

How to Extract Data from Oracle Fusion Cloud Using OIC

This guide explains how to extract data from Oracle Fusion Cloud using Oracle Integration Cloud (OIC). It walks through building a scheduled integration that triggers an export job in Oracle Fusion ERP Cloud, and a companion app-driven integration that receives the completed file and writes it to an FTP directory. Following these steps, you can set up a repeatable, automated process to export bulk data from Oracle Fusion ERP Cloud on a schedule.

Prerequisites

Before you begin, make sure you have:

- Access to Oracle Fusion ERP Cloud with permissions to view ESS job definitions and Setup and Maintenance.
- An Oracle Integration Cloud (OIC) instance with the **ERP adapter** and a **SOAP connection** configured.
- An FTP directory target to receive the exported files.
- The **ERP Integration Service** WSDL URL for your Fusion environment.

Overview of the two export methods

There are two approaches for exporting bulk data from Oracle Fusion ERP Cloud using OIC:

- **Method 1:** Combines a scheduled integration with an app-driven integration exposed over SOAP/HTTPS. This method is covered in this guide.
- **Method 2:** A separate approach, covered elsewhere.

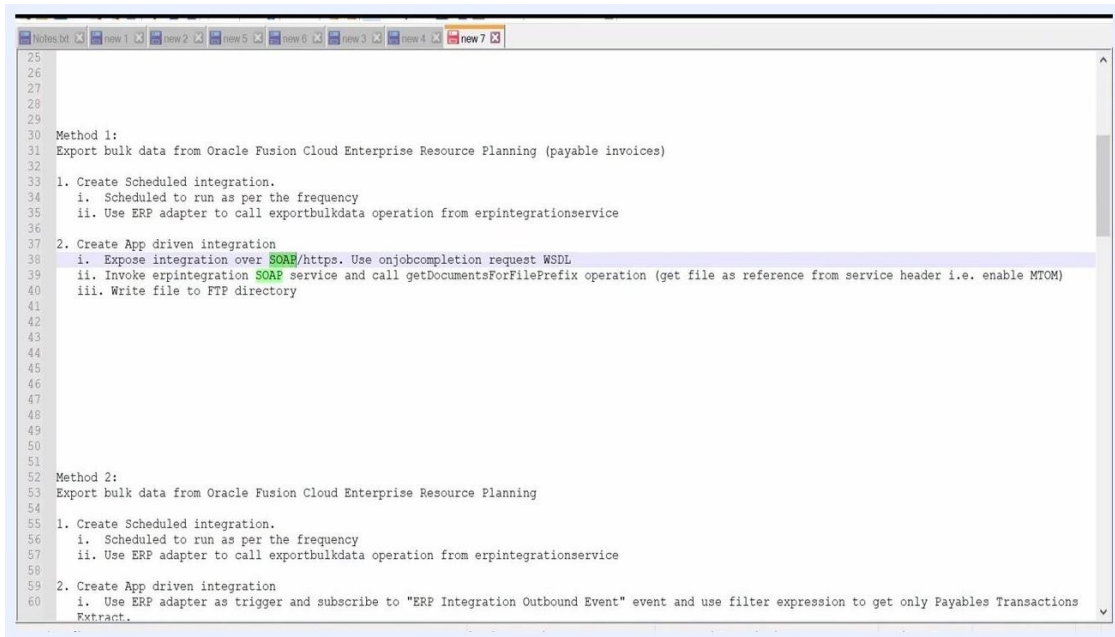
Method 1 uses a scheduled integration to trigger the export job, and a callback (app-driven) integration to retrieve the resulting file once the job completes.

Part 1: Plan the integration design

1. Review the two-part integration approach

Design a scheduled integration in OIC that runs at the frequency you require (for example, daily or hourly). This integration uses the **ERP adapter** to call the [exportbulkdata](#) operation from the [erpintegrationservice](#).

You then expose a second integration over **SOAP/HTTPS** using the **onjobcompletion request WSDL**. When the scheduled integration completes the export, Fusion invokes this app-driven integration as a callback and passes a **document ID**. This callback integration uses a **SOAP connection** (not the ERP adapter) to call the [erpintegration](#) SOAP service and its [getDocumentsForFilePrefix](#) operation, retrieving the file as a reference from the service header with MTOM enabled, and writing it to an FTP directory.

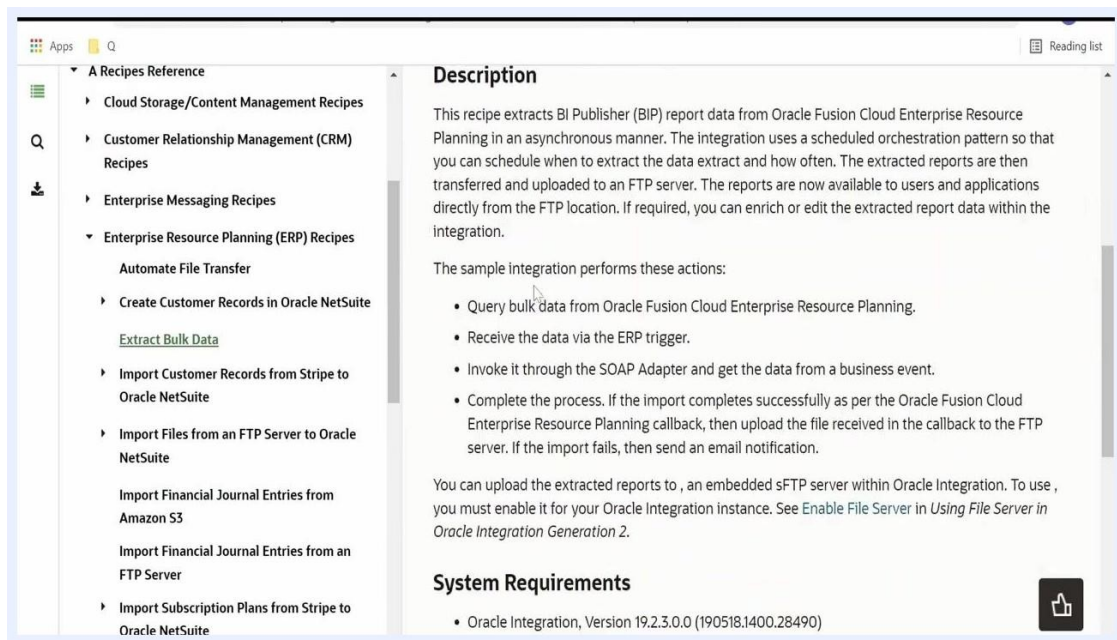


```
25  
26  
27  
28  
29  
30 Method 1:  
31 Export bulk data from Oracle Fusion Cloud Enterprise Resource Planning (payable invoices)  
32  
33 1. Create Scheduled integration.  
34 i. Scheduled to run as per the frequency  
35 ii. Use ERP adapter to call exportbulkdata operation from erpintegrationservice  
36  
37 2. Create App driven integration  
38 i. Expose integration over SOAP/https. Use onjobcompletion request WSDL  
39 ii. Invoke erpintegration SOAP service and call getDocumentsForFilePrefix operation (get file as reference from service header i.e. enable MTOM)  
40 iii. Write file to FTP directory  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52 Method 2:  
53 Export bulk data from Oracle Fusion Cloud Enterprise Resource Planning  
54  
55 1. Create Scheduled integration.  
56 i. Scheduled to run as per the frequency  
57 ii. Use ERP adapter to call exportbulkdata operation from erpintegrationservice  
58  
59 2. Create App driven integration  
60 i. Use ERP adapter as trigger and subscribe to "ERP Integration Outbound Event" event and use filter expression to get only Payables Transactions  
Extract.
```

Text editor window showing the steps for creating an app-driven integration, including exposing the integration over SOAP/HTTPS, using the onjobcompletion request WSDL, invoking the erpintegration SOAP service, calling getDocumentsForFilePrefix, and writing the file to an FTP directory.

2. Enable MTOM for document retrieval

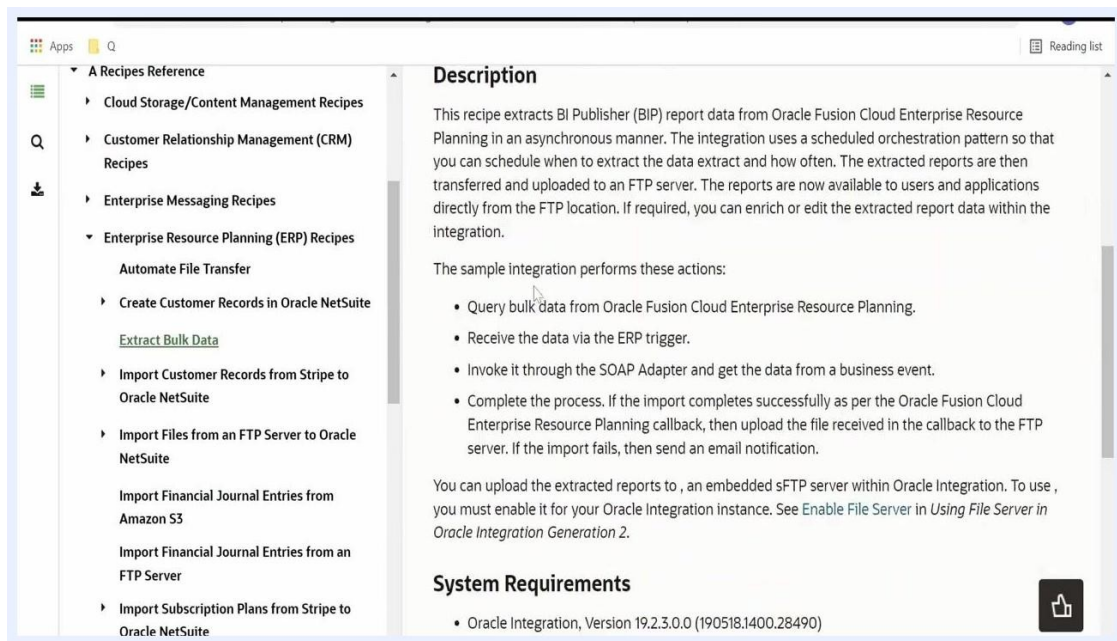
Use the **getDocumentsForFilePrefix** operation in the SOAP service to fetch documents from UCM (Universal Content Management). Enable **MTOM** (Message Transmission Optimization Mechanism) when calling this operation so that large files can be received as references in the service header. After fetching the file, write it to an FTP directory. If your use case requires syncing data with other applications, adapt this file-handling step accordingly.



Text editor window showing detailed steps for Method 1, including enabling MTOM for the **getDocumentsForFilePrefix** operation and writing the file to an FTP directory.

3. Review the Extract Bulk Data recipe documentation

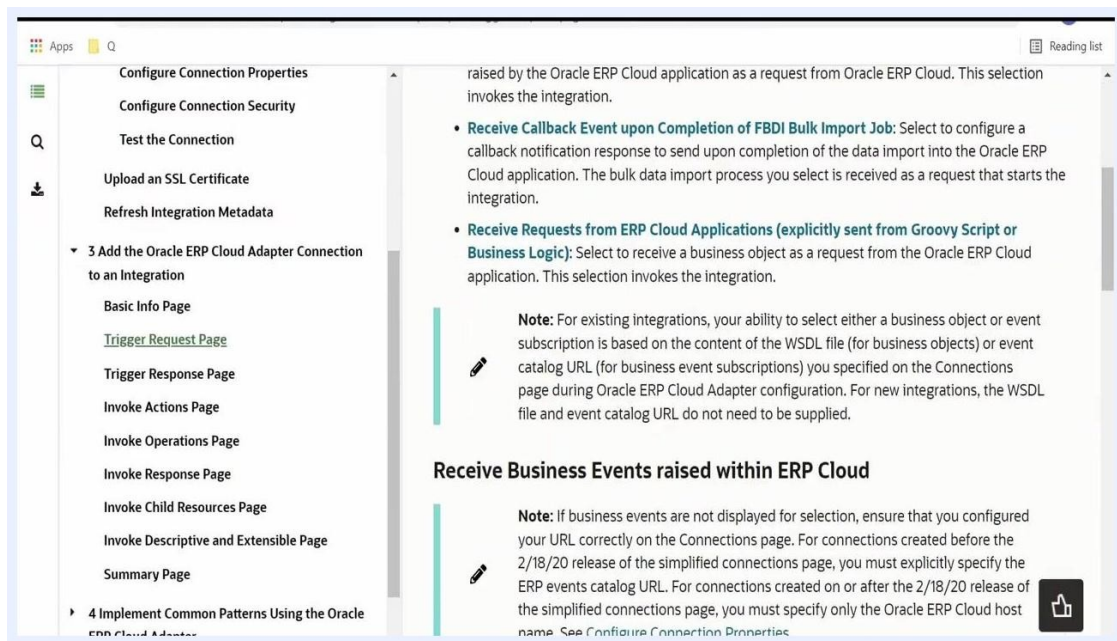
Refer to Oracle's official documentation for step-by-step instructions on building this integration, including how to configure the **ERP Cloud Adapter** for both trigger and callback scenarios, how to generate the WSDL for the ERP Integration Service, and the required hostnames and available operations.



Oracle documentation page describing the Extract Bulk Data recipe, listing actions such as querying bulk data, receiving data via the ERP trigger, invoking the SOAP Adapter, and uploading to FTP.

4. Configure the ERP Cloud Adapter trigger

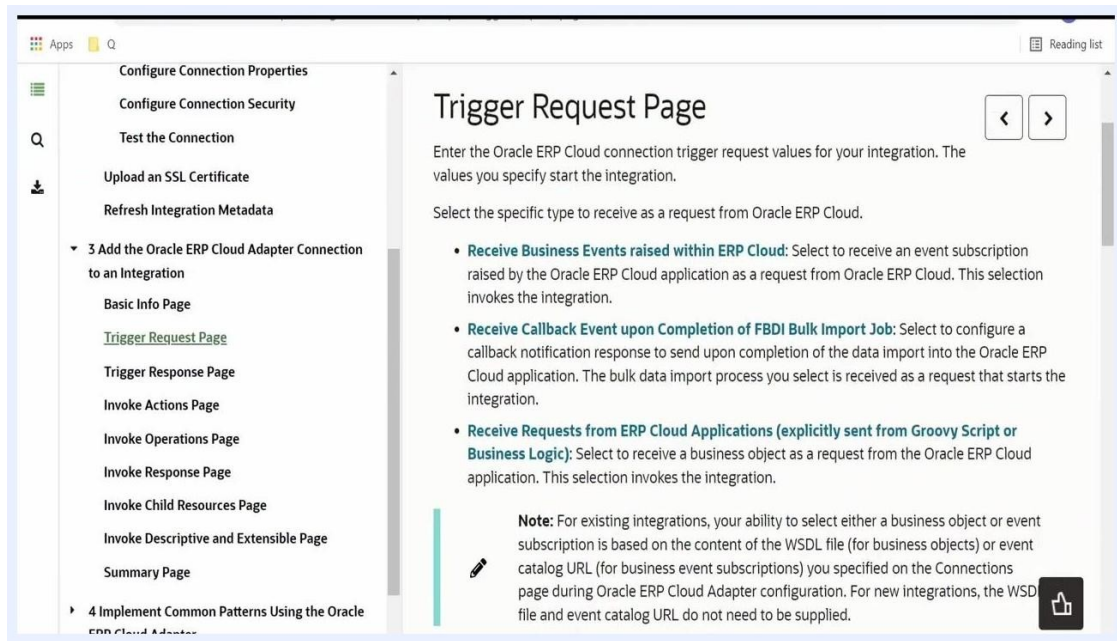
Follow the documentation to set up connection properties and security, test the connection, and upload SSL certificates if required. Add the adapter to your integration and configure the **Trigger Request Page**, choosing the appropriate trigger type such as business event, callback event, or explicit request.



Oracle documentation page showing the Trigger Request Page, with options for receiving business events, callback events, and requests from ERP Cloud applications.

5. Review the `getDocumentsForFilePrefix` request payload

The `getDocumentsForFilePrefix` operation retrieves files matching a specified prefix and account. The request payload includes a **prefix** (string) to search for, an **account** (string) for the security account, and **comments** (string) that are not present in the files. Review the documentation for the expected request and response payloads.

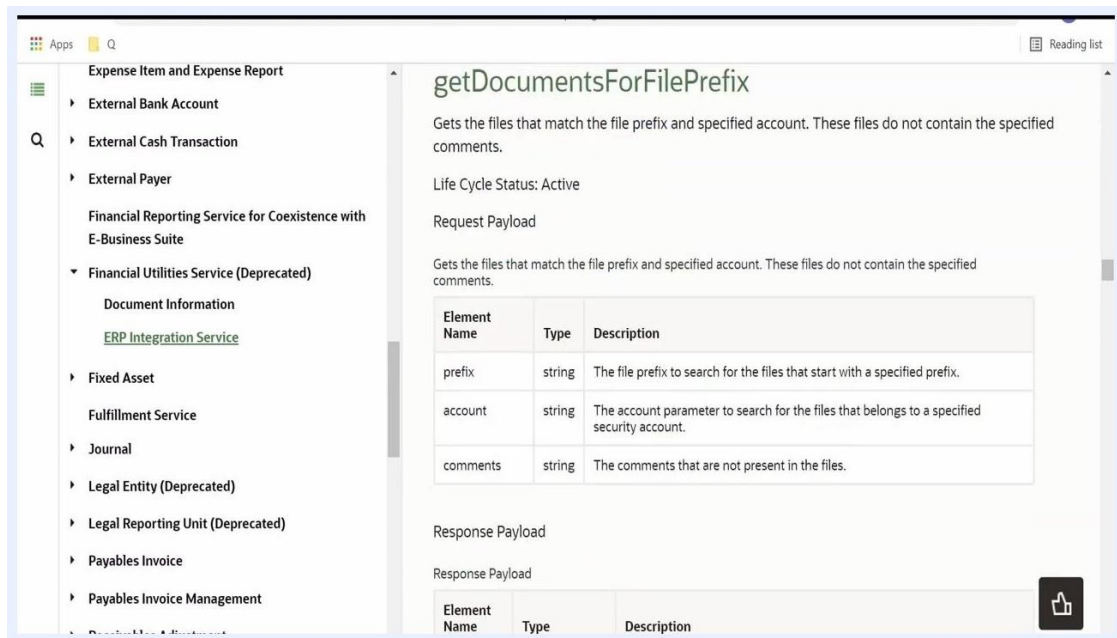


Oracle documentation page showing details for the `getDocumentsForFilePrefix` operation, including request payload fields: **prefix**, **account**, and **comments**.

Part 2: Build the scheduled integration

6. Create the scheduled orchestration and select the export operation

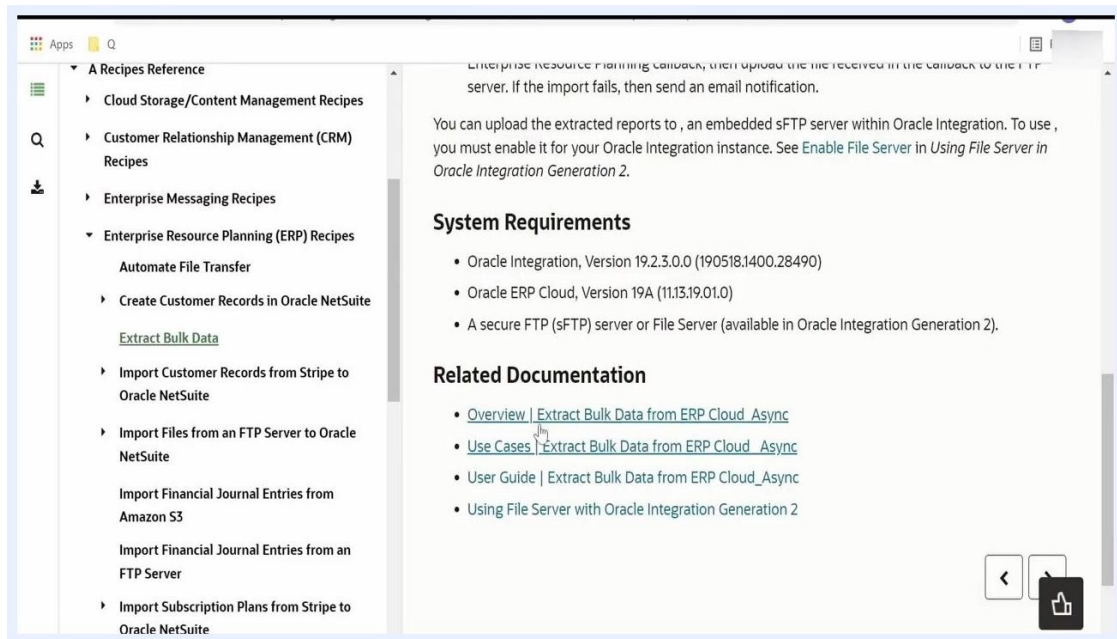
In Oracle Integration Cloud, create a **Scheduled Orchestration** for the ERP bulk extract. Configure the Oracle ERP Cloud Endpoint by selecting the **ErpIntegrationService** as the service and the **exportBulkData** operation. Confirm the operation is active and review its description for context.



Oracle Integration Cloud screen showing configuration of the ERP Cloud Endpoint, with **ErpIntegrationService** and the **exportBulkData** operation selected.

7. Locate the exportBulkData operation in the ERP adapter

In the integration designer, select the ERP adapter and find the **exportBulkData** operation. Review the business object and operation details listed for your selected object, such as Payable Transactions.



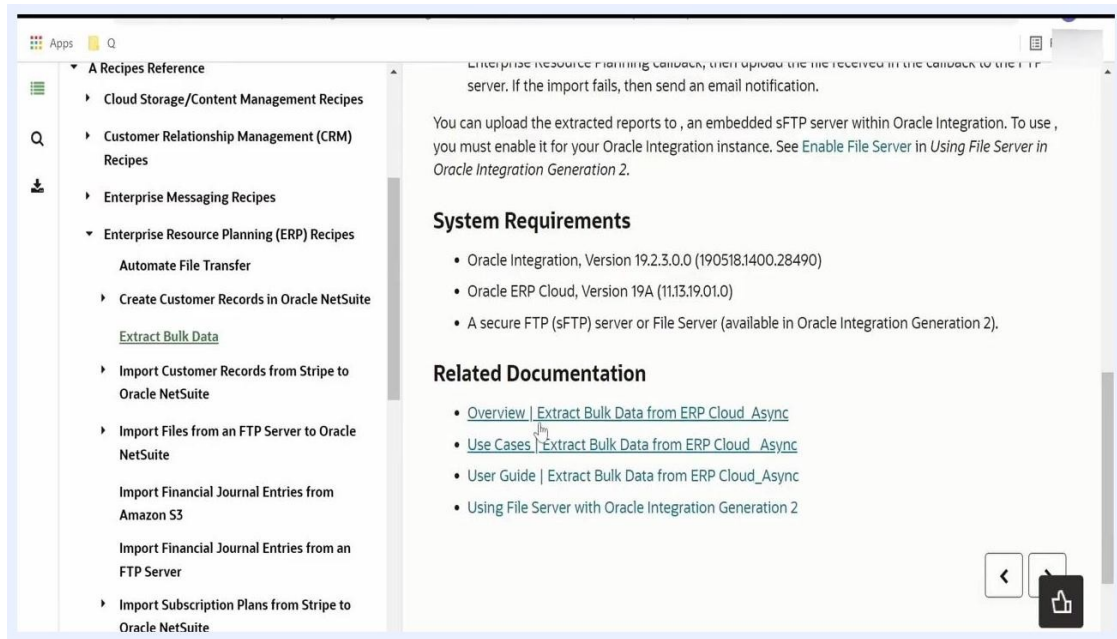
Screen showing the exportBulkData operation documentation, listing required request payload fields: **jobName**, **parameterList**, **jobOptions**, **callbackURL**, and **notificationCode**.

8. Map the required fields in the mapper

In the mapping step, provide values for each required field in the exportBulkData request:

- **Job Name:** the ESS job package and definition name.
- **Parameter List:** the parameters required by the ESS job.
- **Job Options:** options such as file type.
- **Callback URL:** the URL of your OIC app-driven integration.

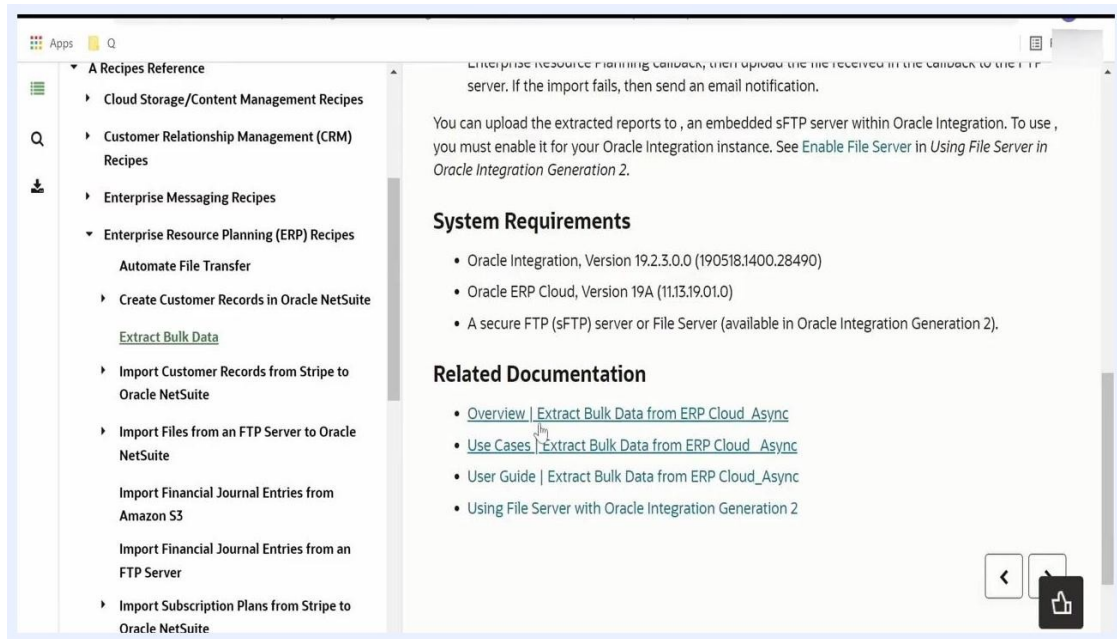
- **Notification Code:** the code for your required notifications (for example, "30" for email and bell notification).



Oracle Integration Cloud mapping screen showing the mapping of Job Name, Parameter List, Job Options, Callback URL, and Notification Code for exportBulkData.

9. Reference the use case document for job name and parameters

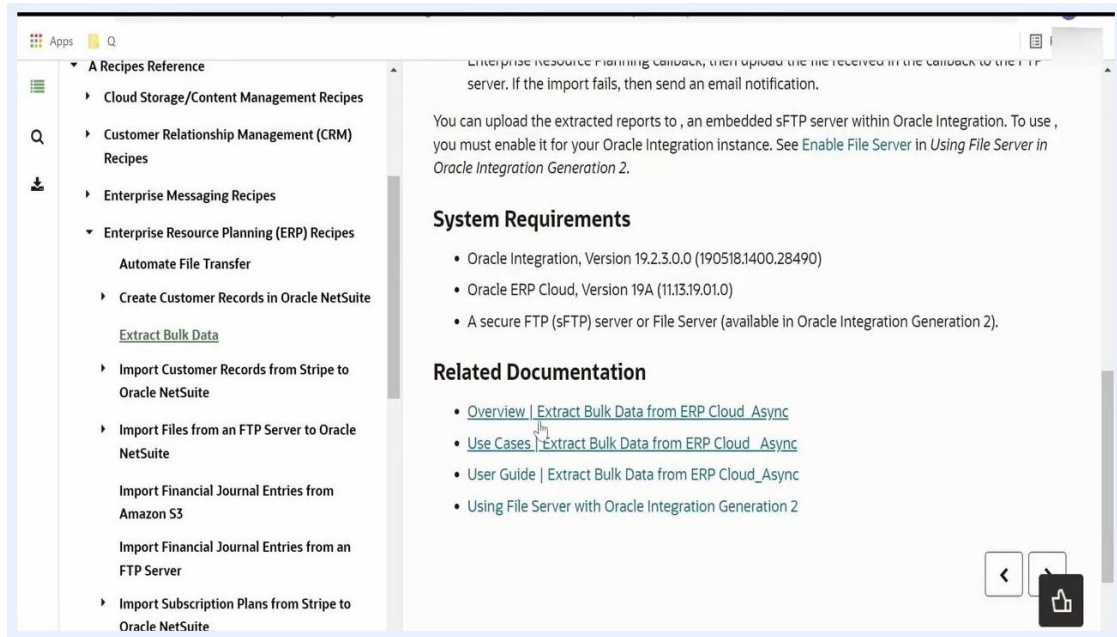
Access the use case document linked under **Related Documentation** and follow its instructions to navigate to **Setup and Maintenance** in Oracle ERP Cloud.



Documentation page showing Related Documentation links, including Use Cases and the User Guide for extracting bulk data from ERP Cloud.

10. Search for the ESS job definition

In Oracle ERP Cloud, go to **Setup and Maintenance** and search for **Manage Enterprise Scheduler Job Definitions for Financial, Supply Chain, and Related Applications**. Locate the job definition for your business object, such as Payables Transactions Extract.

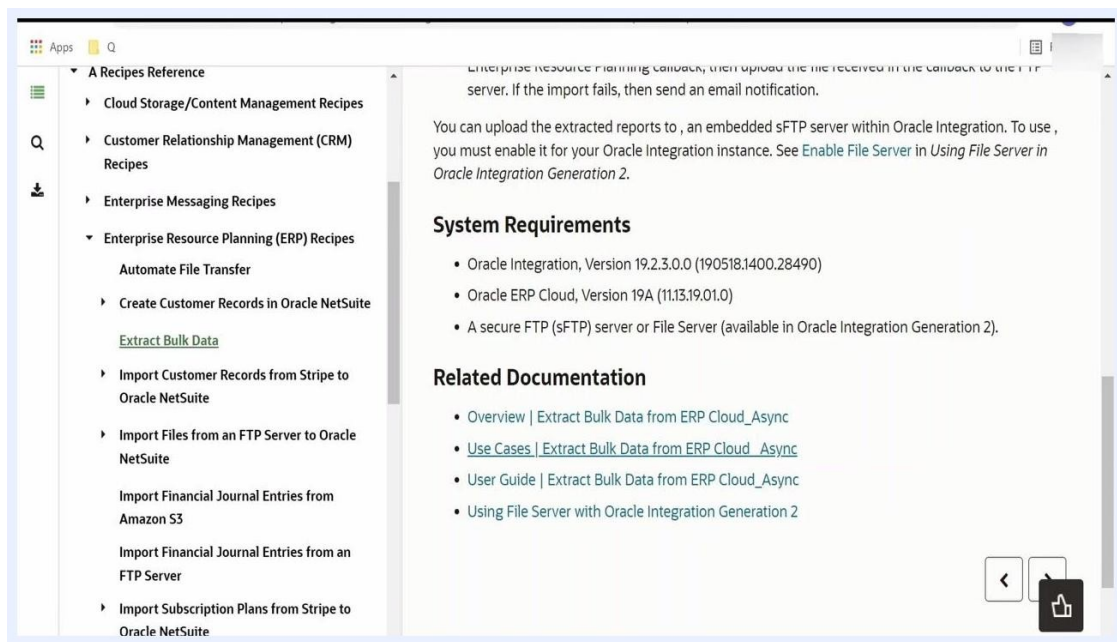


ESS Job Definition screen showing details for Payables Transactions Extract, including path, name, application, and description.

11. Copy the path and name for the job name field

In the job definition details, note the **Path** and **Name** fields. Use the syntax `<Path>,<Name>` for the **Job Name** field in your integration mapping. For example:

`/oracle/apps/ess/financials/commonModules/shared/common/outbound/,PayablesTransactionsExtract.`



ESS Job Definition screen with the Path and Name fields highlighted, and syntax guidance for the Job Name field.

12. Review and provide the parameter list

Scroll to the **Parameters** section of the job definition and review the **Parameter Prompt**, **Data Type**, and **Required** fields. Enter the required parameters in the **Parameter List** field of your integration mapping. For more detail, check the parameter prompts or review existing scheduled process instances for sample values.

Manage Enterprise Scheduler Job Definitions and Job Sets for Financial, Supply Chain Management, and Related Applications ?

Manage Job Definitions Manage List of Values Sources Manage Job Sets

Edit Job Definition Save and Close Cancel

Job Definition [Show More](#)

Display Name Payables Transactions Extract Job Application Name FscmEss

Name PayablesTransactionsExtract ☐ Enable submission from Enterprise Manager

Path /oracle/apps/ess/financials/commonModules/shared/common/outbound/ Job Type BIPJobType

Application Financials Common Module ☐ Bursting report

Description Extracts detailed information for Payables business objects like standard invoice, prepayment, credit memo, and others. Class Name oracle.xdo.service.client.scheduler.BIPJobExecutable

Retries Default Output Format

Job Category Report ID /Financials/Payables/Invoices/PayablesTransactionsExtract.xdo

Timeout Period Priority

☐ Allow Multiple Pending Submissions

☒ Enable submission from Scheduled Processes

PayablesTransactionsExtract: Parameters User Properties

Actions View

Parameter Prompt	Data Type	Page Element	Default Value	Read Only	Required
ATTRIBUTE1	Numeric	List of values	—	☐	☒ Required

ESS Job Definition screen showing parameter prompts, such as ATTRIBUTE1 and ATTRIBUTE2, with their data types.

13. Set job options and the callback URL

In the **Job Options** mapping, enter the required options, such as `ExtractFileType="ALL"`. For **Callback URL**, provide the URL of your OIC app-driven integration so it can receive job completion notifications.

Manage Enterprise Scheduler Job Definitions and Job Sets for Financial, Supply Chain management, and Human Resources

Manage Job Definitions

Manage List of Values Sources

Manage Job Sets

Save and Close

Cancel

Job Definition

Show More

Display Name

Payables Transactions Extract

Name

PayablesTransactionsExtract

Path

/oracle/apps/ess/financials/commonModules/shared/common/outbound/

Application

Financials Common Module

Description

Extracts detailed information for Payables business objects like standard invoice, prepayment, credit memo, and others.

Retries

Job Category

Timeout Period

Job Application Name

FscmEss

Job Type

BIPJobType

Class Name

oracle.xdo.service.client.scheduler.BIPJobExecutable

Default Output Format

Report ID

/Financials/Payables/Invoices/PayablesTransactionsExtract.xdo

Priority

Allow Multiple Pending Submissions

Enable submission from Enterprise Manager

Enable submission from Scheduled Processes

✓

PayablesTransactionsExtract: Parameters

User Properties

Syntax: <Path>,<name>

Parameter Prompt

Data Type

Page Element

Default Value

Read Only

Required

ATTRIBUTE1

Numeric

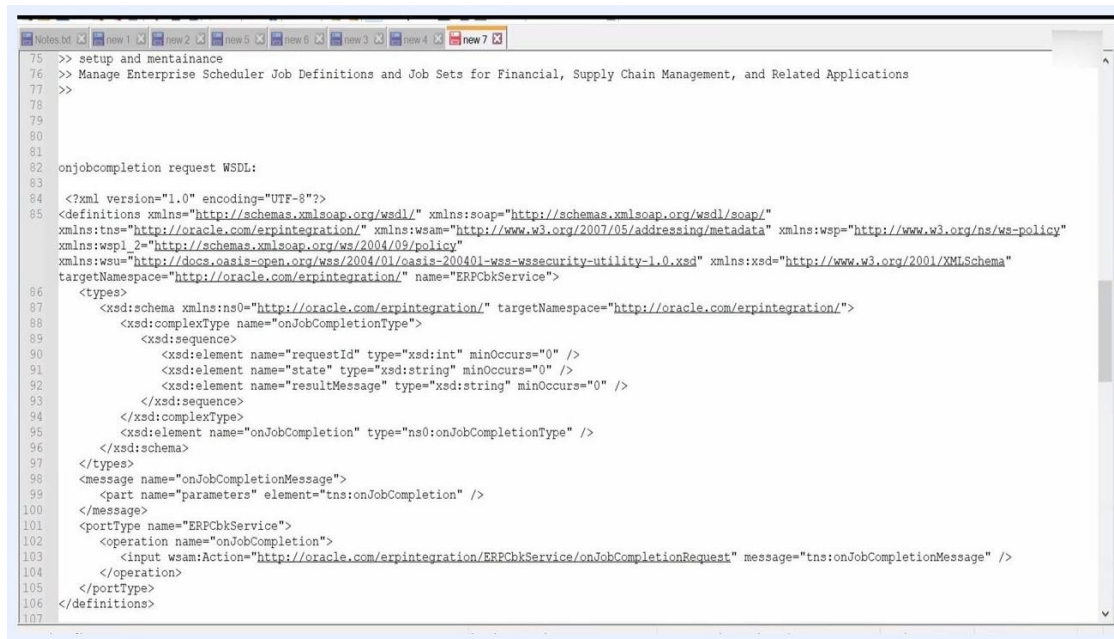
List of values

Required

Oracle Integration Cloud mapping screen showing Job Options set to `ExtractFileType="ALL"`.

14. Configure the notification code

The notification code is made up of two digits. The first digit controls the notification channel: **1** for email, **2** for bell, and **3** for both email and bell. The second digit controls the trigger condition: **0** to send in any case, **1** to send on success, and **2** to send on failure. For example, to send an email only when the job fails, combine **1** (email) with **2** (on failure) for a code of **12**.



```
75 >> setup and maintenance
76 >> Manage Enterprise Scheduler Job Definitions and Job Sets for Financial, Supply Chain Management, and Related Applications
77 >>
78
79
80
81
82 onjobcompletion request WSDL:
83
84 <?xml version="1.0" encoding="UTF-8"?>
85 <definitions xmlns="http://schemas.xmlsoap.org/wsdl/" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
86   xmlns:tns="http://oracle.com/erpintegration/" xmlns:wsam="http://www.w3.org/2007/05/addressing/metadata" xmlns:wsp="http://www.w3.org/ns/ws-policy"
87   xmlns:wsp1_2="http://schemas.xmlsoap.org/ws/2004/09/policy"
88   xmlns:wsm="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
89   targetNamespace="http://oracle.com/erpintegration/" name="ERPCbkService">
90   <types>
91     <xsd:schema xmlns:ns0="http://oracle.com/erpintegration/" targetNamespace="http://oracle.com/erpintegration/">
92       <xsd:complexType name="onJobCompletionType">
93         <xsd:sequence>
94           <xsd:element name="requestId" type="xsd:int" minOccurs="0" />
95           <xsd:element name="state" type="xsd:string" minOccurs="0" />
96           <xsd:element name="resultMessage" type="xsd:string" minOccurs="0" />
97         </xsd:sequence>
98       </xsd:complexType>
99       <xsd:element name="onJobCompletion" type="ns0:onJobCompletionType" />
100     </xsd:schema>
101   </types>
102   <message name="onJobCompletionMessage">
103     <part name="parameters" element="tns:onJobCompletion" />
104   </message>
105   <portType name="ERPCbkService">
106     <operation name="onJobCompletion">
107       <input wsam:Action="http://oracle.com/erpintegration/ERPCbkService/onJobCompletionRequest" message="tns:onJobCompletionMessage" />
108     </operation>
109   </portType>
110 </definitions>
```

Notification code table showing the first- and second-digit values and their meanings.

15. Enter the notification code

In the **Notification Code** field, enter the appropriate value. For example, **30** means email and bell notification sent in any case, while **31** sends the notification only on success.

Name

PayablesTransactionsExtract

Path

/oracle/apps/ess/financials/commonModules/shared/common/outbound/

Application

Financials Common Module

Description

Extracts detailed information for Payables business objects like standard invoice, prepayment, credit memo, and others.

Retries

Job Category

Timeout Period

Data Control Name

Task Flow Name

File Groups

L.XML, O.PDF

☐

Enable time statistic

Database Trace

Job Type

BiPJobType

Bursting report

Class Name

oracle.xdo.service.client.scheduler.BiPJobExecutable

Default Output Format

Report ID

/Financials/Payables/Invoices/PayablesTransactionsExtract.xdo

Priority

Allow Multiple Pending Submissions

Date Increment Procedure

Date Argument Numbers

Enable database optimizer mode

Database Rollback Segment

☒

Enable submission from Scheduled Processes

System Application

PayablesTransactionsExtract: Parameters

User Properties

Actions

View

Detach

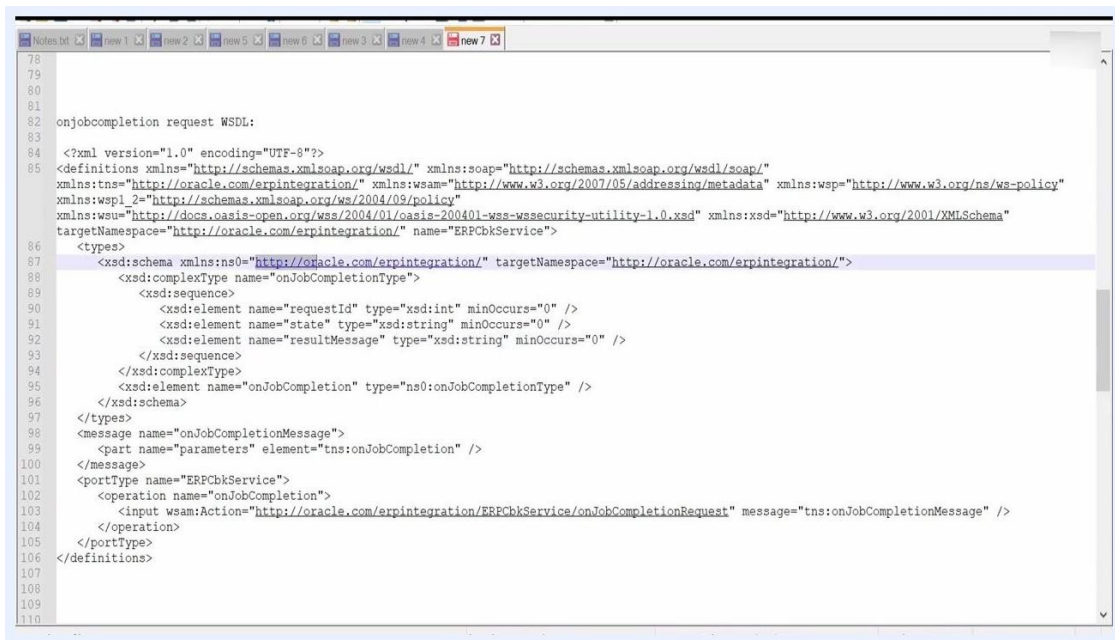
Parameter Prompt	Data Type	Page Element	Default Value	Read Only	Required
ATTRIBUTE1	Numeric	List of values			Required
ATTRIBUTE2	String	Text box			Required

Text editor showing integration steps, including an explanation of the notification code and a reference to the app-driven integration.

Part 3: Build the callback integration

16. Prepare the WSDL file for the callback integration

The WSDL (Web Services Description Language) file defines the structure and operations for the callback SOAP service. It includes a complex type for job completion data ([requestId](#), [state](#), [resultMessage](#)) and defines the [onJobCompletion](#) operation. Make sure the [targetNamespace](#) matches your integration requirements.

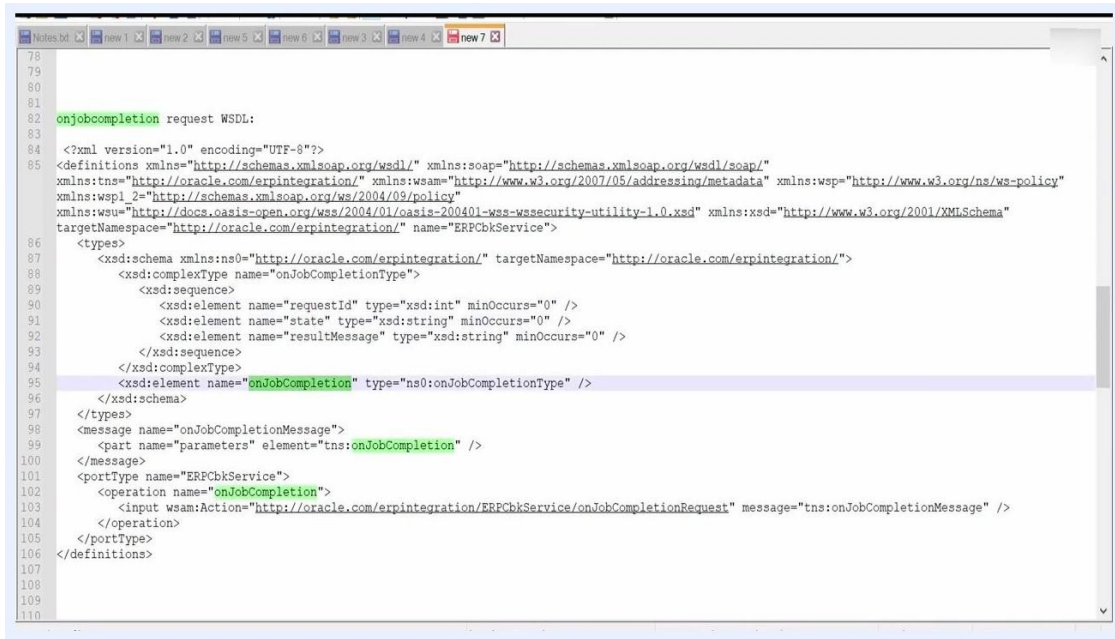


```
78
79
80
81 onjobcompletion request WSDL:
82
83
84 <?xml version="1.0" encoding="UTF-8"?>
85 <definitions xmlns="http://schemas.xmlsoap.org/wsdl/" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:tns="http://oracle.com/erpintegration/" xmlns:wsam="http://www.w3.org/2007/05/addressing/metadata" xmlns:wsp="http://www.w3.org/ns/ws-policy"
  xmlns:wsp1_2="http://schemas.xmlsoap.org/ws/2004/09/policy"
  xmlns:wsu="http://docs.oasis-open.org/ws-s2004/01/oasis-200401-ws-wssecurity-utility-1.0.xsd" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://oracle.com/erpintegration/" name="ERPChkService">
  <types>
  <xsd:schema xmlns:ns0="http://oracle.com/erpintegration/" targetNamespace="http://oracle.com/erpintegration/">
  <xsd:complexType name="onJobCompletionType">
  <xsd:sequence>
  <xsd:element name="requestId" type="xsd:int" minOccurs="0" />
  <xsd:element name="state" type="xsd:string" minOccurs="0" />
  <xsd:element name="resultMessage" type="xsd:string" minOccurs="0" />
  </xsd:sequence>
  </xsd:complexType>
  <xsd:element name="onJobCompletion" type="ns0:onJobCompletionType" />
  </xsd:schema>
  </types>
  <message name="onJobCompletionMessage">
  <part name="parameters" element="tns:onJobCompletion" />
  </message>
  <portType name="ERPChkService">
  <operation name="onJobCompletion">
  <input wsam:Action="http://oracle.com/erpintegration/ERPChkService/onJobCompletionRequest" message="tns:onJobCompletionMessage" />
  </operation>
  </portType>
  </definitions>
107
108
109
110
```

WSDL file structure for the onJobCompletion SOAP operation, showing the XML schema and operation details.

17. Identify the key WSDL elements

Confirm the key elements in your WSDL: [requestId](#), [state](#), [resultMessage](#), and the [onJobCompletion](#) operation.



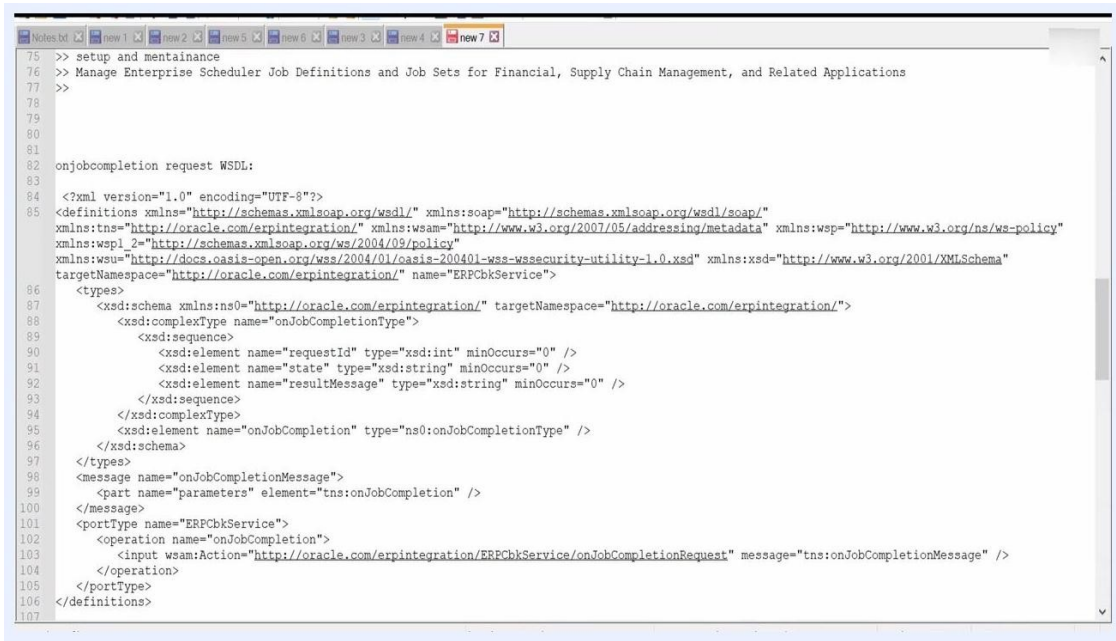
The image shows a screenshot of a text editor window displaying a WSDL file. The editor has a tab bar at the top with tabs labeled 'new 1' through 'new 7'. The main text area shows the WSDL content with line numbers 78 to 110 on the left. Several elements are highlighted in green: the text 'onJobCompletion request WSDL:' on line 82, the 'onJobCompletion' element name on line 95, and the 'onJobCompletion' operation name on line 103. The WSDL content includes standard XML headers, namespace declarations, a schema definition for 'onJobCompletionType' with elements 'requestId', 'state', and 'resultMessage', a message definition 'onJobCompletionMessage', and a port type 'ERPCbkService' with the 'onJobCompletion' operation.

```
78
79
80
81
82 onJobCompletion request WSDL:
83
84
85 <?xml version="1.0" encoding="UTF-8"?>
86 <definitions xmlns="http://schemas.xmlsoap.org/wsdl/" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
87 xmlns:tns="http://oracle.com/erpintegration/" xmlns:wsam="http://www.w3.org/2007/05/addressing/metadata" xmlns:wsp="http://www.w3.org/ns/ws-policy"
88 xmlns:wsp1_2="http://schemas.xmlsoap.org/ws/2004/09/policy"
89 xmlns:wsp1_2="http://schemas.xmlsoap.org/ws/2004/09/policy"
90 xmlns:wsp1_2="http://schemas.xmlsoap.org/ws/2004/09/policy"
91 xmlns:wsp1_2="http://schemas.xmlsoap.org/ws/2004/09/policy"
92 xmlns:wsp1_2="http://schemas.xmlsoap.org/ws/2004/09/policy"
93 xmlns:wsp1_2="http://schemas.xmlsoap.org/ws/2004/09/policy"
94 xmlns:wsp1_2="http://schemas.xmlsoap.org/ws/2004/09/policy"
95 xmlns:wsp1_2="http://schemas.xmlsoap.org/ws/2004/09/policy"
96 xmlns:wsp1_2="http://schemas.xmlsoap.org/ws/2004/09/policy"
97 xmlns:wsp1_2="http://schemas.xmlsoap.org/ws/2004/09/policy"
98 xmlns:wsp1_2="http://schemas.xmlsoap.org/ws/2004/09/policy"
99 xmlns:wsp1_2="http://schemas.xmlsoap.org/ws/2004/09/policy"
100 xmlns:wsp1_2="http://schemas.xmlsoap.org/ws/2004/09/policy"
101 xmlns:wsp1_2="http://schemas.xmlsoap.org/ws/2004/09/policy"
102 xmlns:wsp1_2="http://schemas.xmlsoap.org/ws/2004/09/policy"
103 xmlns:wsp1_2="http://schemas.xmlsoap.org/ws/2004/09/policy"
104 xmlns:wsp1_2="http://schemas.xmlsoap.org/ws/2004/09/policy"
105 xmlns:wsp1_2="http://schemas.xmlsoap.org/ws/2004/09/policy"
106 xmlns:wsp1_2="http://schemas.xmlsoap.org/ws/2004/09/policy"
107 xmlns:wsp1_2="http://schemas.xmlsoap.org/ws/2004/09/policy"
108 xmlns:wsp1_2="http://schemas.xmlsoap.org/ws/2004/09/policy"
109 xmlns:wsp1_2="http://schemas.xmlsoap.org/ws/2004/09/policy"
110 xmlns:wsp1_2="http://schemas.xmlsoap.org/ws/2004/09/policy"
111
```

WSDL file with the onJobCompletion elements highlighted.

18. Locate your integration in the OIC dashboard

Open the Oracle Integration Cloud dashboard and navigate to the **Integrations** section from the left sidebar. Search for your integration by name, then locate the one you want to configure, such as **ERP Bulk Extract Callback TEST01**. Click the integration name to open its configuration details.

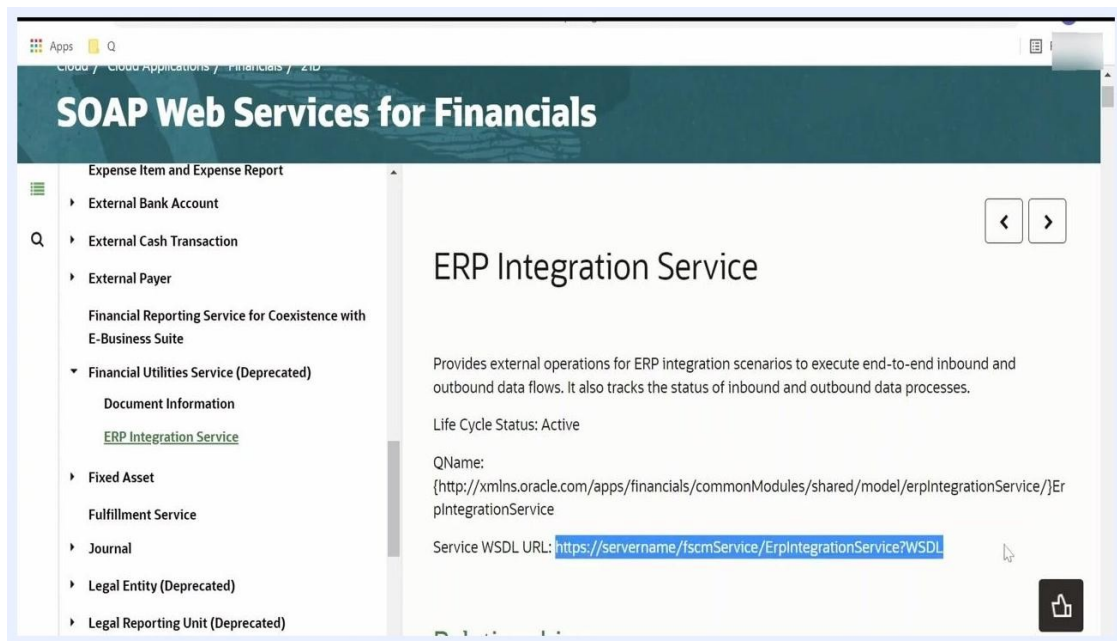


```
75 >> setup and maintenance
76 >> Manage Enterprise Scheduler Job Definitions and Job Sets for Financial, Supply Chain Management, and Related Applications
77 >>
78
79
80
81
82 onjobcompletion request WSDL:
83
84 <?xml version="1.0" encoding="UTF-8"?>
85 <definitions xmlns="http://schemas.xmlsoap.org/wsdl/" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
86   xmlns:tns="http://oracle.com/erpintegration/" xmlns:wsam="http://www.w3.org/2007/05/addressing/metadata" xmlns:wsu="http://www.w3.org/2004/09/policy"
87   xmlns:wspl_2="http://schemas.xmlsoap.org/ws/2004/09/policy"
88   xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
89   targetNamespace="http://oracle.com/erpintegration/" name="ERPCbkService">
90   <types>
91     <xsd:schema xmlns:ns0="http://oracle.com/erpintegration/" targetNamespace="http://oracle.com/erpintegration/">
92       <xsd:complexType name="onJobCompletionType">
93         <xsd:sequence>
94           <xsd:element name="requestId" type="xsd:int" minOccurs="0" />
95           <xsd:element name="state" type="xsd:string" minOccurs="0" />
96           <xsd:element name="resultMessage" type="xsd:string" minOccurs="0" />
97         </xsd:sequence>
98       </xsd:complexType>
99       <xsd:element name="onJobCompletion" type="ns0:onJobCompletionType" />
100     </xsd:schema>
101   </types>
102   <message name="onJobCompletionMessage">
103     <part name="parameters" element="tns:onJobCompletion" />
104   </message>
105   <portType name="ERPCbkService">
106     <operation name="onJobCompletion">
107       <input wsam:Action="http://oracle.com/erpintegration/ERPCbkService/onJobCompletionRequest" message="tns:onJobCompletionMessage" />
108     </operation>
109   </portType>
110 </definitions>
```

Oracle Integration Cloud dashboard showing a list of integrations, with "ERP Bulk Extract Callback TEST01" highlighted.

19. Configure the SOAP endpoint headers

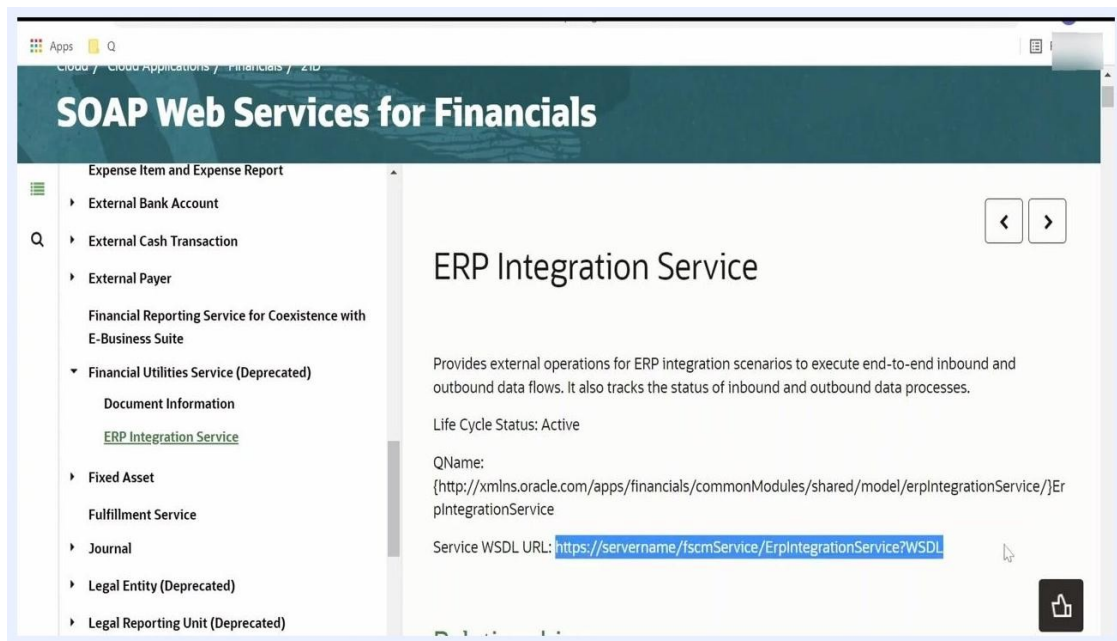
Begin configuring the SOAP endpoint for your callback integration. In the **Configure Headers** step, under **Configure MTOM Attachment Options**, check **Accept attachments in response** and leave **Send attachments in request** unchecked. Under **Configure Headers**, leave the option as **No** unless your application requires custom headers, then click **Next**.



Configure SOAP Endpoint dialog showing the Headers step, with MTOM options and header configuration.

20. Obtain the ERP Integration Service WSDL URL

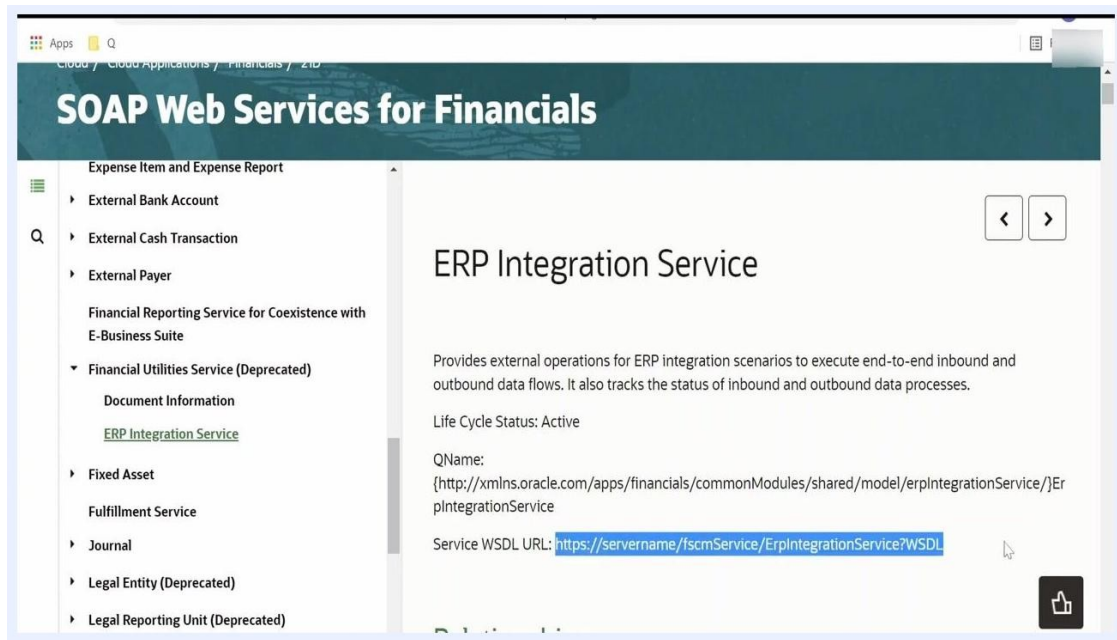
Navigate to the **ERP Integration Service** documentation within Oracle ERP Cloud and locate the **Service WSDL URL** field. Replace [servername](#) in the URL with your actual host name, then copy the full WSDL URL (for example, <https://yourhost/fscmService/ErpIntegrationService?WSDL>). Paste this URL into your browser to verify access and retrieve the WSDL definition.



ERP Integration Service documentation with the WSDL URL highlighted for copying.

21. Map source fields to target fields

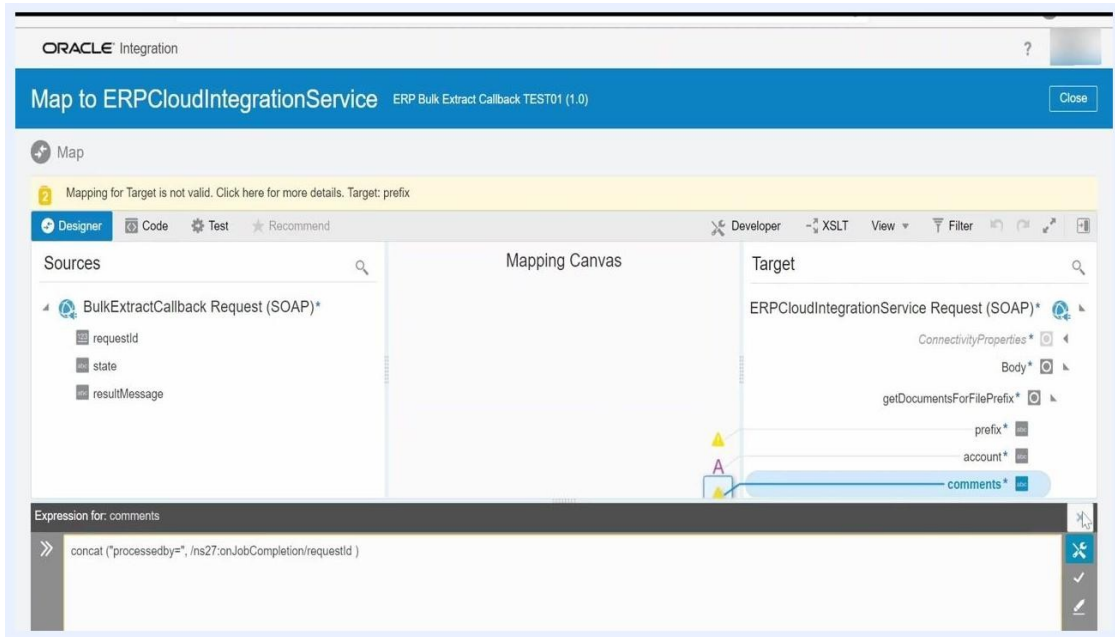
In the mapping designer, map fields from the **BulkExtractCallback Request (SOAP)** source to the **ERPCloudIntegrationService Request (SOAP)** target. The key source fields are `requestId`, `state`, and `resultMessage`, and the target operation is `getDocumentsForFilePrefix`. For the `prefix` field, use a value such as `_<requestId>_VIP_REPORT01`, where `<requestId>` is mapped dynamically. This prefix is used as the file prefix in your UCM system.



Mapping designer showing source fields (`requestId`, `state`, `resultMessage`) and target fields (`prefix`, `account`, `comments`).

22. Set expressions for target fields

For the `comments` field in the target, use an expression to concatenate a label with the request ID, for example: `concat("processedby=", /ns27:onJobCompletion/requestId)`. This ensures traceability in your integration logs and output files.



**Mapping designer showing the expression for the comments field:
`concat("processedby=", /ns27:onJobCompletion/requestId)`.**

23. Review the integration orchestration flow

Confirm that your integration flow includes the following steps, in order:

BulkExtractCallback (trigger), **Map to ERPCloudIntegrationService**, **ERPCloudIntegrationService** (SOAP call), **Map to WriteExtract**, and **WriteExtract** (writes the file to FTP or another target). Confirm the flow is activated and ready for execution.

The screenshot displays the Oracle Integration 'Track Instances' interface. The left sidebar contains navigation links for Integrations, Dashboards, Integrations, Agents, Tracking (selected), and Errors. The main content area shows a table of instances for '1 Instance' with filters for Time Window and Instance Id. The table lists the start time, instance ID, status (Success), and duration. The right panel shows the 'Activity Stream' for the selected instance, detailing the sequence of messages and the final successful completion of the process.

Primary Identifier	Insta...	Status	Business Identifiers	Duration
start Time: 2022-01-24 ERP Bulk Extract TE... 1.0.	2	Success		Trigg... Just now Succ... Just now Dura... 877 milli...

Activity Stream for Instance Id: 29200476

- Message sent to InitiateExtract
- Request: 'Export Bulk Data in ERP Cloud' completed successfully in 810 ms - BUSINESS_SERVICE: ErpIntegrationService
- Message received from InitiateExtract

Message:

```
<nstrgmpr:exportBulkDataResponse xmlns:ns0="http://xmlns.oracle.com/apps/financials/commonModules/shared/model/erpIntegrationService/types/" xmlns:nstrgmpr="http://xmlns.oracle.com/cloud/adaptor/erp/InitiateExtract_REQUEST/types">
  <nstrgmpr:result xmlns="http://xmlns.oracle.com/apps/financials/commonModules/shared/model/erpIntegrationService/types/">5388620</nstrgmpr:result>
</nstrgmpr:exportBulkDataResponse>
```

08:58:17:456 AM Processing completed successfully

Oracle Integration orchestration flow showing all steps from the callback through to the file write.

Part 4: Track and verify the integration

24. Track integration instances and retrieve the request ID

Go to the **Tracking** section in Oracle Integration Cloud and locate the instance of your integration run by **Instance Id** or start time. Review the **Activity Stream** for the instance to see each step's status, and copy the **request ID** generated by Fusion for reference or troubleshooting.

The screenshot shows the Oracle Integration Cloud interface. The left sidebar contains navigation links: Integrations, Dashboards, Integrations, Agents, Tracking (highlighted), and Errors. The main content area is titled 'Track Instances' and features a search bar with filters for 'Time Window : In Retention Period' and 'Instance Id : 29200476'. Below the search bar is a table with columns: Primary Identifier, Instance, Status, Business Identifiers, and Duration. The table lists one instance with the following details:

Primary Identifier	Instance	Status	Business Identifiers	Duration
start Time: 2022-01-24 ERP Bulk Extract TE... 1.0		Succ		877 milli...

The right panel, titled 'Activity Stream', shows the 'Instance Id: 29200476' and a list of messages. The first message is 'Message sent to InitiateExtract' with a status of 'Request: 'Export Bulk Data in ERP Cloud' completed successfully in 810 ms - BUSINESS_SERVICE: ErpIntegrationService'. Below this, a 'Message' box displays XML data:

```
<ns:exportBulkDataResponse>
  xmlns:ns="http://xmlns.oracle.com/apps/financials/commonModules/shared/model/erpIntegrationService/types/"
  xmlns:nstrgpr="http://xmlns.oracle.com/cloud/adapters/erp/InitiateExtract_REQUEST/types"
  <nstrgpr:result>
    xmlns="http://xmlns.oracle.com/apps/financials/commonModules/shared/model/erpIntegrationService/types/">5388628</nstrgpr:result>
</ns:exportBulkDataResponse>
```

At the bottom of the activity stream, a status bar indicates '08:58:17:456 AM Processing completed successfully'.

Track Instances screen showing a successful run, instance ID, and activity stream with request details.

25. (Optional) Search for request status in Fusion

If needed, use the request ID to search for the process status in the **Scheduled Processes** section of Oracle Fusion. This step is optional.

The screenshot displays the 'Track Instances' interface in Oracle Fusion. The left sidebar contains navigation links for Integrations, Dashboards, Integrations, Agents, Tracking, and Errors. The main area shows a search for '1 Instance' with filters for 'Time Window: In Retention Period' and 'Instance Id: 29200476'. A table lists instance details:

Primary Identifier	Insta	Status	Business Identifiers	Duration
start Time: 2022-01-24 ERP Bulk Extract TE... 1.0.	z	● Succ		Trigg... Just now Succ... Just now Dura... 877 milli...

The right panel, 'Activity Stream', shows messages for 'Instance Id: 29200476'. It includes a 'Message sent to InitiateExtract' and a 'Message received from InitiateExtract'. The received message is an XML payload:

```
<ns:exportBulkDataResponse xmlns:ns="http://xmlns.oracle.com/apps/financials/commonModules/shared/model/erpIntegrationService/types/" xmlns:nstrgpr="http://xmlns.oracle.com/cloud/adapters/erp/InitiateExtract_REQUEST/types">
  <nstrgpr:result
    xmlns="http://xmlns.oracle.com/apps/financials/commonModules/shared/model/erpIntegrationService/types">
    $388620
  </nstrgpr:result>
</ns:exportBulkDataResponse>
```

At the bottom, a status bar indicates '08:58:17:456 AM Processing completed successfully'.

Track Instances screen with the request ID and the exportBulkDataResponse XML message.

26. Wait for the callback invocation

After submitting the request, wait a few minutes for Oracle Fusion to call your callback integration. Monitor the integration instance for the callback execution and confirm the process completes successfully.

The screenshot displays the Oracle Integration 'Track Instances' interface. The left sidebar contains navigation links for Integrations, Dashboards, Integrations, Agents, Tracking, and Errors. The main area shows a table of instances with columns: Primary Identifier, Instance, Status, Business Identifiers, and Duration. A single instance is listed with a status of 'Succ' (Success). The right panel, titled 'Activity Stream', shows the message flow for Instance Id: 29200476, including 'Message sent to InitiateExtract' and 'Message received from InitiateExtract'. The bottom status bar indicates 'Processing completed successfully' at 08:58:17.456 AM.

Primary Identifier	Instance	Status	Business Identifiers	Duration
start Time: 2022-01-24 ERP Bulk Extract TE... 1.0.		Succ		Trigg... Just now Succ... Just now Dura... 877 milli...

Activity Stream
Instance Id: 29200476
Ascending

Message sent to InitiateExtract
Request: 'Export Bulk Data in ERP Cloud' completed successfully in 810 ms - BUSINESS_SERVICE: ErpIntegrationService
Message received from InitiateExtract

Message

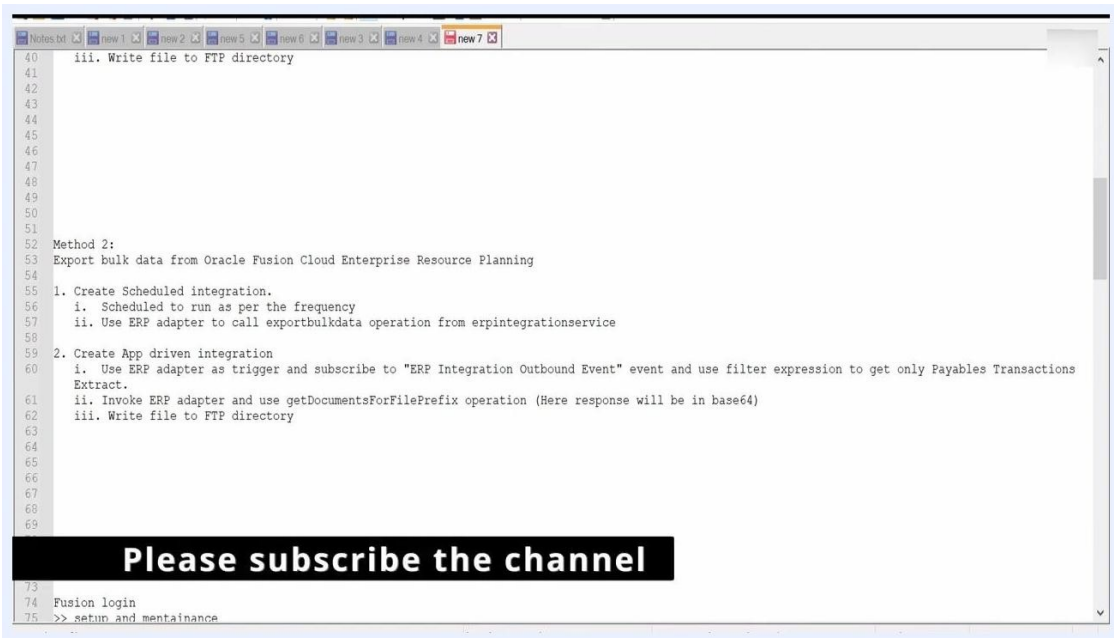
```
<nstrgmpr:exportBulkDataResponse
xmlns:ns0="http://xmlns.oracle.com/apps/financials/commonmodules/shared/model/erpIntegrationService/types/"
xmlns:nstrgmpr="http://xmlns.oracle.com/cloud/adapters/erp/InitiateExtract_REQUEST/types">
  <nstrgmpr:result
xmlns="http://xmlns.oracle.com/apps/financials/commonmodules/shared/model/erpIntegrationService/types/">$388630</nstrgmpr:result>
</nstrgmpr:exportBulkDataResponse>
```

08:58:17.456 AM Processing completed successfully

Track Instances screen showing successful completion and the exportBulkDataResponse message.

27. Search and verify the integration instance using the request ID

Open the Oracle Integration Cloud dashboard and navigate to the **Tracking** section from the left sidebar. In the **Track Instances** screen, enter the request ID you copied earlier (for example, [5388620](#)) into the search bar. Confirm that the matching instance appears, check that the **Status** column shows **Succeeded**, review the **Received** and **Succeeded** timestamps to confirm recent execution, and note the **Duration** of the run.



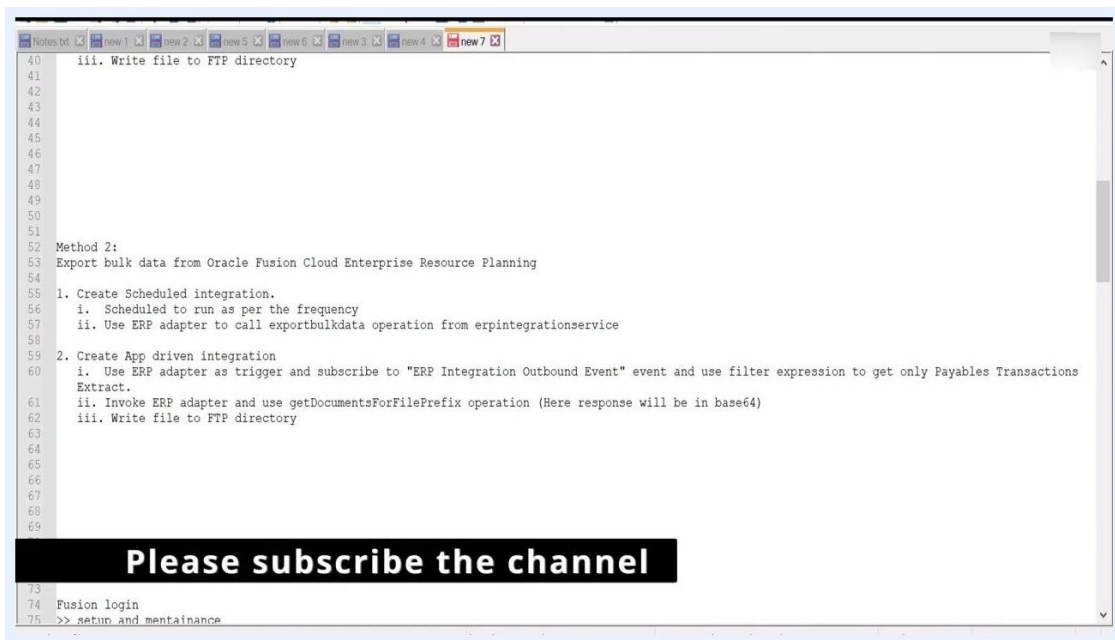
```
40      iii. Write file to FTP directory
41
42
43
44
45
46
47
48
49
50
51
52  Method 2:
53  Export bulk data from Oracle Fusion Cloud Enterprise Resource Planning
54
55  1. Create Scheduled integration.
56     i. Scheduled to run as per the frequency
57     ii. Use ERP adapter to call exportbulkdata operation from erpintegrationservice
58
59  2. Create App driven integration
60     i. Use ERP adapter as trigger and subscribe to "ERP Integration Outbound Event" event and use filter expression to get only Payables Transactions
61        Extract.
62     ii. Invoke ERP adapter and use getDocumentsForFilePrefix operation (Here response will be in base64)
63     iii. Write file to FTP directory
64
65
66
67
68
69
70
71
72
73
74  Fusion login
75  >> setup and maintenance
```

Please subscribe the channel

Oracle Integration Cloud Track Instances screen showing request ID 5388620, succeeded status, and instance details.

28. Review the activity stream for file write completion

Click the instance row to open the **Activity Stream** panel. Verify that the following steps completed successfully: instance created and enqueued for processing, **Trigger BulkExtractCallback** processed, data mapping completed, **Invoke ERPCloudIntegrationService** executed, and **Invoke WriteExtract** executed. Look for a message confirming the file was written to the SFTP server in binary format, note the file name (for example, [PayablesExtract.xml](#)), and confirm all steps show green status indicators.



```
40    iii. Write file to FTP directory
41
42
43
44
45
46
47
48
49
50
51
52 Method 2:
53 Export bulk data from Oracle Fusion Cloud Enterprise Resource Planning
54
55 1. Create Scheduled integration.
56   i. Scheduled to run as per the frequency
57   ii. Use ERP adapter to call exportbulkdata operation from erpintegrationservice
58
59 2. Create App driven integration
60   i. Use ERP adapter as trigger and subscribe to "ERP Integration Outbound Event" event and use filter expression to get only Payables Transactions
61   Extract.
62   ii. Invoke ERP adapter and use getDocumentsForFilePrefix operation (Here response will be in base64)
63   iii. Write file to FTP directory
64
65
66
67
68
69
70
71
72
73
74 Fusion login
75 >> setup and maintenance
```

Please subscribe the channel

Track Instances screen with the Activity Stream showing the successful WriteExtract step and file write confirmation.

Summary

To extract data from Oracle Fusion Cloud using OIC with Method 1, you:

- Copy and verify the ERP Integration Service WSDL URL.
- Configure the SOAP endpoint, including MTOM and header options.
- Map source fields to target fields, using dynamic expressions where needed.
- Review and activate the orchestration flow.
- Track integration runs and use the request ID for monitoring and troubleshooting.
- Wait for the callback to be invoked by Fusion, confirming end-to-end integration.

This completes the setup for consuming the SOAP connection as a trigger and handling the callback integration in Oracle Integration Cloud.

What's next

With the scheduled integration and callback integration both configured and verified, your bulk data extract from Oracle Fusion ERP Cloud will run automatically at the frequency you set, delivering files to your FTP directory each time the export job completes.