RedThunder.Blog

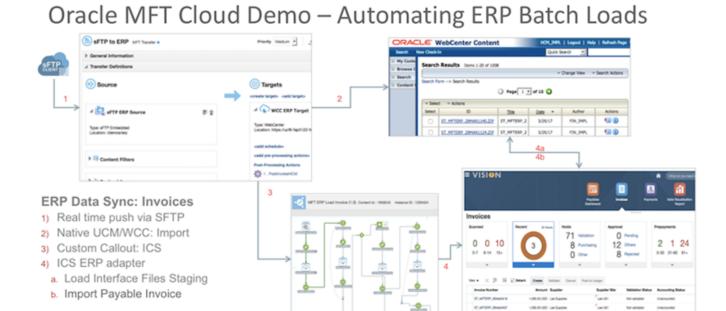
Demystifying cloud technologies...

Automate ERP Cloud Batch Uploads using Oracle MFT and Integration Cloud Service

In this blog, we will use Oracle Managed File Transfer (MFT) and Integration Cloud Service (ICS) to automate batch uploads into Oracle ERP Cloud. We will perform this upload with invoices, a typical use-case for organisations that work with numerous vendors and would like their vendors to bulk invoice them through say an SFTP file drop.

Steps are as follows:

- 1. Drop the file into an SFTP folder that is managed by MFT
- 2. MFT picks it up and uploads it into ERP Cloud's WebCenter Content file repository
- 3. MFT calls ICS to trigger a multi-step orchestration to load ERP Cloud
- 4. Using the ICS ERP Cloud adapter, the ICS orchestration first loads the file into staging tables and import the invoice into final tables



Folks who have read my previous blog, <u>Using MFT Cloud Service to Automate HCM</u>

<u>Batch Uploads – Part 1</u> will notice how similar this is to HCM Data Loading, except that we use ICS to orchestrate a few post-load steps.

That is because HCM and ERP Cloud (also SCM and SalesCloud) are all Oracle Fusion Applications, modules built on the same platform. Some of you may even be on "Global Shared Instances", essentially running both your HCM and ERP SaaSes in the same instance.

About MFT

MFT is Oracle's strategic platform for batch file transfer in all our SaaS services (and more). It provides cool capabilities such as zip/unzip, splitting large files, PGP encryption, retries, embedded FTP/SFTP services that are HA-clustered with user credentials that can be stored in your external Identity Management System. All offered through a dashboard interface.

In short, no more messing around in command-line, OS-level scripting tools to manage your file transfers!

MFT also provides a flexible callout extension framework to trigger downstream activities after a file has been transferred. In this case, we are triggering ICS to perform a series of activities to load ERP Cloud.

Read my blog <u>Using MFT Cloud Service to Automate HCM Batch Uploads – Part 2</u> to understand how the MFT callout extension works.

About ICS

Oracle Integration Cloud Service (ICS) delivers best-in-class Hybrid Integration. Built on our flagship Oracle SOA Suite and skinned to deliver a Line-of-Business SaaS experience, ICS provides a simple and powerful integration platform in the cloud that accelerates integration projects from months to days.

Highlights of ICS include:

- No install SaaS experience. Just login and integrate.
- Browser-based development platform. Click and drag to map.
- Host of pre-built <u>adapters</u> for many SaaS and legacy applications and a Cloud market-place to get additional third-party ones
- Secure hybrid integration through an Agent the punches-out through the firewall rather than VPN.
- Rich monitoring dashboards that provide real-time insight into transactions

ICS provides capabilities to string multiple transactions and transformations into a complex orchestration. A requirement for this ERP loading scenario.

Batch Loading ERP Cloud Manually

Batch Loading ERP Cloud through the browser consists of the following steps:

1. Create ERP Cloud Payload

- 2. Upload the file into WebCenter Content
- 3. Load the Interface File into Staging Tables
- 4. Import Invoices into Final Tables

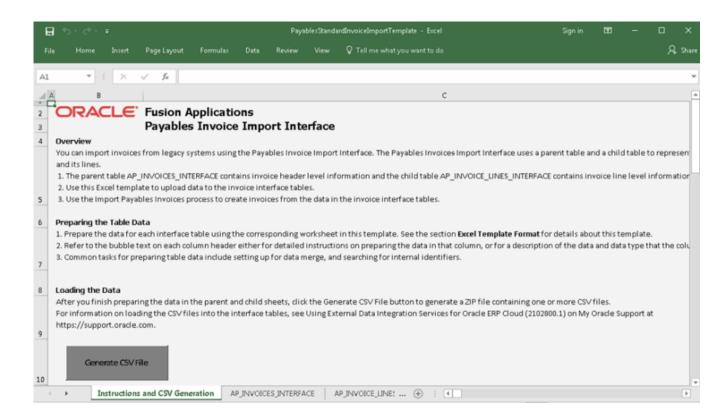
Create an ERP Cloud Payload

ERP Cloud provides Excel Macros with sample content to guide payload creation. The output files are Zipped CSVs. If you're integrating between 2 enterprise systems, you could request the vendor's system to generate the CSVs directly.

For Accounts Payable, we use **PayablesStandardInvoiceImportTemplate.xlsm**, grab it from our **Git** or **ERP** Cloud's Documentation.

*Note: My integration is for Fusion Release 11, so if you're reading my blog from Year 2050, best to get a new copy!

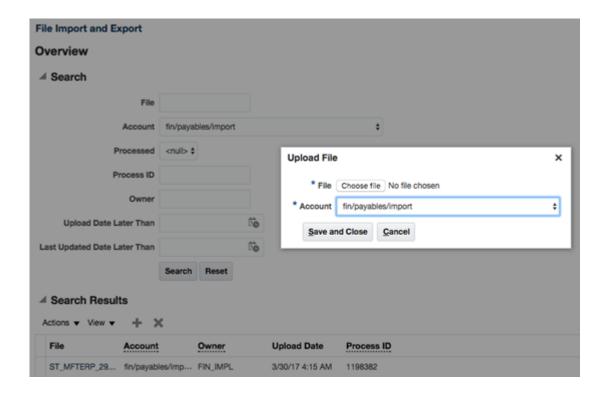
Key in the relevant payload and hit [Create CSV File]



Upload File into WebCenter Content

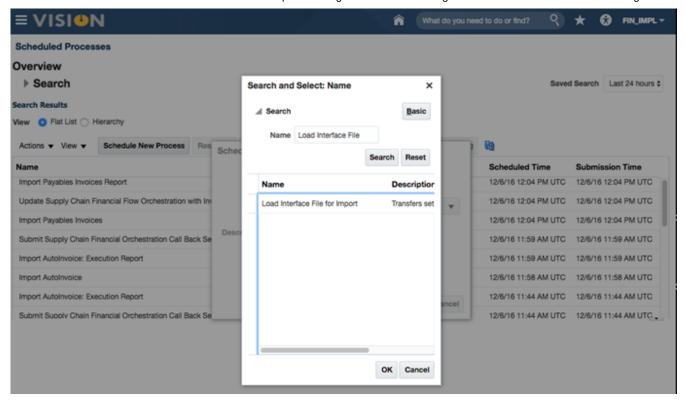
Navigate to [Tools] -> [File Import and Export] and perform the upload. Maximum size of the import through this method Is 2GB.

*Note: Make sure to select the Account matching that task you're trying to perform. For payables, I'm selecting "fin/payables/import", otherwise this file cannot be seen by the [Load Interface File for Import] job.



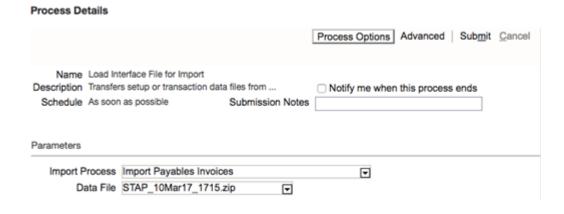
Load Interface File for Import

Next we need to load the file's data into Staging Tables. Batch processes in ERP Cloud run as Scheduled Process, so we will go to [Tools] -> [Scheduled Process] and then [Schedule New Process] -> [Load Interface File for Import]



Select the [Import Payables Invoices] and the file previously uploaded.

*Note: As you've select [Import Payables Invoices], you should only be able to see files uploaded to the relevant account "fin/payables/import"



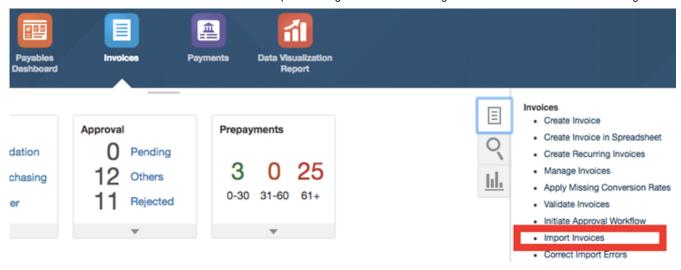
Import Payables Invoices

Now you're ready to load the final tables. Select [Tools] -> [Scheduled Process] and then [Schedule New Process] -> [Import Payables Invoices]

Select the [Business Unit] / [Ledger] you would like to import the invoices to. Select your [Source] depending the field you've entered in the payload. In my situation, I've used the term [External]. This essentially pulls up the rows in the Staging Table that have source indicator of [External].

Process Details Process Options | Advanced | Submit Cancel Name Import Payables Invoices Description Creates Oracle Fusion Payables invoices from in... Notify me when this process ends Schedule As soon as possible Submission Notes **Parameters** Business Unit US1 Business Unit 虿 ਓ * Ledger US Primary Ledger * Source +1 Import Set Invoice Group • Hold Reason Accounting Date 20 Purge Summarize Report Number of Parallel Processes

Alternatively, can you access the same screen from [Invoices] -> [Import Invoices]



Automating with Web Services

Batch processes in ERP Cloud run as Scheduled Processes and every Scheduled Process can be called by the ERP Integration Web Service.

The WSDL looks as below:

<u>https://</u>{ERP_BASE_URL}/publicFinancialCommonErpIntegration/ErpIntegrationS ervice?WSDL

There are methods to orchestrate the data load. Firstly, end-to-end jobs for upload, download and view logs:

- loadAndImportData (Inbound): Imports data into Oracle ERP Cloud.
- submitJobWithOutput (Outbound): Exports data from Oracle ERP Cloud.
- downloadESSJobExecutionDetails: Retrieves data and log files programmatically.

Or individual jobs to give you greater control of the load:

- uploadFileToUcm: Uploads files to UCM
- submitESSJobRequest: Load file from UCM to interface tables
- getEssJobStatus: Check if the job is done
- submitESSJobRequest: Load data from interface tables to application tables

- getEssJobStatus: Check if the job is done
- downloadESSJobExecutionDetails: Retrieves data and log files programmatically

The end-to-end jobs are a quick and dirty way to load data. However, this inserts the file as a Base64 SOAP payload. For large files and greater requirements such as retries, security and better traceability, we will use MFT as the transfer method for (a) and then perform (b)-(f) through ICS.

Filling in the Web Service Parameters

As the ERP Integration Web Service is a generic service, you will need point it to the necessary job to perform the activity. You will need:

- 1. jobPackageName
- 2. jobDefinitionName
- 3. Parameters for the job

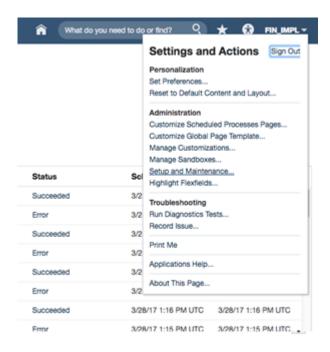
Here's a sample for Loading Interface File for Import:

```
<soapenv:Body>
<typ:jobPackageName>/oracle/apps/ess/financials/commonModules/s
hared/common/interfaceLoader</typ:jobPackageName>
<typ:jobDefinitionName>InterfaceLoaderController</typ:jobDefinition
Name>
<!-Zero or more repetitions:->
<typ:paramList>1</typ:paramList>
<typ:paramList>1943545</typ:paramList>
<typ:paramList>N</typ:paramList>
<typ:paramList>N</typ:paramList>
<typ:paramList>N</typ:paramList>
<typ:paramList>N</typ:paramList>
<typ:paramList>#NULL</typ:paramList>
</typ:submitESSJobRequest>
</soapenv:Body>
```

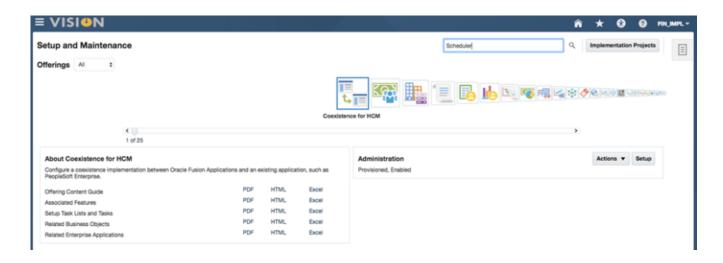
Finding information requires a little digging into the UI, but it is pretty straight-forward and repeatable once you get the drift.

jobPackageName and jobDefinitionName

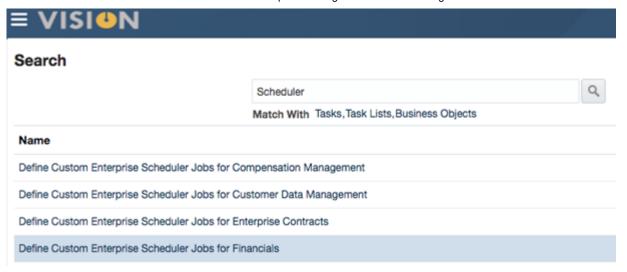
Goto [Setup and Maintenance] under Login



Search for [Scheduler]



Select [Define Custom Enterprise Scheduler Jobs for Financials]

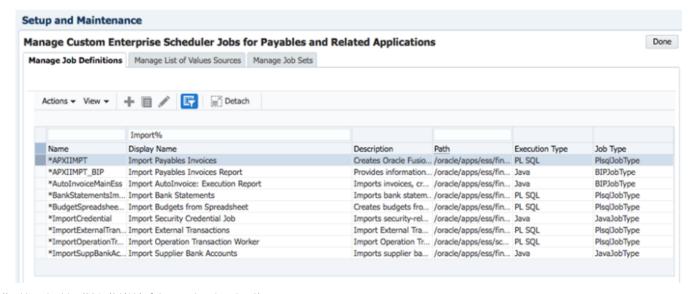


Select the List of Jobs to get details on

- [Manage Custom Enterprise Scheduler Jobs for Financials Common Applications] -> Search for [Load Interface File for Import]
- [Manage Custom Enterprise Scheduler Jobs for Payables and Related Applications] -> Search for [Import Payables Invoices]

Task List: Define Custom Enterprise Scheduler Jobs for Financials

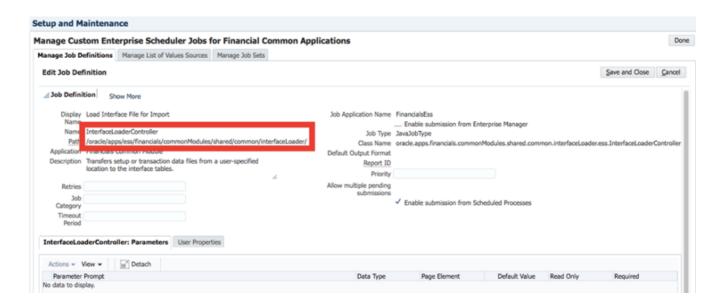




Select the Job Definition then click on [Actions] -> [Edit]. You will now see the parameters for the Payload.

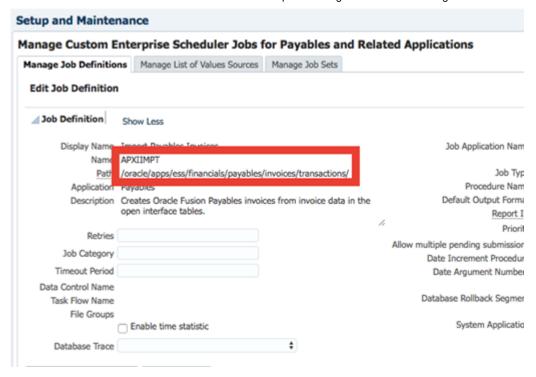
For [Load Interface for Import], we get the parameters:

- jobDefinitionName: InterfaceLoadController
- jobPackageName: /oracle/apps/ess/financials/commonModules/shared/common/interfaceLoader



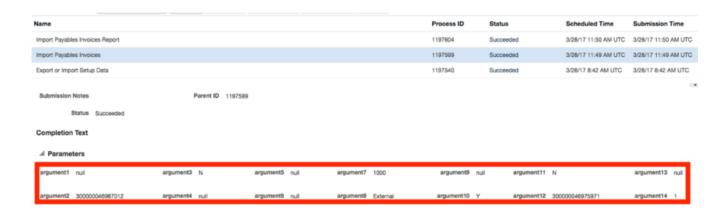
For [Import Payables Invoices], we get the parameters:

- jobDefinitionName: APXIIMPT
- jobPackageName: /oracle/apps/ess/financials/payables/invoices/transactions



Get Job Parameters

Now for the parameters, first run a batch load manually, then go to [Tools] ->
[Scheduled Processes] and select the job you have triggered manually. The
parameters are listed at the bottom of the job. Substitute necessary changes per your
data, e.g. LedgerID and DocID



Final SOAP Payload

This give us finally the SOAP payload we can utilise.

Loading Interface File for Import:

```
<soapenv:Body>
<typ:jobPackageName>/oracle/apps/ess/financials/commonModules/s
hared/common/interfaceLoader</typ:jobPackageName>
<typ:jobDefinitionName>InterfaceLoaderController</typ:jobDefinition
Name>
<!-Zero or more repetitions:->
<typ:paramList>1</typ:paramList>
<typ:paramList>1943545</typ:paramList>
<typ:paramList>N</typ:paramList>
<typ:paramList>N</typ:paramList>
<typ:paramList>#NULL</typ:paramList>
</typ:submitESSJobRequest>
</soapenv:Body>
```

Import Payables Invoices:

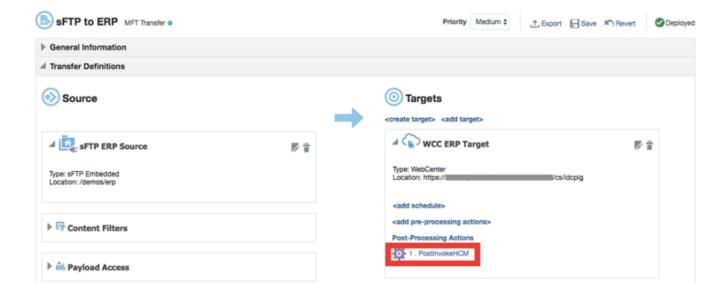
```
<soapenv:Body>
<typ:submitESSJobRequest>
<typ:jobPackageName>/oracle/apps/ess/financials/payables/invoices/t
ransactions/
</typ:jobPackageName>
<typ:jobDefinitionName>APXIIMPT</typ:jobDefinitionName>
<!-Zero or more repetitions:->
<typ:paramList>#NULL</typ:paramList>
<typ:paramList>30000046987012</typ:paramList>
<typ:paramList>N</typ:paramList>
<typ:paramList>#NULL</typ:paramList>
<typ:paramList>#NULL</typ:paramList>
<typ:paramList>#NULL</typ:paramList>
<typ:paramList>1000</typ:paramList>
<typ:paramList>External</typ:paramList>
<typ:paramList>#NULL</typ:paramList>
<typ:paramList>Y</typ:paramList>
<tvp:paramList>N</tvp:paramList>
```

```
<typ:paramList>30000046975971</typ:paramList>
```

- <typ:paramList>#NULL</typ:paramList>
- <typ:paramList>1</typ:paramList>
- </typ:submitESSJobRequest>
- </soapenv:Body>

Create MFT Transfer to Perform the WebCenter Upload

Now let's create the transfer process. This allows us to drop a file into MFT's SFTP server that will be automatically uploaded to ERP's WebCenter. Follow the steps from my <u>previous blog</u> for HCM and substitute with the ERP Cloud's information instead.

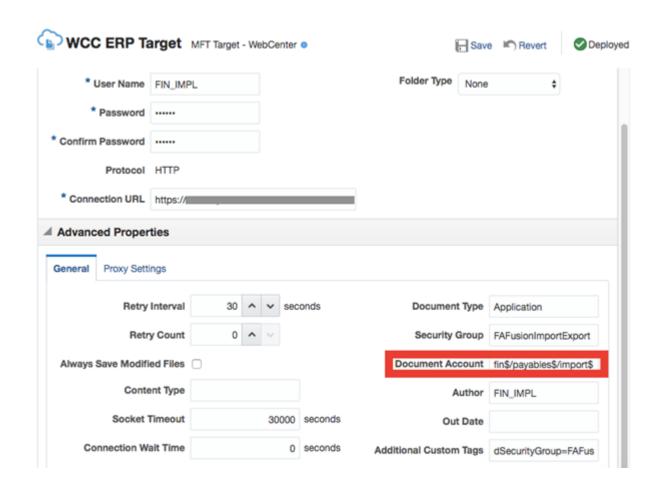


As HCM and ERP are just modules deployed on the same platform, the WebCenter end-point is the same: https://[ERP_Base_URL]/cs/idcplg. In fact, if you are running a "Global Shared Instance", you should see both your HCM and ERP files being drop in the same place!

You've probably noticed in the manual process prior, I've explained that Document Account must be "fin/payables/import". Otherwise this file cannot be seen by the

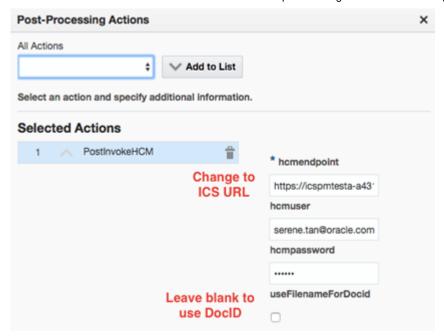
Load Interface File for Import job. Enter that information in [Advanced Properties] of the target:

- Document Type: Application
- Security Group: FAFusionImportExport
- Document Account: fin\$/payables\$/import\$
- Author: {YOUR USER}
- Additional Custom Tags: dSecurityGroup=FAFusionImportExport,dExtension=zip,dWebExtension=zip,dF ormat=application/zip



As a shortcut, I'm going to use the <u>Java Callout Code</u> created in <u>Using MFT Cloud</u> <u>Service to Automate HCM Batch Uploads – Part 2</u>, but instead of triggering HCM directly, I will point the Callout to ICS instead.

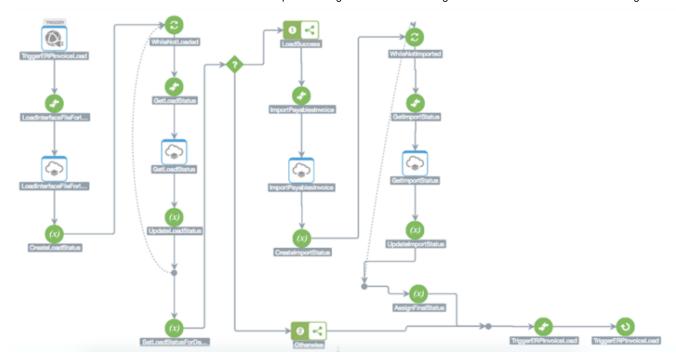
*Note: HCM Data Loader uses the name of the file (ContentID) as the load source, whereas ERP expects the DocumentID. Make sure you're passing in the right parameter.



Create an ICS Orchestration for the Scheduled Processes

For this orchestration to work, you will need at least ICS version 17.1.3, which has "While" loops.

The final output looks like this. You can grab the <u>sample orchestration</u> from GitHub and import it directly to your ICS.

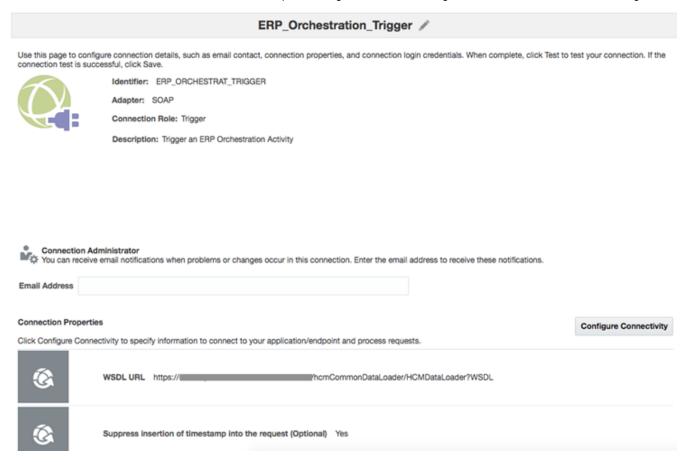


Essentially the steps are:

- ICS SOAP Trigger that begins the process (called by MFT)
- · Call ESS job to Load Interface File into staging tables
- Create a variable to hold the Load Status in the While Loop
- Keep calling GetESSJobStatus till the Load Status returns SUCCEEDED or FAILED
- Exit if the Load Status is FAILED
- Call ESS job to Import Payables Invoice into final tables
- · Create a variable to hold the Import Status in the While Loop
- Keep calling GetESSJobStatus till the Import Status return SUCCEEDED or FAILED
- Return SUCCEEDED or FAILED based the ESS Job Result

Create an ICS SOAP Trigger

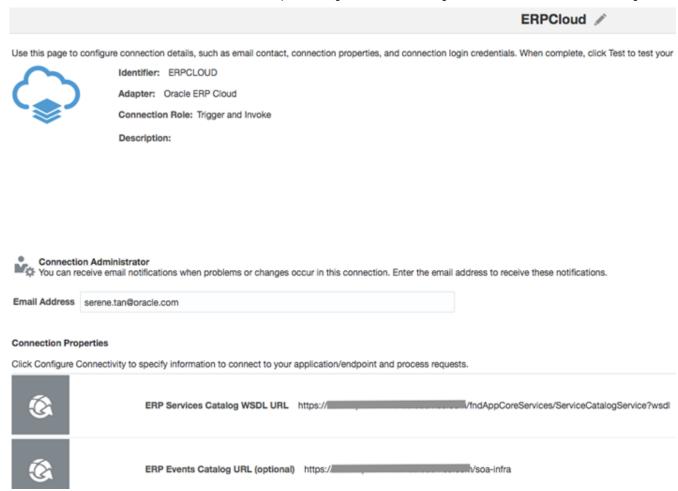
First create a new SOAP connector in ICS. As a shortcut, I am using the HCM WSDL as the SOAP definition for the ICS callout. If you're using my demo orchestration code, you can grab the WSDL from here.



Create an ERP Cloud Connection

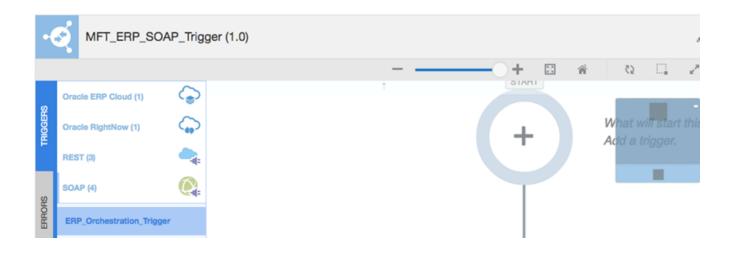
Create an ERP Cloud Connection, your connection WSDL should be similar to: https://ERP_BASE_URL /fndAppCoreServices/ServiceCatalogService?WSDL

If you also wish to listen to events created by ERP Cloud (not covered in this blog), you can enter https://{ERP_BASE_URL}/soa-infra



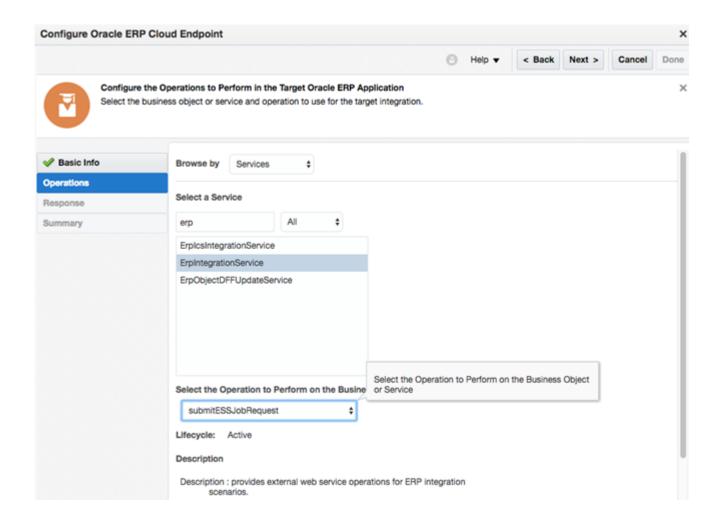
Create an ICS Orchestration

Now create an ICS orchestration that is triggered by the SOAP Trigger.

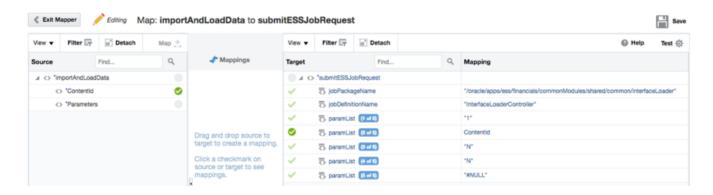


Load Interface File

To call the Load Interface File job, drag the connector onto the orchestration and select [Browse by Service] -> [ERPIntegrationService] -> [submitESSJobRequest]. This will essentially perform the Web Service call we described manually.

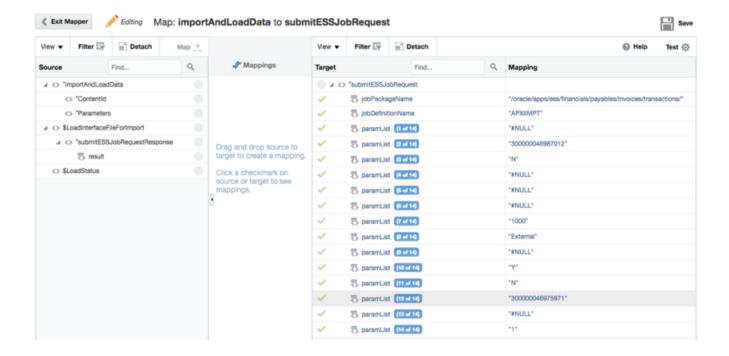


Perform the mapping to pass in the relevant payload. The MFT transfer will return a DocID from the WebCenter transfer. This is the reference to the file uploaded in the system. We have saved this value as ContentId in the Java Callout which will then be passed through ICS into the parameters. Note that the other values have come from our exploration of the [Scheduled Processes] run manually.



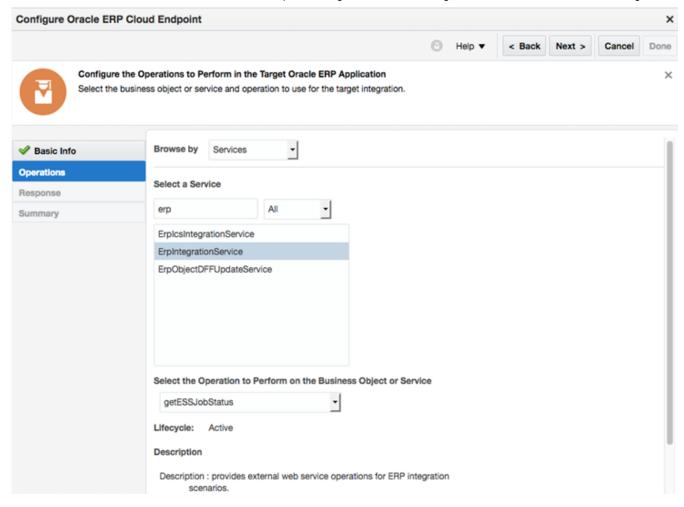
Import Payables Invoice

We will call the same Web Service for Import Payables Invoice with different parameters. In this situation, we have hardcoded the LedgerID. In the real-world, we would pass this in through ICS parameters.

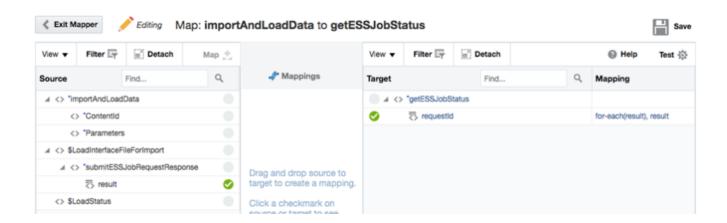


Get Status of Load or Import

To check the status of a job, drag the connector onto the orchestration and select [Browse by Service] -> [ERPIntegrationService] -> [getESSJobStatus].



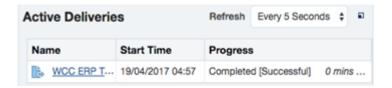
Perform the mapping to pass in the relevant payload. SubmitESSJobRequest returns a JobID, which we will keep querying till it returns SUCCEEDED or FAILED.



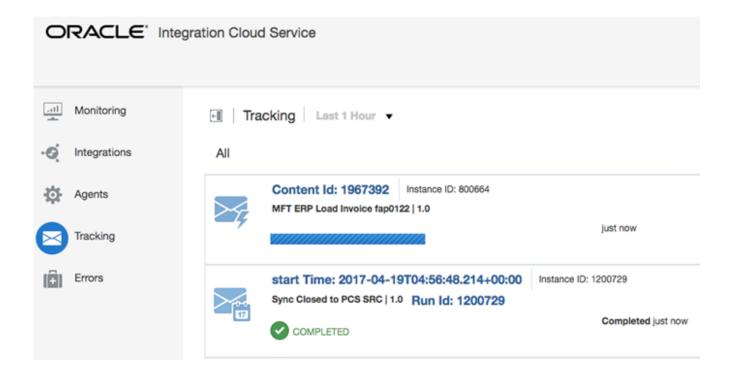
Perform the Transfer and View Success

Now grab your payload created earlier and drop the file into MFT's SFTP embedded server.

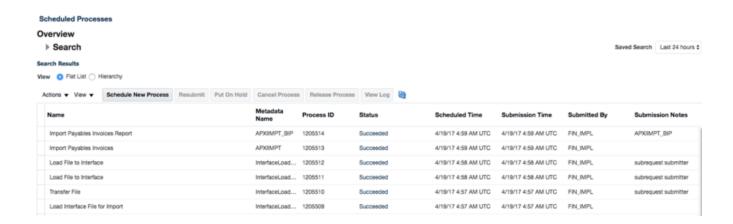
Navigate to MFT and you can see the upload in progress.



Navigate to ICS and you can the orchestration has kicked off.

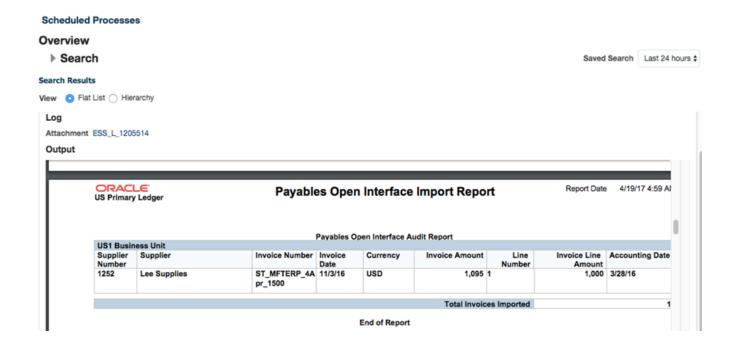


Navigate to [Scheduled Process] and you can see the jobs that have been called.

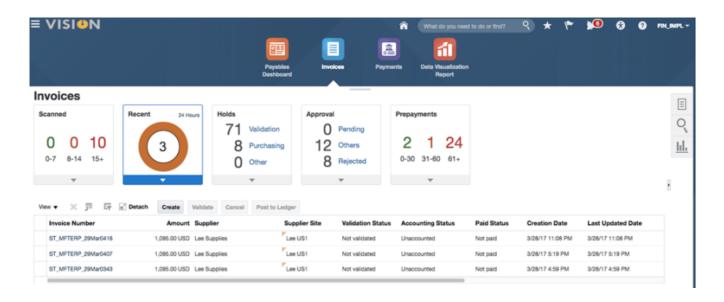


Click on [Import Payables Invoices Reports] -> [Succeeded] and you will see a BIPublisher (PDF) report and the items that have been inserted / failed. You can also

retrieve this information through the Web Service call – downloadESSJobExecutionDetails.



Now finally if [Submit Invoice Import] has succeeded, navigate to [Invoices] and you can see that the new Invoices has been created under the [Recent] tab!



* Content in this blog is thanks to: Dave Berry, Rakesh Singh and Juned Ahmed Syed

Share this:





2 bloggers like this.



Author: Serene Tan

I'm a Cloud Solutions Architect (Platform-as-a-Service) for Oracle based in Melbourne. I love integrating and connecting anything – humans, applications, machines, businesses – to create an end-to-end holistic experience. My motto. NO MORE SILOES.

View all posts by Serene Tan



Serene Tan / April 19, 2017 / ERP Cloud, Integration Cloud Service, MFT

One thought on "Automate ERP Cloud Batch Uploads using Oracle MFT and Integration Cloud Service"



September 24, 2018 at 4:00 pm

Very detailed presentation and helps in our coding. Thank you.



RedThunder.Blog / @