVal Value As String) _
    Implements Microsoft.Office.Interop.SmartTag.ISmartDocument.OnRadioGroupSelectChange
        Dim r As Range = CType(Target, Range)
        r = r.XMLNodes(4).Range
        r.Text = Value
End Sub

Public Sub InvokeControl(ByVal ControlID As Integer, ByVal ApplicationName As String, _
    ByVal Target As Object, ByVal Text As String, ByVal Xml As String, ByVal LocaleID As Integer) _
    Implements Microsoft.Office.Interop.SmartTag.ISmartDocument.InvokeControl
        MessageBox.Show("Verzonden", "U2U Afwezigheid", MessageBoxButtons.OK, MessageBoxIcon.Information)
End Sub
```

Codevoorbeeld 5. aanroep OnListOrComboSelectChange-methode

```

<manifest xmlns=
  "http://schemas.microsoft.com/office/xmlexpansionpacks/2003">
  <version>1.0</version>
  <uri>AfwezigheidsRapport</uri>
  <solution>
    <solutionID>U2U_AfwezigheidsRapport</solutionID>
    <type>smartDocument</type>
    <alias lcid="1033">U2U Afwezigheidsrapport</alias>
    <file>
      <type>solutionActionHandler</type>
      <managed />
      <version>1.1</version>
      <filePath>bin/AfwezigheidsRapport.dll</filePath>
      <CLSNAME>AfwezigheidsRapport.AfwezigheidsRapportSmartDoc
        </CLSNAME>
    </file>
  </solution>
  <solution>
    <solutionID>Schema Files</solutionID>
    <type>schema</type>
    <alias lcid="1033">U2U Afwezigheidsrapport schema</alias>
    <file>
      <runFromServer />
      <type>schema</type>
      <version>1.0</version>
      <filePath>AfwezigheidsRapport.xsd</filePath>
    </file>
  </solution>
</manifest>

```

Codevoorbeeld 6.

des die Office aanroept. Wanneer de gebruiker een item in de combobox kiest, wordt de `OnListOrComboSelectChange`-methode aangeroepen; zie codevoorbeeld 5. Andere methodes die we hier implementeren zijn de `OnTextBoxContentChange`, `OnRadioGroupSelectChange` en `InvokeControl`. Deze laatste wordt aangeroepen voor controls zoals de button. In elk van deze methodes kunnen we het `controlID`-argument gebruiken om te zien over welke control het gaat. Het `Value`-argument bevat de gekozen waarde, `Target` verwijst naar het document en gebruiken we om het document zelf aan te passen.

Invullen van elementen in het document Voor elk soort controls vullen we het eigenlijke document in door gebruik te maken van het `Target`-argument, dat een instance is van `Microsoft.Office.Interop.Word.Range`. Via dit `range`-object bereiken we de XML-elementen van het

document, die op hun beurt weer een `range`-object hebben dat aanpasbaar is via de `Value`-property. We kunnen ook andere eigenschappen aanpassen met bepaalde properties. Wat betreft de afwerking van de code: alle andere methodes van deze interface kunnen we gewoon blanco laten.

Installeren van de nieuwe Smart Document library

Microsoft Office 2003 zal onze .NET-assemblies laden mits deze aan een aantal voorwaarden voldoen. Onze assemblies moeten de `FullTrust`-permissie hebben. Dit kan eenvoudig door de assembly een `strong-name` te geven, en die `strong-name` `FullTrust`-permissie te geven met de 'Microsoft .NET Framework 1.1 Configuration', die is te vinden onder administrative tools. We openen onder 'Runtime Security Policy' en binnen 'Machine' de zogenaamde 'Code_Groups'; zie afbeelding 6. Door

rechts te klikken op 'All_Code' en te kiezen voor 'New', komen we bij de 'Create Code Group'-wizard terecht. We geven de codegroep de naam 'Office Smart Documents', en kiezen op de volgende wizard-pagina voor 'Strong Name' om onze assembly via een `strong-name` te importeren.

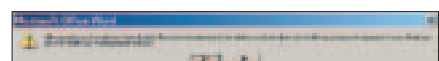


Afbeelding 6. Microsoft .NET Framework 1.1 Configuration

Werken met XML Expansion Packs

Nu komt het erop neer ons document te associëren met onze library, wat kan met behulp van een XML Expansion Pack. In ons voorbeeld is dit de file met de naam `Manifest.xml`; zie codevoorbeeld 6. Wel moet worden gecontroleerd of het `filePath` correct naar de assembly refereert, dat de `CLSNAME` overeenkomt met de class-name, en dat deze file in de projectfolder staat. Het path is namelijk relatief gedefinieerd.

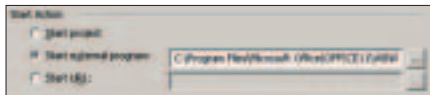
Er is echter een probleem. XML Expansion Packs moeten zijn voorzien van een certificate, want anders wil Office deze packs niet laden. Als developer kan je bij uitzondering de beveiligings-optie uitschakelen, om deze optie opnieuw in te schakelen voor de release-mode. Dit doen we door de 'disable-ManifestSecurityCheck.reg'-registratie uit te voeren. Ook dit bestand is te vinden bij de downloads. Aan een open 'Afwezigheidsrapport.doc'-document, voegen we een XML Expansion Packs toe via de 'Templates and Add-Ins'-dialoog. Office zal ons hierbij waarschuwen dat de security niet optimaal is, wat in dit stadium echter geen probleem is en vooral ons programmeerwerk eenvoudig houdt; zie afbeelding 7. Ten slotte wordt de Smart Documents Library geladen.



Afbeelding 7.

Debuggen van een Smart Documents Library

Een Smart Documents Library debuggen gebeurt vrij eenvoudig. We doen dit door eerst in de projectproperties, via de Debugging-tab (Configuration Properties) te kiezen voor WINWORD.EXE als external program voor het debuggen; zie afbeelding 8. Daarna plaatsen we een breakpoint op de SmartDocXmlType-Count-property om de debugger daar automatisch te laten stoppen.



Afbeelding 8. Debugging

Mogelijke problemen?

Het bouwen en installeren van Smart Documents gaat niet altijd probleem-

loos. Enkele veel voorkomende problemen zijn:

"The XML expansion pack is invalid or cannot be loaded"? Dit betekent dat er een fout in het XML Expansion Pack-bestand zit. Denk erom dat XML case-sensitive is!

"The XML expansion pack or the Smart Document program is missing or invalid"? Dit komt meestal door security-problemen met security. Het is mogelijk dat de assembly geen FullTrust-security heeft.

Debugger stopt niet op het breakpoint? Dit probleem ontstaat doordat Office de verkeerde instantie van de assembly gebruikt, bijvoorbeeld de instantie in de download-cache van de debugger. Om dit

probleem te verhelpen, start je de GACUTIL /CDL. Een andere reden kan zijn dat een oude versie van je assembly zich bevindt in de <DRIVE>:\Documents and Settings\<USER>\Local Settings\Application Data\Microsoft\Schemas-folder. Door deze te verwijderen, en opnieuw je Expansion Pack via het XML Expansion Pack-scherm te installeren, los je het probleem op. Een andere oorzaak kan zijn dat een andere Office-applicatie, zoals Outlook, werkzaam is.

Referenties

<http://www.u2u.net/> (de volledige code van het in dit artikel gebruikte voorbeeld)
<http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnofftalk/html/office09042003.asp>
<http://www.officezealot.com/Office2003Zone/>
<http://msdn.microsoft.com/office/>

© Accenture Technology Solutions 2003. All rights reserved.

/Re-boot/

accenture
Technology Solutions

/Turn innovation into effective technology solutions/
/Make more of your expertise/

www.accenturetechnologiesolutions.nl