

Test

Team Edition for Database Professionals

Die Neue Rolle im Visual Studio Team System

Test der neuen Lösung von Microsoft für Datenbankspezialisten. Es wird aufgezeigt, was begeistert, was fehlt und wie man dieses Produkt im Jungle der Datenbank Helferlein bewerten kann.



Inhalt

Einführung3
Installation.....3
Features3
 DB Schema übertragen4
 DB Daten übertagen4
 Testdaten generieren4
 Prüfung von Skripten.....5
Was fehlt denn noch?5
Preis.....6
Fazit6

Einführung

Im Rahmen der Einführung des Visual Studio 2005 hat Microsoft im Rahmen der Team Suite schon Ende 2005 die Version für Architekten, Developer und Tester herausgebracht. Im Dezember 2006 hat Microsoft zusätzlich noch etwas für die Datenbankspezialisten auf den Markt gebracht. Diese neue Rolle nennt sich nun in der Langform Microsoft Visual Studio Team Edition for Database Professionals. Im Folgenden Kurzttest wird das Produkt einfach nur noch DB Tools genannt, man könnte es aber auch Data Dude nennen, wie es schon im Englischen durchaus üblich scheint.

Installation

Man kann eine 180 Tage Testversion aus dem Internet herunterladen. Dies sind bei dem Tools for Database Professionals nur 10 MB, aber man muss diese auf eine Testversion vom Visual Studio Team Suite on Top installieren. Diese Version kann man aus dem MSDN Medienset nehmen oder man kann gleich die neueste DVD als ISO Image mit den notwendigen Daten herunterladen. Es gilt hier die gleiche Sprachversion zu nehmen, wie auch später die DB Tools. Bestehende Visual Studio Versionen, die nicht Team System heißen müssen vor der Installation entfernt werden.

Ist man stolzer Besitzer der kompletten Visual Studio Team Suite, mit aktiver Subscription, so kann man diese Version direkt installieren und muss dann nur das Plug-In aus dem MSDN Bereich Download Bereich nachinstallieren.

Die DB Tools haben die gleichen Anforderungen an die Hardware wie ein normales Visual Studio 2005. Es muss jedoch mindestens die Express Edition des SQL Servers 2005 installiert sein. Da man diesen vor der Nutzung starten muss und meist die SQL Server Konsole offen hat, empfiehlt es sich 1 GB Hauptspeicher. Vor der Installation sollten idealer Weise noch mindestens 10 GB auf der Festplatte frei sein. Die Installation des SP1 für das Visual Studio ist unter Windows Vista zwingend erforderlich und unter XP SP2 und Server 2003 SP1/SP2 dringend anzuraten.

Die genauen Empfehlungen an die Hardware kann man auch recht einfach auf der Microsoft MSDN Seite ansehen. Man sollte allerdings nicht die minimal mögliche Variante auswählen, da man sonst nur noch am Kaffeetrinken ist...

Deinstallation der Team Edition für Developer, Installation und Patches haben auf dem Testsystem circa 90 Minuten in Anspruch genommen. Es wurde aber kein Neustart benötigt. Der Test wurde mit der englischen Version vorgenommen, da hier auch die Beta Extensions früher verfügbar sind. Laut einer Studie von Microsoft werden sowieso die meisten Visual Studio Installationen in Englisch vorgenommen. Die MSDN Hilfe wurde jedoch in Deutsch installiert, was ohne Probleme möglich ist. Um Speicherplatz zu sparen, kann man auch die MSDN online nutzen. Hierfür muss jedoch immer eine Internet Verbindung verfügbar sein. Als weitere Tools habe ich nur noch die Verbindung zum Team Foundation Server hergestellt, Enterprise Architect als UML Tool installiert und einige DLL's konfiguriert. Der SQL Server war auf dem Testsystem sowieso schon mit SQL Studio installiert.

Features

Was kann man mit den DB Tools groß machen ist die Frage am Anfang, wenn man das Tool noch nicht kennt. Wenn man die anderen Editionen des Team Systems kennt, denkt man vielleicht

auch, der Server ist Extraklasse, aber die Tools die in den Editionen enthalten sind lohnen sich nicht. Doch weit gefehlt, die DB Tools bieten deutlich mehr, als Ihre Konkurrenten. Die gebotenen Funktionen gliedern sich in 4 wesentliche Funktionen, die es aber so richtig in sich haben. Diese Funktionen hat schon fast jeder Entwickler einmal benötigt und es hat ihm zumeist viel Zeit und Nerven gekostet. Neben den als wesentlich erkannten 4 Hauptfunktionen bieten die DB Tool noch einiges mehr. Dies sollte man dann aber selbst erkunden und bewerten. Kommen wir nun zu den 4 Kernfunktionen...

DB Schema übertragen

Man kann das Schema einer Datenbank einlesen und mit einer anderen vergleichen. Dieser Vergleich kann zum Beispiel von der Produktion aus gehen und dann mit der Entwicklungsversion verglichen werden. Ist das Schema analysiert, so werden alle Unterschiede sehr übersichtlich auf Tabellenebene angezeigt. Die festgestellten Änderungen kann man dann auf der jeweils anderen Seite nachziehen. Hierzu kann man sich ein T-SQL Skript generieren lassen. Man muss nicht alle Änderungen nachziehen sondern kann selbstverständlich selektiv Änderungen auslassen. So kann man innerhalb kürzester Zeit ein Update der Datenbank erzeugen, was man dann per Verteilpaket, z.B. bei einem neuen Release der Software, verteilen kann.

Die Integrationsprüfung und die Korrektheit ist beim den vorgegebenen Datenbanken SQL Server 2000 oder 2005 extrem gut. Die Qualität der Skripte ist absolut Produktionstauglich und es kann sehr viel Zeit gespart werden, wenn man bisher die Skripte per Hand geschrieben hat.

DB Daten übertragen

Eben wurde das Schema ohne Daten übertragen... Hier geht es nun die Füllstand, wie zum Beispiel Testdaten oder Differenzen einer Migration zu übertragen. Auch hier ist das Vorgehen mit den DB Tools sehr intuitiv, sicher und absolut Produktionstauglich.

Zuerst führt man eine Analyse der Datenbank durch, diese geht in der Regel in wenigen Sekunden von der Hand, bei großen Datenbanken kann die Zeit je nach Rechner natürlich etwas ansteigen. Vor der Analyse muss man nur auswählen, welche Datensätze man betrachten will. Updates, Inserts oder Delete Statements oder doch alle zusammen? Danach geht's los und man erhält grafisch für jede einzelne Tabelle eine Auswertung. Die Übersicht zeigt, wie viele Update, Insert und Delete Kommandos notwendig wären. Sind die richtigen Datensätze ausgewählt, so kann man sich wiederum ein Skript generieren lassen. Für das Skript kann man alle Tabelle nehmen oder selektiv auswählen, welche relevant sind. Das Skript kann natürlich wieder vor der Ausführung bearbeitet werden. Der Übertrag einer komplexen Datenbank mit Daten war mit dem Tool ohne einen Fehler möglich! Somit konnten die Produktivdaten innerhalb kürzester Zeit in die Entwicklung übernommen werden.

Es wurde wieder viel Zeit durch das Tool gespart, aber auch die Qualität der Testumgebung mit realistischen Daten deutlich erhöht. Bei diesem Feature überzeugten wieder die Übersichtlichkeit und die Usability. So kann man immer genau sehen, wie viele Datensätze genau betroffen sind und ob es sich um ein Insert, Update oder Delete handelt.

Testdaten generieren

Hunderte Testdaten per Hand eintippen ist immer noch sehr verbreitet. Es bringt dem Entwickler extrem viel Spaß, äh Frust und meist auch keinen großen Nutzen. Der Datenbestand ist meist doch zu klein und es werden besonders wichtige Sonderfälle meist nur mit einem „funktionie-

renden“, konstruierten Datensatz abgebildet. Das generieren per Code, z.B. aus Visual Studio, ist meist zu aufwändig und es wird oft ohne ausreichende Testdaten gearbeitet.

Mit den DB Tools sollte sich das ändern – und wie es sich ändert. Es können sogar alle Beziehungen mit 1:n in Werten angegeben werden. So sind in wenigen Sekunden viele Testdaten sehr sauber mit FK Referenzen erzeugt. Man kann sogar die Anzahl der n Werte anpassen und damit die Anzahl der erzeugten Datensätze genau bestimmen. In der Übersicht werden alle Tabellen anschaulich mit genauen Zahlen oder Abhängigkeiten gezeigt. Wiederum braucht man nicht alle Tabellen in die Datengenerierung einzubeziehen und kann bestimmte Tabellen ohne Daten belassen. Bestehende Daten in der Datenbank können, müssen aber nicht gelöscht werden.

Beim ersten Test ist man aber etwas geschockt, aus welchem Datenmüll die Testdaten bei Strings bestehen. Für viele Tests reichen Sie daher nicht aus und man muss bei Strings möglichst eine reguläre Expression wählen. Dies ist bei kleineren Datenbanken ein gutes Vorgehen. Bei sehr großen Datenbanken mit vielen Spalten wünschte man sich hier aber die Standardeinstellung von Datenmüll auf die erstellte reguläre Expression umstellen zu können. So entsteht ein kleiner Aufwand, der aber mit dem Nutzen absolut zu rechtfertigen ist.

Doch was sieht man da... Man kann auch Zufallsdaten aus einer Datenbank als Werte nehmen! Somit kann man, ein wenig Aufwand vorausgesetzt, ordentliche Beispieldaten mit der Hilfe von anderen bestehenden Datenbanken erzeugen. Man kann z.B. eine Namensdatenbank für die Testdatengenerierung nutzen. Einfach super dieses Feature, weil nun auch wirklich realistische Zufallsdaten möglich sind.

Prüfung von Skripten

Beim Importieren von Skripten in die DB Tools wird deutlich, dass das Tool von den Visual Studio Entwicklern kommt. Es kann die Logik der Skripte prüfen und zeigt eventuelle Fehler im T-SQL sofort an.

Bei den zum Test importierten Datenbanken wurden jeweils 3-4 Fehler gefunden, die die Entwickler im SQL Studio übersehen haben. Zwar waren die Fehler nicht wichtig, weil es sich um abgelöste Funktionen handelt, die nicht mehr angesprochen werden, doch war es interessant zu sehen, dass diese Fehler gefunden wurden. Speziell fehlerhafte Referenzen werden absolut sauber gefunden.

Dieses Feature ist sehr hilfreich und kann gegebenenfalls Bugs anzeigen, die sonst in der Produktion später mal zum Tragen kommen könnten... So spart man zwar nicht auf den ersten Blick wenig Zeit, doch wenn man rechnet, dass man hier einen produktionsrelevanten Fehler früher findet, dann lohnt es sich wieder sehr deutlich. Daher auch hier wieder die Bewertung sehr gut für dieses Feature.

Was fehlt denn noch?

Stellt sich die Frage, ob bei 4 so tollen Features noch etwas fehlt bei den DB Tools? Ja, da gibt es noch etwas. Das Analysieren von Logdateien des SQL Server geht mit diesen Tools leider noch immer nicht. So gibt es auf dem Markt nur ein einziges Tool, was dies kann. Der Lumigent Log Explorer ist nicht von Microsoft und kann in Deutschland nur mit Umwegen bezogen werden (Distribution in der Schweiz). Daher wäre hier ein Markt für Microsoft, zumal man das Tool schon intern nutzt und mit den Transaktionen eine perfekt funktionierende Basis hat.

In einem Projekt wurde diese Funktion selbst programmiert und mit 70 Tagen war der Aufwand hier astronomisch hoch. Diese 70 Tage könnte man lieber in andere Features stecken und die Funktion aus einem seriösen Tool, wie dem DB Tool, nutzen. Hiermit würde sich der Nutzen von dem DB Tool dann auch nochmals Potenzieren. Mal sehen, ob es dieses von vielen heiß ersehnte Feature vielleicht in einer nächsten Version geben wird.

Tja, eine Idee gäbe es noch, man könnte den Namen ändern. Der Name der DB Tools ist mit dem Langnamen Microsoft Visual Studio 2005 Team Edition for Database Professionals doch ein wenig lang. Aber aus dem Englischen sind ja bereits lustige Namen wie z.B. Data Dude im Internet zu finden. Hat man dann einen ordentlichen Namen gefunden, so kann man auch seinem Chef erklären, warum das viele Geld für dieses Spitzenprodukt gut angelegt ist.

Preis

Man muss Kunde des MSDN Paketes sein. In der Team Suite oder der DB Pro Rolle ist dieses Tool enthalten. Es lässt sich nicht kaufen, sondern nur Mieten oder einfacher gesagt Abonnieren. Die Miete beträgt, gemessen an einem 3 Jahres Abonnement rund 2500€ pro Jahr. Eine Verlängerung ist dann etwas günstiger. Auch soll ein Upgrade von anderen MSDN Subscriptions verfügbar sein. Preise hierfür müssen jedoch am besten direkt bei Microsoft angefordert werden.

Viel Geld für ein tolles Tool, aber auch andere Datenbanken kosten auch viel Geld... So kostet ein DB2 ein Vielfaches von einem SQL Server. Dieses Geld kann man dann ja in solche Tools investieren, wo mir von der Konkurrenz nur ERWIN bekannt ist, was aber auch separat, sogar von einer anderen Firma, nämlich CA bezogen werden kann.

Fazit

Man bekommt ein tolles Tool zu einem für Microsoft Verhältnisse recht hohen Preis. Zwar ist der Weg mit der MSDN Lizenz als einzige Bezugsquelle etwas fragwürdig, passt aber in die Schiene einer Professionellen Entwicklungsumgebung. Nur mit der MSDN Lizenz kann man wirklich professionell und mit allen Hilfen Programmieren. Dies gilt aber nur für .net Entwickler! Was soll zum Beispiel ein Java Entwickler machen der sonst das Visual Studio nicht braucht? Ich würde sagen eine andere Datenbank...

Für .net und T-SQL Entwickler ist diese hochpreisige Tool aber sein Geld absolut wert. Wenn ein Java Entwickler den Nutzen des Tools sieht, so kann er es aber fast auch nicht mehr umgehen, wenn man den Gewinn an Zeit rechnet – vorausgesetzt er arbeitet mit dem Microsoft SQL Server 2000 oder 2005.

Vor allem ist das Produkt keine Spielerei, wie manche andere Produkte in diesem Umfeld, sondern eine Komponente, die hilft effektiv zu arbeiten und viel Zeit zu sparen. Sehr intuitiv zu bedienen und schon mit der Version 1 ein absolutes Highlight.