Dr. Nico Jacobs en Ir. Wim Coorevits, trainers en consultants bij U2U

Microsoft SQL Server 2005 Integration Services smaakt naar meer

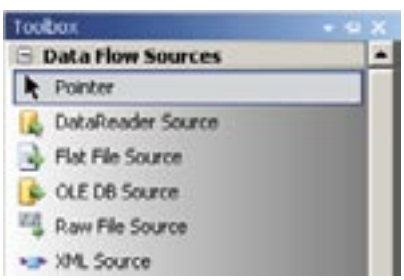
ETL-TOOL MET ARTIFICIAL INTELLIGENCE

Microsoft vaart al enige tijd onder de vlag 'innovatie door integratie'. De nieuwste editie van hun ETL-tool (Extract, Transform, Load) bewijst dit nog eens door de naam Integration Services te dragen. In dit artikel nemen we deze opvolger van SQL Server Data Transformation Services (DTS) onder de loupe.

De I in I(C)T blijft een belangrijk gegeven: hoe je het ook wendt of keert, het basisgegeven van de informatica blijft het genereren, transporteren en opslaan van gegevens of data. In heel wat bedrijven zijn gegevens in verschillende formaten opgeslagen. Vaak moeten deze gegevens omgevormd en gecombineerd worden om in een bepaalde toepassing van nut te zijn. Een typisch voorbeeld van dit laatste zijn analyse- en rapportagetoepassingen. Het is dan ook in deze context dat ETL-tools erg populair zijn om gegevens van allerlei bronnen op te vragen (extraheren), om te vormen (transformeren) en op te slaan (laden). Er zijn legio voorbeelden van het gebruik van dergelijke tools: het combineren van bedrijfseigen met bedrijfsvreemde databases, het op elkaar afstemmen van databasesna een fusie van twee bedrijven, of het overpompen van gegevens uit een database naar een andere database. Microsoft bundelt zijn ETL-tool samen met SQL Server. Voor SQL Server 2000 was dit Data Transformation Services, maar net zoals de andere onderdelen van dit pakket, is ook de ETL-tool danig gewijzigd in de 2005-editie. Zodanig zelfs dat Microsoft er een nieuwe naam aan gegeven heeft: Microsoft SQL Server 2005 Integration Services (SSIS). Integratie wordt mogelijk door data van allerlei gegevensbronnen te lezen en vervolgens om te vormen alvorens ze weer te gaan opslaan.

Databronnen

Integration Services blijft nog steeds trouw aan het uitgangspunt van Data Transformation Services: met dit product kunt u data van ongeacht welke gegevensbron die OLE DB ondersteunt als bron of eindbestemming van uw dataconversie gebruiken; SQL Server hoeft dus niet eens hierin betrokken te zijn. Als we kijken naar de lijst van mogelijke gegevensbronnen (zie afbeelding 1), dan zien we hierin al enkele nieuwigheden ten opzichte van Data Transformation Services. Zo heeft Microsoft ondersteuning voor XML-bestanden toegevoegd: uw gegevens kunnen uit een XML-bestand worden gelezen. Een ander belangrijke nieuwigheid, die vooral ontwikkelaars op prijs zullen stellen, is dat nu ook een DataReader als gegevensbron kan gebruikt worden.



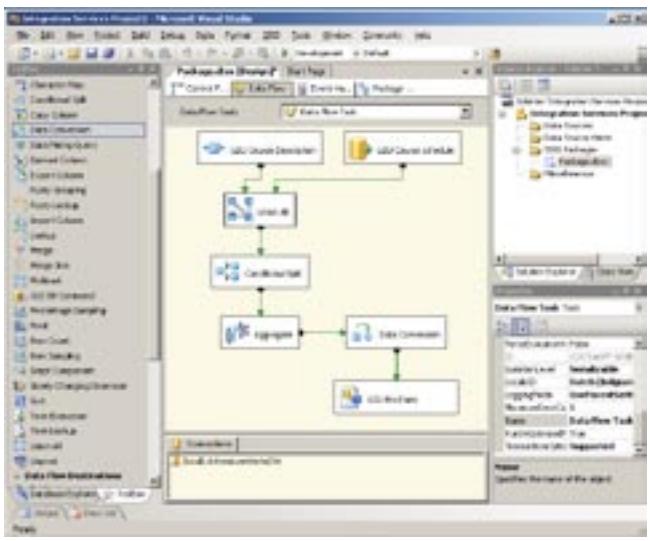
Afbeelding 1. Naast OLE DB en tekst aanvaardt Integration Services nu ook invoer uit XML en zelfs rechtstreeks uit een DataReader

Visual Studio Solution

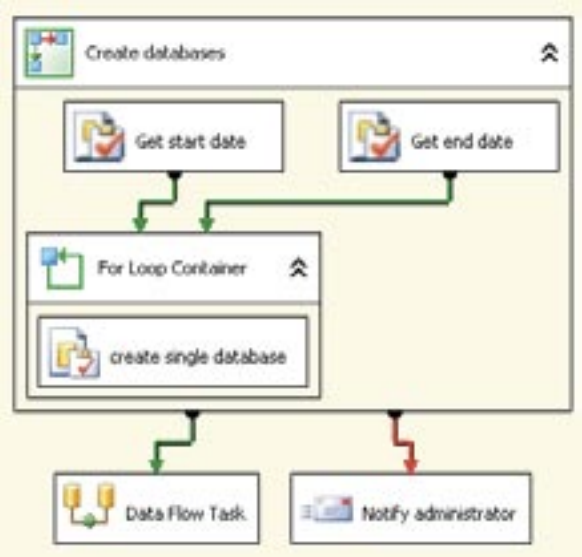
Wat ook nog steeds hetzelfde is als in Data Transformation Services, is dat er twee interactieve manieren zijn om een 'package' of pakket (een verzameling ETL-processen) te ontwerpen; ofwel via een Import/Export-wizard, ofwel via een 'designer'. Deze laatste is echter grondig gerestyled ten opzichte van Data Transformation Services. Om te beginnen is deze designer geïntegreerd in de Business Intelligence Studio, wat betekent dat in Visual Studio wordt ontworpen; zie afbeelding 2. Een ander verschil is dat een pakket als XML wordt opgeslagen. Dit betekent dat we onder andere gebruik kunnen maken van Microsoft Visual SourceSafe om pakketten onder versiecontrole te plaatsen. Dit houdt eveneens in dat we pakketten programmatisch kunnen genereren of wijzigen (met bijvoorbeeld XSLT) mocht dit gewenst zijn. Een ander belangrijke nieuwigheid is de scheiding tussen de controlflow en de dataflow: waar in Data Transformation Services deze beide elementen in een complex schema vervlochten zaten, heeft elk element nu een apart venster, wat complexe packages meteen overzichtelijker maakt.

Controlflow

Afbeelding 3 toont de grafische weergave van de controlflow van een pakket. De atomaire bouwstenen van zo'n flow zijn de taken.



Afbeelding 2. Een package maakt deel uit van een Solution en wordt in Visual Studio ontwikkeld



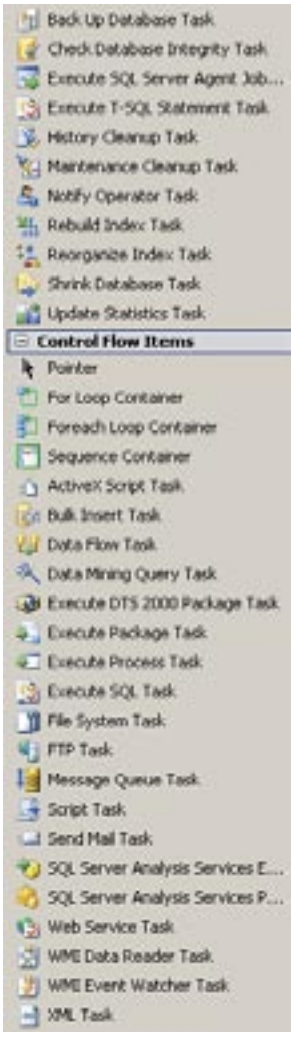
Afbeelding 3. Overzicht van controlflow

Voorbeelden van dergelijke taken zijn het versturen van e-mail, het uitvoeren van een consoleopdrachten of het opstarten van een ander pakket. De controlflow in afbeelding 3 start eerst twee processen op om de condities voor een lus te bepalen. Vervolgens wordt de lus opgestart die in elke iteratie een database aanmaakt. Als de lus uiteindelijk is afgelopen, wordt een Data Flow-taak gestart, tenzij ze faalt, want dan wordt er een e-mail verzonden. Integration Services bevat een aanzienlijk aantal taken dat in Data Transformation Services niet aanwezig was (zie afbeelding 4 voor een lijst met alle controlflow-taken). Zo is er nu een XML-taak die toelaat om XML-documenten te valideren, transformeren, samenvoegen, DiffGrams te berekenen, Xpath-queries uit te voeren en XML-documenten te 'patchen'. Een andere nieuwkomer is de webservice-taak, waarmee je een webservice kunt oproepen. De uitvoer van zo'n webserviceoproep kan zowel naar een bestand worden geschreven als in een variabele worden geladen om verder in het pakket gebruikt te worden. Twee andere nieuwe taken zijn de WMI Data Reader- en de WMI Event Watcher-taak, die de integratie tussen databasegeoriënteerde taken en besturingssysteemgeoriënteerde taken vergemakkelijken.

Containers

Opvallende nieuwigheden in deze controlflow zijn ook de containers. Met deze bouwblokken zijn complexe, geneste operaties eenvoudig op te bouwen. Er zijn drie soorten containers: Sequence-containers, For Loop-containers en ForEach Loop-containers. Een sequence-container is gewoon een verzameling van andere taken: het zorgt voor een scope waarbinnen deze andere taken uitvoeren; zie bijvoorbeeld de Create databases-container in afbeelding 3. Wanneer de container geactiveerd wordt, activeert deze op zijn beurt alle taken binnen in de container. Dit heeft een aantal voordelen. Zo kunnen bijvoorbeeld variabelen gedefinieerd worden die zo'n container als scope hebben. Een container kan ook in de toestand 'disabled' geplaatst worden, waardoor meteen alle taken binnen in deze container op non-actief geplaatst worden. Ook kan in de grafische voorstelling een container worden ingeklapt, waardoor een diagram overzichtelijker wordt.

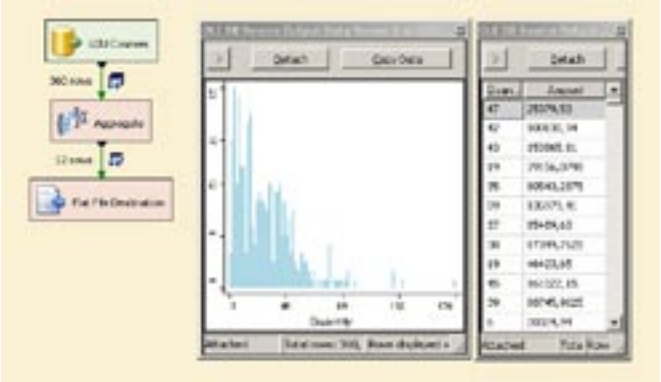
Beide iteratie-containers zijn het equivalent van de gelijknamige lussen in Visual Basic .NET. Bij een For Loop-container wordt een variabele aangepast totdat een bepaalde stopconditie is bereikt. Voor elke iteratie worden alle taken die zich in de container bevinden uitgevoerd, alsof ze een apart package zijn. Doordat SQL Server Integration Services variabelen heeft, net zoals in Data Transformation Services , kunnen deze gebruikt worden binnen in de loop om het gedrag van de taken daarin aan te passen. Een voorbeeld: stel dat we een aantal tabellen moeten aanmaken, eentje voor elk burgerlijk jaar, maar we kunnen het beginjaar en eindjaar niet hard coderen in ons pakket, dan worden deze gegevens uit de database zelf ingelezen. In zo'n scenario zouden we eerst een 'Execute SQL Task' gebruiken om uit de database het begin- en eindjaar in te lezen, en vervolgens een For Loop-container een variabele laten lopen van het beginjaar tot aan het eindjaar. In de container plaatsen we een Execute SQL-statement dat, gebruikmakend van de variabele, de juiste tabellen aanmaakt; zie afbeelding 3.
Een For Each Loop itereert over een verzameling, zoals alle bestanden in een folder. Dit kan handig zijn als we bijvoorbeeld alle logbestanden in een bepaalde directory in de database moeten laden, waarbij we niet op voorhand hoeven te weten hoeveel bestanden er precies zullen zijn.



Afbeelding 4. De beschikbare controlflow-taken

Dataflow

De eigenlijke datatransformatietaken worden op de dataflow-tab gedefinieerd. Waar we voor complexe opdrachten vaak aangewezen waren op het schrijven van een ActiveX-script of een complex SQL-statement, worden nu vele datatransformaties eenvoudig mogelijk door de vele bouwstenen die de designer ons aanreikt. Zo zijn er nu datatransformatietaken zoals aggregaties berekenen, sorteren, lookup doen enzovoort. Naast deze taken zijn er ook enkele 'intelligente' taken bijgekomen. Een voorbeeld van deze laatste is de Fuzzy Lookup. Deze gaat na of een record bij benadering voorkomt in een referentietabel, in tegenstelling tot de klassieke lookup, die enkel een exacte match toelaat. Zo kunt u op een automatische manier licht vervuilde data (typfouten, afkortingen, et cetera) opschonen. Tabel 1 toont het eindresultaat van zo'n Fuzzy Lookup, waarbij naast de kolom met ingelezen namen, een kolom is toegevoegd met de naam die volgens de Fuzzy Lookup de correcte referentienaam is. Naast deze naam vindt u ook statistieken die weergeven hoe groot de overeenkomst is tussen de referentienaam en de ingelezen naam. Hoewel de techniek die hierachter schuil gaat gelijkenissen vertoont met spellingscorrectietechnieken in tekstverwerkers (het opzoeken van woorden in een referentielijst) verschilt de aanpak bij de Fuzzy Lookup toch danig. Niet alleen kan (en moet) u bij de Fuzzy Lookup zelf uw referentielijst aanbieden, de aanpak is ook een taalneutrale techniek. Het systeem gaat er niet van uit dat de ingelezen termen tot een specifieke taal behoren.
Een andere taak die bij dergelijke taken kan helpen is de Fuzzy Grouping, waarbij ook zonder een referentietabel gelijkenissen tussen 'vervuilde' data kunnen worden opgespoord. Bij deze aanpak wordt een soort clusteringtechniek gebruikt, waardoor gekeken wordt hoe 'ver' elk woord van andere 'vervuilde' woorden is verwijderd. Wat ook handig is in de dataflow is dat elke taak waarbij er records kunnen zijn die falen, we deze records heel eenvoudig kunnen omleiden naar ieder andere component. Uit een component ontspringen immers twee pijlen: een groene, waarover de correct verwerkte data stromen, en een rode, waar-



Afbeelding 5. Dataflow met enkele dataviewers in actie

over de gefaalde data stromen. Zo kunnen we in het voorbeeld van de fuzzy lookup bijvoorbeeld eerst een gewone lookup doen, waarna we alleen de records voor welke de gewone lookup faalt doorsturen naar de (rekenintensievere) fuzzy lookup.

Gemakkelijker ontwerpen, testen, debuggen en deployen

Niet alleen heeft Microsoft aan de aangeboden functionaliteit gesleuteld, ook het gemak waarmee packages ontwikkeld en getest kunnen worden is sterk verbeterd. Zo wordt tijdens de uitvoering van een pakket binnen de designer grafisch weergegeven welke taken al opgestart zijn, welke succesvol afgerond zijn en welke gefaald hebben. Het is ook mogelijk om op elke connectie in een dataflow een dataviewer te plaatsen. Deze toont dan tijdens de uitvoering de data die over deze connectie stromen. Dit is op vele manieren mogelijk: ofwel de gedetailleerde data in een grid bekijken, of bijvoorbeeld een grafische weergave in de vorm van een histogram. Afbeelding 5 toont een dataflow in uitvoering: De eerste stap is beëindigd (groen), de twee volgende stappen zijn in uitvoering (geel). Rechts tonen twee dataviewers de gegevens die van de eerste naar de tweede taak stromen, zowel als histogram als in grid-formaat. Zo kunnen snel onregelmatigheden in de data worden opgespoord. Deze dataviewers fungeren op de dataflow tevens als breakpoint.

Een ander hulpmiddel bestaat uit breakpoints: zowel op elke taak als op elke lijn code die in een script van het pakket staan, kan een breakpoint worden gezet. Op die manier kan de uitvoering tijdelijk stopgezet worden, en kan de toestand van het pakket worden geïnspecteerd, vergelijkbaar zoals dit met gewone .NET-applicaties gebeurd. Tot slot is het ook mogelijk om logging en

ApproximateName	CorrectName	Similarity
Identity Confoozion Device	Identity Confusion Device	88%
Global Navigashunal System	Global Navigational System	91%
Multi-Purpose Rubber Ban'	Multi-Purpose Rubber Band	91%
Nonsplosive Cigar	Nonexplosive Cigar	92%
Tellykinesis Spoon	Telekinesis Spoon	92%
Cloakin' Device	Cloaking Device	92%
Th' Incredible Versatile Paperclip	The Incredible Versatile Paperclip	92%
Contack Lenses	Contact Lenses	93%
Fake Moestache Translato'	Fake Moustache Translator	93%
Effeckive Flashlight	Effective Flashlight	95%
Pocket Protecko' Rocket Pack	Pocket Protector Rocket Pack	95%
Unyversal Repair System	Universal Repair System	96%
Ultra Violet Attack Defenner	Ultra Violet Attack Defender	97%

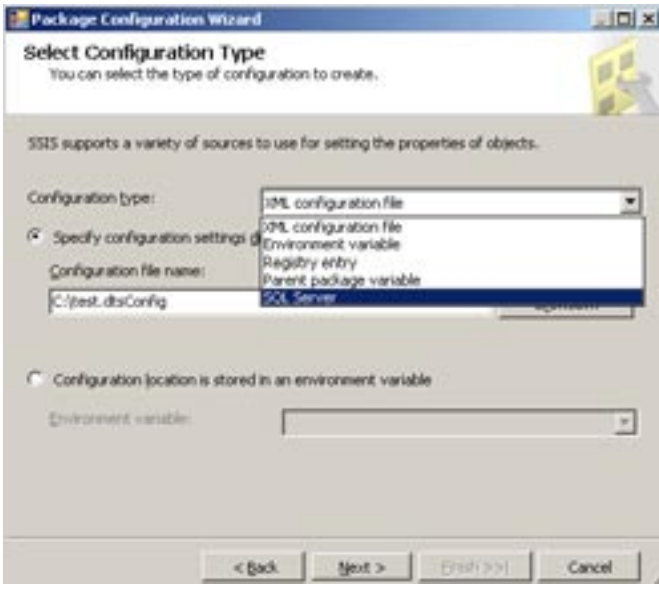
Tabel 1. Eindresultaat van een Fuzzy Lookup

Afbeelding 6. Event handlers kunnen op elke component geplaatst worden

event handling te doen: SQL Server Integration Services bevat een verzameling van events die zich kunnen voordoen, variërend van een 'Start Processing Task'-event tot een 'Could not open Connection-event'. Elk pakket kan een aantal loggers bevatten. Zo zijn er textfile-loggers, xml-loggers, database-loggers en windows event log-loggers. Elk van deze types loggers kan worden geconfigureerd om bepaalde events in een log weg te schrijven. Deze laatste manier van werken is vooral interessant om na afloop van de uitvoering te kunnen analyseren wat er precies is gebeurd. Willen we tijdens de uitvoering echter op een bepaald event reageren, dan kunnen we een event handler toevoegen. Zo'n handler is niet meer of minder dan een controlflow, wat betekent dat we in deze handler dus de beschikking hebben over alle taken (inclusief dataflow-taken) die Integration Services aan boord heeft. En zelfs op de event handlers kunnen event handlers gedefinieerd worden. In afbeelding 6 is te zien dat event handlers op elke component van een Integration Services-pakket geplaatst kunnen worden, én zelfs op andere event handlers.

Integration Services helpt ook bij de deployment van een project. Zo heeft men, naast de klassieke mogelijkheid om deployment vanuit Visual Studio rechtstreeks te doen, ook de mogelijkheid om een deployment-utility te bouwen bij het builden van een project. Dit kan je aanvinken bij de properties van het project; zie afbeelding 7. Er wordt dan automatisch een kleine applicatie aangemaakt die naar een andere server gekopieerd kan worden en daar de deployment doet. Een ander deployment-probleem zijn site-afhankelijke instellingen. Stel dat je een Integration Services-pakket wil laten draaien op meer servers, maar elke server heeft eigen eigenschappen, zoals connection-strings, time-outs en smtp-servers. We willen natuurlijk niet op elke server onze pakketten gaan bewerken. De oplossing die Integration Services hiervoor biedt is de Package Configuration. Hiermee kunnen we bij het aanmaken van een pakket aangeven dat bepaalde instellingen van het pakket niet hard-gecodeerd in het pakket zitten, maar uit een configuratiebestand komen. Dit biedt ons een waaier van mogelijkheden, zoals de configuratiegegevens opslaan in een XML-bestand, de registry, de SQL Server- database of Windows-omgevingsvariabelen. En zelfs de precieze locatie van deze configuratiegegevens kan op zijn beurt weer uit pakketvariabelen worden gelezen. Hierdoor is het bijvoorbeeld mogelijk op elke server een tabel te plaatsen die als string de locatie van het configuratiebestand bevat. Afbeelding 8 toont een eerste stap in de Package Configuratie Wizard. Met deze wizard kunnen we zowat elk aspect van een pakket configuratiebestanden uitlezen, en daardoor site-afhankelijk bepalen.

Afbeelding 7. Aanmaken van een deployment-utility aangeven in project Properties



Afbeelding 8. Eerste stap in de Configuratie Creatiewizard

Tot slot

SQL Server Integration Services kan met recht de grote broer van Data Transformation Services genoemd worden. Een mooie integratie in Visual Studio brengt veel ontwikkelingsgemak met zich mee, van een gemakkelijke Solution- en Projectgerichte aanpak tot integratie met Visual Studio als slagroom op de taart. Maar niet alleen de buitenkant is gewijzigd. Ook wat in de taart zit smaakt beter. Vele nieuwe taken maken dat de meeste opdrachten nu gemakkelijker en overzichtelijker ontwikkeld kunnen worden, zonder dat er scripts hoeven worden geschreven. Door artificiële intelligentie te integreren behoort automatische data-cleansing tot de mogelijkheden dankzij taken als Fuzzy Lookup en Fuzzy Grouping. Tot slot wordt het bakken van de taart ook gemakkelijker gemaakt dankzij allerlei debug- en deploy-mogelijkheden. Voor al dit lekkers, rep je je naar je MSDN subscription om daar de laatste release van SQL Server 2005 te downloaden. SQL Server Integration Services vormt een onderdeel van alle versies van SQL Server 2005, met uitzondering van SQL Server 2005 Express.

Dr. Nico Jacobs is trainer en consultant bij U2U (www.u2u.net), waar hij zich voornamelijk in Microsoft SQL Server en ADO.NET specialiseert. Zijn e-mailadres is nico@u2u.be.
Ir. Wim Coorevits is trainer en consultant bij U2U (www.u2u.net), waar hij zich voornamelijk in .NET en Microsoft SQL Server specialiseert.
Zijn e-mailadres is wim.coorevits@u2u.be.

Nuttige internetadressen

http://www.microsoft.com/sql/bi/integrate/productinfo/foresterreport.asp
http://www.sqlis.com/
http://www.datawarehouse.com/article/?articleid=5290
http://msdn.microsoft.com/SQL/sqlwarehouse/SSIS/default.aspx