



# 'Na bouw van een ASP.NET site wil je niet meer terug naar ASP/COM+'

ONLINE ACCOUNTANCYDIENST COLUMBUS VOLLEDIG ONTWIKKELD MET VISUAL STUDIO .NET

Deze case beschrijft de technische ins en outs bij de ontwikkeling van de ASP.NET-site voor Columbus, een online accountancydienst.

Voor de architectuur van Columbus was een aantal voorwaarden gesteld:

- de site moet eenvoudig schaalbaar zijn door inzet van extra server;
- wegens veiligheidsredenen moeten de webpagina's gescheiden worden van de business logic en de data-opslag.

Als referentiemodel voor de architectuur is een voorbeeldsite van Microsoft zelf gekozen: Duwamish 7.0. Dit is een fictieve e-commerce site voor online boekverkoop. Duwamish laat de 'best practices' zien voor architectuur en design van een schaalbare webapplicatie, en is een bewezen model voor een succesvolle implementatie van een ASP.NET-website. Microsoft biedt meer van dit soort complete voorbeeldsites (inclusief code) aan, waaronder:

- IBuySpy Store (een online winkel voor spionnen);
- IBuySpy Portal (een voorbeeld portal site - download op [www.ibuyspy.com](http://www.ibuyspy.com));
- PetShop (een .NET port van de J2EE PetShop blueprint site);
- Fitch and Mather (een fictieve online stock-broker).

Een overzicht van de architectuur van Duwamish staat in afbeelding 1.

De architectuur van Columbus bestaat uit vier logische lagen:

### 1. Webpagina-laag

Deze laag biedt de toegang tot de site, en bestaat uit ASPNET Web Forms en code-behind bestanden. De Web Forms tonen uiteindelijk de HTML, terwijl de echte code (die bijvoorbeeld wordt uitgevoerd bij het klikken van buttons en links) in de code-behinds staat.

### 3. Facadelaag

De enige functies die door de webpagina's aangeroepen mogen worden staan in deze facadelaag. Feitelijk is deze laag niets meer dan een interface naar de business rules-laag, en wordt deze

gebruikt om de interne functies af te schermen. De Facade-laag kan indien gewenst worden benaderd via Remoting (het aanroepen van .NET-objecten op een andere server). Je kunt Remoting zien als de .NET implementatie van DCOM.

### 4. Business rules-laag

Hier gebeurt het echte werk. De gegevens uit de data-laag worden hier omgevormd en samengevoegd. Alle bedrijfsregels zijn hier geïmplementeerd.

### 5. Data-laag

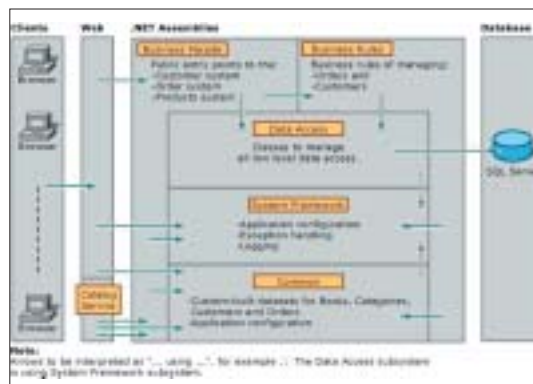
De data-laag biedt de toegang tot de database.

De opsplitsing in deze vier lagen bleek erg goed te werken. Vooral het gebruik van de

facadelaag verplicht de software-ontwikkelaars om na te denken welke functionaliteit het systeem aan de user-interface (de webpagina's) wil aanbieden.

## Inzet BizTalk

Voor Columbus is Biztalk 2000 ingezet om de deelprocessen, die nodig zijn bij het verwerken en calculeren van de financiële bedrijfsgegevens, visueel inzichtelijk en op een robuuste manier



Afbeelding 1. Architectuur van Duwamish

## De case

Columbus is een via Internet beschikbare accountancydienst (gericht op zowel de top 100 bedrijven als het MKB), die managers helpt hun bedrijf beter te monitoren en te sturen. Columbus biedt een heldere en gebruikersvriendelijke interface voor de financiële status van een bedrijf over meerdere perioden, waarbij trends en zelf ingestelde indicatoren door middel van 'stoplichtjes' aangeven of er bijsturing nodig is. De klant levert gegevens uit zijn boekhoudprogramma via Internet aan, waarna de accountant deze analyseert, en adviezen via de site teruggeeft. Dit alles wordt natuurlijk met de hoogste internetbeveiliging (client- en servercertificaten) omgeven. Uiteraard bevat Columbus nog veel meer functionaliteit, maar dat ligt buiten de scope van deze case. Meer informatie is te vinden op <http://www.newcolumbus.com>.

### De feiten:

Opdrachtgever	KPMG Columbus
Projectuitvoerder	Macaw
Server software	Windows 2000, IIS 5, SQLServer 2000, Biztalk 2000
Technologieën	ASP.NET, Web Forms, Server Controls, C#, Xml, .NET Remoting
Ontwikkelmethodiek	Rational Unified Process (RUP)



aan elkaar te koppelen. Het visueel koppelen van de verschillende onderdelen werkte erg prettig voor de opdrachtgever, zodat deze duidelijkheid kreeg over welke processen elkaar opvolgden. Bovendien was het voor de software-ontwikkelaars overzichtelijker en beter hanteerbaar om ingewikkelde functionaliteit te bouwen in kleinere deelprocessen.

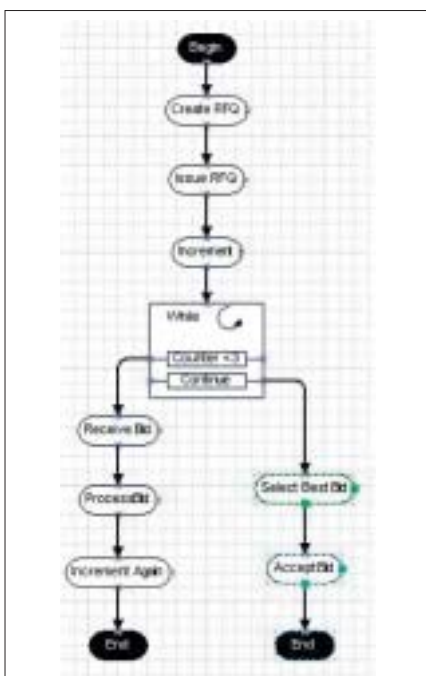
## Visual Studio.NET en SourceSafe

Columbus is volledig gebouwd met Visual Studio .NET, wat de krachtigste tool is om snel en eenvoudig een .NET-applicatie te bouwen (en te debuggen). Vanuit het oog-

punt van een software-ontwikkelaar zijn de meest in het oog springende functies:

- *Intellisense*. Was al aanwezig in Visual Interdev, maar nu een stuk uitgebreider.
- *Drag-click-code mogelijkheden*. Het bouwen van een webapplicatie gaat nu net zo eenvoudig als een desktopapplicatie. Sleep de gewenste webcontrols (bijvoorbeeld een button of invoerveld) op je pagina, dubbelklik op het object en je komt direct in de code-behind functie.
- *Debugging, watching en tracing*. De debug-mogelijkheden van Visual Studio .NET zijn erg sterk, en helpen om snel fouten op te sporen.

Aan de bouw van Columbus heeft een team van acht softwareontwikkelaars tegelijkertijd gewerkt. Hiervoor was het nodig om een source control-systeem te gebruiken, waarvoor Visual SourceSafe uitermate geschikt bleek. De integratie met Visual Studio .NET is naadloos, maar de plaats van de sources is voor de doorgewinterde ASP-ontwikkelaar wellicht even wennen. Waar je vroeger met een team ontwikkelde op een centrale webserver, ontwikkelt iedereen nu op z'n eigen lokale webserver. Dit is ook logisch omdat in ASP.NET de 'hele' site foutloos moet kunnen compileren, voordat deze opgestart kan worden. Het zou anders erg lastig zijn als je de centrale code niet kunt starten omdat een andere ontwikkelaar ergens een fout heeft gemaakt.



Afbeelding 2. BizTalk, de volgorde van de processen in beeld gebracht

Duits, Nederlands en Vlaams) beschikbaar zou zijn. Meertaligheid in .NET is te omschrijven met twee begrippen:

### • Globalisatie

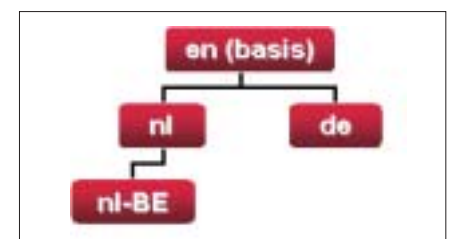
Hieronder vallen de verschillende formatteringen voor bijvoorbeeld data en getallen. Bijna alle objecten in .NET zijn globaliseerbaar. Door het zetten van de gewenste 'culture' zal bijvoorbeeld het kalender control zich automatisch in een bepaalde taal kunnen afbeelden.

### • Localisatie

Met localisatie wordt bepaald in welke taal een vertaling van een tekst wordt getoond. Een voorbeeld hiervan is een knop die in het Engels de tekst 'next' toont, en in het Nederlands de tekst 'volgende'.

Voor localisatie moet het systeem beschikken over zogeheten 'resource files', één per taal, waarin de in de site gebruikte teksten voor een bepaalde taal staan opgesomd in XML-vorm. Deze XML 'resource files' worden vervolgens door VS.NET automatisch gecompileerd tot 'satellite assemblies' - feitelijk kleinere DLL's waarin geen code, maar alleen de vertaalde teksten staan.

Om een site gebruik te laten maken van deze 'satellite assemblies', worden ze in een boom-structuur onder de bin directory geplaatst. Afbeelding 3 toont de basisapplicatie in het Engels (en), met daaronder de Nederlandse (nl) en Duitse (de) vertalingen. De Vlaamse satellite assembly (nl-BE) hoeft alleen de teksten te bevatten die afwijken van de Nederlandse.



Afbeelding 3

## Toveren met ASP.NET Wizards

In Columbus wordt veel gebruik gemaakt van zogenaamde "wizards", die de gebruiker helpen bij het uitvoeren van bepaalde taken. In de tijd van ASP 3.0 waren deze wizards niet echt triviaal om te bouwen,

## Meertaligheid in .NET

Voorwaarde bij de bouw van Columbus was dat de site in meerdere talen (Engels,

want tussen alle wizard-schermen moesten de reeds ingevulde waarden tijdelijk worden opgeslagen en eventueel weer worden ingevuld in de juiste controls als de gebruiker een stap terug deed in de wizard.

Met ASP.NET en Web Forms is dit heel eenvoudig geworden. In ASP.NET kunnen alle schermen van de wizard namelijk op slechts één pagina gezet worden. Alle controls die bij hetzelfde scherm horen, worden geplaatst in panels, waarvan er altijd maar één op 'visible' wordt gezet. De overige schermen worden dan op de server gefilterd, en zijn niet zichtbaar in de uiteindelijke HTML. Bovendien zorgt een Web Form er zelf voor dat de invoer op het scherm blijft behouden bij een post naar zichzelf. Zie afbeelding 4 hoe de ontwikkeling van een ASP.NET Web Form wizard eruit ziet.

### Microsoft Building Blocks

Iedereen die wel eens .NET-code heeft geschreven voor het aanroepen van stored procedures in SQL Server, weet dat dit nogal omslachtig is. Afbeelding 5 bevat als voorbeeld een stuk C# code waarbij de stored procedure 'getProductsByCategory' aangeroepen wordt voor categoryId 27.

Om het benaderen van SQL Server databases gemakkelijker te maken, biedt Microsoft een 'Data Access Application Block'.

```
// connectie openen...
SqlConnection connection = new SqlConnection (connectionString);
connection.Open();
// command zetten...
SqlCommand command = new SqlCommand("getProductsByCategory",
    connection);
command.CommandType = CommandType.StoredProcedure;
// parameters vullen...
SqlParameter idParam = command.Parameters.Add("@CategoryId",
    SqlDbType.Int);
idParam.Value = 27;
// dataset ophalen...
SqlDataAdapter adapter = new SqlDataAdapter(command);
DataSet products = new DataSet();
adapter.Fill (products);
// afsluiten connectie...
connection.Close();
```

Afbeelding 5.

```
DataSet products = SqlHelper.ExecuteDataset(connectionString,
    CommandType.StoredProcedure, "getProductsByCategory", 27);
```

Afbeelding 6.

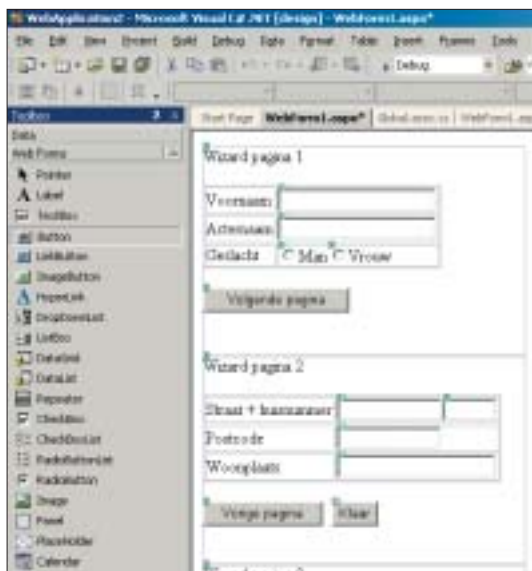
Dit Building Block bevat sourcecode en documentatie, waarmee de beste manieren worden aangeboden om op een snelle en schaalbare manier SQL Server te benaderen. Dezelfde aanroep van 'getProductByCategory' wordt dan ook meteen een stuk korter (zie afbeelding 6).

Om deze Building Block te downloaden zoek je in msdn.microsoft.com op "Data Access Application Block". Tip: kijk ook eens naar het Building Block voor het managen van Exceptions.

kopiëren van de bestanden, vandaar de veelgebruikte term 'XCopy deployment'.

### Nooit meer terug

Voor software-ontwikkelaars geldt: Nadat je een ASP.NET site gebouwd hebt, wil je niet meer terug naar ASP/COM+.



Afbeelding 4. Screenshot van de ontwikkeling van een ASP.NET Web Form wizard

### XCopy deployment

Het uitrollen van een site naar een productie-omgeving is met ASP.NET veel eenvoudiger geworden. Waar bij ASP/COM+ projecten nog extra instellingen werden gedaan in de registry (bijvoorbeeld voor het opslaan van de connectiestring) en de IIS admin tool (bijvoorbeeld voor het configureren van alternatieve error-pagina's), zijn dit soort instellingen in ASP.NET allemaal te doen in één bestand: de Web.config. Het installeren van een site is nu een kwestie geworden van