

Importing Data into Oracle ERP Cloud using Oracle Integration Cloud

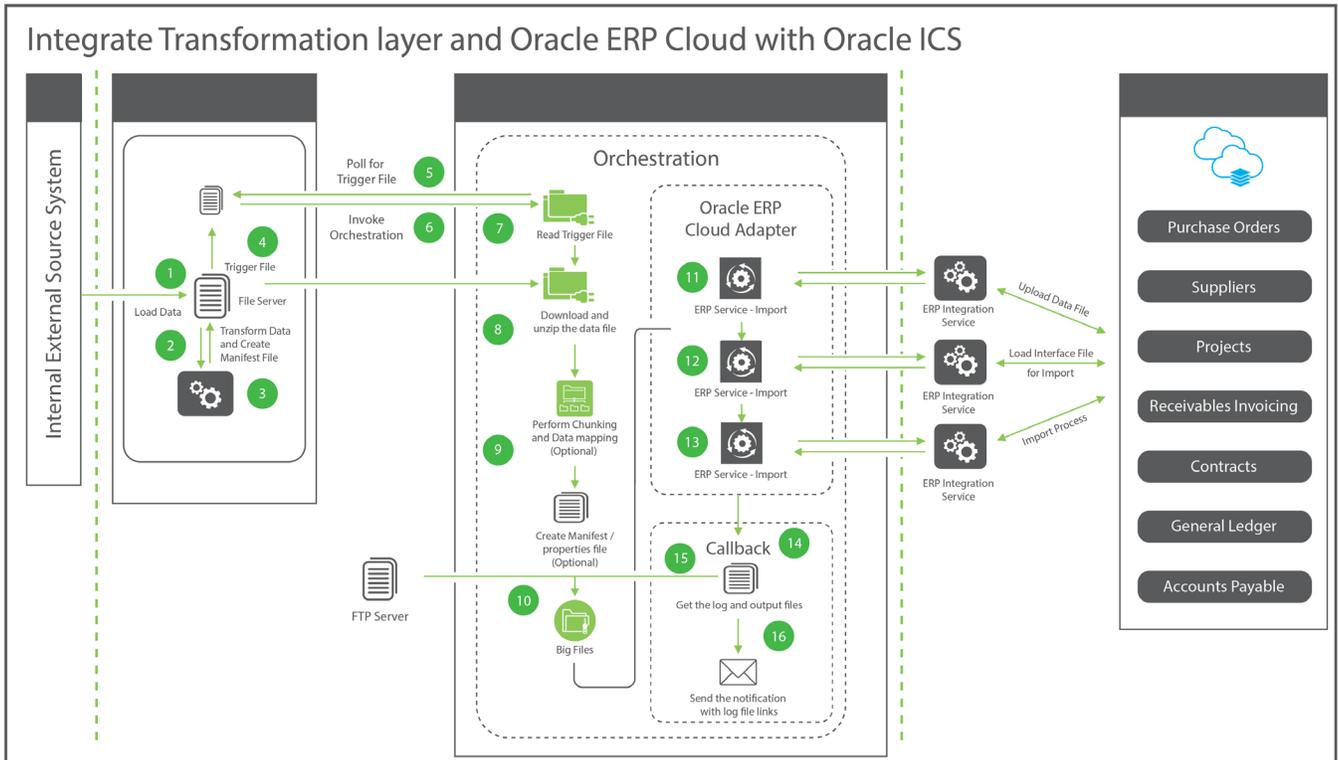
keste

Introduction

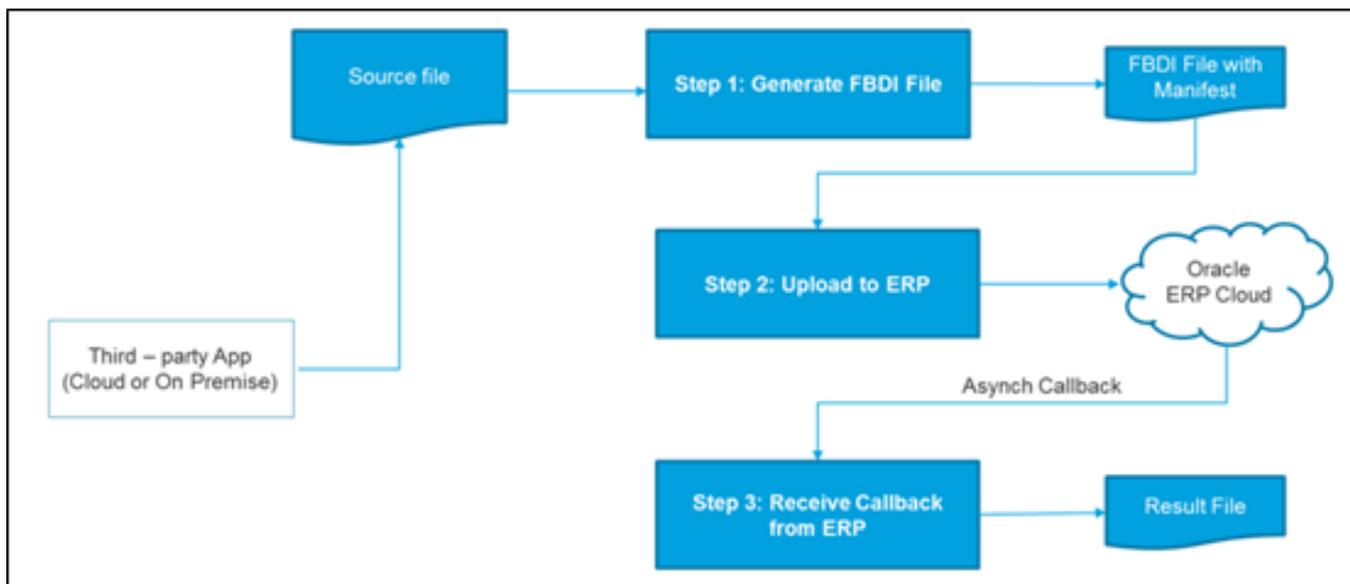
Oracle Integration Cloud (OIC) can be leveraged to import data from external feeds (published in SFTP) to Oracle ERP Cloud. OIC simplifies how you build integrations in the cloud, letting you connect securely to applications and services both in the cloud and on-premises. OIC exposes a browser-based user interface through which the integration is first designed, then activated and managed. It also provides connectors/adapters to easily interact with a number of popular SaaS applications. One such connector that OIC provides is the Oracle ERP Cloud Adapter which is used to connect to Oracle Cloud ERP. Oracle Cloud ERP (ERP) offers the most complete and integrated Enterprise Resource Management solutions for operations and performance management.

Solution Design

The diagram below describes the entire flow involving the various steps while importing the data into ERP Cloud using OIC.



The entire solution is broken into 3 steps as shown in the diagram below.



Step 1: Generate FBDI File

The first step is to generate File Based Data Import (FBDI) file for ERP's consumption. Most of the bulk integration with ERP is through FBDI File. This is a flat file which looks as below:

```
MEM,1,2016/03/01,Balance Transfer,Remote Transfer,USD,2016/03/01,A,01,000,1212,0000,000,,,,,,,,,,,,,99.00,,,Team A Team Batch5,,,Orange Journals,,,,,,,,,,,,,123,,,,,,,,,,,,,END
MEM,1,2016/03/01,Balance Transfer,Remote Transfer,USD,2016/03/01,A,01,000,4111,0000,000,,,,,,,,,,,,,99.00,,,Team A Team Batch5,,,Orange Journals,,,,,,,,,,,,,123,,,,,,,,,,,,,END
MEM,1,2016/03/01,Balance Transfer,Remote Transfer,USD,2016/03/01,A,01,000,1212,0000,000,,,,,,,,,,,,,99.00,,,Team A Team Batch5,,,Orange Journals,,,,,,,,,,,,,123,,,,,,,,,,,,,END
MEM,1,2016/03/01,Balance Transfer,Remote Transfer,USD,2016/03/01,A,01,000,4111,0000,000,,,,,,,,,,,,,99.00,,,Team A Team Batch5,,,Orange Journals,,,,,,,,,,,,,123,,,,,,,,,,,,,END
MEM,1,2016/03/01,Balance Transfer,Remote Transfer,USD,2016/03/01,A,01,000,1212,0000,000,,,,,,,,,,,,,99.00,,,Team A Team Batch5,,,Orange Journals,,,,,,,,,,,,,123,,,,,,,,,,,,,END
MEM,1,2016/03/01,Balance Transfer,Remote Transfer,USD,2016/03/01,A,01,000,4111,0000,000,,,,,,,,,,,,,99.00,,,Team A Team Batch5,,,Orange Journals,,,,,,,,,,,,,123,,,,,,,,,,,,,END
MEM,1,2016/03/01,Balance Transfer,Remote Transfer,USD,2016/03/01,A,01,000,1212,0000,000,,,,,,,,,,,,,99.00,,,Team A Team Batch5,,,Orange Journals,,,,,,,,,,,,,123,,,,,,,,,,,,,END
MEM,1,2016/03/01,Balance Transfer,Remote Transfer,USD,2016/03/01,A,01,000,4111,0000,000,,,,,,,,,,,,,99.00,,,Team A Team Batch5,,,Orange Journals,,,,,,,,,,,,,123,,,,,,,,,,,,,END
MEM,1,2016/03/01,Balance Transfer,Remote Transfer,USD,2016/03/01,A,01,000,1212,0000,000,,,,,,,,,,,,,99.00,,,Team A Team Batch5,,,Orange Journals,,,,,,,,,,,,,123,,,,,,,,,,,,,END
MEM,1,2016/03/01,Balance Transfer,Remote Transfer,USD,2016/03/01,A,01,000,4111,0000,000,,,,,,,,,,,,,99.00,,,Team A Team Batch5,,,Orange Journals,,,,,,,,,,,,,123,,,,,,,,,,,,,END
MEM,1,2016/03/01,Balance Transfer,Remote Transfer,USD,2016/03/01,A,01,000,1212,0000,000,,,,,,,,,,,,,99.00,,,Team A Team Batch5,,,Orange Journals,,,,,,,,,,,,,123,,,,,,,,,,,,,END
MEM,1,2016/03/01,Balance Transfer,Remote Transfer,USD,2016/03/01,A,01,000,4111,0000,000,,,,,,,,,,,,,99.00,,,Team A Team Batch5,,,Orange Journals,,,,,,,,,,,,,123,,,,,,,,,,,,,END
MEM,1,2016/03/01,Balance Transfer,Remote Transfer,USD,2016/03/01,A,01,000,1212,0000,000,,,,,,,,,,,,,99.00,,,Team A Team Batch5,,,Orange Journals,,,,,,,,,,,,,123,,,,,,,,,,,,,END
MEM,1,2016/03/01,Balance Transfer,Remote Transfer,USD,2016/03/01,A,01,000,4111,0000,000,,,,,,,,,,,,,99.00,,,Team A Team Batch5,,,Orange Journals,,,,,,,,,,,,,123,,,,,,,,,,,,,END
MEM,1,2016/03/01,Balance Transfer,Remote Transfer,USD,2016/03/01,A,01,000,1212,0000,000,,,,,,,,,,,,,99.00,,,Team A Team Batch5,,,Orange Journals,,,,,,,,,,,,,123,,,,,,,,,,,,,END
```

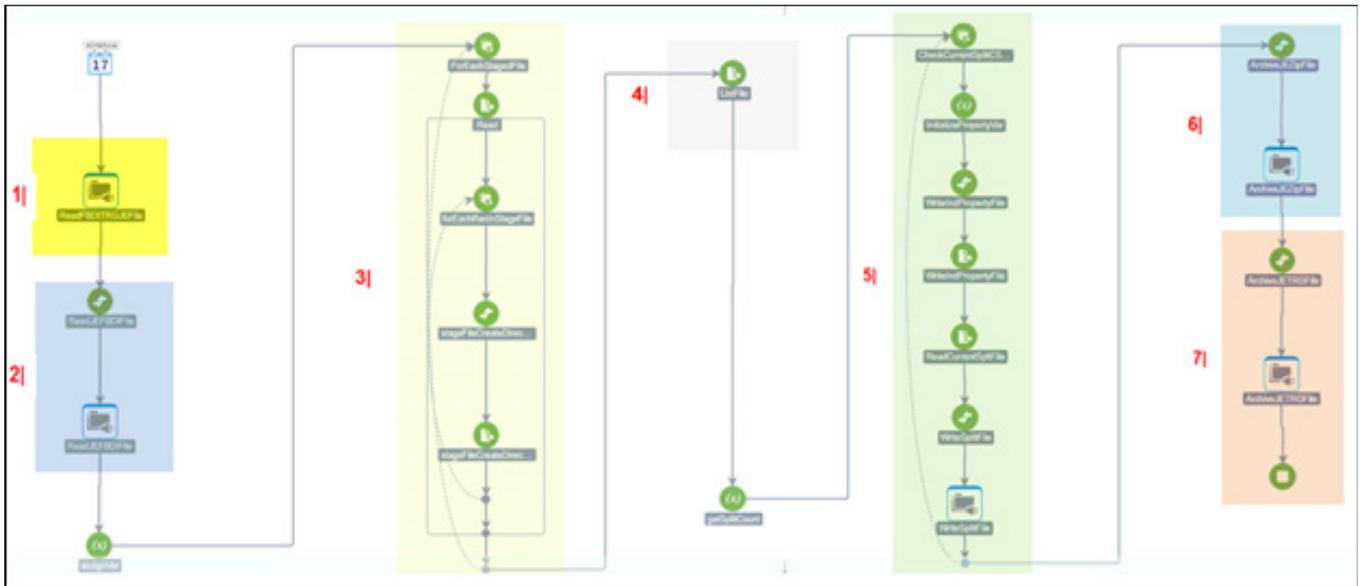
The source application's data file may not be in the FBDI format always. In this case the source application data format needs to be converted to FBDI format. This can be achieved by applying cross - references master data values from Oracle ERP cloud in a custom layer by either making the changes in the source system to burst out the data in FBDI template or by using procedures in a database layer and applying transformation to get FBDI formatted files.

OIC mapper functionality (XSLT) is then used to map from source to the destination format. However, there may be cases where the source generated file may be too large, for example, 500 MB. Anyone who has experience dealing with on premise integration products like Fusion Middleware (FMW) can tell that reading large file in-memory is not a good approach. This is because there are chances of running out of memory reading the large file in-memory.

Since OIC is a PaaS offering, there has been checks on the file sizes that can be read with FTP Adapter. OIC limits the largest file that can be read in-memory using FTP Adapter to maximum of 5 MB.

With this being the case, how could one read a 500 MB source file, transform to FBDI format and then upload to ERP?

Fortunately, OIC has ability to chunk read the large source file. This feature is called the Stage Read. Stage Read is a map reduce feature within OIC. Prior to making use of Stage Read in the OIC flow, one must make sure to download the file locally to OIC where OIC can read the file in chunks. The reason why the file must be downloaded to OIC prior to chunk reading it is because chunk reading from FTP Server is very expensive.



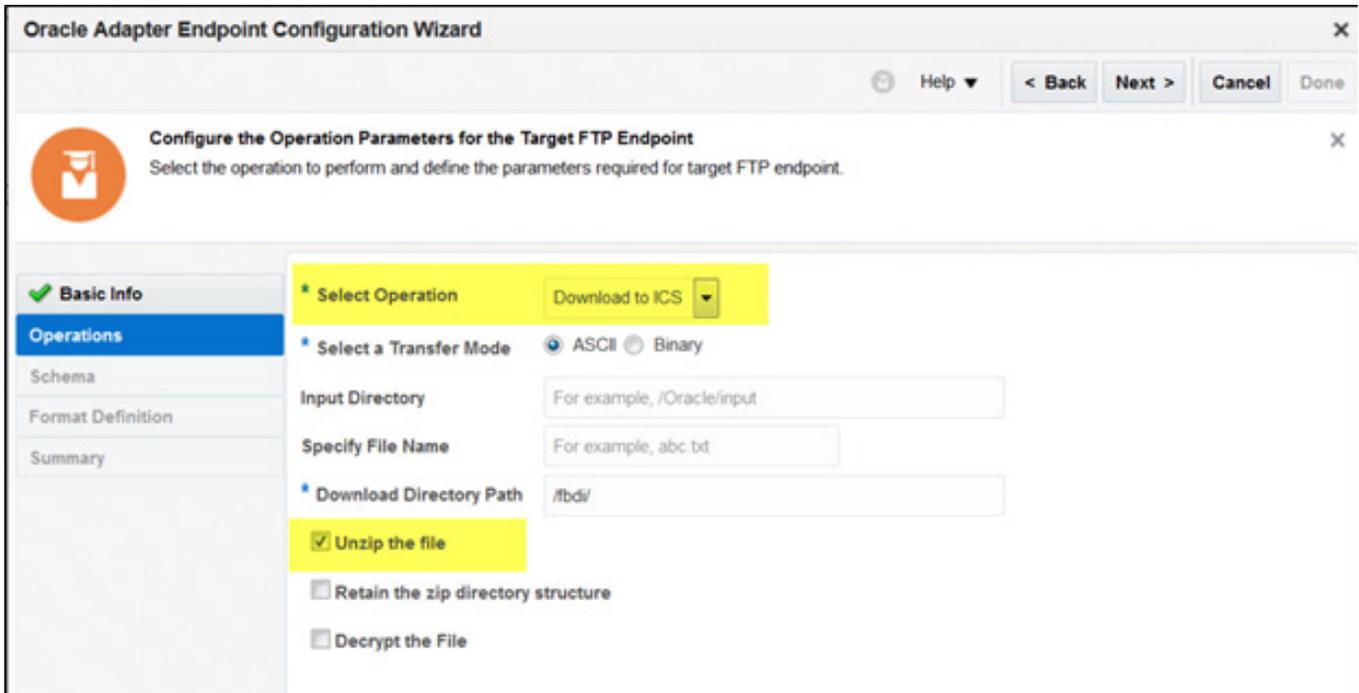
The flow has been marked with logical numbers to explain the different steps involved to generate the FBDI format.

1. Read Trigger File

This is scheduled OIC integration that gets triggered based on a scheduled configured. The first step is to read the trigger file that will be put in FTP location. The trigger file is zero-byte file having the same name as the source data file. The trigger file provides the name of the source data file to be read by the downstream activities in this OIC flow. A trigger file is used to make sure the source application can write the data file completely before OIC can pick up the data file to read. FTP Adapter in OIC does support the minimum age property but sometimes the source application may be trying to append to data file if there were many records. So using minimum age may not work in such scenarios and hence the need of having a trigger file.

2. Download Source Data File

The next step is to download the source data file into OIC. As explained earlier the reason behind downloading the source data file is because the chunking of the file on FTP Server is expensive operation. The OIC flow was written with the requirement that the source data file is a compressed file (.zip file) that may contain one or more source data format files.



3. Chunk Read and Transform to FBDI

In this step we make use of the Stage Read feature in OIC to chunk read file. Then every chunk is transformed from source format to FBDI format and then output FBDI file is written. The outer for-loop ensures that if there are multiple files as a part of the source zip file is read and transformed to FBDI formatted file.

4. List File

List file is needed for OIC to determine how many FBDI files are written out.

5. Generate Property File and Write zip file

OIC uses ERP Adapter to upload FBDI file to ERP. Although the ERP Adapter makes uses of ERP provided webservices internally, it obfuscates many complexities involved in the raw ERP webservices. The ERP Adapter expects a zip file containing FBDI file and Property file to upload to ERP. The property file is a manifest file that will contain information on Jobname and Parameter list that ERP needs to upload the FBDI file into ERP.

6. Archive Source File

The source file is archived for the future on an FTP location for re-submission purposes if necessary in the future.

7. Archive Trigger File

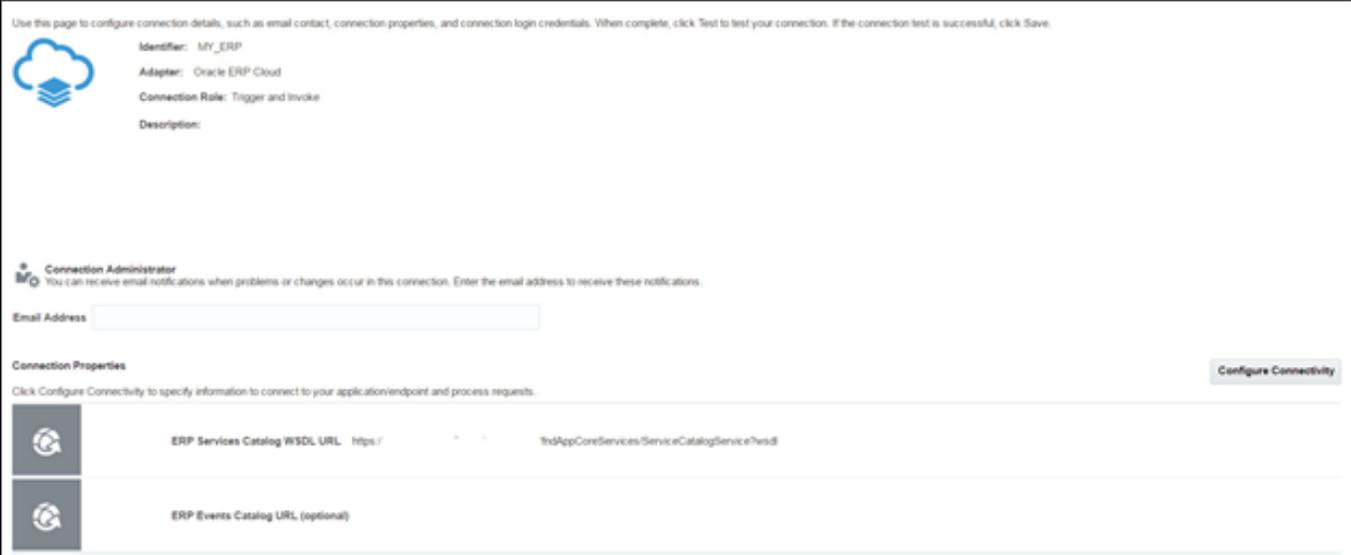
The trigger file is archived for the future on an FTP location for re-submission purposes if necessary in the future.

Step 2: Upload to ERP

The second step is to consume the FBDI file generated in Step 1 and upload the file to ERP. ERP exposes webservices (mainly SOAP) to upload data to ERP. While these services can be made use directly using a SOAP call, there are some complexities involved in terms of dealing with binary file like zip file to upload data and receive data. Retrieving and uploading the zip files using a simple SOAP call maybe little cumbersome. Fortunately, OIC has an ERP Adapter that abstracts several of these complexities. So, for all the cases, the ERP Adapter is used.

Configuring ERP Adapter Connection

To configure the ERP Adapter the serviceCatalog wsdl is needed. The service catalog wsdl is `https://<ERP Host Name>/fndAppCoreServices/ServiceCatalogService?wsdl`. The serviceCatalog WSDL is a catalog of all services exposed by the ERP. The catalog exposes services from all the module such as finance, project management, order management, inventory management and so on.



Use this page to configure connection details, such as email contact, connection properties, and connection login credentials. When complete, click Test to test your connection. If the connection test is successful, click Save.

 Identifier: MY_ERP
Adapter: Oracle ERP Cloud
Connection Role: Trigger and Invoke
Description:

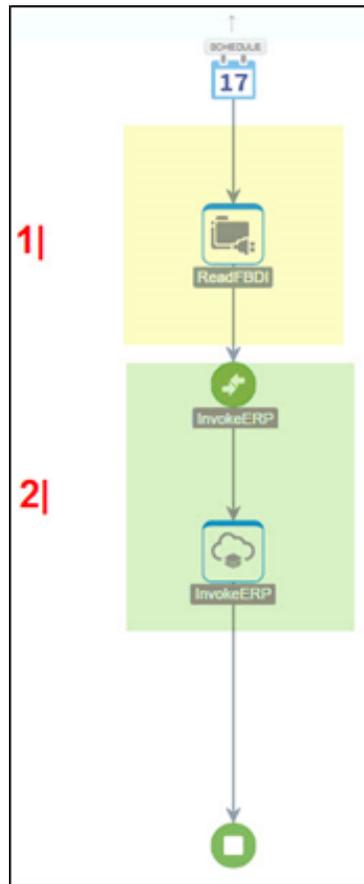
 **Connection Administrator**
You can receive email notifications when problems or changes occur in this connection. Enter the email address to receive these notifications.

Email Address

Connection Properties Configure Connectivity
Click Configure Connectivity to specify information to connect to your application endpoint and process requests.

	ERP Services Catalog WSDL URL	https://	fndAppCoreServices/ServiceCatalogService?wsdl
	ERP Events Catalog URL (optional)		

The OIC flow to upload the journal to ERP is shown below



1. Read FBDI Zip File

The flow is triggered by a scheduler. When the scheduler kicks off the process, the first activity reads the FBDI zip file created in the previous flow. Since the file is a zip file there is no schema needed to read the file.

1. Read FBDI Zip File

The flow is triggered by a scheduler. When the scheduler kicks off the process, the first activity reads the FBDI zip file created in the previous flow. Since the file is a zip file there is no schema needed to read the file.

The screenshot shows the 'Configure Oracle ERP Cloud Endpoint' wizard. The title bar includes 'Configure Oracle ERP Cloud Endpoint', a help icon, and navigation buttons: '< Back', 'Next >', 'Cancel', and 'Done'. A welcome message reads: 'Welcome to the Oracle ERP Cloud Endpoint Configuration Wizard. This wizard helps you configure an endpoint using the Oracle ERP Cloud connection. You will be asked to specify configuration parameters and define an operation for the service.' A sidebar on the left lists 'Basic Info', 'Operations', 'Response', and 'Summary', with 'Basic Info' selected. The main content area contains the following fields:

- A required text field: '* What do you want to call your endpoint?' with the value 'InvokeERP' entered.
- A text area: 'What does this endpoint do?' with the placeholder text 'Describe the endpoint's purpose and detail'.
- A radio button question: 'What actions would you want to perform on this endpoint?' with two options: 'Create, Update, Delete, Read Business Object' and 'Import Data into Financials Cloud Application' (which is selected).

The screenshot shows the second step of the wizard: 'Configure the Operations to Perform in the Target Oracle ERP Application'. The title bar and navigation buttons are the same as in the previous step. The instruction reads: 'Select the business object or service and operation to use for the target integration.' The sidebar now shows 'Basic Info' with a green checkmark and 'Operations' selected. The main content area contains:

- A required text field: '* Import Job Name' with the value 'import Journal' entered.
- A dropdown menu: 'Import Job Name' with the selected value 'Import Journals'.
- A checkbox at the bottom: 'Reuse job property file uploaded separately in respective UCM Account'.

- ✔ Basic Info
- ✔ Operations
- Response
- Summary

Notifications

Please provide notification preferences for communicating job completion status.

• Notification Mode

• Occurrence

Callback

Please provide the integration flow name and version for the callback

• Integration Flow Identifier

• Integration Flow Version

↓

- This is the identifier for the callback ICS flow.
- Not the identifier of the ICS flow needs to be provided NOT the name
- The callback ICS must exist in the same ICS instance it cannot reside on a different ICS instance

The ERP Adapter takes the zip file containing the FBDI file and Property File as input. Since the read activity has read the file, just the file reference and name need to be passed to the ERP Adapter as shown below.

Source	Mappings	Target	Mapping
<ul style="list-style-type: none"> ✚ <> *schedule <li style="padding-left: 20px;"><> *startTime <li style="padding-left: 20px;"><> *filename ✚ <> \$ReadFBDI <li style="padding-left: 20px;">✚ <> *ReadFileResponse <li style="padding-left: 40px;">✚ <> *FileReadResponse <li style="padding-left: 60px;">>< *FTPResponseHeader <li style="padding-left: 40px;">✚ <> *ICSFile <li style="padding-left: 60px;"><> *FileReference <input checked="" type="checkbox"/> <li style="padding-left: 40px;">✚ <> *Properties <li style="padding-left: 60px;"><> *filetype <li style="padding-left: 60px;"><> *directory <li style="padding-left: 60px;"><> *filename <input checked="" type="checkbox"/> <li style="padding-left: 60px;"><> *lastModifiedTime <li style="padding-left: 60px;"><> *creationTime <li style="padding-left: 60px;"><> *size <li style="padding-left: 60px;"><> *checksum 	<p>Drag and drop source to target to create a mapping.</p> <p>Click a checkmark on source or target to see mappings.</p>	<ul style="list-style-type: none"> ✚ <> *ImportBulkData <li style="padding-left: 20px;">✚ <> *ImportBulkData <li style="padding-left: 40px;">>< OutboundERPHeaderType <li style="padding-left: 20px;">✚ <> *ICSFile <li style="padding-left: 40px;">✔ <> *FileReference FileReference <li style="padding-left: 40px;">✚ <> *Properties <li style="padding-left: 60px;"><> *filetype <li style="padding-left: 60px;"><> *directory <li style="padding-left: 60px;">✔ <> *filename filename <li style="padding-left: 60px;"><> *lastModifiedTime <li style="padding-left: 60px;"><> *creationTime <li style="padding-left: 60px;"><> *size <li style="padding-left: 60px;"><> *checksum 	

Once the properties file is prepared then we can use it in two ways:

1. Properties file stored on UCM account

Manually upload the properties file to the UCM account of the import process (i.e. fin/journal/import). And then refer the properties file from the Oracle ERP Cloud Adapter endpoint. This properties file name can also be given in the payload in the jobOptions tag as below:

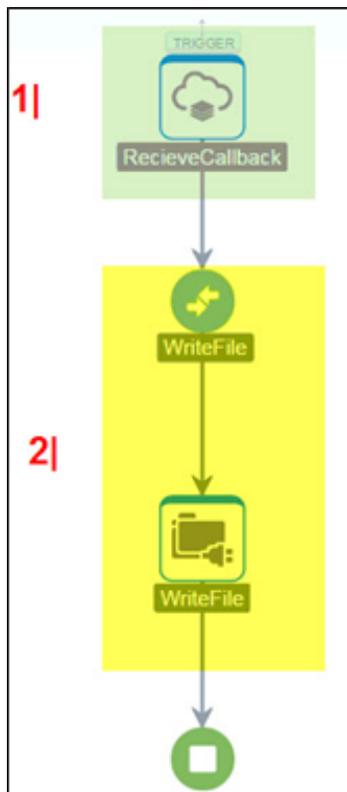
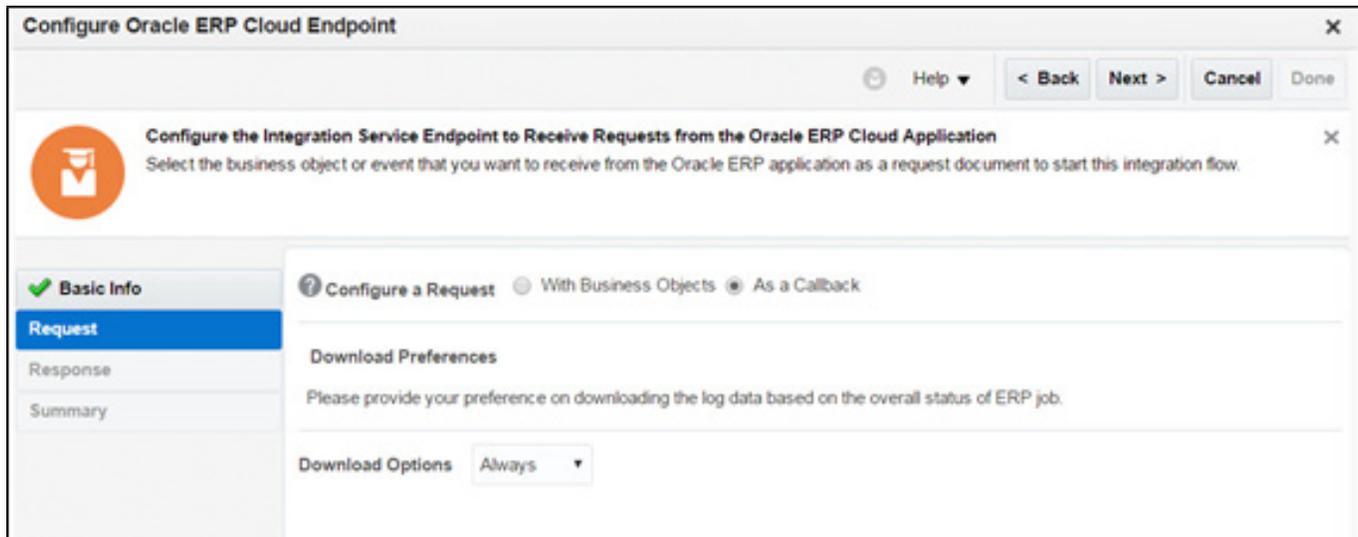
```
<ns1:jobOptions> JobDetailFileName=<fileName.properties></ns1:jobOptions>
```

2. Payload Parameter file in ZIP data file

While creating the zip of the data file, include the properties file in the ZIP itself.

Step 3: Receive Callback from ERP

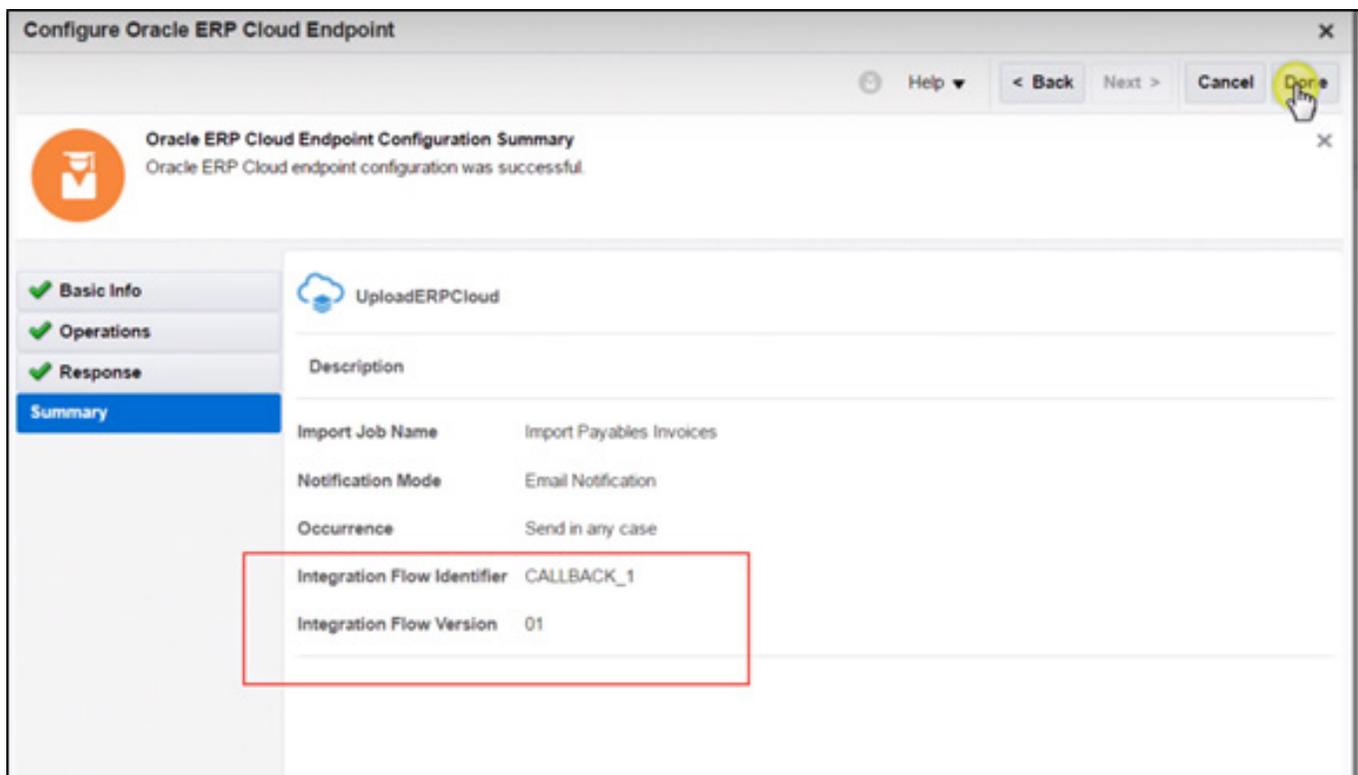
In this OIC flow a callback from ERP is received. It is possible to receive a callback on SUCCESS, ERROR or BOTH. See configuration below.



1. Receive Callback from ERP

The flow is triggered by when ERP asynchronously calls back OIC. The callback was configured when the ERP adapter was configured to invoke - Invoke ERP using ERP Adapter.

Callback Integration Identifier and its version needs to be defined in the Oracle ERP Cloud Adapter.



2. Write Result File

The callback from ERP contains a zip file that has all the results of the jobs that ran on the ERP. This can be written out to a FTP location.

1.1.1 Data Received From ERP

The ERP Callback has 2 parts.

JSON Message

The JSON message contains all the jobs that ran in ERP and its statuses. The JSON looks something like this –

```
{
  "JOBS":[
    {
      "JOBNAME":"Load Interface File for Import",
      "JOBPATH":"/oracle/apps/ess/financials/commonModules/shared/common/
interfaceLoader",
      "DOCUMENTNAME":"JTest1.zip",
      "REQUESTID":"603547",
      "STATUS":"SUCCEEDED",
      "CHILD":[
        {
          "JOBNAME":"Load File to Interface",
          "JOBPATH":"/oracle/apps/ess/financials/commonModules/shared/common/
interfaceLoader",
          "REQUESTID":"603549",
          "STATUS":"SUCCEEDED"
        },
        {
          "JOBNAME":"Transfer File",
          "JOBPATH":"/oracle/apps/ess/financials/commonModules/shared/common/
interfaceLoader",
          "REQUESTID":"603548",
          "STATUS":"SUCCEEDED"
        }
      ]
    },
    {
      "JOBNAME":"Import Journals",
      "JOBPATH":"/oracle/apps/ess/financials/generalLedger/programs/common",
      "REQUESTID":"603550",
      "STATUS":"ERROR"
    }
  ],
  "SUMMARYSTATUS":"ERROR"
}
```

ZIP File

This zip file sent from the ERP contains the results of all the jobs that ran on ERP. The content of the zip file is explained in the picture below.

Name	Size	Packed Size	Modified	Created	Accessed	Attributes	Encrypted	Comment	CRC	Method	Host OS	Ver
FailedRecords_GL_INTERFACE.csv	331 104	4 345	2017-09-22 02:44						28822082	Deflate	FAT	
603864.log	275	209	2017-09-22 02:44						C584AD4A	Deflate	FAT	
603863.log	15 919	2 162	2017-09-22 02:44						C6EA11CC	Deflate	FAT	
603862.log	208	167	2017-09-22 02:43						32F0A17A	Deflate	FAT	
603861.log	2 717	929	2017-09-22 02:43						81YDA584	Deflate	FAT	
IGInterface.csv	187 083	1 771	2017-09-22 02:43						9E8D482	Deflate	FAT	
JobDetails.properties	197	149	2017-09-22 02:43						13F48AF7	Deflate	FAT	