



SharePoint® eMagazine

**Web Content Management:
the SharePoint Server 2010 way**

Themes in SharePoint 2010

**Suggestions for Social and Search
in SharePoint 2010**

**Community Kit for SharePoint:
Development Tools Edition**

**How to manage your Feature Lifecycle
using Feature Versioning**

Architecting Service Applications

**Incorporating External Data
using WCF**



Contents

Let us know	3
Web Content Management the SharePoint Server 2010 way	4
Themes in SharePoint 2010	12
Suggestions for Social and Search in SharePoint 2010	20
Community Kit for SharePoint: Development Tools Edition	26
How to manage your Feature Lifecycle using Feature Versioning	33
About DIWUG	41
Architecting Service Applications	45
Incorporating External Data in SharePoint 2010 using Windows Communication Foundation (WCF)	52
About the authors	62
SharePoint Comic	64



Colofon

DIWUG SharePoint eMagazine

Nr. 2, May 2010

Publisher:

Dutch Information Worker UserGroup
Second edition, May 2010

Editors:

Marianne van Wanrooij (lead)
marianne@information-worker.org
Mirjam van Olst
mirjam.vanolst@macaw.nl

Special thanks to:

Hannah Swain (reviewer)
and all the authors and sponsors!

Design and layout:

Barth Sluyters
<http://home.12move.nl/barthsluyters/>

©2010. All rights reserved. No part of this magazine may be reproduced in any way without prior written permission of Diwug or the author. All trademarks mentioned in this magazine are the property of their respective owners.



Let us know

Editor's note

With over 8700 downloads the first edition of the DIWUG SharePoint eMagazine was a big success. We received a lot of positive responses and great tips from the community. One of the most asked questions was if there are more magazines coming. After a short deliberation we decided to publish this downloadable magazine on a regular basis. We're going to publish the DIWUG SharePoint eMagazine 3 to 4 times a year. We want to offer you articles for developers, IT-pro's and end/power users. Of course we can't do this without the help of authors and sponsors! And since we're on the subject... we can always use articles and sponsors for our next issues. So let us know if you're interested to write an article or want to place an advertorial or advertisement.

The decision to publish this eMagazine made us aware we need to change our website as well. We're working on a SharePoint 2010 environment where you can download all of the issues and register for alerts if you like to stay up to date for new issues. We're planning to have that finished before the 3rd issue will be available.

With SharePoint 2010 on the market now for the public you probably are looking for a way to upgrade the features from 2007 to 2010. Read the article written by Anders Rask for tips and tricks! If you're interested in the WCM feature improvements the article of Waldek Mastyskarz would be of interest to you. Also the new Theming architecture by Wictor Wilen is a 'must read' for WCM developers and consultants. Wesley Hackett gives us an introduction to the CKS for developers. Andries de Haan gives us more information on Suggestions and last but not least Ragavan Sreeni gives us a step by step sample on how to create an External Content Type using WCF.

I hope you'll enjoy this issue of the DIWUG SharePoint eMagazine. And don't forget to give us your feedback!

Marianne van Wanrooij
lead editor
DIWUG SharePoint eMagazine



This issue is sponsored by

[See page](#)

Virto	11
Macaw	19
Winvision	24
Getronics	32
iDevteam	44

You are invited to click on the ad's for direct contact.



Web Content Management the SharePoint Server 2010 way

by Waldek Mastykarz

Web Content Management was introduced as a part of the SharePoint platform back in 2006. Soon people found out that while it was there, it was far from perfect. This was true both for end users and developers. The User Interface was confusing and users seemed to quickly lose track of what they were doing. As for developers, they had little control of the rendered HTML so implementing custom branding and delivering great Internet-facing sites was pretty challenging. SharePoint 2010 got a major makeover in different areas. One thing is for sure: delivering great Web Content Management on the SharePoint platform was never that easy.

The ribbon – one for all

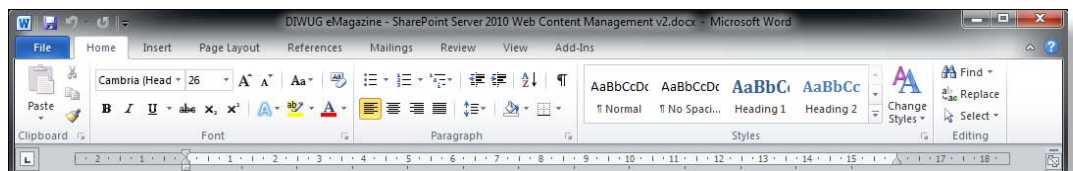


Figure 1: The ribbon in Microsoft Word 2010.

The ribbon was introduced as an alternative for the toolbar about four years ago. Research on usability of the Microsoft Office clients showed that many users experienced the toolbar as rather complex, finding it difficult to locate certain functions. Compared to the toolbar, the ribbon is based on the current context; it highlights menu options which are applicable to what the user is currently doing. This makes it easier to discover the available functionality.

The ribbon has been around for nearly four years now and many users experience it as one of the biggest improvements within the Office suite. If you look at SharePoint 2007, it resembles pre-ribbon Office: so much to offer but not all could be discovered intuitively. The ribbon has now been shipped as part of SharePoint 2010.

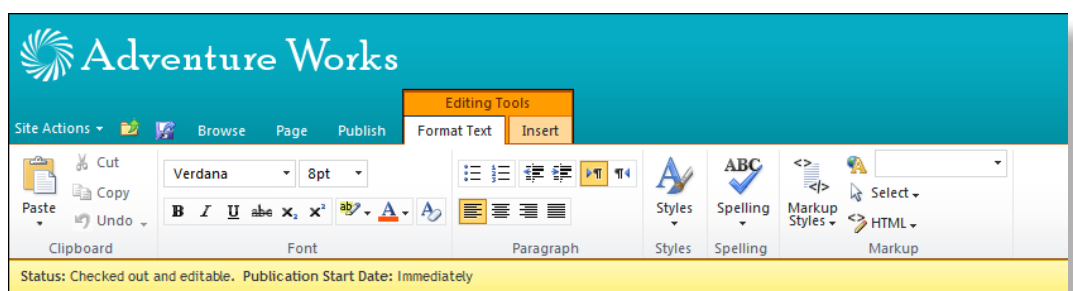


Figure 2: The ribbon in SharePoint 2010.

My precious...

Although it looks perfect in all supported browsers, the ribbon is made using nothing more than HTML and JavaScript. That means that it uses standard technology available on every PC all over the world and is accessible for users with disabilities. The best part of it all is that it's fully customizable. You can add new buttons to existing groups, new groups to existing tabs or brand new tabs – context dependent or not, which will provide the functionality for your custom solution. All of it can be done using supported API: both declaratively using XML and through the SharePoint API.



Below you see an example of XML which adds a custom button to the Content Group from the Insert tab.

```
<?xml version="1.0" encoding="utf-8"?>
<Elements xmlns="http://schemas.microsoft.com/sharepoint/">
  <CustomAction Id="MaventionLoremIpsum"
    Location="CommandUI.Ribbon"
    Sequence="20">
    <CommandUIExtension>
      <CommandUIDefinitions>
        <CommandUIDefinition
          Location="Ribbon.EditingTools.CPInsert.Content.Controls._children">
          <Button Id="Ribbon.EditingTools.CPInsert.Content>LoremIpsum"
            Alt="Insert Lorem ipsum"
            Command="MaventionInsertLoremIpsum"
            LabelText="Insert Lorem ipsum"
            Sequence="20"
            Image16by16="/_layouts/$Resources:core,Language;/images/formatmap16x16.png" Image-
            16by16Top="-16" Image16by16Left="-16"
            Image32by32="/_layouts/$Resources:core,Language;/images/formatmap32x32.png" Image-
            32by32Top="-32" Image32by32Left="-32"
            TemplateAlias="o1"
            ToolTipTitle="Insert Lorem ipsum"
            ToolTipDescription="Insert Lorem ipsum text"/>
          </CommandUIDefinition>
        </CommandUIDefinitions>
      <CommandUIHandlers>
        <CommandUIHandler Command="MaventionInsertLoremIpsum"
          CommandAction="javascript:alert('Lorem ipsum');"/>
      </CommandUIHandlers>
    </CommandUIExtension>
  </CustomAction>
</Elements>
Thank!
```

Listing 1: XML snippet that adds a custom button to the Content Group on the Insert tab of SharePoint 2010 ribbon.

The most important piece is the ID of the location where you want to add the button. You can find the ID in at least three ways. First of all you can navigate to the SharePoint SDK on MSDN where the most common IDs are described. The second way is to open the 14\TEMPLATE\GLOBAL\XML\CMDUI.XML file where the standard menu options are described. Not only will this allow you to discover the IDs but also layout templates supported for the given group. The last approach is to use the Internet Explorer 8 Developer Tools and select the group. The ID of the group is displayed as the ID of the list item.

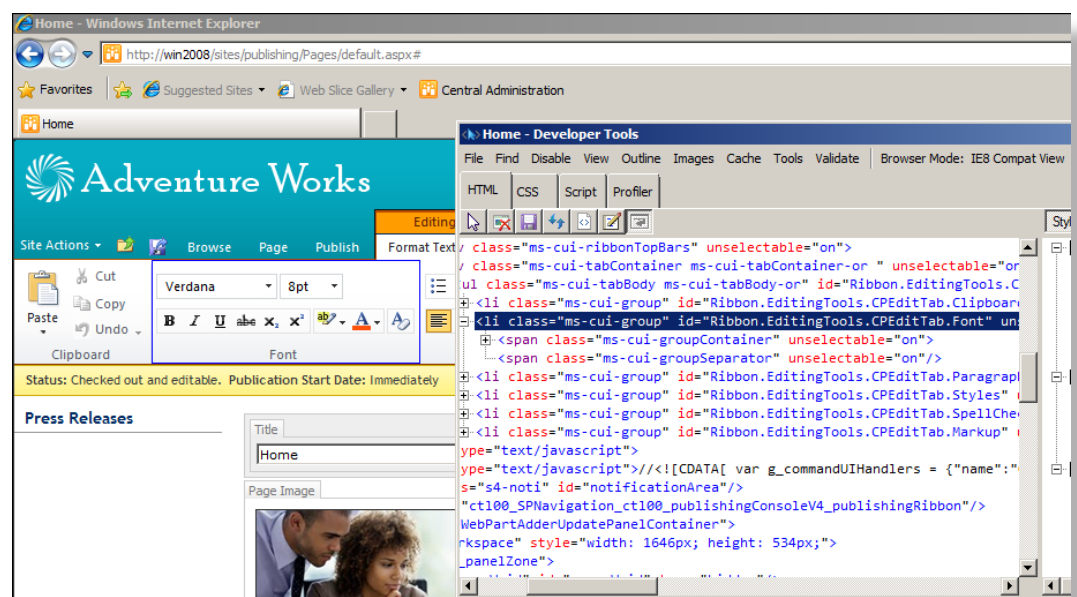


Figure 3: Discovering the ID of a ribbon Group using the Internet Explorer Developer Tools.



After you find out the right ID, don't forget to append Controls._children to it. One last tip: don't forget to clear your browser's cache each time after changing the XML of your ribbon customization. The contents of the ribbon are cached in the browser; without clearing the cache, your changes will not be visible. A good way to clear your cache is to use > Always Refresh from Server option from the Internet Explorer 8 Developer Tools. The ribbon ships with SharePoint Foundation 2010. Although it's not specific to Web Content Management solutions, the odds are high that if you will provide custom functionality with your WCM solution, it will be available through the ribbon.

Content Query Web Part v4

Probably one of the most frequently used Web Parts in Web Content Management solutions built on the SharePoint platform is the Content Query Web Part (CQWP). This Web Part allows you to roll up content from within your Site Collection using a easy and high-performance method. By using XSLT presentation templates you can separate the query results from the UI and create different overviews in a matter of minutes.

The Content Query Web Part shipped with SharePoint Server 2010 has two new features which simplify creating dynamic and reusable content aggregations: slots and content-to-content.

Slots – and it's not about Vegas

If you have worked with the CQWP in Office SharePoint Server 2007, then you know that creating new presentation templates wasn't as easy as we all wanted it to be. You were required to provide the internal names of all fields that you wanted to include in the template. If you wanted to reuse the template across different queries, you had to export the Web Part, tweak its XML and import it again. While you could cope with this approach as a developer, there was little chance of a user changing this all by himself. To simplify this process, the slots have been introduced in the Content Query Web Part in SharePoint Server 2010.

A slot in the CQWP is a placeholder that describes the position of some content in the XSLT presentation template. In technical terms: every `<xsl:value-of select="@myslot" />` becomes a slot which can be bound to a field from a query through Web Part properties. To see how it works, let's create a new Item Template:

```
<xsl:template name="DetailedOverview"
              match="Row[@Style='DetailedOverview']" mode="itemstyle">
  <xsl:variable name="SafeLinkUrl">
    <xsl:call-template name="OuterTemplate.GetSafeLink">
      <xsl:with-param name="UrlColumnName" select="'LinkUrl'"/>
    </xsl:call-template>
  </xsl:variable>
  <xsl:variable name="DisplayTitle">
    <xsl:call-template name="OuterTemplate.GetTitle">
      <xsl:with-param name="Title" select="''"/>
      <xsl:with-param name="UrlColumnName" select="'LinkUrl'"/>
      <xsl:with-param name="UseFileName" select="1"/>
    </xsl:call-template>
  </xsl:variable>
  <p>
    <strong>
      <xsl:value-of select="$DisplayTitle"/>
    </strong><br />
    <xsl:value-of select="@Author"/>, <xsl:value-of select="@Date"/>
  </p>
  <p>
    <xsl:value-of select="@RollupImage"/>
    <xsl:value-of select="@Details"/>
  </p>
  <p>
    <a href="{ $SafeLinkUrl }">Read more</a>
  </p>
</xsl:template>
```

Listing 2: New Item Template.



When you add a Content Query Web Part to a page and go to its properties, you will see your custom Item Template in the templates list – nothing new here. But as soon as you select your custom template, the list of available slots changes:

The screenshot shows the 'Fields to display' section of the Content Query Web Part properties. It includes a 'Link' slot with 'URL Path;', an 'Author' slot, a 'Date' slot, a 'RollupImage' slot, and a 'Details' slot. The 'Item style' is set to 'DetailedOverview' and 'Play media links in browser' is checked.

Figure 4: List of available slots changes after selecting the Item Style.

In every slot you can now enter one of more fields from the query. If the value of the first field provided is empty, the next will be bound and so on.

Slots allow you to create semantic presentation templates that describe what you are showing and how you are showing it instead where the data comes from.

Content Query (dynamic) Web Part

Another new concept implemented in the Content Query Web Part in SharePoint Server 2010 is content-to-content. If you can remember, in Office SharePoint Server 2007, there was no out of the box way to dynamically filter a query, such as filtering an overview or displaying selected products. You would either have to subclass the Content Query Web Part and do the dynamic filtering yourself or create a brand new Web Part with the risk that it would be slow and inflexible.

The Content Query Web Part in SharePoint Server 2010 ships with the ability to filter a query dynamically based on either a query string variable or a value from a field from the current page.

Let's look at the following Web Part which shows an overview of some press releases. As you can see every press release belongs to a category.

The screenshot shows a web part titled 'Press Releases' with a list of four items. Each item has a title and a category.

Press release 1	Category A
Press release 2	Category B
Press release 3	Category C
Press release 4	Category A

Figure 5: Press Releases overview. Every Press Release belongs to a category.



Imagine that you would like to provide an overview of press releases from Category A. All you need to do is include a list of all available categories somewhere on the page. Then you can modify the CQWP to retrieve the category value from the query string:

Additional Filters: ⓘ

Show items when:

Title
begins with
Press

☒ And ☐ Or

Category
is equal to
[PageQueryString:c]

☐ And ☒ Or

<no filter>
is equal to

Figure 6: Configuring Content Query Web Part to retrieve the category from the query string *c* parameter.

By using the PageQueryString token in the filter value, you can retrieve the filter from the query string and filter the query results dynamically based on the selected category. The best part is that if the value of the query string parameter is empty, the whole filter is excluded from the query, making it extremely easy to provide fully functional content aggregations using nothing more than standard functionality!

One more thing that the content-to-content concept allows you to do is to filter the query results based on the value of a field from the current page. Imagine that you would like to display related press release on the detail page of every press release. In Office SharePoint Server 2007 you would very likely end up creating a custom Web Part. In SharePoint Server 2010 you can use the standard functionality of the CQWP to get the job done!

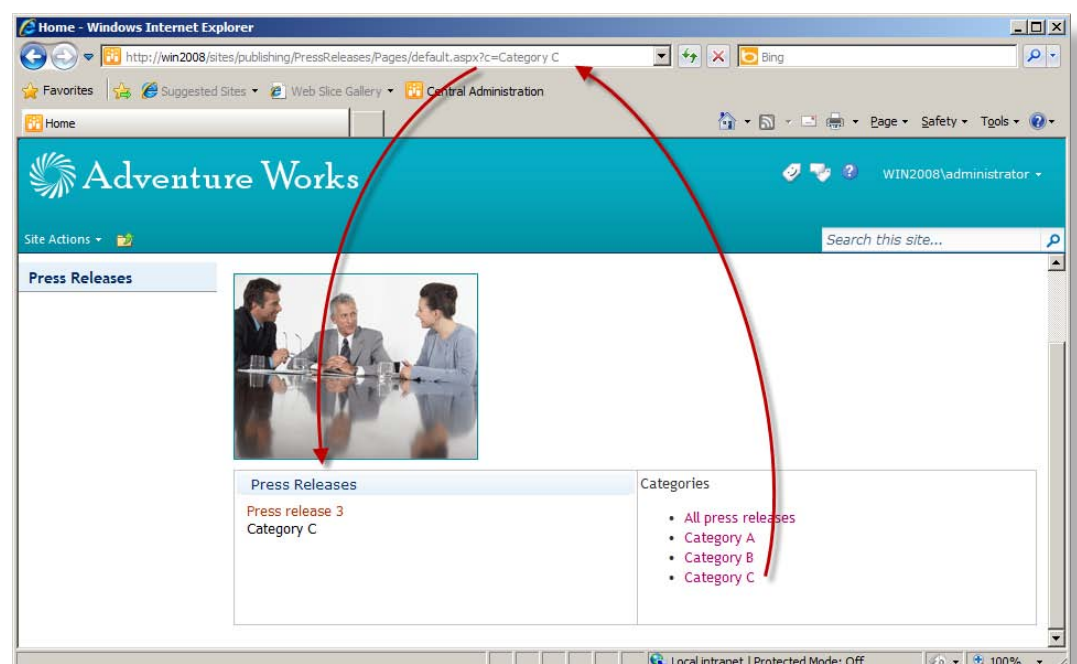


Figure 7: Press Releases overview filtered using the category from the query string.



Using the value of a page field is very similar to filtering using query string. The only difference is that you need to use the PageFieldValue token and the internal name of the field that you want to use to filter the results.

A word or two about branding

Branding in SharePoint 2007 consisted of two parts: themes and Master Pages. While working on Internet-facing websites, you mostly used Master Pages which would allow you to deliver highly customized branding according to the corporate identity of your customer. While using Master Pages gave you flexibility, it was quite complex to do what the web designers asked from you. On the other hand there were the simple themes. Applying mostly to collaboration solutions, themes allowed you to change colors, fonts, perhaps a background and that's it. The most important thing is that you had to use either the one or the other but never both at the same time.

SharePoint 2010 introduces new branding possibilities. Not only has the theming engine totally changed, it can now also be combined with master pages!

Theme-X

SharePoint 2010 leverages themes that are recognised from the Office clients. For example, using Microsoft PowerPoint, you can easily specify a number of colors and fonts, then export these settings as a theme. This will produce a .thmx file which can be uploaded to SharePoint. The theme itself is still nothing more than a definition of fonts and colors. The magic happens when you create custom branding that uses themes to create dynamic branding.

The first thing that you can do to leverage the new theming engine is to replace colors. Take a look at the following picture:

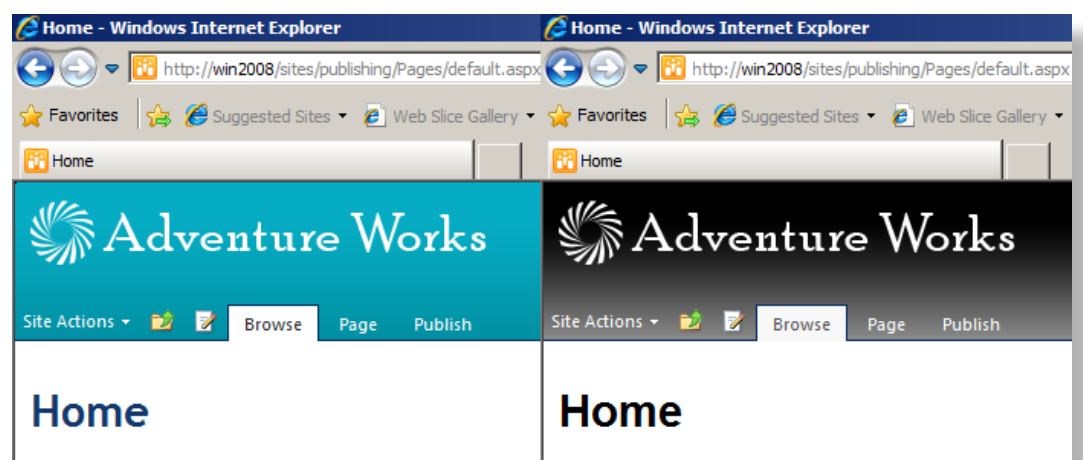


Figure 8: Adventure Works site recolored using a Theme.

Although the color schemas of both sites look totally different, all that changed is the theme. The theming engine supports three tokens that allow you to create dynamic brandings: ReplaceColor, ReplaceFont and RecolorImage. While the application of the first two is pretty obvious, the third one is really powerful. Using the RecolorImage token you can change colors of your images and what allows you to create themed branding with just a few lines of CSS code.

Using tokens from the theme in your custom branding is very simple. All you have to do is to prepend the CSS property that you want to replace with the color or font from the theme, for example:

```
body {  
    /* [ReplaceColor(themeColor:"Light2-Medium")] */ background-color: #f00;  
}
```

Listing 2: Replacing the background color using a Theme.



When no theme is applied, the background color of the body would be red. But after you select a theme, the background color would be set to whatever the Light2 color is.

Standard standards

One of the biggest challenges while working on Web Content Management solutions on Office SharePoint Server 2007 was the lack of control of the generated HTML markup. The standard markup was terrible: it wasn't compliant with any standard, it wasn't accessible and you had to do a lot of work to get it right.

The rendered HTML markup has been redesigned in SharePoint 2010. First of all, SharePoint Foundation 2010 targets the XHTML 1.0 Strict standard. And although it's not 100% there, it's very close, which is very impressive considering the rich functionality that it ships with and the amount of time the SharePoint Team had to get it done. If you look at the source of a Publishing Site you will see more semantic HTML: Web Parts are now rendered as div's and the whole page is just more responsive and better supported in browsers other than Internet Explorer.

What's on the menu?

One of the pain points while working with custom branding and Office SharePoint Server 2007 WCM solutions was branding the navigation. Out of the box SharePoint used the standard ASP.NET menu control, which was rendered as a table. Although it looked okay, the markup consisted of tables and was terrible to brand.

In SharePoint 2010 the menu control has been improved. The UseSimpleRendering attribute has been added which allows you to render the menu as an unordered list instead of a table. Let's compare the HTML rendered by default and after using the simple rendering mode.

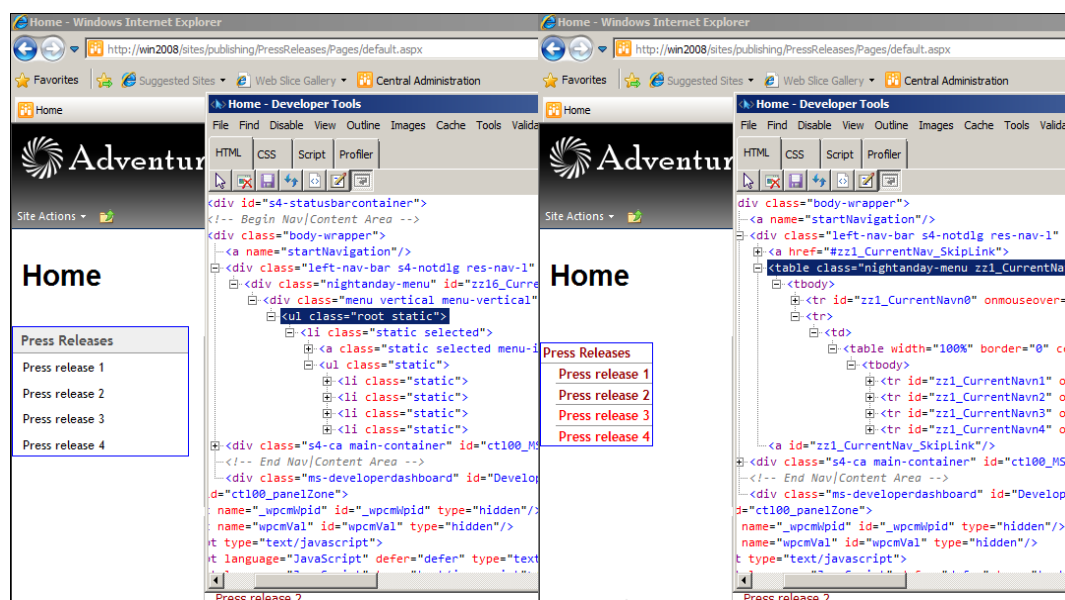


Figure 8: Comparing HTML markup of a SharePoint menu. On the left the new menu using the simple rendering mode produces an unordered list. On the right menu using the classic rendering mode using tables.

See the difference? What seems like a single attribute on the outside is a redesign of the control on the inside. But the most important thing is, that it's available out of the box which allows you to benefit from it in your WCM solutions and focus on the really custom pieces.



Summary & Links

SharePoint Server 2010 ships with many improvements and new features which simplify designing and developing Web Content Management solutions. By using the standard functionality you can deliver rich solutions and minimize the amount of custom development what results in a solution that is more scalable and better maintainable. As many of these improvement have been done in SharePoint Foundation 2010 you can benefit of them also while working with collaboration and search solutions.

SharePoint 2010:

[---http://sharepoint2010.microsoft.com/Pages/default.aspx](http://sharepoint2010.microsoft.com/Pages/default.aspx)

SharePoint 2010 SDK:

[---http://msdn.microsoft.com/en-us/library/ee557253\(office.14\).aspx](http://msdn.microsoft.com/en-us/library/ee557253(office.14).aspx)

SharePoint 2010 Developer Center:

[---http://msdn.microsoft.com/en-us/sharepoint/default.aspx](http://msdn.microsoft.com/en-us/sharepoint/default.aspx)

SharePoint Web Content Management:

[---http://blog.mastykarz.nl/tag/wcm/](http://blog.mastykarz.nl/tag/wcm/)

Content Query Web Part:

[---http://blog.mastykarz.nl/tag/content-query-web-part/](http://blog.mastykarz.nl/tag/content-query-web-part/) ■



Virto Workflow Activity Kit – Additional Activities for Designing Custom Workflows in Microsoft SharePoint 2007 & 2010.

Microsoft SharePoint allows for customizing business processes with its more than 30 standard activity blocks. However, the standard activities are not always sufficient for organizing specific business processes.



Virto SharePoint Workflow Activity Kit expands the set of the standard SharePoint activities, providing users **with wider opportunities to create custom activity sets**, and thus, better customize workflows and business processes.

The Workflow Kit is a constantly developing and expanding set of **no-code and ready-to-use activity blocks** that can be easily added to SharePoint Designer to build a new custom workflow within a minute.

By now the Workflow Kit contains a number of activities for **e-mailing, instant messaging, active directory management, sending SMS as well as text, InfoPath, HTTP, document library and SharePoint site management** activities.

With these complementary activities one can easily design workflows of any complexity and length!

VirtoSoftware Ltd. T +(353) 1 442 91 81

Rich UI Web Parts for Microsoft SharePoint

Virto
Software.com



Themes in SharePoint 2010

by Wictor Wilén

Branding and theming SharePoint is one key factor for great end-user adoption. If the user does not like the interface he or she will not use it. SharePoint 2010 introduces a new way of theming an out-of-the-box or branded SharePoint site. Think of themes as light-weight branding and they are configured per site and can be a great way for the site owners to make their site more personal.

One problem that I have with third party solutions is that they often contain some kind of own images or CSS definitions which not always fit into the currently used brand. This is also one reason that some say no to third party solutions "they just don't fit in". Using the new theming engine in SharePoint 2010 all SharePoint solution vendors can make their applications themes aware and by that get a better end-user acceptance.

Themes in SharePoint 2010

The theming in SharePoint 2010 have had a total make over compared to its predecessors. Themes are easily customizable and can be added to SharePoint without any fiddling with files on the SharePoint Servers. Designers can now use the web browser or use Microsoft PowerPoint to create and edit themes. The themes are stored as files in a new site collection gallery called the Theme Gallery.

Theming a SharePoint Site

The themes in SharePoint is applied to a specific site. By going to Site Settings and selecting Site Theme under the Look and Feel section the theme can be changed. To enable or change a theme select a theme from the list and select apply, see figure 1. The theme engine then applies theme customization to all the necessary files and within a few seconds the theme is applied to the site. If you would like to revert to the default settings then select the Default option at the top of the list. This will use the default CSS files and images.

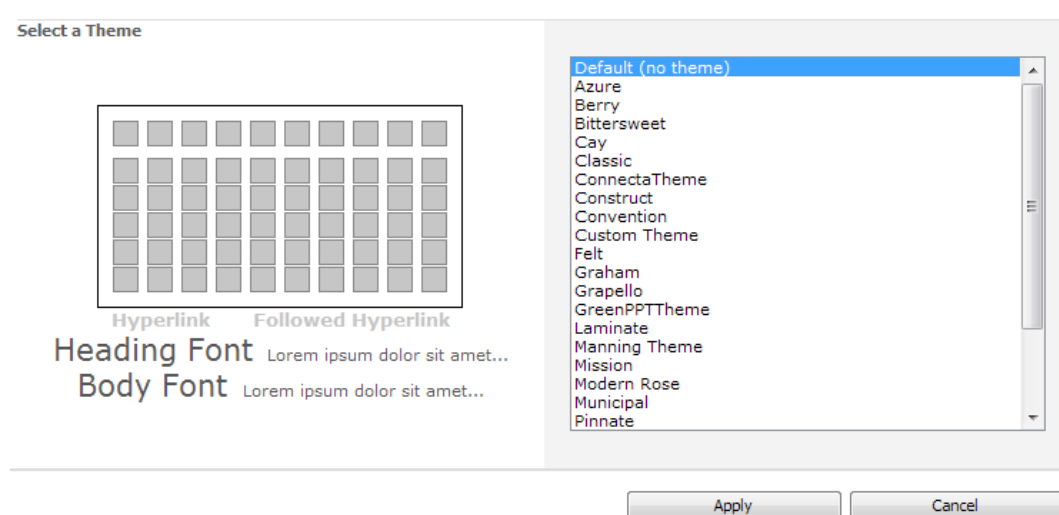


Figure 1: SharePoint Foundation 2010 allows you to easily change the theme on a site.

If you are using SharePoint Server 2010 you will see a few more options compared to SharePoint Foundation 2010. The server version of the Site Theme page allows designers to customize the theme in the browser by specifying the colors and fonts. There is also a preview function that opens up a new window with the selected or customized theme temporarily applied. While the Foundation version of Site Theme administration applies a theme to the current site the server version allows you to push down themes to sub-sites and configure theme inheritance for sub-sites if the publishing features are enabled.



The permissions required to apply themes are the Apply Theme and Border permissions. By default only users or group with the Design permission level are entitled to change the theme. In cases where you do not want users, even though granted control on the site collection, to change the theme you can restrict this from the Central Administration. In Central Administration you go to Application Management > Manage Web Applications and select the web application for which you like to disable theme editing. Click on User Permissions in the Security group in the Ribbon and deselect the Apply Theme and Borders permission. This will disallow users from applying a new theme but it does not hide the Site Theme application page from the Site Settings page.

Working with the Theme Gallery

The Theme Gallery which exists in the Site Collection is accessed by going to Site Settings > Galleries > Themes or by manually entering the URL which is `/_catalogs/theme/`. This is a standard SharePoint library that contains all the Themes available in the site.

This library can be used to configure the available themes in the site. For instance you can remove the themes that you do not want your users to use or upload new themes. The gallery inherits the permissions from the top-level site so Members have contribute permissions and Owners full control.

Theme files

Each theme in the theme gallery consists of a name and a file. This file is of the type Microsoft Office Theme and has the extension `.thmx`. This is the same format that for instance Microsoft PowerPoint uses for themes. This means that you can take a theme made in PowerPoint and export it and upload it to the Theme gallery. Even though it is a cool feature and is very demo friendly I do not think that this is good to teach your end-users. Unfortunately the themes exported from PowerPoint will not look as good in SharePoint as they did in PowerPoint, unless very carefully created, just test it with your standard corporate presentation. If you are going to create a theme and have the proper permissions rather use the web interface in SharePoint to create it.

If you still want to use your PowerPoint theme in SharePoint you select the Design tab in PowerPoint. Expand the Themes gallery in the Ribbon and select Save Current Theme, see figure 2. Once the document is saved navigate to the Theme gallery in SharePoint and upload the file. The theme is now available for use in the Site Collection.

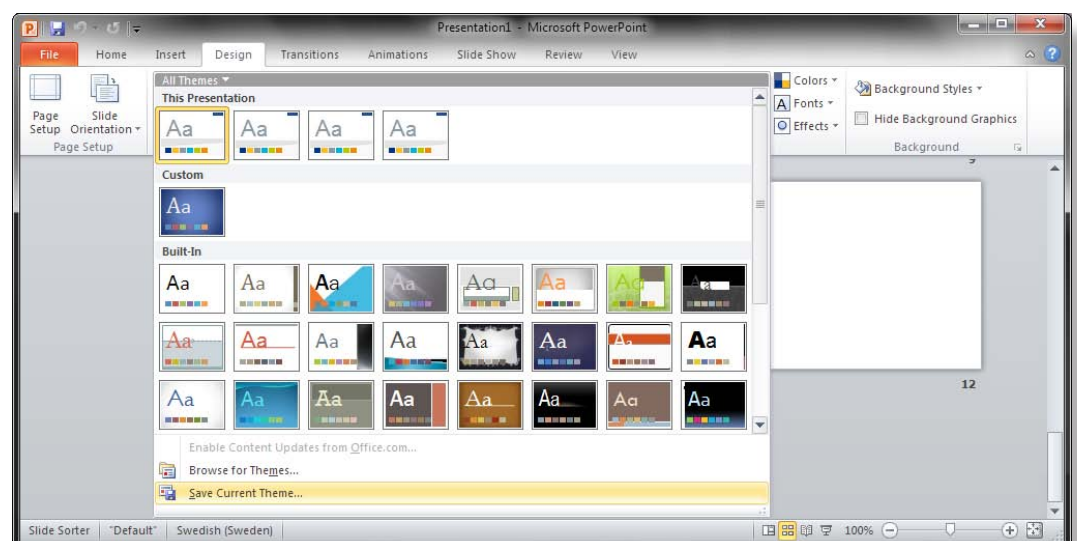


Figure 2: PowerPoint can be used to export a theme which then can be imported into SharePoint 2010.



The.thmx files are ZIP-compressed containers conforming to the Open Packaging Convention, which is specified in the Office Open XML standards ISO/IEC 29500:2008. The container consists of a file structure containing XML files and resources. To take a peek on how it looks like just rename the .thmx file to .zip and extract the files.

Centrally manage themes?

Unfortunately there are no out-of-the-box ways to manage the themes through the user interface in the Central Administration – an opportunity for a community contribution here. By using features or feature stapling the features can be managed though. In all SharePoint deployments you need to consider which themes to be available in the Theme gallery and create a solution that manages those themes.

A solution for this could be a feature that is stapled on specific Site Definitions or on the global site definition. Using a Feature Receiver unwanted themes can be removed and with a feature element module you can add new themes. Themes are added to the gallery using a Module in the elements manifest file like in listing 1.

```
<Elements xmlns="http://schemas.microsoft.com/sharepoint/">
  <Module Name="ThemeAdder"
    Url="_catalogs/theme"
    List="123"
    RootWebOnly="TRUE">
    <File Path="ThemeAdder\WictorsTheme.thmx"
      Url="WictorsTheme.thmx"
      IgnoreIfAlreadyExists="True"
      Type="GhostableInLibrary"/>
  </Module>
</Elements>
```

Listing 1: A Module can be used to provision new themes to the Theme gallery.

The Theme gallery is a list template with the ID equal to 123 and is always located at the _catalogs/theme URL. A top-level site or site collection scoped feature can use a module like the one above to provision new theme files.

The SharePoint Theme Engine

SharePoint 2010 contains a new Theme Engine which replaces the old one. This theme engine is used to read the theme information from the .thmx files and automatically create CSS files and recolor images using that information.

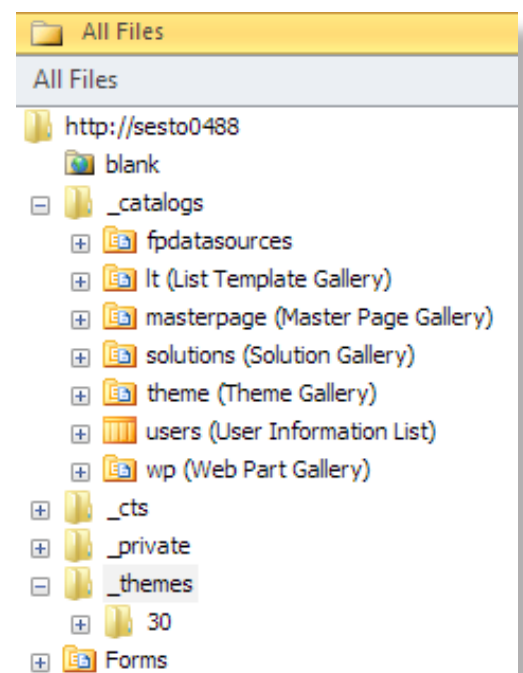


Figure 3: SharePoint Designer 2010 can be used to browse the contents of a site and even access the generated files of SharePoint Theming engine.



The Theme Engine will start processing all themable files when a new theme is applied. It will then extract the files from the .thmx file and traverse all the themable files and replace colors and fonts and recolor images according to the rules in the CSS files. These files are then stored in a folder created in the top-level site called `_themes`. When a new theme is applied a new sub folder of the `_themes` folder is created and the old one is removed. Each folder is given a new number and each file is post-fixed with a hash-key to minimize the risk for client caching of the themable files.

SharePoint Designer 2010 can be used to inspect the generated files, see figure 3. You can even use SharePoint Designer to fine tune your CSS files so you don't have to re-apply the theme over and over again. Remember to save the changes to your local drive since the files will be removed once you apply a new theme.

Creating themable CSS files

The themes in SharePoint 2010 is a great improvement of theme management in SharePoint but what is even greater is that you can adapt your own CSS files and images to take advantage of the theming engine. I will show you how to add a Web Part specific CSS file including images that will use the current theme. This allows you to build solutions that can be used using any theme and always looks good and fits into SharePoint.

Use theme colors in CSS

To use themes in Web Parts, application pages or other SharePoint solutions then you need to use the out-of-the-box classes or create your own CSS files and CSS selectors. To make a CSS file themable you have to follow a few simple rules:

- Add the CSS file to `/_layouts/<LCID>/Styles/Themable` folder
- Register the CSS files using the **SharePoint:CssRegistration** control
- Set the **EnableCssTheming** property to true on the `CssRegistration` control

The Theme engine also accepts that CSS files are deployed into the Site Collection in the Style Library folder. The CSS file has to be in the `~SiteCollection/Style Library/Themable` or in the `~SiteCollection/Style Library/~language/Themable` folders.

To create and build a Web Part that uses a custom CSS we need to create a new empty SharePoint 2010 project in Visual Studio 2010 and select to deploy it as a farm solution.

In order to add a themable CSS file to a Visual Studio 2010 SharePoint solution you add a new SharePoint mapped folder by choosing Project > Add SharePoint Mapped folder. In the dialog expand the following nodes `TEMPLATE/LAYOUTS/1033/STYLES/Themable` and select OK. Then add a new empty CSS file to the Styles folder in Visual Studio and name it `CustomCss.css`.

Note: LCID 1033 is the US-English locale and you have to supply one CSS file per installed language pack.

In Visual Studio 2010 add a new Visual Web Part to the project which will be used to test the theme. Add the code as in listing 2 to the user control of the Visual Web Part.

```
<SharePoint:CssRegistration
  runat="server"
  ID="cssreg"
  EnableCssTheming="true"
  Name="themable/CustomCss.css" />

<asp:Label
  runat="server"
  ID="lblText"
  CssClass="myLabel"
  Text="Themes are great!" />
```

Listing 2: The `CssRegistration` control is used to register CSS files in SharePoint.



The `CssRegistration` control is used to register the CSS file with SharePoint. Using this control also allows us to register it in several Web Parts or other items and only have it render once. During runtime the style sheets will be rendered using the `CssLink` control. The name of the CSS file is specified in the `Name` attribute of the control. The name must be prefixed with `themable/`. The `EnableCssTheming` property is set to `true` to inform SharePoint that this CSS can be themed. A `Label` control is also added which has a custom CSS class specified.

Note: the `CssRegistration` class is not accessible in the Sandbox and cannot be used in a Sandboxed Web Part.

Return to the CSS file and add a new class called `myLabel`, which is used by the `Label` control in the Visual Web Part. Listing 3 shows how the CSS class should look like:

```
.myLabel {  
  /* [ReplaceColor(themeColor:"Accent6-Darkest")] */ border:1px solid black;  
  /* [ReplaceColor(themeColor:"Light2-Lighter")] */ background-color:Silver;  
}
```

Listing 3: Using the `ReplaceColor` attribute colors are changed in the CSS files.

The listing shows that we have added CSS comments to the properties in the class. These comments contains attributes that is used by the SharePoint theme enging to theme the CSS property values. The `ReplaceColor` theme attribute is used to change the value of any colors in the property value.

Property	Description
themeColor	The color in the format Color[-Shade] (see figure X.X)
color	A specific color. Can be used instead of themeColor
themeShade	Sets shade of the color. Value range 0.0 to 1.0
themeTint	Sets tint of the color. Value range 0.0 to 1.0

Table 1: Available theme properties for the `ReplaceColor` theme attribute.

For instance `[ReplaceColor(color:"red",themeTint:"0.5")]` creates a pink color. Figure 4 shows how the colors are organized in the Site Theme settings page. You can use this as a reference when choosing value for the `themeColor` property.

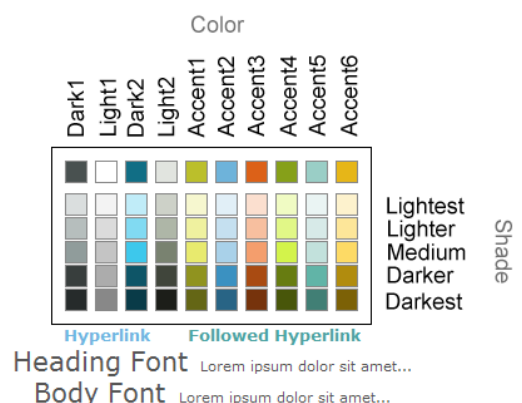


Figure 4: All themes are based on four basic colors and six accents of which all have five default shades.

If you now build and deploy your solution to SharePoint you will see that the default colors in the CSS file will be used. The themed files are only generated when you switch themes. This makes iterative work quite hard. A solution to this is to add an Event Receiver to your solution which you use during development. This feature applies a Theme to the site every time it is deployed. The `FeatureActivated` method of such an Event Receiver could look like listing 4, depending on what scope you are using for your feature.



```
public override void FeatureActivated(
    SPFeatureReceiverProperties properties) {

    SPSite site = properties.Feature.Parent as SPSite;
    ThmxTheme.SetThemeUrlForWeb(site.RootWeb,
        "/_catalogs/theme/Classic.thmx");
}
```

Listing 4: You can improve the development experience when working with Themes by re-applying the theme upon feature activation.

If you have the need to have a very specific or different CSS file for sites that does not use a theme then you could add a secondary, non-themed, CSS file to the TEMPLATE/LAYOUTS/<LCID>/STYLES folder and remove the themable/ prefix from the CssRegistration control. Then SharePoint will use the non-themed CSS file when no theme is selected and the themed CSS when a theme is selected.

Use theme fonts in CSS

Just as you can replace colors in the CSS property values you can also replace the fonts. This is done using the ReplaceFont theme attribute. The ReplaceFont attribute has one property called themeFont which has two possible values:

- MajorFont – font for headings
- MinorFont – font for body text

The ReplaceFont attribute will not replace the current font in the CSS property value. It will instead add the themed font and use the default font as the fallback value. Listing 5 shows how to replace a font in a CSS selector.

```
/* [ReplaceFont(themeFont:"MajorFont")] */ font-family:Vivaldi;
```

Listing 5: Fonts are replaced using the ReplaceFont theme attribute.

Use theme colors on images

Something that is really great about the Theme engine is that it can recolor images. The recoloring is done using the **RecolorImage** theme attribute. Listing 6 shows how a background image is recolored using a theme color and the Filling method.

```
/* [RecolorImage(
    themeColor:"Light2",
    method:"Filling")]*/
background:url("/_layouts/images/camera.png") no-repeat;
```

Listing 6: A powerful feature of the SharePoint 2010 Theme engine is to recolor images.

The RecolorImage theme attribute have several parameters, see table 2.

Property	Description
themeColor	The color in the format Color[-Shade] (see figure 4)
method	Specifies the recoloring method. Valid values are Blending, Filling and Tinting.
excludeRectangle	Specifies a rectangle excluded for recoloring
includeRectangle	Specifies a rectangle included for recoloring, everything else is excluded.
detach	Should be set to true if the same image is recolored differently in different CSS selectors
darkColor	Specifies the dark color when creating duotone recolorings
lightColor	Specifies the light color when creating duotone recolorings
darkThemeColor	Specifies the dark theme color when creating duotone recolorings
lightThemeColor	Specifies the light theme color when creating duotone recolorings

Table 2: Properties of the RecolorImage Theme attribute.



Recoloring can be quite hard and requires a lot of testing. I recommend you to use gray scaled images in the themable CSS file. This will also make the resulting images look better. This is one case when you might not have the same CSS file in the LAYOUTS and Themable folders. The non-themable CSS file might contain a fully colored image instead.

When using the darkColor and lightColor or darkThemeColor and lightThemeColor duotone recoloring is done. The light pixels of the image will be replaced with the light color and the dark pixels are replaced with the dark color. Listing 7 shows how to duotone recolor an image and also excluding a rectangle from recoloring.

```
/* [RecolorImage(
    lightThemeColor:"Light2",
    darkThemeColor:"Dark2",
    method:"Filling",
    excludeRectangle:{x:0,y:20,width:48,height:8})*]*/
background:url("/_layouts/images/camera.png") no-repeat;
```

Listing 7: Recoloring images can be an advanced task and might take a lot of iterations to make it look good.

Figure 5 shows the original image to the left and in the middle the image that we first (listing 6) recolored. To the right is the second recolored image (listing 7) that used two different colors for recoloring.



Figure 5: Images can be recolored using a number of different settings.

Summary

The SharePoint 2010 Theme engine is superior to the themes in SharePoint 2007 and it allows the users to create their own themes easily using the web interface or using PowerPoint.

Making the SharePoint solutions themes aware is an easy approach to better end-user adoption and will make the application look more professional. Although it can be tricky to implement it is all worth it in the end.

Happy branding! ■



... BY THE WAY ...

Want to start up an incremental crawl on the content source using PowerShell?

```
$CrawlContent = Get-SPEnterpriseSearchServiceApplication|
                Get-SPEnterpriseSearchCrawlContentSource
$CrawlContent.StartIncrementalCrawl()
```

That should do the trick!

Tip from Albert-Jan Schot

appieschot@xs4all.nl

www.digiwijs.nl

Wat doe jij volgende maand?

Jij bouwt het wereldwijde intranet voor een grote multinational, waar de meer dan 25.000 gebruikers dagelijks met veel plezier mee werken. Je opent de poorten naar kritische ERP, BI en HR systemen en integreert die met de modernste Document Management en Social Networking applicaties.

Jij bent namelijk een specialist die wel raad weet met SharePoint 2010, Nintex Workflow en WSF. Iemand die altijd op zoek is naar nieuwe kennis en als eerste met de technologie van morgen wil werken, maar ook iemand die geniet van het succes dat de klant vandaag boekt met jouw innovatieve oplossingen.

Jij bent ondernemend en deelt graag je visie en kennis met je omgeving, zowel fysiek als online, en werkt samen met designers, architecten, developers en consultants aan de beste oplossingen voor je klant. Je weet hoe je werk en privé vlekkeloos kunt combineren en onafhankelijk van tijd en plaats de optimale bijdrage levert aan je team.

Volgende maand werk jij bij Macaw.

SharePoint Consultants
SharePoint Developers
Project Managers



Macaw is Nederlands grootste, uitsluitend in Microsoft technologie gespecialiseerde, IT dienstverlener. Onze kennis is gebundeld in vier Solution Centers die zich richten op het realiseren van collaboratieve portalen, ECM systemen, internet en commerce sites, business solutions, CRM systemen en applicatiebeheer en hosting diensten

Bezoek voor meer informatie over deze en andere vacatures onze website www.echtleukwerk.nl of neem contact op met onze afdeling Recruitment via 020-8510510 of recruitment@macaw.nl.



Suggestions for Social and Search in SharePoint 2010

by Andries den Haan

In its fourth generation, Microsoft SharePoint Server 2010 obviously offers a wide range of innovations and improvements throughout the whole platform.

One thing to notice is that behavioral information about users that work with the system is logged and processed. This data becomes available to analyze usage and to support users to connect and communicate with peers and in finding relevant information.

Examples of applying this capability are the "Colleague/Keyword Suggestions" in the "User Profile Service" and the "Search Query Suggestions" in the "Search Service". As these features are both quite complex, a quick peek under the hood is in order to determine their benefit for the new ways of working.

Colleague/keyword suggestions

The new "Social Computing" capabilities in SharePoint Server 2010 offer various components for large scale organizations to build "business networking" solutions.

By leveraging the "User Profile Service Application", end users can maintain their personal profile and colleague/peer list in the "My Site" host.

The personal profile offers (e.g. via "People Search") the option to find colleagues with a similar interest, connect and track each other's activities in the system.

Specific parts of the profile data and colleague list can be imported from directory services or Line- of-Business systems like HR or CRM systems (using Business Connectivity Services). Other parts (depending on configuration) can be populated and maintained by users.

Various fields in the personal profile may contain terms or keywords that the user can input to describe their expertise or areas of interest. This data may also be exposed in "(People) Search", so colleagues with a similar interest or questions on a specific area of expertise can connect.

The colleagues list (depending on the configuration) may be populated with direct peers from a directory service based upon the organizational structure.

Behavioral based profile keyword and colleague suggestions are very cool new features!

SharePoint Server 2010 combined with Outlook 2010 now provides an innovative solution to support "Business Networking" through behavioral based profile keyword and colleague suggestions.

Through their own behavior, users may receive colleague and keyword suggestions, using sent e-mails as a basis.

If the E-mail Analysis functionality in Outlook 2010 is activated (this can be deactivated for privacy reasons), colleagues and keyword from sent emails are analyzed and imported into SharePoint. This functionality is provided by the "Microsoft SharePoint Server Colleague Import Add-in".

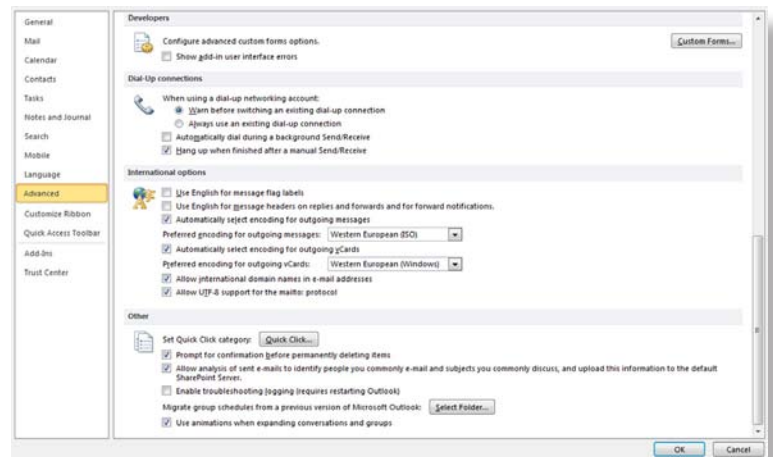


Figure 1: SharePointSuggestions.

Based upon a fixed schedule (once per 15 days), a job checks the "Sent" folder of the users' email box for items sent in the last 15 days.

Through analysis of the sent e-mail commonly used (new) terms are identified as "Keyword Suggestions". The same job also looks at the "To" field of sent e-mail and uses an algorithm to populate a weighted list of "Colleague suggestions".

After the analysis the "Colleague and Keyword suggestions" are sent to the standard configured SharePoint Server (see the link for "Enable SharePoint Server 2010 Colleague in Outlook 2010" at the end of this article) and stored in the "UserSuggestions" table of the "Profile" database.

After import, the keywords become available as suggestions in the personal profile for the user directly below the default "Ask Me About" en "Interests" fields. Clicking the keywords adds them to the profile fields.



Figure 2: SharePointSuggestions.

The "Colleague suggestions" are offered as suggestions on the "Colleagues" page through the "View Suggestions" button next to "Add Colleagues".

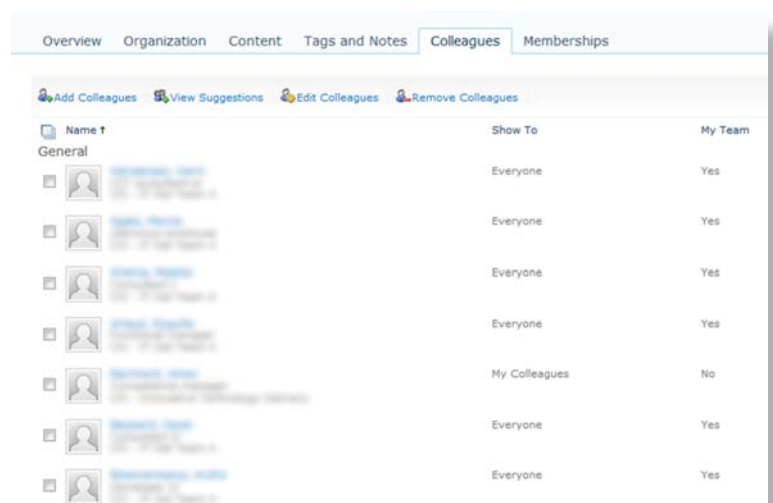


Figure 3: SharePointSuggestions.



Apart from this, a notification email is sent to users that have not updated their profile in the past period through the “My Site Suggestions Email Job” (which runs by default on the 15th of each month).

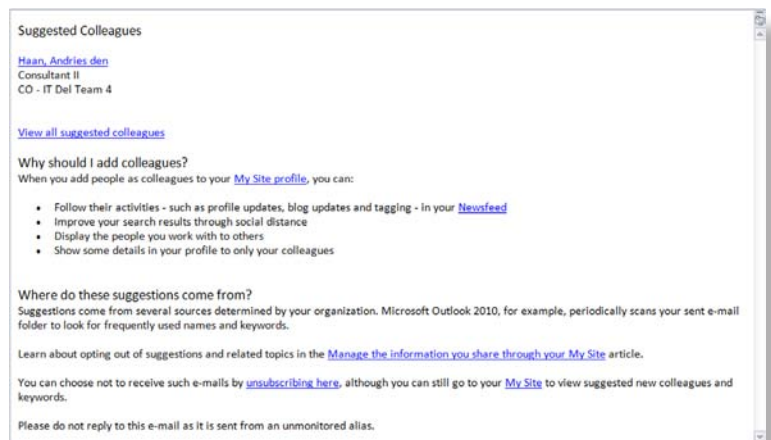


Figure 4: SharePointSuggestions.

This e-mail offers “Colleague and Keyword suggestions” to the recipient.

When users have updated their colleague list through “Colleague suggestions”, the activities of these new contacts (e.g. profile changes, status updates, ratings, comments and blog posts) are shown in the “Activity Feed” on the “My Newsfeed” page of the “My Site Host” and via the Outlook Social Connector in Outlook (if configured for “My Site”).

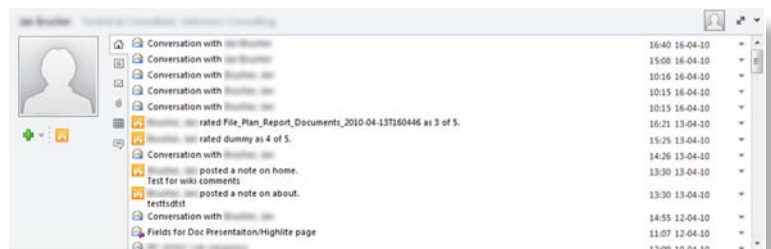


Figure 5: SharePointSuggestions.

Search Query Suggestions

With the ongoing growth of data (documents, web pages, and business data) in large organizations, the more important it is to provide an intuitive and efficient interface to find and discover relevant information.

SharePoint 2010 provides (amongst others) a “Search Center” (powered by SharePoint or FAST search service applications) with a “Refinement panel” to filter search results based upon metadata like file type, site, author or date.

“Search Query Suggestions” introduces a more efficient way of entering search queries by providing search term suggestions based upon previous queries by other users.

“Search Query Suggestions” are offered on two moments when users work with search:

- Before the actual search query when the user starts typing the query terms (“Pre-Query Search Suggestions”);

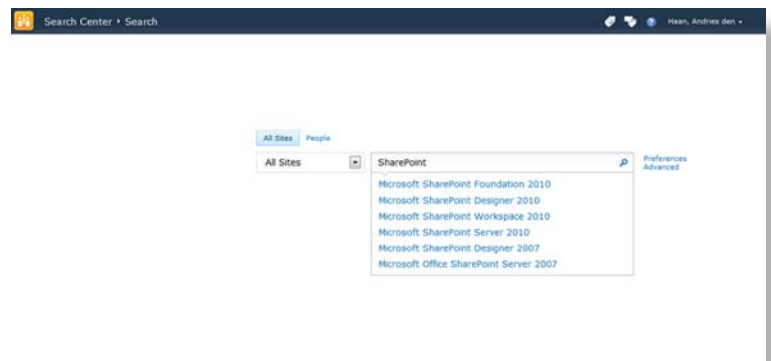


Figure 6: SharePointSuggestions.

- After the search query is submitted and search results are provided a “Related searches” pane is offered (“Post-Query Search suggestions”).

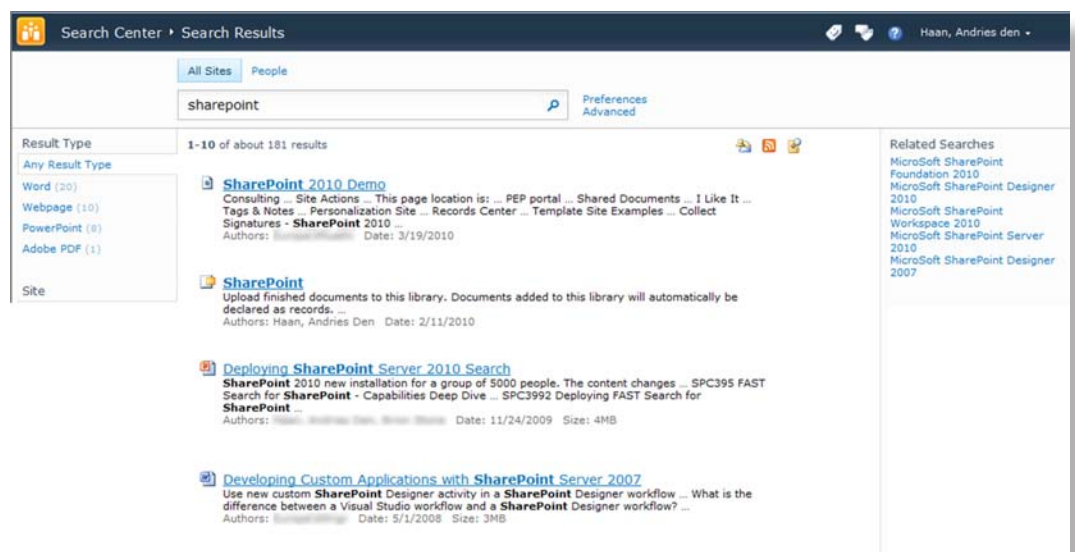


Figure 7: SharePointSuggestions.

The “Search Query Suggestions” are based upon the behavior of all users working with the “Search Service Application” via e.g. the “Search Center”. The query terms that are entered, are recorded in the “Query log”, after which they are processed and offered as suggestions.

Search query suggestions in SharePoint 2010 improve the search experience

However, there is a precondition to consider before a term becomes a candidate “Query Suggestion”. A term may become a candidate, after users have clicked through to an item (e.g. document, page or business data) in the search results based upon the query term at least 6 times.

The “Prepare Query Suggestions” timer job (runs once a day by default) processes the candidates and populates or updates the list with available query suggestions.

- Obviously a correct working “Search Service Application” needs to be available (crawling, processing search queries and returning results);
- The Query log needs to record and process submitted search queries (the “Query Logging” timer job runs every 15 minutes by default).

To test whether this functionality works and to manually add or remove query suggestions, a set of PowerShell commands is available:



- Open the "SharePoint 2010 Management Shell" window and request the current "Search Service Application":

```
$searchapp = Get-SPEnterpriseSearchServiceApplication
```

- To manually add a "Search Query Suggestion" (e.g. "SharePoint" or "SharePoint Server"), the following command is executed each time (where "<suggestion>" is replaced with the term to be added as "Query Suggestion"):

```
New-SPEnterpriseSearchLanguageResourcePhrase -SearchApplication $searchapp  
-Language En-Us -Type QuerySuggestionAlwaysSuggest -Name "<suggestion>"
```

- To actually use the added terms in e.g. the "Search Center", the "Prepare Query Suggestions" Timer Job needs to run via de "Central Administration" website or the following PowerShell command:

```
Start-SPTimerJob -Identity "prepare query suggestions"
```

- To list the available suggestions, the following command is executed:

```
Get-SPEnterpriseSearchQuerySuggestionCandidates -SearchApplication  
$searchapp
```

- The following command is used to remove terms from the list (where "<suggestion>" is replaced with the term that needs to be removed):

```
Remove-SPEnterpriseSearchLanguageResourcePhrase -SearchApplication  
$searchapp -Language En-Us -Type QuerySuggestionAlwaysSuggest -Identity  
"<suggestion>"
```

Working as a professional

for Winvision

Winvision is looking for Dutch speaking SharePoint specialists.
Are you the professional in SharePoint technology we are looking for?
Apply for a job at Winvision!

As a SharePoint professional at Winvision you will:

- Work alongside Microsoft on the newest technologies in TAP programs;
- Face technical challenges daily;
- Work with highly skilled, enthusiastic colleagues;
- Learn by sharing knowledge with other SharePoint specialists at Winvision.

Do you share our passion for Microsoft technology and are you not afraid to innovate and think out-of-the-box? Go to our website, www.werkenbijwinvision.nl, for more information on working at Winvision and send in your application.

Winvision

Working for Winvision means working for an organization that really appreciates you for your knowledge and expertise. We are a trusted partner in planning, developing, implementing, operating and hosting high-quality software solutions. All are based on the Microsoft Platform that help our clients obtain or exceed their primary business objectives. As a Microsoft Gold Partner Winvision is one of the top Microsoft partners worldwide.





To setup a scenario with behavioral based search suggestions, the previously named timer jobs can be used to speed up the process (as long as there is at least a 6 times click-through rate of the term based search results). The PowerShell commands can be used either to check the list with existing query suggestions or add/remove suggestions manually.

“Search Query Suggestions” are available in the following products:

- Microsoft Search Server Express 2010
- Microsoft Search Server 2010
- Microsoft SharePoint Server 2010
- Microsoft FAST Search Server for SharePoint 2010

Conclusion

The suggestion features in SharePoint 2010 promise a lot of opportunities to build robust business networking and search solutions.

The pre- and post-query search suggestions take Microsoft technology to the next level of Enterprise Search solutions and put the pressure on the competition in the search solution market.

The “Colleague and keyword suggestions” provide the foundation to build interesting solutions in an enterprise context. However, it would be expected that colleague suggestions would be based upon e.g. common profile terms or keywords between users instead of having a dependency with Outlook 2010 to provide the suggestions.

Fortunately the API provides the power to build non-Outlook dependent Colleague suggestions as commonly seen on public social networks like LinkedIn or Facebook. It's therefore still valid to say that SharePoint is not a solution by its self, but a platform to build solutions upon.

References

- Enable SharePoint Server 2010 Colleague in Outlook 2010:
[http://technet.microsoft.com/en-us/library/ff384821\(office.14\).aspx](http://technet.microsoft.com/en-us/library/ff384821(office.14).aspx)
- Social Computing in SharePoint Server 2010 (Beta):
<http://technet.microsoft.com/en-us/sharepoint/ee263906.aspx>
- Plan for social computing and collaboration (SharePoint Server 2010):
[http://technet.microsoft.com/en-us/library/ee662531\(office.14\).aspx](http://technet.microsoft.com/en-us/library/ee662531(office.14).aspx)
- What's new in enterprise search (SharePoint Server 2010):
[http://technet.microsoft.com/en-us/library/ee667266\(office.14\).aspx](http://technet.microsoft.com/en-us/library/ee667266(office.14).aspx) ■



... BY THE WAY ...

Add userrole Read and Upload

By default SharePoint 2010 will provide you a lot of userroles. Each of them are usefull, but sometimes you need a role for Read and Upload, in this case users cannot edit or delete their uploaded documents, handy for studentwork!

How to achieve this? Go on the toplevel to Site Actions, Site Permissions. In the Ribbon click on Permission Levels and Add a Permission Level.

Give it the name: Read and Upload and the following checkboxes must be checked:

Add items, View items, View Application Pages, View Pages. If you use columns in your library you must add an additional userole.

Tip from Jan Ligtenberg, trainer/consultant APS IT-diensten, www.apsitprojecten.nl



Community Kit for SharePoint: Development Tools Edition

by Wes Hackett

Visual Studio 2010 includes new integrated tools for SharePoint 2010 development. The teams at Microsoft have done a superb job creating a set of tools to aid the creation of SharePoint 2010 development projects. The Microsoft tools include item and project templates, deployment infrastructure and an extension to the Server Explorer allowing interrogation of a SharePoint site from within Visual Studio.

Introduction

One of the best aspects of the new tools is that they incorporate extensibility points so that you and I can build on top of the Visual Studio environment to meet any tooling requirements that might be required. This is where the Community Kit for SharePoint: Development Tools Edition fits in. The CKSDev extends the Visual Studio 2010 SharePoint project system with advanced templates and tools. Using these extensions you will be able to find relevant information from your SharePoint environments without leaving Visual Studio. You will have greater productivity while developing SharePoint components and you will have greater deployment capabilities on your local SharePoint installation.

There are four core areas of CKSDev. Environment covers the enhancements to the Visual Studio IDE including features such as the new SharePoint References tab on the Add Reference dialog. Exploration extends the SharePoint Explorer with advanced information about the SharePoint sites and features to interact with the site without leaving Visual Studio. Content includes new project and item templates to bring features such as LINQ to SharePoint, Custom Action and Sandboxed Visual Web Part to improve development productivity. The Deployment area provides enhancements to the deployment steps and mechanisms from Visual Studio such as auto quick deploy when working with ascx files.

Environment

Visual Studio 2010 includes numerous IDE enhancements which aid the productivity for a developer. CKSDev adds some new features to enhance the IDE even further. As a SharePoint developer it's common to need references to at least the Microsoft.SharePoint assembly. This normally involves browsing to the SharePoint root and locating the assembly in the ISAPI folder. The SharePoint references tab now appears on the 'Add Reference' dialog listing all the common SharePoint assemblies removing the need to browse through the Root.

To access the SharePoint references tab right mouse on the selected project and choose the 'Add Reference' option. The standard dialog is now displayed. If you look along the top you'll see a tab marked SharePoint. This is the tab containing commonly used SharePoint assemblies. Select the required references and click ok. Those SharePoint assemblies will now be listed within the project references as shown in Figure 1.

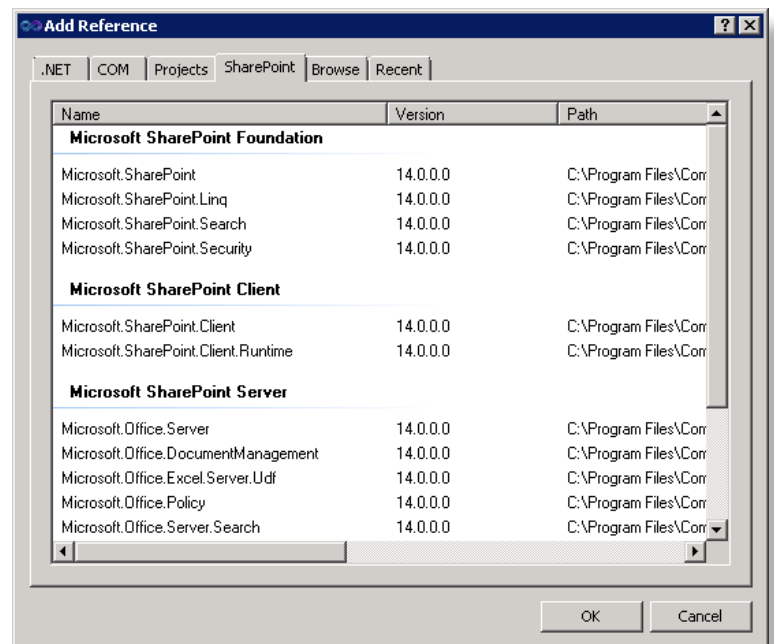
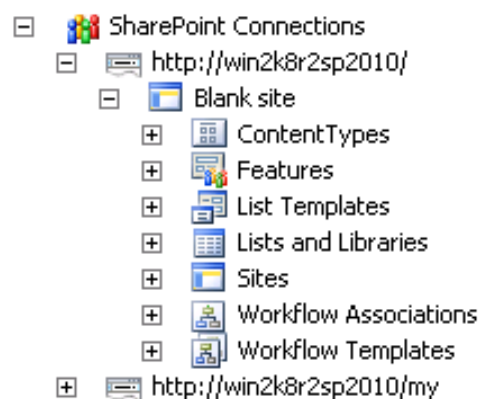


Figure 1: The SharePoint references tab.

Exploration

Server Explorer has been available within Visual Studio for several versions now allowing connections to be made to SQL servers and various event logs. The latest version introduces a SharePoint connection. From the new Server Explorer you can see various elements within that web application. The intrinsic features include, list and galleries, sub sites, features and content types as shown in Figure 2.

Microsoft Intrinsic Nodes



CKSDev additional nodes:

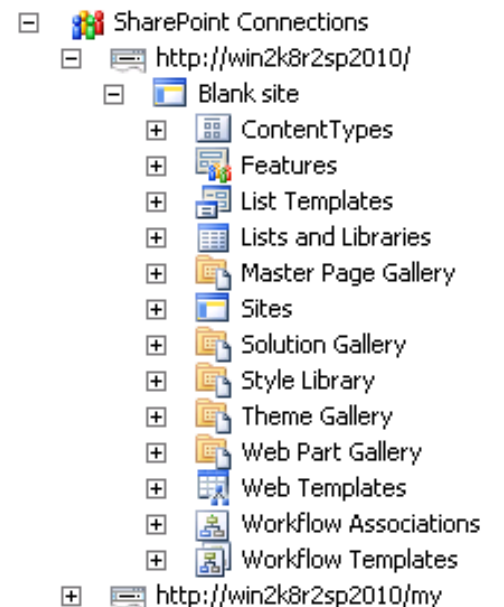


Figure 2: The intrinsic and CKSDev Server Explorer nodes comparison.

CKSDev introduces new nodes to the server explorer to enhance the existing nodes and introduce new information from the site. One of the new CKSDev delivered nodes is the Gallery nodes. Take for example the Master Page Gallery. Expanding this node within the explorer lists the page layouts and master pages within the gallery as shown in Figure 3.

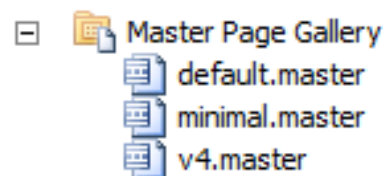


Figure 3: The master page gallery node.

Selecting a particular node provides the usual properties about that item but also new file IO operations. This allows you to interact with the file without leaving Visual Studio. As you would expect before you can make changes to the file it must be checked out to you. This is one of the options available from the item context menu.

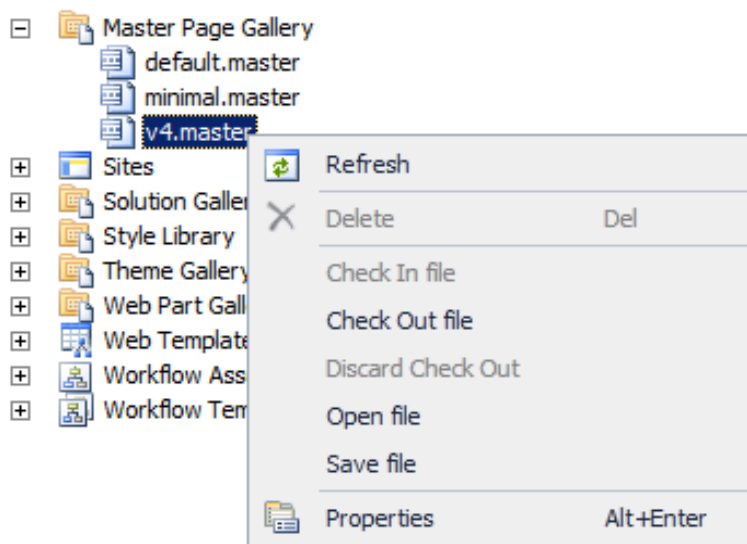


Figure 4: The file operations.

Once checked out the file can be opened directly into Visual Studio. The item will open in the main code window. Once open you can make the required changes then commit them back to the site via the item context menu. The opened document must be the active document window when this is invoked as it has not been brought into the solution structure at this point.

The CKSDev server explorer extensions also provide features which allow you to bring existing SharePoint elements from your site directly into your Visual Studio solution. From the context menu of the Content Type node the option to import the content type is now available.

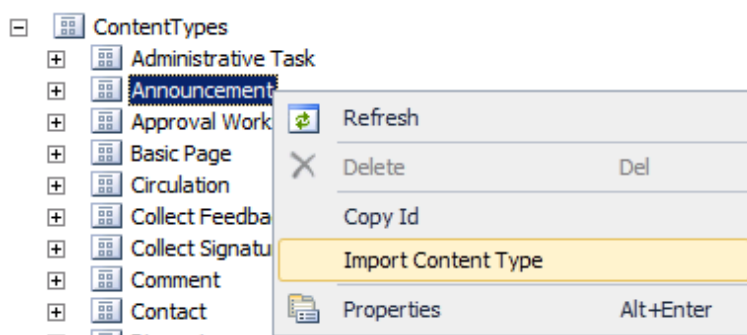


Figure 5: The import content type menu option.

This will import the content type and generate a SPI within the project for the content type. Should you select an OOB content type you are warned before import.

SharePoint 2010 introduces LINQ to SharePoint via a command line tool called SPMetal. CKSDev provides a generate SPMetal Definition feature from each site node. This menu option executes the SPMetal tool with the URL of the Site currently selected in the SharePoint Server Explorer. The generated definition is being displayed in Visual Studio so that you can easily add it to your project.

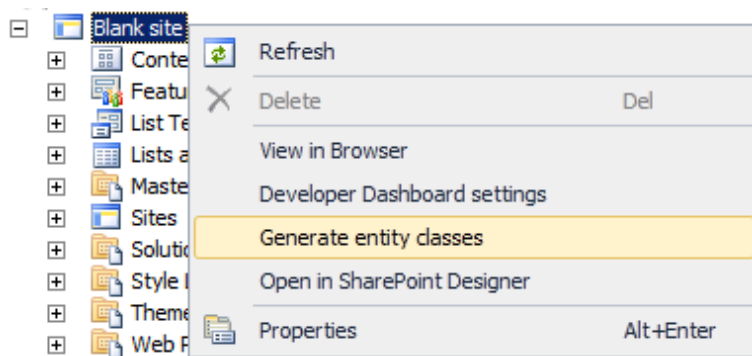


Figure 6: The generate SPMetal menu option.

Content

Visual Studio 2010 SharePoint tools are focused on the concept of a SharePoint Project Item (SPI) rather than a layout mimicking the SharePoint root folder structures. And the new Item and Project templates reflect this new project layout. CKSDev complements the existing SPI templates adding more of the commonly used SharePoint artifacts.

The SharePoint Console project creates a standard console application with the SharePoint assembly added and correctly targeted to the .Net 3.5 Framework. Useful for rapid prototyping the pre-configured console project removes the errors of .Net 4.0 targeting.

CKSDev has three new item templates for Custom Action Group, Custom Action and Hide Custom Action SharePoint artifacts. These item templates start with a visual wizard to aid the developer to complete the required attributes. Once the wizard is completed a new SPI is created using the supplied information saving time and reducing the need to hand craft the xml manifests as shown in Figure 7.



Figure 7: The item templates in the new item dialog.

These new SPI's can be packaged using the standard package and feature designers in the same fashion as the intrinsic versions as shown in Figure 8.

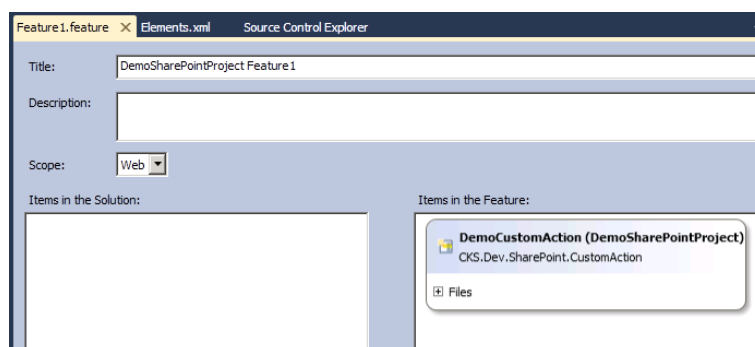


Figure 8: CKSDev custom action in the feature designer.

One of the most useful SharePoint features is the delegate control. SharePoint's user interface features many delegate controls providing a feature based mechanism to



override and replace user interface elements without the need to modify the source files. Based on IDs for the location and with a precedence to indicate the order in which they display. CKSDev has an SPI for delegate controls to aid the developer with the creation of these extremely useful elements. Capturing the basic control details and also the properties for its configuration and parsing these details into a fully formatted element manifest ready for deployment as shown in Figure 9.

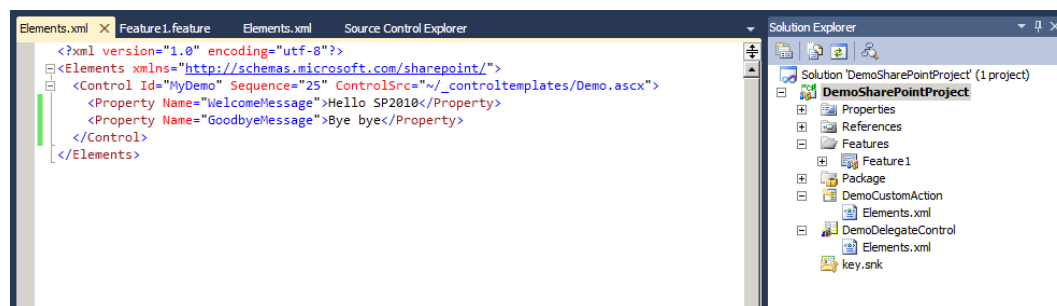


Figure 9: The delegate control SPI.

The new SharePoint sandbox features don't go unnoticed with CKSDev, the Sandboxed visual web part allows developers to still code and build the interface with the aid of an ASCX control and then deploy to the sandbox. Sandboxing removes the ability to host files on the file system which removes the standard Microsoft visual web part from the available templates. CKSDev uses supported techniques to compile the ASCX content into the assembly before the deployment to SharePoint giving you a visual design surface while still allowing a code deployment to the sandbox as shown in Figure 10.

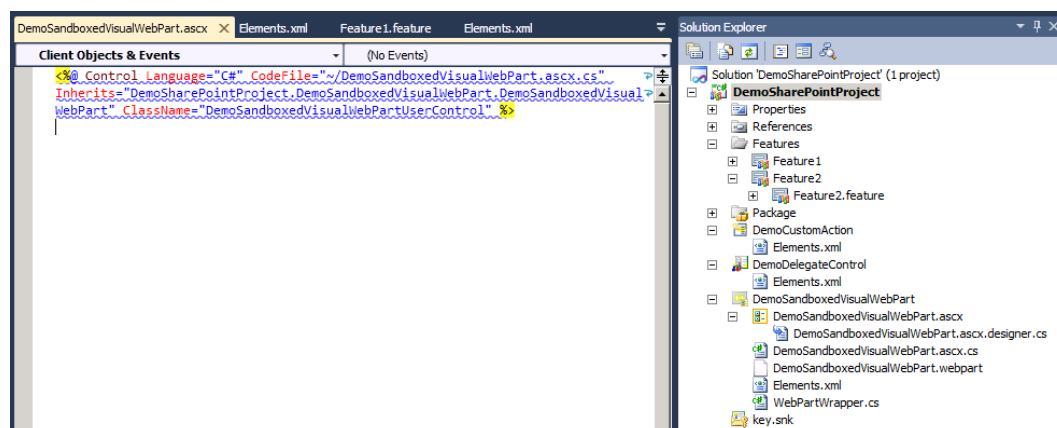


Figure 10: The visual sandbox web part SPI.

Deployment

A key change for SharePoint development has been the work around deployment from within Visual Studio. For SharePoint 2007 the available options from WSPBuilder and VSeWSS 1.3 CTP included Recycle App Pools, Quick Deploy/Copy to 12. These were great productivity boosters for a developer. How many times have you changed a few lines of html in an ascx file to then spend 2 to 3 minutes getting those changes into the SharePoint site. The goal behind Quick Deploy techniques is simple, to make this deployment as fast as possible for certain development scenarios, thus supporting the usual rapid iterative build-test-fix-repeat cycle that developers go through.

Developers familiar to the SharePoint Root/12 Hive folder structure within a SharePoint VS project will have noticed that the new tools deal in SharePoint Project Items which doesn't replicate the SharePoint Root structure within the project. The CKSDev quick deploy will deploy the changes without the need to build the entire package and it also honours those SPI's included in the package. Only SPI's within a package will be 'quick deployed'. Figure 11 shows the CKSDev deployment context menu.

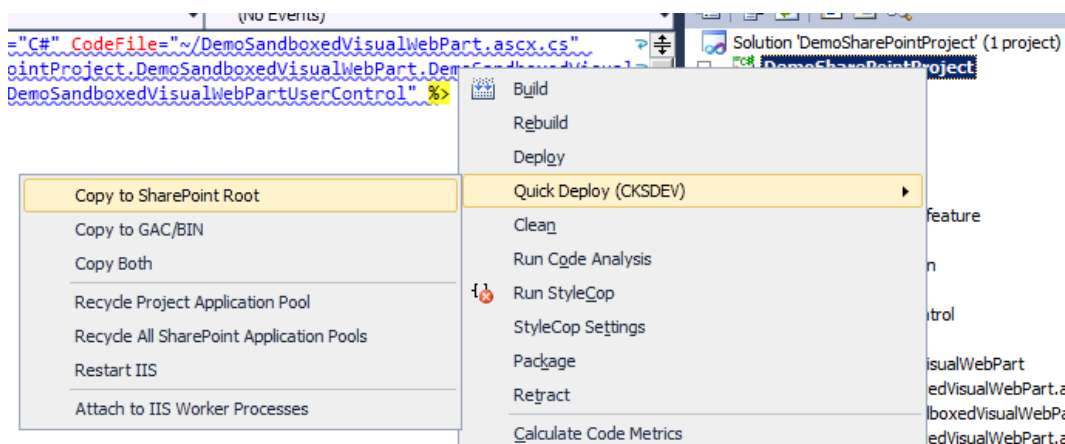


Figure 11: The CKSDev deployment context menu.

There are also options on the project to auto quick deploy so that when SPI's valid for quick deployment are saved they are automatically deployed to the SharePoint root as shown in Figure 12.

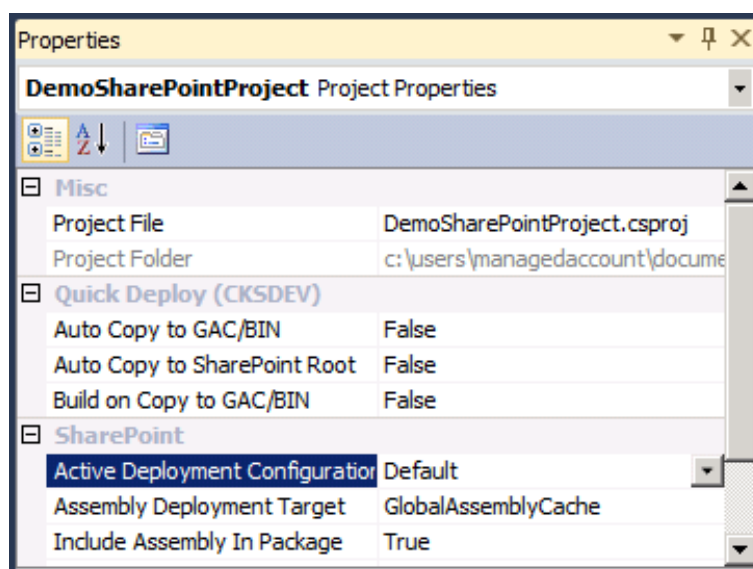


Figure 12: The deployment project properties.

Other deployment enhancements are the Run PowerShell script, warm up site, reset IIS, copy assemblies. Each provides a deployment step to improve the flexibility of the deployment configurations.

What's next?

The CKSDev project is aimed to assist SharePoint developers to be more productive. If you have ideas then please visit the CodePlex site and submit them in the discussion board or contact a team member.

Visit <http://cksdev.codeplex.com> for up to date information and documentation or follow us on Twitter #CKSDev

The first release of CKSDev will be available shortly in line with the SharePoint launch. ■



LAAT INFORMATIE VOOR JE WERKEN

Informatie wordt steeds crucialer voor een goede bedrijfsvoering. De eisen van medewerkers, klanten en partners veranderen voortdurend. Effectief en efficiënt (samen) werken vraagt om bedrijfsinformatie die altijd en overal toegankelijk en uitwisselbaar is. Getronics Consulting levert advies en technische oplossingen, van visie- en strategie-vorming tot implementatie en beheer, voor het optimaliseren van uw informatievoorziening



How to manage your Feature Lifecycle using Feature Versioning

by Anders Rask

Upgrading custom build artifacts in SharePoint 2007 was often a pain-point. The concept of Features was good if you wanted to deploy lists, content types, custom actions etc. but if you wanted to upgrade these artifacts you used to need to either re-activate your features or create new features with different names, since the feature manifest containing the new logic is only processed when a feature is activated.

The first approach would cause problems due to different deactivation behavior depending on the type of artifact. Often a lot of event receiver logic would have to be baked in to handle upgrade scenarios.

The second approach would get rid of the problems caused by deactivating features, but would often need to come with a lot of event receiver code.

The good news is that upgrading features has received some attention in the new version, so it is now possible to upgrade features both declaratively and programmatically. You can even declare branches for different actions depending on version, or have element manifests being applied at update only.

When we talk about features in SharePoint, it is important to distinguish between feature definitions and feature instances. A feature definition can be seen as a template for creating features. In the object model a feature definition is called `SPFeatureDefinition`. A feature instance is a feature that has been "instantiated" from a feature definition. It lives on either farm, web application, site or web scope. In the object model a feature instance is called `SPFeature`.

Declarative Feature upgrade

The declarative feature upgrade manifest contains an `UpgradeActions` element. Here you can declare actions that should only be applied for certain feature version ranges, including adding fields to content types, provision, move or rename files and more.

Currently the following elements can be placed inside an `UpgradeActions` element:

- `VersionRange`
- `CustomUpgradeAction`
- `ApplyElementManifests`
- `AddContentTypeField`
- `MapFile`

The optional `VersionRange` element gives you the opportunity to target feature upgrade to specific version ranges (e.g. between version 1.0.0.0 and version 1.2.0.0). The declarative logic specified inside the `VersionRange` elements will then only be executed if the version number is within the version range. This gives you the ability to branch upgrades with different behavior for each version.

`CustomUpgradeActions` contains actions and parameters for custom code that is referenced in the `UpgradeActions` element (see more below under programmatic feature upgrade). `CustomUpgradeActions` can both be placed inside the `UpgradeActions` element and inside the `VersionRange` element.

`ApplyElementManifests` is what you probably will use a lot when upgrading features: it will include an element manifest that is only triggered on upgrade. This makes it easy to add new artifacts to an existing feature. The element can be placed under `UpgradeActions` or `VersionRange` elements.



The optional `AddContentTypeField` makes it possible to easily add new fields to existing content types. By adding a `PushDown="TRUE"` attribute to the element the change is pushed down from the site content types to every list content type. This was really a pain to do both declaratively and in code in SharePoint 2007, so that's a really helpful change in SharePoint 2010.

`MapFile` can be used to move or rename files during feature upgrade.

Programmatic Feature upgrade

The changes are not only declarative. There are several changes to the object model regarding feature upgrades.

For once there's now a synchronous event called `FeatureUpgrading`. This gets fired for each matching `VersionRange` when a feature is upgraded. You can pass parameters to this event declaratively through the `CustomUpgradeAction` element.

New in SharePoint 2010 is also the Feature Upgrade Query Object Model. This can be used to identify what features are installed, what versions they have, and if they need to be upgraded. A `QueryFeature` method has been added to `SPSite`, `SPContentDatabase`, `SPWebApplication`, `SPWebService` and `SPAdministrationWebApplication` classes for this purpose.

`SPFeature` now has a `Version` property and an `Upgrade` method. SharePoint uses the `Version` property of the `SPFeature` to determine if a given activated feature needs to be upgraded by comparing it with the `SPFeatureDefinition` version.

Upgrading Features

So what triggers a feature upgrade? First it is important to understand when a feature doesn't upgrade: When we upload a new version of our feature definition by upgrading our solution, the feature definition is upgraded to the new version. The existing feature instances that are activated throughout the farm aren't upgraded yet though. SharePoint keeps track of feature definition and instance versions, and is able to determine what features are out of sync with the feature definitions, but it will not upgrade anything automatically. Let's say you now activate the feature somewhere in your farm. The feature will be based on the new feature definition, and hence be "up to date" (as in not needing to be upgraded). But activating the feature in this way will only trigger the "usual" event handlers `FeatureInstalled` and `FeatureActivated`, not any upgrade logic.

What this means is, that you should still take care of that new feature instances based on new versions will have the appropriate declarative and programmatic logic to make them identical to feature instances that have been upgraded from earlier versions. This is something that will surely confuse a lot of people and I'm sure we will see a lot of features that will be left in different upgrade stages.

So how do you upgrade features?

There is actually little out-of-the-box support for this: If you build a sandboxed solution (as opposed to a farm solution) your solution will be deployed to the solution gallery in a site collection. In the solution gallery an upgrade button is shown that will trigger a feature upgrade, but there is no support for upgrading features in farm solutions, apart from doing a B2B in-place upgrade (`psconfig.exe -cmd upgrade -inplace b2b`) which should be considered slightly overkill. In Visual Studio the story is the same: no tools exist at the moment for upgrading features, but since the tools are easy to extend, we will probably see this integration soon.

Because of the above, you will have to activate the feature yourself by calling the `Upgrade` method of `SPFeature`.

Throughout this article PowerShell will be used for both solution deployment and feature upgrade. It is of course also a possibility to use C# code to create application pages, console applications or similar to trigger feature upgrades.



An example of activating features through code can be seen below:

```
# get site context
$sites = Get-SPSite http://<your_site>
# get a collection of features that need upgrade
$qfr=$sites.QueryFeatures([Microsoft.SharePoint.SPFeatureScope]::Site,$true)
# iterate features and upgrade.
# note that if features contain dependencies to other features
# you might need to upgrade features in a specific order
foreach($feature in $queryResults)
{
    write-host $feature.Version;
    $feature.Upgrade($false);
    write-host $feature.Version;
}
```

Listing 1: Activating features through PowerShell code.

So what are the best practices for using versions when creating features? A catch here is, that Visual Studio doesn't actually give a feature a version to begin with. You will have to do it yourself. Below are some practices for when to increment versions.

Good practices for versioning Features

You should always increase your feature version number and if necessary add declarative or programmatic upgrade logic when changing a feature. This way you have better control of what version is deployed where in your farm.

Also don't follow the SharePoint product versions. Many of the 3rd party tools today actually uses v.12.0.0.0 as "default" for SPFeatureDefinitions. Instead you should follow your own product lifecycle when versioning features.

The version is a standard System.Version that follows the following logic:

major.minor[.build[.revision]]

A major version could follow your deployment cycle, eg. 3.0.0.0 when launching a big deployment. For patches and bugfixes increment build version, eg. 3.0.1.0. Minor versions should be changed when releasing service packs, eg. 3.1.2.0.

First Task: getting the basics in place

To demonstrate how feature upgrade could work in a real-life scenario, let's assume you have been given the task to create a product list based on a custom list. This list will have two extra fields Product description and Product Details.

To achieve this task we will create 4 artifacts:

- Two site columns
- A site content type ProductContentType referencing the site columns (inherits from Contact)
- A list definition ProductListDefinition (using ProductContentType)
- A list instance ProductListInstance (using ProductListDefinition)

For this task we will use Visual Studio tools for SharePoint 2010. In this article I will assume the reader has some knowledge about how the tools work for creating SharePoint artifacts, but where appropriate I will elaborate a bit more. The Visual Studio solution will be available for download from my blog <http://andersrask.spoint.me>.

First create an empty SharePoint project called ProductList and add an empty element called ProductList; make it a Farm solution. Rename elements.xml to fields.xml. Now rename the feature name and title to ProductSiteColumns.



Now define two site columns in fields.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<Elements xmlns="http://schemas.microsoft.com/sharepoint/">
  <Field
    ID="{19604698-384D-4a75-8121-0EEE2B04C391}"
    Name="ProductDescription"
    StaticName="ProductDescription"
    DisplayName="Product Description"
    Type="Text" />
  <Field
    ID="{18722C6E-B131-407e-AA05-11A0D0496397}"
    Name="ProuctDetails"
    StaticName="ProuctDetails"
    DisplayName="Product Details"
    Type="Note" />
</Elements>
```

Listing 2: Site columns manifest.

Double click your feature definition to set the version property to 1.0.0.0 .

The feature should contain only your ProductSiteColumns and be Site scoped.

Now add another item called ProductContentType based on the Content Type template. Base it on the Item content type.

Create a new feature ProductContentTypes and set scope to Site. Set version property to 1.0.0.0 .

```
<?xml version="1.0" encoding="utf-8"?>
<Elements xmlns="http://schemas.microsoft.com/sharepoint/">
  <!-- Parent ContentType: Item (0x01) -->
  <ContentType ID="0x0100133b496536644858b88af6114c988f09"
    Name="ProductContentType"
    Group="Custom Content Types"
    Description="Product Content Type"
    Inherits="TRUE"
    Version="0">

    <FieldRefs>
      <FieldRef
        ID="{19604698-384D-4a75-8121-0EEE2B04C391}"
        Name="ProductDescription"/>
      <FieldRef
        ID="{18722C6E-B131-407e-AA05-11A0D0496397}"
        Name="ProuctDetails"/>
      <FieldRef
        ID="{ED66AD6E-77F3-471c-957E-67CBFCD07A04}"
        Name="ProductExpirationDate"/>
    </FieldRefs>
  </ContentType>
</Elements>
```

Listing 3: The content type manifest.

Now create a new item EmployeeList, based on Empty SharePoint template. Add a new item ProductListDefinition based on the List Definition from Content Type template. Make sure you use the ProductContentType for the list definition, and make sure you have checked Add a list instance to this list definition.

Split up the two artifacts into separate feature definitions, both with Web scope. Set the version property to 1.0.0.0 for both features.

For the sake of simplicity in this example I am not defining any dependencies between features. It is a good practice however to define activation dependencies between features that depend on each other. For example a content type feature is dependent on the site column feature, so you would typically hide the site column feature and define a feature dependency in the content type feature to ensure site columns get activated before the content type is activated. For activation dependencies in Share-Point 2010 you can define a MinimumVersion attribute to indicate from what version of the parent feature the feature depend upon.



In SharePoint 2010 we also can define dependencies between solutions. Even though solutions are not “activated”, it is still called activation dependencies and can be defined in the solution manifest. Three important things to note regarding solution dependencies though: solution dependencies do not automatically secure that dependent solutions are deployed. They just give you an error if you try to activate a solution that is dependent on another solution, and that solution isn’t deployed. Also you cannot have a farm based solution that is dependent on a user solution (Sandboxed solution). Last but not least: you will not receive any errors if you try and retract a solution that another solution is dependent on!

Update the list definition manifest with your own type id, display name, name and description:

```
<?xml version="1.0" encoding="utf-8"?>
<Elements xmlns="http://schemas.microsoft.com/sharepoint/">
  <ListTemplate
    Name="ProductListDefinition"
    Type="60010"
    BaseType="0"
    OnQuickLaunch="TRUE"
    SecurityBits="11"
    Sequence="410"
    DisplayName="ProductListDefinition"
    Description="Product List Definition"
    Image="/_layouts/images/itgen.png"/>
</Elements>
```

Listing 4: List definition manifest.

Also update the list instance manifest to point to the list definitions type id, and give it a title, url and description:

```
<?xml version="1.0" encoding="utf-8"?>
<Elements xmlns="http://schemas.microsoft.com/sharepoint/">
  <ListInstance Title="Product List"
    OnQuickLaunch="TRUE"
    TemplateType="60010"
    FeatureId="c3f4896b-76e9-411e-a4f0-224fe0e3aa40"
    Url="Lists/Products"
    Description="Products">
  </ListInstance>
</Elements>
```

Listing 5: List definition manifest.

The project should look something like this:

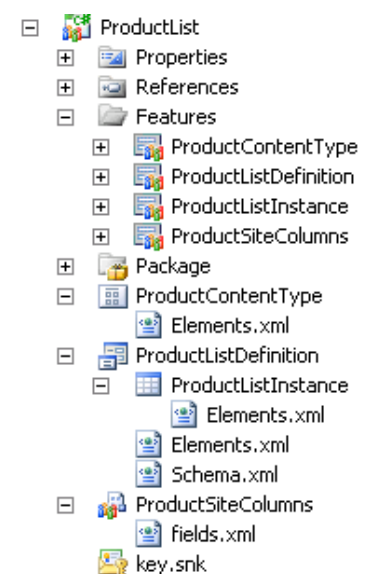


Figure 1: ProductList project.



You should now deploy these 3 solutions to your farm. This could either be done using Deploy from Visual Studio, or manually using PowerShell.

If you haven't already started looking into PowerShell, you should do so! STSADM.EXE is still around for legacy reasons, but when you have worked for some time with PowerShell trust me, you don't want to go back to pesky batch files!

Using PowerShell from the command line means you will have to do a lot of typing. There are already a lot of PowerShell cmdlets out there to make your job easier. For the reason of simplicity I will show how to add and install solutions directly from the SharePoint 2010 Management Shell (aka PowerShell).

Start by packaging your solution: right click your project and select Package. To add the package to the solution store, start up the SharePoint 2010 Management Shell and go to where the package was built (bin/debug or bin/release under your project, depending on your build configuration).

In the example below I am adding and deploying the ProductList solution:

```
# we have to make the path absolute
$solutionPath = Resolve-Path(".\ProductList.wsp")
# add solution to store
Add-SPSolution -LiteralPath $solutionPath
# deploy solution
Install-SPSolution -Identity ProductList.wsp
```

Listing 6: Adding a solution with PowerShell.

Check in *Central Administration > System Settings > Manage Farm Solutions* that the deployment went ok.

Activate your site collection (Site) scoped features first, then go to a site and activate the site (Web) scoped features –or activate them through PowerShell (the command is `Enable-SPFeature -identity <featureName> [-url <siteurl>]`). Make sure to activate the features in the right order, if you haven't defined activation dependencies between your features: first your site columns, then your content type, then your list definition, then your list instance.

Now you should have a working Product List that you can add data to.

Second task: upgrade using declarative logic

As always, things change. Your customer tries out the new list and determines that they need to do a few changes. Specifically they need a new DateTime field called `ProductExpirationDate`.

To achieve this, we need to add a new manifest to our `ProductSiteColumns`, called `fields2.xml`:

```
<?xml version="1.0" encoding="utf-8"?>
<Elements xmlns="http://schemas.microsoft.com/sharepoint/">
  <Field
    ID="{ED66AD6E-77F3-471c-957E-67CBFCD07A04}"
    Name="ProductExpirationDate"
    StaticName="ProductExpirationDate"
    DisplayName="Product Expiration Date"
    Type="DateTime"
  />
</Elements>
```

Listing 7: The new site column manifest.

Remember to set the deployment type to `ElementManifest` to make Visual Studio 2010 include this manifest automatically in the feature manifest.

We separate the two manifests so that we can distinguish between a new feature instance being created, and an old feature instance being upgraded.

Now we need to add some declarative logic to the `ProductSiteColumns` feature that will take care of both new feature instances and upgraded feature instances. When we use the Feature Designer in Visual Studio 2010 it takes care of updating the Feature



manifest file when we change our artifacts. We do however have the possibility to go from designer view to manifest view, by clicking the Manifest button:

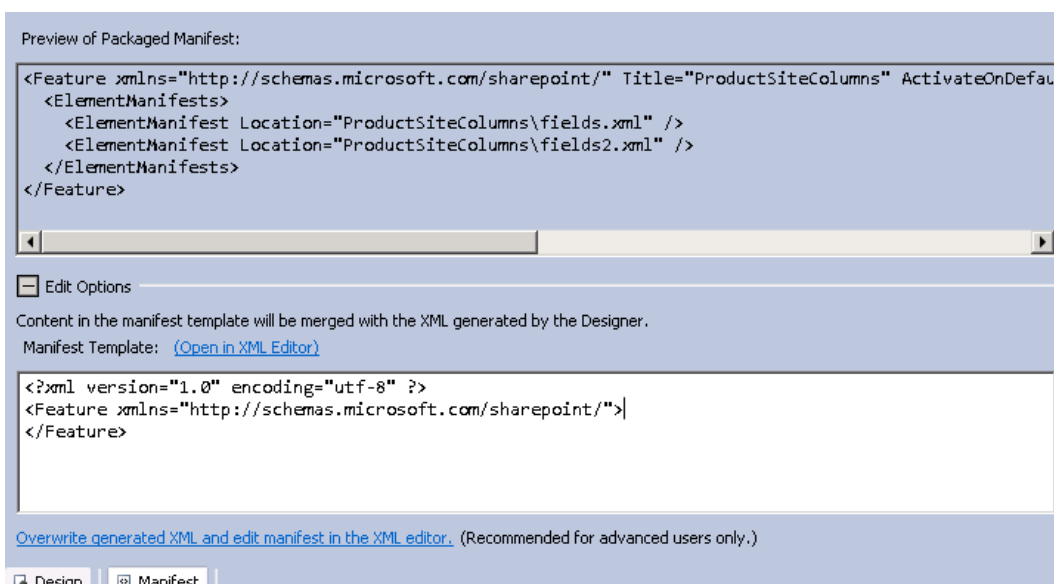


Figure 2: Previewing Site Columns manifest.

If we click on Edit Options we get a small window where we can edit the manifest template. The manifest template is an XML file that Visual Studio 2010 will automatically merge with our feature manifest when we package our solution. To make editing easier either click on Open in XML Editor or simply open the ProductSiteColumns.Template.xml file in Solution Explorer:

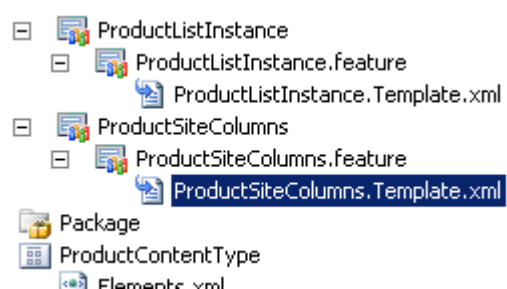


Figure 3: ProductSiteColumns.Template.xml.

Visual Studio will take care of the declarative logic for adding the new element manifest, but we need to add upgrade code ourselves:

```
<?xml version="1.0" encoding="utf-8" ?>
<Feature xmlns="http://schemas.microsoft.com/sharepoint/">
  <UpgradeActions>
    <VersionRange EndVersion="1.0.1.0">
      <ApplyElementManifests>
        <ElementManifest Location="ProductSiteColumns\fields2.xml" />
      </ApplyElementManifests>
    </VersionRange>
  </UpgradeActions>
</Feature>
```

Listing 8: Declarative upgrade code for the new site column.

Notice the VersionRange element inside the UpgradeActions element. This will ensure that when the upgrade is triggered, the current version of the feature instance will be matched against the version ranges defined in the feature manifest.

What happens inside the version range? The ApplyElementManifests element will ensure that the provisioning code that is already defined in the ElementManifests element in the feature manifest will also be triggered for an upgrade. Remember that feature manifest are only triggered once, and that is when the feature is activated.



So to make sure that the already instantiated features get up to date, the manifest is also provisioned when the feature is upgraded.

Finally increment the version to 1.0.1.0 for the ProductSiteColumns solution.

We now need to add the new site column to our content type feature. This is done in a similar fashion:

Edit the ProductContentType.Template.xml and add the following declarative upgrade logic:

```
<?xml version="1.0" encoding="utf-8" ?>
<Feature xmlns="http://schemas.microsoft.com/sharepoint/">
  <UpgradeActions>
    <VersionRange EndVersion="1.0.1.0">
      <AddContentTypeField
        ContentTypeId="0x0100133b496536644858b88af6114c988f09"
        FieldId="{ED66AD6E-77F3-471c-957E-67CBFCD07A04}"
        PushDown="TRUE" />
    </VersionRange>
  </UpgradeActions>
</Feature>
```

Listing 9: Declarative upgrade code for the new site column.

Again we use the UpgradeActions and VersionRange element, but to add the new site column we use the AddContentTypeField that points to both content type and field. Notice the PushDown="True" attribute.

This ensures that the change is propagated down to all the list content type instances of the site content type. This was something that in SharePoint 2007 required a lot of effort, and a lot of custom code to get done right, so this is a warmly welcomed new feature!

At the end increment feature version to 1.0.1.0 for your content type feature.

Now you need to package and deploy your two upgraded solutions. From the SharePoint 2010 Management Console this could be done like this:

```
# we have to make the path absolute
$solutionPath = Resolve-Path(".\ProductList.wsp")
# update solution. Note you don't need to trigger the upgrade job like you
have to do with STSADM
Update-SPSolution -Identity ProductList.wsp -LiteralPath $solutionPath
```

Listing 10: Upgrading a solution with PowerShell.

Now we have upgraded the feature definitions. We can check the version of our definitions in the site collection, again using PowerShell:

```
Get-SPFeature "<featureId>" | select Version
# Output:
Version
-----
1.0.1.0
```

Listing 11: Checking feature definition version with PowerShell.

If you want you can now try and activate the list definition and instance features on another sub site and check that the new features being instantiated contain the new field. Also validate that the list already instantiated does not contain the new field. This is expected.

We will now use PowerShell to query the site collection to get a collection of features that need to be upgraded, and upgrade the features in a loop:



```
$sites = Get-SPSite http://<siteurl>
# $true indicates that it should return only features that need upgrade
# return value is a SPFeatureQueryResultCollection containing SPFeature's
$qr = $sites.QueryFeatures([System.Guid]"<featureId>", $true)
# iterate all features and upgrade
foreach($feature in $queryResults)
{
    write-host $feature.Version;
    # bool parameter indicates if an upgrade should be forced
    # if force is anything like in SharePoint 2007 it should be avoided
    # if at all possible (basically it swallows exceptions, including
    # any errors you might have in your code...)
    $feature.Upgrade($false);
    write-host $feature.Version;
}
```

Listing 12: Get a collection of features needing upgrade and upgrade using PowerShell.

Do this for both features, starting with the site columns feature, then verify that your new field is present in the content type and in the list instances.

So what happened here? SharePoint checked the instantiated features version number against the feature definition, and determined that your features needed upgrade.

Calling the upgrade method on SPFeature, triggers the declarative upgrade actions to be executed. SharePoint again checks what version ranges matches the current version of the feature and executes the logic inside one or more version range elements.

About DIWUG

DIWUG is a platform for people that are interested in information worker solutions. Several times a year, DIWUG organizes events where members can meet, share knowledge and see interesting presentations in an informal setting.



The DIWUG SharePoint eMagazine is a free downloadable magazine written by MCM's, MVP's and other well known people from the SharePoint community. The target audience are IT-pros, developers and end (power) users. All articles are in English.

We're looking for authors!

Of course we cannot do this without content. We're looking for articles on SharePoint (2007/2010). These articles can be for developers, (power) end users or IT-pros. If you'd be interested to write an article, please let us know by emailing us a brief summary of the topic you want to write about. If English is not your native language, don't hesitate to contribute. We will help you with the English spelling and grammar.

... and sponsors

If you're interested in sponsoring this eMagazine please let us know and we'll send you the sponsor possibilities.

Contact information:

Marianne van Wanrooij; marianne@information-worker.org

Mirjam van Olst; mirjam.van.olst@macaw.nl

Mart Muller; mart@diwug.nl

<http://www.diwug.nl>



Third task: Programmatic upgrade

Well you know customers. Once the above enhancements were provisioned, they have another feature request for you.

The customer has decided that the Employment Date should only show date, and not time. To do this you will have to change the format of the field defined in your site columns to be DateOnly -no big deal! But again you need to think about all the existing instances of your code. The solution is to attach a feature receiver to your list instance feature, and add some logic in the new FeatureUpgrading event that handles the above requests for all instantiated lists.

Event receivers are basically .NET code that is executed when a certain event happens. This event, for features, could be when it was activated, installed, uninstalled or deactivated –and now in SharePoint 2010 when a feature is upgraded. You can use the manifest to declare upgrade actions and send custom parameters with those actions, into the upgrade event.

You create an event receiver by right-clicking your feature and selecting Add Event Receiver. This will include the code file and add the necessary wiring inside the feature.

Also remember to set the project property Include Assembly In Package to True when you are using code in your project.

Add the following code to your ProductListDefinitionEventReceiver:

```

public override void FeatureUpgrading(
    SPFeatureReceiverProperties properties, string upgradeActionName,
    System.Collections.Generic.
IDictionary<string, string> parameters)
{
    // get the feature to which the receiver properties apply
    SPFeature feature = properties.Feature;
    SPWeb parentSite = feature.Parent as SPWeb;
    if (parentSite == null)
    {
        throw new Microsoft.SharePoint.Upgrade.SPUpgradeException(
            "Feature does not have web scope");
    }
    // check what upgrade action was called and act accordingly
    switch (upgradeActionName)
    {
        case "V2VUpgrade":
            DoV2VUpgrade(parentSite, feature, parameters);
            break;
        // add other action names here later
        default:
            break;
    }
}

/// <remarks>Note: This code is boilerplate code only! Add proper null
/// checks and exception handling.</remarks>
private void DoV2VUpgrade(SPWeb parentSite, SPFeature feature,
    System.Collections.Generic.IDictionary<string, string> parameters)
{
    foreach (SPList list in parentSite.Lists)
    {
        // fetch lists instantiated by this feature
        if (feature.DefinitionId == list.TemplateFeatureId)
        {
            // here we check the parameters parsed from manifest
            if (parameters.ContainsKey("FieldName"))
            {
                string fieldName = parameters["FieldName"];
                // set date field to DateOnly
                SPFieldDateTime field = (SPFieldDateTime)list.Fields[fieldName];
                field.DisplayFormat = SPDateTimeFieldFormatType.DateOnly;
                field.Update();
            }
        }
    }
}

```

Listing 13: C# event handler code to provision changes on existing list instances.



Now you need to add upgrade code to the manifest:

```
<?xml version="1.0" encoding="utf-8" ?>
<Feature xmlns="http://schemas.microsoft.com/sharepoint/">
  <UpgradeActions>
    <VersionRange BeginVersion="1.0.0.0" EndVersion="1.0.2.0">
      <CustomUpgradeAction Name="V2VUpgrade">
        <Parameters>
          <Parameter Name="ViewName">All Items</Parameter>
          <Parameter Name="FieldName">Product Expiration Date</Parameter>
        </Parameters>
      </CustomUpgradeAction>
    </VersionRange>
  </UpgradeActions>
</Feature>
```

Listing 14: Declarative upgrade code for the event handler.

Now change the version on the list definition feature to 1.0.2.0.

To make the behavior for upgrades and normal provisioning the same we must now change the field2.xml definition to contain the format setting. In this example we can do this declaratively, but if there is no declarative way of doing what you want to do at feature activation, you can add code to the FeatureActivated event. Often it would be a good practice to re-use the methods defined in FeatureUpgrade in your FeatureActivated events to ensure the logic is equal.

```
<?xml version="1.0" encoding="utf-8"?>
<Elements xmlns="http://schemas.microsoft.com/sharepoint/">
  <Field
    ID="{ED66AD6E-77F3-471c-957E-67CBFCD07A04}"
    Name="ProductExpirationDate"
    StaticName="ProductExpirationDate"
    DisplayName="Product Expiration Date"
    Type="DateTime"
    Format="DateOnly"
  />
</Elements>
```

Listing 15: fields2.xml code.

Update the version of the site columns feature to 1.0.2.0, both in the version range of the ProductSiteColumns.Template.xml and version property on feature.

Now we are ready to upgrade our two updated features again using PowerShell.

Tip: If you have C# code that is executed during upgrade, you can attach the PowerShell process in Visual Studio and apply break-points in your code. You will then be able to step through the code as it executes.

When the update has completed verify that the date field inside the instantiated lists contain the new format.

Summary

We have taken a lap around feature upgrade and seen how we can use this to manage our feature lifecycle in SharePoint 2010 in a supported way. Feature upgrade should rank very high on your list of favorite additions to SharePoint 2010, especially if you are a developer. Considering this it would be nice to have more support for upgrading features in Visual Studio and in the SharePoint user interface. Hopefully the community will step up soon to build support for feature upgrades in PowerShell or Visual Studio, as this would definitely help increase user adoption.

If you want more info on this subject, visit MSDN Library

[http://msdn.microsoft.com/en-us/library/aa544511\(office.14\).aspx](http://msdn.microsoft.com/en-us/library/aa544511(office.14).aspx). ■



SHAREPOINT TRAINING

IDEVTEAM PROVIDES EXPERT-LED, HANDS ON, SMALL CLASS SIZE AND FOCUSED SHAREPOINT TRAINING, AND COACHING SERVICES FOR DEVELOPERS, IT PROFESSIONALS AND END USERS. WE ONLY WORK WITH MICROSOFT CERTIFIED TRAINERS AND MICROSOFT CERTIFIED TECHNOLOGY SPECIALISTS. OUR COURSES ARE DELIVERED ON-SCHEDULE IN THE NETHERLANDS AND ON-SITE IN EUROPE AND MIDDLE-EAST.

SHAREPOINT CONSULTING

WITH YEARS OF SHAREPOINT TRAINING EXPERTISE AND ENTERPRISE SHAREPOINT CONSULTING EXPERIENCE, OUR TRAINERS AND CONSULTANTS PROVIDE A ROBUST PORTFOLIO OF COST-EFFECTIVE SHAREPOINT TRAINING AND CONSULTING SERVICES. AS A MICROSOFT PARTNER OUR GOAL IS TO DELIVER HIGH QUALITY TRAINING AND CONSULTING SERVICES IN THE SHAREPOINT AREA.

FOR MORE INFORMATION VISIT WWW.IDEVTEAM.NL OR CONTACT INFO@IDEVTEAM.NL

YES, WE'RE HIRING!

Ready for your next big step?
SharePoint & Freelance?

<http://join.idevteam.nl>





Architecting Service Applications

by Octavie van Haaften

With MOSS 2007 there was this thing called Shared Service Provider (SSP). This SSP provided you Search, User Profiles, Excel Services and the Business Data Catalog all together. The SSP had its own web application and database. Now with SharePoint Server 2010 there is no SSP, but it is split up in service applications. Most service applications have one or more databases and each service application can have its own application pool and can be associated with one or more web applications.

In this article I want to show you how this change of concept affects your architecture.

Overview

SharePoint 2010 offers lots of service applications out of the box. I'll start with a short overview of a few service applications.

Business Data Connectivity Service

With this service you can read data from or write data to external systems like SQL Server databases, SAP or Web Services. With MOSS2007 you had to assemble an XML definition file for your BDC application. Now, you can use the SharePoint Designer to create your application definition.

Managed Metadata Service

With this service you can manage metadata independently of your items. This is done by means of terms and term sets. These terms are then associated with your items. Content types can now be used across webs, site collections and farms!

Search and Search Administration Service

With this service you can crawl and index your content. With the Search functionality in your site users can query the content. SharePoint 2010 provides new capabilities like suggestions, ranking and relevance.

User Profile Service

With this service you can synchronize profiles with a directory service or business system like SAP. You can use these profiles for your social computing features like My Sites, social tagging and People Search.

PerformancePoint Services

With this service you can monitor and analyze your business by providing tools for creating dashboards, reports, scorecards and KPI's. Microsoft Office PerformancePoint 2007 was a standalone server product, but it is now integrated with SharePoint 2010 as a service.

Other services are: Web Analytics, Secure Store, Usage and Health Data Collection, State, Security Token Service, Word Automation Services, Excel Services, Access Services. As you can see there are a lot more services than in MOSS2007. Besides these standard service applications you can develop your own service application for SharePoint 2010. But that's a whole other topic.

The Benefits

With all these separate service applications you have the flexibility to install and use whatever you need. Web applications connect to service applications through service application proxies (figure 1). It is now possible that a service application can serve multiple web applications. It is also possible to have multiple instances of a service application. Each instance can serve a different web application. This ensures complete isolation and dedication.



Name	Type
Application Discovery and Load Balancer Service Application	Application Discovery and Load Balancer Service Application
Application Discovery and Load Balancer Service Application Proxy_d174277f-2c29-4be4-b62d-940b5b54fd12	Application Discovery and Load Balancer Service Application Proxy
Application Registry Service	Application Registry Service
Application Registry Service	Application Registry Proxy
Business Data Connectivity	Business Data Connectivity Service Application
Business Data Connectivity	Business Data Connectivity Proxy
Search Administration Web Service for Search Service Application	Search Administration Web Service Application
Search Service Application	Search Service Application
Search Service Application	Search Service Application Proxy

Figure 1: Overview Service Applications and Proxies.

The way Microsoft has developed these service applications makes SharePoint 2010 more flexible and scalable than its predecessor MOSS 2007. However, one thing to consider is storage. Where the old SSP had basically one database for everything, you will now end up with lots of databases. Most service applications have one or more databases. When installing 10 service applications you can have more than 10 databases. This is something to consider when planning the SQL environment and also backup/restore scenarios.

Logical Architecture

In this part I will cover some logical designs for both single farm and multiple farms. These logical designs consist of web applications and application pools.

Single Farm

Web applications can be configured to use those services that are needed instead of all services that are deployed. Connections can be managed individually or in groups. These are the service application proxy groups. By default, all proxies are part of the Default group (figure 2).

Service Application Connections
Choose the service applications that this Web application will be connected to. A Web application can be connected to the default set of service applications or to a custom set of service applications. You can change the set of service applications that a Web application is connected to at any time by using the Configure service application associations page in Central Administration.

Edit the following group of connections: **default**

Name	Type
☒ Business Data Connectivity Service Application	Business Data Connectivity Proxy
☒ Managed Metadata Service Application	Managed Metadata Service Connection
☒ Search Service Application	Search Service Application Proxy
☒ Secure Store Service	Secure Store Service Application Proxy
☒ State Service	State Service Proxy
☒ WSS_UsageApplication	Usage and Health Data Collection Proxy

Figure 2: Default Group Service Applications.

In the previous figure all service applications serve both web applications by means of the default group. But what if web application B doesn't need Search? Then you will need a custom group (figure 3).

... BY THE WAY ...

Next issue September 2010

You are invited to send us your presentation. See About DIWUG at [page 41](#) for further details.



Service Application Connections

Choose the service applications that this Web application will be connected to. A Web application can be connected to the default set of service applications or to a custom set of service applications. You can change the set of service applications that a Web application is connected to at any time by using the Configure service application associations page in Central Administration.

Edit the following group of connections: [custom]

Name	Type
☒ Business Data Connectivity Service Application	Business Data Connectivity Proxy
☒ Managed Metadata Service Application	Managed Metadata Service Connection
☐ Search Service Application	Search Service Application Proxy
☐ Secure Store Service	Secure Store Service Application Proxy
☐ State Service	State Service Proxy
☐ WSS_UsageApplication	Usage and Health Data Collection Proxy

Figure 3: Custom Group Service Applications.

When creating web applications you have the choice to use the default group of create a custom group and select those services you need. It is not possible to create all groups first and then associate a group with a web application. However, you can have an overview of all web applications and their group associations. In Central Administration select Application Management and then Configure Service Applications Associations (figure 4).

Site Actions			
Give Feedback System Account			
Microsoft SharePoint 2010 Central Administration > Service Application Associations			
This page controls the web application or service application to proxy group associations.			
View: Web Applications			
Central Administration Application Management System Settings Monitoring Backup and Restore Security Upgrade and Migration General Application Settings Configuration Wizards	Web Application / Service Application	Application Proxy Group	Application Proxies
	SharePoint - PublishingPortal80 (http://publishingportal/)	custom	Business Data Connectivity Service Application Managed Metadata Service Application State Service
	SharePoint - 80 (http://winsrv2008r2/)	default	Business Data Connectivity Service Application Managed Metadata Service Application WSS_UsageApplication Search Service Application Secure Store Service State Service
	SharePoint - CollaborationPortal80 (http://collaborationportal/)	custom	Managed Metadata Service Application Search Service Application

Figure 4: Service Application Associations.

You can modify groups associations by just clicking on the group.

It's easy to get lost on all the web applications and their associated service applications. Before installing and configuring all the service applications, take some time to draw a picture for this logical design.

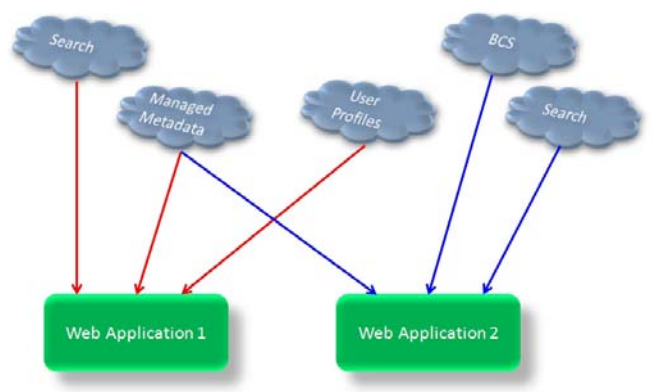


Figure 5: Service Applications Design.



As you can see a service application can be connected to multiple web applications. It is also possible to have multiple instances of a service application that are connected to one web application. For example, the Managed Metadata service application is set up with a Site Collection Content Type Hub. The content types are published to the connected web application. You could have a second instance of the Managed Metadata service application that is set up with another Site Collection Content Type Hub. When connecting to the same web application, it now uses published content types from both hubs.

Multiple Farms

Let's throw in some complexity. How difficult will it be to share data across multiple farms? Not that difficult. Service applications are WCF services and web applications connect through proxies over http/https. All you have to do is publish the service application and make it available for other server farms (figure 6).

Figure 6: Publishing a Service Application.

In the other server farm you connect the web application by using the published URL (figure 7).

Figure 7: Connecting to a remote Service Application.



In this way you could have an Enterprise Service Farm servicing all other farms. Figure 8 shows an example design where Farm A contains all the service applications that are shared with Farms B and C.

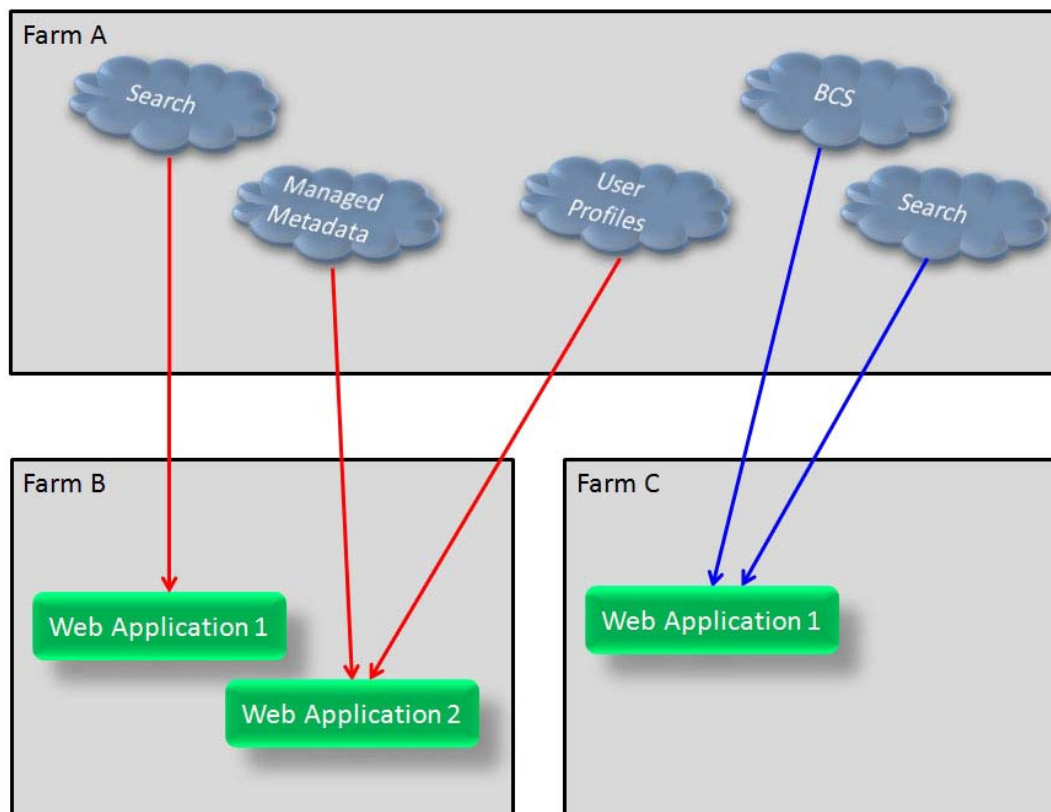


Figure 8: Cross-Farm Service Applications Design.

Great isn't it? But... (there is always a but). Not all service applications can be shared across server farms. There are service applications that can only be shared within a single server farm such as Excel Services. However, you can have both local and published service applications in your group that is associated with the web application (figure 9).

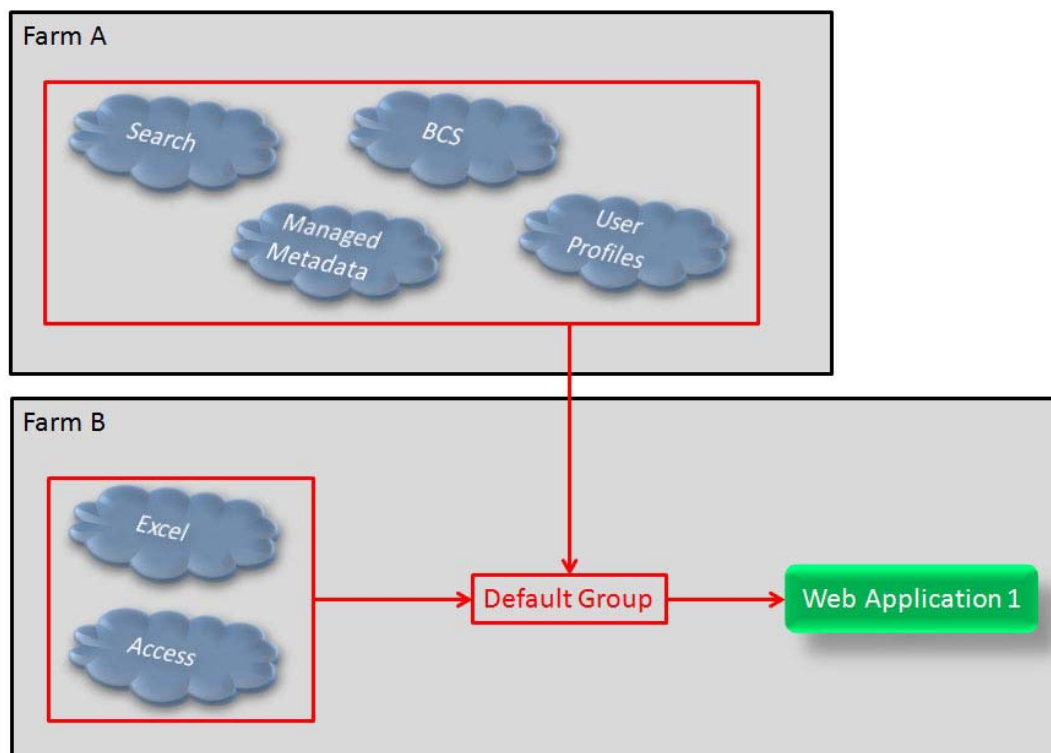


Figure 9: Cross and Single Farm Service Applications Group.



Here's a list of services that can be shared across server farms:

- Search
- User Profiles
- Managed Metadata
- Business Connectivity Services
- Web Analytics
- Secure Store Services

These cannot:

- Access Services
- Excel Services
- Word Automation
- Word Viewer
- Visio Graphics
- State Services
- Usage and Health
- PerformancePoint
- Project Server

Physical Architecture

Until now you should have a pretty good picture about the SharePoint 2010 service applications and how to use and configure them for web applications. For choosing the right topology, a lot of other considerations come in play. For example, how many users will my solution serve, do I have to optimize for Search, what are the requirements for availability, isolation, what about disaster recovery and so on. The more considerations, the more architecture possibilities you will have. I will not cover it all, but they all share the common web server role, application server role and database server.

Web Server Role

This server hosts the web pages and their web parts to process requests. There are no components for service applications. However, in small farms this server could also act as the query server for Search. For high availability, you should consider at least 2 load balanced web servers.

Application Server Role

This server is meant to be for service applications. Basically, you can deploy each service application to a dedicated application server. Depending on the requirements you could group most service applications and deploy them to one application server. Search on the other hand, could be a different story. You might have a dedicated application server for crawling (or maybe more), while all the other service applications are on another.

Database Server

Keep in mind that almost every service application has 1 or more databases. The User Profile service has 3 databases: Profiles, Synchronization and Social Tagging. If you have a large environment for Search, you can have multiple property and crawl databases. For database sizes the best practices and limits are the same as with MOSS2007. So keep the databases about 100 Gb. For high availability you have 2 options: implementing a SQL Server cluster or setting up another SQL server acting as the fail-over server for mirrored databases. In SharePoint 2010 you can configure the fail-over database server on a per database level, but you need to set up the mirroring yourself.



Figure 10 shows you an example topology. There are 2 load balanced web servers handling requests for 10,000 – 20,000 users. Then we have 2 application servers. One is dedicated for Search and acts as a Crawler and Query server. The application server contains all other service applications needed and maybe even acting as a second Query server as well (you could have Index Partitions on both servers for redundancy and increasing performance). And we have 2 database servers. One is also dedicated for Search and contains all crawler and property databases. The other database server contains all other SharePoint databases.

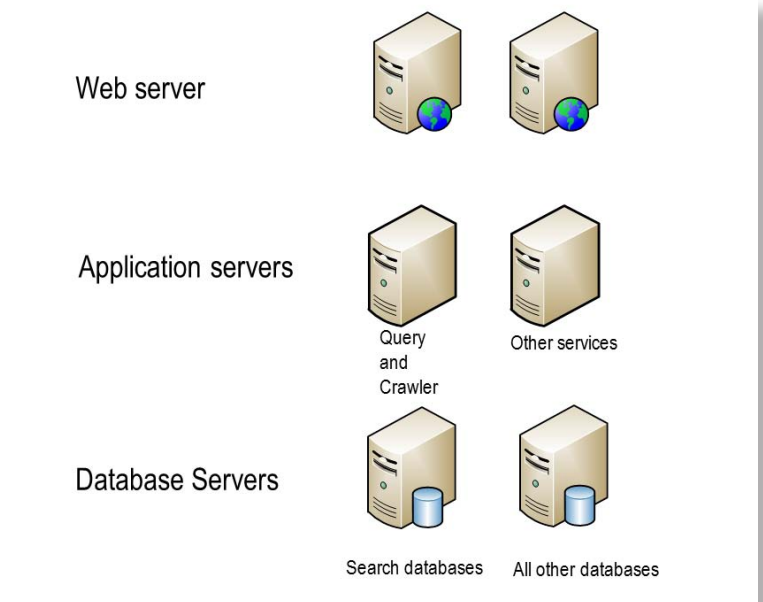


Figure 10: Example Topology.

Conclusion

Yes, it's powerful. It's scalable. It's flexible. SharePoint 2010 offers it all. No more all-or-nothing like MOSS2007, but you can decide what web application can use what services. There are more than a dozen services to choose from and some services can even be shared across farms. But you could also end up having a complex environment that's hard to manage. Pay a little more attention to your design and you're good to go!

For more information I could suggest reading articles about SharePoint 2010 on the TechNet site ([http://technet.microsoft.com/en-us/library/cc303422\(office.14\).aspx](http://technet.microsoft.com/en-us/library/cc303422(office.14).aspx))

... BY THE WAY ...

How to use the Silverlight Video- and audiowebpart

Out of the box SharePoint 2010 will provide you the Silverlight Mediawebpart. If you want to add videos to your Teamsite its just easy now. But before you can use this webpart you must assure that on the toplevel of SharePoint the Publishing Infrastructure feature is activated.

Check it was adminrights, go to Site Actions, Site Settings, Site collection features.

If needed activated the SharePoint Server Publishing Infrastructure feature.

After enabling the feature you can easily edit the page of a Teamsite for example.

Go to your Teamsite and click on Site Actions, Edit page. Click anywhere in the page and click in the Ribbon on Insert, wait a few seconds and hit the button for Video and Audio.

Configure the webpart in the Ribbon with the righth videolocation, size and publish it. Voila!

Tip from Jan Ligtenberg, trainer/consultant APS IT-diensten, www.apsitprojecten.nl



Incorporating External Data in SharePoint 2010 using Windows Communication Foundation (WCF)

by Sreenivasa Ragavan Ramadurai (reviewed by the editors)

Business Connectivity Services is one of the easiest and quickest ways to bring external data into SharePoint. BCS, formerly Known as Business Data Catalog (BDC) in MOSS 2007, provides the ability to Create, Read, Update, and Delete (CRUD Operations) the data from external System directly from SharePoint Unified user Interface. Even Data from external systems can be used to drive other key workloads in SharePoint such as Search and Composites. [BCS could also stand for Bringing external Content to SharePoint]

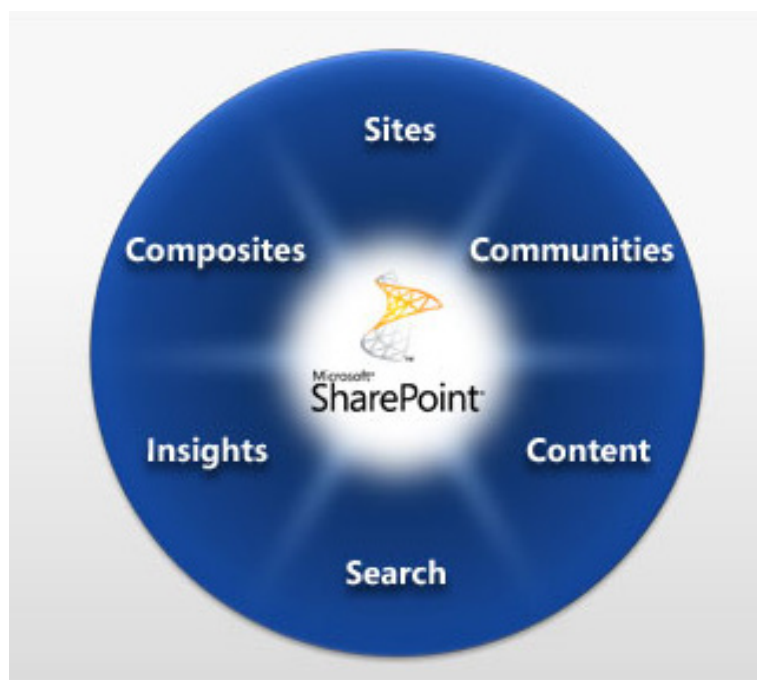


Figure 1: The business connectivity services, one of the new features within SharePoint 2010.

Business Connectivity Services enables you to connect to external data sources, such as databases, Web services, and other Line of Business (LOB) systems like SAP or Siebel. Out of the box there are three ways you can connect to External System

- SQL Database (SPD 2010 -SharePoint Designer)-This method requires no code.
- NET Assembly Connector using VS.NET 2010.
- WCF /XML Web Services.

This article describes a walkthrough how to use Windows Communication Foundation (WCF) to bring External Data into SharePoint 2010.

External Content Type (ECT)

In SharePoint 2010 Microsoft BCS represents external data as an External Content Type. External Content Types are reusable metadata descriptions of connectivity information and data definitions plus the behaviors you want to apply to a certain category of external data. External Content Types enables you to manage and reuse the metadata and behaviors of a business entity such as Customer or Order from a central location, and enable users to interact with that external data and processes in a more meaningful way.

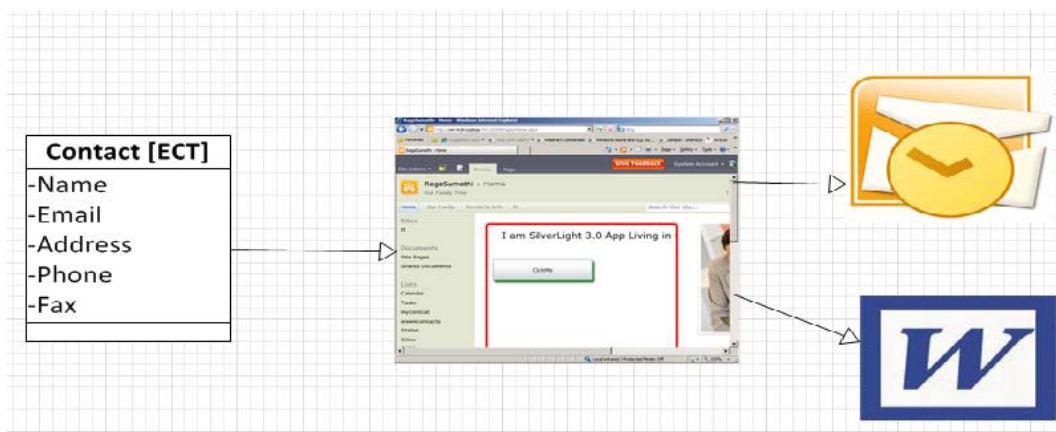


Figure 2: Bringing external information to SharePoint and use the out-of-the-box functionality from SharePoint to export this data to Office client applications..

For example consider the entity called "Contact", you can create an ECT once and reuse it anywhere you need it. Another powerful capability is being able to use the external data that is managed by SharePoint in Microsoft Office client applications. For example, users can take the external content offline using the SharePoint Workspace application just as if it was any other SharePoint content, or they can export it to Microsoft Office Outlook.

Windows Communication Foundation (WCF).

Windows Communication Foundation (WCF) is part of .NET framework (3.0/3.5/4.0) and is designed to build SOA based applications for distributed computing and high interoperability.

This article shows how to bring external data into SharePoint using a WCF service. This example shows Customers from a SQL server database. The database called Chinook is a sample database you can download from Codeplex.com. Although you can create an ECT directly from SQL server, for sake of this article we create a WCF service first.

Step 1: Create a simple WCF service

First we need to create the WCF Service and host the service in IIS. Before we can start creating the contract for the service we create a data model using LINQ to SQL. From this database we are going to use customer table (entity) to bring to SharePoint.

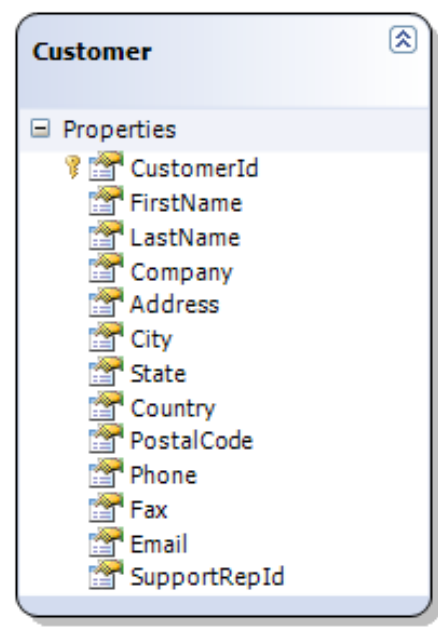


Figure 3: Customer entity.



Next we determine the WCF Service Contract and Operation Contracts. This is necessary to enable to Create, Read, Update and Delete records.

In Visual Studio we create an interface class for the Service Contract.

```
[ServiceContract]
public interface IMusicService
{
    //Create new Customer
    [OperationContract]
    void Create(Customer customer);

    //Delete a customer
    [OperationContract]
    bool Delete(int custId);

    //Select Single Customer
    [OperationContract]
    Customer GetCustomer(int custId);

    //Select All Customer
    [OperationContract]
    List<Customer> GetAllCustomer();
}
```

Listing 1: Interface of the Service.

Next the implementation of the above WCF Service Operation contracts needs to be created. As you can see in listing 2 it uses the IMusicService interface we just created.

```
namespace MusicWcfService
{
    public class MusicServiceImpl : IMusicService
    {
        public void Create(Customer customer)
        {
            using (MusicAlbumDataDataContext dbconn =
                new MusicAlbumDataDataContext())
            {
                dbconn.Customers.InsertOnSubmit(customer);
                dbconn.SubmitChanges();
            }
        }

        public bool Delete(int custId)
        {
            using (MusicAlbumDataDataContext dbconn =
                new MusicAlbumDataDataContext())
            {
                try
                {
                    var deleteQuery = (from cust in dbconn.Customers
                                        where cust.CustomerId == custId
                                        select cust);
                    dbconn.Customers.DeleteOnSubmit(
                        deleteQuery.SingleOrDefault());
                    return true;
                }
                catch (Exception ex)
                {
                    return false;
                }
            }
        }

        public Customer GetCustomer(int custId)
        {
            using (MusicAlbumDataDataContext dbconn =
                new MusicAlbumDataDataContext())
            {
                var queryCustomer = from cust in dbconn.Customers
                                    where cust.CustomerId == custId
                                    select cust;
            }
        }
    }
}
```



```
        return queryCustomer.SingleOrDefault();
    }
}

public List<Customer> GetAllCustomer()
{
    using (MusicAlbumDataDataContext dbconn =
        new MusicAlbumDataDataContext())
    {
        var queryAllcustomer = from cust in dbconn.Customers
                                select cust;

        return queryAllcustomer.ToList();
    }
}
```

Listing 2: Implementation of the Service methods.

Once you deployed the service in IIS and configured the settings you should be able to browse to the service from Internet Explorer. This should look like figure 4.

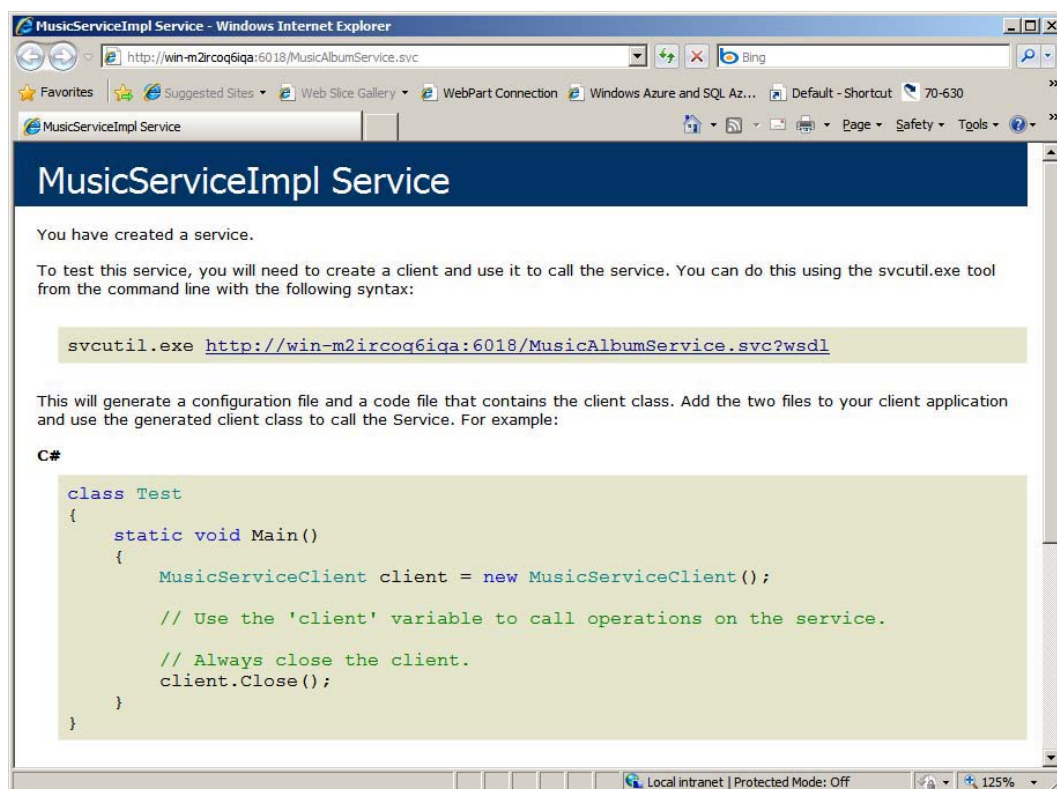


Figure 4: Example how the WCF Service should look like after deployment in IIS.

Once the WCF service is running we can go to step 2.

Step 2: Creating External Content Type In SharePoint Designer 2010 From Our WCF Service

After deploying the WCF service in IIS, start up SharePoint 2010 Designer and create the BCS External Content type from this WCF Service.

1. Open any site in SharePoint 2010 Designer.
2. Click **“External Content Type”** in the Ribbon to create a new External Content type as shown in figure 5.

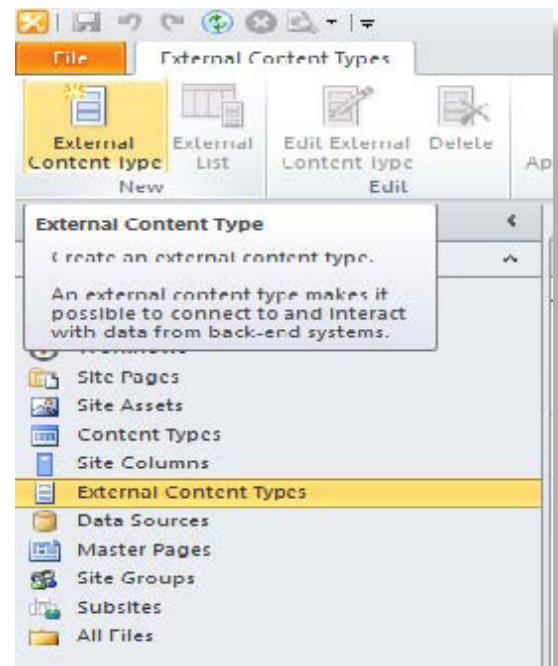


Figure 5: Create External Content Type button.

In the appeared window you can set the properties of the External Content Type

3. Fill in the **"Name"** and other properties.
4. Click on the link: **"Discover external data sources and define operations"**. This will open a new window.

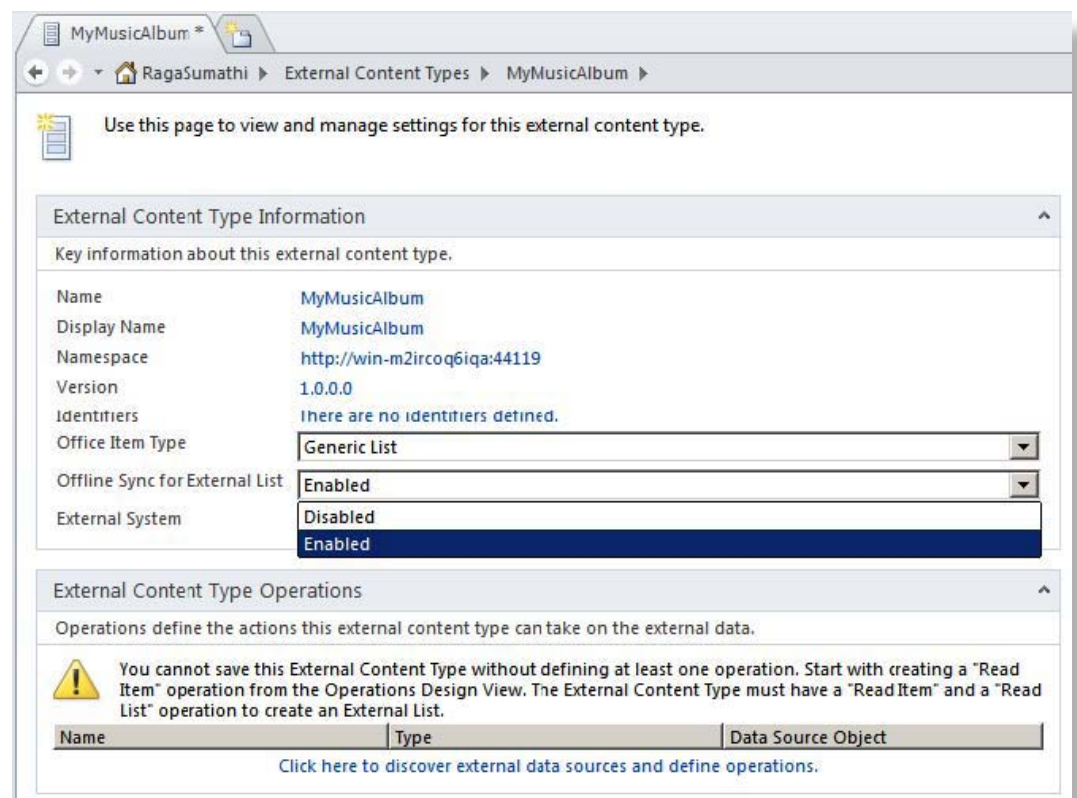


Figure 6: Properties of an External Content Type.

5. In the new window, click **"Add Connection"** and select **"WCF Service"**.

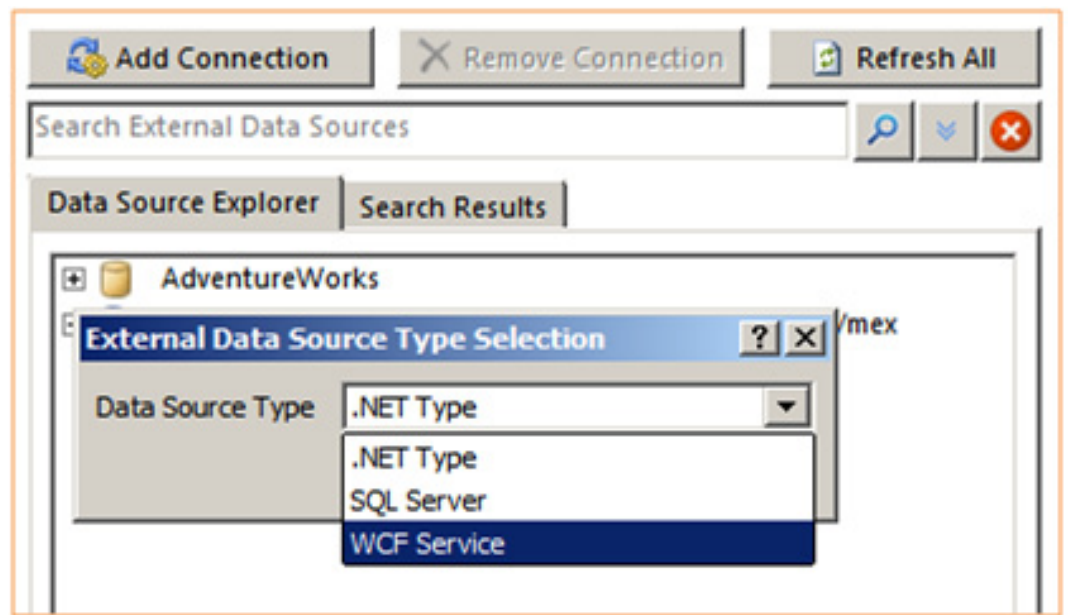


Figure 7: Add connection based on a WCF service.

6. The properties "**Service Metadata URL**", "**Service Endpoint URL**" will be the url to the deployed WCF service we created in Step 1. Fill in the used namespace and click "**ok**". This will download the metadata from the WCF Service contract and generate the BCS External Content Type for you.

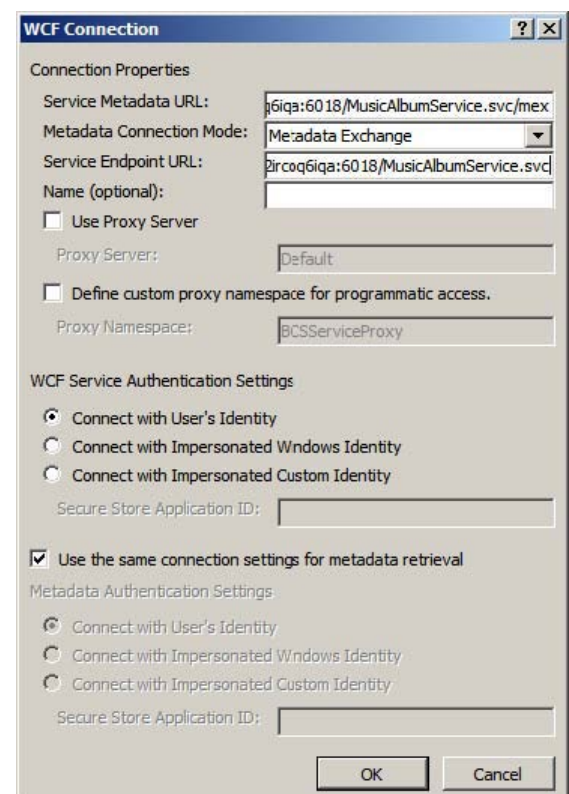


Figure 8: adding the properties of the WCF connection.

In this property window you can also set the security settings. If you're WCF uses Windows Authentication or a Custom Identity you can set the credentials here.

7. Click "**ok**", to close the connection window.

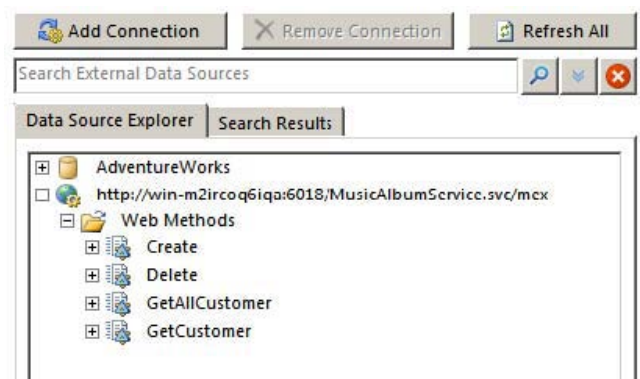


Figure 9: WCF web methods.

Now the connection is made to the WCF service and all web methods are populated (figure 9). At this point we should create the appropriate BCS operations. In our example, we will use GetAllCustomer as "Read List" operation, and Getcustomer and as "Read Item" operation.

Now we need to generate the method for CRUD (Create, Read, Update, and Delete) a customer and the Read List of customers. Here is the step by step procedure for the Read List method.

8. Right click "**GetAllCustomer**" Operation Contract [Web Method] and select "**New Read List Operation**", a wizard will open up.

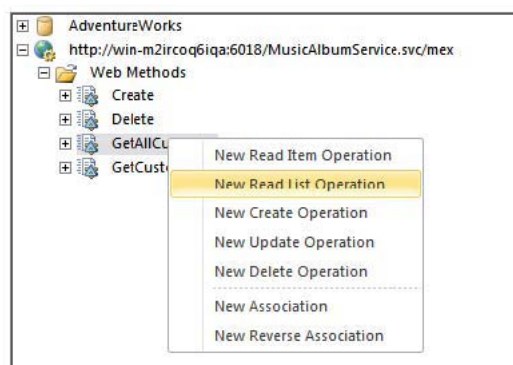


Figure 10: Create the BCS Read List Operation based on a web method.

In the following wizard you can view the data source elements and choose which of these elements should be shown in the list. Every element in the list can be configured. You can which of the elements is the Identifier, if the field is required, read only and if the field can be shown in a picker. This will validate the forms which will be created using this ECT in a SharePoint list.

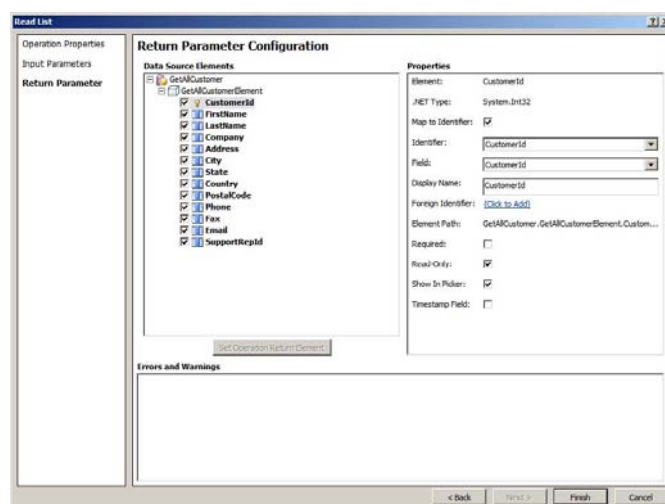


Figure 11: setting the properties on elements.



9. Save the **"External Content Type"**, by pressing **"Save"** button in the SharePoint 2010 Designer at the top.

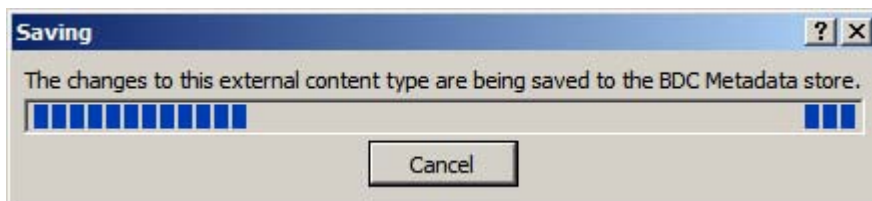


Figure 12: Save content type.

Step 3: Create Profile Page and External List

Profile pages for an instance of a Business Data Connectivity service application site are stored in a site in the farm that hosts the Business Data Connectivity service. You can specify the site where profile pages are hosted.

Creating Profile Page can be created in two ways:

1. From within SharePoint Designer 2010.
Within the Ribbon click on **"Create Profile Page"** from the External Content Type context bar.

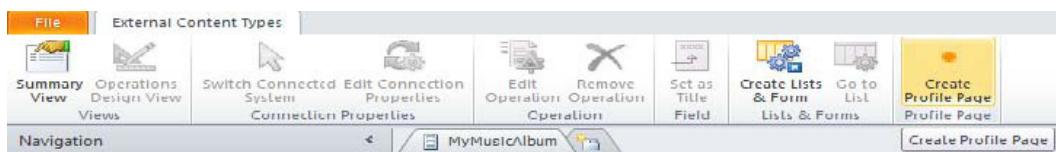


Figure 13: Create Profile Page using SharePoint Designer.

2. From the Central Administration.
Go to manage Service Applications within Application Management.
Click on Business Data Connectivity Models
Choose View External Content Types.
For the External Type you want to create the page for, click on Configure.

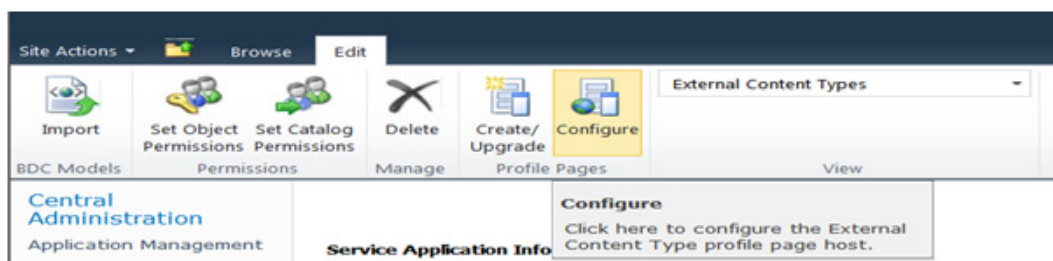


Figure 14: Configure ECT Profile page host from the central administration.

Step 4: Creating External List

The next step is creating a list based on the External Content Type we've just created. These last steps shows how the user will see the data from the external content source.

1. Create a new item from **"site actions"** , **"Create"**
Choose **"external list"**. (figure 15)

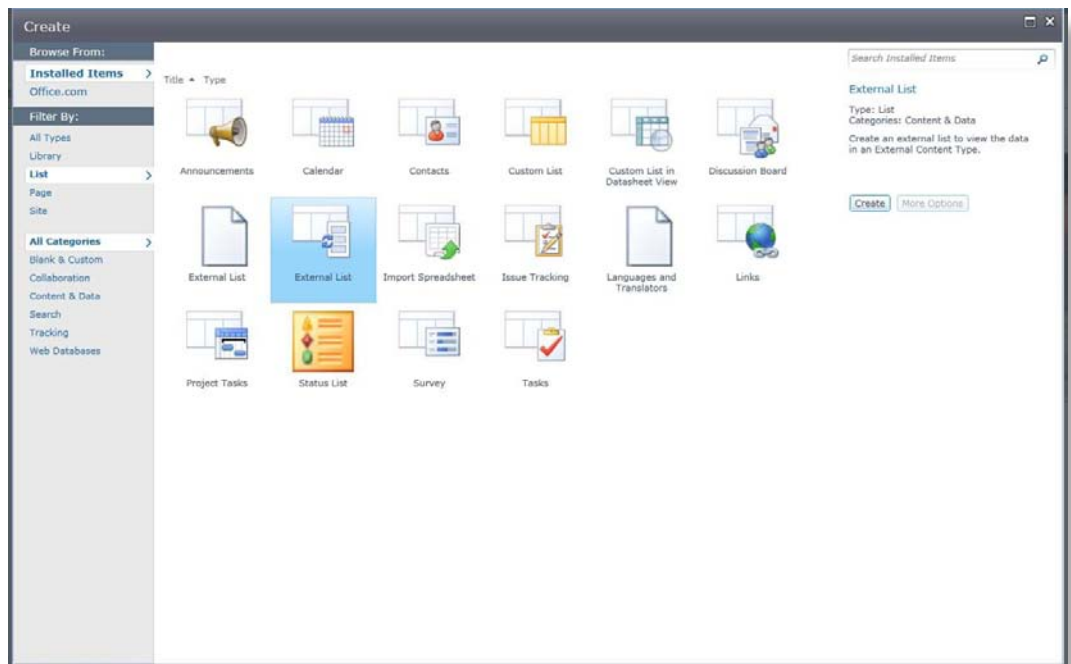


Figure 15: Creating a new list based on External List.

This will open a window where you can set the name and description for the list. Also it shows the Data source configuration. Here you can search for the External Content Type that are available for you.

2. Choose the ECT you created in step 3.

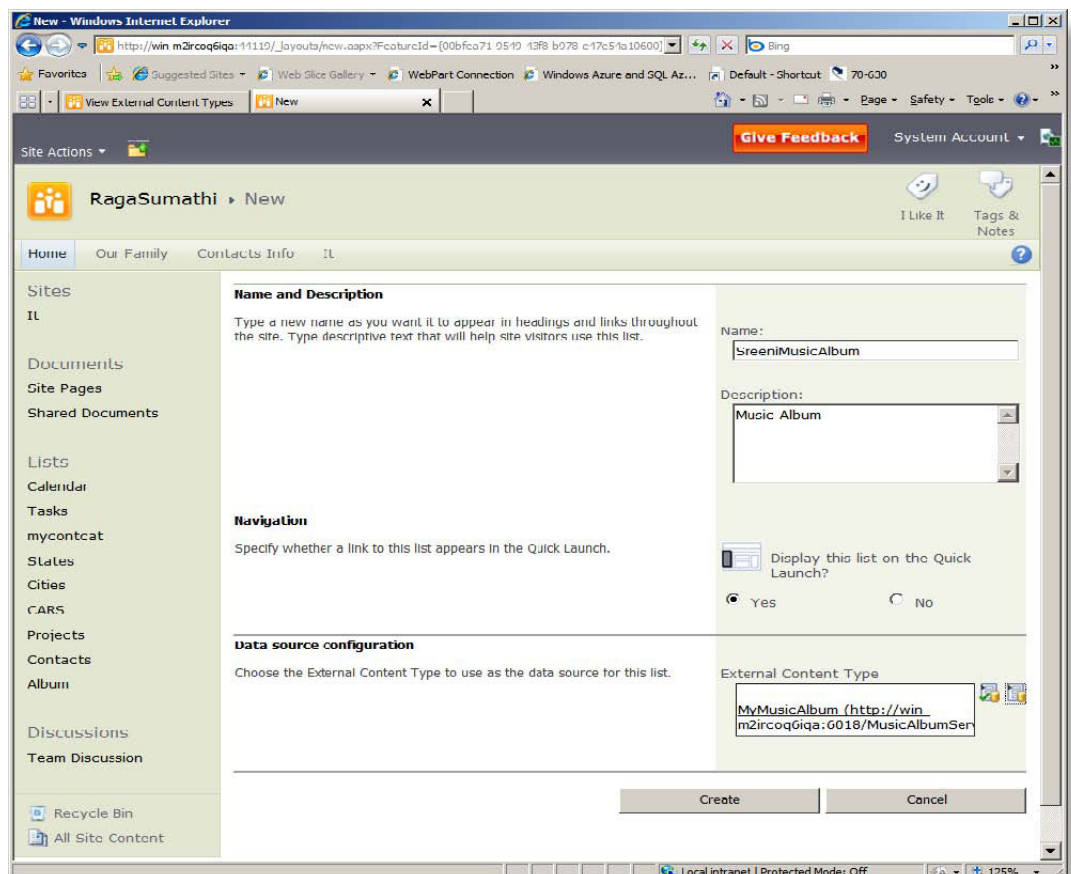


Figure 16: Creating a list based on the ECT.

In the dialog box you can set if you like SharePoint to create the InfoPath forms for you.

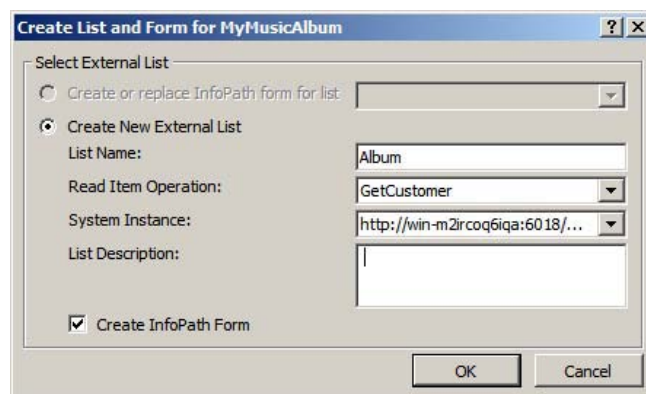


Figure 17: Create List and Form for the ECT.

And that's it! Browsing to the list you will see the content of the external source shown like it's a "normal" SharePoint list. Also the behavior of the items in the list is the same as a normal list. Clicking on a item, selecting the item, even deleting the item is not different from a SharePoint list item.

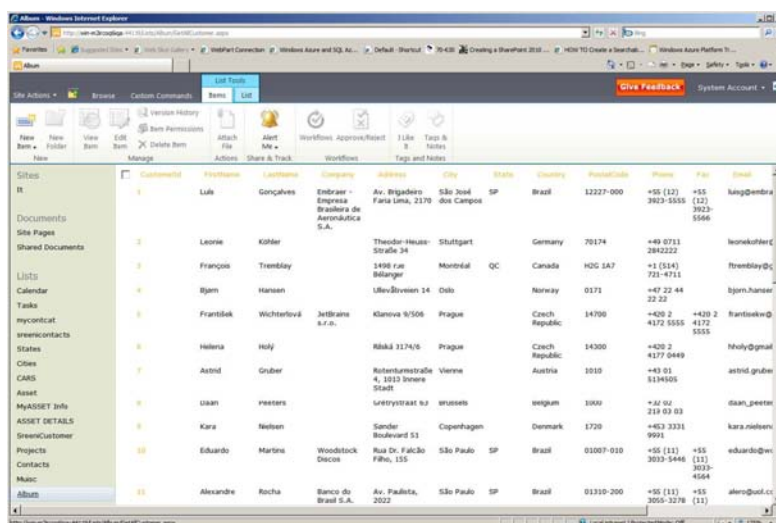


Figure 18: List based on the External Content Type.

The forms in the Dialogs are the same as for SharePoint list items. This is important for the usability of the product. Users have the same user interface and gets the same experience they used to have within SharePoint.

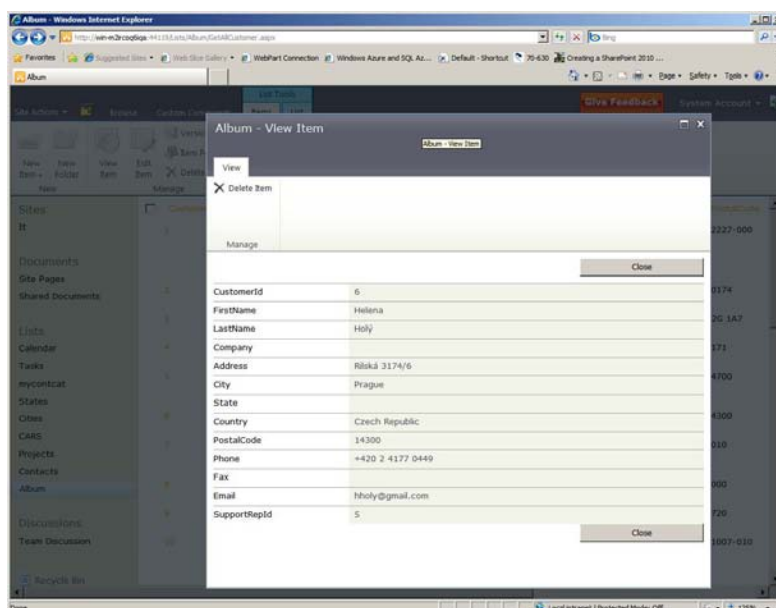


Figure 19: The dialogs are the same as any other dialog.



Conclusion

The Business connectivity services is an ideal way to show and use data from an external source, without losing the context of SharePoint. Using WCF services makes it easy to create the methods needed to add, read, list, update and delete records from the external source. For Line of Business applications the WCF service can be created by the developers of these applications, because you don't necessarily need knowledge of SharePoint or ECTs to create WCF services. And Power users can create the ECT with SharePoint Designer. This is a huge difference compared with creating your own web parts in 2007 to have this kind of functionality.

I would like to thank my Carrier Advisor Ryan Dlay who also works with me at Hitachi consulting for doing proof reading.

... BY THE WAY ...

How to use Calender overlays

Now in SharePoint 2010 Calendars it is possible to use Callendar overlays. If you have a Topsite called Projects and a few subsites for project you can aggregate the appointments into one Calendar.

And of course you need siteaccess to view the related appointments.

Add a Calender to your Topsite, and of course you need Calendars on the Projectsites. Click on the Calendar in the Topsite. In the Ribbon click on Calendar and you will see the button Calendars Overlay.

Click on New Calender and provide the form with the rigth information. After submitting the form you will view got an overview of all projectcalendars.

Tip from Jan Ligtenberg, trainer/consultant APS IT-diensten, www.apsitprojecten.nl



About the authors



Web Content Management the SharePoint Server 2010 way

Waldek Mastykarz is a Dutch SharePoint Server MVP specialized in Web Content Management solutions and development on the Microsoft SharePoint platform. Working as SharePoint Consultant at Mavention, Waldek is responsible for designing and delivering high quality SharePoint Solutions.

Blog: <http://blog.mastykarz.nl>

Web: <http://mavention.nl> / Twitter @waldekm



Themes in SharePoint 2010

Wictor Wilén is a SharePoint Server MVP who works as a Share-Point Architect at Connecta AB. Wictor have been working with SharePoint since the early ages of 2001 and is an active community member and speaker. He is also co-organizer of the Swedish SharePoint User Group. He is currently writing his first book called SharePoint 2010 Web Parts in Action (<http://www.sharepointwebpartsinaction.com>) scheduled for release fall 2010.

Email: wictor@wictorwilen.se / Twitter: @wictor

Blog: <http://www.wictorwilen.se>



Suggestions for Social and Search in SharePoint 2010

Andries den Haan is very excited about web technologies supporting collaboration, content management, enterprise search and social media.

And ...working with SharePoint products and technologies since 2001. Connect, Share and Discover!

Email: andries.denhaan@getronics.com / Twitter: @awdhaan

Blog: awdhaan.blogspot.com



Community Kit for SharePoint: Development Tools Edition

Wes Hackett is a Solutions Architect at Content and Code in the UK. Wes has been building WCM and ECM SharePoint systems for the last three years. As the technical lead for the RNIB SharePoint programme he built the reusable accessibility solution which underpins www.rnib.org.uk providing AAA level accessible SharePoint editing. Wes is currently working on a global SP2010 collaboration portal and the Community Kit for SharePoint: Development Tools Edition.

Email: wesley.hackett@contentandcode.com

Blog: <http://weshackett.spaces.live.com/blog/>



How to manage your Feature Lifecycle using Feature Versioning

Anders Rask is a SharePoint architect and developer living in Århus, Denmark. He has worked as a developer with SharePoint since Tahoe -for good and bad.

Anders is actively participating in the SharePoint community on forums, especially SharePointOverflow.com, twitter and lately by blogging on SharePoint. Anders is also a SharePoint Runner (#sprunners) and dedicated fan of #SharePint.

Email: anders.rask@gmail.com / Twitter: @andersrask

Blog: andersrask.spoint.me



Architecting Service Applications

Octavie is a SharePoint consultant and team lead developer for Ordina Microsoft Solutions in the Netherlands. His main focus is SharePoint 2007 / 2010 IT Pro and development with Visual Studio. Within the company he organizes events regarding Microsoft development platform. On external events where Ordina is a sponsor, he is helping with the sponsor activities.

Email: octavie.van.haaften@ordina.nl / Twitter: @eivatco

Blog: <http://octavievanhaaften.wordpress.com>



Incorporating External Data in SharePoint 2010 using Windows Communication Foundation (WCF)

Sreenivasa Raghavan, Ramadurai is working as an Architect with Hitachi Consulting in Dallas, Texas, USA. My main areas of interest are: Building Distributed applications, Service-Oriented Architectures, Design Patterns and Practices, Cloud Computing (AZURE) BizTalk Server, MOSS, WF, WCF, WPF, SilverLight, ADO.NET DataServices (REST), Odata.

Email: ragavan@hotmail.com

Blog: <http://mstecharchitect.blogspot.com/>



SharePoint Comic

by Dan Lewis

