



Document Excellence through Innovation

ACTIVEDOCS OPUS or MICROSOFT TECHNOLOGIES for DOCUMENT AUTOMATION

Prepared by: Nick Chivers
Director of Product Marketing

Audience: ActiveDocs Evaluator

Abstract: A comparison of the Document Automation capabilities of ActiveDocs Opus and a collection of Microsoft® technologies.

KANSAS CITY

5921 NW Barry Rd
Suite 202
Kansas City, MO 64154, USA
Ph +1 816 746 1999
Fax +1 816 746 1997

AUCKLAND

Level 6, 27 Gillies Ave
Newmarket, Auckland 1023
Post: P O Box 289
Auckland 1140, New Zealand
Ph +64 9 520 5650, Fax +64 9 520 4944

BRISBANE

192 Ann Street
Suite 150
Brisbane, QLD 4000
Australia
Ph +61 7 3040 6616

info@activedocs.com | www.activedocs.com





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 © 2000-2011 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.



Contents

1	Executive Summary	1
2	Introduction	2
3	ActiveDocs Opus	3
3.1	ActiveDocs Opus Document Automation	3
3.2	Technology Base	3
4	Microsoft Technologies	4
4.1	The Open XML Document Format.....	4
4.2	Open XML SDK.....	4
4.3	Microsoft SharePoint.....	4
4.4	Word Automation Services.....	4
4.5	Microsoft InfoPath and Forms Server	5
4.6	Microsoft Word	5
5	Document Automation Features	6
5.1	Modular Design	6
5.2	Template and Snippet Design in Microsoft Word.....	7
5.3	Design Management, Version Control, and Publishing Control.....	8
5.4	Security and management of user access.....	9
5.5	Web interface for Document Creation and Management	9
5.6	Wizard-driven browser-based interactive document creation	9
5.7	Server-side Document Assembly.....	10
5.8	Document and Answer storage.....	11
5.9	High-performance scale-up and scale-out capability	11
5.10	Managed document delivery through multiple channels.....	12
5.11	A well-defined programmatic interface for document generation	12
5.12	Integrated support for multiple workflows	12
6	Conclusion.....	14

1 Executive Summary

This paper compares ActiveDocs Opus with a collection of out-of-the-box Microsoft Technologies with a view to explaining why ActiveDocs Opus is a cost effective and less risky way of building maintainable Document Automation Solutions for the enterprise.

ActiveDocs Opus is made possible by a large number of enabling technologies found in Microsoft Office and Server products. Creating an enterprise-class Document Automation solution is only possible with significant development resource and choosing ActiveDocs Opus over using Microsoft Technologies comes directly down to a Buy-versus-Build decision. To build such a solution should only be attempted by organisations with significant Microsoft Office Development expertise and a project timeline and budget which can accommodate the Development effort required.

2 Introduction

There is a simple concept at the heart of Document Automation which is usually expressed as “template + data = document”. Implementing Document Automation in an enterprise environment requires attention to every aspect of that concept.

- **Template** development and maintenance must be supported by a secure and robust environment that encourages a modular approach to reduce redundant and repetitive content. The environment must balance responsiveness with risk mitigation. Responsiveness because government regulations and other external drivers cannot wait; risk mitigation because your organisation’s legal teams (and perhaps marketing, and management, and...) won’t thank you for publishing changes to sensitive information unless they can see it and approve it first.
- **Data** must be easy to capture or made securely accessible for document generation, to impose validation, to eliminate unnecessary data transcription, and to mitigate the downstream risks associated with inaccurate transcription and other errors.
- Interactive **document creation** with the Document Automation solution must be an attractive, accessible, secure, and easy-to-use option for end users, who will otherwise fall back into error-prone and potentially ruinous practices such as cut-and-paste from old documents, or creating poorly branded documents from scratch.
- Programmatic **document creation** should be fully supported using open standards such as web services without necessitating different template design methods.
- Post-creation **document processing** such as approval, finalisation, variable formatting, and flexible storage and delivery options, should be supported.

The purpose of this document is to compare the Document Automation capabilities of ActiveDocs Opus and a collection of Microsoft® technologies which includes:

- Word & other Office applications
- SharePoint
- Word Automation Services
- InfoPath
- The Open XML standard and the Open XML SDK

3 ActiveDocs Opus

3.1 ActiveDocs Opus Document Automation

ActiveDocs Opus is a purpose-built Document Automation solution which has evolved from experience with hand-coded automation in Word templates, to the development of a business-user-oriented Desktop product to perform the same tasks, and two generations of truly enterprise-grade Server-based suites.

ActiveDocs Opus is a complete integrated package for creating and using document Templates, providing:

- Modular Design
- Design in Microsoft Word
- Design Management, Version Control, and Publishing Control
- Security and management of user access to templates and documents
- Web interface for Document Creation and Management
- A structured, interactive, web-based document creation interface – the ActiveDocs Opus Document Wizard – which is dynamically and automatically created from each template on demand
- Document and Answer storage in a variety of locations and applications including Microsoft SharePoint and WebDAV-compliant applications
- Document assembly in multiple formats including all Microsoft® Word formats, XPS, PDF, ODF(T), HTM, TIFF, and Named Drivers
- High-performance scale-up and scale-out capability
- Document delivery through multiple channels
- A well-defined programmatic interface for document generation, answer data consumption, etc, allowing integration with other applications
- Integrated support for multiple workflows for Document Control, Approval, and Finalization

3.2 Technology Base

ActiveDocs Opus is developed using Microsoft development tools on, and for, the Microsoft .NET platform. The application uses Microsoft's Open XML SDK.

ActiveDocs Opus is designed for server-side deployment on Microsoft Windows Server 2008 and 2003 and client-side deployment on Microsoft Windows 7, Vista and XP.

ActiveDocs Opus stores its data on Microsoft SQL Server 2008 and 2005, integrates with Microsoft Word for Template Design, and integrates with other Office applications for optional client-side document creation.

4 Microsoft Technologies

A number of Microsoft technology products could be relevant to a Document Automation solution:

- The Open XML document format
- Open XML SDK
- SharePoint
- Word Automation Services
- InfoPath and Forms Server
- Word

None of the products individually can provide an integrated, secure, controlled, enterprise-grade solution. A potential architecture is:

- SharePoint for the template repository and general user environment
- Word and/or the Open XML SDK for template design
- InfoPath / Forms Server for interactive web-based data capture
- Word Automation Services and the Open XML SDK for document generation

4.1 The Open XML Document Format

An Open XML document or template is a package containing a set of XML files, image binaries, and any other resources required for the document. An Open XML document can be created or altered programmatically.

4.2 Open XML SDK

The Open XML SDK is a Software Development Kit for working with Open XML format documents. This product is for software developers and is designed to allow the development of custom-built applications that can manipulate Open XML documents. ActiveDocs Opus uses parts of the Open XML SDK.

4.3 Microsoft SharePoint

Prior to the launch of Microsoft's "2010" product line-up, the term "SharePoint" referred to two products, Windows SharePoint Services (WSS) and Microsoft Office SharePoint Services (MOSS). For 2010, these are rebranded as SharePoint Foundation Server 2010 and SharePoint Server 2010 respectively. The term "SharePoint" will, presumably, continue to refer to two products.

ActiveDocs Opus integrates with SharePoint as a repository for Templates, Snippets, and Documents.

4.4 Word Automation Services

Word Automation Services is a feature of SharePoint 2010. Word Automation Services enables server-side conversion of document formats without using Word itself on the server.

ActiveDocs Opus and its predecessor, ActiveDocs Enterprise, have used Word on the server reliably and robustly for many years. Improvements to ActiveDocs Opus Composition Server's document assembly engine, combined with the downsizing of Word files since the introduction of the Word 2007 formats, have resulted in a benchmark of 135,000 documents per hour including conversion to DOC format using Word on the server.

ActiveDocs will use Word Automation Services when its performance, reliability, and other capabilities, are proven to match or exceed what ActiveDocs can do with Word.

4.5 Microsoft InfoPath and Forms Server

InfoPath is a forms tool that enables the design, creation, and use of sophisticated XML-based data capture forms that can reference external data stores as both sources and destinations for data. InfoPath 2007 introduced development of forms for so-called “symmetric” deployment in both Windows client (desktop) and server (web) environments, subject to a few design element restrictions.

Forms Server, also known as Office Forms Services or Office Forms Server, integrates with SharePoint. Using these services, InfoPath web forms may be accessible through intranets, extranets, and the internet.

4.6 Microsoft Word

Word is a ubiquitous and highly-capable product for word processing, potentially making it an ideal choice for template development and possibly as an environment in which document automation is hosted.

5 Document Automation Features

This section explains in detail the ActiveDocs Opus features listed in section 2 and considers how those features could be replicated in a customised solution using the Microsoft technologies listed in section 3.

5.1 Modular Design

ActiveDocs Opus achieves modular design through the use and reuse of Design Components.

- **Data Views** enable access to data sources via common standards like ODBC, OLE-DB, and Web Services. Data Views can access tables and views, may optionally use custom SQL and Stored Procedures (in applicable data sources e.g. SQL Server databases), and can filter incoming data.
- **Design Item Sets** encapsulate common **Design Items** such as **Active Fields** for data, **Rules** for business logic, **Snippet Links** for common content, **Repeating Item Groups** for multiple-instance data sets, and **Graphs**. Design Item Sets can use **Data Views** and thereby link common Active Fields to data sources.
- **Snippets** encapsulate common content. Snippets may be static content, or may contain ActiveDocs Opus technology including **Data Views**, local **Design Items**, and **Design Item Sets**.
- **Templates** are the core component. Templates, like **Snippets**, may be static content, or more usually contain ActiveDocs Opus technology including **Data Views**, local **Design Items**, and **Design Item Sets**. Templates, via local **Snippet Links** or via Snippet Links from Design Item Sets, use Snippets to provide common content.
- **Template Sets** are combinations of multiple **Templates**. Template Sets can use **Rules** to control the inclusion or exclusion of individual Templates when creating the document. Templates may be repeated to enable mail-merge or similar operations. Template Sets can specify how section breaks are applied when composite documents are built from multiple Templates.

Replication using Microsoft Technologies

An equivalent solution using a customised application based on Microsoft Technologies might replicate “templates” and “snippets” using Microsoft Word (see below). It is possible to envisage a partial replication of the Template Set / Template / Snippet structure that would leverage Word’s “Building Blocks” functionality, but the complete structural control and dependency management afforded by ActiveDocs Opus Designer’s Content Manager (see below) would be absent.

The closest that a Microsoft Technologies solution comes to replicating both Data View and Design Item Set functionality is by incorporating InfoPath forms into Word templates, or by using InfoPath as the primary front-end.

For the purposes of Template design, neither option is business-user-friendly. Both options are discussed under some of the following headings.

5.2 Template and Snippet Design in Microsoft Word

ActiveDocs Opus Design Tools is a Microsoft Word add-in which allows Designers of Templates and Snippets to add, modify, remove, visualise, deploy, and browse ActiveDocs Opus functionality in a WYSIWYG environment. ActiveDocs Opus functionality can be deployed throughout the entire Template or Snippet, including headers and footers.

The primary design element is the Active Field, a placeholder for data. Active Fields have a type (e.g. text, date, number, image, etc), optional presentation formatting, may carry validation (e.g. mandatory/optional, range checks, etc), and can be linked to preset lists or external data sources (via ActiveDocs Opus Data Views) with optional contextual filtering.

Designers can create Groups of Active Fields. Grouping has manifold benefits:

- Logical association of related fields
- Business logic (Rules) can be applied to Groups
- Grouping determines how fields are collectively presented during interactive document creation
- Repeating Item Groups are supported for multiple-instance data such as lists and table

Rules are powerful implementations of business logic based on current data, dynamically. Rules can include or exclude Groups (see above) and/or parts of the output document, from single characters to entire sections. Rules can be evaluated for both True and False, and can be nested to multiple levels.

The “interview” for interactive document generation is created automatically from the Template/Snippet based on the Groups and Rules and other ActiveDocs Opus functionality incorporated in the Template.

Replication using Microsoft Technologies

The obvious candidate is Word, optionally partnered with the Open XML SDK and/or InfoPath.

There are features in Word which allow for the replication of some ActiveDocs Opus functionality such as fields with drop-down lists, dates, etc, with some degree of validation. Word allows these controls to be used at document creation time, but they are not 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 Opus Template provides precise structure and control.

Controls appear to work in isolation so repeating the same data value, such as a person’s name, in multiple controls could prove complex (ActiveDocs Opus allows fields to be used in multiple places). External data sources can be used but only by integrating an InfoPath form into the document and by involving software developers in the template creation process. Neither of these is likely to be attractive to business users; ActiveDocs Opus is specifically designed to work for business users by abstracting the technicalities.

The Open XML SDK provides the capability to add controls and validation programmatically. The SDK is designed for Software Developers and would require the decidedly non-trivial development of a separate user interface for creating, selecting, and positioning such controls on the document.

Regardless of how the controls are added, for the end user interview, Word provides no structure. The only solution for structured and controlled interactive use is to bypass the Word functions and use InfoPath instead to create a structured interview process.

This is not a business-user-friendly approach and creates hurdles for future maintenance because the “interview” and the “document” must be kept aligned and there is no automatic way of doing this unless there is programmatic control of the InfoPath and template design – in other words, the imposition of a higher layer of control and the complexities which that introduces.

Furthermore, InfoPath only enables the creation of PDF output based on the form and its data. To create Word documents, InfoPath requires the use of Microsoft’s Visual Studio Tools for Office. Worked examples that show the complexity involved in creating Word documents from InfoPath are at <http://enterprise-solutions.ikreate.nl/infopath2007/convert-infopath-form-into-word-2007-document-xslt-c-sharp.htm> and <http://blogs.3sharp.com/davidg/archive/2008/08/08/4935.aspx>. It goes without saying that this, also, is not a business-user-friendly solution.

5.3 Design Management, Version Control, and Publishing Control

ActiveDocs Opus Content Manager is a user interface designed to allow Template Designers and Administrators to work easily with large or small **Catalogs of Design Components** across multiple environments (e.g. Development, Testing, and Production).

Content Manager uses a tree structure to show the various Design Component types, which may be subdivided by Folder structures and viewed alternatively via cross-type Categories.

Content Manager incorporates Check-Out/Check-In and version-aware Dependency Management.

Content Manager integrates with **ActiveDocs Opus Version Control** to provide version management and version history for all Design Components. Version Control incorporates **Publishing Control** with Publish Now and Deferred Publishing options. Publishing integrates with **Design Approval** so that changes to the content of Snippets and Templates may be automatically referred to nominated experts for pre-Publishing Approval.

ActiveDocs Opus Version Control also supports the recreation of a document using the exact versions of Templates, Snippets, Data Views and Design Item Sets with which it was originally created.

Replication using Microsoft Technologies

Microsoft SharePoint is the obvious candidate for replicating Content Manager’s interface, if not all of its functions (given the inability of a Microsoft Technology solution to support all of the ActiveDocs Opus Design Component types).

Although SharePoint 2010 supports relationships between multiple lists (and could therefore support some degree of dependency management) it is inevitable that customised development would be required to ensure that dependencies could be recognised at all (a necessary precursor to managing them). Version-awareness would add another layer of complexity to dependency management.

It is worth pointing out that ActiveDocs Opus Version Control extends to version control of dependencies, a feature specific to ActiveDocs Opus. While basic Version Control is common, and products like SharePoint support versioning, the requirement to manage versioning when there are dependant items all with their own version history add another

significant level of complexity that is not managed at all in SharePoint. It is this extended Version Control that allows ActiveDocs Opus to recreate a document using the exact versions of Templates, Snippets, Data Views and Design Item Sets with which it was originally created.

Combining any workable Publishing options and Design Approval would require customised development of the SharePoint 2010 solution.

5.4 Security and management of user access

ActiveDocs Opus can integrate directly with Windows Active Directory. Alternatively, it has its own security module or can be integrated with third-party security systems. Whatever method is used, users belong to groups and subsites, and the groups and subsites provide the rights and roles for the users.

Design rights – access to different classes of Design Components – and document rights – for creation etc – are all finely controlled.

Replication using Microsoft Technologies

It is likely that an acceptable level of user access management and security could be implemented in a SharePoint-based solution.

5.5 Web interface for Document Creation and Management

The **ActiveDocs Opus Express Wizard** is the web interface for administrators and end users, for document creation and post-creation document actions, and for some other functions. All functions are security controlled.

Document creation is effected via the ActiveDocs Opus Document Wizard (see below). After creation, the document may be subject to Approval and/or Finalization workflow (see below), and can be downloaded, edited, saved, printed, re-assigned, deleted, or purged, depending on configurations and permissions set in the system.

Options for storage are outlined below, and depending on the configuration, documents stored outside of ActiveDocs Opus may still be accessed from within it.

Other functionality available through the Express Wizard includes common administrative functions and reporting. ActiveDocs Opus ships with a number of useful reports for analysing and managing document creation and other aspects of system usage. Customised reporting is also supported and the relevant parts of the ActiveDocs Opus database schema are explicitly published for that purpose.

Replication using Microsoft Technologies

It is likely that an acceptable level of user access for some of these functions could be implemented in a SharePoint-based solution, provided that the functions themselves could be adequately replicated.

5.6 Wizard-driven browser-based interactive document creation

The **ActiveDocs Opus Document Wizard** is rendered automatically from ActiveDocs Opus Templates or Template Sets, and presents a series of forms based on the Active Field Groups defined by the Template(s) and any Snippet(s) that they reference.

Groups may be Rule-driven and therefore may be bypassed dynamically based on current data.

The Document Wizard includes the ability to load previous Answers (i.e. the responses given in previous Document Wizard interviews) and includes a pop-up Navigation pane to allow the user to move easily between groups.

The Document Wizard supports a high degree of additional customisation and integration via pre-populated answer sets, launching as a pop-up from other applications, and start / step / finish / complete event handling.

Replication using Microsoft Technologies

As noted, any form of structured interview for interactive document creation will necessarily involve the use of InfoPath. In turn, this requires Forms Server.

Forms Server, also known as Office Forms Services or Office Forms Server, integrates with SharePoint. Using these services, InfoPath web forms may be accessible through intranets, extranets, and the internet, in much the same way as the ActiveDocs Opus Document Wizard.

While the limitations of InfoPath with regard to output document formats and the significant effort required to keep the “interview” and the document aligned have been noted, it is likely that the functionality of the Document Wizard could be replicated in InfoPath, albeit with significant effort and only on a per-template basis. A generic solution would require the development of a significantly complex overarching application.

Any such solution would carry a large caveat related to a significant weakness with InfoPath: if the form changes, then the underlying XML schema changes, which renders any historical data invalid when used to regenerate forms and severely compromises the reuse of the “answers” used to create other or previous documents. This is in marked contrast to ActiveDocs Opus where the maximum possible reuse of answers is deliberately supported, even between different generations of the product and its predecessors.

5.7 Server-side Document Assembly

ActiveDocs Opus Composition Server assembles documents on the server. Documents are created in the latest Word format by default, using elements of the Open XML SDK to do so and without using Word itself.

Conversion to other formats, including all previous Word formats, XPS, PDF, ODF (Text), HTM, TIFF, and Named Drivers, is achieved by using a single well-managed instance of Word on the server.

Improvements to ActiveDocs Opus Composition Server’s document assembly engine, combined with the downsizing of Word files since the introduction of the Word 2007 formats, have resulted in a benchmark of 135,000 documents per hour including conversion to DOC format using Word on the server.

ActiveDocs will use Word Automation Services when its performance, reliability, and other capabilities, are proven to match or exceed what ActiveDocs products can do with Word on the server.

Replication using Microsoft Technologies

The obvious candidates are the Open XML SDK for ‘native format’ document assembly and Word Automation Services for format conversion. Together, these technologies are suggested as a 1:1 replacement of the existing desktop object model (Microsoft Word Team Blog at http://blogs.msdn.com/microsoft_office_word/archive/2009/10/26/introducing-word-automation-services.aspx).

It is important to note that document creation via the Open XML SDK, and document conversion via Word Automation Services, can only be achieved by the creation of custom software applications which will use those technologies to create and convert documents.

It is also important to note that Word Automation Services is said to only support the same output formats as Word itself. It is not clear whether any specialised conversion such as “Named Drivers” will be supported directly (implying that the product could not only perform document conversion but could also replicate Word’s use of output devices) or would require additional customisation.

Perhaps the most important point is the apparent disconnection between the preference for a well structured InfoPath-based interactive document creation scenario, and the use of server-side document assembly (which does not use InfoPath). The InfoPath solution would need to be supported or ‘completed’ by a server-side assembly process. The real challenge, however, is keeping the document’s data requirements, and the InfoPath forms’ functionality, aligned.

This last point is no issue for ActiveDocs Opus, as the data requirements and the interactive use of the Template are fully integrated through a common set of fields and groups.

5.8 Document and Answer storage

Dependent on certain settings and/or workflow configurations, documents and answers may be stored in ActiveDocs Opus or forwarded automatically for storage in other locations and/or applications (including WebDAV-compliant systems, SharePoint libraries, shared folders, and named Web Services) directly or via Web Services.

Using workflow, it is possible to dynamically alter the document storage settings (and, in fact, any workflow settings) on a per-document basis. Thus, for example, Sales Proposals for particular Customers and/or for sales above or below certain thresholds might be directed to different document libraries, applications, folders, or Web Services, for storage.

Replication using Microsoft Technologies

Assuming a SharePoint™ environment for the initial storage of documents and the configuration of suitable workflows, it should be possible to instigate workflows based on document properties and thereby replicate the functionality described above.

5.9 High-performance scale-up and scale-out capability

ActiveDocs Opus Composition Server is the server-side part of ActiveDocs Opus that encompasses all of the server-side technology that supports the ActiveDocs Opus Express Wizard, Document Wizard, Document Assembly, Document Delivery, and ultimately the use of the ActiveDocs Opus Designer and Client functions from the desktop.

Composition Server can be scaled-up for multi-threading on multi-CPU servers, and/or can be scaled out for multiple server support. In the latter case, the ActiveDocs Opus servers can be configured to work in parallel off the same document creation job queue and to store the session state on a database server and/or can be managed by a network sharing appliance.

ActiveDocs Opus uses a SQL Server back-end and the configuration of this aspect will have an impact on the performance of the system as a whole.

Replication using Microsoft Technologies

It has to be assumed that the SharePoint server-based environment posited as the likely backbone of a solution based on Microsoft Technology would be scalable in the same ways as the ActiveDocs Opus solution.

5.10 Managed document delivery through multiple channels

ActiveDocs Opus Composition Server has a built-in delivery module designed to integrate with print, email, and fax infrastructure for document delivery. For each delivery type, the delivery module supports a collection of multiple named queues which can be configured to run at different times, days of week, and days of the month.

The print integration supports OCR technology and job-sorting via document properties so that, for instance, all of a customer's documents can be grouped together with start-and-end OCR marks to cause all of the documents to be "stuffed" into the same envelope.

Replication using Microsoft Technologies

Any functionality that is not natively supported by the proposed collection of Microsoft Technologies could be custom-developed.

5.11 A well-defined programmatic interface for document generation

The ActiveDocs Opus Solutions Studio and Automated Document Production modules offer a well-defined programmatic interface for document generation, allowing integration with other applications for Automated Mode document generation.

Automated Mode document creation uses a Web Service and Job XML (XML command stream) to specify the Template(s) and Answers to be used. The Templates used for this mode of document creation have no special features that set them apart from Templates used for interactive (User Driven) document creation and indeed the same Templates can be used in both modes.

If a Template uses Data Views to select its data from external data sources, then the data can be automatically populated and only minimal 'seed' data (e.g. a Customer ID) needs to be supplied via Job XML. This is a significant feature that reduces the ongoing development and maintenance workload for the application that is using ActiveDocs Opus.

Replication using Microsoft Technologies

There is a clear disconnection between the use of the technology in User-Driven Mode and the use of the technology in Automated Mode.

In User-Driven Mode, the preference noted above is for a solution based on InfoPath forms to ensure a well-structured interview process that can integrate data from external sources.

In Automated Mode, the preference would be to use the Open XML SDK and Word Automation Services as discussed under "Server-side Document Assembly", above.

To be able to use the same "templates" for both Modes, the User-Driven Mode's InfoPath solution would need to be supported or 'completed' by a server-side assembly process which could be re-used for Automated Mode. The real challenge, however, is keeping the document's data requirements, and the InfoPath forms' functionality, aligned.

This last point is no issue for ActiveDocs Opus, as the data requirements and the interactive use of the Template are fully integrated through a common set of fields and groups, and the Templates are able to be used in both modes transparently.

5.12 Integrated support for multiple workflows

ActiveDocs Opus supports a set of workflows for Document Control, Approval, and Finalisation, which may be applied at almost any level of granularity, from global to per-document.

Document workflows may incorporate specifications for Approval and/or Finalization. Approval is off or on, and if on, can be instigated manually or automatically, with predetermined or selectable single or multiple Approvers. Finalization may specify Draft and Final document properties and/or watermarks, the formats in which the draft and final documents are to be created, and where the final document is to be stored.

Workflows may be preset and may also be adjusted on a per-document basis so, for example, the fact of a Sales Proposal's total cost being over or under a particular threshold may cause the document's workflow to be altered (e.g. Approval may only be required if the value exceeds \$1m).

Replication using Microsoft Technologies

Assuming (1) a SharePoint environment for the initial storage of documents and the configuration of suitable workflows, and (2) easily repeatable document generation processing (whatever combination of Word, InfoPath, Open XML SNK and Word Automation Services that comprises the remainder of the solution) it should be possible to instigate workflows based on document properties and thereby replicate to some degree the functionality described above.

6 Conclusion

ActiveDocs Opus is an integrated solution designed for business users to meet the challenges of enterprise Document Automation. It has evolved consistently to meet the changing requirements of Document Automation.

A number of Microsoft technology products have potential for use in Document Automation. Individually, none can provide an integrated, secure, controlled, enterprise-grade solution.

Any solution based on Microsoft Technologies therefore requires integration of selected technologies. Significant development time and cost would be required to integrate the necessary features to create a workable solution.

Even with Microsoft Technologies working together there are significant challenges such as the alignment of template design with the creation and maintenance of the user interface. The ability of business units, rather than IT, to maintain templates under such a solution is questionable. If the business cannot create and manage its own templates then the utility of the solution is compromised, as ownership is diluted and responsiveness is lost to long IT maintenance cycles.

Turning the vision into reality means creating one of a class of what has been described as “cumbersome, expensive, one-off solutions” to which the usual Build-versus-Buy questions and arguments apply:

- Can your organisation predict how its document automation needs will change during the next few years?
- Does your organisation have the in-house development and project management expertise required, or the resources pay an external organisation?
- Can your organisation tolerate the long development cycle of a custom build?
- Does your organisation want to support, maintain and enhance the application as its needs and external factors change, or pay an external organisation to do so?
- Does your organisation understand the risks of one-off development v purpose-built commercial software?
 - Unknown costs v known costs
 - Estimated delivery time v guaranteed delivery
 - Limited functionality v extensive functionality
 - Lack of ongoing product development and support v ongoing development and support as operating systems and other applications evolve
 - Potential loss of expertise (whether in-house or external) v an experienced team in a company dedicated to a product
 - The expense of testing one-off applications v dedicated quality assurance and a product refined from feedback from a worldwide client base
 - Unknown performance v proven performance and scalable architecture
 - Your organisation's resources directed from away from other business areas as the product is developed v dedicated development and testing teams

It is worth reiterating the points made at the start of this paper:

- ActiveDocs Opus is made possible by a large number of enabling technologies found in Microsoft Office and Server products.
- Creating an enterprise-class Document Automation solution is only possible with significant development resource.
- Choosing ActiveDocs Opus over using Microsoft Technologies comes directly down to a Buy-versus-Build decision.
- To build such a solution should only be attempted by organisations with significant Microsoft Office Development expertise and a project timeline and budget which can accommodate the Development effort required.