

SharePoint 2010 voor ontwikkelaars

MEEST IN HET OOG SPRINGENDE VERANDERINGEN IN NIEUWE VERSIE

Wouter van Vugt

Een SharePoint-ontwikkelaar is goed uitgerust, heeft de securiteit van een horlogemaker en vindt het niet erg om met Spartaanse ontwikkelmethodes te werken. Vanaf SharePoint 2010 zal hier echter het nodige aan gaan veranderen. De F5-Experience zal het leven van dagelijkse en incidentele SharePoint ontwikkelaars volledig omgooien.

Vanaf deze nieuwe editie van SharePoint zijn vrijwel alle obstakels die ervoor zorgden dat SharePoint als complex werden beschouwd verstoppt achter een nieuwe laag aan krachtige tools en APIs. Werken met CAML? Niet meer direct nodig met integratie van Linq. Tools voor het maken van Diamond Directive Files (DDF, dat is toch zóó 1980)? Nog maar exact één keuze; Visual Studio 2010. Integratie via Web Services? Nope, objectmodellen aan de server én in de client. Enorm veel interessante features om mee aan de slag te gaan. SharePoint-ontwikkelaars zullen (gelukkig) nog altijd goed uitgerust zijn, want SharePoint blijft natuurlijk wel een hele gewone ASP.NET applicatie. In dit artikel krijg je een introductie in al het lekkers wat Microsoft voor ontwikkelaars heeft bekookst. Te veel om op enkele pagina's tekst te vatten, en derhalve komen hierna meer artikelen over wat SharePoint 2010 allemaal gaat brengen. Hier alvast een overzicht van de hoofdpunten.

Visual Studio 2010

Een kernvraag die typisch bij het beginnen van een SharePoint-project wordt gesteld is welke tool gebruikt gaat worden voor het bouwen van de solution packages (installeerbare pakketten voor SharePoint). Met de komst van Visual Studio 2010 is deze vraag volledig overbodig. Vanaf deze versie is SharePoint eindelijk een volwaardig onderdeel van de ontwikkelomgeving net als de Windows, Web en Office applicaties. De SharePoint Tools in deze populaire IDE zijn van de grond af opgebouwd met als belangrijkste kenmerk extensibility. Hierdoor is het niet meer nodig om halfgeïntegreerde tools te gebruiken zoals WSP Builder, of om te vechten met de black-box van Visual Studio Extensions for WSS.

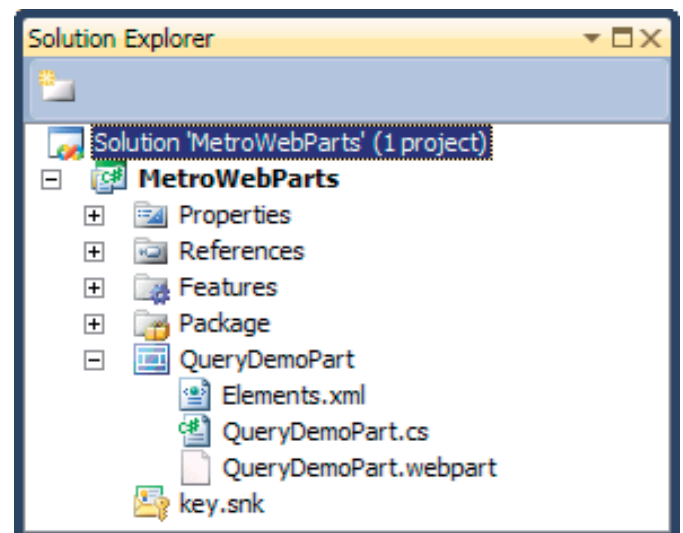
Het kernbegrip in de Visual Studio 2010 SharePoint Tools is de SPI, ofwel SharePoint Project Item. Een SPI is een logische samenpakking van files, geplaatst in wat op het filesysteem een directory is, maar wat tevens in Visual Studio een programmeerbaar concept vormt. Zo kun je aan een SPI event handlers koppelen, zodat wanneer een file in de directory van naam of inhoud verandert alle relevante files mee worden aangepast. Neem bijvoor-

beeld de <Module> SPI. Wanneer je in de map files bijplaatst, worden deze gelijk toegevoegd als <File> elementen ter deployment naar het SharePoint virtuele filesysteem. Ten behoeve van deployment van SPIs kun je via het Property Window in Visual Studio exact configureren waar een file binnen SharePoint gedeplouy moet worden.

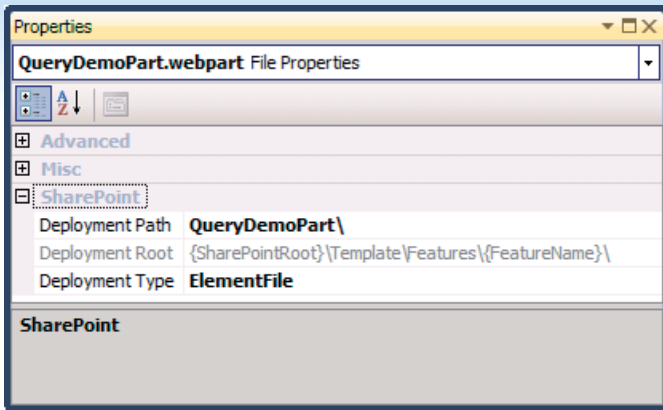
Via het extensibility model is het mogelijk om aan een SPI dingen te koppelen zoals menu-items en visuele designers. Dit maakt het werken met SharePoint-elementen eenvoudiger. Zo hoef je bijvoorbeeld geen List en menu identifiers te onthouden (Announcements is?...104 natuurlijk!)

De SharePoint Tools worden gecomplementeerd met designers voor Features en Solution Packages waarbij de SharePoint-elementen zowel door de designer maar ook handmatig aan te passen zijn.

Ook een handigheid is de SharePoint Explorer. Geïntegreerd met de Server Explorer functie toont deze tool de inhoud van Share-



FIGUUR 1: EEN PROJECT MET EEN WEB PART SPI.



FIGUUR 2: DEPLOYMENTCONFIGURATIE VAN DE WEB PART DEFINITIE.

Point servers en vitale kenmerken die via de user interface verborgen blijven.

Qua extensibility is er veel te doen rondom de SharePoint Tools in Visual Studio 2010. De tools zijn eenvoudig uitbreidbaar door middel van MEF-componenten. MEF is het Managed Extensibility Framework dat door Visual Studio gebruikt wordt als nieuw ontwikkelmodel. Door deze MEF extensibilitylaag van SharePoint Tools hoef je minder met COM interfaces te werken en wordt het een stuk eenvoudiger om nieuwe SPI's te maken, bestaande SPI's uit te breiden en te werken met de SharePoint Explorer. Ook aan de Explorer kun je nodes toevoegen en uitbreiden. Helemaal sterk zijn de uitbreidbare deployment steps. Hiermee kan je het deployment proces volledig naar je hand zetten. Xcopy deployment of een volledige solution install, resetten van de test site, allemaal dingen die eenvoudig worden toegevoegd.

Laten we eens nader gaan kijken wat voor soorten oplossingen er met deze tools gebouwd kunnen worden.

Sandboxed Solutions

SharePoint kan op twee manieren worden uitgebreid. Declaratief met behulp van XML, wat ruwweg gezien kan worden als productconfiguratie, maar ook imperatief, met behulp van code. Juist dit laatste is wat beheerders zenuwachtig maakt, want het betreft code die geen onderdeel is van het originele product. Het is in SharePoint 2007 niet eenvoudig om deze code aan banden te leggen. Ook wanneer code executeert onder de user account van de ingelogde gebruiker (impersonatie) kan via `SPSecurity.RunWithElevatedPrivileges` toch een blok code met verhoogde permissies worden geëxecuteerd zonder dat hier een wachtwoord voor nodig is. Het is voor beheerders ondoenlijk om alle code die wordt gedeployed te onderzoeken op aanroepen die als onveilig beschouwd zou kunnen worden.

Naast de enorme verandering in de ontwikkelmethode is er een even zo grote verandering op het gebied van deployment. SharePoint als platform laat het toe om verantwoordelijkheid voor de dagelijkse operatie naar eindgebruikers te verleggen. Zo kunnen gebruikers zelf sites aanmaken en content beheren binnen hun deel van de omgeving, en dat zonder tussenkomst van ontwikkelaars en systeembeheer. Echter, het toevoegen van nieuwe functionaliteit was altijd een taak waar beheerders aan te pas kwamen. In SharePoint 2010 is ook deze verantwoordelijkheid voor een groot deel naar eindgebruikers verlegd, waarbij de beheertak natuurlijk wel het nodige kan configureren over wat wel en juist niet is toegestaan. Het algehele ontwikkelmodel verandert niet. Nog altijd maak je een gewone solution package.

Het grootste verschil tussen een normale solution en een sandboxed solution is de manier van deployment. In plaats van installatie met beheertools wordt de oplossing gedeployed naar een site collectie door de site administrator, typisch een eindgebruiker. Binnen dit nieuwe niveau van deployment kun je echter geen zaken installeren die site collecties overstijgen, zoals files die in de layouts folder moeten worden geplaatst.

Elke site collectie heeft ten behoeve van dit nieuwe deployment-model een nieuwe documentbibliotheek genaamd de Solution Gallery. De declaratieve kant van de oplossing werkt zoals we gewend zijn van het oude model. Content-type definities, modules, custom actions, alles werkt zoals normaal. De code wordt echter op een speciale manier geëxecuteerd. In plaats van in-process executie zoals plaats vindt bij installatie door beheerders, wordt de code out-of-band in een apart worker-process uitgevoerd. Door strenge Code Access Security restricties en het onbreken van SharePoint framework code die als onnodig wordt beschouwd, kan onbekende code geen kwaad doen buiten de site collectie waar de oplossing actief is. Zo is `SPSecurity.RunWithElevatedPrivileges` geen probleem meer, want deze aanroep is simpelweg niet aanwezig in de sandbox. Is het nu toch nodig om dingen uit te voeren met een verhoogde security-context dan kun je gebruik maken van voorgeconfigureerde proxies die altijd inzichtelijk te maken zijn voor beheerdoeleinden. Deze proxies installeer je samen met je solution.

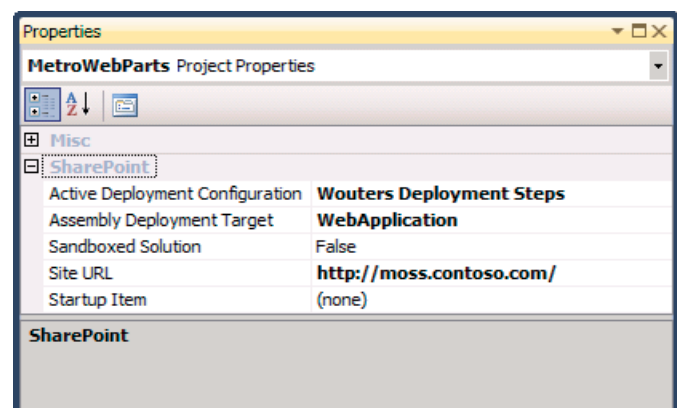
Toevoegen van nieuwe functionaliteit is voor een groot deel naar de eindgebruikers verlegd

Om het kraakhelder te maken; het is de best practice om een oplossing als sandboxed solution te ontwikkelen en deployen. De voordelen uiteten zich vooral in minder beheer en betere monitoring.

Nu we via het ontwikkelmodel ook naar het deploymentmodel hebben gekeken, is het de hoogste tijd om eens nader te onderzoeken welke goodies er voor ontwikkelaars aan het product zijn toegevoegd.

Linq To SharePoint

Een van de SharePoint black-arts die een developer moet leren om de zwarte band met gele slip te halen is het gebruiken van CAML, een XML-taal om SharePoint data mee te raadplegen.



FIGUUR 3: PROJECT CONFIGURATIE MET DEPLOYMENT OPTIES.

Er is een nieuw smoelwerk aangebracht dat het aantal vereiste kliks en navigaties tot een minimum terugbrengt

Hoewel krachtig, is CAML niet de meest eenvoudige taal om te leren. Zo geven foute queries soms geen resultaat, wat het onderscheid tussen een query die gewoon niets heeft gevonden en een query die een fout bevat moeilijk te achterhalen valt.

De kern van Linq To SharePoint zijn klassestructuren gegenereerd met de tool SPMetal.exe. Deze gegenereerde klassen bieden een getypeerde weergave van een SharePoint-lijst, hetzelfde concept als Typed DataSets in ADO.NET. Door nu tegen deze klassen Linq-queries te schrijven krijg je toegang tot IntelliSense en refactoring support van Visual Studio. CAML zelf is in geen velden of wegen te bekennen. Onderwater worden de queries vertaald naar de overeenkomstige XML-structuur en geëxecuteerd via het SharePoint-objectmodel. De opgehaalde list items worden vertaald naar hun overeenkomstige klassen die je bijvoorbeeld kunt verbinden met ASP.NET controls via data-binding.

```
EntitiesDataContext context = new EntitiesDataContext
("http://server");
var parts = from part in context.PartCatalog
            where part.Product.Description.Contains("Aspirin")
            select part;
```

Je gaat de kracht van Linq To SharePoint pas echt appreciëren op het moment dat je lijsten via joins gaat aflopen. SharePoint-lijsten zijn vanaf de 2010 versie geschikt voor het opslaan van relationele data. Hierbij kun je gebruik maken van database constructies als cascading en restrict deletes en unique constraints. Linq To SharePoint is volledig op de hoogte van al dit moois en weet de juiste queries te genereren op basis van de object relaties die je afloopt in de Linq-query.

Client Object Models

Naast de complexiteit van het werken met SharePoint data binnen SharePoint zelf, is het ophalen van data ten behoeve van clientapplicaties niet veel eenvoudiger. De Lists.asmx web service gebruikt om SharePoint data naar buiten toe aan te bieden is gebaseerd op dezelfde CAML query taal. Daarnaast worden de resultaten als XML strings aangeboden, met uitbundig gebruik van de System.Xml namespace ten gevolg.

SharePoint 2010 pakt groots uit met maar liefst drie aparte client objectmodellen, waarmee het ophalen en werken met SharePoint data uniform verloopt over verschillende platformen. De drie modellen zijn voor .NET, SilverLight en JavaScript. Qua bereik van de functionaliteit kun je ruwweg alles binnen de site collectie benaderen. Voor de veelgebruikte klassen zoals SPList en SPListItem zijn equivalenten ontwikkeld voor gebruik op de client, maar dan zonder de SP prefix zoals regelmatig gebruikt wordt in de server assemblies.

```
using (ClientContext clientContext =
    new ClientContext("http://server"))
{
    // Load the site collection properties
    Site siteCollection = clientContext.Site;
    clientContext.Load(siteCollection);
    clientContext.ExecuteQuery();
    // Print a property
    Console.WriteLine(siteCollection.Url);
}
```

```
// Load specific lists, only load the Title
clientContext.Load(clientContext.Web,
    x => x.Lists
        .Include(l => l.Title)
        .Where(l => l.Title.Contains("Items")));
clientContext.ExecuteQuery();
foreach (List list in clientContext.Web.Lists)
{
    Console.WriteLine(list.Title);
}
```

Een belangrijk kenmerk van de client objectmodellen is dat er heel specifiek gekozen moet worden welke gegevens opgehaald moeten worden. Een list item in SharePoint bevat al gauw vijftig waarden waarvan de meeste onzichtbaar zijn en dienen voor de interne huishouding. Niet slim om alles over een lijstje naar de client te sturen natuurlijk. Over dat lijstje; er wordt gebruik gemaakt van een gemeenschappelijke portal die queries ontvangt en JSON informatie terugstuurt.

Ribbon Extensibility

Een van de zaken waar veel ontwikkelaars mee zullen gaan stoeien is de nieuwe Fluent interface, je SharePoint customizations moeten natuurlijk wel wat knopjes hebben voor de interactie met gebruikers. Net als in de clientapplicaties is er een nieuw smoelwerk aangebracht wat het aantal vereiste kliks en navigaties terugbrengt tot een minimum. En met een nieuwe Ribbon UI komt ook een nieuw extensibility model.

Een aantal mooie developer aspecten van de ribbon op een rij:

- Natuurlijk gebouwd bovenop de ASP.NET Ajax bibliotheek
- Renderbaar zowel client-side als server-side.
- ondersteunt de Quick Access Toolbar en Jewel (Office Button)
- Toevoegen / verwijderen / aanpassen van alle elementen
- Commando's zowel declaratief als via JavaScript
- Uitgebreid client object model voor het ontwikkelen van je eigen 'page components'
- Integreerbaar met multi-selectie in list views

Genoeg moois om van te gaan smullen. De kern van Ribbon extensibility bevat twee elementen. Ten eerste een XML-model voor het maken van nieuwe tabs, groups, buttons enzovoorts. Daarnaast is het nodig om een in JavaScript geschreven klasse te maken, bekend als de page component. Het Ribbon framework communiceert met deze klasse voor het beheren van de in XML gedefinieerde elementen. Voor de pure server developer wordt het dus tijd om weer eens dieper in JavaScript te duiken. Krachtig spul, die Ribbon.

Business Connectivity Services

Wat begint als kleinschalig platform voor teamwerk ontwikkelt zich tot centrale spil in de dagelijkse bedrijfsvoering. Derhalve is het van enorm belang dat het eenvoudig is om binnen SharePoint met data van externe bronnen te kunnen werken. In de 2007 versie werd het voor non-developers mogelijk om met XML definitie externe web services en SQL databases te tonen binnen SharePoint door middel van voornamelijk Web Parts. De Business Data Catalog liet je toe om met XML modellen je data te modeleren. Dit XML-programmeren was echter niet zo eenvoudig zon-

Hoog tijd om verder te bladeren er meer te lezen over al het moois dat binnenkort beschikbaar komt!

der de nodige technische know-how. Wat vroeger BDC heette gaat nu door het leven als Business Connectivity Services ofwel BCS. Ook wel bekend als BDC++ +++++. Met behulp van BCS wordt het nu mogelijk om externe databronnen nog dieper te integreren in het jasje van SharePoint, waarbij zowel lezen als schrijven mogelijk is. Daarnaast faciliteren Office applicaties zoals Outlook en de SharePoint Workspace (a product formerly known as Groove) de toegang tot de gegevens, inclusief offline mogelijkheden en automatische sync. Als voorbeeld kan je de Contacts tabel uit de Northwind database bewerken via SharePoint lijsten in de browser UI, hierna kan je deze offline nemen naar Outlook, waar elke Northwind Contact als Outlook Contact beschikbaar is. Wanneer je in de trein een Outlook Northwind Contact toevoegt en op het werk aankomt, staat hij na een sync meteen in de Northwind Database. No coding required!

Business Connectivity Services wordt via SharePoint Designer en de browser beheerd. Het is hierbij niet nodig om XML te bewerken. Alles zit keurig achter deurtjes in de user interface verstopt. Daarnaast is het mogelijk om met Visual Studio 2010 .NET modellen op te stellen, deze kunnen dan net als web services, SQL en

WCF dienen als data bron voor het ophalen en wegschrijven van data. Een ander sterk kenmerk is dat de lijsten in SharePoint ook echt volwaardig als lijst te benaderen zijn. Het is transparant voor de SPList of er externe of SharePoint data wordt benaderd. Zo kun je bijvoorbeeld een External Data List gebruiken als bron voor Lookup kolommen. Aardig cool!

Conclusie

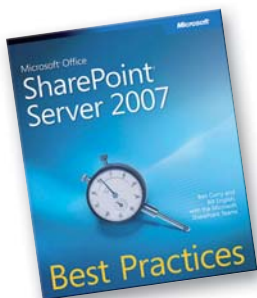
Wat nou conclusie? Dit verhaal is zeker nog niet klaar! Hoog tijd om verder in deze speciale eenmalige uitgave van SharePoint Magazine te bladeren er daar meer te lezen over al het moois dat binnenkort beschikbaar komt!



.....
Wouter van Vugt, maakt deel uit van een team dat door de Microsoft Developer Platform Evangelism groep is gevraagd om SharePoint 2010 training te ontwikkelen en te doceren.

(Advertentie)

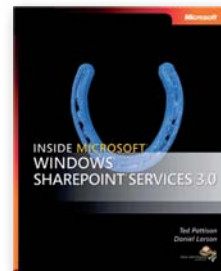
computercollectief
 computerbooks & software
comcol.nl



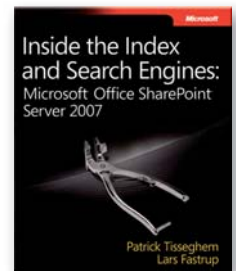
14131-A5
Microsoft Office SharePoint Server 2007 Best Practices € 42,50
 Gain expert insights on what works, where to make tradeoffs, and how to implement the best decisions for your organization.



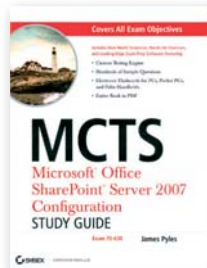
11618-A4
Inside Microsoft Office SharePoint Server 2007 € 31,90 Take a detailed look at SharePoint Server 2007 internals. Includes code samples.



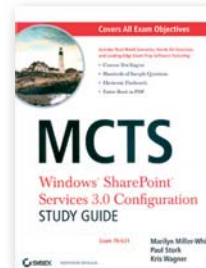
10824-A1
Inside Microsoft Windows SharePoint Services 3.0 € 35,90
 This guide includes C# and XML code samples.



2126-G3
Inside the Index and Search Engines € 35,90
 Optimize the user search experience in your workplace.



14054-A4
MCTS: Microsoft Office SharePoint Server 2007 Configuration Study Guide € 39,90 Exam 70-630
 Covers All Exam Objectives.



16958-A7
MCTS: Windows SharePoint Services 3.0 Configuration Study Guide € 39,90 Exam 70-631
 Covers All Exam Objectives.



11639-A4
Microsoft SharePoint Products and Technologies Administrator's Pocket Consultant € 21,90

SharePoint 2010 forthcoming titles:
Microsoft Press
 > Microsoft Office SharePoint Server 2010 Administrator's Pocket Consultant (April '10)
 > Microsoft Office SharePoint Designer 2010 Step by Step (Jan. '10)
 > Microsoft Office SharePoint Server 2010 Administrator's Companion (May '10)
 > Inside Microsoft Office SharePoint 2010 (May '10)
 > MCTS Self-Paced Training Kit (Exam 70-667): Configuring Microsoft Office SharePoint Server 2010
Apress
 > Pro SharePoint 2010 Development Techniques (Mar. '10)
 > Building the SharePoint 2010 User Experience (Mar. '10)
 > SharePoint 2010 User's Guide: Learning Microsoft's Collaboration and Productivity Platform (Mar. '10)