



DOCUMENT AUTOMATION SOFTWARE

## ACTIVEDOCS V MICROSOFT® WORD

Prepared by: Nick Chivers  
Director of Product Marketing

Audience: ActiveDocs Evaluator

Abstract: The purpose of this document is to compare the features and capabilities of ActiveDocs and Microsoft® Word 2007, 2010 and 2013, with particular emphasis on Microsoft® Word's automation potential.

---

**OVERLAND PARK**

Southcreek Office Park  
7301 West 129th Street  
Suite 160  
Overland Park, KS 66213, USA  
Ph +1 913 888 1999

**LONDON**

199 Bishopsgate  
London  
EC2M 3TY  
United Kingdom  
Ph +44 20 3290 1788

**AUCKLAND**

Level 6, 27 Gillies Avenue  
Newmarket, Auckland 1023  
Post: PO Box 289  
Auckland 1140, New Zealand  
Ph +64 9 520 5650

**BRISBANE**

192 Ann Street  
Brisbane, QLD 4000  
Post: PO Box 604  
Paradise Point QLD 4216, Australia  
Ph +61 7 3040 6616



## Copyright

Information in this document is subject to change without notice. Companies, names, and data used in examples herein are fictitious unless otherwise noted. No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without the express written permission of ActiveDocs Limited.

Copyright © ActiveDocs™ Limited. All rights reserved.

Microsoft is a registered trademark and Microsoft SQL Server, Microsoft Access, Microsoft Outlook, and Microsoft Windows are trademarks of Microsoft Corporation in the United States and/or other countries.

Other product and company names herein may be the trademarks of their respective owners.

**Disclaimer:** While ActiveDocs has taken care to ensure the accuracy and quality of this document, all content including fitness for a particular purpose are provided without any warranty whatsoever, either expressed or implied. In no event shall ActiveDocs, or its employees, be liable for any direct, indirect, incidental or consequential, special or exemplary damages resulting from the use of this document or from the use of any products described in this guide. Any persons or businesses mentioned within this document are strictly fictitious. Any resemblances to existing or deceased persons, or existing or defunct businesses, are entirely coincidental. This document will be updated regularly and changes will be included in later versions. If you experience any discrepancies in the content of this document, please e-mail [info@activedocs.com](mailto:info@activedocs.com).



# Contents

<b>1</b>	<b>Introduction .....</b>	<b>1</b>
<b>2</b>	<b>Microsoft Word 2007 .....</b>	<b>2</b>
<b>3</b>	<b>Detailed Comparison .....</b>	<b>3</b>
3.1	General .....	3
3.2	Template Design.....	3
3.3	Interactive document generation from Templates .....	4
3.4	Programmatic/Unattended document generation from Templates.....	6
<b>4</b>	<b>Key Points.....</b>	<b>7</b>

# 1 Introduction

The purpose of this document is to compare the features and capabilities of ActiveDocs and Word 2007, 2010, and 2013, with particular emphasis on Word's automation potential.

## 2 Microsoft Word 2007, 2010, and 2013

A Word document or template is a Zip file, or package, containing a set of XML files, image binaries, and any other resources required for the document. It seems that Word itself can be seen as just one of the available ways of writing, reading, and presenting Word files, albeit a convenient one.

Open XML, and the easily-manipulated Word file structure, enables less difficult programmatic interaction with the document. It is not necessary to use Word itself to generate a document or template from scratch or when using a template to create a document. However, this ignores functions like repagination, updating field codes, Table of Contents, printing, etc, which cannot easily (if at all) be replicated without using Word.

The key to the Word programmability is the custom XML Data Store which allows elements in the document to be mapped to, and set and retrieved via, XML. Any Zip manipulation library (including Microsoft's own NetFX System.IO.Packaging) can be used to access the contents of this store and thus expose its contents for programmatic manipulation. When the updated package/file is opened in Word, the changes are simply there.

For unattended document generation, XML mapping means that programmatic generation from templates might be automated in a similar way to ActiveDocs Composition Server's automated assembly engine. The caveat is that ActiveDocs Composition Server provides a full support and interface infrastructure which would have to be developed separately if another solution were chosen.

For interactive document generation, some ActiveDocs Composition Server functionality is replicated in Word where elements referenced in the XML Data Store can be controls – e.g. drop-down lists, dates with pickers, etc. Word allows these controls to be used at document creation time, but they aren't rendered as controls when the document is presented or printed. Use of these controls in Word cannot be structured and controlled. By contrast, an ActiveDocs Template provides precise structure and control.

Word's new controls are an improvement and likely to be popular with document designers, especially in combination with a degree of validation and new features that allow regions of the document to be made un-modifiable and/or undeletable. One caveat is that the controls appear to work in isolation so repeating the same data value, such as a person's name, in multiple controls could prove complex. A second caveat is that external data sources can be used but only by integrating an InfoPath form into the document or by involving software developers in the template creation process – something that ActiveDocs is specifically designed to avoid.

This means that some ActiveDocs Active Field types and some related functionality can be replicated in Word. Other ActiveDocs functionality such as structured interaction, rules, conditional content, the ability to re-use Active Fields in a Template to automatically repeat the same value in multiple locations in a document, and the extension of these features to Template Sets, is not so easy to replicate. Features that Word on its own cannot provide are ActiveDocs' well-structured Designer environment, full support for reusable Template components, Wizard-driven interactive document creation, and the ActiveDocs Composition Server's full support infrastructure and API.

## 3 Detailed Comparison

This section of the document outlines differences between Word and ActiveDocs in the context of creating documents from templates. Where the products or their capabilities are broadly similar, comparisons are not made.

### 3.1 General

ActiveDocs Designer and ActiveDocs Composition Server make a complete integrated package, providing:

- Fully structured Template design with data access
- Full support for component-based Template design and deployment, including relationship and dependency management
- Template management including package-based deployment
- Security and management of user access to Templates and documents
- Profile fields associated with users and the (optional) business areas to which they have access, providing pre-set values for incorporation into finished documents.
- Web interface
- Template storage and retrieval
- Template Set handling
- Structured interactive document creation interface (the Document Wizard)
- Answer storage and re-use
- Document assembly and production in multiple formats including optional PDF
- Scale-up and scale-out capability
- Document storage and retrieval
- Document delivery through multiple channels
- A well-defined programmatic interface for document generation, answer data consumption, etc, allowing integration with other applications.

Word is not a package. The build-or-buy argument always applies if a complete solution like ActiveDocs is required. Out of the box, Word and/or its file format provide only:

- Basic template authoring
- Unstructured interactive document creation
- The possibility of programmatic document generation and manipulation

### 3.2 Template Design

In the ActiveDocs design environment, Templates are one type of “Design Component” supporting the key concepts of consistency and reusability. Templates themselves can be incorporated into Template Sets to enable creation of single or multiple documents from combinations of Templates. Templates, in turn, may be built from other reusable design components such as: Data Views, which provide design-time and run-time linkages to corporate and other data sources; Snippets, which may be thought of as Template fragments for encapsulating common content, both active and passive; Design Item Sets, where the key building blocks of Active Fields and Rules may be created and re-used.

ActiveDocs Designer provides two powerful interfaces for Template design and management. The first, Content Manager, provides a rich interface for the management of Templates and other reusable Design Components across the enterprise. The second, Design Tools, is a Microsoft Word add-in that leverages the familiar Word environment for Template development while adding full support for the rich ActiveDocs feature set.

ActiveDocs Designer allows input fields to be grouped, allowing associated data items to be kept together. This gives structure to the Template design process and more importantly to the interactive document creation process via the ActiveDocs Document Wizard. Word has no apparent way of structuring the data requirements independently of the structure of the document.

To allow users to select from online data during document creation from a template in Word requires the integration of an InfoPath form into the template, or the involvement of a software developer. In turn, this requires the InfoPath product and InfoPath skills to create and integrate the form, or the template creation and maintenance process becomes heavily dependent on software development skills. ActiveDocs provides immediate linkage to data sources via its own built-in facility (the Data View) and allows data to be cached in the Template, providing useful offline functionality. ActiveDocs is specifically designed to put Template design and management in the hands of the business and to remove unnecessary dependence on software development or other IT skills.

ActiveDocs' rules feature allows groups and document fragments (from characters to whole documents) to be optional, powerfully enhancing the Template or Template Set with optional processing based on run-time data values. "Optional processing" refers to both data capture and the inclusion or exclusion of content from the finished document, and applies to sets as well as single documents. Word does not directly support such use of rules and optional content, requiring complex customised solutions instead.

An ActiveDocs Active Field can be used in multiple locations in a Template and in a Template Set. During interactive document creation the field will only be prompted for once, and during automated document creation it only needs to be supplied once. In Word, equivalent fields rendered as controls need additional programming to ensure that one value can be repeated through the document, and extending such functionality to template sets adds another layer of complexity.

ActiveDocs provides powerful calculation features for setting field default values and can reference and use other field values in these calculations. It can also use the values of Active Fields to dynamically filter data – so for instance the selection of a customer means that only that customer's invoices are displayed for retrieval.

ActiveDocs allows user and business area (subsidiary) profile data to be included in Templates as fixed or default values. Profile data is customised per installation but might usefully include the user's name, job title, and phone number, and the business area's name and address.

ActiveDocs provides Repeating Items, typically for lists rendered as table rows where the actual number of repetitions cannot be known at design time.

### 3.3 Interactive document generation from Templates

To use a Word solution, every user needs access to the Word application itself. In essence, this means that every user needs Word on the desktop. ActiveDocs provides the Express Wizard – a web interface for choosing Templates – and the Document Wizard – which parses and runs any ActiveDocs Template to elicit answers from the user – and the Composition Server which runs the Document Compiler to create documents from the Template and the answers, returning each completed document to another web page where the user can view, deliver, download, and otherwise manage them. In respect of those documents Word is only required on the desktop if the documents are to be downloaded and modified there.

In Word the end user needs access to the template itself and can modify the content and presentation of any unprotected areas. Templates can be obtained from anywhere the user finds them, even out of date versions. ActiveDocs provides centrally managed Template access through the Express Wizard browser interface, and its interactive Document Wizard abstracts the Template and only presents and prompts for information that is required. The remainder of the content, and all presentation, is kept intact.

ActiveDocs allows user and business area profile data to be included in Templates as fixed or default values. Profile data is customised per installation but might usefully include the user's name, job title, and phone number, and the business area's name and address. This means that such information may be automatically included in the document.

ActiveDocs Designer allows the input fields (Active Fields) to be grouped, allowing associated data items to be kept together. This gives structure to the Design process and more importantly to the interactive document creation process. The ActiveDocs Document Wizard follows the designer's group structure and prompts for each group on a separate page, and it prompts for every required field.

An ActiveDocs Active Field can be used in multiple locations in a Template and in a Template Set. During interactive document creation the field will only be prompted for once. In Word, equivalent fields rendered as controls would need additional programming to ensure that one value could be repeated through the document, and extending such functionality to document sets would add another layer of complexity.

Word provides no structure for creating the document from the template, requiring instead that the user browse the entire template looking for items to complete, and complete every instance of repeated information. To do these effectively and accurately means the user needs familiarity with the document, whereas ActiveDocs can be used efficiently by a user who has never seen the document.

In Word, access to data sources is only possible by integrating InfoPath, a separate product requiring different designing skills, or by involving specific software development expertise in the creation and maintenance of the template. ActiveDocs is able to provide immediate access to available ODBC/OLE sources through Data Views, the design and maintenance of which can be completely in the hands of the business.

ActiveDocs' calculation feature means that default values can be calculated on the fly while the user is creating the document, even using values already entered. The use of default values can significantly reduce the time and effort of creating new documents.

ActiveDocs' Repeating Items allow any number of repetitions, typically for lists or table rows (e.g. invoices).

ActiveDocs allows groups and document fragments (from characters to whole documents) to be optional, powerfully enhancing the Template or Template Set with optional processing based on run-time data values. "Optional processing" refers to both data capture and the inclusion or exclusion of content from the finished document, and applies to sets as well as single documents.

ActiveDocs automatically handles Template Sets very efficiently with optimised data capture, and allows entire documents in the set to be included or excluded based on run-time data values.

ActiveDocs Composition Server's server-based document assembly and delivery is multithreading-capable and is scalable up and out. It is capable of interacting with other applications, which may for example use ActiveDocs' Document Wizard to create documents interactively. Other applications may also provide and consume answer data (e.g. for updating databases, perhaps to implement full round-trip support), and receive completed documents.

ActiveDocs stores and allows reuse of answers. Word would require external functionality to achieve this.

### 3.4 Programmatic/Unattended document generation from Templates

An ActiveDocs Active Field can be used in multiple locations in a Template and in a Template Set. During automated document creation it only needs to be supplied once. In Word, equivalent fields (whether rendered as controls or not) would need additional programming to ensure that one value could be repeated through the document, and extending such functionality to document sets would add another layer of complexity.

ActiveDocs allows groups and document fragments (from characters to whole documents) to be optional, powerfully enhancing the Template or Template Set with optional processing based on run-time data values. “Optional processing” refers to both data capture and the inclusion or exclusion of content from the finished document, and applies to sets as well as single documents.

ActiveDocs automatically handles Template Sets very efficiently and allows entire documents in the set to be included or excluded based on run-time data values.

ActiveDocs Composition Server’s server-based document assembly and delivery is multithreading-capable and is scalable up and out. It is capable of interacting with other applications, which may for example use Composition Server’s web service to create documents automatically, provide and consume answer data (e.g. for updating databases, perhaps to implement full round-trip support), and return documents to the other application.

The Word file format allows programmatic generation of new documents from scratch without using Word (although there are non-trivial considerations such as repagination, updating field codes, Table of Contents, printing, etc). While third-party software components have long been available to do this, the apparent benefit of the Open XML Word paradigm is the way document contents may be manipulated programmatically.

This could be seen as putting Word on a par with ActiveDocs but there are important differences. An application sends Job XML (typically specifying the Template, data values, and the output format & channel) to ActiveDocs Composition Server’s web service which handles the assembly, conversion, archiving, and delivery of the document. Without ActiveDocs, an application opens a template, programmatically modifies its contents, saves the result as a document, saves it to a specified location, then programmatically redirects it to a printer, fax, or email destination, etc; if this is done without Word itself, the application may also be required to repaginate the completed document and update its fields (including TOC) programmatically – a practical impossibility.

There is a core software development requirement in both solutions, that the application must map data to XML tags. The remainder of the work is clearly easier with the ActiveDocs solution, using the ActiveDocs Composition Server web services interface which neatly abstracts the mechanics of the assembly, storage, and delivery while providing web-based accessibility.

Of course, an organisation wishing to use the Word solution would soon identify the need for a generic interface allowing an application to pass a template name, data values, and delivery requirements, to a network or Web service... sounds familiar? Word on its own is insufficient, considerable additional software development is required, and the usual build-or-buy considerations now apply.

Even supposing that an organisation was willing to go down the “build” route, it must be kept in mind that: (1) templates can also be used interactively and the ActiveDocs interactive solution is arguably superior; (2) ActiveDocs Composition Server offers more than just automation support; and (3) a custom automation solution would likely provide little more (and probably less) than ActiveDocs already provides. Does an organisation want different solutions for interactive and automated document creation?

## 4 Key Points

ActiveDocs is an integrated and complete solution from Template design to document delivery.

ActiveDocs Designer supports modular Template design and enables a wide range of input field types and a variety of ways to populate them, including integrated access to external data sources. Word's controls cover some of the same ground, but Word needs a separate product (InfoPath) to enable access to external data sources.

ActiveDocs Composition Server provides out-of-the-box user security management, Template access and management, online document creation, document assembly, document delivery, document access, and answer data storage and re-use. Document creation/assembly works generically with any ActiveDocs Template, interactively or automatically. ActiveDocs Composition Server also provides for seamless integration with other applications.

ActiveDocs uses field grouping and conditional processing to provide structure to the Template, which improves the interactive experience. ActiveDocs Document Wizard abstracts the Template, and just focuses on the data required to create the document. Word provides no structure and requires the end user to interact with the template itself.

ActiveDocs supports conditional processing, optional content, and re-use of fields with optimised data capture, for single Templates and for Template Sets.

Of itself, Word only provides basic template creation and unstructured document creation from templates. The Word file structure and open XML offer better programmatic document generation potential than older versions, but the build-or-buy argument becomes relevant if the Word solution is to be integrated at all with its ICT environment.

Word users need Word on their desktop and direct access to the templates to interactively create documents, with the usual application rollout/management and template version problems. ActiveDocs document creation is a centralised server application, end users don't necessarily require direct access to the Word application, and they don't need direct access to the Templates to interactively create documents.

In conclusion, Microsoft Word – or, mostly, its file format – may have the potential for better templates and easier programmatic manipulation than its predecessors, but it still lacks most of the functionality of the ActiveDocs suite and is no closer than its predecessors to providing a full and integrated solution for document automation.