

Klaas Brant over IBM's DB2 Universal Database

# Oh, Oh, ODBC ...

Van de zomer lag ik tijdens de hittegolf op het strand van Scheveningen en door mijn hoofd speelde het liedje van Harry Klorkestein: *Oh, oh, Den Haag*. Eenmaal weer aan het werk schrijf ik een programma om gegevens uit de database te halen. Omdat het programma 'database-onafhankelijk' moet zijn, kies ik er voor om ODBC te gebruiken. Deze interface heeft immers de belofte database-onafhankelijk te zijn. Er bestaan ODBC interfaces voor bijna alle databases en zelfs vele niet-databases zoals spreadsheets.

ODBC is een initiatief van Microsoft gebaseerd op de CLI standaard van de X-open groep. Het grote voordeel van een nieuwe standaard gebaseerd op een oude standaard, is dat je de voordelen van de oude standaard hebt maar de vrijheid hebt om dingen toe te voegen aan de standaard zonder bij jan en alleman te gaan vragen of dat goed is. Zo is ODBC inmiddels aan versie 3 toe en is het krachtig middel om databases te benaderen.

Het is dan ook niet zo verwonderlijk dat database-leveranciers in de rij staan om hun database ODBC-versie X compliant te laten verklaren. Waar die rij precies is, is niet duidelijk. Wie op zoek gaat naar hoe je ODBC compliant kan worden komt bedrogen uit, want hiervoor zijn geen procedures te vinden. Met andere woorden: iedere leverancier kan beweren dat zijn interface ODBC compliant is.

Als DB2 man was mijn eerste test met DB2 for Windows. Mijn gedachte was: "als het loopt, dan loopt het ook met SQL Server, Oracle, MySQL etcetera". Maar al tijdens het programmeren met DB2 kom ik vreemde zaken tegen. Je bent namelijk ODBC compliant, zonder dat je alle API's uit de interface ondersteunt. Als je pakweg zo'n 80 procent ondersteunt dan zit het wel snor en noem je jezelf ook 'compliant'. Dus als je de connectie met de database maakt moet je aan ODBC vragen wat er wel en niet ondersteund wordt. Dit is begrijpelijk, want sommige leveranciers zijn slechts versie 2 compliant en andere zijn voor ruim 80 procent compliant met versie 3.

Overigens zal het vrijwel onmogelijk zijn om alles te ondersteunen omdat er in de API calls ook zaken zitten die zeer moeilijk in de database te implementeren zijn. Denk maar

eens aan 'nested units of recovery'. Eens soort commit binnen een grotere commit scope? Maar via de ODBC API kun je er ook achterkomen wat wel en niet ondersteund is. Als je een programma wil maken dat geschikt is voor meerdere databases kan zorg je ervoor dat je geen speciale zaken gebruikt zoals 'Array Inserts' en je bepaalt de standaard set van API's die de database minimaal moet ondersteunen. Een soort 'geen fratsen' aanpak dus. Je programma gaat van start en zijn al je API's er dan is de database geschikt voor je programma. Overigens zijn ontbrekende API's vaak nog wel door eigen routines te vervangen.

Toen mijn programma op DB2 voor Windows liep, was de volgende test een DB2 voor z/OS (mainframe). Dat ging minder goed, want deze DB2 had andere ideeën over het verstrekken van metadata. Al snel werd het duidelijk dat ondanks dat ik ODBC gebruik, ik nog steeds in de coding moet weten met welke soort database gepraat wordt. Zelfs binnen de DB2 family is dat een feit. Dit is trouwens geen verrassing, "DB2 is geen DB2" roepen de Oracle-fanaten al heel lang. Geen ramp, want ik kan ook zonder de metadata. Wellicht zijn metadata ook fratsen?



Ik doe wat aanpassingen en nu loopt het programma op meerdere DB2-soorten. Tijd voor een niet-DB2 database, tijd voor SQL Server. Al snel komt mijn programma tot stilstand op de insert van een NULL value. Na wat speurwerk en met behulp van Internet kom ik er achter dat Microsoft in zijn wijsheid heeft beslist dat ook bij een NULL value het veld een lengte heeft. Dit is onzin natuurlijk, want NULL is niets. Maar goed, speciaal voor SQL Server heeft een NULL value een lengte van 1. De ODBC-documentatie heeft hier overigens geen mening over.

Ik zal u niet verder gaan vervelen met hoeveel if/ten/else- en case-constructies het programma nu heeft. Maar het *Oh, oh, Den Haag* is inmiddels overgegaan in *Oh, oh, ODBC*. Zal het in de toekomst net zo gaan met de .NET standaard?

**Klaas Brant** (kbrant@kbce.nl) is DB2-specialist en directeur van KBCE b.v. Meer informatie over DB2 is te vinden op [www.kbce.nl](http://www.kbce.nl) en [www.db2-times.com](http://www.db2-times.com)