Hazardous Applications Notice

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 that 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.

Third Party Content, Products, and Services Disclaimer

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.
Preface

About This Document

This document will help you to understand how to integrate Oracle Documaker with the Oracle Revenue Management (ORMB) application on various supported platforms. It also explains how the bills and correspondence or letters are printed in the PDF format when Oracle Documaker is integrated with the ORMB application.

Intended Audience

This document is intended for the following audience:

- End-Users
- Administrators
- Consulting Team
- Implementation Team

Organization of the Document

The information in this document is organized into the following sections:

<table>
<thead>
<tr>
<th>Section No.</th>
<th>Section Name</th>
<th>Description</th>
</tr>
</thead>
</table>
Contents

1. Overview .......................................................................................................................... 1
2. Integration Process for Bill .............................................................................................. 3
   2.1 Bill Display and Batch Extract Process Flows ............................................................... 3
   2.2 Online Bill Display ....................................................................................................... 4
   2.3 Online Bill Display - Implementation Details (Banking) .............................................. 5
   2.4 Online Bill Display Algorithm (Banking) .................................................................... 8
   2.5 Sample XSL for Bill Extract - Documaker ................................................................... 9
   2.6 Banking Bill Extract Algorithm Type .......................................................................... 10
   2.7 Bill Route Type – Banking Route Type ....................................................................... 12
   2.8 Business Services (Banking) ..................................................................................... 13
   2.9 Batch Bill Process Flow ............................................................................................. 14
3. Integration Process for Correspondence .......................................................................... 16
   3.1 Online Letter Image - Implementation Details ............................................................. 19
   3.2 Online Letter Image Algorithm .................................................................................. 22
   3.3 Sample XSL for Letter Extract - Documaker ............................................................... 23
   3.4 Letter Template – Letter Extract Implementation Details ......................................... 24
   3.5 Letter Template – Letter Extract – (Auto Pay) Implementation Details ....................... 26
   3.6 Letter Template – Letter Extract – (Credit Card Expiry) Implementation Details ............ 27
   3.7 Letter Template – Letter Extract – (Notice of Intent to Cancel) Implementation Details .... 28
   3.8 Letter Template – Letter Extract – (Past Due) Implementation Details ....................... 29
   3.9 Business Service for Letter Generation ....................................................................... 31
   3.10 Batch Letter Process Flow .......................................................................................... 32
4. Setup for Oracle Documaker 12.6.0 Integration ................................................................. 34
   4.1 Documaker Setup For Linux ......................................................................................... 34
      4.1.1 Oracle Documaker Configuration for Batch Mode ............................................... 35
      4.1.2 Oracle Documaker Configuration for Online Mode ............................................... 36
   4.2 Setup ORMB for Oracle Documaker ............................................................................. 38
      4.2.1 Defining an Outbound Message Type ..................................................................... 38
      4.2.2 Defining an XAI Sender ......................................................................................... 39
      4.2.3 Defining an External System ............................................................................... 39
1. Overview

Oracle Revenue and Billing Management (ORMB) generates bills and correspondence or letters. Oracle Documaker presents bills and correspondence in the Portable Document Format (PDF).

ORMB sends an extract in the form of an XML message to Oracle Documaker. The Oracle Documaker application in turn generates Portable Document Format (PDF) files for bills and correspondences. These PDF files are rendered through a browser to the customer.

When you click the Display Bill button shown in ORMB-bill user interface for Bills, the integrated solution formats and creates a PDF version of the bill and displays it to you in a browser window. Similarly, PDF files are generated for correspondence when you click the online Display Letter button in ORMB-customer contact user interface. Sample extracts and templates are provided for bills and correspondence for Oracle Documaker integration including:

**Bills** -
- Banking Bill for Individual
- Banking Bill for IGA

**Correspondence** -
- Expiring credit card (with Manual Customer Contact)
- Auto-pay changes confirmation (with Manual Customer Contact)
- Notice of intent to cancel
- Past due date

The end to end process for the mass generation and distribution of documents using the ORMB to Oracle Documaker integration is as follows:

**Bills** -
- ORMB calculates and generates charges for a bill.
- ORMB extracts billing related data using a scheduled batch process. Further this extracted data could be sent to Oracle Documaker.
- Using the sample templates provided, Oracle Documaker formats the bill.

**Correspondence** -
- ORMB extracts the customer contact and related information as part of a batch process. Further this extracted data could be sent to Oracle Documaker.
- Oracle Documaker formats the correspondence using the sample templates.

Bills and correspondence can also be viewed in real time in order to answer customer questions or send copies of the bill and correspondence. The process for this is:
- Click the Online Bill Display button for bills or click the Display Letter button of customer contacts for correspondence.
• ORMB extracts the single bill or correspondence and sends it to Oracle Documaker.
• Oracle Documaker formats the bill or correspondence based on the templates.
• A PDF copy of the bill or correspondence is rendered in a separate browser window.

**Note:** This document covers the basic configuration details of Oracle Revenue Management and Billing (ORMB) artifacts used for Oracle Documaker integration. For more specific details, please refer to ORMB documentation and Online Help.

**Assumptions, Constraints and Dependencies:**

• Oracle Documaker Standard Edition 12.6.0 is tested and certified only for the Banking module on Linux platform.
• Oracle Documaker supports Online mode and Batch mode for Bill and Letter generation. In Online mode, Documaker can be invoked via Enterprise Web Processing Services (EWPS) or through native call. In case of Batch mode, Documaker can be invoked only via native call.
  
  For invoking Documaker through native call in Online or Batch mode, you need to install Documaker and ORMB on the same application server. However, while invoking Documaker via Web Services, you can install Documaker and ORMB on different application servers.
• All sample bills and correspondence are branded as “Alamere” and use the “Alamere” logo and address. We assume you will make a copy of the supplied template and make the necessary changes to brand these documents for your organization.
• Each bill type is associated with a template set in the Oracle Documaker application.
• Each correspondence type is associated with a template set in Oracle Documaker application.
• A customer contact is created within ORMB for Auto Pay and Expiring Credit Card. This capability is not included as a pre-configured capability at this time and must be configured as part of an implementation project.
• Current XSL files have been designed as per the sample Documaker templates. Implementation teams can modify the provided XSLs if additional fields need to be displayed, in addition to those provided with the base algorithm type released with the product.
• Documaker Linux integration requires all file paths which it refers to, to be in lower case. So all the file paths configured in ORMB Bill and Letter algorithms as well as those used in Documaker setup should be in lower case.
• There are limitations with respect to lengths of fields displayed on the Documaker templates.
• Documaker supports extract files only in XML format, flat files are not supported.
2. Integration Process for Bill

2.1 Bill Display and Batch Extract Process Flows

Step 1: On clicking the Display Bill button, ORMB extracts bill information, generates an XML file corresponding to format specified by the Documaker template and sends the XML message to Oracle Documaker.

Step 2: Oracle Documaker formats the bill based on the rules in the template and generates a PDF file.

Step 3: ORMB displays the PDF file in a separate browser window.

Note:
If you have integrated Oracle Documaker Standard Edition 12.6.0 with ORMB, the above mentioned steps are performed by an algorithm which is created using the C1_ONLNBILL algorithm type.

The algorithm type C1-DOCMAKALG which was earlier used with Oracle Documaker Standard Edition is deprecated and will no longer be supported. In its place, C1_ONLNBILL algorithm type should be used in all the future releases.

This algorithm is attached to the Online Bill Display system event in the Algorithms tab of the Installation Options – Framework screen.
2.2 Online Bill Display

For banking bills the Algorithm Type C1_BNKBLEX has to be associated with Wrapper Algorithm created using C1_ONLNBILL Algorithm Type.

The algorithm identifies the extract algorithm based on the bill routes defined with the main customer of an account. The following table shows the list of extract algorithms to be set up at bill route type. Each bill route type corresponds to a bill type.

<table>
<thead>
<tr>
<th>Bill Route Type on Main Customer</th>
<th>Bill Route Code</th>
<th>Bill Routing Method</th>
<th>Batch Control</th>
<th>Extract Algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Banking Route Type</td>
<td>BNKRTYP</td>
<td>Postal</td>
<td>POSTROUT</td>
<td>CM_BNKBLEX</td>
</tr>
</tbody>
</table>
The following table shows the list of Algorithm types and algorithm entities.

<table>
<thead>
<tr>
<th>Algorithm Entity</th>
<th>Algorithm Type</th>
<th>Algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Installation – Online Bill Display (Web Services Integration)</td>
<td>C1_ONLNBILL</td>
<td>CM_ONLNBILL</td>
</tr>
<tr>
<td>Bill Route Type – Banking Route Type</td>
<td>C1_BNKBLEX</td>
<td>CM_BNKBLEX</td>
</tr>
</tbody>
</table>

**Note**: Each extract algorithm will internally invoke six (for Banking) business services to extract the data and transform the same into output data as required by the Oracle Documaker application.

It is recommended to define all the required soft parameters for each and every algorithm mentioned in the document. For Oracle Linux please note that paths mentioned for soft parameters are case sensitive.

### 2.3 Online Bill Display - Implementation Details (Banking)

The following table provides information about the C1_ONLNBILL algorithm type:

<table>
<thead>
<tr>
<th>Algorithm Type</th>
<th>C1_ONLNBILL</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Description</strong></td>
<td>Wrapper Algorithm Type for Online Bill Display</td>
</tr>
<tr>
<td><strong>Detailed Description</strong></td>
<td>This Algorithm is created for online bill view. It identifies respective algorithms based on the bill route type configuration corresponding to respective bills. The integration with third party application can be through batch script or web services. Sleep time needs to be configured in case of integration using batch script. When integrating using web services, XAI outbound message configuration is required. Two folders need to be created one for XML extract and other for storing the Dat files. Input File Path is the root folder in which Dat file folder and Extract folder will be created. Output file path is the full path to the folder where third party application e.g. Oracle Documaker will generate PDF files. Create this folder if it does not exist. For native mode invocation of Documaker, Input File Path, Dat and Extract folder names and Output File Path should exist in lower case. Bill View folder parameter is used to create the URL for calling a program that displays the generated PDF to the user. The bill view folder path will be relative to the “context root” of the web server. The value for this parameter should be specified as ‘billView’, if the default URL pattern mapping of the program is to be used.</td>
</tr>
</tbody>
</table>
Bat file name is the name of the bat file which will invoke application specific executable. It internally sends two parameters one for the input DAT file name and other for the expected output PDF file name. Output Http URL is the web server address under which the ‘Output file path’ folder is configured.

**Invocation point(s)**

<table>
<thead>
<tr>
<th>Name</th>
<th>Online Bill Display – (View Bill)</th>
</tr>
</thead>
</table>

**Program Type**

<table>
<thead>
<tr>
<th>Name</th>
<th>Java</th>
</tr>
</thead>
</table>

**Name**

<table>
<thead>
<tr>
<th>Name</th>
<th>com.splwg.ccb.domain.billing.billtransform.OnlineBillDisplayAlgo</th>
</tr>
</thead>
</table>

**Parameters**

<table>
<thead>
<tr>
<th>Name</th>
<th>Required (Yes/No/Conditional)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input File Path</td>
<td>Yes</td>
<td>File path where Dat and Extract folders have been created. For Native mode invocation of Documaker, this path should be in lower case.</td>
</tr>
<tr>
<td>Encoding Class Name</td>
<td>No</td>
<td>This feature is not supported in this release.</td>
</tr>
<tr>
<td>Outbound Message Type</td>
<td>Conditional</td>
<td>Message type to be sent to the external system. This parameter is required if Invocation Mode parameter value is “WEB”</td>
</tr>
<tr>
<td>External System</td>
<td>Conditional</td>
<td>External system to be configured as per interface of external System. This parameter is required if Invocation Mode parameter value is “WEB”</td>
</tr>
<tr>
<td>Outbound Message Business Object</td>
<td>Conditional</td>
<td>Business Object created using existing MO (F1-OUTMSG) and default Application Service (F1-DFLTS). This parameter is required if Invocation Mode parameter value is “WEB”</td>
</tr>
<tr>
<td>Parameter</td>
<td>Required</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------</td>
<td>----------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Output File Path</td>
<td>Yes</td>
<td>Name of the file path where PDF will be generated. Ensure that this file path should be same as the billView Output Directory Path (property <code>ouaf.runtime.billView.directoryPath</code>) provided in <code>&lt;SPLBASE&gt;/etc/conf/root WEB-INF/classes/spl.properties</code>, where <code>&lt;SPLBASE&gt;</code> is the environment variable pointing to the root directory of ORMB installation. For Native mode invocation of Documaker, this file path should exist in lower case. <strong>Note:</strong> In case the value of Output File Path has to be modified, change the value of billView Directory Path as both the values should be in sync.</td>
</tr>
<tr>
<td>Dat Folder Name</td>
<td>Yes</td>
<td>Used by the external system to get the path of extract files. This folder is created under the location specified in Input File Path parameter. For Native mode invocation of Documaker, this folder name should be in lower case.</td>
</tr>
<tr>
<td>Bat File Name</td>
<td>Conditional</td>
<td>BAT file name for converting Documaker extract XML files into PDF. This parameter is required if Invocation Mode parameter value is “NATV”</td>
</tr>
<tr>
<td>Output HTTP URL</td>
<td>Yes</td>
<td>URL name where output PDF will be displayed.</td>
</tr>
<tr>
<td>Bill View Folder</td>
<td>Yes</td>
<td>Used for creating the URL for displaying the generated PDF. Specify this value as ‘billView’ for the default behavior.</td>
</tr>
</tbody>
</table>
Sleep Time for Online Display | Conditional | Sleep time needs to be configured in case of integration using executable. The time needs to be configured in milliseconds (e.g. to configure 5 seconds enter 5000 as value). This parameter is required if Invocation Mode parameter value is “NATV”

Configuration Folder | Conditional | The Master Resource Library name/Library Id to be used in Documaker. This parameter is required if Invocation Mode parameter value is “WEB”

Invocation Mode | No | This indicates the external report generation tool to be invoked. Supported values for Documaker are: Valid values are: NATV – For calling native Documaker WEB – For calling Documaker using web services. When not specified, value of this parameter is “NATV”. Note: This parameter is case-sensitive.

Detailed Design | The Online Bill Display Wrapper Algorithm will call specific Bill Extract Algorithm based on the Bill Route Type configured for the bill.

### 2.4 Online Bill Display Algorithm (Banking)

The following table lists the parameters defined in the sample algorithm named CM_ONLNBILL. These are sample values and can be modified as per the user configuration.

<table>
<thead>
<tr>
<th>Algorithm Name</th>
<th>CM_ONLNBILL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameters</td>
<td></td>
</tr>
<tr>
<td>Input File Path</td>
<td>/spl/temp/</td>
</tr>
<tr>
<td>Encoding Class Name</td>
<td>(Not supported in this release)</td>
</tr>
<tr>
<td>Outbound Message Type</td>
<td>C1-DOCMAKOM</td>
</tr>
</tbody>
</table>
### External System
- **C1-DOCMAKEXTSYS**

### Outbound Message Business Object
- **C1-DOCMAKBO**

### Output File Path
- `/spl/temp/billview`

### Dat Folder Name
- `data`

### Bat File Name
- `<DOCUMAKER_INSTALL_DIR>/ormb/input/rmbonlineprint.sh`

### Output HTTP URL
- `http://<host>:<port>/ouaf/`

### Bill View Folder
- `billView`

### Sleep Time for Online Display
- `5000`

### Configuration Folder
- `ormb`

### Invocation Mode
- `WEB`

---

**Note:** For more information on how to define Outbound Message Type and External System, refer to the [Setup ORMB for Oracle Documaker 12.6.0](#) section.

---

### 2.5 Sample XSL for Bill Extract - Documaker

ORMB provides below XSL for generating extract XML for Documaker from raw ORMB XML for Individual and IGA accounts:

<table>
<thead>
<tr>
<th>Type of Account</th>
<th>XSL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Individual</td>
<td><code>&lt;SPLEXBASE&gt;/splapp/xai/schemas/samples/billTransform/ORMB_DM_Banking_Bill_Extract.xsl</code></td>
</tr>
<tr>
<td>IGA</td>
<td><code>&lt;SPLEXBASE&gt;/splapp/xai/schemas/samples/billTransform/ORMB_DM_Comm_Banking_List_Bill.xsl</code></td>
</tr>
</tbody>
</table>

These XSLs need to be configured in the Banking Bill extract algorithm.
2.6 Banking Bill Extract Algorithm Type

The following table provides information about the C1_BNKBLEX algorithm type:

<table>
<thead>
<tr>
<th>Algorithm Type</th>
<th>C1_BNKBLEX</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>Banking Bill Extract Algorithm Type</td>
</tr>
<tr>
<td>Detailed Description</td>
<td>This algorithm type is created for extracting Banking bills. This will internally invoke six business services to extract the data and transform the same into output data as required by a third party application. Six base business services invoked are C1-BnkBillHeaderExtract, C1-InvoiceSummaryExtract, C1-BnkAcctSmry, C1-BnkServiceSmry, C1-BnkServiceActivityExtract and C1-BnkPriorPerdAdjstmntExtract for extracting Banking Bill Header, Invoice Summary, Account Summary, Service Summary, Service Activity and Prior Period Adjustments for individual and commercial list (IGA) bills. A custom business service can also be configured in case any additional enrichment of the output XML needs to be done. The algorithm program transforms the internal ORMB XML into outbound XML as per the XSL configured on respective algorithms.</td>
</tr>
<tr>
<td>Invocation point(s)</td>
<td>Bill Route Type</td>
</tr>
<tr>
<td>Program Type</td>
<td>Java</td>
</tr>
<tr>
<td>Name</td>
<td>com.splwg.ccb.domain.billing.billtransform.BankingBillExtractAlgo</td>
</tr>
<tr>
<td>Parameters</td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Required (Yes/No)</td>
</tr>
<tr>
<td>Tax Identifier Code</td>
<td>No</td>
</tr>
<tr>
<td>Tax Component</td>
<td>No</td>
</tr>
<tr>
<td>Charge Calc Line Characteristic</td>
<td>No</td>
</tr>
<tr>
<td>Tax Calc Line Value</td>
<td>No</td>
</tr>
<tr>
<td>Discount Calc Line Value</td>
<td>No</td>
</tr>
<tr>
<td>Custom Business Service</td>
<td>No</td>
</tr>
<tr>
<td>Table Row</td>
<td></td>
</tr>
<tr>
<td>-----------</td>
<td></td>
</tr>
<tr>
<td>XSL File Name For Individual Account</td>
<td>No</td>
</tr>
<tr>
<td>XSL File Name For IGA Account</td>
<td>No</td>
</tr>
<tr>
<td>Extract Folder Name</td>
<td>No</td>
</tr>
<tr>
<td>Dat Folder Name</td>
<td>No</td>
</tr>
<tr>
<td>Bill Date Format</td>
<td>No</td>
</tr>
<tr>
<td>Date Format for other dates in the Bill</td>
<td>No</td>
</tr>
<tr>
<td>External System Input String</td>
<td>No</td>
</tr>
<tr>
<td>SA Characteristic Type</td>
<td>No</td>
</tr>
</tbody>
</table>
### 2.7 Bill Route Type – Banking Route Type

The following table lists the parameters defined in the sample algorithm named CM_BNKBIEX. These are sample values and can be modified as per the user configuration.

<table>
<thead>
<tr>
<th>Algorithm Name</th>
<th>CM_BNKBIEX</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameters</td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Value</td>
</tr>
<tr>
<td>Tax Identifier Code</td>
<td>TAXID</td>
</tr>
<tr>
<td>Tax Component</td>
<td></td>
</tr>
<tr>
<td>Charge Calc Line Characteristic</td>
<td>CHRG_TYP</td>
</tr>
<tr>
<td>Tax Calc Line Value</td>
<td>TAX</td>
</tr>
<tr>
<td>Discount Calc Line Value</td>
<td>DISCOUNT</td>
</tr>
<tr>
<td>Custom Business Service</td>
<td>&lt;SPLEBASE&gt;/splapp/xai/schemas/samples/billTransform/ORMB_DM_Banking_Bill_Extract.xsl</td>
</tr>
<tr>
<td>XSL File Name For Individual Account</td>
<td>&lt;SPLEBASE&gt;/splapp/xai/schemas/samples/billTransform/ORMB_DM_Banking_Bill_Extract.xsl</td>
</tr>
<tr>
<td>XSL File Name For IGA Account</td>
<td>&lt;SPLEBASE&gt;/splapp/xai/schemas/samples/billTransform/ORMB_DM_Comm_Banking_List_Bill.xsl</td>
</tr>
<tr>
<td>Extract Folder Name</td>
<td>extract</td>
</tr>
<tr>
<td>Dat Folder Name</td>
<td>data</td>
</tr>
<tr>
<td>Bill Date Format</td>
<td>MMMM,dd,yyyy</td>
</tr>
<tr>
<td>Date Format for other dates in the Bill</td>
<td>MMMM,dd,yyyy</td>
</tr>
</tbody>
</table>
### 2.8 Business Services (Banking)

The following base business services are invoked in bill generation:

- C1-BnkBillHeaderExtract
- C1-InvoiceSummaryExtract
- C1-BnkAcctSmry
- C1-BnkServiceSmry
- C1-BnkServiceActivityExtract
- C1-BnkPriorPerdAdjstmntExtract

These business services are used for extracting Bill Header, Invoice Summary, IGA Account Summary, Service Summary, Service Activity and Prior Period Adjustments, respectively, for Individual and IGA account bills.

A custom business service can also be configured in case any additional enrichment of the output XML needs to be done.

The Banking Bill Extract algorithm internally invokes these business services to extract the data and transform the same into output data as per the XSL configured on it.

Please refer to ORMB Banking User Manual for more information on these business services.

<table>
<thead>
<tr>
<th>External System Input String</th>
<th>XML_FILE_INPUT</th>
</tr>
</thead>
<tbody>
<tr>
<td>SA Characteristic Type</td>
<td>C1_SAFCD</td>
</tr>
<tr>
<td>IGA Account Characteristic Type</td>
<td>C1_F_IGA</td>
</tr>
<tr>
<td>Charge Characteristic Type</td>
<td>CUORIGAM</td>
</tr>
<tr>
<td>Exchange Rate Characteristic Type</td>
<td>EXCHG-RT</td>
</tr>
<tr>
<td>Pricing Currency Characteristic Type</td>
<td>CUPRICCD</td>
</tr>
<tr>
<td>Price Item Relationship Flag</td>
<td>SVC</td>
</tr>
</tbody>
</table>
2.9 Batch Bill Process Flow

To generate bills through batches, in addition to the configuration done for Banking Bill Extract Algorithm (C1_BNKBLEX), also configure the Batch Control of bill extract (POSTROUT).

In the batch control, set the value of FILE_NAME parameter to the dat file name which will be generated with the extract information as shown below:

FILE_NAME = <<Path of Dat Folder>>/<<Output file name for bill>>

Here, <<Path of Dat Folder>> is the one created during Bill Extract algorithm (C1_BNKBLEX) configuration.
For example,

    FILE_NAME = /spl/temp/data/5412563416

For more information regarding POSTROUT batch configuration, please refer to ORMB Online Help.

**Note:** Ensure that all files names along with the complete path referred to or generated by the batch control should exist in lower case. This includes the FILE_NAME parameter passed to the batch control as well as the extract file paths mentioned in the generated dat file.
3. Integration Process for Correspondence

**Step 1:** On click of “Display Letter” button, ORMB extracts the letter information and generates an XML corresponding to formats required by the Documaker templates. The XML message is then sent to Oracle Documaker.

**Step 2:** Oracle Documaker will format and will generate a PDF file.

**Step 3:** ORMB will receive the PDF and will display the PDF file in a separate browser window.

A wrapper algorithm ‘C1_XLETOLALG’ is defined in Installation Options - Framework for Online Letter Image.
Based on letter extract Algorithm Type (C1-LETXTALG) an algorithm ‘CM_XLETOLALG’ is used as wrapper algorithm. The algorithm will identify the extract algorithm based on the letter template defined with customer contact type. Each letter template is associated with the extract algorithm.

Following table shows list of extract algorithms for each letter types:

<table>
<thead>
<tr>
<th>Customer Contact Type</th>
<th>Letter Template</th>
<th>Batch Control</th>
<th>Extract Algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>AUTOPAY Auto Pay source Change</td>
<td>AUTO-PAY-CHG (Auto Pay Change)</td>
<td>LTRPRT</td>
<td>CM_XLETEXTAU (Letter Extract Algorithm for Auto Pay changes.)</td>
</tr>
<tr>
<td>CREDIT CARD Expiry Correspondence</td>
<td>CC-EXP-TMPL (Credit Card Expiry Letter)</td>
<td>LTRPRT</td>
<td>CM_XLETEXTCC (Algorithm for credit card expiry letters.)</td>
</tr>
<tr>
<td>Notice of Intent to Cancel</td>
<td>C1-NI-TMPL (Notice of Intent to Cancel)</td>
<td>LTRPRT</td>
<td>CM_XLETEXTNI (Algorithm for notice of intent letters.)</td>
</tr>
<tr>
<td>Past Due Letter</td>
<td>C1-OVR-DUE (Over Due Letter)</td>
<td>LTRPRT</td>
<td>CM_XLETEXTPD (Algorithm for past due letters.)</td>
</tr>
</tbody>
</table>

Following table shows list of Customer Contact Type settings:

<table>
<thead>
<tr>
<th>Contact Type</th>
<th>Contact Class</th>
<th>Description</th>
<th>Contact Action</th>
<th>Letter Template</th>
<th>Characteristics</th>
</tr>
</thead>
<tbody>
<tr>
<td>AUPC</td>
<td>APAY</td>
<td>Auto Pay source Change</td>
<td>Send Letter</td>
<td>AUTO-PAY-CHG (Auto Pay Change)</td>
<td>Auto Pay Characteristics on Account (Required : YES)</td>
</tr>
<tr>
<td>EXPCRCARD</td>
<td>CC</td>
<td>Credit Card Expiry</td>
<td>Send Letter</td>
<td>CC-EXP-TMPL (Credit Card Expiry Letter)</td>
<td>• Name on File for Credit Card Expiry Customer Contact (Required : YES)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Correspondence</td>
<td></td>
<td></td>
<td>• Expiration Date for Credit Card Expiry Customer Contact (Required : YES)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>• Account Number for Credit Card Expiry Customer Contact (Required : YES)</td>
</tr>
<tr>
<td>Contact Type</td>
<td>Contact Class</td>
<td>Description</td>
<td>Contact Action</td>
<td>Letter Template</td>
<td>Characteristics</td>
</tr>
<tr>
<td>---------------</td>
<td>---------------</td>
<td>------------------------</td>
<td>----------------</td>
<td>-----------------------</td>
<td>-----------------</td>
</tr>
<tr>
<td>OD REMIND</td>
<td>C&amp;C (Credit and collection contacts)</td>
<td>Past Due Letter</td>
<td>Send Letter</td>
<td>C1-OVR-DUE (Over Due Template)</td>
<td>NIL</td>
</tr>
<tr>
<td>NOTICE_CNCL</td>
<td>C&amp;C (Credit and collection contacts)</td>
<td>Notice of Intent to Cancel Letter</td>
<td>Send Letter</td>
<td>C1-NI-TMPL (Notice of Intent Letter)</td>
<td>NIL</td>
</tr>
</tbody>
</table>

Following table shows list of new Algorithm types, algorithm entities and algorithms which can be created during implementation.

<table>
<thead>
<tr>
<th>Algorithm Entity</th>
<th>Algorithm Type</th>
<th>Algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Installation – Online Letter Image</td>
<td>C1_XLETOLALG</td>
<td>CM_XLETOLALG</td>
</tr>
<tr>
<td>Letter Template – Letter Extract</td>
<td>C1-LETXTALG</td>
<td>CM_XLETXTAU</td>
</tr>
<tr>
<td>Letter Template – Letter Extract</td>
<td>C1-LETXTALG</td>
<td>CM_XLETEXTCC</td>
</tr>
<tr>
<td>Letter Template – Letter Extract</td>
<td>C1-LETXTALG</td>
<td>CM_XLETEXTNI</td>
</tr>
<tr>
<td>Letter Template – Letter Extract</td>
<td>C1-LETXTALG</td>
<td>CM_XLETEXTPD</td>
</tr>
</tbody>
</table>

Following characteristic types are required for configuring algorithms for letter extraction.

<table>
<thead>
<tr>
<th>Char Code</th>
<th>Description</th>
<th>Type</th>
<th>Characteristic Entity</th>
</tr>
</thead>
<tbody>
<tr>
<td>CCID</td>
<td>Customer Contact Id</td>
<td>Foreign Key Value (FK Reference – Customer Contact)</td>
<td>Overdue Process Log</td>
</tr>
<tr>
<td>C1_AUPC</td>
<td>Auto Pay Characteristic for Account</td>
<td>Adhoc Value</td>
<td>Customer Contact</td>
</tr>
<tr>
<td>C1_CCACT</td>
<td>Account Number for Credit Card Expiry Customer Contact</td>
<td>Adhoc Value</td>
<td>Customer Contact</td>
</tr>
<tr>
<td>Char Code</td>
<td>Description</td>
<td>Type</td>
<td>Characteristic Entity</td>
</tr>
<tr>
<td>-----------</td>
<td>--------------------------------------------------</td>
<td>-------------------------------------</td>
<td>-----------------------------------------</td>
</tr>
<tr>
<td>C1-CCDT</td>
<td>Expiration Date for Credit Card Expiry Customer Contact</td>
<td>Adhoc Value (Date Type)</td>
<td>Customer Contact</td>
</tr>
<tr>
<td>C1-CCNM</td>
<td>Name on File for Credit Card Expiry Customer Contact</td>
<td>Adhoc Value</td>
<td>Customer Contact</td>
</tr>
<tr>
<td>OVDBILL</td>
<td>Bill ID</td>
<td>Foreign Key Value (FK Reference – C1-BLACT)</td>
<td><strong>Overdue Process Log</strong>&lt;br&gt;<strong>Overdue Process Collecting On</strong></td>
</tr>
</tbody>
</table>

From the above characteristics, C1_AUPC, C1-CCACT, C1-CCDT and C1-CCNM are shipped with the product.

Characteristics CCID and OVDBILL are required to be created as per the details provided in the table.

### 3.1 Online Letter Image - Implementation Details

The following table provides information about the C1_XLETOLALG algorithm type:

<table>
<thead>
<tr>
<th>Algorithm Type</th>
<th>C1_XLETOLALG</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>Wrapper Algorithm Type for Online Letter Display</td>
</tr>
<tr>
<td>Detailed Description</td>
<td>This Wrapper algorithm will internally call letter extract algorithm as per the Letter Template configuration. It will invoke external system interface to send request to external system. It will extract URL from the response and present letter online. The integration with third party application can be through batch script or web services. Sleep time needs to be configured in case of integration using batch script. When integrating using web services, XAI outbound message configuration is required. Two folders need to be created one for XML extract and other for storing the Dat files. Input File Path is the root folder in which Dat file folder and Extract folder will be created. Output file path is the full path to the folder where third party application e.g. Oracle Documaker will generate PDF files. Create this folder if it does not exist. For native mode invocation of Documaker, Input File Path, Dat and Extract folder names and Output File Path should exist in lower case. Bill View folder parameter is used to create the URL for calling a program that displays the generated PDF to the user. The bill view...</td>
</tr>
</tbody>
</table>
folder path will be relative to the “context root” of the web server. The value for this parameter should be specified as 'billView', if the default URL pattern mapping of the program is to be used.

Bat file name is the name of the bat file which will invoke application specific executable. It internally sends two parameters one for the input DAT file name and other for the expected output PDF file name.

Output Http URL is the web server address under which the ‘Output file path’ folder is configured.

<table>
<thead>
<tr>
<th>Invocation point(s)</th>
<th>Installation Options – Framework (Online Letter Image)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Program Type</td>
<td>Java</td>
</tr>
<tr>
<td>Name</td>
<td>com.splwg.ccb.domain.admin.letterTemplate.OnlineLetterDisplayAlgo</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Name</th>
<th>Required (Yes/No/Conditional)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>File Path</td>
<td>Yes</td>
<td>File path where Dat and Extract folders have been created. For Native mode invocation of Documaker, this path should be in lower case.</td>
</tr>
<tr>
<td></td>
<td>Encoding Class Name</td>
<td>No</td>
<td>This feature is not supported in this release.</td>
</tr>
<tr>
<td></td>
<td>Outbound Message Type</td>
<td>Conditional</td>
<td>Message type to be sent to the external system. This parameter is required if Invocation Mode parameter value is “WEB”</td>
</tr>
<tr>
<td></td>
<td>External System</td>
<td>Conditional</td>
<td>External system to be configured as per interface of external System. This parameter is required if Invocation Mode parameter value is “WEB”</td>
</tr>
<tr>
<td></td>
<td>Outbound Message Business Object</td>
<td>Conditional</td>
<td>Business Object created using existing MO (F1-OUTMSG) and default Application Service (F1-DFLTS).</td>
</tr>
<tr>
<td>Parameter</td>
<td>Required</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>-------------------------</td>
<td>----------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>Output File Path</td>
<td>Yes</td>
<td>Name of the file path where PDF will be generated. Ensure that this file path should be same as the billView Output Directory Path (property <code>ouaf.runtime.billView.directoryPath</code>) provided in <code>&lt;SPLBASE&gt;/etc/conf/root/WEB-INF/classes/spl.properties</code>, where <code>&lt;SPLBASE&gt;</code> is the environment variable pointing to the root directory of ORMB installation. For Native mode invocation of Documaker, this file path should exist in lower case. <strong>Note:</strong> In case the value of Output File Path has to be modified, change the value of billView Directory Path as both the values should be in sync.</td>
<td></td>
</tr>
<tr>
<td>Dat Folder Name</td>
<td>Yes</td>
<td>Used by the external system to get the path of extract files. This folder is created under the location specified in Input File Path parameter. For Native mode invocation of Documaker, this folder name should be in lower case.</td>
<td></td>
</tr>
<tr>
<td>Bat File Name</td>
<td>Conditional</td>
<td>BAT file name for converting Documaker extract XML files into PDF. This parameter is required if Invocation Mode parameter value is “NATV”.</td>
<td></td>
</tr>
<tr>
<td>Output HTTP URL</td>
<td>Yes</td>
<td>URL name where output PDF will be displayed.</td>
<td></td>
</tr>
<tr>
<td>Parameter</td>
<td>Value</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>------------------------------------------------</td>
<td>-------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>Bill View Folder</td>
<td>Yes</td>
<td>Used for creating the URL for displaying the generated PDF. Specify this value as ‘billView’ for the default behavior.</td>
<td></td>
</tr>
<tr>
<td>Sleep Time for Online Display</td>
<td>Conditional</td>
<td>Sleep time needs to be configured in case of integration using executable. This parameter is required if Invocation Mode parameter value is “NATV”</td>
<td></td>
</tr>
<tr>
<td>Configuration Folder</td>
<td>Conditional</td>
<td>The Master Resource Library name/Library Id to be used in Documaker. This parameter is required if Invocation Mode parameter value is “WEB”</td>
<td></td>
</tr>
<tr>
<td>Invocation Mode</td>
<td>No</td>
<td>This indicates the external report generation tool to be invoked. Supported values for Documaker are:</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Valid values are:</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>NATV – For calling native Documaker</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>WEB – For calling Documaker using web services.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>When not specified, value of this parameter is “NATV”. Note: This parameter is case-sensitive.</td>
<td></td>
</tr>
</tbody>
</table>

### Detailed Design

The Online Letter Image Wrapper Algorithm will be using separate Algorithm Type as C1_XLETOLALG. This Wrapper Algorithm will call specific Letter Extract Algorithm according to the Letter Template configured on Customer Contact.

### 3.2 Online Letter Image Algorithm

The following table lists the parameters defined in the sample algorithm named CM_XLETOLALG. These are sample values and can be modified as per the user configuration.

<p>| Algorithm Name | CM_XLETOLALG |</p>
<table>
<thead>
<tr>
<th>Parameters</th>
<th>Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Input File Path</td>
<td>/spl/temp/</td>
</tr>
<tr>
<td></td>
<td>Encoding Class Name</td>
<td>(Not supported in this release)</td>
</tr>
<tr>
<td></td>
<td>Outbound Message Type</td>
<td>C1-DOCMAKOM</td>
</tr>
<tr>
<td></td>
<td>External System</td>
<td>C1-DOCMAKEXTSYS</td>
</tr>
<tr>
<td></td>
<td>Outbound Message Business Object</td>
<td>C1-DOCMAKBO</td>
</tr>
<tr>
<td></td>
<td>Output File Path</td>
<td>/spl/temp/billview</td>
</tr>
<tr>
<td></td>
<td>Dat Folder Name</td>
<td>data</td>
</tr>
<tr>
<td></td>
<td>Bat File Name</td>
<td>&lt;DOCUMAKER_INSTALL_DIR&gt;/ormb/input/rmbonlineprint.sql</td>
</tr>
<tr>
<td></td>
<td>Output HTTP URL</td>
<td>http://&lt;host&gt;:&lt;port&gt;/spl/</td>
</tr>
<tr>
<td></td>
<td>Bill View Folder</td>
<td>billView</td>
</tr>
<tr>
<td></td>
<td>Sleep Time for Online Display</td>
<td>5000</td>
</tr>
<tr>
<td></td>
<td>Configuration Folder</td>
<td>ormb</td>
</tr>
<tr>
<td></td>
<td>Invocation Mode</td>
<td>WEB</td>
</tr>
</tbody>
</table>

**Note:** For more information on how to define Outbound Message Type and External System, refer to the Setup ORMB for Oracle Documaker 12.6.0 section.

### 3.3 Sample XSL for Letter Extract - Documaker

ORMB provides below XSL for generating extract XML for Documaker from raw ORMB XML for Letters:

<table>
<thead>
<tr>
<th>Type of Letter</th>
<th>XSL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Auto Pay</td>
<td>&lt;SPLEBASE&gt;/splapp/xai/schemas/samples/letterTransform/ORMB_DM_AutoPayExtract.xsl</td>
</tr>
<tr>
<td>Credit Card Expiry</td>
<td>&lt;SPLEBASE&gt;/splapp/xai/schemas/samples/letterTransform/ORMB_DM_CcExpiryExtract.xsl</td>
</tr>
<tr>
<td>Notice of Intent to Cancel</td>
<td>&lt;SPLEBASE&gt;/splapp/xai/schemas/samples/letterTransform/ORMB_DM_NoticeOfIntentExtract.xsl</td>
</tr>
<tr>
<td>Past Due</td>
<td>&lt;SPLEBASE&gt;/splapp/xai/schemas/samples/letterTransform/ORMB_DM_OverDueExtract.xsl</td>
</tr>
</tbody>
</table>

These XSLs need to be configured in the corresponding extract algorithm.
### 3.4 Letter Template – Letter Extract Implementation Details

The following table provides information about the C1-LETXTALG algorithm type:

<table>
<thead>
<tr>
<th>Algorithm Type</th>
<th>C1-LETXTALG</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>Letter Extract Algorithm for letter changes.</td>
</tr>
<tr>
<td>Detailed Description</td>
<td>Letter Extract Algorithm Type for extracting letter records from RMB for different types of Auto Pay changes, Credit Card expiry, Past due and Notice of Intent to Cancel letters and transform the message to outbound message format.</td>
</tr>
<tr>
<td>Invocation point(s)</td>
<td>Letter Template</td>
</tr>
<tr>
<td>Program Type</td>
<td>Java</td>
</tr>
<tr>
<td>Name</td>
<td>com.splwg.ccb.domain.admin.letterTemplate.LetterExtractAlgo</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Name</th>
<th>Required (Yes/No)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>External System Input String</td>
<td>No</td>
<td>Input String for External System from where it will pick up path for extract file.</td>
</tr>
<tr>
<td></td>
<td>Letter Specific Business Service</td>
<td>No</td>
<td>Name of Letter Specific Business Service.</td>
</tr>
<tr>
<td></td>
<td>Custom Business Service</td>
<td>No</td>
<td>Name of Custom Business Service for extracting custom information specific to the corresponding Letter type.</td>
</tr>
<tr>
<td></td>
<td>Name of XSL file for generating output XML</td>
<td>No</td>
<td>XSL file used for XML transformation from raw ORMB XML to Documaker XML</td>
</tr>
<tr>
<td>Parameter</td>
<td>Allow Nulls</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>------------------------------------------------</td>
<td>-------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>Extract Folder Name</td>
<td>No</td>
<td>Extract folder where the Documaker XML will get extracted after the XSLT. This folder should be created under the location specified as ‘Input File Path’ in Online Letter Image algorithm (C1_XLETOLALG). For Native mode invocation of Documaker, through Online Letter Image algorithm or via batches, this folder name should be in lower case.</td>
<td></td>
</tr>
<tr>
<td>Dat Folder Name</td>
<td>No</td>
<td>Used by the external system to get the path of extract files. This folder should be created under the location specified as ‘Input File Path’ in Online Letter Image algorithm (C1_XLETOLALG). This is the same folder specified in ‘Dat Folder Name’ parameter in Online Letter Image algorithm. For Native mode invocation of Documaker, through Online Letter Image algorithm or via batches, this folder name should be in lower case.</td>
<td></td>
</tr>
<tr>
<td>Letter Process Date Format</td>
<td>No</td>
<td>Letter Process Date Format</td>
<td></td>
</tr>
<tr>
<td>Due Date format for Overdue Bill Processing</td>
<td>No</td>
<td>Due Date format for Overdue Bill Processing</td>
<td></td>
</tr>
<tr>
<td>Effective Date for Auto Pay Change</td>
<td>No</td>
<td>Effective Date Format for Auto Pay Change</td>
<td></td>
</tr>
<tr>
<td>Event Type Code</td>
<td>No</td>
<td>Event Type Code For Overdue Processing</td>
<td></td>
</tr>
<tr>
<td>Number of Days</td>
<td>No</td>
<td>No. of days in which notice of intent to cancel letter to be sent.</td>
<td></td>
</tr>
<tr>
<td>Char Type Code for Customer Contact</td>
<td>No</td>
<td>Characteristic Type used in Overdue and AutoPay Letter.</td>
<td></td>
</tr>
<tr>
<td>Char Type Code for Over Due</td>
<td>No</td>
<td>Characteristic Type used in Overdue Letter.</td>
<td></td>
</tr>
<tr>
<td>----------------------------</td>
<td>----</td>
<td>--------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>Char Type Code for Account Number</td>
<td>No</td>
<td>Characteristic Type used in Credit Cards Expiry and AutoPay Letter.</td>
<td></td>
</tr>
<tr>
<td>Char type code for Credit Card Expiry Date</td>
<td>No</td>
<td>Characteristic Type used in Credit Cards Expiry Letter.</td>
<td></td>
</tr>
<tr>
<td>Char type code for Name on Credit Card</td>
<td>No</td>
<td>Characteristic Type used in Credit Cards Expiry Letter.</td>
<td></td>
</tr>
</tbody>
</table>

### 3.5 Letter Template – Letter Extract – (Auto Pay) Implementation Details

The following table lists the parameters defined in the sample algorithm named CM_XLETEXTAU. These are sample values and can be modified as per the user configuration.

<table>
<thead>
<tr>
<th>Algorithm Name</th>
<th>CM_XLETEXTAU</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Parameters</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Name</strong></td>
<td>Value</td>
</tr>
<tr>
<td>External System Input String</td>
<td>XML_FILE_INPUT</td>
</tr>
<tr>
<td>Letter Specific Business Service</td>
<td>C1-AutoPayLetterExt</td>
</tr>
<tr>
<td>Custom Business Service</td>
<td></td>
</tr>
<tr>
<td>Name of XSL file for generating output XML</td>
<td>&lt;SPLBASE&gt;/splapp/xai/schemas/samples/letter_transform/ORMB_DM_AutoPayExtract.xsl</td>
</tr>
<tr>
<td>Extract Folder Name</td>
<td>extract</td>
</tr>
<tr>
<td>Dat Folder Name</td>
<td>data</td>
</tr>
<tr>
<td>Letter Process Date Format</td>
<td>MMMM DD, YYYY</td>
</tr>
<tr>
<td>Due Date format for Overdue Bill Processing</td>
<td></td>
</tr>
<tr>
<td>Effective Date for Auto Pay Change</td>
<td>MM-DD-YYYY</td>
</tr>
<tr>
<td>Event Type Code</td>
<td></td>
</tr>
<tr>
<td>Number of Days</td>
<td></td>
</tr>
<tr>
<td>Char Type Code for Customer Contact</td>
<td>CCID</td>
</tr>
<tr>
<td>Char Type Code for Over Due</td>
<td></td>
</tr>
<tr>
<td>Char Type Code for Account Number</td>
<td>C1_AUPC</td>
</tr>
</tbody>
</table>
Detailed Design

The AutoPay letter extract algorithm uses common Algorithm Type (C1-LEXTALG) for AutoPay Letter.
Base Letter Service (C1-BaseLetterExt) will be called from the algorithm and Base auto pay service (C1-AutoPayLetterExt) will be passed as soft parameter.

3.6 Letter Template – Letter Extract – (Credit Card Expiry) Implementation Details

The following table lists the parameters defined in the sample algorithm named CM_XLETEXTCC. These are sample values and can be modified as per the user configuration.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>CM_XLETEXTCC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>Letter Extract Algorithm for extracting letter records from RMB for credit card expiry letters and transform the message to outbound message format</td>
</tr>
<tr>
<td>Detailed Description</td>
<td>Letter Extract</td>
</tr>
<tr>
<td>Invocation point(s)</td>
<td>Letter Template</td>
</tr>
<tr>
<td>Program Type</td>
<td>Java</td>
</tr>
<tr>
<td>Name</td>
<td>com.splwg.ccb.domain.admin.letterTemplate.LetterExtractAlgo</td>
</tr>
<tr>
<td>Parameters Name</td>
<td>Value</td>
</tr>
<tr>
<td>External System Input String</td>
<td>XML_FILE_INPUT</td>
</tr>
<tr>
<td>Letter Specific Business Service</td>
<td>C1-CcExpiryLetterExt</td>
</tr>
<tr>
<td>Custom Business Service</td>
<td></td>
</tr>
<tr>
<td>Name of XSL file for generating output XML</td>
<td><code>&lt;SPLBASE&gt;/splapp/xai/schemas/samples/letter_transform/ORMB_DM_CcExpiryExtract.xsl</code></td>
</tr>
<tr>
<td>Extract Folder Name</td>
<td>extract</td>
</tr>
<tr>
<td>Dat Folder Name</td>
<td>data</td>
</tr>
<tr>
<td>Letter Process Date Format</td>
<td>MMMM DD, YYYY</td>
</tr>
<tr>
<td>Due Date format for Overdue Bill Processing</td>
<td></td>
</tr>
<tr>
<td>Effective Date for Auto Pay Change</td>
<td></td>
</tr>
</tbody>
</table>
3.7 Letter Template – Letter Extract – (Notice of Intent to Cancel) Implementation Details

The following table lists the parameters defined in the sample algorithm named CM_XLETEXTNI. These are sample values and can be modified as per the user configuration.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>CM_XLETEXTNI</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Description</strong></td>
<td>Letter Extract Algorithm for extracting letter records from RMB for Notice of Intent to Cancel letters and transform the message to outbound message format</td>
</tr>
<tr>
<td><strong>Detailed Description</strong></td>
<td>Letter Extract</td>
</tr>
<tr>
<td><strong>Invocation point(s)</strong></td>
<td>Letter Template</td>
</tr>
<tr>
<td><strong>Program Type</strong></td>
<td>Java</td>
</tr>
<tr>
<td><strong>Name</strong></td>
<td>com.splwg.ccb.domain.admin.letterTemplate.LetterExtractAlgo</td>
</tr>
<tr>
<td><strong>Parameters</strong></td>
<td><strong>Name</strong></td>
</tr>
<tr>
<td>External System Input String</td>
<td>XML_FILE_INPUT</td>
</tr>
<tr>
<td>Letter Specific Business Service</td>
<td>C1-PastDueLetterExt</td>
</tr>
<tr>
<td>Custom Business Service</td>
<td></td>
</tr>
<tr>
<td>Name of XSL file for generating output XML</td>
<td>&lt;SPLEBASE&gt;/splapp/xai/schemas/samples/letterTransform/ORMB_DM_NoticeOfIntentExtract.xsl</td>
</tr>
</tbody>
</table>
Extract Folder Name | extract
---|---
Dat Folder Name | data
Letter Process Date Format | MMMM DD, YYYY
Due Date format for Overdue Bill Processing | MMMM DD, YYYY
Effective Date for Auto Pay Change
Event Type Code | INS-INT-CANC
Number of Days | 10
Char Type Code for Customer Contact | CCID
Char Type Code for Over Due | OVDBILL
Char Type Code for Account Number
Char type code for Credit Card Expiry Date
Char type code for Name on Credit Card

**Detailed Design**
The Notice of Intent to Cancel letter extract algorithm will use common algorithm type (C1-LETXTALG). Base Letter Service (C1-BaseLetterExt) will be called from the algorithm and Base credit card service (C1-PastDueLetterExt) will be passed as soft parameter.

### 3.8 Letter Template – Letter Extract – (Past Due) Implementation Details

The following table lists the parameters defined in the sample algorithm named CM_XLETSEXTPD. These are sample values and can be modified as per the user configuration.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>CM_XLETSEXTPD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>Letter Extract Algorithm for extracting letter records from RMB for Past Due letters and transform the message to outbound message format</td>
</tr>
<tr>
<td>Detailed Description</td>
<td>Letter Extract</td>
</tr>
<tr>
<td>Invocation point(s)</td>
<td>Letter Template</td>
</tr>
<tr>
<td>Program Type</td>
<td>Java</td>
</tr>
<tr>
<td>Name</td>
<td>com.splwg.ccb.domain.admin.letterTemplate.LetterExtractAlgo</td>
</tr>
<tr>
<td>Parameters</td>
<td>Name</td>
</tr>
<tr>
<td>------------</td>
<td>-------------------------------------------</td>
</tr>
<tr>
<td></td>
<td>External System Input String</td>
</tr>
<tr>
<td></td>
<td>Letter Specific Business Service</td>
</tr>
<tr>
<td></td>
<td>Custom Business Service</td>
</tr>
<tr>
<td></td>
<td>Name of XSL file for generating output XML</td>
</tr>
<tr>
<td></td>
<td>Extract Folder Name</td>
</tr>
<tr>
<td></td>
<td>Dat Folder Name</td>
</tr>
<tr>
<td></td>
<td>Letter Process Date Format</td>
</tr>
<tr>
<td></td>
<td>Due Date format for Overdue Bill Processing</td>
</tr>
<tr>
<td></td>
<td>Effective Date for Auto Pay Change</td>
</tr>
<tr>
<td></td>
<td>Event Type Code</td>
</tr>
<tr>
<td></td>
<td>Number of Days</td>
</tr>
<tr>
<td></td>
<td>Char Type Code for Customer Contact</td>
</tr>
<tr>
<td></td>
<td>Char Type Code for Over Due</td>
</tr>
<tr>
<td></td>
<td>Char Type Code for Account Number</td>
</tr>
<tr>
<td></td>
<td>Char type code for Credit Card Expiry Date</td>
</tr>
<tr>
<td></td>
<td>Char type code for Name on Credit Card</td>
</tr>
</tbody>
</table>

**Detailed Design**

The Past Due letter extract algorithm will use common algorithm type (C1-LEXTALG). Base Letter Service (C1-BaseLetterExt) will be called from the algorithm and Base credit card service (C1-PastDueLetterExt) will be passed as soft parameter.
3.9 Business Service for Letter Generation

The following base business services are invoked in letter generation:

(a) Base business service for extracting generic letter extract information
   - C1-BaseLetterExt

(b) For generating Auto Pay Letter Extract
   - C1-AutoPayLetterExt

(c) For generating Credit Card Expiry Letter Extract
   - C1-CcExpiryLetterExt

(d) For generating Notice of Intent to Cancel and Past Due Letter Extract
   - C1-PastDueLetterExt

A custom business service can also be configured in case any additional enrichment of the output XML needs to be done.

Please refer to ORMB Banking User Manual for more information on these business services.
3.10 Batch Letter Process Flow

To generate letters through batches, in addition to the configuration done for Letter Extract Algorithm (C1-LETXTALG), also configure the Batch Control of letter extract (LTRPRT).

In the batch control, set the value of the FILE_PATH parameter to the Dat folder created during Letter Extract algorithm (C1-LETXTALG) configuration.
For example,

```bash
FILE_PATH = /spl/temp/data/
```

For more information regarding LTRPRT batch configuration, please refer to ORMB Online Help.

**Note:** Ensure that all files names along with the complete path referred to or generated by the batch control should exist in lower case. This includes the FILE_PATH parameter passed to the batch control as well as the extract file paths mentioned in the generated dat file.
4. Setup for Oracle Documaker 12.6.0 Integration

4.1 Documaker Setup For Linux

**Step 1:** Download Oracle Documaker Standard Edition v12.6.0 for Linux from Oracle Software Delivery Cloud or My Oracle Support.

Please refer to Oracle Documaker Standard Edition Documentation for more details.

**Note:** Ensure that you follow Documaker Standard Edition System Requirements guide available at the Documentation link above before proceeding with the installation.

**Step 2:** Install Oracle Documaker Standard Edition. Verify that `<<DOCUMAKER_INSTALL_DIR>>/rel126` is created on the server.

`<<DOCUMAKER_INSTALL_DIR>>` is the root directory path of the Documaker installation.

**Note:** Ensure there are no spaces in the directory names for installation and the complete installation path along with the directory names is in lower case.

**Step 3:** Download patch number 28865465 from My Oracle Support.

**Step 4:** Extract patch file. It contains ormb.zip file.

**Step 5:** Extract ormb.zip in the `<<DOCUMAKER_INSTALL_DIR>>` such that the ormb folder is present at the same level as rel126 folder created in Step 2 above.

This will extract all sample Documaker specific templates and shell script files developed for bill and letter generation.

**Note:** For Linux, use binary FTP or similar tool to transfer the entire directory.

**Note:** Ensure that the Oracle Revenue Management and Billing userid (typically cissys) has execute, read, write permissions to all files in the `<<DOCUMAKER_INSTALL_DIR>>/ormb` and `<<DOCUMAKER_INSTALL_DIR>>/rel126` folders.

### Sample Directory Structure

<table>
<thead>
<tr>
<th>Directory</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;&lt;DOCUMAKER_INSTALL_DIR&gt;&gt;/rel126</code></td>
<td>Documaker Binaries</td>
</tr>
<tr>
<td><code>&lt;&lt;DOCUMAKER_INSTALL_DIR&gt;&gt;/ormb</code></td>
<td>ORMB templates and driver programs</td>
</tr>
</tbody>
</table>
Follow the steps below to configure Documaker in Batch mode and Online mode.

### 4.1.1 Oracle Documaker Configuration for Batch Mode

In Batch mode, Documaker can be invoked only via native call.

**Step 1:** Modify the `fsisys.ini` file in `<DOCUMAKER_INSTALL_DIR>/ormb` folder. Replace `<<<###Absolute Path###>>` with `<DOCUMAKER_INSTALL_DIR>/ormb`.

**Step 2:** Modify `rmbbatchprint.sh` script in `<DOCUMAKER_INSTALL_DIR>/ormb/input` and configure `LD_LIBRARY_PATH` and `DOC1HOSTDIR` environment variables as per the Documaker setup.

For example:

```bash
export LD_LIBRARY_PATH=${JAVA_HOME}/jre/lib/i386/server:$LD_LIBRARY_PATH
export DOC1HOSTDIR=/spl/documaker_12_6
```

where `<JAVA_HOME>`: path of JDK referenced by Documaker,

```
DOC1HOSTDIR: root directory path of Documaker installation. It is same as
<<<DOCUMAKER_INSTALL_DIR>>>
```

**Step 3:** To generate Bill/Letter PDF files, execute `rmbbatchprint.sh` script using the following command:

```
<<<DOCUMAKER_INSTALL_DIR>>>/ormb/input/rmbbatchprint.sh <DATFILE> <OUTPUTDIR>
```

where `<DATFILE>`: the dat file generated by ORMB bill/letter extract batches,

`<OUTPUTDIR>`: directory where the Bill/Letter PDF files should be generated

For example,

```
<<<DOCUMAKER_INSTALL_DIR>>>/ormb/input/rmbbatchprint.sh
/spl/temp/data/843792232946 /spl/temp/billview
```

The above command will generate PDF files from the extract files specified in

```
/spl/temp/data/843792232946 dat file generated by bill/letter extract batches
```

and store the PDF files in

```
/spl/temp/billview folder.
```

**Note:** All files names along with the complete path with directory names referred to by `rmbbatchprint.sh` script should exist in lower case. This includes .dat file path and Output Directory path arguments passed to the script as well as the extract file paths mentioned in the input dat file.
4.1.2 Oracle Documaker Configuration for Online Mode

In Online mode, Documaker can be invoked through native call or via web services.

Native Mode Invocation:

**Step 1:** Modify the fsisys.ini file in `<DOCUMAKER_INSTALL_DIR>/ormb` folder. Replace `<<<Absolute Path>>>` with `<DOCUMAKER_INSTALL_DIR>/ormb`.

**Step 2:** Modify `rmbonlineprint.sh` script in `<DOCUMAKER_INSTALL_DIR>/ormb/input` and configure `LD_LIBRARY_PATH` and `DOC1HOSTDIR` environment variables as per the Documaker setup.

For example:

```bash
export LD_LIBRARY_PATH=<JAVA_HOME>/jre/lib/i386/server:$LD_LIBRARY_PATH
export DOC1HOSTDIR=/spl/documaker_12_6
```

where `<JAVA_HOME>`: path of JDK referenced by Documaker,

```
DOC1HOSTDIR: root directory path of Documaker installation. It is same as `<DOCUMAKER_INSTALL_DIR>`
```

Web Services Invocation:

**Note:** Enterprise Web Processing Services (EWPS) ‘doPublish’ request is used for Documaker Web Services invocation. This integration is only supported for Online mode.

For more details regarding setup and configuration of EWPS, please refer to Documaker Standard Edition Documentation.

**Step 1:** Ensure that Docupresentment is installed as part of the Documaker setup.

Docupresentment provides Enterprise Web Processing Services (EWPS) war file to invoke Documaker via web services.

**Step 2:** Install and configure EWPS.

As part of EWPS setup, add ORMB configuration with following details in dap.ini file of Docupresentment:

- **workspace name:** `ormb` (The workspace name is same as the one specified for ‘Configuration Folder’ parameter of Online Bill Display (C1_ONLNBILL)/Online Letter Image (C1_XLETOLALG) algorithm of ORMB.

  This configuration has to be specified as [Config:workspace_name ] and should be added in the Configurations control group in dap.ini file.)

- **INI file:** `<DOCUMAKER_INSTALL_DIR>/ormb/fsisys.ini`

  `<DOCUMAKER_INSTALL_DIR>/ormb/fsiuser_online_web.ini`
For example,

```
[ Config:ormb ]
    INIFile = <<DOCUMAKER_INSTALL_DIR>>/ormb/fsisys.ini
    INIFile = <<DOCUMAKER_INSTALL_DIR>>/ormb/fsiuser_online_web.ini

[ Configurations ]
    Config = ormb
```

For more details regarding EWPS and Docupresentment configuration, refer to [Introduction to Enterprise Web Processing Services](#) document.

**Step 3:** Modify the fsisys.ini file in `<<DOCUMAKER_INSTALL_DIR>>/ormb` folder. Replace `<<###Absolute Path###>>` with `<<DOCUMAKER_INSTALL_DIR>>/ormb`.

**Step 4:** Modify the fsiuser_online_web.ini file in `<<DOCUMAKER_INSTALL_DIR>>/ormb` folder. Replace `<<###Absolute Path###>>` with `<<DOCUMAKER_INSTALL_DIR>>/ormb`. 
4.2 Setup ORMB for Oracle Documaker

**Note:** The steps mentioned in this section are applicable only for Oracle Documaker integration via EWPS.

To communicate with an external system, you need to configure outbound messages in Oracle Revenue Management and Billing. To configure outbound messages, you need to do the following:

- Define the Outbound Message Type
- Define the XAI Sender
- Define the External System

4.2.1 Defining an Outbound Message Type

To define an outbound message type (for example, C1-DOCMAKOM):

1. Login to Oracle Revenue Management and Billing.
2. Click the **Menu** link in the **Actions/Navigation** area. A list appears.
3. Select the **Admin Menu** option from the list.
4. From the **Admin Menu**, select **O** and then click the **Add** icon corresponding to the **Outbound Message Type** menu option. The **Outbound Message Type** screen appears. It contains the following fields:

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
<th>Mandatory (Yes or No)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Outbound Message Type</td>
<td>Used to specify the outbound message type.</td>
<td>Yes</td>
</tr>
<tr>
<td>Description</td>
<td>Used to specify the description for the outbound message type.</td>
<td>Yes</td>
</tr>
<tr>
<td>Detailed Description</td>
<td>Used to specify additional information about the outbound message type.</td>
<td>No</td>
</tr>
<tr>
<td>Business Object</td>
<td>Used to indicate the business object that defines business rules and schema for the outbound message type.</td>
<td>Yes</td>
</tr>
<tr>
<td>Priority</td>
<td>Used to indicate the relative priority for processing outbound messages of this type.</td>
<td>Yes</td>
</tr>
</tbody>
</table>

1. The business object is created using existing MO (F1-OUTMSG) and default Application Service (F1-DFLTS). Sample business object C1-DOCMAKBO is shipped with the product which can be specified in the Business Object field above.
2. Enter the other required details.
3. Click **Save** in the **Actions/Navigation** area. The outbound message type is defined.
4.2.2 Defining an XAI Sender

When messages are routed to an external system via XAI, each message must be associated with an XAI Sender, which informs the system how to send the message. XAI senders are responsible for defining outbound message destinations and for responding to the XAI executer.

To define an XAI sender (for example, C1-DOCMAKXAI):

1. Login to Oracle Revenue Management and Billing.
2. Click the Menu link in the Actions/Navigation area. A list appears.
3. Select the Admin Menu option from the list.
4. From the Admin Menu, select M and then click the Add icon corresponding to the Message Sender menu option. The Message Sender screen appears.
5. Enter the name of the Message sender in the respective field.
6. Enter the description for the Message sender in the respective field.
7. Select the Real-time sender from the Invocation Type list.
8. Enter RTHTTPSNDR in the Message Class field.
9. Select the Active check box to indicate that the Message sender is active.
10. Select the UTF-8 message encoding from the MSG Encoding list.
11. Click the Context tab. The Context tab appears.
12. Select HTTP Method (POST/GET) from the Context Type list. Set Context Value as POST.
13. Select HTTP Transport Method from the Context Type list. Set Context Value as sendReceive.
14. Select HTTP URL 1 from the Context Type list.
15. Enter the address of WSDL on the application server where Oracle Documaker is installed in the Context Value field.
   For example, enter http://<host>:<port>/ewps-axis2/services/DocumentService?wsdl
16. Click Save in the Actions/Navigation area. The XAI sender is defined.

4.2.3 Defining an External System

Once you define the outbound message type and XAI sender, you need to define an external system and determine the types of outbound messages that will be sent to the external system in real-time.

To define an external system (for example, C1-DOCMAKEXTSYS):

1. Login to Oracle Revenue Management and Billing.
2. Click the Menu link in the Actions/Navigation area. A list appears.
3. Select the Admin Menu option from the list.
4. From the Admin Menu, select E and then click the Add icon corresponding to the External System menu option. The External System screen appears.
5. Enter the name of the external system in the respective field.
6. Enter the description for the external system in the respective field.
7. Select the **Outbound Message Types** check box.
8. Enter the type of outbound messages you want to send to the external system in the **Outbound Message Type** field.
9. Select the **Real-time** option from the **Processing Method** list.
10. Enter the name of the XAI sender through which you want to send the outbound message in the **Message Sender** field.
11. Select the **OUAF** option from the **Date/Time Format** list.
12. Click **Save** in the **Actions/Navigation** area. The external system is defined.