

Microsoft® SharePoint® Online

Microsoft SharePoint Online Developer Guide

Published: December 2008

Information in this document, including URL and other Internet Web site references, is subject to change without notice. Unless otherwise noted, the companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted in examples herein are fictitious. No association with any real company, organization, product, domain name, e-mail address, logo, person, place, or event is intended or should be inferred. 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 Microsoft Corporation.

Microsoft may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Microsoft, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

© 2008 Microsoft Corporation. All rights reserved.

Microsoft, Excel, InfoPath, JScript, MSN, SharePoint, SQL Server, Visual C#, Visual Studio, and Windows are trademarks of the Microsoft group of companies.

All other trademarks are property of their respective owners.

Contents

1	Getting Started.....	6
1.1	Welcome to the Microsoft SharePoint Online Developer's Guide	6
1.2	About SharePoint Online.....	6
1.3	How SharePoint Online Works	6
1.4	Features of SharePoint Online	7
	Comparison of SharePoint Online Standard and Dedicated Offerings.....	7
1.5	SharePoint Online Standard Features at a Glance.....	8
1.6	Capabilities and Limitations of SharePoint Online.....	9
1.6.1	Customization Capabilities	9
1.6.2	Customization Limitations	9
1.7	Solution Scenario.....	10
2	Creating SharePoint Online Sites	11
2.1	Introduction	11
2.2	Selecting a Default Site Template.....	11
2.2.1	Differences between SharePoint Online and Office SharePoint Server	11
2.2.2	Collaboration Site Templates	12
2.2.3	Meetings Workspaces	13
2.2.4	Application Templates	13
2.3	Creating a Site Collection from the Microsoft Online Services Administration Center (MOAC)	16
2.4	Creating a Custom Site Template	21
2.5	Installing a Site Template	24
2.6	Removing a Site Template	25
2.7	Creating a New Site Using a Site Template.....	25
2.8	Example: Creating a New Site when Initiating a New Project	26
2.9	Example: Creating Blogs for the Knowledge Base in the Portal	27

3 Customizing SharePoint Online Applications 28

3.1 About Customizing SharePoint Online Applications28

3.2 Customizing Through the Browser28

 3.2.1 Applying Themes28

3.3 Customizing Master Pages Using SharePoint Designer 200729

 3.3.1 Creating a Minimal Master Page29

 3.3.2 Creating a Master Page from the Default Master35

3.4 Customizing Page Layouts.....37

 3.4.1 Opening a Page Layout for Editing37

 3.4.2 Adding Custom Content to a Page Layout38

 3.4.3 Inserting a Content Field39

3.5 Customizing Cascading Style Sheets.....39

 3.5.1 Applying a Custom Style Sheet to a SharePoint Online Site40

 3.5.2 Example: Changing the Background Color of the Page Using CSS Tools in SharePoint Designer 2007 .43

3.6 Managing Site Navigation in SharePoint Online Site.....45

 3.6.1 Navigation Editing and Sorting45

3.7 Example: Designing the Home Page48

 3.7.1 Example: Changing the Banner Image50

 3.7.2 Example: Customizing the Home Page Navigation Bar51

 3.7.3 Example: Adding Custom CSS to the Master Page52

4 Using a Data Form Web Part in SharePoint Online Applications..... 53

4.1 About the Data Form Web Part53

 4.1.1 Creating a Data Form Web Part Using SharePoint Designer53

4.2 Examples: Adding, Editing, Deleting, Sorting, and Filtering List Items Using the Data Form Web Part 55

4.3 Sorting and Grouping Data in a Data View60

4.4 Applying Conditional Formatting to a Data View.....60

4.5 Connecting Form Web Part to another Web part.....61

4.6 Example: Integrating Microsoft Live Search Web Service Using the Form Web Part.....62

4.7 Example: Adding Page Numbers to a Data Form Web Part.....74

5	Using SharePoint Online Web Services.....	77
5.1	List of Available SharePoint Online Web Services.....	77
5.2	List of Unsupported SharePoint Online Web Services	81
5.3	Example: Displaying All List Names from a SharePoint Online Site Collection	84
5.4	Example: Fetching List Collection Data from an SharePoint Online Site.....	89
5.5	Example: Fetching List Items from a SharePoint Online Site.....	93
6	Integrating Microsoft Silverlight with a SharePoint Online Page	97
6.1	Example: Using Silverlight to Fetch Data from a SharePoint Online Site Using Web Services.....	97
7	Workflows Using SharePoint Designer	108
7.1	About Workflows.....	108
7.2	Creating a No-Code Workflow Using SharePoint Designer	108
7.3	Collecting Data from the User and Processing It in the Workflow	113
7.4	Example: Send a SOW Document to the Project Manager for Review.....	127
8	Index	131

1 Getting Started

1.1 Welcome to the Microsoft SharePoint Online Developer's Guide

This guide describes the services that Microsoft® SharePoint® Online (standard offering) provides developers to start working with SharePoint Online. This guide also contains conceptual overviews and detailed information about the features that SharePoint Online supports.

In this guide you will learn about:

- Getting started with SharePoint Online
- Understanding SharePoint Online offerings
- Creating SharePoint Online sites
- Installing site templates
- Customizing SharePoint Online applications
- Integrating live data into SharePoint Online
- Using SharePoint Online Web Services
- Creating custom no-code workflows in SharePoint Online
- Extending SharePoint Online with Microsoft Silverlight

1.2 About SharePoint Online

- SharePoint Online, built on top of Microsoft Office SharePoint Server 2007, provides a single, integrated location where users can efficiently collaborate on tasks such as finding organizational resources, searching a site, managing content and workflow, and leveraging business insight to make better-informed decisions.
- SharePoint Online enables the users to easily create and manage custom and project-focused sites for collaboration including document sharing.
- Microsoft handles setup, provisioning, ongoing maintenance, and upgrades. This reduces the workload on your IT resources and helps them work on core strategic initiatives that help you to move your business forward.

1.3 How SharePoint Online Works

SharePoint Online simplifies IT management by removing the need to deploy, configure, monitor, update, or upgrade a collaboration solution on premises. From the Microsoft Online Services Administration Center, IT administrators can create new sites and provide access to specific users. SharePoint Online utilizes redundant and

geographically dispersed data centers. Each data center houses a reliable and redundant infrastructure to support the service. SharePoint Online offers you a comprehensive set of functionalities spread across the portal, collaboration, search, content management, and business process forms. With SharePoint Online, your enterprise can be up and running quickly.

1.4 Features of SharePoint Online

SharePoint Online provides the following features:

- Use of https for secure Internet access
- Standard templates including wikis, blogs, and surveys
- SharePoint Online site creation with online discussion areas, shared documents and meeting workspaces, document libraries with version control, and surveys
- Off-the-shelf content management features for documents and Web content
- Ability to create custom workflows from standard workflow activities and actions with Microsoft Office SharePoint Designer
- Site-wide search capability
- E-mail alerts when documents and information have been changed or added to a site
- Sign In application provides single-sign-in capability
- Built-in recycle bin
- Simple per-user monthly fee
- 99.9% scheduled uptime with financially secured service level agreements
- All upgrades and maintenance handled transparently
- Web form and round-the-clock phone-based tier-2 support for IT administrators

Comparison of SharePoint Online Standard and Dedicated Offerings

Standard Offering	Dedicated Offering
Multiple customers share a common infrastructure	Each customer has its own infrastructure
Customer priority: rapid deployment	Customer priority: heavy customization
No lower seat limit	For customers with more than 5000 seats
Cost efficiency optimized for customers with up to 20,000 users	Optimized for customers with 20,000+ users

1.5 SharePoint Online Standard Features at a Glance

	Online & Server
	Server Only

Portal	Collaboration	Content Management	Search	Business Process Forms	Business Intelligence
<ul style="list-style-type: none"> • RSS content syndication • Privacy and security • Audience targeting (by group only) • Site and document aggregation • Site Manager • Mobile device support • Portal site templates • SharePoint Sites and Documents Roll-up Web Parts • Client integration • Integration with Microsoft Office SharePoint Designer (except site backup and restore) • My Site personal site • Site Directory • User profiles 	<ul style="list-style-type: none"> • All standard templates • Wikis • Blogs • People and group lists • Calendars • E-mail integration • Task coordination • Surveys • Document collaboration • Issue tracking • Social Networking Web Part • Mail-enabled lists • Real-time presence and communication 	<ul style="list-style-type: none"> • Business document workflow support • Document information panel • Document action bar • Retention and auditing policies • Navigation controls • Content authoring • Page layouts • Site variations • Content publishing and deployment • Content management site templates • Policies, auditing, and compliance • Records repository • E-mail content as records • Legal holds • Integration with Microsoft information rights management 	<ul style="list-style-type: none"> • Security • People search • Business data search • Enterprise content sources • Cross site-collection search • Administration and management • Indexing controls 	<ul style="list-style-type: none"> • Forms libraries • Out-of-the-box workflows • Custom no-code workflows • Custom code workflows • Browser-based forms • Centralized forms management and control • Design once development model • Form Import Wizard • Integrated deployment model for no-code forms • Compatibility Checker 	<ul style="list-style-type: none"> • Integrated business intelligence dashboards • Key performance indicators • Filter Web Parts • Integrated flexible spreadsheet publishing • Sharing, management, and control of spreadsheets • Web-based business intelligence using Microsoft Office Excel® services • Data connection libraries • Business Data Catalog • Business data Web Parts

and the profile store • Colleagues and Memberships Web Parts		(IRM) • WYSIWIG Web content editor • Content staging • Slide libraries • Document management site templates • High-fidelity Web sites with consistent branding		• Business data actions • Report Center
---	--	---	--	--

1.6 Capabilities and Limitations of SharePoint Online

The standard version provides many of the features you expect from Office SharePoint Server, such as document sharing, collaboration, and workflows. In addition, you can use Office SharePoint Designer 2007 to design and modify your site's appearance and capabilities.

1.6.1 Customization Capabilities

The customization capabilities of SharePoint Online enables the developers and designers to:

- Use Office SharePoint Designer 2007 to create and deploy no-code workflows, customize content types, taxonomy, and branding via master pages and layouts. You can also create and deploy site templates.
- Use the Data Form Web Part to create applications to mash up, filter, roll up, and render SharePoint data or data consumed from a Web service such as RSS feeds in new ways.
- Use Microsoft Office InfoPath® to design forms for workflows, provided the forms contain no custom code.
- Use the Office SharePoint Server Web services to access and manipulate SharePoint data remotely.

1.6.2 Customization Limitations

In the current release of the services, the following actions are not supported:

- Use inline code, build coded workflows, or develop Office InfoPath forms with coded business logic.
- Deploy features, solutions, pluggable authentication providers, Web Parts, site definitions, or other modifications that require deployment and configuration on the server.
- Modify built-in SharePoint files, web.config settings, security policy, and other elements.

- Make configuration changes that affect the Web server or the Microsoft .NET Framework.
- Make changes or add capabilities that require a custom database or changes to the database schema.

1.7 Solution Scenario

This solution scenario is used in the examples in this guide.

Software service programs are often carried out in multiple locations to leverage skills and cost structures. The key concern is the visibility of the operations, status, and teams across time zones collaborating more effectively, to the project and program governance boards.

The **CEO** of Contoso would like to see the status of all the projects that Contoso is handling, along with the exceptions and the audit reports. He can provide his vision to all the teams and can drill into individual projects in case he needs more information.

The **program manager** in Contoso is handling multiple projects and has a team working under him that is placed in various locations in the region. The program manager would like to see project health indicators and detailed status reports for each project. He would also like to ensure that there is learning and knowledge transfer among teams.

He also performs a set of activities such as initiating new projects, tracking project status, and identifying issues. Because he is also the member of the administrator group, he can perform certain administrator-related activities and has full control of his project space.

The project status will be updated by the project managers, who are working on individual projects and are also responsible for project artifacts, governance reports, and minutes. In addition to resource management, the project manager will also manage issues, risks, and schedules. He will be responsible for **project artifacts** that include project-specific documents containing statements of work (SOW), software requirement specifications (SRS), design documents (high-level, detailed), and governance artifacts such as reports and minutes of meetings.

A **knowledge base** is a reservoir of information that contain links to **blogs** and a **document store**. The document store contains such materials as standard documents, best practices, and process documents.

The software quality assurance team and an architect will be responsible for the process and the technical content in the document store at the program level.

For this solution scenario, Contoso developed a collaboration portal using SharePoint Online and Microsoft Exchange Online.

2 Creating SharePoint Online Sites

2.1 Introduction

A Site Collection is a set of web sites on a web server, all of which have the same owner and share administration settings. Each site collection contains exactly one top-level Web site, and can also contain one or more subsites. There can be multiple site collections on each Web server.

A subsite is a complete Web site stored in a named subdirectory of the top-level Web site. Each subsite can have administration, authoring, and browsing permissions that are independent from the top-level Web site and other subsites. A subsite can also have subsites of its own. Because every site below the top-level site is actually a subsite, each subsite is generally called simply a site.

Application template addresses a business scenario and provides a base of functionality that can be either used directly out of the box, or customized for company specific needs.

Custom templates are a way of packaging up a set of changes to an existing site definition and making those available for new sites and lists. Every custom template is based on a site definition. Custom templates are stored in the database and are made available through the central template gallery or through site collection template galleries.

2.2 Selecting a Default Site Template

The site templates included in Microsoft SharePoint Online contain pages, lists, libraries, and other elements or features that support a wide range of specific content publishing and content management designed to meet the requirements of your organization.

The standard site templates available in SharePoint Online applications are grouped into the following categories:

- Collaboration
- Meetings
- Custom

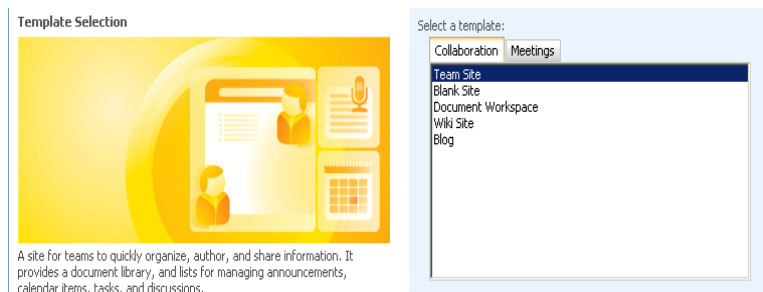
2.2.1 Differences between SharePoint Online and Office SharePoint Server

- SharePoint Online doesn't have a global site template library. Therefore you cannot create site collections from a custom site template. Because new site collections for SharePoint Online are created in the Microsoft Online Services Administration Center, only the default options are available. However, you can create subsites for SharePoint Online that use the custom templates.

- In this release of SharePoint Online, Server Admin templates are not supported, nor are any other custom templates that require adding code on the server side.

2.2.2 Collaboration Site Templates

The site templates in the collaboration group are designed to help the teams within an organization to work together on projects, collaborate on documents, and share information.

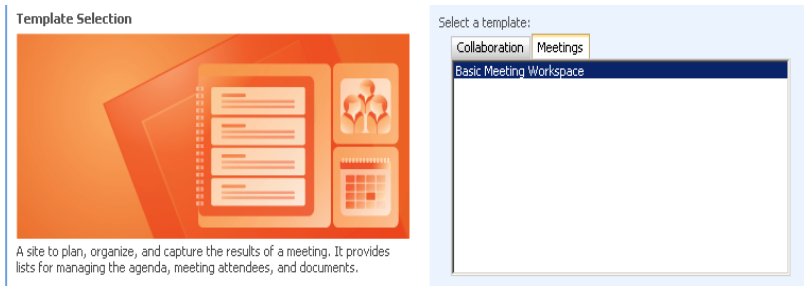


- **Team Site:** Select this site template to create a site that teams can use to create, organize, and share information. The template includes a document library, an announcements list, a calendar, a contacts list, and a links list by default.
- **Blank Site:** Select this site template to create a site with a blank home page that you want to customize. The template includes a site image Web Part and tools to insert other Web Parts. This template is useful when you do not want to use any of the existing templates, Web Parts, lists, or libraries as a starting point.
- **Document Workspace:** Select this site template to create a site that will help you coordinate the development of one or more related documents with other people. This site template provides tools to share and update files, and to keep people informed about the status of the files. It includes a document library, an announcements list, a tasks list, a members list, and a links list.
- **Wiki Site:** Select this site template to create a site where the users can quickly and easily add, edit, and link Web pages. This site template includes wiki pages, a links list, and a wiki page library.
- **Blog:** Select this site template to create a site where the users can post information quickly and allow people to comment on it. Sometimes known as weblogs, blogs are online journals where you can share your ideas quickly in an informal, chronological format.

This site template includes a posts list for storing blog posts, a blogs list for links to other blogs, a categories list, a comments list, a links list for links to related resources, a photo library, and tools to manage your posts and other resources.

2.2.3 Meetings Workspaces

The site template in the meetings group is designed to help teams within an organization to manage basic types of meetings. There is one template in the meetings group.

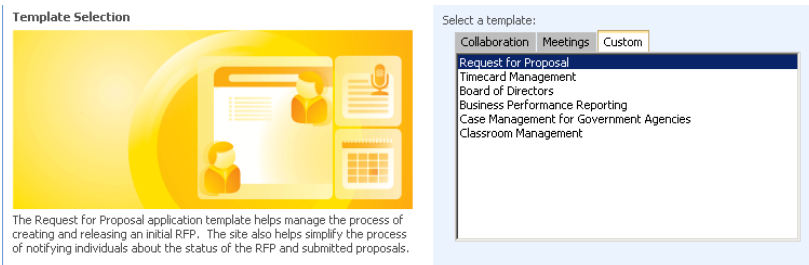


- **Basic Meeting Workspace:** Select this site template to create a site that will help you plan, organize, and track your meetings with the rest of your team. The template includes an objectives list, an attendees list, an agenda, and a document library.

2.2.4 Application Templates

Microsoft has developed forty application templates that can be downloaded at no extra charge. Each application template addresses a business scenario and provides a base of functionality that can be either used directly out of the box, or customized for company specific needs. These custom templates (*.stp files) are site templates that are easy for any site administrator to install in the template gallery without requiring server administration access.

Note: SharePoint Online is not installed with these custom templates. You must upload and install each template yourself. The **Custom** tab appears only when you have uploaded at least one custom template to SharePoint Online.



- **Board of Directors:** This application template provides a single location for an external group of members to store and locate common documents such as quarterly reviews, shareholder meeting notes, and annual strategy documents. The template also tracks tasks, issues, and calendar items so that the board members have a single location from which to view relevant information.
- **Business Performance Reporting:** This application template helps organization managers to track customer satisfaction through a combination of surveys and discussions.
- **Case Management for Government Agencies:** This application template helps case managers track the status and tasks required to complete their work. When a case is created, standard tasks and documents are also created that are modified based on the work that each case manager has completed.
- **Classroom Management:** This application template helps instructors and students organize and store information that is related to a particular class. The template includes document libraries to store assignments and lecture notes as well as calendars and announcement capability to enable communication from the instructor to students.
- **Clinical Trial Initiation and Management:** This application template helps teams manage the processes of tracking clinical trial protocols, objective setting, subject selection, and budget activities.
The site provides useful Microsoft Office Word 2007 templates as well as the capability to create, track, and assign tasks and issues related to a particular clinical trial.
- **Competitive Analysis:** This application template helps teams organize information about competitors and their products.

The site provides useful 2007 Microsoft Office system documents to help perform Strengths, Weaknesses, Opportunities, and Threats (SWOT) analysis and make use of other useful competitive analysis techniques. Links to industry, company, and product news can also be included to provide a single location for all competitive information.

- **Discussion Database:** This application template provides a location where team members can create and reply to discussion topics.
- **Disputed Invoice Management:** This application template helps accounts payable departments track open invoices, along with the potential savings associated with paying the invoice early.

The site includes useful templates for analyzing the reasons for the invoices being disputed, as well as tracking whom to contact for more information.
- **Employee Activities:** This application template helps you to manage the creation and attendance of events for employees.

Activity owners use the site to review proposals for new activities and create event calendar items. Employees use the site to sign up for activities, as well as track the activities attended in the past.
- **Employee Self-Service Benefits:** This application template provides tools for an organization to inform employees about available benefits, as well as enable them to enroll for each benefit.
- **Employee Training Scheduling and Materials:** This application template helps instructors and employees manage courses and related materials.

Instructors use the site to add new courses and organize course materials. Employees use the site to schedule attendance for a course, track courses they have attended, and provide feedback.
- **Equity Research:** This application template helps teams collaborate on researching stocks and other equities.

It provides a central location for teams to store documents, post links, track news, and hold discussions related to the equities that are tracked by the site.
- **Integrated Marketing Campaign Tracking:** This application template helps marketing managers track the implementation and success of outbound marketing activities.

The template allows a manager to create marketing activities and track the results of those activities, such as responses generated and sales completed.
- **Manufacturing Process Management:** This application template helps teams to model and track manufacturing processes as well as tasks and issues that arise in the upkeep of these processes.
- **New Store Opening:** This application template helps a team manage the opening of new store locations or remodeling of an existing store location.

The site provides a single location to manage tasks, issues, and documents for all store opening processes, enabling end users to view relevant information and providing project managers with insight across the entire project.

- **Product and Marketing Requirements Planning:** This application template enables teams to manage the process of collecting and documenting requirements for new products.

The site provides several 2007 Office system templates for providing useful techniques for marketing, product, and steering committee actions as well as a template for meeting notes and financial information.

- **Request for Proposal:** This application template helps manage the process of creating and releasing an initial request for proposal (RFP), collecting submissions of proposals, and formally accepting the selected proposal from among those submitted.

The site also helps simplify the process of notifying individuals about the status of the RFP and submitted proposals.

- **Sports League:** This application template helps an intracompany league administrator manage a baseball league.

The site tracks team information, players, captains, and scheduled team activities such as games, practices, and social events. The site also enables discussions between league members through a league discussion board.

- **Team Work:** This application template provides a place where project teams can upload background documents, track scheduled calendar events, and submit action items that result from team meetings.

The site also tracks the creation and purpose of "subteams" and enables discussion on topics created by members of the team.

- **Timecard Management:** This application template helps teams track hours spent working on various projects.

The site enables team members to "punch in" on a particular project and "punch out" when they cease work. The system automatically generates the time worked by project, and can show managers who is working on a particular project, total hours versus budgeted time, and the details of who worked on each project that is entered into the site.

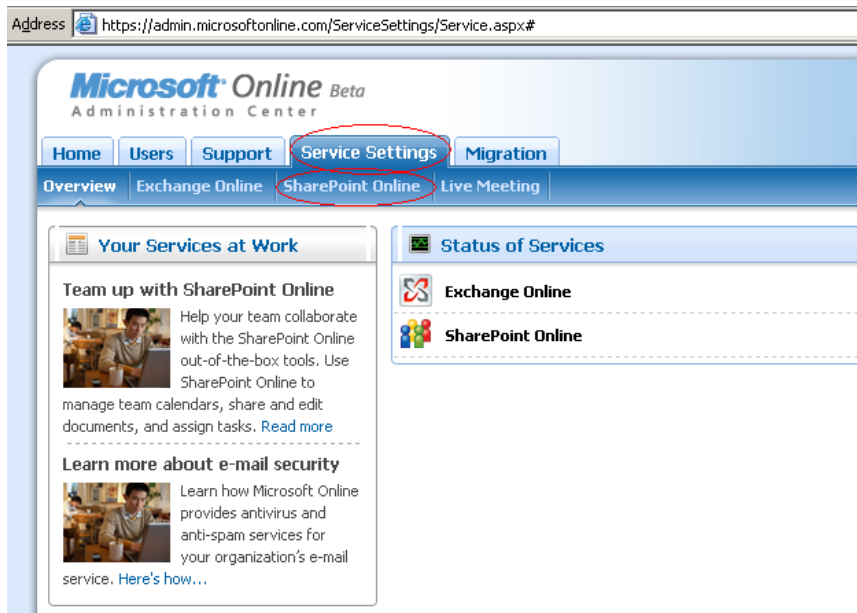
2.3 Creating a Site Collection from the Microsoft Online Services Administration Center (MOAC)

A SharePoint Online site collection is a hierarchical set of sites that can be managed together. Sites within a site collection have common features, such as shared permissions; galleries for templates, content types, and Web Parts; and often a shared common navigation interface. Site collections can include different types of subsites, along with team sites, meeting workspaces, document workspaces, blogs, and wikis.

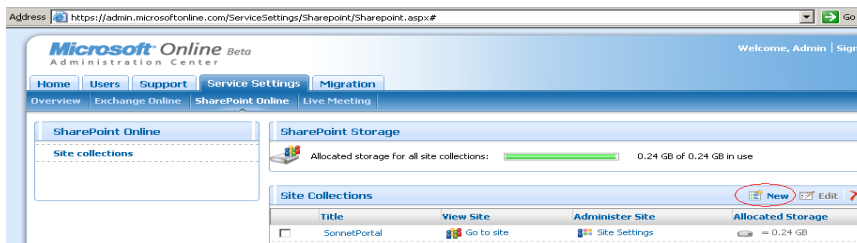
Administrators of your SharePoint Online service can create organization-wide site collections. The users can create new subsites within each site collection. To create your site collections, go to the Microsoft Online Services Administration Center. (A site collection can only be created via the Administration Center.) Then add users.

To create a site collection through the Administration Center

1. Go to the Administration Center (<https://admin.microsoftonline.com>).
2. Log on to the Administration Center with administrator credentials.
3. Click the **Service Settings** tab, and then click the **SharePoint Online** subtab.



4. In the **Site Collections** pane, click **New**.



- Under **Site Information**, enter the information requested. Type a title and the description for the site collection. Select a template, and then type an e-mail address to receive access requests and notifications for the site.

New SharePoint Online Site Collection

Site Collection Properties

Confirmation

Site collection properties
Enter information for your new SharePoint Online site collection.

Site Information * Required fields

* Title:

* Description:

Template:
Team Site

* Send access requests and notifications to this e-mail address:

Note: Email address must match the address of site owner or administrator

Site Storage Information

You can allocate up to 50 GB of storage to an individual site collection.

* Allocated storage for this site collection:
 GB

Remaining storage available for this site collection:
0.26 GB

Help for this wizard

Create Cancel

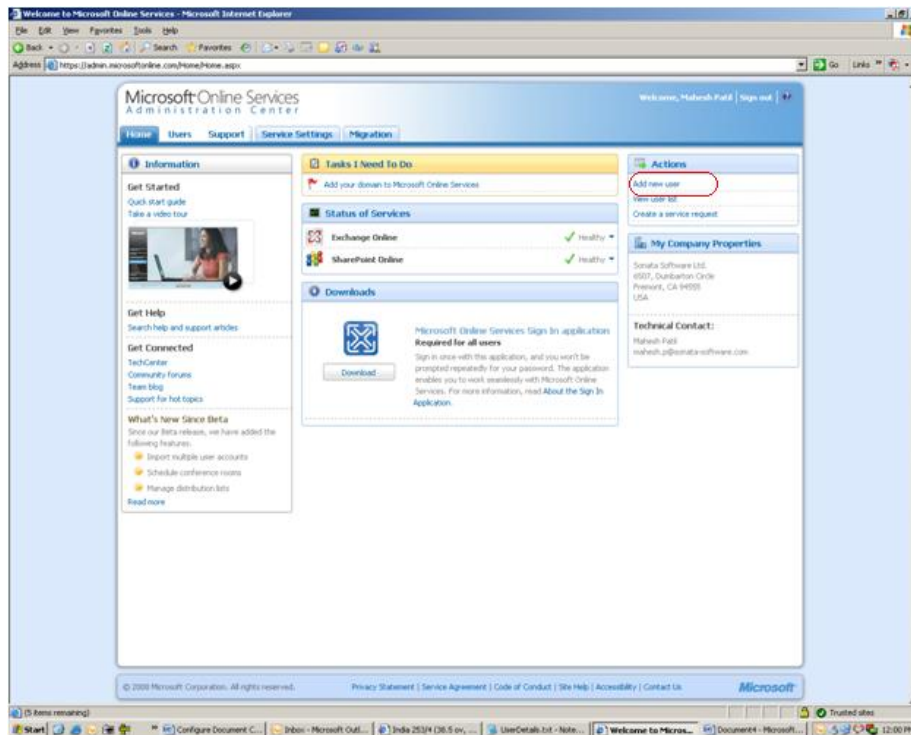
- Under **Site Storage Information**, choose the amount of storage to allocate to this collection. When you finish, click **Create**.

Note: The storage size must be entered as 0.25 or 0.75.

- On the **Confirmation** page, click **Finish**.

To add users

- Log on to the Administration Center with admin privileges.
The **Information** tab appears.



2. In the **Actions** pane, click **Add new user**.
The **New User Wizard** appears.
3. On the **General** page of the **New User Wizard**, enter information in the required fields, and then click **Next**.

Note: Fields marked with red asterisks (*) are mandatory.

New User [Close]

General properties | **General**
Enter the user's information

Security settings

Services | **General** * Required fields

Confirmation

* User name: Domain: [v]

* First: * Last: * Display:

Employee Information

Job title:

Department: Office:

Phone and address

Office phone: Mobile phone: Fax number:

Street address:

City: State or province:

ZIP or postal code: Country or region:

Help for this wizard [Next] [Cancel]

4. On the **Security Settings** page, under **Do you want this user to have service administrator permissions?**, select **Yes** if you want to assign service administrator permissions; otherwise select **No** (the default).
5. Under **Do you want to enable this user's account?**, select **Yes** (the default).

New User ⌵

General properties

Security settings
Enter the user's permission levels security settings

Services

Confirmation

Password: **Automatically you will get the password here**

Do you want this user to have service administrator permissions?
Caution: Administrators have permissions to access and change users' service resources, passwords, and permissions.

Yes
 No

Do you want to enable this user's account?

Yes
 No

[Help for this wizard](#)

2.4 Creating a Custom Site Template

A custom site template is a feature that lets you create multiple sites with a consistent look and built-in functionality. Application templates described above are also custom templates. Here we show how to create one such custom template. This feature is particularly useful when you need to create new client sites quickly and apply updates to a group of sites. You can create custom SharePoint Online site templates to control the following:

- Style sheets and headers for site administration pages
- .aspx program files that you want to include in a site (for example, default.aspx)
Note: The .aspx pages cannot be used to execute custom server side code. You can reference existing subjects or add code that runs on the client.
- Web Parts that you want to include on each Web Part page in a site

- Lists and document libraries that you want to create as part of the site

In addition to customizing a site template, you can customize the base .aspx files such as default.aspx or Allitems.aspx , which are executed when SharePoint Online displays a site.

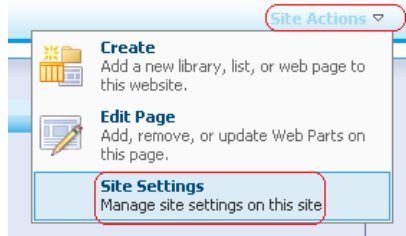
The following key steps will help you create a custom site template:

- Create a site using any default site template of SharePoint Online. For more information, see section 2.6, [Creating a New Site Using a Site Template](#).
- As per your requirements, customize the template: add Web Parts, document libraries, lists, style sheets, etc, that you want to have as part of the site.
- Save the site as a custom site template.

Formatted: Font: Arial, Italic

To save the site as a custom site template

1. Open the SharePoint Online site that you have created, using administrator privileges.
2. On the **Site Actions** menu, click **Site Settings**.



3. Under Look and Feel, click Save site as template.

Note: If you do not see **Save site as template** in the **Look and Feel** section, then go to **Site Collection feature** in the **Site Collection Administration** section, deactivate **Office SharePoint Server Publishing Infrastructure**, and then go to **Site features** in the **Site Administration** section, and deactivate **Office SharePoint Server Publishing**.
4. In the **File Name** box, type **MainTemplate** as the name for the template file.

Note: You can enter a name of your choice.
5. In the **Template name** and **Template description** boxes, type **CustomTemplateMain** for both the template name and description.

The name and description of this template appear on the Web site template picker page when users create new Web sites.
6. Clear the **Include Content** check box, and click **OK** twice.

Note: Include content in your template if you want the new Web sites that were created from this template to include the contents of all the lists and document libraries in this Web site. Some customizations, such as custom workflows, are present in the template only if you choose to include content. Including content can also increase the size of your template.

Home > Site Settings > Save as Template

Save Site as Template

Use this page to save your Web site as a site template. Users can create new Web sites from this template.

File Name
Enter the name for this template file.

File name: MainTemplate .stp

Name and Description
The name and description of this template will be displayed on the Web site template picker page when users create new Web sites.

Template name: CustomTemplateMain

Template description: CustomTemplateMain

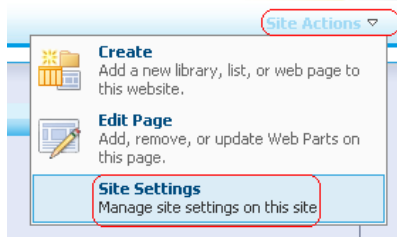
Include Content
Include content in your template if you want new Web sites created from this template to include the contents of all lists and document libraries in the Web site. Some customizations, such as custom workflows, are present in the template only if you choose to include content. Including content can increase the size of your template.

Include Content

Caution: Item security is not maintained in a template. If you have private content in this Web site, enabling this option is not recommended.

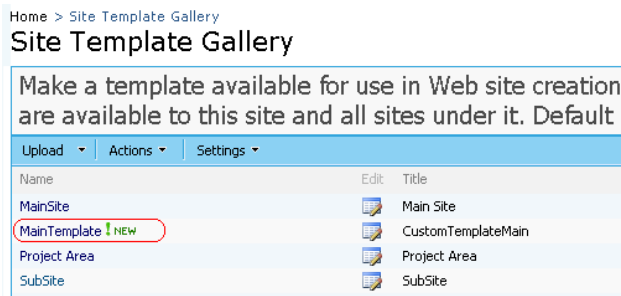
OK Cancel

7. On the **Site Actions** menu, click **Site Settings**.

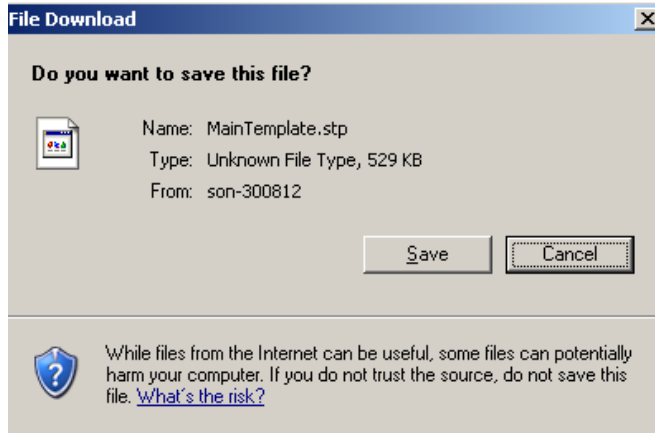


8. Under **Galleries**, click **Site templates**.

9. Click **Main Template**, which you created in step 4.



10. In the **File Download** dialog box, click **Save** to save a local copy (specify path).



Now this custom site template is ready to be uploaded to the SharePoint Online site.

2.5 Installing a Site Template

To install a site template

1. Log on to the SharePoint Online site as a member of the Owners group.
2. On the **Site Actions** menu, click **Site Settings**. (If the publishing feature is activated, also click **Modify All Site Settings**.) Note that Publishing needs to be activated under site collection administration/site collection features and then site administration/site features in that order.
3. In the **Galleries** section, click **Site templates**.
Note: If you do not see **Site templates** in the **Galleries** section, you might not be at a top-level site. In the **Site Collection Administration** section, click **Go to top-level site administration**.
4. Click **Upload** to save an application template to this SharePoint Online site.
Note: To upload more than one application template, click **Upload Multiple Files**.
5. Click **Browse**.
The **Choose File** dialog box opens.
6. Browse to the folder where the *template_name.stp* file is stored (where *template_name* is the name of the relevant template), select the file, and click **Open**.
7. Click **OK**.

The uploaded template appears in the custom group templates list.

2.6 Removing a Site Template

To remove a site template

1. Log on to the top-level SharePoint Online site as a member of the Owners group.
2. On the **Site Actions** menu, click **Site Settings**.
3. In the **Galleries** section, click **Site templates**.
4. In the list of site templates, click the application template that you want to remove, and then click **Edit**.
5. Confirm that this is the application template to remove, and then click **Delete Item**.
6. Click **OK** to confirm the deletion.

The application template is now unavailable to SharePoint Online sites, and it has been removed from the SharePoint Online site template gallery.

Note: Because this is a soft deletion, you can click the **Recycle Bin** link, select the template, and then either click **Restore Selection** to get the template back into the Gallery or click **Delete selection** to delete permanently.

2.7 Creating a New Site Using a Site Template

To create a new site, you can use either the default templates that are available in SharePoint Online or the custom templates that are uploaded to the SharePoint Online site.

To create a new site

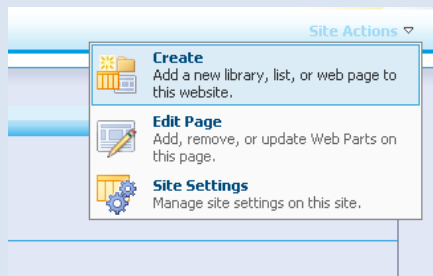
1. Log on to your SharePoint Online site with admin credentials.
2. Click **View all Site Content**
3. Click **Create**.
4. Under **Web Pages**, click **Sites and workspaces**.
5. On the **New SharePoint Site** page, fill in the information about your new site.
6. In the **Template Selection** section, click the **Custom** tab to use a custom template, or click the **Collaboration** or **Meetings** tab to use a default template.
7. Select a site template that you want to use to create the site.
8. Select a site template that you want to use to create the site.
9. Click **Create**.

2.8 Example: Creating a New Site when Initiating a New Project

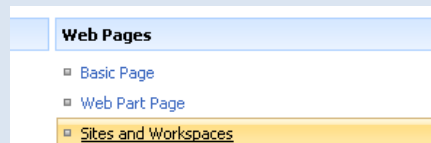
When a program manager initiates a new project, the administrator needs to perform the following steps to create a site for the initiated project.

To create a new site

1. Log on to your SharePoint Online site with admin credentials.
2. Click **View all site settings**.
3. Click **Create**.



4. Under **Web Pages**, click **Sites and workspaces**.



5. On the **New SharePoint Site** page, fill in the information about your new site.
6. In the **Template Selection** section, click the **desired template**
7. Click **Create**.

Home > Create > New SharePoint Site

New SharePoint Site

Use this page to create a new site or workspace under this SharePoint site. You can specify a title, Web site address, and access permissions.

Create Cancel

Title and Description
Type a title and description for your new site. The title will be displayed on each page in the site.

Title:

Description:

Web Site Address
Users can navigate to your site by typing the Web site address (URL) into their browser. You can enter the last part of the address. You should keep it short and easy to remember.
For example, <https://sonatasoftware1microsoftonlinecom-1.sharepoint.com/.../lastname>

URL name:

Template Selection

Select a template:

Collaboration Meetings

Team Site
Blank Site
Document Workspace
Wiki Site
Blog

A site for a person or team to post ideas, observations, and expertise that site visitors can comment on.

Permissions
You can give permission to access your new site to the same users who have access to this parent site, or you can give permission to a unique set of users.
Note: If you select **Use same permissions as parent site**, one set of user permissions is shared by both sites. Consequently, you cannot change user permissions on your new site unless you are an administrator of this parent site.

User Permissions:
 Use same permissions as parent site
 Use unique permissions

Navigation Inheritance
Specify whether this site shares the same top link bar as the parent. This setting may also determine the starting element of the breadcrumb.

Use the top link bar from the parent site?
 Yes No

2.9 Example: Creating Blogs for the Knowledge Base in the Portal

Using the blogs, users can share any information about their technologies and their projects. The administrator can create blog sites using the following steps:

To create a blog

1. Log on to the SharePoint Online site as a member of the Owners group.
2. On the **Site Actions** menu, click **Site Settings**.
3. In the **Site Administration** section, click **Sites and workspaces**.
4. Click **Create**.
5. On the **New SharePoint Site** page, fill in the information about your new site.
6. In the **Template Selection** section, under **Select a template**, click the **Collaboration** tab.
7. Click the **Blog** template.
8. Click **Create**.

3 Customizing SharePoint Online Applications

3.1 About Customizing SharePoint Online Applications

You can customize Microsoft SharePoint Online sites by adding or removing pages, changing the appearance of the pages, changing the site navigation, and making other customizations to fit the user requirements.

You can customize the look and feel of the SharePoint Online applications using:

- Site themes
- Master pages
- Page layouts

3.2 Customizing Through the Browser

You can perform basic customization from within the browser, using links from the **Home**, **Create**, and **Site Settings** pages of the Web site. From the browser you can perform basic customizations such as the following:

- Add a list.
- Change the layout of the home page.
- Change the picture on the home page.
- Add an out-of-the-box Web Part to a Web Part page.
- Change a site's display name (not the URL).

3.2.1 Applying Themes

You can apply themes and borders to your SharePoint Online sites. A theme is a unified set of design elements and color schemes that give your pages an attractive and consistent appearance.

A theme changes the appearance of a page. In addition it enables you to do the following:

- Manage the appearance of your pages in one convenient place.
- Choose a theme as the default appearance for your SharePoint Online site, to have the theme applied to all the new and existing pages.
- Change or remove the default theme, to have the changes automatically applied to the entire site.
- Apply themes to individual pages.

3.3 Customizing Master Pages Using SharePoint Designer 2007

Master pages, a Microsoft ASP.NET 2.0 feature, lets child pages inherit and reuse a common design. This makes development and maintenance of the site much easier. For new pages, you do not have to redo the work that you did in the master page. At the time of rendering, the page brings in the code from the master so it is always up to date.

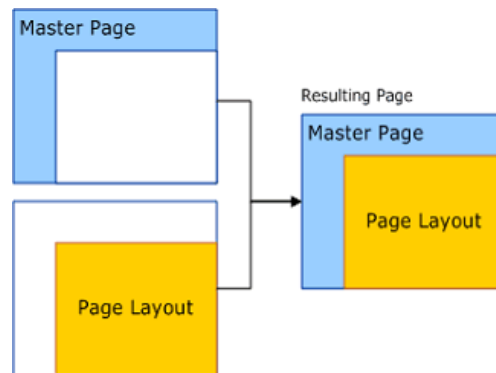
Master pages provide a consistent look and feel for all the pages in a SharePoint Online site. By changing the styles or content of a master page, you can quickly customize the appearance of an entire site.

A master page (any ASP.NET page with a .master extension) is a template with placeholders for content. Much like a regular ASP.NET page that invokes user controls to render content in Microsoft .NET Framework syntax, these placeholders are implemented using the `<ASP:ContentPlaceHolder>` tag. Placeholders can be nested, so that a top-level placeholder may have multiple levels of child placeholders, much as a parent `<div>` tag can have many nested child `<div>` tags.

The master page by itself usually contains very little actual content, other than headers and navigation. It is just a framework for one or more layout pages that comprise the body of the page.

You can open the SharePoint Online site in Microsoft Office SharePoint Designer 2007 and edit it there.

The following figure shows the master pages concept.



3.3.1 Creating a Minimal Master Page

A master page contains references to elements that you can share across multiple pages in an SharePoint Online site. This includes navigation, search controls, logon controls, and banner images.

A master page can also contain the cascading style sheet (CSS) and ECMAScript (as well as Microsoft JScript[®] and JavaScript) references that define the overall look and feel

of your site. Commonly, every site and therefore every page in your site collection uses the same master page to present a consistent user experience across the entire site collection.

Depending on your requirement, you can use a different master page for one or for all of the sites in your site hierarchy, to distinguish the various areas of your portal.

3.3.1.1 Master Page Galleries

When you provision a site collection in SharePoint Online, the system creates a master page gallery that contains all the master pages and page layouts for that site collection. If the site collection uses either the collaboration or meeting workspace templates, the master page gallery includes several master pages that are provided with SharePoint Online, such as BlueBand.master.

You can use any of these master pages as they are, or you can customize them fully to create unique branding for your site.

Home > Master Page Gallery

Master Page Gallery

Use the master page gallery to store master pages. The master pages in this gallery are available to this site and any sites underneath it.

Type	Name	Modified	Modified By	Checked Out To	Approval Status	Contact	Hidden Page	Associated Content Type
	Editing Menu	9/22/2008 11:21 PM	Admin		Approved			
	en-us	8/21/2008 6:35 AM	Admin		Approved			
	Preview Images	9/22/2008 11:21 PM	System Account		Approved			
	AdvancedSearchLayout.aspx	8/11/2008 8:43 AM	System Account		Approved			
	ArticleLeft.aspx	8/21/2008 6:35 AM	Admin		Approved			Article Page
	ArticleLinks.aspx	8/21/2008 6:35 AM	Admin		Approved			Article Page
	ArticleRight.aspx	9/30/2008 10:17 PM	Admin		Approved			Article Page
	BlackBand.master	8/21/2008 6:35 AM	Admin		Approved			
	BlackSingleLevel.master	8/21/2008 6:35 AM	Admin		Approved			
	BlackVertical.master	8/21/2008 6:35 AM	Admin		Approved			
	BlankWebPartPage.aspx	9/30/2008 10:17 PM	Admin		Approved			Welcome Page
	BlueBand.master	8/21/2008 6:35 AM	Admin		Approved			
	BlueGasBand.master	8/21/2008 6:35 AM	Admin		Approved			
	BlueTabs.master	8/21/2008 6:35 AM	Admin		Approved			
	BlueVertical.master	8/21/2008 6:35 AM	Admin		Approved			
	default.master	9/15/2008 2:32 AM	Admin		Approved			
	default_copy(1).master	9/15/2008 2:27 AM	Admin		Draft			
	DefaultLayout.aspx	9/30/2008 10:16 PM	Admin		Approved			Welcome Page
	NewsHomeLayout.aspx	8/11/2008 8:43 AM	System Account		Approved			
	OrangeSingleLevel.master	8/21/2008 6:35 AM	Admin		Approved			
	PageFromDocLayout.aspx	8/21/2008 6:35 AM	Admin		Approved			Article Page
	PageLayoutTemplate.aspx	8/21/2008 6:35 AM	Admin		Approved		Yes	
	PeopleSearchResults.aspx	8/11/2008 8:43 AM	System Account		Approved			
	RedirectPageLayout.aspx	8/21/2008 6:35 AM	Admin		Approved			Redirect Page
	ReportCenterLayout.aspx	8/11/2008 8:43 AM	System Account		Approved			
	SearchMain.aspx	8/11/2008 8:43 AM	System Account		Approved			
	SearchResults.aspx	8/11/2008 8:43 AM	System Account		Approved			
	TabletPageLayout.aspx	8/11/2008 8:43 AM	System Account		Approved			
	Test.master	10/6/2008 6:16 AM	Admin	Admin	Draft			

3.3.1.2 Why Begin with a Minimal Master Page

Creating and completing a master page to begin your SharePoint Online site customization requires planning and time. This section shows you how to create a minimal master page that includes only the minimal functionality that SharePoint Online requires to provide a stable platform on which you can build your own master pages.

Creating a minimal master page can help you avoid the time-consuming process of backing code out of a preexisting .master page such as BlueBand.master, or removing functionality and then rebuilding it when your requirements change.

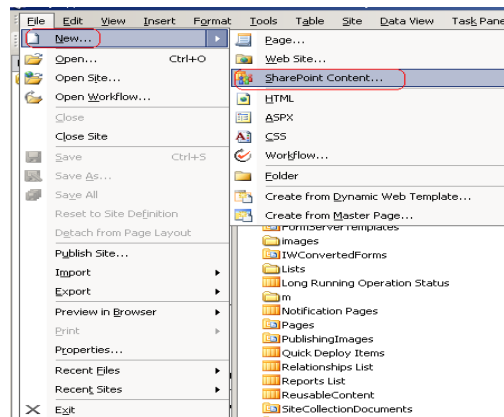
You can, of course, create a master page from scratch. However, it is not recommended, because an empty master page does not include all the content placeholders that the SharePoint Online site page model needs to work correctly.

The sample code in the following procedure includes only the elements that the SharePoint Online Site page model requires: the necessary content placeholders and controls to work with the page layouts that are included in a default SharePoint Online site. The default SharePoint Online site requires a master page that includes a title, branding, logon functionality, search functionality, breadcrumb functionality, and basic structural elements such as page areas, separators, borders, consoles, and description placeholders.

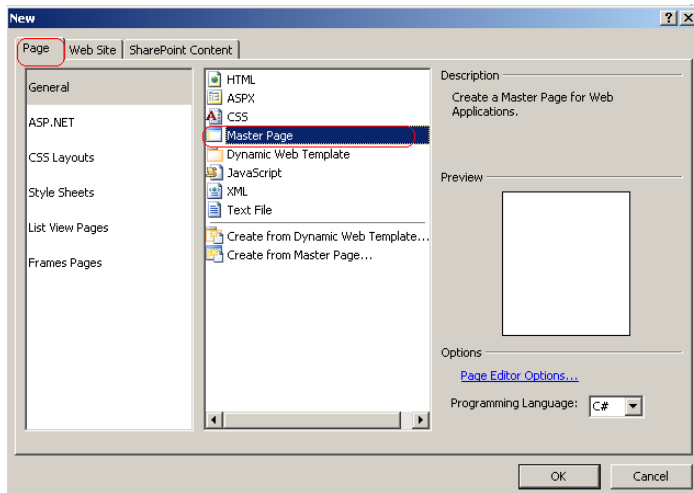
The following procedure uses Office SharePoint Designer 2007 as the master page design environment. You can, however, use a text editor.

To create a minimal master page

1. Open Office SharePoint Designer 2007.
2. On the **File** menu, point to **New**, and then click **SharePoint Content**.



3. In the **New** dialog box, click the **Page** tab, and then double-click **Master Page** to create a new master page.



4. Click **Design** to show the master page in design view.
You should see the header, the left margin areas, and the content placeholders in the master page.
5. Click **Code** to show the master page in code view.
6. Copy the following code into the master page.
The following code contains:
 - The necessary namespaces that must be referenced in the master page.
 - All the necessary content placeholders of the page that you must provide in the master page.
 - A content placeholder for the page title, and links to CSS and ECMAScript (or Microsoft JScript or JavaScript) files that run on the server.


```

<!-- Identifies this page as a .master page written in Microsoft Visual C# and
registers tag prefixes, namespaces, assemblies, and controls. -->
<%@ Master language="C#" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<%@ Import Namespace="Microsoft.SharePoint" %>
<%@ Register Tagprefix="SPSWC"
Namespace="Microsoft.SharePoint.Portal.WebControls"
Assembly="Microsoft.SharePoint.Portal, Version=12.0.0.0, Culture=neutral,
PublicKeyToken=71e9bce111e9429c" %>
<%@ Register Tagprefix="SharePoint"
Namespace="Microsoft.SharePoint.WebControls" Assembly="Microsoft.SharePoint,
Version=12.0.0.0, Culture=neutral, PublicKeyToken=71e9bce111e9429c" %>
<%@ Register Tagprefix="WebPartPages"
Namespace="Microsoft.SharePoint.WebPartPages"
Assembly="Microsoft.SharePoint, Version=12.0.0.0, Culture=neutral,
PublicKeyToken=71e9bce111e9429c" %>
<%@ Register Tagprefix="PublishingWebControls"
Namespace="Microsoft.SharePoint.Publishing.WebControls"
Assembly="Microsoft.SharePoint.Publishing, Version=12.0.0.0, Culture=neutral,
PublicKeyToken=71e9bce111e9429c" %>
<%@ Register Tagprefix="PublishingNavigation"
Namespace="Microsoft.SharePoint.Publishing.Navigation"
Assembly="Microsoft.SharePoint.Publishing, Version=12.0.0.0, Culture=neutral,
PublicKeyToken=71e9bce111e9429c" %>
<%@ Register TagPrefix="wssuc" TagName="Welcome"
src="~/_controltemplates/Welcome.ascx" %>
<%@ Register TagPrefix="wssuc" TagName="DesignModeConsole"
src="~/_controltemplates/DesignModeConsole.ascx" %>
<%@ Register TagPrefix="PublishingVariations" TagName="VariationsLabelMenu"
src="~/_controltemplates/VariationsLabelMenu.ascx" %>
<%@ Register Tagprefix="PublishingConsole" TagName="Console"
src="~/_controltemplates/PublishingConsole.ascx" %>
<%@ Register TagPrefix="PublishingSiteAction" TagName="SiteActionMenu"
src="~/_controltemplates/PublishingActionMenu.ascx" %>
<!-- Uses the Microsoft Office namespace and schema. -->
<html>
  <WebPartPages:SPWebPartManager runat="server"/>
  <SharePoint:RobotsMetaTag runat="server"/>

  <!-- The head section includes a content placeholder for the page title and links to
CSS and ECMAScript (JScript, JavaScript) files that run on the server. -->
  <head runat="server">
    <asp:ContentPlaceHolder runat="server" id="head">
      <title>
        <asp:ContentPlaceHolder id="PlaceholderPageTitle" runat="server" />
      </title>
    </asp:ContentPlaceHolder>

```

```

<Sharepoint:CssLink runat="server"/>
<asp:ContentPlaceHolder id="PlaceHolderAdditionalPageHead" runat="server" />
</head>

<!-- When loading the body of the .master page, SharePoint Server 2007 also
loads the SpBodyOnLoadWrapper class. This class handles .js calls for the master
page. -->
<body onload="javascript:_spBodyOnLoadWrapper();">
  <!-- The SPWebPartManager manages all of the Web part controls, functionality,
and events that occur on a Web page. -->
  <form runat="server" onsubmit="return _spFormOnSubmitWrapper();">
    <wssuc:Welcome id="explitLogout" runat="server"/>
    <PublishingSiteAction:SiteActionMenu runat="server"/>
    <PublishingWebControls:AuthoringContainer id="authoringcontrols"
runat="server">
      <PublishingConsole:Console runat="server" />
    </PublishingWebControls:AuthoringContainer>
    <!-- The PlaceholderMain content placeholder defines where to place the page
content for all the content from the page layout. The page layout can overwrite any
content placeholder from the master page. Example: The PlaceholderLeftNavBar can
overwrite the left navigation bar. -->
    <asp:ContentPlaceHolder id="PlaceholderMain" runat="server" />
    <asp:Panel visible="false" runat="server">
      <!-- These ContentPlaceHolders ensure all default SharePoint Server pages
render with this master page. If the system master page is set to any default master
page, the only content placeholders required are those that are overridden by your
page layouts. -->
      <asp:ContentPlaceHolder id="PlaceholderSearchArea" runat="server"/>
      <asp:ContentPlaceHolder id="PlaceholderTitleBreadcrumb" runat="server"/>
      <asp:ContentPlaceHolder id="PlaceholderPageTitleInTitleArea" runat="server"/>
      <asp:ContentPlaceHolder id="PlaceholderLeftNavBar" runat="server"/>
      <asp:ContentPlaceHolder ID="PlaceholderPageImage" runat="server"/>
      <asp:ContentPlaceHolder ID="PlaceholderBodyLeftBorder" runat="server"/>
      <asp:ContentPlaceHolder ID="PlaceholderNavSpacer" runat="server"/>
      <asp:ContentPlaceHolder ID="PlaceholderTitleLeftBorder" runat="server"/>
      <asp:ContentPlaceHolder ID="PlaceholderTitleAreaSeparator" runat="server"/>
      <asp:ContentPlaceHolder ID="PlaceholderMiniConsole" runat="server"/>
      <asp:ContentPlaceHolder id="PlaceholderCalendarNavigator" runat="server" />
      <asp:ContentPlaceHolder id="PlaceholderLeftActions" runat="server"/>
      <asp:ContentPlaceHolder id="PlaceholderPageDescription" runat="server"/>
      <asp:ContentPlaceHolder id="PlaceholderBodyAreaClass" runat="server"/>
      <asp:ContentPlaceHolder id="PlaceholderTitleAreaClass" runat="server"/>
      <asp:ContentPlaceHolder id="PlaceholderBodyRightMargin" runat="server" />
    </asp:Panel>
  </form>
</body>
</html>

```

Note: You should not delete any of the ContentPlaceHolders because they ensure that all default Office SharePoint Server pages render with this master page. If the system master page is set to any default master page, the only content placeholders that are required are those that are overridden by your page layouts. If a specific content placeholder is not required, instead of deleting, set the **visible** value to False.

7. On the **File** menu, click **Save As**, provide a unique file name with the .master extension, and then save the file to the master page gallery (/_catalogs/masterpage) in your site collection.

3.3.2 Creating a Master Page from the Default Master

These are the three key steps to create a simple master page:

- Open a SharePoint Online site in Office SharePoint Designer 2007.
- Locate and create a copy of the default.master page.
- Apply the new master page to content pages in the site.

3.3.2.1 Opening a SharePoint Online Site in Office SharePoint Designer 2007

The first step for creating a simple master page is to open the SharePoint Online site for which the new master page has to be created.

To open a SharePoint Online site in Office SharePoint Designer 2007

1. Open Office SharePoint Designer 2007.
2. Click the **File** menu, and then click **Open Site**.
The **Open Site** dialog box appears.
3. Enter the URL. The URL will look like <https://contoso1.microsoftonline.com-7.sharepoint.microsoftonline.com/default.aspx> and you can obtain this URL by going to the appropriate SharePoint page and copying the URL from the address bar in your browser.
4. and then click **Open**.

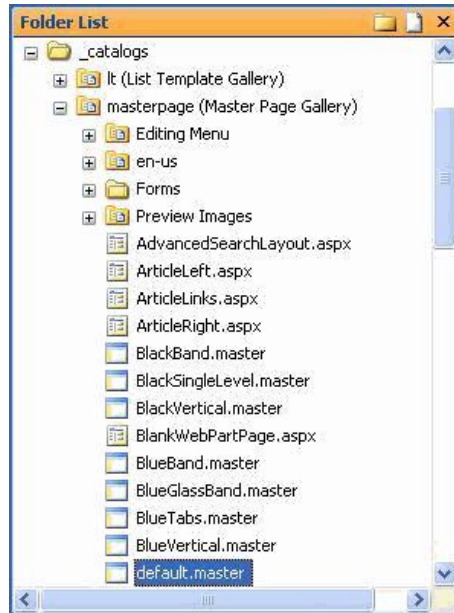
3.3.2.2 Locating and Create a Copy of the default.master Page

By default, a single master page named default.master is applied to all the pages in a SharePoint Online site. Copying default.master and then modifying the copy is a good option for creating a new master page. If this default master page already meets the requirement and only a few changes are required, then the default master page can be used as it is. The default master page is located in the Master Page Gallery.

To create a copy of the default.master master page

1. In Office SharePoint Designer 2007, locate the master page named default.master in the **Folder List** panel.

The default.master master page is located in the masterpage (Master Page Gallery) folder, which is located under the _catalogs folder.



2. In the **Folder List**, right-click **default.master**, and then click **Copy**.
3. Right-click the masterpage (Master Page Gallery) folder, and then click **Paste**.

The new master page is named default_copy(1).master. You can now modify and rename this page.

Note: When a master page is copied, all content pages that are attached to the original master page (in this case, default.master) remain attached to the original master page and not to the copy. After you copy and customize default.master, you can apply the new master page to all of the content pages in the site.

3.3.2.3 Applying the New Master Page

When a new master page is set as the default master page, all of the pages that are attached to the current master page are also attached to the new master page.

To apply the new master page to the site

1. Open the **Folder List** panel in Office SharePoint Designer 2007.
2. Under the _catalogs folder in the **Folder List** panel, go to the masterpage (Master Page Gallery) folder.

3. Right-click the page that will be used as the new master page, and then click **Set as Default Master Page** to set the page as the new Site Master Page.

A message appears, noting that any pages that contain content regions that do not appear in the new default master page will appear broken.

4. If the content regions in the new default master match the content regions in the site's content pages, click **Yes** to continue.

3.4 Customizing Page Layouts

Page layouts are page templates that define how a page should look, what page fields and content fields are available, and exactly which elements should be present on the page (such as lists and libraries).

Publishing in SharePoint Online site automates the workflow process and makes it easy to create new pages in the browser, based on a publishing page. After the page has been created and edited, the content owner submits the page for review. When the reviewer approves the article and publishes it, the page becomes visible to other users on the site.

After you create a page layout, you can fully customize it by using Office SharePoint Designer 2007.

3.4.1 Opening a Page Layout for Editing

All publishing pages in a site are stored in a site-level document library called Pages, where you can identify which page layout was used to create each page.

To open the Pages document library

1. From your sign in tool, click on the My Company Portal and choose a SharePoint site.
2. On the **Site Actions** menu, click **View All Site Content**, and then click the Pages document library.

The publishing feature automatically creates the Pages document library, which contains all of the pages in a site that are created from page layouts. You can look in the **Page Layout** column to see the page layout that was used for a specific page and then click the link to view that page in the browser.

Note: If the Pages document library is not available, on the **Site Actions** menu, click **Site Settings**, and then click **Site Collection Features** (in the top-level site). Then activate **Office SharePoint Server Publishing Infrastructure Feature**.

3. To open the page layout that you want to customize in Office SharePoint Designer 2007, use one of the following methods:

Method 1:

- a. Open your SharePoint Online site in Office SharePoint Designer 2007. The URL will look like <https://contoso1.microsoftonlinecom->

7.sharepoint.microsoftonline.com/default.aspx and you can obtain this URL by going to the appropriate SharePoint page and copying the URL from the address bar in your browser.

- b. Double click the page in the Pages document library.
- c. When prompted, click **Edit Page Layout**.

Method 2:

- a. From your sign in tool, click on the My Company Portal and choose a SharePoint site.
- b. On the **Site Actions** menu, click **Site Settings**.
- c. In **Site Settings** Page, Under **Galleries**, click **Master pages**.
- d. Click the required page layout, and then click **Edit** in Office SharePoint Designer 2007.

3.4.2 Adding Custom Content to a Page Layout

After you open a page layout for editing in Office SharePoint Designer 2007, you can edit it as you would edit any other page that is attached to a master page. When you add custom content to a page layout, the content appears on all the publishing pages that are subsequently created using that page layout.

To add custom content to a page layout

1. With the page layout open in Office SharePoint Designer 2007, click the placeholder in which you want to add content, click the arrow that appears, and then click **Create Custom Content** to unlock the placeholder for editing.



2. Now you can edit this section of the page by using the features that you commonly use to edit pages. For example, you can use any of the following:
 - **Tables:** Click **Insert Table** on the **Table** menu to create tables to align content.
 - **Graphics:** Click and drag images from the Images folder to this section of the page.
 - **Text:** Click and then type where you want the text to appear.
 - **Web Part zone:** Insert a Web Part zone to give visitors of the page the option to add or remove Web Parts such as views of document libraries and lists.

- **Web Parts:** Insert views of document libraries, lists, and more into the page.
3. On the **File** menu, click **Save** to save the page.

3.4.3 Inserting a Content Field

You can display information about content (metadata) that you have inserted into a page layout by using content fields. For example, in a news site, if the page layout includes a field for the body of the article, you can also insert another field that displays the date and time when the body of the article was last modified. This is done by adding a content field to a page layout.

To add a content field to a page layout

1. With the page layout open in Office SharePoint Designer 2007, click the placeholder in which you want to add content, click the arrow that appears, and then click **Create Custom Content** to unlock the placeholder for editing.

Note: When you are creating a table that you want to add to an existing page, it is sometimes easier to create the table on a blank page and then copy the table and paste it into the placeholder on the page where you want to publish it.

By doing this, you can get the format and structure for the information arranged just the way you want it. Then follow the rest of these steps to insert the page fields into the allocated spaces in the table.

2. On the **Task Panes** menu, click **Toolbox**.
3. From the **Content Fields** section, drag the content field that you want to insert into the editable region.

The control is inserted into the content on the page.

Note: If the toolbox is visible but the items in the **Page Fields** section of the toolbox are not visible, you are probably not editing a page layout. **Page Fields** and **Content Fields** are available only for page layouts.

3.5 Customizing Cascading Style Sheets

Cascading style sheets (CSS) provides one of the best ways to update and customize the user interface (UI) of a SharePoint Online site. You can create and deploy your own CSS class by creating a CSS file, through which you can apply styles for images, background, font, and colors. You can define your own set of classes that are different from the default CSS that comes with SharePoint Online site. This also isolates your CSS classes in a separate file, so changes made to the CSS files do not overwrite the changes you made.

In SharePoint Online, to customize the site UI, you create a CSS file and change the link of the CSS file in master pages.

Because there is no access to physical files, you can upload the CSS file to a style library folder in Office SharePoint Designer 2007, or you can use the browser interface to apply a customized style sheet and then use it.

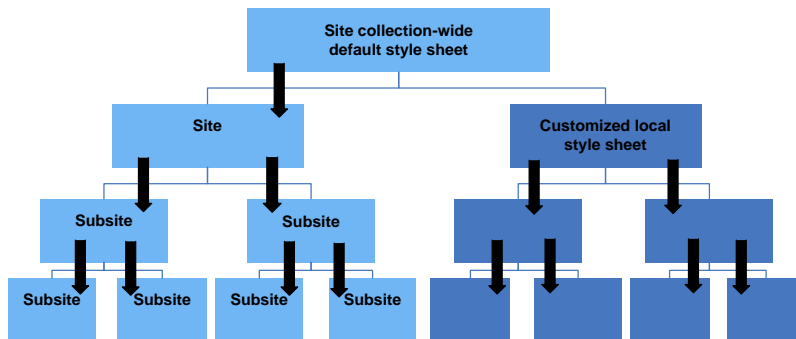
3.5.1 Applying a Custom Style Sheet to a SharePoint Online Site

3.5.1.1 Applying a Style Sheet to a Site and All of Its Subsites

If you are working in a SharePoint Online site environment, you can use the browser interface to apply a customized style sheet to multiple sites in a single operation.

This process takes advantage of a functionality called inheritance. A site can be configured to inherit the styles used by its parent site.

In the following illustration, the second-level site on the right contains a customized local version of the default style sheet. Each of the six subsites in that site collection is configured to inherit its styles from its parent site. Thus, all seven sites in the site collection use the same customized version of the default style sheet. Arrows indicate inheritances.



To apply a customized style sheet to all subsites:

1. In the browser, open the SharePoint Online site in which the customized style sheet resides.
2. On the **Site Actions** menu, do one of the following:
 - Click **Site Settings**.
 - Point to **Site Settings**, and then click **Modify All Site Settings**.
3. On the **Site Settings** page, under **Look and Feel**, click **Master page**.
4. On the **Site Master Page Settings** page, in the **Alternate CSS URL** section, click **Specify a CSS file to be used by this publishing site and all sites that inherit from it**, and then type the full path for the customized style sheet, including the file name, in the box.

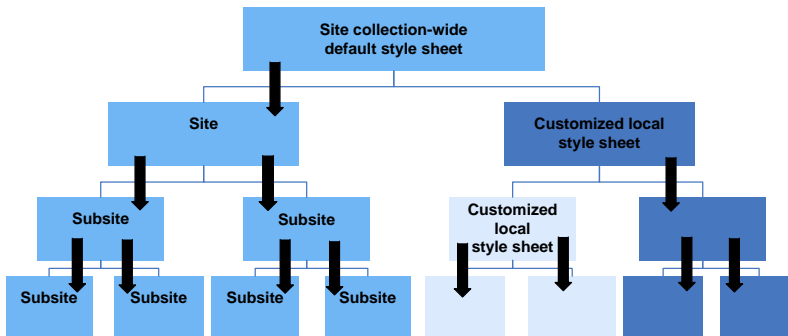
Note: Because the `_styles` folder is a hidden folder and cannot be viewed in the browser, you cannot use the browse (...) button to find and select a customized version of a style sheet file that is stored in a `_styles` folder.

However, you can copy and paste the first part of the path, including the site name, from the **Address** box at the top of the browser. Be sure to stop after the site name, and do not include any external folder or file names within the site. For example, if the path in the **Address** box is `https://SiteName/Subsite_1/_Layouts/ChangeSiteMasterPage.aspx`, copy and paste only the `https://SiteName /Subsite_1/` portion, and then type the rest of the path manually. (The rest of the path is usually `_styles/customized_style_sheet`, where `customized_style_sheet` is the name of the customized style sheet that you want to apply.)

5. Select the **Reset all subsites to inherit this alternate CSS URL** check box, and then click **OK**.

The customized style sheet, which was already applied to the current site, is now also applied to all of its subsites.

In the following illustration, a second customized local version of the default style sheet was added to the site collection on the right and applied to all of the subsites of the site where it resides. Again, arrows indicate inheritance.



3.5.1.2 Configuring a Subsite to Inherit Styles from Its Parent Site

If you are working in a SharePoint Online site environment, you can use the browser interface to configure a single subsite to inherit the styles that are used by the site immediately above it, the parent site.

To configure a single subsite to inherit the styles of its parent site:

1. In the SharePoint Online site, open the sub site to which you want to apply the styles.

2. On the **Site Actions** menu, do one of the following:
 - Click **Site Settings**.
 - Point to **Site Settings**, and then click **Modify All Site Settings**.
3. On the **Site Settings** page, under **Look and Feel**, click **Master Page**.
4. On the **Site Master Page Settings** page, in the **Alternate CSS URL section**, click **Inherit Alternate CSS URL from parent of this site**.
5. Select or clear the **Reset all subsites to inherit this alternate CSS URL** check box, and then click **OK**.

3.5.2 Example: Changing the Background Color of the Page Using CSS Tools in SharePoint Designer 2007

You can use the CSS tools in Office SharePoint Designer 2007 to identify the styles used in the master page, and then you can modify those styles. For example, if you want to change the background color of the page, you should first identify the styles where the background color is defined, and then change those styles to apply the required background color.

To change the background color

1. Open Office SharePoint Designer 2007.
2. On the **File** menu, click **Open Site**.
3. Specify your SharePoint Online site name.
4. Click in the default_copy(1).master file where you want to modify the style.

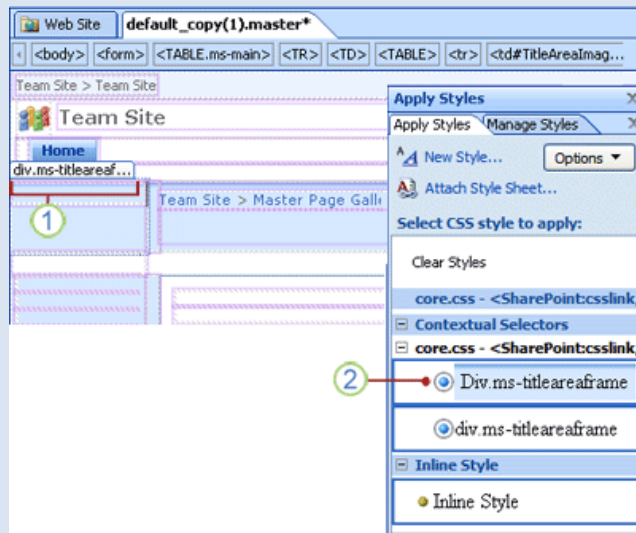
Note: Some content areas that appear in Design view do not have styles applied to them, and therefore no corresponding style appears in the **Apply Styles** task pane. For example, if you select a content placeholder, no style appears in the **Apply Styles** task pane. Content placeholders, however, do appear inside HTML elements such as the <td> tag.

5. On the **Task Panes** menu, click **Apply Styles**.
6. In the **Apply Styles** task pane, click **Options**, and then click **Show Styles Used On Selection**.

Each style applied to the area where the cursor is located is outlined in blue, and a preview of the style appears inside the outline.

In the following illustration, div.ms-titleareafame is selected in Design view. In the **Apply Styles** task pane, three styles that are applied to div.ms-titleareafame appear. The top style, Div.ms-titleareafame, defines the background color.

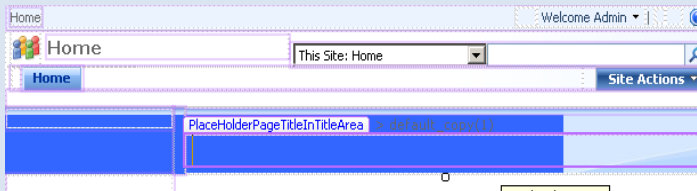
Note: Because styles are case-sensitive, the uppercase Div.ms-titleareafame is different from the lowercase div.ms-titleareafame.



- Div.ms-titlereaf... is selected in Design view.
 - The three styles applied to div.ms-titlereaf... appear in the **Apply Styles** task pane.
7. Right-click the style that you want to modify, and then click **Select All X Instance(s)**, where X is the number of instances in which the style is applied on the page. For this example, right-click the top style, Div.ms-titlereaf...
 8. After you select all instances of the style, right-click the style again, and then click **Modify Style**.
 9. In the **Modify Style** dialog box, make the changes that you want, and then click **OK**.

To follow the example, in the **Modify Style** dialog box, under **Category**, click **Background**. In the **background-color** list, click **Blue**, and then click **OK**.

The new style is applied to the page. Continue modifying the styles until the master page looks the way you want it to appear.



3.6 Managing Site Navigation in SharePoint Online Site

Links for navigation (called navigation items) appear in both the top link bar and on the left side of pages on the Quick Launch. You can customize the SharePoint Online site navigation by adding, removing, or reordering links. You can also hide links to subsites and pages. You can even add links to pages outside the site.

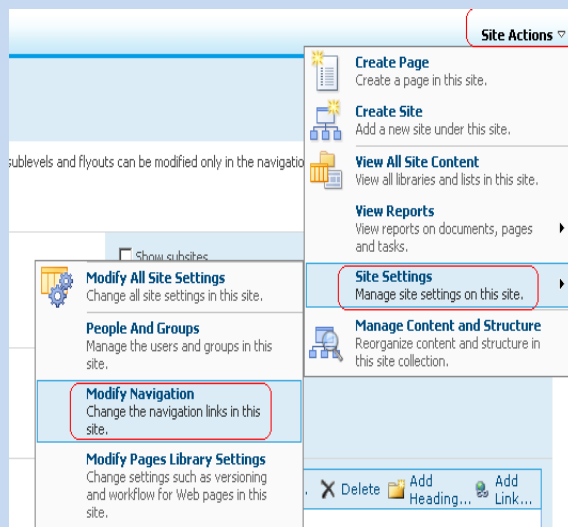
3.6.1 Navigation Editing and Sorting

To customize navigation for a SharePoint Online site, go to the **Site Navigation Settings** page and do one of the following:

1. On the **Site Actions** menu, point to **Site Settings**, and then click **Modify Navigation**.
2. In the **Look and Feel** column, click **Navigation**.

Note: The **Navigation** command appears under **Look and Feel** only if the publishing features are enabled for your site and you have the Full Control or Design permission level.

For non-publishing sites, the **Quick Launch** and **Top link bar** commands appear under **Look and Feel**, instead of **Navigation**.



3.6.1.1 Reordering Navigation Items

You can use the **Move Up** and **Move Down** buttons in the **Navigation Editing and Sorting** section to reorder navigation items and to nest items within headings. The availability and function of the **Move Up** and **Move Down** buttons depend on the following factors:

- How you configured sorting
- The type of item you selected
- The position of the item you selected

The following table shows the behavior of the **Move Up** or **Move Down** buttons when you select the **Sort Manually** option.

Navigation Item	Action	Position of navigation item	Result
Non-heading item	Click Move Up	Not at the top of the local list	The item moves up one position in the local list.
Non-heading item	Click Move Up	Top of the local list	If the item is nested, the item moves above the heading it was nested in, and becomes no longer nested. If the item is not nested, it becomes nested under the heading directly above its original position
Non-heading item	Click Move Down	Not at the bottom of the local list	The item moves down one position in the local list.
Non-heading item	Click Move Down	Bottom of the local list	If the item is nested, the item becomes no longer nested. If the item is not nested, the item becomes nested under the heading directly below its original position.
Heading item	Click Move Up	Not relevant	The item moves up one position. Any items nested within it move with it.
Heading item	Click Move Down	Not relevant	The item moves down one position. Any items nested within it move with it.

The following table shows the behavior of the **Move Up** and **Move Down** buttons when you select the **Sort Automatically** option.

Navigation Item	Action	Result
Non-heading item	Click Move Up	If the item is nested, the item becomes nested in the heading above the heading under which the item is currently nested. If the item is not nested, it becomes nested in the first heading above it.
Non-heading item	Click Move Down	If there is a heading below it, the item becomes nested in the heading below it. If there is no heading below the item, the item becomes no longer nested.

3.6.1.2 Adding, editing, or deleting a heading or link

To add a new heading or link, or to edit an existing heading or link

- In the **Navigation Editing and Sorting** section, do one of the following:
 - To edit a heading or link, click the heading or link, and then click **Edit**.
 - To add a new heading, click **Add Heading**.
 - To add a new link, click **Add Link**.

Note: If you select a heading, the link will be added under that heading. If you select an item, the link will be added to the same level as the selected item.
- In the dialog box that appears, configure the following:
 - Title** box: Type the title of the navigation item as you want it to appear.
 - URL** box: Type the URL for the heading or link. The URL is required for links, but is optional for headings.
If you want to open the URL that you specify to confirm that it is the correct URL, select the **Open link in new window** check box. When you click **OK**, the URL opens in a new window.
 - Description** box: Type a description for the heading or link.
- Click **OK**.

To delete an existing heading or link

- In the **Navigation Editing and Sorting** section, click the link or heading to delete.
- Click **Delete**.

3.6.1.3 Managing navigation item visibility

You can hide navigation items individually in the **Navigation Editing and Sorting** section.

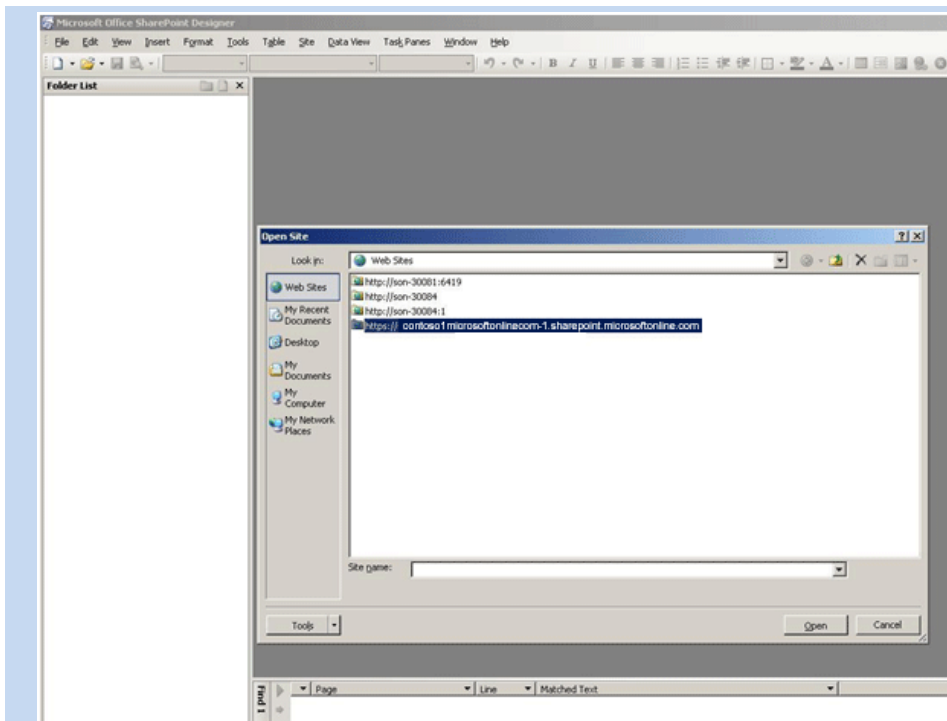
- If you want to hide an item that is currently visible, click the item, and then click **Hide**.
- If you want to show an item that is hidden, click the item, and then click **Show**.

3.7 Example: Designing the Home Page

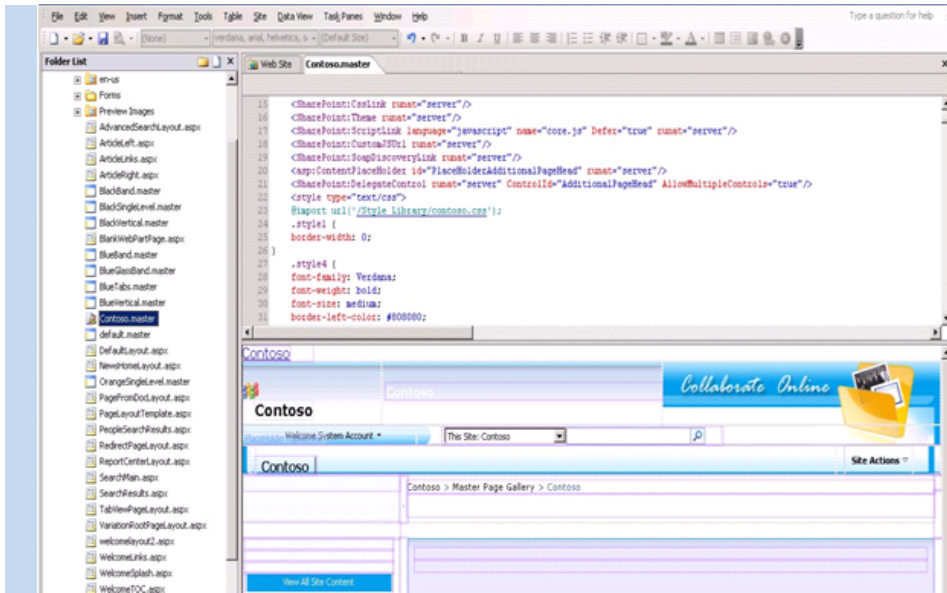
This scenario provides the detailed steps for customizing the home page.

To open the site

1. Go to Office SharePoint Designer 2007 and open your SharePoint Online site.
2. On the **File** menu, click **Open site**.
The **Open Site** dialog box appears.



3. Enter the URL of your site in the **Site Name** box, and then click **Open**.
4. Open the `_catalogs` folder.
5. Open the Masterpage folder, and then click the `default.master` file.
6. Copy `default.master`, and then immediately paste it in the Masterpage folder to create the new file `default_copy(1).master`,
7. Rename the file as `Contoso.master`.
8. Open the custom `Contoso.master` master page.

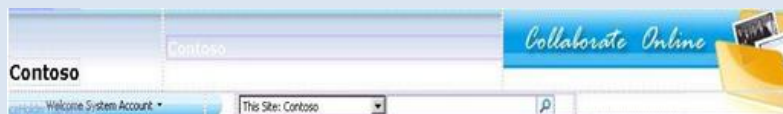


Now, you are ready to customize your master page. The master page is similar to an ASP.NET page, which has HTML tags, content placeholders, and SharePoint controls.

3.7.1 Example: Changing the Banner Image

This scenario explains where and how to change the banner image in the master page.

In the master page you can modify the top banner with table alignment. To do this you need to change the contents inside the **PlaceHolderGlobalNavigation** content placeholder. The `<SharePoint:SiteLogoImage>` tag is responsible for the banner image in the home page. Add the image to the Style Library in Office SharePoint Designer 2007, and set the **LogImageUri** attribute in the `<SharePoint:SiteLogoImage>` tag to the URL of the image.



Code snippet:

```
<SharePoint:SiteLogoImage id="onetidHeadbnnr0" LogImageUri="/Style
```

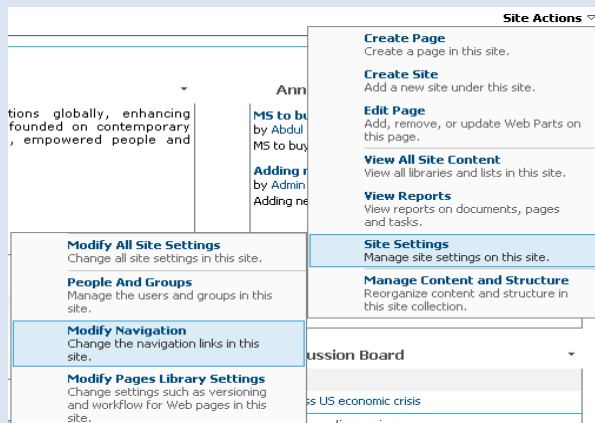
```
Library/images/contosoimages/logo.jpg" runat="server" Visible="true"/>
```

3.7.2 Example: Customizing the Home Page Navigation Bar

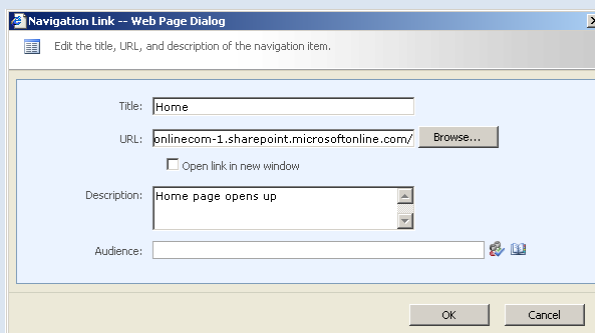
This scenario explains how to add custom links; for example, how to add the **Home** link in the quick launch (the left navigation bar).

To customize the home page navigation bar

1. On the **Site Actions** menu, click **Site Settings**, and then click **Modify Navigation**.



2. Click **Current Navigation**, and then click **Add Link**.
The following dialog box appears.



3. Add the appropriate information to the **Title**, **URL**, and **Description** boxes, and then click **OK**.
4. Click **OK** in the **Current Navigation** page.

3.7.3 Example: Adding Custom CSS to the Master Page

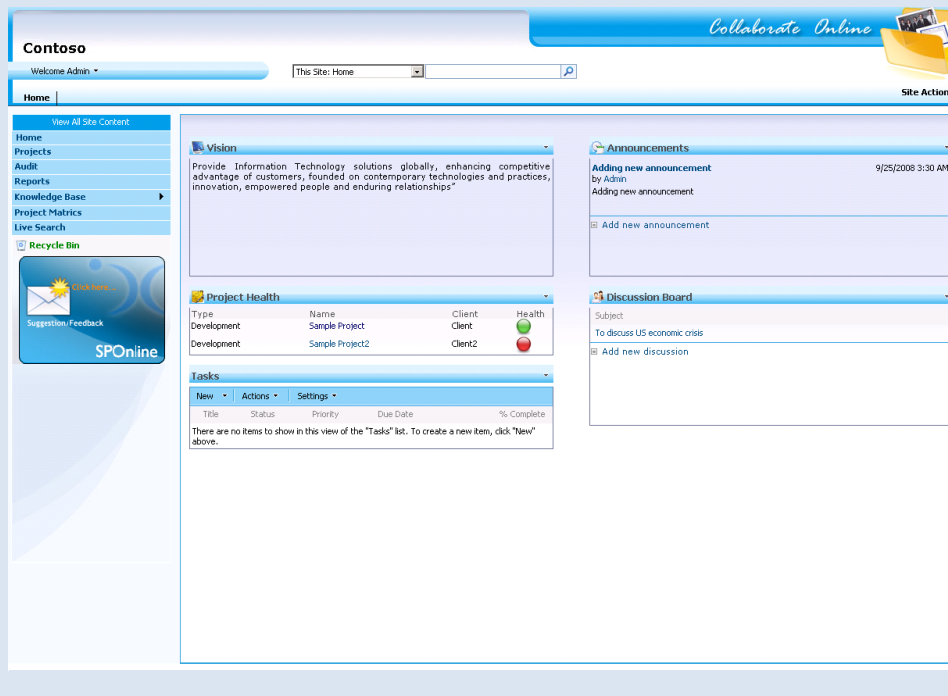
This scenario explains how to add a custom CSS to a master page.

In the master page, you can change the CSS in two ways. You can change either the inline style or the CSS file itself. If you have written a custom CSS, you must attach it to the master page as shown in the following example:

```
<style type="text/css">
  @import url('/Style Library/contoso.css');
</style>
```

Alternatively you can use **SharePoint:CssLink** and **SharePoint:CssRegistration** Web controls.

The customized home page will appear like the following example:



4 Using a Data Form Web Part in SharePoint Online Applications

4.1 About the Data Form Web Part

A Data Form Web Part displays a live and customizable view of a data source. A Data Form Web Part can display data from a wide variety of sources including database queries, XML documents, Web Services, SharePoint, and server-side scripts.

You can create and modify a Data Form Web Part using Microsoft Office SharePoint Designer 2007. With SharePoint Designer, you can create data sources as well to be displayed within a Data Form Web Part. When creating a Data Form Web Part, SharePoint Designer will retrieve the data from the data source as Extensible Markup Language (XML) and display the data by using Extensible Style sheet Language Transformations (XSLT).

Many of us use "Data View" and "Data Form" rather interchangeably. In Windows SharePoint Services v2, we shipped a web part called the DataViewWebPart (DVWP). This web part uses XSLT to transform data from Data Retrieval Services to HTML. In Windows SharePoint Services v3, we shipped a web part called the DataFormWebPart (DFWP). The DataFormWebPart still uses XSLT, but now uses ASP.Net 2.0 Data Source Controls for data access. In addition to "view" functionality, the DataFormWebPart also introduced "form" functionality to write back to various data sources. We use Data View generically to refer to the feature set and all of SharePoint Designer's UI still uses the term Data View.

4.1.1 Creating a Data Form Web Part Using SharePoint Designer

To create a Data Form Web Part, you select a data source and then use a Data View to present the data.

To create a Data Form Web Part

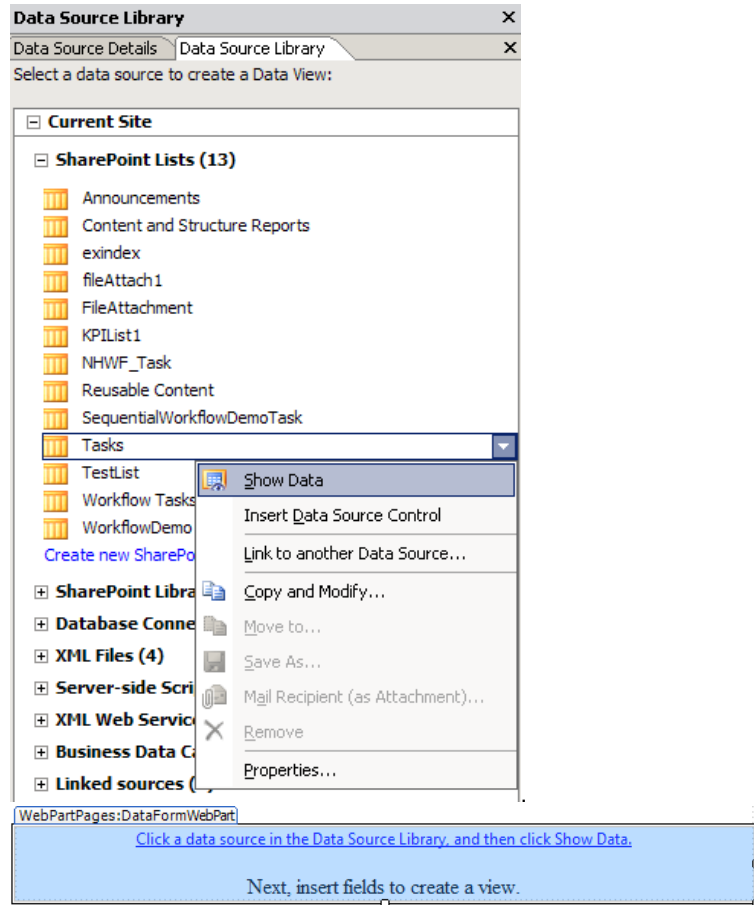
1. In Office SharePoint Designer 2007, click the **File** menu, and then click **New**.
2. In the **New** dialog box, double-click **ASPX**.
3. Click **OK**

A new page with a <form> tag opens.



4. Click in the form page.
5. On the Data View menu, click **Insert Data View**.

An empty Data View is displayed on the page, and the **Data Source Library** task pane opens



6. In the **Data Source Library (It may be any document library or list)** task pane, locate and click the data source, and then click **Show Data**.

Note: With Office SharePoint Designer 2007, you can connect to different data sources like SharePoint Lists and libraries, databases, XML files, XML Web Services etc. The Data Source Library pane in the Office SharePoint Designer 2007, shown above, lists all the different types of data sources available in SharePoint Online.

By default, every Microsoft SharePoint list and library in a site has a corresponding data source connection in the Data Source Library.

7. In the **Data Source Details** task pane, click the fields that you want to insert in the Data View.
8. Click **Insert Selected Fields as**, and then click one of the following:
 - **Multiple Item View:** Displays multiple records in columns with headings in the Data View.
 - **Single Item View:** Displays a single record in the Data View.

Note: **Single Item View** displays a single record in the Data View. **Multiple Item View** displays multiple records, with the fields in columns, and with headings for each column.

The fields that you selected in the **Data Source Details** task pane now appear in a table.

After you have created a Data View, you can:

- Sort and group the data in the Data View. For details, see section 4.3. *Sorting and Grouping in a Data View*.
- Apply formatting to the Data View, when the data in the Data View meets certain conditions. For more details, see section 4.4, *Applying Conditional Formatting to a Data View*

Note: We can modify a Data Form Web Part using **CAML** or **XSLT**, through Office SharePoint Designer 2007

4.2 Examples: Adding, Editing, Deleting, Sorting, and Filtering List Items Using the Data Form Web Part

This scenario explains how to add, edit, and delete project list items. The list stores the details about the project.

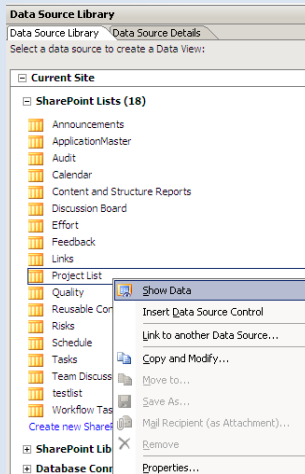
To add, edit, delete, sort, and filter list items using the Data Form Web Part

1. Create a custom list named **Project list** with the following columns, if already not available:
 - Project Name (single line of text)

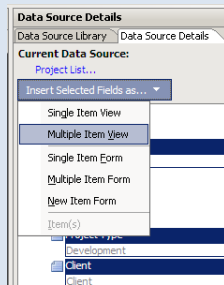
- Project Type (Single line of text)
- Client (Single line of text)

Note: Make all three columns as mandatory columns.

2. Open your SharePoint Online site in Office SharePoint Designer 2007.
3. Open the page where you want to add the Data View Web Part.
4. Click **Insert**, point to **SharePoint controls**, and then click **Data View**.
5. On the **Data Source Library** tab, expand **SharePoint Lists**.
6. In **SharePoint Lists**, click **Project List**, and then click **Show Data**.



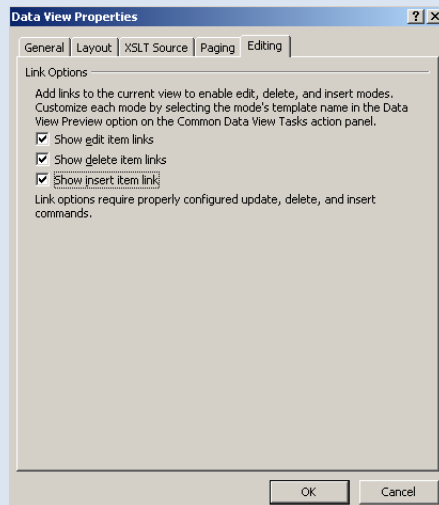
7. Using the CTRL key, select each of the columns you need to show in your Data Form Web Part.
8. Click **Insert Selected Fields as**, and then click **Multiple Item View**.



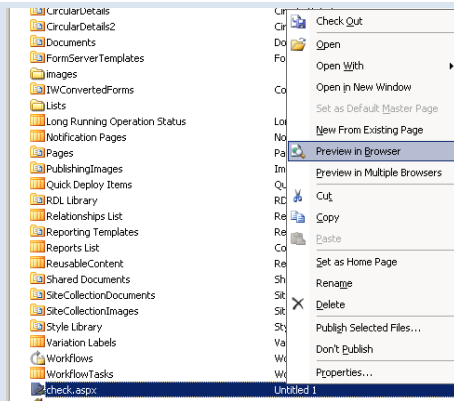
9. Click the arrow at the top right corner of the Data Form Web Part. The **Common Data View Tasks** pane appears.



10. Click the **Paging** task.
The **Data View Properties** dialog box appears.

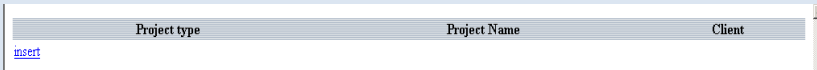


11. Click the **Editing** tab, and then select the following check boxes:
 - **Show edit item links**
 - **Show delete item links**
 - **Show insert item link**
12. Click **OK**.
13. Go to the page that you added, right-click the page, and then click **Preview in Browser**.

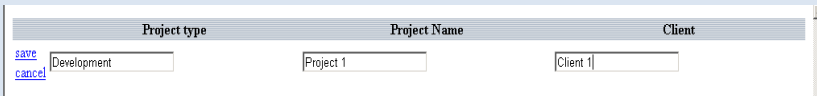


14. Enter the appropriate credentials.

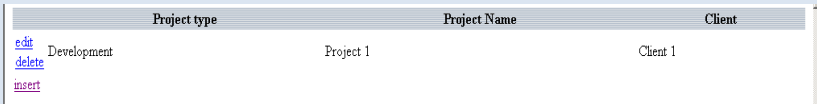
15. Click **insert** to add the item.



16. Enter the values in all the fields, and then click **save** to save the item.



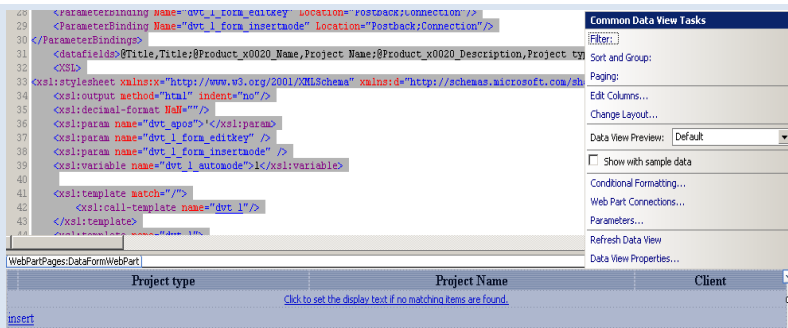
The following screen appears:



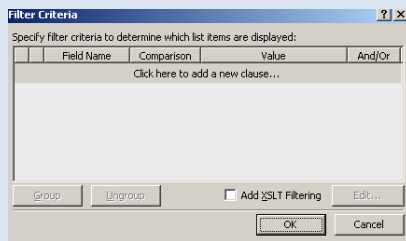
17. Do one of the following:

- Click **edit** to edit the added item.
- Click **delete** to delete an item.
- Click **insert** to create a new item.

18. Go to Office SharePoint Designer 2007, and click the arrow at the top right corner of the Data View Web Part.

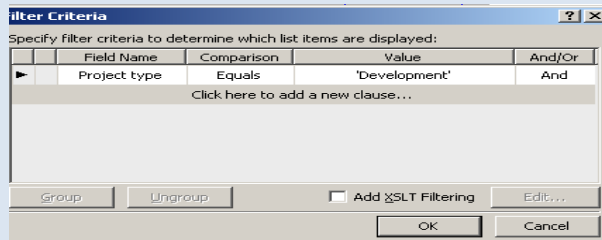


19. Click the **Filter** task.



20. In the **Filter Criteria** pane, click **Click here to add a new clause**.

21. In the **Field Name** column, select the required field to filter the data: for example, **Project type**.



22. From the next columns, select the required comparison and the corresponding value.

Note: To add more conditions, click **And** or **Or** from the **And/OR** column, according to your requirements. This example shows only one condition.

23. Click **OK**.

Only filtered items are displayed in the Data Form Web Part.

24. Click the arrow at the right top corner of the Data Form Web Part.

25. Click the **Sort and Group** task.
26. Under **Available fields**, select the column name according to which you want to sort; for example, select **Client**.
27. Select the order of sorting by selecting **ascending** or **descending** according to your requirements.

Note: You can add as many fields as required.

28. Click **OK**.

The sorted items will appear in the Data View Web Part.

4.3 Sorting and Grouping Data in a Data View

You can add a toolbar to a Data View so that the users can filter, sort, or group data in the Data View by using the toolbar in the browser. You can also turn on filtering and sorting on column headers in a Data View, so that the users can filter or sort the data by clicking the column headers, as in a list view.

To add the toolbar to a Data View

1. Open the page that contains the Data View.
2. Right-click the Data View, and then click **Show Common Control Tasks**.

Tip: You can also click the Data View and then click the arrow in the upper right corner to show the **Common Data View Tasks** list. Click the arrow again to hide the list.

3. Click **Data View Properties**.
The **Data View Properties** dialog box opens.
4. On the **General** tab, do one of the following:
 - To add the toolbar, select the **Show toolbar with options for** check box, and then select the required check boxes below it.
 - To remove the toolbar, clear the **Show toolbar with options for** check box.
 - To enable sorting and grouping in column headers, select the **Enable sorting and filtering on column headers** check box. (It is applicable only to the Data View with basic table layout.)

4.4 Applying Conditional Formatting to a Data View

You can use conditional formatting to easily create a Data View that applies specific styles to data values when the data meets the criteria that you specify. You can also set conditions that change the visibility of a data value, so you can show or hide data.

To apply conditional formatting to a Data View

1. Right-click the Data View, and then click **Conditional formatting**.
The **Conditional Formatting** task pane opens.

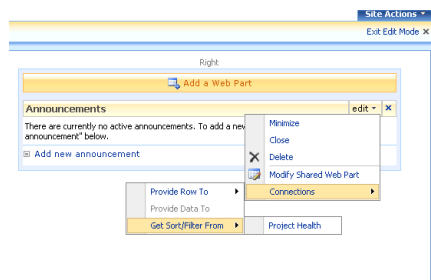
2. In the **Data View** list, click the items in the fields that contain the data you want to format.
3. Click **Create**, and then click **Apply Formatting**.
The **Condition Criteria** dialog box opens to enable you to create a condition.
4. In the **Condition Criteria** dialog box, place your mouse pointer anywhere in the first row, and click.
5. Under **Field name**, click the arrow, and then click the field that you want to use in the condition.
6. Under **Comparison**, click the arrow, and then click the operator that you want to use in the condition.
7. Under **Value**, click the arrow, and then click **More Fields**.
8. In the **More Fields** dialog box, click the field that you want to use to complete the condition.
9. Click **OK**.
10. In the **Condition Criteria** dialog box, click **OK**.
11. In the **Modify Style** dialog box, select the options required to create the style for your conditional formatting.
12. Click **OK**.

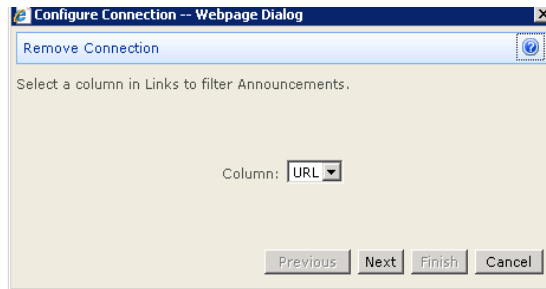
4.5 Connecting Form Web Part to another Web part

To use the Form Web Part, you connect it with another Web Part that is capable of filtering through a Web Part connection, such as the List View . You can connect the Form Web Part to one or more Web Parts on the Web Part page.

To connect the Form Web Part to another Web Part

1. If you are not already in edit mode, on the **Site Actions** menu, click **Edit Page**.
2. On the **Web Part** menu, point to **Connections**, point to **Provide Form Value To**, and then click the name of the Web Part to which you want to link.





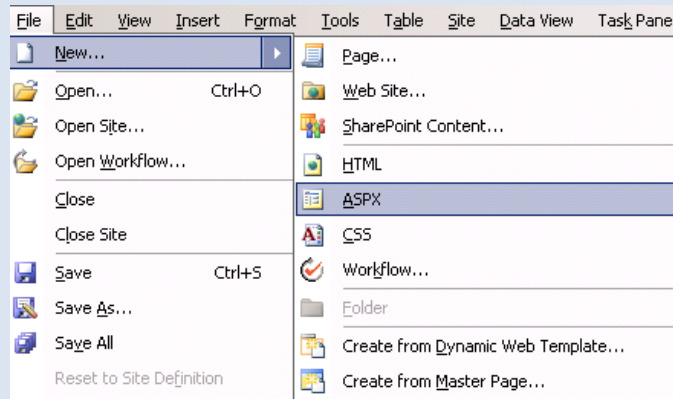
3. In the above Dialog Box, select a column.
4. Click **Next** then click **Finish**.

4.6 Example: Integrating Microsoft Live Search Web Service Using the Form Web Part

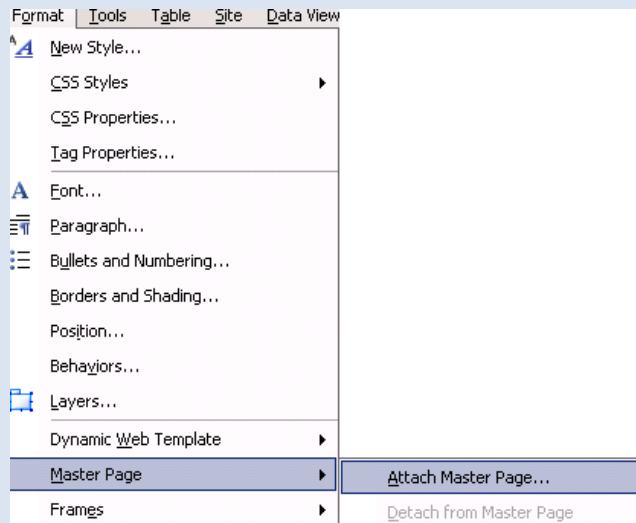
In this scenario you learn how to connect third-party Web services using the FormWeb Part. The first task is to define the Live Search data source. The second task is to integrate Live Search data source to a Form Web Part in a SharePoint Online page.

Initial Steps: Creating SharePoint Online application page.

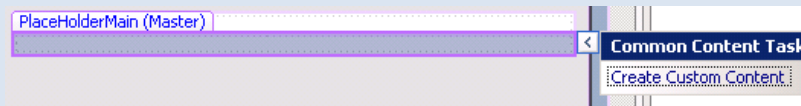
1. Open Office SharePoint Designer 2007.
2. On the **File** menu, click **Open Site**.
3. Enter the site URL. The URL will look like <https://contoso1.microsoftonline.com-7.sharepoint.microsoftonline.com/default.aspx> and you can obtain this URL by going to the appropriate SharePoint page and copying the URL from the address bar in your browser.
4. In Designer, Click on **File**, point to **New**, click on **ASPX** .



5. To attach the page to master page, click on **Format**, point to **Master Page** then click **Attach Master Page**.

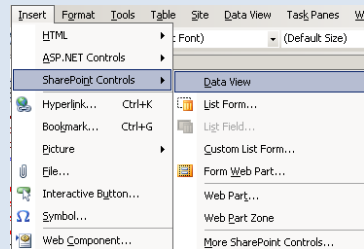


6. On the page click on **PlaceholderMain**, Click on the arrow at right and click on **Create Custom Content**.



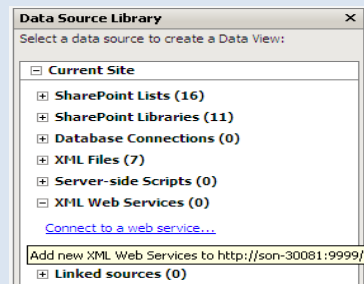
To define the Live Search data source

1. On the **Insert** menu, point to **SharePoint Controls**, and then click **Data view**.



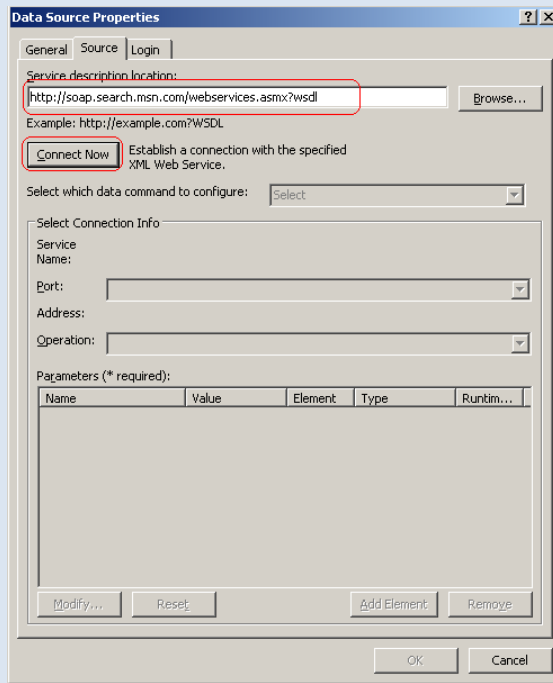
The **Data Source Library** task pane appears on the right.

2. Click to expand **XML Web Services**, and then click **Connect to a web service**.

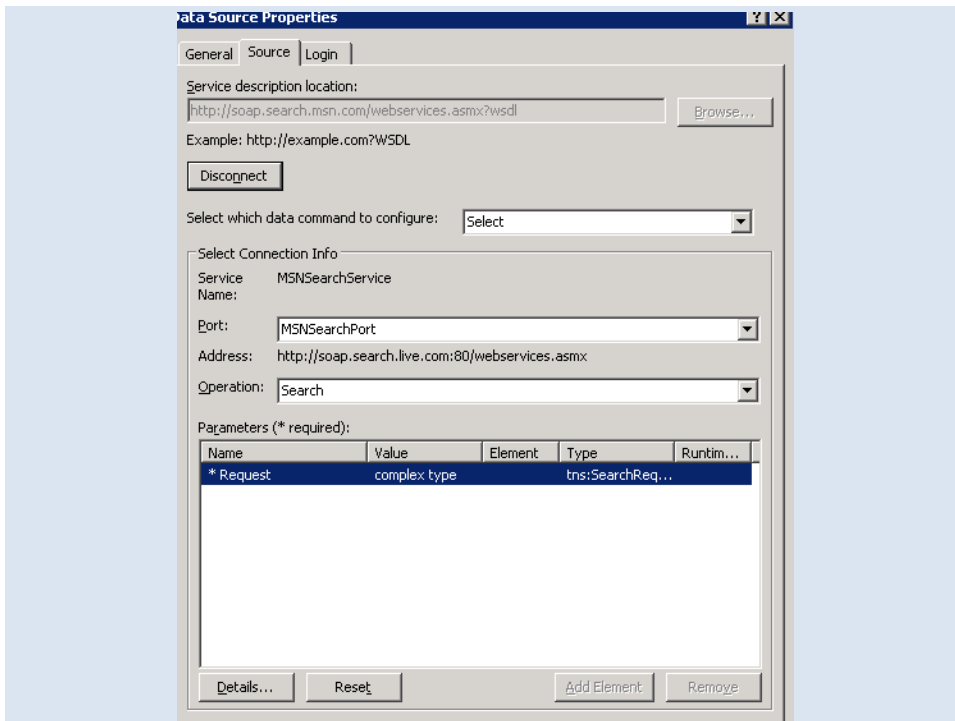


The **Data Source Properties** dialog box appears.

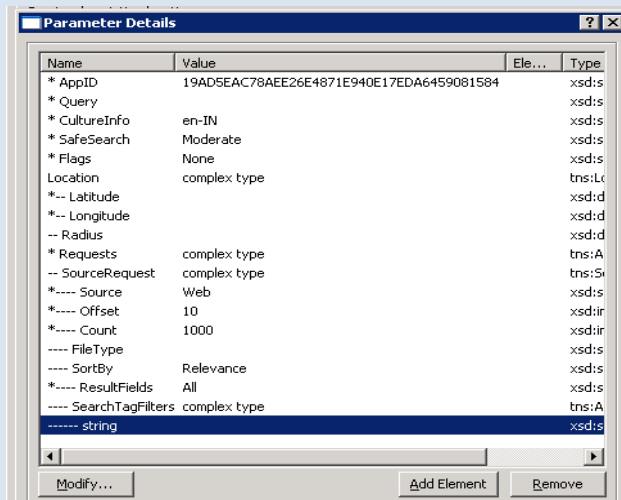
3. On the **General** tab, enter the name for the service, its description, and the details.
4. On the **Source** tab, in the **Service description location** box, enter the URL of the specified Web Service; this example uses <http://soap.search.msn.com/webservices.asmx?wsdl>.



5. Click **Connect Now**.
6. Under **Select Connection Info**, in the **Operation** list, click **Search**.

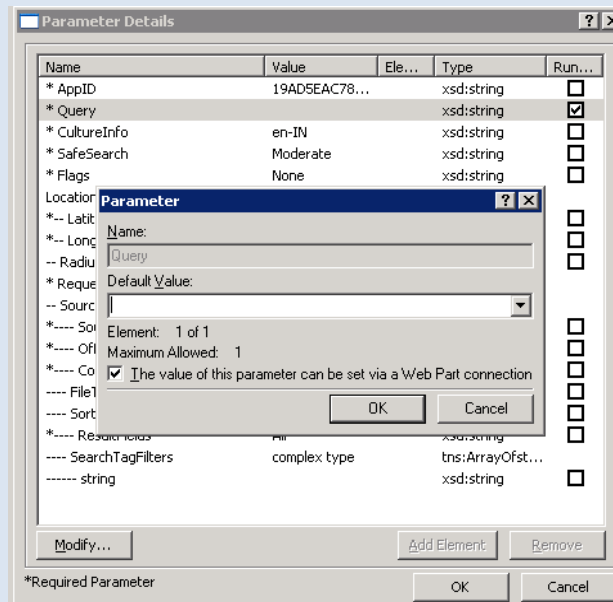


7. Under **Parameters (* required)**, in the **Name** column, double-click *** Request**. The **Parameter Details** dialog box appears.
8. In the **Value** column, add the parameter values that are shown here:



9. For each required parameter (marked with *), double-click the parameter.
10. In the **Parameter** dialog box that appears for each required parameter, click **OK**.

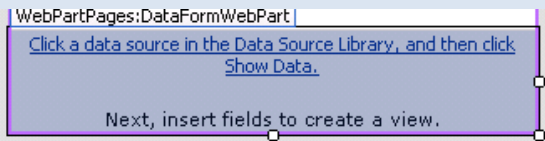
Note: For the **Query** parameter, you must select the check box for the value of this parameter that can be sent via a Web Part connection.



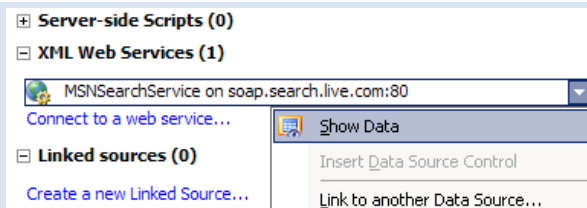
11. Click **OK** in the **Parameter Details** dialog box.
12. Click the **Login** tab in Data Source Properties, and then select the radio button, **Don't attempt to authenticate**.
13. Click **OK**.

To connect Form Web Part with Live Search Data Source.

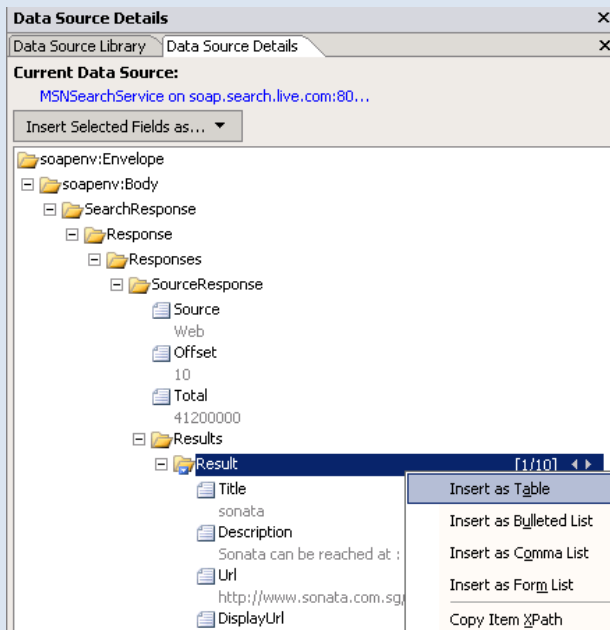
1. Click on the newly added Form Web part in Split view



2. In Data Source Library under XML Web Services, click on Live Search Data Source and click **Show Data**.

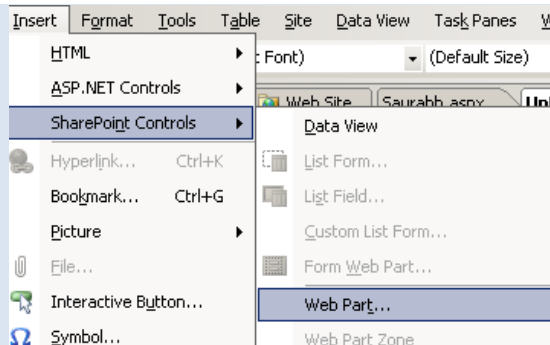


3. In the Data Source Details Pane, right click on **Result** and click on **Insert as Table**

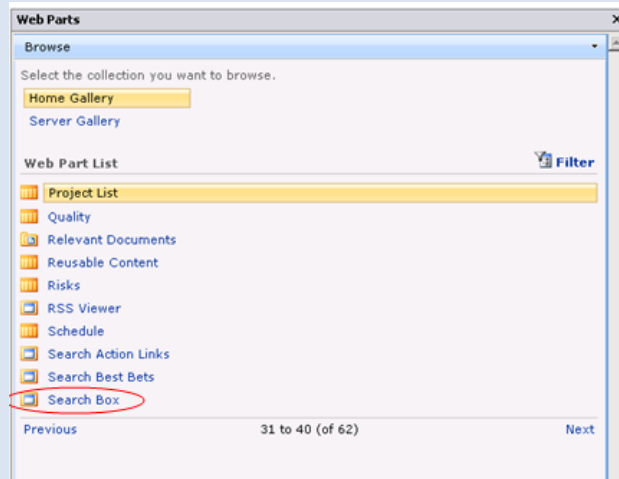


To Add Search Box to the page

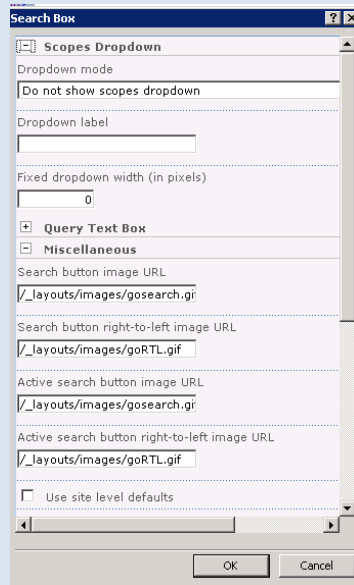
1. In menu, click on **Insert**, Point to **SharePoint Controls** and click on **Web Part**.



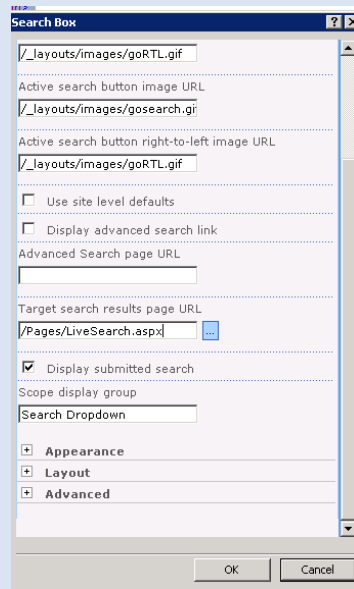
2. From the right pane, add the SharePoint Search Box Web Part to the page, to enter the search query.



3. Right-click **Search Box**, and then click **Web Part Property**.
4. In the **Search Box** pane, click **Scopes Dropdown**.
5. In the **Dropdown mode** box, select **Do not show scopes dropdown**.

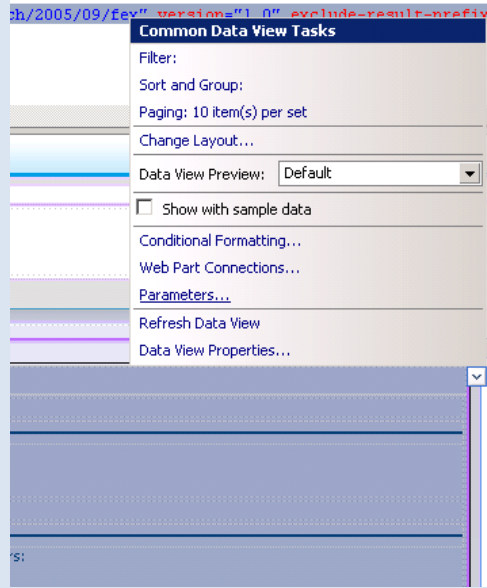


6. In the middle of the **Search Box** pane, click **Miscellaneous**.
7. In the **Target search results page URL** box, use the browse (...) button to enter the current page.



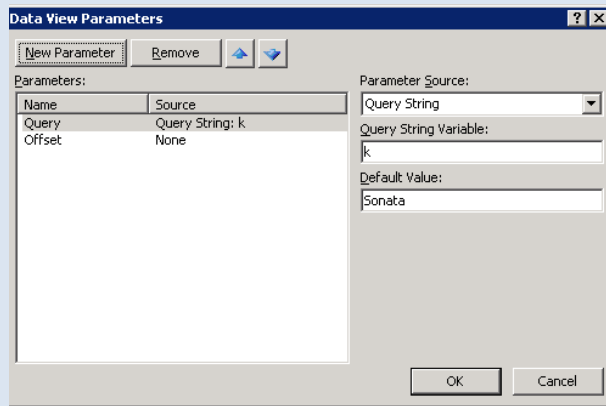
To integrate the search box with the Form Web Part

1. In the Form Web Part, click on arrow at right, top corner.
2. In the **Common Data View Tasks** pane, click **Parameters**.



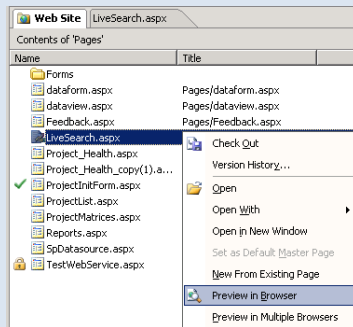
The **Data View Parameters** dialog box appears.

- 3. In the **Parameter Source** list, click **Query String**
- 4. In the **Query String Variable** list, enter **k**.

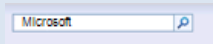


- 5. Click **OK**.

6. Save the current page.
7. In the Folder list, right click the page which we have created now.
8. In the menu that appears, click **Preview in browser**.



9. Enter the credentials.
10. Enter the word to search.



11. Click the search icon  to get the search results.

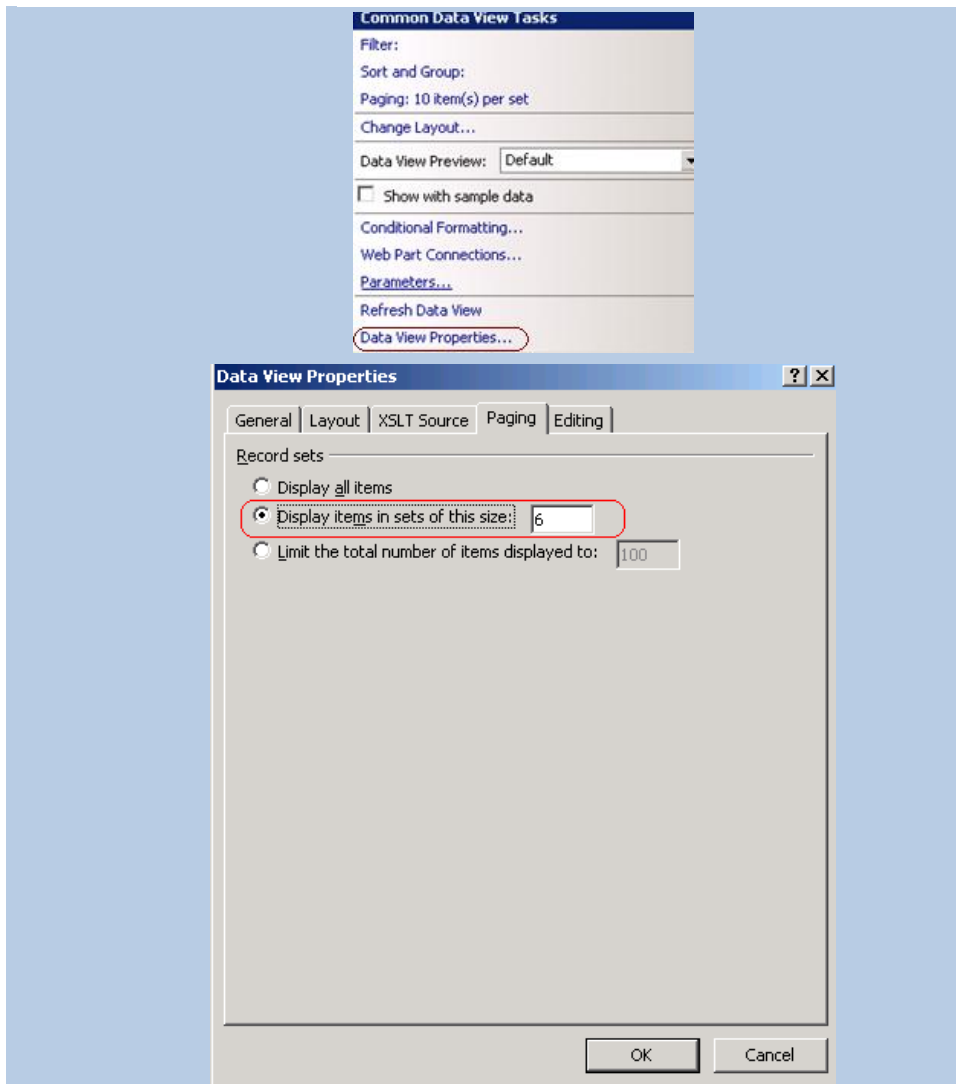
4.7 Example: Adding Page Numbers to a Data Form Web Part

This example shows how to modify the Data Form Web Part using XSLT.

XSLT can be used for modifying DVWP. This scenario illustrates how to bring paging in DVWP. Normally you will get a **next** link that you need to click to go to the next page. By doing this modification we can specifically click on individual page numbers to go to a required page.

The solution makes use of the fact that the navigation links that are created by default pass a parameter that indicates what row number to start rendering from. By looping through the row data and checking the current position, you can simply output a page number, as shown here: $((position() - 1) \div \$RowLimit) + 1$. Every time you get to a row number, that becomes the start of the next page: $position() \bmod \$RowLimit$

This solution assumes you have created a Data View. Click the Data View Properties, go to the Paging tab and enter the size in the text box against "Display items in sets of this size" through Office Sharepoint Designer 2007. (Please refer the two images below)



To add page numbers to a Data View Web Part

1. In Office SharePoint Designer 2007, view the HTML (XSLT) source of your Data View.
2. Find `<xsl:template name="dvt_1.navigation">` and add `<xsl:param name="Rows"/>` as the last parameter.

3. Somewhere within the <table> tag, insert the following code:

```
<tr><td colspan="2" class="ms-vb">
  <xsl:for-each select="$Rows">
    <xsl:if test="(position() mod $RowLimit) = 1">
      <a>
        <xsl:attribute name="href">javascript: <xsl:value-of
xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
select="ddwrt:GenFireServerEvent(concat('dvt_firstrow={',position(),'})')"/></xsl:attribute>
        <xsl:value-of select="((position() - 1) div $RowLimit) + 1"/></a>
        <xsl:text
xmlns:ddwrt="http://schemas.microsoft.com/WebParts/v2/DataView/runtime"
ddwrt:nbsp-preserve="yes" disable-output-escaping="yes">&nbsp;</xsl:text>
      </xsl:if>
    </xsl:for-each>
  </td></tr>
```

4. Find <xsl:call-template name="dvt_1.navigation"> and add <xsl:with-param name="Rows" select="\$Rows"/> as the last parameter.

Note: This should be enough to get you started. You could add an <xsl:if test> statement to prevent the current page number from being created as a link.

5 Using SharePoint Online Web Services

The Microsoft Office SharePoint Online Web Services are a subset of the Windows SharePoint Services 3.0 Web Services APIs. These APIs provide methods that allow you to access SharePoint Online data from Windows applications, customer applications, and other instances of SharePoint. In SharePoint Online applications, you must use the Data Form Web Part to connect to any Web service.

5.1 List of Available SharePoint Online Web Services

The following table lists each of the Web services that are available in SharePoint Online, along with its path, a description, and a link to further information in MSDN.

Web Service	Path from <site>	Description	MSDN Link
Alerts	/_vti_bin/alerts.aspx	Provides methods for working with alerts for list items in a SharePoint Online site	Alerts Web service
Copy Web Service	/_vti_bin/Copy.aspx	Provides methods for copying items between locations in the SharePoint environment Add files to a Distribution List (DL), copy files from one DL to another, download files from a DL	Copy Web service
Document Workspace	/_vti_bin/DWS.aspx	Provides methods for managing Document Workspace sites and the data they contain	Document Workspace Web service
Note: The following method of Document Workspace service is not available in SharePoint Online <ul style="list-style-type: none"> • FindDwsDoc 			
Imaging	/_vti_bin/Imaging.aspx	Provides methods that enable you to create and manage	Imaging Web service

Web Service	Path from <site>	Description	MSDN Link
		picture libraries	
Lists	/_vti_bin/Lists.asmx	Provides methods for working with lists and list data	Lists Web service
Note: The following method of Lists service is not available in SharePoint Online <ul style="list-style-type: none"> • AddDiscussionBoardItem 			
Meetings	/_vti_bin/Meetings.asmx	Provides methods that enable you to create and manage Meeting Workspace sites	Meetings Web service
People	/_vti_bin/People.asmx	Provides methods for working with security groups	People Web service
Permissions	/_vti_bin/Permissions.asmx	Provides methods for working with the permissions for a site or list	Permissions Web service
Site Data	/_vti_bin/SiteData.asmx	Provides methods that return metadata or list data from sites or lists in Windows SharePoint Services	Site Data Web service
Sites	/_vti_bin/sites.asmx	Provides methods for returning information about the site templates for a site collection	Sites Web service
Search	/_vti_bin/spsearch.asmx	Provides methods for remotely performing searches within a Windows SharePoint Services deployment	Search Web service
Users and Groups	/_vti_bin/UserGroup.asmx	Provides methods for working with users, site groups, and cross-site groups	Users and Groups Web service

Web Service	Path from <site>	Description	MSDN Link
Versions	/_vti_bin/versions.asmx	Provides methods for working with file versions	Versions Web service
Views	/_vti_bin/Views.asmx	Provides methods for working with views of lists in Windows SharePoint Services	Views Web service
Web Part Pages	/_vti_bin/webpartpages.asmx	Provides methods for working with Web Parts	Web Part Pages Web service
<p>Note: The following methods of Web Part Pages service are not available on SharePoint Online</p> <ul style="list-style-type: none"> • AssociateWorkflowMarkup • ExecuteProxyUpdates • GetAssemblyMetaData • GetDataFromDataSourceControl • GetFormCapabilityFromDataSourceControl • RemoveWorkflowAssociation • ValidateWorkflowMarkupAndCreateSupportObjects 			
Webs	/_vti_bin/Webs.asmx	Provides methods for working with sites and subsites	Webs Web service
<p>Note: The following methods of Webs service are not available in SharePoint Online</p> <ul style="list-style-type: none"> • CustomizeCss 			
Publishing Service	/_vti_bin/PublishingService.asmx	Provides methods to work remotely with the publishing service	Publishing Service Web service
<p>Note:</p> <ol style="list-style-type: none"> 1. The following methods of Publishing service are not available on SharePoint Online <ul style="list-style-type: none"> • ExportObjects • GetObjectStatusCollection • GetObjectStatusCollectionWithExclusions • ImportObjects 			

Web Service	Path from <site>	Description	MSDN Link
2. The publishing service does not have link in the MSDN and hence no URL is provided in the MSDN Link column			

5.2 List of Unsupported SharePoint Online Web Services

The following table provides the list of each excluded SharePoint Online Web Services, with its path, description, and MSDN link.

Web Service	Path from <site>	Description	MSDN Link
Area Web Service	/_vti_bin/areaservice.asmx	Provides an area interface for remote clients	Not applicable
Authentication Web Service	/_vti_bin/Authentication.asmx	Provides ability for Web Services to operate in conjunction with forms authentication	Not applicable
BDC Fields Resolver Web Service	/_vti_bin/bdcfieldsresolver.asmx	Business Data Catalog Fields Resolver Web Service for Clients	Not applicable
BDC Web Service	/_vti_bin/businessdatacatalog.asmx	Business Data Catalog Metadata Web Service	Not applicable
CMS Content Area Toolbox Info Web Service	/_vti_bin/contentAreaToolboxService.asmx	This Web service is designed for the Office SharePoint Designer client.	Not applicable
List Data Retrieval Web Service	/_vti_bin/DspSts.asmx	Provides a method for performing queries against lists in Windows SharePoint Services	Not applicable
Excel Services Web Service	/_vti_bin/ExcelService.asmx	Provides methods to call the Microsoft Office Excel Services remotely or work as a callback for Web Parts	Not applicable
Forms Web Service	/_vti_bin/Forms.asmx	Provides methods for returning forms used in the user interface when working with the contents of a list	Not applicable

Web Service	Path from <site>	Description	MSDN Link
Forms Services Proxy Web Service	/_vti_bin/FormsServiceProxy.aspx	Provides methods for Forms Services to proxy requests to remote data sources from Web-based forms	Not applicable
Forms Services Web Service	/_vti_bin/FormsServices.aspx	Provides methods to call Forms Services remotely or to work as a callback for Web-based forms	Not applicable
Official File Web Service	/_vti_bin/officialfile.aspx	Provides methods for sending files to a records repository	Not applicable
Published Links Web Service	/_vti_bin/publishedlinksservice.aspx	Microsoft Office clients and other applications can obtain the list of published links on the server that are targeted to the current user	Not applicable
Search Web Service	/_vti_bin/search.aspx	Allows access to Enterprise Search results from client applications and Web applications outside of the context of a SharePoint Online site	Not applicable
SharePoint Directory Management Web Service	/_vti_bin/sharepointemailws.aspx	Provides methods for remotely managing distribution groups	Results in error
Slide Library Web Service	/_vti_bin/SlideLibrary.aspx	Provides methods for slide library callbacks or remote publishing of slides	Not Applicable
Spell Check Web Service	/_vti_bin/SpellCheck.aspx	Provides methods for spell checking callbacks	No Results

Web Service	Path from <site>	Description	MSDN Link
Search Crawl Web Service	/_vti_bin/spscrawl.asmx	Provides methods for remote Office SharePoint Server 2007 farms to crawl a local farm	Not applicable
User Profile Change Web Service	/_vti_bin/userprofilechangeservice.asmx	Provides methods to query the user profiles change log remotely	Not applicable
User Profile Web Service	/_vti_bin/userprofileservice.asmx	Provides a user profile interface for remote clients	Not applicable
Admin	/_vti_adm/admin.asmx	Provides methods for managing a deployment of Windows SharePoint Services, such as for creating or deleting sites	Not Applicable
Content Deployment Remote Import	/_vti_adm/ContentDeploymentRemoteImport.asmx	Provides methods for a remote farm to use content deployment to get content into a local farm	Not applicable
Search Web Service	SearchAdmin.asmx	Office SharePoint Server 2007 Search Administration Web service	Not Applicable
Search Application Web Service	/SSP/Search/SearchAdmin.asmx	Office SharePoint Server 2007 Search Application Administration Web service	Not applicable
Excel Service Soap	/SSP/ExcelCalculationServer/ExcelService.asmx	Office SharePoint Server 2007 Excel Services Application Web service	Not applicable

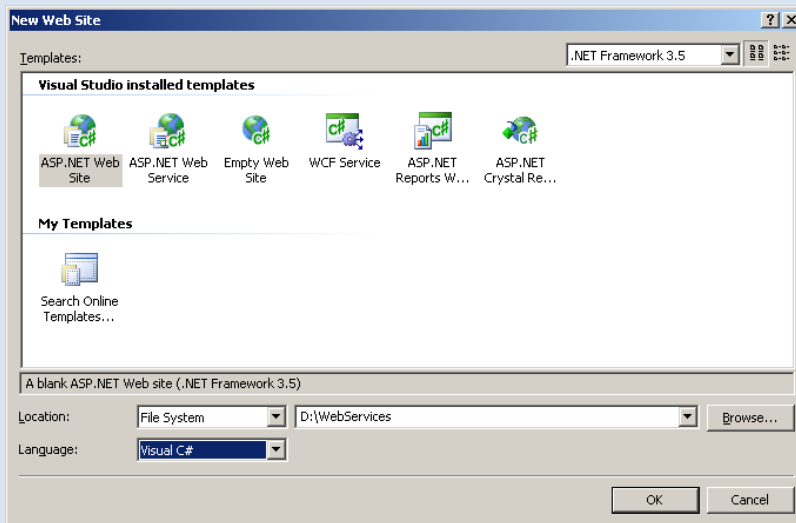
5.3 Example: Displaying All List Names from a SharePoint Online Site Collection

This example shows how managed code can be run on your computer to access data using SharePoint Online Web Services.

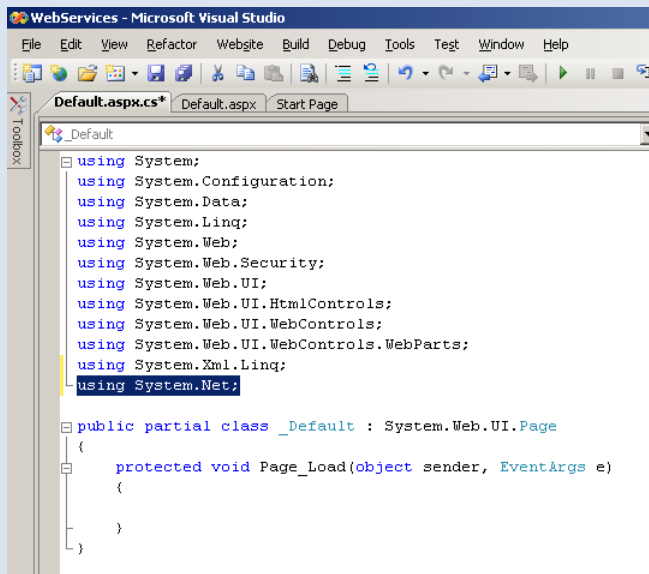
In this example we will see how the Microsoft .NET Framework managed code can access data from a SharePoint Online site. This example creates a custom .aspx page that lists all the list names from the SharePoint Online site.

To display all list names from a SharePoint Online Application

1. Open Microsoft Visual Studio® 2008, click the **File** menu, and then click **New Web Site**.



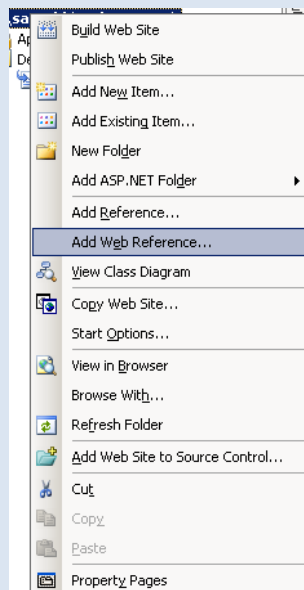
2. Add the namespace **System.Net**.



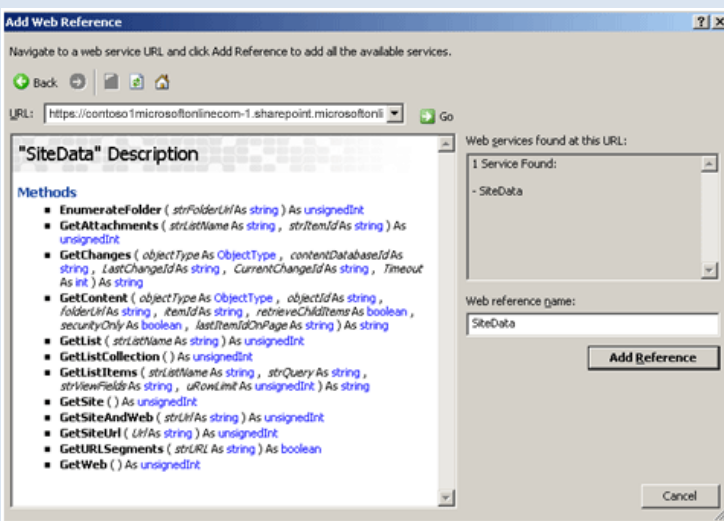
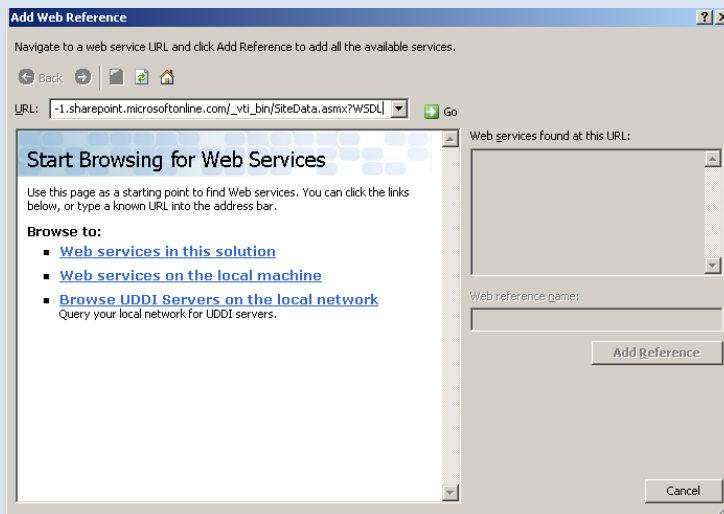
```
using System;
using System.Configuration;
using System.Data;
using System.Linq;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.HtmlControls;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Xml.Linq;
using System.Net;

public partial class _Default : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
    }
}
```

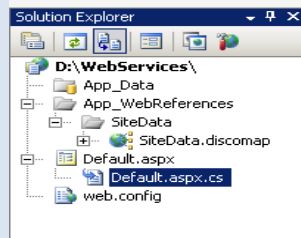
3. In **Solution Explorer**, right-click the solution, and then click **Add the Web Reference**.



4. Click the **Source** tab, and then in the **Service Description Location** box, enter `https://<SPOnlineSiteName>/_vti_bin/lists.asmx?WSDL`, where `<SPOnlineSiteName>` is SharePoint Online sites URL. The URL will look like <https://contoso1microsoftonline.com/7.sharepoint.microsoftonline.com/default.aspx> and you can obtain this URL by going to the appropriate SharePoint page and copying the URL from the address bar in your browser, and then click **Go**.



5. Enter the Web reference name, and then click **Add Reference**.
You will see the added Web Service reference in **Solution Explorer**.



6. In the **Page Load** function add the following code, and enter the user name and password for your SharePoint Online site.

```

SiteData.SiteData site = new SiteData.SiteData();
System.Net.NetworkCredential myCredential = new
System.Net.NetworkCredential();
myCredential.UserName = "<username>";
myCredential.Password = "<password>";

site.Credentials = myCredential;

SiteData._sList[] lists;
site.GetListCollection(out lists);
foreach (SiteData._sList list in lists)
{
    listdisp.InnerHtml += list.Title.ToString() + "<br/>";
}

```

7. Open **Default.aspx**, click the **Source** tab, and then copy the following code into the <DIV> tag.

```

<asp:Label ID="Label1" runat="server" Font-Bold="True" Font-Size="12pt"
    Text="List Collection in SPOne site" Width="233px"
    BorderColor="#404040">
</asp:Label><br/>
<asp:Panel ID="Panel1" runat="server" Height="100%" Width="232px"
    BorderColor="Gray" BorderWidth="2px">
    <span id="listdisp" runat="server" style="position: static"/>
</asp:Panel>

```

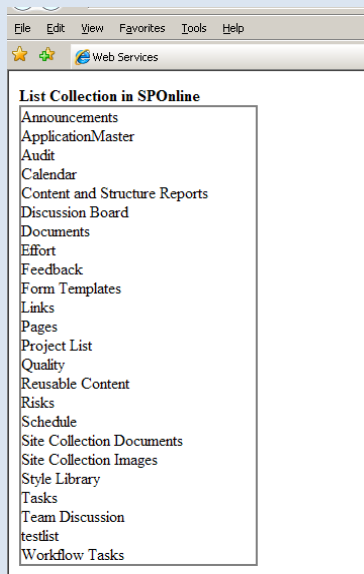
The code should appear as shown in the following figure.



```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Default.aspx.cs" Inherits="_Default" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
<title>Web Services</title>
</head>
<body>
<form id="form1" runat="server">
<div>
<asp:Label ID="Label1" runat="server" Font-Bold="True" Font-Size="12pt"
Text="List Collection in SPOOnline" Width="233px" BorderColor="#404040">
</asp:Label><br/>
<asp:Panel ID="Panel1" runat="server" Height="100px" Width="233px" BorderColor="Gray" BorderWidth="2px">
<asp:ListDisp ID="ListDisp1" runat="server" style="position: static;">
</asp:Panel>
</div>
</form>
</body>
</html>
```

On running the application, the .aspx page should display all the available SharePoint lists of the site, as shown in the following figure.

Note: Results will vary depending on your site and the parameters that you pass.

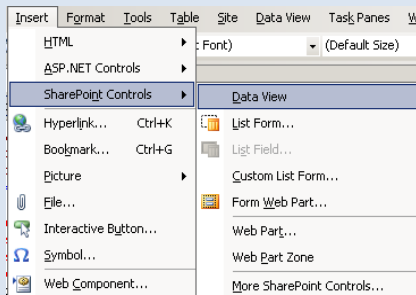


5.4 Example: Fetching List Collection Data from an SharePoint Online Site

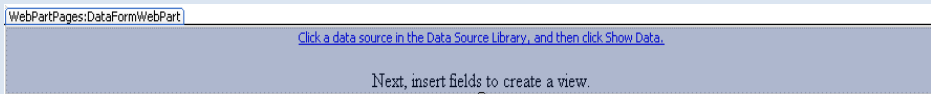
This scenario explains how to fetch list collection data from a SharePoint Online site using the List Web service, and how to render data using DVWP.

To fetch list collection data from a SharePoint Online site

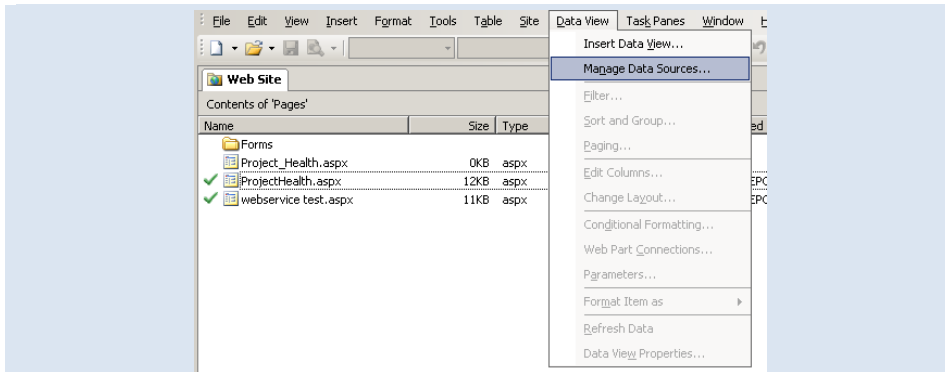
1. Open Office SharePoint Designer 2007.
2. On the **File** menu, click **Open Site**.
3. Give the site URL of your site.
4. Open the page where you want to display the data.
5. Click the placeholder where you want to add the Data View Web Part.
6. On the **Insert** menu, point to **SharePoint Controls**, and then click **Data View**.



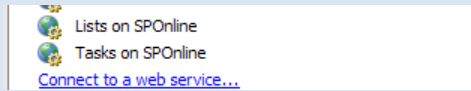
The **Data View** control appears on the page, as shown in the following figure.



7. To set the data source for this Data View, on the **Data View** menu, click **Manage Data Sources**.

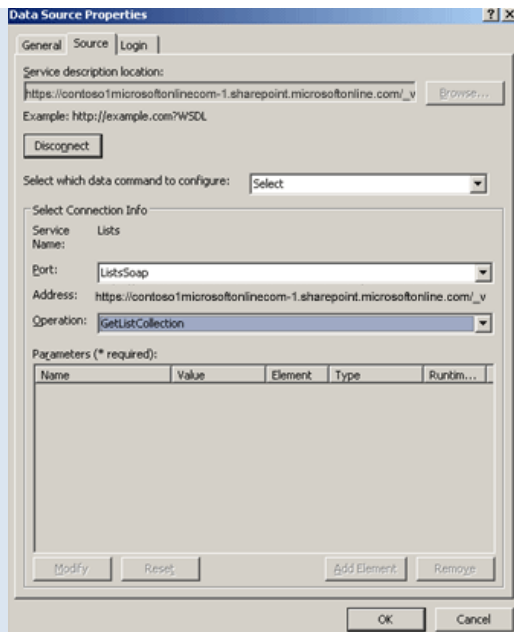


8. In the **Data Source Library** task pane, click **Connect to a web service**.



The **Data Source Properties** dialog box appears.

9. In the **Source** tab, in the **Service Description Location** box, enter `https://<SPOnlineSiteName>/_vti_bin/lists.aspx? WSDL`, where `< SPOnlineSiteName >` is SharePoint Online sites URL. The URL will look like <https://contoso1microsoftonline.com-7.sharepoint.microsoftonline.com/default.aspx> and you can obtain this URL by going to the appropriate SharePoint page and copying the URL from the address bar in your browser.

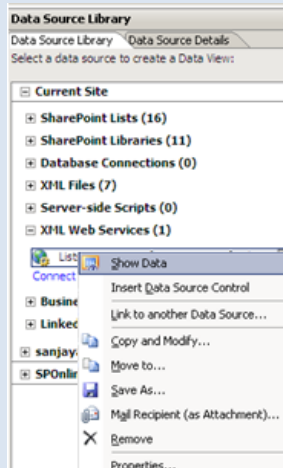


10. In the **Operation** list, click **GetListCollection**.

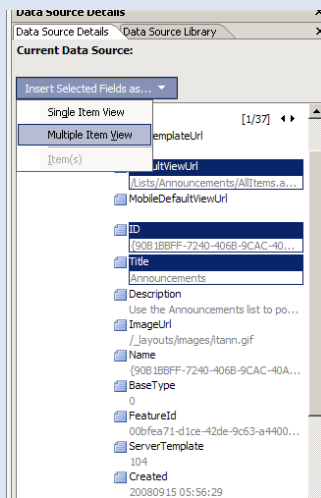
11. Click **OK**.

The service now appears in the **Data Source Library** task pane.

12. Click the service, and then click **Show Data**.



13. Using the CTRL key, select the columns that you want to display.



The results appear as shown in the following figure.

Note: Results will vary depending on your site and the parameters that you pass.

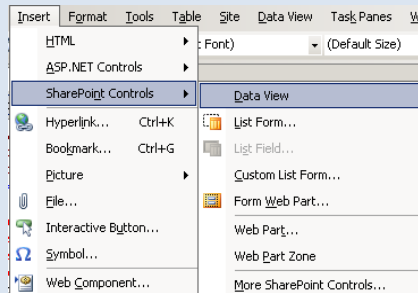
Title	DefaultViewUrl	ID
Announcements	/Lists/Announcements/AllItems.aspx	(90B1BBFF-7240-406E-9CAC-40A08CEF48E1)
BDI	/Lists/BDI/AllItems.aspx	(9FF917A2-6C11-4BB0-B1A4-882ABA6665A6)
Cache Profiles	/Cache Profiles/AllItems.aspx	(6053D270-D64D-4830-825D-DE2D5504B6CE)
Calendar	/Lists/Calendar/calendar.aspx	(B577E732-43C5-4147-B967-0B6A3E2D8A57)
Content and Structure Reports	/Reports List/AllItems.aspx	(1F6287C9-887C-44B4-867F-0E9BE686797F)
Converted Forms	/TW/ConvertedForms/AllItems.aspx	(34E5FEA0-2562-4E8E-8312-9A03694F8998)
Discussion Board	/Lists/Discussion Board/AllItems.aspx	(F94B1967-1315-4E29-AABA-AC51218ADDB6)
Document Store	/Document Store/Forms/AllItems.aspx	(90696A5E-6317-47C9-93F6-21168E5ABFA5)
Effort	/Lists/Effort/AllItems.aspx	(D342C5A7-9B98-4AB9-B072-836AE4023D8D)
Feedback	/Lists/Feedback/AllItems.aspx	(2F148EF0-4F16-425B-9DE4-1073B6D0A452)

5.5 Example: Fetching List Items from a SharePoint Online Site

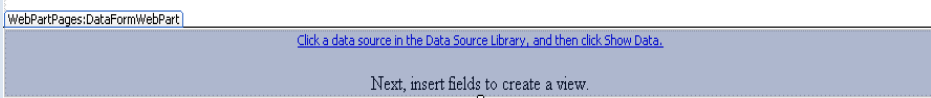
This scenario explains how to fetch list items from a SharePoint Online site using the List Web service, and how to render data using DVWP.

To fetch list items from a SharePoint Online site

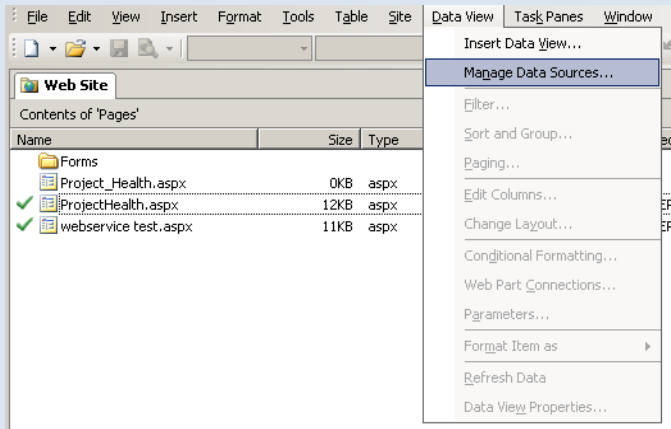
1. Open Office SharePoint Designer 2007.
2. On the **File** menu, click **Open Site**.
3. Enter the site URL of your site.
4. Open the page where you want to display the data.
5. Click the Content Placeholder in page where you need to add the Data Form Web Part.
6. On the **Insert** menu, point to **SharePoint Controls**, and then click **Data View**.



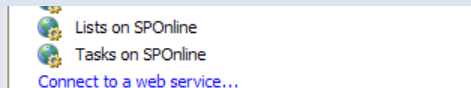
The **Data View** control appears on the page, as shown in the following figure.



7. To set the data source for this Data View, on the **Data View** menu, click **Manage Data Sources**.



8. In the **Data Source Library** task pane, click **Connect to a web service**.

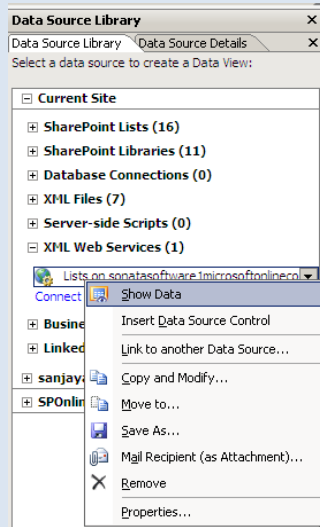


The **Data Source Properties** dialog box appears.

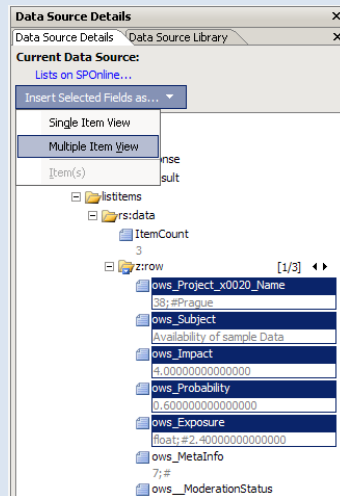
9. Click the **Source** tab, and then in the **Service Description Location** box, enter `https://<SPOnlineSiteName>/_vti_bin/lists.asmx? WSDL`, where `< SPOnlineSiteName >` is SharePoint Online sites URL. The URL will look like <https://contoso1microsoftonline.com-7.sharepoint.microsoftonline.com/default.aspx> and you can obtain this URL by going to the appropriate SharePoint page and copying the URL from the address bar in your browser.
10. Under **Select Connection Info**, in the **Operation** list, click **GetListItems**.
11. Under **Parameters (* required)**, click **listName**.
12. Click **Modify**, and then enter the name of the list from which you want to fetch the items.
13. Click **OK**.

The service now appears in the **Data Source Library** task pane.

14. Click the service, and then click **Show Data**.



15. Using the CTRL key, click the columns that you want to display and then click **Insert Selected Fields as** and then click **Multiple Item View**.



The results for this example appear as shown in the following figure.

Note: Results will vary depending on your site and the list parameters that you pass.

WebPartPages:DataFormWebPart 1 - 10

ows_Project_x0020_Name	ows_Subject	ows_Impact	ows_Probability	ows_Exposure
38.#Prague	Availability of sample Data	4.000000000000000	0.600000000000000	float:#2.400000000000000
38.#Prague	Change requests	6.000000000000000	1.000000000000000	float:#6.000000000000000
39.#Toronto	Change requests	6.000000000000000	0.500000000000000	float:#3.000000000000000

6 Integrating Microsoft Silverlight with a SharePoint Online Page

This section provides an example of integrating the Silverlight application with a SharePoint Online site.

By integrating Silverlight with a SharePoint Online site, you can create a rich user interface. Though writing custom code is not allowed in SharePoint Online, you can write code in a Silverlight application, import the files (.xap) to a SharePoint Online site, and display the data in an .aspx page.

6.1 Example: Using Silverlight to Fetch Data from a SharePoint Online Site Using Web Services

This scenario explains how to fetch list items from the SharePoint Online site using the Lists Web service in SilverLight. The first task is to create a Silverlight application using the Lists Web service in SharePoint Online. The second task is to integrate the Silverlight application by importing the resulting .xap file to the SharePoint Online site and rendering data in an .aspx page.

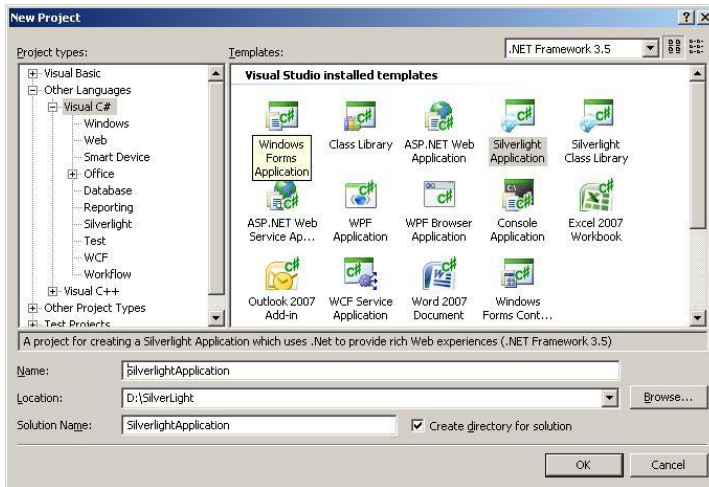
Create a custom list named ProjectList with the following fields.

- **Title**(Single Line of Text)
- **Type**(Single Line of Text)
- **Client**(Single Line of Text)

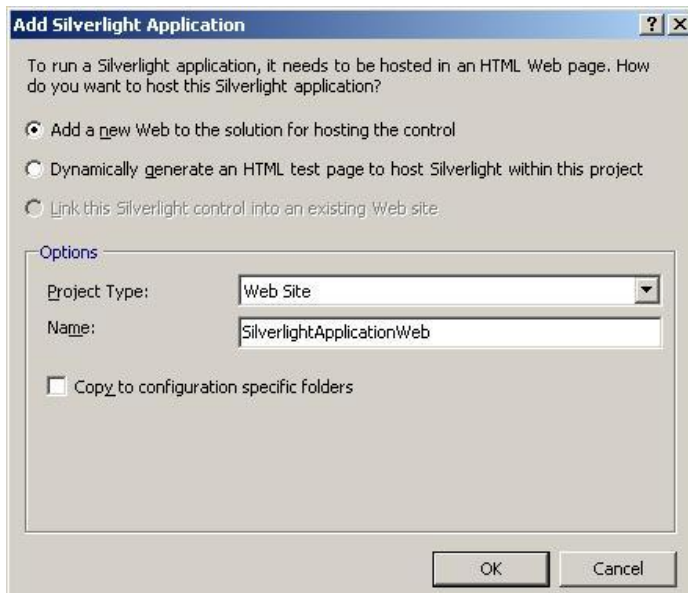
To create a Silverlight application

In this step you fetch list items using the Lists Web service in SharePoint Online

1. Open Visual Studio 2008.
2. On the **File** menu, click **New Project**.
3. In the **New Project** dialog box, under **Templates**, click the Silverlight Application project template, enter SilverlightApplication in the **Name** box and the location in the **Location** box, and then click **OK**.

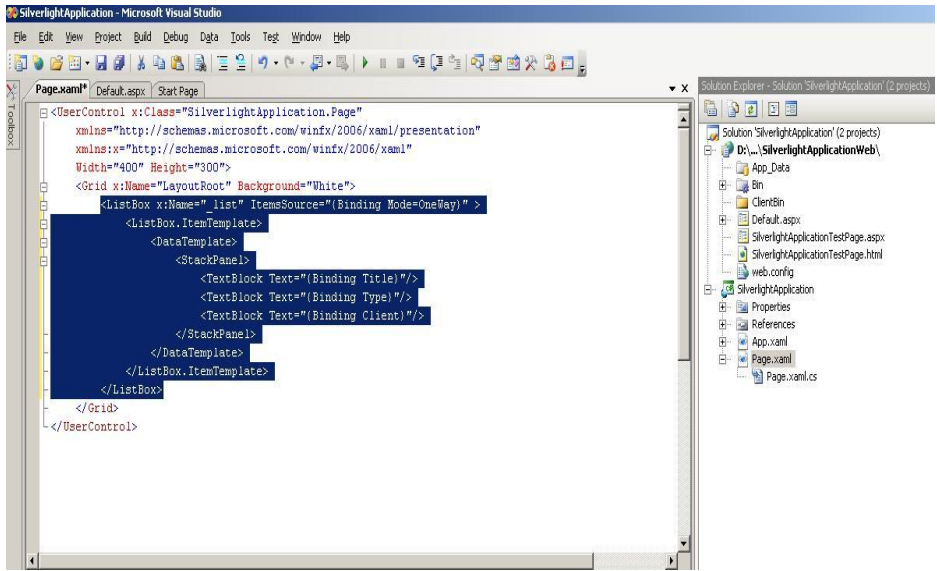


4. In the **Add Silverlight Application** dialog box, leave the default options, and click **OK**.

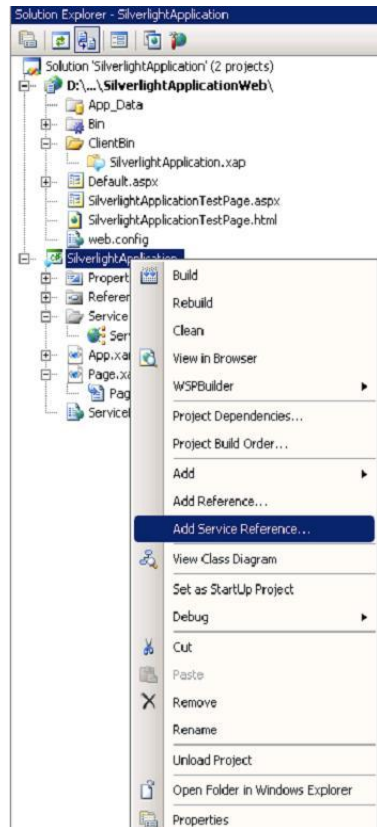


5. In **Solution Explorer**, open the Page.xml page under SilverlightApplication.
6. On the **Page.xml** tab, add the following code between the <Grid> tags:

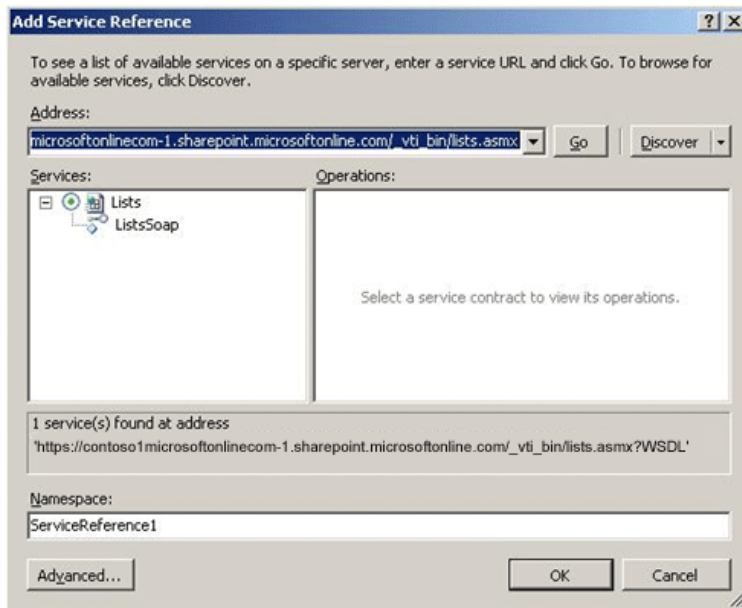
```
<ListBox x:Name="_list" ItemsSource="{Binding Mode=OneWay}" >
  <ListBox.ItemTemplate>
    <DataTemplate>
      <StackPanel>
        <TextBlock Text="{Binding Title}"/>
        <TextBlock Text="{Binding Type}"/>
        <TextBlock Text="{Binding Client}"/>
      </StackPanel>
    </DataTemplate>
  </ListBox.ItemTemplate>
</ListBox>
```



7. In **Solution Explorer**, right-click SilverLightApplication, and then click **Add Service Reference** to add the service reference to the Lists Web service of the SharePoint Online site.



8. Click the **Source** tab, and then in the **Service Description Location** box, enter `https://<SPOnlineSiteName>/_vti_bin/lists.asmx? WSDL`, where `<SPOnlineSiteName >` is SharePoint Online sites URL. The URL will look like <https://contoso1microsoftonlinecom-7.sharepoint.microsoftonline.com/default.aspx> and you can obtain this URL by going to the appropriate SharePoint page and copying the URL from the address bar in your browser.
9. Click **OK** to add the service reference to the Lists Web service.



10. Open **page.xml.cs** and add the following code snippets into the respective places.

11. Copy this code snippet in the **page** class constructor:

```
public Page()
{
    InitializeComponent();
    ListsSoapClient proxy = new ListsSoapClient();
    proxy.GetListItemsCompleted += new
    System.EventHandler<GetListItemsCompletedEventArgs>(proxy_GetListItemsCompleted);
    XDocument doc = XDocument.Parse("<Document><Query /><ViewFields
/><QueryOptions /></Document>");
    XElement query = doc.Element("Query");
    XElement viewFields = doc.Element("ViewFields");
    XElement queryOptions = doc.Element("QueryOptions");
    proxy.GetListItemsAsync("ProjectList", "", query, viewFields, "10",
    queryOptions, "");
}
```

Formatted: French (Switzerland)

Formatted: French (Switzerland)

12. Copy this code snippet after the **page** class constructor (this will fetch the list items and bind the data):

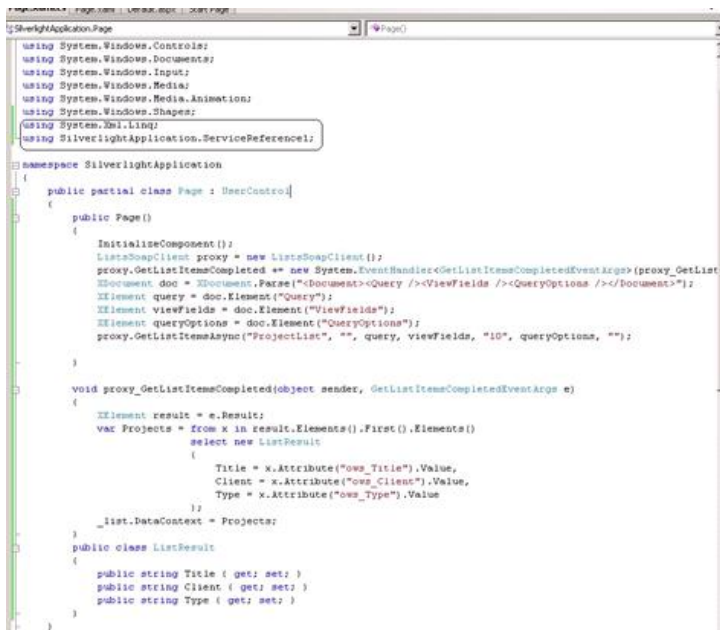
```
void proxy_GetListItemsCompleted(object sender, GetListItemsCompletedEventArgs e)
{
    XElement result = e.Result;
    var Projects = from x in result.Elements().First().Elements()
    select new ListResult {Title=x.Attribute("ows_Title").Value ,
    Client = x.Attribute("ows_Client").Value, Type=x.Attribute("ows_Type").Value };
    _list.DataContext = Projects;
}
```

13. Add this code snippet after the **proxy_GetListItemsCompleted** event handler:

```
public class ListResult
{
    public string Title { get; set; }
    public string Client { get; set; }
    public string Type { get; set; }
}
```

Note: Add the **System.Xml.Linq** and the **SilverlightApplication.ServiceReference1** namespaces in the **page.xml.cs**.

14. The **page.xml.cs** now looks as shown in the following figure:

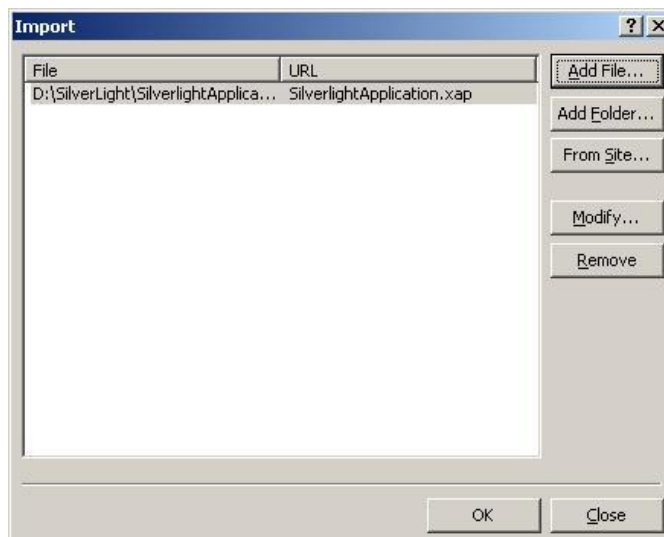


15. Save and build the solution.

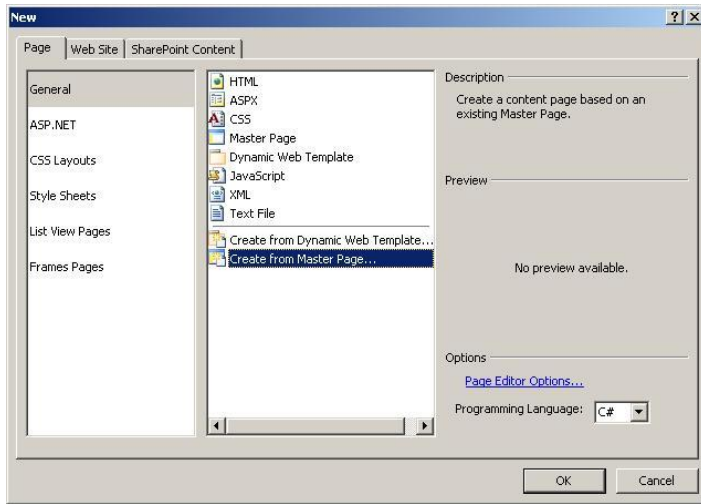
To integrate the Silverlight application with a SharePoint Online site

In this step you import the SilverlightXAP file to the SharePoint Online site and display the results in an aspx page.

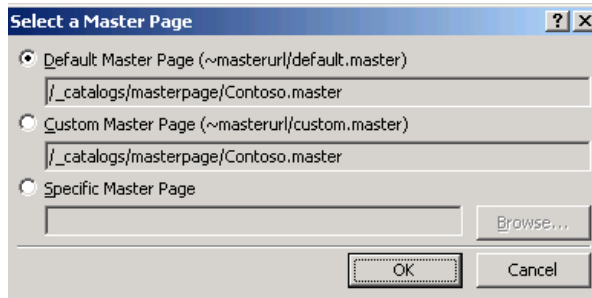
1. Open Office SharePoint Designer 2007.
2. On the **File** menu, click **Open Site**.
3. Enter the site URL of your SharePoint Online site.
4. On the **File** menu, point to **Import**, and then click **File**.
The **Import** dialog box appears.
5. In the **Import** dialog box, click **Add File**, select SilverlightApplication.xap, and then click **OK**.



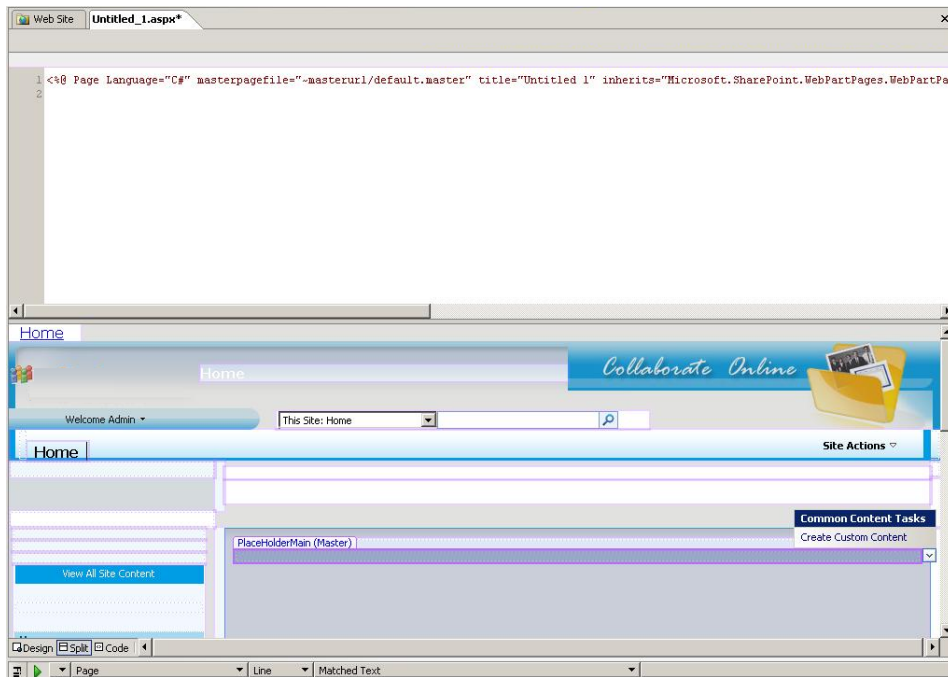
6. On the **File** menu, click **New**, and then click **Page**.
The **New** dialog box appears.



- 7. Click **Create from Master Page**, and click **OK**.
- 8. Leave the default options in the **Select a Master Page** dialog box, and then click **OK**.



- 9. Click **Create Custom Content** in the resulting page.



10. In the Silverlight Web application, open SilverlightApplicationTestPage.html.

11. Copy the following code snippet:

```
<object data="data:application/x-silverlight," type="application/x-silverlight-2-b2" width="100%" height="100%">
<param name="source" value="SilverlightApplication.xap"/>
<param name="onerror" value="onSilverlightError" />
<param name="background" value="white" />
</object>
```

Note: Some unwanted code has been removed.

12. Go to Office SharePoint Designer 2007, and copy the above code snippet to the content placeholder of the newly created .aspx page.

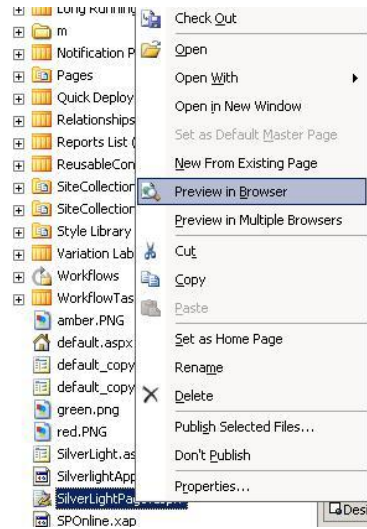
13. On the **File** menu, click **Save**, and enter the file name (for example, **SilverlightPage.aspx**).



```
1 <%@ Page Language="C#" masterpagefile="-masterurl/default.master" title="Untitled 1" inherits="Microsoft.SharePoint.WebF
2 <asp:Content id="Content1" runat="server" contentplaceholderid="PlaceHolderMain">
3   <object data="data:application/x-silverlight," type="application/x-silverlight-2-b2" width="100%" height="100%">
4     <param name="source" value="SilverlightApplication.xap" />
5     <param name="onerror" value="onSilverlightError" />
6     <param name="background" value="white" />
7   </object>
8 </asp:Content>
```

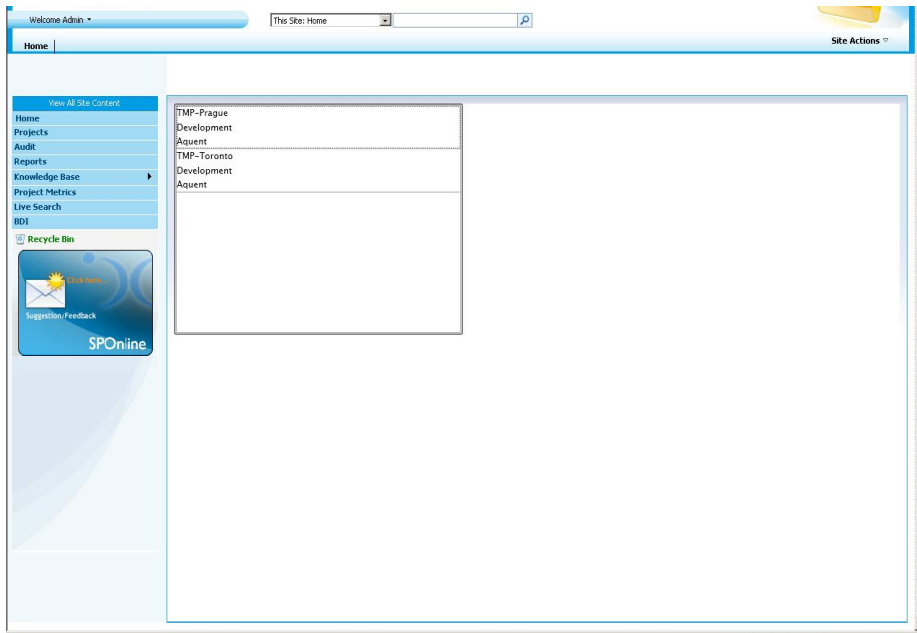
14. Right-click the saved .aspx page (SilverlightPage.aspx), and then click **Preview in browser**.

15. Enter the credentials when prompted.



16. Enter the credentials when prompted.
The results appear as shown in the following figure.

Note: Results vary depending on the data in your list.



7 Workflows Using SharePoint Designer

7.1 About Workflows

Workflows are automations of a business process, in whole or in parts. During the course of a workflow, documents, information, or tasks are passed from one participant to another for action, according to a set of procedural rules.

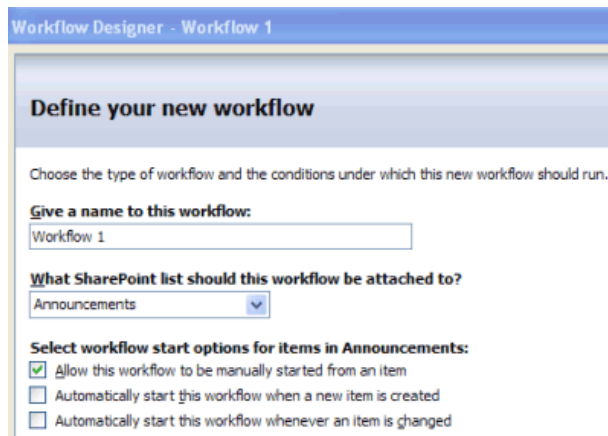
With Microsoft Office SharePoint Designer 2007, you can design workflows that add no-code application logic to your Microsoft SharePoint Online sites and applications. Using the Workflow Designer, you create rules that associate conditions and actions with items in SharePoint lists and libraries. Changes to items in lists or libraries trigger actions in the workflow.

7.2 Creating a No-Code Workflow Using SharePoint Designer

The basic steps of creating a workflow in Microsoft Office SharePoint Designer 2007 are first to define the workflow, and then to create the rules by choosing conditions and actions.

To create a no-code workflow using Office SharePoint Designer 2007

1. Open Office SharePoint Designer 2007.
2. On the **File** menu, click **Open Site**.
3. In the **Open Site** dialog box, browse and select the SharePoint Online site where you want to create the workflow, and then click **Open**.
4. On the **File** menu, point to **New**, and then click **Workflow**.
The **Workflow Designer** opens.

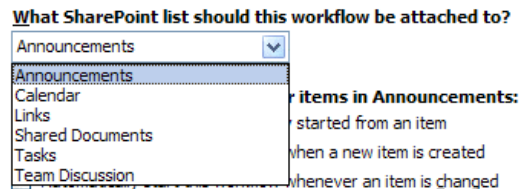


The screenshot shows the 'Workflow Designer - Workflow 1' window. The main heading is 'Define your new workflow'. Below this, it says 'Choose the type of workflow and the conditions under which this new workflow should run.' There are three sections: 1. 'Give a name to this workflow:' with a text box containing 'Workflow 1'. 2. 'What SharePoint list should this workflow be attached to?' with a dropdown menu showing 'Announcements'. 3. 'Select workflow start options for items in Announcements:' with three checkboxes: 'Allow this workflow to be manually started from an item' (checked), 'Automatically start this workflow when a new item is created' (unchecked), and 'Automatically start this workflow whenever an item is changed' (unchecked).

5. In the **Give a name to this workflow** box, type a name for this workflow.

Site visitors will see this name when they view the **Workflow Status** and **Workflows** pages in the browser. By default, new workflows are automatically named Workflow 1, Workflow 2, and so on.

- Click an option in the **What SharePoint list should this workflow be attached to?** list.



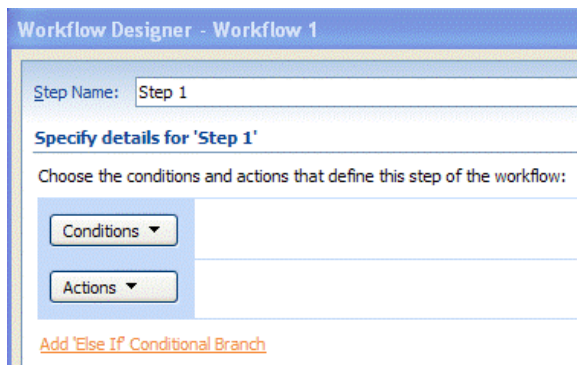
You must attach your workflow to a SharePoint list or a document library. The list you choose here determines where the workflow participants start the workflow. For example, if you want a workflow to start when an item is created or changed in Shared Documents, choose that library. The attached list is where you check the workflow status and history, or manually start a workflow.

- Under **Select workflow start options for items in Announcements**, select one or more of the check boxes.

Note: You must select at least one option if you want to run the workflow.

If you do not select any option, you can design the workflow and then click **Finish**, but the workflow cannot be initiated. Later, you must open the workflow (using the **Open Workflow** command on the **File** menu), select an initiation option, and then click **Finish**. This is useful for saving a workflow that you are in the process of designing.

- Click **Next**.
- In the **Step Name** box, type a name for the first step of your workflow.

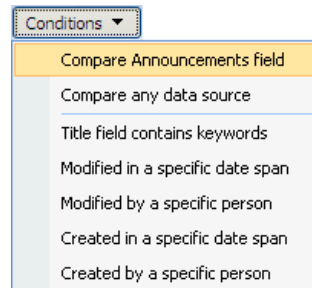


For the next several steps, in the **Specify details for 'Step 1'** section you will be creating the rules for your workflow by choosing the actions that you want it to perform and any conditions that must be satisfied for those actions to occur.

Note: Not every step must contain a condition; for example, you can have steps that are simply lists of actions to be performed. However, every step can have at most one conditional branch that contains an action without an associated condition.

This is because a conditional branch that does not have a condition acts as the "Else" branch in an "Else If" conditional branch. If condition A is true, do action B; otherwise ("else") do action C. As such, it would not make sense to have more than one "Else" branch because the last branch would never be reached.

10. For each condition that you want to specify, click **Conditions**, and then click that condition in the **Conditions** list.



11. Repeat this step until you have specified all the conditions that you want to include.

Each conditions that you choose appears in the box to the right of the **Conditions** button, as a joined If statement that includes hyperlinked parameters.

Note: If you do choose more than one condition, the conditions are joined, meaning that all the conditions must be satisfied before the workflow performs the associated actions. Multiple conditions create a progressive filter in which the order matters, because the first condition is evaluated first.

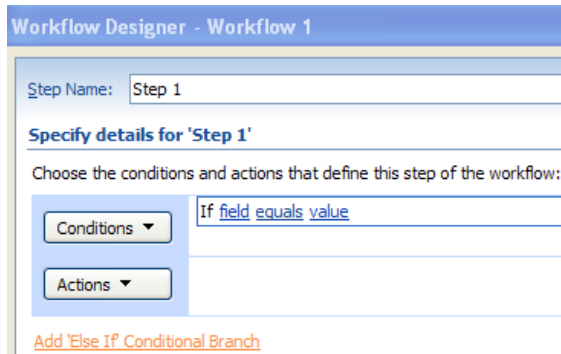
To move a condition up or down in a list of conditions, click the condition, click the down arrow that appears, and then click either **Move Condition Up** or **Move Condition Down** in the list.

Office SharePoint Designer 2007 provides a number of predefined conditions. In addition, you can create *advanced conditions* and *custom conditions*.

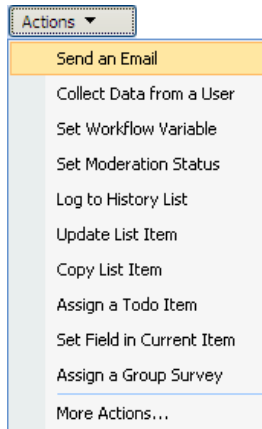
With custom conditions, you can compare a field in the current **Conditions** list with a value. With advanced conditions, you can compare one value to another value. This allows you to create a comparison between a field in any list and a value from a wide range of sources.

- 12. For each condition, click each hyperlinked parameter, and then choose a value for the required parameter.

For example, a condition might include links for choosing a field in the list, a comparison, and a value.



- 13. For each action that you want to specify, click **Actions**, and then click that action in the **Actions** list.



- 14. Repeat this step until you have specified all the actions that you want to include.

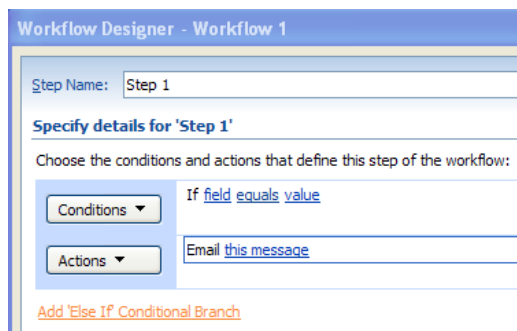
Note: If the action you want does not appear in the list, click **More Actions**, click the action that you want, and then click **Add**.

Note: If you do create more than one action for a condition, the actions can run either sequentially (joined by **then**) or at the same time (joined by **and**).

To switch between running in sequence and running in parallel, click the arrow in the upper right corner of the rule, and then click either **Run All Actions in Sequence** or **Run All Actions in Parallel**.

15. For each action, click each hyperlinked parameter, and then choose a value for the required parameter.

For example, an action might include a link for creating an e-mail message.



16. To add a conditional branch, click **Add 'Else If' Conditional Branch**, and then repeat steps 10 through 13 to create another rule.

A rule is a set of related conditions and actions, as shown in the previous illustration. When the condition is true, the workflow performs the associated actions. However, if the condition is not true, you can specify additional conditions by adding conditional branches.

For example, you can create three different conditions for when a document status is changed to Approved, Rejected, or Pending. You can also create an Else condition, which causes the workflow to perform the action only when none of the previous conditions have been satisfied. To create an Else condition, the last action in the rule must be an action with no conditions. The workflow evaluates all the conditions; if none of them are satisfied, the workflow performs the actions in the final branch that has no condition.

17. When this step of the workflow is complete, click **Next**.
18. For each step in the workflow, repeat steps 10 to 14 to create additional sets of conditions and actions, or rules.

Note: As you add rules, keep in mind that each rule can hold only one set of Else If conditional branches, where the actions in each branch are performed only when the associated condition is satisfied. You must add additional rules when:

- You have multiple sets of Else If conditional branches to be evaluated.
 - You need to separate a branched statement from a non-branched statement.
-

19. To check the workflow for errors before you exit the Workflow Designer, click **Check Workflow**.

If there is a workflow error, the hyperlink changes color and asterisks appear before and after the invalid parameter. In addition, under **Workflow Steps**, an error symbol appears next to each step that contains an error.

20. Check any errors by making sure that the parameters entered are valid.

21. Click **Finish**.

The workflow is saved and attached to the list that you specified.

7.3 Collecting Data from the User and Processing It in the Workflow

You can collect data from a workflow participant, and then use that data in your workflow for making further decisions. In this process, the first task is to collect data from a user, by creating a custom task form to collect data from a user, and then storing the ID of the task item in a variable. The second task is to set a workflow variable, by using the task ID variable to look up the user response and then storing that particular response in a new variable. The final task is to use the variable to make decisions, by using the **Compare any data source** condition.

To collect data from a user

In this procedure you create a custom task form to collect data from a user and store the ID of the task item in a variable.

1. Open Office SharePoint Designer 2007.
2. On the **File** menu, click **Open Site**.
3. In the **Open Site** dialog box, browse and select the SharePoint Online site where you want to create the workflow, and then click **Open**.
4. On the **File** menu, point to **New**, and then click **Workflow**.
5. On the first page of the **Workflow Designer**, in the **Give a name to this workflow** box, name the workflow **Document review**.
6. In the **What SharePoint list should this workflow be attached to?** list, click **Documents for review** to select the list or the library to which the workflow will be attached.

7. Under **Select workflow start options for items in Shared Documents**, do the following:
 - Clear the **Allow this workflow to be manually started from an item** check box.
 - Select the **Automatically start this workflow whenever an item is created** check box.
 - Clear the **Automatically start this workflow whenever an item is changed** check box.

The screenshot shows the 'Workflow Designer - Document review' window. The main heading is 'Define your new workflow'. Below this, there is a section for naming the workflow, a dropdown menu for selecting the SharePoint list, and a section for selecting start options. The 'Automatically start this workflow when a new item is created' option is selected.

Workflow Designer - Document review

Define your new workflow

Choose the type of workflow and the conditions under which this new workflow

Give a name to this workflow:
Document review

What SharePoint list should this workflow be attached to?
Documents for review

Select workflow start options for items in Documents for review:

- Allow this workflow to be manually started from an item
- Automatically start this workflow when a new item is created
- Automatically start this workflow whenever an item is changed

8. Click **Next**.
9. Click **Actions**, and then click **Collect Data from a User** in the **Actions** list.

Note: If this action does not appear in the list, click **More Actions** to see the full list.
10. In the **Collect data from this user** action, click the first hyperlinked parameter; in this example it is **data**.
11. On the first page of the **Custom Task Wizard**, click **Next**.
12. On the second page of the **Custom Task Wizard**, for this example do the following (see the following figure):
 - In the **Name** box, type **Review document** as the name for this task.

Note: The task name must be unique in the current site.
 - In the **Description** box, type **Please review this document. Then choose either Approved or Rejected, and click Complete Task.** as the instructions to the recipient of this task.

This description appears at the top of the custom task form, so it should include here any instructions that you want the workflow participant to see, not just a general description.

Wizard

Enter a title and description that explains to the recipient the purpose of this task. This information will be displayed in the item added to the Tasks list.

Name:
Review document

Description:
Please review ths document. Then choose either Approved or Rejected, and click Complete Task.

You can set other task fields, like Expires, by adding a secondary workflow to the Tasks list.
[Click here for more information.](#)

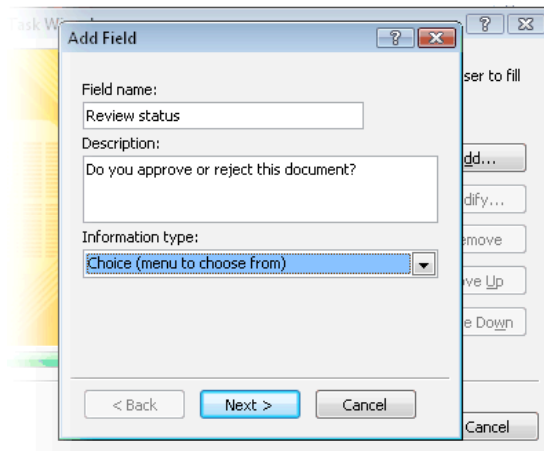
< Back Next > Cancel

13. Click **Next**.

14. Click **Add** to define a custom form field for this task.

15. On the first **Add Field** page, for this example do the following (see the following figure):

- In the **Field name** box, type **Review status** as the name for this field.
- In the **Description** box, type **Do you approve or reject this document?** as the question for which this form field is storing the answer.
- In the **Information type** list, click **Choice (menu to choose from)** to format the field appropriately for the type of information that you are collecting.



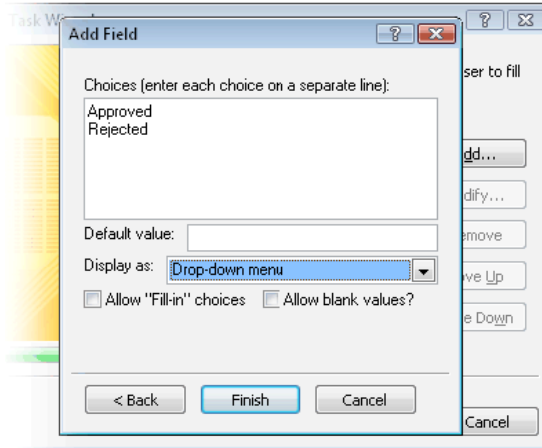
16. Click **Next**.

The next page of the wizard varies depending on which information type you selected on the previous page.

17. On the second **Add Field** page, for this example do the following (see the following figure):

- In the **Choices (enter each choice on a separate line)** box, type **Approved**, press ENTER, and then type **Rejected**.
- In the **Display as list**, click **Drop-down menu**.

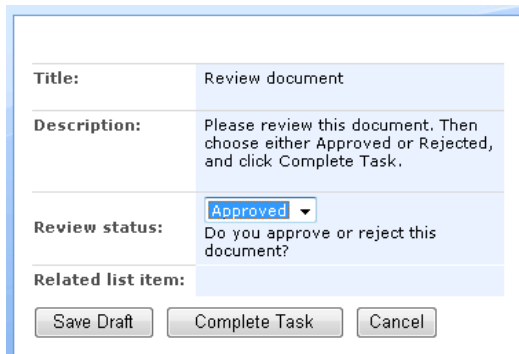
- Clear the **Allow blank values?** check box, to permit the reviewer to choose either **Approved** or **Rejected**.



Rejected.

18. Click **Finish** twice to return to the **Workflow Designer**.

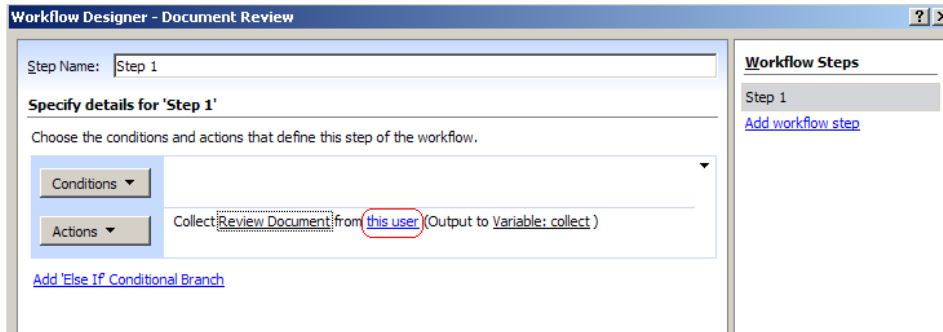
The following illustration shows how this task will be displayed to the workflow participant in the browser when the participant edits the task. This is the custom task form (`Review document.aspx`) that you just designed by using the Custom Task Wizard.



In the **Workflow Designer**, you now set the second and third parameters of the **Collect data from this user** action. (You began setting the first parameter in step 10.)

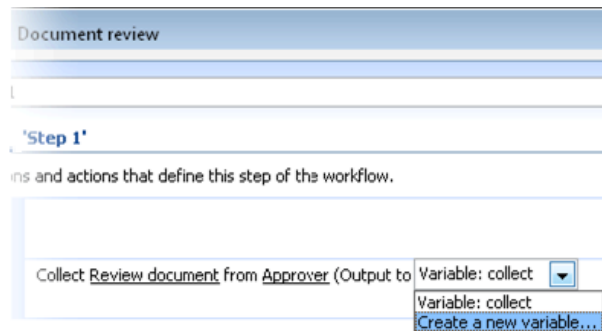
The action now appears as **Collect Review Document from this user (Output to Variable: collect)** next to the **Actions** button.

19. In the action, click the second hyperlinked parameter, which is **this user** (see the following figure).

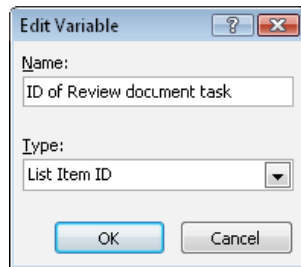


20. In the **Select Users** dialog box, type a name or e-mail address, or click an existing user or group in the list; for this example type **Approver**.
21. Click **Add**, and then click **OK**.
22. In the action, click the third hyperlinked parameter, **Variable: collect**, and then click **Create a new variable** from the list (see the following figure).

Note: Click **Create a new variable** instead of using the default name (**Variable: collect**) when your workflow will have multiple variables and you want to give this variable a more descriptive name.



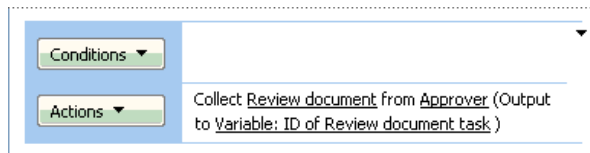
23. In the **Edit Variable** dialog box, in this example type **ID of Review document task** to describe the value stored by the variable.



For this variable, **List Item ID** is the only choice in the **Type** list.

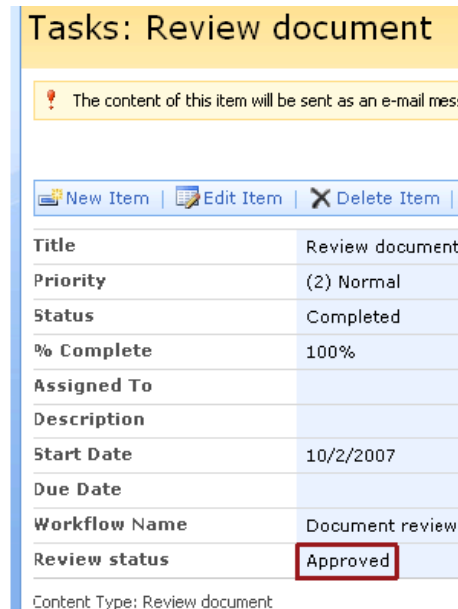
24. Click **OK**.

At this point, your action should look like this.



To set a workflow variable

In this procedure, you use the task ID variable to look up the user response, and then store that particular response in a new variable. At this point in the workflow, a workflow participant submits a completed task. The review status submitted by that person is stored in a field or column in the Tasks list, as shown here.



The screenshot shows a task list item titled "Tasks: Review document". At the top, there is a yellow banner with a warning icon and the text "The content of this item will be sent as an e-mail message". Below this is a toolbar with three buttons: "New Item", "Edit Item", and "Delete Item". The main content is a table with the following fields and values:

Title	Review document
Priority	(2) Normal
Status	Completed
% Complete	100%
Assigned To	
Description	
Start Date	10/2/2007
Due Date	
Workflow Name	Document review
Review status	Approved

At the bottom of the table, it says "Content Type: Review document". The "Review status" cell is highlighted with a red border.

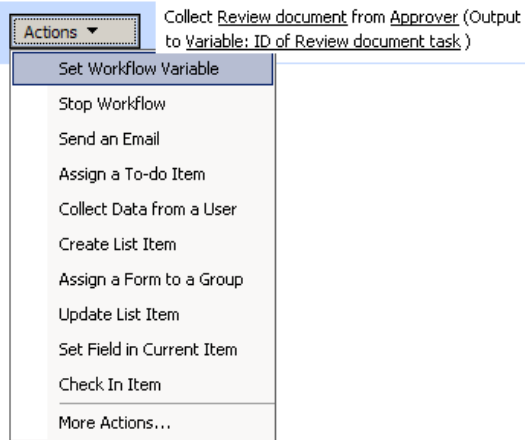
How do you get at this information to use it in another step in the workflow?

You use the **Set Workflow Variable** action to look up the value of the **Review status** field for the task that the user edited, and store the value of this field in a new variable. In this example, the new variable is also named **Review status**.

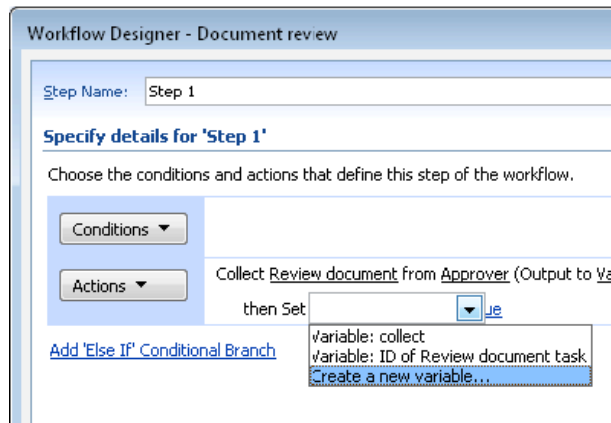
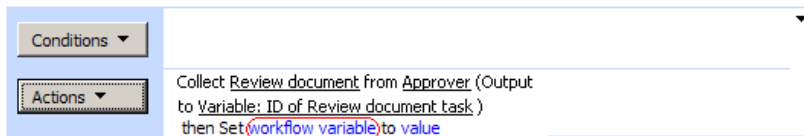
By looking up and storing the review status in a new variable, you can now use this variable as the basis for future comparisons. This is a best practice, because the workflow now makes it clear where the data is coming from, and this data can be reused more easily (multiple times, if necessary).

1. Click **Actions**, and then click **Set Workflow Variable**.

Note: If this action does not appear in the list, click **More Actions** to see the full list.

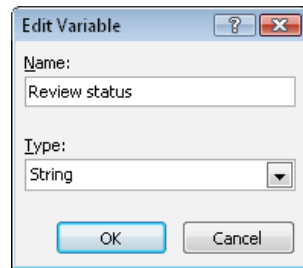




2. In the action, click **workflow variable**, and then click **Create a new variable** from the list.

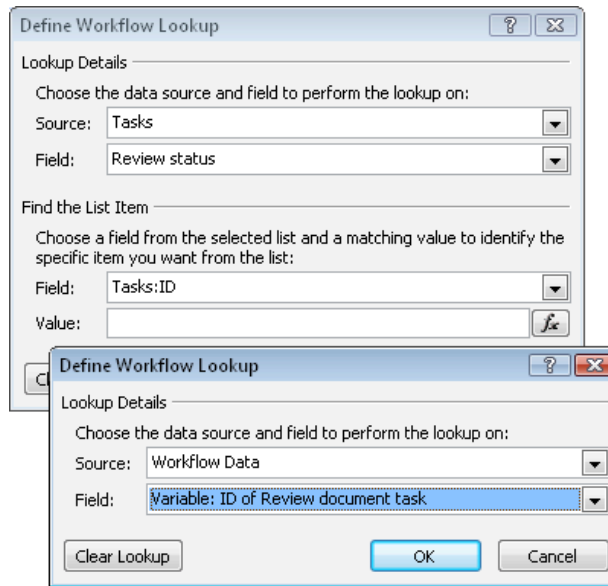


3. In the **Edit Variable** dialog box, for this example do the following:

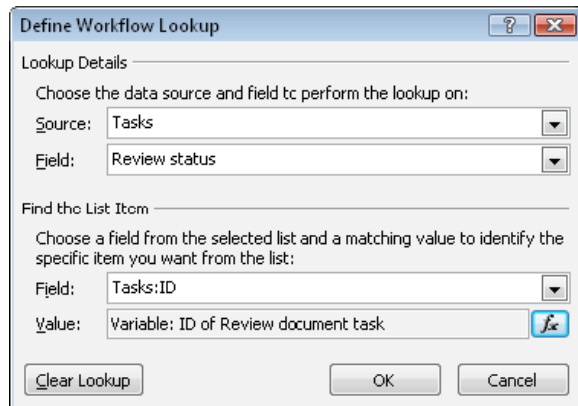
- In the **Name** box, type **Review status** as the name that describes the value stored by the variable.
- In the **Type** list, click **string** as the type of data to be stored in the variable. This variable will store the review status as **Approved** or **Rejected**, which are text strings.



4. Click **OK**.
5. In the action, click **value**, and then click the **Display data binding** icon .
6. In the **Define Workflow Lookup** dialog box, choose the following options:
 - In the **Lookup Details** section:
 - **Source** list: Click **Tasks** (if more than one tasks list is available, choose the list that contains the custom form field that you created in the previous section).
 - **Field** list: Click **Review status** (choose the custom form field that you created in the previous section).
 - In the **Find the List Item** section:
 - **Field** list: Click **Tasks:ID**.
 - **Value** box: Click the **Display data binding** icon .
7. In the second **Define Workflow Lookup** dialog box, choose the following options:
 - **Source** list: Click **Workflow Data**.
 - **Field** list: Click **Variable: ID of Review document task**.



- 8. Click **OK** to return to the first page of the **Define Workflow Lookup** dialog box. The final workflow lookup should resemble the following figure.

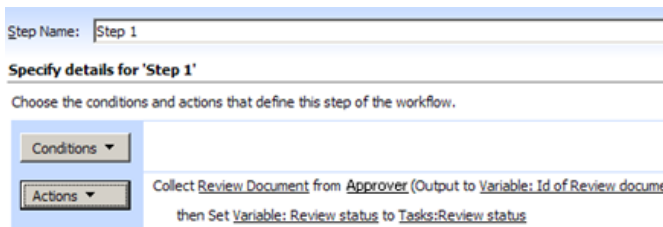


This lookup says, “Go to the **Review status** field in the **Tasks** list. Then go to the row in the **Tasks** list whose ID matches the ID stored in the variable for this instance of the workflow.”

If the workflow runs and creates a task whose ID is 3, this lookup retrieves the value of the **Review status** column from the third row in the **Tasks** list. The next time the workflow runs, this ID may be 4, so the lookup goes to the fourth row, and so on.

9. Click **OK** again.

At this point, your rule should look like this:

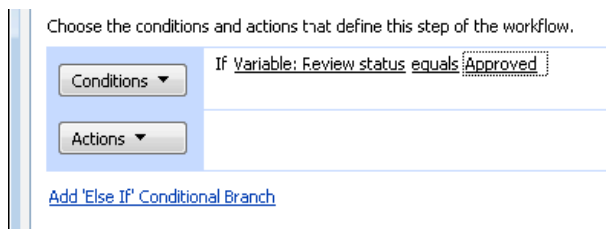


To use a variable to make decisions

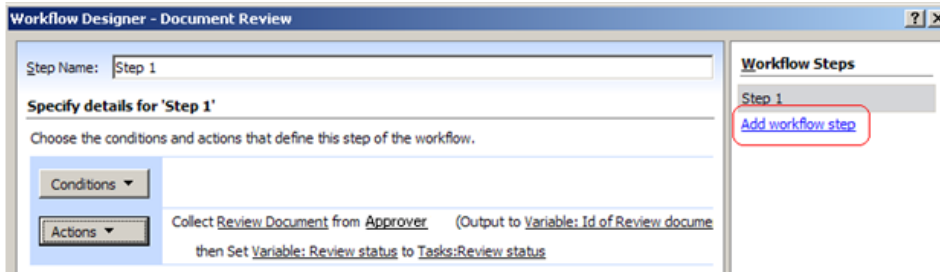
Now that you have the task ID variable (in this example, **Variable: Review status**) you are ready to use it as the basis for making decisions in the workflow. To do this, you use the **Compare any data source** condition, creating two conditional branches. For example, you can add a step to your workflow that says “If **Variable: Review status** equals **Approved**, run these actions; if **Variable: Review status** equals **Rejected**, run different actions.”

Create the first conditional branch

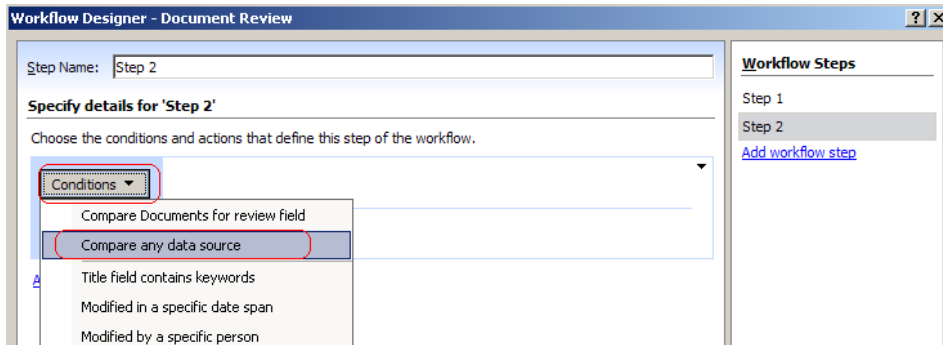
If the value stored in Variable:Review status equals **Approved**, the workflow will perform any actions associated with this condition.




1. In the Workflow Designer, under **Workflow Steps**, click **Add workflow step**.



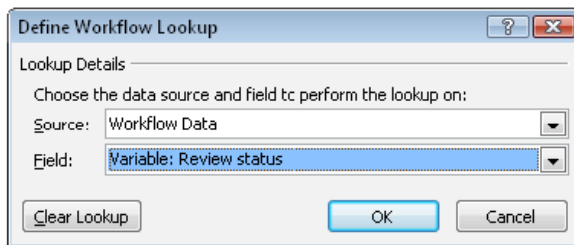
2. Click **Conditions**, and then click **Compare any data source**.



3. In the **Compare any data source** condition, click the first hyperlinked parameter, and then click the **Display data binding** icon .

4. In the **Define Workflow Lookup** dialog box, choose the following options:

- **Source** list: Click **Workflow Data**.
- **Field** list: Click **Variable: Review status**.



5. Click **OK**.


6. In the condition, leave **equals** as the value for the second parameter.

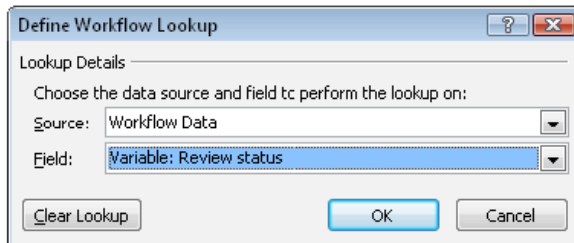
7. Click the third hyperlinked parameter, type **Approved**, and then press ENTER.



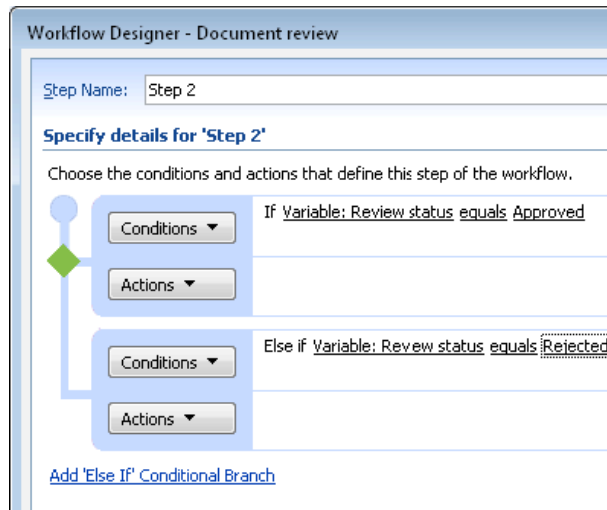
Create the second conditional branch

The second conditional branch handles the case where the value stored in **Variable: Review status** equals **Rejected**.

8. Click **Add 'Else If' Conditional Branch**.
9. Click **Conditions**, and then click **Compare any data source**.
10. In the condition, click the first hyperlinked parameter, and then click the **Display data binding** icon .
11. In the **Define Workflow Lookup** dialog box, choose the following options:
 - **Source** list: Click **Workflow Data**.
 - **Field** list: Click **Variable: Review status**.



12. Click **OK**.
13. In the condition, leave **equals** as the value for the second parameter.
14. In the condition, click the second hyperlinked parameter, type **Rejected**, and then press ENTER.
 Before you add any actions to the condition in each branch, this step of the workflow should look like this:



15. Click **Actions**, and then add the actions that you want each conditional branch to perform.

7.4 Example: Send a SOW Document to the Project Manager for Review

This scenario is applicable when a team member needs to get a project artifact reviewed and approved by the project manager. The project artifacts, such as detail designs and high-level designs, are stored in the Document Store library.

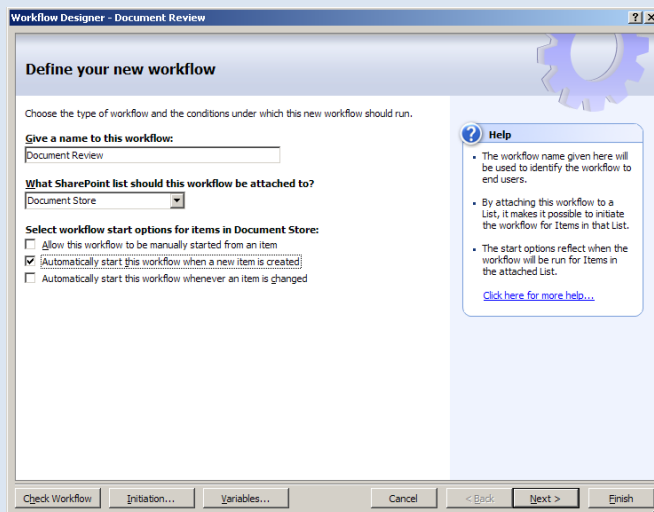
Prerequisite: In the SharePoint Online site, create a new document library. For this example, name it **Document Store**.

To create a workflow for routing a document for review

1. Open Office SharePoint Designer 2007.
2. On the **File** menu, click **Open Site**.
3. In the **Open Site** dialog box, select the SharePoint Online site where you want to create the workflow, and then click **Open**.
4. On the **File** menu, point to **New**, and then click **Workflow**.
5. On the first page of the Workflow Designer, In the **Give a name to this workflow** box, type **Document Review**.
Site visitors will see this name when they view the **Workflow Status** and **Workflows** pages in the browser.
6. In the **What SharePoint list should this workflow be attached to?** list, click

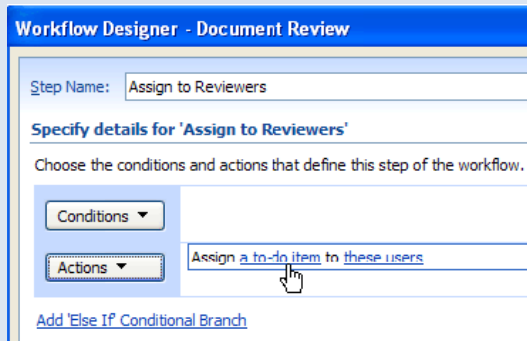
Document Store.

7. Under **Select workflow start options for items in Document Store**, do the following:
 - Clear the **Allow this workflow to be manually started from an item** check box.
 - Select the **Automatically start this workflow when a new item is created** check box.
 - Clear the **Automatically start this workflow whenever an item is changed** check box.



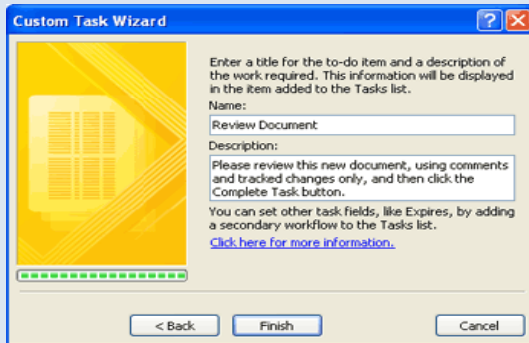
8. Click **Next**.
9. Click **Actions**, and then click **Assign a To-do Item** in the **Actions** list.

Note: If this action is not available in the list, click **More Actions** to see the full list.
10. In the action, click the first hyperlinked parameter, in this example it is a **to-do item**.



11. On the first page of the **Custom Task Wizard**, click **Next**.
12. On the second page of the **Custom Task Wizard**, for this example do the following:
 - In the **Name** box, type **Review Document** as the name for this task.
Note: The task name must be unique in the current site.
 - In the **Description** box, type **Please review this new document, using comments and track changes only, and then click the Complete Task button**.

This description appears at the top of the custom task form, so it should include the instructions that the workflow participant should use.



13. Click **Finish**.
The following illustration shows how this task will appear to the workflow participant in the browser. This is the custom task form (`Review Document.aspx`) that you just designed using the **Custom Task Wizard**.

Title:	Review Document
Description:	Please review this new document, using comments and tracked changes only, and then click the Complete Task button.
Related list item:	Document for review
<input type="button" value="Complete Task"/> <input type="button" value="Cancel"/>	

14. In the action, click the Second hyperlinked parameter, in this example it is **these users**.

Choose the conditions and actions that define this step of the workflow.

Conditions ▾	
Actions ▾	Assign Review Document to these users

15. In the **Select Users** dialog box, type a name or e-mail address, or click an existing user or group in the list, and then click **Add**.
16. Repeat this step to include other participants to whom you want to assign this task.

Note: In the **Assign a To-do Item** action, the workflow creates one task for each person and group that you select. It does not expand the group and assign a separate task to each person in the group.

Instead of entering a specific user or group name, or e-mail address, you can store this information in a separate list, and then use workflow lookup to get this information. This way, you can configure various users and groups as workflow participants without modifying the workflow itself.

17. When all the appropriate participants appear in the **Selected Users** box, click **OK**.

8 Index

A

- actions108
- add workflow124
- adding custom content38
- adding, editing, or deleting a heading or link.....43
- applying a style sheet40
- applying master page36
- applying theme28
- available SharePoint Online Web Services.....77

C

- capabilities and limitations9
- changing background color.....43
- collaboration emplates12
- conditional branch124
- conditions108
- configuring a subsite.....41
- create master page35
- create site17
- creating blog site27
- creating custom site template.....21
- creating new site25, 26
- custom templates13
- customization
 - capabilities.....9
 - customizing cascading style sheets39
 - customizing master page29
 - customizing page layouts.....37
 - customizing SharePoint Online site.....28
 - customizing web sites.....28

D

- data centers7
- data view
 - conditional formatting60
 - create53
 - sorting and grouping60
- default site template.....11
- design of home page43

E

- else if conditional branch 126
- excluded SharePoint Online Web Services 81

F

- features 7
- form web part 61

I

- inserting content field 39
- installing site template..... 24
- integrating third party web services 62
- introduction 6

L

- locate and create default.master 35

M

- master page
 - apply 36
 - customize 29
- master page galleries 30
- meetings templates 13

N

- navigation editing and sorting 43
- no code workflow 108

O

- opening page layout 37
- opening SharePoint Online site 35
- overview 6

P

- page layout
 - add custom content 38
 - insert content field..... 39

open37
 process user data in workflow113

R

removing site template25
 reordering navigation items43

S

set workflow variable 119, 120
 SharePoint Online Web Services77
 available.....77
 excluded81
 site template
 install24
 remove.....25
 site templates
 collaboration.....12
 blank site.....12
 blog.....12
 document workspace.....12
 team site12
 wiki site12
 custom13
 board of directors.....14
 business performance reporting.....14
 case management for government agencies ...14
 classroom management14
 clinical trial initiation and management.....14
 competitive analysis site.....14
 discussion database.....15
 disputed invoice management.....15
 employee activities site.....15
 employee self-service benefits15
 employee training scheduling and materials15
 equity research15
 integrated marketing campaign tracking15
 manufacturing process management15
 new store opening15
 product and marketing requirements planning .16
 request for proposal16
 sports league.....16
 team work site16
 timecard management16
 default site templates11
 meetings13
 basic meeting workspace13
 solution scenario10

W

workflows 108

