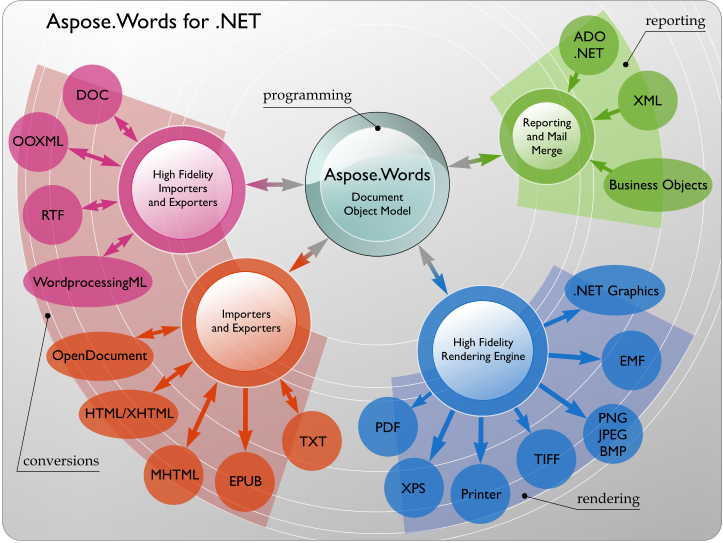
**Evaluation Only. Created with Aspose.Words. Copyright 2003-2020 Aspose Pty Ltd.**

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.