

Van binaire documentformaten naar Open XML

MICROSOFT OFFICE-DOCUMENTEN OPEN VOOR ONTWIKKELAARS

Binnenkort komt Microsoft met een nieuwe versie van Office. Naast de gedaantewisseling in de gebruikersinterface van Microsoft Office 2007 is er nog een grote verandering. Het nieuwe documentformaat is volledig open en biedt tal van mogelijkheden voor ontwikkelaars, die voorheen onmogelijk waren. In dit artikel maken we kennis met het nieuwe documentformaat.

De Microsoft Office-bestandsformaten die we tot op de dag van vandaag gebruiken zijn ontwikkeld in 1994 en met Office 97 geïntroduceerd. Zo'n tien jaar terug zag de wereld er anders uit. In de oude binaire wereld werden floppy disks voornamelijk gebruikt om bestanden uit te wisselen. Het interne geheugen van de pc was in die tijd zeer beperkt, waardoor een compact formaat erg belangrijk was. De mogelijkheid om een document op te slaan was eigenlijk alleen nodig zodat je er later weer aan verder kon werken. Deze 'elektronische' versie van het document was slechts een tussenfase, het eindresultaat was een geprint document. Overheadsheets en dia's waren de belangrijkste outputvorm van PowerPoint in die tijd. Het moet meer dan 15 jaar geleden zijn, dat ik op stap ging met een diaprojector en een carrousel met dia's. De mogelijkheid om 35mm dia's te maken zit tot op vandaag nog in PowerPoint 2003. De documentformaten waren niet dusdanig opgezet dat ze gemakkelijk in een ander programma ingelezen konden worden. Een bepaald bestandsformaat hoorde een-op-een bij een applicatie. Er zijn in de loop der jaren honderden verschillende bestandsformaten ontstaan. Een aardig overzicht is te vinden op www.webopedia.com/quick_ref/fileextensions.asp. Vaak was de opbouw van het bestandsformaat gelijk aan de manier waarop de data werden opgeslagen in het interne geheugen. Het aanpassen van de inhoud van het document door een andere applicatie resulteerde vaak in een corrupt bestand.

De nieuwe open wereld

Tien jaar later ziet de computerwereld er behoorlijk anders uit. Bijna alles en iedereen is verbonden via een breedband internetverbinding waarmee we dagelijks allerlei informatie uitwisselen. Het elektronische document is tegenwoordig vaak het eindproduct, en uitprinten doen we alleen als we een document moeten controleren. Documenten worden massaal bewaard en dienen als input voor een ander persoon of proces. Er zijn in de loop der jaren bedrijven en instanties geweest die aan Microsoft de garantie gevraagd hebben of de door hun opgeslagen Office-documenten na 120 jaar nog te openen zijn. Ik weet niet of er iemand is die zo ver in de toekomst kan kijken. Aan de andere kant, heeft een mens vandaag de dag een redelijke kans heeft om 100 jaar of ouder te worden. Het is dus uiterst reëel om te eisen, dat iets wat je vandaag maakt en opslaat, in de toekomst opnieuw gebruikt kan worden. Hergebruik is waar het tegenwoordig om draait en de afmeting van het bestand is minder van belang. De oude binaire bestanden kunnen gezien worden als een container voor zowel de content als voor de presentatie. Vandaag de dag gaat het veel meer om de content. Een volgende applicatie bepaalt wel hoe de data gebruikt wordt of opnieuw gepresenteerd gaat worden.

De manier hoe mensen documenten maken en bewerken is ook door de jaren veranderd. Er wordt steeds vaker gezamenlijk aan een document gewerkt en documenten bevatten veel meer 'live' verbindingen met andere documenten en gegevensbronnen dan vroeger. Door de opmars van het internet zijn er in Office wel formaten bij gekomen zoals html en rtf, maar die hebben nooit de plaats van het binaire formaat overgenomen. Met Office 2000 introduceerde Microsoft een html-bestandsformaat met document-properties in xml. Met elke nieuwe versie van Office is de ondersteuning om gegevens te exporteren naar xml uitgebreid; zie afbeelding 1. Met Office 2007 introduceert Microsoft een nieuw bestandsformaat voor zijn Office-toepassingen: Open XML. Dit wordt tevens het standaard (default) bestandsformaat en het is volledig gebaseerd op xml.

Open XML

De klanten van Microsoft hebben in de afgelopen jaren miljarden Office-documenten gemaakt. Het is dus belangrijk dat de overgang soepel verloopt en dat er geen informatie verloren gaat. De nieuwe formaten zijn met compatibiliteit en performance in het achterhoofd ontwikkeld. De nieuwe formaten zijn dan ook compatibel met de huidige formaten. Gebruikers raken door het nieuwe formaat geen informatie kwijt en krijgen geen andere opmaak. De nieuwe formaten zijn daarentegen helemaal opnieuw opgezet en bieden uitbreidingsmogelijkheden voor allerlei scenario's. Het nieuwe documentformaat voor Office 2007 is gebaseerd op xml en zip-formaat en heet 'Open XML'.

Het zip-formaat brengt verscheidene voordelen met zich mee. Als eerste natuurlijk de compressie van alle inhoud, en alles wordt als losse bestanden opgeslagen. Het is dan ook vrij eenvoudig een enkel element - bijvoorbeeld een foto - te vervangen zonder het



Afbeelding 1. Evolutie van bestandsformaten

Microsoft Office Word 2007	
.docx	Word-document
.docm	Word-document met macro's
.dotx	Word-template
.dotm	Word-template met macro's
Microsoft Office Excel 2007	
.xlsx	Excel-werkboek
.xlsm	Excel-werkboek met macro's
.xltx	Excel-template
.xltm	Excel-template met macro's
.xlam	Excel add-in
.xlsb	Excel-binary werkboek
Microsoft Office PowerPoint 2007	
.pptx	PowerPoint-presentatie
.pptm	PowerPoint-presentatie met macro's
.potx	PowerPoint-template
.pptm	PowerPoint-template met macro's
.ppsx	PowerPoint-show
.ppsm	PowerPoint-show met macro's

Tabel 1. Overzicht met nieuwe bestandsextensies

hele document open te breken of mogelijk te beschadigen. Als laatste is de kans op bestandscorruptie kleiner. Mocht een onderdeel toch corrupt raken, dan blijven de andere onderdelen bereikbaar. Een foto wordt in zijn eigen native formaat opgeslagen. Elk Office-document is dus eigenlijk een zip-file. Toch worden de bekende Office-bestandsextensies gebruikt, maar dan wel met de toevoeging van een letter. Zie tabel 1 voor een overzicht van de nieuwe bestandsextensies. Omdat documenten met macro's als een potentieel gevaar worden gezien, is er duidelijk onderscheid gemaakt tussen documenten met macro's/vba-code en documenten zonder. In een docx kunnen geen macro's zitten. Wanneer in de package van een docx-bestand toch macro's geplaatst worden, zal Word dit bestand niet openen en als corrupt beschouwen. Het is hierdoor makkelijker om bestanden met embedded code te herkennen, te weren of om met behulp van een klein programmaatje de embedded code te verwijderen.

Microsoft heeft de specificatie van Open XML uit handen gegeven aan Ecma International. Ecma beheert verschillende standaarden, te weten: ECMAScript (Jscript/JavaScript), DVD+RW, DVD-RW en natuurlijk C#. Op dit moment is de Technical Committee (TC45) bezig met het standaardisatieproces. Draft 1.3 van de specificatie is te downloaden van de Ecma-site. Het is al zeer compleet en alles is gedetailleerd beschreven. Denk even na voordat je hem uitprint, want het zijn namelijk 4081 pagina's. De voorzitter van TC45 is niemand minder dan Jean Paoli (Microsoft). Hij is een van de architecten van de xml-standaard. Zijn naam prijkt als auteur op de voorkant van de W3C XML 1.0-specificatie; zie het interview met Jean Paoli in het .NET Magazine #7. Andere leden van Ecma TC45 komen van: Apple, Barclays Capital, BP, The British Library, Essilor, Intel, NextPage, Novell, Statoil, Toshiba en The United States Library of Congress.

Iedereen die gebruik wenst te maken van Open XML kan dat doen, want het is 100% royalty free. Voor Open XML heeft Microsoft een speciale licentie 'Covenant Not to Sue'. Deze verklaring is bovendien 'non revocable' – het kan niet ongedaan worden gemaakt – en het is 'Non Discriminatory' wat betekent dat concurrenten van Microsoft het ook zonder enig bezwaar en zonder restricties of kosten kunnen toepassen. Dus iedereen die gebruik wenst te maken van het Open XML-formaat kan dat doen, er hoeft niets getekend te worden.

Interoperabiliteit

Het Open XML-formaat is vanaf Office 2007 het standaard documentformaat voor Word, Excel en PowerPoint. Vanzelfsprekend lezen en schrijven de Office-applicaties nog steeds de oude doc-,

Even een zijstapje. Hoe zit het met het andere documentformaten als OpenDocument Format, XML Paper Specification en PDF?

OpenDocument Format (ODF) is het formaat voor OpenOffice van Sun. ODF is ook op xml gebaseerd, toch zijn de beide formaten niet compatibel. ODF is ook ondergebracht bij een standaardisatiecommissie, OASIS. Er is een open source-project gestart dat als doel heeft translators te maken. Op basis van de specificatie van Ecma en OASIS worden deze translators gemaakt. Op dit moment is er een prototype translator beschikbaar die als add-in voor Microsoft Office Word 2007 kan worden geïnstalleerd. Hiermee is het mogelijk ODF-documenten te openen en op te slaan vanuit Word. Het open source-project heeft ook plannen om een add-in voor Excel en PowerPoint te maken. Tot nu toe zijn er vier sponsors van dit open source-project en één daarvan is Microsoft. XML Paper Specification (XPS) is recent door Microsoft aangekondigd als onderdeel van Windows Vista. Dit is een paginageoriënteerd formaat (fixed lay-out), dat ook xml gebruikt om de inhoud te beschrijven. Het maakt net als het Open XML-formaat gebruik van de Open Packaging Conventions (zip-formaat). Ondersteuning voor XPS komt ook als add-in voor Office beschikbaar. De XML Paper Specification wordt ook royalty free aangeboden aan ontwikkelaars.

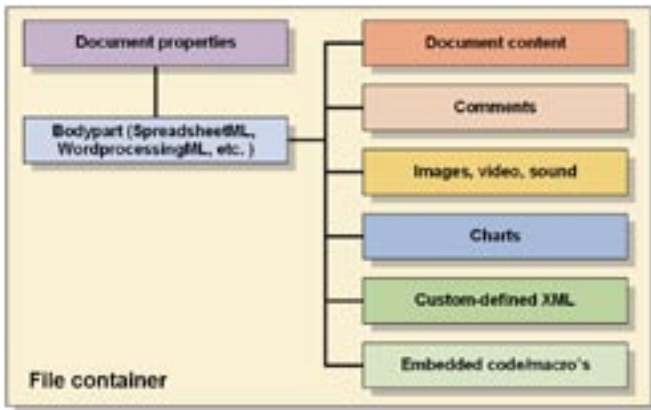
Portable Document Format (PDF) staat hoog op het verlanglijstje van gebruikers. PDF is door Adobe ontwikkeld en is inmiddels een iso-standaard. Sinds jaren stromen de verzoekjes bij Microsoft - meer dan 120.000 per maand(!) - binnen om 'save as PDF' te ondersteunen. Een van de grote voordelen van PDF is de fixed lay-out en dat het breed geaccepteerd is. Het goede nieuws is dat er voor PDF ook een add-in zal komen. Alle Office-applicaties (Word, PowerPoint, Excel, Access, InfoPath, Publisher, Visio, Project en OneNote) kunnen het document als PDF opslaan. De PDF's gemaakt door Office ondersteunen live hyperlinks, cross-document links, bookmarks, enzovoort. Publisher 2007 gaat een stapje verder en ondersteunt specifieke zaken voor pre-press, zoals het CMYK-kleurenprofiel en printing page marks.

xls-, ppt- bestanden. Wat misschien niet vanzelfsprekend is, is dat Microsoft het Open XML-formaat ook beschikbaar stelt aan gebruikers van Office 2000, XP en 2003. Dit gaat via een add-in, waarvan al een bètaversie beschikbaar is; zie de lijst met referenties voor de url. Open XML voldoet aan alle eisen die de overheid oplegt met betrekking tot standaarden voor het vastleggen van overheidsdocumentatie.

De structuur van Open XML

Een Open XML-bestand is modulair opgebouwd en alle onderdelen - parts genaamd - zijn geplaatst in een zip-container. Een part is in essentie een bestand. Alle gezamenlijke parts vormen het document. De meeste parts zijn xml-bestanden die de applicatiedata, metadata en gebruikersdata beschrijven. De container kan ook niet-xml-bestanden bevatten zoals afbeeldingen en OLE-objecten. Deze bestanden worden in hun eigen formaat - bijvoorbeeld jpg voor een afbeelding - in de container geplaatst. Parts kunnen wijzigingen bevatten naar andere parts. Omdat de container een zip-bestand is worden alle parts individueel gecomprimeerd. Het voordeel van het zip-formaat is dat eindgebruikers slechts één bestand zien. Ze kunnen op dezelfde manier met het Open XML-document werken zoals ze gewend zijn met de oude binaire formaten. Als ontwikkelaar kun je met standaard tools de zip-container openen en de individuele parts benaderen en aanpassen. Je kunt bijvoorbeeld in een Excel-bestand de juiste sheet opzoeken en daarin een waarde muteren.

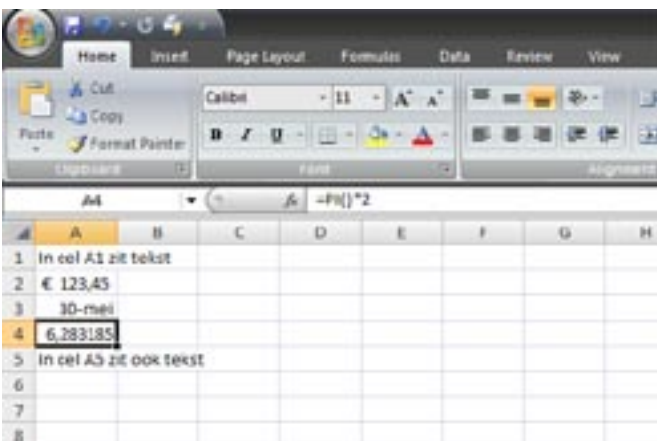
Het Open XML-bestandsformaat van Excel heet *SpreadsheetML*. Het is geïntroduceerd in Office 2002, echter, toen was het nog niet mogelijk alle onderdelen in een spreadsheet te beschrijven. Nu is dat wel het geval. Het formaat voor tekstdocumenten heet *Word-processingML* en is in Word 2003 geïntroduceerd. Ook toen waren er nog beperkingen. Binaire bestanden zoals afbeeldingen werden ook in XML beschreven. Dit maakte het bestand log en lastig te



Afbeelding 2. Structuur van een Open XML-bestand

bewerken. Voor PowerPoint en tekeningen is er nu ook een formaat *PresentationML* en *DrawingML*. Er gaan veel elementen in een Office-document. Een deel daarvan zit in alle Office-documenten zoals: document-properties, styles, charts, hyperlinks, commentaar, annotation. Andere elementen zijn specifiek voor een bepaalde applicatie zoals een worksheet in Excel en slides in PowerPoint. Zie afbeelding 2 voor de structuur van een Open XML-bestand.

Laten we naar een voorbeeld van Excel kijken. Ik heb een sheet gemaakt met een aantal waarden; zie afbeelding 3. Wanneer we dit opslaan als xlsx en vervolgens de container openbreken, dan zie je hoe de structuur er daadwerkelijk uitziet; zie afbeelding 4. De file [Content_Types].xml in de root van de container bevat een opsomming van alle parts. Omdat het hier om een Excel-document gaat, is de body in dit geval 'xl'. In de folder 'xl' worden alle parts opgeslagen die de inhoud van het document bepalen. Wanneer je te maken hebt met een Word-document is de body 'word', en voor een PowerPoint-presentatie 'ppt'. In de map xlworksheets is de worksheet sheet1.xls te vinden. Hierin zit de inhoud van de verschillende cellen. Na wat puzzelen herken je de verschillende waarden. Wat niet lukt, is om de tekst in cel A1 en A5 te vinden. Alle strings worden gezamenlijk opgeslagen in sharedStrings.xml; zie codevoorbeeld 2. Het attribuut t="s" bij cel A1 en A5 geeft aan dat deze cel een string bevat en de waarde <v>0</v> is het indexnummer naar de string in sharedStrings.xml. Het is een zero-based index. Op het eerste gezicht ziet dit er omslachtig uit. Omwille van performance – bij de eindgebruikers – is voor een stringtable gekozen. De stringtable bespaart veel ruimte, omdat in een spreadsheet traditioneel veelvuldig dezelfde tekstwaardes gebruikt worden. Een bijkomstigheid is dat het nu eenvoudig is om de stringtable te vertalen. In cel A4 is te zien dat zowel de formule (pi * 2) als het resultaat van de formule opgeslagen worden. De formule wordt gewoon in

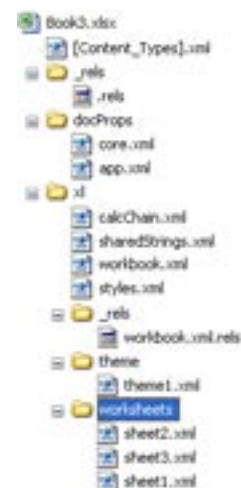


Afbeelding 3. Excel-spreadsheet

tekst weergegeven net zoals deze is ingegeven in Excel. In dit geval is de berekening eenvoudig, maar je moet er niet aan denken dat je zelf alle formules moet doorrekenen, daar heb je immers Excel voor. Iets lastiger is het datumveld dat opgeslagen wordt als serieel getal. Dit is het interne datumformaat van Excel. De waarde 1 staat voor 1 jan 1900 en loopt tot 31 dec 9999. De waarde in cel A3 is 38867 en geeft het aantal dagen na 1 jan 1900 aan. Naast parts die content bevatten, zijn er ook 'Relationship-items'. Relationship-items specificeren hoe alle document-parts samen het document vormen. Relationships worden ook opgeslagen in xml-parts en je treft ze altijd aan in de folder met de naam '_rels'. In de relationships staat bijvoorbeeld welke afbeelding op welke slide voorkomt. Elke part met content is tenminste eenmaal vermeld in één van de relationship-bestanden. In een part wordt gebruikgemaakt van een relationship ID. Er wordt dus niet rechtstreeks naar een andere part verwezen, maar via een relationship ID. Wat hieraan handig is, is dat je met behulp van het relationship-item kunt controleren hoeveel werkboeken een Excel-bestand bevat of wat voor type een bepaalde part is. Deze relationships zijn niet geheel onbelangrijk. Je kunt straffeloos document-parts verplaatsen en toevoegen, maar de relationships moeten wel mee geüpdate worden. De folder \docProps bevat parts met properties. Dit zijn App.xml met applicatiespecifieke properties en Core.xml met gezamenlijke properties die gelden voor alle bestanden die gebaseerd zijn op het Open Packaging Conventions documentformaat. In de folder \docProps tref je ook nog thumbnails aan als je daar tenminste gebruik van maakt bij het opslaan van het document. Als laatste worden in \docProps ook digital signatures opgeslagen. Ten slotte bevat xl\theme\theme1.xls de opmaak van de spreadsheet. Zoals je kunt zien is het een duidelijk formaat en heeft alles een logische plaats. De spreadsheetdata zijn opgeslagen in rijen en kolommen, en de opmaak en content zijn gescheiden. Het is dus vrij eenvoudig om bijvoorbeeld de opmaak van het document aan te passen. Veel van de parts die je in afbeelding 4 kunt zien zijn optioneel. Wanneer je zelf een xlsx aanmaakt, dan is een aantal parts verplicht. Minimaal moet de [Content_Types].xml bekend zijn en de relationship-file_rels_rels moet aanwezig zijn; en natuurlijk een bestand met de inhoud.

Development

De eenvoudigste manier om een Open XML-bestand te creëren of te bewerken is via PackagePart-class (System.IO.Packaging), onderdeel van WinFX. Je hebt al kennigemaakt met SpreadsheetML. In codevoorbeeld 3 is te zien hoe vanaf nul een docx gemaakt wordt. Als eerste wordt de xml-structuur van het document opgezet; zie codevoorbeeld 4 voor het resultaat. Vervolgens wordt de package gemaakt en wordt het part 'word\document.xml' met de inhoud wegschreven. Dan wordt het xml-bestand met de relaties opgeslagen en ten slotte wordt de package-container gesloten. Het resultaat is



Afbeelding 4. Structuur van het Excel-bestand

```

<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<worksheet xmlns="http://schemas.openxmlformats.org/spreadsheetml/
2006/5/main"
xmlns:r="http://schemas.openxmlformats.org/officeDocument/2006/
relationships">
<dimension ref="A1:A4" />
<sheetViews>
<sheetView tabSelected="1" workbookViewId="0">
<selection activeCell="A2" sqref="A2" />
</sheetView>
</sheetViews>
<sheetFormatPr defaultRowHeight="15" />
<cols>
<col min="1" max="1" width="10.28515625" customWidth="1" />
</cols>
<sheetData>
<row r="1" spans="1:1">
<c r="A1" t="s">
<v>0</v>
</c>
</row>
<row r="2" spans="1:1">
<c r="A2" s="2">
<v>123.45</v>
</c>
</row>
<row r="3" spans="1:1">
<c r="A3" s="1">
<v>38867</v>
</c>
</row>
<row r="4" spans="1:1">
<c r="A4">
<f>PI()*2</f>
<v>6.2831853071795862</v>
</c>
</row>
<row r="5" spans="1:1">
<c r="A5" t="s">
<v>1</v>
</c>
</row>
</sheetData>
<printOptions />
<pageMargins left="0.7" right="0.7" top="0.75" bottom="0.75"
header="0.3" footer="0.3" />
<headerFooter />
</worksheet>

```

Codevoorbeeld 1.

```

<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<sst xmlns="http://schemas.openxmlformats.org/spreadsheetml/2006/5/
main"
count="2" uniqueCount="2">
<si>
<t>In cel A1 zit tekst</t>
</si>
<si>
<t>In cel A5 zit ook tekst</t>
</si>
</sst>

```

Codevoorbeeld 2.

```

using System;
using System.IO;
using System.IO.Packaging;
using System.Xml;

public class CreateDOCX
{
    static void Main()
    {
        string WordprocessingML = "http://schemas.openxmlformats.org/word
processingml/2006/3/main";
        string fileName = "myfirst.docx";
        string bodyText = "Hello World";

        // Set up the nested structure ...
        XmlDocument xmlStartPart = new XmlDocument();
        XmlElement tagDocument = xmlStartPart.CreateElement("w:document",
WordprocessingML);
        xmlStartPart.AppendChild(tagDocument);
        XmlElement tagBody = xmlStartPart.CreateElement("w:body",
WordprocessingML);
        tagDocument.AppendChild(tagBody);
        XmlElement tagParagraph = xmlStartPart.CreateElement("w:p",
WordprocessingML);
        tagBody.AppendChild(tagParagraph);
        XmlElement tagRun = xmlStartPart.CreateElement("w:r",
WordprocessingML);
        tagParagraph.AppendChild(tagRun);
        XmlElement tagText = xmlStartPart.CreateElement("w:t",
WordprocessingML);
        tagRun.AppendChild(tagText);
        XmlNode nodeText = xmlStartPart.CreateNode(XmlNodeType.Text,
"w:t", WordprocessingML);
        nodeText.Value = bodyText;
        tagText.AppendChild(nodeText);

        // Create a new package (Open XML document) ...
        Package pkgOutputDoc = null;
        pkgOutputDoc = Package.Open(fileName, FileMode.Create,
FileAccess.ReadWrite);

        // Save the main document part (document.xml) ...
        Uri uri = new Uri("/word/document.xml", UriKind.Relative);
        PackagePart partDocumentXML = pkgOutputDoc.CreatePart(uri,
"application/vnd.openxmlformats-officedocument.wordprocessingml.
document.main+xml");
        StreamWriter streamStartPart = new StreamWriter
(partDocumentXML.GetStream(FileMode.Create,
FileAccess.Write));
        xmlStartPart.Save(streamStartPart);
        streamStartPart.Close();
        pkgOutputDoc.Flush();

        // Create the relationship part, close the document ...
        pkgOutputDoc.CreateRelationship(uri, TargetMode.Internal,
"http://schemas.openxmlformats.org/officeDocument/2006/relation
ships/officeDocument", "rId1");
        pkgOutputDoc.Flush();
        pkgOutputDoc.Close();
    }
}

```

Codevoorbeeld 3.

een zip-container genaamd myFirst.docx met de volgende items:

- een indexbestand [Content_Types].xml
- een relationships file \rels\rels
- een part met de content \word\document.xml.

Op de MSDN-site is nog een aantal code-snipjets te vinden. Microsoft heeft als doel om ongeveer 50 verschillende samples beschikbaar te hebben tegen het einde van het jaar. Daarnaast is er een validation-library aangekondigd waarmee je een controle kunt

```

<?xml version="1.0" encoding="utf-8" ?>
<w:document xmlns:w="http://schemas.openxmlformats.org/wordproce
singml/2006/3/main">
  <w:body>
    <w:p>
      <w:r>
        <w:t>Hello World</w:t>
      </w:r>
    </w:p>
  </w:body>
</w:document>

```

Codevoorbeeld 4.

uitvoeren of het bestand nog steeds valide is na jouw mutaties. Een andere aardige tool is PackageExplorer, geschreven door Wouter van Vugt die werkzaam is bij Info Support. Hiermee is het mogelijk om op eenvoudige wijze de hele xml-structuur van een Open XML-document te bekijken. De tool inclusief de source-code is te vinden op codeplex. In theorie heb je aan Notepad voldoende om Open XML-documenten te maken en te manipuleren.

Echt open

Dat Open XML daadwerkelijk open is, bewijzen de ontwikkelaars van Novell. De open source spreadsheet genaamd Gnumeric is nu al in staat (hoewel nog op beperkte wijze) om een spreadsheet in het Open XML-formaat te openen en op te slaan. Nu het bestandsformaat eindelijk open is en alle uitbreidingen die Microsoft in de afgelopen jaren in Office heeft gestopt, is Office een echt development-platform geworden. De structuur van Open XML is zo duidelijk dat je al vrij snel begrijpt hoe het in elkaar steekt. Met een paar regels code of standaard tools trek je het bestand open en kan het aanpassen beginnen. Je kunt bijvoorbeeld heel eenvoudig een copyrightvermelding toevoegen aan elk document, of controleren of alle velden ingevuld zijn, of een PowerPoint-presentatie in een klap in de huisstijl aanpassen. Dankzij de eenvoud van Open XML is het mogelijk op de server met een service Office-bestanden te genereren. Hiervoor is geen (niet ondersteunde) Office-installatie op de server meer nodig. Bovendien is het snel. In een klein testje waar de opmaak van een aantal grote documenten werd aangepast, bleven de verwerkingstijden per document onder de seconde. In dezelfde tijd kun je niet eens Word opstarten en het document openen. Kortom, de mogelijkheden zijn ongekend. OpenXML is zo open dat er oplossingen voor vele nieuwe scenario's gebouwd kunnen worden.

Robert Fransen is freelance consultant met zeer brede ICT-kennis en -ervaring. Zijn eerste regels code zijn geschreven in machinecode voor een zelfgebouwde computer met 6502-processor. Tegenwoordig is hij is parttime werkzaam bij Microsoft Nederland in het webteam, verantwoordelijk voor onder andere de MSDN-website. Zijn e-mailadres is robert.fransen@center.nl

Referenties

Algemene informatie: www.microsoft.com/office/xml
 Developers forum: www.openxmldeveloper.org
 Open XML walkthrough: <http://msdn2.microsoft.com/en-us/library/ms771890.aspx>
 Excel team blog: <http://blogs.msdn.com/excel>
 PackageExplorer: www.codeplex.com/Wiki/View.aspx?ProjectName=PackageExplorer
 Open XML voor Office 2003: www.microsoft.com/office/preview/beta/converter.aspx
 ODF add-in voor Office: <http://sourceforge.net/projects/odf-converter>
 Ecma TC 45: www.ecma-international.org/memento/TC45.htm
 Blog van Office productmanager Brian Jones: http://blogs.msdn.com/brian_jones
 Sample-code: www.microsoft.com/downloads/details.aspx?FamilyID=8d46c01f-e3f6-4069-869d-90b8b096b556&displaylang=en