



OPEN CURSOR

ABIS heeft tijdens de 'winterstop' niet stilgezeten.

We hebben een aantal technische projecten afgerond; nu kunnen we de conclusies verwerken - o.a. in Exploring DB2. Op basis van deze projecten zullen onze DB2-specialisten hun opgedane ervaringen met u, DB2-professional, delen.

In dit nummer presenteren we voor het eerst een tekst van een externe DB2-specialist. We hopen dat dit initiatief voor een aantal onder u een motivatie zal zijn, om ook een aantal eigen praktische ervaringen in een artikel te verwerken.

Dit voorjaar besteden we extra aandacht aan te weinig gekende, applicatie-specifieke aspecten van DB2: stored procedures, temporary tables, cursoren. U kent de basis, wij bieden u een relevante uitdieping!

Het ABIS DB2-team.

IN DIT NUMMER:

- De kenmerken en de evolutie van online reorganisatie in DB2, met praktijkvoorbeelden van Sidmar - *Online Reorg in DB2.*
- *Dossier 8* behandelt een aantal wijzigingen m.b.t. 'stage 1' query-optimalisatie, en distributiestatistieken.
- We bieden een tweede artikel in de reeks van 3 rond *.NET - DB2 Data Provider verenigt .NET-clients - 2.*
- *Cursusplanning mrt 2004 - mei 2004.*

CLOSE CURSOR

In het volgende nummer staan we stil bij een aantal fysieke karakteristieken van LOB-management. We lichten de integratie tussen DB2 en MQSeries toe, en we ronden af met een laatste artikel rond DB2 en .NET: waar zijn uw cursors gebleven?

Tot dan!

Online Reorg in DB2 *Davy Goethals (Sidmar)*

In dit artikel beschrijven we het mechanisme van de online reorg in DB2 voor OS/390 en z/OS en gaan we nader in op enkele nieuwigheden van V7, zoals de FASTSWITCH- en DRAIN-opties en het EXEC SQL-commando. Met deze features wordt het mogelijk om praktisch alle DB2-tablespaces en -indexen online te reorganiseren zonder, of met een minimale, onderbreking van de applicaties. We illustreren dit met enkele voorbeelden uit de praktijk van Sidmar.

Waarom reorganiseren?

Als een DB2-tabel regelmatig gewijzigd wordt (via SQL INSERT, UPDATE, DELETE of via de LOAD-utility) zullen de data zich in de tabel na verloop van tijd fysiek niet meer in een optimale positie bevinden met extra I/O en dalende antwoordtijden of doorlooptijden van applicaties als gevolg. Hetzelfde geldt voor de key-informatie in de indexen op die tabel.

Het REORG TABLESPACE-utility zal de rijen in de tabellen van een tablespace terug sorteren in een optimale volgorde zoals aangegeven via de clustering index. Tevens zullen de bijhorende indexen terug opgebouwd worden met voor elke key de nieuwe RID (record-id) waarden. De RID bevat de fysieke plaats van de rij(en) in de tablespace die behoren bij een bepaalde key.

Het REORG INDEX-utility zal de keys in een index terug sorteren in stijgende of dalende volgorde volgens de definitie van de index, maar zal geen RID-waarden aanpassen.

Regelmatig tablespaces en/of indexen reorganiseren is noodzakelijk: het is één van de factoren die optimale performantie garandeert voor de applicaties. Dit is echter niet voldoende - andere factoren zoals een goede design, voldoende indexen en accurate RUNSTATS-gegevens zijn even onontbeerlijk.

De klassieke REORG

Tot en met DB2 V4 bestond het reorganiseren van een tablespace, tablespace-partitie, index of index-partitie uit het ontladen van de data uit het object (UNLOAD-fase), het sorteren van de data in de goede volgorde (SORT-fase) en het terug opbouwen van het object (RELOAD-fase). In het geval van een reorganisatie van een tablespace met indexen werd dit nog gevolgd door een extra fase voor het herbouwen van de indexen (BUILD-fase). Opties zorgden ervoor dat bepaalde fases gelijktijdig worden uitgevoerd; in elk geval was het object steeds volledig onbeschikbaar voor applicaties tijdens de RELOAD- en BUILD-fases. Deze manier van reorganiseren bestaat vandaag nog steeds en wordt nu REORG SHRLEVEL NONE genoemd.

Technisch gesproken wordt in het begin van de uitvoering van het utility het object in status UTRO gezet en alle writers geDRAINed, zodanig dat het object tijdens de UNLOAD-fase niet meer kan wijzigen en enkel nog beschikbaar is voor leesoperaties. DRAINen betekent dat de actieve applicaties hun werk mogen verder zetten tot het einde van hun UOW maar dat er geen nieuwe applicaties of nieuwe UOWs meer toegelaten worden op het object. In het begin van de RELOAD-fase wordt het object in UTUT-status geplaatst, en ook de resterende readers worden geDRAINed, zodanig dat het object volledig vrij komt en kan herbouwd worden (het object is dus volledig onbeschikbaar voor SQL-applicaties).

Applicaties en permanente beschikbaarheid

Naarmate de hoeveelheid tabellen en applicaties stijgt, wordt de boven bedoelde onbeschikbaarheid voor de meeste installaties onaanvaardbaar. Sinds DB2 V5 werd dan ook door IBM een zogenaamde online reorg geïntroduceerd, met als doel de onbeschikbaarheid van de te reorganiseren objecten te reduceren en in bepaalde gevallen quasi volledig te elimineren. Dit als één van de vele elementen om 24x7 processing te kunnen bereiken.

Uitgangspunt bij een online reorg van een object is dat het object zelf fysisch niet langer herbouwd wordt, maar dat er een nieuw object aangemaakt wordt met de gereorganiseerde data (een zogenaamd shadow-object). Tijdens het aanmaken van het shadow-object blijft het oorspronkelijke object volledig beschikbaar. Pas wanneer alle samenhangende shadow-objecten herbouwd zijn, nemen ze de plaats in van de oorspronkelijke objecten tijdens de zogenaamde SWITCH-fase. Enkel tijdens deze SWITCH-fase zijn de data onbeschikbaar voor de applicaties (i.p.v. traditioneel gedurende de volledige RELOAD- en BUILD-fases).

Online REORG SHRLEVEL REFERENCE

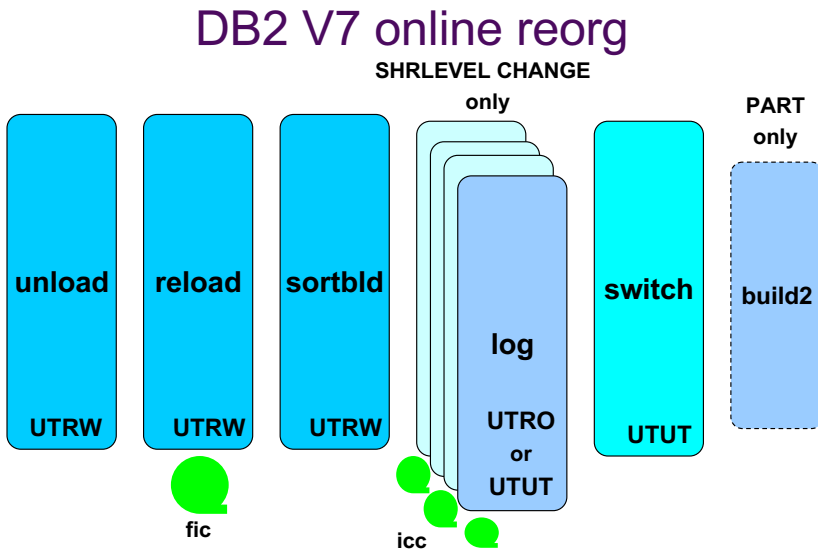
Met deze manier van reorganiseren blijft het object gedurende de ganse tijd beschikbaar voor leesoperaties. Enkel tijdens de SWITCH-fase zal een korte onbeschikbaarheid optreden tijdens het vervangen van basisobject (-en) door shadow-object (-en).

Meer technisch gesproken wordt bij de start van het utility in de UNLOAD-fase het object in UTRO-status geplaatst en alle writers geDRAINed, zodanig dat het object niet meer kan wijzigen en enkel nog beschikbaar is voor leesoperaties. De shadow-objecten worden volledig opgebouwd tijdens de RELOAD- en BUILD-fases en in het begin van de SWITCH-fase wordt het object in UTUT-status geplaatst. Alle resterende readers worden geDRAINed, zodanig dat de oorspronkelijke objecten volledig vrij komen en kunnen vervangen worden door hun respectievelijke shadows. Tijdens de SWITCH-fase is het object volledig onbeschikbaar voor SQL-applicaties.

Online REORG SHRLEVEL CHANGE

Dit is de tweede en meer complete vorm van online reorg. Inderdaad, het object blijft gedurende de ganse operatie beschikbaar voor zowel lees- als update-operaties. Enkel juist voor en tijdens de SWITCH-fase zal een korte onbeschikbaarheid optreden tijdens het vervangen van basisobject(-en) door shadow-object(-en) - zie ook figuur 1.

Figuur 1: Online REORG SHRLEVEL CHANGE



Meer technisch gesproken wordt bij de start het object in status UTRW geplaatst en worden de shadow-objecten opgebouwd; de basisobjecten mogen nog wijzigen. Na het volledig opbouwen van de shadow-objecten tijdens de RELOAD- en BUILD-fases worden de laatste wijzigingen die nog niet in de shadow-objecten aanwezig zijn, bepaald en eveneens toegepast op de shadow-objecten, voorafgaand aan de SWITCH-fase. Dit gebeurt in een nieuwe fase die men de LOG-fase noemt. Tijdens de LOG-fase worden alle wijzigingen sinds het begin van de reorganisatiejob bepaald door het afschannen van de DB2-log en het toepassen van de updates uit de log op de shadow-objecten. Omdat tijdens dit scannen ook nog nieuwe updates kunnen plaats vinden is de LOG-fase een iteratief proces waarbij in enkele stappen geprobeerd wordt het einde van de log in te halen om zo alle updates tot het laatste moment mee te hebben.

De laatste iteratieve stap van de LOG-fase wordt de LASTLOGAPPLY-fase genoemd. In het begin van de LASTLOGAPPLY-fase zal DB2 het object in UTRO-status plaatsen en alle writers DRAINen. Enkel leesoperaties worden nog toegelaten en er kunnen geen nieuwe updates meer bijkomen. Na het toepassen van alle resterende logrecords begint dan de SWITCH-fase: het object wordt in UTUT-status geplaatst, en de resterende readers geDRAINed. De oorspronkelijke objecten komen aldus volledig vrij, en kunnen vervangen worden door hun respectievelijke shadow-objecten. Tijdens de SWITCH-fase is het object volledig onbeschikbaar voor alle SQL-applicaties.

Tijdens elke LOG-iteratie wordt de full image copy (fic), die tijdens de RELOAD-fase werd gemaakt, aangevuld met een incrementele image copy (iic) om de inline copy te vervolledigen - zie ook figuur 1.

Sinds DB2 V7 bestaat er een nieuwe optie DRAIN ALL, welke toelaat om in het begin van de LASTLOGAPPLY het object in UTUT-status te zetten en zowel de readers als de writers onmiddellijk te DRAINen.

DB2 beslist de LASTLOGAPPLY-fase in te gaan wanneer de tijd nodig om de laatste logrecords toe te passen kleiner geschat wordt dan de waarde in seconden opgegeven via de MAXRO-optie. Om deze LOG-fase dus zo kort mogelijk te houden is het aan te raden om het utility uit te voeren tijdens een periode van lage activiteit. Het is ook mogelijk de LASTLOGAPPLY dynamisch te manipuleren via de DEFER-, LONGLOG- en DELAY-opties, waar we in dit artikel niet verder zullen op ingaan.

De mapping-tabel

De mapping-tabel wordt tijdens de RELOAD-fase van een REORG SHRLEVEL CHANGE opgebouwd, en bevat voor elke rij in de te reorganiseren tablespace de oorspronkelijke en nieuwe RID. Deze tabel zal tijdens de LOG-fase gebruikt worden om de vertaling te doen van de oorspronkelijke RID zoals die voorkomt in de DB2-logrecords naar de nieuwe RID in de shadow-objecten. Deze mapping-tabel is een DB2-tabel met bijhorende index die vooraf moet zijn aangeemaakt.

Door gebruik te maken van het nieuwe V7 EXEC SQL-commando kunnen de mapping-tabel en bijhorende fysieke objecten aanmaken in het begin van de utility step zelf, en verwijderen aan het eind. Bijkomende job steps voor en na de utility steps, om die dynamische DDL-statements uit te voeren, worden dan overbodig wat een vereenvoudiging van de JCL betekent.

Reorg van individuele partities - de BUILD2-fase

Een speciaal geval is de online reorg van één of meerdere partities van een partitioned tablespace. Om te vermijden dat een volledige non-partitioned index (NPI) zou onbeschikbaar worden tijdens de LASTLOGAPPLY- en/of SWITCH-fases wordt een NPI in dit geval nooit vervangen door zijn shadow-dataset maar wordt de NPI zelf aangepast na de SWITCH-fase in een aparte BUILD2-fase vanuit zijn shadow-dataset. Tijdens deze fase zijn de NPI's tijdelijk onbeschikbaar.

baar voor applicaties. Dit probleem wordt in DB2 V8 verder aangepakt door de invoering van zogenaamde data partitioned secondary indexen (DPSI 's) die ook fysisch gepartitioneerd zullen zijn.

Terminate van een online REORG

Een groot voordeel van een online reorg t.o.v. de klassieke reorg is dat de online reorg nooit het object zelf aanpast, maar enkel het shadow-object. In geval er tijdens de reorganisatie iets mis loopt volstaat TERMINATE van het utility om het basisobject terug vrij te geven en de shadow-objecten te schrappen. Er zijn nooit aanvullende recovery-acties nodig op het basisobject!

DB2 V7 fastswitch

In DB2 V5 en V6 werd tijdens de SWITCH-fase het basisobject genamed naar een tijdelijk object en het shadow-object genamed naar de oorspronkelijke naam. In geval van bv. een partitioned tablespace kon dit betekenen dat een ganse reeks VSAM-clusters moesten worden genamed. Daar elke rename van een VSAM-cluster tot 1 sec kan oplopen werd dit soms problematisch voor de doorlooptijd van de SWITCH-fase. Hetzelfde geldt voor een tablespace met enkele tientallen indexen. De nieuwe werkwijze in DB2 V7 is de FASTSWITCH waarbij tijdens de SWITCH-fase enkel de verwijzingen in de DB2-catalog en -directory worden aangepast om vanaf nu de shadow-datasets te gaan gebruiken als actieve datasets. In beide gevallen worden op het einde natuurlijk ook nog de oorspronkelijke objecten geschrapt.

Technisch gesproken kan de instance node van een DB2 VSAM-cluster nu I0001 of J0001 zijn, waarbij de actieve dataset instance-prefix 'I' of 'J' nu opgeslagen wordt in de DB2-catalog en -directory. Na een succesvolle online reorganisatie zal de instance van een object dus gewijzigd zijn. Dit is vooral van belang voor uitvoering van standalone utilities die in de JCL rechtstreeks naar de VSAM-naam verwijzen.

Fastswitch wordt geactiveerd door de FASTSWITCH-optie in de ZPARMS (default gedrag) of door de FASTSWITCH-optie in het REORG-commando.

Disk space

Daar de shadow-objecten en de mapping-tabel ook DB2-objecten zijn moet er voldoende disk space voorzien worden om deze objecten tijdelijk te kunnen bevatten. Voor grote tabellen met kleine beschikbaarheidseisen kan het dus nog altijd nuttig zijn om toch een klassieke reorganisatie te overwegen.

DB2 V7 nieuwe drain opties

Een veel voorkomend probleem met online reorg is dat het DRAINEN van alle applicaties tijdens de LOG- en/of SWITCH-fases niet lukt, omdat er nog enkele applicaties locks vasthouden op het object. In dit geval krijgt het reorg utility een timeout waarna het in abend ging en daarbij soms het object in UTRO- of UTUT-status achterliet. Het

was dan aan de DBA om zo snel mogelijk ofwel het utility te herstarten (niet mogelijk in alle fases van online reorg bv. niet in de LOG-fase), ofwel te beëindigen via het TERMINATE-commando welke de shadow-datasets schrapt en het oorspronkelijke object terug volledig beschikbaar maakt in RW-status. De tijd dat het utility wacht op de DRAIN vooraleer in timeout te gaan was IRLMRWT x UTIMOUT met IRLMRWT de timeoutwaarde voor SQL-applicaties en UTIMOUT de vermenigvuldigingsfactor voor utilities.

Een eerste verbetering werd geïntroduceerd met de TIMEOUT TERM-optie die ervoor zorgt dat na een timeout tijdens het DRAINen het utility onmiddellijk en automatisch beëindigd wordt (in tegenstelling met de default TIMEOUT ABEND waar het utility in abend gaat en manuele restart of terminate nodig zijn). Goed nieuws voor de applicaties door het onmiddellijk opheffen van een eventuele restrictieve status, maar slecht nieuws voor de DBA die later de reorganisatie moest herbeginnen.

Bijkomend gebeurde het ook vaak dat nieuwe binnenkomende applicaties in timeout gingen door een uitstaande DRAIN die langer duurt dan het IRLMRWT timeout interval.

Vanaf V7 is het mogelijk om het DRAINen beter te controleren via een aantal nieuwe drain opties:

- DRAIN_WAIT: het interval in seconden waarna het DRAINen stopt als in die tijd de DRAIN niet gelukt is. In combinatie met de RETRY-optie zal een gewone nieuwe LOG-iteratie gestart worden.
- RETRY: hiermee specificeert men hoeveel maal men de DRAIN zal laten herproberen vooraleer toch een terminate aan te vragen.
- RETRY_DELAY: de tijd tussen twee DRAIN-pogingen in.

Door DRAIN_WAIT kleiner te kiezen dan de IRLMRWT time-out voor SQL-applicaties en door een voldoende aantal keren te herproberen met voldoende tussentijd kan men er voor zorgen dat de online reorganisatie vroeg of laat toch succesvol zal beëindigen, en dit zonder timeouts van het utility of van de applicaties. Het effect van de DRAIN op de antwoordtijden blijft natuurlijk wel bestaan. Hoe beter de commit-frequentie van de applicaties, hoe minder RETRY's er nodig zijn, en hoe sneller de REORG afgerond zal zijn!

Design aspecten

Tijdens de design van een applicatie blijft het dus belangrijk om aandacht te schenken aan de commit-frequentie van de applicatie, vermits de DRAIN maar lukt als alle actieve UOWs afgelopen zijn. Een actieve UOW duurt normaal tot de volgende commit-instructie of einde van de thread als er geen commits gebeuren. Uitzondering is een open cursor-with-hold, die locks houdt over commit points heen tot aan de commit volgende op de close van de cursor. Vergeet ook geen commits te nemen in applicaties die enkel lezen met uncommitted read: ook deze houden de DRAIN tegen.

Opsporen van slechte commit-frequenties

Wanneer een DRAIN mislukt wegens een actieve UOW zal DB2 de message DSNU1122I uiten (vb: DSNU1122I JOB jobname PERFORMING REORG WITH UTILID utilid UNABLE TO DRAIN database.tablespace RETRY 1 OF 120 WILL BE ATTEMPTED IN 120 SECONDS).

De applicatie die de DRAIN verhindert kan dan gemakkelijk opgespoord worden via het commando “-DISPLAY DATABASE(dbname) SPACENAM(*) CLAIMERS LIMIT(*)”.

Zie [Voorbeelden online reorg](#) op p. 13.

Een online reorg plannen

Zoals eerder gezegd is het aangeraden om een online reorg te plannen tijdens een periode van lage activiteit. Dit om de LOG-fase kort te houden door het aantal te scannen en toe te passen logrecords te beperken, en om het effect van de DRAIN op antwoordtijden te minimaliseren. Maar door gebruik te maken van de nieuwe V7 DRAIN-opties is het mogelijk om een online reorg ook uit te voeren tijdens een piekperiode zonder timeouts te veroorzaken. De lengte van de LOG-fase, met eventueel meerdere pogingen om te DRAINen is dan echter moeilijk te voorspellen.

Conclusies

In dit artikel hebben we pogen aan te tonen hoe het met de nieuwe features van de DB2 V7 REORG utility mogelijk is praktisch alle DB2-tablespaces en indexen online te reorganiseren zonder noemenswaardige operationele problemen. Noodzakelijke voorwaarden zijn wel dat er voldoende DB2-diskspace voorhanden is en dat de commit-frequentie van de applicaties voldoende hoog is (ook voor uncommitted read-applicaties).

Binnen Sidmar worden aldus wekelijks, tijdens een dalperiode, alle tablespaces en indexen van onze productie systemen, die in aanmerking komen voor reorganisatie, automatisch bepaald en gereorganiseerd, en dit zonder één enkele applicatie uit dienst te nemen.

Enige uitzonderingen zijn de DB2 catalog- en directory-objecten en LOB-tablespaces welke in DB2 V7 nog steeds dienen gereorganiseerd te worden tijdens een stilstand van de applicaties met een klassieke reorganisatie. Online reorg van één enkele of meerdere partities van een gepartitioneerde tablespace is ook enkel toegelaten tijdens een stilstand wegens mogelijke onbeschikbaarheid van de niet gepartitioneerde indexen.

D. Goethals is systeemingenieur bij SIDMAR, waar hij zich voornamelijk bezig houdt met de software infrastructuur op de centrale mainframe computer. Zijn belangrijkste aandachtspunten zijn het DB2, IMS en MQseries. Davy kan worden gecontacteerd op davy.goethals@sidmar.arcelor.com.

DOSSIER 8

Over stage 1 en distributiestatistieken

Elke nieuwe release van DB2 bevat een aantal wijzigingen - verbeteringen - op het niveau van de optimizer. Concreet gaat het dan meestal om een aantal nieuwe elementen waarmee de optimizer kan rekening houden bij het bepalen van de meest efficiënte accesspaden. Een tweetal belangrijke vernieuwingen worden in wat volgt even aangehaald.

Voorreest is er natuurlijk een uitbreiding van de lijst met stage 1 - indexeerbare predicaten.

De weg die IBM in DB2 versie 7 heeft gekozen wordt consequent verder uitgewerkt. Inderdaad, sinds versie 7 zijn where-condities van het type 'col1 = col2 expressie', 'col1 operator col2 expressie', en 'col1 between col2 expressie 1 and col3 expressie 2' reeds stage 1 - indexeerbaar als aan een aantal voorwaarden is voldaan. Concreet: in alle boven aangehaalde voorbeelden moet col1 een kolom zijn van de inner join tabel, moet de definitie van de data types in de bewerking identiek of compatibel zijn, en moet de lengte van col2 steeds minstens gelijk zijn aan de lengte van col1. Indien aan deze voorwaarden niet is voldaan blijven de predicaten stage 2, en dus niet indexeerbaar.

In versie 8 is de eis wat betreft compatibiliteit van de datatypes, met inbegrip van de lengte van de kolommen, zo goed als verdwenen (een aantal beperkingen blijven bestaan, en hebben te maken met het door mekaar gebruiken van verschillende CCSIDs, bepaalde float/decimal/float-conversies, etc.). Deze versoepeling komt voornamelijk queries ten goede, die een join uitvoeren op kolommen met niet-compatibele datatypes. Ook applicaties ontwikkeld in programmeertalen die DB2-specifieke datatypes niet ondersteunen, zullen hier mogelijk een aanzienlijk performantievoordeel bij halen - denk maar aan C, die geen decimal type kent, of Java, waarin een vaste-lengte-karaktertype ontbreekt.

Daarnaast moet ook aandacht worden besteed aan de mogelijkheid tot het genereren van distributiestatistieken, voor niet-geïndexeerde kolommen, dan wel voor de non-leading kolommen in een multi-kolom-index. Sinds DB2 versie 5 was hiervoor een externe applicatie DSTATS beschikbaar - een speciale catalogotabel SYSCOLDIST werd hiervoor aangemaakt en aangewend. Sinds DB2 versie 8 is het gebruik van DSTATS niet meer vereist - de functionaliteit van het RUNSTATS utility werd namelijk overeenkomend uitgebreid. Merk echter op dat inline gegenereerde statistieken nog steeds geen distributiestatistieken genereren.

RUNSTATS in DB2 versie 8 genereert: frequentiedistributie voor kolommen en kolomgroepen, cardinaliteit voor kolomgroepen, en de frequentie waarmee een opgegeven aantal waarden in een kolom of kolomgroep aanwezig is.

Aan de hand van deze uitbreidingen is het mogelijk voor de DB2 optimizer een betere inschatting te maken van de effectiviteit van alternatieve uitvoerpaden - de evaluatie van de join-volgorde, de keuze voor synchrone of asynchrone I/O, of de inschatting van de bruikbaarheid van de alternatieve predikaten wordt hierdoor vergemakkelijkt, en correcter.

Kris Van Thillo (ABIS)

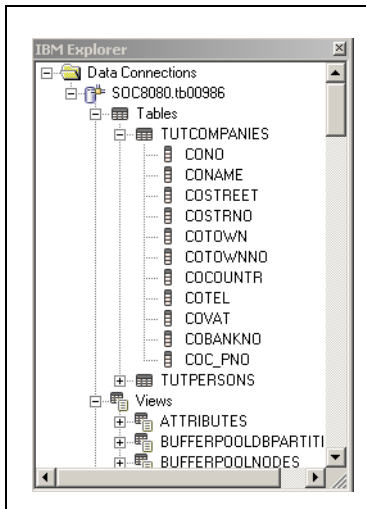
DB2 Data Provider verenigt .NET-clients - 2

Katrien Platteborze (ABIS)

Inleiding

In een vorig artikel werd de DB2 Data Provider voor .NET-clients in-geleid. Het IBM DB2 .NET enablement-project heeft twee aspecten, enerzijds de dataprovider, deze voorziet in een aantal klassen die connecties met een DB2-datasource mogelijk maken, die toelaten een commando's uit te voeren en data op te halen. Bij het bouwen van applicaties zal men deze klassen gebruiken in combinatie met een aantal ADO.NET-klassen die onafhankelijk zijn van de data source - bv. het DATASET-object. Een cruciale vraag bij ADO.NET-applicaties is "Waar is mijn cursor naartoe?" of beter nog, wanneer men denkt aan cursor in de strikte betekenis van het woord (een server cursor dus), "Bestaan er nog cursors?". Op deze problematiek gaan we in een volgend nummer in. Anderzijds heeft IBM een aantal add-ins voor Visual Studio - dé ontwikkel omgeving voor .NET - ontwikkeld. In dit artikel bekijken we deze add-ins en de systeemvereisten.

IBM Explorer



Parallel aan de Visual Studio Server Explorer die gebruikt wordt voor het beheer van data-connecties, ontwikkelde IBM de IBM Explorer met hetzelfde doel. Wanneer men met een DB2 Data Provider een DB2 data source wil benaderen zal men uitsluitend met deze laatste werken omdat de algemene Server Explorer alleen ODBC- of OLE DB-connecties aankan. Voor een welbepaalde connectie naar een database ziet men de tabellen - weliswaar zonder owner -, views, stored procedures en functies. Het owner-probleem kan men oplossen door met verschillende filters te werken voor verschillende connecties naar dezelfde database.

Er is echter een spijtige voorwaarde: elke connectie moet met een andere user en paswoord gelegd worden. De connectieinformatie wordt gecached op de client en kan elk ogenblik vernieuwd worden. Vanuit de IBM Explorer kan men direct aan het werk, zowel richting DB2 als richting .NET.

DB2 georiënteerd:

- ophalen van data;
- droppen van objecten;
- source code van procedures en UDF's bekijken;
- ophalen van catalog-informatie zoals bv. datatype in het Visual Studio properties scherm;
- testruns van procedures en functies gevolgd door rollback.

Screenshots? - zie: [DB2-addins in Visual Studio](#) op p. 16

.NET georiënteerd:

- genereren van ADO.NET-code.

Door het slepen van een tabel of view op een Windows form wordt er automatisch een DB2 connection-object en een DB2 data adapter-object aangemaakt. De data adapter wordt gegenereerd met DML-commando's naar keuze (hiertoe dient men de data adapter configuratie-wizard te doorlopen). De code wordt automatisch aangepast. Eens het zover is, is de knop 'Generate Dataset' dicht bij de hand en is men nog maar een paar lijnen code verwijderd van een werkende applicatie die data ophaalt, toelaat ze te bewerken en terugstuurt naar de database. Ook voor stored procedures en functies kan men via drag en drop code genereren.

DB2 Database Project

Het DB2 Database Project laat toe server-objecten met behulp van scripts aan te maken, gebruik makende van een SQL-editor. Zowel DB2 DDL als DML kan uitgevoerd worden. Output-berichten worden getoond in een output-scherm waar de DB2 messages afgezonderd van andere opgevraagd kunnen worden. De DB2 database-projecten worden toegevoegd aan de Solution Explorer. Andere add-ins zijn:

- DB2 help;
- Echte DB2 development and administration tools zoals Development Center, Command Center,...;
- Aangepaste toolbox met DB2 klassen.

Installatie

Voor de ontwikkeling gebruiken wij op de client een Visual Studio .NET 2002-installatie en een Application Development V8.1 met service pack 2 DB2-client. Deze versie is de eerste versie waarvoor er .NET-ondersteuning is. Wanneer men met Visual Studio 2003 werkt heeft men service pack 3 of 4 nodig. De database bevindt zich op een DB2 UDB voor LUW V8.1. Mits de installatie van DB2 connect software is DB2 voor OS/390 of z/OS V7 ook mogelijk. Wanneer men eerst Visual Studio en dan de DB2-client installeert, is het niet nodig de DB2 add-ins te registreren in Visual Studio. In het omgekeerde geval kan dit zeer eenvoudig via de DB2-menu's die men vanaf de 'Start' balk kan openen of via db2vsreg.exe. Aan db2vsreg.exe kan men opties toevoegen om alleen de add-ins of alleen de documentatie te vragen of ze te verwijderen.

CURSUSPLANNING MRT - APR - MEI 2004

DB2 concepten	375 EUR	08/03(L)
DB2 for OS/390, een totaaloverzicht	1625 EUR	01-05/03 (L), 29/03-02/04 (W), 24-28/05 (w)
DB2 UDB, een totaaloverzicht	1625 EUR	01-05/03 (L), 24-25/05&01-03/06 (w)
RDBMS concepten	325 EUR	01/03 (L), 29/03 (W), 24/05 (W)
Basiskennis SQL	325 EUR	02/03 (L), 30/03 (W), 25/05(W)
DB2 for OS/390 basiscursus	975 EUR	03-05/03 (L), 31/03-02/04 (W)
DB2 UDB basiscursus	975 EUR	03-05/03 (L)
SQL workshop	700 EUR	11-12/03 (L), 15-16/04 (W)
DB2 for OS/390 programmering voor gevorderden	700 EUR	08-09/03 (L), 17-18/05 (W)
Gebruik van DB2 procedural extensions	350 EUR	10/03 (L), 19/05(W)
DB2 for OS/390: SQL performance	1200 EUR	24-26/03 (L)
DB2 UDB applicatieperformance	400 EUR	08/06 (W)
Database applicatieprogrammering met Java	800 EUR	29-30/04 (L)
Fysiek ontwerp van relationele databases.	700 EUR	03-04/05 (L)
DB2 for OS/390 database administratie	1600 EUR	15-18/03 (W), 24-27/05(L)
DB2 for OS/390 operations and recovery	1500 EUR	21-23/04(L)
DB2 UDB systeembeheer en performance	400 EUR	30/04 (L)
DB2 UDB en zijn extenders: XML en text search	200 EUR	12/03 (L)
DB2 UDB integratie met MQSeries	200 EUR	12/03 (L)

Plaats: L = Leuven; W = Woerden; details en extra cursussen: www.abis.be

Postbus 220
Diestsevest 32
BE-3000 Leuven
Tel. 016/245610
Fax 016/245691
training@abis.be



Postbus 122
Pelmolenlaan 1-K
NL-3440 AC Woerden
Tel. 0348-435570
Fax 0348-432493
training@abis.be

Bijlagen

Voorbeelden online reorg

We starten met een JCL-voorbeeld van een online REORG SHRLEVEL CHANGE van een tablespace.

Voorbeeld 1 - online REORG SHRLEVEL CHANGE (tablespace)

```
/**
/** ONLINE REORG SHRLEVEL CHANGE VAN DSQDBCTL.DSQTSC1
/**
//STEP01 EXEC PGM=DSNUTILB,PARM='DB2T,SIDDAGOY'
// INCLUDE MEMBER=DB2TSLIB
//SYSTEMPL DD DSN=SIDM.SOURCE.PARMLIB(Z1REORG),DISP=SHR
//UTPRINT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
EXEC SQL CREATE DATABASE SIDDAGOY ENDEXEC
EXEC SQL CREATE TABLESPACE SIDDAGOY IN SIDDAGOY
        USING STOGROUP SYSDEFLT PRIQTY 1 SECQTY 1 ERASE NO
        BUFFERPOOL BP1 FREEPAGE 0 PCTFREE 5
        LOCKSIZE TABLE SEGSIZE 64 ENDEXEC
EXEC SQL CREATE TABLE MAPTB.SIDDAGOY
        (TYPE CHAR(1) NOT NULL, SOURCE_RID CHAR(5) NOT NULL,
        TARGET_XRID CHAR(9) NOT NULL, LRSN CHAR(6) NOT NULL)
        IN SIDDAGOY.SIDDAGOY ENDEXEC
EXEC SQL CREATE UNIQUE INDEX MAPIX.SIDDAGOY
        ON MAPTB.SIDDAGOY
        (SOURCE_RID ASC, TYPE ASC, TARGET_XRID ASC, LRSN ASC)
        USING STOGROUP SYSDEFLT PRIQTY 73728 SECQTY 18432 ERASE NO
        BUFFERPOOL BP2 FREEPAGE 0 PCTFREE 10 ENDEXEC
REORG TABLESPACE DSQDBCTL.DSQTSC1
        COPYDDN(TSYSOCP1)
        SHRLEVEL CHANGE MAPPINGTABLE MAPTB.SIDDAGOY
        MAXRO 20 DRAIN ALL
        DRAIN_WAIT 20 RETRY 120 RETRY_DELAY 60 TIMEOUT TERM
        SORTDEVT 3390 SORTNUM 6
        WORKDDN(TSYSUT1,TSORTOUT) UNLDDN(TSYSREC)
        DISCARDN(TSYSDISC) PUNCHDDN(TSYSPUNC)
        STATISTICS TABLE ALL
        INDEX(ALL KEYCARD FREQVAL NUMCOLS 1 COUNT 10
                FREQVAL NUMCOLS 2 COUNT 10
                FREQVAL NUMCOLS 3 COUNT 10
                FREQVAL NUMCOLS 4 COUNT 10
                FREQVAL NUMCOLS 5 COUNT 10)
EXEC SQL DROP DATABASE SIDDAGOY ENDEXEC
/**
```

In dit voorbeeld wordt de mapping-tabel aangemaakt en gedropt in het utility step zelf door gebruik te maken van de nieuwe EXEC SQL-syntax. Om conflicten te vermijden tussen reorganisatiejobs die in parallel lopen wordt deze aangemaakt in een nieuwe database die dezelfde naam heeft als de jobnaam. In onze installatie kunnen geen 2 jobs met dezelfde naam gelijktijdig gestart worden. Om dezelfde reden bevatten de mapping-tabel en de index erop deze jobnaam om een unieke naam te bekomen. De mapping-tabel wordt gealloceerd

met een kleine spaceallocatie omdat in de praktijk enkel de mapping index intern zal gebruikt worden. Bij succesvol uitvoeren van de job wordt de mapping-tabel terug gedropt.

De reorg utility maakt voor alle werk- en outputdatasets zoals COPYDDN, WORKDDN, etc. gebruik van DB2-templates om deze datasets dynamisch te alloceren. Dit maakt de uitvoerings-JCL heel eenvoudig met nog slechts 5 DD-kaarten: STEPLIB, SYSTEMPL, SYSPRINT, UTPRINT en SYSIN .

De FASTSWITCH-optie is default actief in de ZPARMS en moet hier niet meer expliciet gespecificeerd worden.

Om timeouts te vermijden van zowel het utility zelf als applicaties die de tabel wensen te benaderen wordt gebruik gemaakt van de volgende DRAIN-opties:

- MAXRO 20: begin met de laatste iteratie van de LOG-fase als die minder dan 20 sec geschat wordt.
- DRAIN ALL: DRAIN alle applicaties (zowel readers als writers in het begin van de laatste iteratie.
- DRAIN_WAIT 20: stop het DRAINen als dit niet lukt binnen de 20 sec en start een nieuwe gewone LOG-fase Deze waarde is kleiner dan de timeoutwaarde voor SQL-applicaties IRLMRWT die 25 sec bedraagt.
- RETRY 120 RETRY_DELAY 60: probeer de DRAIN 120 maal te herhalen met 60 sec tussenpause.
- TIMEOUT TERM: doe onmiddellijk TERMINATE van het utility als dit na 120 maal (na meer dan 2 uur) nog niet gelukt is (TIMEOUT ABEND in combinatie met DRAIN ALL is zinloos vermits het utility toch niet herstartbaar is in de LOG fase en er toch altijd TERMINATE nodig is) .

Tenslotte wordt tijdens de REORG-fase ook RUNSTATS-statistieken verzameld (STATISTICS keyword) en een INLINE COPY aangemaakt (altijd verplicht bij online REORG SHRLEVEL CHANGE).

Voorbeeld 2 - online REORG SHRLEVEL REFERENCE (tablespace)

```
REORG      TABLESPACE DSQDBCTL.DSQTSC1
           SORTDATA  SORTKEYS NOSYSREC
           COPYDDN(  TSYSOCP1 )
           SHRLEVEL  REFERENCE
           DRAIN_WAIT 20  RETRY 120  RETRY_DELAY 60  TIMEOUT TERM
           SORTDEVT 3390 SORTNUM 6
           WORKDDN(  TSYSUT1,TSORTOUT ) UNLDDN(TSYSREC)
           DISCARDN(  TSYSDISC ) PUNCHDDN(TSYSPUNC)
           STATISTICS TABLE ALL
           INDEX(ALL  KEYCARD  FREQVAL NUMCOLS 1 COUNT 10
                  FREQVAL NUMCOLS 2 COUNT 10
                  FREQVAL NUMCOLS 3 COUNT 10
                  FREQVAL NUMCOLS 4 COUNT 10
                  FREQVAL NUMCOLS 5 COUNT 10)
```

Een REORG SHRLEVEL REFERENCE wordt uitgevoerd met dezelfde eenvoudige JCL door ook gebruik te maken van DB2-templates. Aanmaak van een mapping-tabel en bijhorende objecten is hier niet nodig. De performantie-opties SORTDATA SORTKEYS en NOSYSREC worden expliciet toegevoegd daar ze niet default zijn voor REORG SHRLEVEL REFERENCE. De FASTSWITCH-mode is geactiveerd in de ZPARMS. De DRAIN-opties worden identiek gekozen als bij REORG SHRLEVEL CHANGE maar slaan hier op het begin van de UNLOAD-fase (naar UTRO) en de SWITCH-fase (naar UTUT). Ook hier worden INLINE STATISTICS en een INLINE COPY in parallel aangemaakt.

Met dezelfde JCL wordt ook een online REORG SHRLEVEL CHANGE van een index gestart. De SYSIN utility input ziet er dan als volgt uit.

Voorbeeld 3 - online REORG SHRLEVEL CHANGE (index)

```

REORG      INDEX Q.PROFILEX
           SHRLEVEL CHANGE  MAXRO 20    DRAIN ALL
           DRAIN_WAIT 20  RETRY 120    RETRY_DELAY 60    TIMEOUT TERM
           SORTDEVT 3390  SORTNUM 6
           WORKDDN(TSYSUT1)
           STATISTICS KEYCARD  FREQVAL NUMCOLS 1  COUNT 10
                                FREQVAL NUMCOLS 2  COUNT 10
                                FREQVAL NUMCOLS 3  COUNT 10
                                FREQVAL NUMCOLS 4  COUNT 10
                                FREQVAL NUMCOLS 5  COUNT 10

```

De DRAIN-opties zijn identiek als bij een reorg tablespace.

Voor de volledigheid geven we ook nog een SYSIN-voorbeeld van een REORG SHRLEVEL REFERENCE van een index.

Voorbeeld 4 - online REORG SHRLEVEL REFERENCE (index)

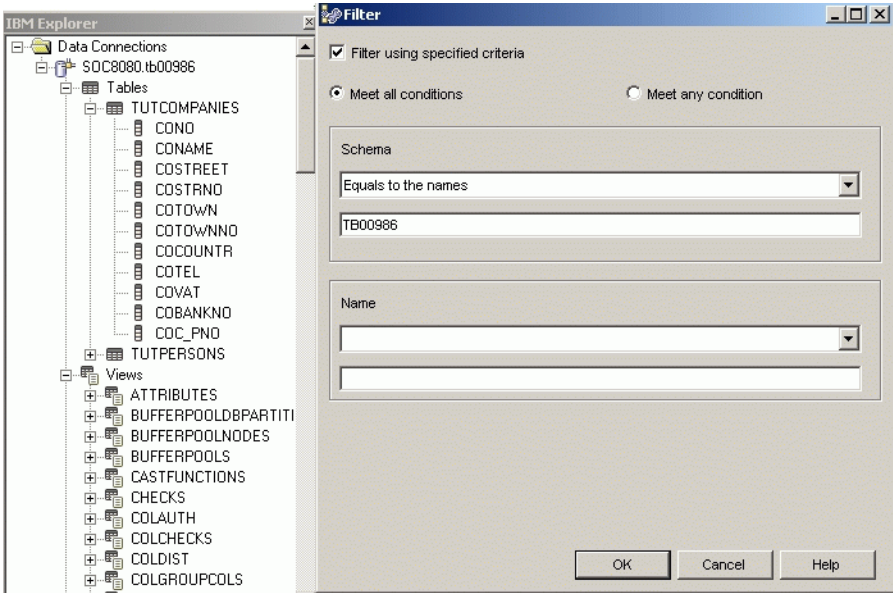
```

REORG      INDEX Q.PROFILEX
           SHRLEVEL REFERENCE
           DRAIN_WAIT 20  RETRY 120    RETRY_DELAY 60    TIMEOUT TERM
           SORTDEVT 3390  SORTNUM 6
           WORKDDN(TSYSUT1)
           STATISTICS KEYCARD  FREQVAL NUMCOLS 1  COUNT 10
                                FREQVAL NUMCOLS 2  COUNT 10
                                FREQVAL NUMCOLS 3  COUNT 10
                                FREQVAL NUMCOLS 4  COUNT 10

```

DB2-addins in Visual Studio

Figuur 1: Filter die gebruikt wordt om aan te geven welke tabellen bij deze connectie bruikbaar zijn

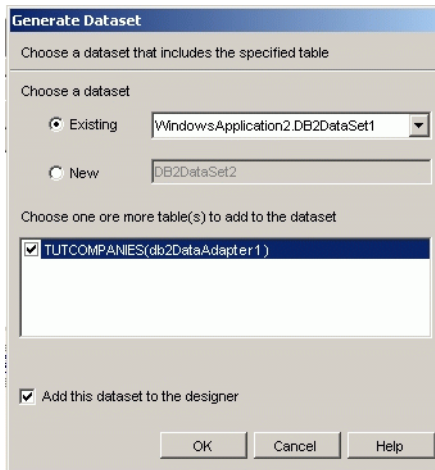


Figuur 2: Properties en data van de tutcompnies tabel

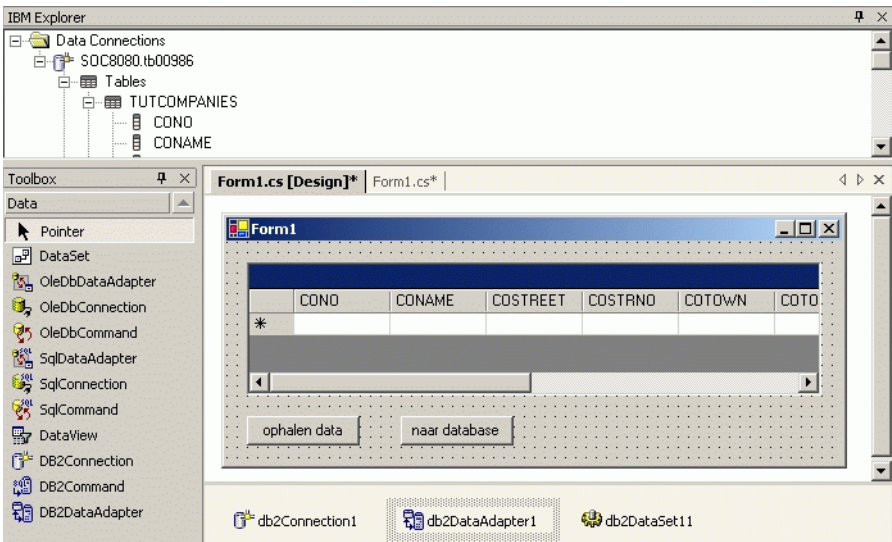
The screenshot displays the IBM Explorer interface. On the left, the 'Data Connections' tree shows a connection to 'SOC9090.tb00986' with a folder for 'Tables' containing 'MYTABLE', 'TUTCOMPANIES', and 'TUTPERSONS'. The 'Properties' window on the right shows the 'DatabaseObjectProperties' for 'TUTCOMPANIES', with fields for 'Name' (TUTCOMPANIES), 'Owner' (TB00986), and 'Type' (TABLE). Below this, the 'Data Results' window shows a table with 8 columns: CONO, CONAME, COSTREET, COSTRNO, CDTOWN, CDTOWNNO, and COCOL. The table contains 15 rows of data, with the first row highlighted.

	CONO	CONAME	COSTREET	COSTRNO	CDTOWN	CDTOWNNO	COCOL
▶	32	BERENDSE	EXSELSIOR	67 B45	BRUSSEL	1220	B
	25	DIEMEN	HEERBAAN	250	AMSTERDA	1014 BA	NL
	22	BELGISCHE	HOOFDSTR	675	ANTWERPE	2000	B
	21	PROCESS	DESGUINLEI	109 B56	ANTWERPE	2030	B
	20	OLIVER	GASTSTRAA	243	HASSELT	3500	B
	17	POUDRE BE	RUE D'EAU	67	NAMUR	7200	B
	15	DIRECT SOF	CORPORATI	14	BIRMINGHA	B2 4RN	GB
	13	FLYGT	HOOGSTRA	67	TERNAT	1200	B
	12	BROWN AG	HELGA FEL	3	MUNICH	8906	D
	11	C&C INTERN	GOSWELL R	521	LONDON	EC1N7JQ	GB
	10	BET MECHA	INDUSTRIEP	678	MECHELEN	1240	B
	9	TECHNISOF	ZWARTEWE	112	ROTTERDA	1420-WK	NL
	6	ASC COMPA	KAAL 345	67 b5	ANTWERPE	2030	B
	5	COVERDAT	WARMTEST	67	NAARDEN	1411 AR	NL
	4	ESCON	ANTONIUSL	2	GENT	8200	B
	3	ABIS N.V.	LADEUZEPL	17	LEUVEN	3000	B
	2	DATAWISHE	AVENUE LA	45 B6	BRUSSEL	1060	B
	1	LOC COMPU	RING	189	BRUSSEL	1020	B

Figuur 3: Via dit scherm laat men een dataset genereren



Figuur 4: Via de toolbox kan men een datagrid en buttons (niet afgebeeld) op het Windows Form zetten. Nadat men via het properties scherm van de datagrid deze laatste verbonden heeft met de de dataset (voor datasource en datamember waarden respectievelijk naam dataset en naam tabel kiezen) wordt deze direct aangepast aan de tabel en verschijnen de kolom namen.



Figuur 5: Wanneer men een nieuw project opstart kan men kiezen voor een DB2 databaseproject. In de solution explorer kan men dan bijvoorbeeld via de 'add new item' menu op de scripts map een script kiezen om een DB2 server-object aan te maken. Het script wordt geopend in de SQL-editor.

