

# Commerce Server 2002

UITSTEKENDE BASIS VOOR KLEINE OF GROTE WEBWINKEL

Microsoft Commerce Server 2002 is begin 2002 verschenen en was één van de eerste producten van Microsoft die .NET enabled was. Waar je in Commerce Server 2000 nog aangewezen was op het gebruik van ASP, kun je nu gebruik maken van ASP.NET en dat biedt veel voordelen. In dit artikel wordt getoond wat nodig is om snel een complete webwinkel of interne inkoopsite te bouwen en waar de webontwikkelaar op moet letten.

Wat heb je allemaal nodig om een winkel op het internet te beginnen? Voor een leuke winkel heb je het volgende nodig: 1) iets om producten mee te beheren; 2) een database om gegevens over klanten in op te slaan; 3) een winkelmandje; 4) iets om kortingsacties mee te beheren, zodat deze automatisch uitgerekend worden; en 5) weten hoeveel klanten dagelijks langskomen en in welke producten ze het meest geïnteresseerd zijn. Ook nodig is een gemakkelijk beheer van alle informatie rond klantenservice, algemene voorwaarden, interessante informatie over je producten, enzovoort. Wanneer je al een back-office hebt waarin gegevens over je producten staan of waarin orders verwerkt worden, heb je iets nodig om deze gegevens op de website te krijgen en vice versa. Aan de eerste vijf wensen kun je met Commerce Server voldoen, de andere twee wensen kun je met andere Microsoft-producten vervullen.

## Productcatalogus

Welke onderdelen bevat Commerce Server en hoe kunnen we die gebruiken in onze winkel? Bovenaan ons boodschappenlijstje staat de mogelijkheid om producten te beheren. Alles wat met producten te maken heeft, wordt in Commerce Server opgeslagen in de productcatalogus, die je kunt beheren via de BizDesk (waarover later meer). Je kunt in Commerce Server verscheidene catalogi aanmaken, waarbij onderscheid gemaakt wordt tussen een basiscatalogus en een virtuele catalogus. In een basiscatalogus sla je informatie over producten op. In de virtuele catalogus kun je producten uit één of meer basiscatalogi verzamelen, waarbij je de prijs van producten kunt beïnvloeden door er rekenregels op toe te passen. Hierdoor kunnen meer (virtuele) catalogi ontstaan. Waarom zou je nu meer catalogi willen gebruiken? Welnu, als je actief bent in meer landen, dan kun je elke catalogus op een ander land richten, maar je kunt ook elke catalogus op een andere doelgroep richten of, als je met B2B bezig bent, voor iedere klantgroep een andere catalogus maken.

### Wat is Commerce Server?

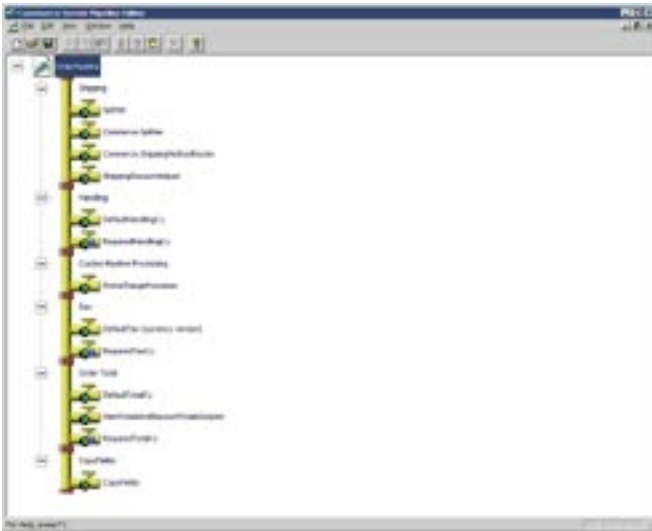
Microsoft Commerce Server bestaat uit een aantal componenten, die samen een goede basis vormen om een e-commerce site te bouwen. Alles wat je nodig hebt vind je in de doos. Er wordt zelfs een aantal voorbeeldsites meegeleverd die je, als je weinig eisen stelt aan de vormgeving van je winkel, zou kunnen gebruiken om je eerste winkel mee te starten.

Elk product in de productcatalogus kan van een bepaald producttype zijn. Dat is handig omdat de eigenschappen van bijvoorbeeld een spijkerbroek anders zijn dan die van een pak melk. Daarom kun je in Commerce Server meer producttypen aanmaken. Elk producttype heeft een aantal eigenschappen die je uiteraard ook zelf kunt instellen en elke eigenschap kan ook weer van een bepaald type zijn, bijvoorbeeld tekst, integer, money of multiple choice. Verder wordt onderscheid gemaakt tussen een los product en een productfamilie met haar varianten. Wat betekent dit precies? Een productfamilie is bijvoorbeeld een spijkerbroek. De varianten zijn de verschillende maten en kleuren waarin een spijkerbroek te verkrijgen is. Elke productfamilie kan dus één of meer varianten hebben. Eigenschappen kunnen zowel per familie als per variant worden opgeslagen.

Om cross-selling mogelijk te maken kun je per product de relatie en het soort relatie met andere producten vastleggen. Dit kun je goed gebruiken om niet alleen alternatieve of gelijkwaardige producten vast te leggen, maar ook om een bijverkoop vast te leggen (bijvoorbeeld een verzekering bij een mobiele telefoon). Elk product kan in één of meer categorieën zitten. Ook van een categorie kun je de eigenschappen zelf instellen en het is dus ook mogelijk om meer categorietypen aan te maken. Al deze mogelijkheden maken de productcatalogus zeer flexibel. Je kunt er dus de meeste producten wel in kwijt. Met de producten en categorieën is overigens nog wel een klein probleem. Commerce Server maakt voor elke eigenschap van een product- of categorietype een veld aan in de tabel waarin de producten en categorieën worden opgeslagen. Als je uitgebreide product-definities hebt en veel verschillende producttypen gebruikt in één catalogus, heb je kans dat je tegen de maximale lengte van een rij in SQL Server 2000 aanloopt.

## Winkelmandje en ordersysteem

Commerce Server biedt een winkelmandje met een aantal interessante mogelijkheden. Natuurlijk kun je er producten in stoppen en weer uithalen, maar dankzij de pipeline worden ook kortingsacties, verzendkosten, BTW en het totaal van het winkelmandje berekend. De pipeline, die overigens niet alleen door het winkelmandje gebruikt wordt, lijkt inderdaad op een pijplijn: je stopt er aan één kant iets in, er wordt een aantal bewerkingen op uitgevoerd en aan de ander kant komt er iets uit. De bewerkingen kunnen lopen van het controleren op de voorraad van de bestelde producten tot het opzoeken van de verzendkosten voor een order. Er wordt bij Commerce Server



Afbeelding 1. De Pipeline-editor

standaard een aantal pipeline-componenten meegeleverd, maar het is niet zo heel erg moeilijk om zelf een nieuwe component te schrijven. Dat kun je op een aantal manieren doen. De simpelste manier is via een stukje VBScript, zoals in codevoorbeeld 1 staat. Hierin wordt een aantal velden uit het winkelmandje naar de definitieve order gekopieerd.

Hiermee kun je makkelijk en snel een nieuwe component toevoegen, maar het is wel lastig om te debuggen. Bij echte high-performance sites kun je VBScript-componenten beter niet gebruiken. Een betere manier is een component in C# of VB.NET te schrijven. Dit is niet helemaal triviaal omdat je in feite een COM-component aan het schrijven bent, maar met de meegeleverde template is het goed te doen. Bovendien is de performance beter dan met een VBScript-component, ondanks dat er wat vertaling van COM naar .NET plaatsvindt bij de aanroep. De snelste mogelijkheid is natuurlijk een pipeline-component in C++ te schrijven. Na registratie van de nieuwe component kun je deze in de pipeline-editor toevoegen aan een bestaande pipeline of een heel nieuwe pipeline maken.

Iets waar het winkelmandje en de pipeline-standaard niet goed mee om kunnen gaan, zijn decimale hoeveelheden; bijvoorbeeld 1,5 liter verf. Hier zul je zelf een pipeline-component voor moeten schrijven waarin je eerst de hoeveelheid met 100 vermenigvuldigt om, nadat alle berekeningen zijn uitgevoerd, het subtotaal voor dat product weer door 100 te delen.

## Campagnes

Onder campagnes verstaat Commerce Server alles wat met direct mail, banners en kortingsacties te maken heeft. De mogelijkheden voor direct mail zijn niet zo uitgebreid, maar voldoende voor simpele toepassingen. De ondersteuning voor kortingsacties is wel uitgebreid. Zo kun je korting geven op specifieke producten, producten uit een bepaalde categorie of op een hele order. Ook kun je korting geven wanneer iemand ook een ander product koopt, of wanneer een product binnen een bepaalde tijdsperiode gekocht wordt (happy hour discounts). Een korting kan percentageel zijn of in de vorm van een bepaald bedrag. De korting kan toegespitst worden op de eigenschappen van een gebruiker. De pipeline berekent automatisch de juiste korting voor de producten die een klant in zijn winkelmandje heeft zitten. In de praktijk zijn de kortingsacties iets minder praktisch. Over het algemeen heeft een bedrijf al een eigen back-office waarin kortingsacties verwerkt zijn. Het is dan een hele toer om deze acties aan te laten sluiten op de acties in Commerce Server.

## Profile System

Het Profile System is de plaats om gegevens over klanten op te

### Option Explicit

```
Function MSCSExecute(config, orderform, context, flags)
    Dim objProfileService, user_id, objUserProfile, org_id, objOrgProfile

    MSCSExecute = 1

    'Copy some lineitem-level fields
    Dim item

    For Each item In orderform.value("items")
        item.value("cy_unit_price") = item.value("_cy_iadjust_regularprice")
        item.value("cy_lineitem_total") = item.value("_cy_oadjust_adjustedprice")
        item.value("description") = Mid(item.value("_product_description"),1,127)
        item.value("saved_product_name") = item.value("_product_Product Name")
        item.value("is_promotion_product") = item.value("IsPromotionProduct")
    Next

    'Copy some orderform-level fields
    orderform.value("saved_cy_oadjust_subtotal") =
        orderform.value("_cy_oadjust_subtotal")
    orderform.value("saved_cy_total_total") = orderform.value("_cy_total_total")

    ' Save shipping total
    orderform.value("cy_shipping_total") = orderform.value("_cy_shipping_total")

End Function
```

Codevoorbeeld 1. Zelf een nieuwe component schrijven

slaan. Dit onderdeel van Commerce Server is het beste te vergelijken met een database met wat extra faciliteiten. Op zich niet schokkend dus, maar die extra faciliteiten maken wel het verschil. Het Profile System is zo opgezet dat het onafhankelijk is van de bron van je gegevens. Dat werkt niet alleen voor 'tabellen', maar ook voor 'records'. Een record in het Profile System kan dus verrijkt worden met gegevens uit meer bronnen. Je kunt op die manier optimaal gebruik maken van gegevens uit bestaande bronnen (zoals NAW-gegevens uit een SQL Server-database) en deze aanvullen met gegevens die specifiek voor de website bedoeld zijn (zoals gebruikersnaam en wachtwoord uit Active Directory).

## BizDesk

Al deze zaken worden beheerd vanuit de BizDesk, een HTML-applicatie voor het beheer van sites die met Commerce Server gemaakt zijn. De BizDesk is volledig ontwikkeld in ASP en je krijgt alle broncode. Uitbreiden is dus mogelijk, maar omdat de BizDesk van een nogal ingewikkelde architectuur uitgaat, is het niet makkelijk. Wel simpel is het om het menu uit te breiden met zelf ontwikkelde applicaties.

## Datawarehouse en rapportages

Alle webserver-logs en gegevens over aankopen worden dagelijks verzameld en ingelezen in het datawarehouse. Het datawarehouse maakt gebruik van SQL Server Analysis Services om via OLAP allerlei rapporten te tonen. Die rapporten variëren van het aantal bezoekers per uur tot rapporten over het aantal verkopen en de meest bezochte productcategorieën. Voor de liefhebber is er de mogelijkheid om zelf rapporten te maken, maar daar is wel wat kennis van OLAP en cubes voor nodig.

## Predictor

Stel dat we nu - net als Amazon.com - een 'andere klanten kochten ook'-feature willen hebben. Hoe doen we dat? Commerce Server biedt hiervoor de predictor. Op basis van de het klik- en aankoopgedrag van klanten van een site kan de predictor uitspraken doen over welke overige producten interessant kunnen zijn bij een bepaald product. Ook kan de predictor

statistisch verantwoorde uitspraken doen over eigenschappen van een gebruiker die zijn gebaseerd op zijn aankoop- en klikgedrag. Om goed met de predictor om te kunnen gaan en om verantwoorde resultaten te krijgen, heb je wel enige kennis van statistieken nodig en moet het aantal bezoekers op je site redelijk hoog zijn.

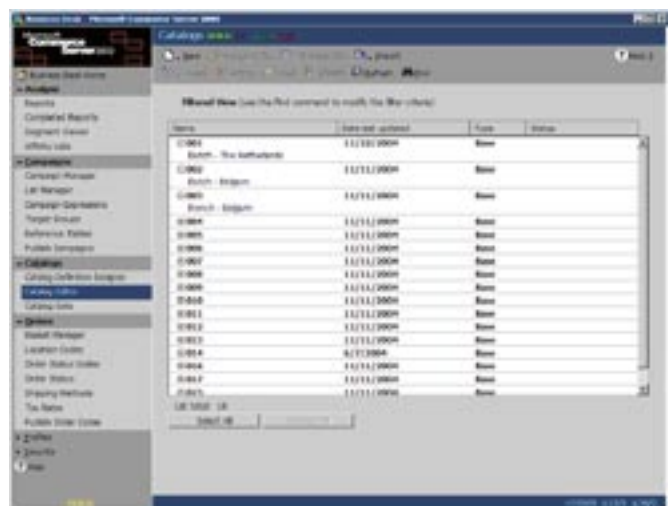
## Meertaligheid

Commerce Server ondersteunt meertaligheid, maar alleen in de productcatalogus. Je kunt van een producteigenschap instellen dat deze in verscheidene talen ingevoerd kan worden. Waar je dan wel op moet letten is dat deze eigenschappen niet meegenomen kunnen worden in een free text search index. Dit maakt dat meertaligheid een stuk minder aantrekkelijk wordt. De productcatalogus is dus het enige onderdeel van Commerce Server dat meer talen ondersteunt. Campagnes en kortingsacties kunnen niet omgaan met meer talen, wat betekent dat als je kortingsacties voor meer talen wilt maken, je deze ook meer keren aan moet maken. Kortom, het volledig doorvoeren van meertaligheid vergt veel handmatige aanpassingen en uitbreidingen.

## Commerce Server en .NET

Commerce Server 2002 is een van de eerste producten die echt .NET enabled zijn, en dat is te merken. De voornaamste manier om Commerce Server aan te spreken in deze versie is nog steeds de COM-interface. Die interface is zeer uitgebreid en het biedt de mogelijkheid om alle onderdelen van Commerce Server te besturen. De .NET-interface biedt slechts een gedeelte van de functionaliteit die de COM-interface biedt, maar dit zijn wel de onderdelen die je het meest zult gebruiken. Functies die je normaal niet in een site gebruikt, zoals het toevoegen van producten aan de catalogus of het gebruiken van de predictor, zijn niet meegenomen in de .NET-interface. Als je hier dus iets mee wilt doen, ben je aangewezen op de Primary Interop Assemblies. De .NET-interface richt zich primair op het gebruik in ASP.NET. Daarbuiten is het haast niet bruikbaar, dus een losse WinForms-applicatie gebruiken om gegevens over producten op te halen is via de .NET interface niet of nauwelijks te realiseren. De COM-interface kan hier echter wel voor gebruikt worden. De .NET-interface maakt goed gebruik van de mogelijkheden van het .NET Framework en is een verademing in vergelijking met de 'oude' COM-interface. Daarnaast is er goed nagedacht over hoe de implementatie van een e-commerce website makkelijker te maken is voor ontwikkelaars. Codevoorbeeld 2 laat het ophalen van een productcatalogus met de COM-interface zien.

Er wordt eerst een catalogmanager gemaakt die vervolgens wordt geïnitieerd met een connectiestring waarna de catalogus opgehaald kan worden. Overigens is die connectiestring



Afbeelding 2. BizDesk: de Catalog-editor

eerst via een ander stukje code uit een configuratieobject van Commerce Server gehaald. Vergelijk dit eens met de .NET-versie; zie codevoorbeeld 3.

Uiteraard wordt onder water bijna hetzelfde gedaan, maar doordat veel beter gebruik gemaakt wordt van bijvoorbeeld configuratie via web.config en er impliciet van alles gebeurt, wordt het leven van de ontwikkelaar wel een stuk eenvoudiger gemaakt. Commerce Server biedt ook functionaliteit die overlapt met functionaliteit uit ASP.NET. Zo heeft Commerce Server objecten die speciaal bedoeld zijn voor het cachen van allerlei gegevens. ASP.NET beschikt op dit gebied over meer en soms ook over betere mogelijkheden. Denk bijvoorbeeld maar aan page-caching. In dit soort gevallen is het dan ook beter om gebruik te maken van de mogelijkheden van ASP.NET. Het leidt naar een beter groeipad voor toekomstige versies van Commerce Server en maakt het ontwikkelen een stuk eenvoudiger.

De integratie met Visual Studio .NET is erg goed. Ontwikkelen kun je doen in je vertrouwde omgeving. Daar waar je met ASP nog wel eens met meer ontwikkelaars op dezelfde server ontwikkelt, kan elke ontwikkelaar nu in zijn eigen omgeving aan de gang zonder andere ontwikkelaars in de weg te zitten. Vanuit Visual Studio is een aantal onderdelen van de BizDesk, de pipeline-editor en de Commerce Server-manager makkelijk bereikbaar. Er is ook een aparte tool om eenvoudig databinding-expressies voor eigenschappen van een product of categorie aan te maken.

Het grootste voordeel van het gebruik van .NET schuilt echter in de type-checking die de compiler uitvoert. Dit maakt de ASP.NET-sites veel robuuster dan de ASP-tegenhangers. Door bovendien gebruik te maken van de mogelijkheden van objectoriëntatie is onderhoud en uitbreiding veel makkelijker.

## Content management

Wanneer je met de bouw van een e-commerce site start merk je dat zo'n site eigenlijk uit twee gedeeltes bestaat. De kern bestaat uit allerlei functionaliteit die met de producten te maken heeft die je wilt verkopen, zoals de productcatalogus en informatie over klanten. Maar daar omheen heb je vaak veel functionaliteit die daar niets mee te maken heeft: informatie over openingstijden, verzending, klantenservice, reviews over de producten die je verkoopt enzovoort. Al die informatie moet ook onderhoudbaar zijn. Daarvoor is een contentmanagementsysteem nodig. Uiteraard ben je vrij in de keuze of en welk contentmanagementsysteem je gebruikt. Microsoft Content Management Server beschikt over een speciale koppeling met Commerce Server, waardoor je niet alleen je 'normale' content kunt beheren, maar ook het beheer van productinformatie direct op de website kunt uitvoeren.

## Productbeheer

Ik ben nog geen klant tegengekomen die niet zelf al een systeem heeft draaien met gegevens over zijn producten.

```
Dim mscsCatalogManager
Dim oCatalog

Set mscsCatalogManager = Server.CreateObject("Commerce.CatalogManager")
Call mscsCatalogManager.Initialize(dictConfig.s_CatalogConnectionString,
ADO_CONNECTION_STRING)
```

```
Set oCatalog = mscsCatalogManager.GetCatalog(sCatalogName)
```

Codevoorbeeld 2. Ophalen van de productcatalogus met de COM-interface

```
Catalog.ProductCatalog catalog = CommerceContext.Current.CatalogSystem.
GetCatalog( catalogName );
```

Codevoorbeeld 3. Dezelfde actie in de .NET-versie



Afbeelding 3. De Databinding Expression Builder

Omdat het verstandig is om gegevens op één plaats te beheren, is structureel beheer van producten via de BizDesk dan ook meestal niet de beste oplossing. Dit betekent dat gegevens uitgewisseld moeten worden tussen de back-office en de website. Om dat mogelijk te maken heeft Microsoft een XML-schema gedefinieerd voor de productcatalogus. Via de BizDesk kun je een XML-bestand inlezen dat aan dit schema voldoet om een productcatalogus in te lezen. Het nadeel van deze methode is dat je altijd een volledige catalogus in moet lezen. Er is geen mogelijkheid om aan te geven dat je één eigenschap van een aantal producten wilt inlezen. Daardoor is het in veel gevallen niet de ideale oplossing. De conclusie kan dan zijn dat je zelf een importcomponent schrijft, waarbij je de Commerce Server API producten laat toevoegen. Helaas stuit dat op het praktische bezwaar dat de API bij producten toevoegen erg langzaam is, waardoor het importeren voor een beetje productcatalogus al snel te lang duurt. Uit ervaring weet ik dat een combinatie van de API en een zelfontwikkelde koppeling vaak het beste resultaat oplevert.

### Orderverwerking

Commerce Server biedt geen sterke tools om orders te beheren. Dit is logisch want de orderverwerking is in ieder bedrijf anders geregeld. Je kunt dit op twee manieren oplossen. Je kunt de BizDesk aanpassen en hier faciliteiten maken om orders te verwerken. Dit is een goede optie voor kleine bedrijven die zelf geen back-office hebben. Het komt dan ook veel vaker voor dat orders vanuit de website naar de back-office verstuurd worden. Microsoft BizTalk Server is uiteraard een zeer goede kandidaat om dit probleem op te lossen. Door gebruik te maken van de mogelijkheid om orders in XML-formaat te exporteren is de integratie met BizTalk Server snel te realiseren.

### Wat brengt de toekomst?

Microsoft heeft aangekondigd dat begin 2006 een nieuwe versie van Commerce Server uitgebracht wordt. Op dit moment is op de website van Microsoft alleen wat algemene informatie beschikbaar over deze nieuwe release. De verwachting is dat de nieuwe versie gebruik zal maken van de kracht van andere pro-

ducten. SQL Server Reporting Services zal bijvoorbeeld gebruikt worden voor de rapportages, waardoor het veel makkelijker wordt om zelf nieuwe rapportages toe te voegen. Deze nieuwe versie zal volledig .NET enabled zijn in plaats van de gedeeltelijke .NET-implementatie die er nu is. Hierbij gaan webservices een belangrijke rol spelen. Iets waar je nu al wat van kunt zien als je feature pack 1 installeert. Tot de nieuwe versie uitkomt, biedt Commerce Server 2002 een prima basis voor een kleine of grote webwinkel.

Erwin Werkman is werkzaam bij The Vision Web, [www.thevisionweb.nl](http://www.thevisionweb.nl)

#### Nuttige internetadressen

##### Commerce Server 2002 Trial:

<http://www.microsoft.com/commerceserver/evaluation/trial/default.asp>

##### Commerce Server 2002 Feature Pack 1 Trial:

<http://www.microsoft.com/downloads/details.aspx?>

FamilyId=B2DFE80C-C6AB-450C-ACA6-3FE99B36A86C&displaylang=en

<http://www.microsoft.com/commerceserver/techinfo/development/default.asp>

<http://msdn.microsoft.com/library/en-us/dnanchor/html/commerceserver.asp>

Commerce Server 2006: <http://www.microsoft.com/commerceserver/2006preview/>

## Abonnee service

### Nog geen abonnee?

Geef je op voor het .NET Magazine en ontvang gratis alle volgende nummers.

#### Voor Nederland

Schrijf je in op [www.microsoft.nl/netmagazine](http://www.microsoft.nl/netmagazine)

#### Voor België

Schrijf je in op [www.microsoft.com/belux/nl/msdn/community/magazine.mspix](http://www.microsoft.com/belux/nl/msdn/community/magazine.mspix)

### Nummers gemist?

Heb je niet alle nummers van het .NET Magazine? Je kunt de nummers 3, 4, 5, 6, 7 en 8 nog nabestellen. Stuur een e-mail met jouw naam en adresgegevens en welke edities je wilt ontvangen naar [abonnementen@msdevmag.com](mailto:abonnementen@msdevmag.com). Dan sturen wij ze kosteloos naar je toe.

### Verhuisd?

Het .NET Magazine op je nieuwe adres ontvangen? Of thuis? Wijzig jouw adresgegevens op [www.microsoft.nl/netmagazine](http://www.microsoft.nl/netmagazine).

### Overige vragen?

Je kunt alle artikelen in PDF-formaat en de codevoorbeelden downloaden van [www.microsoft.nl/netmagazine](http://www.microsoft.nl/netmagazine).

### Suggesties?

Heb je een leuk idee voor een onderwerp, of een suggestie om het blad te verbeteren? Laat het ons weten op [redactie@msdevmag.com](mailto:redactie@msdevmag.com).