



Ernst Peter Tamminga

Managing consultant werkzaam bij XCESS expertise center b.v.
www.xcess.nl

Office 2003 en XML

DE GEBRUIKER AAN DE XML!

Microsoft Office 2003 bevat vele uitbreidingen om met XML-gegevens om te gaan. Deze XML-mogelijkheden kun je ruwweg opdelen in: Office-documenten opslaan als XML en de bewerking van XML. In dit artikel gaat auteur Ernst Peter Tamminga in op de XML-mogelijkheden van Word 2003, Excel 2003 en InfoPath 2003.

Word 2003 is in de opslag van XML het meest uitgebreid in de Office 2003 familie. De Word XML-opslag is volledig compatibel met de 'traditionele' .doc binaire opslag. Excel 2003 is nog niet zover als Word 2003, maar biedt wel veel betere opslag van spreadsheets in XML-formaat dan enig andere voorgaande Excel-versie. Word 2003 en Excel 2003 springen er uit als het gaat om het omzetten van documentgegevens naar bruikbare XML-data; data die daarna gemakkelijk door andere applicaties kunnen worden opgepakt. Het koppelen van Word-documenten en Excel-spreadsheets aan XML-schema's biedt ongekennde mogelijkheden om ongestructureerde data (documenten, spreadsheets) te vertalen naar gestructureerde data (XML), zonder dat de gebruiker zijn 'normale' Office-omgeving hoeft te verlaten. InfoPath 2003 is een volledig nieuw programma in de Office suite. InfoPath is primair gericht op het maken van flexibele invoerformulieren die XML als uitvoer en invoer hebben.

Met XML lukt het wel

XML, XML en XML. Het gonst al enige jaren in de kantoren van de developers: met XML kun je het wel! XML heeft zich

de afgelopen jaren een vaste plaats veroverd als het gaat om het definiëren, communiceren en vastleggen van gegevens (informatie) tussen verschillende applicaties, databases, systemen. Meer en meer onderdelen baseren zich op XML als het gaat om configuratie, opslag, invoer, uitvoer, enzovoort. Als het gaat om de vastlegging van gestructureerde

Het koppelen van Word-documenten en Excel-spreadsheets aan XML-schema's biedt ongekennde mogelijkheden

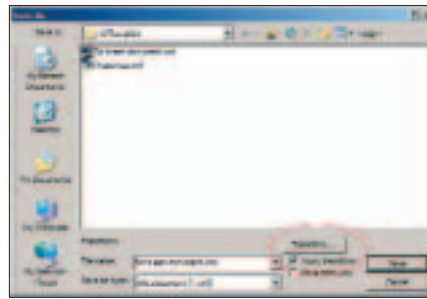
gegevens is dit ongetwijfeld een hele grote stap voorwaarts (geweest), maar hoe zit het met Office-toepassingen, zoals Word, Excel? In de praktijk die ik tegenkom, worden dit soort Office-onderdelen meer gebruikt dan enig andere toepassing. En juist met de Office-toepassingen wordt veel gestructureerde en misschien nog meer ongestructureerde gegevens aangemaakt. Office-onderdelen zijn in hoge mate productiviteitstoepassingen voor de kenniswerkers. Daarbij ontstaat

de noodzaak om vanuit 'reguliere' applicaties (die natuurlijk op .NET/SQL zijn gebaseerd!) Office-documenten aan te maken, zodat de gebruiker daarmee verder kan. Menige routine is geschreven om Word-samenvoegingen automatisch uit te voeren. Je automatiseert Word door het Word objectmodel aan te spreken. Ook Excel verheugt zich in de nodige belangstelling. Vul automatisch getallen in Excel-spreadsheetcellen in, eveneens door het aanspreken van het Excel objectmodel, om bijvoorbeeld flexibele analyses te maken.

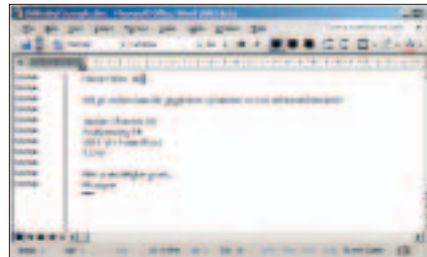
Is het maken van een Office-document met gegevens voor de gevorderde Office-developer gesneden koek, de omgekeerde weg is een heel ander verhaal. Het is meer dan een grote uitdaging om gestructureerde gegevens uit geproduceerde Office-documenten te isoleren als invoer voor verwerking in andere applicaties. Lukt dit extraheren vaak nog wel in een Excel-spreadsheet, door cellen een bepaalde naam te geven om daaraan te refereren, het teruglezen van gewijzigde teksten uit Word-documenten, zoals offertebedragen en percentages vereist de nodige uren van het betere programmeerwerk. Een extra uitdaging is dat Office-



Figuur 1



Figuur 3



Figuur 4



Figuur 2

applicaties 'fat client' toepassingen zijn, en daarom prima op een werkstation kunnen draaien, maar niet op een (web)server geïnstalleerd en opgestart kunnen of mogen worden. Een enorme drempel als je vanuit een webapplicatie Office-documenten wilt aanmaken of lezen. Aan alles komt een eind, ook aan deze problemen: zie aldus Microsoft Office 2003. En het is onvermijdelijk: ook de Office-gebruiker gaat aan de XML. Office 2003 biedt nuttige en handige uitbreidingen op de XML-mogelijkheden die in de voorgaande Office suites beschikbaar waren.

XML als opslagformaat

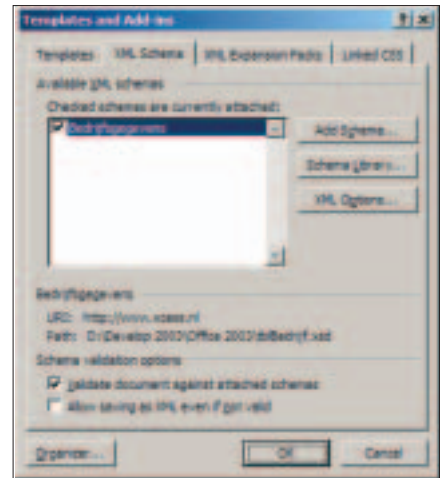
De ondersteuning van XML als opslagformaat door Excel 2003 en Access 2003 is sterk uitgebreid ten opzichte van de voorgaande versies; en bereikt nieuwe hoogtes in Word 2003. Als je wilt kun je het doc-formaat geheel en al vergeten. Het Word XML (WordML) bestand is geheel uitwisselbaar met het doc-bestand, tot en met de VBA-macro's aan toe. Alle onderdelen in het Word-document kunnen in de XML-ver-

sie worden aangewezen aan de hand van de XML-tags. Een voorbeeld daarvan is te zien in figuur 1 in vergelijking met figuur 2.

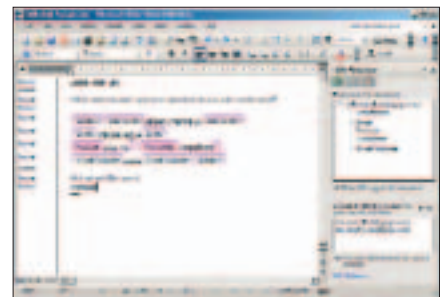
Wat in figuur 2 (de XML-weergave van de documenteigenschappen) opvalt is de verwijzing naar de diverse schema's die gebruikt worden om het XML-bestand te definiëren. De Word-documentstructuur is aan de hand van een XML-schema gedefinieerd. Alle onderdelen van Word-documenten zijn terug te vinden: teksten, stijlen, sjablonen, eigenschappen, lettertype, enzovoort. Word 2003 ondersteunt eigen gedefinieerde XSD's en heeft uitgebreide mogelijkheden om transformaties met XSLT door te voeren bij het wegschrijven van documenten in XML-formaat; zie figuur 3.

Wat kun je hiermee bereiken? Haast te veel om op te noemen. Laat ik een voorbeeld uitwerken: de gebruiker ontvangt een Word-document van buiten, waarin adresgegevens opgenomen zijn. Deze adresgegevens moeten gestructureerd en gecontroleerd in een adressendatabase worden opgenomen. Hieronder de beschrijving van een scenario waarin de gebruiker deze handeling zelf verricht aan de hand van een schema (xsd) dat beschikbaar is.

Stap 1: De gebruiker heeft een document ontvangen (zie figuur 4), opent het



Figuur 5



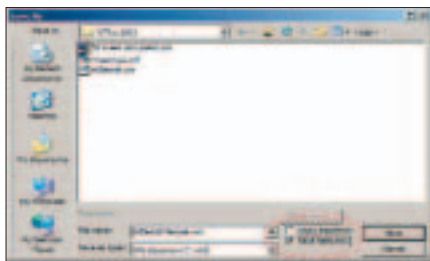
Figuur 6

document en koppelt het bestaande schema (xsd) aan het document (zie figuur 5). Motto: de gebruiker wil een schone set adresgegevens aanmaken vanuit het aangeleverde Word-document.

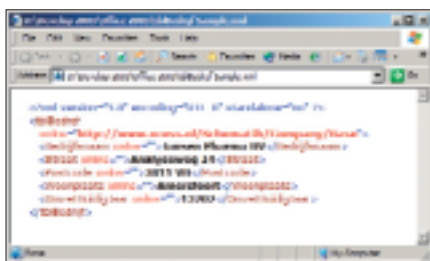
Het koppelen van een xsd is bereikbaar vanuit de optie 'Templates en AddIns', waarbij nu naast de Word-templates die kunnen worden gekoppeld, een nieuwe tab beschikbaar is: XML Schema's. Wijs het schema aan (xsd-file), geef er een naam aan voor gebruik in Word (Bedrijfsgegevens) en sluit de dialoog. Een nieuw XML-taakscherm komt naar voren in het gewone Word-venster: werken met XML structure, zie figuur 6.

Stap 2: Het deel dat gegevens bevat wordt gemarkeerd en gekoppeld aan de definitie. Daarna markeert de gebruiker steeds precies de gegevens in het document die met een element samenhangen en koppelt daar steeds een XML-onderdeel (tag) aan. Gewoon door aanklikken.

Als er in het schema validatieregels zijn opgenomen en het veld voldoet niet aan de regels, dan krijgt de gebruiker daarvan een melding. De aldus gemarkeer-



Figuur 7



Figuur 8

de velden zijn duidelijk herkenbaar; zie figuur 6. Door de checkbox 'Show XML tags in the document' kan de zichtbaarheid van de markering aan- en uitgezet worden. Zijn alle velden gemarkeerd en correct (er is geen fout meer zichtbaar in het Elements-venster), dan kunnen de gegevens als XML worden opgeslagen door de 'Save As'-optie te gebruiken. Wil je alleen de data overhouden, kies dan voor de checkbox 'Save data only', zoals in figuur 7 zichtbaar is.

De gegevens zijn nu voor verdere verwerking in andere applicaties als gecontroleerde en gevalideerde XML-data beschikbaar, zie figuur 8.

Wordt het document als geheel bewaard, dan blijven de XML Tags in het document aanwezig. Als de tekst wordt aangepast, volstaat één Save As-optie om een nieuw XML-databestand aan te maken. Is ervoor gekozen een compleet Word-document in XML-formaat te bewaren, dan zijn de toegekende tags nog steeds in het Word XML (WordML) document herkenbaar (zie figuur 8). Het schema dat gebruikt is om de velden in de tekst te markeren is als aparte namespace in de kop van het XML-document opgenomen. Je kunt een xsl gebruiken worden om de gegevens (velden) uit de WordML te filteren.

Omdat het schema bewaard blijft bij



Figuur 9

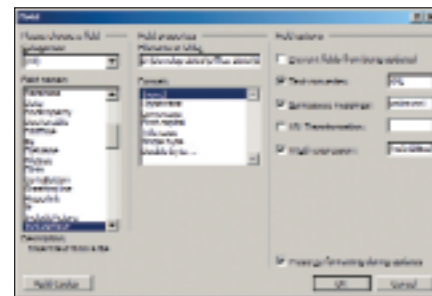
het Word-document, kun je een Word-document gaan gebruiken als flexibel invoerformulier van XML-data. Laat Word het document openen, dan opent Word ook de XML-datadefinitie. Geef de gebruiker daarna gelegenheid om de gegevens in het document te wijzigen. De XML tags blijven rondom de gegevensvelden zitten. Je hebt nu de combinatie van een ogenschijnlijk 'ongestructureerd' Word-tekstdocument, waarin gestructureerde gegevens zijn opgenomen.

Uitbreidingen van het Objectmodel

Word VBA en het Word-objectmodel zijn op allerlei plekken uitgebreid om de XML-bewerkingen als schemakoppeling, transformaties, en dergelijke onder programma (macro) besturing mogelijk te maken. Gebruik bijvoorbeeld de volgende VBA-code om een xsl-transformatie van het actieve document uit te voeren, zie listing 1.

Een nieuw leven voor Word-fields

Het omgaan met XML in Word is nog verder doorgevoerd: XML-data kunnen ook de bron vormen voor document fields. Zo kan een field in een document verwijzen naar XML-data die regelmatig wordt vernieuwd vanuit een externe databron.



Figuur 10

Voer de volgende stappen uit om een koppeling te maken vanuit een Word-field naar XML-gegevens:

1. Vanuit het Insert-menu, kies Field
2. In de Fields name lijst, kies voor IncludeText
3. In de Filename or URL, tik de naam van het XML-bestand in dat de benodigde gegevens bevat.

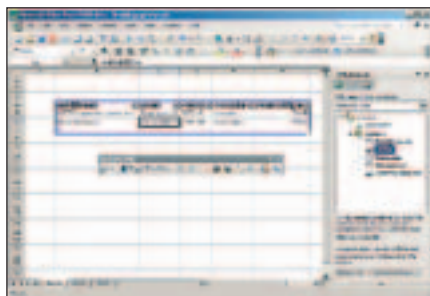
Wanneer je nu op OK klikt, dat worden de XML-data in het Word-document als Field ingevoerd. Wil je een selectie van de XML-data, voer dan aanvullend de volgende stappen uit (zie ook figuur 10):

1. Geef namespace mapping op door het aanklikken van de Namespace mappings checkbox, en type de bijbehorende namespace mappings in het bijbehorende tekstvak (dat overigens in de bèta versie van Office 2003 veel te klein is).
2. Als je nog een transformatie op de aangeleverde XML-data wilt uitvoeren, staat xsl tot je beschikking. Klik de XSL transform file checkbox op aan, en tik de naam van het xsl-bestand dat je wilt gebruiken.
3. Klik ook de XPATH expression checkbox op aan en tik de Xpath-expressie in het tekstvak ernaast.
4. Klik op OK om de geselecteerde data als veld in je document op te nemen.

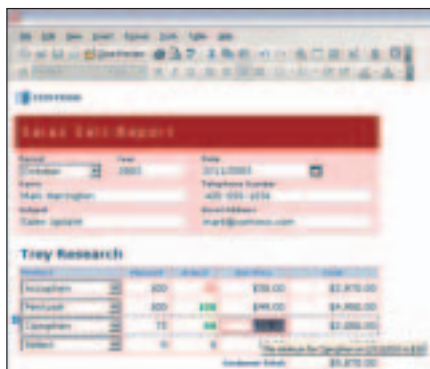
De ruimte in dit artikel is helaas onvoldoende om nog veel meer op de XML-mogelijkheden in Word in te gaan. Zo ben ik bijvoorbeeld niet ingegaan op de mogelijkheid van Word-XML om 1:n relaties in een document weer te geven, waarbij de n-kant als tabel te zien is. Geloof me, dit is het topje van de ijsberg. Met de XML-mogelijkheden van Word krijg je gereedschap in handen om gestructureerde data, ongestructureerde teksten en ongestructureerde gebruikers van Word op een controleerbare manier te laten samenwerken.

```
ActiveDocument.TransformDocument _
    Path:="D:\MyXSL\TransformMyDoc.xsl", DataOnly:=True
```

Listing 1



Figuur 11



Figuur 12

Excel 2003 XML Mogelijkheden

Ook Excel 2003 heeft een enorme stap voorwaarts gemaakt in het omgaan met XML. Excel 2003 ondersteunt XML-mogelijkheden als het tonen en benaderen van zelfgedefinieerde XML, en is niet langer beperkt tot platte Excel-geformatteerde en XML-spreadsheetgeformatteerde gegevens als in Excel 2002. Je kunt een mapping maken tussen een eigen XML-schema en het Excel-workbook om vervolgens met het taakvenster voor XML-source *drag en drop schema-elementen* in het Worksheet te krijgen, zie figuur 11.

Met wat fantasie heb je ook in Excel 2003 een XML-editor gekregen. Aangezien Excel op vrijwel elke desktop voorkomt, heeft iedere gebruiker vanaf Office 2003 een gestructureerde XML-editor! Ook hier geldt dat er in het objectmodel door de developer de nodige objecten zijn toegevoegd die het via VBA-code bewerken van de XML mogelijk maken. XmlDataBinding, XmlMap, XmlNamespace, XmlSchema en XPath-objecten, als onderdeel van de XmlMaps, XmlNamespaces en XmlSchemas collecties van Excel 2003 stellen de developer in staat spreadsheets te koppelen aan XML-datasources, XML-datastructuren te mappen op Excel-worksheetdata, XML-gegevensfragmenten te selecteren met XPath, en meer, veel meer.

XML-schemakoppeling met Word- of Excel-documenten zijn ook de basis van Smart Documents, documenten die logica ingebouwd hebben (op basis van XML-structuur) en die in het taakvenster aangeven wat een gebruiker op die plek in het document (of cel in de Excel-sheet) kan doen. Dynamische en zelf programmeerbare taakvensters (task panes) dus. Smart Documents is een apart onderwerp voor een artikel.

Nieuw in Office 2003: InfoPath

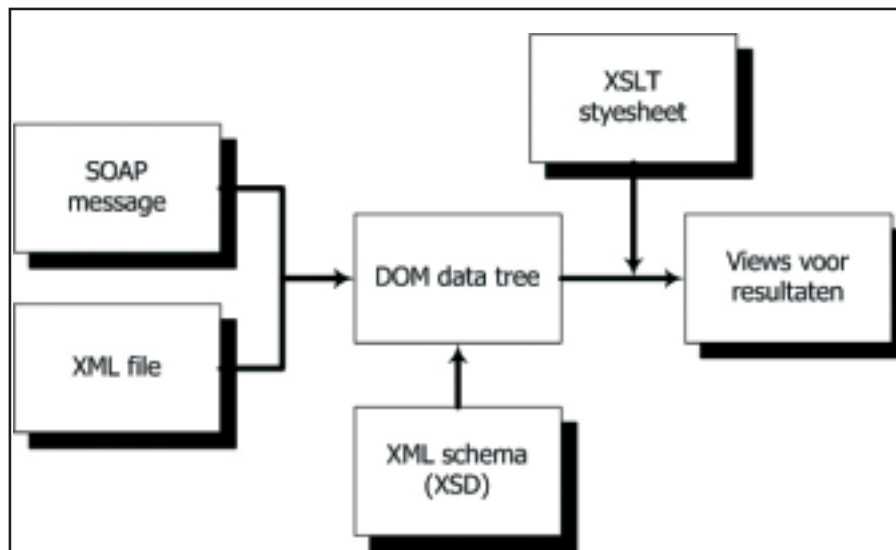
Met Excel en Word hebben we nu dus mogelijkheden gekregen om gegevens in een Excel-spreadsheet of een Word-document te verbinden met XML-gegevens. Hierdoor kunnen we eenvoudig

documenten koppelen aan XML-data. Met InfoPath is een geheel nieuwe loot aan de Office-stam toegevoegd met maar één doel voor ogen: invoerschermen maken waarmee gebruikers XML-gegevens kunnen aanmaken. Een voorbeeld van zo'n invoerscherm staat in figuur 12.

InfoPath 2003 is geheel opgebouwd rondom XML: XML als input, XML-schema's (XSD's) voor de configuratie van de data, XSLT-stylesheets voor de presentatie. InfoPath verbergt een belangrijk deel van de complexiteit van XML voor developer en gebruiker. De structuur van het werken met InfoPath is opgebouwd rondom het laden van een XML-document in het Document Object Model (DOM). Wanneer je een XML-document in InfoPath opent, wordt een representatie van dit document opgebouwd als DOM tree. Een getransformeerde weergave van deze tree wordt aan de eindgebruiker getoond; zie ook figuur 13.

De DOM-structuur wordt vervolgens door InfoPath gebruikt om de door de gebruiker geraadpleegde of ingevoerde gegevens te editen, te valideren, te bewaren en eventueel door te sturen naar een volgende applicatie. InfoPath is hiermee het Office-programma voor dynamische formulieren, en bevat daarmee ongetwijfeld functionaliteit die je normaal alleen aantreft in een formulieren/programmeeromgeving, zoals een .NET/Windows-toepassing. Of InfoPath gemakkelijker werkt dan het ontwikkelen van een invoerscherm in een .NET-programma, en je dus kiest voor InfoPath in plaats van .NET, hangt ongetwijfeld af van je vaardigheden als .NET-developer. Feit is dat InfoPath geconcentreerd is op het maken van XML vanuit gebruikersinvoer. InfoPath bevat daarmee gereedschappen die het werken in deze omgeving snel en gemakkelijk moet maken.

Een voorbeeld hiervan is het *valideren van gegevens* die door een gebruiker worden ingevoerd. Het resultaat van invoer en wijzigen in een InfoPath-document resulteert altijd in gegevens in de DOM-tree. Wanneer gegevens worden



Figuur 13

```
<salesReport>
  <customers>
    <customer>
      <price></price>
    </customer>
  </customers>
  <sales>
    <maxPrice></maxPrice>
  </sales>
</salesReport>
```

Listing 2

```
<xsf:customValidation>
  <xsf:errorCondition
    match="/salesReport/customers"
    expressionContext="customer/price"
    expression=". > ../../sales/maxPrice">
  <xsf:errorMessage
    type="modeless"
    shortMessage="Prijs hoger dan maximum">
  </xsf:errorMessage>
</xsf:errorCondition>
</xsf:customValidation>
```

Listing 3

```
<xsf:domEventHandlers>
  <xsf:domEventHandler
    handlerObject="msoxd__ZipCode"
    match="/Report/Contacts/Contact/Address/ZipCode"/>
</xsf:domEventHandlers>
```

Listing 4

```
function msoxd__ZipCode::OnBeforeChange(eventObj)
{
  // Code for the event handler function
}
```

Listing 5

ingevoerd, zijn er drie niveaus van documentvalidatie:

- Schema validation: wanneer een gebruiker gegevens invoert, komen deze in het XML-document terecht, mits de DOM-tree valide blijft zoals vastgelegd in het XML-schema.
- Declaratieve validatie met Xpath.
- Script gebaseerde validatie.

Om deze stappen te illustreren hier een voorbeeld van declaratieve validatie met XPath en scriptvalidatie.

Stel dat de gebruiker (in een XML-document) een prijs moet invullen die altijd kleiner moet zijn dan een maximumprijs die in een ander veld beschikbaar is; zie Listing 2.

Deze validatie kan met een Xpath-validatie worden opgegeven; zie Listing 3.

Ook is het mogelijk met VBScript of JScript geprogrammeerde validaties uit te laten voeren. Om dit getriggert te krijgen, moeten events worden opgevangen: nodename::OnBeforeChange(), nodename::OnValidate() en nodename::OnAfterChange(). Een voorbeeld hiervoor is gegeven in Listing 4 en Listing 5.

De behoefte aan flexibele invoerformulieren als basis voor allerlei (administratieve) processen is in de omgeving waarin we werkzaamheden verrichten zeker aanwezig. InfoPath

moet de drempel en inspanning om die formulieren te maken verkleinen en is, zoals meer en meer producten, gebaseerd op XML en rondom XML. Met de tijd gaan we ongetwijfeld een verschuiving zien van helemaal uitgeprogrammeerde invoerschermen naar flexibele formulieren die met InfoPath gemaakt zijn.

XML op de desktop

XML op de desktop. Met de komst van Office 2003 is dat meer dan ooit realiteit geworden. Gezien het feit dat er meer Office-gebruikers zijn dan developers, mag je verwachten dat er een XML-vraag ontstaat. Reken erop dat er de nodige uitdagingen voor ons IT-professionals zullen komen om het XML-gebruik in goede banen te leiden. Kennis van Office en zeker de nieuwe mogelijkheden van Office 2003 is daarbij essentieel, met als basis een gedegen kennis van XML en alle onderdelen die daarbij bruikbaar zijn. XML dus ook op onze eigen desktop.

Referenties

Algemene informatie over Office 2003 kun je uitstekend halen op: <http://www.microsoft.com/office/>

Wil je het verschil met de voorgaande Office-versie weten, kijk dan op: <http://www.microsoft.com/office/preview/editions/compare.asp>

Wil je als developer meer achtergrond over Office 2003 weten, kijk dan op: <http://msdn.microsoft.com/office>

Wil je een introductie op de XML-mogelijkheden van Office 2003, kijk dan op: http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dno2k3ta/html/odc_XMLinOffice2003_SummaryDoc.asp