

**Oracle® Insurance Policy
Administration**

**Oracle Insurance Integration
Accelerator for Oracle
Insurance Policy
Administration (OIPA) and
Oracle Insurance Documaker**

How-To Guide

Version 1.0

Documentation Part Number: E23637_01

October 2011

Copyright © 2010, 2011 Oracle and/or its affiliates. All rights reserved.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this software or related documentation is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS

Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle USA, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

This software is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications which may create a risk of personal injury. If you use this software in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure the safe use of this software. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software in dangerous applications.

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

This software and documentation may provide access to or information on content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.

Where an Oracle offering includes third party content or software, we may be required to include related notices. For information on third party notices and the software and related documentation in connection with which they need to be included, please contact the attorney from the Development and Strategic Initiatives Legal Group that supports the development team for the Oracle offering. Contact information can be found on the Attorney Contact Chart.

The information contained in this document is for informational sharing purposes only and should be considered in your capacity as a customer advisory board member or pursuant to your beta trial agreement only. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described in this document remains at the sole discretion of Oracle.

This document in any form, software or printed matter, contains proprietary information that is the exclusive property of Oracle. Your access to and use of this confidential material is subject to the terms and conditions of your Oracle Software License and Service Agreement, which has been executed and with which you agree to comply. This document and information contained herein may not be disclosed, copied, reproduced, or distributed to anyone outside Oracle without prior written consent of Oracle. This document is not part of your license agreement nor can it be incorporated into any contractual agreement with Oracle or its subsidiaries or affiliates.

This documentation is in prerelease status and is intended for demonstration and preliminary use only. It may not be specific to the hardware on which you are using the software. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to this documentation and will not be responsible for any loss, costs, or damages incurred due to the use of this documentation.

CONTENTS

GETTING STARTED	7
Prerequisites for This Document	7
Documentation Audience	7
<i>About this Document</i>	8
<i>Integration Steps</i>	9
STEP 1. REQUIREMENTS STUDY	10
<i>Pre-Integration Planning and Considerations</i>	11
OIPA Considerations	11
Documaker Considerations	11
Integration Considerations	11
<i>Analysis of Model Documents</i>	12
Related Documentation	12
<i>Integration Checklist</i>	13
STEP 2. PROJECT PLANNING.....	17
<i>Resource Requirements</i>	18
<i>System Requirements</i>	19
<i>Oracle Education</i>	20
Oracle Insurance Policy Administration	21
Oracle Documaker.....	22
<i>Oracle Support</i>	23
STEP 3. IMPLEMENTATION.....	25
<i>Configuring OIPA</i>	26
Step 1. Data Mapping and Identification of Data Gaps	26
Step 2. Add Data to Transaction Results	28
Step 3. Add comments to delineate math variables	28
Step 4. Add the required MathVariables	28
Step 5. Run Transaction and Test Required Data Generated	29
Step 6. Add Web Service Information.....	29
Step 7. Construct Template for Data Transformation.....	30
Step 8. Update Services File for New Service	32
<i>Configuring Documaker</i>	33
Structured Documents	33
Related Documentation	34
On Demand Documents	35
Best Practice	36
Related Documentation	36
<i>Configuring Transport</i>	37
File Transport Configuration.....	37
Service Registry Definition	37
Recommended Filesystem Storage	37
Batch Execution Script	38
Real-time Transport Configuration.....	38
Service Registry Definition	38
Update the Template for EWPS.....	38
STEP 4. TEST AND DEPLOYMENT.....	41
<i>OIPA Deployment</i>	42
<i>Documaker Deployment</i>	43
Deployment Steps	43
APPENDIX A: REAL-TIME SAMPLE	44

APPENDIX B: BATCH SAMPLE	45
GLOSSARY OF TERMS.....	46
APPENDIX C: PROCEDURES IN OIPA INSTALLATION TO RUN EXAMPLE	47
OIPA INSTALLATION STEPS	47
Refer to Update the Template for EWPS section of this guide.....	48
APPENDIX D: PROCEDURES IN RUNNING DOCUMAKER ENVIRONMENT	49
DEPLOYMENT STEPS INSTALLED DIRECTORY STRUCTURES	49
ALAMERE STUDIO INSTALLATION STEPS	51
<i>Before you begin</i>	<i>51</i>
<i>Verify Documaker Batch Installation.....</i>	<i>51</i>
<i>Verify Web Processing Installation.....</i>	<i>51</i>

Figures

Figure 1: Integration Steps	Error! Bookmark not defined.
Figure 2: Requirement Study	10
Figure 3: OIPA and Documaker Integration Process.....	13
Figure 4: OIPA Workflow	13
Figure 5: Documaker Structured Document Workflow.....	14
Figure 6: Documaker On Demand Document Workflow.....	15
Figure 7: Project Planning	17
Figure 8: Implementation.....	25
Figure 9: Math Variable Snippet.....	28
Figure 10: Math Variable with comments Snippet	28
Figure 11: Web Service Information Snippet	29
Figure 12: Template Ready for Transformation Snippet.....	31
Figure 13: Service-Registry XML Snippet	32
Figure 14: Oracle Documaker Architecture.....	33
Figure 15: Overview	35
Figure 16: Integration Configuration Workflow.....	39
Figure 17: Deployment	41
Figure 18: On Demand Sample.....	44
Figure 19: Structured Sample	45
Figure 20: Folder Structure	47

Tables

Table 1: Integration Checklist..... 16

Table 2: Oracle Insurance Policy Administration Version and Documentation 19

Table 3: Documaker Products and Documentation 19

Table 4: Product Name and Number 23

Table 5: Requesting a CSI 23

Table 6: Registration on Oracle Support Site..... 24

Table 7: Completed Integration Checklist 40

Table 8: Alamere Folders and Files 49

Getting Started

Welcome to the Oracle Insurance Policy Administration (OIPA) and Oracle Documaker How to Guide for integrating OIPA and Documaker. The integration enables OIPA to automatically send the data using a seamless data transfer technique to Documaker for generating correspondence and policy pages using the OIPA Term Life chassis. The integration provides a flexible toolkit that allows you to span other business services such as billing and claims processes for the OIPA Term Life chassis.

This guide describes step-by-step how to use the integration tools and the configuration files that should be updated for use with integration. Our goal is to provide the necessary information to develop, plan, and execute an OIPA and Documaker integration project.

PREREQUISITES FOR THIS DOCUMENT

Prior to reading this guide, complete the OIPA and Oracle Documaker Questionnaire and complete the OIPA and Documaker Requirement Study documents.

DOCUMENTATION AUDIENCE

This document is intended for the following audiences:

- **Insurance Global Business Unit (IGBU) product(s) trained specialists** — Project team members who will focus on helping customers integrate OIPA and Documaker. Project team members will have an in-depth working knowledge of OIPA, Documaker, XML configuration, business process flow and data mapping.
- **Oracle customers** — Individuals who are responsible for the OIPA and Documaker Integration project for their organization. Customer project team members will have an in-depth working knowledge of their business requirements and have taken or plan to take OIPA training and Oracle Documaker training.

ABOUT THIS DOCUMENT

This document provides an overview and explanation of the specific steps and requirements needed to integrate OIPA and Oracle Documaker for both structured and on-demand documents. This guide is designed to explain how to leverage the OIPA Extensibility Framework and the Outbound Services to build robust communication with the Documaker product.

The guide will begin with a high-level description of how to integrate the two products.

This document will:

- provide an overview of the configuration process
 - ♦ extension setup
 - ♦ transaction XML
 - ♦ transformer
- provide a high-level OIPA and Documaker integration checklist
- provide an OIPA and Documaker step-by-step integration guide
- provide “best practices”
- provide references to related documentation for each topic
- provide a glossary for reference to OIPA and Documaker specific terminology

The OIPA and Documaker integration includes two key categories of documents:

- **Structured documents:** Structured documents typically suggest high volume statements, bills, and policy print (see Scenario 1).
- **On Demand documents:** Real-time generation of documents implies no editing is allowed such as letters that confirm transactions that happened successfully, or letters to confirm a face increase (see Scenario 2).

INTEGRATION STEPS

The steps below illustrate the suggested key components needed to successfully complete an OIPA and Documaker integration implementation. The integration process begins once Step 1 has been completed and continues through Step 4. The complete project timeline will vary based on factors outside the control of the development cycle, such as customer specific requirements, stringency, and length of testing by the customer. Each step is detailed in subsequent pages.

The integration steps can be managed as a series of steps or depending on the project team composition the OIPA and Documaker configuration steps can run concurrently.

Figure 1: Integration Steps

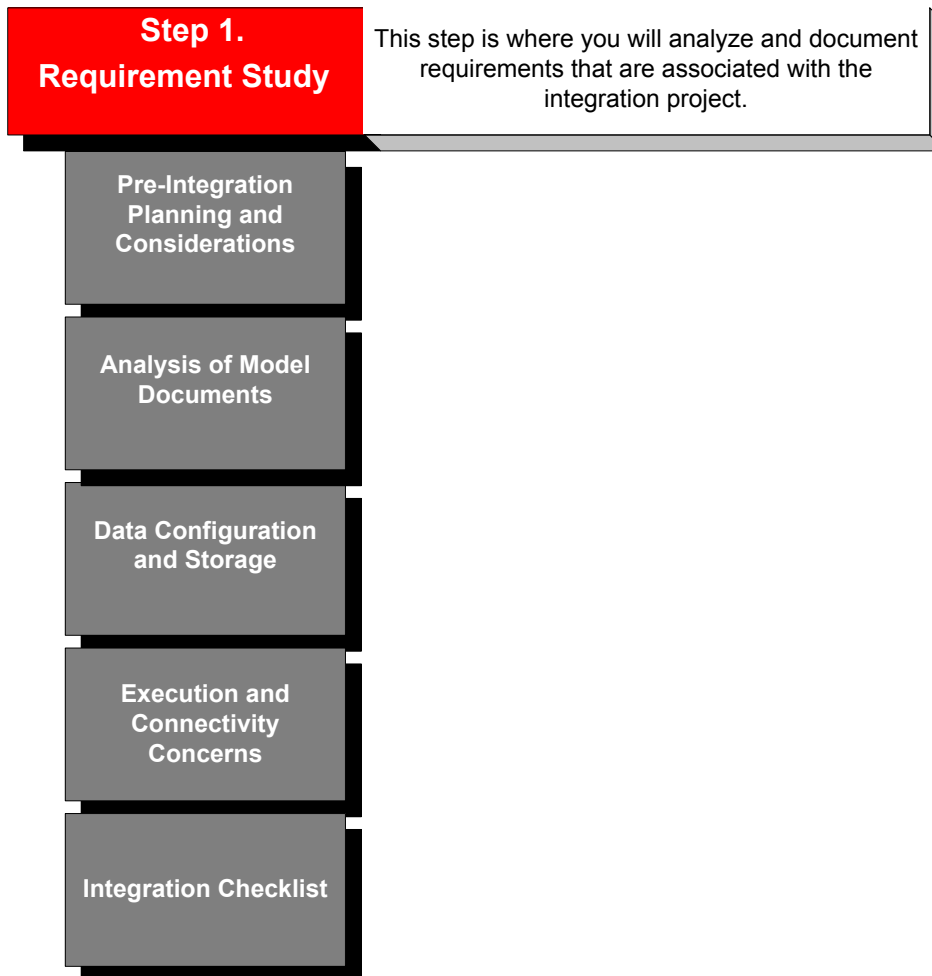
Step 1. Requirements Study	This step is where you will analyze and document requirements that are associated with the integration project.
Step 2. Project Planning	This step identifies: project resources, skills, and system requirements, education needs, and changes to the Oracle support process.
Step 3. Implementation	This step identifies: the OIPA and Documaker implementation steps.
Step 4. Test and Deployment	Final step, execute integration testing and deploy resources.

Step 1. Requirements Study

The requirement study should be completed prior to starting the integration configuration steps and is critical to the success of the project. Each requirement must be measurable, testable, and where possible each requirement should map the OIPA data to Documaker resources. Each requirement must be defined sufficiently so the developer can construct from the requirement study document and testers can define a test matrix.

The requirements study questionnaire should come from the services organization implementing the integration.

Figure 2: Requirement Study



PRE-INTEGRATION PLANNING AND CONSIDERATIONS

OIPA and Oracle Documaker use different file structures, data acquisition models and logic constructs. The integration requires that the content, logic, and variable definitions be provided by OIPA to Documaker in a format that Documaker can consume. A large portion of the integration is handled using tools (transformation, scripts); however, some manual preparation of the environments and post-import tasks are required.

OIPA CONSIDERATIONS

- Identify the business documents to be included with the integration
- Identify the business transactions that provide the content data
- Identify and coordinate variable field definitions in OIPA and Documaker
- Install OIPA
- Install Extension for OIPA

DOCUMAKER CONSIDERATIONS

- Printers required for output documents
- Fonts used within the documents
- True Type Fonts for display
- Printer/device specific fonts
- Install Documaker products

INTEGRATION CONSIDERATIONS

- Production requirements for document transmission (real time vs. batch)
- Each document should have its own message template
- A new Documaker form will be setup for each FLT template
- Determine what content is required in the return message
- Determine storage requirements for documents and archived transactions

ANALYSIS OF MODEL DOCUMENTS

The analysis of business documents and data is critical to the success of an OIPA and Documaker integration project. Document analysis defines form types, recipients, print output types and maps variable data from the OIPA database to DMStudio resources.

Recommended Analysis Questions

- ? What documents will be included in the integration project?
- ? What business activity triggers each document, i.e., billing, collections, policy processing, or claims?
- ? What business data is needed to complete the document?
- ? Does the data need to be translated or formatted?
- ? Who receives a copy of the document (Owner, Beneficiary, Agent, or Agency)?

Pre-built Letters and Term Life Policy Pages

OIPA provides a flexible, rules-based policy administration solution that supports event processing like policy issue, billing, collections, policy processing and claims. For each business process, customer facing documents may be required, such as a policy change confirmation letter.

To help with the integration effort Oracle Insurance provides a pre-built set of Life OIPA data templates and documents that are branded as Alamere Insurance and include:

<u>Correspondance</u>	<u>Transactions or Events</u>
Cancellation/Termination Letter	PolicyCancellationRequest
Policy Change Confirmation Letter	BaseFaceDecrease, and ModeChange
Grace Letter	Grace
Lapse Letter	Lapse
Death Benefit Letter	DeathBenefitLetter
Term Life Policy Pages	Policy Pages (Issue)

RELATED DOCUMENTATION

1. OIPA Rules Palette User Guide
2. Oracle Documaker Studio User Guide
3. Oracle Documaker Studio On-line Help
4. Oracle Documaker FAQ
5. Oracle Documaker Server System Reference Guide
6. Oracle Documaker Batch Processing reference Guide

INTEGRATION CHECKLIST

The first building block to a successful integration is planning. The Insurance Global Business Unit, also known as **IGBU**, has provided an example for the OIPA and Documaker Integration project.

The second building block to a successful integration is a roadmap that defines integration best practices and steps. The IGBU has created a proven integration roadmap (*see checklist on next page*) that will help you manage your integration project. This guide is a blend of the OIPA and Documaker best practices and steps suggested by the IGBU. These integration steps will overlap with the implementation steps used by an implementation team

Figure 3: OIPA and Documaker Integration Process



Figure 4: OIPA Workflow

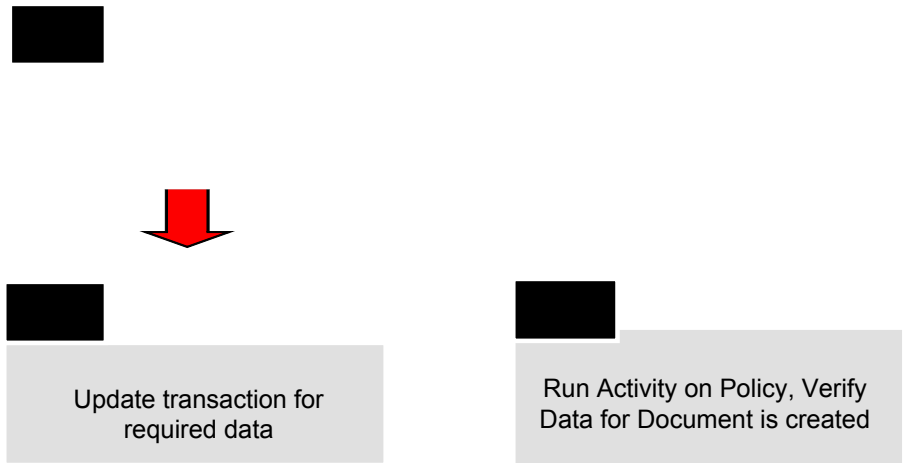


Figure 5: Documaker Structured Document Workflow

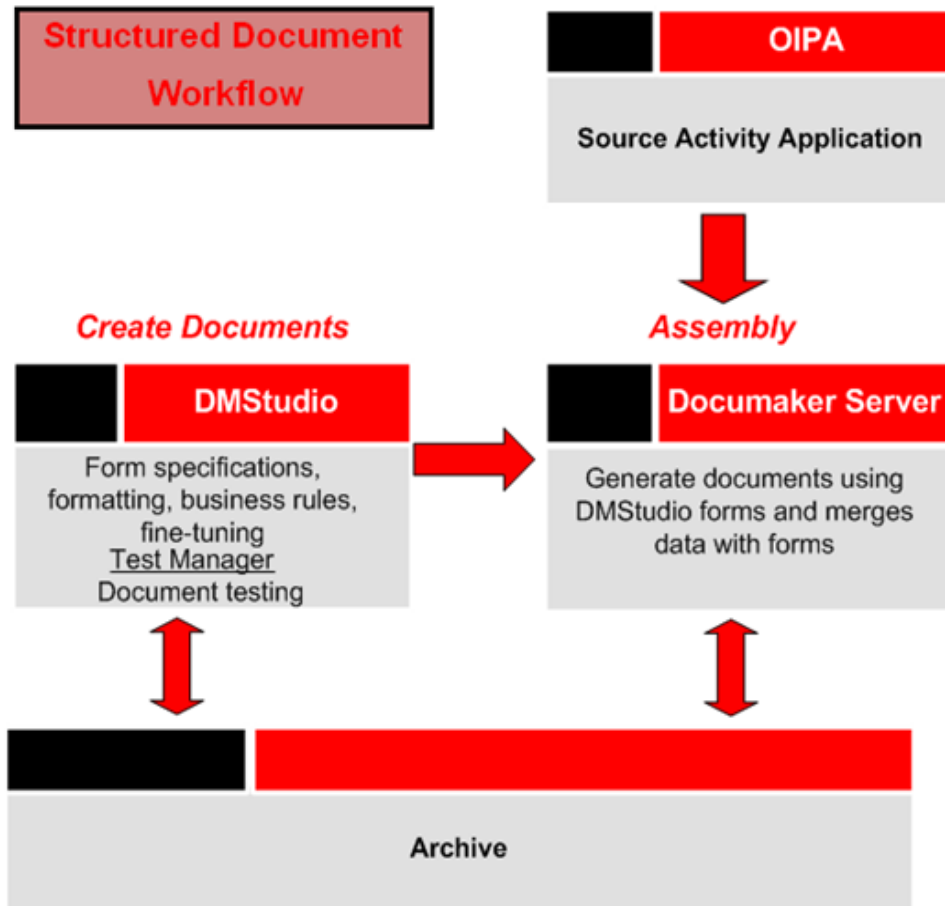


Figure 6: Documaker On Demand Document Workflow

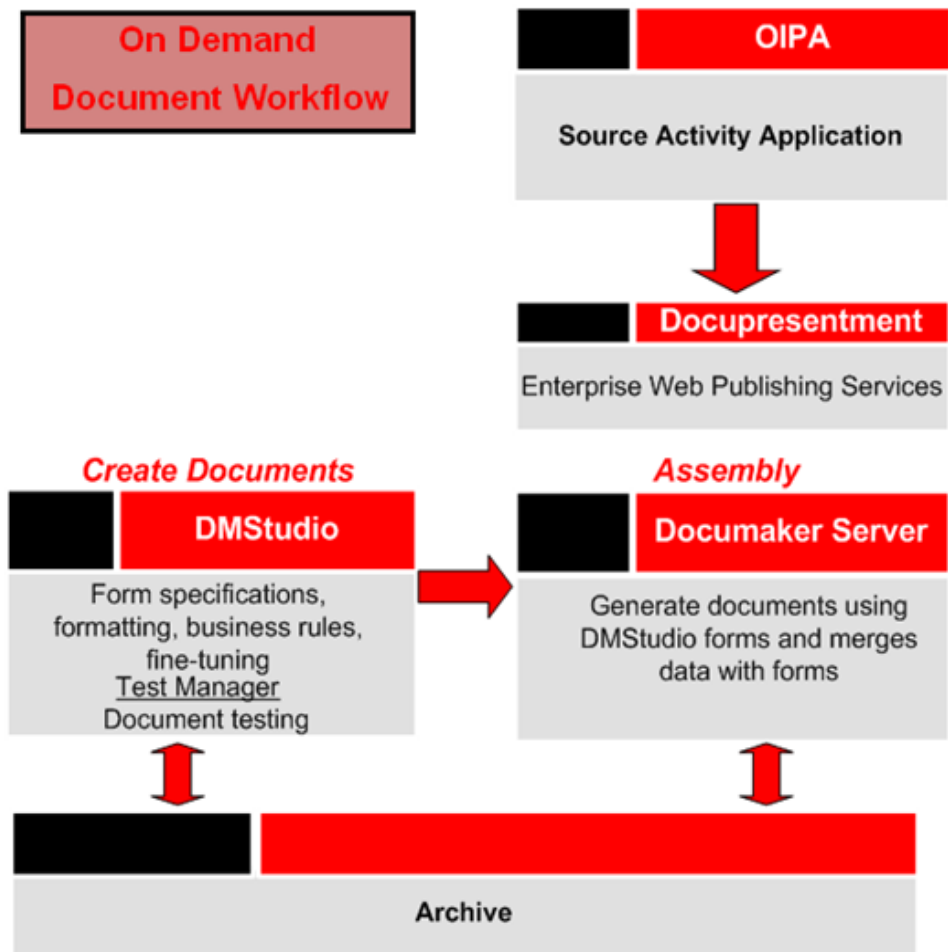


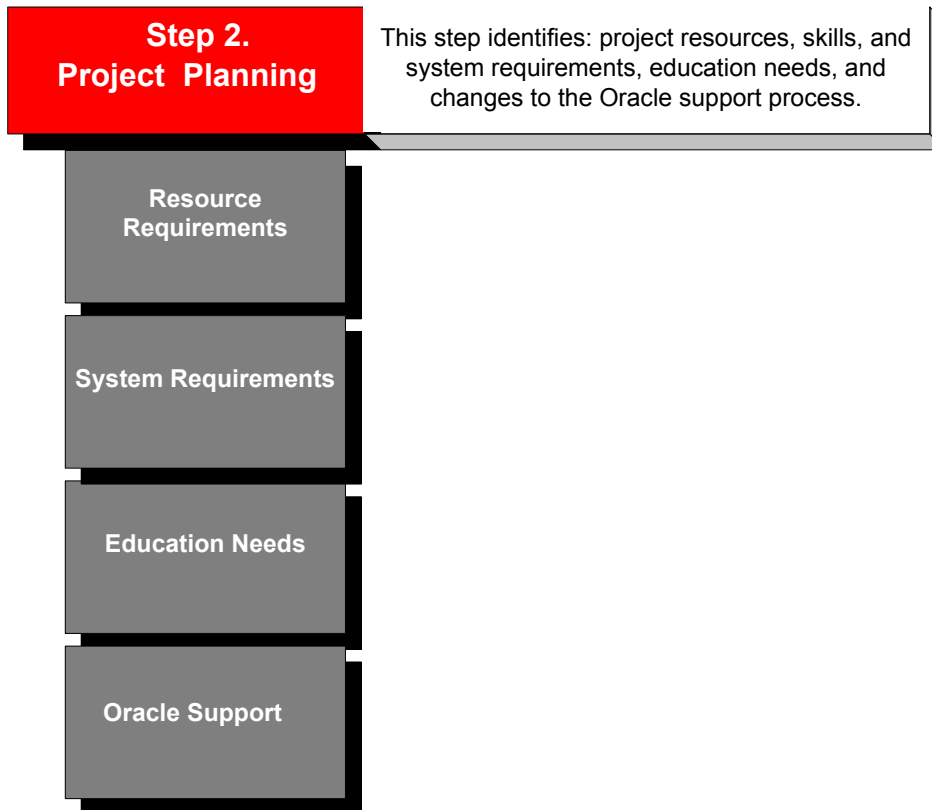
Table 1: Integration Checklist

<u>Step</u>	<u>Sub-Step</u>	<u>Complete</u>
1. Requirement Study		
	Pre-Integration Analysis and Considerations	
	Analysis of Model Documents	
	Data Configuration and Storage	
	Execution and Connectivity	
	Model Document and Data Analysis	
2. Project Planning		
	Resource Requirements	
	System Requirements	
	Education Needs	
	Oracle Support	
3. Implementation		
	Configuring OIPA	
	Data Mapping and gap analysis	
	Add data to transaction results	
	Add comments	
	Add Required Math Variables	
	Run Transaction and Test	
	Add Web Services Information	
	Construct Template for Data Transformation	
	Update Services File for New Service	
	Integration Process and Checklist	
	Configuring Documaker	
	Structured Documents	
	On Demand Documents	
	Configuring Transport	
4. Test and Deployment		
	OIPA	
	Documaker	

Step 2. Project Planning

The key to a successful project is in the planning. The project goal is to provide a seamless integration between OIPA and Documaker by defining roles, responsibilities, and deliverables.

Figure 7: Project Planning



RESOURCE REQUIREMENTS

It is recommended that a project team be comprised of IGBU product(s) trained specialists and customer trained personnel. The team's goal is to provide a seamless integration implementation between OIPA and Documaker. Definitions of project personnel roles and responsibilities are provided below:

Implementation Team Resources

Project Manager (PM)

The Project Manager will be the central contact point for the Customer team. Project Managers work together to set key deliverable dates for the migration project. The PM manages resources and task assignments at the customer side, and provides status reporting.

Product SME

The role of the product(s) Subject Matter Expert (SME) is to work with you and your staff to help with your integration project. The SME is involved in all phases of the project from pre-analysis, to running the integration tools, to composing integration templates, composition documents using DMStudio and testing.

Technical Contact

The technical contact reviews the Requirements Study Questionnaire responses and performs any technical tasks such as database connection configuration and any other non-business functions.

Customer Trained Resources

Project Manager (PM)

The Project Manager will be the central contact point for the Implementation team. Project Managers work together to set key deliverable dates for the migration project. The PM manages resources and task assignments at the customer side and provides status reporting.

Business SME

The Business SME is someone who is familiar with your model documents, new model document processes and business level requirements. This person will assist with resolution of mapping questions, provide test matrices and test XML files.

Technical Contact

The technical contact is the contact for database connectivity issues, technical questions, and technical deliverables as needed from the customer.

SYSTEM REQUIREMENTS

This guide provides a list of required products used by the integration development team. The product prerequisites and product installation steps are outlined in the product documentation and can be found on the [Oracle Technology Network site](#). You will need to install the products separately before attempting to use the integration components.

Table 2: Oracle Insurance Policy Administration Version and Documentation

<u>Oracle Insurance Policy Administration Product</u>	<u>Version</u>
Version	V 9.3 or higher
Documentation	Oracle Technology Network

Table 3: Documaker Products and Documentation

<u>Oracle Documaker Products</u>	<u>Version</u>
Oracle Documaker v11.5 for Microsoft Windows (32-bit)	V.11.5 or higher
Documaker Server	V.11.5
Documaker Studio	V.11.5
Docupresentment 2.2 p7 for Microsoft Windows (32-bit)	V 11.5
EWPS	V 11.5
Documentation	Oracle Technology Network

ORACLE EDUCATION

Oracle University is the premier provider of training for Oracle technologies and products. One of the largest global corporate training organizations, Oracle University offers thousands of in-class, onsite, Web-based, and CD-ROM courses for learners around the world. Oracle University can help customize a complete training plan that matches the customer's requirements. We train entire project teams according to their learning needs, timeline, and budget, or help individual learners find the right training by using our easy job-role learning paths. Oracle education advisors are experienced in identifying learning objectives and the needs of customers. Their mission is to provide customers with end-to-end educational approaches that meet the unique education requirements of each organization and exceed the expectations of our customers.

Oracle University courses are available in a variety of training formats to provide students with flexible learning options. In addition to courses taught at Oracle Learning Centers, we offer private events, training at customer sites and custom training. Delivery methods include Instructor-Lead, Live or Recorded Web Class training, Self-Study Online and CD-ROM. The courses and associated training formats and locations are subject to availability. For additional information please visit

http://education.oracle.com/pls/web_prod-plq-dad/db_pages.getpage?page_id=3

ORACLE INSURANCE POLICY ADMINISTRATION

We recommend that developers and take the four-day course for XML configurors, *Oracle Insurance Policy Administration Configuration I v9*.

In addition, there is a three-day course for advanced configuration, *Oracle Insurance Policy Administration Configuration: Advanced Math 9*. Additional configuration training is available via Oracle Consulting and may be provided as part of a customer's Oracle Insurance Policy Administration implementation.

- **Oracle Insurance Policy Administration Configuration I v9:** This course covers the basic fundamentals of rule configuration using version 9 of Oracle Insurance Policy Administration and the Rules Palette. You will learn how to use the system, how to properly use the building blocks for configuration, and how to use pre-defined XML elements for configuration. Any job role associated with implementing a policy administration system should attend this course.

Participants will learn how to use the front-end of the system for policy administration. The lifecycle of a policy is reviewed to provide a foundation for configuration tasks. Then you will learn how to configure business rules, system screens, fields, transactions, security and math elements. You will perform hands on configuration work for each topic, and at the end of class, you will perform an individual configuration project.

- **Oracle Insurance Policy Administration Configuration: Advanced Math 9:** This course covers advanced math configuration using version 9 of the Oracle Insurance Policy Administration product. You will learn how to configure arrays, perform functions and calculations on arrays, configure loops, and create user defined functions using predefined XML elements. Only those job roles associated with configuring should attend this course.

In this course, you will gain skills that are necessary for configuring calculations and business event processing. You will learn the various types of arrays available for configuration, how to fill them and perform various functions to manipulate their data. You will become skilled at configuring loops. Lastly, you will learn to configure reusable functions that calculate return values and output parameters according to incoming parameters. You will perform extensive hands on configuration work and complete and test for each topic.

ORACLE DOCUMAKER

We recommend that system administrators take the three-day course for document designers, *Oracle Insurance: Introduction to Documaker Studio*. In addition, there are several courses that are recommended for system administrators to support Oracle Documaker in an operational environment. Additional administrative training is available via Oracle Consulting and may be provided as part of a customer's Oracle Documaker implementation.

- **Oracle Insurance: Documaker Studio Batch Processing XML File:** Using XML file for input to batch, this class focuses on hands-on lab sessions, allowing for realistic implementation experience through independent decision-making and the application of the Oracle Documaker implementation process. Participants will develop the skills required to implement an end-to-end Oracle Documaker batch system using Documaker Studio. Also available for flat file batch processing.
- **Oracle Insurance: DAL Workshop for Batch Processing:** This course provides students with a full understanding of the purpose, benefits, general functionality, and locations for DAL scripts. Upon completion of the course, students will come away with a full understanding of the purpose, benefits, general functionality, and locations for DAL scripts. Terminology necessary for the creation of DAL scripts will be discussed in detail. Students learn to use several of the most common built-in functions and will have the opportunity to work with them through structured hands-on exercises. Exercises have been developed so that participants must learn the thought process to creating DAL scripts.

ORACLE SUPPORT

My Oracle Support is a personalized, collaborative support platform that incorporates service request creation, product knowledge base, access to patches and updates, and a community forum, all in one place. More information on each function is available from the full context-sensitive Help capability, located in the top-right corner of your screen after logging into My Oracle Support.

The OIPA and Documaker integration **does not** impact your Oracle support. However, when requesting support about the OIPA and Documaker integration, we recommend starting with the OIPA product number listed below:

Table 4: Product Name and Number

<u>Product Name</u>	<u>Product Number</u>
Oracle Insurance Policy Administration for J2EE	5279

Note: There are no changes to your Customer Support Identifier (CSI) number; however, if you do not already have a CSI number, the next two sections will guide you through requesting a CSI number and registering on Oracle's support site.

Requesting a SI (Support Identifier) Number

You will need a valid SI number to log a support inquiry. The SI number identifies your licensed products, maintenance level, and contract duration. You can request your SI number by contacting the Oracle Support Hub at 1-800-223-1711 (North America).

Table 5: Requesting a CSI

<u>Steps</u>	<u>Description</u>
Contact Oracle Support Hub	1-800-223-1711
Indicate the type of support request	Select 2, non-technical support
Provide Oracle Support Hub with your company name	Your company name is used to locate your licensed products, maintenance levels, and contract duration.
Provide Oracle Support Hub with the product name	Oracle Insurance Policy Administration for J2EE

Registration on the Oracle Support Site

The registration process uses a number of pages to gather relevant information so Oracle Support can contact you. To register, you need to visit <https://support.oracle.com> then click **New User?** Register and complete the pages listed below. Click **Next** to navigate from one page to the next.

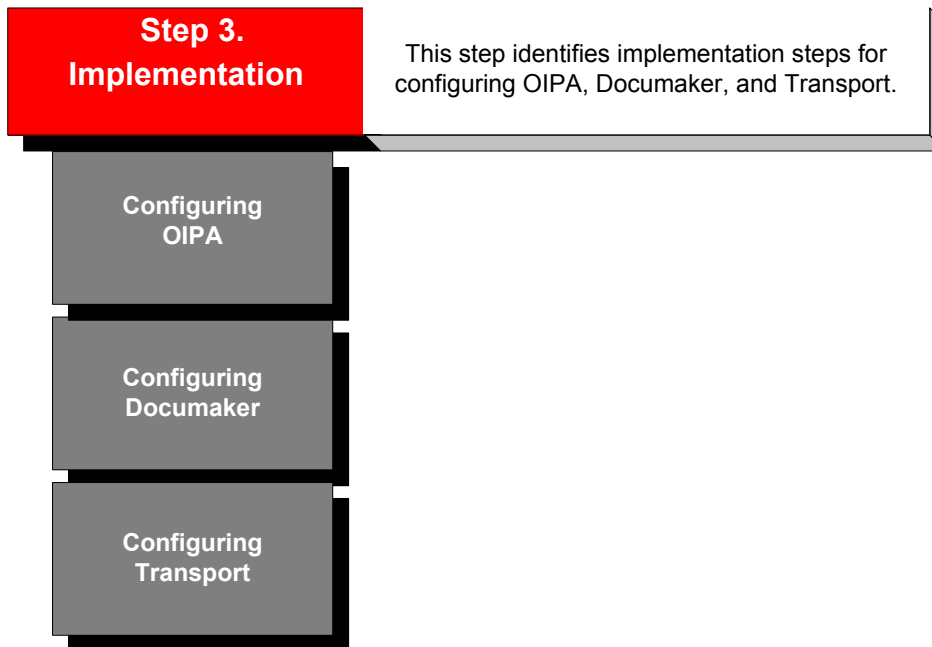
Table 6: Registration on Oracle Support Site

Prompt	Description
E-mail	E-mail is needed to login to Oracle Support and used to receive communication from Oracle Support.
Support Identifier, Terms of Use	You must have a valid support identifier which is also called SI number. SI numbers are associated with Oracle products so you may have multiple SI numbers.
Contact Information	Contact information is gathered so Oracle Support can communicate with you regarding your support requests.
Send and Wait	Send and wait page allows you to review information entered prior to submitting your registration request.

Step 3. Implementation

The implementation step provides an explanation, at a high-level, of the process and the representative tasks that the integration team followed when enabling the integration between OIPA and Documaker. We understand that each system implementation is unique, with different requirements, systems, and an environment; however, using a consistent methodology promotes efficiency and supportability.

Figure 8: Implementation



CONFIGURING OIPA

This section covers the OIPA configuration requirements to enable the OIPA and Documaker integration for Structured and On Demand documents. The paragraphs that follow will guide you through each of the configuration steps to construct or append a sample transaction for integration.

Step 1. Data Mapping and Identification of Data Gaps

Start with the report requirements and use the report field naming convention for the transform template field names. Then review the existing policy transaction for a MathVariable that maps to the report field requirement. When there is no MathVariable to map, record in the comment section how the MathVariable will be obtained, and use the existing report field name.

The ModeChange transaction example on the next page illustrates a policy owner confirmation letter.



INSURANCE

Report Field Name	Description	Transform Template (modechange) Field Name	Policy Transaction MathVariable if exists	Comments
ProcessDate	Letter process date	ProcessDate	EffectiveDateMV	
OwnerFullNm	Policy Owner's First and Last Name	OwnerFullNm	No	Create SQL call
OwnerAddrLine1	Policy Owner's 1 st street address	OwnerAddrLine1	No	Create SQL call
OwnerAddrLine2	Policy Owner's 2 nd street address	OwnerAddrLine2	No	Create SQL call
OwnerAddrLine3	Policy Owner's City, State and Postal Code	OwnerAddrLine3	No	Create SQL call
PolicyNum	Policy Number	PolicyNum	No	Create Activity Field variable
InsuredFullNm	Policy Insured's First and Last Name	InsuredFullNm	No	Create SQL call
SalutationNm	Policy Owner's Salutation and Last Name	SalutationNm	No	Create SQL call
OldMode	Policy's previous premium mode	OldMode	No	Need short description through SQL call
NewMode	Policy's new premium mode	NewMode	No	Need short description through SQL call
NewModalPremium	Policy's new premium modal amount	NewModalPremium	CurrYearPolicyModalPremium	
EffectiveDt	Policy Start Date of new premium modal	EffectiveDt	NextPaidToDate	
CSRFullNm	Customer Service Rep's First and Last Name	CSRFullNm	No	Create SQL call
ServicingAgentFullNm	Policy's Servicing Agent's First and Last Name	ServicingAgentFullNm	No	Create SQL call
ServicingAgentAddrLine1	Policy's Servicing Agent's 1 st street address	ServicingAgentAddrLine1	No	Create SQL call
ServicingAgentAddrLine2	Policy's Servicing Agent's 2 nd street address	ServicingAgentAddrLine2	No	Create SQL call
ServicingAgentAddrLine3	Policy's Servicing Agent's City, State and Postal Code	ServicingAgentAddrLine3	No	Create SQL call

Step 2. Add Data to Transaction Results

Open the Rules Palette for the OIPA system and check out the transaction to be modified (or CopyBook if transaction only calls the CopyBook).

Find the <MathVariables> section and go to the bottom in order to append the section for the variables that are needed for the confirmation letter. Again, we're using the ModeChange transaction as the example:

Figure 9: Math Variable Snippet

```
</MathVariable>
  <MathVariable VARIABLENAME="NegativeAdvanceMonths" TYPE="EXPRESSION" LOG="Yes"
  DATATYPE="INTEGER">(AdvanceMonths*-1)
</MathVariable>
```

Step 3. Add comments to delineate math variables

Figure 10: Math Variable with comments Snippet

```
</MathVariable>
  <MathVariable VARIABLENAME="NegativeAdvanceMonths" TYPE="EXPRESSION" LOG="Yes"
  DATATYPE="INTEGER">(AdvanceMonths*-1)
</MathVariable>
  <!-- Prepare Data for Integration -->
```

Step 4. Add the required MathVariables

When mapping MathVariables add comments before and after the closing MathVariables tag. Additional comments to break up sections are discretionary.

```
<MathVariable VARIABLENAME="NegativeAdvanceMonths" TYPE="EXPRESSION" LOG="Yes"
  DATATYPE="INTEGER">(AdvanceMonths*-1)</MathVariable>
<!-- Prepare Data for Integration -->
<!-- add CSR '11' -->
<MathVariable VARIABLENAME="CSRGUID" TYPE="SQL" DATATYPE="TEXT">SELECT AsRole.ClientGUID FROM
AsRole JOIN AsCode ON AsCode.CodeValue = AsRole.RoleCode WHERE AsCode.CodeName = 'AsCodeRole' AND
AsRole.RoleCode = '11' AND AsRole.PolicyGUID = '[Policy:PolicyGUID]'
```

Step 5. Run Transaction and Test Required Data Generated

The next step when new configuration is completed is to test the transaction to see if the MathVariables are correctly produced. Check-in your configuration in the Rules Palette. Next, set up a policy using one of the use cases and run the ModeChange activity to confirm the creation and values associated with these MathVariables. Resolve any issues and once the data is correctly produced, resume the Rules Palette to make the final configuration changes to the transaction.

Step 6. Add Web Service Information

The last step in the configuration modification process is to add the Web Service information to the MathVariables section right after the completion of the above data integration MathVariables. A gain, the recommendation is to start the section with a comment that delineates what function the section provides but is optional. The bolded areas represent required configuration for the Web Service. The only variable piece is the name of the ServiceID, which in our example is modechange. The rest of the configuration is the variables that will be passed to the template for transformation into the Documaker data format. Please note that there are two sections of configured variables. The first labeled “Common” is generally reusable variables for client letter confirmation. The second labeled “Specific Transaction Variables” reflect ones generally applicable to this transaction. There is no requirement for this separation, just a convention to assist with readability and configuring other transactions for integration.

Figure 11: Web Service Information Snippet

```
<MathVariable VARIABLENAME="NewModeShortDescription" TYPE="SQL" DATATYPE="TEXT">SELECT
ShortDescription FROM AsCode WHERE CodeName = 'AsCodeMode' and CodeValue = '[NewModeMV]'</MathVariable>
<!-- Integration Service -->
<MathVariable VARIABLENAME="ServiceID" TYPE="VALUE"
DATATYPE="TEXT">modechange</MathVariable>
<MathVariable VARIABLENAME="ServiceResult" TYPE="VALUE" DATATYPE="TEXT"/>
<MathVariable VARIABLENAME="ServiceError" TYPE="VALUE" DATATYPE="TEXT"/>
<MathVariable VARIABLENAME="OWSCRequest" TYPE="PROCESS" NAMESPACE="com.oi.owsc"
OBJECT="MathPlugin" DATATYPE="OBJECT">
<Parameters>
<!-- OWSC Variables -->
<Parameter NAME="service.id" TYPE="INPUT">ServiceID</Parameter>
<!-- Common Variables -->
<Parameter NAME="ProcessDt" TYPE="INPUT">EffectiveDateMV</Parameter>
<Parameter NAME="PolicyNumr" TYPE="INPUT">PolicyNumberMV</Parameter>
<Parameter NAME="CSRFullNm" TYPE="INPUT">CSRFullNm</Parameter>
<Parameter NAME="InsuredFullNm" TYPE="INPUT">InsuredFullNm</Parameter>
<Parameter NAME="SalutationNm" TYPE="INPUT">OwnerPreferredSalutation</Parameter>
<Parameter NAME="OwnerFullNm" TYPE="INPUT">OwnerFullNm</Parameter>
<Parameter NAME="OwnerAddrLine1" TYPE="INPUT">OwnerAddrLine1</Parameter>
<Parameter NAME="OwnerAddrLine2" TYPE="INPUT">OwnerAddrLine2</Parameter>
<Parameter NAME="OwnerAddrLine3" TYPE="INPUT">OwnerAddrLine3</Parameter>
<Parameter NAME="ServicingAgentFullNm" TYPE="INPUT">ServicingAgentFullNm</Parameter>
<Parameter NAME="ServicingAgentAddrLine1"
TYPE="INPUT">ServicingAgentAddrLine1</Parameter>
<Parameter NAME="ServicingAgentAddrLine2"
TYPE="INPUT">ServicingAgentAddrLine2</Parameter>
<Parameter NAME="ServicingAgentAddrLine3"
TYPE="INPUT">ServicingAgentAddrLine3</Parameter> -->
<!-- Specific Transaction Variables -->
<Parameter NAME="OldMode" TYPE="INPUT">OldModeShortDescription</Parameter>
```

```

<Parameter NAME="NewMode" TYPE="INPUT">NewModeShortDescription</Parameter>
<Parameter NAME="NewModalPremium"
TYPE="INPUT">CurrYearPolicyModalPremium</Parameter>
<Parameter NAME="EffectiveDt" TYPE="INPUT">NextPaidToDate</Parameter>
<!-- Output Parameter -->
<Parameter NAME="returnValue" TYPE="OUTPUT">ServiceResult</Parameter>
<Parameter NAME="errorValue" TYPE="OUTPUT">ServiceError</Parameter>
</Parameters>
</MathVariable>
</MathVariables>
</Math>

```

Step 7. Construct Template for Data Transformation

Once the transaction is producing the data for the integration and is configured to send the data via the Web Service, the data needs to be transformed into the Documaker format. The Web Service uses the template to set up the correct format including the KEY1, KEY2, LIBRARY and FORM tags that Documaker uses to access the correct form in its library. Besides retrieving the correct form, multiple forms can be retrieved as in our ModeChange example. A copy of the letter and cover sheet with mailing address goes to the servicing agent. Finally the variable fields are set up by matching the Parameter name used in the transaction.

Please note that logic can be configured in the template to remove variables that contain empty values through as conditional if checking for blank values. For instance in our ModeChange example, AddressLine2 would frequently be blank and it is desired to not pass the empty variable to Documaker so a blank line is not created in the form.

Based on our Modechange example, the following is the ModeChange template:

Figure 12: Template Ready for Transformation Snippet

```

<?xml version="1.0" encoding="UTF-8" ?>
<DOCUMENT TYPE="RPWIP" VERSION="11.5">
  <DOCSET NAME="">
    <KEY1 NAME="KEY1">ALAMERE INSURANCE COMPANY</KEY1>
    <KEY2 NAME="KEY2">Life</KEY2>
    <TRANSACTIONID NAME="KEYID">${PolicyNum}</TRANSACTIONID>
    <LIBRARY NAME="" CONFIG="Alamere">Alamere</LIBRARY>
    <ARCEFFECTIVEDATE NAME="CREATETIME">20100503</ARCEFFECTIVEDATE>
    <KEY1 NAME="KEY1">ALAMERE INSURANCE COMPANY</KEY1>
    <KEY2 NAME="KEY2">LIFE</KEY2>
    <KEYID NAME="KEYID">1111</KEYID>
    <TRANCODE NAME="TRANCODE">EN</TRANCODE>
    <STATUSCODE NAME="STATUSCODE"/>
    <DESC NAME="DESC"/>
    <LOCID NAME="LOCID"/>
    <SUBLOCID NAME="SUBLOCID"/>
    <JURISDICTN NAME="JURISDICTN"/>
    <TRNNAME NAME="TRNNAME"/>
    <QUEUEID NAME="QUEUEID"/>
    <GUIDKEY NAME="GUIDKEY">50EEFBD7103C40C0875EF5DC12FB6773</GUIDKEY>
    <FIELD NAME="CSRFullNm">${CSRFullNm}</FIELD>
    <FIELD NAME="ServicingAgentFullNm">${ServicingAgentFullNm}</FIELD>
    <FIELD NAME="ServicingAgentAddrLine1">${ServicingAgentAddrLine1}</FIELD>
    <#if ServicingAgentAddressLine2?? && ServicingAgentAddressLine2 != "">
    <FIELD NAME="ServicingAgentAddrLine2">${ServicingAgentAddrLine2!}</FIELD>
    </#if>
    <FIELD NAME="ServicingAgentAddrLine3">${ServicingAgentAddrLine3}</FIELD>
    <FIELD NAME="ProcessDt">${ProcessDt?date?string.long}</FIELD>
    <FIELD NAME="OwnerFullNm">${OwnerFullNm}</FIELD>
    <FIELD NAME="OwnerAddrLine1">${OwnerAddrLine1}</FIELD>
    <#if OwnerAddressLine2?? && OwnerAddressLine2 != "">
    <FIELD NAME="OwnerAddrLine2">${OwnerAddrLine2!}</FIELD>
    </#if>
    <FIELD NAME="OwnerAddrLine3">${OwnerAddrLine3}</FIELD>
    <FIELD NAME="PolicyNum">${PolicyNum}</FIELD>
    <FIELD NAME="InsuredFullNm">${InsuredFullNm}</FIELD>
    <FIELD NAME="SalutationNm">${SalutationNm}</FIELD>
    <FIELD NAME="OldMode">${OldMode}</FIELD>
    <FIELD NAME="NewMode">${NewMode}</FIELD>
    <FIELD NAME="NewModalPremium">${NewModalPremium?string.currency}</FIELD>
    <FIELD NAME="EffectiveDt">${EffectiveDt?date?string.long}</FIELD>
    <GROUP NAME="" NAME1="ALAMERE INSURANCE COMPANY" NAME2="LIFE">
    <FORM NAME="Agent Cover Letter">
    <RECIPIENT NAME="AGENT" COPYCOUNT="1"/>
    <SHEET/>
    </FORM>
    <FORM NAME="PCC-01">
    <RECIPIENT NAME="AGENT" COPYCOUNT="1"/>
    <RECIPIENT NAME="OWNER" COPYCOUNT="1"/>
    <SHEET/>
    </FORM>
    </GROUP>
  </DOCSET>
</DOCUMENT>

```

Step 8. Update Services File for New Service

Once the template is set up, the service-registry.xml needs to be updated for the new service and associated template. Open this file in your XML editor and add the <Service> section as follows. Make sure the Service ID name found in the transaction configuration matches the Transform File Name and is set up in the service-registry.xml. Repeat this process for additional templates by adding another service section to the existing file.

Please note that type="soap" is for real time processing and type="file" is for batch processing.

Figure 13: Service-Registry XML Snippet

```
<Services>
  <Service id="facechange" type="soap">
    <TemplateName>FaceChangeTemplate.ftl</TemplateName>
    <WSDLLocation>http://prod-vapp1.adminserver.com:8080/ewps-axis2/services/DocumentService?wsdl</WSDLLocation>
    <ServiceName>{http://webservices.docucorp.com/ewps/service/2005-12-01}DocumentService</ServiceName>
    <ServicePort>{http://webservices.docucorp.com/ewps/service/2005-12-01}EWPSDocumentServicesSoap</ServicePort>
  </Service>
  <Service id="modechange" type="soap">
    <TemplateName>ModeChangeTemplate.ftl</TemplateName>
    <WSDLLocation>http://prod-vapp1.adminserver.com:8080/ewps-axis2/services/DocumentService?wsdl</WSDLLocation>
    <ServiceName>{http://webservices.docucorp.com/ewps/service/2005-12-01}DocumentService</ServiceName>
    <ServicePort>{http://webservices.docucorp.com/ewps/service/2005-12-01}EWPSDocumentServicesSoap</ServicePort>
  </Service>
  <Service id="policyprint" type="soap">
    <TemplateName>PolicyPrintTemplate.ftl</TemplateName>
    <WSDLLocation>http://prod-vapp1.adminserver.com:8080/ewps-axis2/services/DocumentService?wsdl</WSDLLocation>
    <ServiceName>{http://webservices.docucorp.com/ewps/service/2005-12-01}DocumentService</ServiceName>
    <ServicePort>{http://webservices.docucorp.com/ewps/service/2005-12-01}EWPSDocumentServicesSoap</ServicePort>
  </Service>
  <Service id="policyprintfile" type="file">
    <TemplateName>PolicyPrintFileTemplate.ftl</TemplateName>
    <Directory>/opt/OIPA/documaker_input/policyprint/</Directory>
    <FileNameType>Random</FileNameType>
  </Service>
</Services>
```

Updating the service-registry.xml completes the configuration on the OIPA system and the data handoff is ready for Documaker.

An adjunct to the process is for the OIPA Implementation team to pass a mocked up XML file representing the output expected from the transformation process. This allows the Documaker Implementation team to test the form prior to running end-to-end and makes debugging simpler for both teams by isolating potential failure points.

CONFIGURING DOCUMAKER

This section covers the Documaker configuration requirements to enable the OIPA and Documaker integration for Structured documents and On Demand documents. The pages that follow establish the required Documaker products and required configuration options to complete the Documaker integration implementation.

STRUCTURED DOCUMENTS

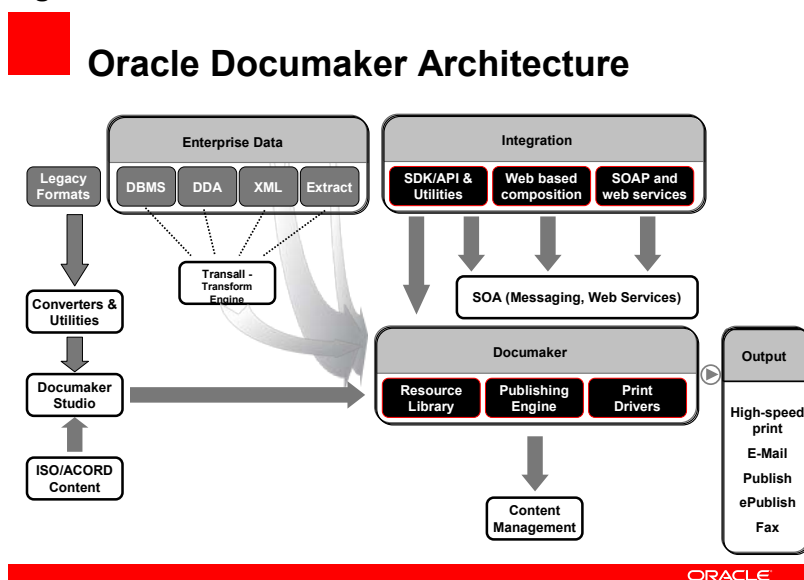
Product(s)

- Documaker Studio
- Documaker Server

Structured documents typically suggest high volume statements, bills, and policy print. Structured documents use **Documaker Studio** to compose documents and **Documaker Server** to process the OIPA XML files, merge data with Studio documents, and render documents in multiple formats.

Documaker Server is built using a multi-layered design, with each layer accessible via documented APIs. The layers range from the lowest level system calls and Web Services, and support SOA standards. Software Developer Kits (SDKs) consisting of toolboxes and libraries are available. They contain collections of common APIs.

Figure 14: Oracle Documaker Architecture



All of the components of Documaker use the same architecture and invoke common system APIs to read and write data, load and unload library resources, and to manage the objects and attributes inside the intelligent documents.

Configuration Options

- Documaker uses configurable INI and system files to tell the system how it should operate. To enable the OIPA and Documaker integration configure Documaker to accept a **list of XML files**.
- Ensure Documaker is setup to process the **Import XML file**.

Note: When processing Structured documents, OIPA and Documaker support sending and processing a list of XML files. The integration does not support processing a single file having concatenated XML files.

RELATED DOCUMENTATION

1. Oracle Documaker Installation Guide
2. Oracle Documaker Server System Reference Guide
3. Oracle Documaker Rules Reference Guide
4. Oracle Documaker Importing and Exporting XML Files with Documaker Server
5. Oracle Documaker Working with XML files

ON DEMAND DOCUMENTS

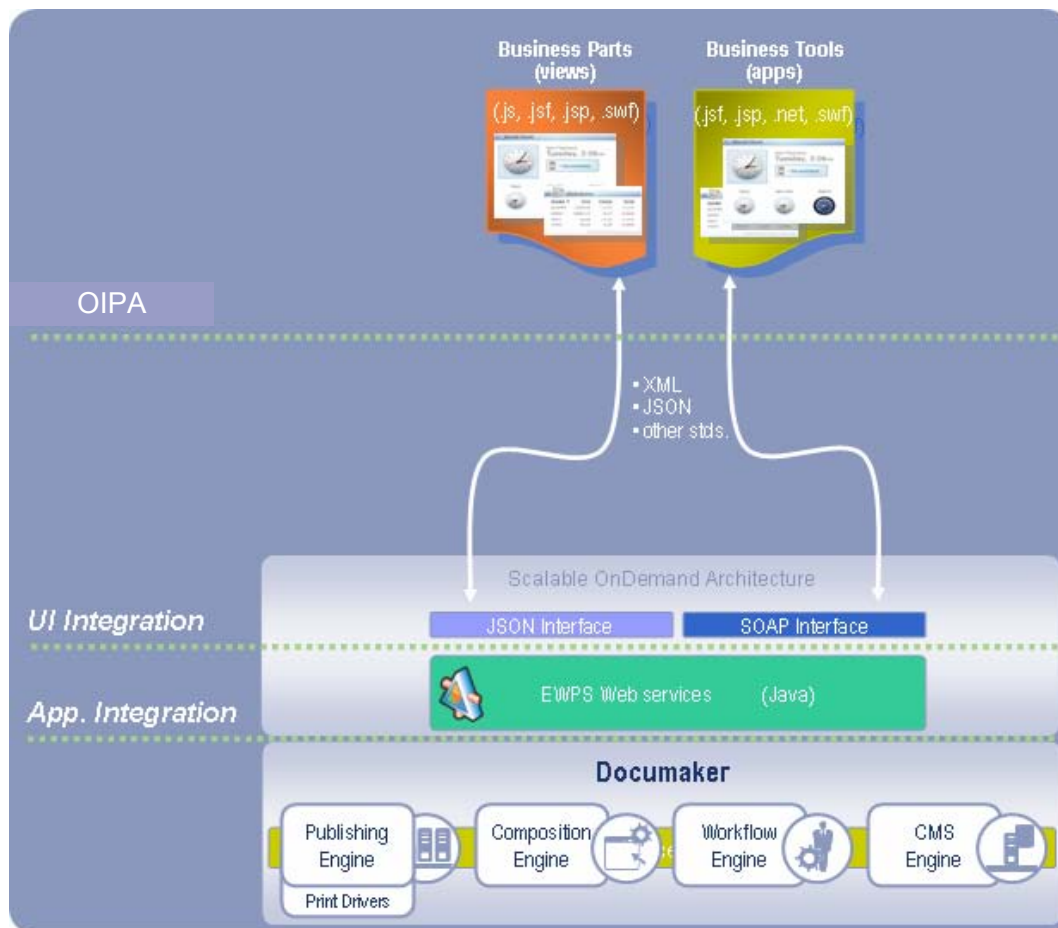
Product(s)

- Documaker Studio
- Documaker Server
- Docupresentment
- Enterprise Web Processing Services

On Demand documents refer to real-time generation of documents and imply no editing is allowed. On Demand documents use **Documaker Studio** to compose documents, **Docupresentment** to bridge to Documaker and the other publishing modules as well as integration with Web Services, **Enterprise Web Processing Services** to enable self-service publishing solutions, and **Documaker Server** to consume the OIPA XML file and render documents.

The Enterprise Web Processing Services (EWPS) framework offers functionality via a set of established and interoperable standards such as XML and Web Services. This allows a multitude of enterprise applications — including policy production and claims correspondence — to be designed and developed around a core functional infrastructure.

Figure 15: Overview



Configuration Options

- Documaker uses configurable INI and system files to tell the system how it should operate. To enable the OIPA and Documaker integration you need to configure Documaker to accept a **single XML file**.
- Ensure the Documaker system is setup to process the **Import XML file**.
- Add the **RPRUNRP** INI option to the your INI file.

```
< RPDRunRP >  
Executable = path to the gendaw32.exe  
Directory  = path to the working directory  
UserINI    = name of the FSIUSER file
```

Note: If the UserINI option does not include a drive letter, the system will look at the Directory option to find the path, so the full UserINI name becomes: Drive letter: Path\File Name.

BEST PRACTICE

1. When using the IDSServer INI option, set doPublishAttachment option to **No**. This eliminates sending the PDF encoded file in the response.
2. When changing Documaker INI files stop and restarted the Documaker service.

RELATED DOCUMENTATION

1. Oracle Documaker Installation Guide
2. Oracle Documaker Introduction to Enterprise Web Processing Services
3. Oracle Documaker Server System Reference Guide
4. Oracle Documaker Rules Reference Guide
5. Oracle Documaker Importing and Exporting XML Files with Documaker Server
6. Oracle Documaker Internet Documaker Server Guide

CONFIGURING TRANSPORT

The decision to choose real time or file transport depends on the business requirements surrounding document production.

If immediate production is required, real-time transport is the logical choice. Real time transport leverages the EWPS component for Documaker. When the OIPA activity is processed, a real time call is made to EWPS and Documaker creates the document. Subsequent processing can commence immediately.

If immediate production is not required, the file transport allows for document production to be delayed until a batch processing window is available. When the OIPA activity is processed, the extension writes the requested data to the specified directory. A script is then used to sweep these files together for Documaker. The script can be run on any schedule appropriate for the environment (hourly, daily, etc.). Subsequent processing must be delayed until after the script has run and the document has been produced.

FILE TRANSPORT CONFIGURATION

Service Registry Definition

A file transport is configured by specifying a type of "file" in the service description. Make sure the ID value matches the corresponding ServiceID from the OIPA transaction. A file transport also takes values for a directory and a file naming convention. Refer to the extension documentation for further explanation of these values.

```
<Service id="ServiceID" type="file">
  <TemplateName>ServiceTemplate.ftl</TemplateName>
  <Directory>/opt/OIPA/integration_work/ServiceID/xml</Directory>
  <FileNameType>Random</FileNameType>
</Service>
```

Recommended Filesystem Storage

It is important to organize the output from the file transport so that it can be effectively managed and passed to Documaker. The suggested directory is as follows:

- \$OUTPUT_ROOT/ - The top level directory for integration output

- ServiceID/ - One directory for each service producing batch output

- xml/ - Each service directory will have an xml directory for raw XML output

- pdf/ - Each service directory will have a pdf directory for finished PDF output

If OIPA and Documaker do not reside on a common system, the \$OUTPUT_ROOT can be shared. If volume is anticipated, this should leverage a high speed SAN for storage.

Batch Execution Script

Either the operating system scheduler or a commercial enterprise scheduler can execute this script. Please refer to the *Integration Studio Installation Guide* for further steps and detail.

REAL-TIME TRANSPORT CONFIGURATION

Service Registry Definition

A real time service call is configured by specifying a type of “soap” in the service configuration. Make sure the id value matches the corresponding ServiceID from the OIPA transaction. The SOAP transport method also requires the address of the WSDL for the EWPS service endpoint. The service name and port should remain the same, however, these values can be validated in the WSDL should an update to Documaker take place.

```
<Service id="ServiceID" type="soap">
  <TemplateName>ServiceTemplate.ftl</TemplateName>
  <WSDLLocation>http://server.company.com:8080/ewps-axis2/services/DocumentService?wsdl</WSDLLocation>
  <ServiceName>{http://webservices.docucorp.com/ewps/service/2005-12-01}DocumentService</ServiceName>
  <ServicePort>{http://webservices.docucorp.com/ewps/service/2005-12-01}EWPSDocumentServicesSoap</ServicePort>
</Service>
```

Update the Template for EWPS

The transform template defined previously is sufficient for defining the data Documaker needs to produce a requested document. For real time delivery, additional information needs to be added to satisfy the EWPS endpoint.

Append the following to the beginning of the template.

```
<#assign base64 = "com.oj.owsc.constructor.freemarker.Base64EncoderDirective"?new()>
<doPublishRequest xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xsi:type="doPublishReq_Import" xmlns="http://webservices.docucorp.com/ewps/schema/2005-12-01">
  <LibraryId>ALAMERE</LibraryId>
  <DistributionOptions xsi:type="DistributionOptions_PREDEFINED" source="PREDEFINED">
    <Priority>REALTIME</Priority>
  </DistributionOptions>
  <SourceType>IMPORT</SourceType>
  <Import>
    <ImportFile xsi:type="ImportFile_ATTACH" d3p1:contentType="" location="ATTACH" xmlns:d3p1="http://www.w3.org/2005/05/xmlmime">
      <@base64>
```

Append the following to the end of the template.

```
</@base64>
</ImportFile>
</Import>
</doPublishRequest>
```

These sections encode the Documaker XML for transport and provide EWPS with information about how to process the request.

Figure 16: Integration Configuration Workflow

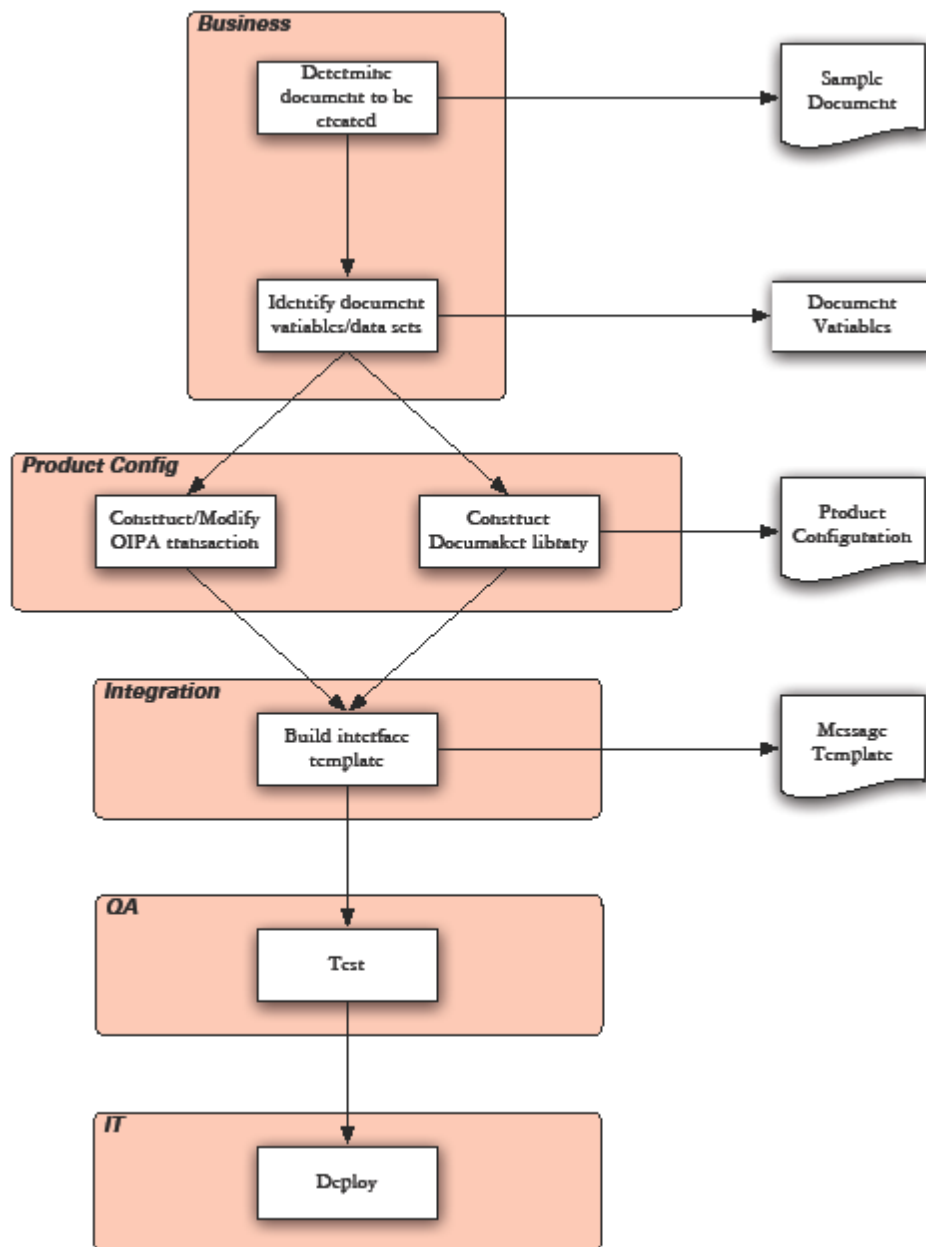


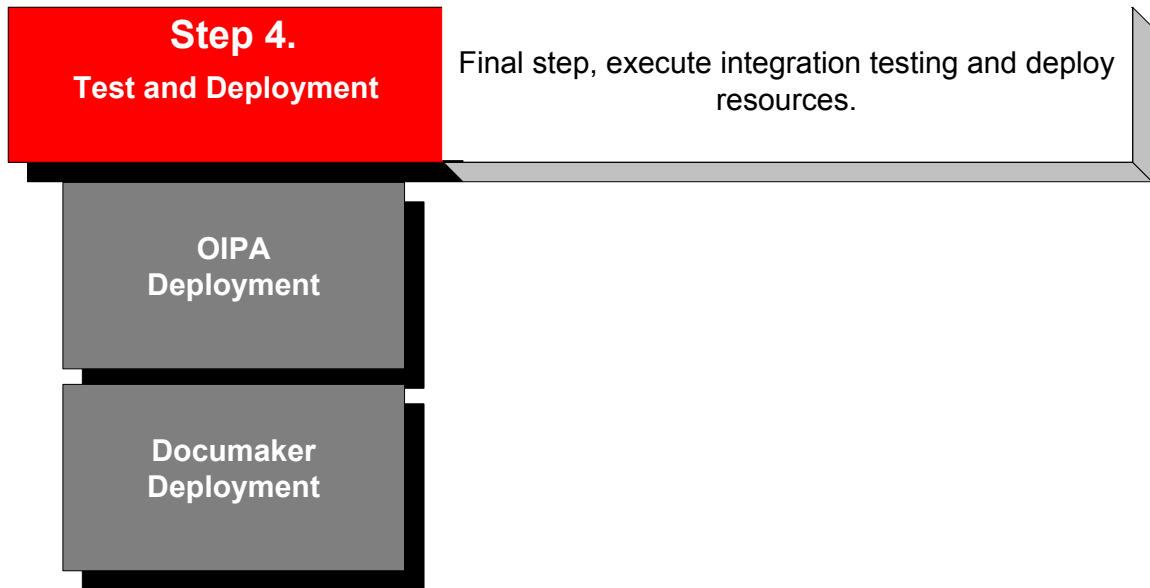
Table 7: Completed Integration Checklist

<u>Step</u>	<u>Sub-Step</u>	<u>Complete</u>
1. Requirement Study		
	Pre-Integration Analysis and Considerations	✓
	Analysis of Model Documents	✓
	Data Configuration and Storage	✓
	Execution and Connectivity	✓
	Model Document and Data Analysis	✓
2. Project Planning		
	Resource Requirements	✓
	System Requirements	✓
	Education Needs	✓
	Oracle Support	✓
3. Implementation		
	Configuring OIPA	✓
	Data Mapping and gap analysis	✓
	Add data to transaction results	✓
	Add comments	✓
	Add Required Math Variables	✓
	Run Transaction and Test	✓
	Add Web Services Information	✓
	Construct Template for Data Transformation	✓
	Update Services File for New Service	✓
	Integration Process and Checklist	✓
	Configuring Documaker	✓
	Structured Documents	✓
	On Demand Documents	✓
	Configuring Transport	✓
4. Test and Deployment		
	OIPA	✓
	Documaker	✓

Step 4. Test and Deployment

Deployment is the last development step, using Documaker's deployment tool you can deploy a library to the next environment in your document automation life cycle.

Figure 17: Deployment



OIPA DEPLOYMENT

OIPA features a comprehensive rule migration process for migrating configuration data between environments. Modifications to OIPA rules to invoke and configure integration functionality will be handled under this migration process. In addition, the OSC templates and service registry located on each server must be manually migrated between environments. Typically this involves a one-time setup of the service registry per environment and then a migration of template files when the data has changed.

DOCUMAKER DEPLOYMENT

DMStudio's deployment feature consists of several interrelated steps with possible transitions between them. Deployment can be interpreted as a *general process* that has to be customized according to specific project requirements or characteristics.

DEPLOYMENT STEPS

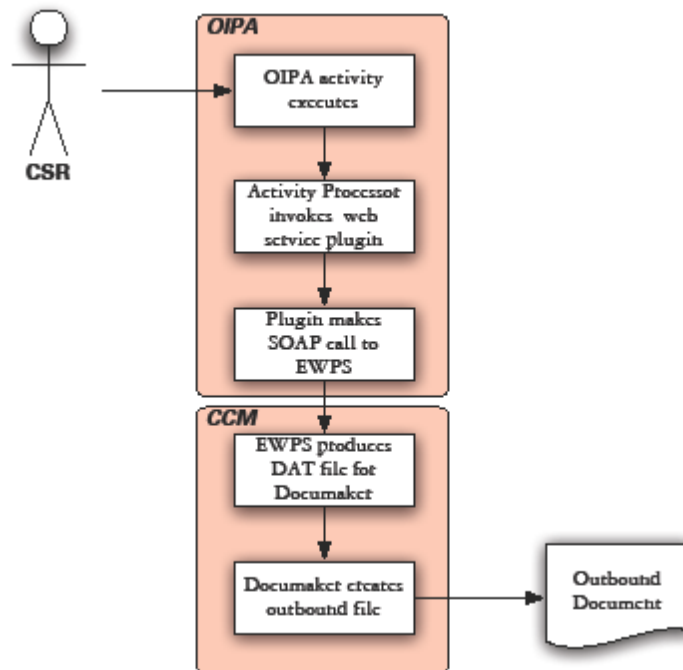
Requirements Phase

Deployment is an important and critical undertaking with any development project. As a best practice, the deployment requirements phase should take into consideration the following questions:

1. **Deployment owner**, who will own the deployment process?
2. **Deployment type**, what type of deployment your company will use?
3. **Deployment resources**, how are resources tagged for deployment?
4. **Deployment scripts**, what and how many deployment scripts are needed?

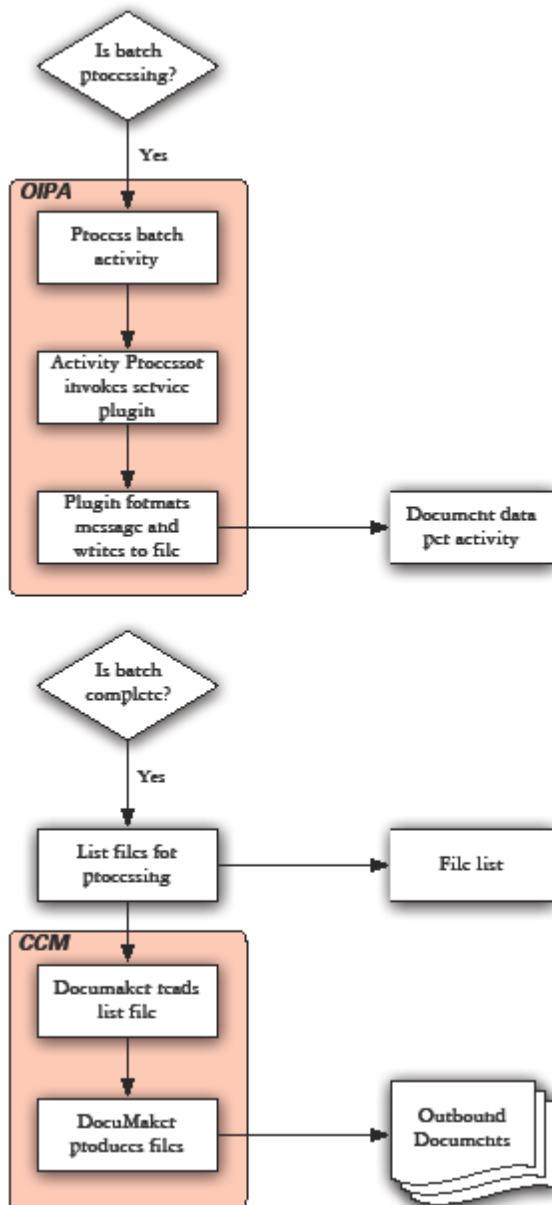
Appendix A: Real-Time Sample

Figure 18: On Demand Sample



Appendix B: Batch Sample

Figure 19: Structured Sample



Glossary of Terms

Structured documents: Structured documents typically suggest high volume statements, bills, and policy print.

On Demand documents: Real-time generation of documents implies no editing is allowed. For example, letters that confirm transactions that happened successfully, such as a letter to confirm a face increase.

Appendix C: Procedures in OIPA installation to run example

This appendix should be used in conjunction with the OIPA installation guide. By following the steps below in an OIPA installation you may view the working example of the Integration Accelerator.

OIPA Installation Steps

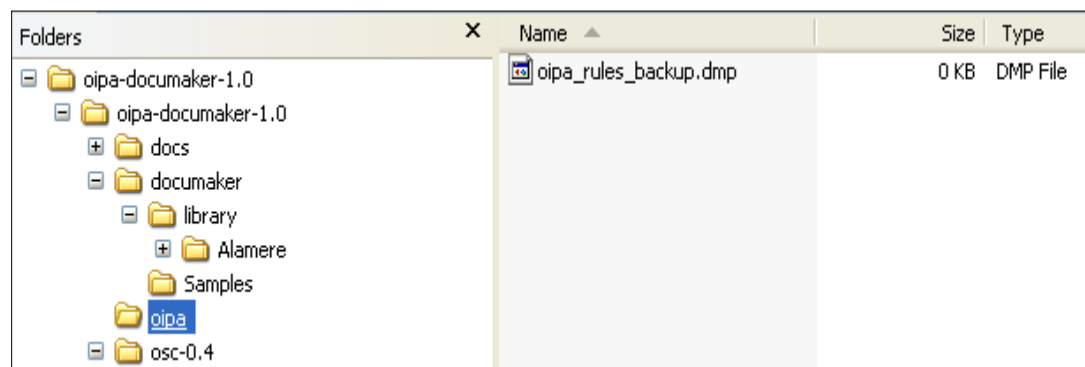
1. Database Import

Import OIPA database dump (.dmp) file using the below 'imp' command

```
imp <UserName>/<Pwd> FULL=y File=<oipa_rules_backup.dmp>
```

Refer to the folder structure below to locate the "oipa_rules_backup.dmp" file.

Figure 20: Folder Structure



2. Add Transaction Rules

(This step is required only if rules are not a part of dump file)

Refer to **Configuring OIPA** section of [How to guide for OIPA-Documaker](#).

3. Add Service Information to service-registry.xml

Refer to **Configuring OIPA** and **Service Registry Definition** section of [How to guide for OIPA-Documaker](#).

4. Add entries to osc.properties

Refer to [OSC Installation Guide](#) for more information.

5. Modifications to Coherence Cache Config

Add the following as mentioned in the [OSC Installation Guide](#):

```
<!-- Map Templates to Distributed Near Scheme -->
<cache-scheme-mapping>
  <cache-mapping>
    <cache-name>OSC_Templates</cache-name>
    <scheme-name>OSCScheme</scheme-name>
  </cache-mapping>
</cache-scheme-mapping>
<!-- OSC Distributed In-memory Cache -->
<cache-schemes>
  <local-scheme>
    <scheme-name>SampleMemoryScheme</scheme-name>
  </local-scheme>
  <distributed-scheme>
    <scheme-name>OSCScheme</scheme-name>
    <backing-map-scheme>
      <local-scheme>
        <scheme-ref>SampleMemoryScheme</scheme-ref>
      </local-scheme>
    </backing-map-scheme>
  </distributed-scheme>
</cache-schemes>
```

6. Add FTL Templates

Refer to **Update the Template for EWPS** section of this guide.

Appendix D: Procedures in running documaker environment

The steps in this guide describe how to:

- Install and configure the Alamere Studio files
- Verify that the system is working correctly

Deployment Steps Installed Directory Structures

The Alamere installation is a repository of print resources that have been integrated with the Oracle Insurance Policy Administration system. The Alamere installation creates folders and files developed for use with OIPA and are identified in the table below:

Table 8: Alamere Folders and Files

<u>Root Directory</u>	<u>Level 1</u>	<u>Level 2</u>	<u>Level 3</u>	<u>Content</u>
DOCSERV/				
	MSTRRES/			
		ALAMERE		Alamere.dxm Alamere.DXS AlamereLifeRun.bat fsisys.ini fsisyspub.ini fsiuser.ini fsiuserpub.ini men.res translat.ini Userinfo.dbf Userinfo.mdx
			ARC/	APPIDX.dbf APPIDX.mdx
			BDFLib/	ALAMERE.bdf
			DEFLIB/	AfgJob.jdt AfgJob_dopublish.jdt appidx.dfd EPFDB.dbf EPFDB.mdx lbylog.dbf lbylog.mdx MASTER.dbf MASTER.LBY MASTER.mdx NEWTRN.DFD RCBDFDFL.DFD Rel113.FXR rules.lst TRNDFDFL.DFD wip.dfd
			EDLLIB/	Standard Studio folder; not applicable to the Alamere Workspace.

INSURANCE

<u>Root Directory</u>	<u>Level 1</u>	<u>Level 2</u>	<u>Level 3</u>	<u>Content</u>
			FONTLIB/	Font files
			FORLIB/	Stores FAP files when opened in Studio.
			FORMS/	Forms specifies the directory path of library forms and logos
			GRPLIB/	Group file location
			DATA/	AGENCY.bch AGENT.bch BENEFICIARY.bch Counter.tbl ErrFile.dat error.bch LogFile.dat mfp.XML MsgFile.dat NaFile.dat NewTrn.dat OWNER.bch PolFile.dat
			INPUT/	Sample XML files: Cancellation_Letter.xml Death_Benefit_Letter.xml Grace_Letter.xml Lapse_Letter.xml Mode_Change_Confirmation_Letter.XML policyprintd.dat Policy_Print.xml
			PRINT/	Contains Generated PDF files: Also contained in "oipa-documaker-1.0\documaker\Sample Generated Letters" directory
			TABLE/	Standard Studio folder; not applicable to the Alamere Workspace.
			HELP/	Standard Studio folder; not applicable to the Alamere Workspace.
			WIP/	Standard Studio folder; not applicable to the Alamere Workspace.

ALAMERE STUDIO INSTALLATION STEPS

Before you begin

1. Create the directory structure to store Alamere Studio resources -
C:\Docserv\Mstrres
2. Download and review the readme.txt and release notes.
3. Unzip the zip file to the directory defined in step 1. The installation creates a folder called Alamere and 14 sub-folders.

Verify Documaker Batch Installation

1. Verify the installation finished properly by validating directories against the ALAMERE folders ([#Illustration 1: ALAMARE FOLDERS AND FILES](#)).

2. Go to the Alamere folder, open the ALAMERELIFERUN.BAT file. Update the batch file so it points to your local Documaker DLL files, default location is C:\FAP\DLL.

PATH = C:\FAP\DLL

Echo Y|Del ..\output*. * >NUL

Set UserINI=

C:\FAP\DLL\GenDaW32.Exe -INI=C:\Docserv\Mstrres\Alamere\FSIUSER.ini

Note: The Documaker Batch System uses the FSIUSER.INI and FSISYS.INI files.

3. To ensure your Alamere library was properly installed and configured, execute Documaker batch system using the sample test cases provided in the INPUT folder. Before running the XML files, copy the content of the installed PRINT and DATA folders into an archive folder (YYYY MM DD) and then execute **ALAMERELIFERUN.BAT**.

- Compare output files (NAFile, Polfile, errfile) with the files. Do they match?
 - If no, review installation step and re-run or contact Oracle Support.
 - If yes, the installation was successful.
- Create an archive folder (YYYY MM DD) and save your output files.

Verify Web Processing Installation

1. Complete the On Demand installation steps identified in this guide.

Note: The Documaker Batch System uses the FSIUSERPUB.INI and FSISYSPUB.INI files.

2. Go to the Alamere folder, open the FSISYSPUB.INI file and locate the **RPRUNRP** INI option. Ensure the Executable path is configured for your Documaker installation.

< RPDRunRP >

BaseLocation = c:\Docserv\mstres\Alamere\data\

Directory = c:\Docserv\mstres\Alamere\data\

Executable = c:\FAP\DLL\gendaw32.exe

UserINI = c:\Docserv\mstres\Alamere\fsiuserpub.ini

Note: If the UserINI option does not include a drive letter, the system will look at the Directory option to find the path, so the full UserINI name becomes: Drive letter: Path\File Name.

Add Company Forms

1. Insert your company's development, deployment, and integration procedures.