

# Zoeken met Search Server 2008 Express

EENVOUDIG IN TE ZETTEN ALS OPLOSSING OP WEBSITES

Reg Ernst en Vincent Middelkoop

Microsoft Search Server 2008 Express is een gratis zoekstelsel dat je op eenvoudige wijze kunt inzetten als zoekoplossing voor elke .net website. In dit artikel beschrijven we aan de hand van enkele praktijkvoorbeelden hoe je zoekopdrachten kunt verfijnen met metadata en hoe je vanuit een .net webapplicatie in Search Server Content Sources en Search Scopes kunt genereren. Voor ontwikkelaars geven we codevoorbeelden voor het gebruik van het op SharePoint gebaseerde Search Server Object Model en beschrijven we vervolgens de stappen om deze maatwerk code naar de Search Server omgeving te deployen.

**Search Server 2008** is een uitbreiding op SharePoint services en biedt een zoekoplossing die heel wat verschillende soorten opslagsystemen kan doorzoeken en indexeren. Je kunt informatie vinden opgeslagen in websites, SharePoint sites, gedeelde netwerkmappen en publieke mappen van Exchange. Standaard worden alle bekende Office formaten doorzocht. Door het toevoegen van extra 'Filters' kunnen ook andere bestandstypen worden doorzocht. Voorbeelden hiervan zijn de TIFF iFilter, die een OCR-technologie toevoegt, waardoor gescande documenten en faxen automatisch geïndexeerd kunnen worden, en de gratis PDF iFilter van Adobe, waarmee PDF's doorzocht kunnen worden. Search Server 2008 biedt uitgebreide zoekfunctionaliteit, rapportage- en beheermogelijkheden en is bovendien schaalbaar en uitbreidbaar.

Naast de gratis 'Express' versie, bestaat er ook een betaalde Search Server 2008 versie die op meerdere servers kan draaien en daarmee geschikt is voor grote organisaties. Wij hebben de Express versie gebruikt die bedoeld is voor een hoeveelheid informatie waarbij één fysieke zoekserver voldoende is. Search Server kent geen limieten op de omvang van documenten. Wanneer je bij de installatie van Search Server de gratis SQL Server Express mee installeert en gebruikt, heb je wel te maken met een beperking van 4 GB op de omvang van de databasefile. Wanneer je bij installatie een bestaande SQL Server gebruikt, heb je deze beperking niet.

Belangrijk voor een zoekstelsel is het doorzoeken van sites en servers op zoek naar informatie om te indexeren, ofwel crawlen. In Search Server kun je voor het definiëren van een crawl-opdrachten 'Content Sources' aanmaken. Een Content Source is een inhoudsbron waar data opgeslagen wordt, zoals een website, een

gedeelde netwerkmap of een online database. Per opgegeven Content Source in Search Server kun je instellen wanneer en hoe vaak er gecrawld moet worden, hoeveel niveaus diep, volledige of incrementele crawl, etcetera.

Voor het bepalen van bereik van zoekopdrachten in de Search Server database kun je 'Search Scopes' aanmaken. Per Search Scope kun je middels 'Scope Rules' instellen wat al dan niet in scope is, zoals welke Content Source, website adres of metadata waarde. Voor het koppelen van Search Server 2008 Express als zoekoplossing aan je webapplicatie biedt Search Server standaard een SOAP-webservice. De URL van deze webservice voor zoekopdrachten is `http://<NaamServer>/_vti_bin/search.asmx`. In de webservice request (de zoekopdracht) geef je o.a. de Search Scope en zoektermen op. De XML-response (het zoekresultaat) kun je vervolgens als DataSet verwerken in je webapplicatie.

## Zoekopdracht verfijnen met MetaData

Er kunnen ook Search Scopes aangemaakt worden aan de hand van 'metatag' waardes. Metatags zijn standaard HTML-tags die een bepaalde eigenschap aan een pagina kunnen meegeven. Door het gebruik van metatags in combinatie met Search Server kan er gefilterd worden op specifieke eigenschappen van een pagina. In een Sitecore CMS project waar filtering en verfijning een grote rol speelde hebben wij deze functionaliteit ingezet. Om dit succesvol toe te passen moeten er een aantal stappen gevolgd worden. Ten eerste moeten er dynamische metatags op de pagina's geplaatst worden. Denk hierbij bijvoorbeeld aan een paginatype, rubriek en filtereigenschappen voor een bepaalde pagina. De metatags krijgen vervolgens een waarde mee uit het CMS.

Tevens hebben wij het 'Id' van een CMS-item meegenomen. Op basis hiervan is het mogelijk om eventueel de CMS context te gebruiken. In de code kunnen dan alle velden en relaties opgehaald worden.

Voorbeeld van dynamisch gegenereerde Metatags binnen een HTML pagina:

```
<meta name="Provinces" content="Gelderland,Utrecht" />
<meta name="Section" content="Nieuws" />
<meta name="Nieuwstype" content="Regionaal, Met _ afbeelding" />
<meta name="Date" content="2009-11-02" scheme="YYYY-MM-DD" />
<meta name="PageId" content="5B8FF542-3B07-4647-BE30-134AA4BC53D2"/>
```

Als de metatags aanwezig zijn moet er een crawl gestart worden. Vervolgens slaat Search Server alle metatags op die tijdens een crawl langskomen. Deze zijn vervolgens te benaderen in de Search Server beheeromgeving onder de naam 'crawled properties'. Om deze daadwerkelijk te gebruiken moeten ze gekoppeld worden aan een zogenaamde 'managed property'. Met een Search Server query kunnen dan bijvoorbeeld alle pagina's uit de rubriek 'Nieuws' met als type 'Regionaal' en als type 'Met afbeelding' worden opgehaald. Vervolgens hebben wij ervoor gezorgd dat binnen de resultaten van deze Search Scope nog verder verfijnd kan worden. Dit gaat aan de hand van DataTable select statements.

Voorbeeld dynamische query voor een gefilterde DataSet:

```
<QueryText language="en-US" type="MSSQLFT">
SELECT Title, Provinces, Section, Nieuwstype, Date, PageId
FROM SCOPE()
WHERE "scope"='Nieuws' AND
CONTAINS (Nieuwstype, 'Regionaal')
AND CONTAINS (Nieuwstype, 'Met afbeelding')
AND CONTAINS (Provinces, 'Gelderland')
ORDER BY Rank DESC --
</QueryText>
```

In dit specifieke project wordt het geheel gepresenteerd als 'facenavigatie' ofwel als kolom met een overzicht van meerdere filtermogelijkheden waar een bezoeker op kan klikken. Tevens wordt het aantal resultaten achter deze mogelijkheden geplaatst, zodat een bezoeker weet wat hij/zij kan verwachten. Als er geen resultaten zijn voor een specifieke filter actie dan wordt het filter verborgen. In de filterkolom kunnen actieve filtermogelijkheden ook weer uitgezet worden. Daarnaast is er een filter kruimelpad bovenaan de pagina om aan te geven waarop gefilterd is.

## Genereren van Content Sources en Search Scopes via een webservice

In een ander project hebben wij Search Server 2008 Express als zoekoplossing ingezet voor een ASP.NET webapplicatie voor het genereren van websites. Hierbij moesten we rekening houden met de eis dat binnen gegenereerde websites afzonderlijk gezocht moest kunnen worden. We hadden bedacht om in Search Server per gegenereerde website een Content Source aan te maken om deze website te kunnen crawlen en een Search Scope om het zoekbereik op deze website te bepalen. In de beheerschermen van Search Server is dit natuurlijk mogelijk, maar voor het opnemen van deze stappen in het automatische proces van websites genereren konden we in Search Server geen 'out-of-the-box' webservice vinden. Als oplossing hebben we met behulp van het op SharePoint gebaseerde Search Server Object Model een webservice gebouwd en deze als uitbreiding gedeployed naar de Search Server omgeving.

## Codevoorbeeld Search Server Object Model

```
using Microsoft.SharePoint;
using Microsoft.Office.Server.Search.Administration;

//need the site collection to modify settings
using (SPSite spSite = new SPSite (searchServerUrl))
{
    // get the SP Site search server instance
    SearchContext searchContext = SearchContext.GetContext (spSite);

    // get content sources
    Content content = new Content (SearchContext);
    ContentSourceCollection ContentSourceCollection = content.ContentSources;

    //create the new content source
    ContentSource contentSource = ContentSourceCollection.Create(typeof(WebContentSource), contentSourceName);

    //update startaddress of content source
    if (startAddress.LastIndexOf("/") >= 0)
    {
        string[] startAddressArray = startAddress.Split (char.Pars("|"));
        foreach (string startAddressItem in startAddressArray)
        {
            contentSource.StartAddress.Add(new Uri(startAddressItem));
        }
    }
    else
    {
        contentSource.StartAddresses.Add(new Uri(startAddress));
    }
    contentSource.Update();

    // start full crawl
    contentSource.StartFullCrawl();
}
```

## Stappen om een custom webservice te deployen naar een Search Server-omgeving

1. Maak een strong name voor de assembly met behulp van bijvoorbeeld SN.exe.

```
Neem de strong name op als page directive in de markup van de webservice (asmx). Voorbeeld:
<%@ Webservice Language="C#" CodeBehind="Service1.aspx.cs"
Class="Custom.Search.Service1"%>
<%@ Assembly Name="Custom.Search, Version=1.0.0.0, Culture=neutral,
PublicKeyToken=a16feb6896306e77"%>
```

2. Creëer een disco en wsdl file voor de webservice (disco.exe).
  - Service1.aspx
  - Service1disco.aspx
  - Service1wsdl.aspx
3. Kopieer de webservice bestanden (asmx, disco en wsdl) naar de iis webservice virtual van Search Server (C:\Program Files\Common Files\Microsoft Shared\web server extensions\12\ISAPI).
4. Neem de webservice dll op in de GAC van Search Server (C:\WINDOWS\assembly).
5. Reset IIS.

Herhaal stap 3 t/m 6 voor iedere deploy-ronde.



.....  
**Reg Ernst en Vincent van Middendorp**, zijn beide (Senior) Software Ontwikkelaar bij Evident Interactive.