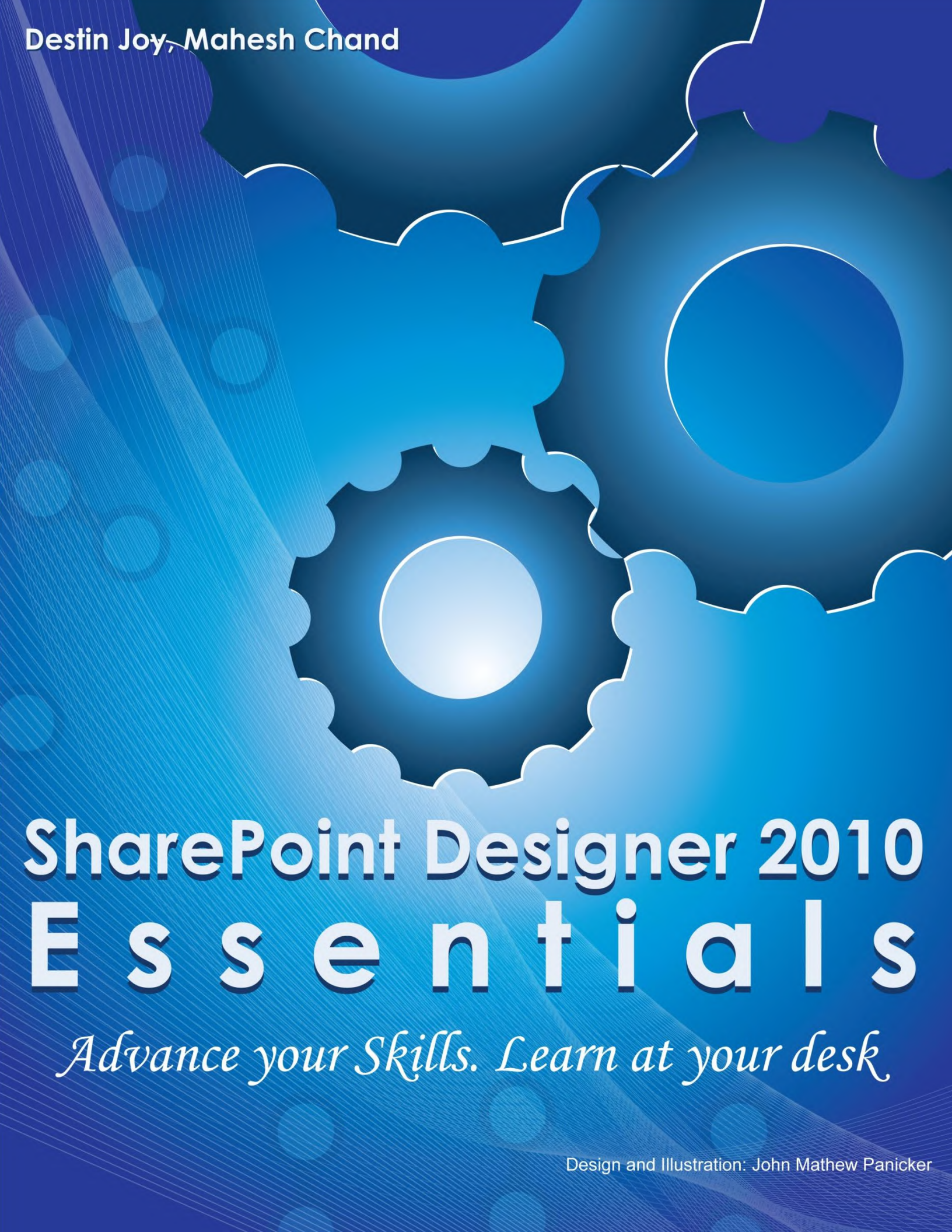




Destin Joy, Mahesh Chand

SharePoint Designer 2010 Essentials

Advance your Skills. Learn at your desk

Design and Illustration: John Mathew Panicker

Copyright

This is a preliminary document and may be changed substantially. The information contained in this document represents the current view of the authors as of the date of publication. This EBook is for informational purposes only. WE MAKE NO WARRANTIES, EXPRESS, IMPLIED, OR STATUTORY, AS TO THE INFORMATION IN THIS DOCUMENT.

Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in, or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of its authors.

Unless otherwise noted, the example companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted in this document are fictitious, and no association with any real company, organization, product, domain name, e-mail address, logo, person, place, or event is intended or should be inferred.

© 2011 Mindcracker. All rights reserved.

Message from the Authors

Thank you for being a part of C# Corner and the Mindcracker Network, a free online community for IT developers and professionals.

We would like to thank **Dan Sullivan** and **Sam Hobbs** for checking for the technical accuracy and grammatical errors. We would also like to thank **Aneesh Bhargavan** and **Jibin Koshy** for providing the code samples. We would also like to thank **John Panicker** and **Jophin Varghese** for the design and **Hari Kumar** and **Deepa Gopinathan** for reviewing the book.

We have always been a big believer and advocate of free knowledge and education for all. To continue this belief, we personally have begun writing free distributable books for our members. Please feel free to share this book with your friends and co-workers.

Also, do not forget to share your knowledge and spread the word around about C# Corner and the Mindcracker Network.

Cheers!

Destin Joy
Microsoft MVP, SharePoint Server



Mahesh Chand
Microsoft MVP, Visual C#



Contents

Abstract.....	1
System Requirements.....	1
Supported Operating Systems.....	1
Computer and processor:	1
Memory:	1
Hard disk:	1
Additional Operating System Details:	2
Additional:	2
Introduction	3
Summary of what's New.....	3
Governance	5
Redesigned User Interface Introduction.....	6
File Tab.....	7
Site Summary Page	7
Navigation Pane	8
Gallery Pages	9
Summary Pages.....	9
List & Content Type Editors	10
Navigation Bar	11
Ribbon.....	11
Zoom to Contents.....	12
All Files View	12
Workflows introduction	13
Reusable Workflows.....	13
Export Workflows.....	13
Association Columns	13

Associate Workflows with Content Types.....	13
Site Workflows	14
Edit the Workflows Included with SharePoint Server	14
Design a Custom Approval Process	14
Impersonation Steps.....	14
Visio Integration	15
Workflow Visualizations.....	15
InfoPath Forms.....	15
Sub Steps & New Workflow Actions	15
XSLT List View Web Part Introduction.....	16
Shared XSLT templates for better performance	16
Reverting the XSLT Customizations: Made Easy.....	16
Create Data Views from the Ribbon	16
Apply Conditional Formatting from the Ribbon.....	17
Share View Styles and Apply them from the Ribbon.....	17
External Content Types and BCS Introduction	17
Create External Content Types in SharePoint Designer 2010.....	17
External Lists	18
External Lists in a SharePoint Workspace.....	18
External Lists in Outlook.....	18
Business Data Columns, Workflow, and Word	18
How to Install SharePoint Designer 2010.....	19
Share Point Designer 2010 Best Practices.....	24
Recommendations.....	25
Working with SharePoint Master Pages	26
Differences Between the Master Page and Content Pages	26
Benefits of Master Pages.....	27
Types of SharePoint 2010 Master Pages	28

Required Place Holders.....	30
How to Customize the Master Page	33
Customizing Master Pages in SharePoint Designer 2010	34
How to Deploy a Master Page	36
Working with Pages	37
Enable AJAX Options in List View Web Parts	42
Views and Forms Web Part in SharePoint Designer 2010.....	45
Working with Data View Web Parts	47
How to Preview a Data View Web part.....	48
How to Use a Data View Web Part to Show Data from SQL Server	49
Data View Migration Issues.....	54
XSLT List View Web part.....	56
How to Create an XSLT List View Web part in Designer 2010	56
How to modify XSLT List View Web Part using conditional formatting	57
XSLT List Form Web Part	61
Data Form Web Part.....	61
List Form Web Part	61
InfoPath Form Web Part	62
Adding Custom Actions to the Ribbon UI	62
Adding a Custom Action to a SharePoint List Item	62
Adding Custom Action to a List Form Ribbon	65
Working with Workflows	67
SharePoint Workflow Demo.....	74
Exporting a SharePoint Designer 2010 Workflow to Visual Studio	81
Working with External Content types	89
Creating External Data Column for a List	96
Creating an External List.....	99
Customizing a Blog Site Using SharePoint Designer 2010	101

Backup & Restore in SharePoint Designer 2010	103
Backup and Restore a Web Site.....	103
Import Web Site Wizard	104
Publish Web Site	104
Conclusion	104

Abstract

This EBook is designed to give SharePoint developers an introduction and overview of the features in Microsoft® SharePoint® Designer 2010. It begins with a brief introduction and a summary of what's new, followed by a more in-depth discussion of the most pertinent features of Designer 2010.

We can use SharePoint Designer 2010 to create end to end solutions on top of SharePoint using functionality such as the Data View web part, Workflow designer, manipulating web part zones, creating page layouts and use of the built-in reports; just to name a few. We truly believe in it and always mention to our customers that they should very seriously consider using SharePoint Designer in their environment and to take full advantage of their SharePoint investment.

System Requirements

To install SharePoint Designer 2010 in your environment, your infrastructure must leverage the following minimum requirements:

Supported Operating Systems: Windows 7, Windows Server 2003 R2 (32-Bit x86), Windows Server 2003 R2 x64 editions, Windows Server 2008, Windows Server 2008 R2, Windows Vista Service Pack 1, Windows XP Service Pack 3.

SharePoint Designer 2010 only connects to SharePoint 2010. To connect to SharePoint 2007 and earlier sites you must continue to use SharePoint Designer 2007. If you elect to install both applications side-by-side, you must download the 32-bit version of SharePoint Designer 2010 and install it after installing SharePoint Designer 2007. Please also note that 64 bit Office applications will not run if SharePoint Designer 2007 is installed. 64-bit Office 2010 applications should only be used if not connecting to SharePoint 2007 servers.

Computer and processor: 500 MHz processor or higher.

Memory: 256 megabytes (MB) of RAM or higher.

Hard disk: 2.5 GB; a portion of the disk space will be freed after installation if the original download package is removed from the hard drive.

Additional Operating System Details: Terminal Server and Windows on Windows (WOW) (which allows installing 32-bit versions of Office 2010 on 64-bit operating systems) are supported.

Additional: Use of graphics hardware acceleration requires DirectX 9.0c compatible graphics card with 64.

Introduction

Welcome to the *SharePoint Designer 2010 Essentials guide*. The goal of this guide is to help you gain sufficient knowledge and understanding of SharePoint Designer 2010 and how to work with it.

Using SharePoint Designer 2010, you can rapidly create SharePoint solutions in response to business needs. Compose no-code solutions that encompass a variety of common scenarios; from collaborative sites and web publishing to Line-Of-Business data integration, business intelligence solutions and human workflows, all leveraging the building blocks available in SharePoint in an easy to use environment. Developers can use SharePoint Designer 2010 to get a quick start on SharePoint development projects.

The following are the list of major benefits of SharePoint Designer 2010.

- **Rapidly Develop SharePoint Solutions without Code:** Build composite SharePoint applications by configuring or designing components without having to write any code. The improved user interface, easy connectivity to external systems, improved views and forms, and enhanced workflow design capabilities allow us to respond more rapidly to our business needs.
- **Easily Connect with External Data:** With SharePoint Designer 2010, we can easily integrate external data into any SharePoint solution. Discover available data sources, create External Content Types that define the data operations you want to enable in SharePoint and have SharePoint Designer auto-generate SharePoint external lists and required forms.
- **Deploy Safe Customizations with Confidence:** SharePoint Designer 2010 provides control over where and how developers use SharePoint Designer throughout all levels of your SharePoint deployment. Preserve a consistent brand and layout across a site collection or web application by controlling customization.

Summary of what's New

SharePoint Designer 2010 includes numerous new capabilities and features but only some of them are highlighted in this book. Listing 1 shows a brief comparison between SharePoint 2007's and SharePoint 2010's features.

No	Block	Features	SharePoint Designer 2007	SharePoint Designer 2010
1	UI Related	Ribbon Feature	No	Yes
		Quick Launch Navigation	No	Yes
		Site Structure Information	No	Yes
2	Advancement in list and Document library	Custom Action Modifications	No	Yes
		Customize workflows associated with current lists	No	Yes
		Add or customize content types	No	Yes
3	Site based	Modify or create Groups	No	Yes
		Manage Site Settings	No	Yes
4	XSLT advancement	Dynamically show views of your data via XSLT List View Web Parts	No	Yes
5	Workflow Enhancement	Many more actions added	No	Yes
		Export and Import workflow from Visio	No	Yes
		Reusable Workflow	No	Yes
		Site Workflow	No	Yes

6	Content Types	Add or modify content Type	No	Yes
		Attach content type to list	No	Yes

Listing 1

We will discuss these features and some more features later in this book.

Governance

With all of the new and enhanced features in SharePoint 2010, one which we cannot skip over is how access for SharePoint Designer is handled. There were a lot of dissociations about how governance related to SharePoint Designer can be handled. With SharePoint 2010, there are no obscure methods or actions required to limit SharePoint Designer's access, they've moved it into the Site Collection Administration.

To view/modify these settings, go up to your site collection level, look under the Site Collection Administration custom action group on the Site Settings page and click on the SharePoint Designer Settings link towards the bottom. See Figure 1.

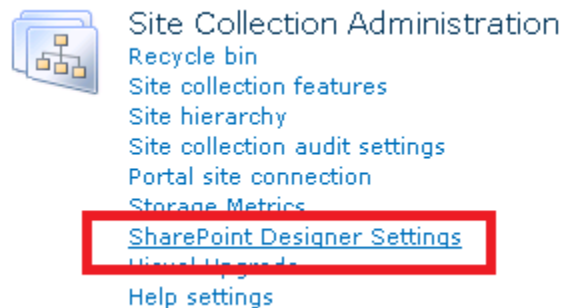


Figure 1

We can enable or disable SharePoint designer access to the entire site collection. We can even go as far as enabling the detaching of pages from the site definition, enable or

disable the ability to modify Master Pages and Page Layouts and the ability to manage the site hierarchy, as you can see in Figure 2.

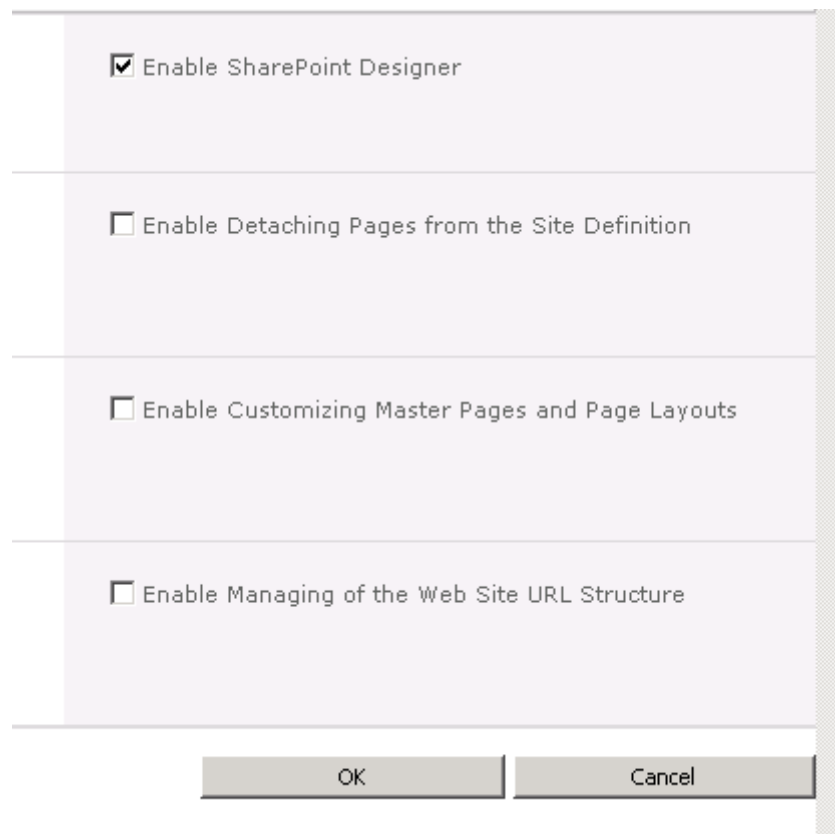


Figure 2

Redesigned User Interface Introduction

The user interface of SharePoint Designer 2010 has been completely redesigned to “put the ‘SharePoint’ in SharePoint Designer.” All of the important components in a site or solution — lists and content types, views and forms, workflows, data sources and external content types, page layouts and master pages can be easily found and customized. We can also see how components are related to each other.

File Tab

When we start SharePoint Designer 2010, the new File tab makes it easy to get started. There are options to open existing sites or create new sites. We can open pages or sites that we have recently worked with. We can also add new components such as lists and workflows to the current site. A Help menu and Option tab are also available. The Option tab allows us to customize some of the features.

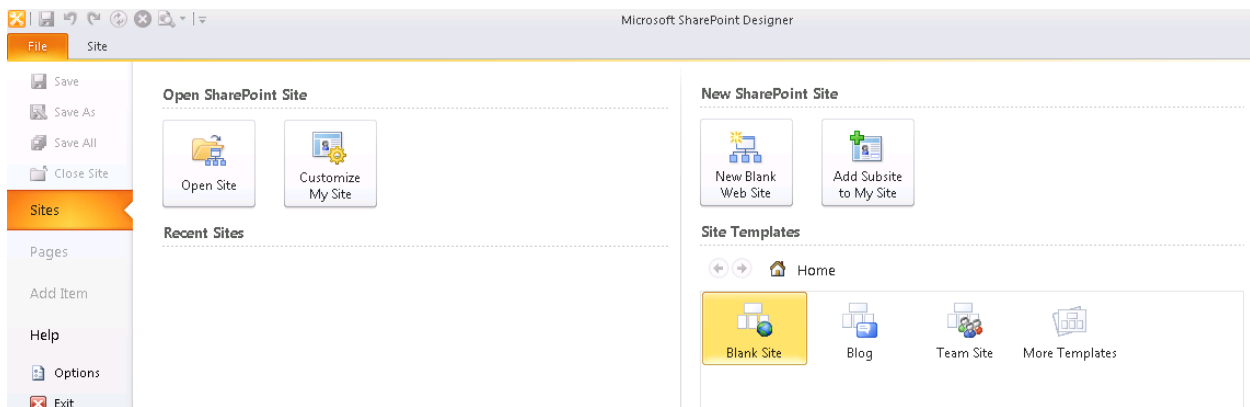


Figure 3

Site Summary Page

SharePoint Designer 2010 has a summary page where we can change settings or manage various aspects of our site, as you can see in Figure 4. We can manage permissions directly from SharePoint Designer 2010. Like every summary page, the site summary page shows related components. We can easily view and navigate to all subsites in the current site.

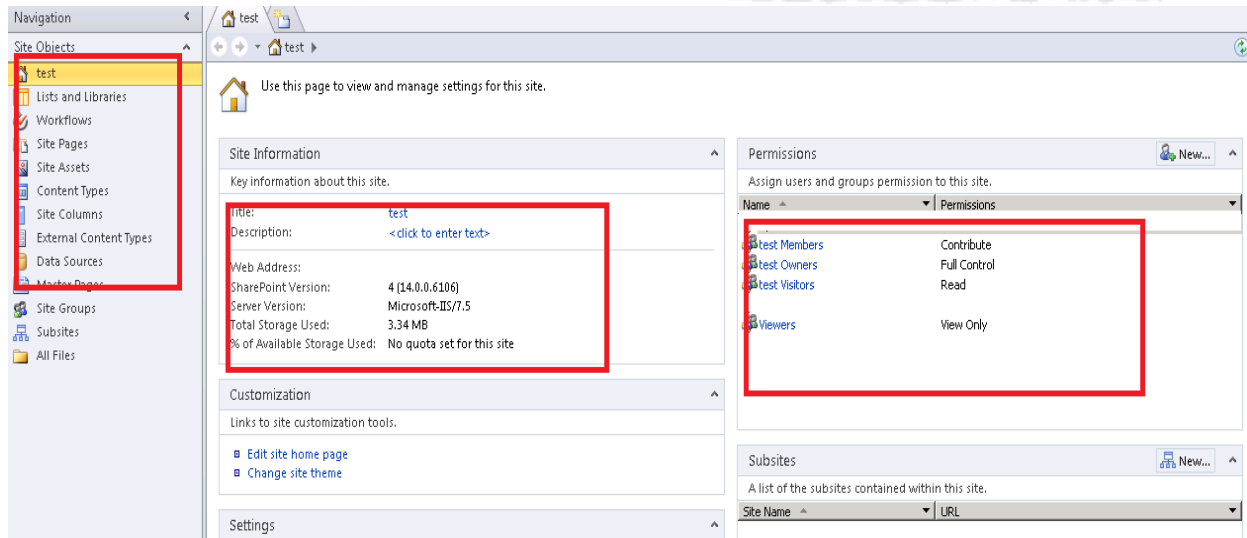


Figure 4

Navigation Pane

The new improved Navigation pane makes it easy to find and navigate to all of the important components in a site that you use to build a solution. We can just work with the navigation pane to work on different tasks like creating a workflow or lists etc. See Figure 5.

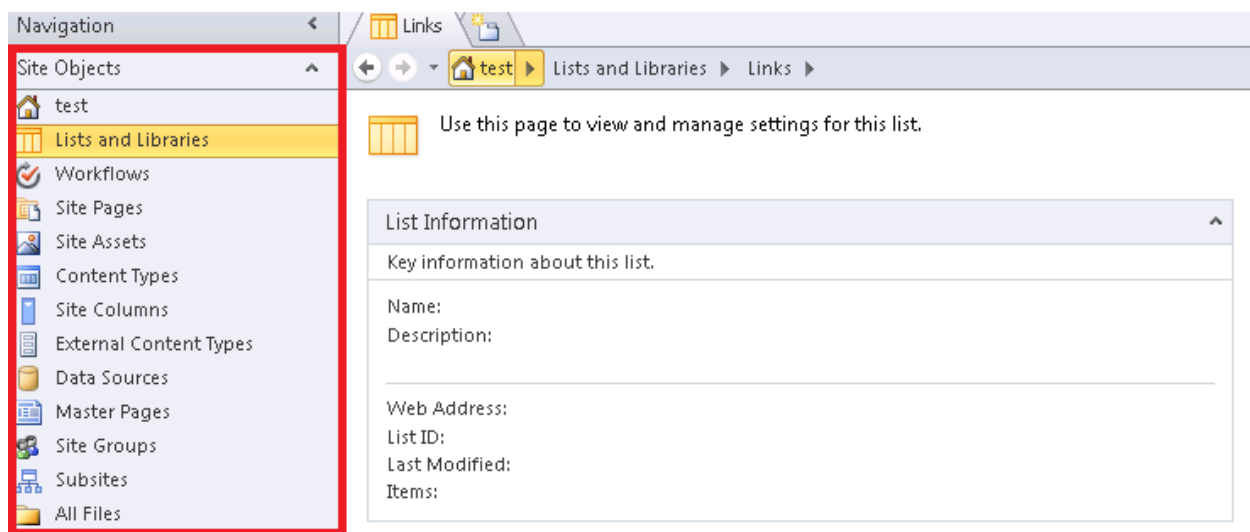


Figure 5

Gallery Pages

Gallery pages are one of the good features in SharePoint Designer 2010, as you can see in Figure 6. It is very easy to find and edit any of the important components of our solution. In the Navigation pane, clicking any component will show you a gallery of all such components in the current site. In the screen shot shown below, we have selected the workflows; we can see the details in the page as shown below. On a gallery page, just click any component to edit it.

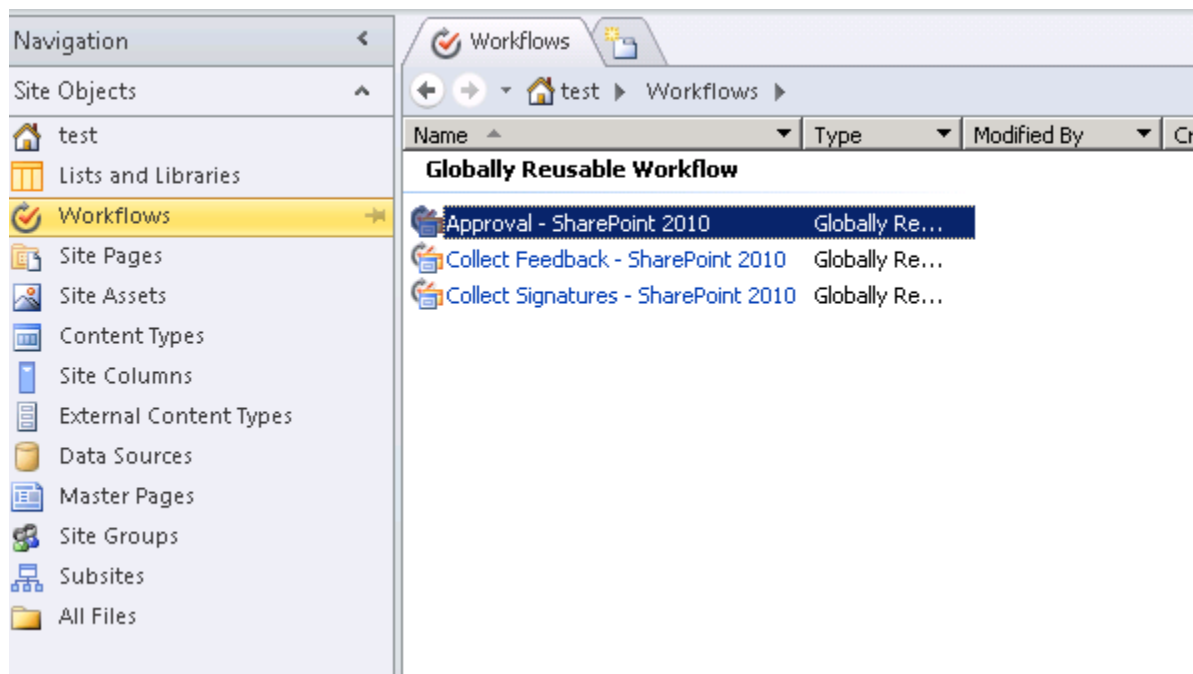


Figure 6

Summary Pages

A summary page gives us an all-up interpretation or portal into all of the pieces that make up a module. For example, the summary page for a list shows you the forms and workflows for that list. From a summary page, you can edit the settings for that module like name, permissions or even edit the component itself. Figure 7 shows a summary page.



Use this page to view and manage settings for this list.

List Information

Key information about this list.

Name:

Links

Description:

Use the Links list for links to Web pages that your team members will find interesting or useful.

Web Address:

/Lists/Links/AllItems.aspx

List ID:

{F54B06C}

Last Modified:

6/17/2011 2:49 PM

Items:

0

Customization

Links to list customization tools.

Edit list columns

Permissions for this list

Settings

General settings for this list.

General Settings

Display this list on the Quick Launch

Views

Views display list data in prescribed orders and selections.

Name	Type	Default
All Links	HTML	Yes

Forms

Forms are used to display and edit data contained within this list.

File Name	Type	Default
DispForm.aspx	Display	Yes
EditForm.aspx	Edit	Yes
NewForm.aspx	New	Yes

Figure 7

List & Content Type Editors

We can use SharePoint Designer 2010 to edit all of the important components that comprise a SharePoint solution including lists and content types. As you can see in Figure 8, you use lists and content types to model your data, so these components are often at the core of a solution. SharePoint Designer 2010 also provides a full-page editor for both lists and content types. You can quickly edit the columns and settings of a list or content type in one place, without having to navigate between pages as in the browser.

Settings

General settings for this list.

General Settings

Display this list on the Quick Launch

Hide from browser

Advanced Settings

Allow attachments

Display New Folder command on the New menu

Require content approval for submitted items

Create a version each time you edit an item

Allow management of content types

Content Types

A content type is a reusable collection of columns and settings that you can apply to your list.

Name	Show on New Menu	Default
Folder		
Link	Yes	Yes

File Name	Type	Default
DispForm.aspx	Display	Yes
EditForm.aspx	Edit	Yes
NewForm.aspx	New	Yes

Workflows

A list of workflows associated with this list.

Name	Start Settings
There are no items to show in this view.	

Custom Actions

Custom actions add buttons to the SharePoint Ribbon and options to the List Item Menu.

Name	Button Location	Sequence
There are no items to show in this view.		

Figure 8

Navigation Bar

If we navigate through the components in a site, the Navigation bar can help us to understand where we are in the site and how the various components are related to each other. The Navigation bar displays breadcrumbs that show where you are in the site hierarchy. There are also forward and backward buttons. The following example shows a list named Links. As you can see in Figure 9, we have navigated to DispForm.aspx.

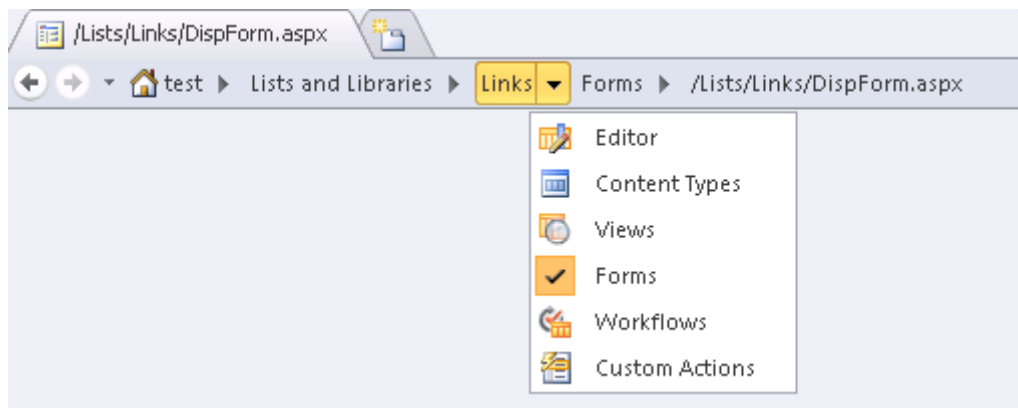


Figure 9

Ribbon

The Ribbon is a new concept introduced in SharePoint 2010. Figure 10 shows how the ribbon looks like in SharePoint Designer 2010. It makes us more efficient and dynamic in SharePoint Designer 2010. The Ribbon shows all of the important features in the current context for us to work on. For example, when we are working with a list, all the items that may be needed can be seen on the ribbon.

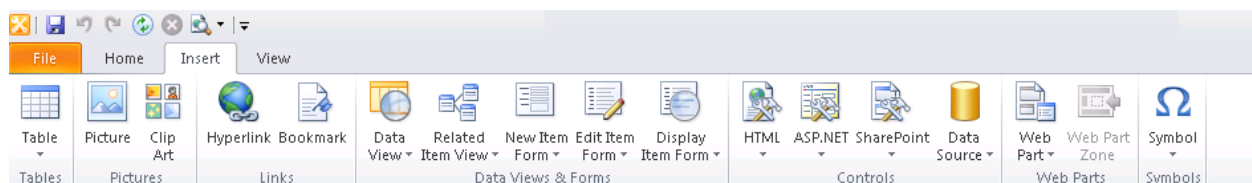


Figure 10

Zoom to Contents

A SharePoint page can have a lot going on with content placeholders, Web Part zones, controls, and other page elements. When we're working with a SharePoint page, often we want to emphasize a single element, such as a list web part or a Web Part zone or a Data View. In SharePoint Designer 2010, we have an option to select any of those elements and then right click, select Zoom to Contents. Once done, the Design view and Code view will show only the control that we selected. So we can eliminate other code and controls while working with a particular component. See Figure 11.

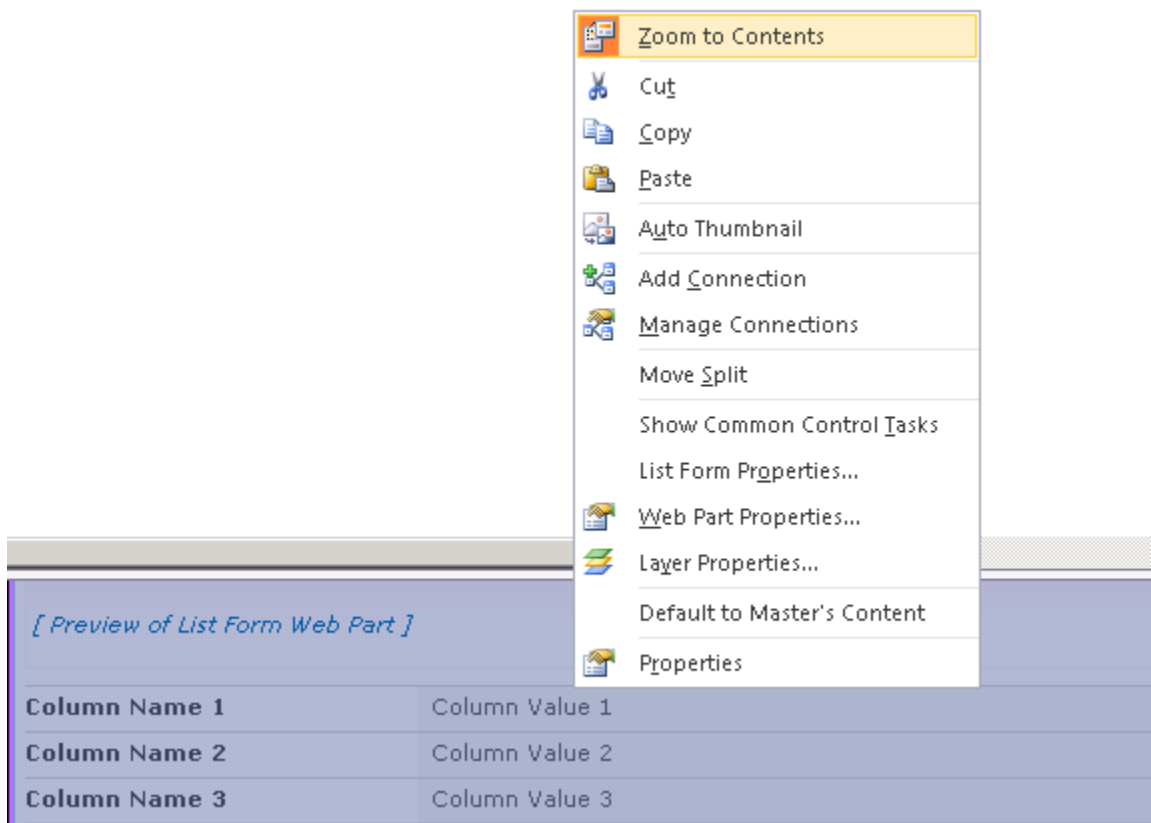


Figure 11

All Files View

The new user interface in SharePoint Designer 2010 focuses on surfacing all of the important components in a SharePoint solution. If you used the Folder List in previous versions of SharePoint Designer, you can still have the experience of working with the files and folders in a site by using the All Files view.

Workflows introduction

SharePoint Designer 2010 includes major improvements to workflows. You can quickly design workflows that ease a wide range of business and human processes. Many new building blocks also mean that workflows are even more flexible, powerful and capable of modeling more complex business logic and process.

Reusable Workflows

With SharePoint Designer 2010, you can create workflows that are easily reusable. You can create a reusable workflow in the top-level site in the site collection, and that workflow is globally reusable; meaning that the workflow can be associated to any list or library in the site collection. You can also create a reusable workflow in any sub site in the site collection; this workflow is available for reuse in that particular sub site. This is a new type of workflow in SharePoint Designer 2010.

Export Workflows

You can export a reusable workflow from one site collection and then upload and activate that workflow in a different site collection. SharePoint Designer 2010 supports exporting a workflow as a solution package or .wsp file. Later in this book, we have explained how to export Designer 2010 workflows to Visual Studio 2010.

Association Columns

If your reusable workflow requires certain columns to be present in the list or library that it is associated with, then you can add those columns as association columns. Association columns get added automatically to a list or library when a reusable workflow is associated with that list or library. This is one of the great features.

Associate Workflows with Content Types

When you create a reusable workflow, you can choose to filter your reusable workflow to a specific content type. If you associate a workflow to a site content type, you make that workflow available for all items of that content type in every list and library where that content type is used.

Site Workflows

A site workflow is associated with a site. It is not associated to a list, library, or content type. So unlike most workflows, a site workflow is not running on a specific list item. In the browser, you can start a site workflow or view the status of running site workflows by clicking Site Workflows on the Site Actions menu.

Edit the Workflows Included with SharePoint Server

One of the weaknesses of SharePoint Designer 2007 was that we were not able to edit any of the out-of-the-box workflows. In SharePoint Designer 2010, this problem has been rectified. The three most popular workflows in SharePoint Server: the Approval, Collect Feedback, and Collect Signatures have been completely rebuilt as declarative workflows. That means that they are now fully customizable. These workflows are event-driven, and all of the important events in the workflow are textured in the workflow editor, both for each task in the process and for the process as a whole. For example, you can easily add conditions and actions to define what happens when each task is assigned, expired, or completed.

Design a Custom Approval Process

SharePoint Designer 2010 includes three new approval actions: Assign Item for Approval, Assign Item for Feedback, and General Task Process. The main three workflows included in SharePoint Server 2010 out of the box are: Approval, Collect Feedback, and Collect Signatures workflows. The approval actions surface all of the important events in an approval process, making it easy for you to design a new interactive workflow process where many users can interact or collaborate on a specific document.

Impersonation Steps

In the previous version of SharePoint Designer, a workflow always impersonated the user who started the workflow and ran with that user's permissions. That means we can run a workflow with current user context only. In many instances, we need to give unnecessary permissions to users who are involved in the workflow process. In SharePoint Designer 2010, you can use impersonation steps to have the workflow perform actions by impersonating the workflow author instead of the workflow initiator. Impersonation steps are very useful in approval and publishing scenarios, where the people submitting content for approval and the people approving content have different permissions.

Visio Integration

Most organizations have a business analyst who best understands the process that a workflow needs to capture. With SharePoint Designer 2010 and Visio Professional 2010, a business analyst can sketch out workflows graphically in Visio. Visio Professional 2010 includes a Microsoft SharePoint Workflow template that includes shapes for all of the workflow conditions and actions. You can start in Visio and then export to SharePoint Designer, or you can start in SharePoint Designer and then export to Visio. You can iterate on a workflow design with many round trips between both programs without losing any of the workflow settings.

Workflow Visualizations

You can easily view the status of any running workflow on the Workflow Status page. With workflow visualization, a Visio diagram of your workflow is automatically created and displayed in a Visio Web Part on the Workflow Status page. The workflow visualization shows a live view of the exact workflow. Workflow visualizations require Visio Professional 2010 to be installed on your computer and Visio Graphics Services running on the server.

InfoPath Forms

If your server is running SharePoint Server 2010, all of the workflow forms: the association form, initiation form, and task forms are InfoPath 2010 forms. It is easy to change the appearance and layout of InfoPath forms, and you can add validation rules to an InfoPath form. To edit a workflow form in InfoPath 2010, just click the form and InfoPath opens directly from SharePoint Designer 2010.

Sub Steps & New Workflow Actions

SharePoint Designer 2010 includes many new building blocks such as sub steps and new workflow actions. For example, if your workflow is associated with a document library or if it is filtered to the Document content type, a group of background Document Set actions appear. As you know, document set is a new feature in SharePoint Server 2010 whereby a group of documents are treated as a single unit, so a workflow action for a document set will repeat on all items in that document set. Other new workflow actions include utility actions to manipulate strings and dates; a new relational action that uses Active Directory to look up a person's manager; and new list actions that a workflow can use to set permissions on a list item.

XSLT List View Web Part Introduction

You can easily share custom styles for your list views with others in the site collection by uploading the styles to the Style library. Anyone with proper permissions can then use SharePoint Designer 2010 to apply these custom styles to their own list views with just one click. Every standard list view in SharePoint 2010 products is now a Data View specifically, an XSLT List View Web Part. This means that you can easily customize any standard list view in SharePoint Designer 2010. You can also modify the view using all of the options in the browser, such as adding or removing columns or filtering and sorting. An XLV is fully customizable both in SharePoint Designer and in the browser.

Shared XSLT templates for better performance

The XSLT ListView WebPart uses shared XSLT style sheets that reside in the _layouts directory on the server. Every standard list view on the entire server uses the same set of shared XSLT templates. An XLV uses one template to format each field (or column), one template to format the row, and one template to format the table. When you customize the appearance of an XLV, you generate overriding XSLT templates only for the specific templates that you modify; that means it will affect only the list or view you are modifying

Reverting the XSLT Customizations: Made Easy

There are situations when you customize an XSLT List View Web Part and later decide that you want to undo those changes, or if you encounter errors in the XSLT that you are unable to fix; you can always revert an XLV to how it first appeared before you made any changes. When you revert the XSLT for an XLV, that XLV goes back to using the shared XSLT templates on the server. You can choose to revert the entire view or just specific XSLT templates, such as the item template. The same process happens for the master page also. We have explained the same process in detail in the master page section.

Create Data Views from the Ribbon

The Data View is a powerful feature for viewing and editing data from a wide range of data sources including both data in SharePoint lists and data external to SharePoint. The Ribbon makes it easy to insert a Data View with just a couple of clicks. You can then further customize your Data View by using all of the options under List View.

Apply Conditional Formatting from the Ribbon

Conditional formatting can give your Data Views more impact by highlighting the information that's critical to your team. You can apply conditional formatting rules to your Data Views directly from the Ribbon. In SharePoint Designer 2007, you had to use the designer for the conditional formatting. In Design 2010, you still have the option to use the Designer to format conditions

Share View Styles and apply them from the Ribbon

You can easily share custom styles for your list views with others in the site collection by uploading the styles to the Style library in the top-level site in the site collection. Anyone with proper permissions can then use SharePoint Designer 2010 to apply these custom styles to their own list views with just one click.

External Content Types and BCS Introduction

SharePoint 2010 includes new capabilities for deeply integrating external business data and processes into the user experience in Microsoft Office and SharePoint. These capabilities include new features, services, and tools in the SharePoint platform. With Business Connectivity Services (BCS), you can connect to external data from Microsoft SQL Server, Web services, or .NET assemblies. You can connect to this external data from the client or the server, whether online or offline and you can create, read, update, and delete data in these external data sources.

In previous versions of SharePoint, there was the concept of a Business Data Entity, which was a type of external business data available in the Business Data Catalog. We can use SharePoint Designer to create an external content type. This has been explained later in this book.

Create External Content Types in SharePoint Designer 2010

In the previous version, SharePoint did not support authoring an application definition in the Business Data Catalog. SharePoint Designer 2010 provides the tools to connect to external data, create external content types, publish those external content types to the Business Data Catalog on the server and auto-generate both an external list and all

of the forms required by that external list. With SharePoint Designer 2010, you can make use of external data from LOB systems such as SAP or Siebel without writing a single line of code.

External Lists

An external list is the key building block for BCS solutions. An external list looks and behaves virtually the same as any SharePoint list but contains data from an external source. An external list makes it easy for people to read and write external data within the familiar experience of working with a SharePoint list. The main drawback of BDC is that it is read only data. You can create an external list and its forms from within SharePoint Designer 2010. Creating an external list also creates views and forms for the browser, and a customizable Outlook 2010 form and an InfoPath 2010 form. The InfoPath form can be used instead of the default XSLT views and forms to view and edit data in the external list in the browser, and the Outlook form is used to display external data in Outlook 2010.

External Lists in a SharePoint Workspace

After you create an external list, you can just click the Connect to SharePoint Workspace button on the server Ribbon to make that external list available in SharePoint Workspace 2010. SharePoint workspace is a new feature in SharePoint Designer 2010.

External Lists in Outlook

You can also connect an external list to Outlook 2010. When you create an external content type in SharePoint Designer 2010, you can map it to a native Office item type such as a contact or task so that the external data looks and behaves just like a contact or task item in Outlook. These new features enhance the power of SharePoint Designer 2010.

Business Data Columns, Workflow, and Word

After you create an external content type with SharePoint Designer 2010, you can add a Business Data column to a list or library. A Business Data column is associated with an external content type and represents a field from that ECT. If a document library uses a Word document for its template, you can edit the template in Word by adding Quick Parts, which are content controls that are bound to Business Data columns. When

people create a new document based on that Word template, external data is displayed inside the document in the content controls.

How to Install SharePoint Designer 2010

1. Double Click on the file of your SharePoint Designer 2010 install. See Figure 12.

Name ^	Date modified	Type	Size
 SharePointDesigner	9/6/2010 2:49 PM	Application	285,743 KB

Figure 12

2. You will be prompted with a screen that looks like Figure 13.
3. Click Run.

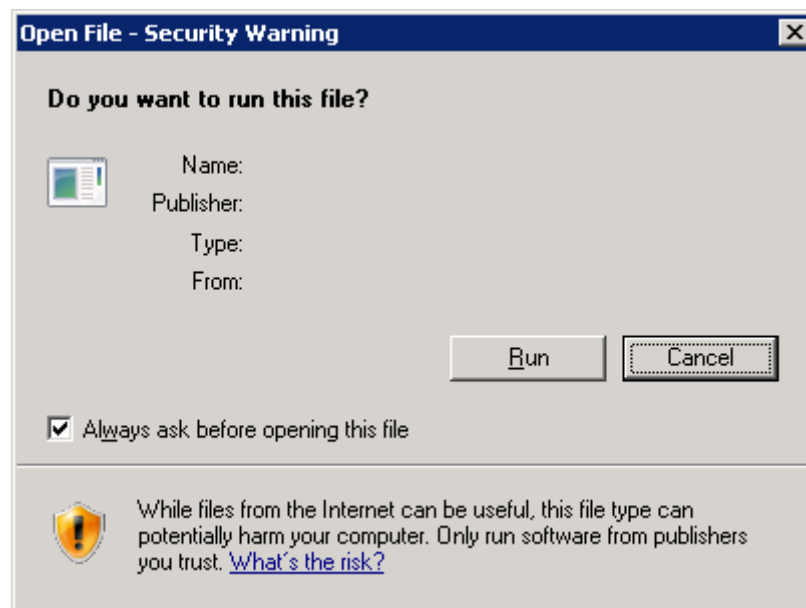


Figure 13

4. Once clicked, you will get a screen that looks like Figure 14. On this screen, click on the "Install Now" button to continue.



Figure 14

5. Click Continue
6. The installation will progress as shown in Figure 15.

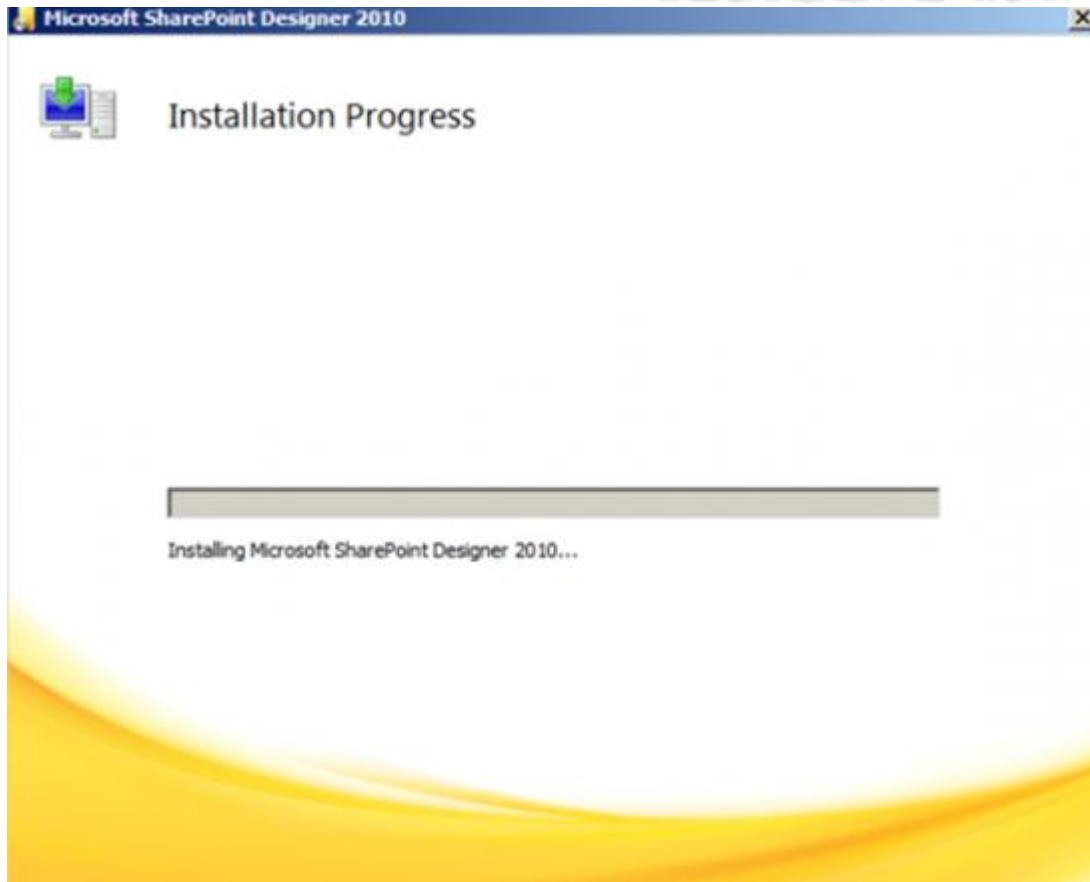


Figure 15

7. Upon completion of the installation you will get the success message as shown in Figure 16 and then Figure 17.



Figure 16

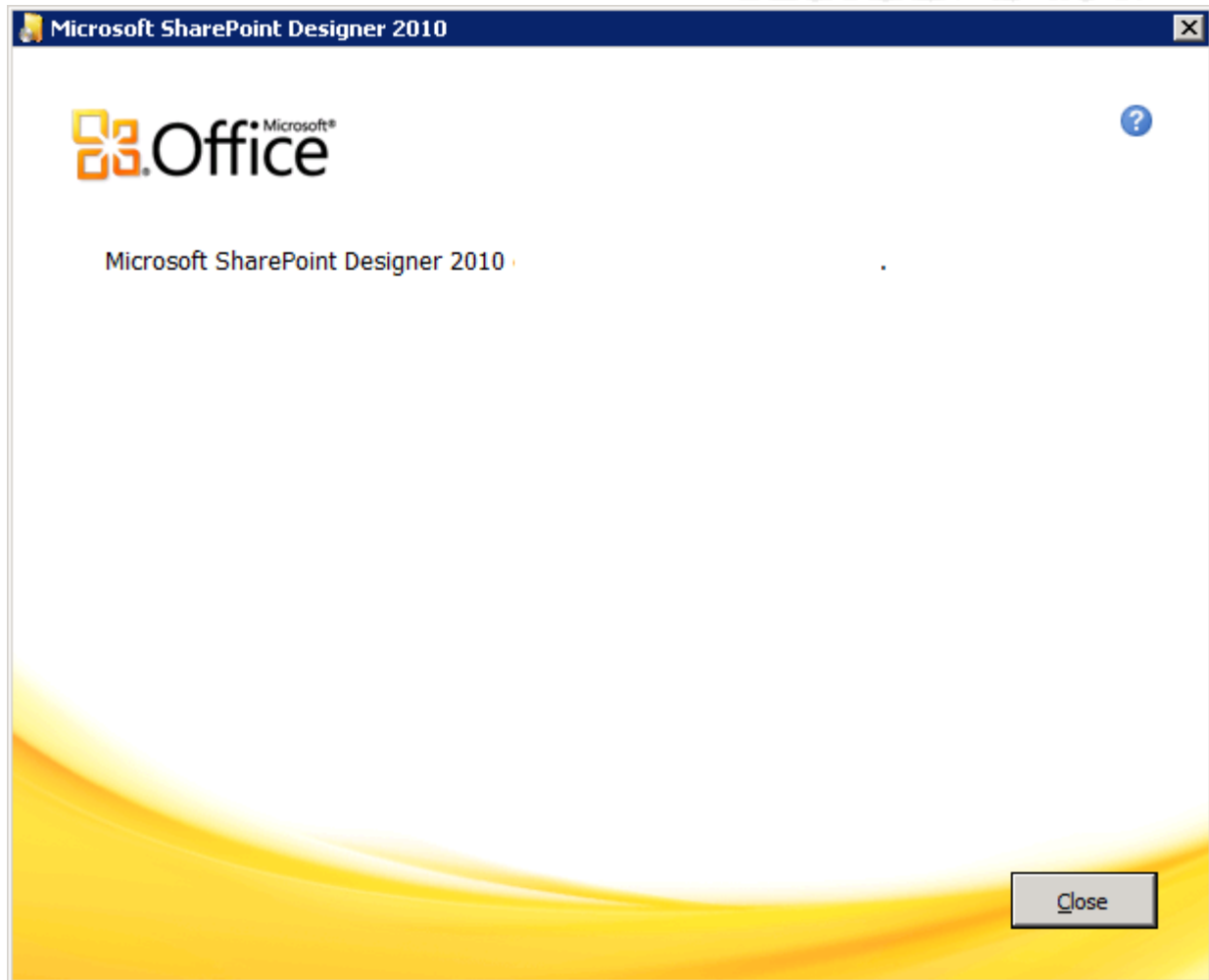


Figure 17

8. Upon completion of that click on Start, point to SharePoint, then you will see the Designer option there. From there, you can open Designer 2010. See Figure 18.

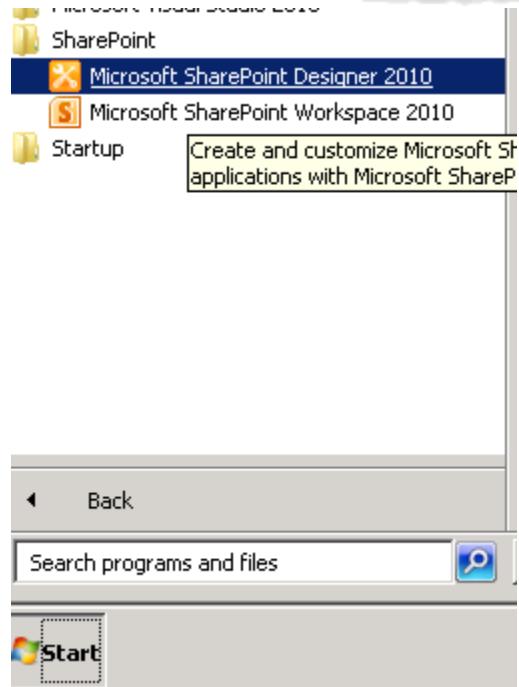


Figure 18

Share Point Designer 2010 Best Practices

SharePoint Designer 2010 is the tool to configure and customize SharePoint 2010. In this session, we discuss some of the development scenarios where SharePoint Designer 2010 can be helpful and identify situations where using SharePoint Designer 2010 may be more of a burden than a help. Listing 2 shows some of the case scenarios for SharePoint Designer 2010.

No.	Best Practices	Comments
1	Avoid customizing a Pages based on a File in the "14" Hive.	
2	Avoid installing SharePoint Designer 2010 on SharePoint Server	
3	Before developing a new Web Part, check the following a. Is it already provided Out of the Box? b. Whether a Data View Web Part can be used	

4	Do not edit the Out of the Box CSS files	
5	Do not start customizing from Production directly.	It is recommended to develop the solution in the developer interface, and then migrate the solution to the development server. After thorough testing, the same should be migrated to staging and then Production
6	Do not delete files which are part of a previous solution. In case of any issues, the developer should be in a position to roll back to a previous solution.	
7	Do not modify any system or application pages	
8	Do not apply a custom master page to system application pages. When we change the master page, we have two options, site master page and system master page. Try to not change the system master page.	
9	Avoid customizing pages which are frequently accessed and require less page load time.	
10	Check out the page and do your modifications If editing an existing page: a. Create a Copy b. Make your Alterations to your Copy c. Make your Copy active	
11	Web Part Properties in Web Part Zones are not versioned so be careful.	
12	Enable versioning for the artifacts that are being customized. In case of any issues, this will help with rolling back to a previous version.	

Listing 2

Recommendations

1. Do not allow end users (non-technical) to use SharePoint Designer

2. Reusable workflows should be exported to Visual Studio and then enhanced
3. Master page customizations done in SharePoint Designer should be exported to Visual Studio and a WSP should be created for deployment
4. In SharePoint Designer 2007, there was a packaging difference between .stp site templates and .wsp files. In 2010, only .wsp is the way to go.

Working with SharePoint Master Pages

The master pages in SharePoint 2010 utilize the features from ASP.NET 2.0. SharePoint master pages provide the interface and overall layout of the pages on a SharePoint site. The common elements of a page – its header, navigation links, Site Actions quick launch and top menu etc. are placed in the same areas regardless of the page we are viewing. This helps users immediately recognize where they are and helps them easily navigate the site. This continuity of look and feel is accomplished using master pages.

We will discuss the following points:

- Consolidate some of the basics of working with SharePoint 2010 master pages
- Highlight some of the changes between the previous version and the new version
- Provide insight into some of the more common and useful customizations

Master pages define the common layout and interface for SharePoint pages. The master page displays the persistent elements that you expect to see when you navigate from page to page on a site. These elements can include a company logo, title, navigation menu, search box, and the colors or graphics associated with the business or organization. The individual pages on a SharePoint site, Default.aspx, wiki pages, a list view are handled as content pages on the site. When these pages are viewed in a browser, they are combined with the master page to produce a single, continuous web page. The master page displays the persistent elements and layout.

Differences Between the Master Page and Content Pages

In Figure 19, you see the general areas of a page organized by the master page and those areas organized by the content page.



Figure 19

Even though the master page and content pages are separate, they are fused together at runtime to display as a single web page in the web browser. Both pages are ASP.NET pages. Master pages have .master as the file extension and they contain the necessary ASP code and content regions to render both pages on a SharePoint site.

If you open the SharePoint Site in SharePoint Designer, you can see the master page and content page separate from one another.

Benefits of Master Pages

Master pages provide functionality that developers have traditionally created by copying existing code, text, and control elements repeatedly; using framesets; using include files for common elements; using ASP.NET user controls; and so on. The advantages of master pages include the following:

- They allow you to centralize the common functionality of your pages so that you can make updates in just one place.
- They make it easy to create one set of controls and code and apply the results to a set of pages. For example, you can use controls on the master page to create a menu that applies to all pages.
- They give you fine-grained control over the layout of the final page by allowing you to control how the placeholder controls are rendered.
- They provide an object model that allows you to customize the master page from individual content pages.

Types of SharePoint 2010 Master Pages

In SharePoint 2010, if you open a site in the Designer using the menu option, you will see something like Figure 20.

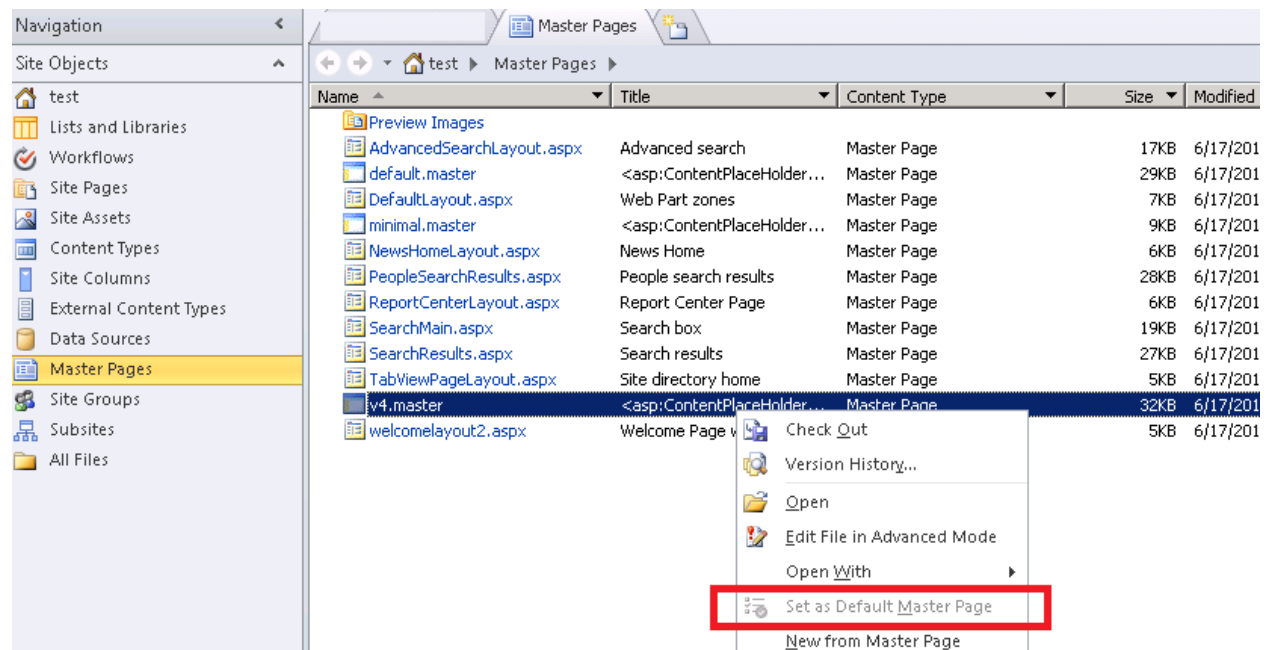


Figure 20

You can see the use of different type of master pages in SharePoint in Listing 3.

No	Master page Name	Description	Use
1	v4.master	Used for content and administration pages. Responsible for providing the	Team Site home page, list and library pages,

	Default Master Page	interface and layout for SharePoint 2010. Site actions are updated for 2010 and appear on left Ribbon bar is available	and site settings page.
2	minimal.master This the Minimal master page	Contains minimal SharePoint content. This can be used for full-screen functionality and when we need to customize sharePoint master page with rich UI. The minimal.master is an out-of-the box master page. These trimmed-down custom master pages are commonly referred to as Starter Master Pages in SharePoint 2010. This master page contains almost nothing. It is used by the Search Center and Office Web Applications.	The Search results page uses this
3	Publishing master page nightandday.master	Used for pages on a SharePoint Server publishing-enabled site.	The home page, about us, or press release page on a publishing intranet, extranet, or Internet-facing site.
4	2007 default master page Default. Master	Used to support legacy SharePoint sites, which haven't been visually upgraded to SharePoint 2010. Site actions on right side and are the same as SharePoint 2007 version No ribbon bar.	Home page, site pages, and list pages on a SharePoint 2007 site before visual upgrade can use this master page

Listing 3

Required Place Holders

When we work with SharePoint, we need the following place holders to render the page correctly in the browser. The table listed in Listing 4 shows the most frequently required place holders. Even though you don't require these place holders, don't delete any of them; just make its visibility property false.

No	Placeholder Control	Description
1	<code><asp:ContentPlaceHolder id="PlaceHolderQuickLaunchTop" runat="server"></code>	Top of the Quick Launch menu.
2	<code><asp:ContentPlaceHolder id="PlaceHolderQuickLaunchBottom" runat="server"></code>	Bottom of the Quick Launch menu.
3	<code><asp:ContentPlaceHolder id="PlaceHolderPageTitle" runat="server"/></code>	Title of the site.
4	<code><asp:ContentPlaceHolder id="PlaceHolderAdditionalPageHead" runat="server"/></code>	Placeholder in the head section of the page used to add extra components.
5	<code><asp:ContentPlaceHolder id="PlaceHolderBodyAreaClass" runat="server"/></code>	The class of the body area.
6	<code><asp:ContentPlaceHolder ID="SPNavigation" runat="server"></code>	Control used for additional page editing controls.
7	<code><asp:ContentPlaceHolder id="PlaceHolderSiteName" runat="server"></code>	Name of the site where the current page resides.
8	<code><asp:ContentPlaceHolder id="PlaceHolderPageTitleInTitleArea" runat="server" /></code>	Title of the page, which appears in the title area on the page. This is the place where we give Title
9	<code><asp:ContentPlaceHolder id="PlaceHolderPageDescription" runat="server"></code>	Description of the current page.

	<code>runat="server"/></code>	
10	<code><asp:ContentPlaceHolder id="PlaceHolderSearchArea" runat="server"></code>	Section of the page for the search controls.
11	<code><asp:ContentPlaceHolder id="PlaceHolderGlobalNavigation" runat="server"></code>	Breadcrumb control on the page.
12	<code><asp:ContentPlaceHolder id="PlaceHolderTitleBreadcrumb" runat="server"></code>	Breadcrumb text for the breadcrumb control.
13	<code><asp:ContentPlaceHolder id="PlaceHolderGlobalNavigationSiteMap" runat="server"></code>	List of subsites and sibling sites in the global navigation on the page.
14	<code><asp:ContentPlaceHolder id="PlaceHolderTopNavBar" runat="server"></code>	Container used to hold the top navigation bar.
15	<code><asp:ContentPlaceHolder id="PlaceHolderHorizontalNav" runat="server"></code>	The navigation menu that is inside the top navigation bar.
16	<code><asp:ContentPlaceHolder id="PlaceHolderLeftNavBarDataSource" runat="server" /></code>	The placement of the data source used to populate the left navigation bar.
17	<code><asp:ContentPlaceHolder id="PlaceHolderCalendarNavigator" runat="server" /></code>	Date picker used when a calendar is visible on the page.
18	<code><asp:ContentPlaceHolder id="PlaceHolderLeftNavBarTop" runat="server"/></code>	Top section of the left navigation bar.
19	<code><asp:ContentPlaceHolder id="PlaceHolderLeftNavBar" runat="server"></code>	Quick Launch bar.
20	<code><asp:ContentPlaceHolder id="PlaceHolderLeftActions" runat="server"></code>	Additional objects above the Quick Launch bar.

21	<code><asp:ContentPlaceholder id="PlaceholderMain" runat="server"></code>	Main content of the page.
22	<code><asp:ContentPlaceholder id="PlaceholderFormDigest" runat="server"></code>	Container where the page form digest control is stored.
23	<code><asp:ContentPlaceholder id="PlaceholderUtilityContent" runat="server"/></code>	Additional content at the bottom of the page. This is outside of the form tag.
24	<code><asp:ContentPlaceholder id="PlaceholderTitleAreaClass" runat="server"/></code>	The class for the title area (in the head tag). Customizations that add a WebPartZone in a content tag to this placeholder will cause an error.
25	<code><asp:ContentPlaceholder id="PlaceholderPageImage" runat="server"/></code>	Not part of the UI, present for backward compatibility.
26	<code><asp:ContentPlaceholder id="PlaceholderTitleLeftBorder" runat="server"></code>	Not part of the UI, present for backward compatibility.
27	<code><asp:ContentPlaceholder id="PlaceholderMiniConsole" runat="server"/></code>	Not part of the UI, present for backward compatibility.
28	<code><asp:ContentPlaceholder id="PlaceholderTitleRightMargin" runat="server"/></code>	Not part of the UI, present for backward compatibility.
29	<code><asp:ContentPlaceholder id="PlaceholderTitleAreaSeparator" runat="server"/></code>	Not part of the UI, present for backward compatibility.
30	<code><asp:ContentPlaceholder id="PlaceholderNavSpacer" runat="server"></code>	Not part of the UI, present for backward compatibility.
31	<code><asp:ContentPlaceholder id="PlaceholderLeftNavBarBorder" runat="server"></code>	Not part of the UI, present for backward compatibility.
32	<code><asp:ContentPlaceholder id="PlaceholderBodyLeftBorder" runat="server"></code>	Not part of the UI, present for backward compatibility.

34	<pre><asp:ContentPlaceHolder id="PlaceHolderBodyRightMargin" runat="server"></pre>	Not part of the UI, present for backward compatibility.
----	--	--

Listing 4

How to Customize the Master Page

Before you start customizing master pages, it is very important to understand how default master pages are stored and managed in SharePoint 2010 as this can affect your SharePoint Server performance. The default primary master page, v4.master, is part of a site definition that is stored on the server running SharePoint 2010. The location as you know it is "**\Program Files\Common Files\Microsoft Shared\web server extensions\14\TEMPLATE\GLOBAL**". The site definition files are cached in memory on the server when first accessed at runtime and the information in each file is pulled from the cache for each subsequent request. This allows the page to be reused across multiple sites and reduces unnecessary storage and retrieval of the page. All of this improves the performance and scalability of SharePoint.

When a master page is customized for the first time, a new edited copy of the page is stored in SharePoint's content database instead of the server file system. That is why you see a message the first time you customize a master page in SharePoint Designer 2010, warning that you're about to customize a page and it will no longer be based on the site definition. This won't affect other SharePoint sites that are still using the original master page. If, at any time, you want to revert to the original master page and no longer use your customized master page, you can reset the master page to its site definition. This brings your site back to its original state and at the same time, creates a copy of your customized master page and places it in the Master Page Gallery in case you want to use it again. One more thing you want to keep in mind while working with master page is to check it out and check in each version when you feel you achieved a successful milestone. This will help you to go back to any previous version if anything happened to your customized master page.

We have two options, as listed in Listing 5, to use when we going to customize a master page.

No	Approach	Advantages	Disadvantages	When to Use
1	By customizing a copy of v4.master	We are using the existing SharePoint look and feel, with its features, brand, and design elements, and customizing it	it can be challenging as you try to understand how everything works in the v4.master page	For minimal branding or customization efforts, like changing the header of your SharePoint pages.
2	By creating a blank custom master page or using the Minimal.Master Page	This approach is more flexible when you know the design which you're trying to create and you're working off existing plans and prototypes.	This approach can be challenging because you need to know how to implement a design in SharePoint	When we are creating the whole design from scratch.

Listing 5

Customizing Master Pages in SharePoint Designer 2010

To customize master pages in SharePoint Designer 2010, open SharePoint Designer 2010 from the site itself or from your Start menu. Then go to file and select open web site. This will open your site in SharePoint Designer 2010; click Master Pages in the Navigation pane. If you don't see Master Pages in the Navigation pane, make sure they are enabled on the site. We have explained the how-to in the Governance section. You can see the following master pages by default: v4.master, minimal.master, and default.master in the Master Page Gallery as shown in Figure 21.

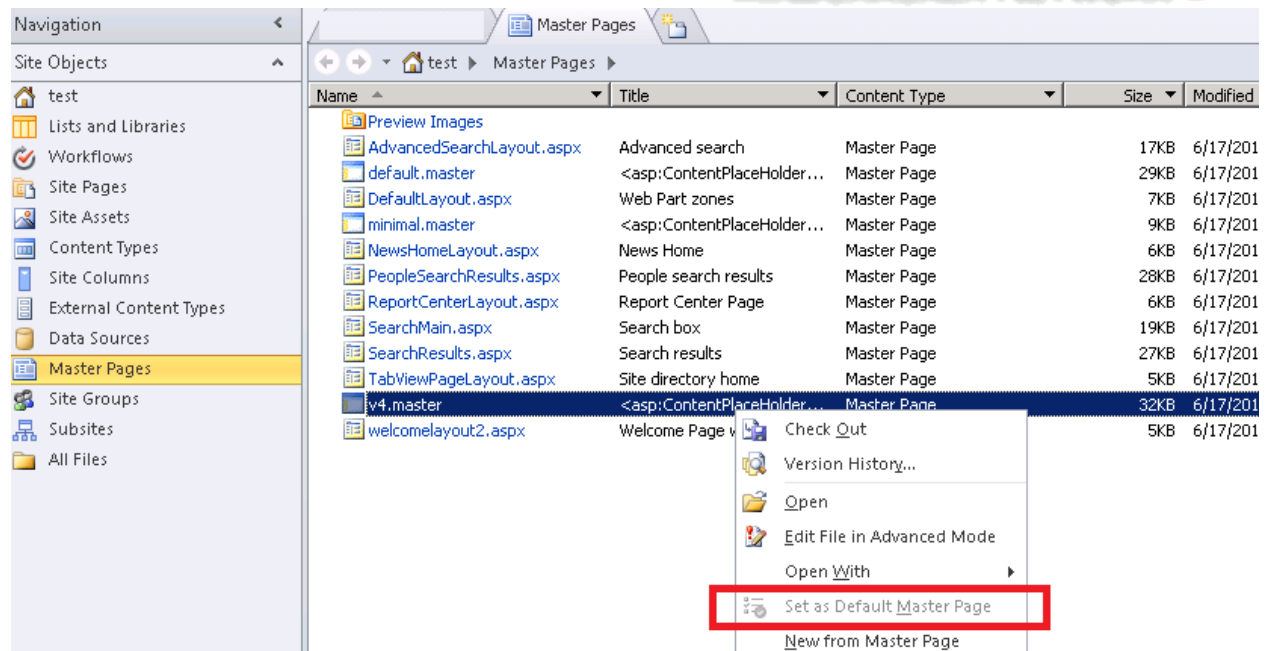


Figure 21

To view the contents of the primary master page, click v4.master or first make a copy of it and then open the copy by checking it out as explained earlier. This opens the master page in the SharePoint Designer 2010 full-screen page editor where you can edit the page in Design view, Split view, or Code view. You can see how the master page looks like as you customize it. It is however better to see the page in a browser to get better clarity on the page customization. There are also tools available for you to navigate and customize tags, styles, scripts, and content regions as shown in Figure 22.

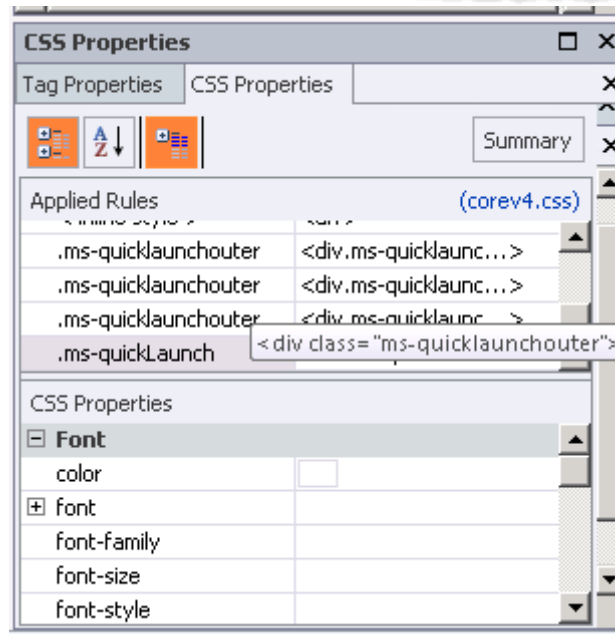


Figure 22

When customizing a master page, you can add new content and functionality, change the placement of objects on the page, resize the Web Part Zones, etc. For the most part, you're changing the layout and positioning of elements on the page. If you want to customize the colors, graphics, text formatting, and other style attributes, we recommend you create your own custom styles and add them to the master page directly or place them in an external Cascading Style Sheets (CSS) that you attach to the master page. You can even keep the CSS file in the document library.

While checking the page when you progress with designer, keep in mind to check other web part zones, document library, lists, calendar etc. to make sure they are also working fine. Your customized master page needs to be able to render this content and functionality while providing a good user experience.

How to Deploy a Master Page

As you know, when we are finished customizing a master page, the next step is to deploy it to SharePoint. One option is to save and reuse the master page by applying it to individual sites from SharePoint Designer 2010, or if you're running SharePoint Server 2010, apply it to the site collection from the Site Master Page Settings page.

A more scalable approach is to deploy the master page as a solution package and then activate it as a SharePoint feature. Then, when you want to install or uninstall the master

page, you just activate or deactivate that feature. This approach requires using Microsoft Visual Studio 2010 to create the necessary modules, deployment XML, and event receivers for the feature. We are not giving more explanation on this approach as it is out of scope of this book.

Working with Pages

Publishing in Microsoft SharePoint Server 2010 is fundamentally a means of authoring and displaying information by using a set of templates. With regard to publishing in SharePoint Server 2010, there are two fundamental templates to understand: master pages and page layouts. Master pages and page layouts dictate the overall look and feel of your SharePoint site. Master pages contain controls that are shared across multiple page layouts, such as navigation, search, or language-preference for multilingual sites. Page layouts contain field controls and Web Parts. By using master pages, you can provide a consistent look and feel for your site. You can use master pages to position items that must be shared by all pages, such as navigational controls, company logos, and copyright notices. Within master pages, you can use user controls (.ascx controls), Web server controls, and Web Parts. The top-level SharePoint Server site for a site collection hosted on SharePoint Server 2010 has a special document library called the Master Page and Page Layout Gallery. All page layouts and master pages are stored in this document library. The Master Page and Page Layout Gallery support versioning and workflow, so you can use those features for your master pages and page layouts. Page layouts can be used by all page instances that are based on that page layout. Master pages can be used by all page instances in a site. Page instances based on the same page layout in different sites can use different master pages.

To do this, navigate to the top level site of the Site Collection then Site Actions / Site Settings / Master pages (under the Galleries section). Then switch over to the Documents tab in the Ribbon and then click New > Page Layout. See Figure 23.

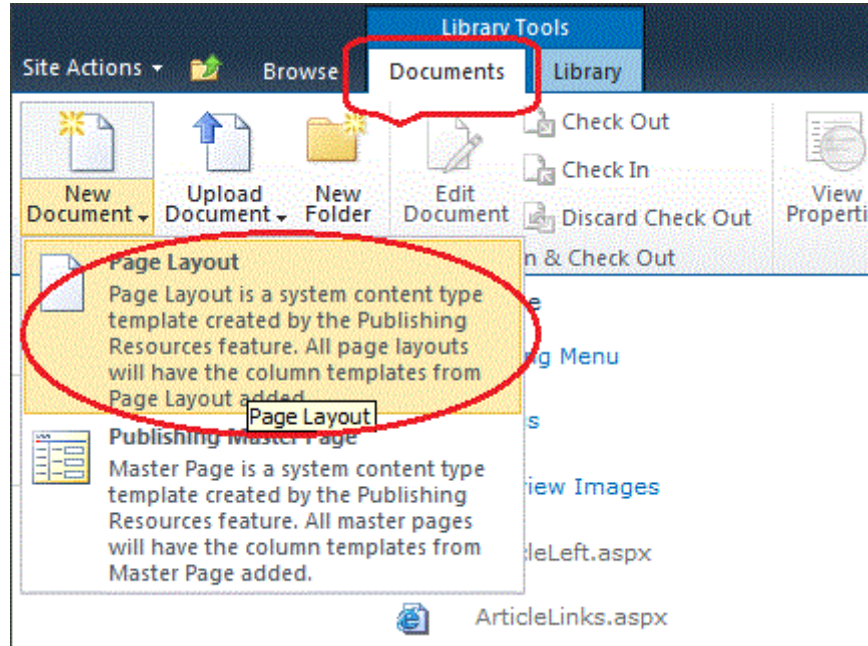


Figure 23

Select an Article Content Type to associate with; give it an URL and Title. There's also a link on this page to create a new Content Type. You might be interested in doing this if you wanted to, say, add more editing fields or metadata properties to the layout. See Figure 24.

Content Type Group

Content Type Name

Description:
 Page is a system content type template created by the Publishing Resources feature. The column templates from Page will be added to all Pages libraries created by the Publishing feature.
[Create a new site content type](#)

URL Name:
 .aspx

Title:

Description:

Variation Labels

Available Labels		Labels for Layout
	Add > < Remove	

Figure 24

Click OK.

Open the page that we created in SharePoint Designer, as you see in Figure 25.

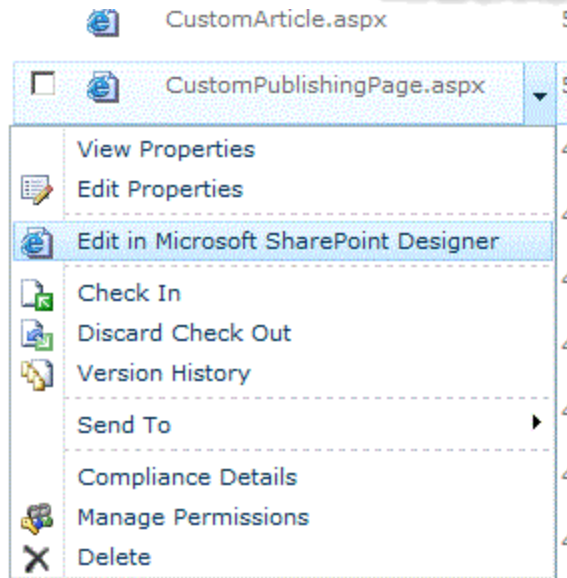


Figure 25

You can choose to build your page manually by dragging your SharePoint Controls onto the page and laying them out as you'd like. You can even copy another article page and paste it into the page then arrange it at your convenience.

Check-in, publish, and approve the new Page Layout. Side note: We like to add the Check-In/Check-Out/Discard or Undo-Checkout buttons to all of my Office Application's Quick Access Toolbars for convenience.

Almost there! Navigate to your publishing site, in this case the Enterprise Wiki Site, then go to Site Actions > Site Settings > Page layouts and Site Templates (Under Look and Feel). You'll be able to make the new Page Layout available for use within the site. See Figure 26.

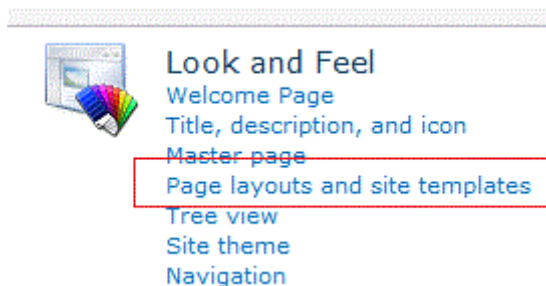


Figure 26

Specify the page that we created and click OK. See Figure 27.

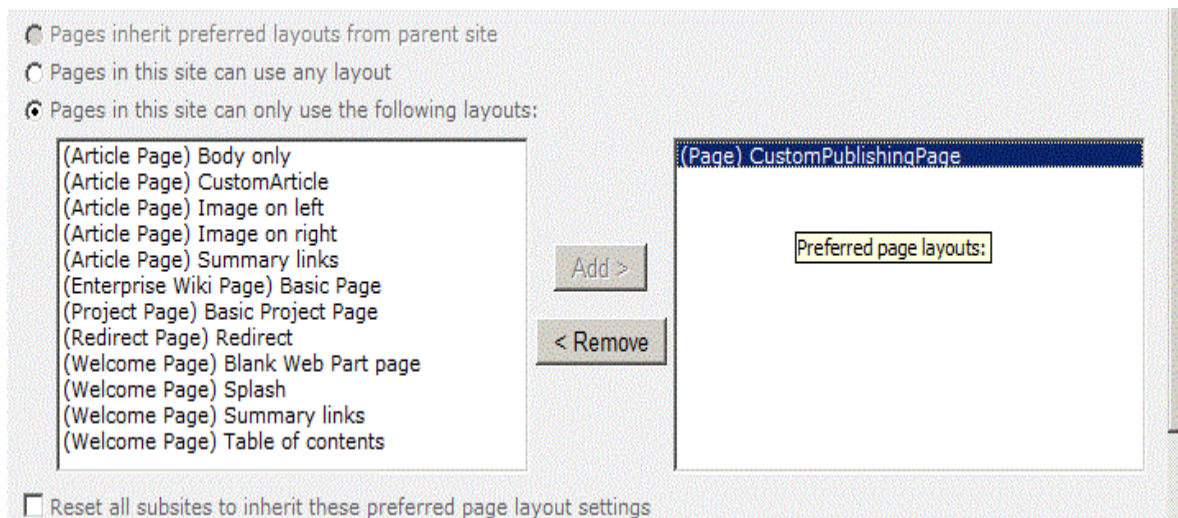


Figure 27

You are almost done.

Go to Site Actions then select View all site Content.

Then click on the Create link. Select Page from the left navigation.

Then click on the Publishing Page and click on Create. See Figure 28.

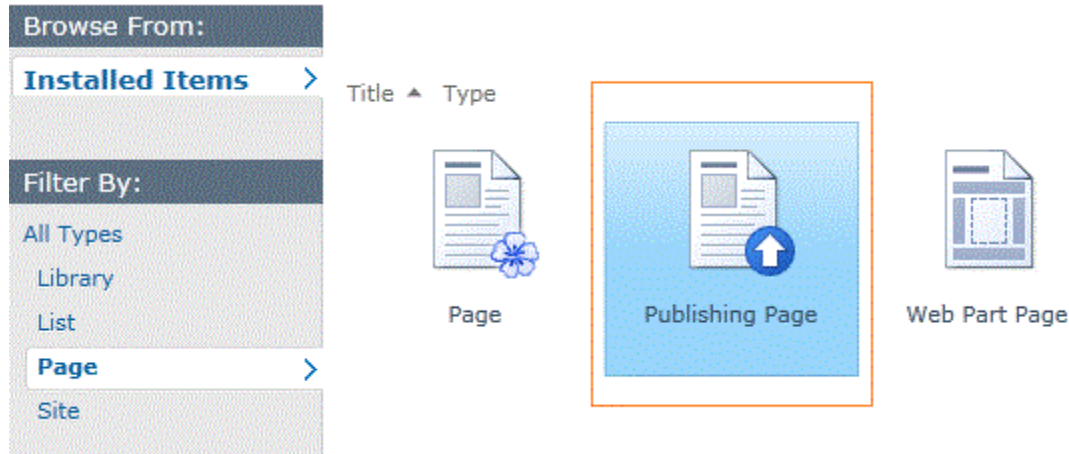


Figure 28

You can see that our custom page template was also added; select that and create a new publishing page from the new page template. See Figure 29.

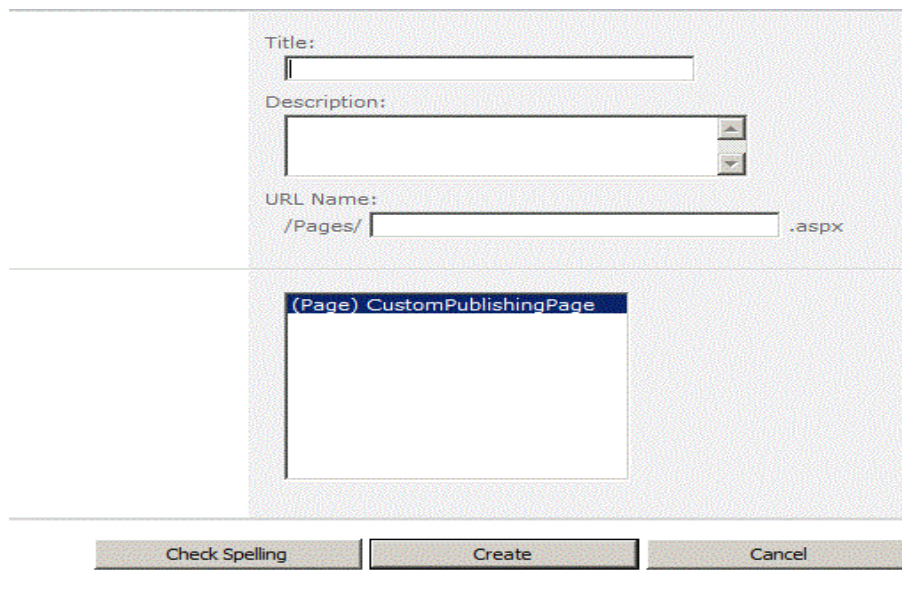


Figure 29

Enable AJAX Options in List View Web Parts

When we add a Web Part from the List and Libraries category, this list basically shows every List and Document Library we have on the SharePoint site. Actually, behind the scenes, a Data View Web Part is being used to display the List or Document Library data.

As you can see in Figure 30, we used an employee custom list and added it to the web part zone.

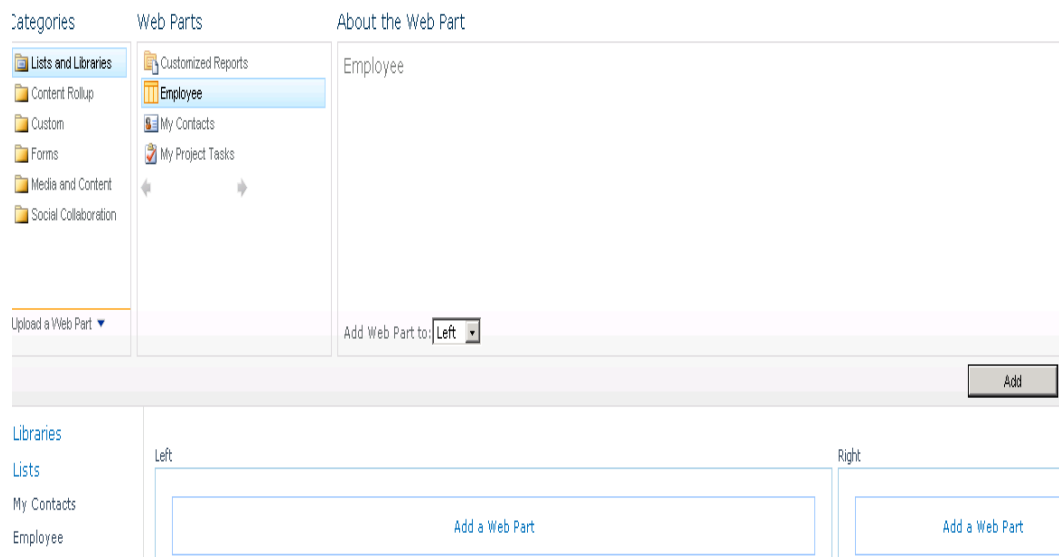


Figure 30

We have added this web part into the zone as you see in Figure 31.

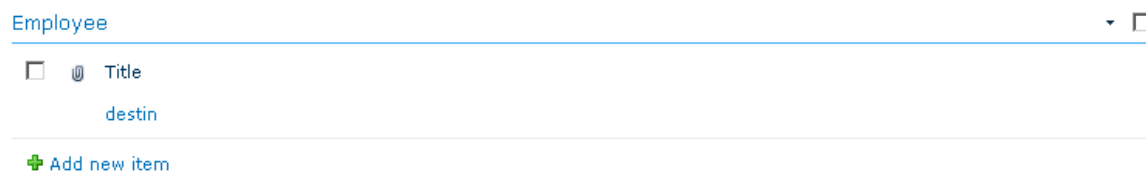


Figure 31

When you edit a Web Part once it has been added to a page, you'll notice there is a new **AJAX Options** section in the Web Part properties. AJAX stands for Asynchronous JavaScript and XML and is a web development technique to build more interactive, rich web sites. The AJAX Options are disabled by default, but we can enable this as shown in Figure 32.

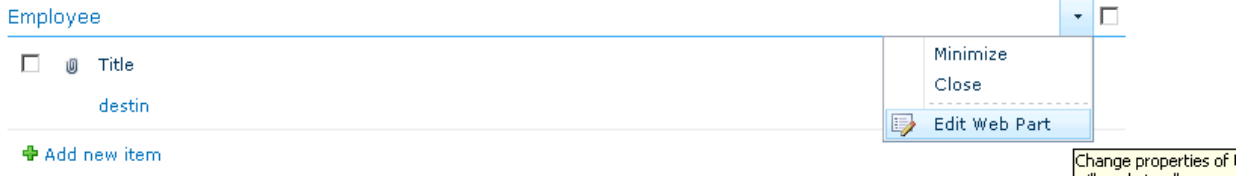


Figure 32

Once you open in Edit mode, you can see a screen like Figure 33.

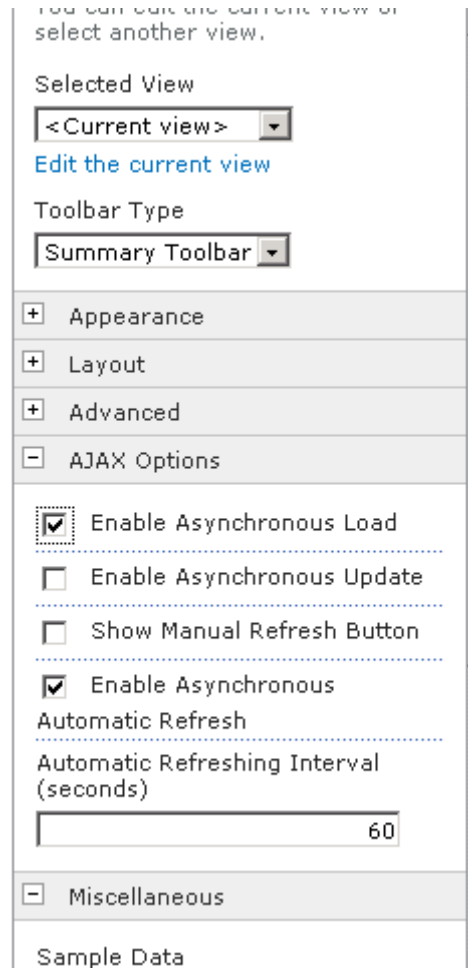


Figure 33

1. **Enable Asynchronous Update:** Enabling this option will make paging, sorting, filtering work without full page refreshes.
2. **Show Manual Refresh Button:** Enabling this option will show an icon to allow the user the refresh the data manually, without refreshing the rest of the page.
3. **Enable Asynchronous Automatic Refresh:** When enabled, the Web Part will dynamically refresh the date it's showing, without completely reloading the page. The interval can be specified in the textbox as shown above.
4. **Automatic Refreshing Interval:** Specifies the interval used in the previous option.
5. **Enable Asynchronous Load:** The Web Part will initially be displayed without any data in it. But once the page is loaded, the Web Part will asynchronously fetch the data afterwards. When the data is being loaded, the Web Part will display an animation. This option can be used to reduce the page load time.

Views and Forms Web Part in SharePoint Designer 2010

In SharePoint Designer 2010, we can create list views and data views as well as list forms and data forms. You create these views and forms to help site visitors interact with the data on the site and, in the case of a custom solution, manage how data is presented and modified. Every view and form you create in Microsoft SharePoint Designer 2010 is stored in a Web Part on an ASPX page. The Web Part contains the necessary code to render the view or form on the page as well as allow users to change the appearance of the data or submit data to the data source. Depending on which view or form you choose and the type of data source you're using, a different Web Part is added to the ASPX page.

The table in Listing 6 shows a quick comparison of the various Web Parts that are used for the views and forms you create in SharePoint Designer 2010. We will discuss more about each of them in the following sections.

No	Web Part Name	Data Source can be connected with	Use	Advantages/Dis Advantages
1	XSLT List View Web Part (XLV)	SharePoint lists SharePoint libraries External Lists	The standard and often default view associated with lists and libraries. The XLV is also used when you create a view on a page where the data source is a list or library.	Full ribbon support in SharePoint, so the view can be customized in browser/Designer.XLVs, however, Can be used for SharePoint lists and libraries, no other data sources.
2	XSLT List Form Web Part (XLF)	External lists	External lists in SharePoint.	It can be used to write data to an external data source. But it can only be customized by using SharePoint Designer.
3	Data Form Web Part (DFWP)	SharePoint lists SharePoint libraries Databases Web Services XML files	Used by default when you create a data view to a data source other than a list and library. You can use a DFWP for lists and libraries when Data Source Details pane is used.	Can be used to display data from any data source- including lists, libraries, databases, and web services.
	Data View Web Part (DVWP)	SharePoint lists Share Point libraries Database	This is mostly used in 2007 and supported in an upgrade from 2007 to SharePoint 2010.	Not used for new sites in SharePoint 2010, but available for backwards compatibility

		Web Services XML files		
	List Form Web Part (LFWP)	SharePoint lists SharePoint libraries	The default list form used for SharePoint lists and libraries.	Easy to use form for users. To customize the form, we need to replace them with a DFWP.
	List View Web Part (LVWP)	SharePoint lists SharePoint libraries	Used for highly sophisticated view, like the Calendar view, Gantt view, and Datasheet view.	Well formatted and powerful view of the list or library. To customize the view, you must use SharePoint Designer
	InfoPath Form Web Part (IFWP)	SharePoint lists	Used to host InfoPath-based list forms. The IFWP is created when you customize list form in InfoPath 2010.	Highly customizable and powerful forms generated by InfoPath 2010 form templates. Cannot be customized in SharePoint Designer. Need to use InfoPath 2010

Listing 6

Working with Data View Web Parts

A Data View Web part is a great way to display data with filtering, grouping, and user desired formatting. It was the most useful web part in SharePoint 2007 and has been improved a lot and is extended in SharePoint 2010. In SharePoint 2010, you have a XSLT List view web part and a data view web part. We will discuss about a XSLT list view later in this book. The OLD data view web part is still available and can be used in the same manner as we used it in SharePoint 2007. To add a data view web part onto a page, follow the following steps.

Open SharePoint Page and in Site Actions, select ->edit in SharePoint Designer. Or use the method we mentioned earlier

1. Select a zone on the Page where you want to add your Data view web part.
2. Select the Insert tab -> click on Data View.

3. In the drop-down of the Data view menu click on "Empty Data view. This will insert an Empty Data view in your selected zone.
4. Click on "Click here to select a data source" to select the data source or list.
5. Select the list/library you want to display in the data from.
6. Drag and drop the columns in the data view web part from the right hand column list (Data Source we selected)

See Figure 34.

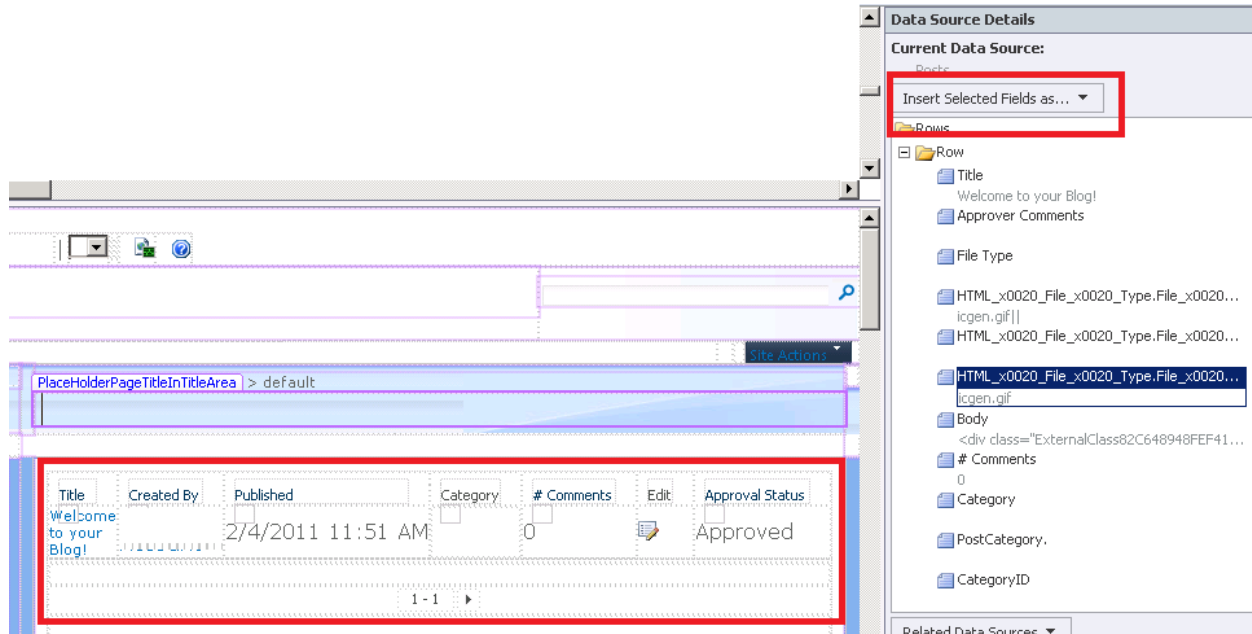


Figure 34

You will see you have the Data view web part on your page. You can use the formatting options available on your ribbon or you can modify the XSLT either in Designer or on the page itself by editing the web part properties in the Browser.

How to Preview a Data View Web part

When we use designer to edit data view in SharePoint 2010 on the **Design** tab, there's a section called **Preview**. This title is a little ambiguous, because there are some pretty powerful options in this section. We can do some inline editing. As is deceptive, the view does not need to be created in SharePoint Designer in order to use inline editing; it can just be a regular browser view. See Figure 35.

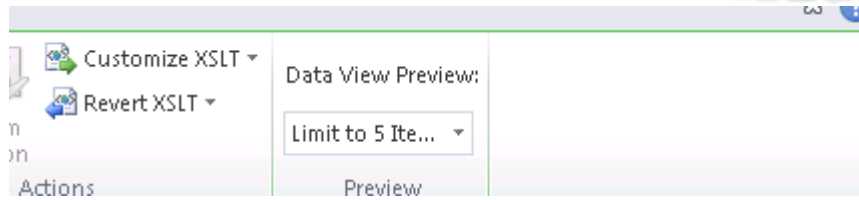


Figure 35

How to Use a Data View Web Part to Show Data from SQL Server

In this demo, make sure you have a database with some data and you can connect to the data. See Figure 36.

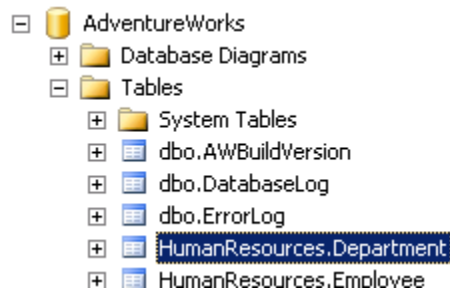


Figure 36

1. Open your designer and open your web application.
2. Create a web part page and give it a suitable name.
3. Now create a new data source by clicking the data source option in the left navigation and choose database. See Figure 37.

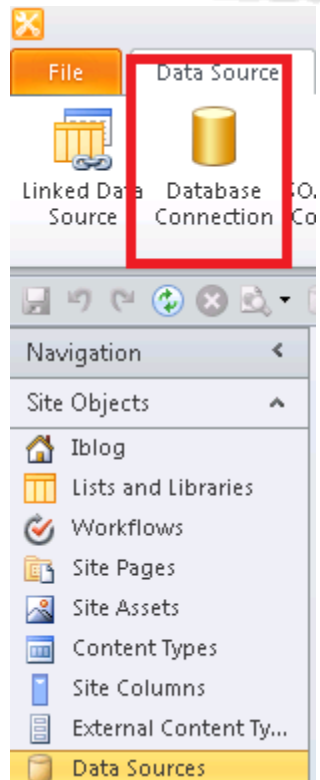


Figure 37

4. Click on Configure Database Connection. See figure 38.

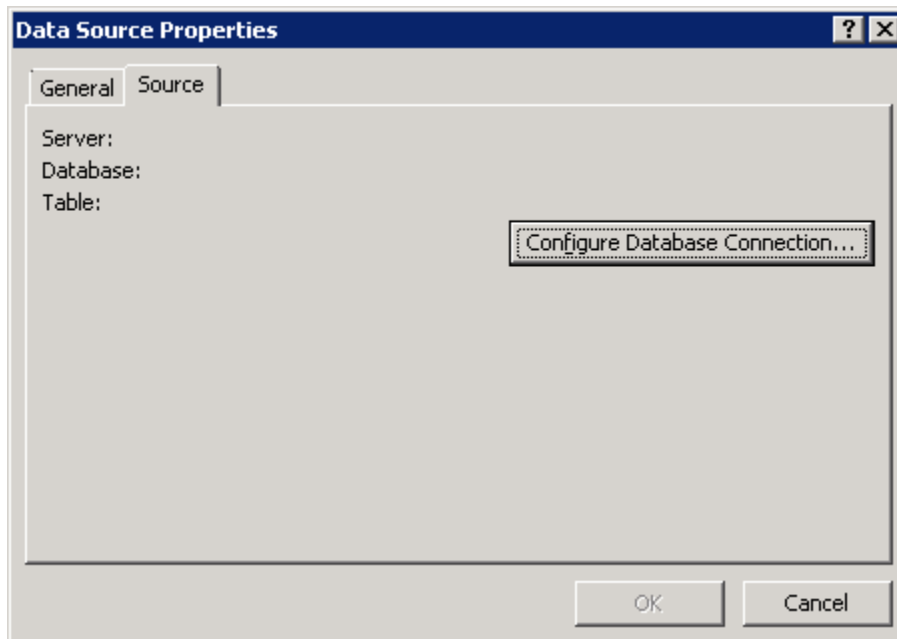
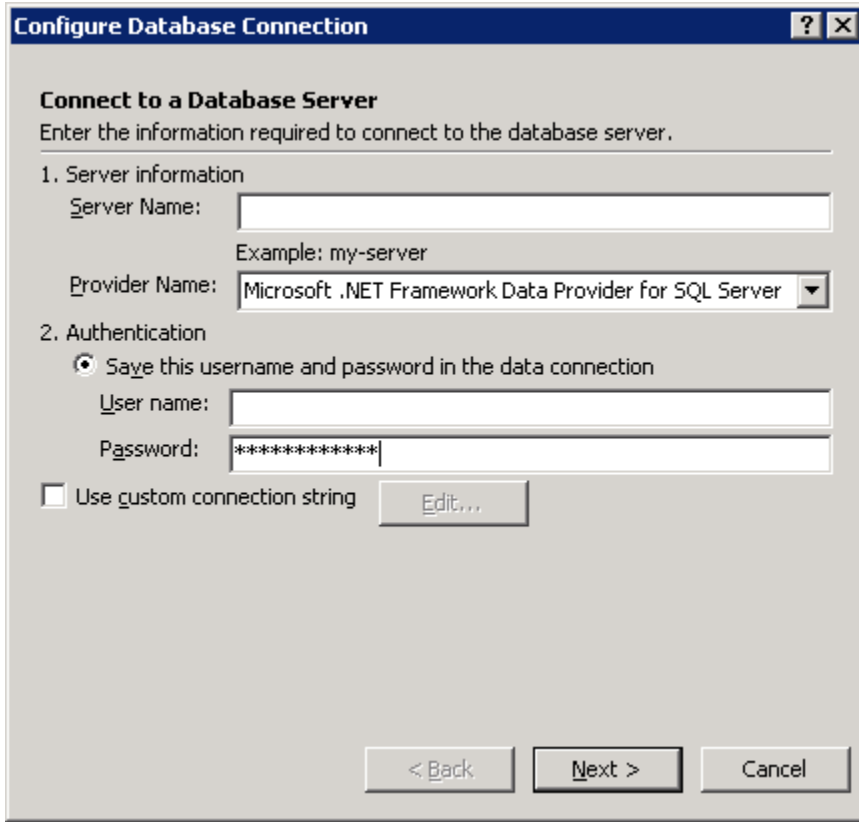


Figure 38

5. Give your Database credentials and click OK. See Figure 39.



Configure Database Connection [?] [X]

Connect to a Database Server
Enter the information required to connect to the database server.

1. Server information

Server Name:
Example: my-server

Provider Name:

2. Authentication

☒ Save this username and password in the data connection

User name:

Password:

☐ Use custom connection string

< Back Next > Cancel

Figure 39

6. Click Ok on the message and wait a few seconds for the DB list to load
7. Once loaded, select your Database name and table. See figure 40.

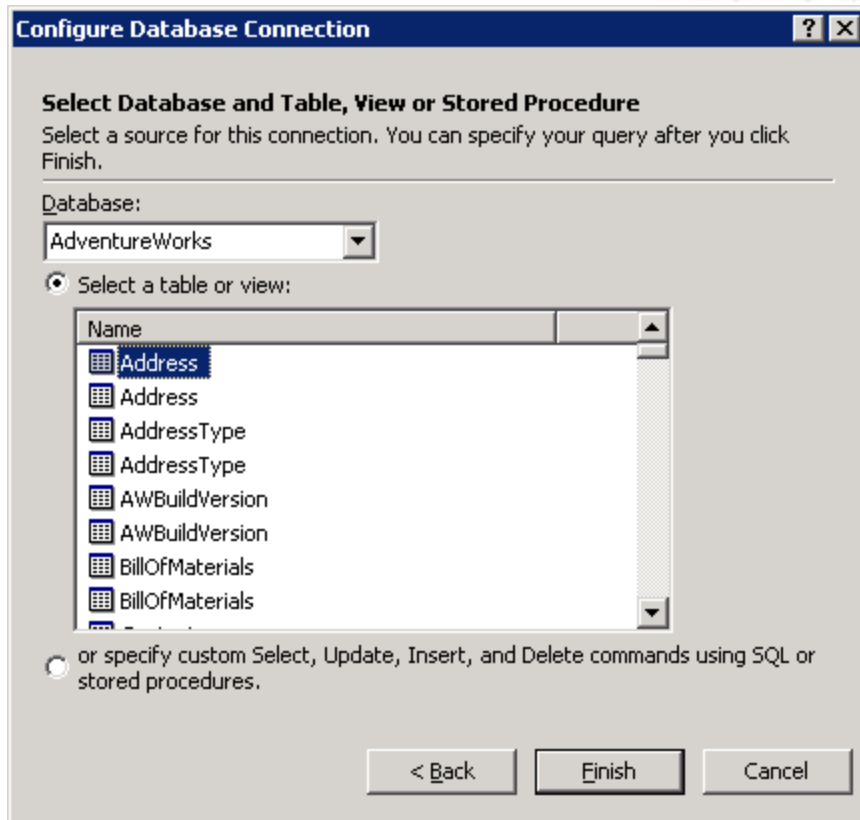


Figure 40

8. You have the option of choosing columns, filtering and sorting from the data source. See figure 41.

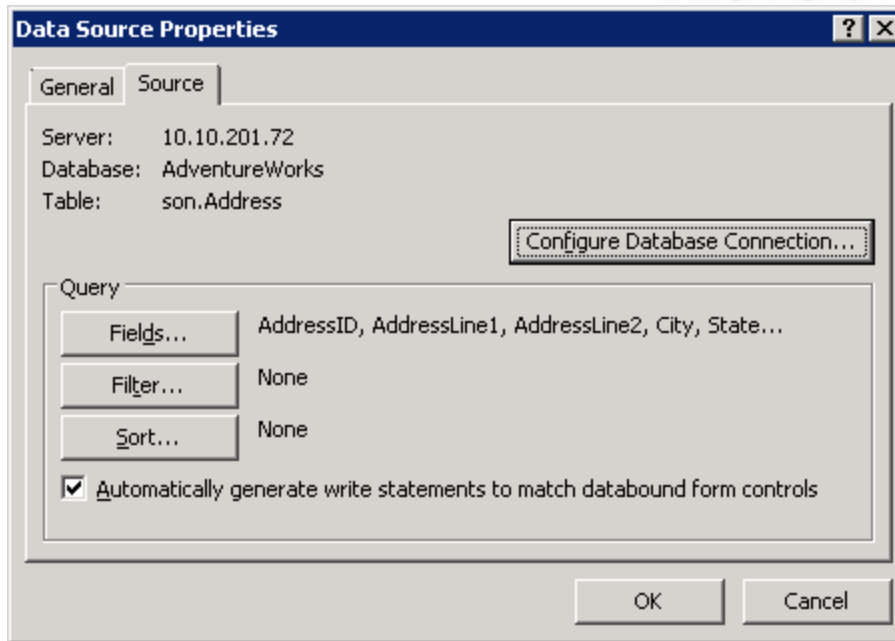


Figure 41

9. We are now going to add the custom database source to the web part page we just created in step 2 and display the data. Edit the file in SPD 2010. See Figure 42.

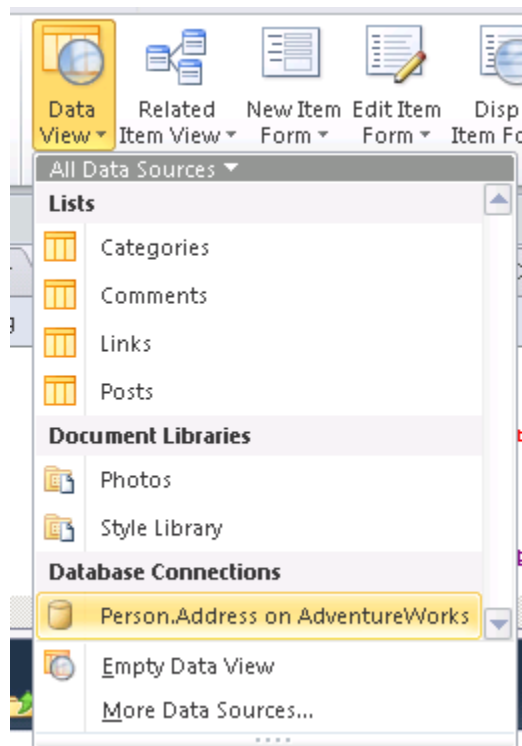


Figure 42

10. Choose insert from the ribbon bar and insert a data view web part in the Main Placeholder.
11. Now you can see the data source we created. Add the same to your page.
12. Preview it in the browser.

Data View Migration Issues

We are familiar with the SharePoint data view web part. Most of us may have experienced issues when we migrate a data view from one server to another server as a Site Template or back/restore. We are explaining how to resolve these issues related with data view migration in SharePoint

Error

The following error message may be a common issue you get:

"Unable to display this Web Part. To troubleshoot the problem, open this Web page in a Windows SharePoint Services-compatible HTML editor such as Microsoft Office SharePoint Designer. If the problem persists, contact your Web server administrator"

SharePoint Designer error: *"The server returned a non-specific error when trying to get data from the data source. Check the format and content of your query and try again. If the problem persists, contact the server administrator."*

Cause:

This is a known issue, actually. The issue is caused by an incorrect ListName, which is a GUID id for the list's instance on a given site instance. When backup/restore or template movement occurs, this GUID may change, therefore the broken data views are pointing to the incorrect data source. If you create a DVWP in SharePoint designer 2007, it will append the GUID of the LIST in the code and would link to the DVWP. If you try to create a new List it will always create a new GUID which will never match up with the DVWP and it breaks.

Solution 1



```
'alse" Description="" PartImageSmall="" DataSourceID="" MissingAssembly="Cannot import this Web Part." Frame
Command="<View><Query><Where><Eq><FieldRef Name="ID"/><Value Type
ParameterKey="ListID" PropertyName="ParameterValues" DefaultValue="{AFA25ABA-7FDB-4DAD-8A7C-12FF0F0E5DE3}"
ParameterKey="ListID" PropertyName="ParameterValues" DefaultValue="{AFA25ABA-7FDB-4DAD-8A7C-12FF0F0E5DE3}"
ParameterKey="ListID" PropertyName="ParameterValues" DefaultValue="{AFA25ABA-7FDB-4DAD-8A7C-12FF0F0E5DE3}"
```

Open the page having the problem in SharePoint Designer 2010 Search for "ListID" and replace it with "ListName" and change the "DefaultValue=" to your List name. This will fix it.

```
ectCommand="<View><Query><OrderBy><FieldRef Name="Created_x0020_Date" Ascen
ter ParameterKey="ListName" PropertyName="ParameterValues" DefaultValue="Cartoon Corner" Name="ListName">
ter ParameterKey="ListName" PropertyName="ParameterValues" DefaultValue="Cartoon Corner" Name="ListName">
ter ParameterKey="ListName" PropertyName="ParameterValues" DefaultValue="Cartoon Corner" Name="ListName">
```

Solutions 2

1. Start SharePoint Designer, and open the page containing broken data view(s).
2. In SharePoint Designer's menu, click Task Panes -> Data Source Library.
3. From the Data Source Library task pane, drag in a new instance of the broken data view's data source (example: Announcements list) onto the page. This will create a new and working data view.
4. Highlight the newly inserted data view, and go into split view (half code, half design). From the code portion highlighted, look for the ListName property. Note: it should be on the first line of the highlighted code, but you may have to scroll to the right in order to see it.
5. Once you find the ListName property, copy everything in between { ... } (without the braces). Store that value, as we will need it later (example: paste into Notepad).
6. Remove the newly inserted data view.
7. Highlight the broken data view, and go into split view (half code, half design). From the code portion highlighted, look for the ListName Property. Note: it should be on the first line of the highlighted code, but you may have to scroll to the right in order to see it.

8. When you have found the ListName property, copy everything between { ... } (without the braces).
9. In SharePoint Designer's menu, click Edit -> Replace. In the "Find What" portion, paste in the broken data views ListName value. In the "Replace With" portion, paste in the previous working data view's List Name value (example: the one stored in Notepad). Click Replace All, and let the process finish. Click close.
10. Perform steps 3 - 9 for each respective data view.

XSLT List View Web part

SharePoint 2010 has a heavy dependence on the XSLT List View Web Part. This replaces the MOSS List View Web Part. In MOSS, the Content Query Web Part was heavily utilized and the XSLT List view Web part is often a better option in SharePoint 2010. The old List View Web part used CAML to provide custom views of the data or lists.

Customizing the XSLT List view Web part display is done using XSLT. The XSLT List View Web part has a default XSLT for display located in the 14 hive at

_layouts/xsl/main.xsl.main.xsl uses vwstyles.xsl and fldtypes.xsl that does most of the work on all the XSLT List View Web part.

The XSLT List View Web Part, or XLV, provides a powerful, highly customizable view of a SharePoint list or library. The XLV is very unique among the Web Parts for views and forms because it can be customized using SharePoint Designer 2010 as well as in the browser using the SharePoint ribbon. As a result, both site designers and site users can customize the view and even take turns customizing it to meet their needs.

When you work with an XLV, there are a few things to keep in mind. The Web Part is designed for SharePoint lists and libraries, which includes external lists. An XLV is not, however, compatible with other data sources, like database connections, Web service connections, or XML files. The view itself is not a form, in that it's not intended to be used as a new item or edit item form, though you can enable inline editing so users can update individual fields in the view if they like.

How to Create an XSLT List View Web part in Designer 2010

1. Open SharePoint Designer 2010.
2. Click the Open Site button and in the Open Site dialog window, type the URL
3. Open the page where you want to add the XSLT List view web part

4. In the ribbon menu, click Edit File, and then click Edit File in Advanced Mode.
5. Click the Insert tab on the ribbon, and then click Data View.
6. In the drop-down list, click Employees. The SharePoint Designer adds the XsltListViewWebPart Web Part to the Home page. See Figure 43.

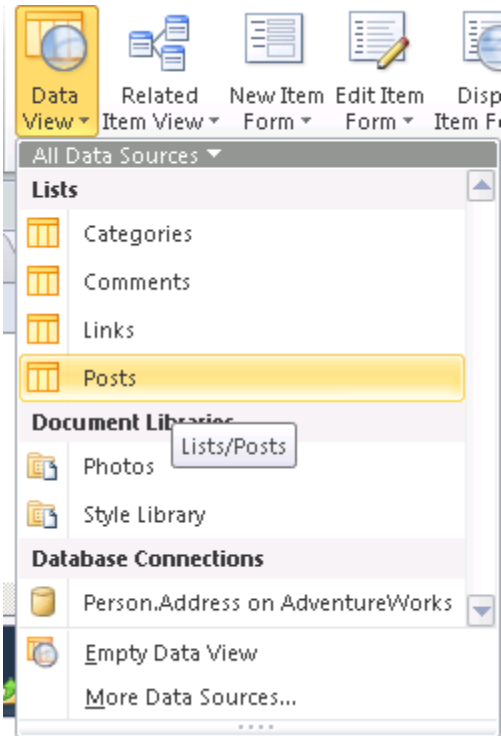


Figure 43

7. Click **Save** at the top of the page. If you are prompted for confirmation, click **Yes**. See Figure 44.

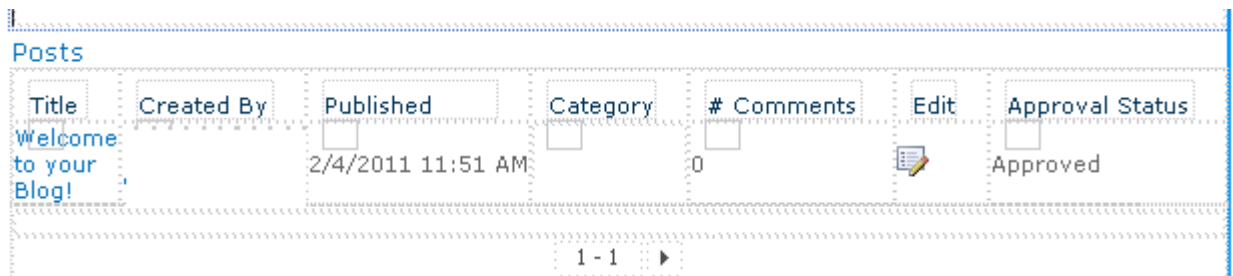


Figure 44

How to modify XSLT List View Web Part using conditional formatting

1. In SharePoint Designer 2010 and in the design

2. Right click on the XSLT view
3. Select Conditional Formatting. See figure 45.

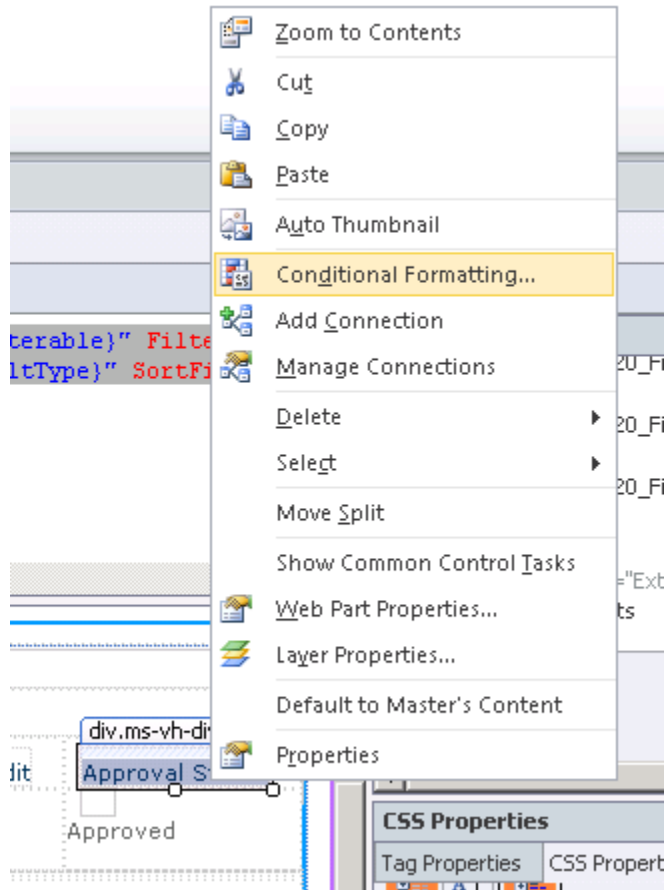


Figure 45

4. Click on Apply formatting as shown from right hand side menu. See Figure 46.

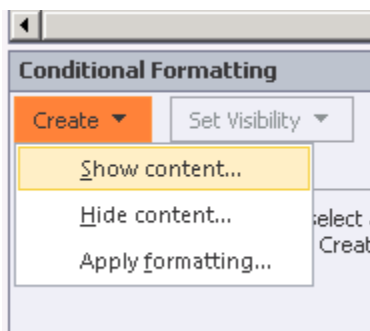


Figure 46

5. You will get a screen like in figure 47.

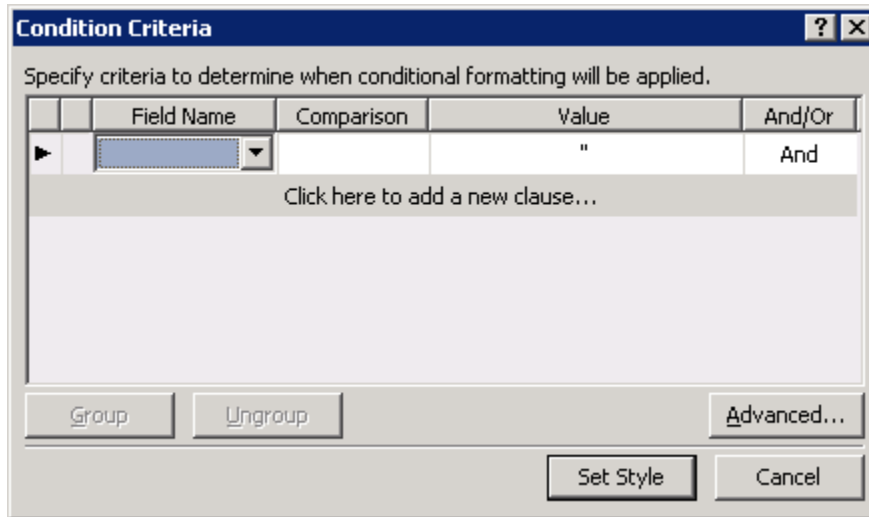


Figure 47

6. In my case I am selecting approval Status equals "Approved" then I want to show Approved Status fields in a Red color. Select your field as shown below and click Apply style. See Figure 48.

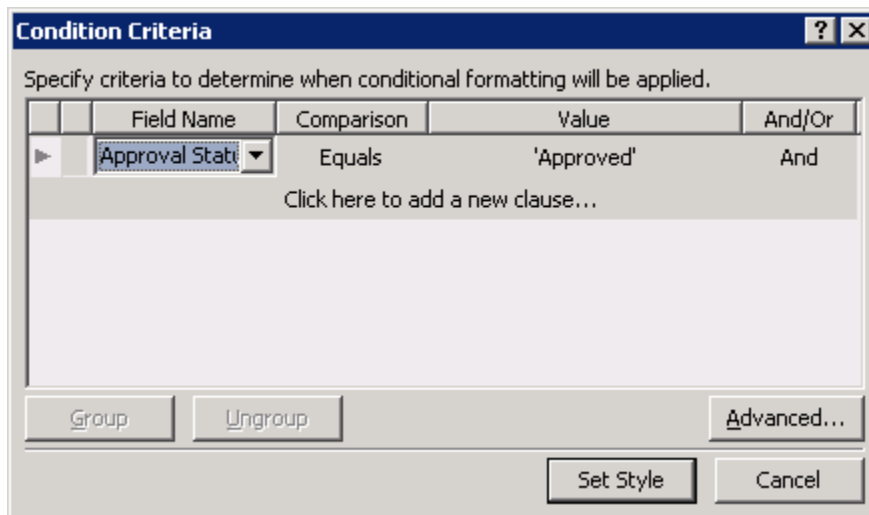


Figure 48

7. You will get the following screen, as mentioned earlier I have selected the background color to be red. See Figure 49.

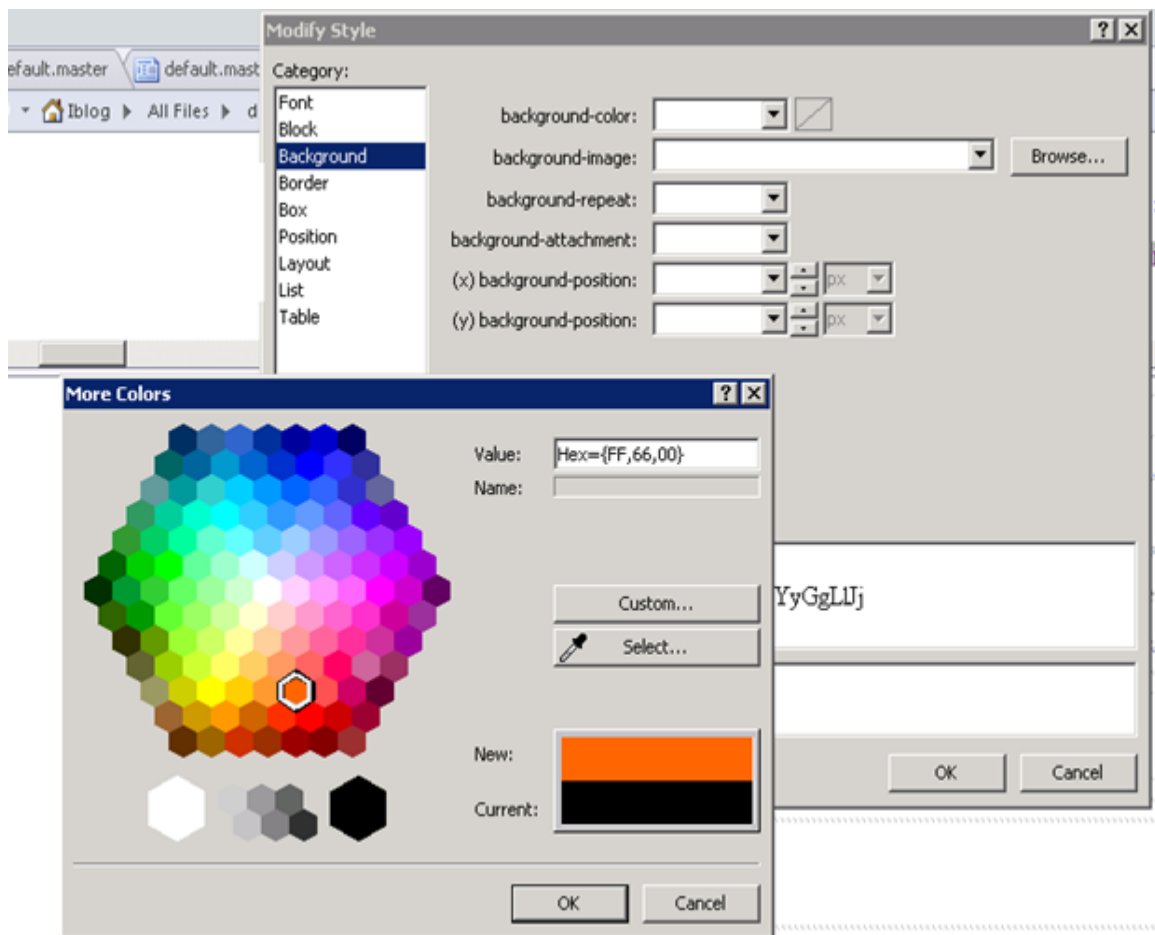


Figure 49

8. Now check your XSLT list view web part; you can see that particular cell is changed with a red color. See Figure 50.



Figure 50

You can use the same technique for any other formatting of data.

XSLT List Form Web Part

The XSLT List Form Web Part, or XLF, holds the list form used for external lists in SharePoint. When you create an external content type for an existing business data source and then create an external list for this data source, an XLF is automatically created. The XLF provides an important functionality for your site by allowing site users to use the form to read/write data to an external business data source. We can customize an XLF in the code view of SharePoint Designer only.

Data Form Web Part

Most of us may be used to this web part SharePoint Designer 2007. The Data Form Web Part, or DFWP can be used to display read-only views of data and user-input forms for data. The DFWP can be used for all of the available data sources in SharePoint Designer, including lists and libraries, database connections, Web Services, XML files, and more. DFWPs are used in various places in SharePoint Designer 2010. For example, when you create a new item, edit item, or display item form, a DFWP is added to the page. When you create a data view to a database connection, Web service, or XML file, a DFWP is added to the page.

List Form Web Part

This is the web part we mostly see in default list associated with SharePoint lists and libraries. These forms are highly formatted in a way that makes them easy to use by site users. If you don't want to customize them keep it like that. It is recommended to customize these forms with much care; you can simply use them as-is since they meet so many business purposes. If you want to change their look and feel or form behavior, please create a copy of the form in SharePoint Designer 2010 and make it the default form for the list or library. See figure 51.

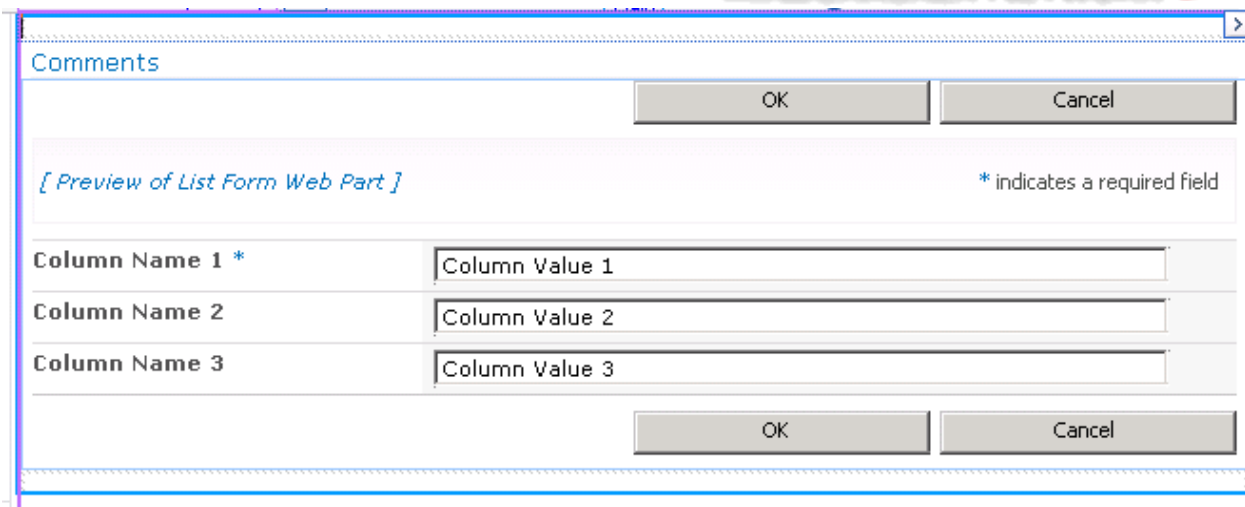


Figure 51

InfoPath Form Web Part

The InfoPath Form Web Part (IPWP) is a kind of browser form Web Part that is updated by an associated InfoPath form template. Anytime you convert list forms to InfoPath 2010, you will see a new InfoPath form template file with an extension of .xsn and new ASPX pages with this Web Part placed on them. The new ASPX pages replace the original form pages that you started with in SharePoint. InfoPath 2010 is a powerful form design tool that you can use to create powerful, rule-based, and heavily branded forms for SharePoint lists, workflows, and form libraries.

Adding Custom Actions to the Ribbon UI

Adding a Custom Action to a SharePoint List Item

In this Section we are showing you how to create a custom Action for a SharePoint list using SharePoint Designer 2010

1. Open the site in Designer 2010
2. Open your list and open the list called Announcement.
3. Once it is opened, click on Custom Actions from the List Settings

4. Select List Item Menu. See figure 52.

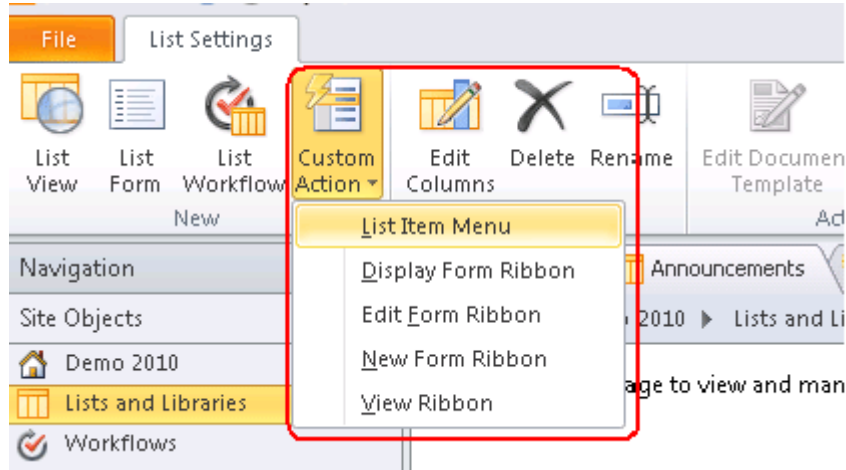


Figure 52

5. You will get a pop up .Give custom Action Name and description

7. We have the following three actions

- Navigate to a Form for this list
- We can Initiate a Workflow
- Or, we can Navigate to a URL

8. Proceed with the Navigate to a URL option

9. For this we need as a calendar list. Provide the list URL, as shown in figure 53.



Create Custom Action

Add a custom action to your list

Enter a name and description for your new custom action

Name:
Click Here

Description:

Select the type of action

Custom Actions can be used to run a workflow, show an existing form, or navigate to another Web page.

☐ Navigate to form:
/Lists/Announcements/DispForm.aspx?ID={ItemId}

☐ Initiate workflow:

☒ Navigate to URL:
lender.aspx

Figure 53

10. Click Ok and save the changes

11. Go to your list and click on any item you have (Menu); you can see the new menu and click here, as shown in figure 54.

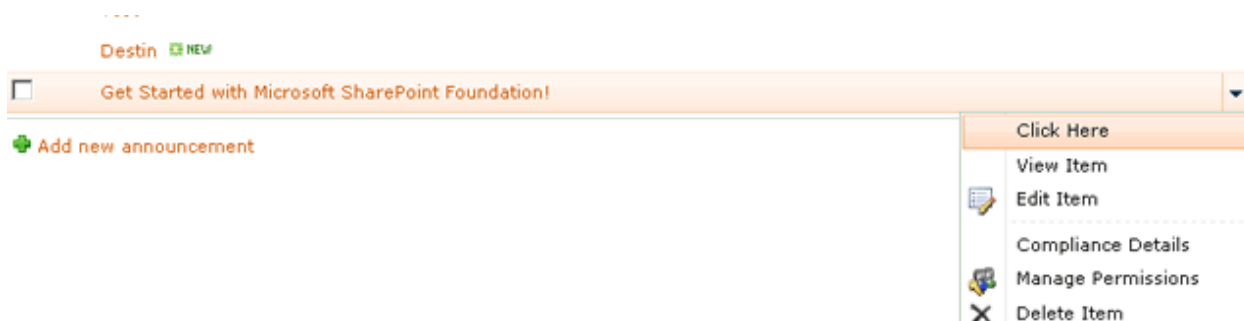


Figure 54

Adding Custom Action to a List Form Ribbon

In this section, we are showing you how to create a custom Action for a SharePoint list using SharePoint Designer 2010

1. Open the site in Designer 2010
2. Open your list, assuming there is a list called Announcement and open it in designer.
3. Once it is opened, from the List Settings, click on Custom Actions
4. Select Display Form Menu, as you can see in Figure 55.

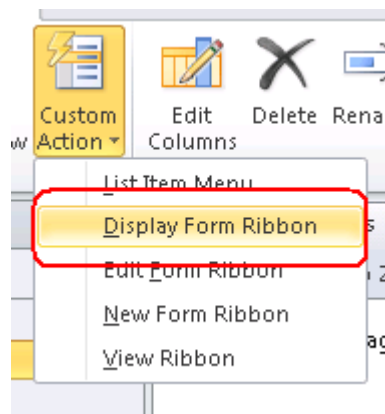


Figure 55

5. You will get a pop up as shown below
6. Give a Custom Action Name, description
7. We get the following three actions
 - Navigate to a Form for this list
 - Initiate a Workflow
 - Or, Navigate to a URL
8. Proceed with Initiate Workflow

9. We have a default Approval workflow named APPROVAL and there is an option to start this workflow in the Display Form Ribbon. See figure 56.

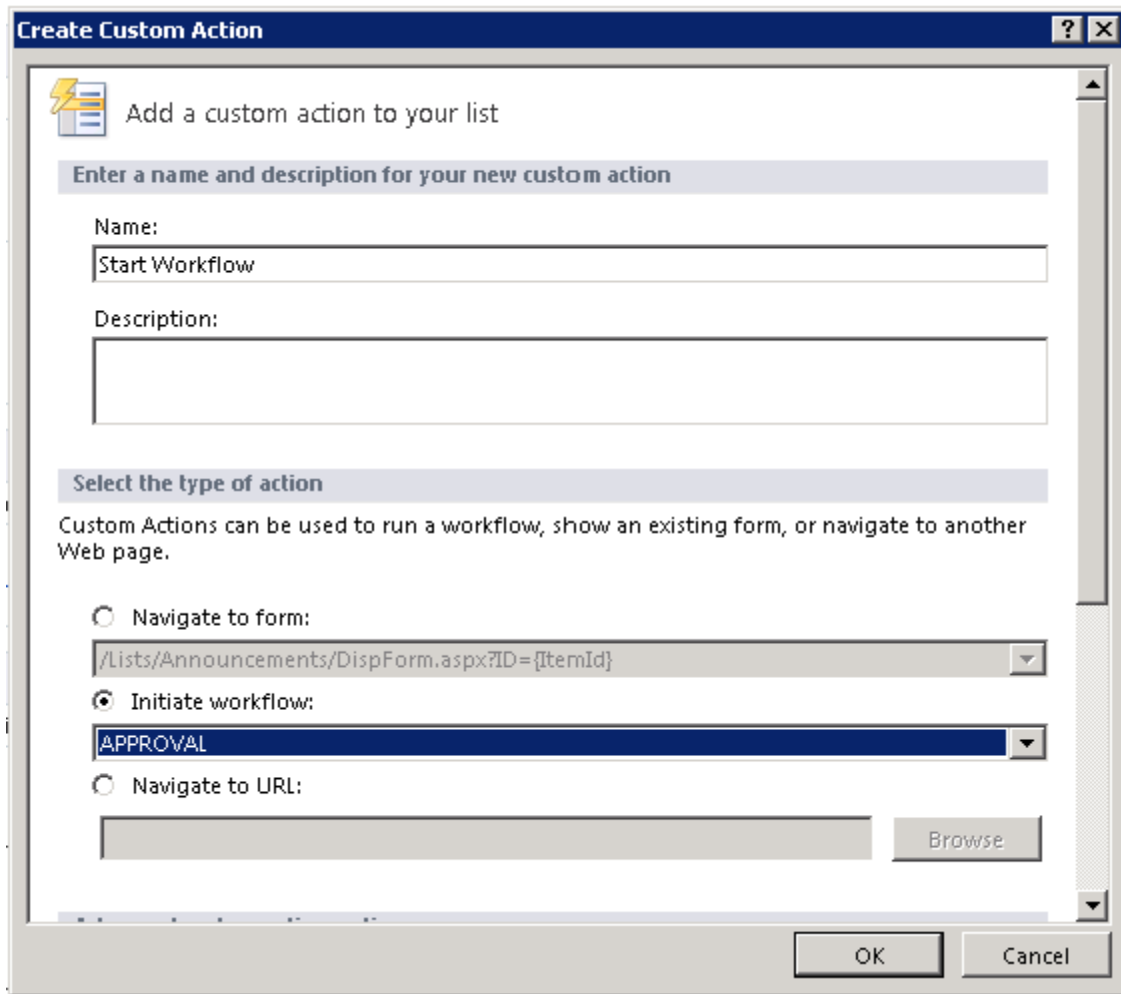


Figure 56

10. Click Ok and save the changes
11. Go to your list and click on any item to open in Display form.
12. You can see the new Ribbon item is added in figure 57.

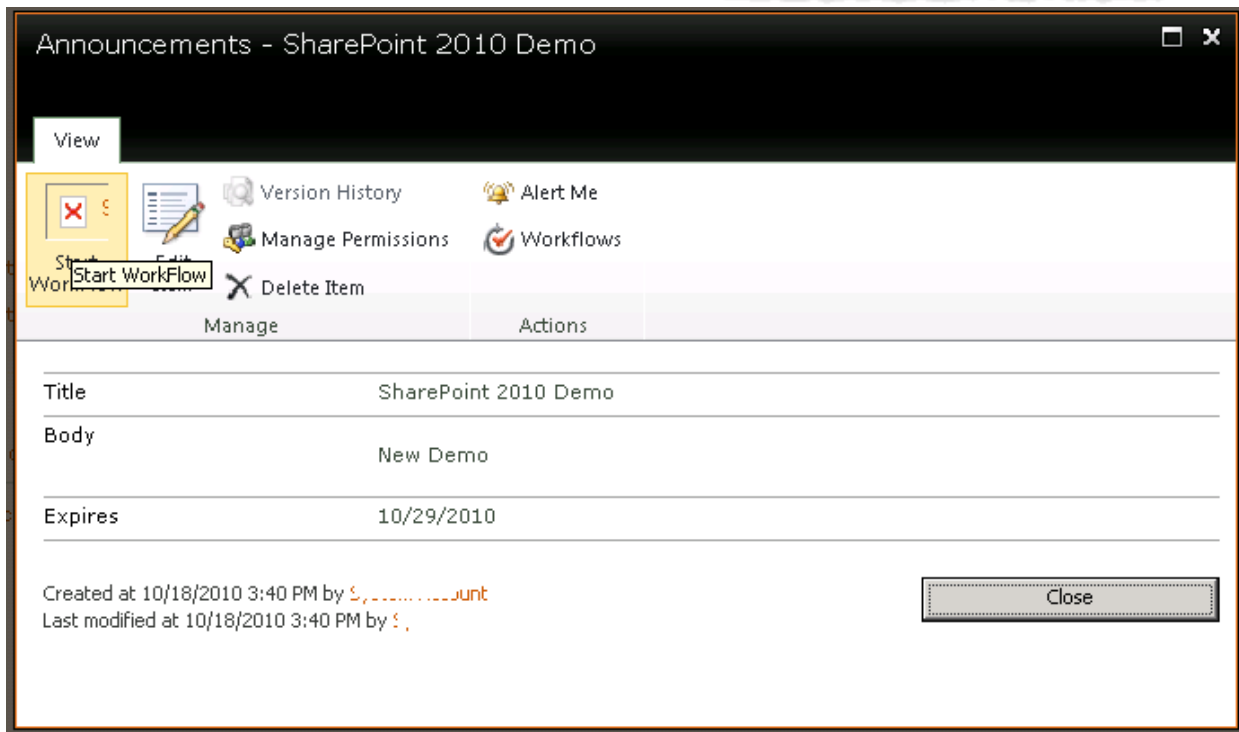


Figure 57

Working with Workflows

A workflow is sometimes described as a series of tasks that produce an outcome. In the context of Microsoft SharePoint Products and Technologies, a workflow is defined more narrowly as the automated movement of documents or items through a sequence of actions or tasks that are related to a business process. Workflows can be used to consistently manage common business processes within an organization by enabling the organization to attach business logic to documents or items in a SharePoint list or library. Business logic is basically a set of instructions that specifies and controls the actions that happen to a document or item.

Workflows can streamline the cost and time required to coordinate common business processes, such as project approval or document review, by managing and tracking the human tasks involved with these processes. Web designers can design no-code workflows for use in a specific list or library by using SharePoint Designer 2010. These workflows are created from a list of available workflow activities, and the Web designer

who creates the workflow can deploy the workflows directly to the list or library where they will be used.

There are two ways to view the workflow actions list in SharePoint Designer 2010. While editing a workflow:

- On the Workflow tab, in the Insert group, click Conditions to open the list of workflow actions.
- Alternately, you can click inside a workflow step, in the search box that appears type words that are in the name of the condition that you want, such as add, and then press Enter. Actions and conditions that contain the words you typed appear below the text box.

Actions are prearranged in categories based on their area of action in a workflow. Actions that affect an item's behavior are listed under List Actions; actions relevant to document sets under Document Set Actions; and custom workflow actions under Custom Actions. The following are the basic categories for actions

- Core Actions
- Document Set Actions-If your workflow is associated with a library or the document Content type.
- List Actions
- Relational Actions-If your SharePoint site is running SharePoint Server 2010, not available in Foundation.
- Task Actions-If your SharePoint site is running SharePoint Server 2010, not available in Foundation.
- Task Behavior Actions-appear only when you customize tasks in pre-built workflow actions, such as Start Approval Process and Start Feedback Process.
- Utility Actions
- Core Actions

The General actions can be seen in Listing 7 below.

No	Actions	Description
	Pause for Duration	This action is initially displayed in a workflow step as Pause for 0 days, 0 hours, 5 minutes. Use this action to pause the workflow for a specific duration in days, hours, or minutes. This will be useful if you want to send reminder mails after certain time duration.
	Send an Email	This action is initially displayed in a workflow step as Email these users. Use this action to send an email to a user or a group. It is commonly used to send a confirmation e-mail to workflow. You can easily specify a dynamic value in your subject such as a lookup or a string.
	Set Workflow Status	This action is initially displayed in a workflow step as Set workflow status to Canceled. Use this action to set the status of the workflow. The default options are Canceled, Approved, and Rejected.
1	Add a comment	This action is initially displayed in a workflow step as Comment: comment text. Use this action to leave informative comments in the workflow designer for reference purposes. This is especially helpful when there are other users co-authoring the workflow. This is more like we are giving information to our co-worker. Following is an example of what the action might look like in a workflow step, Comment: This variable provides the manager info for the workflows initiator. If we open the workflow in Microsoft Visio 2010 will also be able to view the comments.
2	Add Time to Date	This action is initially displayed in a workflow step as Add 0 minutes to date (Output to Variable:date). Use this action to add a specific time in minutes, hours, days, months, or years to a date, and stores the output value in as a variable. The date can be a current date, specific date, or a lookup. Following is an example of what the action might look like in a workflow step, Add 9 days to Current Item: Modified (Output to: Variable:

		A week from Modified)
3	Do Calculation	This action is initially displayed in a workflow step as Calculate value plus value (output to Variable: calc). Use this action to perform a calculation, such as add, subtract, multiply, or divide two values, and stores the output value in a variable. Please find the examples of what the action might look like in a workflow step, Calculate 50 divided by 10 (Output to Variable: Number of items)
4	Log to History List	Use this action to log a message about the workflow into its History list. (In some SharePoint out of the box workflow the history can be seen in some task list). The workflow history list can be helpful in troubleshooting issues with the workflow. This action is initially displayed in a workflow step as Log this message to the workflows history list. For Example Log A Month from Modified variable set to the workflow history list
	Set Time Portion of Date/Time Field	This action is initially displayed in a workflow step as Set time as 00:00 for date (Output to Variable: date). Use this action to create a timestamp, and stores the output value in a variable. You can set the time in hours and minutes and add a current date, specific date or a lookup.
	Set Workflow Variable	This action is initially displayed in a workflow steps as Set workflow variable to value. Use this action to set a workflow variable to a value.
	Stop Workflow	This action is initially displayed in a workflow step as Stop workflow and logs this message. Use this action to stop the current instance of the workflow and log a message to the Workflow History list.
	Check In Item	This action is initially displayed in a workflow step as Check in item in this list with comment: comment. Checks in an item that is checked out.
	Check Out Item	This action is initially displayed in a workflow step as Check out item in this list. Use this action to check out an item in

		list or library.
	Copy List Item	This action is initially displayed in a workflow step as Copy item in this list to this list. Use this action to copy a list item to another list. If there is a document in the list item, the workflow also copies the document to the destination list. But make sure you have same columns in the destination list also
	Create List Item	This action is initially displayed in a workflow step as Create item in the list (Output to Variable: create). This action can be used to create a new list item in the list that you specify. You can supply the fields and values in the new item.
	Delete Item	This action is initially displayed in a workflow steps as Delete item in this list. Use this action to delete an item from list or library.
	Discard Check Out Item	This action is initially displayed in a workflow step as Discard check out of item in this list. Use this action if an item is checked out, changes have been made to it, and you want to get rid of the changes and check the item back to the list or library.
	Set Content Approval Status	This action is initially displayed in a workflow step as Set content approval status to this status with comments. If you have content approval enabled in your list, use this action to set the content approval status field to a value such as Approved, Rejected, or Pending.
	Set Field in Current Item	This action is initially displayed in a workflow step as Set field to value. Use the action to set a field in the current item to a value specifically in list or library.
	Update List Item	This action is initially displayed in a workflow step as Update item in list Use this action to update a list item. You can specify the fields and the new values in those fields. This will be helpful after some approval or something you want to make one of the field in the list item to some specific value.
	Wait for Field Change in Current Item	This action is initially displayed in a workflow steps as Wait for field to equal value. This action pauses the workflow until the field in the current item has changed to a new value. This will be helpful if you want to do some action if item is approved or rejected

Assign a Form to a Group	This action is initially displayed in a workflow step as Assign a custom form to these users. It enables you to create a custom task form with customized fields.
Assign a To-do Item	This action is initially displayed in a workflow step as Assign to-do to these users. Use this action to assign a task to each of the participants, prompting them to perform their tasks and then, when they are done, to click the Complete Task button on their task form.
Collect Data from a User	This action is initially displayed in a workflow step as Collect data from this user (Output to Variable: collect). Use this action to assign a task to the participant, prompting them to provide the needed information in a custom task form. This action will be useful if we are developing an approval workflow
Extract Substring from End of String	This action is initially displayed in a workflow step as Copy 0 characters from then end of string (Output to Variable: substring). Use this action to copy n number of characters from the end of a string and store the output value in a variable. You have to define the number of characters that the workflow has to copy from the string.
Extract Substring from Index of String	This action is initially displayed in a workflow steps as Copy from string, starting at 0 (Output to substring). Use this action to copy all characters starting from character n in a string and store the output in a variable
Extract Substring from Start of String	This action is initially displayed in workflow step as Copy 0 characters from the start of string (Output to Variable: substring). Use this action to copy n number of characters from the start of a string and stores the output in a variable. You must define the number of characters that the workflow has to copy from the string
Extract Substring of String from Index with Length	This action is initially displayed in a workflow step as Copy from string, starting at 0 for 0 characters (Output to Variable: substring). Use this action to copy n number of characters starting from a particular character in a string and stores the output in a variable. You must define the number of characters that the workflow has to copy from the string.

Find Interval Between Dates	This action is initially displayed in a workflow step as Find minutes between date and date (Output to Variable: time). Use this action to calculate the time difference, in terms of minutes, hours or days, between two dates and stores the output in a variable.
Start Document Set Approval Process	This action is initially displayed in workflow steps as Start Approval process for the contents of this Document Set with the users specified by this column. It starts an approval action on a document set to specific users.
Capture a version of the Document Set	This action is initially displayed in workflow steps as Capture a version of this Document Set that includes the type versions of the contents with comment.
Send Document Set to Repository	This action is initially displayed in workflow steps as Submit Document set using this action to this destination content organizer with this explanation (Output to Variable: submit file result). It allows you to move or copy the document set to a document repository.
Set Content Approval Status of the Document Set	Use this action to set content approval of a document set to Approved, Rejected, or Pending
Start Approval Process	This action is initially displayed in workflow steps as Start Approval process on Current Item with these users. It routes a document for approval. Approvers can approve or reject the document, reassign the approval task, or request changes
Start Custom Task Process	This action is initially displayed in workflow steps as Start Task process on Current Item with these users. The Start Custom Task Process action is an approval process template that you can use if the above approval actions do not meet your needs.
Declare Record	This action displays in a workflow steps as Declare this item as a record. Use this action to identify a document as a record and apply the record restrictions settings you may have defined in the Record Declaration Settings page in SharePoint.

	Undeclare Record	This action displays in a workflow steps as Undeclare the item as a record. Use this action to undo any record declaration settings that are applied to a record.
	Lookup Manager of a User	This action is initially displayed in workflow steps as Find Manager of this user (output to Variable: manager). Use this action to look up a user's manager. The output value is then stored in a variable. This is a very useful action in Designer 2010
	Add List Item Permissions	This action is initially displayed in workflow steps as Add these permissions to item in list. It grants the permissions of the specific permission levels for an item to specific users.
	Parent Permissions	This action is initially displayed in workflow steps as Inherit parent permissions from item in this list. If your item has unique permissions, you can use this action to make the item inherit the parent permissions from the list.
	Remove List Item Permissions	This action is initially displayed in workflow steps as Remove these permissions from item in list. It removes permissions from an item for specific user or users.
	Replace List Item Permissions	This action is initially displayed in workflow steps as Replace these permissions of item in list. It replaces an item's current permissions with the new permissions that you specify in the action

Listing 7

Some list actions are available only when you click inside an impersonation step in a workflow.

SharePoint Workflow Demo

In this section, we are showing a demonstration of SharePoint Designer 2010 custom workflows

1. Open your SharePoint 2010 site in SharePoint 2010 Designer.
2. Go to File -> Add Item. See Figure 58.

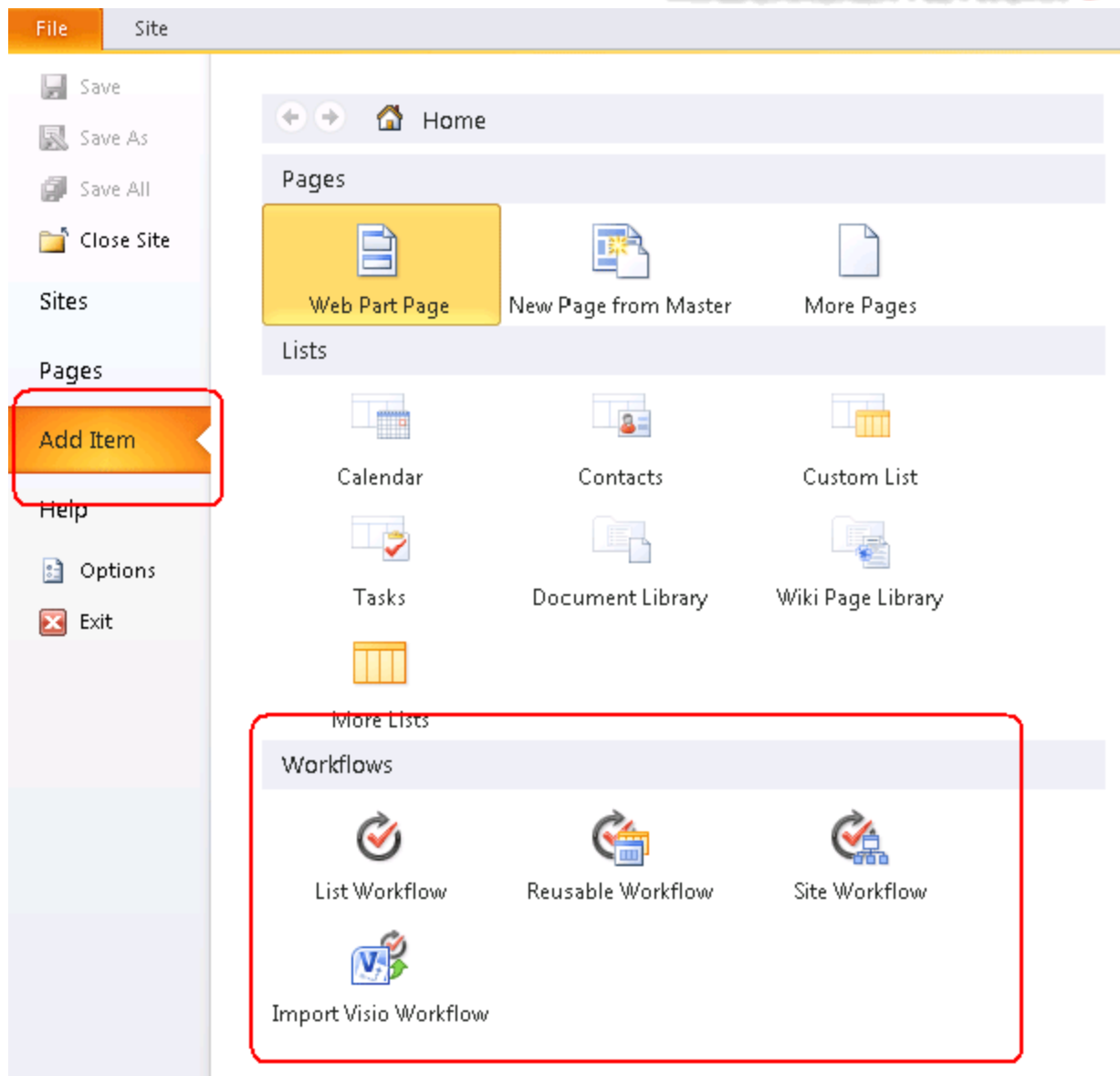


Figure 58

3. Create a new workflow. We can choose from one of the following workflow types:

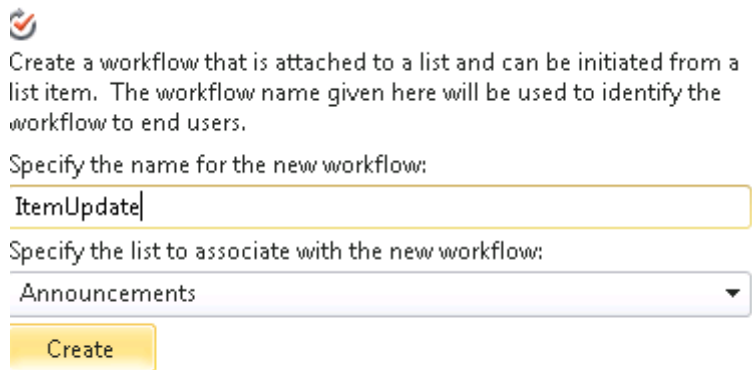
- **List Workflow** - This type of workflow can be attached only to SharePoint lists or libraries. This kind of workflow can be useful if your requirements are specific to a list or document library
- **Reusable Workflow** - It is more complex than a List workflow. It can be attached to a content type. Later, this content type can be used with a list. One more

advantage of this type of workflow is that it can be imported to Visual Studio and we can write code to enhance this type of workflow

- **Site Workflow** - This type of workflow operates on site level.
- **Import Visio Workflow** - Flexibility that it can be imported/exported to Visio.

4. Select the List Workflow

Give a name and select the list for the workflow to be attached. See figure 59.



Create a workflow that is attached to a list and can be initiated from a list item. The workflow name given here will be used to identify the workflow to end users.

Specify the name for the new workflow:

ItemUpdate

Specify the list to associate with the new workflow:

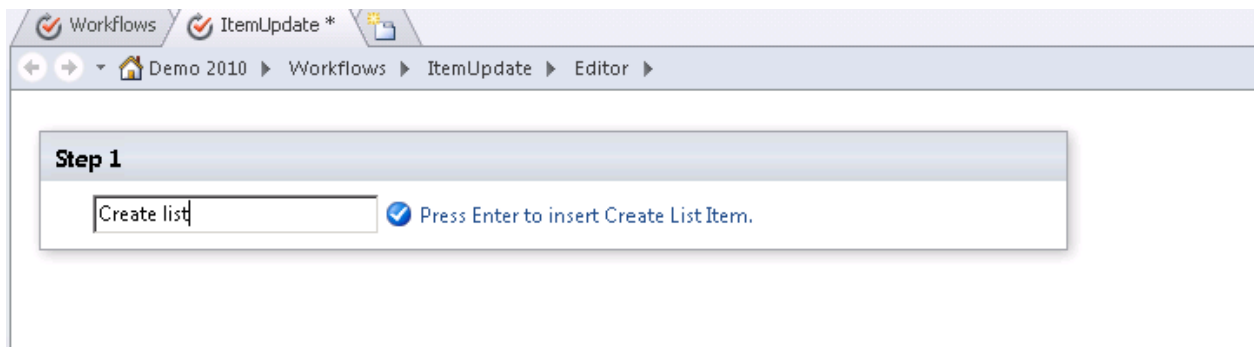
Announcements

Create

Figure 59

5. From the following screen, you have to choose the Action.

6. We have chosen Create List Item. See figure 60.



Workflows ItemUpdate * 

← → Demo 2010 > Workflows > ItemUpdate > Editor >

Step 1

Create list  Press Enter to insert Create List Item.

Figure 60

8. Assume that in this demo we have a calendar list and an announcement list .We are trying to show you whenever an item is added to Announcement, an event will be added to the Calendar.

9. Select my Calendar list as shown below. See figure 61.

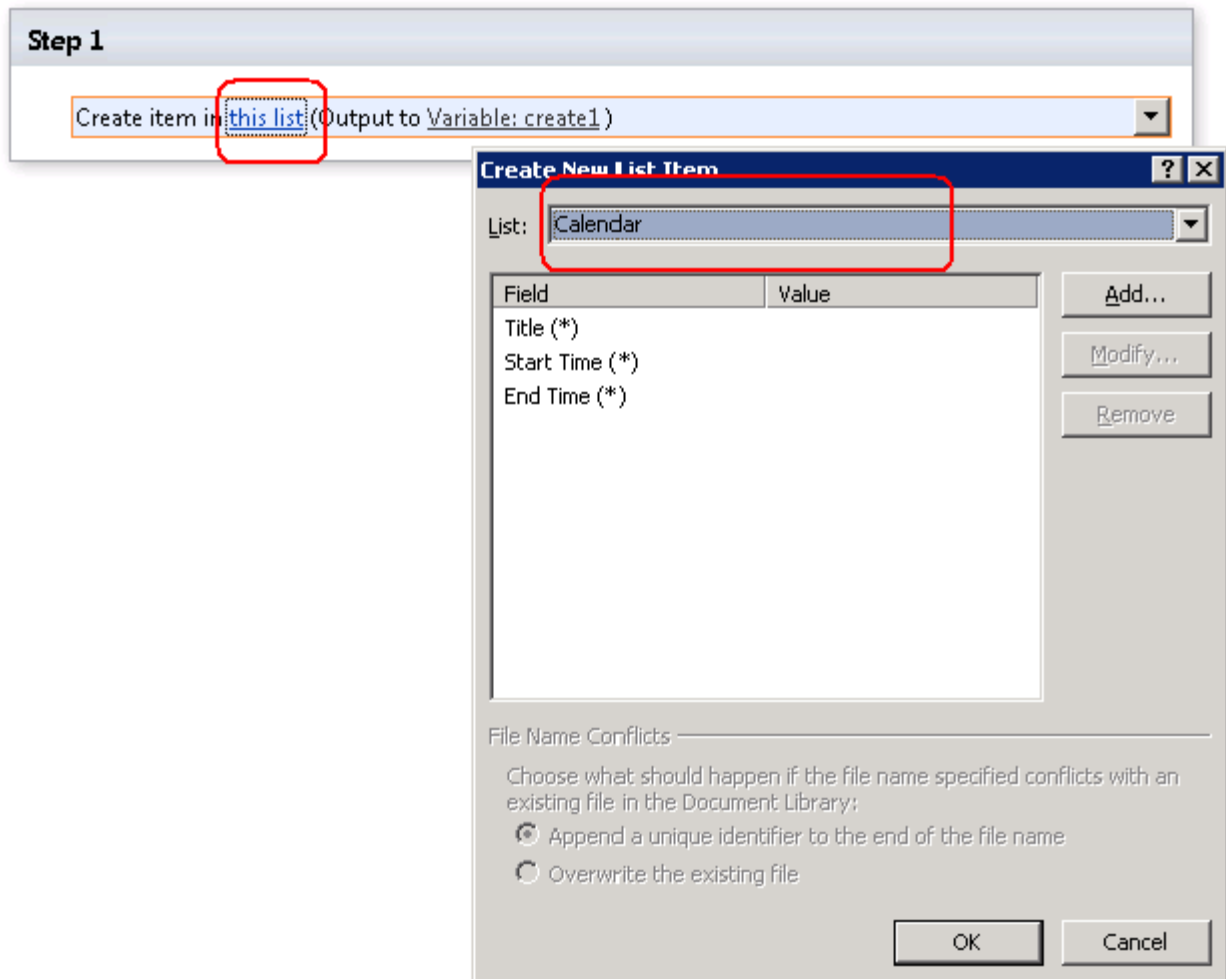


Figure 61

10. Set the Calendar title to the Announcement Title. See Figure 62.

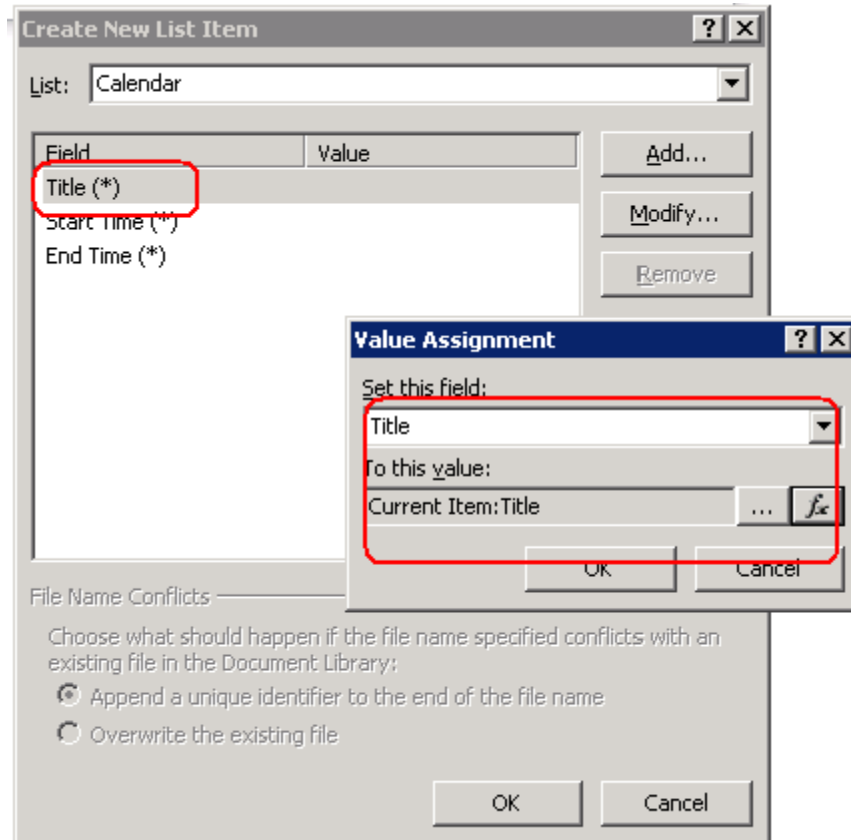


Figure 62

11. Set the Calendar Start Date to the Announcement Created Date. See figure 63.

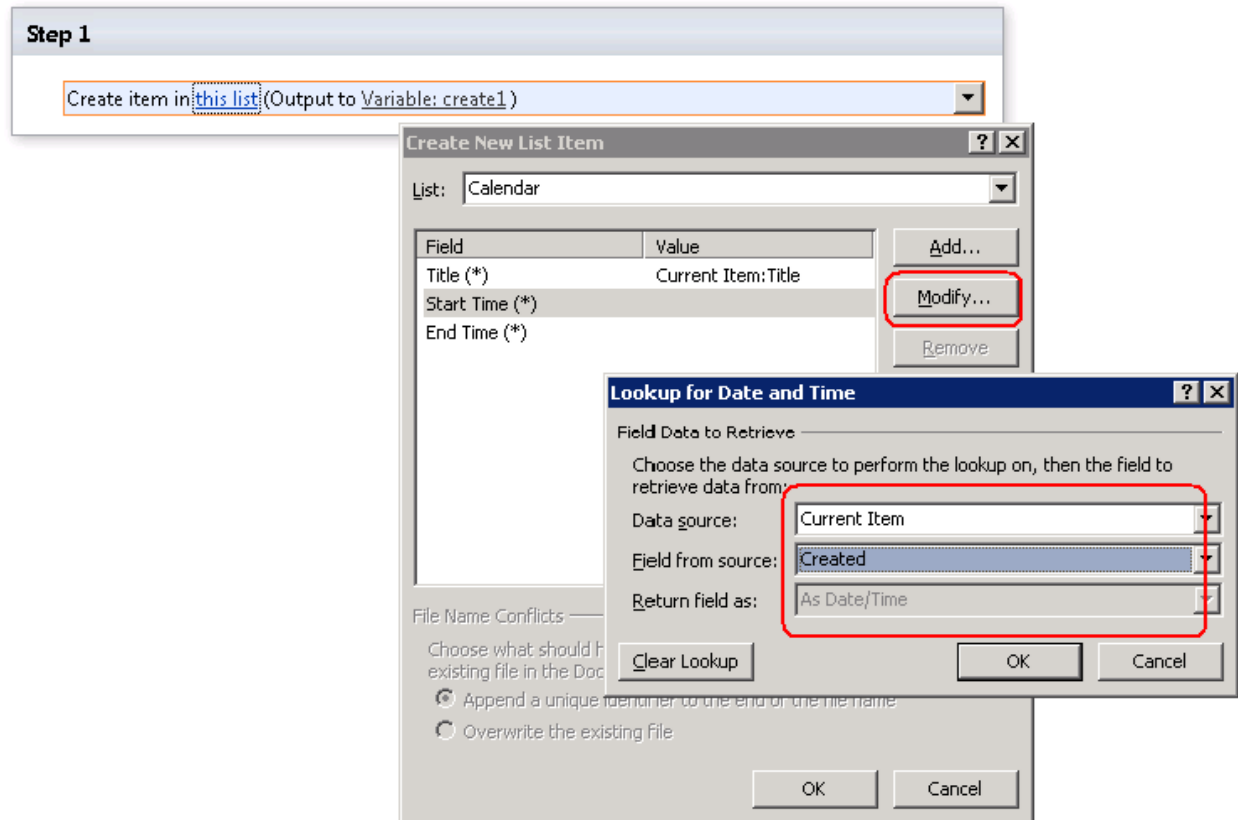


Figure 63

12. Set the Calendar End Date to the Announcement Expires Date. See figure 64.

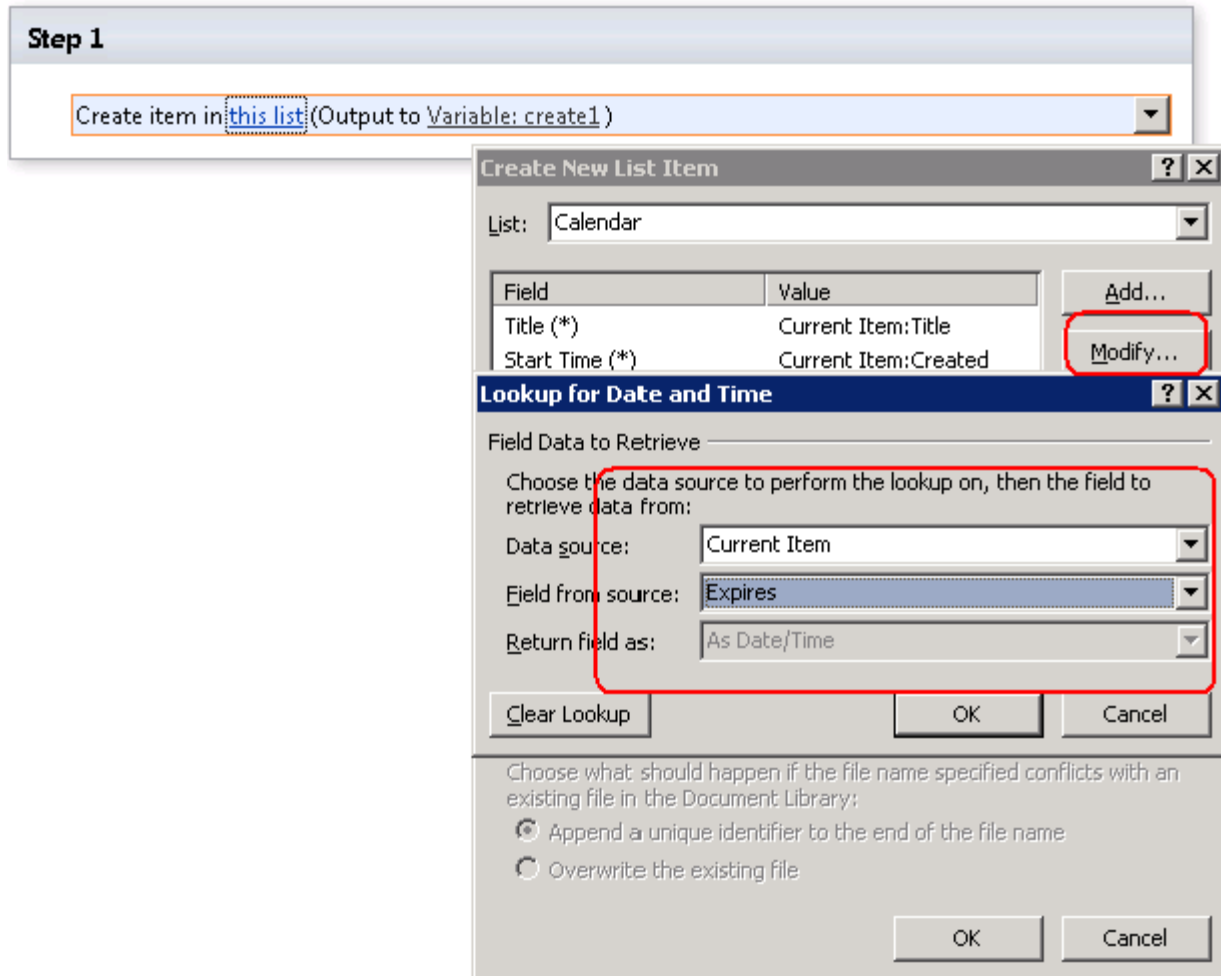


Figure 64

13. When done, you will get a screen like below. See Figure 65

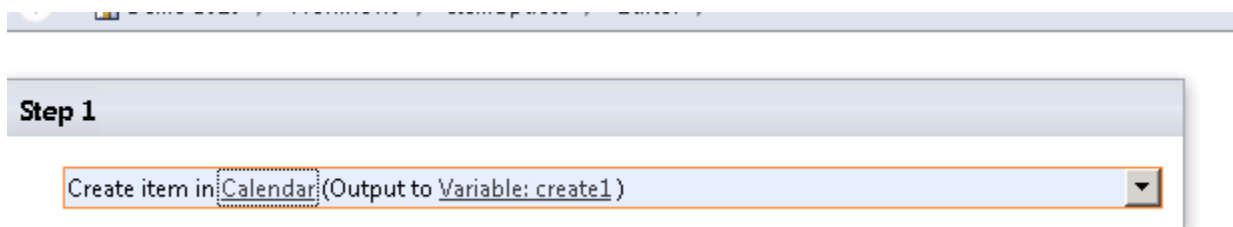


Figure 65

14. Save your Changes. See figure 66.



Figure 66

15. If you want to change the workflow settings, do as shown below. See figure 67.

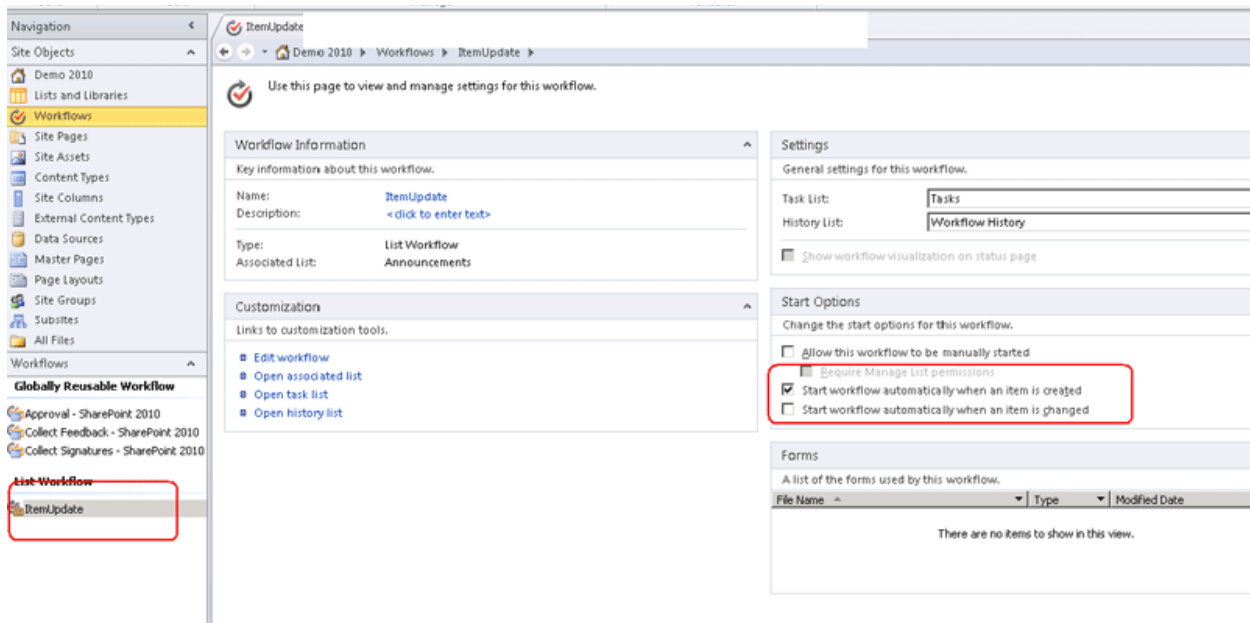


Figure 67

Exporting a SharePoint Designer 2010 Workflow to Visual Studio

In this Section we are showing you how to export a SharePoint Designer 2010 workflow to Visual Studio.

1. Open your site in SharePoint Designer 2010.
2. From the Ribbon click Reusable Workflow. See figure 68.

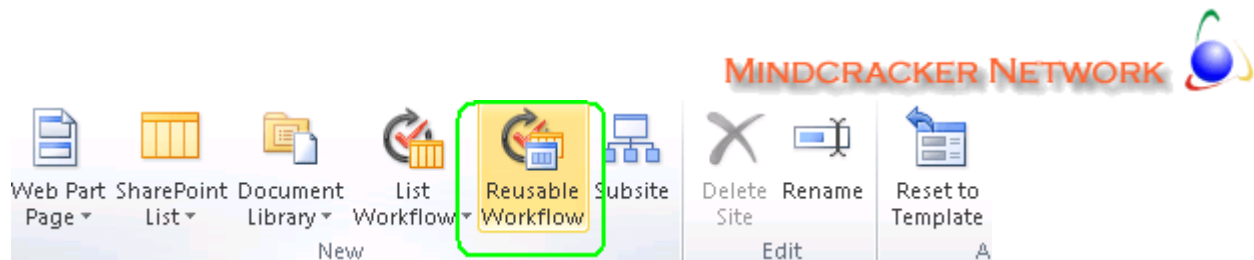


Figure 68

3. For the next screen, give a proper name for your workflow then select the Content Type you want to associate with this workflow. See figure 69.

 The screenshot shows a 'Create Reusable Workflow' dialog box with a title bar containing a question mark and a close button. The main area has a header 'Add a new reusable workflow to your site' with a circular arrow icon. Below this is a section titled 'Enter a name and description for your new workflow'. It contains a 'Name:' label followed by a text box with 'Designer Demo' entered, and a 'Description:' label followed by an empty text box. Below this is another section titled 'Pick a base content type to limit this workflow to'. It contains a 'Content Type:' label followed by a dropdown menu currently showing 'Discussion'. At the bottom right are 'OK' and 'Cancel' buttons.

Figure 69

4. Select a Simple approval workflow and click Ok as shown below. See Figure 70.

 The screenshot shows a configuration screen titled 'Step 1'. It features a single text box with the text 'Start Approval process on Current Item with these users'. The text 'these users' is highlighted with a dashed blue border, indicating it is a clickable link or a selection point.

Figure 70

5. You have to save the workflow by clicking Save. See Figure 71.

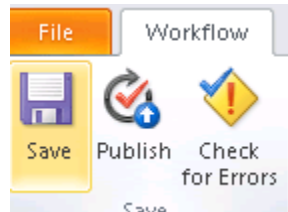


Figure 71

6. Now you have to publish the workflow to be able to reuse it in all the sites from the site collection in SharePoint.

7. Click on Publish Globally. See Figure 72.

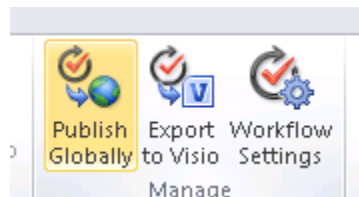


Figure 72

8. You will get a conformation like below; click OK. See Figure 73.

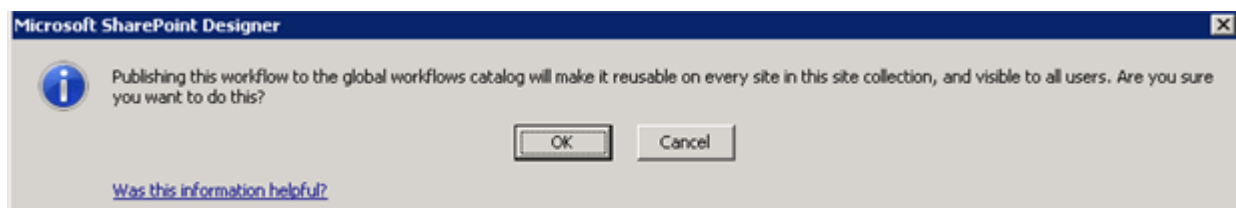


Figure 73

9. Click on Workflow settings. See Figure 74.



Figure 74

10. Then click on Save as template. See Figure 75.

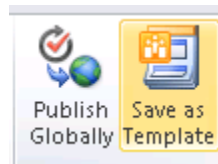


Figure 75

11. Your workflow will be saved as a template in your Site Assets. See Figure 76.

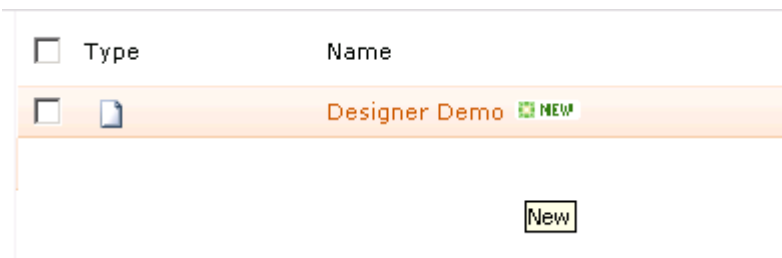


Figure 76

12. Save your workflow template locally. See Figure 77.

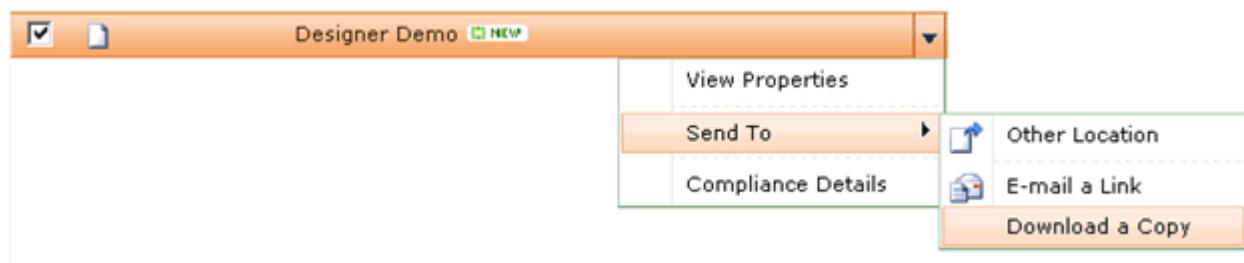


Figure 77

13. Open Visual Studio 2010 and Select Import SharePoint Solution Package. See Figure 78.

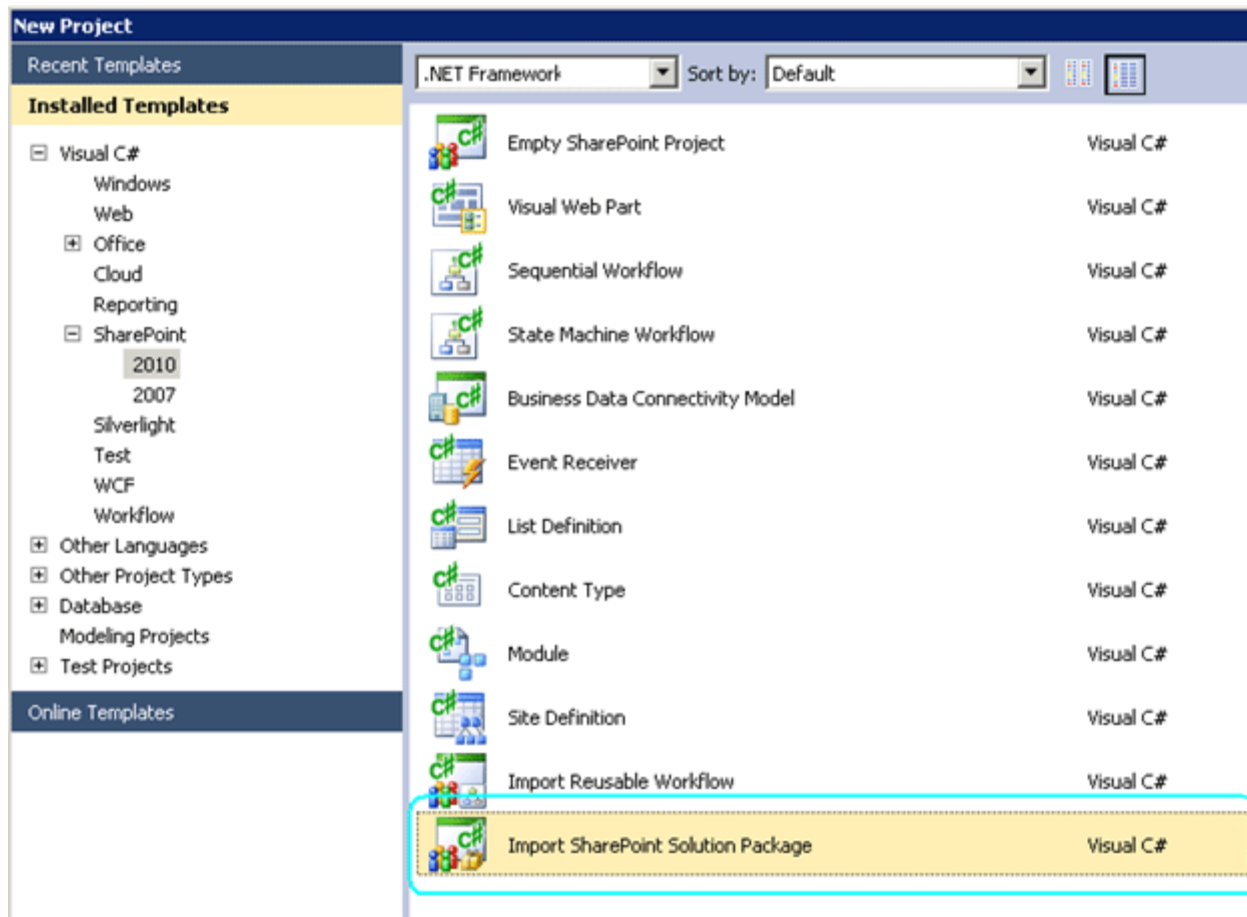


Figure 78

14. Click OK.

15. Give the web application URL then click Next. See Figure 79.

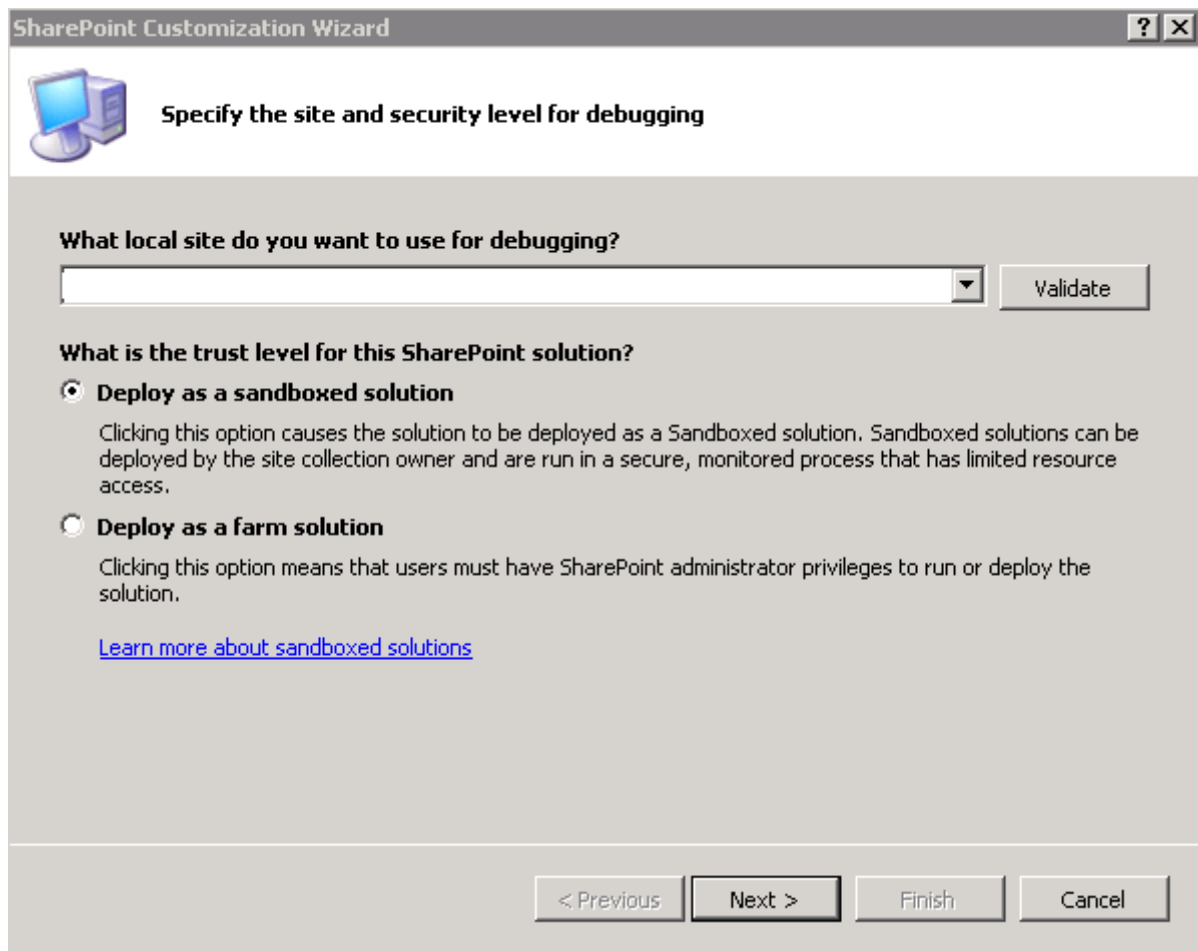


Figure 79

16. From the next screen, browse to our Workflow template and click Next. See Figure 80.

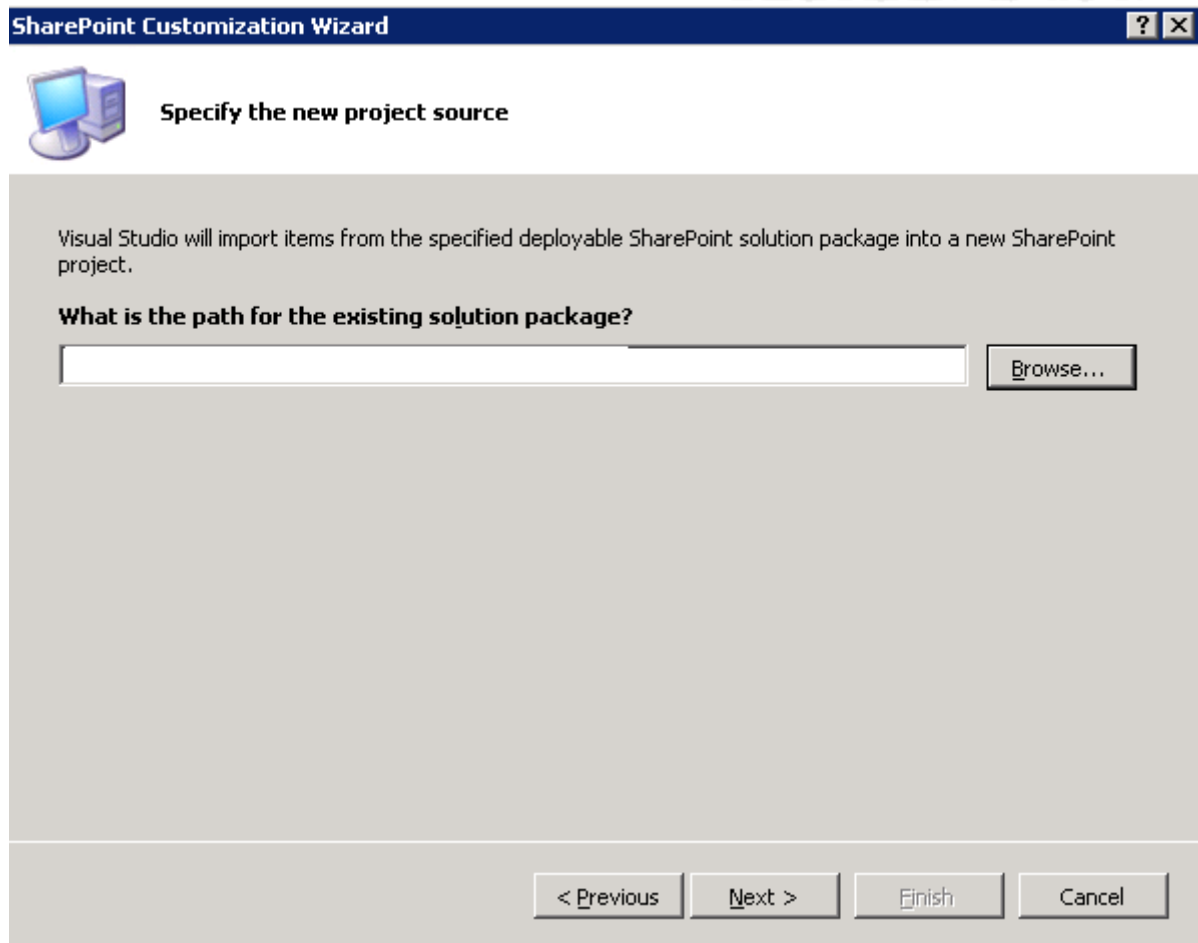


Figure 80

17. You will get the following screen; then click Finish. See Figure 81.

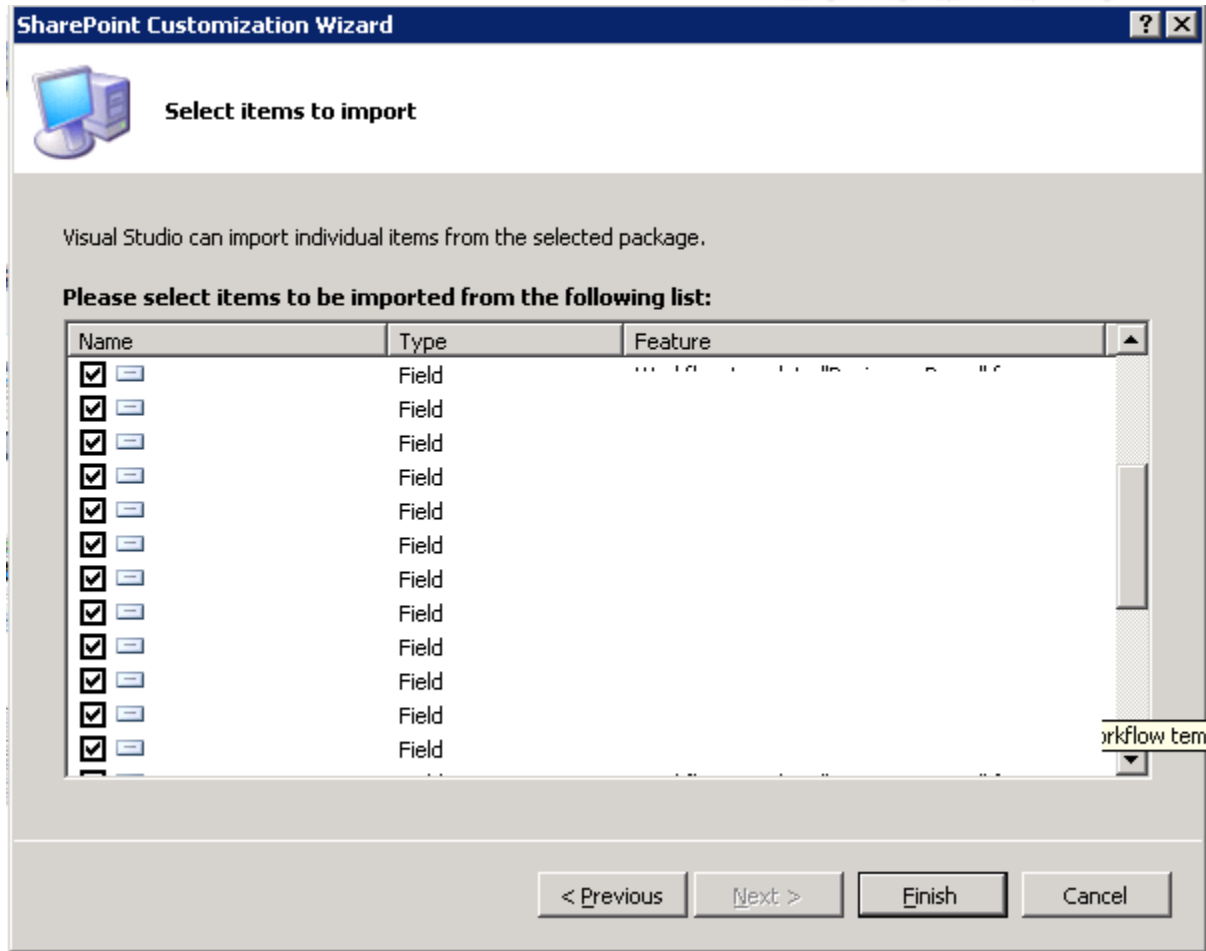


Figure 81

18. Your workflow has been imported to Visual Studio. See Figure 82.

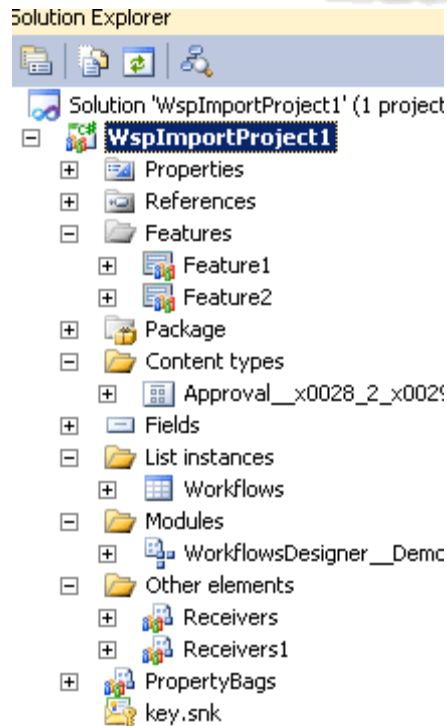


Figure 82

19. After you have modified the Workflow. deploy the solution.

Working with External Content types

Implementation-wise, external content types are the evolution of BDC entities from SharePoint Server 2007. They are metadata descriptions of the connectivity settings, data structures and stereotyped operations – like create, read, update and delete - that allow access to business entities as exposed by external systems (such as databases, web services, line-of-business applications, among others). In addition to the new stereotyped operations supporting read/write access to external data, external content types in SharePoint 2010 can also describe SharePoint and Office integration behaviors like mappings to native Office item types including Contact, Task, Appointment and Post as well as off line data mapping to Outlook 2010 and SharePoint Workspace 2010. These are very important new aspects as they are the basis for providing familiar experiences for information workers to interact with external data inside SharePoint and Office.

An external content type represents a reusable description of an object that can be used in a composite application, such as "Customer," "Order," or "Contact" (recall that these

were formerly called BDC entities). The external content type also includes the object's data fields; the methods to create, read, update, or delete that object; actions that users can take on the object; and information that supports connecting to the external data source where the object is stored. Notice that both read and write capability is included, which is an improvement over the BDC capability in SharePoint 2007. Also, batch and bulk operation support is available. External content types represent data that is stored in an external data source, such as Microsoft SQL Server, SharePoint Server, and Web services. External content types are created using SharePoint Designer 2010, Visual Studio® 2010, and by importing an application model that contains one or more external content types into a BCS service application.

There are three options

- Creating an External Content type.
- Creating an External Data column for list.
- Creating an External List

Before starting, you should have a SQL Server Table. We have a database with the name BCS and a table named BCS_customer with two columns; NameID and Name

Creating an External Content Type

1. Open your site in SharePoint Designer 2010.
2. Click on External Content Type.

3. Click on New External Content Type. See Figure 83.

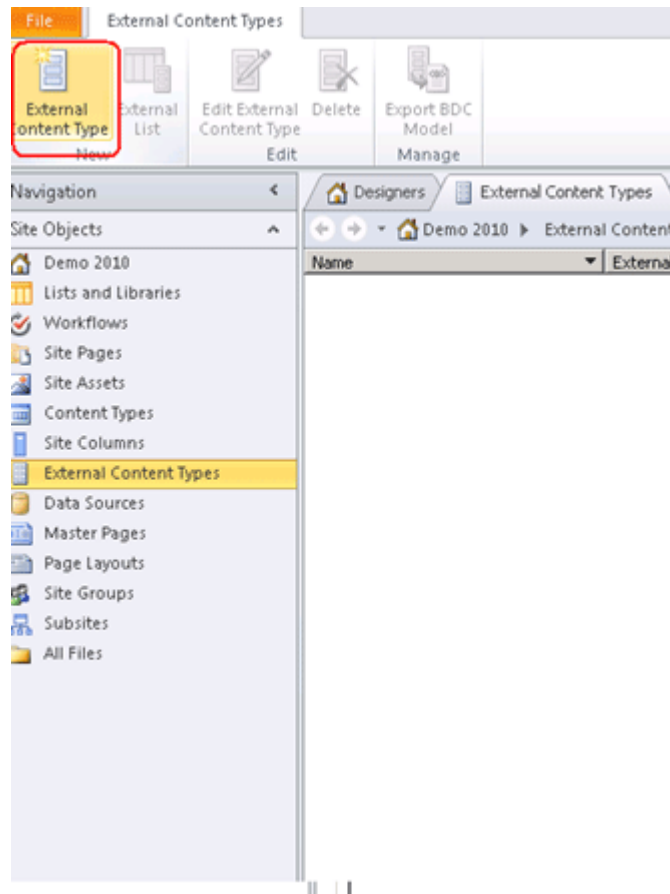


Figure 83

You will get a screen like the one shown below. Click on Add Connection. See Figure 84.

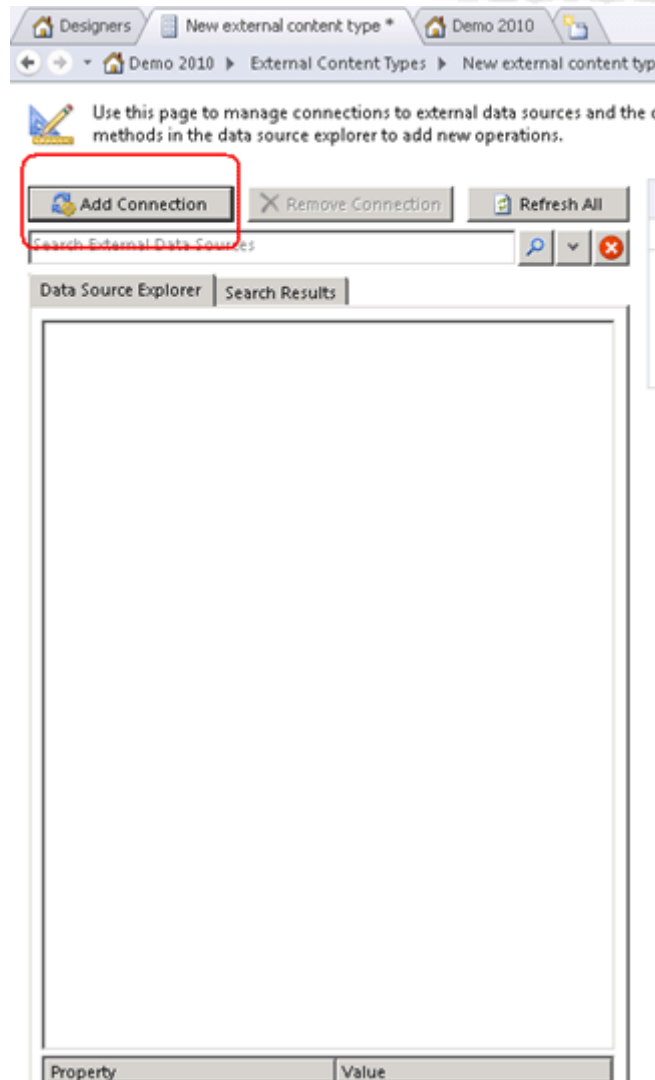


Figure 84

4. There are many options available. Select the SQL Server option. See Figure 85.

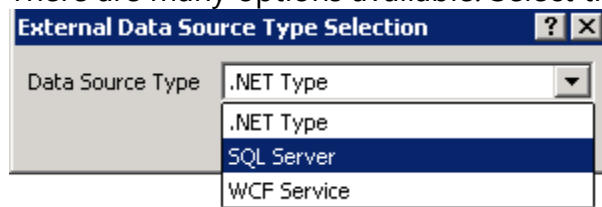


Figure 85

5. Give the data connection with the database server name and database name. You can also configure out which identity is used to connect to the database. Please make sure that you granted the proper permissions on the SQL Server itself for the appropriate user.

6. After that it will add the tables in the connection as shown below. Click on the table and select Create All Operations. See Figure 86.

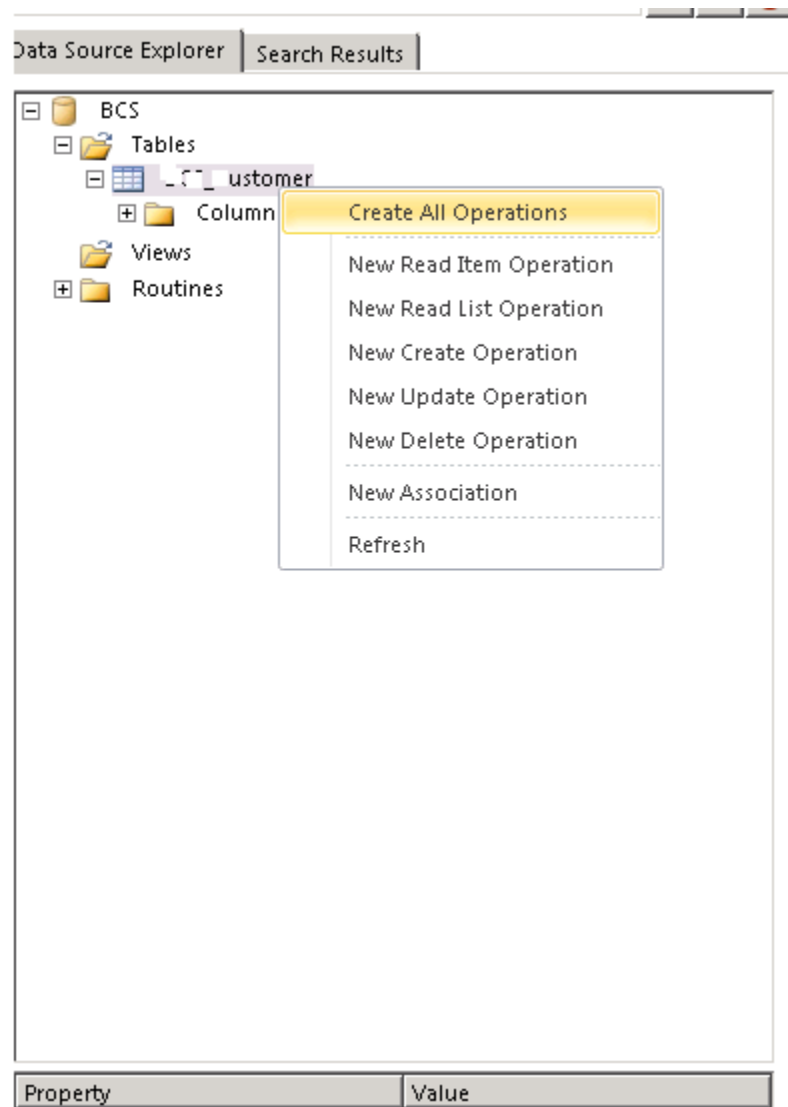


Figure 86

On the next screen, you have to select the columns you want to display. See Figure 87.

All operations

Operation Properties

Parameters

Filter Parameters

Parameters Configuration

Define input parameters for this operation.

☒ **Data Source Elements**

Name	Type
NameID	System.String
Name	System.String

Properties

Data Source Element: Name

.NET Type: System.String

Map to Identifier: ☐

Identifier:

Field:

Display Name:

Foreign Identifier: [\(Click to Add\)](#)

Required: ☐

Read-Only: ☐

Office Property:

Show In Picker: ☐

Timestamp Field: ☐

Errors and Warnings

 No fields have been selected to be shown in the external item picker control. By default, all fields will be displayed in the external item picker dialog, which may not be the user experience you want. Select a small subset of elements that best describes an item and then select the 'Show in Picker' check box for each one.

 The identifier NameID configured on the data source element NameID is not marked as read-only. Enabling identifiers to be updated requires a 'PreUpdaterField' to be configured on the corresponding Update operation. This is not supported by SharePoint Designer. If you need this functionality edit the external content type outside of SharePoint Designer. For more information see the documentation for 'PreUpdaterField' available at msdn.microsoft.com

To configure Filter Parameters, click Next.

< Back Next > Finish Cancel

Figure 87

7. If successfully added, you will get a screen that looks like Figure 88. Click on Create Profile Page.

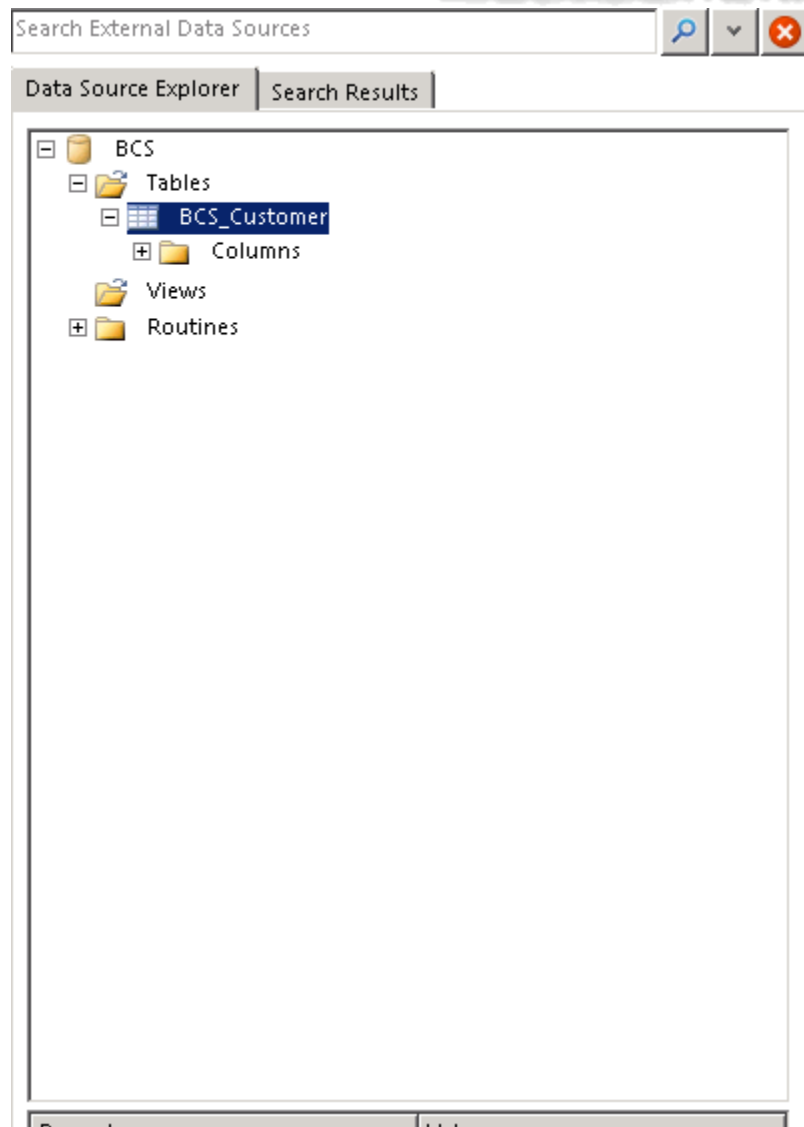


Figure 88

8. You are done with External Content Type creation. See Figure 89.

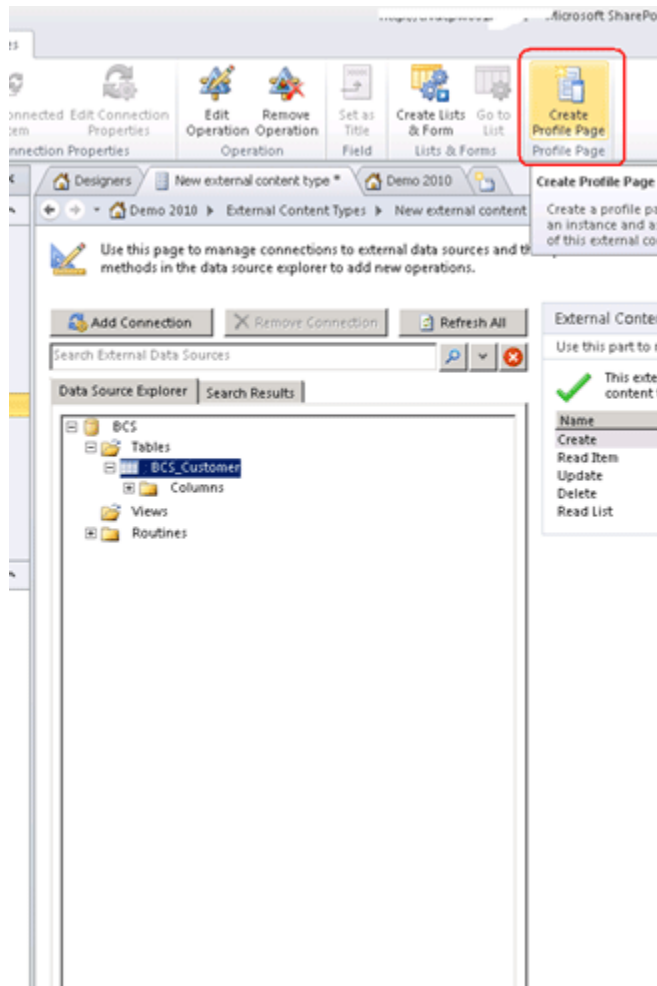


Figure 89

Creating External Data Column for a List

9. Go to the SharePoint list and create a column of type External Data.

Select the External Content Type we just created. See Figure 90.

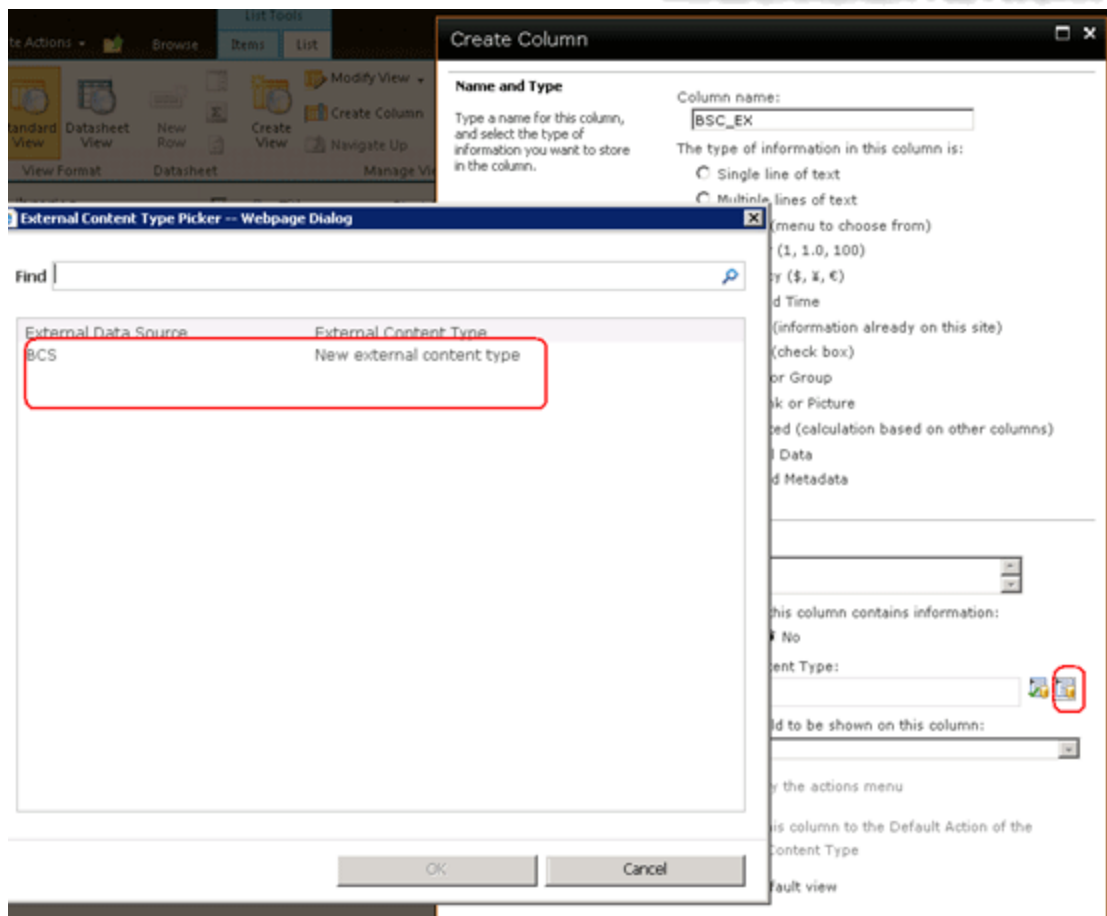


Figure 90

Click OK Select the options as shown in Figure 91.



Create Column

☐ Single line of text
☐ Multiple lines of text
☐ Choice (menu to choose from)
☐ Number (1, 1.0, 100)
☐ Currency (\$, ¥, €)
☐ Date and Time
☐ Lookup (information already on this site)
☐ Yes/No (check box)
☐ Person or Group
☐ Hyperlink or Picture
☐ Calculated (calculation based on other columns)
☒ External Data
☐ Managed Metadata

Additional Column Settings

Specify detailed options for the type of information you selected.

Description:

Require that this column contains information:
☐ Yes ☒ No

External Content Type:

Select the Field to be shown on this column:

☒ Display the actions menu

☒ Link this column to the Default Action of the External Content Type

Add a column to show each of these additional fields:
☒ Select all fields

☒ Name
☒ NameID

☒ Add to default view

Figure 91

10. In the following figure, you can see the data populated from the BCS_Customer Table.

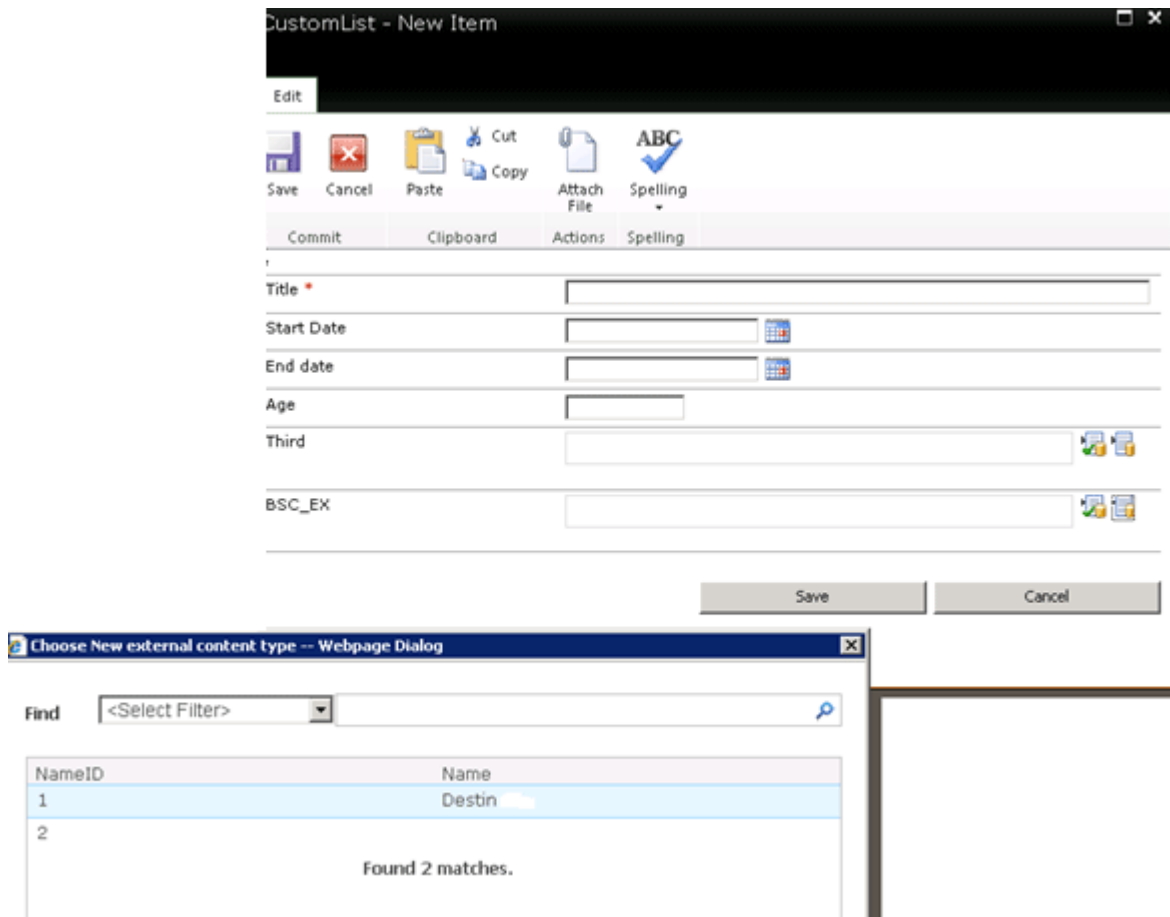


Figure 92

Creating an External List

11. You have to go back to your Designer and click on the Create list and Form

Give a proper name for your list. Click Ok. See Figure 93.

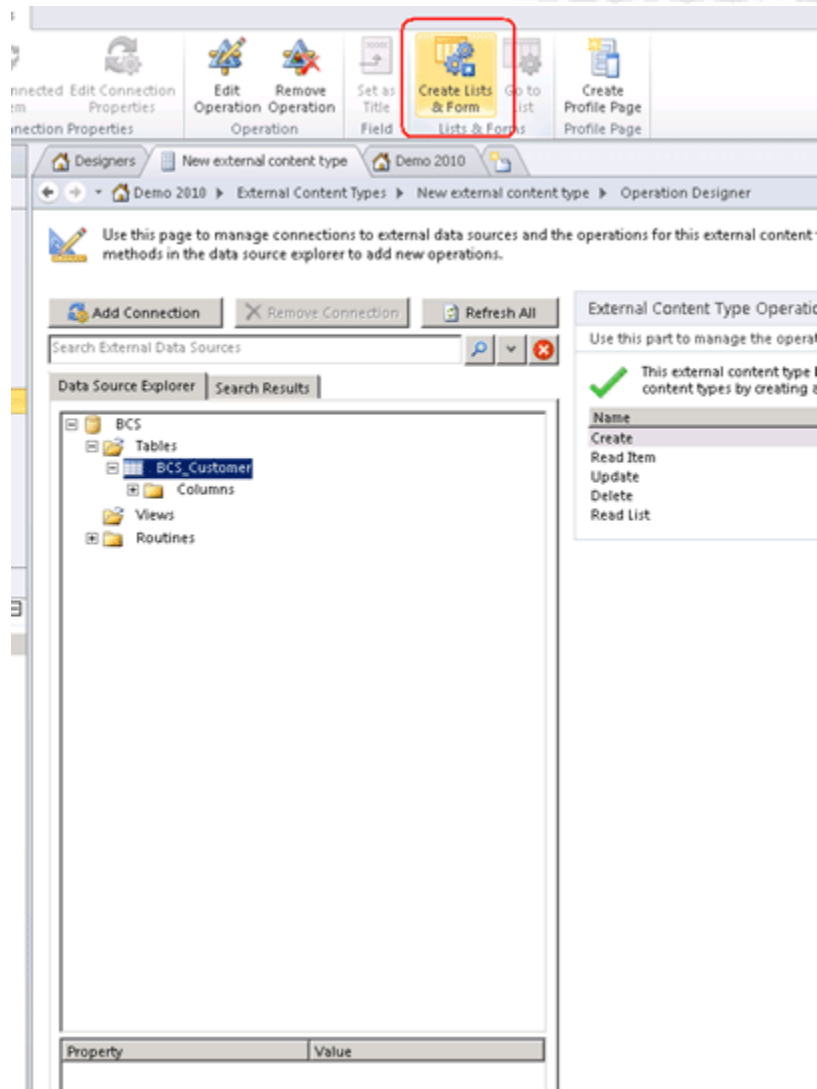


Figure 93

12. Once done, you will get an external list connected to our SQL table that you can add/update/delete data from. See Figure 94.

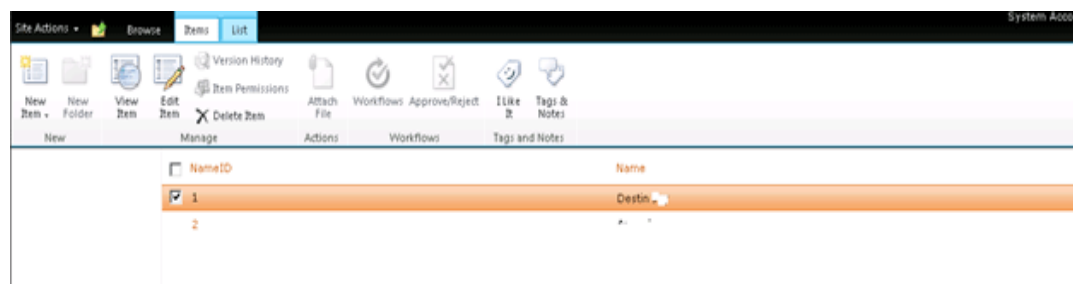


Figure 94

Customizing a Blog Site Using SharePoint Designer 2010

In this section, we are describing how to customize a SharePoint 2010 Blog Post. There are very few articles on customizing SharePoint 2010 blog posts. This will be useful if you work with SharePoint 2010 blog customization.

We are showing how to remove the table column with the post date in it; i.e. we have marked it with a Red circle. It is not as easy as it was in SharePoint 2007. If you open the page in the Designer then you can't even see this control in it. See figure 95.

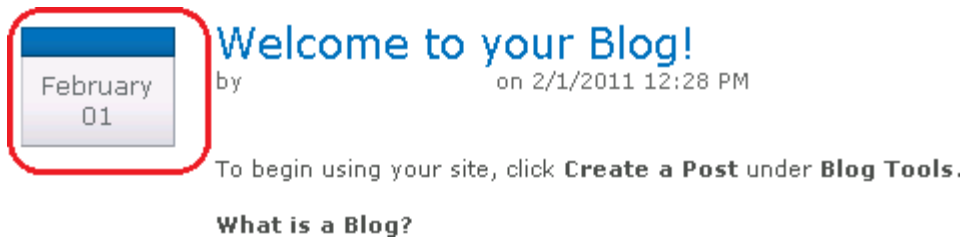


Figure 95

1. Open the blog.xsl, which you can find the file under the following location:

C:\Program Files\Common Files\Microsoft Shared\Web Server Extensions\14\TEMPLATE\LAYOUTS\XSL. See Figure 96.

 AssetPicker	28-02-2010 01:02	XSLT Stylesheet	25 KB
 blog	26-03-2010 21:24	XSLT Stylesheet	40 KB
 File icon	26-03-2010 21:24	XSLT Stylesheet	105 KB

Figure 96

2. Create a folder in the root of your site using the designer named XSL as shown below in Figure 97.



Figure 97

3. Copy or paste the file that we loaded or create a file named blog.xml and paste the contents. Change the name of the file; eg: Blogmine.xml.
4. In SharePoint Designer, open the Post.aspx file; it is under the post list.
5. Click on the Advanced Mode button from the ribbon. If you don't then it won't allow you to edit the page
6. Click on the empty cell where the Post will appear as shown below.

```

34      <tr valign="top" class="s4-die">
35      <td class="ms-pagebreadcrumb">
36          <asp:SiteMapPath SiteMapProvider="SPContentMapProvider" id="ContentMap" SkipLinkText="" />
37      </td>
38  </tr>
39  </ContentTemplate>
40  </SharePoint:UIVersionedContent>
41  <tr>
42      <td>
43          <table cellpadding="0" cellspacing="0" >
44              <tr>
45                  <td valign="top">
46                      <WebPartPages:WebPartZone runat="server" FrameType="TitleBarOnly" ID="Left" Title="
47                      <WebPartPages:XsltListViewWebPart runat="server" Description="Use the Posts list fo
48                      <ParameterBinding Name="dvt_sortdir" Location="Postback;Connection"/>
49                      <ParameterBinding Name="dvt_sortfield" Location="Postback;Connection"/>
50                      <ParameterBinding Name="dvt_startposition" Location="Postback" DefaultValue="" />
51                      <ParameterBinding Name="dvt_firsttrou" Location="Postback;Connection"/>
52                      <ParameterBinding Name="OpenMenuKeyAccessible" Location="Resource(wss,OpenMenuK
53                      <ParameterBinding Name="open_menu" Location="Resource(wss,open_menu)" />
54                      <ParameterBinding Name="select_deselect_all" Location="Resource(wss,select_dese
55                      <ParameterBinding Name="idPresEnabled" Location="Resource(wss,idPresEnabled)" />

```

7. Click on the "Tag Properties" window of SharePoint Designer on the right side of the screen; locate the property "XslLink" and type in the location of the new XSL file you created above; for me it will be /blog/xsl/blogmine.xml.

Open your file you just created blogmine.xml and open it in edit mode. Search for the code <td class = "ms-leftblogdate"> and remove this entire TD because it was making

the design look like this:

```

09      <xsl:variable name="showtopdate"><xsl:if test=".../@BaseViewID='0' or .../@BaseViewID='
10      <!-- Posts.aspx should show draft posts-->
11      <xsl:variable name="ShowDrafts"><xsl:if test=".../@BaseViewID='7'">1</xsl:if></xsl:variab
12      <xsl:variable name="thisNode" select="."/>
13      <xsl:if test="$ShowDrafts=1 or $thisNode/@ModerationStatus. = 0">
14      <table>
15      <tr class="ms-blogpostdatecol" valign="top">
16      <!-- <td class="ms-leftblogdate">
17      <div id="PostDateTopBox" class="ms-postcalendardateboxtop">
18      </div>
19      <xsl:if test="$MasterVersion = 4">
20      <div id="PostDateBottomBox" class="ms-postcalendardateboxbottom" style="overflow-x:hidden">
21      <div>
22      <xsl:value-of select="$thisNode/@PublishedDate.MonthDayOnly" disable-output-escaping="yes"
23      </div>
24      </div>
25      </xsl:if>
26      <xsl:if test="$MasterVersion != 4">
27      <div id="PostDateBottomBox" class="ms-postcalendardateboxbottom" style="overflow-x:hidden;f
28      <div>
29      <xsl:value-of select="$thisNode/@PublishedDate.MonthDayOnly" disable-output-escaping="yes"
30      </div>
31      </div>
32      </xsl:if>
33      </td-->
34      <td class="ms-rightblogpost">
35      <table dir="None" cellspacing="0" cellpadding="0" width="100%" border="0">
36      <tbody>
37      <tr>

```

8. Check in all of your changes.

9. Open the blog in the browser and you can see the changes.

Backup & Restore in SharePoint Designer 2010

Backup and Restore a Web Site

SharePoint Designer 2010 is designed to work with content on an existing SharePoint Web site. To back up a Web site, SharePoint administrators have to use the Central Administration Web Site for the collection. This feature is not supported in Designer 2010. In order to get a valid backup of a 2010 environment, you really need to work with the server administrators to do the backup. There is no realistic way to do it with only client access.

To restore a Web site, an administrator for the SharePoint Products and Technologies Web site uses the Central Administration Web site for the collection. In order to get a valid restore of a 2010 environment you really need to work with the server administrators to do the backup. There is no realistic way to do it with only client access.

Import Web Site Wizard

SharePoint Designer 2010 is designed to work with content on an existing SharePoint Web site. The ability to import a Web site from another server or from a file location is removed in 2010.

Publish Web Site

SharePoint Designer 2010 is designed to work with content on a SharePoint Web site. The ability to create, edits, or publish a disk-based Web site and work with remote Web sites is removed. SharePoint Designer 2007 did support disk-based Web sites.

Conclusion

SharePoint Designer 2010 is the core and the only design tool for designing, prototyping, and customizing both SharePoint Server 2010 and SharePoint Foundation 2010 sites; from small sites to enterprise-level internet sites and intranets.

Our recommendation to all of you is to utilize this great tool to its maximum extent. When you get a requirement, analyze if the given requirement can be implemented with the out of the box functionalities in SharePoint. Else, check if it can be done in Designer. If both the ways won't work, then you have to go for custom coding.

This is my first book and I am sincerely expressing my heartfelt thanks to my co-author and guide Mr. Mahesh Chand. I also would like to extend my thanks to my friends Aneesh Bhargavan and Jibin Koshy for all their help. Also, I would like to thank John Panicker and Jophin for the design work and Hari Kumar for the proof reading and editing.