ActiveDocs

DOCUMENT AUTOMATION SOFTWARE

USING ACTIVEDOCS IN AUTOMATED MODE

Prepared by:	Nick Chivers
	Director of Product Marketing
Audience:	ActiveDocs Evaluator

Abstract: This document provides an overview of the use of ActiveDocs by other applications to create documents in Unattended or Automated Mode.

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

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 © ActiveDocs^{TM} 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 Introduction

ActiveDocs is the third generation of ActiveDocs' document automation products, and the second generation of ActiveDocs' server-based products.
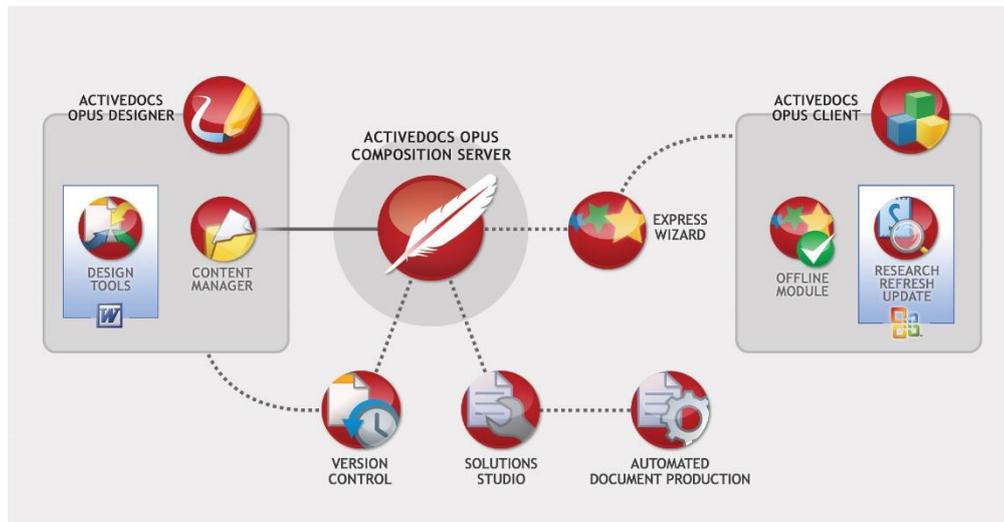

FIGURE 1 THE ACTIVEDOCS SUITE

ActiveDocs creates documents by inserting data into specified markers in templates. This simple concept is enabled through two core technologies. The first is ActiveDocs Designer which provides a rich environment for the design and management of Templates and other Design Components. The second is ActiveDocs Composition Server which provides a powerful XML-based document assembly engine, the Document Compiler, designed to merge data with Templates to create documents.

ActiveDocs provides two methods for supplying data to the Document Compiler. The first method is the web-based ActiveDocs Document Wizard which provides an interactive user interface to elicit from an end user (Document Creator) the data required for any ActiveDocs Template. This is called User-Driven Mode. The second method permits the unattended supply of data in an XML stream called Job XML from another application via the ActiveDocs Solutions Studio integration module. This is called Automated Mode.

This document outlines the use of ActiveDocs to create documents using Automated Mode.

2   Templates

The use of ActiveDocs Designer to create and manage Templates and other Design Components is well-documented elsewhere. For the purposes of this document it is necessary only to consider the end product of the Template design process.

An ActiveDocs Template consists of fixed content and variable content. The variable content is a collection of ActiveDocs markers representing Design Items like Snippet Links, Selection Lists, Repeating Items, and Active Fields.

Dear EMPLOYEE_FirstName,

We are delighted to offer you the full time position of EMPLOYMENT_Position_Title with Dalmore Corporation, for our EMPLOYMENT_Employment_Type contract.

As discussed at your recent interview with us, you will report to EMPLOYMENT_Manager, EMPLOYMENT_Manager_Title.

The terms and conditions of our full time offer of employment are set out below. We request that you confirm your acceptance of our offer of employment by signing this letter and returning it to us within five (5) working days.

Commencement   You will commence employment on EMPLOYMENT_Start_Date on a probationary period for three (3) months. The probationary period will enable both you and Dalmore Corporation to determine if the position is suitable to your skills, qualifications and capacity. During this period of probation, if your employment is terminated, the notice period is one week.

FIGURE 2 AN EMPLOYMENT CONTRACT TEMPLATE

Active Fields, shown as shaded markers above, are essentially placeholders for data that is required to create the finished document. Whether documents are created in interactive User-Driven Mode or in unattended Automated Mode, data needs to be supplied for the Active Fields.

In interactive User-Driven Mode, data is acquired from the Document Creator (or derived from other data) via the ActiveDocs Document Wizard.

In unattended Automated Mode, data is supplied in an XML stream called Job XML which is passed to ActiveDocs Composition Server's Document Compiler via a web services interface.

Any ActiveDocs Template developed for interactive use in User-Driven Mode can also be used in Automated Mode.
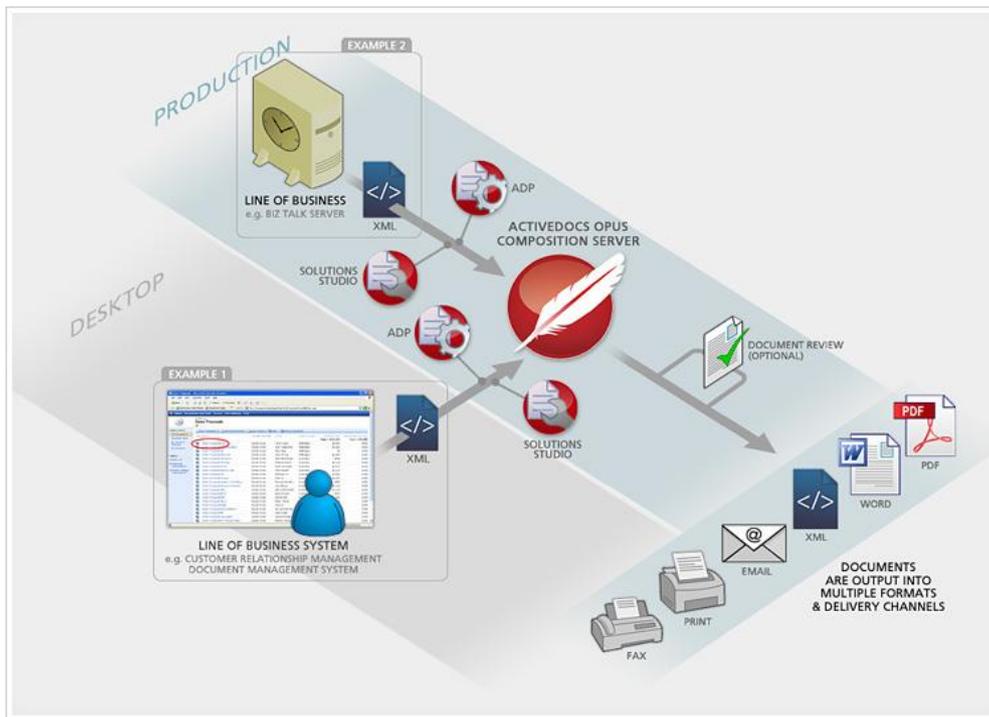
3   Automated Mode

FIGURE 3 ACTIVEDOCS AUTOMATION EXAMPLES

There are two scenarios in which Automated Mode is used. The first is user-initiated (Example 1, above) and the second is fully automatic (Example 2, above). Both use the same principle of supplying data via Job XML to the Document Compiler.

3.1   User-initiated Automated Mode

Consider a common scenario in which a Customer Relationship Management (CRM) application is being used at a call centre. A customer contacts the call centre and informs the call handler of a change of address. As part of the change of address procedure, the call handler is required to produce a short letter to the customer thanking the customer for the information and confirming the new address details.

Without any document creation system, the call handler is obliged to manually create such a document. With a document creation system that supports only the interactive use of templates, the call handler would invoke a suitable template (the Change Of Address Acknowledgement Letter) and would enter or retrieve the required data (Customer Name, Account Number, Old Address, New Address, and so on).

However, we can see that the CRM actually has all of the data required to complete the document. There is no need for the call handler to be involved in creating the document other than to initiate its creation; the CRM can provide all of the variable information itself. Why not simply provide the call handler with a "Change Of Address Acknowledgement Letter" button to click, so the call handler can close the current transaction and move on to handling the next call? ActiveDocs supports this drive for efficiency with user-initiated Automated Mode.

The implementation is straightforward: clicking the button initiates a software process in which the required information is marshalled and transformed into ActiveDocs Job XML, which also specifies the Template to be used; ActiveDocs Composition Server's web service is contacted and the Job XML is sent to it; the document is assembled. Delivery can be automated through ActiveDocs Composition Server's Document Delivery service via print, fax, or email.

The user-initiated scenario described above can be further automated by dispensing with the button. The CRM application already knows that a change of address event has taken place and can produce the letter as a fully automatic response; this is a small scale example of the fully automatic use of the Automated Mode described below.

3.2   Fully automatic Automated Mode

Large scale fully automatic use of Automated Mode is fully supported by ActiveDocs. Consider another common scenario in which an incumbent application is generating customers' invoices or monthly statements.

Monthly statement runs, invoicing, and other batch-type processes, have common characteristics. They produce large numbers of documents, and they tend to be hard to change. Reformatting the output in response to external changes – rebranding, legislation, sales and marketing initiatives – requires the use of extensive and expensive IT resources. Such work naturally takes its place in a queue of other IT tasks, so any response to external changes becomes expensive and rapid response becomes impossible.

ActiveDocs addresses the "change problem", firstly by separating the data processing (which seldom changes) and the output formatting (which frequently changes). This can usually be achieved with a one-time change to stop the incumbent application from generating its own output documents, and instead to send its data as Job XML to the ActiveDocs Composition Server.

Using ActiveDocs Composition Server to create the output documents means that ActiveDocs Templates can be used, and this introduces the second and most important way to use ActiveDocs to address the "change problem": ActiveDocs Templates are designed and managed by the business. This puts the business in control of the output content, while simultaneously reducing the reliance on IT and freeing IT resources for other tasks.

With the output content under the business' control, the "change problem" simply becomes "change management". The use of ActiveDocs Templates means that presentation can be improved, rapid response becomes possible as changes take hours rather than months, and output documents can be dynamically customised.

Dynamic customisation means that output content can be adjusted according to the data being processed. Does the customer require statements in another language? Do you want to include optional additional documents for the customer, based on what the customer bought last month? Add a thank-you note for large orders? Include warnings for overdue payments, or propose repayment schedules? ActiveDocs allows the business to design and deploy Templates to do all of this dynamic, optional, personalised document creation and much more.

ActiveDocs addresses the "change problem" without losing sight of the other common requirement of large-scale document production: speed. ActiveDocs Composition Server's Document Compiler is benchmarked at more than 150,000 documents per hour, meaning that performance does not have to be sacrificed to presentation and adaptability.

4 Integration Development

Integrating other applications with ActiveDocs to use Templates in Automated Mode requires the ActiveDocs Solutions Studio and ADP (Automated Document Production) modules to be licensed and installed with the ActiveDocs Composition Server. With these in place, integration development can proceed.

4.1 Templates and Answer Data

The Template is developed using ActiveDocs Designer. The use of ActiveDocs Designer to create and manage Templates and other Design Components is well documented elsewhere.

With the Template deployed on the ActiveDocs Composition Server, the next step is to discover what the Template expects as data ("Answers" in ActiveDocs terminology) to populate its Active Fields. By far the easiest way to do this is to run the Template in User-Driven Mode through the ActiveDocs Document Wizard via the ActiveDocs Express Wizard browser interface.
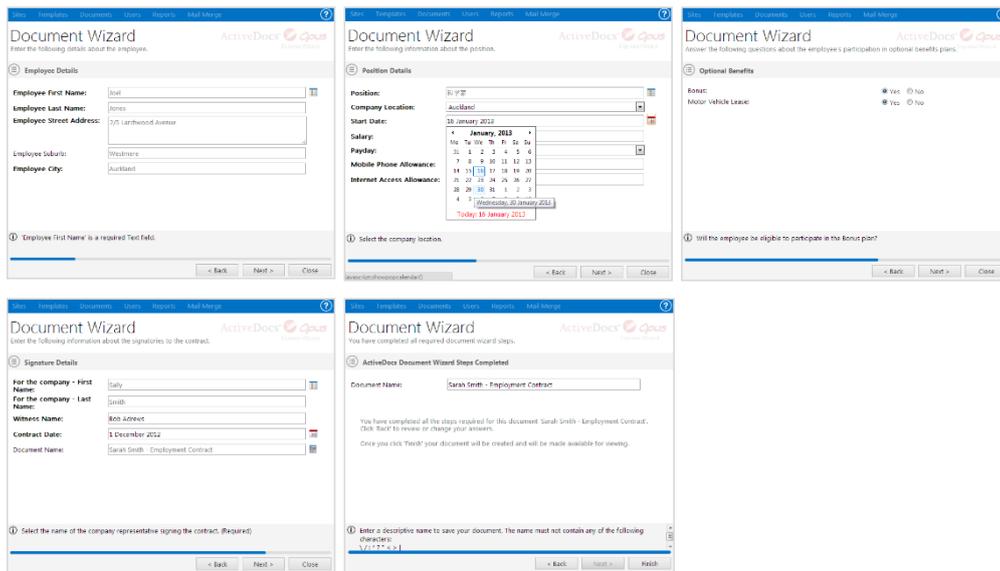

FIGURE 4 THE EMPLOYMENT CONTRACT TEMPLATE IN THE ACTIVEDOCS DOCUMENT WIZARD

When the Template has been used in this way, the completed document and the Answer Data are stored by ActiveDocs Composition Server.
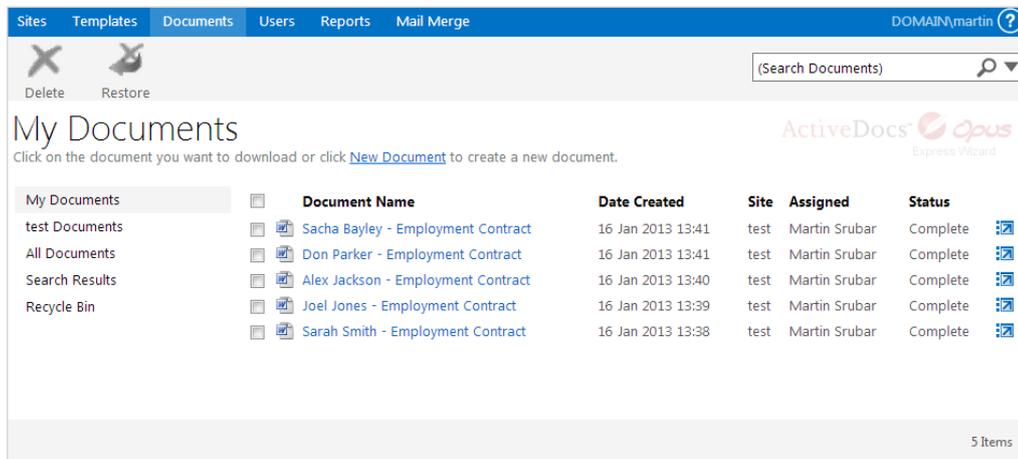

FIGURE 5 DOCUMENTS PAGE OF THE ACTIVEDOCS EXPRESS WIZARD

Importantly, the Answer Data is stored in XML format in a schema which is fully compatible with ActiveDocs Job XML, and the Answer file can be downloaded for examination from the ActiveDocs Express Wizard's Document Actions page.
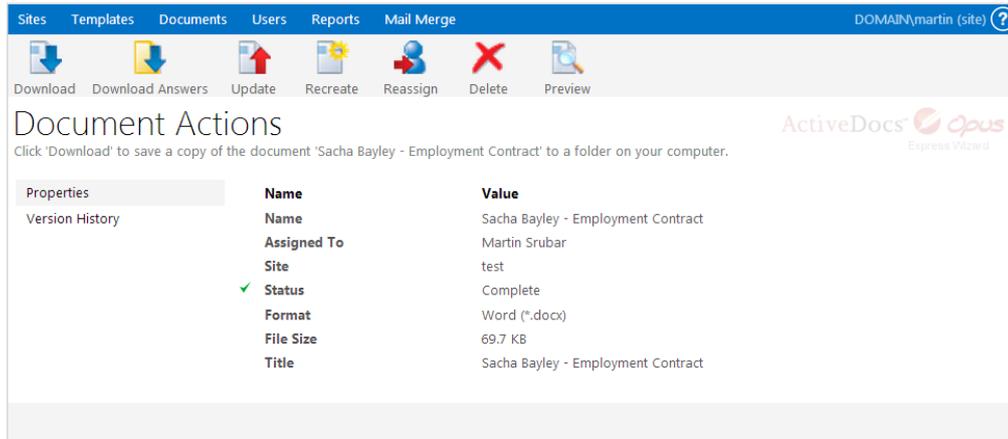

FIGURE 6 DOCUMENT ACTIONS PAGE OF THE ACTIVEDOCS EXPRESS WIZARD

The downloaded Answer data file is in XML format and can be opened with any text editor or an XML-compatible program like Microsoft® FrontPage. The Answer data shown here has been formatted with tabbing for clarity.
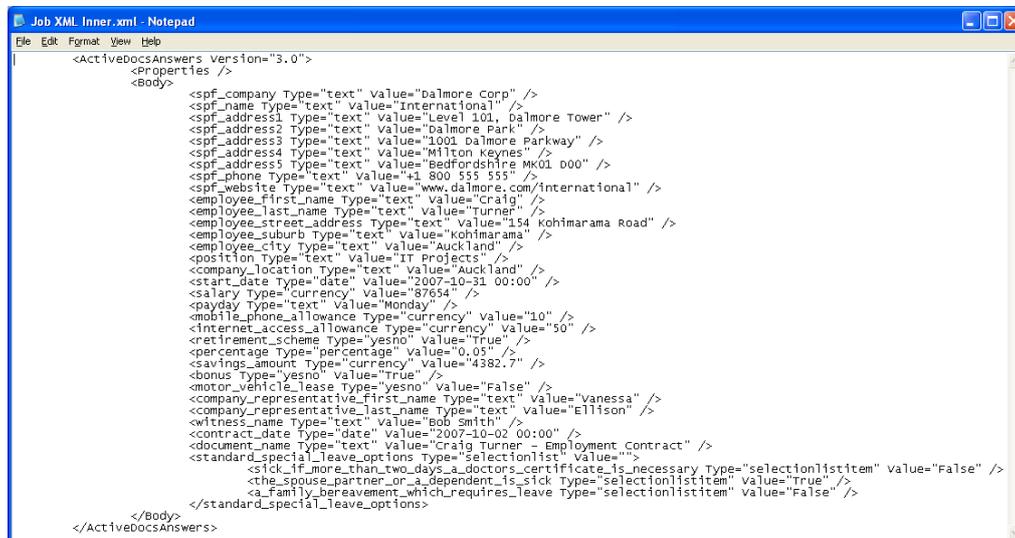

FIGURE 7 SAMPLE ANSWER DATA FOR AN EMPLOYMENT CONTRACT

In the example shown, each line in the ActiveDocs Answers node refers to a particular Active Field and specifies its name, type, and value. The last five lines in the Body area refer to a Selection List.

4.2 Job XML

Answer data forms part of the Job XML stream, usually as part of a Job Item node which pertains to one output document. The full specification for Job XML is included in the ActiveDocs Solutions Studio documentation.

There are endless variations of Job XML. An example of the Job XML before insertion of the Answer data might look like this:

```
Job XML Outer.xml - Notepad
File  Edit  Format  View  Help
<Job Subsite="Opus VM1.1" Format="1"  xmlns="urn:ActiveDocs.Enterprise.Jobv2">

  <TemplateSet>
    <Template>Employment_Contract_ESLO</Template>
  </TemplateSet>

  <JobItem>

  </JobItem>
</Job>
```
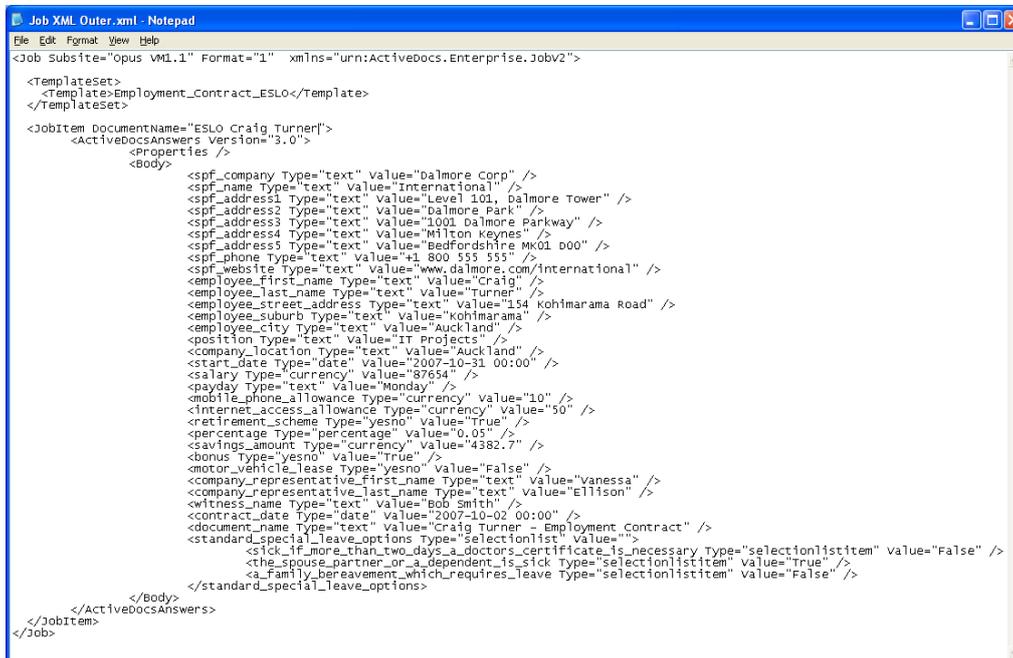
FIGURE 8 AN EXAMPLE OF JOB XML BEFORE ANSWER DATA IS ADDED

"Job" is the top-level node. It specifies the ActiveDocs Composition Server subsite (refer to ActiveDocs Composition Server documentation), the Format of the output document, and the XML namespace. In this example, the document format will be Word ML (value "1"). Other format options are described in the ActiveDocs Solutions Studio documentation. This specification may be given or over-ridden on a per-document basis in the Job Item node.

The Template Set node describes the Template or Templates to be used for all documents in this example. This specification may also be given or over-ridden on a per-document basis in the Job Item node.

Delivery requirements may also be specified, once again at the Job level or at the Job Item level. Delivery methods include print, email, and fax; the specification corresponds to Delivery Queues set up in the ActiveDocs Composition Server's Document Delivery module. For email delivery, specific email addresses are usually specified in the Job XML.

In the expanded example shown below, the Answer data has been added to the Job Item node, and a document name has been specified.

```
Job XML Outer.xml - Notepad
File  Edit  Format  View  Help
<Job Subsite="Opus VM1.1" Format="1"  xmlns="urn:ActiveDocs.Enterprise.Jobv2">

  <TemplateSet>
    <Template>Employment_Contract_ESLO</Template>
  </TemplateSet>

  <JobItem DocumentName="ESLO Craig Turner">
      <ActiveDocsAnswers Version="3.0">
          <Properties />
          <Body>
              <spf_company Type="text" Value="Dalmore Corp" />
              <spf_name Type="text" Value="International" />
              <spf_address1 Type="text" Value="Level 101, Dalmore Tower" />
              <spf_address2 Type="text" Value="Dalmore Park" />
              <spf_address3 Type="text" Value="1001 Dalmore Parkway" />
              <spf_address4 Type="text" Value="Milton Keynes" />
              <spf_address5 Type="text" Value="Bedfordshire MK01 D00" />
              <spf_phone Type="text" Value="+1 800 555 555" />
              <spf_website Type="text" Value="www.dalmore.com/international" />
              <employee_first_name Type="text" Value="Craig" />
              <employee_last_name Type="text" Value="Turner" />
              <employee_street_address Type="text" Value="154 Kohimarama Road" />
              <employee_suburb Type="text" Value="Kohimarama" />
              <employee_city Type="text" Value="Auckland" />
              <position Type="text" Value="IT Projects" />
              <company_location Type="text" Value="Auckland" />
              <start_date Type="date" Value="2007-10-31 00:00" />
              <salary Type="currency" Value="87654" />
              <payday Type="text" Value="Monday" />
              <mobile_phone_allowance Type="currency" Value="10" />
              <internet_access_allowance Type="currency" Value="50" />
              <retirement_scheme Type="yesno" Value="True" />
              <percentage Type="percentage" Value="0.05" />
              <savings_amount Type="currency" Value="4382.7" />
              <bonus Type="yesno" Value="True" />
              <motor_vehicle_lease Type="yesno" Value="False" />
              <company_representative_first_name Type="text" Value="Vanessa" />
              <company_representative_last_name Type="text" Value="Ellison" />
              <witness_name Type="text" Value="Bob Smith" />
              <contract_date Type="date" Value="2007-10-02 00:00" />
              <document_name Type="text" Value="Craig Turner - Employment Contract" />
              <standard_special_leave_options Type="selectionlist" Value="">
                  <sick_if_more_than_two_days_a_doctors_certificate_is_necessary Type="selectionlistitem" Value="False" />
                  <the_spouse_partner_or_a_dependent_is_sick Type="selectionlistitem" Value="True" />
                  <a_family_bereavement_which_requires_leave Type="selectionlistitem" Value="False" />
              </standard_special_leave_options>
          </Body>
      </ActiveDocsAnswers>
  </JobItem>
</Job>
```

FIGURE 9 AN EXAMPLE OF JOB XML WITH ANSWER DATA AND DOCUMENT NAME ADDED

4.3   Self Populating Templates

ActiveDocs Opus Viper introduces this new feature which significantly reduces the amount of Answer data required in the Job XML by enabling automatic retrieval of data from external data sources. For example, a Template might contain a number of Active Fields for customer data like name and address which can all be obtained from a customer database. Provided that the Active Fields are all linked to the data source via an ActiveDocs Data View, the Job XML can be constructed with just a "seed" field, such as the Customer ID, populated.

The advantages of this approach are twofold. Firstly, the Job XML is simpler and smaller. Secondly, and more significantly, changes to the customer fields required by the Template –

e.g. the addition of a Country name – do not require the Job XML to change, so the application which generates it does not need to be updated, and the maintenance effort is thereby reduced.

4.4 Application Integration

To provide user-initiated or fully automatic use of the ActiveDocs Automated Mode, the other application (examples given above: CRM, invoice/statement generation) needs to be modified to use ActiveDocs Composition Server's web service.

Fully documented code examples showing how to specify, open, and use a connection to the web service are included in the ActiveDocs Solutions Studio documentation.

The application needs to generate suitable Job XML, which means that it must match the required format and at least specify a valid subsite, Template(s), and Active Field names. A Job XML stream may contain the details for one document or many documents, using one Template or many Templates.

Finally, the Job XML is submitted to the ActiveDocs Composition Server's web service. Fully documented code examples showing how to submit the Job XML, monitor the progress of the document assembly process, and retrieve the document bytes (if required), are included in the ActiveDocs Solutions Studio documentation.

5   Workflow for Approval and Finalisation

5.1   Document Approval

Document Approval is a switchable feature that allows a nominated group of users, called Document Approvers, to be alerted by email when a document is created within their jurisdiction. Individual Approvers then log on to the system and give their approval, or disapproval, to the document. The original creator of the document can be automatically emailed about these responses.

The feature can be further configured so that approval must be given by all Approvers, or by just one Approver, before the document is considered approved. Documents which are pending approval, or for which approval is not granted, cannot be further processed in the system.

ActiveDocs server-based document automation solutions have always supported post-creation Approval processing of documents, whether generated in Automated Mode or User-Driven Mode. ActiveDocs Opus Viper significantly enhances the Document Approval feature providing options for:

- Nominating sub groups of Approvers
- Allowing the document creator to control the email notification to Approvers
- Allowing the document creator to choose the Approver
- Identifying a default Approver

A separate whitepaper, available on request, provides extended information about Document Approval and its optional integration with Document Finalisation (see below) in ActiveDocs Opus Viper.

5.2   Document Finalisation

Document Finalisation is an ActiveDocs Opus Viper feature which optionally integrates with Document Approval (see above) and works with document creation in both Automated Mode and User-Driven Mode.

Briefly, Document Finalisation allows documents to be automatically or manually finalised with a variety of options:

- Document Save locations
- Document Save formats
- Reassignment
- Notification

Finalisation is integrated with new Draft options which allow selection of the Draft format, document versioning for recreation, selection and specification of a document property for the draft status, and Draft/Final watermarking.

A separate whitepaper, available on request, provides extended information about Document Finalisation and its optional integration with Document Approval in ActiveDocs Opus Viper.

6   Conclusion

ActiveDocs Automated Mode is designed for seamless integration, enabling other applications to leverage the use of ActiveDocs Templates for large-scale and small-scale document creation.

The use of ActiveDocs Templates allows the organisation to take full advantage of features like enterprise-wide standardised formatting and content, combined with dynamic customisation and personalisation of documents.

Using ActiveDocs Automated Mode for document creation reduces the reliance on IT resources by allowing the business to use ActiveDocs Templates which the business owns, designs, and manages for itself.

ActiveDocs, used in Automated Mode or User-Driven Mode or a combination of both, provides the ideal enterprise-wide platform for document creation.