Aspose.Words for SharePoint Documentation

Introducing Aspose.Words for SharePoint

Does your organization base a corporate document depository on Windows SharePoint Services or Microsoft Office SharePoint Server's technologies?

Do you have to utilize multiple document formats in your internal work process or in your interaction with customers and partners?

Would you like to see an affordable and fast tool capable of converting documents right in your SharePoint's document library with high fidelity but without the use of Microsoft Word automation?

Do you generate documents based on database data? Would you like to do it right in your SharePoint document library and save the result in multiple document formats?

Aspose.Words for SharePoint is a unique solution, which enables you to

- Perform high-fidelity document conversion to multiple formats.
- Generate reports based on a Microsoft Word template document and an external data source.

You can click on a document in a SharePoint document library and convert it to any of the supported formats:

- PDF - Adobe Portable Document
- DOCX – Office Open XML
- DOC - Microsoft Word 97 - 2003 Document
- RTF - Rich Text
- MHT - Web Page Archive
- TXT - Plain Text
- XML - Microsoft Word 2003 WordprocessingML
- XML – Flat OPC
- ODT - OpenDocument
- EPUB - IDPF Digital Book
- XPS - XML Paper Specification

You can put a report template in a SharePoint document library and make reports from this template saving the result in any of the formats above.

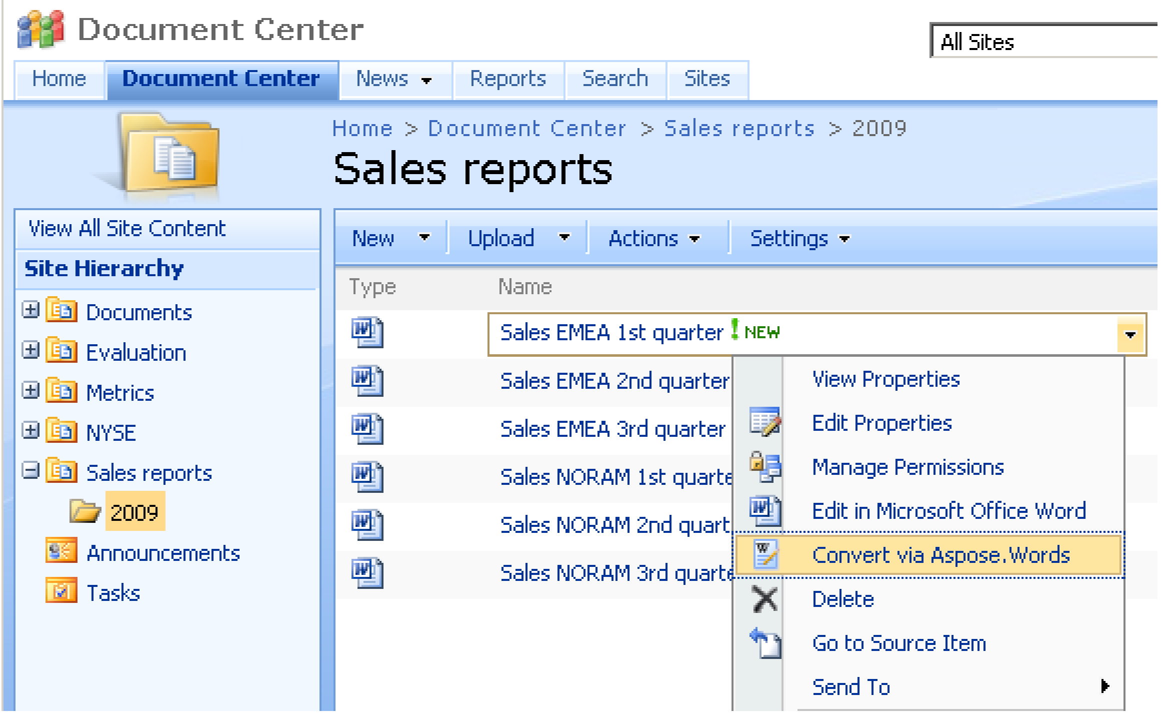
Use Aspose.Words for SharePoint with the following products:

- Windows SharePoint Services 3.0 (WSS)
- Microsoft Office SharePoint Server 2007 (MOSS)
- Microsoft SharePoint Foundation 2010
- Microsoft SharePoint Server 2010
- x32 and x64 servers supported.

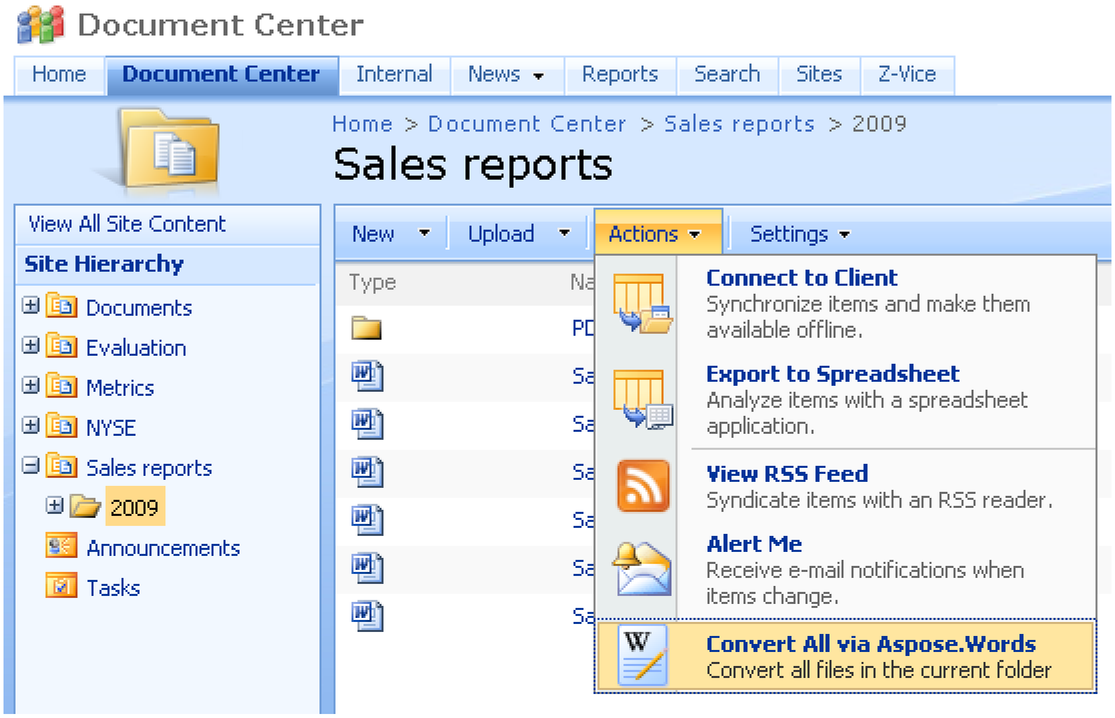
There are no additional system requirements besides those existing for the products above.

Please read on to learn more. This documentation describes the features, installation, evaluation, licensing, common use cases and settings of Aspose.Words for SharePoint.

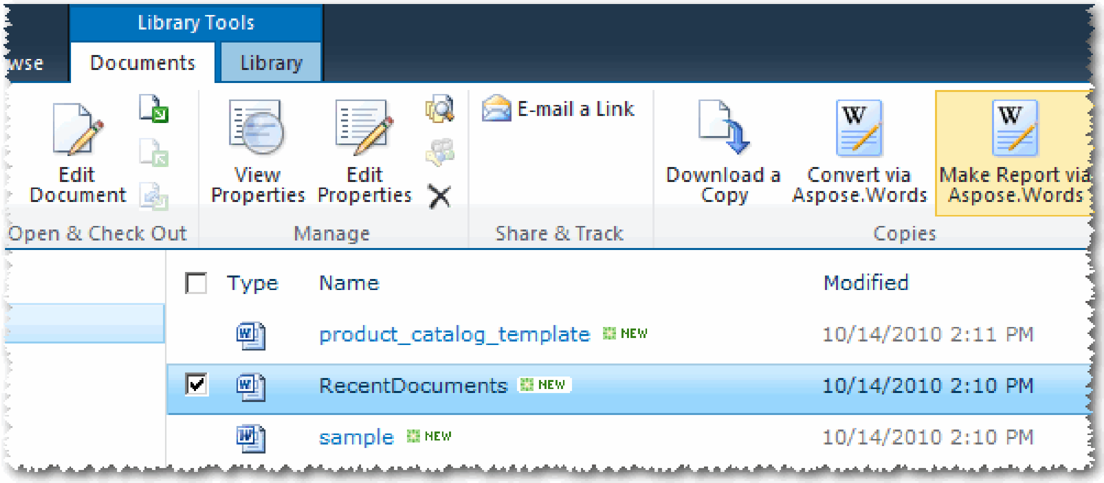
You can use Aspose.Words for SharePoint right from your SharePoint's document library to convert a particular document:



Or to convert all files in the current folder:

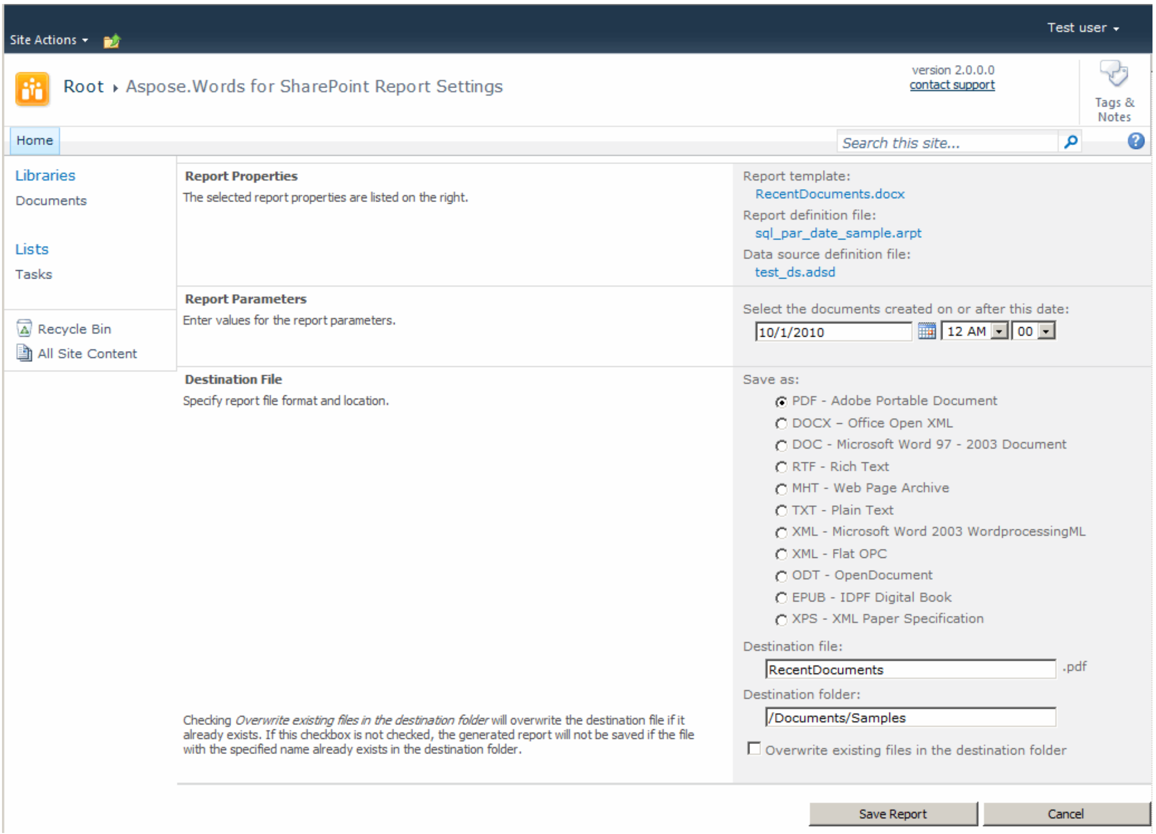

Document Center
Home
Document Center
Internal
News
Reports
Search
Sites
Z-Vice
Home > Document Center > Sales reports > 2009
Sales reports
View All Site Content
Site Hierarchy
Documents
Evaluation
Metrics
NYSE
Sales reports
2009
Announcements
Tasks
New
Upload
Actions
Settings
Type
Connect to Client
Synchronize items and make them available offline.
Export to Spreadsheet
Analyze items with a spreadsheet application.
View RSS Feed
Syndicate items with an RSS reader.
Alert Me
Receive e-mail notifications when items change.
Convert All via Aspose.Words
Convert all files in the current folder

You can start a report from a report template.



□	Type	Name	Modified
		product_catalog_template NEW	10/14/2010 2:11 PM
☑		RecentDocuments NEW	10/14/2010 2:10 PM
		sample NEW	10/14/2010 2:10 PM

Aspose.Words for SharePoint lets you enter report parameters and chose destination format and location.



Report Properties
The selected report properties are listed on the right.

Report template:
RecentDocuments.docx
Report definition file:
sql_par_date_sample.arpt
Data source definition file:
test_ds.adsd

Report Parameters
Enter values for the report parameters.

Select the documents created on or after this date:
10/1/2010 | 12 AM | 00

Destination File
Specify report file format and location.

Save as:

- ● PDF - Adobe Portable Document
- ○ DOCX – Office Open XML
- ○ DOC - Microsoft Word 97 - 2003 Document
- ○ RTF - Rich Text
- ○ MHT - Web Page Archive
- ○ TXT - Plain Text
- ○ XML - Microsoft Word 2003 WordprocessingML
- ○ XML - Flat OPC
- ○ ODT - OpenDocument
- ○ EPUB - IDPF Digital Book
- ○ XPS - XML Paper Specification

Destination file:
RecentDocuments .pdf
Destination folder:
/Documents/Samples

Checking *Overwrite existing files in the destination folder* will overwrite the destination file if it already exists. If this checkbox is not checked, the generated report will not be saved if the file with the specified name already exists in the destination folder.

□ Overwrite existing files in the destination folder

Features

Multiple formats support

Having Aspose.Words for SharePoint installed you can convert documents right from your SharePoint document library between many popular office document formats and most of the conversions is done with high fidelity.

Currently Aspose.Words for SharePoint supports the following input formats:

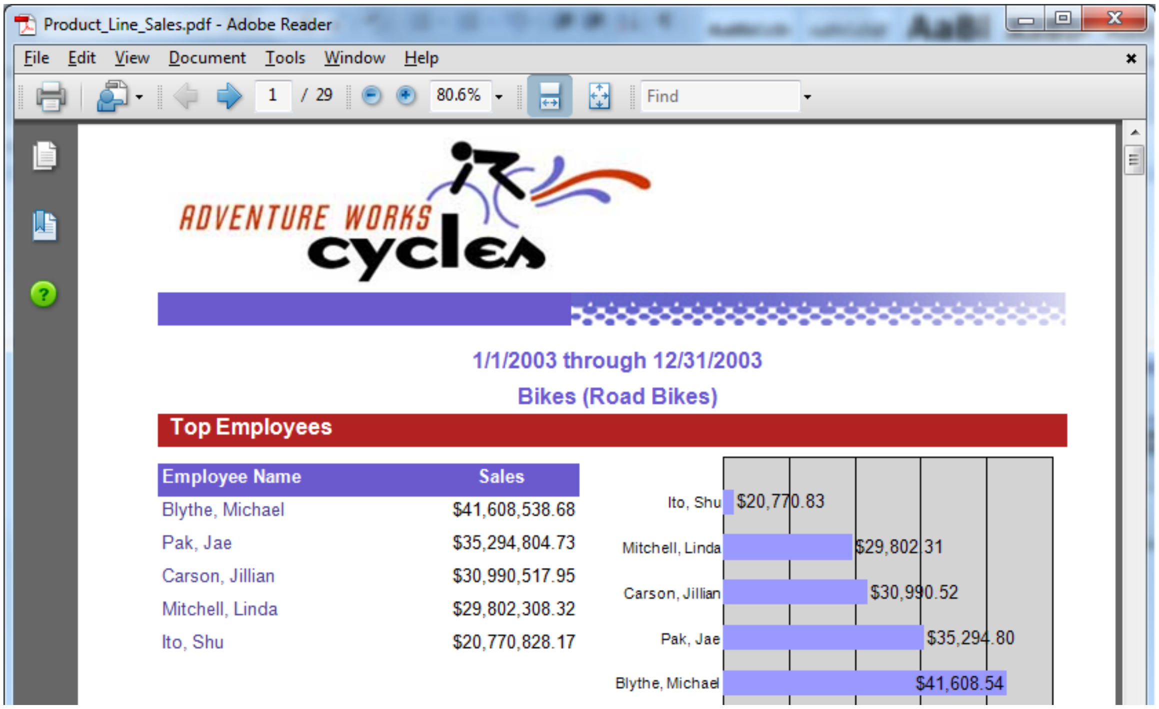
- DOCX – Office Open XML
- DOC - Microsoft Word 97 - 2003 Document
- RTF - Rich Text
- MHT - Web Page Archive
- XML - Microsoft Word 2003 WordprocessingML
- XML - Flat OPC
- ODT - OpenDocument

To generate documents, Aspose.Words for SharePoint relies on a built-in version of Aspose.Words for .NET, the market leading document-processing component from Aspose. **DOC, OOXML, RTF and WordprocessingML** copies of the same document **will look identical to each other**, thanks to the high degree of support for the underlying file formats, provided by Aspose.Words. Microsoft Word is not used to generate documents.

The full list of output formats supported by Aspose.Words for SharePoint is below:

- PDF - Adobe Portable Document
- DOCX – Office Open XML
- DOC - Microsoft Word 97 - 2003 Document
- RTF - Rich Text
- MHT - Web Page Archive
- TXT - Plain Text
- XML - Microsoft Word 2003 WordprocessingML
- XML - Flat OPC
- ODT - OpenDocument
- EPUB - IDPF Digital Book
- XPS - XML Paper Specification

Adobe Portable Document (PDF) document generated by Aspose.Words for SharePoint from Microsoft Word 97 – 2003 (DOC) source.



Product_Line_Sales.pdf - Adobe Reader
File Edit View Document Tools Window Help
1 / 29
80.6%
Find
ADVENTURE WORKS cycles

1/1/2003 through 12/31/2003

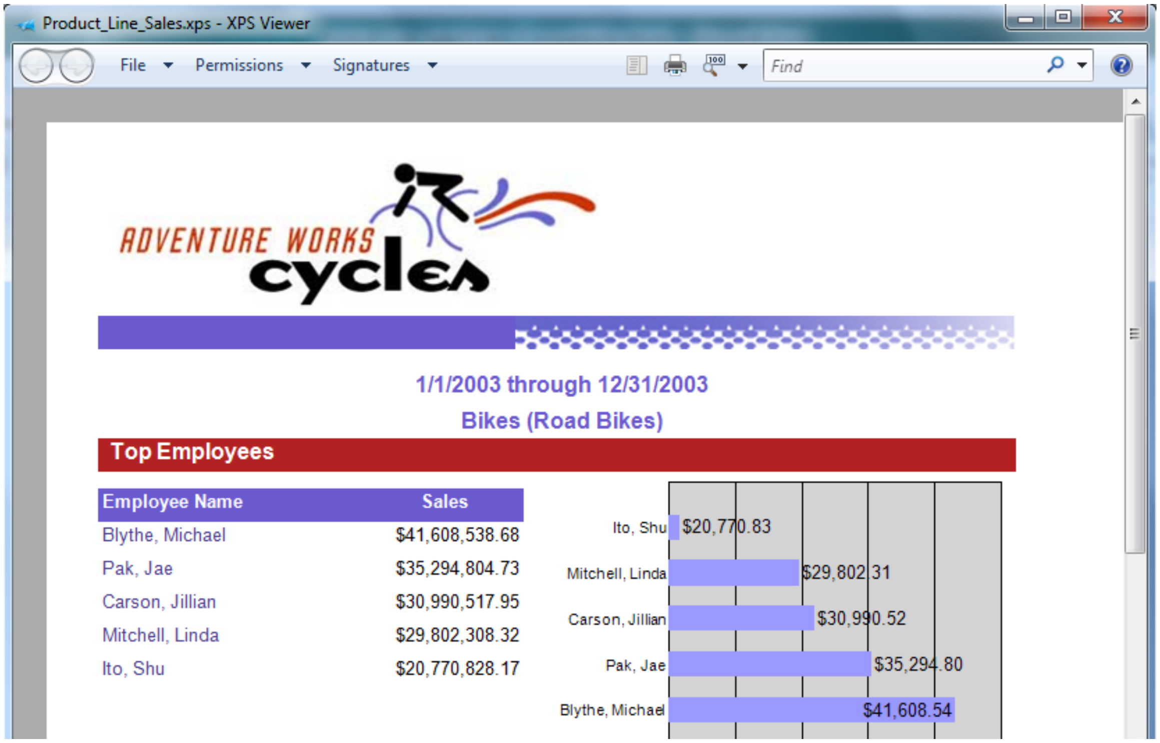
Bikes (Road Bikes)

Top Employees

Employee Name	Sales
Blythe, Michael	$41,608,538.68
Pak, Jae	$35,294,804.73
Carson, Jillian	$30,990,517.95
Mitchell, Linda	$29,802,308.32
Ito, Shu	$20,770,828.17

Ito, Shu $20,770.83
Mitchell, Linda $29,802.31
Carson, Jillian $30,990.52
Pak, Jae $35,294.80
Blythe, Michael $41,608.54

XML Paper Specification (XPS) document generated by Aspose.Words for SharePoint from Microsoft Word 97 – 2003 (DOC) source.



Product_Line_Sales.xps - XPS Viewer
File Permissions Signatures
Find
ADVENTURE WORKS cycles

1/1/2003 through 12/31/2003

Bikes (Road Bikes)

Top Employees

Employee Name	Sales
Blythe, Michael	$41,608,538.68
Pak, Jae	$35,294,804.73
Carson, Jillian	$30,990,517.95
Mitchell, Linda	$29,802,308.32
Ito, Shu	$20,770,828.17

Ito, Shu $20,770.83
Mitchell, Linda $29,802.31
Carson, Jillian $30,990.52
Pak, Jae $35,294.80
Blythe, Michael $41,608.54

Different conversion options

The following options are available during conversion setup:

Files concatenation

If you select a folder or multiple files for conversion, you can use “Combine Files” checkmark to combine files in different formats into single document. You can set the order in which source files are placed in the combined document.

Convert files in subfolders

This option is available under SharePoint 2007 when a folder is selected for conversion. You can recursively convert all subfolders of the selected folder by using this option. The same subfolder structure is created below the destination folder during conversion. Under SharePoint 2010, the folders are always processed recursively. To process files without subfolders, just select the needed files in a document library view before starting conversion. Do not checkmark any subfolders.

Save destination folder option

Once you have selected Destination folder you can use this option to not have to go through the same steps again. It will save your selection and prepopulate Destination folder field upon next conversion session.

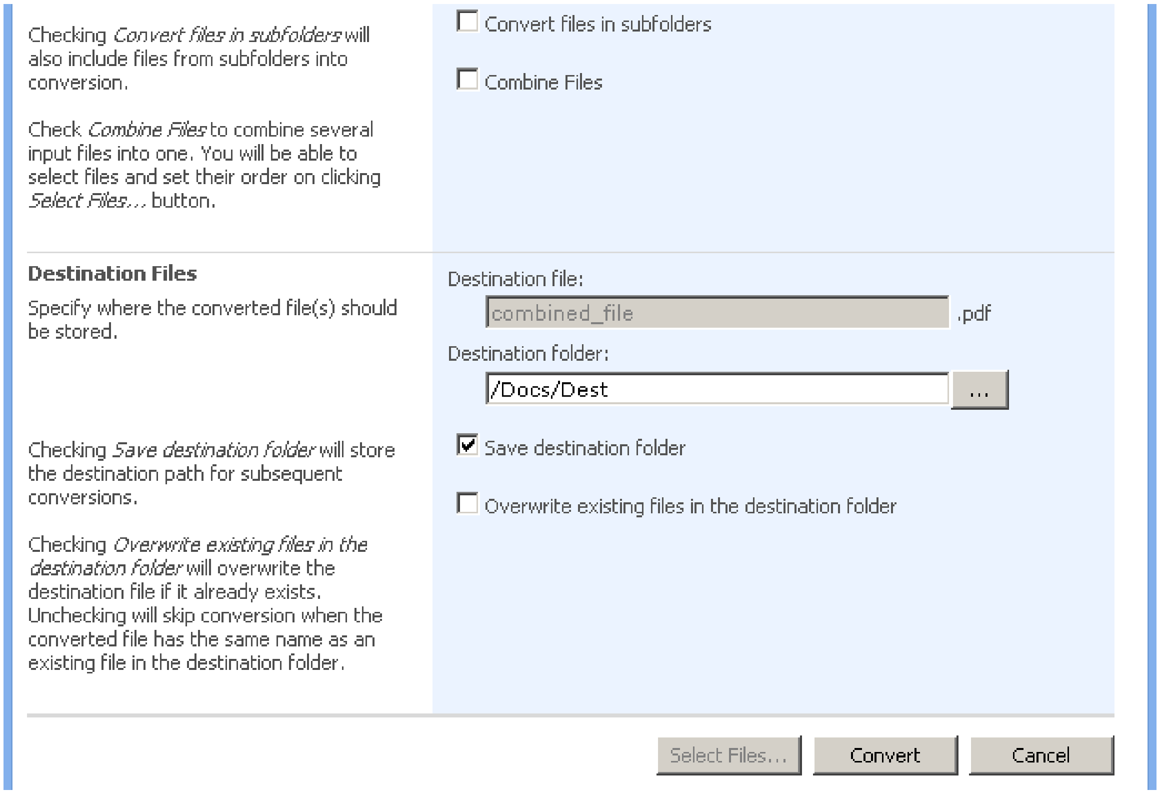
Overwrite existing files in the destination folder

Destination folder you have selected can already contain some of the files you want to convert. This option allows you to skip conversion for such files and process only files which are not presented in the destination folder.

Browse for destination folder

Corporate site can consists of dozens of document libraries with complex URL to them and manual entering of Destination folder can be a headache. So, to not have to remember the whole URL path to the particular library or subfolder within it you can use Browse button to choose Destination folder visually.

Different options will help to adjust the conversion process.



Checking *Convert files in subfolders* will also include files from subfolders into conversion.

Check *Combine Files* to combine several input files into one. You will be able to select files and set their order on clicking *Select Files...* button.

□ Convert files in subfolders

□ Combine Files

Destination Files

Specify where the converted file(s) should be stored.

Destination file: combined_file .pdf

Destination folder: /Docs/Dest

Checking *Save destination folder* will store the destination path for subsequent conversions.

Checking *Overwrite existing files in the destination folder* will overwrite the destination file if it already exists. Unchecking will skip conversion when the converted file has the same name as an existing file in the destination folder.

☑ Save destination folder

□ Overwrite existing files in the destination folder

Report Generation

Use Aspose.Words for SharePoint for generating reports based on a Microsoft Word template file.

- Prepare a template file. Edit the visual report layout using Microsoft Word.
- Put placeholders in the report template where external data must be inserted.
- Prepare an XML report definition describing report properties like data source connection, queries to retrieve the data, report parameters and so on.
- Put report template and report definition in a SharePoint document library.
- Link report template and report definition via report template property.

Now you can use you report template to make reports. Click on the report template and select the appropriate menu item. Enter report parameters and choose saving format and location.

Report template supports

- Regions to repeat for each record in a data source table.
- Nested regions for parent-child relationships in the report data.
- Image placeholders to insert images from a data source.

You can access data from

- Relational databases via SQL queries and Microsoft SQL or OLE DB providers.
- SharePoint lists via CAML queries.
- XML data files located in a SharePoint document library.

In a report definition file the following report properties are specified:

- Data source connection properties.

- Queries to retrieve the data.
- Relationships between data tables returned by different queries.
- Report parameters.

You can specify multiple data source connections and queries. You can combine data returned by the different queries using data relationships in the report definition file.

You can store data source connection description in a separate file and use it with multiple reports.

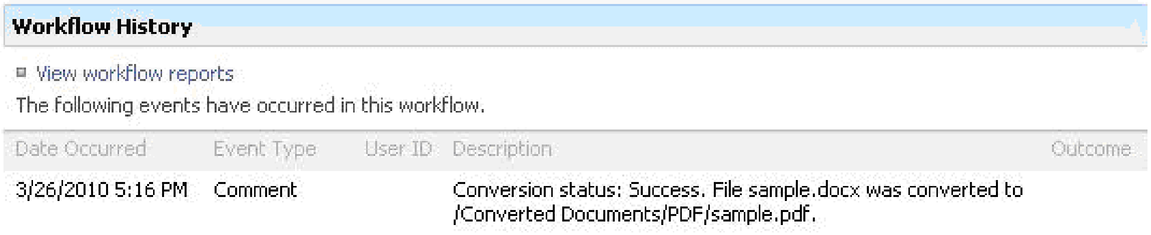
Workflow support

You can use Aspose.Words for SharePoint from a document workflow. Upon installation, these new workflow activities will appear in SharePoint Designer:

- Convert via Aspose.Words
- Make a Report via Aspose.Words

Use them in your workflows to convert documents or build reports.

An entry for the conversion activity logged in the workflow history.



Workflow History

View workflow reports

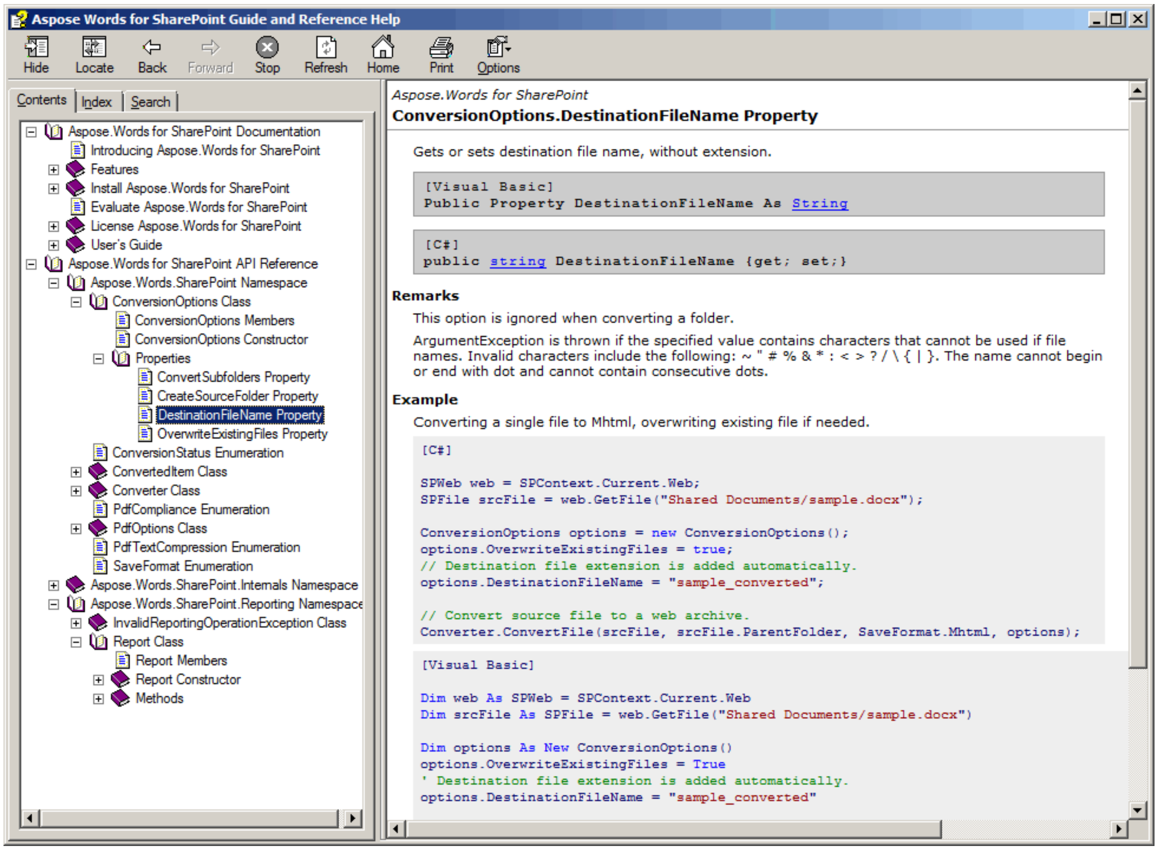
The following events have occurred in this workflow.

Date Occurred	Event Type	User ID	Description	Outcome
3/26/2010 5:16 PM	Comment		Conversion status: Success. File sample.docx was converted to /Converted Documents/PDF/sample.pdf.	

Public API

Customers developing their own SharePoint solution can access our conversion and reporting engine via public API. This includes all supported file formats and conversion options. Both single and multiple files conversions are available. You can save the converted document to stream and specify additional PDF saving options. You can build a report once and save many times to different formats and locations. The detailed interface specification in our API Reference is accompanied with dozens of C# and VB samples.

Usage of each class member is accompanied with C# and VB samples.


Aspose Words for SharePoint Guide and Reference Help
Hide
Locate
Back
Forward
Stop
Refresh
Home
Print
Options
Contents
Index
Search
Aspose.Words for SharePoint Documentation
Introducing Aspose.Words for SharePoint
Features
Install Aspose.Words for SharePoint
Evaluate Aspose.Words for SharePoint
License Aspose.Words for SharePoint
User's Guide
Aspose.Words for SharePoint API Reference
Aspose.Words.SharePoint Namespace
ConversionOptions Class
ConversionOptions Members
ConversionOptions Constructor
Properties
ConvertSubfolders Property
CreateSourceFolder Property
DestinationFileName Property
OverwriteExistingFiles Property
ConversionStatus Enumeration
ConvertedItem Class
Converter Class
PdfCompliance Enumeration
PdfOptions Class
PdfTextCompression Enumeration
SaveFormat Enumeration
Aspose.Words.SharePoint.Internals Namespace
Aspose.Words.SharePoint.Reporting Namespace
InvalidReportingOperationException Class
Report Class
Report Members
Report Constructor
Methods

Aspose.Words for SharePoint

ConversionOptions.DestinationFileName Property

Gets or sets destination file name, without extension.

```
[Visual Basic]
Public Property DestinationFileName As String
```

```
[C#]
public string DestinationFileName {get; set;}
```

Remarks

This option is ignored when converting a folder.

ArgumentException is thrown if the specified value contains characters that cannot be used if file names. Invalid characters include the following: ~ " # % & * : < > ? / \ { | }. The name cannot begin or end with dot and cannot contain consecutive dots.

Example

Converting a single file to Mhtml, overwriting existing file if needed.

```
[C#]

SPWeb web = SPContext.Current.Web;
SPFile srcFile = web.GetFile("Shared Documents/sample.docx");

ConversionOptions options = new ConversionOptions();
options.OverwriteExistingFiles = true;
// Destination file extension is added automatically.
options.DestinationFileName = "sample_converted";

// Convert source file to a web archive.
Converter.ConvertFile(srcFile, srcFile.ParentFolder, SaveFormat.Mhtml, options);
```

```
[Visual Basic]

Dim web As SPWeb = SPContext.Current.Web
Dim srcFile As SPFile = web.GetFile("Shared Documents/sample.docx")

Dim options As New ConversionOptions()
options.OverwriteExistingFiles = True
' Destination file extension is added automatically.
options.DestinationFileName = "sample_converted"
```

World Class Free Technical Support

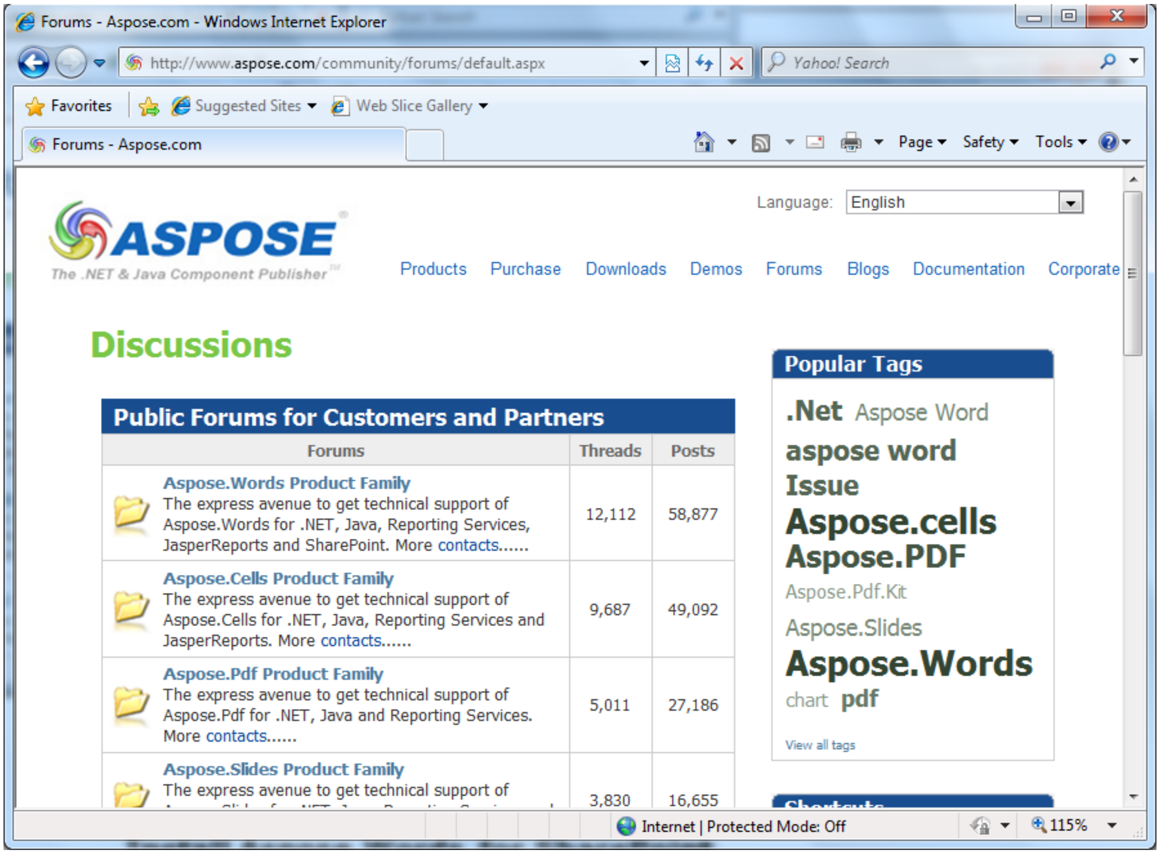
Aspose is renowned for its free and unlimited technical support provided directly by the developers of the products.

If there is a new version of a product available, or if resolving a question results in a fix in the product, all new releases are free if you have an active subscription.

Aspose.Support Forums is the place to not only resolve technical issues, but also to participate in development discussions with the vibrant and growing community of Aspose users. Currently there are over 40,000 users registered at the Aspose website.

Aspose.Blogs is the place to look for information about latest releases and about what Aspose developers have to say.

There is plenty of activity in the Aspose.Support Forums.



Discussions

Public Forums for Customers and Partners

Forums	Threads	Posts
Aspose.Words Product Family The express avenue to get technical support of Aspose.Words for .NET, Java, Reporting Services, JasperReports and SharePoint. More contacts......	12,112	58,877
Aspose.Cells Product Family The express avenue to get technical support of Aspose.Cells for .NET, Java, Reporting Services and JasperReports. More contacts......	9,687	49,092
Aspose.Pdf Product Family The express avenue to get technical support of Aspose.Pdf for .NET, Java and Reporting Services. More contacts......	5,011	27,186
Aspose.Slides Product Family The express avenue to get technical support of	3,830	16,655

Install Aspose.Words for SharePoint

System Requirements

Server Requirements

To install Aspose.Words for SharePoint you need a computer running one of the following:

- Windows SharePoint Services 3.0
- Microsoft Office SharePoint Server 2007
- Microsoft SharePoint Foundation 2010
- Microsoft SharePoint Server 2010

Installing the latest service packs and other updates for these products is recommended.

Client Requirements

You need a browser to access SharePoint sites. The following browsers were checked with Aspose.Words for SharePoint:

- Microsoft Internet Explorer 6, 7, 8
- Mozilla FireFox 3.5
- Google Chrome 3
- Safari 4

Installing Aspose.Words for SharePoint

Aspose.Words for SharePoint is downloadable as Aspose.Words.SharePoint.zip archive.

This archive contains:

Aspose.Words.SharePoint.wsp	SharePoint solution file. Aspose.Words for SharePoint is packaged as a SharePoint solution to facilitate deployment/retraction and feature activation/deactivation across the server farm.
Aspose.Words.SharePoint2010.wsp	A solution file for SharePoint 2010. The setup program automatically detects the installed SharePoint version and uses an appropriate solution file.
Aspose_LicenseAgreement.rtf	End user license agreement
Aspose.Words for SharePoint Guide.pdf	User documentation
Aspose.Words for SharePoint API Reference.chm	User documentation and API reference in the hypertext format
setup.exe	Setup program
setup.exe.config	Setup configuration file
Report.xsd	Report definition schema file
DataSource.xsd	Data source definition schema file

The setup program checks the following conditions before proceeding:

- One of the supported SharePoint versions is installed.
- The user has permission to install SharePoint solutions.
- SharePoint database is online.
- SharePoint Administration service is started.
- SharePoint Timer service is started.

Administration service and Timer service are needed because some setup actions rely on a timer job to propagate to all servers in the server farm.

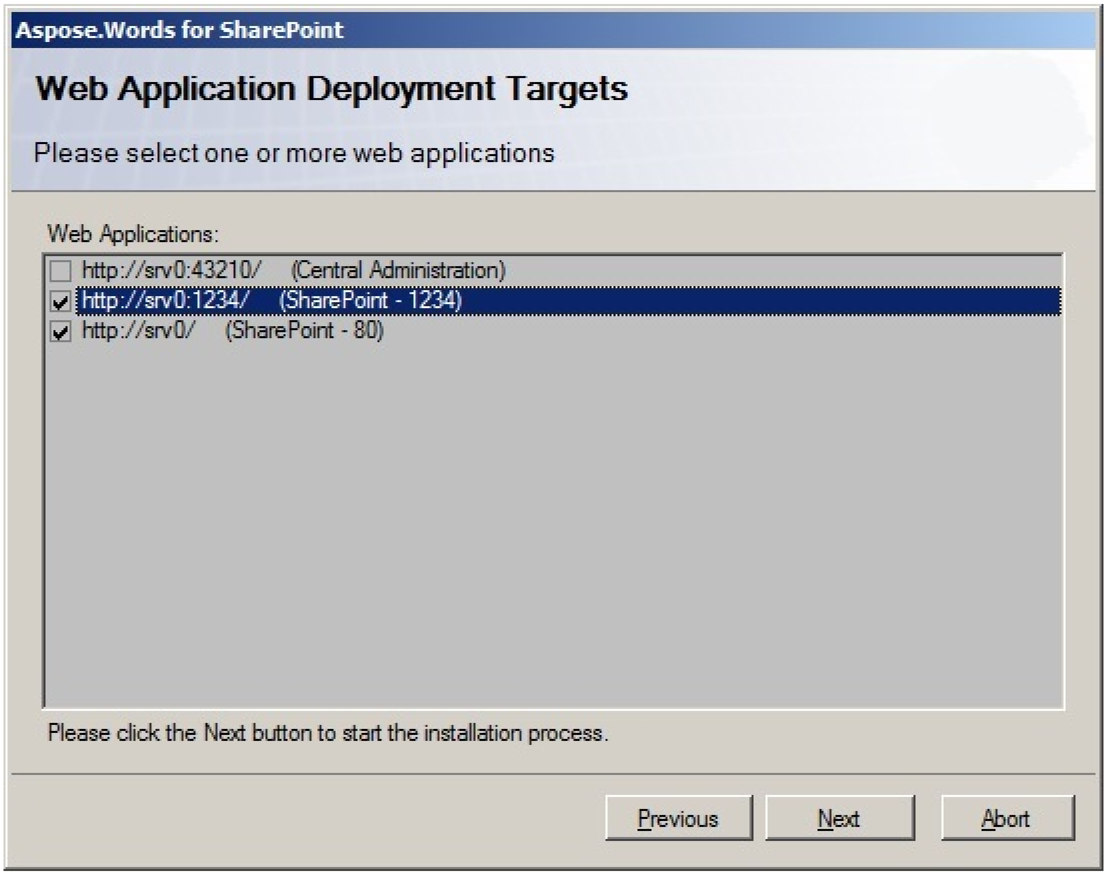
To install Aspose.Words for SharePoint:

- Unpack Aspose.Words.SharePoint.zip to the local drive on the SharePoint server.
- Run setup.exe and follow the instructions on the screen.

The setup program performs the following actions:

- Check installation prerequisites. Setup will not continue if any check fails.
- Display End User License Agreement. The user must accept the agreement in order to proceed.
- Display deployment target selection dialog. The user selects web applications where the feature shall be deployed and activated. See the figure below.
- Deploy the feature to the server farm.
- Activate the feature for the selected web applications and update their virtual directories.
- Display a list of web applications where the feature has been deployed and activated.

Deployment target selection dialog.



Aspose.Words for SharePoint

Web Application Deployment Targets

Please select one or more web applications

Web Applications:

☐ http://srv0:43210/ (Central Administration)
☑ http://srv0:1234/ (SharePoint - 1234)
☑ http://srv0/ (SharePoint - 80)

Please click the Next button to start the installation process.

More installation details

Deployment

Aspose.Words for SharePoint performs the following actions during deployment:

- Install Aspose.Words.SharePoint.dll into Global Assembly Cache and add SafeControl entry to the web.config file.
- Install feature manifest and other necessary files to the appropriate directories.
- Register the feature in the SharePoint database and make it available for the activation at the feature scope.

Activation

Aspose.Words for SharePoint is packaged as a web application level feature and can be activated and deactivated on the web application level.

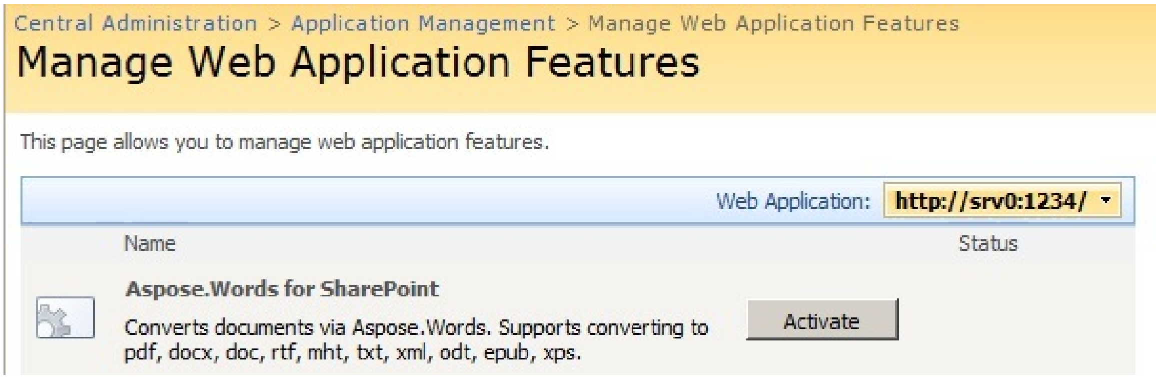
During activation, the feature makes some changes to the virtual directory of web application:

- Add authorized type entry for the workflow activities provided by Aspose.Words for SharePoint
- Add conversion settings page to the sitemap file
- Copy necessary resource files to the App_GlobalResources folder in the virtual directory

Activation and Deactivation after installation

During installation, Aspose.Words for SharePoint is activated for all selected web applications. After installation, SharePoint administrator can use Central Administration utility to activate and deactivate Aspose.Words for SharePoint. Use “Manage Web Application Features” item under “Application Management” to activate and deactivate Aspose.Words for SharePoint after installation.

Activating Aspose.Words for SharePoint on a web application.



Central Administration > Application Management > Manage Web Application Features

Manage Web Application Features

This page allows you to manage web application features.

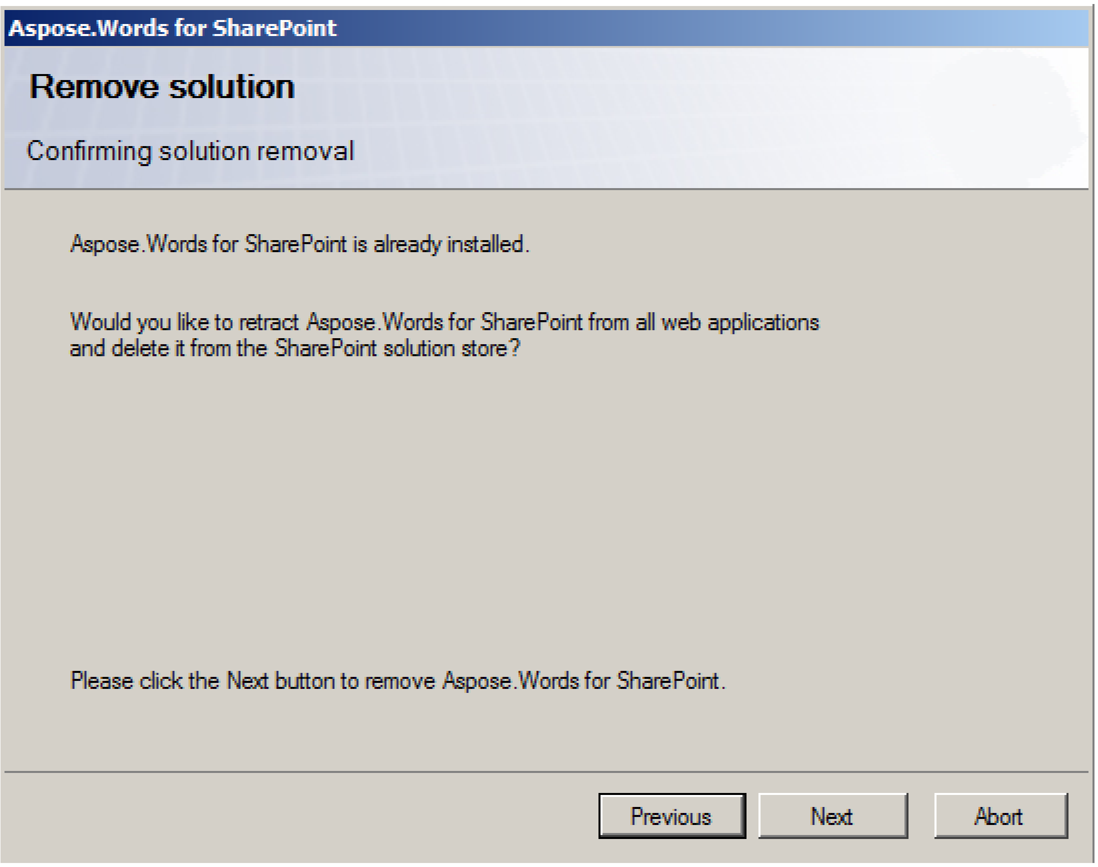
Web Application: http://srv0:1234/

Name	Status
Aspose.Words for SharePoint Converts documents via Aspose.Words. Supports converting to pdf, docx, doc, rtf, mht, txt, xml, odt, epub, xps.	

Uninstalling Aspose.Words for SharePoint

To uninstall Aspose.Words for SharePoint, just run the setup program. If Aspose.Words for SharePoint is already installed, the setup program suggests to remove it.

Choose “Remove” option to uninstall the feature.



Aspose.Words for SharePoint

Remove solution

Confirming solution removal

Aspose.Words for SharePoint is already installed.

Would you like to retract Aspose.Words for SharePoint from all web applications and delete it from the SharePoint solution store?

Please click the Next button to remove Aspose.Words for SharePoint.

During deinstallation, the setup program deactivates Aspose.Words for SharePoint feature for all web applications and retracts the solution from the server farm.

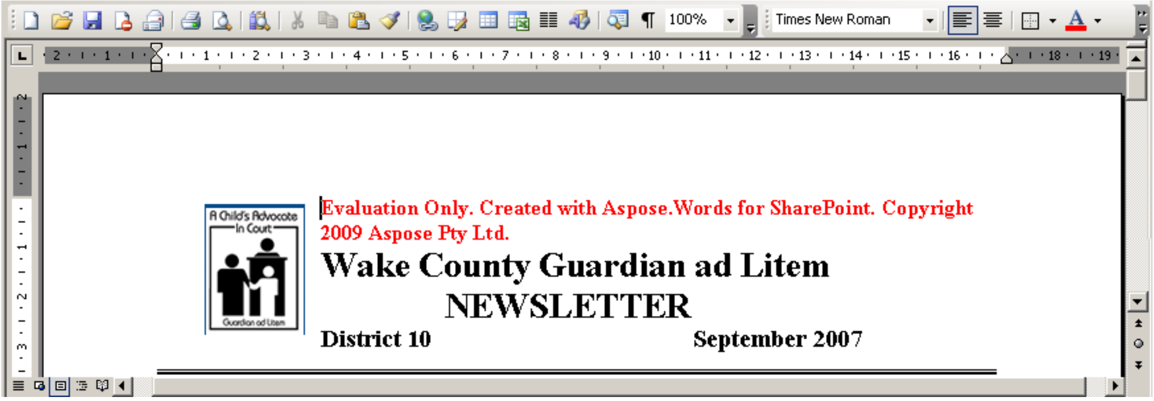
Evaluate Aspose.Words for SharePoint

Make sure to take advantage of the free Aspose.Words for SharePoint evaluation as it has no time limit, and free technical support is provided for evaluation users.

It is the same download for both the evaluation and the paid version of Aspose.Words for SharePoint. Simply download Aspose.Words for SharePoint from the download page, install it and it will work in the evaluation mode.

Evaluation mode injects watermarks into exported reports and limits them to several pages. When you have purchased a license, simply install the license solution over installed evaluation copy of Aspose.Words for SharePoint and it will then work in the licensed mode.

Aspose.Words for SharePoint injects a watermark when working in the evaluation mode.



Evaluation Only. Created with Aspose.Words for SharePoint. Copyright 2009 Aspose Pty Ltd.

Wake County Guardian ad Litem
NEWSLETTER

District 10 September 2007

License Aspose.Words for SharePoint

Installing Aspose.Words for SharePoint License

Once you are happy with your evaluation, you can purchase a license. Before purchasing make sure you understand and agree to the license subscription terms.

The license will be emailed to you after the order has been paid. The license is a .zip archive containing a regular SharePoint solution package.

This archive contains:

Aspose.Words.SharePoint.License.wsp	SharePoint solution package file. Aspose.Words for SharePoint License is packaged as a SharePoint solution to facilitate deployment/retraction across the server farm.
readme.txt	License installation instructions

License installation is performed from the server console via stsadm.exe. The steps required to install the license are below.

Note: The paths are omitted for clarity. You may need to add the actual path to stsadm.exe and/or solution file when executing them.

1. Run stsadm to add the solution to the SharePoint solution store:

```
stsadm.exe -o addsolution -filename Aspose.Words.SharePoint.License.wsp
```

2. Deploy the solution to all servers in the farm:

```
stsadm.exe -o deploysolution -name Aspose.Words.SharePoint.License.wsp -immediate -
force
```

3. Execute administrative timer jobs to complete the deployment immediately

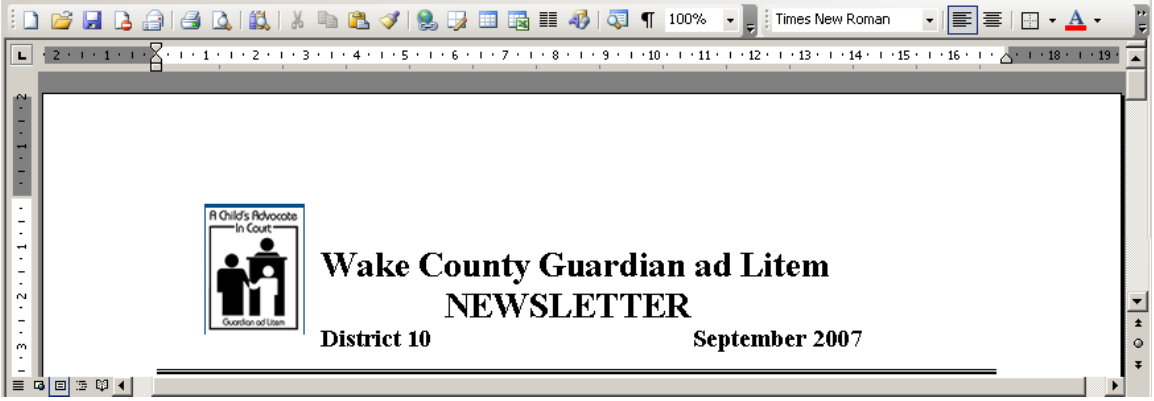
```
stsadm.exe -o execadmsvcjobs
```

Note: You will receive a warning when running deployment step if Windows SharePoint Services Administration service is not started. Stsadm.exe relies on this service and Windows SharePoint Timer Service to replicate solution data across the farm. If these services are not running on you server farm, you may need to deploy the license at each server.

Testing a License

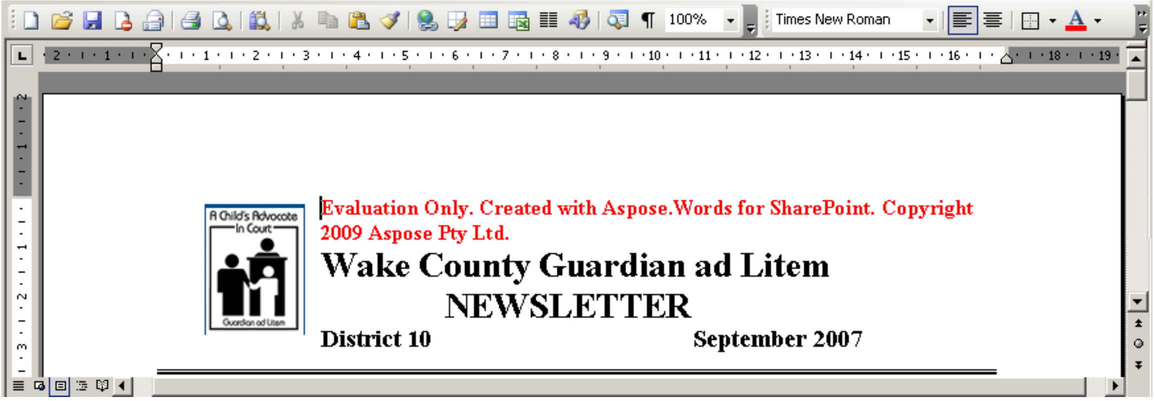
To test that a license has been installed correctly, convert any document into desired format. If the document contains no watermark at the top, the license was activated successfully.

When a valid Aspose.Words for SharePoint License has been installed correctly there will be no evaluation watermark.



Wake County Guardian ad Litem
NEWSLETTER
District 10 September 2007

If there is a problem with the license, Aspose.Words for SharePoint will work in evaluation mode and still include watermarks into the document.



Evaluation Only. Created with Aspose.Words for SharePoint. Copyright 2009 Aspose Pty Ltd.
Wake County Guardian ad Litem
NEWSLETTER
District 10 September 2007

Uninstalling Aspose.Words for SharePoint License

To uninstall Aspose.Words for SharePoint license, please use the steps below from the server console.

1. Retract the license solution from the farm:

```
stsadm.exe -o retractsolution -name Aspose.Words.SharePoint.License.wsp -immediate
```

2. Execute administrative timer jobs to complete the retraction immediately:

```
stsadm.exe -o execadmsvcjobs
```

3. Wait for the retraction to complete. You can use Central Administration to check if the retraction completed under Central Administration -> Operations -> Solution Management

4. Remove the solution from the SharePoint solution store:

```
stsadm.exe -o deletesolution -name Aspose.Words.SharePoint.License.wsp
```

User’s Guide

Reporting

About Reporting in Aspose.Words for SharePoint

With Aspose.Words for SharePoint, you can fill documents with data from external data sources such as databases, SharePoint lists or XML data files. You can save the result in any supported document format. We call this feature Reporting and the final document filled with data is a report.

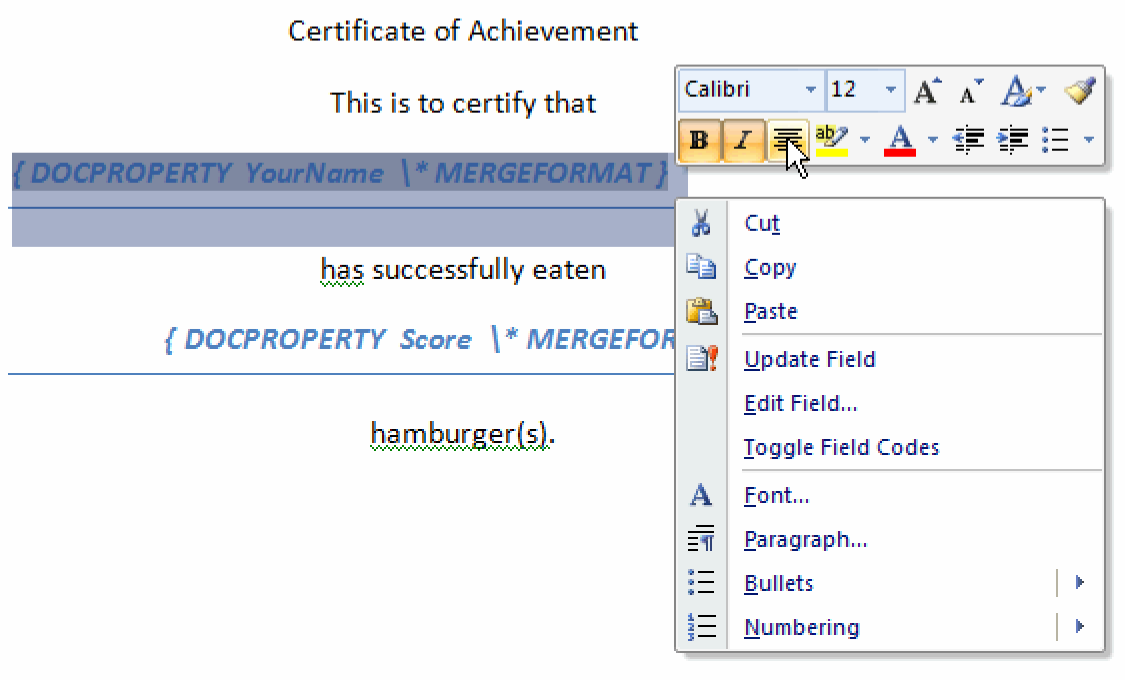
Reporting is built on top of Aspose.Words mail merge feature. However, Aspose.Words is a class library and you can use it only programmatically. Aspose.Words for SharePoint provides a user interface for making reports from templates stored in a SharePoint document library. You can specify saving format and location through the interface, as well as report parameters.

Report template is a Microsoft Word file with placeholders defining where to insert data from the report data source. Regular Microsoft Word mail merge fields serve as placeholders. However, Aspose.Words does much more than the standard Microsoft Word mail merge. You can:

- Define template regions to repeat for each data record in a data source.
- Use nested regions to reflect parent-child relations in the data.
- Insert images instead of merge fields.

Changing the visual report layout is very simple. Just use Microsoft Word to change the report template.

Change the report template layout in Microsoft Word.



Certificate of Achievement

This is to certify that

{ DOCPROPERTY YourName * MERGEFORMAT }

has successfully eaten

{ DOCPROPERTY Score * MERGEFOR

hamburger(s).

Report template also specifies where to find the data source definition. In Aspose.Words for SharePoint, data source definition is a separate XML file, which describes database connections and queries to retrieve the data. Currently the following data sources are supported:

- SQL queries from data sources supported by OLE DB data providers. For Microsoft SQL Server you can use .Net SQL data provider.
- SharePoint lists via CAML queries.
- XML data files located in a SharePoint document library.

There are two sets of tasks when using Aspose.Words for SharePoint reporting. A report user can

- Generate reports from templates and data source descriptions stored in SharePoint document libraries.
- Save the generated reports in any of the supported formats.
- Edit report template to change the layout using Microsoft Word.

A report designer does not need to write the code to get data from a data source. However, he needs to author an XML data source definition where he can

- Describe data source connection details.
- Describe queries to obtain data from a data source.
- Describe relationships between query results.
- Describe report parameters.

You can also build reports from templates programmatically with a few lines of code using Aspose.Words.SharePoint.Reporting namespace.

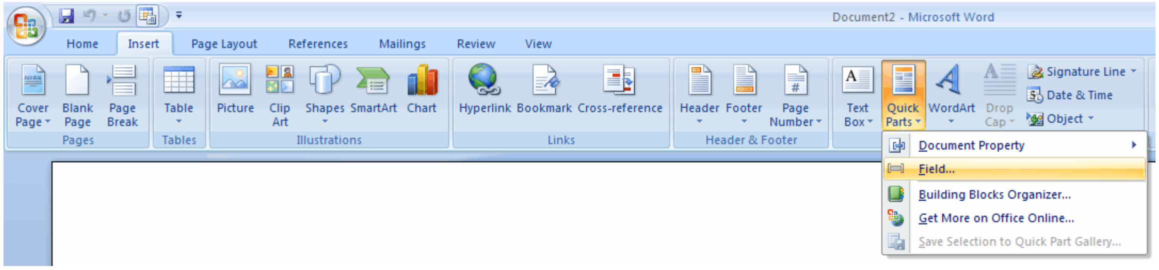
Report Template and Merge Fields

You need a template document before you can generate a report. A template is a Microsoft Word document with placeholders defining where to insert external data. It does not have to be a Microsoft Word template (.dot or .dotx) file. A regular .doc or .docx document will work.

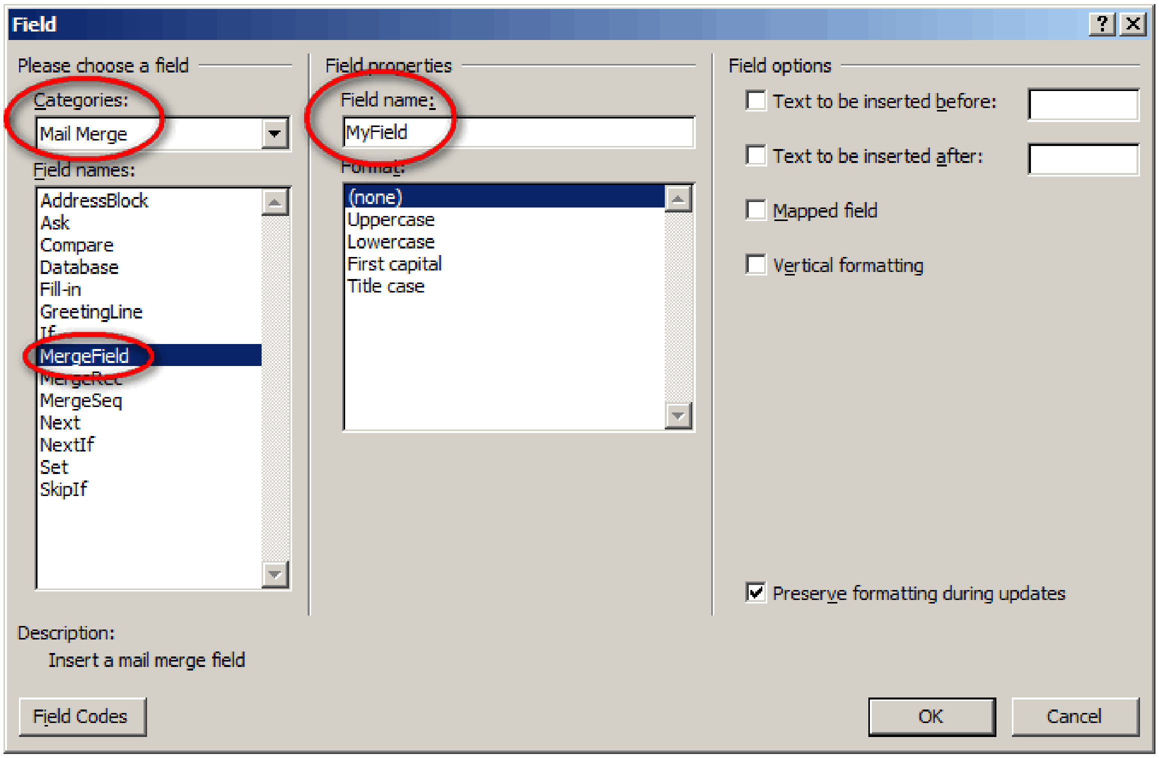
Mail merge fields serve as data placeholders. Field names should match data field names in the data source.

To insert a mail merge field in Microsoft Word:

1. Open your document in Microsoft Word.
2. Select the **Insert** tab of the Ribbon.



3. Select **Field...** from **Quick Parts** drop-down menu to open the **Field** dialog.
4. From the **Field names** list, select **MergeField**.
5. In the **Field name** text box, enter a name for the merge field and press **OK**.


Field

Please choose a field

Categories: Mail Merge

Field names:

- AddressBlock
- Ask
- Compare
- Database
- Fill-in
- GreetingLine
- If
- MergeField
- MergeRec
- MergeSeq
- Next
- NextIf
- Set
- SkipIf

Field properties

Field name: MyField

Format:

- (none)
- Uppercase
- Lowercase
- First capital
- Title case

Field options

□ Text to be inserted before:

□ Text to be inserted after:

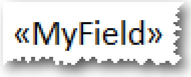
□ Mapped field

□ Vertical formatting

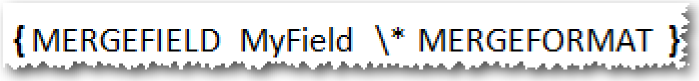
☑ Preserve formatting during updates

Description:
Insert a mail merge field

Now you have a new merge field placed in your document. Microsoft Word shows it like this:


«MyField»

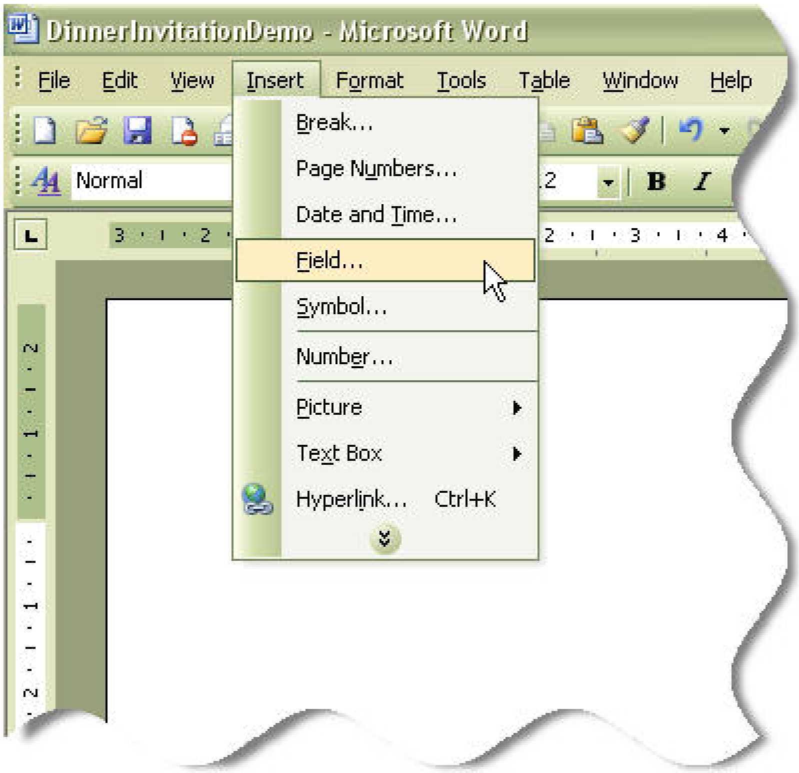
Of course, since a merge field is a regular Microsoft Word field, you can switch between displaying field codes and results in your document in Microsoft Word using the keyboard shortcut Alt+F9. Field codes appear between curly braces:



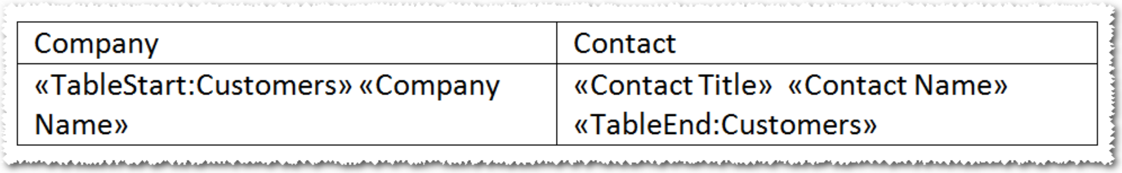
```
{MERGEFIELD  MyField  \* MERGEFORMAT }
```

You can edit merge field name and format switches if needed.

To open the **Field** dialog in Microsoft Word 2003, you need to open the **Insert** menu and select **Field...** item.



An example of a tabular template with multiple field codes.



Company	Contact
«TableStart:Customers» «Company Name»	«Contact Title» «Contact Name» «TableEnd:Customers»

Merge Regions

Every merge field in a report template must belong to a **region**. A region corresponds to a table in the report data source. A region also marks the part of the template, which is repeated for every data row in that table.

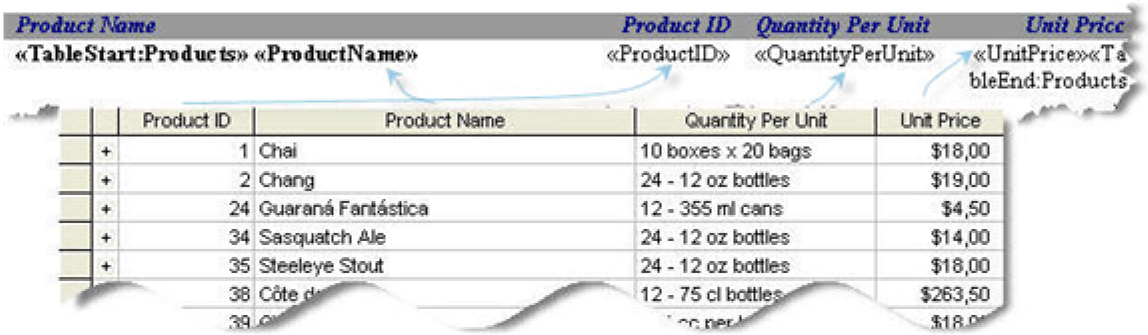
To specify a merge region in a document you need to insert two merge fields to mark the beginning and the end of the region.

To mark the beginning of a merge region, insert a merge field with the name *TableStart:MyTable*, where *MyTable* corresponds to the name of the table. To mark the end of the merge region insert another merge field with the name *TableEnd:MyTable*. Between these marking fields, place merge fields that correspond to the fields of your data source (table columns). These merge fields are populated with data from the first row of the data source, then the whole region is repeated, and the new fields are populated with data from the second row, and so on.

Follow these simple rules when marking a region:

- *TableStart* and *TableEnd* fields must be inside the same section in the document.
- If used inside a table, *TableStart* and *TableEnd* must be inside the same row in the table.
- You should always have a pair of matching *TableStart* and *TableEnd* fields with the same table name.
- You can nest one region inside the other. In this case, the inner region should end before the outer region ends. Use the **Relations** element to describe table relationships in report the data source definition.

Here is an example of a region in a report template:

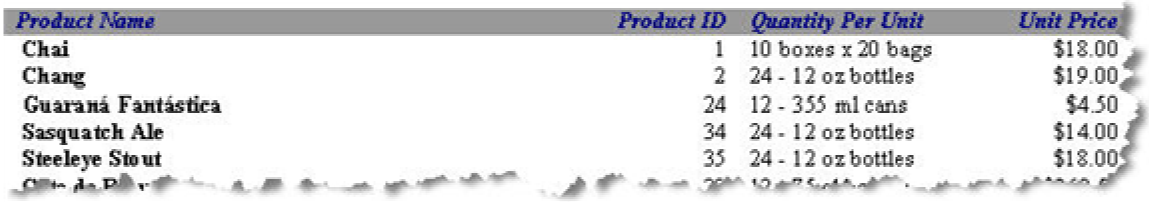


Product Name	Product ID	Quantity Per Unit	Unit Price
«TableStart:Products» «ProductName»	«ProductID»	«QuantityPerUnit»	«UnitPrice»«TableEnd:Products

	Product ID	Product Name	Quantity Per Unit	Unit Price
+	1	Chai	10 boxes x 20 bags	$18,00
+	2	Chang	24 - 12 oz bottles	$19,00
+	24	Guaraná Fantástica	12 - 355 ml cans	$4,50
+	34	Sasquatch Ale	24 - 12 oz bottles	$14,00
+	35	Steeleye Stout	24 - 12 oz bottles	$18,00
	38	Côte d	12 - 75 cl bottles	$263,50

You can see a merge region defined for populating with data from the Products table. Note that both the marking fields TableStart:Products and TableEnd:Products are placed inside the same row of the Word table.

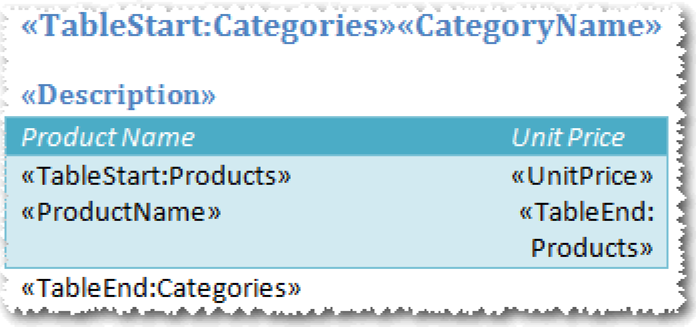
After the report is filled with data, here is the result:



Product Name	Product ID	Quantity Per Unit	Unit Price
Chai	1	10 boxes x 20 bags	$18.00
Chang	2	24 - 12 oz bottles	$19.00
Guaraná Fantástica	24	12 - 355 ml cans	$4.50
Sasquatch Ale	34	24 - 12 oz bottles	$14.00
Steeleye Stout	35	24 - 12 oz bottles	$18.00

Nested Regions

You can use nested regions when there is a relationship between data tables in the data source. Here is an example of a report template for listing product by categories. There are two tables: **Products** and **Categories** linked by **CategoryID** field. The outer region is for category description and the inner region lists products for each category:



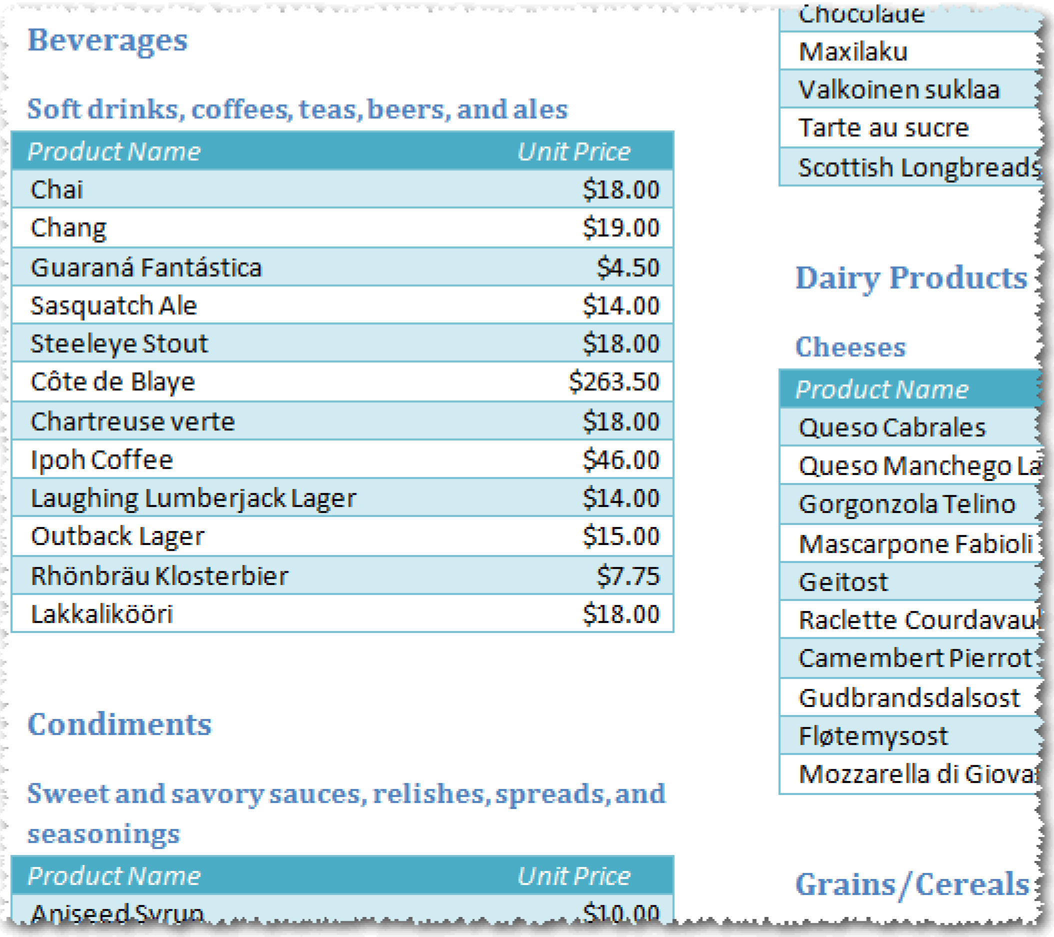
«TableStart:Categories»«CategoryName»

«Description»

Product Name	*Unit Price*
«TableStart:Products» «ProductName»	«UnitPrice» «TableEnd: Products»

«TableEnd:Categories»

The result is products grouped by categories:



Beverages

Soft drinks, coffees, teas, beers, and ales

Product Name	*Unit Price*
Chai	$18.00
Chang	$19.00
Guaraná Fantástica	$4.50
Sasquatch Ale	$14.00
Steeleye Stout	$18.00
Côte de Blaye	$263.50
Chartreuse verte	$18.00
Ipoh Coffee	$46.00
Laughing Lumberjack Lager	$14.00
Outback Lager	$15.00
Rhönbräu Klosterbier	$7.75
Lakkalikööri	$18.00

Condiments

Sweet and savory sauces, relishes, spreads, and seasonings

Product Name	*Unit Price*
Aniseed Syrup	$10.00

Chocolade
Maxilaku
Valkoinen suklaa
Tarte au sucre
Scottish Longbreads

Dairy Products

Cheeses

Product Name
Queso Cabrales
Queso Manchego La
Gorgonzola Telino
Mascarpone Fabioli
Geitost
Raclette Courdavau
Camembert Pierrot
Gudbrandsdalsost
Fløtemysost
Mozzarella di Giova

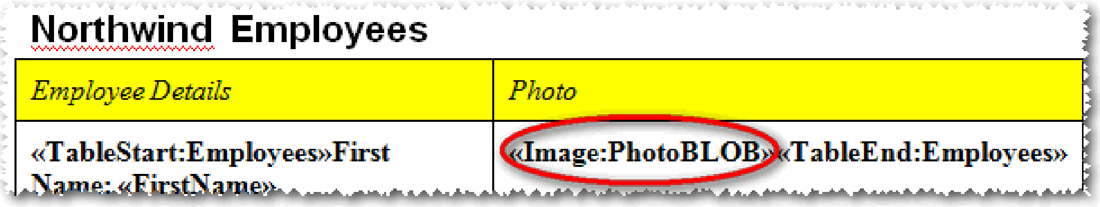
Grains/Cereals

See how to define related tables in a report data source in the **TableRelation** element description.

Merge Fields for Inserting Images

You can insert images from a data source in the report. Image merge fields must have the names starting with **Image:** in the report template. The name part after colon should match data source field (table column) with the image data, just like the names of the other merge fields.

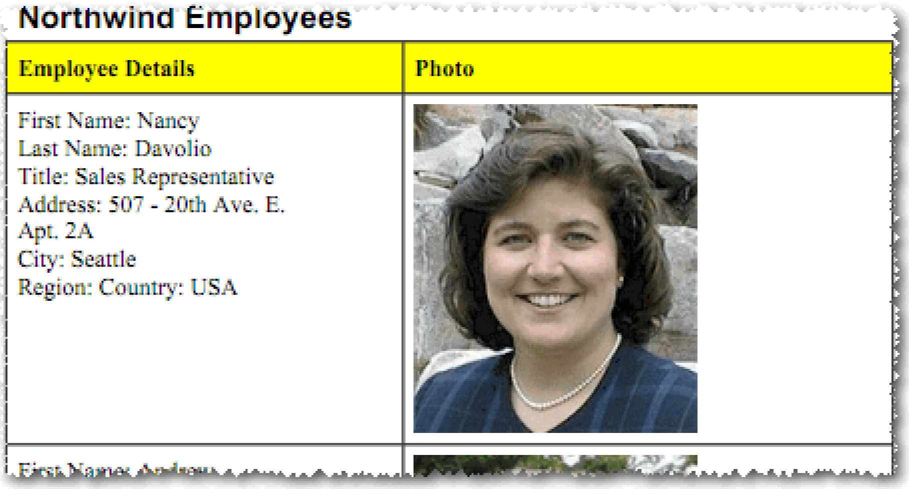
A report template with an *Image:* merge field.



Northwind Employees

Employee Details	*Photo*
«TableStart:Employees»First Name: «FirstName»	«Image:PhotoBLOB»«TableEnd:Employees»

An employee photo is inserted in the report instead of the *Image:* merge field.



Northwind Employees

Employee Details	Photo
First Name: Nancy Last Name: Davolio Title: Sales Representative Address: 507 - 20th Ave. E. Apt. 2A City: Seattle Region: Country: USA	
First Name: Andrew	

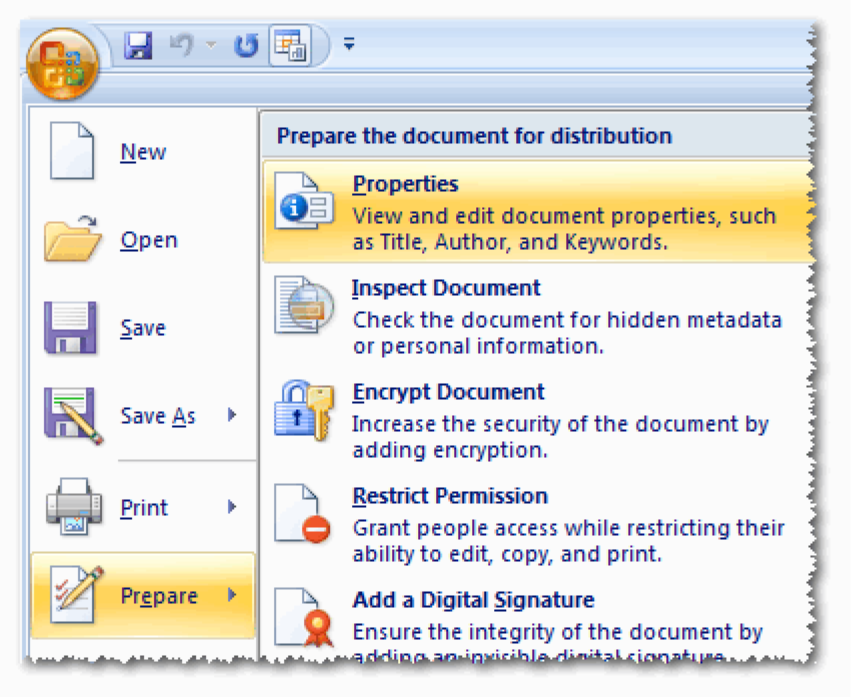
Linking Report Template with Report Definition

A report template does not have a description of data source connection inside. Instead, there is a link to a report definition file stored in a SharePoint document library. Report definition is an XML file, which describes things like data source connection details, queries to retrieve data, report parameters, data relationships and so on.

You should store the link to the report definition in a custom document property named **Aspose.Words.SharePoint.ReportUrl**. To edit this property, follow these steps for Microsoft Word 2007 (see below for the slightly different steps in Microsoft Word 2010):

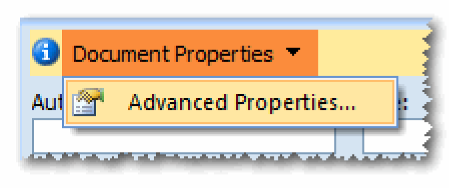
1. Open a report template in Microsoft Word.

2. Select **Prepare** and **Properties** from the **Office** Button menu.



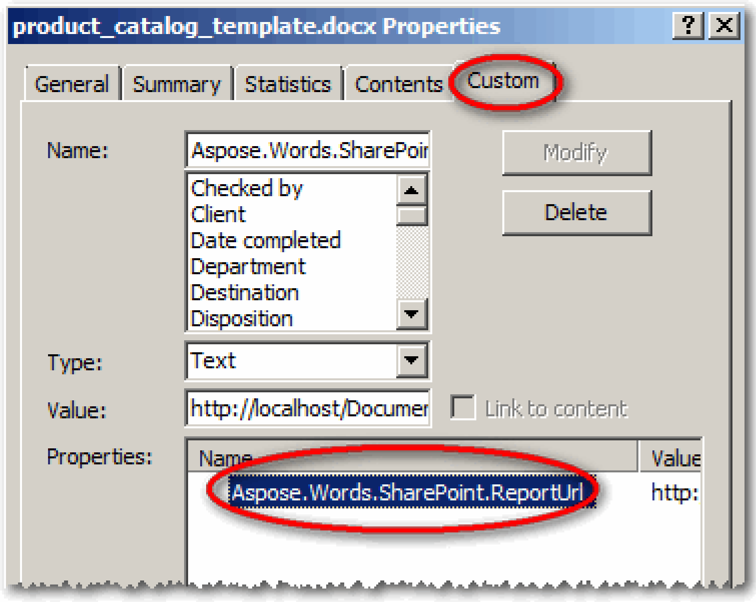
New
Open
Save
Save As
Print
Prepare
Prepare the document for distribution
Properties
View and edit document properties, such as Title, Author, and Keywords.
Inspect Document
Check the document for hidden metadata or personal information.
Encrypt Document
Increase the security of the document by adding encryption.
Restrict Permission
Grant people access while restricting their ability to edit, copy, and print.
Add a Digital Signature
Ensure the integrity of the document by adding an invisible digital signature.

3. Select **Advanced Properties** from the **Document Properties** drop down menu to open the document properties dialog.



4. In the document properties dialog, select **Custom** tab.

5. Select **Aspose.Words.SharePoint.ReportUrl** property from the properties list.



Name: Aspose.Words.SharePoir

Checked by
Client
Date completed
Department
Destination
Disposition

Type: Text

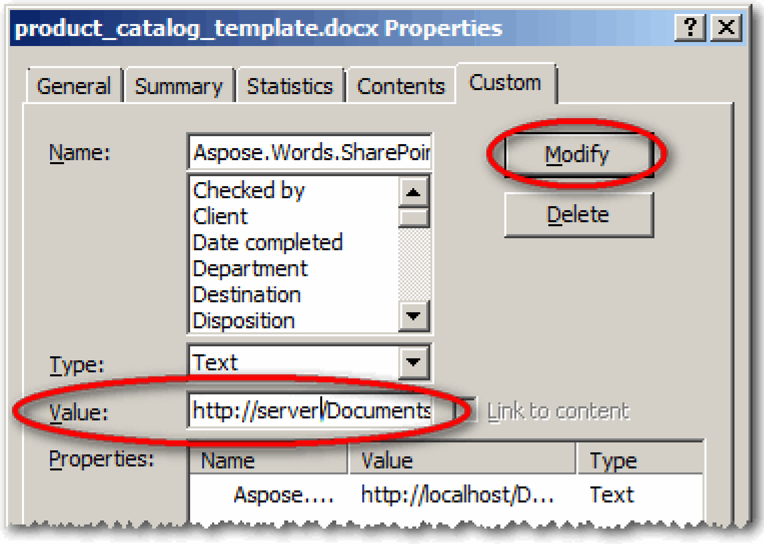
Value: http://localhost/Documer

☐ Link to content

Properties:

Name	Value
Aspose.Words.SharePoint.ReportUrl	http:

6. Edit the report definition URL in the in the **Value** field. On changing the value, **Modify** button becomes enabled. Press it to save the changes.



Name: Aspose.Words.SharePoir

Checked by
Client
Date completed
Department
Destination
Disposition

Type: Text

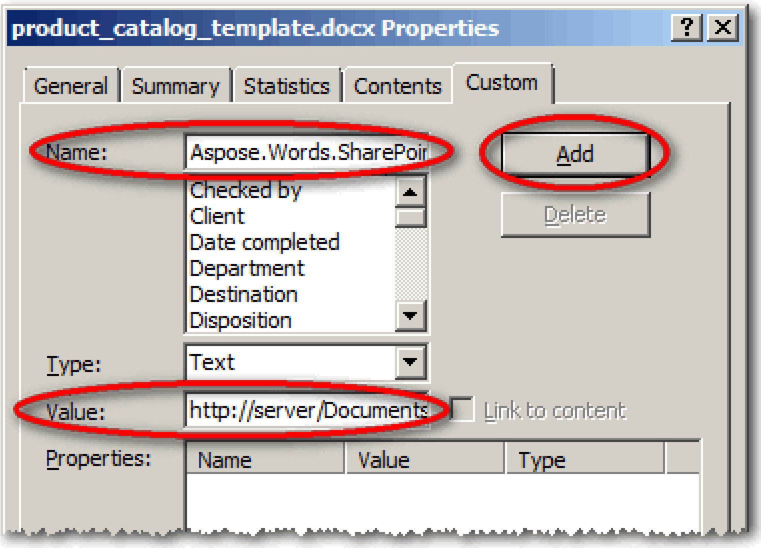
Value: http://server/Documents

☐ Link to content

Properties:

Name	Value	Type
Aspose....	http://localhost/D...	Text

If there is no **Aspose.Words.SharePoint.ReportUrl** property, you can add it in the same dialog by typing the name in the **Name** field, editing **Value** and then pressing **Add** button.



Name: Aspose.Words.SharePoir

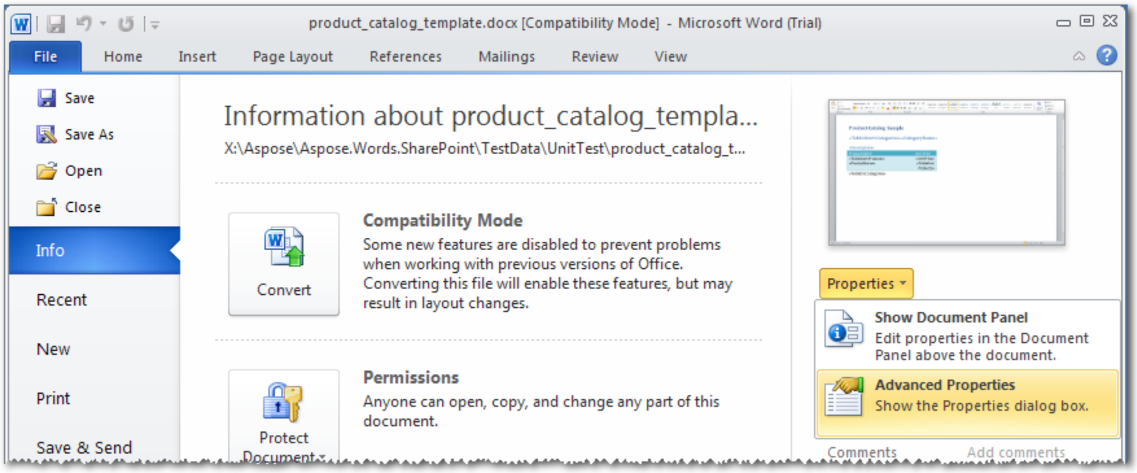
Checked by
Client
Date completed
Department
Destination
Disposition

Type: Text

Value: http://server/Documents

The steps to get to the document properties dialog are slightly different for Microsoft Word 2010.

1. Select the **File** tab and then **Advanced Properties** from the **Properties** drop down menu on the **Info** tab.



Information about product_catalog_templa...

X:\Aspose\Aspose.Words.SharePoint\TestData\UnitTest\product_catalog_t...

Compatibility Mode

Some new features are disabled to prevent problems when working with previous versions of Office. Converting this file will enable these features, but may result in layout changes.

Permissions

Anyone can open, copy, and change any part of this document.

Show Document Panel

Edit properties in the Document Panel above the document.

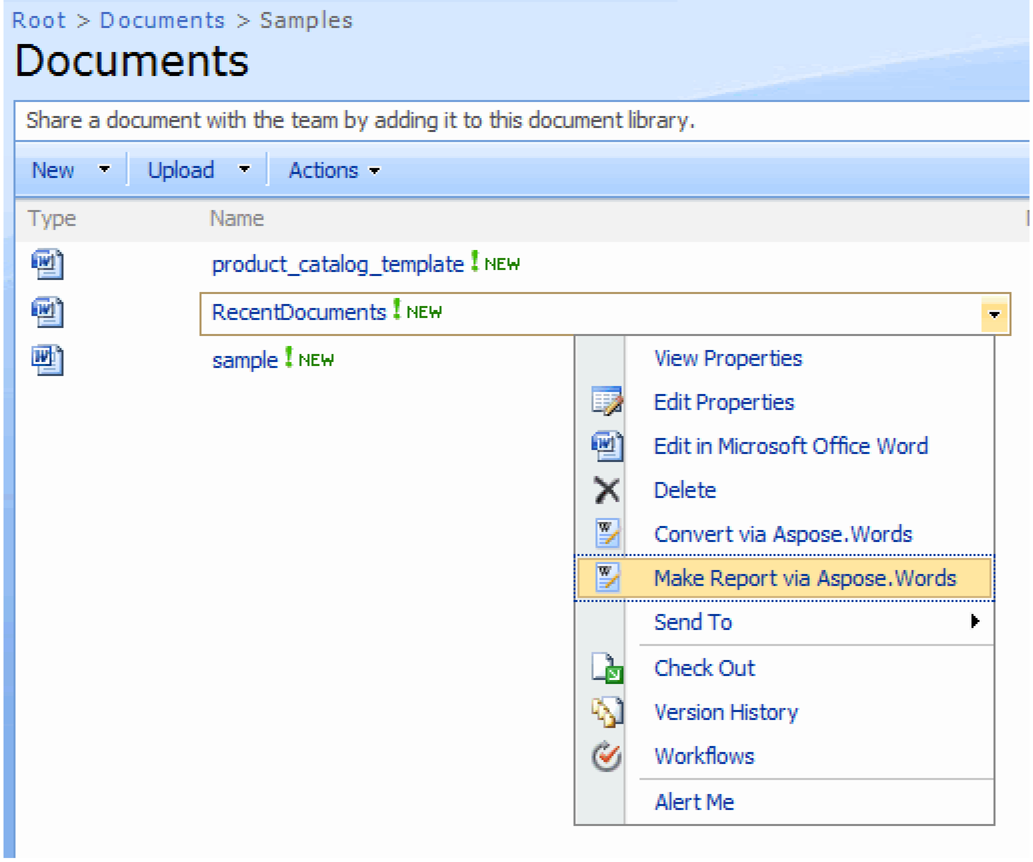
Advanced Properties

Show the Properties dialog box.

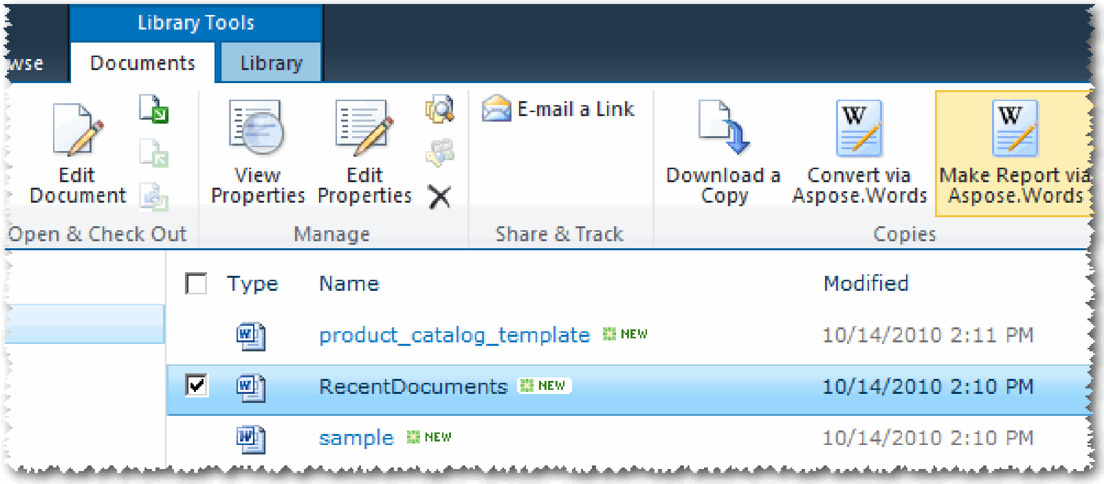
2. This will produce the document properties dialog. Follow the steps for Word 2007 above starting from step 4.

User Interface

To make a report, select a report template in a SharePoint document library and click **"Make Report via Aspose.Words"** item in the template's Edit Control Block.


Root > Documents > Samples
Documents
Share a document with the team by adding it to this document library.
New
Upload
Actions
Type
Name
product_catalog_template NEW
RecentDocuments NEW
sample NEW
View Properties
Edit Properties
Edit in Microsoft Office Word
Delete
Convert via Aspose.Words
Make Report via Aspose.Words
Send To
Check Out
Version History
Workflows
Alert Me

In SharePoint 2010, you can use **"Make Report via Aspose.Words"** button on the ribbon as well. The button is active when you select a single file.



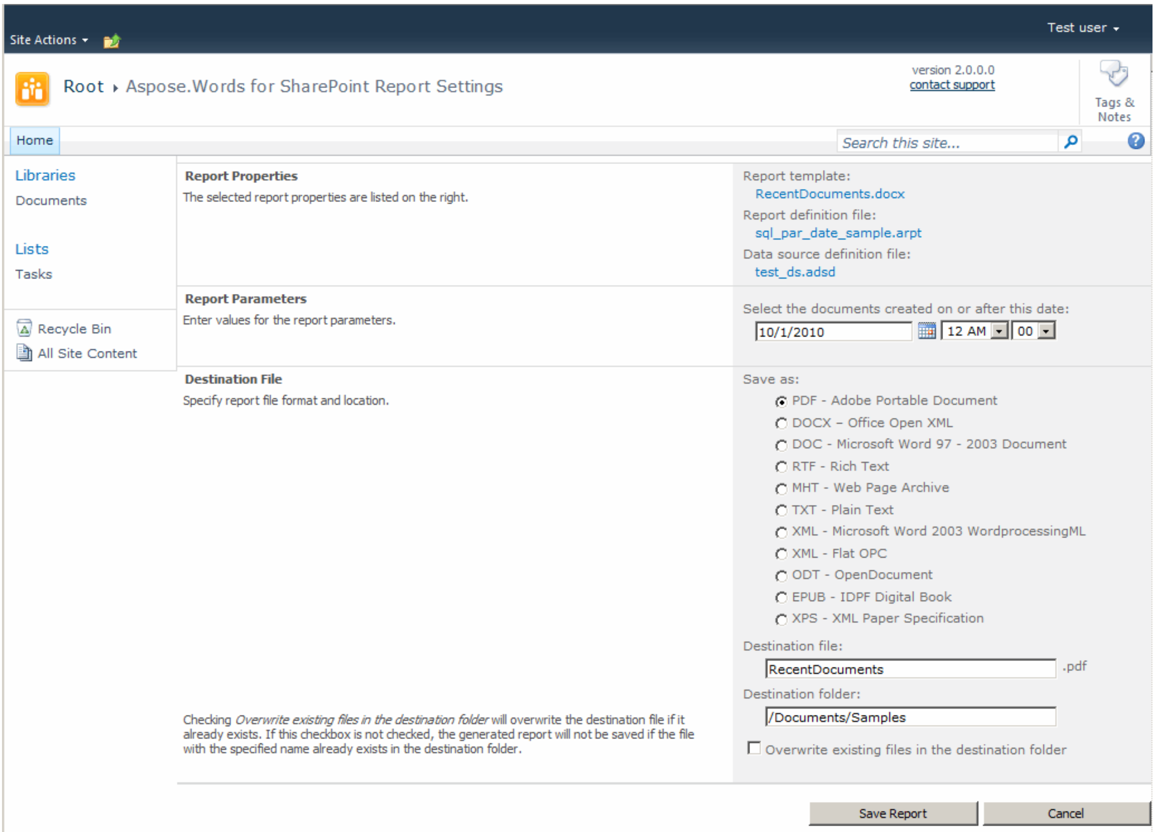
☐	Type	Name	Modified
		product_catalog_template NEW	10/14/2010 2:11 PM
☑		RecentDocuments NEW	10/14/2010 2:10 PM
		sample NEW	10/14/2010 2:10 PM

This will lead you to the report settings screen. The following options are available:

- If the report has parameters, **"Parameters"** section is present. Enter the required parameter values of the appropriate type.
- Select the destination format with the **"Save as"** radio button list.

- Enter the destination file name in the “**Destination file**” edit box. By default, Aspose.Words for SharePoint prepopulates this field with the name of the report template file. The extension is added automatically depending on the selected format.
- Enter the URL of the destination library folder in the “**Destination folder**” edit box. It can be a folder in a different site collection. The default value is where the report template is located.
- Check the **“Overwrite existing files in the destination folder”** checkbox to replace an existing file with the report in case of name clash. By default, existing files are not overwritten. If this box is checked:
    - When the destination library has versioning enabled, a new file version is added for a conflicting file.
    - When versioning is not enabled for the destination library, an existing file with a conflicting name is overwritten.
- Click “**Save Report**” to build and save the report.
- Click “**Cancel**” button to go back to the document library.

Report settings screen.



Report Properties
The selected report properties are listed on the right.

Report template:
RecentDocuments.docx
Report definition file:
sql_par_date_sample.arpt
Data source definition file:
test_ds.adsd

Report Parameters
Enter values for the report parameters.

Select the documents created on or after this date:
10/1/2010 | 12 AM | 00

Destination File
Specify report file format and location.

Save as:
- ● PDF - Adobe Portable Document
- ○ DOCX – Office Open XML
- ○ DOC - Microsoft Word 97 - 2003 Document
- ○ RTF - Rich Text
- ○ MHT - Web Page Archive
- ○ TXT - Plain Text
- ○ XML - Microsoft Word 2003 WordprocessingML
- ○ XML - Flat OPC
- ○ ODT - OpenDocument
- ○ EPUB - IDPF Digital Book
- ○ XPS - XML Paper Specification

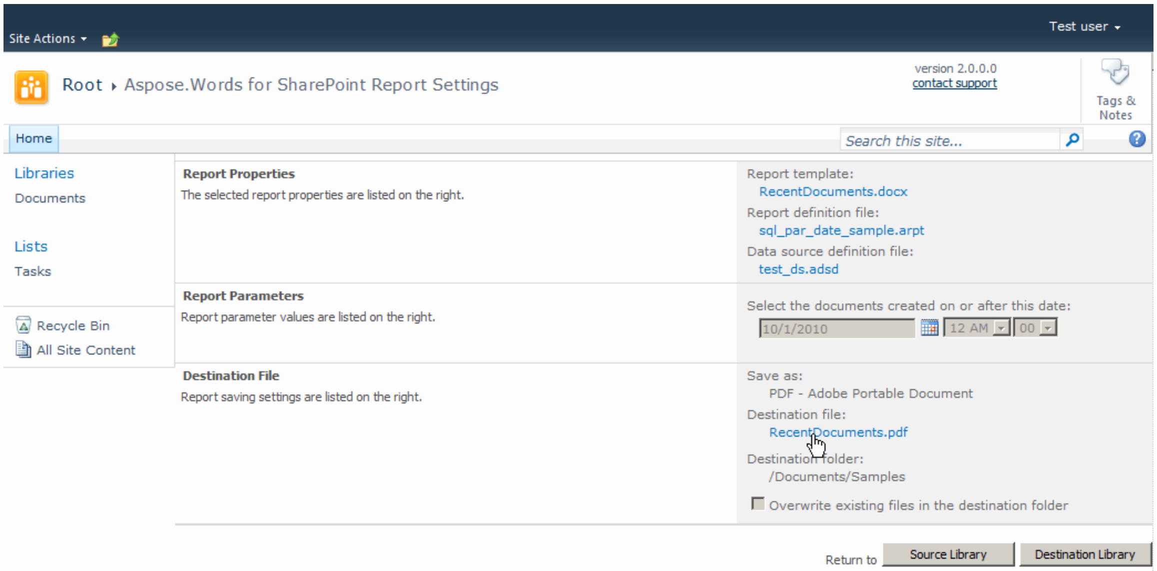
Destination file:
RecentDocuments .pdf

Destination folder:
/Documents/Samples

Checking *Overwrite existing files in the destination folder* will overwrite the destination file if it already exists. If this checkbox is not checked, the generated report will not be saved if the file with the specified name already exists in the destination folder.

□ Overwrite existing files in the destination folder

After saving the report, another screen is presented. It lists the options and parameters used. You can return to the library where the report template is located, go to the destination library or use the hyperlink to open the saved report.



Report Properties
The selected report properties are listed on the right.

Report template:
RecentDocuments.docx
Report definition file:
sql_par_date_sample.arpt
Data source definition file:
test_ds.adsd

Report Parameters
Report parameter values are listed on the right.

Select the documents created on or after this date:
10/1/2010 | 12 AM | 00

Destination File
Report saving settings are listed on the right.

Save as:
PDF - Adobe Portable Document
Destination file:
RecentDocuments.pdf
Destination folder:
/Documents/Samples

□ Overwrite existing files in the destination folder

This documentation was generated using Aspose.Words for SharePoint.

Report Definition File

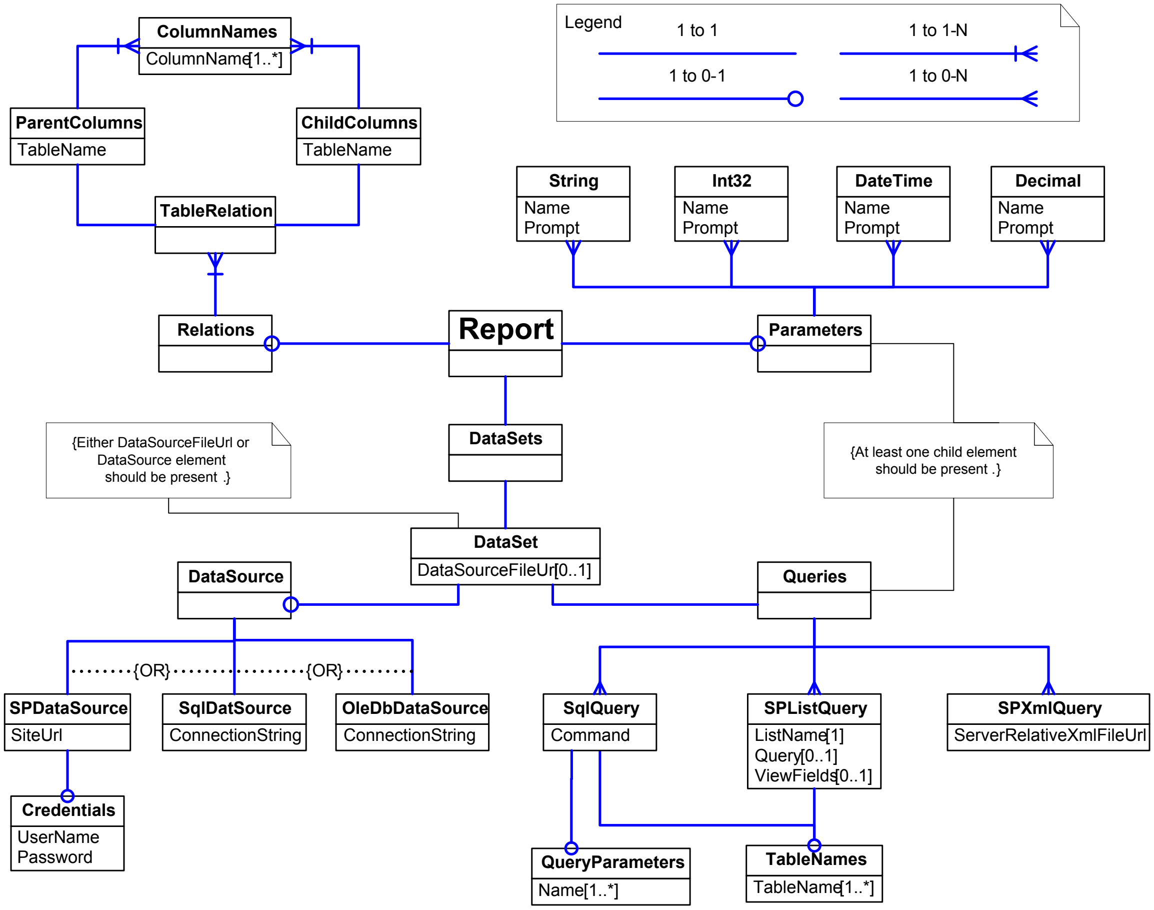
Report definition file is an XML file that describes all about report that report template does not do. That is, it defines everything except the visual report layout. This includes:

- Data source connection properties.
- Queries to retrieve the data.
- Relationships between data tables returned by different queries.
- Report parameters.

A report template must have a link to the report definition file.

If you are designing report for Aspose.Words for SharePoint, you need to create the report definition file. Currently, there is no tool for report definition editing, so you have to edit raw XML. You can look at report definition file as a configuration file where you specify all the details of data retrieving rather than coding them with Aspose.Words API. As far as you do not need a fine level of control that Aspose.Words class library offers, Aspose.Words for SharePoint may be more suitable.

Report definition schema diagram.



Look at the report definition diagram. The main section is **DataSets** that describes data source connection details and queries to retrieve the data. A report can combine data from several queries and have several data source connection. For example, it is possible to get a query result from an SQL database and combines it with list items from a SharePoint list. If the data returned by different queries are related, you describe it in the **Relations** section. The **Parameters** section is for report parameters description.

All elements used in the report definition are defined in *http://www.aspose.com/Words/SharePoint/Reporting* XML namespace.

Either the elements have child elements or a value stored in as text element content. Elements having child elements do not use text content in the report definition.

When the element has text content as a value, leading and trailing whitespace is trimmed from the text before usage. Consider this element:

```
<TableName>MyTable</TableName>
```

It has the same value ("MyTable") as the element below.

```
<TableName>
       MyTable
</TableName>
```

Report definition schema file (Report.xsd) and data source definition schema file (DataSource.xsd) are included in the installation package for you reference.

Report Element

The **Report** element is the root element of the report definition. It can have the following child elements:

Name	***Cardinality***	***Has Children***	***Description***
DataSets	1	Yes	Describes connections to data sources and queries to obtain the data for this report.
Relations	0-1	Yes	Describes relations between data tables returned by different queries.
Parameters	0-1	Yes	Report parameters.

DataSets Element

The **DataSets** element contains information about connections to the data sources and queries to retrieve the data for the report.

Name	***Cardinality***	***Has Children***	***Description***
DataSet	1-N	Yes	Describes a connection to a data source and queries to get the data from this data source.

DataSet Element

The **DataSet** element contains information about a connection to a data source and queries to get the data for the report from this data source.

You describe data source connection properties inside **DataSource** element.

Alternatively, you can store data source connection description in a separate XML file on the same SharePoint farm. This way you can share the same data source connection description between many reports, and update it only once when needed. The **DataSourceFileUrl** element points to the data source description file. The schema of a separate data source description is a subset of the report definition schema with the **DataSource** element at the root level. Data source definition schema file (DataSource.xsd) is included in the installation package for you reference.

Name	***Cardinality***	***Has Children***	***Description***
DataSource	0-1	Yes	Describes a connection to a data source. Either this element or DataSourceFileUrl must be present.
DataSourceFileUrl	0-1	No	Data source description file URL. Either this element or DataSourceFileUrl must be present.

Queries	1	Yes	Queries to get the report data from the specified data source.

DataSource Element

The **DataSource** element contains information about a connection to a data source.

Only one child element from the table below is allowed. The child element defines the data source connection type.

Name	***Cardinality***	***Has Children***	***Description***
SqlDataSource	1	Yes	Describes a connection to a Microsoft SQL Server database.
OleDbDataSource	1	Yes	Describes a connection to a data source via OLE DB provider.
SPDataSource	1	Yes	Describes a connection to a SharePoint server.

SqlDataSource Element

This element contains information about a connection to a Microsoft SQL server database. .Net SqlClient is used for the connection. You can use it only with **SqlQuery** elements to define queries.

Please note that storing database user name and password in the connection string may be insecure.

Name	***Cardinality***	***Has Children***	***Description***
ConnectionString	1	No	Data source connection string. It must not be empty.

OleDbDataSource Element

This element contains information about a connection to a relational database via OLEDB provider. You can use it only with **SqlQuery** elements to define queries.

Please note that storing database user name and password in the connection string may be insecure.

Name	***Cardinality***	***Has Children***	***Description***
ConnectionString	1	No	Data source connection string. It must not be empty.

SPDataSource Element

This element contains information about a connection to a SharePoint server. You can use it with **SPListQuery** or **SPXmlQuery** to get data from a SharePoint server.

Name	***Cardinality***	***Has Children***	***Description***
SiteUrl	1	No	Full SharePoint site URL, including *http://server:port* part. It must not be empty.
Credentials	0-1	Yes	Defines user name and password to use for connecting to a SharePoint server. Omit it to use the current user credentials.

Credentials Element

The **Credentials** element contains information about user name and password to use for connecting to a SharePoint site. You can omit it to use the current user credentials.

Please note that storing user name and password in a report definition file may be insecure.

Name	***Cardinality***	***Has Children***	***Description***
UserName	1	No	User name for connecting to a SharePoint server.
Password	1	No	Password for connecting to a SharePoint server.

Queries Element

The **Queries** element contains information about queries for retrieving report data from the data source. It must contain at least one of the elements from the table below. Use **SqlQuery** with **SqlDataSource** or **OleDbDataSource**. Use **SPListQuery** or **SPXmlQuery** with **SPDataSource**. Other combinations are not supported.

At least one of the child elements should be present.

Name	***Cardinality***	***Has Children***	***Description***
SqlQuery	0-N	Yes	Describes an SQL query to a relational database.
SPListQuery	0-N	Yes	Describes a CAML query to a SharePoint list.
SPXmlQuery	0-N	Yes	Describes XML data file location in a SharePoint document library.

SqlQuery Element

The **SqlQuery** element contains information about an SQL query for retrieving report data from a relational database. Use it with **SqlDataSource** or **OleDbDataSource**.

Command can include parameter placeholders. However, the placeholder syntax is different for **SqlDataSource** and **OleDbDataSource.**

For **SqlDataSource** query, use parameter names starting with "**@".** Parameter values are substituted by name. Parameter names must match report parameters described in the **Parameters** element. See how to use parameter placeholders for **SqlDataSource** on MSDN.

For **OleDbDataSource**, use "**?**" as a parameter placeholder. Parameters are substituted in the order they are listed in the **QueryParameters** element or in **Parameters** element if **QueryParameters** is missing. See how to use parameter placeholders for **OleDbDataSource** on MSDN.

Name	***Cardinality***	***Has Children***	***Description***
TableNames	0-1	Yes	Describes table names to assign to the tables returned by the query.
QueryParameters	0-1	Yes	Sets the order of report parameters used in the query.
Command	1	No	SQL command to get report data from the data source. It must not be empty.

TableNames Element

The **TableNames** element describes how to name the tables returned by its parent query. Use it to match queries with report regions designated by **TableStart:** and **TableEnd:** merge fields in the report template.

Names are assigned to the tables returned by the query in the order they go inside the **TableNames** element. However, normally one query returns only one table.

Name	***Cardinality***	***Has Children***	***Description***

TableName	1-N	No	Table name. It must not be empty.

QueryParameters Element

The **QueryParameters** element set the order in which parameters are substituted in a query. It is useful for queries to OleDbDataSource, because they substitute parameters by order.

Use parameter names defined in the **Name** element inside **Parameter** element.

Name	***Cardinality***	***Has Children***	***Description***
Name	1-N	No	Parameter name. It must match a parameter name defined in the **Parameters** element.

SPListQuery Element

The **SPListQuery** element contains information about CAML query for retrieving report data from a SharePoint list. Use it with **SPDataSource**.

Use the **ListName** element to set the name of the list to query. Use the **SiteUrl** element of the data source to specify subsite if needed.

Set CAML query via the **Query** element. It is best to wrap the element contents in ![CDATA[]] for readability as CAML query is in XML. The same is true for the **ViewFields** element.

In a CAML query, parameters are substituted by name. An entire element inside the query must match a parameter name listed in the **Parameters** element.

If CAML query is not specified, all elements in the list are retrieved.

CAML queries are executed via Lists.asmx web service so the returned field names are often prefixed with "ows_". See CAML query schema description on MSDN.

The queries are executed recursively, traversing subfolders of the list.

Please note that a bunch of fields is always returned regardless of the **ViewFields** element. This is the behavior of the SharePoint Lists web service.

Name	***Cardinality***	***Has Children***	***Description***
TableNames	0-1	Yes	Describes table names to assign to the tables returned by the query.
ListName	1	No	The name of the list from where to retrieve data. It must not be empty.
Query	0-1	No	CAML query to get filter and order the list items.
ViewFields	0-1	No	Describes which columns to get from the list.

Here is an example of the report with the **SPListQuery** element:

```
<Report xmlns="http://www.aspose.com/Words/SharePoint/Reporting" >
  <DataSets>
    <DataSet>
      <DataSource>
        <SPDataSource>
          <SiteUrl>http://localhost</SiteUrl>
        </SPDataSource>
      </DataSource>
      <Queries>
        <SPListQuery>
          <TableNames>
            <TableName>Docs</TableName>
          </TableNames>
          <ListName>Documents</ListName>
          <Query><![CDATA[
            <Where>
              <Contains>
                <FieldRef Name='FileLeafRef'/>
                <Value Type='Text'>s</Value>
              </Contains>
            </Where>
          ]]></Query>
          <ViewFields><![CDATA[
            <FieldRef Name='FileLeafRef' />
            <FieldRef Name='ID' />
          ]]></ViewFields>
        </SPListQuery>
      </Queries>
    </DataSet>
  </DataSets>
</Report>
```

Note that internal field names like **FileLeafRef** are used inside the query. However, Lists web service returns, **FileLeafRef** field as **ows_FileLeafRef**. And it returns about 15 fields instead of just two mentioned in the **ViewFields** element.

SPXmlQuery Element

The **SPListQuery** element contains information an XML data file. This is not actually a query. It just specifies the location of an XML data file via server relative URL. The data file is just converted to a data set by using the.Net System.Data.DataSet.ReadXml() method with inferred schema

The **TableNames** element is allowed inside **SPXmlQuery** by the schema but will be ignored when retrieving data.

There is no way to use parameters with **SPXmlQuery**.

The XML data file referenced by the query is retrieved via SharePoint object model. Currently there is no way to get a file from a SharePoint farm other than the current farm with **SPXmlQuery**.

The **Credentials** element of the **SPDataSource** element is ignored when getting data via **SPXMLQuery**. The current user context is always used.

Name	***Cardinality***	***Has Children***	***Description***
ServerRelativeXmlFileUrl	1	No	Server relative URL of an XML data file.

Relations Element

The **Relations** element contains information about table relations in the report data. Each query returns on or more data tables in a data set. This data set is merged with data sets returned by the other queries. In order to make the nested regions in the report template to work, you have to specify the relations between tables inside the report data set.

Name	***Cardinality***	***Has Children***	***Description***
TableRelation	1-N	Yes	Describes a relation between two tables in the report data set.

TableRelation Element

The **TableRelation** element contains information a parent-child relation between two tables in the report data set.

Name	***Cardinality***	***Has Children***	***Description***
ParentColumns	1	Yes	Describes the parent side of the relation.
ChildColumns	1	Yes	Describes the child side of the relation.

ParentColumns Element

The **ParentColumns** element contains information about columns that make a parent side of the relation between two tables in the report data set.

Name	***Cardinality***	***Has Children***	***Description***
TableName	1	No	Parent table name. Must match a table name set in a **TableName** element for a query.
ColumnNames	1	Yes	Lists the names of the columns that form the parent side of the relation.

ColumnNames Element

The **ColumnNames** element contains information about columns names in a table relation description.

Name	***Cardinality***	***Has Children***	***Description***
ColumnName	1-N	No	Column name in a table specified by .a **ParentColumns** or **ChildColumns** element.

ChildColumns Element

The **ChildColumns** element contains information about columns that make a child side of the relation between two tables in the report data set.

Name	***Cardinality***	***Has Children***	***Description***
TableName	1	No	Child table name. Must match a table name set in a **TableName** element for a query.
ColumnNames	1	Yes	Lists the names of the columns that form the child side of the relation.

Parameters Element

The **Parameters** element contains information about report parameters.

Aspose.Words for SharePoint requests the parameter values from a user when making a report via user interface. When generating controls for entering parameters, it takes parameter type into account.

You set the parameters through the API when making a report programmatically.

Name	***Cardinality***	***Has Children***	***Description***
String	1-N	Yes	Describes a string report parameter.
Int32	1-N	Yes	Describes a 32-bit integer report parameter.
DateTime	1-N	Yes	Describes a report parameter for representing date and time values.
Decimal	1-N	Yes	Describes a decimal report parameter.

Parameter Elements

Currently **String**, **Int32**, **DateTime** and **Decimal** parameter elements all have the same child elements.

However, parameter data type, specified by the element name is important. Aspose.Words for SharePoint uses is to generate controls for parameter entering. Parameter type also affects how the supplied values are converted when passed to queries.

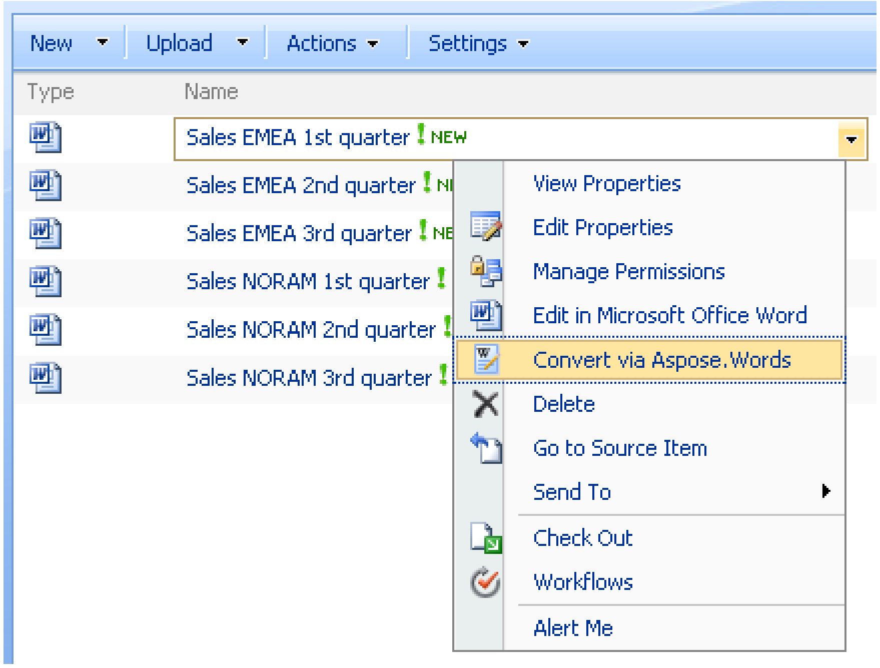
Name	***Cardinality***	***Has Children***	***Description***
Name	1	No	Parameter name to use in query elements like **Command** in **SqlQuery** and **Query** in **SPListQuery**.
Prompt	1	No	Parameter description to display next to a control generated for entering the parameter in the user interface.

Document Conversion

Convert One File

To convert a particular document in a document library please choose “Convert via Aspose.Words” item in the document’s Edit Control Block (ECB).

Selecting “Convert via Aspose.Words” item in the document’s ECB.


New
Upload
Actions
Settings
Type
Name
Sales EMEA 1st quarter ! NEW
Sales EMEA 2nd quarter ! NE
Sales EMEA 3rd quarter ! NE
Sales NORAM 1st quarter !
Sales NORAM 2nd quarter !
Sales NORAM 3rd quarter !
View Properties
Edit Properties
Manage Permissions
Edit in Microsoft Office Word
Convert via Aspose.Words
Delete
Go to Source Item
Send To
Check Out
Workflows
Alert Me

This will lead you to conversion settings screen that allows tuning of the following options:

“**Convert to**” radio button list selects desired destination file format.

“**Destination file**” edit box allows entering the name of destination file. By default, Aspose.Words for SharePoint prepopulates this field with the name of the source file. But you can change it if needed. The extension is added automatically depending on the selected format.

“**Destination folder**” edit box allows entering the URL of the document library or folder where you would like to place the converted file. It can be a folder in the current document library, another library within the current site collection or even library or folder in a different site collection. You can always type a path to the destination folder directly in the edit box, but it may be more convenient to select the destination folder using a graphical view of the sites and lists hierarchy provided by the Browse button (“...”) placed next to the edit box. Only document libraries and folders where the current user can add and edit items are selectable in the Browse dialog.

“Save destination folder” check box will save your selection and prepopulate Destination folder field upon next conversion session. This option saves URLs independently for different users. Moreover, as destination folder is stored at the site collection level you will have different destination folders prepopulated when starting conversion from, for example, a personal site and from a corporate site.

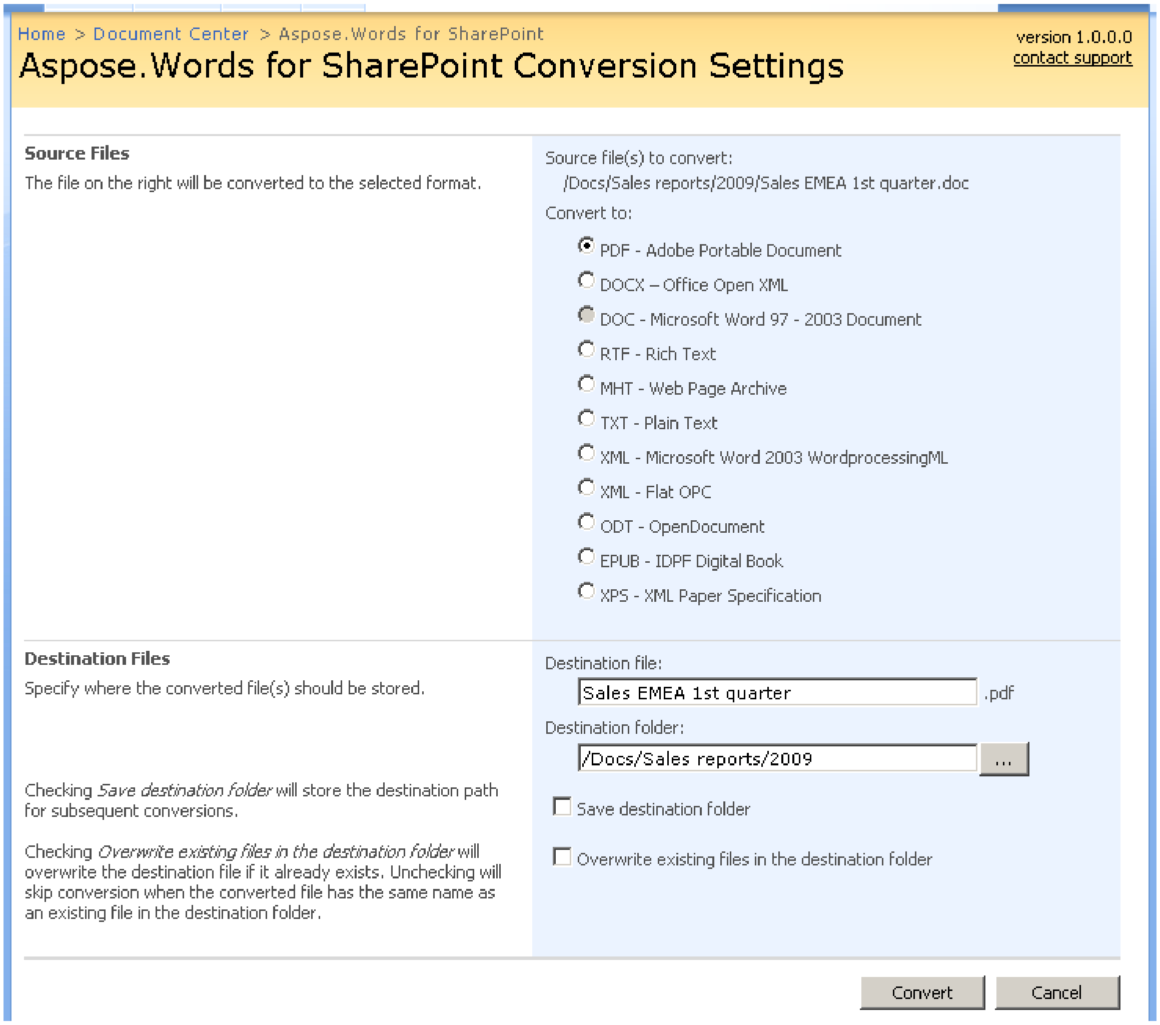
"Overwrite existing files in the destination folder" check box defines what should be done in case when destination file already exits. By default, conflicting files are skipped during conversion. Checking "Overwrite existing files" option will change the default behavior:

- If destination library has versioning enabled, a new file version is added for a conflicting file.
- If versioning is not enabled for the destination library, an existing file with a conflicting name is overwritten.

"**Convert**" button launches conversion process.

"**Cancel**" button gets you back to the library and dismisses all changes in settings.

Conversion Settings screen for one file conversion.



Home > Document Center > Aspose.Words for SharePoint

version 1.0.0.0
contact support

Aspose.Words for SharePoint Conversion Settings

Source Files

The file on the right will be converted to the selected format.

Source file(s) to convert:
/Docs/Sales reports/2009/Sales EMEA 1st quarter.doc

Convert to:

- ● PDF - Adobe Portable Document
- ○ DOCX – Office Open XML
- ○ DOC - Microsoft Word 97 - 2003 Document
- ○ RTF - Rich Text
- ○ MHT - Web Page Archive
- ○ TXT - Plain Text
- ○ XML - Microsoft Word 2003 WordprocessingML
- ○ XML - Flat OPC
- ○ ODT - OpenDocument
- ○ EPUB - IDPF Digital Book
- ○ XPS - XML Paper Specification

Destination Files

Specify where the converted file(s) should be stored.

Checking *Save destination folder* will store the destination path for subsequent conversions.

Checking *Overwrite existing files in the destination folder* will overwrite the destination file if it already exists. Unchecking will skip conversion when the converted file has the same name as an existing file in the destination folder.

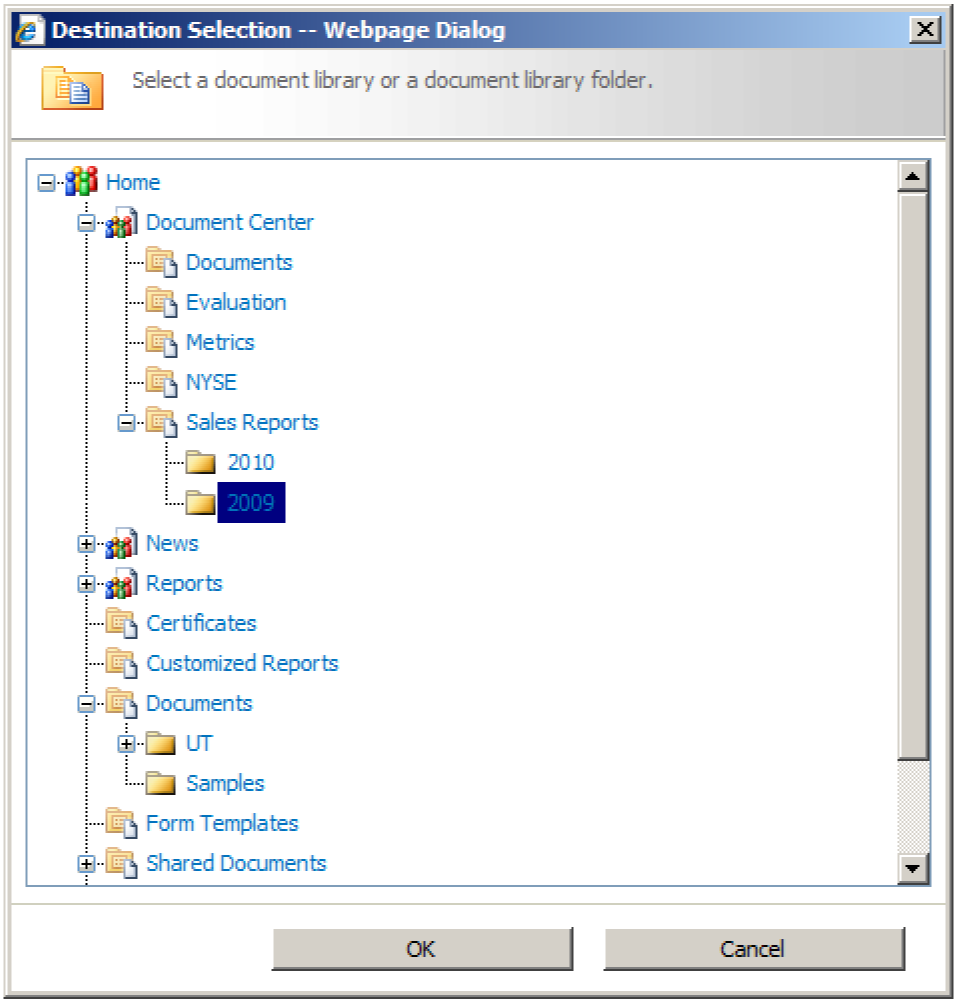
Destination file: Sales EMEA 1st quarter .pdf

Destination folder: /Docs/Sales reports/2009 ...

□ Save destination folder

□ Overwrite existing files in the destination folder

Selecting destination folder using graphical view via Browse button

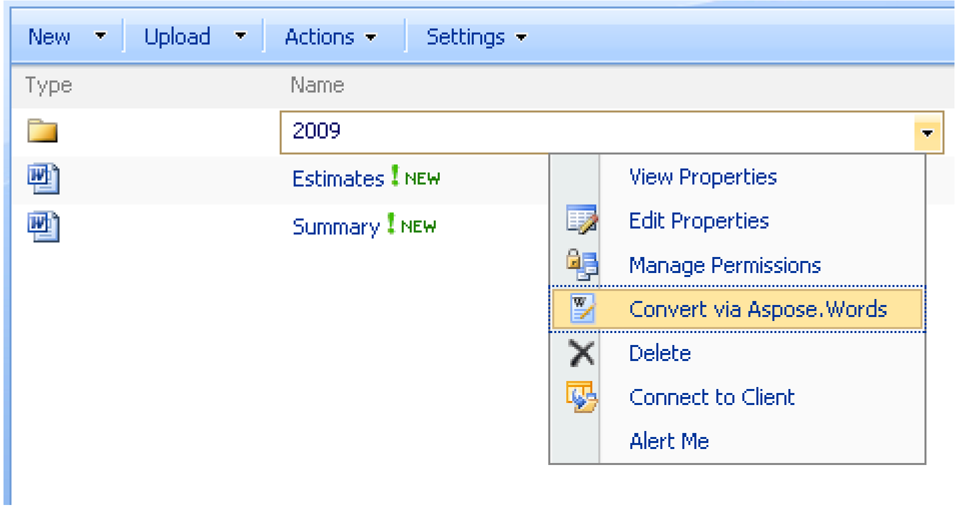


Select a document library or a document library folder.

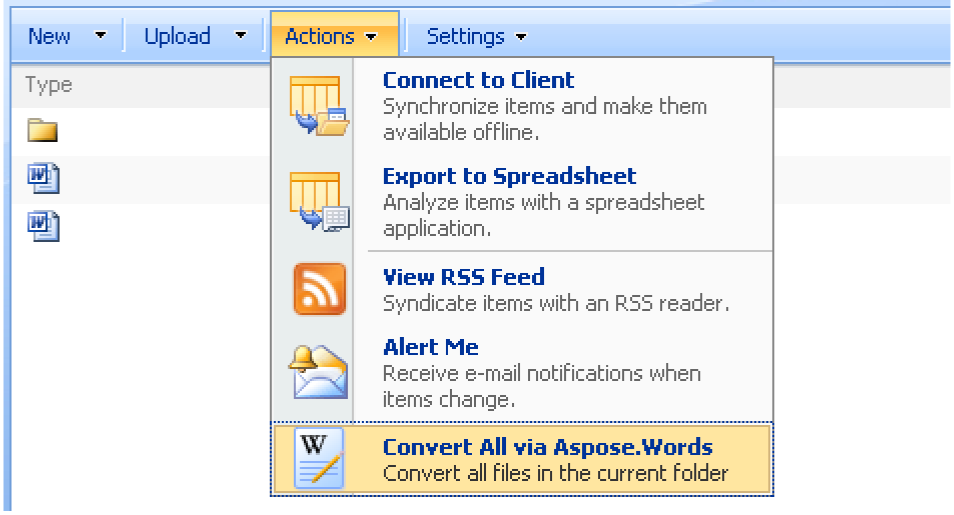
Convert All Files in a Folder

If you want to convert all files in a folder, you can do so either by selecting "Convert via Aspose.Words" option in ECB of desired folder while staying in the parent folder or by selecting "Convert All via Aspose.Words" from the Action menu while staying in the folder itself or in the root of the document library. Whichever way you have chosen, Aspose.Words for SharePoint will try to convert every document in the selected folder or library.

Convert all files in the subfolder.



Convert all files in the current folder.



Connect to Client
Synchronize items and make them available offline.

Export to Spreadsheet
Analyze items with a spreadsheet application.

View RSS Feed
Syndicate items with an RSS reader.

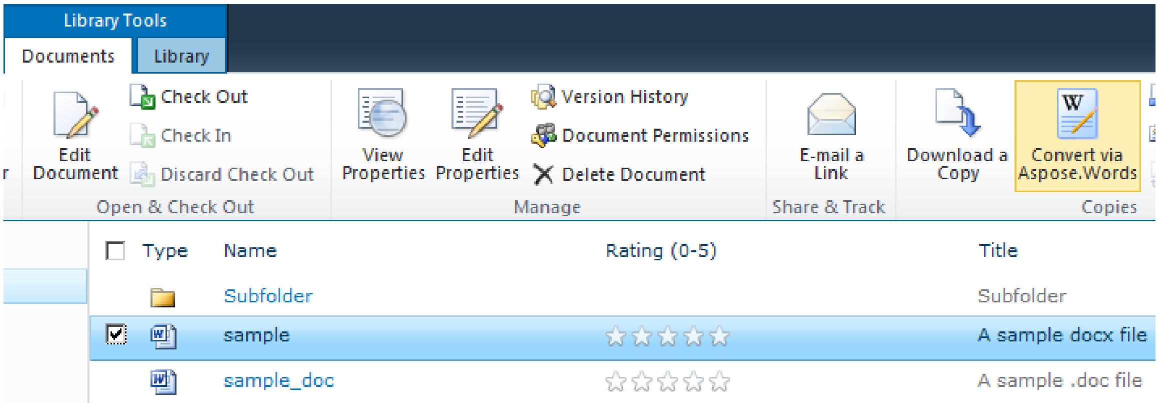
Alert Me
Receive e-mail notifications when items change.

Convert All via Aspose.Words
Convert all files in the current folder

Using Server Ribbon Button under SharePoint 2010

Under SharePoint 2010, server ribbon button is available instead of the custom action menu item used with SharePoint 2007 / WSS 3.

Server ribbon button converts a single file when a single file is selected.

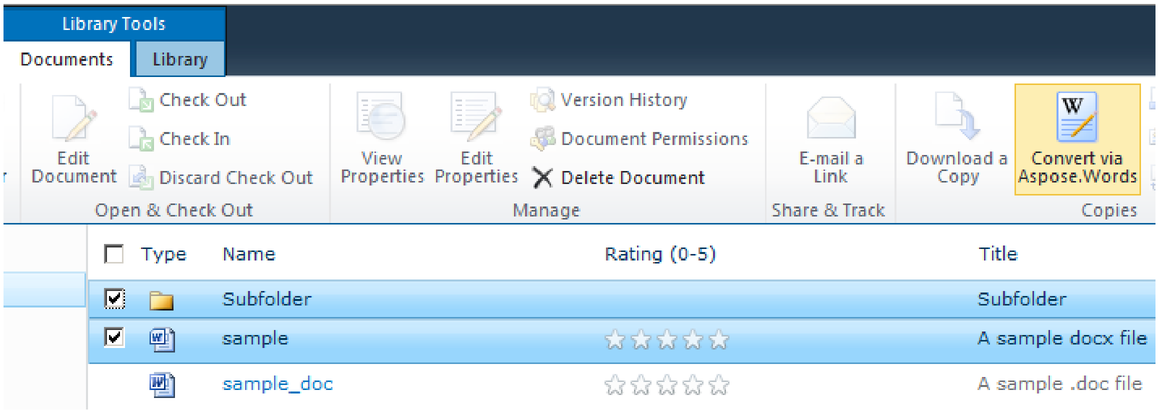


☐	Type	Name	Rating (0-5)	Title
		Subfolder		Subfolder
☑		sample	☆☆☆☆☆	A sample docx file
		sample_doc	☆☆☆☆☆	A sample .doc file

This documentation was generated using Aspose.Words for SharePoint.

When you select a single file or folder in a document library view, “Convert via Aspose.Words” server ribbon button displays conversion settings page for this item. It works the same way as ECB menu item.

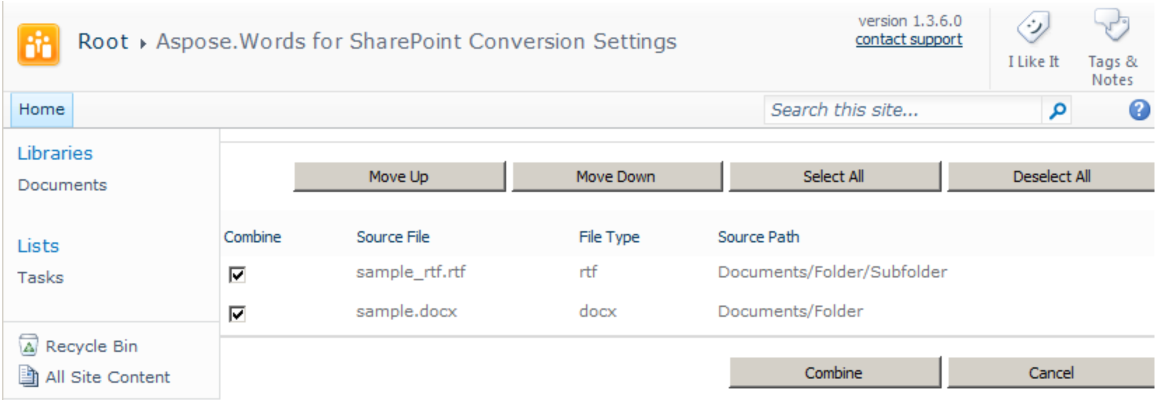
Multiple items selection is supported.



☐	Type	Name	Rating (0-5)	Title
☑	📁	Subfolder		Subfolder
☑	W	sample	☆☆☆☆☆	A sample docx file
	W	sample_doc	☆☆☆☆☆	A sample .doc file

When you select multiple items in a document library view, you can convert or combine selected items only.

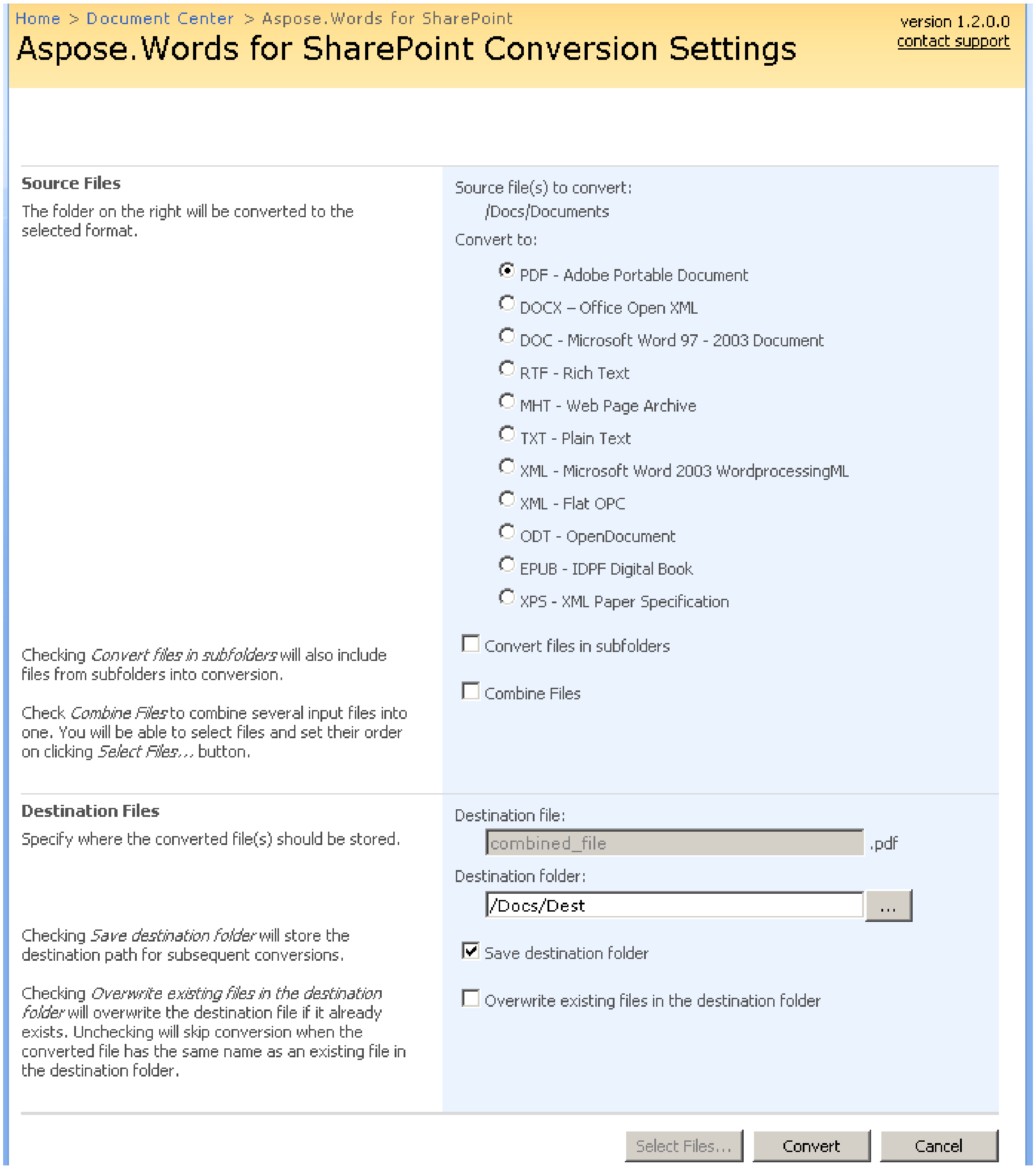
The list of files to combine is populated with the files selected in the document library.



Combine	Source File	File Type	Source Path
☑	sample_rtf.rtf	rtf	Documents/Folder/Subfolder
☑	sample.docx	docx	Documents/Folder

If you select no items in a document library view, “Convert via Aspose.Words” ribbon button is disabled.

Conversion Settings screen for multiple files conversion.



Home > Document Center > Aspose.Words for SharePoint

Aspose.Words for SharePoint Conversion Settings

version 1.2.0.0
contact support

Source Files

The folder on the right will be converted to the selected format.

Source file(s) to convert:
/Docs/Documents

Convert to:

● PDF - Adobe Portable Document
○ DOCX – Office Open XML
○ DOC - Microsoft Word 97 - 2003 Document
○ RTF - Rich Text
○ MHT - Web Page Archive
○ TXT - Plain Text
○ XML - Microsoft Word 2003 WordprocessingML
○ XML - Flat OPC
○ ODT - OpenDocument
○ EPUB - IDPF Digital Book
○ XPS - XML Paper Specification

Checking *Convert files in subfolders* will also include files from subfolders into conversion.

□ Convert files in subfolders

Check *Combine Files* to combine several input files into one. You will be able to select files and set their order on clicking *Select Files...* button.

□ Combine Files

Destination Files

Specify where the converted file(s) should be stored.

Destination file: combined_file .pdf

Destination folder: /Docs/Dest ...

Checking *Save destination folder* will store the destination path for subsequent conversions.

☑ Save destination folder

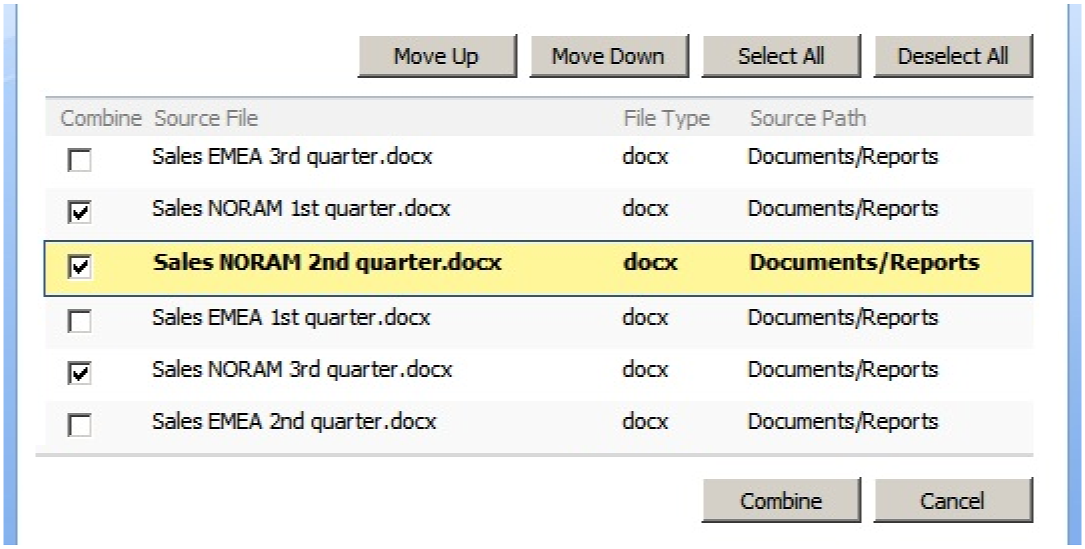
Checking *Overwrite existing files in the destination folder* will overwrite the destination file if it already exists. Unchecking will skip conversion when the converted file has the same name as an existing file in the destination folder.

□ Overwrite existing files in the destination folder

"Convert files in subfolders" check box lets you include all subfolders into conversion. During conversion, Aspose.Words for SharePoint will create the same subfolders tree in the destination folder. This option is for SharePoint 2007 only. Under SharePoint 2010, the subfolders are always processed recursively. To process folder contents without subfolders, just select the needed files in a document library view.

"Combine files" check box lets you combine files into one document. Input files can be in different formats. Setting this check box makes accessible "Select Files…" button that will lead to sequence adjustment screen.

“Select Files” button allows you to adjust combination order.



Combine	Source File	File Type	Source Path
□	Sales EMEA 3rd quarter.docx	docx	Documents/Reports
☑	Sales NORAM 1st quarter.docx	docx	Documents/Reports
☑	**Sales NORAM 2nd quarter.docx**	**docx**	**Documents/Reports**
□	Sales EMEA 1st quarter.docx	docx	Documents/Reports
☑	Sales NORAM 3rd quarter.docx	docx	Documents/Reports
□	Sales EMEA 2nd quarter.docx	docx	Documents/Reports

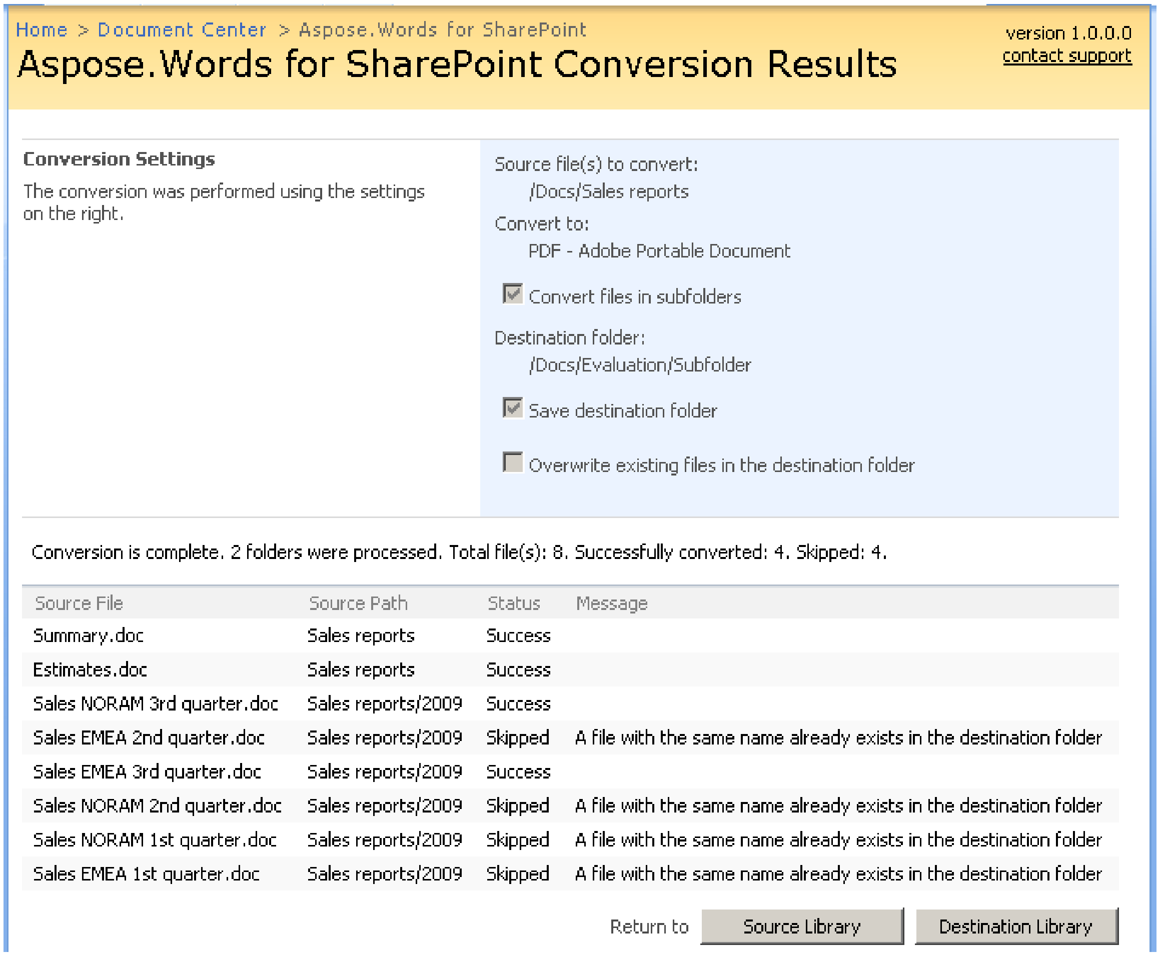
Conversion results screen

After finishing the conversion, Aspose.Words for SharePoint shows Conversion Results screen having three parts:

- “Conversion Settings” pane shows parameters used during conversion.
- Information string shows the totals of the conversion: overall number of processed files, number of files successfully converted and number of skipped files.
- Results list at the bottom of the page shows individual information for each file processed during conversion. Source Path in this list shows relative path from the folder there the conversion was started. You can sort the information in this list by pressing the title of any column.

After getting familiar with the results, you can return either to the source or to the destination library by pressing corresponding button at the end of the page.

Conversion Results screen.



Aspose.Words for SharePoint Conversion Results

Conversion Settings

The conversion was performed using the settings on the right.

Source file(s) to convert:
/Docs/Sales reports

Convert to:
PDF - Adobe Portable Document

☑ Convert files in subfolders

Destination folder:
/Docs/Evaluation/Subfolder

☑ Save destination folder

☐ Overwrite existing files in the destination folder

Conversion is complete. 2 folders were processed. Total file(s): 8. Successfully converted: 4. Skipped: 4.

Source File	Source Path	Status	Message
Summary.doc	Sales reports	Success	
Estimates.doc	Sales reports	Success	
Sales NORAM 3rd quarter.doc	Sales reports/2009	Success	
Sales EMEA 2nd quarter.doc	Sales reports/2009	Skipped	A file with the same name already exists in the destination folder
Sales EMEA 3rd quarter.doc	Sales reports/2009	Success	
Sales NORAM 2nd quarter.doc	Sales reports/2009	Skipped	A file with the same name already exists in the destination folder
Sales NORAM 1st quarter.doc	Sales reports/2009	Skipped	A file with the same name already exists in the destination folder
Sales EMEA 1st quarter.doc	Sales reports/2009	Skipped	A file with the same name already exists in the destination folder

Using Aspose.Words for SharePoint in a Workflow

Workflow summary

You can use Aspose.Words for SharePoint in SharePoint workflows. After installation you can see two new activities available in SharePoint Designer:

- Convert via Aspose.Words
- Make a Report via Aspose.Words

Use "Convert via Aspose.Words” activity to convert documents between supported formats and store the result in a SharePoint document library. Use “Make a Report via Aspose.Words” activity to make a report from a template stored in a SharePoint document library. You can use the output of these activities in other activities.

Converting a Document via Workflow Activity

Let us use SharePoint Designer to create a workflow for converting new documents to the .pdf format and storing the results in another document library.

Select source and destination document libraries

We need to choose source and destination document library for converting documents. Let us assume that we would like to convert all new documents in a library called “Incoming Documents” and store the converted .pdf documents in a library called “Converted Documents” placing the result in the “PDF” subfolder.

If you would like to reproduce this example, you can create these libraries in a SharePoint site you use for testing. Alternatively, you can use existing libraries instead of the suggested example libraries.

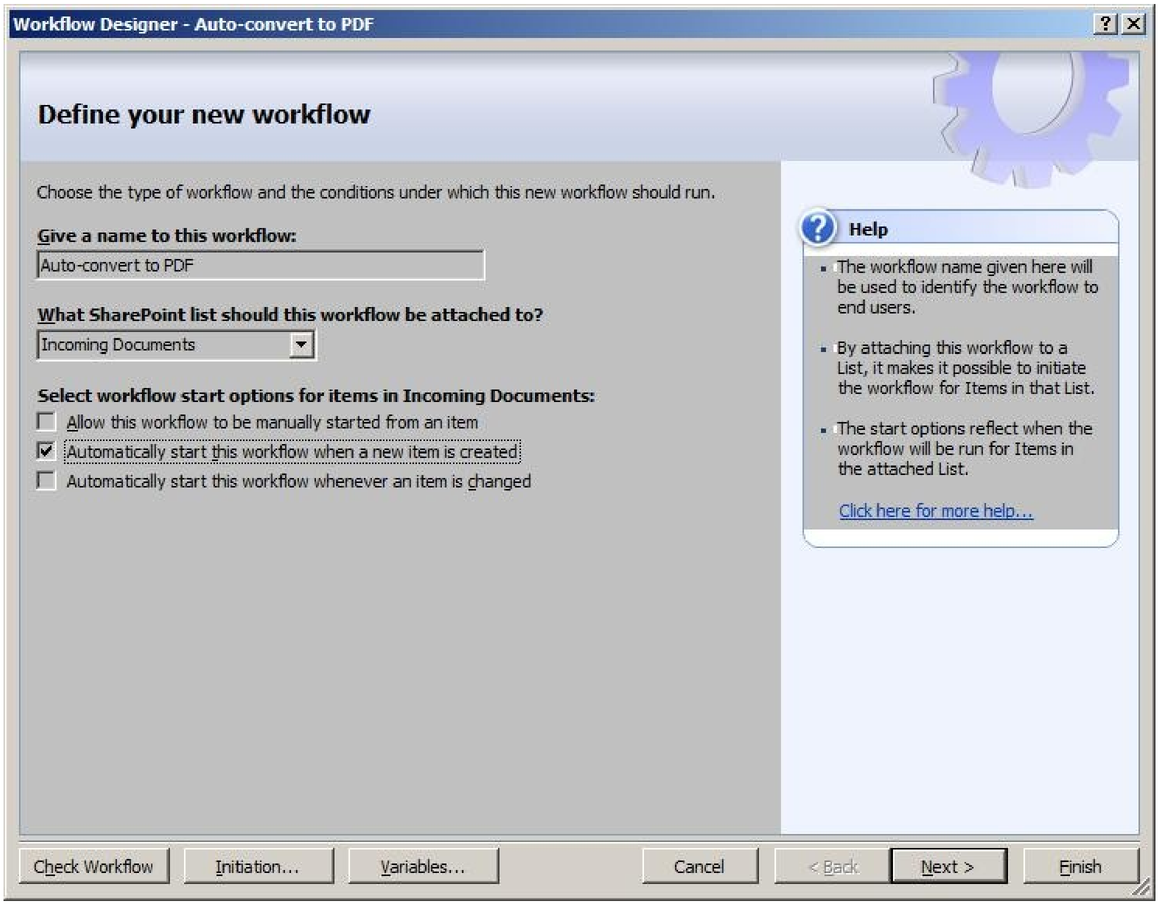
When a workflow is executed, it has the permissions of the user who initiated the workflow. So in order to store the converted documents, users should have the permission to add and update documents in the destination library.

Create a new workflow

To create a new workflow, start SharePoint Designer and connect to the site where the source document library is located. Select “New…” and “Workflow…” from the File menu.

Workflow designer window will appear. Attach the workflow to the “Incoming Documents” library by selecting it in the document library list. Check “Automatically start this workflow when a new item is created” and clear other checkboxes in order to run this workflow for new documents only.

Creating a new workflow.



Workflow Designer - Auto-convert to PDF

Define your new workflow

Choose the type of workflow and the conditions under which this new workflow should run.

Give a name to this workflow: Auto-convert to PDF

What SharePoint list should this workflow be attached to? Incoming Documents

Select workflow start options for items in Incoming Documents:

☐ 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

Help

- The workflow name given here will be used to identify the workflow to end users.
- By attaching this workflow to a List, it makes it possible to initiate the workflow for Items in that List.
- The start options reflect when the workflow will be run for Items in the attached List.

Click here for more help...

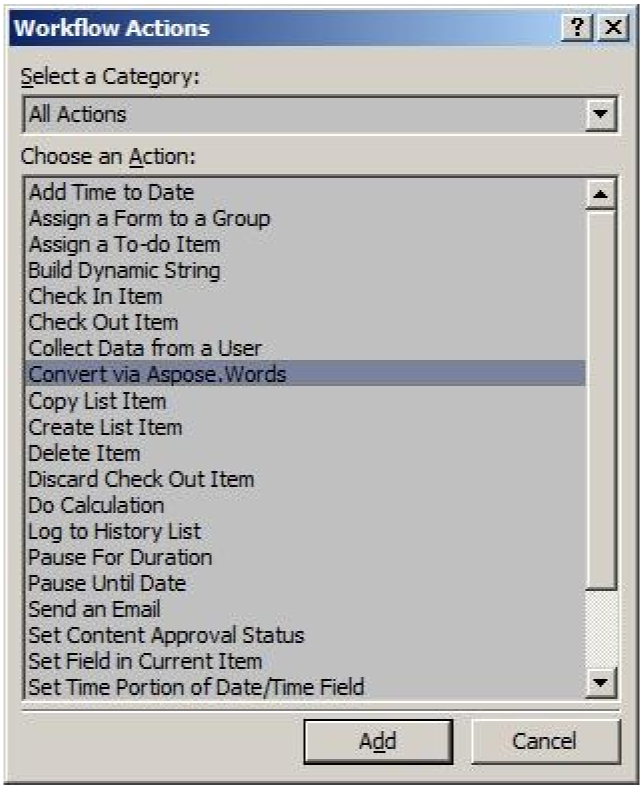
Give your new workflow a descriptive name.

Add “Convert via Aspose.Words” action to your workflow

On clicking “Next” at the previous step, a workflow steps editor will appear. Let us just add an action that will try to convert every new document without any conditions.

Click “Actions” and select “Convert via Aspose.Words” from the list. You may need to select “More Actions…” to produce the complete action list.

Selecting conversion action.



Select a Category: All Actions

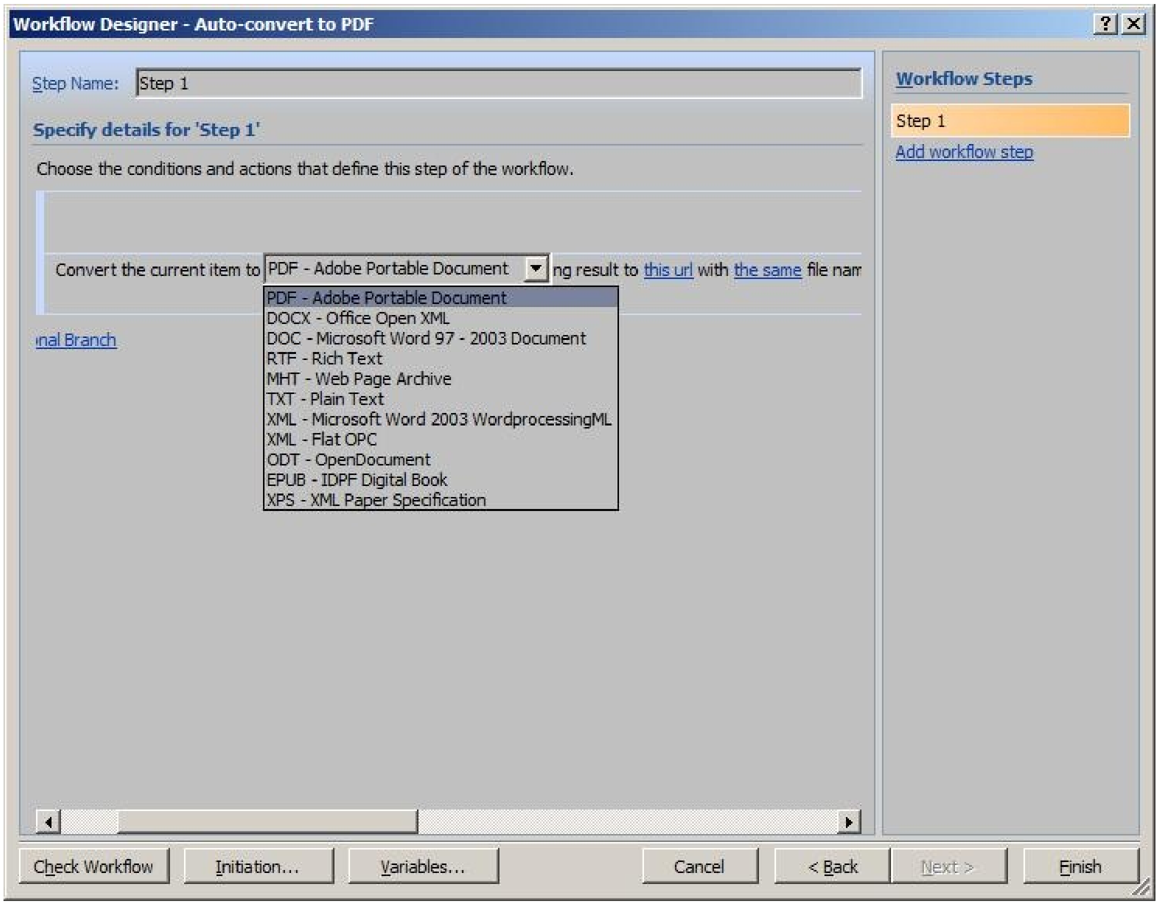
Choose an Action: Convert via Aspose.Words

Configure action parameters

The parameters of the workflow action are similar to the conversion settings available when launching conversion via Aspose.Words from a document library:

- Destination format. You can use all formats supported by Aspose.Words for SharePoint in a workflow.
- Destination folder URL. This is where the converted document is placed. If this parameter is not set, conversion result is placed in the same folder with the source file.
- Destination file name. Specify this parameter to give the converted file a different name. If this parameter is not set, the converted file will have the same name as a source file. An extension is added according to the specified destination format.
- Whether the existing files in the destination library should be overwritten by the conversion result. By default, conversion does not overwrite existing files.

Selecting destination format.



Step Name: Step 1

Specify details for 'Step 1'

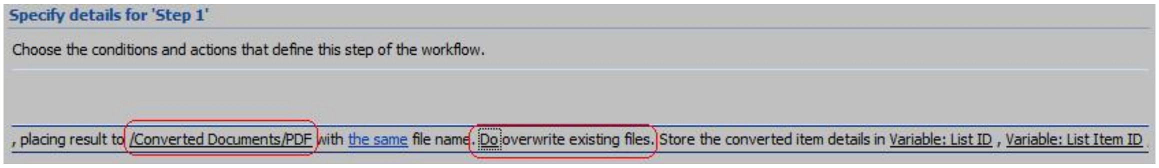
Choose the conditions and actions that define this step of the workflow.

Convert the current item to PDF - Adobe Portable Document ng result to this url with the same file nam

- PDF - Adobe Portable Document
- DOCX - Office Open XML
- DOC - Microsoft Word 97 - 2003 Document
- RTF - Rich Text
- MHT - Web Page Archive
- TXT - Plain Text
- XML - Microsoft Word 2003 WordprocessingML
- XML - Flat OPC
- ODT - OpenDocument
- EPUB - IDPF Digital Book
- XPS - XML Paper Specification

For our example, let us type in the server-relative URL of the destination document library folder. Leave the destination file name parameter at the default ("the same file name"). Select "Do overwrite existing files" to overwrite existing files in the destination folder with the conversion result.

Setting other workflow action parameters.



Specify details for 'Step 1'

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

, placing result to /Converted Documents/PDF with the same file name. Do overwrite existing files. Store the converted item details in Variable: List ID , Variable: List Item ID

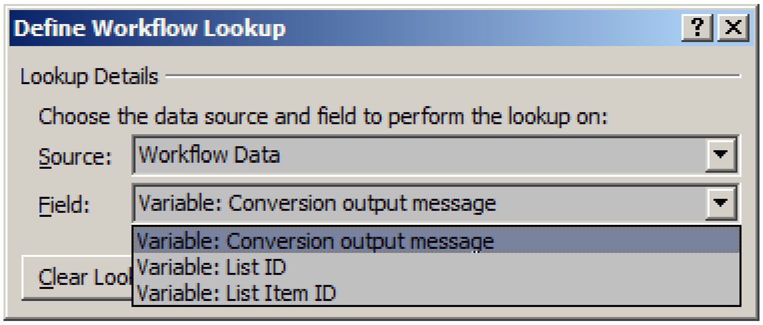
Use action output parameters in other actions

The conversion action sets three workflow variables on execution.

- "List ID". This is the ID of the list where the converted file is saved.
- "List Item ID". This is the id of the document item in the destination list. Use these two parameters in subsequent actions when you need to do something with the converted file.
- "Conversion output message". This is a textual description of conversion outcome.

Let us add another action using one of the conversion output parameters as an argument. Click "Actions" button in the workflow designer to add another action and select "Log to History List". Then click "this message" in the action sentence and "*fx*" button next to it in order to set the message to the conversion output. Select "Workflow Data" as a source and "Variable: Conversion output data" as a field.

Using action output as a parameter for another action.



Choose the data source and field to perform the lookup on:

Source: Workflow Data

Field: Variable: Conversion output message

Save the new workflow

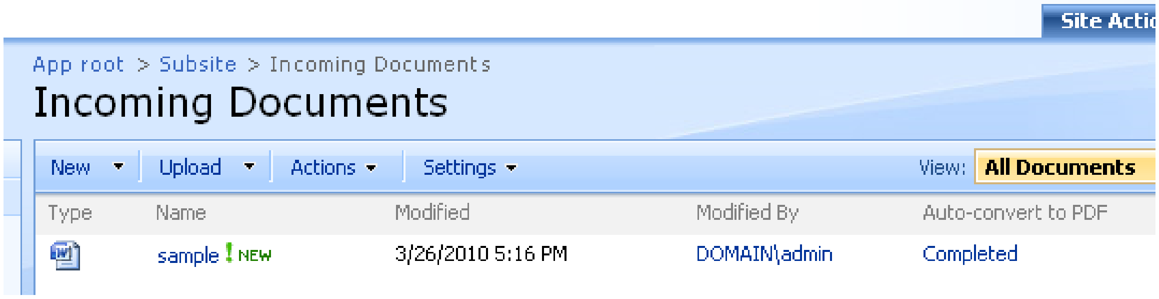
Click “Finish” in the workflow designer window to save the new workflow.

Test the new workflow

Now let us see how it works. Start a browser and go to the “Incoming Documents” library to which our new workflow is attached. Upload a new document. Make sure that document type is supported by Aspose.Words (docx; doc; rtf; mht; xml: Microsoft Word 2003 WordprocessingML or Flat OPC). The workflow should start automatically.

Refresh the browser window. A new column with the workflow status should appear:

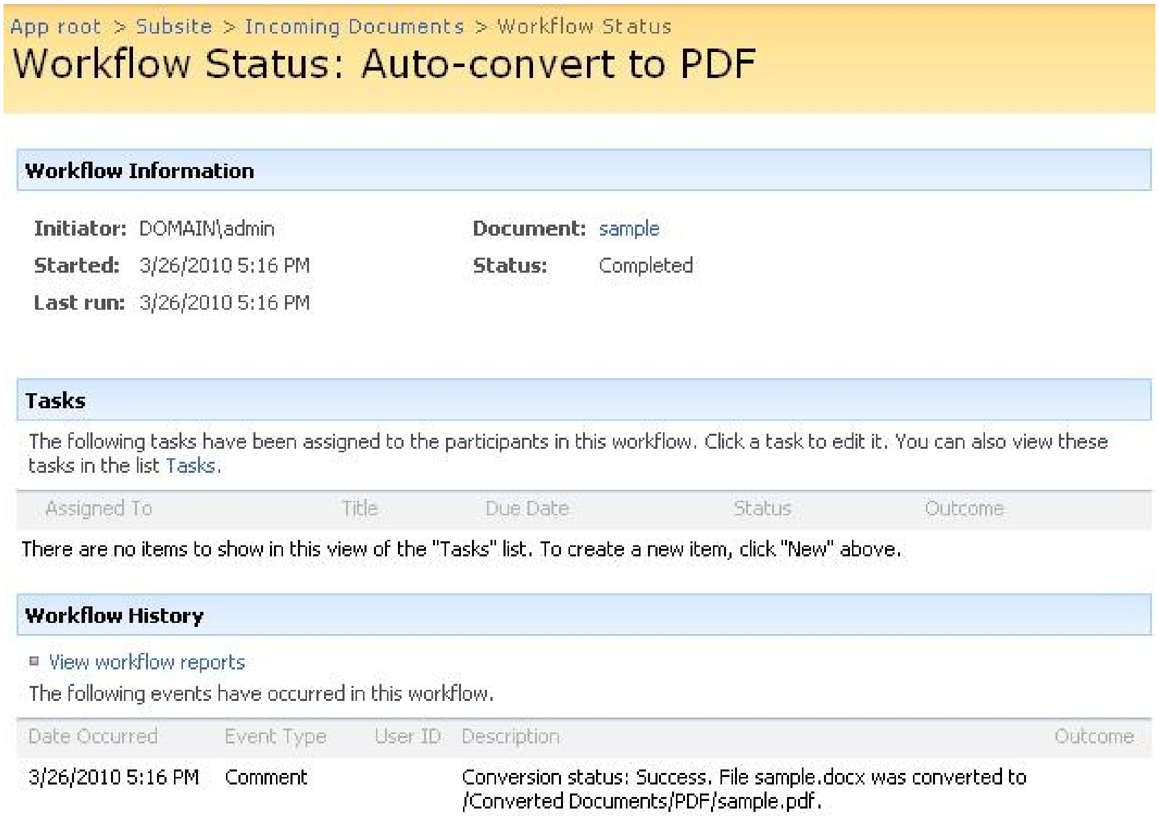
Workflow status column displayed in a document library.



Type	Name	Modified	Modified By	Auto-convert to PDF
	sample ! NEW	3/26/2010 5:16 PM	DOMAIN\admin	Completed

You can click “Completed” in the workflow status column to see that conversion outcome message was logged to the workflow history.

A workflow history entry.



App root > Subsite > Incoming Documents > Workflow Status

Workflow Status: Auto-convert to PDF

Workflow Information

Initiator:	DOMAIN\admin	**Document:**	sample
Started:	3/26/2010 5:16 PM	**Status:**	Completed
Last run:	3/26/2010 5:16 PM		

Tasks

The following tasks have been assigned to the participants in this workflow. Click a task to edit it. You can also view these tasks in the list Tasks.

Assigned To	Title	Due Date	Status	Outcome

There are no items to show in this view of the "Tasks" list. To create a new item, click "New" above.

Workflow History

▫ View workflow reports

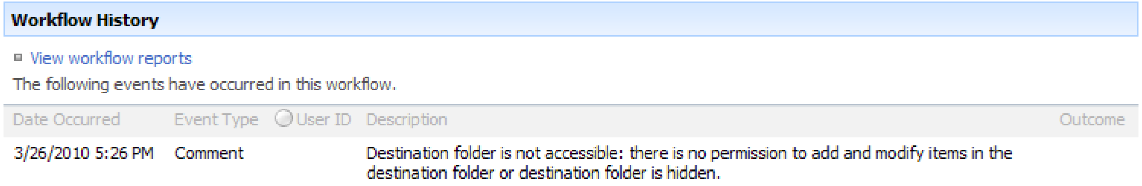
The following events have occurred in this workflow.

Date Occurred	Event Type	User ID	Description	Outcome
3/26/2010 5:16 PM	Comment		Conversion status: Success. File sample.docx was converted to /Converted Documents/PDF/sample.pdf.	

You can go to the destination library and find the converted document there.

Now let us assume that a user with insufficient permissions to update the destination library has started the workflow. The workflow again completes successfully, but the failure is logged in the workflow history:

An unsuccessful conversion logged in the workflow history.



Workflow History

▫ View workflow reports

The following events have occurred in this workflow.

Date Occurred	Event Type	User ID	Description	Outcome
3/26/2010 5:26 PM	Comment		Destination folder is not accessible: there is no permission to add and modify items in the destination folder or destination folder is hidden.	

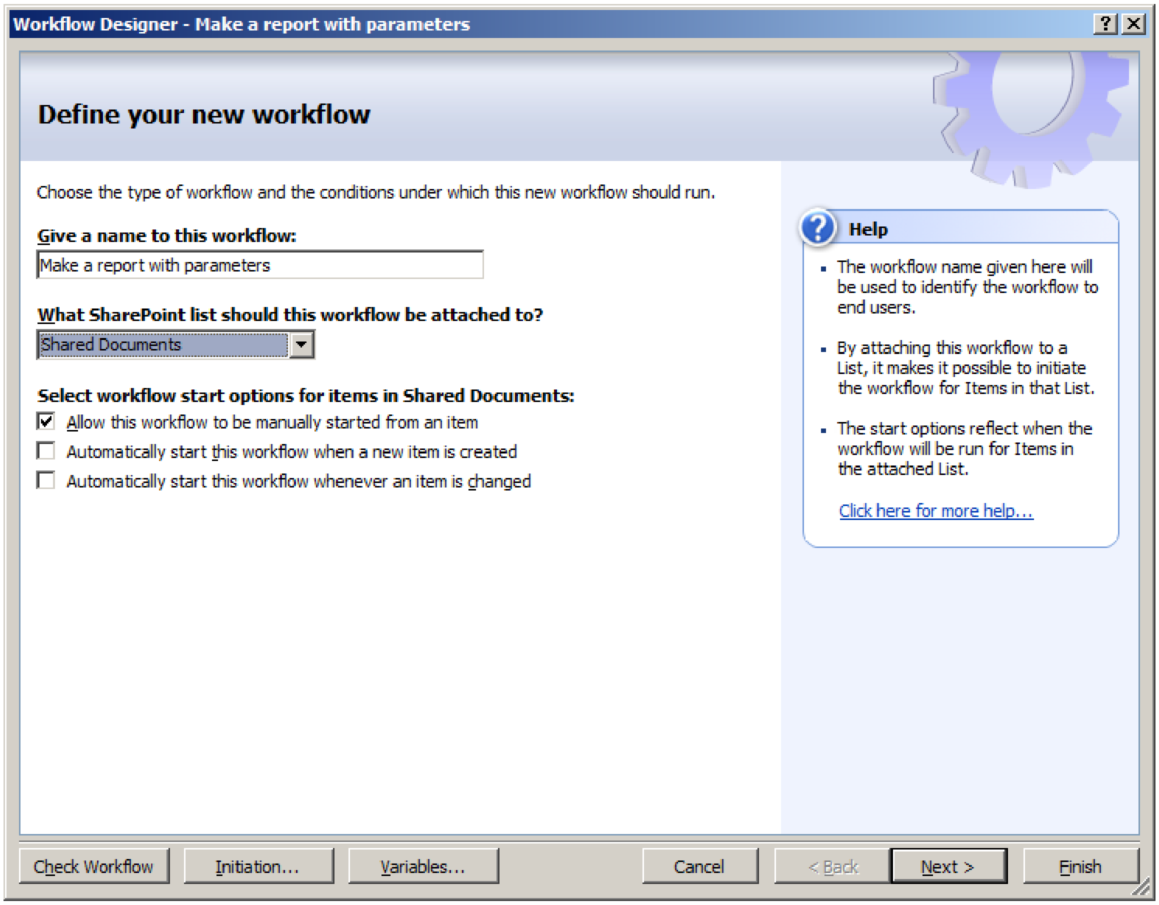
Making a Report via Workflow Activity

Let us walk through the creation of a workflow for building a report with parameters. The steps and screenshot are for SharePoint Designer 2007. The steps for SharePoint 2010 are very similar.

Start SharePoint Designer, open a site and create a new workflow.

Name the workflow and associate it with a document library.

Creating a new workflow.



Workflow Designer - Make a report with parameters

Define your new workflow

Choose the type of workflow and the conditions under which this new workflow should run.

Give a name to this workflow:

Make a report with parameters

What SharePoint list should this workflow be attached to?

Shared Documents

Select workflow start options for items in Shared Documents:

☑ 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

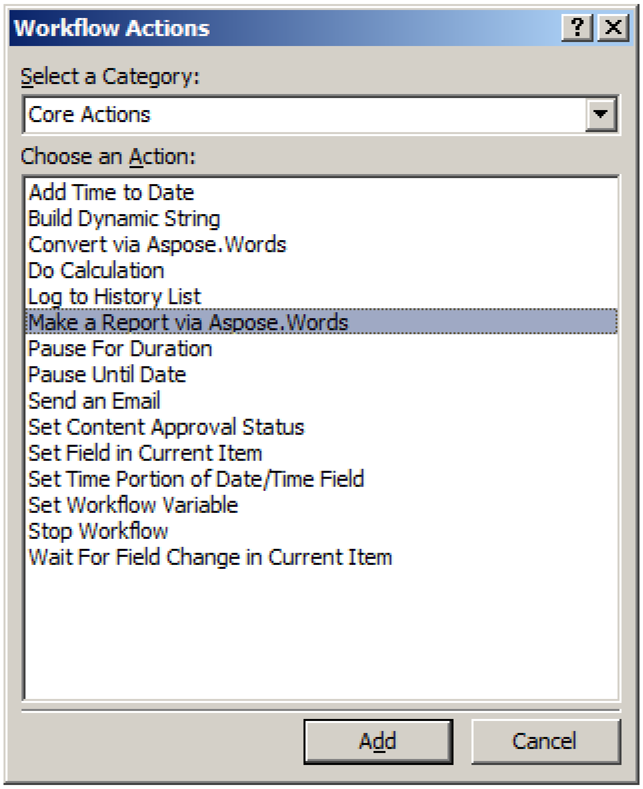
Help

- The workflow name given here will be used to identify the workflow to end users.
- By attaching this workflow to a List, it makes it possible to initiate the workflow for Items in that List.
- The start options reflect when the workflow will be run for Items in the attached List.

Click here for more help...

Add “Make a Report via Aspose.Words” activity to the workflow.

Adding workflow activity for making a report.

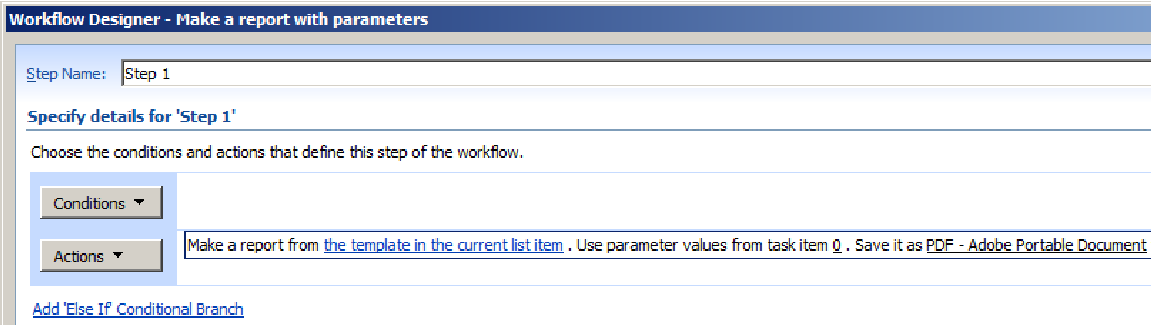


Workflow Actions

Select a Category: Core Actions

Choose an Action: Make a Report via Aspose.Words

By default, the report template is taken from the current list item.



Workflow Designer - Make a report with parameters

Step Name: Step 1

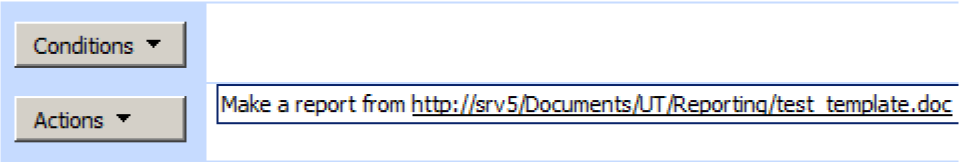
Specify details for 'Step 1'

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

Make a report from the template in the current list item . Use parameter values from task item 0 . Save it as PDF - Adobe Portable Document

However, in this walkthrough we will set a fixed URL to the report template by editing the first workflow parameter.

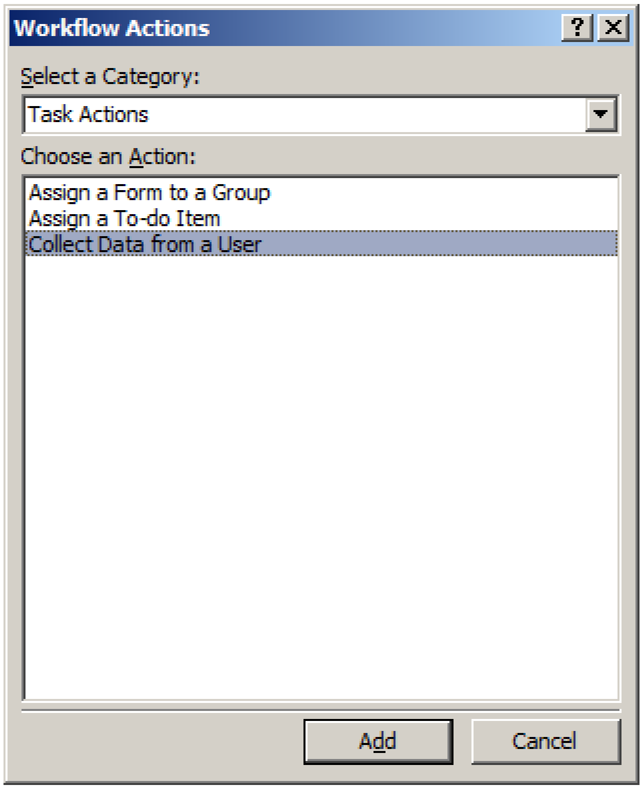
The report template URL is specified in the activity parameter.



Make a report from http://srv5/Documents/UT/Reporting/test_template.doc

Let us assume that we need to provide the report parameter values during workflow execution. You can do it via the built-in “Collect Data from a User” SharePoint activity.

Adding the activity for collecting parameter values.



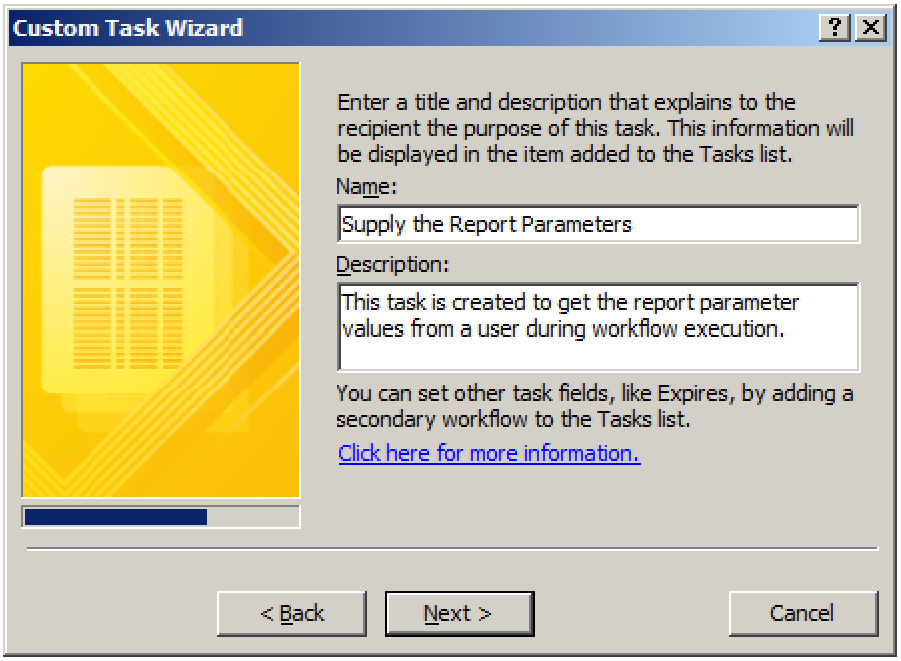
Select a Category: Task Actions

Choose an Action:

Assign a Form to a Group
Assign a To-do Item
Collect Data from a User

Move the added activity up in the workflow sequence so that it executes before building the report. You specify what data to collect in the activity "data" parameter. When the activity executes, a task is created in the workflow task list and is assigned to the specified user.

The task configuration wizard is started on clicking the "data" parameter of the collection activity.



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: Supply the Report Parameters

Description: This task is created to get the report parameter values from a user during workflow execution.

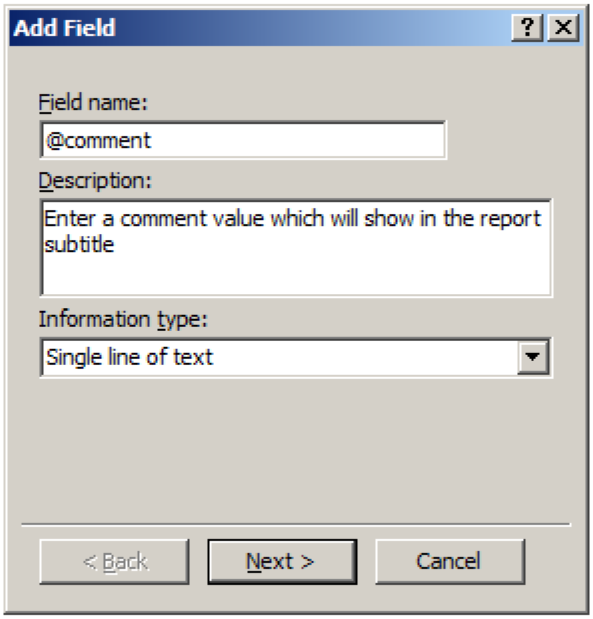
You can set other task fields, like Expires, by adding a secondary workflow to the Tasks list.

Click here for more information.

For example, if the report is expecting two parameters, you should configure the data collection activity to get two fields. The field names should match the parameter names in the report definition. Here is a fragment of the report definition with parameter description.

```
<Parameters>
  <String>
    <Name>@comment</Name>
    <Prompt>This is a test report parameter of type string</Prompt>
  </String>
  <Int32>
    <Name>@numericValue</Name>
    <Prompt>This is a test numeric report parameter</Prompt>
  </Int32>
</Parameters>
```

Configuring the field for the @comment parameter in the data collection activity.

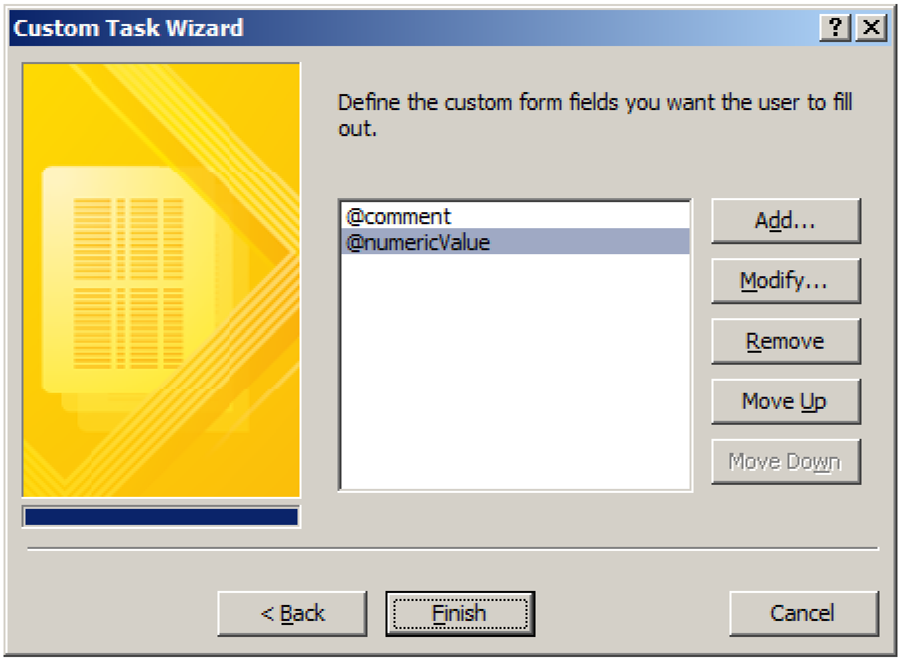


Field name: @comment

Description: Enter a comment value which will show in the report subtitle

Information type: Single line of text

Data collection activity with two fields added for report parameters.

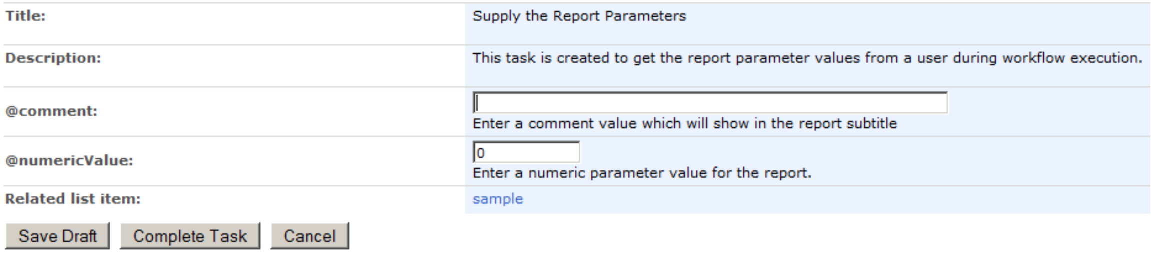


Define the custom form fields you want the user to fill out.

@comment
@numericValue

Please note that field names in the data collection activity match parameter names in the report description. You can specify the user who will provide the parameter values in the "user" parameter of the data collection activity. The user will enter the parameter values and complete the task during workflow execution.

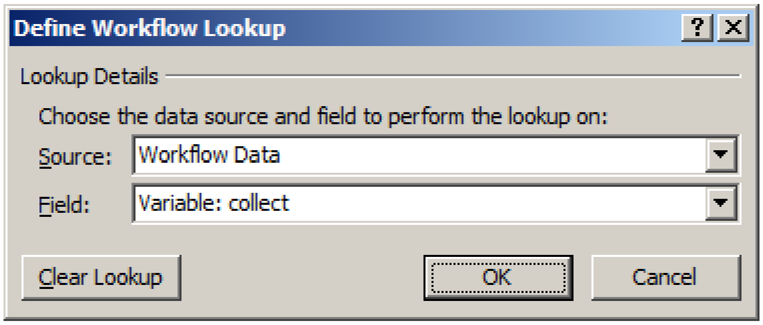
A form for entering parameter values generated by the data collection activity.



Title:	Supply the Report Parameters
Description:	This task is created to get the report parameter values from a user during workflow execution.
@comment:	Enter a comment value which will show in the report subtitle
@numericValue:	0 Enter a numeric parameter value for the report.
Related list item:	sample

You link the collected data to the report generation activity by setting the parameter task item ID in the report activity to the output parameter of the data collection activity. Data collection activity output parameter is named "collect" by default and can be accessed via workflow variables.

Defining a workflow lookup to the data collection output parameter.

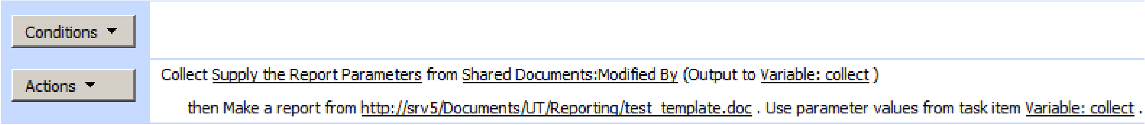


Choose the data source and field to perform the lookup on:

Source: Workflow Data

Field: Variable: collect

Data collection output linked to the report input in the workflow.

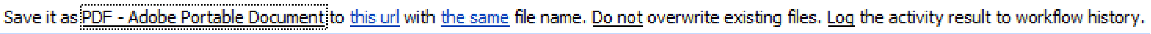


Collect Supply the Report Parameters from Shared Documents:Modified By (Output to Variable: collect)

then Make a report from http://srv5/Documents/UT/Reporting/test_template.doc . Use parameter values from task item Variable: collect .

Most of the remaining report activity parameters are similar to the conversion activity. You can specify the format to save the generated report, destination folder and file name. If the destination folder is not specified, the report is saved to the folder where the report template is. If the file name is not specified, the report will get the same file name as the report template and an extension according to the specified saving format. You can choose whether to overwrite the existing files in case of name clash. However, the report template from which the report is created is never overwritten.

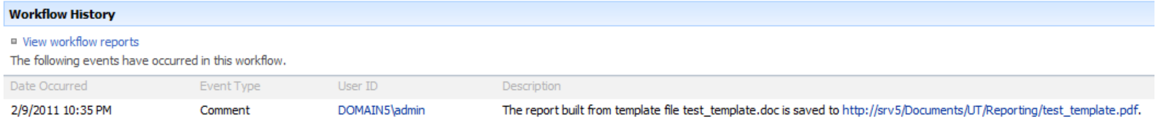
The default input parameter values of the reporting activity.



Save it as PDF - Adobe Portable Document to this url with the same file name. Do not overwrite existing files. Log the activity result to workflow history.

By default, the activity logs its outcome to the workflow history list. You can switch the logging off.

The activity outcome is logged to the workflow history
list.



Workflow History

▫ View workflow reports

The following events have occurred in this workflow.

Date Occurred	Event Type	User ID	Description
2/9/2011 10:35 PM	Comment	DOMAIN5\admin	The report built from template file test_template.doc is saved to http://srv5/Documents/UT/Reporting/test_template.pdf.

The list ID and item ID of the created list item are saved in the activity output parameters. You can use the in subsequent workflow actions.

Troubleshooting

Aspose.Words for SharePoint uses SharePoint tracing logs to log information about unexpected errors. They may be useful when contacting support. The logs are in the LOGS folder of the SharePoint installation:

- c:\Program Files\Common Files\Microsoft Shared\Web Server Extensions\12\LOGS\ for SharePoint 2007.
- c:\Program Files\Common Files\Microsoft Shared\Web Server Extensions\14\LOGS\ for SharePoint 2010.

You can find messages related to Aspose.Words for SharePoint using the line "Aspose.Words for SharePoint" as a keyword. It serves as a diagnostics area under SharePoint 2010 or as a product name under SharePoint 2007. Here is an example of the logged messages.

```
10/21/2010 10:29:48.42    w3wp.exe (0x0A1C)
                          0x0808   Aspose.Words for SharePoint       Exception
                          0000     Unexpected       Message:  Report  template  is  invalid.  An  error
occured when loading report template. CallStack:    at
…
```

There is a known issue with trace logging under SharePoint 2010: after installation, the application pool account (the one used to run w3wp.exe) has insufficient permissions for logging trace messages. You can fix it by manually making the application pool account a member of the "Performance Log Users" group.