# Getting Started

## Introducing Aspose.Words for .NET



Aspose.Words for .NET is a class library that enables your applications to perform a great range of document processing tasks. Aspose.Words supports DOC, DOCX, RTF, HTML, OpenDocument, PDF, XPS, EPUB and other formats. With Aspose.Words you can generate, modify, convert, render and print documents without utilizing Microsoft Word®.

Using Aspose.Words for .NET in your project gives you the following benefits:

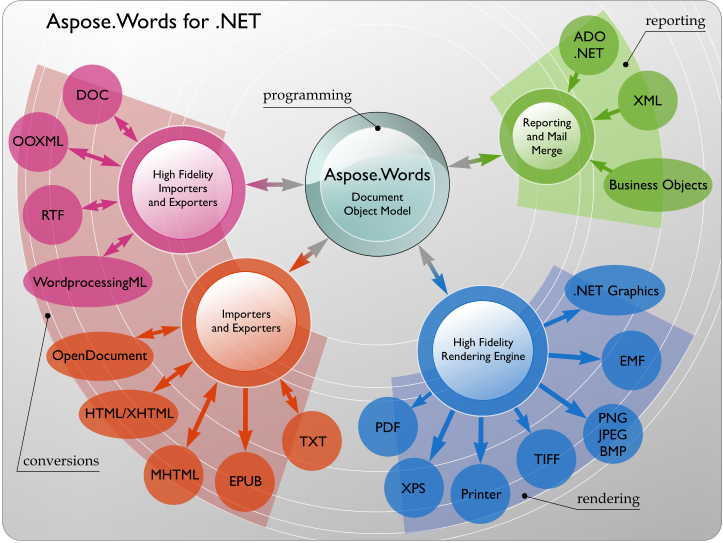
* Rich Set of Features
* No Microsoft Word Automation
* Platform Independence
* Performance and Scalability
* Minimal Learning Curve

##### Rich Set of Features

Aspose.Words for .NET features can be divided into four main areas:

* **Conversions.** High quality conversions to and from DOC, OOXML, RTF, WordprocessingML, HTML, MHTML, TXT and OpenDocument formats.
* **Document Object Model.** Programmatic access through a rich API to all document elements and formatting allows to create, modify, extract, copy, split, join, and replace document content.
* **Rendering.** Convert whole documents or pages to PDF or XPS for server-side document generation or printing. Also convert document pages to TIFF, PNG, BMP or EMF images. Print or render pages to .NET Graphics. All with high fidelity - exactly like Microsoft Word® would have done it.
* **Reporting.** Generate documents or reports from scratch or by filling templates with data from data sources or business objects.

The main feature areas of Aspose.Words for .NET and how they relate to each other.



##### No Microsoft Word Automation

Aspose.Words does not require Microsoft Office to be installed on the machine in order to work. All Aspose components are totally independent and are not affiliated with, nor authorized, sponsored, or otherwise approved by Microsoft Corporation. In short, Aspose.Words is a better alternative to automation in terms of security, stability, scalability/speed, price and features.

##### Platform Independence

Aspose.Words for .NET and Aspose.Words for Java are “twin brothers” products and together cover most of the popular development environments and deployment platforms.

Both Aspose.Words for .NET and Aspose.Words for Java run on Windows, Linux and Mac OS operating systems.

You can use Aspose.Words for .NET to build any type of a 32-bit or 64-bit .NET application including ASP.NET, WCF, WinForms etc. It is possible to use Aspose.Words for .NET via COM Interop from ASP, Perl, PHP and Python. You can also use Aspose.Words for .NET to build applications with Mono.

Aspose.Words for Java is available for Java 1.4, 1.5 and 1.6 and will run in any place where Java is installed. It is possible to use Aspose.Words for Java from Perl, PHP and Python.

##### Performance and Scalability

Aspose.Words is designed to perform great both on a server or client. Aspose.Words is a single .NET assembly that can be deployed with any .NET application by simply copying it. You do not have to worry about any other services or modules.

You can literally generate thousands of documents in minutes with Aspose.Words and that involves opening a document, modifying content and formatting or populating it with data and saving it. Even most of the complex documents are opened and saved on a P4 2.4Ghz 1Gb RAM machine in fractions of a second.

Aspose.Words is multithread safe as long as only one thread works on a document at a time. It is a typical scenario to have one thread working on one document. Different threads can safely work on different documents at the same time.

##### Minimal Learning Curve

Although there are over 150 public classes and enumerations in Aspose.Words, the learning curve is minimal because the Aspose.Words API has been carefully designed with the following goals in mind:

* Borrow best practices from other well-known APIs such as Microsoft Word Automation.
* Borrow best practices from the .NET Framework Design Guidelines.
* Provide a balance of easy usage vs detailed document element manipulation.

Developers, who used Microsoft Word Automation in their projects before, will find names and behavior of many classes, methods and properties familiar. [Document](http://www.aspose.com/documentation/.net-components/aspose.words-for-.net/aspose.words.document.html), [Paragraph](http://www.aspose.com/documentation/.net-components/aspose.words-for-.net/aspose.words.paragraph.html), [Bookmark](http://www.aspose.com/documentation/.net-components/aspose.words-for-.net/aspose.words.bookmark.html), [Range](http://www.aspose.com/documentation/.net-components/aspose.words-for-.net/aspose.words.range.html), [Section.PageSetup](http://www.aspose.com/documentation/.net-components/aspose.words-for-.net/aspose.words.section.pagesetup.html), [Paragraph.ParagraphFormat](http://www.aspose.com/documentation/.net-components/aspose.words-for-.net/aspose.words.paragraph.paragraphformat.html) are some of the Aspose.Words classes.

At the same time, Aspose.Words is quite different from the Microsoft Word Object Model in that it represents the document as a tree of objects more like an XML DOM tree. If you worked with any XML DOM library you will find it is easy to understand and work with Aspose.Words.

## Aspose.Words Features

### Feature Overview

The tables below summarize Aspose.Words for .NET features and provide links to detailed feature descriptions.

Most of the screenshots in this section demonstrate documents generated by Aspose.Words and then opened for viewing in Microsoft Word and other applications.

##### File Formats and Conversions

Many customers worldwide rely on the thorough document import and export features provided by Aspose.Words.

|  |  |
| --- | --- |
| Feature | Description |
| [High-Quality Conversions](#_High-Fidelity_Conversions) | Quickly and reliably, convert between several popular document formats. |
| [Microsoft Word (DOC)](#_Microsoft_Word_(DOC)_1) | The fastest and most complete DOC format implementation you can find. |
| [WordprocessingML (DOCX, XML)](#_Microsoft_Office_Open) | Aspose.Words provides exhaustive support OOXML, Flat OPC and also Word 2003 XML. |

### File Formats and Conversions

#### High-Quality Conversions

The ability to quickly and reliably convert different document formats with a high degree of precision is often enough of a reason to choose Aspose.Words.

With Aspose.Words you can convert documents from any supported load format into any supported save format using just two lines of code:

Example OpenAndSaveToFile

Opens a document from a file and saves it to a different format

[C#]

Document doc = new Document(MyDir + "Document.doc");

doc.Save(MyDir + "Document Out.html");

[Visual Basic]

Dim doc As New Document(MyDir & "Document.doc")

doc.Save(MyDir & "Document Out.html")

##### Fidelity

One of the main goals of Aspose.Words is to provide high-fidelity conversion between document formats. We use the word “high-fidelity” to describe conversions where the output document retains all content and formatting of the original.

Modern word processing document formats are complex and include hundreds of features. Sometimes, when a vendor claims their solution supports a particular document format – it means only the basic features of the format are supported. For example, Microsoft Word documents have advanced features such as footnotes, textboxes, auto shapes, OLE objects, fields, and advanced formatting attributes. Many of these elements often will be ignored in document processing libraries, but not in Aspose.Words.

Aspose.Words’ document object model was designed with Microsoft Word document formats in mind. Therefore, our level of support for DOC, DOCX, RTF and WordprocessingML is unprecedented. Conversion in any direction between these formats is high-fidelity.

Although all of the converters in Aspose.Words are great, not all of them have the “high-fidelity” mark. For example, there is no easy one-to-one match between all features of HTML and Microsoft Word documents. There are no headers/footers, sections, fields and tabs in HTML. Conversely, not all CSS attributes and rules can be mapped to a Microsoft Word document.

Examples of Aspose.Words conversions.

|  |  |
| --- | --- |
| Description | Screenshot |
| **Original DOC file.** |  |
| **Aspose.Words converted DOC to RTF. High-fidelity conversion.** |  |
| **Aspose.Words converted DOC to PDF. High-fidelity conversion.** |  |
| **Aspose.Words converted DOC to ODT. High-fidelity conversion.** |  |
| **Aspose.Words converted DOC to HTML. Not “high-fidelity”, but still very good.** |  |

##### Scalability

Server-side scalable scenarios are where Aspose.Words is used mainly because of its simple and clean design.

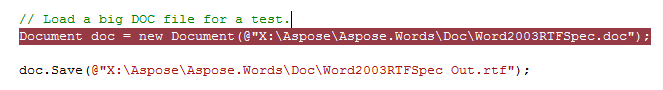
Aspose.Words supports multithreading. [Document](http://www.aspose.com/documentation/.net-components/aspose.words-for-.net/aspose.words.document.html) objects are independent of each other and as long as only one thread at a time is allowed to modify a document (a typical requirement for any collection class), multiple threads can be processing any number of documents at the same time.

##### Speed

Aspose.Words demonstrates high productivity when loading and saving documents. In a test, a DOC file over 5Mb in size was loaded in less than 1 second and then saved in less than 1 second on a P4 3GHz machine. Smaller files can be processed at thousands-per-minute speeds.

On several occasions, our customers have reported to us that they could not believe the speed of Aspose.Words. They stepped over a line of code that loaded a huge document in a debugger and it was so instantaneous that they could not believe the file was loaded into the Document object.

Stepping over a line of code where Aspose.Words loads a big file could be so quick that you might not believe the document was loaded completely.



#### Microsoft Word (DOC)

##### About DOC

|  |  |
| --- | --- |
|  | [DOC](http://en.wikipedia.org/wiki/Microsoft_Word#File_formats) is a word processing document file format that became popular because of its use in all versions of Microsoft Word. It is a proprietary binary format developed by Microsoft. DOC is a not a single format, but a family of formats that evolved with every new Microsoft Word version. |

##### DOC in Aspose.Words

Aspose.Words can read DOC files created by the software listed below. When Aspose.Words writes a DOC file, the same set of software can read it:

* Microsoft Word versions 97 to 2010
* Microsoft Word for Macintosh 98 to X
* Other applications including OpenOffice and AbiWord

##### Comprehensive Support for DOC

The DOC format is very complex because it needs to represent modern word processing documents that have richly formatted content and complex layout. There are hundreds of elements, structures and formatting attributes defined in the DOC format.

The distinct advantage of Aspose.Words is the great extent to which it supports the DOC format. It is hard or impossible to find the same level of support for many important DOC features elsewhere. Aspose.Words for .NET and Aspose.Words for Java support the DOC format equally well.

In addition to all the common DOC features such as paragraphs, tables, styles, lists and fields, Aspose.Words fully supports most of the advanced DOC features:

* Revisions
* Images, textboxes, AutoShapes and group shapes
* Linked and embedded OLE objects, ActiveX controls
* VBA projects with and without digital signatures
* Embedded TrueType fonts
* Encrypted documents
* Documents in right-to-left languages and complex scripts

When shopping for a solution that claims to support DOC files, make detailed enquires about the level to which the DOC features are supported. Create a complex test file and run it through the proposed solution. You will often find that many document elements and formatting will be lost. Shapes, textboxes, fields, columns, OLE objects, revisions, right-to-left text are among the features that usually suffer. Then run the same document through Aspose.Words and enjoy the unmatched completeness of the DOC format implementation.

A complex DOC file generated by Aspose.Words and opened in Microsoft Word.



#### WordprocessingML (DOCX, XML)

##### About WordprocessingML

|  |  |
| --- | --- |
|  | WordprocessingML or WordML is a name for a family of XML-based formats for word processing documents.  WordprocessingML was first introduced in Microsoft Word 2003. WordprocessingML was a significant step by Microsoft towards making the document format open. It is plain XML fotmat.  [Office Open XML](http://en.wikipedia.org/wiki/Office_Open_XML) (OOXML) is the new XML-based format introduced in Microsoft Office 2007 applications. Office Open XML is a container format for several specialized XML-based markup languages. WordprocessingML is the markup language used by Microsoft Office Word to store its DOCX documents. |

##### WordprocessingML in Aspose.Words

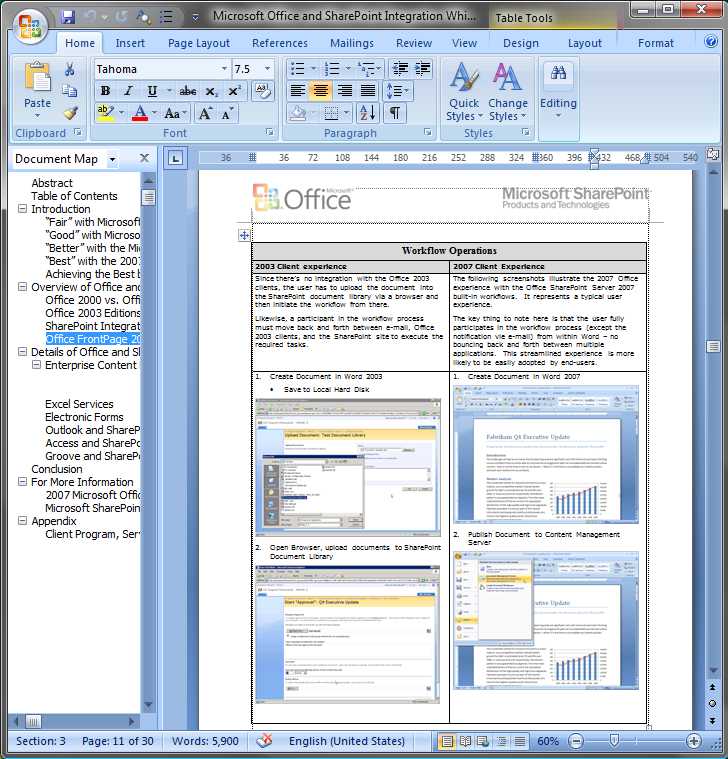
This table explains which "versions" of WordprocessingML are supported by Aspose.Words for .NET:

|  |  |  |
| --- | --- | --- |
| **WordprocessingML “Version”** | **Applicable Standard/Specification** | **Supported** |
| Microsoft Word 2003 | [Microsoft Word 2003 XML](http://en.wikipedia.org/wiki/Microsoft_Office_XML_formats) | Yes |
| Microsoft Word 2007 | [OOXML ECMA-376](http://www.ecma-international.org/publications/standards/Ecma-376.htm) | Yes |
| Microsoft Word 2010 | OOXML ISO/IEC DIS 29500 | Yes |

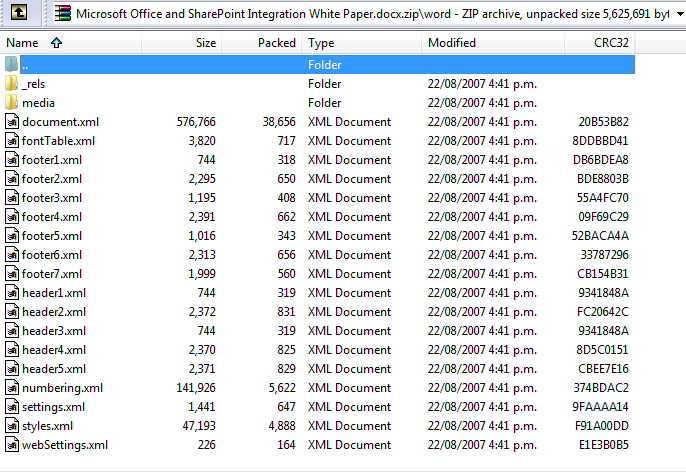
OOXML WordprocessingML documents most often come as DOCX files, which are ZIP packages. In addition to DOCX, Aspose.Words also supports loading and saving OOXML in the “plain XML” [Flat OPC](http://blogs.msdn.com/ericwhite/archive/2008/09/29/the-flat-opc-format.aspx) format.

Aspose.Words provides extensive support for loading, saving and converting WordprocessingML documents. Such all-embracing implementation is possible because Aspose.Words was designed with the structure of Microsoft Word documents in mind (and WordprocessingML is known to mimic the internal representation of Microsoft Word documents).

A DOCX document generated by Aspose.Words and opened in Microsoft Word.



A DOCX document generated by Aspose.Words follows the Open Packaging Convention and can be opened in a ZIP-capable application.



##### OOXML is Open, Why Use Aspose.Words?

Being XML-based, Office Open XML is heralded as an enabling technology. It is true that Office Open XML makes it possible to build document processing and generating applications using just the XML classes without relying on third-party libraries such as Aspose.Words. However, we strongly believe it is still very beneficial to use Aspose.Words when you have to deal with OOXML documents, rather than work through XML or other libraries.

The OOXML specification is several thousand pages long. Being open and standard does not mean being simple. To correctly process or generate OOXML documents one must invest in learning the format well.

In addition to making it simpler to correctly process and generate valid documents, Aspose.Words provides the following important features you would not have when working with OOXML files directly via XML or other third-party libraries:

* Quality conversions between many popular document formats, including conversion to PDF , XPS and printing.
* Ability to build documents from fragments from one or multiple documents, while automatically merging per-document structures such as styles and lists.
* High-level functions such as field update, accept revisions or mail merge can be invoked with just one line of code.
* Access to flat Range-like operations such as find and replace, get/set text of a bookmark, form field, document field or a node.

Consider the following example. It is a simple paragraph that contains text “Hello World” and the word “Hello” is bold. Now imagine you need to write a program that will search for all “Hello World” phrases in the document and replace them with “Goodbye Earth”.

What started out as a seemingly simple task of loading, modifying and saving an XML file does not look so simple anymore. In fact, it has become very complex. It takes a non-trivial algorithm to find and replace flat text across an XML tree. Have you ever wondered why standard XML classes such as **XmlDocument** do not offer find and replace functionality.

A fragment of an Office Open XML document.

<w:p w:rsidR="**00C07F31**" w:rsidRDefault="**003F6D7A**">

<w:r w:rsidRPr="**003F6D7A**">

<w:rPr>

<w:b />

</w:rPr>

<w:t>**Hello**</w:t>

</w:r>

<w:r>

<w:t xml:space="**preserve**">**World.**</w:t>

</w:r>

</w:p>

Implementing even a simple find and replace operation over an Office Open XML document yourself is far from easy. It might be that your boss will enjoy you coding this yourself, but maybe not. Our advice: remember that open and standard does not mean simple, and use Aspose.Words.