Clemens Reijnen is werkzaam als Technologie Adviseur bij Sogeti Nederland BV

Deployment van BizTalk Solutions

MOGELIJKHEDEN VOOR HET UITROLLEN VAN EEN BIZTALK-PROJECT

Het installeren van een BizTalk-project is een activiteit die je op verschillende manieren en met verscheidene tools kunt uitvoeren. Welke je hiervan gebruikt is afhankelijk van de status waarin het project zich bevindt en wie de deployment uitvoert. In dit artikel gaat de auteur in op wat er precies moet gebeuren tijdens het installeren van een BizTalk-project. Ook gaat hij in op de verschillende tools die tijdens het deployment-proces beschikbaar zijn.

Wat altijd hetzelfde blijft tijdens het installeren van een BizTalk-project zijn de vele acties die gedaan moeten worden voor, tijdens en na de installatie. Dit komt voornamelijk doordat een BizTalk-solution een samenvoeging is van assemblies en het configureren van een serveromgeving. Wanneer een complex project met meer orchestrations, poorten, schema's en pipelines geïnstalleerd gaat worden, kan het aantal acties dat doorlopen moet worden makkelijk boven de honderd uitkomen. Het is niet gewenst om al deze acties tijdens het ontwikkelen, testen en in productie nemen van de BizTalk-solution handmatig te doorlopen voor iedere orchestration in het project.

Pre-deployment acties

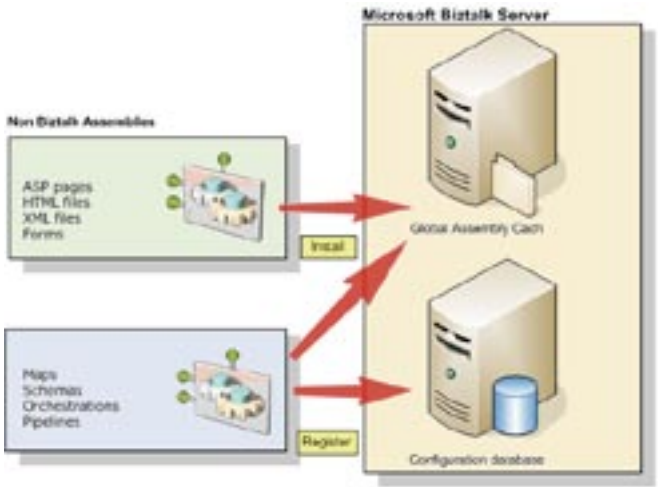
Microsoft BizTalk Server verwacht dat alle gebruikte assemblies zijn geïnstalleerd in de Global Assembly Cache (GAC) en dat de BizTalk-assemblies zijn geconfigureerd in de configuratiedatabase van BizTalk; zie afbeelding 1. Hierdoor moet, voordat met de installatie kan worden begonnen, eerst worden bekeken welke assemblies er al geïnstalleerd zijn in de GAC en of aan alle projectgerelateerde assemblies een strong name is toegewezen.

Wanneer het project nog in de ontwikkelfase zit, kan dit eenvoudig bestudeerd worden met de BizTalk Explorer, die beschikbaar is in Visual Studio. Hier kan de assembly-naam, versie, culture en het public key token worden bestudeerd. Wanneer het project al op een productieserver staat of in een testomgeving waarmee geen connectie kan worden gelegd vanaf de ontwikkelomgeving, dan kunnen ontwikkelaars niet beschikken over de voor ontwikkelaars vertrouwde Visual Studio IDE. In die situatie moeten we terugvallen op de twee mogelijkheden die Microsoft hiervoor biedt.
De eerste is de BizTalk Assembly Viewer. Deze heeft naast het bekijken en verwijderen van assemblies geen toegevoegde waarde. De viewer wordt niet standaard geïnstalleerd bij de installatie van Microsoft BizTalk Server, maar kan eenvoudig worden geconfigureerd door het *registeren* van de dll genaamd BtsAsmExt.dll.
De andere standaard mogelijkheid om al geïnstalleerde BizTalk-assemblies te bekijken, is de BizTalk Administration Console. Hiermee kunnen orchestrations en poorten gestart, ge-enlist en uiteraard bekeken worden. Orchestrations en poorten kunnen niet met de Console worden verwijderd. Natuurlijk kan je ook altijd in de GAC zelf nagaan welke assemblies er instaan en om welke versie het gaat. Het is logisch dat deze acties moeten worden uitgevoerd voordat een nieuw BizTalk-project wordt gedeployed. Het is namelijk niet mogelijk om twee dezelfde assemblies in de GAC te plaatsen. Dit geldt ook voor niet-BizTalk-assemblies in het project.

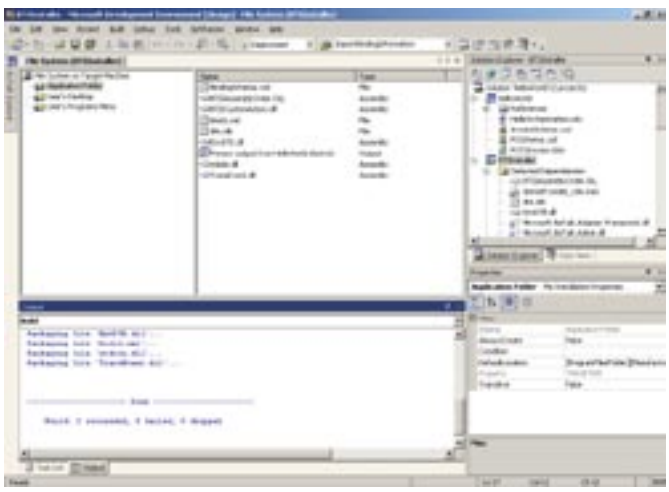
Solution Deployment-methoden

Wanneer er geen assemblies met dezelfde naam en hetzelfde versienummer meer aanwezig zijn, de nieuwe assemblies een strong name hebben gekregen, het project gebuild is en een bindingfile is gemaakt, is het project klaar om geïnstalleerd te worden. Microsoft BizTalk Server biedt vier manieren om een project te deployen.

Visual Studio .NET kan alleen gebruikt worden tijdens de ontwikkelfase van het project en niet voor het installeren en configureren



Afbeelding 1. Installatie en registratie van assemblies.



Afbeelding 2. Deploymentproject BTSintaller in Visual Studio.

van productie- of acceptatieomgevingen. Het deployen van een solution vanuit de Visual Studio IDE is heel eenvoudig. In het Build-menu in de menubalk zijn er de opties Deploy Project om een enkele orchestration te deployen of Deploy Solution om de gehele BizTalk Solution in een keer te installeren en configureren. Met de BizTalk Explorer is het mogelijk om de assemblies te undeployen. Als er echter een connectie is gelegd met een externe machine dienen de assemblies niet uit de GAC worden te verwijderd.

De tweede en derde methoden om een Biztalk-applicatie te deployen zijn hoofdzakelijk gericht op het inrichten van de productieomgevingen. Dit zijn de commandline-applicatie, BTSDeploy.exe, en de op deze commandline-applicatie gebaseerde BiztalkDeployment Wizard. Buiten het deployen en undeployen van assemblies kunnen deze twee tools ook bindingfile's importeren en exporteren. Met de commandline-tool kunnen de verschillende installatiestappen op verschillende servers eenvoudig herhaald worden wanneer deze in een batchprogramma achter elkaar worden gezet. Het uitvoeren van deze batchprogramma's kan eenvoudig door het releasemanagement worden uitgerold. Het toepassen van de Deployment Wizard is beduidend meer werk en zal dan ook nooit gebruikt worden voor grote en complexe projecten. Bij beide procedures dienen natuurlijk alle benodigde assemblies beschikbaar te zijn. Dus nadat de solution is gebuild, moeten de BizTalk-assemblies via het netwerk of een mediadrager benaderbaar zijn. Het is ook mogelijk om de assemblies op de BizTalk Server te kopiëren met behulp van een installatiepakket waarna de commandline-utility, als custom action, in het installatiepakket de verschillende deploy-stappen uitvoert. Dit dient zeer zorgvuldig ingeregeld te worden. Wanneer een configuratiestap van de commandline-tool namelijk een fout genereert, stopt de installatieprocedure en blijft de BizTalk Solution 'half' geïnstalleerd achter op de server. Hierdoor zal het uninstallen van de applicatie ook niet meer lukken en blijft de releasemanager achter met een 'halve' BizTalk Solution.

De laatste en vierde methode die wordt geboden om een BizTalk–solution te deployen, is het maken van een deployment-project met Visual Studio .NET. In de directory 'SDK\Utilities\' staat al een eerste aanzet voor een package-project; zie afbeelding 2. Om een Windows Installer (MSI) Package te genereren moet dit package-project toegevoegd worden aan de BizTalk-solution in Visual Studio. Hierna moet de solution-output van het BizTalk-project worden toegewezen aan het package-project. Visual Studio voegt dan zelf alle benodigde dependencies toe. Dit is in normale Windows- of consoleapplicaties gewenst, maar bij BizTalk-projecten worden ook alle BizTalk-assemblies als dependency herkend, wat als gevolg heeft dat de MSI die gegenereerd wordt al deze dependencies in zich heeft. Wanneer deze applicatie wordt gedeïnstalleerd, worden ook alle BizTalk-assemblies verwijderd van de server. BizTalk zal dan niet meer werken. De enige oplossing is dan het opnieuw installeren van de complete BizTalk-omgeving. Deze assemblies horen dan ook ge-*exclude* te worden uit het project.

De MSI die op deze manier wordt gegenereerd, kan vervolgens worden doorgegeven aan het releasemanagement dat hem op de gebruikelijke manier kan installeren. Het grote voordeel van deze deploy-methode is dat men geen rekening hoeft te houden met de volgorde waarin de orchestrations geïnstalleerd dienen te worden. Dit wordt binnen de MSI gedaan. Deze methode heeft een groot nadeel. Stel dat het installeren van een orchestration mislukt, bijvoorbeeld doordat er een assembly met dezelfde naam en key in de GAC staat, dan nog zal de installatieprocedure slagen. Als je wilt ontdekken wat er verkeerd is gegaan, dan kan dat alleen door het logfile te bekijken dat tijdens de installatie is opgebouwd.

```
using System;
using System.Management;
using System.Text;

namespace StopOrchestrations
{
   public class StopOrchestrations
   {
      public StopOrchestrations()
      {
         ManagementObjectSearcher Searcher = new ManagementObjectSearcher();
         ManagementScope Scope = new ManagementScope(
            "root\\MicrosoftBizTalkServer");

         Searcher.Scope = Scope;
         SelectQuery Query = new SelectQuery();

         StringBuilder queryString = new StringBuilder();
         queryString.Append("SELECT * FROM MSBTS_Orchestration");

         Query.QueryString = queryString.ToString();

         Searcher.Query = Query;
         ManagementObjectCollection QueryCol = Searcher.Get();

         foreach (ManagementObject Inst in QueryCol)
         {
            ManagementBaseObject inParams = Inst.GetMethodParameters("Stop");
            inParams["AutoDisableReceiveLocationFlag"] = 2;
            inParams["AutosuspendOrchestrationInstanceFlag"] = 2;
            ManagementBaseObject outParams = Inst.InvokeMethod("Stop",
               inParams, null);
         }
      }
   }
}
```

Codevoorbeeld 1. Instanties van de BizTalk Core Server Classes aanspreken en gebruiken.

Post-deployment acties

Alle BizTalk-orchestrations maken gebruik van receive- en send-poorten. Hoewel de poorten tijdens de ontwikkelfase aan een orchestration worden gekoppeld, kunnen ze onafhankelijk van de orchestration worden aangemaakt. Deze poorten moeten na het deployen van een BizTalk-assembly nog wel aangemaakt en gekoppeld worden aan de orchestrations. Tijdens de deploy-methode met de Visual Studio IDE kan door middel van een eigenschap worden aangegeven of de binding van de poorten logical, physical, direct, dynamic of role-gebaseerd is. Logical binding ('Specify later') betekent dat de poort pas later aan een orchestration wordt gekoppeld. Tijdens de deploy vanuit de IDE worden de poorten met deze optie niet aangemaakt. Dit moet handmatig gebeuren met een bindingfile. Een poort met de eigenschap Physical binding ('Specify now') wordt direct aangemaakt. Na het undeployen van een assembly worden in de IDE de poorten nooit verwijderd. Dit zal handmatig moeten plaatsvinden.

Voor het creëren van poorten in de test- en productieomgeving moet een bindingfile geïmporteerd worden. Deze bindingfile kan gegenereerd en geïmporteerd worden met behulp van de BizTalk Depoyment Wizard, Deze kan ook meegegeven worden aan de consoleapplicatie BTSInstall. In de meeste gevallen wordt een bindingfile gegenereerd op de ontwikkelmachine. Als er bij de portdefinities afhankelijke configuratie-items staan, moeten deze gegevens aangepast worden, voordat gedeployed wordt op een andere machine. Bijvoorbeeld als een poort gebruikmaakt van een SQL-adapter zal de SQL-serveromgeving tijdens de ontwikkelfase een andere naam hebben dan tijdens productie of test.

Een bindingfile is een xml-bestand dat eenvoudig aan te passen is. Wel kan het erg onoverzichtelijk worden, afhankelijk van de complexiteit van de orchestration en het aantal poorten dat wordt gebruikt.
Nadat de poorten zijn aangemaakt, dienen ze nog ge-*enlist* en gestart te worden voordat de orchestrations kunnen starten. Het enlisten en starten in de productieomgeving kan met de BizTalk Server Administration-applicatie. In de ontwikkelomgeving kan dit met BizTalk Explorer in Visual Studio IDE. Wanneer het enlisten en starten van poorten en orchestrations tijdens de deploy ingeregeld dienen te worden, moet je gebruikmaken van de in de SDK meegeleverde VB-scriptfiles. Deze staan in de '\Admin\WMI' BizTalk Server sample-directory. Alle voorbeelden die worden meegeleverd in de SDK maken gebruik van deze scripts om het voorbeeld op te zetten. Dit is een goede start om te bekijken hoe je deze scripts precies moet gebruiken.

GotDotNet Custom BizTalk Deployment-tools

Alle stappen die zijn beschreven behoren nog tot de standaard-procedures van het installeren van een BizTalk-solution. Tijdens het ontwikkelen van de applicatie wordt een project verscheidene keren op een dag getest. Op deze manier kost het veel tijd om de gehele applicatie in te stellen, om maar niet te spreken over RSI. Dit heeft geleid tot enkele zelfgemaakte tools die het deployment-proces vereenvoudigen en beter beheersbaar maken. Deze tools maken alle gebruik van WMI (Windows Management Instrumentation) Classes. Dit zijn classes die gebruikt kunnen worden om de belangrijkste objecten van Microsoft BizTalk Server aan te kunnen spreken. De VB-scripts die in de SDK zijn te vinden maken ook gebruik van deze classes. Alle beschikbare WMI Classes kunnen bekeken en aangestuurd worden met CIM Studio, dat wordt meegeleverd met de WMI SDK. Om deze classes te kunnen aanspreken in een .NET-project dient er een referentie te zijn naar de System. Management-namespace. Vervolgens is het eenvoudig om instanties van de BizTalk Core Server Classes aan te spreken en te gebruiken; zie codevoorbeeld 1.

Deployment Framework for BizTalk 2004 Applications

De populairste applicatie op GotDotNet is het 'Deploy Framework for BizTalk 2004' van Scott Colestock. Het Deployment Framework kan vanuit de Visual Studio IDE gebruikt worden voor developer-deployments en kan een installatiepakket genereren voor installatie naar een server. Als het om functionaliteit gaat is dit de compleetste custom-tool die op GotDotNet is te vinden. De gehele applicatie maakt gebruik van NAnt, een op XML-gebaseerd softwarebuild-systeem. Om het mogelijk te maken dat NAnt ook BizTalk-applicaties kan deployen, zijn er verschillende extra assemblies aan NAnt toegevoegd die gebruikmaken van het BizTalk Explorer Object Model en de BTSDeploy commandline-tool. Om het framework te kunnen gebruiken moet NAnt het deployment-framework op de computer installeren. Vervolgens hoeven er alleen maar twee NAnt-bestanden aan het project toegevoegd te worden, en de bindingfile en verschillende projectspecifieke namen ingesteld te worden. Via de commandfile van NAnt kan de build en deploy nu gestart worden. Om de deploy te gebruiken in Visual Studio kan de commandfile toegevoegd worden aan de VS.NET External-tools. Voor het deployen naar een server wordt gebruik gemaakt van WiX om een MSI te genereren. WiX is een open-source toolset uitgebracht door Microsoft; in .NET Magazine #6 zijn de mogelijkheden van WiX beschreven door Tom Mertens. De werking van de MSI wordt in XML beschreven en met een aantal tools wordt deze XML omgezet naar een MSI. De gegenereerde MSI bestaat uit de assemblies van de BizTalk-applicatie, inclusief de bindingfiles en de benodigde NAnt-bestanden om de applicatie daadwerkelijk te installeren. De installatieprocedure bestaat dan ook uit twee stappen. Als eerste installeer je het installatiepakket op de server. Deze installatie maakt in het startmenu een menustructuur aan om de Biztalk Solution daadwerkelijk te installeren; zie afbeelding 3.

Afbeelding 3. Menustructuur om de Biztalk Solution te installeren.

BizTalk Transactional Deployment Application

Het grote probleem bij het installeren van een serveromgeving tijdens de productiefase is dat de installatieprocedure in de meeste gevallen uitgevoerd zal worden door iemand die niet de BizTalk-applicatie heeft gemaakt. Dit zal de beheerder of de releasemanager zijn, waarbij de kennis van Biztalk meer op het gebied van monitoring (HAT, Healt and Tracking-monitoring) zal liggen dan op het gebied van enlisten, starten, stopen en binden van de solution en de volgorde waarin de applicatie geconfigureerd dient te worden. Als er iets misgaat tijdens de installatie, omdat bijvoorbeeld een BizTalk-assembly al in de GAC staat, zal in de meeste gevallen de installatie stoppen (BTSDeploy, VB-Scripts) of er zal een fout in een logfile worden gegenereerd (BTSInstaller). Maar altijd is de installatie niet volledig doorlopen en dienen handmatig stappen te worden ondernomen om de 'halve' installatie ongedaan te maken en opnieuw uit te voeren.

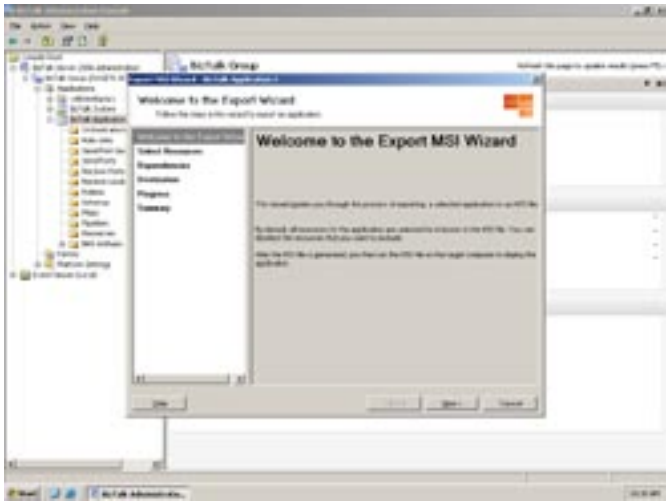
Met behulp van de consoleapplicatie - BizTalk Transactional Deployment Application - kun je de verschillende stappen uitvoeren die tijdens de installatie doorlopen moeten worden. Wanneer een van de stappen niet slaagt, zal de applicatie alle voorgaande stappen ongedaan maken en de stap waarbij de fout optrad melden in de EventLog. De console controleert of alle geplande stappen wel uitgevoerd kunnen worden voordat er daadwerkelijk geïnstalleerd gaat worden. Er wordt bijvoorbeeld gekeken of de benodigde bestanden aanwezig zijn, of de assemblies die geïnstalleerd gaan worden niet al in de GAC staan en natuurlijk of de persoon die de consoleapplicatie uitvoert wel voldoende rechten heeft om de BizTalk-applicatie te installeren. Dit gebeurt door het aanroepen van de 'Can'-methoden binnen de BizTalk-objecten van de consoleapplicatie; zie afbeelding 4.

De volgorde van de stappen die moeten worden doorlopen voor het installeren en configureren moeten in dezelfde volgorde in het configuratiebestand staan. Het instellen van deze installatiestappen zal dus moeten gebeuren door de ontwikkelaar die de applicatie gemaakt heeft en kennis heeft van de onderdelen van de applicatie.

Andere custom-tools

BizTalk 2004 Management Tool is een must-have in de productieomgeving als er na de installatie handmatig instellingen moeten

Afbeelding 4. Class-diagram, Transactional Deployment Application.



Afbeelding 5. BizTalk 2006 Export MSI Wizard.

worden doorgevoerd. Het is ook mogelijk om met deze Windows-applicatie verscheidene assemblies te installeren. Een echte developertool is BizTalkAutoDeploy. Deze wordt geïntegreerd in de Visual Studio IDE waardoor met een muisklik een solution geïnstalleerd kan worden.

BizTalk 2006

Dat het deployen, configureren en managen van een BizTalk Solution geen eenvoudige klus is en veel verschillende stappen omvat is bekend. Microsoft heeft in de komende versie van BizTalk (2006) dan ook duidelijke stappen genomen. Zo zal er een volledig nieuwe Management Console beschikbaar komen, waarbij het eenvoudiger wordt om solutions te beheren en te deployen. BizTalk Server 2006 zal de mogelijkheid hebben om van een complete BizTalk Solutions een MSI te maken met behulp van Microsoft Windows Installer via de Management Console.

Clemens Reijnen is als Technologie Adviseur werkzaam bij Sogeti (www.sogeti.nl). Naast zijn huidige functie geeft hij les op de academie van Sogeti in de verschillende Microsoft-producten en begeleidt hij collega's in de voorbereiding naar het MCSD.NET Certificaat. E-mail: Clemens.Reijnen@Sogeti.nl

Nuttige internetadressen

Microsoft BizTalk Core Server Classes:
http://msdn.microsoft.com/library/en-us/sdk/htm/ebiz_sdk_wmi_reference_core_stxe.asp

Deploying BizTalk Server Assemblies:
http://msdn.microsoft.com/library/en-us/deploying/htm/ebiz_asdepl_howto_nbbr.asp

Deployment Framework for BizTalk 2004 Applications:
www.traceofthought.net/PermaLink,guid,09ad4181-dde2-4906-884b-d023f80fc800.aspx

BizTalk Transactional Deployment Console Application:
www.gotdotnet.com/Workspaces/Workspace.aspx?id=153aedcf-809f-4e6c-850a-9d13acfccc0a

BizTalk 2004 Management Tool:
www.gotdotnet.com/Workspaces/Workspace.aspx?id=992ca223-553c-475a-ac87-da7ae2c9016a

BizTalkAutoDeploy:
www.gotdotnet.com/Workspaces/Workspace.aspx?id=62d94220-c0e0-46d4-a2d6-85d3d911467a

Visualstudio IDE:
http://biztalkers.blogspot.com/2004/09/biztalk-auto-deploy-project-progress_17.html