

SQL Server on the move

SQL SERVER COMPACT EDITION NADER BEKEKEN

Met SQL Server 2005 Compact Edition is SQL Server 2005 nu ook voor mobiele apparaten beschikbaar. SQL Server CE biedt onder andere integratie met Microsoft Visual Studio en het .NET Compact Framework. In dit artikel beschrijft de auteur hoe ver de mogelijkheden gaan in vergelijking met SQL Server 2005 en hoe het zit met synchronisatie en security.

SQL Server 2005 CE is geen nieuw product, maar een evolutie van de SQL Server-familie die enkele jaren geleden is uitgebreid naar het mobiele platform van Microsoft. In 2000-2001 waren we enthousiast over het dynamische duo van eMbedded Visual Basic (eVB) en Pocket Access. Het was toen een geschenk uit de hemel voor Visual Basic- en Access-programmeurs die hiermee op een snelle manier applicaties voor de Pocket PC 2000 zouden kunnen ontwikkelen met oude vrienden zoals het ActiveX Control en ADO. Het is wel grappig wanneer je bedenkt dat het in het begin gebracht werd alsof devices met 32MB RAM een sterk begrensde omgeving hadden ten opzichte van applicaties op de desktop. Zelfs nu vinden we dat nog. In de jaren negentig bouwden we immers applicaties met Visual Basic 3.0 en Access 2.0 en die hadden 4MB RAM beschikbaar. Visual Studio 2003 verscheen met SQL Server CE 2.0 als de lokale opslag voor het Compact Framework. Als antwoord op klanten die een algemene opslag wilden van een device naar Tablet PC, veranderde de naam van SQL Server CE naar SQL Server Mobile 3.0 en verscheen het samen met Visual Studio 2005. Hierin werd de SQL Server CE-engine beschikbaar gemaakt voor het gehele desktopplatform en niet alleen voor devices en de Tablet PC. SQL Server CE is ontworpen om aan de eisen van applicaties tegemoet te komen, waarvoor het belangrijk is om een light versie van de SQL Server 2005-database te hebben.

SQL Server CE

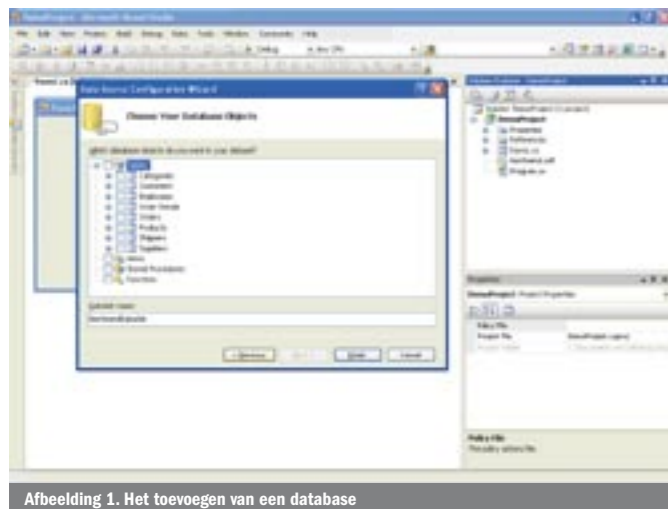
Wanneer kies je voor het gebruik van SQL Server CE en wanneer niet? Als je lokale dataopslag nodig hebt voor de Pocket PC, de Smartphone of je hebt dezelfde dataopslag nodig voor verschillende devices (van smartphone tot desktop) dan kies je voor SQL Server CE. Heb je een database nodig die als een service draait, dan kies je bijvoorbeeld voor SQL Server 2005. Een applicatie communiceert meestal met een back-end database. Wanneer er geen back-end database aanwezig is, de performance een grote rol speelt of de opslagcapaciteit beperkt is, dan kan ook lokale opslag worden gebruikt. Logische keuze is dan een lokale database-engine, waarbij de basisfunctionaliteit van een relationele database aanwezig is. Daarbij moet je denken aan query-functionaliteit, simpele transacties, data-caching en de algemene datatypes (zie tabel 1). De database wordt dan benaderd vanuit één client, maar is niet beperkt tot één connectie. De applicatie kan communiceren met een centrale dataservice en moet altijd blijven functioneren, zelfs als er geen verbinding is. Denk maar eens aan Outlook. Bij de meeste werkzaamheden die je daarin doet, wordt gebruikgemaakt van data die lokaal op jouw pc zijn opgeslagen. Wanneer er een connectie is met de Exchange Server, dan wordt de lokale dataopslag gesynchroniseerd met de centrale dataservice. In tabel 1 staat vermeld wat wel en niet mogelijk is binnen SQL Server CE.

Visual Studio en SQL Server Management Studio

Met SQL Server CE kun je applicaties ontwikkelen binnen Visual Studio 2005 en tevens gebruikmaken van SQL Server Management Studio. Het programmeren van een applicatie die gebruikmaakt van SQL Server CE is eigenlijk niet veel anders dan wat je al gewend bent bij het maken van bijvoorbeeld een Windows-applicatie. Je kunt jouw database gewoon toevoegen aan jouw project (zie afbeelding 1 voor de Northwind-database). De database bestaat uit één datafile (Northwind.sdf) en heeft dus geen logfile. In codevoorbeeld 1 wordt getoond hoe je een connectie kunt maken, hoe je gegevens op kunt halen en data kunt koppelen aan bijvoorbeeld een datagrid. Programmeren tegen een SQL Server 2005 CE-database is bijna identiek aan het programmeren tegen een SQL Server 2005-database. Bij datatoegang naar een dataservice denk je al gauw aan

Optie	Aanwezig binnen SQL Server CE
Installatiegrootte	1.7 MB download size 1.8 MB op schijf
Click Once Deployment	Ja
Installatieopname binnen de applicatie	Ja
Geen administrator nodig bij installatie	Ja
64-bit support	Nee
Runt als een service	Nee
Bestandsformaat	Enkele file
Datafile-opslag op een netwerk-share	Ja
Ondersteuning voor verschillende bestandsextensies	Ja
Database size support	4GB
XML-opslag	Ja als nText
Transact-SQL Algemene query-faciliteiten	Ja
Procedurale T-SQL Select Case, If, features	Nee
Simpele transacties	Ja
Gedistribueerde transacties	Nee
Native XML/XQuery/QPath	Nee
Stored Procedures, views, triggers	Nee
Role Based Security	Nee
Aantal concurrent connecties	256
Zoeken op index	Ja
Referentiële integriteit	Ja
Data-types	Real, Numeric, NVarChar, NText, Image, Integer, DateTime, VarBinary, UniqueIdentifier, Money

Tabel 1.



Afbeelding 1. Het toevoegen van een database

schikbaarheid en concurrency. Als er veel verschillende gebruikers zijn, dan wil je geen onnodige connecties openhouden naar de database. Dit in verband met performance en het wijzigen van identieke records door meer gebruikers. Om dat te bewerkstelligen maak je een kopie van de data, die wijzig je en de wijzigingen geef je door aan de dataservice. ADO.NET maakt dit mogelijk door middel van datasets en data-adapters (disconnected data). Bij SQL Server CE is de database zelf de kopie van de data. Daardoor is het vaak gemakkelijker om een connectie te openen en te koppelen aan een updatable query. SQL Server CE ondersteunt een cursor die vergelijkbaar is met een dataset maar een betere performance kent: `SqlCeResultSet`; zie codevoorbeeld 1. Wat de cursor vergelijkbaar maakt met een dataset is dat hij is te wijzigen en dat je er doorheen kunt lopen. In hoeverre je dit laatste wilt, kun je instellen met de `ResultSetOptions`. In het artikel van Dennis Mulder en Olaf Conijn over de Enterprise Library wordt al genoemd dat het Data Access Block ondersteuning biedt voor SQL Server CE. In codevoorbeeld 2 kun je zien hoe je een connectie maakt met de database door de methode `CreateDatabase` aan te roepen. Hieraan geef je een sleutel mee (`SQLSERVERCE`) waarmee in de app.config de connectiestring kan worden opgehaald; zie codevoorbeeld 3. Vervolgens wordt een nieuwe unieke identifiër opgehaald. Let op: dit voorbeeld is gemaakt met Microsoft Enterprise Library 3.0 - February 2007 CTP met default opties aan tijdens installatie. Binnen SQL Server Management Studio kun je een SQL ServerCE-database openen, beheren (tabellen inclusief inhoud wijzigen, verwijderen of aanmaken) en een nieuwe database creëren.

```
using System.Data.SqlServerCe;

_conn = new SqlCeConnection(@"Data Source=C:\Mobile\example\Northwind.sdf");
SqlCeCommand cmd = new SqlCeCommand();
cmd.Connection = _conn;

//Selecteer gegevens uit de employees tabel.
cmd.CommandText = "SELECT [Employee ID], [Last Name], [First Name],Photo
FROM Employees";

// Openen van de connectie
_conn.Open();

// Maak een scrollable/updateable cursor aan.
SqlCeResultSet resultSet = cmd.ExecuteResultSet(
    ResultSetOptions.Scrollable | ResultSetOptions.Updateable);

// Vul een datasource met de data en koppel deze aan
// bijvoorbeeld een datagrid
this.bindingSource1.DataSource = resultSet;
```

Codevoorbeeld 1.

```
using Microsoft.Practices.EnterpriseLibrary.Data;

namespace EntLibArticle
{
    class Program
    {
        static void Main(string[] args)

            // Open een connectie met de database
            Database db = DatabaseFactory.CreateDatabase("SQLServerCE");
            // Voer een query uit op de database en schrijf
            // het resultaat weg naar de console.
            Console.WriteLine(db.ExecuteScalar(CommandType.Text,
                "select newid()").ToString());
            Console.ReadLine();
        }
    }
}
```

Codevoorbeeld 2.

Security

SQL Server CE is onderdeel van de SQL Server-familie. Je kunt er dus van uitgaan dat het gebruikmaakt van bewezen technologie. Zo kan de database omgaan met apparaten die ineens uitgaan of waarvan de batterij het niet meer doet, waardoor het apparaat niet meer opgeladen kan worden. De belangrijkste vorm van beveiliging bestaat uit het beveiligen van jouw data. Dit kan op de volgende manieren:

- 128-bit RSA-file-gebaseerde encryptie op devices voor database file-security.
- Een encrypted dataformaat met wachtwoordbescherming.
- Een single file format, waardoor document-safety (zie uitleg hieronder) ontstaat.
- Synchronisatie via 128-bit SSL-transport voor veilige en betrouwbare connectie, zelfs door firewalls.

Je wilt natuurlijk alleen die mensen fysieke toegang geven tot de datafile die daartoe ook gerechtigd zijn. SQL Server CE is zo gebouwd dat de gebruiker altijd toegang heeft tot de file. Dit betekent dat een gebruiker langs jouw applicatie kan gaan en via tools als MSQuery de ruwe data kan bekijken en zelfs kan wijzigen. Met SQL Server CE kun je de datafile beschermen met een wachtwoord en encryptie. (Dit geef je aan bij het aanmaken van een database in SQL Server Management Studio). Bescherming van de databasefile door middel van een wachtwoord voegt een laag van bescherming toe die meereist met de datafile. Hierdoor wordt het moeilijker om toegang te verkrijgen, ook al is de file in verkeerde handen gevallen. SQL Server CE ondersteunt wachtwoorden tot veertig karakters lang, waarbij gebruik mag worden gemaakt van letters, getallen, en alle symbolen in het Windows CE Operating System. Het wachtwoord beschermt niet alleen jouw datafile, maar levert ook de hash die wordt gebruikt als de sleutel om jouw databasefile te versleutelen. Microsoft slaat die op aan het begin van de databasefile. SQL Server CE gebruikt het MD5 hashing-algoritme om een 128-bit sleutel te maken voor het RC4 encryptie-algoritme. Wanneer je dit niet voldoende vindt, kun je altijd nog gevoelige data zoals wachtwoorden en creditcardnummers versleutelen voordat ze worden opgeslagen in de database. Een enkel wachtwoord biedt dus totale toegang tot de gehele file. De gegevens worden al verdeeld voordat ze op de lokale machine arriveren. Zelfs wanneer meer mensen één en dezelfde laptop delen, dan nog heeft elke gebruiker zijn eigen datafile met zijn gedeelte van de data. Omdat de datafile van SQL Server CE geen stored procedures of andere code binnen de database toestaat, wordt dit formaat als 'document safe' beschouwd. Anders dan bij data-service file-formats (bijvoorbeeld de datafile van SQL Server 2005), kun je bij SQL Server CE geen code plaatsen in de database waardoor je een event kunt uitvoeren. Document-safety geeft je daarnaast ook de mogelijkheid om, wanneer je bijvoorbeeld een energie-applicatie hebt geschreven, jouw file-extensie te associëren met jouw applicatie (bijvoorbeeld .energie). Om nog veiliger te werken, kun je natuurlijk



Afbeelding 2. Het aanzetten van de encryptie-optie

overwegen om gebruik te maken van een aantal Windows-opties. Denk daarbij aan het gebruik van 'strong passwords', het vragen van een wachtwoord wanneer het apparaat in stand-by-modus gaat en het locken van het apparaat wanneer de screensaver aanslaat.

Een aanvullende beveiligingsoptie die ik hier bespreek, is het Windows Encrypted File System (EFS). Hiermee versleutel je de datafile op het operating system-niveau. Door het selecteren van properties op jouw datafile, kun je advanced opties kiezen en de optie 'Encrypt contents to secure data' aanzetten; zie afbeelding 2. Omdat SQL Server CE in-process draait binnen de hosting-applicatie, die onder de context draait van de huidige gebruiker, is EFS altijd beschikbaar. Wanneer gebruikgemaakt wordt van EFS, kan de gebruiker de file niet kopiëren zonder de mogelijkheid te verliezen om de data te versleutelen. EFS zorgt er ook voor dat gebruikers niet zomaar files kunnen openen en kopiëren die door andere gebruikers zijn versleuteld. Verder voorkomt het mounten van hard drives toegang via een per ongeluk gedeelde folder. Het beveiligen van zowel de authenticatie-informatie als de data, terwijl ze het netwerk doorkruisen, is het moeilijkste gedeelte van het bouwen van een applicatie die gebruikmaakt van SQL Server CE, vooral als het netwerk draadloos is. Daardoor moet je vooral oefenen hoe je 'veilig' gebruikmaakt van de synchronisatietechnieken RDA en Merge Replication, die ik hierna zal behandelen.

Datasynchronisatie

De belangrijkste methoden voor datasynchronisatie die door SQL Server CE worden ondersteund zijn: Remote Data Access (RDA) en Merge Replication. RDA houdt de wijzigingen bij die gemaakt worden in de database van het apparaat en zorgt ervoor dat die wijzigingen worden doorgegeven aan de centrale database. Merge Replication houdt de wijzigingen bij die gemaakt worden in zowel de database van het apparaat als de centrale database. Wanneer de data gesynchroniseerd worden, wisselen beide databases hun wijzigingshistorie met elkaar uit om de databases met elkaar te verenigen. Daarbij gaat Merge Replication door een fase waarin de wijzigingen aan beide kanten worden bekeken om te bepalen welke data exact worden uitgewisseld. In deze fase worden automatisch conflicterende wijzigingen gedetecteerd en wordt actie ondernomen. Hoe deze conflicten worden afgehandeld, is totaal configureerbaar en kan worden aangepast. Je zult merken dat Merge Replication nog een aantal voordelen heeft ten opzichte van RDA. Ten eerste kan de synchronisatie zich afspelen op de achtergrond. Het proces speelt zich dan af op een eigen thread, zodat de user-interface van jouw applicatie actief blijft. Ten tweede kunnen 'process-events' jouw applicatie een update geven van de status van het synchronisatieproces. Daarbij kun je denken aan het exacte moment per tabel waarop wijzigingen worden uitgewisseld en de status van de eerdergenoemde fase, waarin een beslissing wordt genomen over conflicterende wijzigingen.

```
<connectionStrings>
  <add name="Connection String"
        connectionString="Database=Database;Server=(local)\SQLEXPRES;Integrated Security=SSPI"
        providerName="System.Data.SqlClient" />
  <add name="SQLServerCE" connectionString="Data
        Source=C:\Projects\EntLibArticle\EntLibArticle\MyDatabase#1
        sdf;Persist Security Info=False;"
        providerName="System.Data.SqlServerCe" />
</connectionStrings>
```

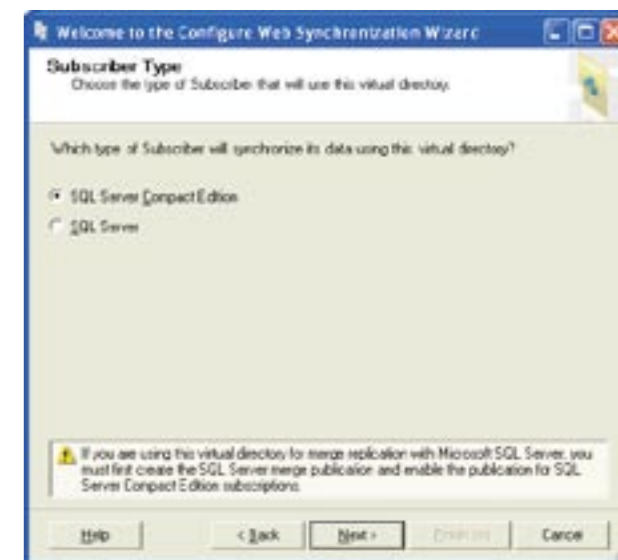
Codevoorbeeld 3.

Via een webserver

Zowel RDA als Merge Replication voeren hun synchronisatieproces uit via een webserver. Datawijzigingen worden verstuurd over Hypertext Transfer Protocol (HTTP) of Hypertext Transfer Protocol Secure (HTTPS). Dit is een enorm voordeel omdat je dan, elke keer dat jouw apparaat online is, data kunt synchroniseren met de centrale database. Daarbij maakt het niet uit of dit gebeurt door middel van een snelle of een langzamere verbinding. Microsoft gebruikt IIS dus als een bemiddelaar tussen SQL Server CE Client Agent met de SQL Server CE Server Agent. Deze laatste is een ISAPI-extensie die op IIS draait. Om dit te kunnen doen, installeer je eerst de SQL Server CE Server Tools. Vervolgens maak je een shared folder aan (bijvoorbeeld Replication-Data) in de Explorer. Via de Web Synchronization Wizard doe je dan de rest van de instellingen. Je komt in deze wizard door onder het Start-menu eerst te kiezen voor Programs, daaronder kies je Microsoft SQL Server 2005 Compact Edition en daaronder selecteer je weer de Configure Web Synchronization Wizard. Laten we die eens opstarten. Zoals te zien in afbeelding 3 kies je een subscriber, bijvoorbeeld SQL Server CE. Daarna wordt gevraagd om een nieuwe Virtual Directory aan te maken. De vraag of je de SQL Server Agent wilt kopiëren en registreren, beantwoord je met ja. Dan geef je aan of je gebruik wilt maken van SSL en hoe je de authenticatie wilt laten verlopen; zie afbeelding 4. In afbeelding 5 geef je de naam van jouw gedeelde folder op. Uiteindelijk krijg je een samenvatting van wat je hebt ingevoerd. Normaliter gebruik je natuurlijk een aparte server voor jouw synchronisatie en jouw data.

Aanmaken RDA-object

In codevoorbeeld 4 zie je hoe je in .NET een RDA-object kunt aanmaken. De url die je daar ziet, is verkregen door middel van de wizard. Wanneer je met RDA nu een PULL-request doet (zie codevoorbeeld 5), zendt de Client Agent SELECT-queries naar de Server Agent. Daarbij wordt er data ontvangen en opgeslagen in een SQL Server CE data-



Afbeelding 3. Het kiezen van een subscriber

base. Doe je een PUSH-request (zie codevoorbeeld 6), dan wordt een lijst met wijzigingen opgehaald uit de lokale database. Die lijst wordt verpakt als een HTTP(S) POST-command, dat naar de server agent wordt gestuurd. Bij Merge Replication gebruikt de Client Agent de Server Agent als een proxy om de subscriptie mogelijk te maken. Wanneer een gebruiker een wijziging maakt, haalt de Client Agent een lijst met wijzigingen op en stuurt deze naar de Server Agent.

De Server Agent heeft dus drie rollen binnen datasynchronisatie:

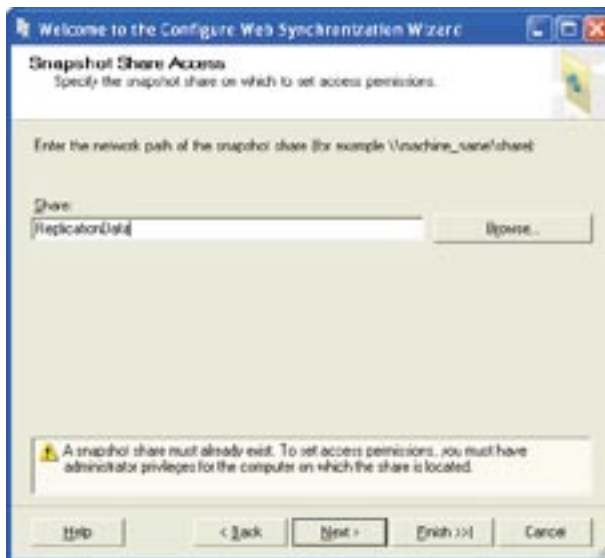
- Het uitwisselen van data
 - Connectie maken met de centrale database uit naam van de client
 - Authenticatie van gebruikers en het beveiligen van de datastroom
- Een uitval van de connectie wordt door de Server Agent opgelost door records op schijf op te slaan. Wanneer de verbinding is hersteld, wordt verder gegaan vanaf het laatste bevestigde record. Omdat de Client Agent het HTTP-protocol gebruikt om te communiceren met de Server Agent, wordt alle toegang tot de ISAPI DLL geauthenticeerd door IIS. Dit proces is exact hetzelfde als het proces voor een reguliere website. Let wel, als je werkt met 'Basic Authentication', dan moet je de datastroom tussen de Client Agent en de Server Agent altijd versleutelen. Voor andere vormen van synchronisatie kun je denken aan webservices en SQL Server CE biedt ook nog de mogelijkheid om te synchroniseren met een Access-database op de desktop. Op dit moment is Microsoft bezig om een nieuwe vorm van synchronisatie aan te kunnen bieden (OCS Sync genaamd). Dit is een synchronisatiemethode die tussen RDA en Merge Replication in staat. Uiteindelijk zal die RDA compleet vervangen. Een groot voordeel van deze synchronisatie is dat je dan niet alleen afhankelijk bent van SQL Server, maar ook kunt synchroniseren met bijvoorbeeld een Oracle-database.

Deployment

Het is handig om te weten hoe jij jouw applicatie in combinatie met de database kunt uitleveren. SQL Server CE kan worden meegenomen in de set-up van jouw applicatie. Om de database op een centrale plek te installeren, heeft SQL Server CE een MSI-deploymentmodel (SQL Server CE wordt dan in een andere folder opgeslagen dan jouw applicatie). ClickOnce kan helpen om de database verplicht in de installatie van jouw applicatie mee te nemen. Hier zijn wel administratieve rechten voor nodig. Omdat SQL Server CE uit DLLs bestaat die in-process binnen de hosting-applicatie draaien, kan SQL Server CE ook uitgeleverd worden met de applicatie. Bij deze vorm van uitleveren zijn geen administratieve rechten nodig en de databasefile komt dan in jouw applicatiefolder te staan. Realiseer je dan wel, dat de DLLs niet meer geregistreerd worden met de Add/Remove Programs en niet meer worden aangeraakt door Microsoft Update. Een voordeel is, dat wanneer je de applicatie wilt verwijderen, er geen extra stappen nodig zijn.



Afbeelding 4. Aangeven of je wel of niet gebruik wilt maken van SSL



Afbeelding 5. Het opgeven van de naam van de gedeelde folder

```
SqlCeRemoteDataAccess rda = new SqlCeRemoteDataAccess();
//url is aangemaakt via de Configure Web Synchronization Wizard
rda.InternetUrl = "http://localhost/Sync/sqlcesa30.dll";
rda.LocalConnectionString = Properties.Settings.Default.
LocalNorthwindConnectionString;
return rda;
```

Codevoorbeeld 4.

```
rda.Pull("Contacts", "SELECT CustomerID, Company
Name, ContactName, ContactTitle, Address, City, Region,
PostalCode, Country, Phone, Fax FROM Customers"
, Properties.Settings.Default.Server0
leDbNorthwindConnectionString,
RdaTrackOption.TrackingOnWithIndexes, "Contacts_Errors");
```

Codevoorbeeld 5.

```
rda.Push("Contacts", Properties.Settings.Default.Server0leDbNorthwind
ConnectionString, RdaBatchOption.BatchingOn);
```

Codevoorbeeld 6.

Gemakkelijk op te zetten

In het voorgaande heb je gezien dat SQL Server CE een gemakkelijk op te zetten database is met een compactheid van 1.7 MB (tabel 1). Je kunt met SQL Server CE er zelfs voor kiezen of je een aparte deployment wilt doen of dat je het met de applicatie mee wilt uitleveren, en welke vorm van beveiliging en synchronisatie voor jou het beste is. Denk er altijd goed over na of SQL Server CE de functionaliteit biedt die jouw applicatie nodig heeft.

Jacqueline van der Holst werkt als consultant bij Avanade (www.avanade.com), een samenwerkingsverband van Microsoft en Accenture. Zij houdt zich op dit moment bezig met LINQ, SQL Server CE en Smart Client-applicaties. Jacqueline's weblog is te vinden op www.jacquelinevan-derholst.nl. Voor vragen en opmerkingen is zij te bereiken op jacquelinev@avanade.com.

Referenties

Algemene informatie SQL Server CE: <http://www.microsoft.com/sql/editions/compact/default.mspix>
 Download SQL Server 2005 CE: <http://www.microsoft.com/sql/editions/compact/register.mspix>
 Download SQL Server 2005 CE Server Tools: <http://www.microsoft.com/downloads/details.aspx?FamilyId=4E45F676-E69A-4F7F-A016-C1585ACF4310&displaylang=en>
 SQL Server CE Books Online: <http://www.microsoft.com/downloads/details.aspx?FamilyId=E6BC81E8-175B-46EA-86A0-C9DACA84C85&displaylang=en>
 Informatie voor het ontwikkelen van mobiele applicaties: <http://msdn2.microsoft.com/en-us/windowsmobile/default.aspx>