

Integrating WebCenter Content

with Documaker Enterprise Edition

A Configuration Guide

Andy Little

Technical Director

Oracle Financial Services Global Business Unit

Rev 1.0, 22 Sep 2017

© 2017

Integration Methods

WebCenter Integration design depends on specific business requirements, and what your application can support with respect to integration models. WCC exposes its services over a wide variety of protocols (web services, JSP, Java API, RIDC, EJB, J2EE, CORBA, RMI, IIOP, ODMA, SOAP, WebDAV, and COM).

Available services are listed [here](#) (there are a lot of them!) The most common integration method is persistent URLs. Another is SOAP. But before you worry about the integration *method* you need to know what you're integrating. More specifically, you need to understand your functional requirements and then map those to exposed WCC services. You can access all services through any of the available APIs. So once you have the requirements mapped to services, you can determine the non-functional requirements (e.g. security, integration methods, etc) and then it comes down to performing the integration. If you want to see SOAP WSDLs, You can login to the CS as an administrator and go to Administration > SOAP WSDLs to see the available service endpoints and download WSDL files for each service. Note that SOAP requests use WSS for authentication.

Example Pattern: Search/Retrieve

A typical integration pattern is to search for a document by some metadata criteria, then retrieve a selected document. With this pattern, we have several options. We can link directly to the document, or we can use a service to return the document by ID. Persistent URLs take this form: `http://<SERVER>:<PORT>/cs/idcplg?IdcService=<SERVICE_NAME>&PARMNAME=PARMVALUE`

Search

This method is used to locate documents given a set of one or more search criteria.

Persistent URL

http://wcc:16200/cs/idcplg?IdcService=GET_SEARCH_RESULTS&QueryText=dDocTitle<Substring>%60Arms%60

SOAP¹ Request

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:sear="http://www.stellent.com/Search/">
  <soapenv:Header/>
  <soapenv:Body>
    <sear:QuickSearch>
      <sear:queryText>
        dDocTitle &lt;Substring&gt; `Arms`
      </sear:queryText>
    </sear:QuickSearch>
  </soapenv:Body>
</soapenv:Envelope>
```

SOAP Results

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <idc:QuickSearchResponse xmlns:idc="http://www.stellent.com/Search/">
      <idc:QuickSearchResult>
        <idc:SearchResults>
          <idc:dID>2</idc:dID>
          <idc:dRevisionID>1</idc:dRevisionID>
          <idc:dDocName>19216813916200000002</idc:dDocName>
          <idc:dDocTitle>Hogwart's Coat of Arms</idc:dDocTitle>
          <idc:dDocType>DigitalMedia</idc:dDocType>
          <idc:dDocAuthor>weblogic</idc:dDocAuthor>
          <idc:dSecurityGroup>Public</idc:dSecurityGroup>
          <idc:dDocAccount/>
          <idc:dExtension>png</idc:dExtension>
          <idc:dWebExtension>png</idc:dWebExtension>
          <idc:dRevLabel>1</idc:dRevLabel>
          <idc:dInDate>9/16/17 8:00 AM</idc:dInDate>
          <idc:dOutDate/>
          <idc:dFormat>image/png</idc:dFormat>
          <idc:dOriginalName>Hogwarts_coat_of_arms_colored_with_shading.svg.png</idc:dOriginalName>
          <idc:url>/cs/groups/public/documents/digitalmedia/mjaw/mdaw/~edisp/
19216813916200000002.png</idc:url>
          <idc:dGif/>
          <idc:webFileSize>0</idc:webFileSize>
          <idc:vaultFileSize>1256024</idc:vaultFileSize>
```

¹ SOAP requests must include authentication headers, not shown here.

```
<idc:alternateFileSize>0</idc:alternateFileSize>
<idc:alternateFormat/>
<idc:dPublishType/>
<idc:dRendition1>D</idc:dRendition1>
<idc:dRendition2/>
<idc:CustomDocMetaData>
  <idc:property>
    <idc:name>xComments</idc:name>
    <idc:value/>
  </idc:property>
  <idc:property>
    <idc:name>xExternalDataSet</idc:name>
    <idc:value/>
  </idc:property>
  <idc:property>
    <idc:name>xIdcProfile</idc:name>
    <idc:value/>
  </idc:property>
  <idc:property>
    <idc:name>xTemplateType</idc:name>
    <idc:value/>
  </idc:property>
  <idc:property>
    <idc:name>xAnnotationDetails</idc:name>
    <idc:value>0</idc:value>
  </idc:property>
  <idc:property>
    <idc:name>xIsACLReadOnlyOnUI</idc:name>
    <idc:value>0</idc:value>
  </idc:property>
  <idc:property>
    <idc:name>xLibraryGUID</idc:name>
    <idc:value/>
  </idc:property>
  <idc:property>
    <idc:name>xPartitionId</idc:name>
    <idc:value/>
  </idc:property>
  <idc:property>
    <idc:name>xWebFlag</idc:name>
    <idc:value/>
  </idc:property>
  <idc:property>
    <idc:name>xStorageRule</idc:name>
    <idc:value>DispByContentId</idc:value>
  </idc:property>
</idc:CustomDocMetaData>
</idc:SearchResults>
<idc:SearchInfo>
  <idc:startRow>1</idc:startRow>
```

```

        <idc:endRow>1</idc:endRow>
        <idc:pageNumber>1</idc:pageNumber>
        <idc:numPages>1</idc:numPages>
        <idc:totalRows>1</idc:totalRows>
        <idc:totalDocsProcessed>1</idc:totalDocsProcessed>
    </idc:SearchInfo>
    <idc:NavigationPages>
        <idc:headerPageNumber>1</idc:headerPageNumber>
        <idc:pageReference>1</idc:pageReference>
        <idc:pageNumber>1</idc:pageNumber>
        <idc:startRow>1</idc:startRow>
        <idc:endRow>1</idc:endRow>
    </idc:NavigationPages>
    <idc:StatusInfo>
        <idc:statusCode>0</idc:statusCode>
        <idc:statusMessage>You are logged in as 'weblogic'.</idc:statusMessage>
    </idc:StatusInfo>
</idc:QuickSearchResult>
</idc:QuickSearchResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

Retrieve

This method uses the document ID in <idc:dID>. Alternately you could directly reference the document's URL as shown in the <idc:url> node. If the document ID is not known, you can also use the document name along with a revision specification — this method is the most likely to be used since the document name may be known by external systems without any knowledge of the internal document ID set by WebCenter. Review the GET_FILE documentation [here](#).

Persistent URL

This example shows the file access using document ID:

http://wcc:16200/cs/idcplg?IdcService=GET_FILE&dID=2

This example shows the file access using document name and revision specification.

http://wcc:16200/cs/idcplg?IdcService=GET_FILE&dDocName=123456789&RevisionSelectionMethod=Latest

SOAP² Request

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:get="http://
www.stellent.com/GetFile/">
    <soapenv:Header/>
    <soapenv:Body>

```

² SOAP requests must include authentication headers, not shown here.

```

    <get:GetFileByID>
      <get:dID>2</get:dID>
    </get:GetFileByID>
  </soapenv:Body>
</soapenv:Envelope>

```

SOAP Results

```

<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <idc:GetFileByIDResponse xmlns:idc="http://www.stellent.com/GetFile/">
      <idc:GetFileByIDResult>
        <idc:FileInfo>
          <idc:dDocName>19216813916200000002</idc:dDocName>
          <idc:dDocTitle>Hogwart's Coat of Arms</idc:dDocTitle>
          <idc:dDocType>DigitalMedia</idc:dDocType>
          <idc:dDocAuthor>weblogic</idc:dDocAuthor>
          <idc:dSecurityGroup>Public</idc:dSecurityGroup>
          <idc:dDocAccount/>
          <idc:dID>2</idc:dID>
          <idc:dRevClassID>2</idc:dRevClassID>
          <idc:dRevisionID>1</idc:dRevisionID>
          <idc:dRevLabel>1</idc:dRevLabel>
          <idc:dIsCheckedOut>0</idc:dIsCheckedOut>
          <idc:dCheckoutUser/>
          <idc:dCreateDate>9/16/17 8:01 AM</idc:dCreateDate>
          <idc:dInDate>9/16/17 8:00 AM</idc:dInDate>
          <idc:dOutDate/>
          <idc:dStatus>RELEASED</idc:dStatus>
          <idc:dReleaseState>Y</idc:dReleaseState>
          <idc:dFlag1/>
          <idc:dWebExtension>png</idc:dWebExtension>
          <idc:dProcessingState>Y</idc:dProcessingState>
          <idc:dMessage/>
          <idc:dReleaseDate>9/16/17 8:01 AM</idc:dReleaseDate>
          <idc:dRendition1>D</idc:dRendition1>
          <idc:dRendition2/>
          <idc:dIndexerState/>
          <idc:dPublishType/>
          <idc:dPublishState/>
          <idc:dDocID>3</idc:dDocID>
          <idc:dIsPrimary>1</idc:dIsPrimary>
          <idc:dIsWebFormat>0</idc:dIsWebFormat>
          <idc:dLocation/>
          <idc:dOriginalName>Hogwarts_coat_of_arms_colored_with_shading.svg.png</idc:dOriginalName>
          <idc:dFormat>image/png</idc:dFormat>
          <idc:dExtension>png</idc:dExtension>
          <idc:dFileSize>1256024</idc:dFileSize>
          <idc:CustomDocMetaData>
            <idc:property>
              <idc:name>xComments</idc:name>

```

```

        <idc:value/>
    </idc:property>
    <idc:property>
        <idc:name>xExternalDataSet</idc:name>
        <idc:value/>
    </idc:property>
    <idc:property>
        <idc:name>xIdcProfile</idc:name>
        <idc:value/>
    </idc:property>
    <idc:property>
        <idc:name>xTemplateType</idc:name>
        <idc:value/>
    </idc:property>
    <idc:property>
        <idc:name>xAnnotationDetails</idc:name>
        <idc:value>0</idc:value>
    </idc:property>
    <idc:property>
        <idc:name>xIsACLReadOnlyOnUI</idc:name>
        <idc:value>0</idc:value>
    </idc:property>
    <idc:property>
        <idc:name>xLibraryGUID</idc:name>
        <idc:value/>
    </idc:property>
    <idc:property>
        <idc:name>xPartitionId</idc:name>
        <idc:value/>
    </idc:property>
    <idc:property>
        <idc:name>xWebFlag</idc:name>
        <idc:value/>
    </idc:property>
    <idc:property>
        <idc:name>xStorageRule</idc:name>
        <idc:value>DispByContentId</idc:value>
    </idc:property>
</idc:CustomDocMetaData>
</idc:FileInfo>
<idc:downloadFile>
    <idc:fileName>Hogwarts_coat_of_arms_colored_with_shading.svg.png</idc:fileName>
    <idc:fileContent>XXXXXFileContentIsHereXXXXXXXX</idc:downloadFile>
</idc:StatusInfo>
    <idc:statusCode>0</idc:statusCode>
    <idc:statusMessage>You are logged in as 'weblogic'.</idc:statusMessage>
</idc:StatusInfo>
</idc:GetFileByIDResult>
</idc:GetFileByIDResponse>
</SOAP-ENV:Body>

```

Integrating Documaker with WebCenter Content

Documaker integrates with WebCenter Content for two reasons: pushing published documents into a repository, and pulling documents for inclusion in a document transaction. The latter happens only when Documaker Interactive, part of Oracle Documaker Enterprise Edition (ODEE), is used. The former happens as part of the Documaker Connector when using Oracle Documaker Standard Edition (ODSE) or part of the Archiver worker process when using ODEE. This guide will focus on the specifics of ODEE integration into WebCenter Content. Some information may be relevant to ODSE integration, but this is covered in detail in the [product documentation](#).

In order to publish documents to WebCenter Content (WCC), Documaker needs only a bit of configuration information, however this basic configuration is usually not sufficient for a full-fledged implementation because most companies need to be able to find documents based on more than document title and date. Therefore, it becomes necessary to define additional data elements (“metadata”) that describe the document in WCC, and it is also necessary to source that data in order to push it into WCC. The source of that data is a combination of:

- source systems, which provide initial transactional data to Documaker;
- user-entered data, if Documaker Interactive is used; and
- data that is computed in Documaker processes.

1. Plan WebCenter Configuration - review planning [documentation](#).
 1. Determine organizational structure (folders)
 2. Determine any document classes or profiles that may be needed
 3. Identify any global metadata elements that are needed
 4. Identify any document class-specific metadata elements that are needed
2. Configure WebCenter Content - the two most common activities are to organize documents into folders according to metadata values, and to define additional metadata values.
 1. To define metadata values first review the [documentation](#), then follow these general steps.
 1. Open WebCenter Content Server administrator.
 2. Select Administration > Admin Applets > Configuration Manager. Run the file that downloads.

3. Click the Add button on the Information Fields tab to create a new metadata element, give the element a name, and click Ok.
4. Define the attributes of the element, such as type, value, requirements, and more, then click Ok.
5. Repeat steps 3 and 4 as necessary, then click Update Database Design. Review the changes and click Ok.
2. To organize documents into folders according to metadata values, first review the documentation, then follow these general steps. You will be creating a *query folder*, also known as a *saved search*.
 1. First, determine where the saved search folder will be located. Ideally you will create an Enterprise Library and/or a subfolder within that library to contain the folder.
 2. Open WCC User Interface and login.
 3. Click Browse. To create a Library, click the Create Library button.
 4. Give the Enterprise Library a unique name, description, and assign a security group³, then click Finish. The library will be created.
 5. To create a search folder, click the Search button, then click the down arrow in the search bar, and choose “Standard Search”.
 6. Input the appropriate search criteria that will define your folder - e.g. choose metadata elements and values with appropriate comparison operators such as *starts*, *matches*, *contains* or *does not contain*. Click Search.
 7. If the results look correct, click the down arrow under Searching Documents and select Create Saved Search.
 8. Give the Saved Search a unique name and description, then select the appropriate library(ies) to house the saved search. Click Save. The saved search will now be available under the libraries selected.
 9. To test the saved search, upload a document and configure the document criteria to match the saved search. In WCC User Interface, click Browse and then open the library that contains the saved search, then click on the saved search. The document(s) matching the saved search criteria appear.
3. Configure Documaker for publishing documents into WebCenter Content
 1. Open Documaker Administrator and login.
 2. Select System 1 > Assembly Line 1 > Archiver and click Configure.
 3. Select DESTINATION > WebCenterContent > Configuration. Set the following properties:

³ Remember: the “Public” security group will be accessible without logging in, so it is advisable to use the “Secure” group.

- destination.name = oracle.documaker.ecmconnector.ucmdestination.UCMDestination
- destination.ucm.connectionstring_0 = idc://<webcenter_cs>:4444
- destination.ucm.importmethod = 0
- destination.ucm.password = <credential password>
- destination.ucm.username = <credential>
- DocumentURL⁴ = http://<webcenter_cs>:16200/cs/groups/secure/documents/document/
- UCM.retry.count = 0

4. Select DESTINATION > WebCenterContent > Defaults. Here you can set any default metadata values for all documents archived into WCC. The property name must be a valid WCC data element. The following are created for you, and you can change these values.

- dDocAccount = docfactory
- dDocAuthor = documentfactory
- dDocType = Document
- default =
- dSecurityGroup = Secure⁵

5. Select DESTINATION > WebCenterContent > Mappings. Here you can set any metadata elements that will be mapped to transaction or document details using Freemarker notation. When you add metadata elements in WebCenter, you will map them here. You can use columns from any of the ODEE processing tables: JOBS, TRNS, BCHS, RCPS, and PUBS. A listing of some of the fields is in the ODEE 12.6 Administration Guide on page 543. To map new metadata values⁶, click the + and enter the element name and mapping. Save your changes - you will need to restart ODDF for the changes to take effect. You should not delete any of the mappings that come with the system on installation, but you can change them per your requirements. **Note:** metadata elements created in WCC will not display an “x” in front of the field name, but in ODEE you will need to enter the element with an x as shown below. Also, be aware of any field data type and length limits that you set in WCC and ensure that the values you map into those fields will fit; failure to do so may cause documents to fail to import.

- dDocType = {PUBS.UNIQUE_ID}
- default =
- primaryFile = {PUBS.PUBUNIQUE_ID}.{PUBS.PUBPRTEXT}
- primaryFileExt = {PUBS.PUBPRTEXT}

⁴ This setting may not be used currently.

⁵ This controls the disposition of the document in WCC. The default options are Public and Secure. You should never use Public for any documents that require authorization to view. Refer to the [system documentation](#) for additional details.

⁶ ODEE provides many customer-use fields for mapping metadata. These fields can be sourced from configuration, extract data, web service request, computed or user-entered, or Documaker-generated. Refer to Documaker documentation on specifics.

- Example: MyNewElement = {TRNS.TRNCUSSTR001}

References

[Service guide for WCC](#)

[Simple Search Example](#)

[Introduction to WebCenter Content Services](#)

[Configuring WebCenter Content Web Services for Integration](#)

[Understanding Libraries](#)

[Understanding Folders](#)