

# Application Lifecycle Management met BizTalk

## EFFECTIEF APPLICATIES ONTWIKKELEN IN TEAMVERBAND

Dennis Mulder

Met de nieuwste release van BizTalk, BizTalk Server 2009, worden we als ontwikkelaars behoorlijk verwend. De belangrijkste verbetering in het product is de volledige ondersteuning van Visual Studio Team System 2008, .NET Framework 3.5 SP1 en ondersteuning voor Windows Server 2008 en SQL Server 2008. Dit artikel gaat kort in op alle nieuwe functionaliteiten.

**Sinds de release** van BizTalk Server 2006 R2 in 2007 is er het nodige veranderd aan het Core Microsoft platform. Om optimaal te kunnen profiteren van nieuwe versies van .NET, Visual Studio, Windows Server en SQL Server is een nieuwe release van BizTalk Server nodig.

BizTalk Server 2009 is een volledige release van BizTalk en dus meer dan alleen maar een servicepack waarin de laatste platformupdates zijn meegenomen. Naast de ondersteuning voor het nieuwe platform biedt BizTalk Server 2009 vernieuwde ondersteuning op het gebied van SOA en Web Services, Business to Business Integratie en Application Lifecycle Management.

### SOA & Web Services

Een integratieserver zoals BizTalk Server wordt veelvuldig ingezet in scenarios waarbij BizTalk Web Services aanbiedt of consumeert of de brug is naar platforms die (nog) geen Web Services ondersteunen. Met de komst van WCF in .NET 3.0 en het WCF Adapter Framework in BizTalk Server 2006 R2 zijn enorme stappen voorwaarts gemaakt op het gebied van SOA & Web Services. Wederom zijn in BizTalk Server 2009 investeringen gedaan op dit gebied. Onderdeel van de nieuwe functionaliteit van BizTalk is een implementatie van UDDI 3.0, hetgeen betekent dat UDDI in Windows Server op termijn zal verdwijnen. Daarnaast zijn er verbeteringen doorgevoerd op het gebied van BAM en zijn er nieuwe WCF gebaseerde adapters voor Oracle EBS, SQL Server, MQSeries, CICS en IMS.

De ESB Guidance, die vlak na de release van BizTalk Server 2009, uitgebracht zal worden bevat zeer nuttige richtlijnen voor het integreren van BizTalk in een SOA-landschap.

### Business to Business integratie

Ook op het gebied van B2B Integratie heeft het BizTalk-team nieuwe functionaliteit ingebouwd. Het RFID-platform is uitgebreid met een mobiel RFID-platform en de laatste ondersteuning van diverse industriestandaarden. Voor de financiële (bancaire) wereld is er verbeterde SWIFT ondersteuning ingebouwd. Daarnaast zijn er verbeteringen doorgevoerd in de EDI en AS2 adapters.

### Application Lifecycle Management

De ALM ondersteuning in BizTalk Server 2009 is voor ons als ontwikkelaars natuurlijk verreweg het meest interessant. Maar wat betekent dat nu eigenlijk ALM ondersteuning? Onder ALM ondersteuning verstaan we – in dit verband – voornamelijk de lifecycle die doorlopen wordt tijdens het ontwikkelen van een applicatie. Het gaat hierin vooral om het ontwerp-, ontwikkel-, build- en testproces.

Voorname probleem in BizTalk Server 2006 R2 was dat de ontwikkeltools voor BizTalk uitsluitend geïnstalleerd konden worden bovenop Visual Studio 2005. Voor het compileren van BizTalk oplossingen had je dan ook altijd Visual Studio 2005 (devenv.exe) nodig. In omgevingen waar men gewend is dit op basis van Team Foundation Server en TeamBuild uit te voeren moest via diverse omwegen een BizTalk applicatie opgenomen worden in het (Daily) buildproces. In BizTalk Server 2009 is het BizTalk projectformaat aangepast, zodat deze aansluiten op het MSBuild projectformaat zoals andere projecttypes sinds Visual Studio 2005 hebben.

Hiermee is het vele malen eenvoudiger en sneller om via een Daily Build proces een BizTalk applicatie te compileren en te deployen naar een testomgeving, zodat de applicatie onderworpen kan worden aan diverse tests. Voor het deployen van een BizTalk applicatie kan gebruik gemaakt worden van de BTSTask.exe, Explorer Object Model, PowerShell of de grote hoeveelheid MSBuild tasks die te vinden zijn op CodePlex.

Op het gebied van Unit Testing bevat BizTalk Server 2009 out-of-the-box Unit Testing ondersteuning voor:

- + Maps
- + Schemas
- + Pipelines

Om de Unit Testing functionaliteit te gebruiken moet deze aangezet worden in de Deployment eigenschappen van een BizTalk project.

Maps kunnen getest worden via de TestMap operatie op een map waarmee het mogelijk is een inputfile en een outputfile mee te geven. Maps worden in BizTalk Server 2009 afgeleid van Microsoft.

BizTalk.TestTools.Mapper.TestableMapBase waarin onder andere de TestMap operatie is gedefinieerd. Na het uitvoeren van een map in je Unit Test kun je in je Unit Test verificaties uitvoeren ten opzichte van de outputfile door deze bijvoorbeeld te vergelijken met een van tevoren gedefinieerde outputfile (known good) of specifieke asserts in te voegen op variabelen in de (XML) file via Xpath expressies. In codevoorbeeld 1 is een voorbeeld te zien van een eenvoudige unit test waarin een map wordt getest.

```
[TestMethod]
public void TestMap_Transformation()
{
    string strSourcePO_XML = testContextInstance.TestDir +
    "..\\..\\..\\SamplePOInput.xml";
    string strDestInvoice_XML = testContextInstance.TestDir +
    "..\\..\\..\\SampleInvoiceOutput.xml";

    // Test de map door gebruik te maken van de TestMap methode
    POTOInvoice testmap = new POTOInvoice();
    testmap.ValidateOutput = false; // We will validate manually
    as an example
    testmap.TestMap(strSourcePO_XML,

    Microsoft.BizTalk.TestTools.Schema.InputInstanceType.Xml,
        strDestInvoice_XML,

    Microsoft.BizTalk.TestTools.Schema.OutputInstanceType.XML);

    // Controle of het outputbestand aangemaakt is
    Assert.IsTrue(File.Exists(strDestInvoice_XML));
}
```

CODEVOORBEELD 1.

Het testen van schemas kan op een vergelijkbare manier als maps door bestaande (know good) instance documenten te valideren tegen de schemas in een oplossing. Ook hiervoor is een methode toegevoegd met de naam ValidateInstance, welke gedefinieerd is in Microsoft.BizTalk.TestTools.Schema.TestableSchemaBase. In codevoorbeeld 2 is een voorbeeld te zien van een eenvoudige unit test waarin een schema wordt getest.

```
[TestMethod]
public void TestSourceSchema()
{
    string strSourcePO_XML = testContextInstance.TestDir +
    "..\\..\\..\\SamplePOInput.xml";

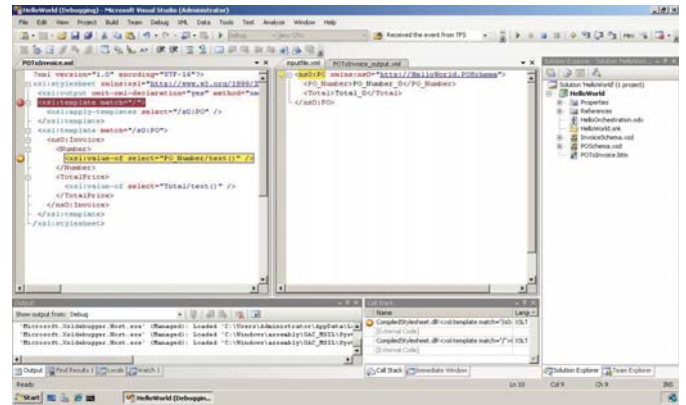
    // Valideer het SamplePOInput bericht tegen het schema via de
    ValidateInstance methode
    POSchema TestPOSchema = new POSchema();
    Assert.IsTrue(TestPOSchema.ValidateInstance(strSourcePO_XML,

    Microsoft.BizTalk.TestTools.Schema.OutputInstanceType.XML));

    testContextInstance.Properties.Add("SourcePath", strSourcePO_XML);
}
```

CODEVOORBEELD 2.

Tenslotte is het testen van een pipeline component nu ook mogelijk doordat pipelines afleiden van de TestableReceivePipeline of TestableSendPipeline classes die gedefinieerd zijn in de Microsoft.BizTalk.TestTools.Pipeline namespace. Beiden bevatten een operatie TestPipeline met diverse overloads om documenten, parts en schema's te voeden aan de Pipeline. De output van de pipeline kan vervolgens wederom vergeleken worden met van tevoren vastgestelde 'known-goods'. Voor het Functioneel (Unit) Testen van Orchestrations wordt geadviseerd gebruik te maken BizUnit, een framework dat ont-



FIGUUR 1.

wikkeld is door Microsoft Services in de UK, wat vrij verkrijgbaar is via CodePlex.

## Het debuggen van maps

De BizTalk mapper is een tool waarmee eenvoudige tot zeer complexe transformaties van berichten ontwikkeld kunnen worden. Uiteindelijk is een BizTalk map een XSLT transformatie. Vanaf BizTalk Server 2009 wordt het mogelijk om BizTalk maps te debuggen! Je kunt met deze functionaliteit de uitvoering van je Unit Tests debuggen en zo vanuit je (C#) Unit Test direct doorstappen naar een XSLT weergave van je map. Zoals je van een debugger mag verwachten, biedt deze functionaliteiten zoals een locals window, watches en breakpoints. Zie figuur 1 voor een screenshot van een BizTalk map met een breakpoint.

## Conclusie

Het heeft even geduurd totdat BizTalk de aansluiting heeft gevonden met de laatste platforminnovatie, maar de verbeteringen die doorgevoerd zijn, zijn absoluut productiviteitsverhogend voor ontwikkelaars. Het is nu echt mogelijk om effectief in teamverband BizTalk Server applicaties te ontwikkelen. Het is vele malen eenvoudiger om best practices zoals Unit Testing, (Daily) Builds, Build Verification Tests en andere functionaliteiten in Visual Studio Team System, Microsofts Application Lifecycle Management Suite, ook in de BizTalk wereld toe te passen.

BizTalk Server 2009 wordt uitgebracht in het tweede kwartaal van 2009. De informatie in dit artikel is gebaseerd op de publieke beta van BizTalk Server 2009.

## Links

BizTalk Server roadmap informatie: <http://www.microsoft.com/biztalk/en/us/roadmap.aspx>

Developing Integration Solutions using BizTalk Server 2006 and Team Foundation Server: <http://msdn.microsoft.com/en-us/library/dd297495.aspx>

BizUnit : <http://www.codeplex.com/bizunit>

SDCTasks : <http://www.codeplex.com/sdctasks>



**Dennis Mulder**, is Senior Development Consultant bij Microsoft Services. Hij is doorgaans vinden op de BTUG (BizTalk Usergroup) of te bereik via email [dennis.mulder@microsoft.com].