

Eindelijk een serieuze plaats voor de database in Visual Studio

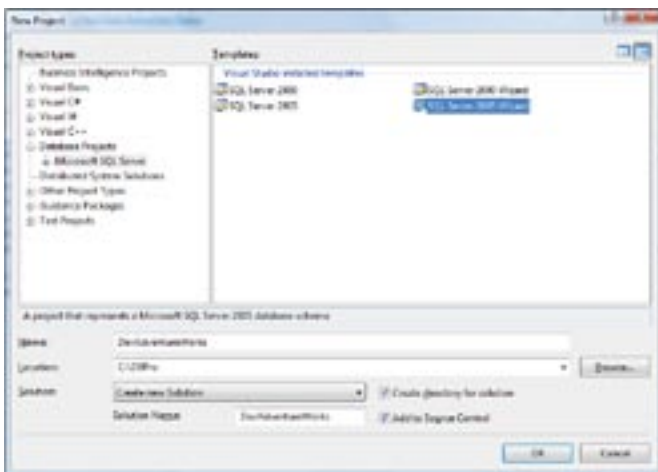
VISUAL STUDIO 2005 TEAM EDITION FOR DATABASE PROFESSIONALS

Vrijwel elke bedrijfsapplicatie gebruikt wel een database. Toch is de ondersteuning van de database in het ontwikkeltraject tot nu toe minimaal geweest. Met de komst van Visual Studio 2005 Team Edition for Database Professionals gaat dit veranderen. Met een download-grootte van nog geen 8 MB vraag je je toch af wat je daarvoor krijgt. Lees verder en ontdek het antwoord op deze vraag.

Team Edition for Database Professionals, hierna kortweg DB Pro, is verkrijgbaar als aparte editie van Visual Studio 2005. DB Pro is echter ook als add-on van bijna 8 MB te installeren op Visual Studio 2005 Team Suite. Dat Microsoft een serieuze rol ziet weggelegd voor de database in het ontwikkeltraject blijkt ook uit de nieuwe MSF Guidance for Database Development. De update voor MSF for Agile Software Development en MSF for CMMI Process Improvement onderkent nu eindelijk de rol van database- developer en databaseadministrator in de levenscyclus van een applicatie. DB Pro richt zich op activiteiten die je typisch terug ziet komen in een ontwikkelproces, te weten: databaseprojectmanagement, datageneratie, unit-testing, refactoring, comparing en building & deploying. We zullen één voor één naar deze onderwerpen kijken, maar we beginnen bij het begin: een databaseproject.

Databaseproject

Uiteraard kun je met DB Pro een nieuwe database maken, maar waarschijnlijk heb je al een bestaande database die je in Visual Studio wil beheren. We gaan in de rest van het artikel uit van een bestaande database, te weten de voorbeelddatabase AdventureWorksLT (te downloaden van de Microsoft- website, zie referenties). Wanneer je DB Pro installeert, krijg je een aantal nieuwe projecttemplates in Visual Studio; zie afbeelding 1. Ondersteunt DB Pro slechts SQL Server 2000 en 2005? Het antwoord is ja, in de huidige eerste versie wel. Sterker nog, DB Pro



Afbeelding 1. DB Pro-projecttemplates

vereist de aanwezigheid van een lokale SQL Server 2005 voor design-time validatie van zowel SQL Server 2000- als 2005-projecten! Helaas biedt DB Pro vooralsnog géén ondersteuning voor SQL Server Analysis Services. Kies voor de SQL Server 2005 Wizard-projecttemplate. Via een aantal wizardstappen kunnen we zo de bestaande AdventureWorksLT-database importeren naar onze nieuwe lokale ('sandboxed') DevAdventureWorksLT-database. Het eindresultaat is een databaseproject met een sql-file per schema-object, zie afbeelding 2. Direct wordt één van de voordelen zichtbaar: source control per databaschema-object!

Datageneratie

Het is lang niet altijd mogelijk, laat staan wenselijk, om productiedata te gebruiken in het ontwikkeltraject. Daarom biedt DB Pro zogenaamde Data Generation Plans voor het genereren van (test)data. Kies 'Add -> Data Generation Plan' op het DevAdventureWorksLT-project en kies een naam voor het plan. DB Pro zal nu een standaard plan samenstellen, waarbij in iedere tabel 50 rijen worden toegevoegd, zie afbeelding 3. Dit en ander standaard gedrag van DB Pro kan overigens worden gewijzigd onder 'Tools -> Options -> Database Tools'.

Selecteer de tabel Customer en de kolom FirstName. Merk op dat deze kolom wordt gevuld met een String Generator. In het Properties-venster kan nu een aantal eigenschappen van deze generator worden opgegeven, waaronder de minimum- en maximumlengte van de gegenereerde string. Verander bijvoorbeeld de Minimum Length in 5 om voornamen te genereren die uit ten minste vijf karakters bestaan. Kies vervolgens de Data Generation Preview-tab om een idee te krijgen van de data die in de Customer-tabel terechtkomen. Toets F5 om de data daadwerkelijk te genereren in een doeldatabase naar keuze. DB Pro zal hierbij vragen of de huidige gegevens verwijderd mogen worden. Afhankelijk van de datatype van de kolom kan er in het plan gekozen worden voor een andere Generator. Zo is er de Regular Expression Generator voor het genereren van bijvoorbeeld telefoonnummers of postcodes. Er is zelfs een Data Bound Generator voor het importeren van data uit een andere database. Ten slotte is het mogelijk je eigen Custom Generators te maken in managed code. Een eenvoudig voorbeeld is een Custom Generator voor het genereren van getallen die alle een veelvoud zijn van een bepaalde factor; zie codevoorbeeld 1. Een Custom Generator komt neer op het maken van een class die is afgeleid van de Generator-class. Vervolgens kunnen er zogenaamde input-properties worden gedefinieerd die als input-

```

using System;
using System.Data.SqlTypes;
using Microsoft.VisualStudio.TeamSystem.Data.DataGenerator;

namespace CustomGenerator
{
    public class FactorGenerator : Generator
    {
        private int _factor;
        private int _generatedNumber;
        Random _random;

        [Input(Name="Factor",
            Description="The integers generated are a multiple of this factor.",
            DefaultValue=1)]
        public int Factor
        {
            set { _factor = value; }
            get { return _factor; }
        }

        [Output]
        public int GeneratedNumber
        {
            get {return _generatedNumber;}
        }

        protected override void OnInitialize(GeneratorInit initInfo)
        {
            _random = new Random(this.Seed);
            base.OnInitialize(initInfo);
        }

        protected override void OnGenerateNextValues()
        {
            _generatedNumber = _random.Next(int.MaxValue/_factor) * _factor;
        }
    }
}

```

Codevoorbeeld 1.
Een Custom Generator voor het genereren van veelvouden van een factor

parameters dienen voor de generator. Deze input-properties kun je configureren in het generatieplan. In het geval van een String Generator is Minimum Length een voorbeeld van zo'n inputparameter, zie nogmaals afbeelding 3. Ook dient ten minste één [Output] property te worden gedefinieerd die de gegenereerde data oplevert. In codevoorbeeld 1 is dit de property GeneratedNumber. Het is mogelijk meer output-properties te definiëren. In het generatieplan kun je de gewenste output-property selecteren. Je kunt ervoor kiezen een generator deterministisch of non-deterministisch te maken. Deterministisch houdt in dat de generator bij iedere run dezelfde resultaten oplevert, non-deterministisch dat de resultaten verschillen. Bovenstaande generator is deterministisch gemaakt door een Random-object te instantiëren met de huidige Seed (zie de override van de methode OnInitialize). Seed is een property van de Generator-baseclass. Deze is te configureren in het generatieplan. Zolang de Seed hetzelfde is, genereert de generator herhaaldelijk dezelfde data. Het feitelijk generen van de data is de verantwoordelijkheid van de methode OnGenerateNextValues. Deze methode wordt aangeroepen voor iedere rij en voor iedere kolom waar gekozen is voor deze generator. Voordat we deze custom generator kunnen gebruiken, moeten er nog drie zaken geregeld worden: signing, registratie en plaatsen in de juiste folder. Zorg er ten eerste voor dat de C#-assembly, die bovenstaande class bevat, wordt gesigned. Gebruik vervolgens `sn.exe -T CustomGenerator.dll` om de public key token van de assembly op te vragen. Nu moeten we DB Pro laten weten dat we ook echt gebruik willen maken van onze custom generator. Dit doen we door de assembly te registreren

```

<?xml version="1.0" encoding="utf-8" ?>
<extensions assembly="CustomGenerator, Version=1.0.0.0,
    Culture=neutral, PublicKeyToken=TokenKey" version="1"
    xmlns="urn:Microsoft.VisualStudio.TeamSystem.Data.Extensions"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="urn:Microsoft.VisualStudio.TeamSystem.Data.
    Extensions Microsoft.VisualStudio.TeamSystem.Data.Extensions.xsd">
    <extension type="CustomGenerator.FactorGenerator" enabled="true" />
</extensions>

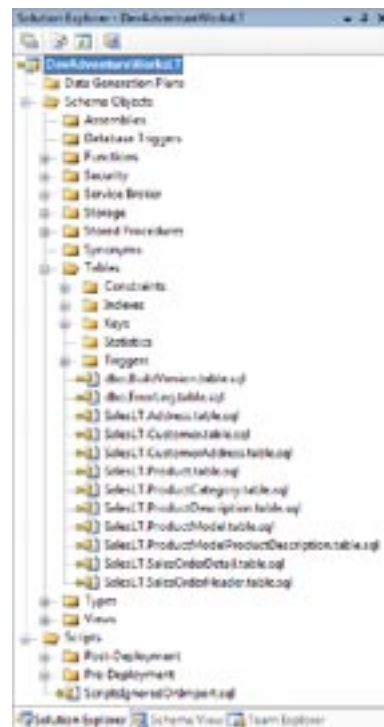
```

Codevoorbeeld 2.
Registratie van onze custom generator in CustomGenerator.Extensions.xml

in een daarvoor bestemde xml-file die we CustomGenerator.Extensions.xml noemen; zie codevoorbeeld 2. Vervang in dit codevoorbeeld TokenKey door de eerder opgevraagde public key token. Ten slotte dienen de xml-file en de assembly in de daarvoor bestemde folders geplaatst te worden. Het betreft [Program Files]Microsoft Visual Studio 8\DBPro voor de xml-file en [Program Files]Microsoft Visual Studio 8\Common\IDE\PrivateAssemblies voor de assembly. Sluit en open Visual Studio en happy data generation.

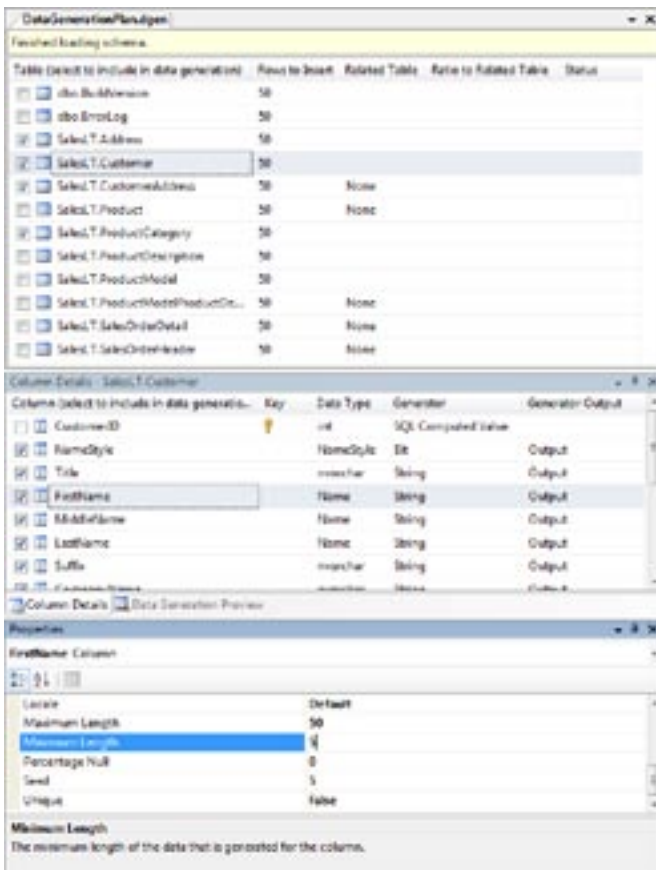
Unit-testing

Nu de database gevuld is met testdata kunnen we ook testscenario's loslaten op de database. Zo kunnen we stored procedures, functions

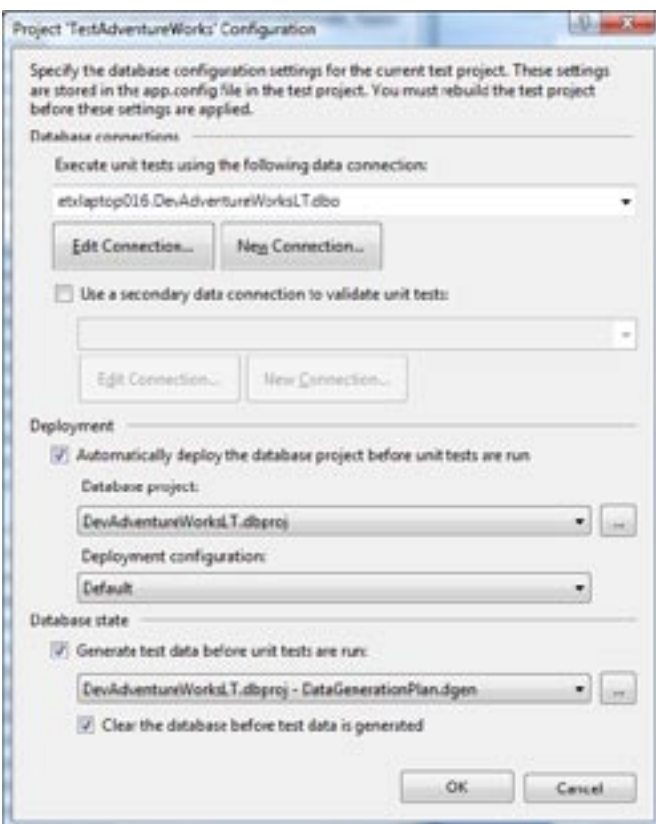


Afbeelding 2. Solution Explorer met Schema Objects

en triggers testen. Kies in de Schema View onder Functions `dbo.ufn-GetAllCategories`. Kies vervolgens 'rechter muisknop -> Create Unit Tests...'. We kunnen er nu voor kiezen de unit-test toe te voegen aan een nieuw of bestaand testproject. We kiezen in dit geval voor een nieuw testproject met de naam TestAdventureWorks en noemen de unit testclass `UserDefinedFunctionTest.cs`. Wanneer we verder gaan, krijgen we een aantal configuratieopties voor het testproject; zie afbeelding 4. In deze dialog bepalen we met welke databaseconnectie de unit-tests worden uitgevoerd en of we een aparte databaseconnectie (met bijvoorbeeld meer rechten) willen gebruiken om het eindresultaat te testen. We kunnen bovendien kiezen of we het databaseproject automatisch willen 'uitrollen' voordat de unit-tests worden uitgevoerd. Ten slotte kunnen we kiezen of we vóór de unit-tests een datageneratieplan willen uitvoeren. Wanneer we OK kiezen, komen we in de unit-test-designer terecht; zie afbeelding 5. Hier bepalen we de testconditie van de unit-test. De user-defined

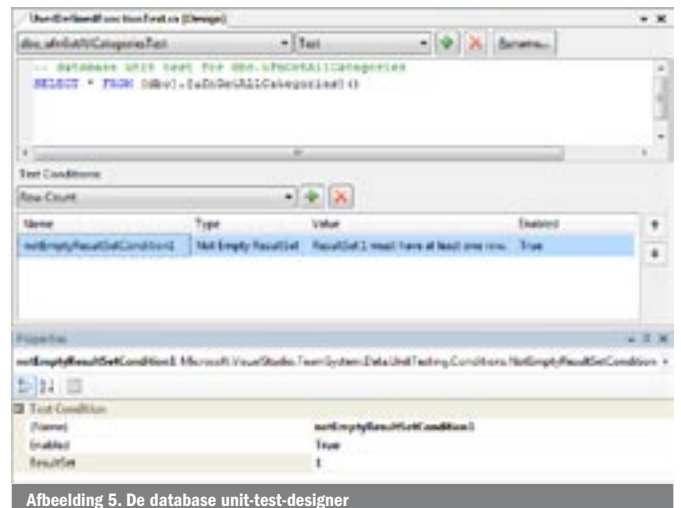


Afbeelding 3. Data Generation Plan



Afbeelding 4. Configureren van het testproject

function `ufnGetAllCategories` is een table-valued function. Hiervan verwachten we in ieder geval dat deze een resultset oplevert van ten minste één rij. Om dit te testen voegen we een Row Count-conditie toe van het type Not Empty Resultset. Kies ten slotte 'Test -> Start selected test ...' om de unit-test te starten.



Afbeelding 5. De database unit-test-designer

Refactoring

De refactoring-opties van DB Pro zijn in de huidige versie beperkt tot precies één: Rename. Rename kan worden toegepast op vrijwel alle schemaobjecten. Om deze optie te gebruiken, klik je een schemaobject aan in de Schema View en kies je 'rechter muisknop -> Refactor -> Rename'. Vervolgens kies je de nieuwe naam van het object en laat je de standaardopties aangevinkt. Wanneer je doorgaat, krijg je een Preview Changes-dialoog. Deze dialoog toont in een boomstructuur alle schemaobjecten, datageneratieplannen en overige SQL-scriptfiles die gewijzigd gaan worden.

Comparing

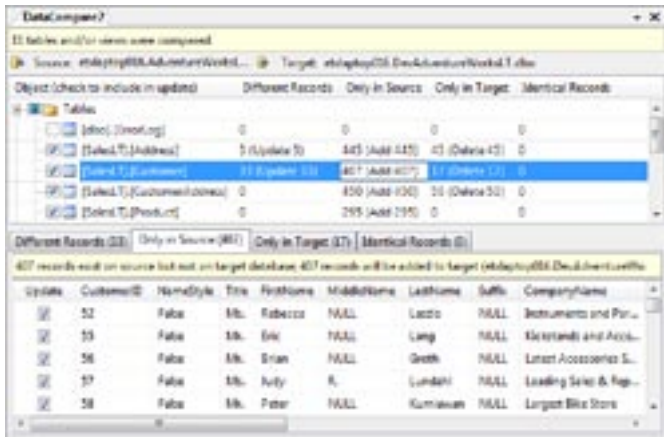
Wanneer je vermoedt dat er een verschil bestaat tussen twee databases, dan wordt het tijd deze te vergelijken. Voorheen was je hiervoor overgeleverd aan third-party tools, zoals die van RedGate. Nu kun je dit doen met DBPro, want Schema Compare en Data Compare in het Data-menu biedt wat de naam doet vermoeden: een optie om twee databaseschema's of de data in twee databases met elkaar te vergelijken. Schema Compare biedt zowel de mogelijkheid schema's van twee online databases met elkaar te vergelijken als die van een online database te vergelijken met de schemaobjecten uit een DB Pro-databaseproject. Data Compare daarentegen vergelijkt concreet de inhoud van tabellen en views; zie afbeelding 6. Data Compare rapporteert hoeveel en welke records verschillend zijn, alleen voorkomen in de ene database, alleen voorkomen in de andere database of exact overeenkomen. Zowel Schema Compare als Data Compare maken inzichtelijk wat de verschillen zijn en bieden de mogelijkheid deze verschillen, met een gegenereerd SQL-script, ongedaan te maken.

Building & deploying

Building en deploying van het databaseproject kunnen vanuit Visual Studio. De eigenschappen die hierbij worden gebruikt, zoals de databaseconnectie voor deployment, worden ingesteld onder de properties van het databaseproject. Hoewel dit handig is voor het deployen naar je lokale 'sandboxed' database wil je voor het deployen naar bijvoorbeeld een centrale testserver een andere strategie gebruiken. DBPro introduceert hiervoor een drietal MSBuild-tasks die je kunt gebruiken, te weten: build, deploy en clean. Helaas missen we in deze release de mogelijkheid een setup-project te maken dat resulteert in een MSI voor het uitrollen van database(wijzigingen).

Een blik op de toekomst

Op het moment van schrijven van dit artikel is service release 1 aangekondigd voor het tweede kwartaal van dit jaar. Naast oplossingen voor bekende kwesties zal deze release officiële ondersteuning bieden voor Microsoft SQL Server 2005 SP2 en Microsoft Windows Vista. Kijken we wat verder vooruit, dan mogen we de



Afbeelding 6. Data Compare

volgende features verwachten in toekomstige releases: designer-support, logische datamodelering, inclusief ondersteuning voor het synchroniseren met het fysieke datamodel, ondersteuning voor andere databases, meer refactoring-opties, IntelliSense en last but not least MSI-deployment.

Database op de kaart

We mogen concluderen dat Visual Studio 2005 Team Edition for Database Professionals eindelijk de database op de kaart heeft gezet van het ontwikkelproces in Visual Studio. De beperking tot ondersteuning van SQL Server 2000 en 2005 zal voor Microsoft-dedicated ontwikkeltrajecten niet zo'n kwestie zijn. Al met al biedt Visual Studio 2005 Team Edition for Database Professionals een brede ondersteuning van het ontwikkeltraject. Van databaseprojectmanagement, datageneratie en unit-testing tot buildmanagement-ondersteuning. Ik zie uit naar de volgende versies, want designer-support, IntelliSense en MSI-deployment zijn welkome aanvullingen.

René Sterrenburg is software-architect bij Qurius Advanced Solutions. Zijn e-mail-adres is rene.sterrenburg@qurius.nl.

Referenties

Visual Studio 2005 Team Edition for Database Professionals: <http://msdn.microsoft.com/vstudio/teamsystem/products/dbpro/default.aspx>
 AdventureWorksLT sample database: <http://www.microsoft.com/downloads>
 Gert Drapers (aka 'Data Dude', de Architect van DB Pro): <http://blogs.msdn.com/gertd>
 Samples-project: <http://www.codeplex.com/vsdbpro>