

Eenvoudig documenteren met Sandcastle

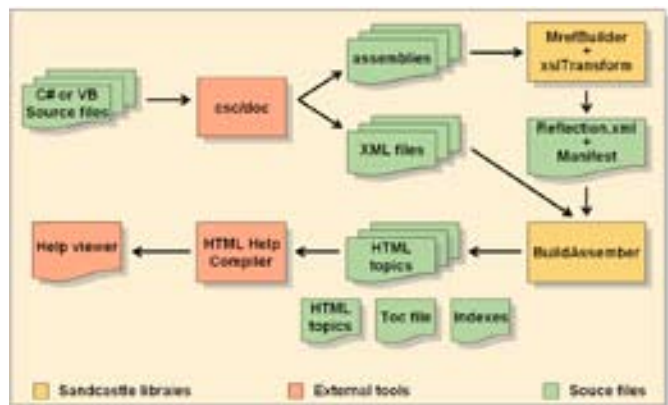
EEN GOED ALTERNATIEF VOOR NDOC

Het documenteren van objecten is nog steeds een ondergeschoven kindje, ondanks de documentatiemogelijkheden van .NET-programmeertalen. Met Sandcastle biedt Microsoft nu een set gereedschappen aan, waarmee eenvoudig documentatie voor .NET-sourcecode kan worden gegenereerd. In dit artikel wordt getoond hoe Sandcastle kan worden gebruikt als alternatief voor het verdwenen NDoc.

Veel C#-ontwikkelaars gebruikten NDoc om documentatie te genereren op basis van de XML-tags in C#-code. Met de introductie van .NET 2.0 kon dit ook met andere .NET-talen zoals Visual Basic .NET. NDoc ondersteunt de nieuwe functionaliteit van .NET 2.0, zoals generics, niet of onvolledig. Bovendien is door diverse oorzaken de ontwikkeling van NDoc eind 2005 volledig gestopt. Op het MSDN Developer Documentation and Help System-forum werd al vanaf medio 2005 over dit probleem gediscussieerd. In mei van 2006 verscheen in een van deze discussies de aankondiging van Sandcastle door Anand Raman, Group Manager Application Division bij Microsoft. In deze aankondiging meldde Anand dat het bedrijf uit Redmond zijn eigen interne documentatiecompilers (codenaam Sandcastle) beschikbaar zou maken aan softwareontwikkelaars.

Sandcastle

Sandcastle is een set gereedschappen waarmee documentatie kan worden gecompileerd, gebaseerd op de XML-tags in .NET-code. Microsoft gebruikt Sandcastle zelf bij het genereren van documentatie voor MSDN en Visual Studio. In juli 2006 werd de eerste Community Technology Preview vrijgegeven, nadat het eerst door Microsoft intensief getest werd door de MSDN-documentatie te genereren. De eerste versie maakte het mogelijk documentatie te genereren voor .NET 2.0-code. De kern van Sandcastle bestaat uit enkele command-line-applicaties die gebruikmaken van XML en XSLT. Met behulp van reflection en de XML-tags uit de code wordt de uiteindelijke documentatie van de assemblies gegenereerd. Het genereren van documentatie met Sandcastle is inzichtelijk gemaakt



Afbeelding 1. Genereren van documentatie met Sandcastle

in afbeelding 1. De .NET-compiler produceert de assemblies en de bijbehorende XML-documentatie. Met behulp van reflection en een aantal XSL-bestanden wordt vervolgens een manifest en een Reflection.XML-bestand gegenereerd. Deze bestanden, aangevuld met de XML-bestanden die door de compiler zijn gegenereerd, worden door de BuildAssembler gebruikt om html-bestanden te genereren. De geproduceerde html dient als basis voor de HTML Help Compiler (HHC) om de uiteindelijke documentatie te genereren.

Omdat Microsoft intern ook gebruikmaakt van Sandcastle voor het genereren van documentatie, wordt voorkomen dat nieuwe .NET Frameworks niet of niet correct worden ondersteund. Sandcastle biedt op dit moment al de mogelijkheid documentatie te compileren die voldoet aan de nieuwe Visual Studio 2008-stijl. Sandcastle biedt daardoor een stabiele basis voor het documenteren van de code.

Zelf aan de slag

Tot zover de theorie, maar hoe werkt Sandcastle in de praktijk? Om Sandcastle te kunnen gebruiken, moeten de volgende onderdelen worden geïnstalleerd:

- Visual Studio 2005 SDK 4.0. Deze bevat onder andere HTML Microsoft Help 2.0 SDK die nodig is om documentatie te compileren.
- De recentste versie van Sandcastle.

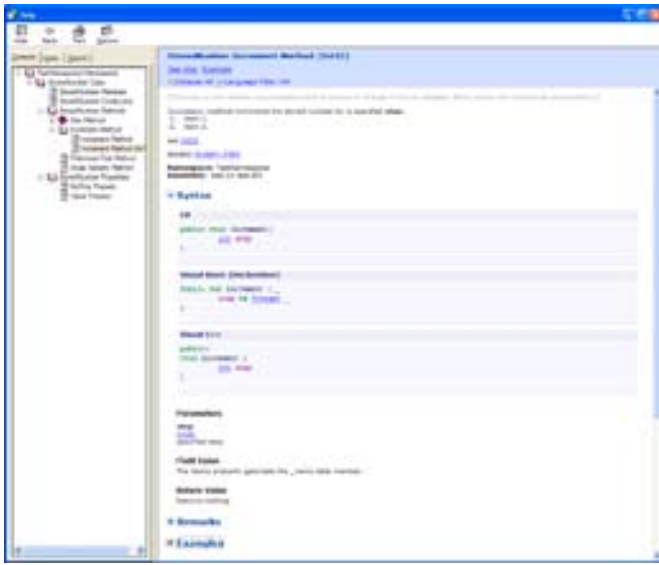
Wanneer beide installaties correct zijn uitgevoerd, kunnen we beginnen met het genereren van documentatie. De eenvoudigste en snelste manier om een indruk te krijgen van Sandcastle is gebruik te maken van de meegeleverde voorbeelden. Deze voorbeelden zijn te vinden in de map C:\Program Files\Sandcastle\Examples\Sandcastle. Je vindt hier de volgende bestanden:

- Test.CS. Voorbeeldcode.
- Build_Sandcastle.bat. Een batchbestand waarmee documentatiebestand (CHM) voor test.cs kan worden gecompileerd.
- Build.proj. Een MS Build-project.

Met behulp van deze voorbeelden kan een CHM-bestand worden gecompileerd met de documentatie voor de Test.cs.

Het proces in detail

We zullen nu de stappen doorlopen om een CHM-bestand te compileren voor de meegeleverde voorbeeldcode. Allereerst moet de voorbeeldcode worden gecompileerd. Daarbij moet ook het XML-bestand met de documentatie-tags worden gecompileerd. Bij de volgende stap wordt informatie over de APIs in een assembly verzameld. Hiervoor wordt de Sandcastle-applicatie MrefBuilder



Afbeelding 2. Een CHM-bestand met daarin de documentatie

gestart. Deze applicatie gebruikt reflection om informatie over APIs (visibility, return types, parameter namen en typen, inheritance-data, is een class abstract of sealed, enzovoort) uit de assemblies te halen. Vervolgens wordt met behulp van de Sandcastle-applicatie XslTransform een aantal XSL-transformaties op dit bestand uitgevoerd. Welke dit zijn, is afhankelijk van de manier waarop de documentatie gepresenteerd moet worden. De juni-CTP van Sandcastle biedt op dit moment standaard XSL-transforms voor de volgende presentatiemogelijkheden:

- Prototype
- Visual Studio 2005
- Visual Studio 2008

Het resultaat van het voorgaande wordt met behulp van XslTransform omgevormd tot een zogeheten Topic Manifest. Dit is niets meer dan een lijst met topics (onderwerpen) die gecompileerd moeten worden. Alle bestanden die later door HHC worden gebruikt, moeten in een outputmap worden opgeslagen. Het batchbestand CopyOutput.bat maakt deze map aan met de daarvoor benodigde structuur. Hierna wordt de Sandcastle-applicatie BuildAssembler uitgevoerd. Deze verzamelt de informatie uit de hiervoor gegenereerde XML en reflection-informatie en genereert daarmee een Manifest.XML-bestand. Dit Manifest.XML-bestand wordt dan met behulp van een XSL-transformatie weer omgevormd tot een projectfile voor HHC. Er worden nog drie XSL-transformaties uitgevoerd met de XslTransform-applicatie voordat de HHC het CHM-bestand kan compileren. De eerste creëert een XML-bestand met daarin de 'Table Of Content' (TOC) informatie. De tweede transformatie gebruikt deze TOC-informatie om een nieuw bestand aan te maken dat door HHC wordt gebruikt om de informatie in de content-pane op te bouwen. De laatste transformatie gebruikt de beschikbare reflection-informatie om een bestand aan te maken waaruit HHC de index kan opbouwen. Ten slotte kan HHC worden aangeroepen om de CHM te compileren. Het eindresultaat is een CHM-bestand met daarin de documentatie; zie afbeelding 2. Alle hiervoor beschreven stappen worden uitgevoerd door het batchbestand en het build-script.

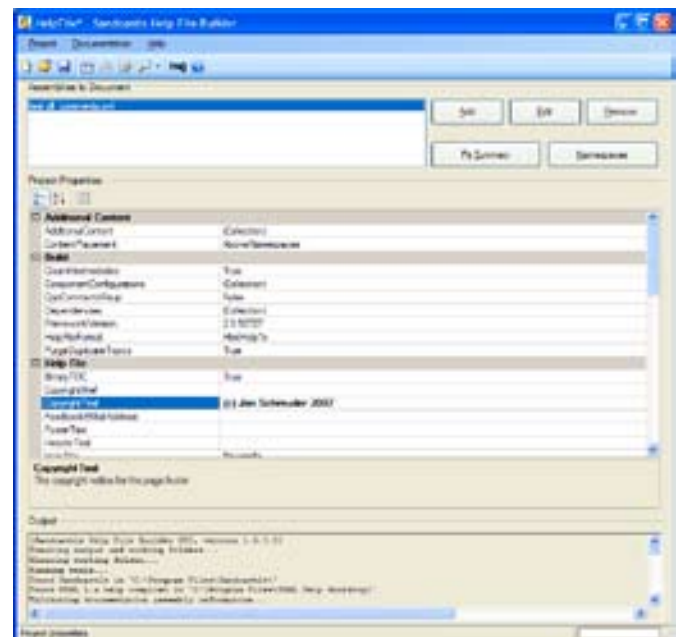
Bewerkelijk

Wat onmiddellijk opvalt is dat er een flink aantal stappen moet worden doorlopen om een CHM-bestand te compileren voor een relatief eenvoudige class. Ook zijn er nog enkele factoren die het gebruik van Sandcastle nog complexer maken wanneer je Sandcastle voor het eerst gebruikt. Referenties naar componenten die geen deel uitmaken van het .NET Framework, zoals Crystal Reports of eigen class-libraries, moeten expliciet worden vermeld bij het uitvoeren van MRefBuilder. Het genereren van documen-

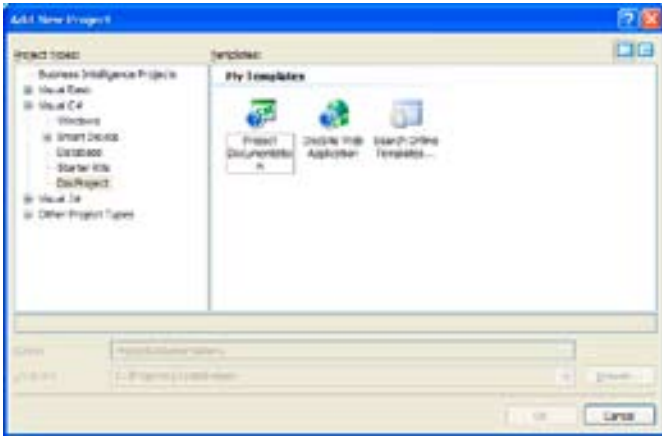
tatie voor solutions die uit meer assemblies bestaan, maakt het gebruik van Sandcastle nog complexer. Het zijn echter wel veelvoorkomende situaties in de dagelijkse praktijk van veel ontwikkelaars. In die complexiteit ligt echter ook een grote mate van flexibiliteit. Het hele proces is verdeeld in een aantal opeenvolgende stappen. Extra transformaties zijn daardoor eenvoudig tussen te voegen, waardoor je als ontwikkelaar invloed blijft houden op het hele proces. Ook bieden deze losse applicaties de mogelijkheid het genereren van documentatie met behulp van een MS-Build-script te automatiseren. Omdat veel van de presentatie door middel van XSL-transformaties wordt geregeld, heb je als ontwikkelaar veel invloed op de look-and-feel van de uiteindelijke documentatie. Je kunt een compleet eigen presentatieformaat definiëren. Het ontbreken van een gebruiksvriendelijke interface kan voor veel ontwikkelaars een drempel zijn om Sandcastle te gebruiken voor het compileren van documentatie. Dit probleem is door de .NET-community zeer voortvarend opgepakt. Al kort na de introductie van Sandcastle in juni 2006 verschenen op diverse weblogs applicaties die batchfiles genereerden voor Sandcastle. Ook verschenen er diverse add-ins en voorbeelden van MS-Build-scripts. Nu, ruim een jaar later, zijn er diverse open source-applicaties beschikbaar op CodePlex die de complexiteit van Sandcastle wegnemen en ook extra functionaliteit toevoegen. Drie van deze applicaties wil ik hier noemen: Sandcastle Help File Builder, DocProject en ScriptDoc.

Sandcastle Help File Builder

Sandcastle Help File Builder (SHFB), ontwikkeld door Eric Woodruff, verscheen al in september 2006 en is een goed alternatief voor ontwikkelaars die voorheen gebruikmaakten van NDoc. De gebruikersinterface (zie afbeelding 3) van SHFB zal Ndoc-gebruikers bekend voorkomen. De versies van SHFB houden gelijke tred met de CTP-releases van Sandcastle. Met SHFB kunnen ontwikkelaars een projectbestand aanmaken, waarmee de documentatie voor de assemblies kan worden gegenereerd. Aan het project kunnen meer assemblies worden toegevoegd, zodat er uiteindelijk één document ontstaat voor alle assemblies in een Visual Studio Solution. Nog eenvoudiger is het om, analoog aan het oude NDoc, een VS 2005-solution te importeren. SHFB maakt dan een projectbestand aan dat is gebaseerd op de VS 2005-solution. Gebruikers van het oude NDoc kunnen zelfs hun oude Ndoc-projecten eenvoudig importeren en converteren. De documentatieprojecten van SHFB kunnen worden uitgevoerd met een consoleapplicatie die in de set-up van SHFB wordt meegeleverd. Op deze manier kun je het genereren van documentatie in een build-proces opnemen.



Afbeelding 3. De gebruikersinterface van SHFB



Afbeelding 4. DocProject beschikt over twee nieuwe projecttemplates

DocProject

Van recentere datum is DocProject, geschreven door Dave Sexton. Deze applicatie integreert volledig met Visual Studio 2005 en voegt twee nieuwe projecttemplates (zie afbeelding 4) toe aan de al aanwezige templates in Visual Studio 2005. Deze projecten kunnen, net als andere projecten, worden toegevoegd aan de solution. De applicatie start vervolgens een wizard die de basisparameters voor het documentatieproject opvraagt. De assemblies die in de solution zijn opgenomen, worden automatisch door de applicatie toegevoegd aan het documentatieproject. De applicatie maakt vervolgens gebruik van MS-Build om de documentatie te genereren. Deze integratie van het genereren van documentatie in het build-proces maakt DocProject uitermate geschikt om in grotere projecten te gebruiken. Door het in het build-proces op te nemen, kunnen ontwikkelaars worden gedwongen om documentatie op te nemen in de sourcecode. Wijzigingen in API's worden op deze manier ook beter up-to-date gehouden. De documentatie sluit op die manier dan ook beter aan op de technische werkelijkheid. DocProject biedt bovendien ook een externe GUI die, net als SHFB, veel op het oude Ndoc lijkt. Dankzij deze externe GUI kunnen ontwikkelaars die met Visual Studio Express-applicaties werken, ook eenvoudig gebruikmaken van Sandcastle.

ScriptDoc

Deze derde applicatie, ontwikkeld door Bertrand Leroy, maakt het mogelijk documentatie te generen op basis van JavaScript-code in .NET-webapplicaties. Zeker nu steeds meer webapplicaties gebruikmaken van Ajax om een betere gebruikerservaring te bieden, is documentatie van JavaScript-code van groot belang. ScriptDoc haalt de documentatie uit de JavaScript-code en genereert daarmee de XML-bestanden en andere gegevens die nodig zijn om documentatie te genereren met Sandcastle.

Aandachtspunten bij Sandcastle

Sandcastle maakt gebruik van reflection om informatie over de gerefererde assemblies te achterhalen. Dit heeft als consequentie dat deze assemblies bij het genereren van de documentatie ook beschikbaar moeten zijn. Sandcastle moet ook weten waar deze staan. Vooral voor GAC-assemblies kan dit lastig zijn, maar Sandcastle en de hiervoor beschreven tools bieden mogelijkheden dit probleem op te lossen. Met Sandcastle kun je zelf op zoek gaan naar assemblies in de GAC, terwijl de tools mogelijkheden bieden om expliciet aan te geven waar de assemblies gevonden kunnen worden. De command-line applicaties van Sandcastle kennen veel opties. Er moeten daardoor zeer uitgebreide command-lineopdrachten worden uitgevoerd wanneer Sandcastle in een buildscript, batchbestand of powershell-script wordt gebruikt. In het bijzonder MRefBuilder gebruikt flink veel van het interne geheugen. Dit is zeker iets om rekening mee te houden wanneer men Sandcastle wil gaan gebruiken. Het werkt redelijk op een machine met Windows XP en 512 Mb intern geheugen, maar een minimum intern geheugen van 1 Gb is aan te bevelen.

Uitstekend alternatief

Met het introduceren van Sandcastle is het mogelijk op relatief eenvoudige wijze documentatie te genereren op basis van informatie in de code. Microsoft maakt zelf ook gebruik van Sandcastle. De MSDN-documentatie wordt met Sandcastle gegenereerd, maar ook de QuickStarts voor Silverlight.NET (<http://silverlight.net/quick-starts/>). De open source-gemeenschap heeft zeer nuttige uitbreidingen voortgebracht, met als exponenten de hiervoor genoemde Sandcastle Help File Builder en DocProject. SHFB is een uitstekend alternatief voor NDoc en ontwikkelaars die met NDoc uit de voeten kunnen, zullen ook direct met SHFB overweg kunnen. DocProject biedt als belangrijk voordeel dat het een projectstructuur biedt die opgenomen kan worden in de solution. Hierdoor is het naadloos in te passen in de builds die door VSTS worden uitgevoerd. Verder biedt DocProject een NDoc-achtige interface aan, zodat ook Ndoc-gebruikers snel aan de slag kunnen. Mijn persoonlijke voorkeur gaat daarom uit naar DocProject. Sandcastle en de hiervoor genoemde applicaties maken documenteren zeer eenvoudig. Er hoeft dan ook niets meer in de weg te staan om documentatie te genereren die bij de projecten behoren die opgeleverd worden.

Jan Schreuder is Microsoft Consultant bij de divisie Energy, Utilities & Telecom van LogicaCMG (<http://www.logicacmg.com/nl>). Voor vragen en opmerkingen kun je hem bereiken op jan.schreuder@logiacmg.com. Zijn blog kun je vinden op <http://bloggingabout.net/blogs/jschreuder>.

Referenties

Sandcastle-blog: <http://blogs.msdn.com/Sandcastle>
 Sandcastle CTP: <http://www.microsoft.com/downloads/details.aspx?FamilyID=e82ea71d-da89-42ee-a715-696e3a4873b2&displaylang=en>
 MSDN Developer Documentation and Help System forum: <http://forums.microsoft.com/MSDN/ShowForum.aspx?ForumID=66&SiteID=1>
 Eric Woodruff's Sandcastle Help File Builder: <http://www.codeplex.com/SHFB>
 Dave Sexton's DocProject: <http://www.codeplex.com/DocProject>
 Bertrand Leroy's ScriptDoc: <http://www.codeplex.com/scriptdoc>

(advertentie MS Press)



Introducing Microsoft Silverlight™ 1.0
 ISBN: 9780735625396
 Author: Laurence Moroney
 Page Count: 256