

# E-Commerce met Commerce Server

## ONTWIKKEL BREED INZETBARE OPLOSSING MET SHAREPOINT

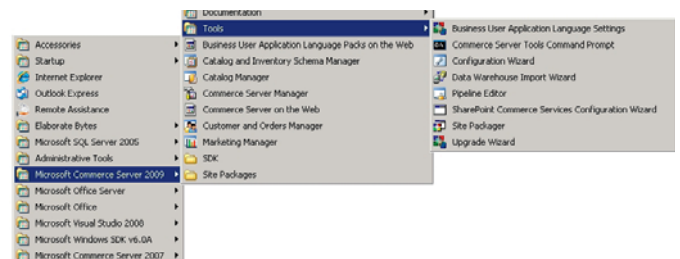
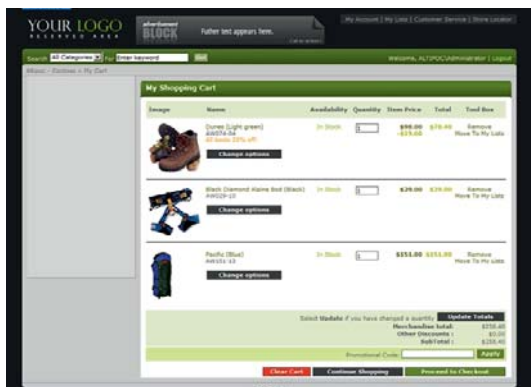
Nick Boumans

Momenteel zijn meer dan 20.000 Nederlandse webshops actief. Dit aantal stijgt nog steeds fors. Aangezien SharePoint vaker wordt ingezet als internet-oplossing, is het logisch dat meer vraag ontstaat naar een E-Commerce-oplossing binnen SharePoint. Naast een maatwerkoplossing voor E-Commerce binnen SharePoint, is Microsoft Commerce Server een mogelijkheid bij het realiseren van een SharePoint E-Commerce-oplossing.

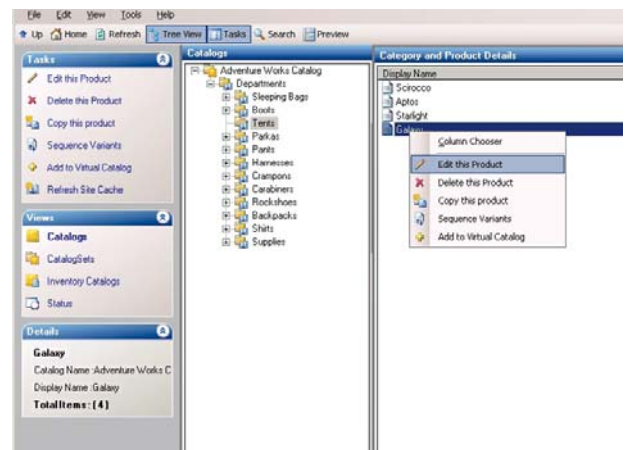
Commerce Server 2009 is een Microsoft-product voor het bouwen van E-Commerce systemen, gebruikmakend van .NET-technologie, waaronder webservices en een abstracte business layer, bekend als de Commerce Server Foundation. Dit product vindt zijn oorsprong in 2000. De huidige versie (2009) is gelanceerd in maart van dit jaar. Commerce Server levert zelf geen Out-of-the Box (OOTB) oplossing voor een E-Commerce portal. Al komt de laatste versie erg dicht in de buurt. SharePoint Commerce Services verzorgen de integratie tussen SharePoint en Commerce Server. Verder is het platform zelf een volledig Service Oriëntated (SOA) enabled platform met daarin BizTalk adapters.

### Voordeel

Het voordeel van Commerce Server is dat men na het doorlopen van de SharePoint Commerce Services wizard de beschikking heeft over een aantal SharePoint WebParts en eventueel een Default Site (Figuur 1). Deze WebParts variëren van een winkelmandje tot een myAccount WebPart. Veel van deze WebParts zijn te customizen door het bijbehorende XSLT aan te passen. In de meegeleverde CommerceSharePointExtensibilityKit zijn tevens de broncodes beschikbaar van deze WebParts en UserControl, welke hulp kunnen bieden bij maatwerk.



De Default SharePoint Site is een volledige voorbeeldsite binnen SharePoint, gebruikmakend van de Commerce Server Features. In principe kunnen we door enkel een eigen catalogus te maken (configureren) en deze te koppelen aan de default site in slechts enkele stappen een eigen E-Commerce oplossing neerzetten. De meegeleverde tools (Figuur 2), die verbinding maken via webservices, kunnen hierbij helpen. Met de 'Catalog and Inventory Schema Manager' tool kunnen schemadefinities gemaakt worden voor categorieën en producten. Hierin worden dus de eigenschappen vastgelegd van producten en categorieën in de catalogus en de catalogus zelf. Hierna kan een catalogus aangemaakt en gevuld worden met de 'Catalog Manager' (Figuur 3).



De Commerce Server SharePoint Site Customizen kan door de masterpage en bijbehorende CSS te customizen. Tevens maken veel WebParts gebruik van een template die gebruik maakt van XSL. Deze templates staan in een Document Library 'Commerce Server Templates' genaamd. Hierin kunnen dus ook maatwerk templates geplaatst worden, die vervolgens gebruikt kunnen worden door in de WebPart properties van een betreffend WebPart aan te geven, dat een betreffende template gebruikt kan worden. Op deze manier zijn de meest voor de hand liggende aanpassingen aan WebParts te verwezenlijken. Hierbij kan men denken aan de volgorde van kolommen in bijvoorbeeld het WebPart dat een winkelmandje weergeeft.

Development met de Commerce Server API maakt het mogelijk om door middel van .NET-Code aanpassingen te verrichten aan bijvoorbeeld de SharePoint E-Commerce site. Door een aantal .dll's toe te voegen als referentie aan een Visual Studio Project wordt het mogelijk gebruik te maken van deze API's. Veelgebruikte dll's zijn:

```
Microsoft.CommerceServer.Runtime.dll
Microsoft.CommerceServer.Catalog.dll
Microsoft.Commerce.Portal.Common.dll
```

Deze .dll's zijn terug te vinden in de CommerceSharePointExtensibilityKit die na installatie op de volgende locatie te vinden is: C:\Program Files\Microsoft Commerce Server 2007\Microsoft

Commerce Server 2009\Sdk\Samples\CommerceSharePointExtensibilityKit.zip. Om de dll's beschikbaar te krijgen zal men het project uit deze kit eerst moeten bouwen, nadat deze gesigned is met een eigen key. Het vertrekpunt van development is de 'CommerceContext'-class die terug te vinden is in de CommerceServer.Runtime.dll. Deze class maakt het onder andere mogelijk om catalogi te managen. Voor de beste performance creëert men slechts één instantie van dit object die men hergebruikt tijdens de lifetime van de applicatie.

Development met de catalogus is een van de meest voorkomende developmentvraagstukken. In de praktijk resulteert dit in wensen als: een WebPart binnen SharePoint waarin producten en categorieën zijn op te vragen of te managen. Los van het feit dat dit met de Commerce Server tools kan, of door een door Microsoft beschikbaar gesteld Silverlight WebPart, voldoen deze oplossingen vaak niet aan de eisen en wenst men hier een eigen draai aan te geven.

Het toevoegen van een product aan een catalogus kan gezien worden als een 'Hello World'-developmentvoorbeeld voor Commerce Server. Zoals eerder aangegeven is het vertrekpunt het verkrijgen van de CommerceContext en daarmee de catalogus waarop men bewerkingen wenst uit te voeren. In Codevoorbeeld 1 is te zien hoe het mogelijk is om een product toe te voegen aan een catalogus. De laatste coderegel is wellicht de meeste belangrijke, aangezien zonder het aanroepen van de item.Save() methode wijzigingen niet opgeslagen zullen worden.

(Advertentie)



Materialise Dental is wereldwijd marktleider in '3D Digital Dentistry'. Daarbij bieden we baanbrekende producten en diensten aan implantologen, orthodonten en hun patiënten. Materialise Dental investeert in innovatieve softwareproducten en diensten voor allerlei dentale toepassingen. Door de brede waaier van technologieën bieden wij unieke en boeiende kansen aan onze werknemers. Wij zijn momenteel op zoek naar kandidaten voor de volgende functies (m/v):

## Release & Deployment Manager

Als Release Manager ben je verantwoordelijk voor de integratie van onze hoogtechnologische producten en diensten binnen de groeiende infrastructuur van onze internationale onderneming. Je werkt nauw samen met onze IT- en softwaredevelopment-afdelingen om het release proces verder te verfijnen en het daarna zelfstandig en nauwgezet toe te passen. Je zorgt ervoor dat nieuwe ontwikkelingen vlot en rimpelloos hun weg vinden naar onze gebruikers en klanten.

Aanbod: Wij bieden je een dynamische en uitdagende job in een professionele werkomgeving. Je kan rekenen op een vliegende start en begeleiding bij de uitbouw van je carrière.

## .Net Ontwikkelaar

Als .Net ontwikkelaar ben je mee verantwoordelijk voor de analyse en ontwikkeling van producten. Software ontwikkelen is voor jou een proces waarin verschillende partijen met eigen specialisaties samenwerken en uitdagingen aangaan om commercieel waardevolle en innovatieve producten aan te bieden. Je wil in een omgeving werken waarin jouw bijdrage hieraan een zichtbaar verschil maakt!

VOOR MEER INFORMATIE:

[www.materialise.com/careers](http://www.materialise.com/careers)

Materialise Dental, Technologielaan 15, 3001 Leuven.

```
// You have to add a reference to the Microsoft.CommerceServer.Catalog.dll
// (CommerceSharePointExtensibilityKit) and the
// Microsoft.CommerceServer.Runtime.dll
BaseCatalog baseCatalog = (BaseCatalog)CommerceContext.Current.CatalogSystem.CatalogContext.GetCatalog("Adventure Works Catalog");
Product item = baseCatalog.CreateProduct("Tents", "Iglo", 30m, "Tents");
item.DisplayName = "Iglo";
item["Description"] = "3mm Tent";
item.Save();
```

#### CODEVOORBEELD 1: HET TOEVOEGEN VAN EEN PRODUCT AAN EEN CATEGORIE IN EEN CATALOGUS.

Een categorie definitie beschrijft de categorie in de catalogus. Wezenlijk is dit een collectie van eigenschappen (properties) die specificeren hoe informatie wordt opgeslagen in een categorie. Bij het creëren van een categorie definitie wordt deze definitie toegevoegd aan het catalog schema.

Om een categorie definitie te maken:

## Zonder installeren en configureren direct aan het werk met de downloadbare Microsoft Commerce Server VPC

- Creëert men een categorie definitie door gebruikt te maken van de CreateDefinition methode van het CatalogContext object.
- Creëert men properties door gebruik te maken van de AddProperty methode van het CatalogDefinition object.
- CatalogDefinition.Save() voor het opslaan van de definitie.

Codevoorbeeld 2 creëert een catalogus definitie "Movie Category" door het aanmaken van een CatalogDefinition object.

```
public static void CreateCategoryDefinition(CatalogContext context)
{
    // Create a category definition for categories in a DVD catalog.
    CatalogDefinition categoryDefinition = context.CreateDefinition("Movie Category", CatalogDefinitionType.CategoryDefinition);
    categoryDefinition.Description = "Categories for DVDs";
    // Create a description property for this category.
    CatalogProperty description = context.CreateProperty("Description", CatalogDataType.String, 500);
    // Add the property to the category definition.
    categoryDefinition.AddProperty(description.Name, DefinitionPropertyType.NormalProperty);
    // Save the definition.
    categoryDefinition.Save();
}
```

#### CODEVOORBEELD 2: HET MAKEN VAN EEN CATEGORIE DEFINITIE.

Het toevoegen en verwijderen van een categorie vormt een logisch vervolg op het creëren van een categorie definitie. Ook bij deze twee bewerkingen dient men eerst de catalogus op te halen indien dit nog niet gebeurd is. Deze catalogus kent respectievelijk de methoden CreateCategory en DeleteCategory. Bij het toevoegen van een categorie dient een categorie definitie opgegeven te worden. Codevoorbeeld 3 en 4 laten zien hoe een categorie toegevoegd of verwijderd kan worden door gebruik te maken van de Catalog API. Bij het verwijderen van een categorie zorgt de methode 'Pur-

geDeletedItems' ervoor dat verwijderde items ook daadwerkelijk worden verwijderd uit de Catalog Database.

```
BaseCatalog baseCatalog = (BaseCatalog)CommerceContext.Current.CatalogSystem.CatalogContext.GetCatalog("Movie Catalog");
// Add a category
Category category = baseCatalog.CreateCategory("MovieCategory", "Adventure");
category.DisplayName = "Adventure";
category["Description"] = "Adventure Movies";
category.Save();
```

#### CODEVOORBEELD 3: HET TOEVOEGEN VAN EEN CATEGORIE.

```
ProductCatalog productCatalog = (ProductCatalog)CommerceContext.Current.CatalogSystem.CatalogContext.GetCatalog("Movie Catalog");

// Delete the category
productCatalog.DeleteCategory("MovieCategory");
// Remove the deleted items from the catalog database
productCatalog.PurgeDeletedItems();
```

#### CODEVOORBEELD 4: HET VERWIJDEREN VAN EEN CATEGORIE.

### Newbie met Commerce Server?

Indien u zich als SharePoint ontwikkelaar nog niet verdiept heeft in Commerce Server is het zeker de moeite waard om hiermee te experimenteren. Indien u geen tijd wilt spenderen aan het installeren en configureren alvorens men kan experimenteren is er een Microsoft Commerce Server VPC beschikbaar die te downloaden is van de Microsoft Connect Site.



#### Links

Blog Nick Boumans: <http://www.sharepointdevelopment.nl>

Commerce Server 2009 VPC: <https://connect.microsoft.com/Downloads/DownloadDetails.aspx?SiteID=643&DownloadID=16926>

MSDN Commerce Server 2009 Forum: <http://social.msdn.microsoft.com/Forums/en-US/commsserver2009/threads>

Inline property editor webpart: <http://www.microsoft.com/downloads/details.aspx?FamilyID=fce87672-90cf-4ead-b100-6bbc24d96f66&displaylang=en>

**Nick Boumans**, is SharePoint specialist en Microsoft Certified Trainer bij Giraffe IT.

Hij blogt op <http://www.SharePointDevelopment.nl>.

