

De laatste loodjes, nu ook in XML

INTRODUCTIE TOT HET MAKEN VAN INSTALLATIEPROGRAMMA'S MET WIX, DE WINDOWS INSTALLER XML TOOLSET

Met WiX, de Windows Installer XML toolset, kun je installatieprogramma's maken die gebruik maken van de Microsoft Windows Installer technologie. De bronbestanden van een WiX-project zijn XML-documenten, waardoor deze toolset heel wat mogelijkheden biedt aan softwareontwikkelaars. Dit artikel legt uit hoe je WiX kunt gebruiken om installatieprogramma's te maken voor .NET-applicaties en belicht enkele voordelen van de Windows Installer XML toolset.

De tijd dat een applicatie enkel bestond uit een alleenstaand uitvoerbaar bestand is al lang voorbij. Bijna alle hedendaagse applicaties hangen af van verschillende uiteenlopende bronnen, zoals andere softwarecomponenten, configuratiebestanden, de registry, snelkoppelingen, en dergelijke. Om te vermijden dat eindgebruikers al deze zaken zelf manueel moeten instellen - u koopt uw wagen toch ook niet als zelfbouwpakket? - zijn ontwikkelaars installatiesoftware gaan maken. Tot op heden loopt er echter nog veel fout bij de installatie van software. Het is ook niet gemakkelijk om een universeel installatiepakket te maken dat in alle omstandigheden goed werkt. Zodra de software vertrekt uit het ontwikkelteam komt deze terecht in een zogehete 'vijandige omgeving'. Dat wil zeggen dat men nooit weet welke software reeds op de computer geïnstalleerd staat. Zo houden niet alle installatieprogramma's rekening met de reeds geïnstalleerde versies van bepaalde bestanden, halen ze niet alle informatie van de applicatie weg bij het verwijderen van software, enzovoort.

Daarom heeft Microsoft de Windows Installer technologie ontwikkeld. Deze technologie biedt een aantal basisfunctionaliteiten voor het installeren van software en is zelfs geïntegreerd in de laatste besturingssystemen van Microsoft. Bovendien helpt de Windows Installer service ook bij het opvolgen van essentiële regels voor het installeren van software. Standaard biedt Windows Installer onder andere de volgende functionaliteiten:

- transactionele installatie met rollback (automatisch verwijderen van reeds geïnstalleerde onderdelen wanneer de installatie een fout tegenkomt).
- zelfherstellende applicaties wanneer een onderdeel (al dan niet opzettelijk) verwijderd werd.
- advertising, waardoor installatie-op-aanvraag mogelijk wordt (zoals het installeren van software wanneer deze voor de eerste maal gebruikt wordt).
- gemakkelijk configureerbaar en aanpasbaar indien nodig.
- herstellen en verwijderen van de geïnstalleerde software.
- standaard API voor controle op geïnstalleerde software.
- en nog veel meer.

En mocht de standaardfunctionaliteit onvoldoende zijn voor de wensen van de ontwikkelaar, dan kan deze nog altijd bijkomende mogelijkheden programmeren via externe bibliotheken die aanroepbaar zijn vanuit de installatie. Helaas bood Microsoft tot voor kort enkel een API aan en enkele rudimentaire tools om met Windows Installer databases te werken. Met de komst van de Windows Installer XML toolset (WiX) verandert dit. Maar voordat we dieper ingaan op Windows Installer en WiX in het bijzonder, is het best om te weten met welk soort bestanden de Windows Installer technologie werkt.

Windows Installer database (.MSI). Een MSI is inderdaad een database waar alle informatie in tabellen wordt bewaard. Elke tabel bevat een aantal velden waarvan één een sleutelveld (primary key) is en andere velden verwijzingen (foreign keys) kunnen zijn naar records in andere tabellen of gewoon informatie kunnen bevatten. Ook de verschillende schermen die getoond worden tijdens de installatie, de te installeren bestanden, registry keys, snelkoppelingen en dergelijke zijn allemaal beschreven als records in verschillende tabellen. Merge modules (.MSM). Omdat sommige bestanden geregeld geïnstalleerd moeten worden door verschillende applicaties, biedt Windows Installer de mogelijkheid deze bestanden afzonderlijk te verpakken. Een merge module is een subset van een Windows Installer database en bevat alle nodige logica om een bepaald onderdeel te installeren. Zo zijn de Microsoft XML Core Services (MSXML 4.0) en de runtime van het oude Visual Basic 6.0 beschikbaar als merge modules. Een bedrijf dat een basisframework heeft voor al zijn applicaties, zou van dit basisframework een merge module kunnen maken. Zo wordt vermeden dat voor elk installatieprogramma dezelfde bestanden en instellingen moeten worden toegevoegd, met een verhoogde kans op fouten. Merge modules kunnen niet afzonderlijk worden geïnstalleerd; ze moeten steeds gebruikt worden vanuit een Windows Installer database.

Transforms (.MST). Transforms staan toe een Windows Installer database te wijzigen zonder het fysieke bestand te wijzigen.

Patch (.MSP). Omdat niet alle software vanaf de eerste publieke release volledig zonder fouten is, laat Windows Installer ook toe om patches, kleine updates aan de bestaande software, te installeren.

In dit artikel zal ik enkel dieper ingaan op het maken van Windows Installer databases (MSI-bestanden), hoewel WiX ook merge modules kan aanmaken. Op het ogenblik van schrijven worden transforms nog niet ondersteund door WiX.

De achtergrond van WiX

Verschiede groepen binnen Microsoft gebruiken de Windows Installer XML toolset intern zelf al gedurende enige tijd, maar begin april 2004 werd de toolset voor het eerst publiek gemaakt. En dan nog wel als open source project op SourceForge, een belangrijke website voor open source softwareontwikkeling; zie de nuttige internetadressen om WiX, als applicatie of de source code, te downloaden. WiX is voor het grootste deel geschreven in Visual C# 2003, maar enkele ondersteunende bibliotheken zijn ontwikkeld in Visual C++ 2003. De toolset bestaat uit een aantal

commandline-interface applicaties. Candle is een compiler die er voor zorgt dat de WiX-bronbestanden worden omgezet naar objectbestanden. Light is een linker die referenties tussen verschillende objecten gaat koppelen en er uiteindelijk ook voor zorgt dat alle informatie en bestanden worden gebundeld in een Windows Installer database. Verder kan je met Dark een bestaande MSI database 'decompileren' naar een WiX-bronbestand. Tallow helpt de auteur van WiX-bronbestanden met een aantal zaken, door bijvoorbeeld een bestaande folderstructuur automatisch naar WiX-broncode om te zetten. Een laatste tool, Lit, kan gebruikt worden om bibliotheken van objecten samen te stellen die later door Light kunnen worden gebruikt.

Windows Installer database

Om de structuur van de WiX-bronbestanden te kunnen begrijpen is het belangrijk om enkele basisconcepten goed te begrijpen. Deze zijn specifiek gebonden aan Windows Installer en daardoor ook aan alle software die helpt bij het aanmaken van Windows Installer databases.

Product. Een product is een bepaalde versie van een applicatie in een specifieke taal. Zo zijn 'Microsoft Office 2003 Professional Edition (Engels)' en 'Microsoft Office 2003 Standard Edition (Nederlands)' twee verschillende producten. Windows Installer identificeert een product aan de hand van een GUID (Globally Unique Identifier), namelijk de Product Code.

Package. Een package identificeert de Windows Installer database zelf. De identificatie van een package gebeurt ook door een GUID, namelijk de Package Code. Deze GUID moet verschillend zijn van andere packages in het geval er ook maar één byte verschil is tussen de twee Windows Installer databases. Mocht dit niet het geval zijn, dan loop je het risico dat Windows Installer de verkeerde database gaat gebruiken voor installatie, herstellen of verwijderen van de applicatie.

Feature. Een feature is een deel van de functionaliteit van de applicatie. De gebruiker kan zelf bepalen welke features van de applicatie hij installeert. Een voorbeeld van verschillende features kan je terugvinden in 'Microsoft Office 2003' waarbij een gebruiker kan bepalen welke applicaties hij wel en welke hij niet wil installeren. Een feature kan ook een aantal onderliggende features hebben; bijvoorbeeld je hoofdapplicatie kan de mogelijkheid bieden om al dan niet een spellingcontrole of voorbeeldbestanden te installeren.

Component. Een component is een deel van de applicatie dat geïnstalleerd moet worden en kan bestaan uit één bestand, verscheidene bestanden, registry-informatie, een snelkoppeling of een combinatie hiervan. In de meeste gevallen zal een component slechts uit één bestand bestaan, hoewel dit absoluut geen vereiste is. Een component kan ook worden geïdentificeerd door middel van een GUID, maar dit is niet verplicht. Als er geen GUID is toegekend aan een component, zal Windows Installer deze na de installatie niet meer automatisch verwijderen. Components zijn tegelijk zeer belangrijk in Windows Installer, maar ook vrij complex. Daarom raad ik je zeker aan om meer over components te lezen in de Windows Installer SDK.

Een eerste WiX-bestand

In het codevoorbeeld wordt getoond hoe de WiX-broncode voor een eenvoudige installatie er uit kan zien. In dit voorbeeld wordt één applicatie geïnstalleerd in de Program Files folder, onder subfolder 'HelloWiXWereld'. Naast de executable wordt ook de C#-broncode van de applicatie in de folder weggeschreven. Verder wordt er ook een snelkoppeling vanuit de Start-menu gemaakt naar de applicatie. Elke WiX-broncode moet 'Wix' als root-element met een vast bepaalde namespace definiëren. Binnen dit element bevindt zich alle informatie voor een bepaalde installatie. Allereerst wordt nu het 'Pro-

duct'-element gedefinieerd; zo weet WiX dat het eindresultaat een Windows Installer database (dus een .MSI) moet zijn. De verschillende attributen van dit element bepalen welk product in deze installatie zit. Eén opmerking over het 'Version'-attribuut: dit geeft de versie van het product aan en de eerste twee delen van de versie moeten beide kleiner zijn dan 256. Alle volgende elementen zullen binnen het 'Product'-element worden gedefinieerd. Als eerste komen we in het codevoorbeeld het 'Package'-element tegen. De attributen van dit element bepalen wat de gebruiker zal zien wanneer hij de eigenschappen van de Windows Installer database opvraagt. Het 'Upgrade'-element zorgt er voor dat Windows Installer kan weten wanneer we een bepaalde versie van een applicatie zullen upgraden. De 'Id' is ook een GUID en wordt de Upgrade Code genoemd. Wanneer een upgrade van de applicatie wordt uitgegeven, kan aan de hand van deze Upgrade Code opgegeven worden welke geïnstalleerde versies moeten worden bijgewerkt. In het voorbeeld is het de eerste versie van de applicatie, dus wordt er enkel een Upgrade Code opgegeven om later te gebruiken als een nieuwe versie zou gelanceerd worden. Een Windows Installer database kan ook verdeeld worden via media

```
<?xml version='1.0'?>
<Wix xmlns='http://schemas.microsoft.com/wix/2003/01/wi*'
  <Product Id='78F91F3A-BA56-4419-9D07-72585329A357'
    Name='.NET Magazine #6 - WiX voorbeeldcode'
    Language='1033' Version='1.0.0.0'
    Manufacturer='Tom Mertens'>

  <Package Id='364CC1BA-6C14-4e17-92F5-BCD0B042E178'
    Description='Voorbeeldcode WiX in .NET Magazine #6'
    Manufacturer='Tom Mertens' InstallerVersion='200'
    Compressed='yes' />

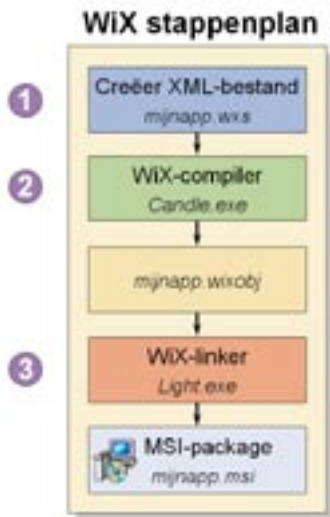
  <Media Id='1' EmbedCab='yes' Cabinet='WixSampl.cab' />

  <Directory Id='TARGETDIR' Name='SourceDir'>
    <Directory Id='ProgramFilesFolder' Name='ProgramF'
      LongName='Program Files'>
      <Directory Id='WiXSamples' Name='WiXSampl'
        LongName='.NET Magazine #6 - WiX Voorbeelden'>
        <Component Id='Shortcut'
          Guid='33B841E8-E6A0-4f79-93E1-B563F7846211'>
          <Shortcut Id='Program shortcut'
            Directory='ProgramMenuFolder'
            Name='Samples' LongName='WiX Voorbeelden'
            Target='Default' Show='normal'
            WorkingDirectory='TARGETDIR' />
        </Component>
        <Component Id='Sample1'
          Guid='26C0460B-705C-4597-B359-498A6A4D4D9A'>
          <File Id='Demo1' Name='demo.wxs'
            LongName='demo.wxs' DiskId='1' src='demo.wxs' />
        </Component>
      </Directory>
    </Directory>
    <Directory Id='ProgramMenuFolder' Name='ProgMenu'>
  </Directory>

  <Feature Id='Default' Title='Alle voorbeelden' Level='1'>
    <ComponentRef Id='Shortcut' />
    <ComponentRef Id='Sample1' />
  </Feature>

</Product>
</Wix>
```

listing



Afbeelding 1. MSI-package maken met WiX

die beperkt zijn in schrijfruimte, bijvoorbeeld diskettes. Om ook grotere installatieprogramma's op diskettes te kunnen verdelen, maakt Windows Installer het mogelijk om bestanden te comprimeren in verschillende Cabinet-bestanden. Via 'Media'-elementen worden verschillende diskettes gedefinieerd. Bij de bestanden zelf zal dan een referentie worden geplaatst om aan te geven in welke Cabinet (of medium) het bestand terecht moet komen. De folderstructuur wordt in WiX door middel van geneste 'Directory'-elementen opgebouwd. Het eerste 'Directory'-element heeft altijd een 'Id'-attribuut met de waarde TARGETDIR. Dit geeft de rootfolder aan waar alles geïnstalleerd wordt. Verder heeft Windows Installer nog een aantal vaste waarden voor 'Id' die naar een specifieke folder verwijzen: ProgramFilesFolder, ProgramMenuFolder, TempFolder, WindowsFolder, DesktopFolder, enzovoort. Met het 'Name'-attribuut wordt de naam van de folder opgegeven in een 8.3-formaat, terwijl het 'LongName'-attribuut gebruikt zal worden voor bestandssystemen die langere bestandsnamen aankunnen. Hoewel .NET-applicaties altijd worden geïnstalleerd op een besturingssysteem (Windows) dat lange namen ondersteunt, kan het gebeuren dat de bestanden ook worden weggeschreven op een medium dat enkel 8.3-bestandsnamen ondersteunt, bijvoorbeeld op een oude netwerkserver. Binnen een 'Directory'-element worden uiteindelijk ook één of meer 'Component'-elementen toegevoegd. En in een 'Component'-element worden de verschillende onderdelen, zoals registry-sleutels en -waarden, bestanden en snelkoppelingen opgegeven. Tot slot moeten ook nog de features worden opgegeven. In dit voorbeeld is er maar één feature, die de twee eerder gedefinieerde componenten bevat.

Aanmaken van de MSI

Nu het WiX-bronbestand klaar is, moet dit gecompileerd worden. Zoals eerder aangegeven, is Candle de WiX-compiler en wordt deze via de opdrachtregel aangeroepen. Geef als argument de naam van het WiX-bronbestand op. Standaard heeft een WiX-bronbestand de extensie '.wxs'. Om het codevoorbeeld te compileren, typ je dan 'candle.exe HelloWiXWereld.wxs'. Nadat de compiler klaar is, kan het gegenereerde object gelinkt worden en kan er een Windows Installer database van gemaakt worden. Dit doe je met Light via de opdrachtregel: 'light.exe HelloWiXWereld.wixobj'. Als alles goed verlopen is, staat er nu een Windows Installer database klaar. Zie ook het stappenplan in afbeelding 1.

Testen

Het installatieprogramma kan nu worden getest door de volgende opdracht te typen: 'msiexec.exe /i HelloWiXWereld.msi'. Na enkele seconden verdwijnt het Windows Installer scherm en kan je de applicatie opstarten via de Start-menu. Ook zal de applicatie toegevoegd zijn in de lijst van geïnstalleerde applicaties (Add/Remove Programs of Software in het configuratiescherm). Langs deze weg kan de applicatie

ook gemakkelijk worden verwijderd. Een alternatief is de applicatie te verwijderen via de opdrachtregel: 'msiexec.exe /x HelloWiXWereld.msi'. Zoals in het begin van dit artikel aangegeven werd, biedt Windows Installer standaard veel functionaliteit. Zo kan ook gemakkelijk getest worden of de voorbeeldapplicatie zichzelf inderdaad ook herstelt. Hiervoor verwijder je de folder waarin de applicatie geïnstalleerd werd. Standaard is dit waarschijnlijk onder C:\Program Files\HelloWiXWereld\. Als je nu de applicatie opnieuw probeert te starten, verschijnt onmiddellijk een venster van Windows Installer en kort hierna start de applicatie op. De applicatie heeft zichzelf inderdaad hersteld. Het enige dat nodig was voor deze functionaliteit, was het koppelen van de snelkoppeling aan de feature. Dit gebeurde door 'Target="DefaultFeature"' als attribuut toe te kennen aan het 'Shortcut'-element. Wanneer een snelkoppeling naar een feature verwijst - via het Target-attribuut - zal Windows Installer controleren dat alle componenten van die feature geïnstalleerd zijn zodra de snelkoppeling wordt aangeklikt.

Verder gaan...

De installatie uit het codevoorbeeld maakt geen gebruik van een mooie user interface die de gebruikers begeleidt tijdens de installatie. Zo kunnen gebruikers momenteel geen locatie ingeven waar ze de applicatie willen installeren, er wordt geen licentieovereenkomst getoond, en de vooruitgang van de installatie is niet erg duidelijk. Toch is dit allemaal ook mogelijk met WiX. Merk op dat in dit artikel enkel de basisonderdelen van een installatie behandeld werden. WiX kan bijvoorbeeld ook werken met include-bestanden, die een deel van de broncode bevatten.

Microsoft Press heeft enkele zeer goede boeken over Windows Installer, en ook de Microsoft Platform SDK bevat heel veel informatie over maken van installatieprogramma's met de Windows Installer SDK. Wanneer je zelf met WiX aan de slag wil gaan, beveel ik je ten sterkste aan om eerst de WiX-documentatie grondig te lezen. Hier staat onder andere een volledig uitgewerkt voorbeeld in om een installatie te maken. Verder wordt elk element uit het XML-document beschreven. Ook het Windows Installer SDK helpbestand uit de Microsoft Platform SDK is onontbeerlijk referentiemateriaal voor het maken van jouw eerste WiX-broncode.

XML: voordelen voor ontwikkelaars

Alhoewel verschillende softwareleveranciers mooi ogende en gebruiksvriendelijke software op de markt brengen om Windows Installer databases aan te maken, biedt WiX ook veel voordelen voor softwareontwikkelaars. Op het ogenblik van schrijven is er echter nog één nadeel aan WiX en dat is het ontbreken van een echt bruikbare visuele tool om de broncode op te bouwen. Dit maakt het gebruik van de toolset wat moeilijker voor ontwikkelaars met weinig ervaring met Windows Installer. Maar de voordelen van WiX zijn ook groot. De broncode bestaat uit zuivere XML en kan dus perfect door eigen software gemanipuleerd worden. Denk maar hoe gemakkelijk het is om het aanmaken van installatiesoftware met WiX te integreren in een automatisch build-proces. Bovendien kan de WiX-broncode gemakkelijk worden vergeleken met een oudere versie van de broncode die zich in een versiebeheersysteem bevindt, zoals Microsoft Visual SourceSafe. Zo'n vergelijking is nog steeds tekstueel - en houdt dus geen rekening met het feit dat XML eigenlijk data is - maar het is toch bruikbaar dan een binaire vergelijking tussen de Windows Installer databases onderling. Windows Installer en WiX bieden echter zo veel mogelijkheden dat dit artikel er slechts enkele heeft kunnen bespreken.

Nuttige internetadressen:

<http://blogs.msdn.com/robmen/>
<http://sourceforge.net/projects/wix/>
http://msdn.microsoft.com/library/en-us/msi/setup/windows_installer_start_page.asp

Tom Mertens geeft zijn klanten advies over het ontwikkelen van applicaties met het .NET Framework. Je kan hem bereiken via tom@itm.be