

AI Agent Studio demo. Using release 25D.

Agent Team of type “Workflow”. COMPLEX EXAMPLE.

Scan-read PDF from FTP and generate JSON with LLM to create Sales Order.

Introduction/Goal.

We will create this demo from scratch.

We have detailed some steps in previous chapters (OIC integrations and FTP server), but this would be the complete example.

DISCLAIMER.

For learning purposes, we use demo Oracle instances for partners, which renew PODS and passwords frequently, as this demo is time consuming we have done by sections, even a part contains screenshots of 26A release which does not affect the explanations.

Sorry for that, when 26A is fully available, it will be easier to manage this as export-import will be possible and we will provide the complete code.

Flow:

1. Create Agent team and trigger parameters: supplier name and file name. (Create Agent team).
2. Read supplier name and get supplier id. (BO).
3. If to check if supplier found with name provided. (IF node).
4. Read PDF file content from FTP server. (REST tool).
5. Create attachment PDF file to supplier. (BO).
6. Get hash code for next step. We will use OIC integration. (REST tool).
7. Business Object to read attachment.
8. Read file from attachment using Document Processor. (Doc. Processor)
9. Use LLM to extract data. (First LLM) and generate purchase requisition payload to use in next creation steps. (outside the scope).
10. Create Sales Order. (BO reused from template provided by Oracle).

Steps.

Steps in Agent Team. AA_CREATE_SO_FROM_FTP_PDF (1).

Agent team Name: AA_CREATE_SO_FROM_FTP_PDF.

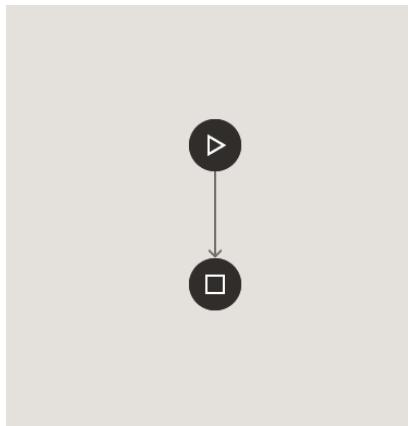
Description: create sales order from PDF file in FTP server.

The screenshot shows the 'AA_CREATE_SO_FROM_FTP_PDF' Agent Team configuration page. The 'Details' tab is selected. The 'Agent Team Name' field contains 'AA_CREATE_SO_FROM_FTP_PDF'. The 'Agent Team Code' field contains 'AACREATESOFROMFTPPDF'. The 'Family' dropdown is set to 'FIN' and the 'Product' dropdown is set to 'Other'. The 'Type' dropdown is set to 'Workflow'. The 'Maximum Interactions' field has a value of '1'. The 'Description' field contains the text 'create sales order from PDF file in FTP server'.

Include new parameters (Triggers) in Agent Team.

The screenshot shows the 'AA_CREATE_SO_FROM_FTP_PDF' Agent Team configuration page with the 'Triggers' tab selected. Under the 'Webhook' section, there are three parameters listed: 'pSupplier' (string), 'pDirectory' (string), and 'pFileName' (string).

Now we have this:



Read supplier and get supplier id. Business Object (2).

Name: AA_GET_SUPP_ID.

Description: get supplier id using supplier name as a parameter.

Note: needed to create and read attachment.

In our instance we have this data to play:

```
{  
    "SupplierId": 300000047414503,  
    "SupplierPartyId": 300000047414501,  
    "Supplier": "Lee Supplies",
```

Edit businessobject - AA_GET_SUPP_ID

Family PRC	Product Supplier
Name AA_GET_SUPP_ID	Code ORA_PRC_SUPPLIER_AAGETSUPPID
Resource Type Monolith resource	Resource Path /fscmRestApi/resources/11.13.18.05/suppliers
Description get supplier id	

Scenario-function.

Edit scenario - AA_GET_SUPP_ID

Name GetSupplierId	Description get supplier id using supplier name as a parameter
Operation Type HTTP GET	Resource Path /fscmRestApi/resources/11.13.18.05/suppliers?q=Supplier='{pSupplier}'&onlyData=true&fields=SupplierId,SupplierPartyId,Supplier,SupplierNumber

Use native authentication Header

Resource PATH:

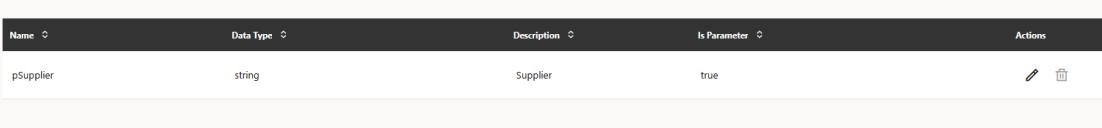
/fscmRestApi/resources/11.13.18.05/suppliers?q=Supplier='{pSupplier}'&onlyData=true&fields=SupplierId,SupplierPartyId,Supplier,SupplierNumber

Name of function: GetSupplierId.

Description: get supplier id using supplier name as a parameter.

Parameter: name of the supplier.

Parameters and Business Object Fields



Name	Data Type	Description	Is Parameter	Actions
pSupplier	string	Supplier	true	

Edit parameter/field

Name
pSupplier

Data Type
String

Description
name of the supplier.

Example.

Add a new example payload

Description
Supplier

Sample Payload
{
 "pSupplier" : "Lee Supplies"
}

Steps in Agent Team. AA_CREATE_SO_FROM_FTP_PDF.

Added previous Business Object to Agent Team as Business Object Function.

Business Object function name: GET_SUPP_ID.

The screenshot shows the 'Agent' interface with a context menu open. The menu items are: Agent, Business Object Function (highlighted with a yellow arrow), If Condition, Document Processor, Send Email, External REST, LLM, Set Variables, Vector DB Reader, and Vector DB Writer. Below the menu, a configuration panel for a 'Business Object Function' is displayed. The panel has fields for Name (GET_SUPP_ID), Code (GETSUPPID), Error Handler, Family (FIN), Product (Other), Business object (AA_GET_SUPP_ID), and Function (GetSupplierId). Under the 'Parameters' section, there is a table with one row: pSupplier (string type). The entire configuration is titled 'GET_SUPP_ID'.

Output.

```
{  
    "$$schema": "http://json-schema.org/draft-07/schema#",  
    "title": "Internal_List1",  
    "type": "object",  
    "properties": {  
        "items": {  
            "type": "array",  
            "items": {  
                "type": "object",  
                "properties": {  
                    "SupplierId": {  
                        "type": "integer"  
                    },  
                    "SupplierPartyId": {  
                        "type": "number"  
                    },  
                    "SupplierNumber": {  
                        "type": "string"  
                    },  
                    "Supplier": {  
                        "type": "string"  
                    }  
                }  
            }  
        }  
    }  
}
```

GET_SUPP_ID

Business Object Function

Parameters

Output

Output Builder Mode

Specification Mode Simple Mode

Properties

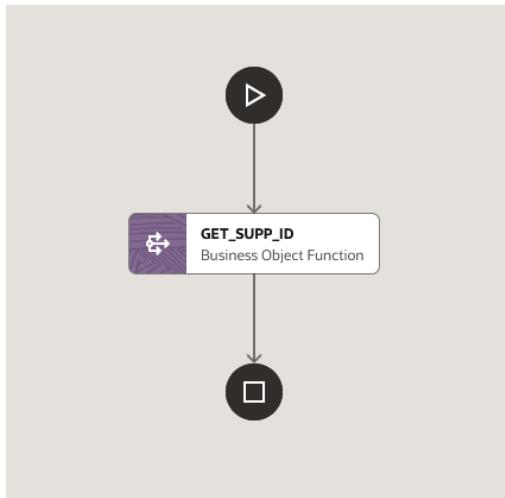
```
{  
  "$schema": "http://json-schema.org/draft-07/schema#",  
  "title": "Internal_List1",  
  "type": "object",  
  "properties": {  
    "items": {  
      "type": "array",  
      "items": {  
        "type": "object",  
        "properties": {  
          "SupplierId": {  
            "type": "integer"  
          }  
        }  
      }  
    }  
  }  
}
```

Parameter and value assigned.

Parameters

Name
pSupplier

Value
{\$context.\$triggers.REST.\$input.pSupplier}



Initial testing.

Payload for initial testing:

```
{
    "pDirectory": "/files/Nordea/AP_Invoices/Outbound/LOG",
    "pFileName": "PO_US165029_5.pdf",
    "pSupplier": "Lee Supplies"
}
```

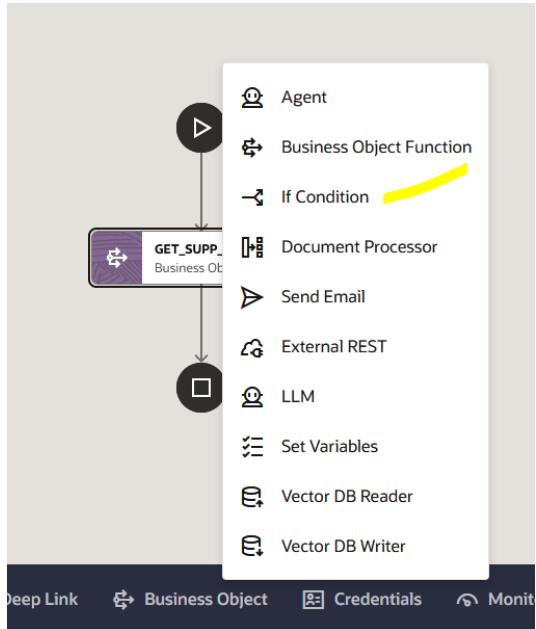
```

{
  "items": [
    {
      "SupplierId": "300000047414503",
      "SupplierPartyId": "300000047414501",
      "Supplier": "Lee Supplies",
      "SupplierNumber": "1252",
      "@context": {
        "key": "300000047414503",
        "headers": {
          "ETag": "ACED0005737200136A6176612E7574696C2E41727261794C6973747881D21D99C7619D0300149000473697A6578700000000777040000007737200116A6176612E6C616E672E496E746567657212E2A0A4F781873802000149000576616C7565787200106A6176612E6C616E672E4E756D62657286AC951D0B94E08B0200007870000001937371007E0002000000457372001B6F7261636C652E6A626F2E646F6D61696E2E4E756C6C56616C75655899C1C58DAABEEB02000149000A6D53514C54797065496478700000000C7371007E0002000000A071007E00077371007E0002000000087371007E0002000000678" }
    }
  ],
  "count": 1,
  "hasMore": false,
  "limit": 25,
  "offset": 0
}

```

Now let's check if supplier is found with IF node (3).

Name: IF_SUPP_FOUND.



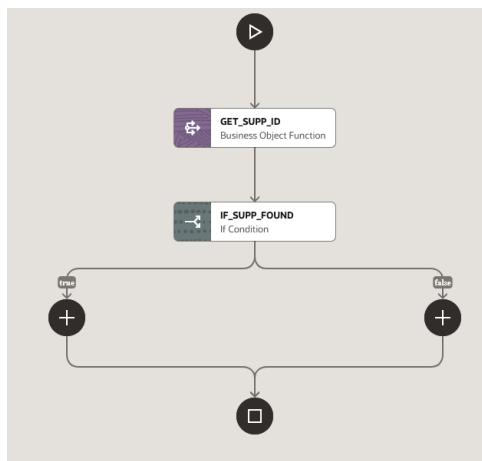
In "If Condition" use "" or null to compare to nulls.

```
{ ${context.$nodes.GETSUPPID.$output.items[0].SupplierId != null} }
```

Other examples:

```
{ ${context.$variables.businessUnitName != ''} }  
{ ${context.$nodes.CREATEORDERSFORPOTOSO.$output.HeaderId != ''} }  
{ ${context.$system.$inputMessage == 'CONVERT_PO_TO_SO'} }
```

Now we have this:

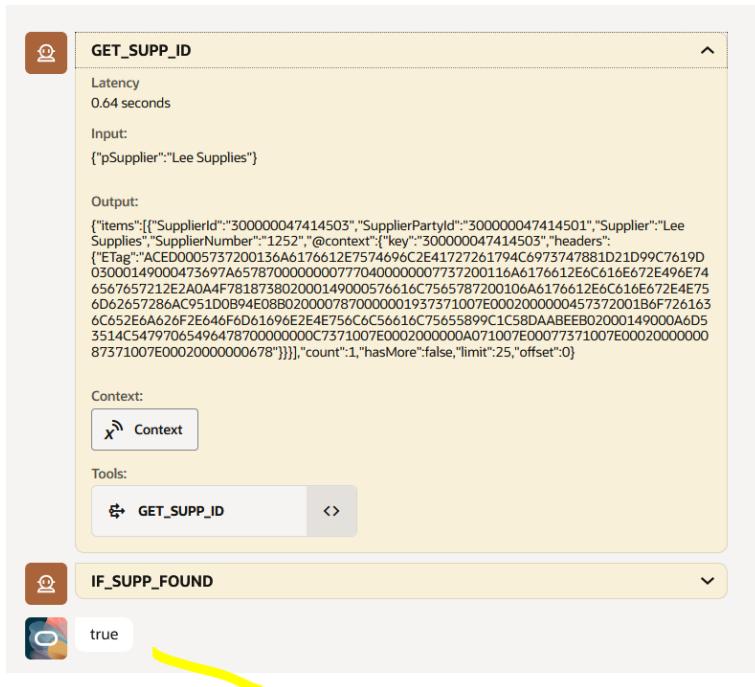


Testing.

Payload for initial testing:

```
{  
    "pDirectory": "/files/Nordea/AP_Invoices/Outbound/LOG",  
    "pFileName": "PO_US165029_0.pdf",  
    "pSupplier": "Lee Supplies"
```

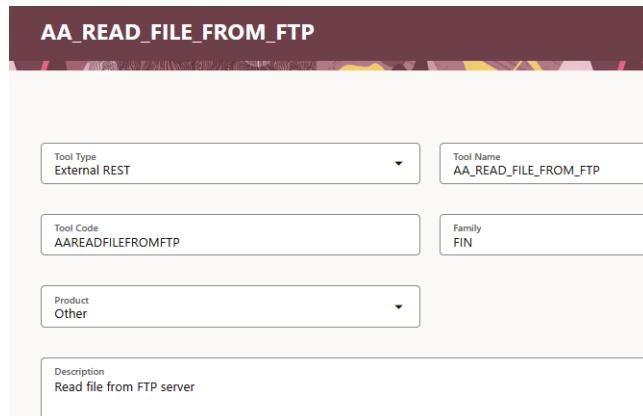
}



Tool of type External REST (4).

Name: AA_READ_FILE_FROM_FTP.

Description: Read file from FTP server.



OIC integration (provided .iar file): XX_AI_FF_GET_FILE_CONTENT (1.0.2)

Purpose: retrieve content of PDF file in base64.

The screenshot displays the configuration and activity stream for the XX_AI_FF_GET_FILE_CONTENT integration. The configuration pane shows a POST request to /ic/api/integration/v1/flows/rest/XX_AI_FF_GET_FILE_CONTENT/1.0 with the URL curl -X POST -H 'Accept: application/json' -H 'Content-Type: application/json' -d '{"pFileName": "PO_US165029_1.pdf", "pDirectory": "/files/Nordex/AP_Invoices/Outbound/LOG"}' 'https://oi3-partners-bot4-saas-yu.integration.ca-montreal-1.ocp.oraclecloud.com/ic/api/integration/v1/flows/rest/XX_AI_FF_GET_FILE_CONTENT/1.0/'. The activity stream on the right shows the execution flow starting with a wire message received by Trigger GetFilesFromFTP, followed by a message started processing, a message received by Data Mapper, data mapping completed, and finally an invoke of GetFiles.

Authorization.

Instance URL.

<https://oic3-partners-bots4saas-yu.integration.ca-montreal-1.ocp.oraclecloud.com>

require human approval

Enable if you want human approval before running the tool.

Authorization Functions

Authorization

Instance URL
isaas-yu.integration.ca-montreal-1.ocp.oraclecloud.com

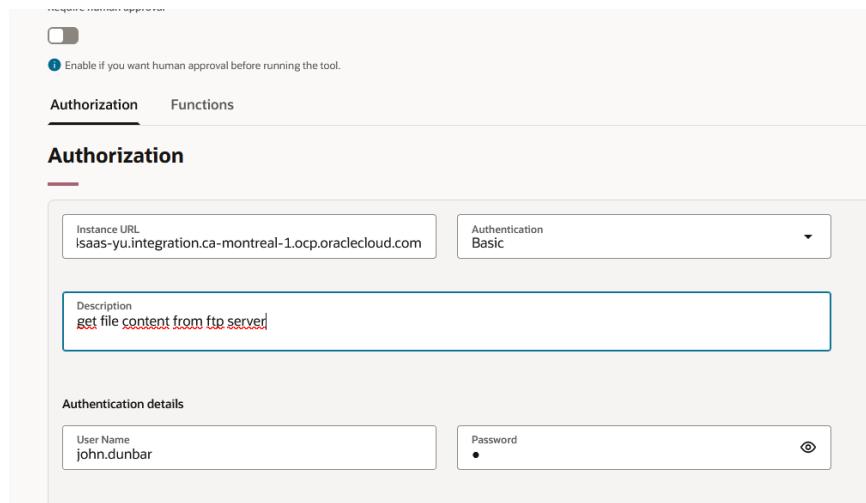
Description
`get file content from ftp server`

Authentication
Basic

Authentication details

User Name
john.dunbar

Password
• 



Function Endpoint.

Name: get_content.

Description: get file content in base64.

Resource path URL:

/ic/api/integration/v1/flows/rest/XX_AI_FF_GET_FILE_CONTE/1.0/

Operation type: POST.

Body template:

```
{  
  "pFileName": "{pFileName}",  
  "pDirectory": "{pDirectory}"  
}
```

Add Endpoint

Name get_content	Operation Type HTTP POST
Resource Path <code>/ic/api/integration/v1/flows/rest/XX_AI_FF_GET_FILE_CONTE/1.0/</code>	
Description get file content	
Body Template { "pFileName": "(pFileName)", "pDirectory": "(pDirectory)" }	

Parameters.

Body Template { "pFileName": "(pFileName)", "pDirectory": "(pDirectory)" }		
Parameters	Sample Queries	Headers
+ Add		
Name pDirectory	Data Type String	
Description pDirectory		
Name pFileName	Data Type String	
Description		

Sample Query.

Parameters Sample Queries Headers

```
Query
{
  "pFileName": "PO_US165029_1.pdf",
  "pDirectory": "/files/XXX/AP_Invoices/Outbound/LOG"
}
```

Description
Payload

Payload for test:

```
{
  "pDirectory": "/files/Nordea/AP_Invoices/Outbound/LOG",
  "pFileName": "PO_US165029_0.pdf",
  "pSupplier": "Lee Supplies"
}
```

Tool is now completed.

Tools Library

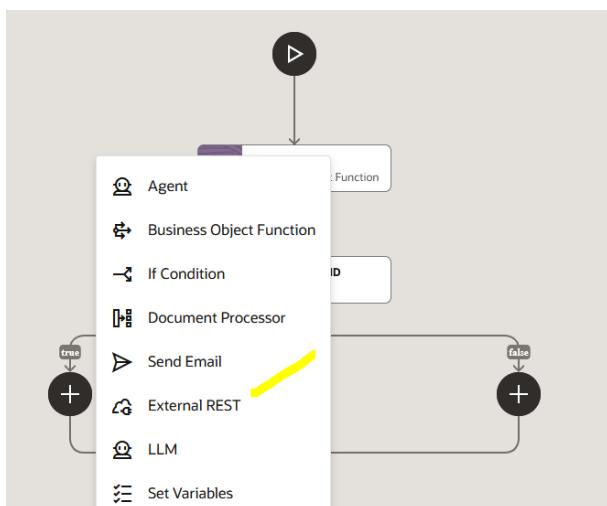
aa_read X Search by tool name, code, or de...

Family Product Tool Type Created By Created On

1 item + Add

Tool Name	Description	Tool Code	Type
AA_READ_FILE_FROM_FTP	get file content	AAREADFILEFROMFTP	External REST

Now in Agent Team add new node of type External REST to include previously created tool.



Name in Agent Team: GET_FILE_CONTENT.

Tool name to use: AA_READ_FILE_FROM_FTP.

GET_FILE_CONTENT
External REST

Name GET_FILE_CONTENT	Code GETFILECONTENT	Error Handler
Family FIN	Product Other	Tool Name AA_READ_FILE_FROM_FTP
Function get_content		

Parameters

pDirectory	string
pFileName	string

Output

Output Builder Mode
 Specification Mode Simple Mode

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "properties": {
    "data_base64": {
      "type": "string"
    }
  }
}
```

Output.

Output

Output Builder Mode
 Specification Mode Simple Mode

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "properties": {
    "data_base64": {
      "type": "string"
    }
  }
}
```

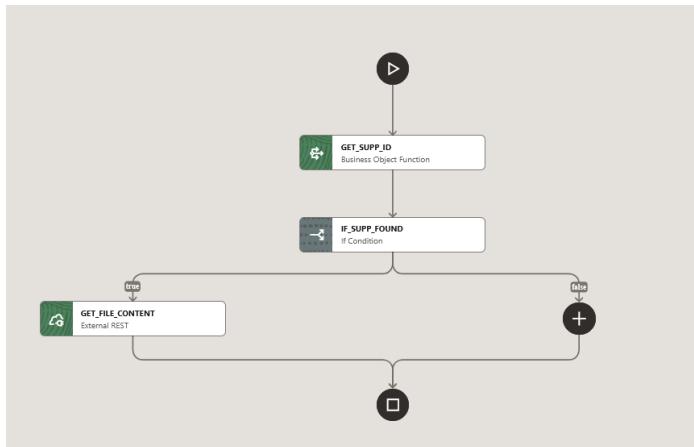
Output payload.

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "properties": {
    "data_base64": {
      "type": "string"
    }
  }
}
```

Parameters in REST tool and values assigned from trigger variables.

Parameters	Parameters																
<table border="1"> <tr> <td>Name pDirectory</td> <td>Name pDirectory</td> </tr> <tr> <td>Value {{\$context.\$triggers.REST.\$input.pDirectory}}</td> <td>Value {{\$context.\$triggers.REST.\$input.pFileName}}</td> </tr> <tr> <td>Type {{ to start seeing context-sensitive expressions.}}</td> <td>Type {{ to start seeing context-sensitive expressions.}}</td> </tr> <tr> <td>Description</td> <td>Description</td> </tr> </table>	Name pDirectory	Name pDirectory	Value {{\$context.\$triggers.REST.\$input.pDirectory}}	Value {{\$context.\$triggers.REST.\$input.pFileName}}	Type {{ to start seeing context-sensitive expressions.}}	Type {{ to start seeing context-sensitive expressions.}}	Description	Description	<table border="1"> <tr> <td>Name pFileName</td> <td>Name pFileName</td> </tr> <tr> <td>Value {{\$context.\$triggers.REST.\$input.pFileName}}</td> <td>Value {{\$context.\$triggers.REST.\$input.pFileName}}</td> </tr> <tr> <td>Type {{ to start seeing context-sensitive expressions.}}</td> <td>Type {{ to start seeing context-sensitive expressions.}}</td> </tr> <tr> <td>Description</td> <td>Description</td> </tr> </table>	Name pFileName	Name pFileName	Value {{\$context.\$triggers.REST.\$input.pFileName}}	Value {{\$context.\$triggers.REST.\$input.pFileName}}	Type {{ to start seeing context-sensitive expressions.}}	Type {{ to start seeing context-sensitive expressions.}}	Description	Description
Name pDirectory	Name pDirectory																
Value {{\$context.\$triggers.REST.\$input.pDirectory}}	Value {{\$context.\$triggers.REST.\$input.pFileName}}																
Type {{ to start seeing context-sensitive expressions.}}	Type {{ to start seeing context-sensitive expressions.}}																
Description	Description																
Name pFileName	Name pFileName																
Value {{\$context.\$triggers.REST.\$input.pFileName}}	Value {{\$context.\$triggers.REST.\$input.pFileName}}																
Type {{ to start seeing context-sensitive expressions.}}	Type {{ to start seeing context-sensitive expressions.}}																
Description	Description																

Now we have.



Payload for test:

```
{
    "pDirectory": "/files/Nordea/AP_Invoices/Outbound/LOG",
    "pFileName": "PO_US165029_0.pdf",
    "pSupplier": "Lee Supplies"
}
```

AA_CREATE_SO_FROM_FTP_PDF

GET_SUPP_ID

IF_SUPP_FOUND

GET_FILE_CONTENT

{
 "data_base64": "JVBERi0xLjYNCiXi48/
 TDQoxIDAgb2JqDQo8PA0KL1R5cGUgL0NhGFsb2cNCi9QYWdlcyAzIDAgUg0KPj4NCmV
 uZG9iag0KMiaWlG9iag0KPDwgL01vZERhdGUgKEQ6MjAyNTAxMDYpIA0KL1R5cGUgL01
 uZm8NCi9Qcm9kdWNlcia0T3JhY2xIIEfUYWx5dGljcyBQdWJsaXNoZXlpDQo+Pg0KZW5k
 b2JqDQozlDAgb2JqDQo8PA0KL1R5cGUgL1BhZ2VzDQovS2IkcyBbDQo1IDAgUg0KMTlg
 MCBSDQoxNCawlFINC10NCi9Db3VudCAzDQo+Pg0KZW5kb2JqDQo0IDAgb2JqDQo8PA0
 KL1Byb2NTZXQgWyAvUERGIC9UZXh0f0NCi9Gb250IDw8IC94ZG9Gb250MSAxOSAwIFI
 gIA0KL0YxIDggMCBSDQovRjlgOSAwlFINC19GMyAxMCAwIFINCj4+DQovWE9iamVjdCA8
 PCANCi9JbTAgNyAwlFINC19JbTEgMTEgMCBSDQo+Pg0KPj4NCmVuZG9iag0KNSAwlG9i
 ag0KCTw8DQoJCS9UeXBICS9QYWdlDQoJCS9QYXJlbnQJMyAwlFINCgkJL01ZGhQm94C
 VsgMCAwIDYxMi4wIDc5Mi4wID0Nyb3BCb3gJWyAwlDAgNjEyLjAgNzkyLjAgXQ
 }

Create attachment PDF file to supplier. Business Object (5).

Goal: creating attachments in suppliers.

Name: AA_ATTACH_SUPP.

Description: attach PDF file to supplier.

Resource path:

The screenshot shows the 'Create a new business object' screen. The 'Name' field is set to 'AA_ATTACH_SUPP'. The 'Description' field contains the text 'attach PDF file to supplier'. The 'Resource Path' is set to '/fscmRestApi/resources/11.13.18.05/suppliers'. The 'Business Object Functions' section lists a single function named 'attach_pdf_to_supplier' with the description 'Attach PDF to supplier.' and the resource path '/fscmRestApi/resources/11.13.18.05/suppliers/{id}/child/attachments'.

Scenario-function.

Name: attach_pdf_to_supplier

Description: Attach PDF to supplier.

Operation: POST.

Resource path.

/fscmRestApi/resources/11.13.18.05/suppliers/{id}/child/attachments

REST API To investigate:

<https://docs.oracle.com/en/cloud/saas/procurement/25d/fapra/op-suppliers-supplierid-child-attachments-post.html>

The screenshot shows the 'Edit scenario - AA_ATTACH_SUPP' screen. The 'Name' field is set to 'attach_pdf_to_supplier'. The 'Description' field contains the text 'Attach PDF to supplier.'. The 'Operation Type' is set to 'HTTP POST'. The 'Resource Path' is set to '/fscmRestApi/resources/11.13.18.05/suppliers/{id}/child/attachments'. The 'Operation Payload' field contains the following JSON code:

```
{ "DatatypeCode": "FILE", "FileName": "{file_name}", "CategoryName": "FROM_SUPPLIER", "UploadedFileContentType": "application/pdf", "ContentRepositoryFileShared": "false", "Title": "title", "Description": "demo", "FileContents": "content" }
```

Payload.

```
{
    "DatatypeCode": "FILE",
    "FileName": "{file_name}",
    "CategoryName": "FROM_SUPPLIER",
    "UploadedFileContentType": "application/pdf",
    "ContentRepositoryFileShared": "false",
    "Title": "{title}",
    "Description": "demo",
    "FileContents": "{content}"
}
```

Parameters and Business Object Fields

Add Parameter	Add Field	Add Field from Specification	
Name	Data Type	Description	Is Parameter
id	string	supplier id	true
file_name	string	file name	true
title	string	file title	true
content	string	file content	true

Parameter values:

- Id: supplier id.
- The rest self-explained.

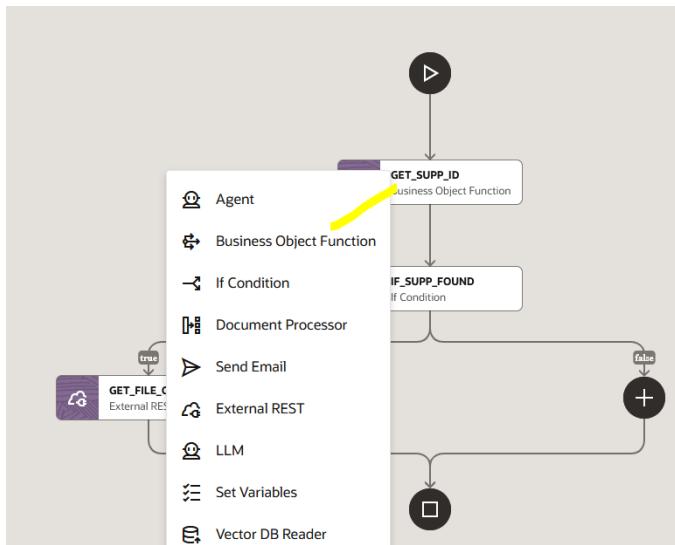
Sample payload.

Example Usages		
+ Add		
Description	Sample Payload	Actions
payload	{"DatatypeCode": "FILE", "FileName": "file_name1", "CategoryName": "FROM_SUPPLIER", "UploadedFileContentType": "application/pdf", "ContentRepositoryFileShared": "false", "Title": "title1", "Description": "demo", "FileContents": "content1"}	

```
{
    "DatatypeCode": "FILE",
    "FileName": "file_name1",
    "CategoryName": "FROM_SUPPLIER",
    "UploadedFileContentType": "application/pdf",
    "ContentRepositoryFileShared": "false",
    "Title": "title1",
    "Description": "demo",
    "FileContents": "content1"
}
```

Steps in Agent Team. AA_CREATE_SO_FROM_FTP_PDF.

Include Business Object Function in Agent Team.



Name: ATTACH_SUPP_FILE.

Using Business Object: AA_ATTACH_SUPP.

The screenshot shows the configuration interface for the business object function "ATTACH_SUPP_FILE".

General Information:

- Name: ATTACH_SUPP_FILE
- Code: ATTACHSUPPFILE
- Error Handler: (empty)
- Family: FIN
- Product: Other
- Function: attach_pdf_to_supplier

Parameters:

Parameter	Type
id	string
file_name	string
title	string
content	string

Value of parameters.

Parameter 1.	Parameter 2.				
<p>Parameters</p> <table border="1"> <tr> <td>Name id</td> <td>Type String</td> </tr> <tr> <td>Value <code> {{\$context.\$nodes.GETSUPPID.\$output.items[0].SupplierId}}</code></td> <td></td> </tr> </table>	Name id	Type String	Value <code> {{\$context.\$nodes.GETSUPPID.\$output.items[0].SupplierId}}</code>		<p>Name title</p> <p>Type String</p> <p>Value <code> {{\$context.\$triggers.REST.\$input.pFileName}}</code></p> <p>Type [] to start seeing context-sensitive expressions.</p>
Name id	Type String				
Value <code> {{\$context.\$nodes.GETSUPPID.\$output.items[0].SupplierId}}</code>					
Parameter 3.	Parameter 4.				
<p>Name title</p> <p>Type String</p> <p>Value <code> {{\$context.\$triggers.REST.\$input.pFileName}}</code></p> <p>Type [] to start seeing context-sensitive expressions.</p>	<p>Name content</p> <p>Type String</p> <p>Value <code> {{\$context.\$nodes.GETFILECONTENT.\$output.data_base64}}</code></p> <p>Type [] to start seeing context-sensitive expressions.</p>				

Values assigned:

```

id      : {{$context.$nodes.GETSUPPID.$output.items[0].SupplierId}}
file_name : {{$context.$triggers.REST.$input.pFileName}}
title   : {{$context.$triggers.REST.$input.pFileName}}
content  : {{$context.$nodes.GETFILECONTENT.$output.data_base64}}

```

Output.

ATTACH_SUPP
Business Object Function

Parameters

name_name	string
title	string
content	string

Output

Output Builder Mode Specification Mode Simple Mode

```

Properties
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "properties": {
    "AttachedDocumentId": {
      "type": "integer"
    }
  }
}

```

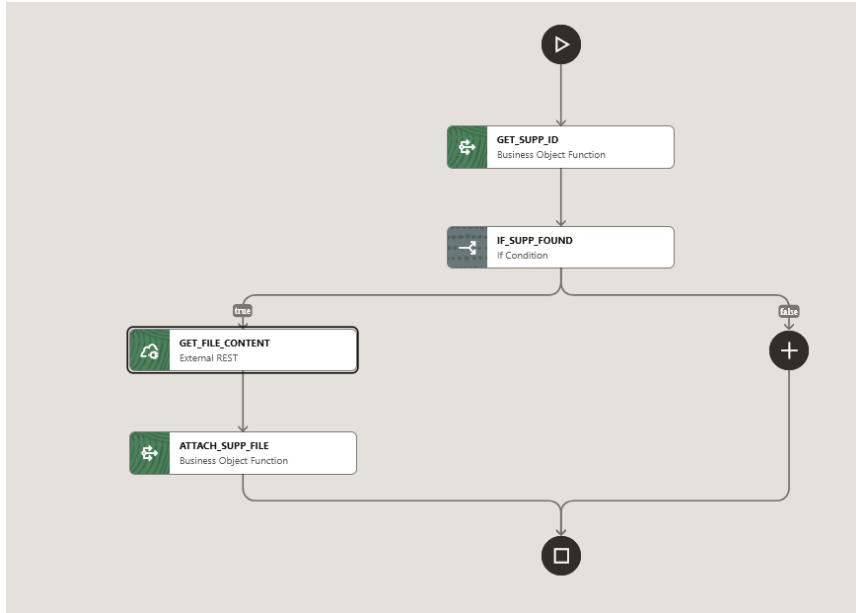
Output payload.

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "properties": {
    "AttachedDocumentId": {

```

```
        "type": "integer"
    }
}
}
```

We now have this:



Note.

Example of payload used in investigation with Postman to attach file to supplier:

```
{  
    "DatatypeCode": "FILE",  
    "FileName": "REST Attachment.txt",  
    "CategoryName": "FROM_SUPPLIER",  
    "UploadedFileContentType": "text/plain", >>>> "application/pdf",  
    "ContentRepositoryFileShared": "false",  
    "Title": "REST Attachment.txt",  
    "Description": "description",  
    "FileContents": "UmFnaGF2YQ=="  
}
```

The screenshot shows the Postman interface with a successful API call. The URL is <https://fa-etit-saasfademo1.ds-fa.oraclepdemos.com:443//fscmRestApi/resources/11.13.18.05/suppliers/300000047414503/child/atta...>. The Body tab is selected, showing the JSON payload. The response status is 201 Created, and the execution time was 9.85 s with 2 parallel tests.

```
1 {  
2     "DatatypeCode": "FILE",  
3     "FileName": "REST Attachment.txt",  
4     "CategoryName": "FROM_SUPPLIER",  
5     "UploadedFileContentType": "text/plain",  
6     "ContentRepositoryFileShared": "false",  
7     "Title": "REST Attachment.txt",  
8     "Description": "description",  
9     "FileContents": "UmFnaGF2YQ=="  
10 }
```

Body Cookies Headers (28) Test Results

201 Created 9.85 s 2

{ } JSON ▾ ▶ Preview Visualize

```
1 {  
2     "AttachedDocumentId": 300000320196992,  
3     "LastUpdateDate": "2025-11-23T10:31:29.228+00:00",  
4     "LastUpdatedBy": "JOHN.DUNBAR",  
5     "DatatypeCode": "FILE",  
6     "FileName": "REST Attachment.txt",  
7     "DmFolderPath": null,  
8     "DmDocumentId": "UCMFA07098912",  
9     "DmVersionNumber": "7151053",  
10    "Url": null,  
11    "CategoryName": "FROM_SUPPLIER",
```

Testing.

Payload.

```
{  
    "pDirectory": "/files/Nordea/AP_Invoices/Outbound/LOG",  
    "pFileName": "PO_US165029_5.pdf",  
    "pSupplier": "Lee Supplies"  
}
```

The screenshot shows a Fusion application interface. At the top, there is a code editor containing a JSON payload:

```
{  
    "pDirectory": "/files/Nordea/AP_Invoices/Outbound/LOG",  
    "pFileName": "PO_US165029_5.pdf",  
    "pSupplier": "Lee Supplies"  
}
```

Below the code editor, there are two sections: "Context:" and "Tools:". The "Context:" section contains a single item: "Context". The "Tools:" section has three items: "GET_FILE_CONTENT" (which is selected), "File", and "Copy".

At the bottom of the interface, there is a large text area displaying the generated file content. The content is a long string of binary data represented as hex values (e.g., 0x45, 0x4c, etc.), which is the PDF file itself.

Testing 2.

Payload.

```
{
    "pDirectory": "/files/Nordea/AP_Invoices/Outbound/LOG",
    "pFileName": "PO_US165029_3.pdf",
    "pSupplier": "Lee Supplies"
}
```

The screenshot shows a web-based application interface. At the top, there is a large JSON object representing a file upload or document metadata. Below this, there are two smaller preview boxes for attachments:

- Attachment 1:** A PDF file named "PO_US165029_5.pdf". The content of the PDF is a long string of encoded data, likely a self-contained exploit or proof-of-concept code.
- Attachment 2:** A smaller file named "PO_US165029_5.pdf" which is identical to the first one.

Notice in SaaS the attachment file.

The screenshot shows the Oracle Fusion Applications Supplier Management screen. The supplier record for "Lee Supplies" is displayed. In the "Attachments" section of the "General" tab, there are two attachments listed:

- PO_US165029_5.pdf (2 more...)

In the next step we will get the hash code to get the attachment content using our OIC integration, the purpose of this step is to read the PDF content in base64 using document processor.

This OIC integration will be needed in further steps to retrieve the file content from attachment. The goal is to get the hash code needed.

The screenshot shows the 'XX_GET_SUPP_HASH_ATTACH (1.0.o)' integration in the OIC interface. The 'Request' section shows a POST request to '/ic/api/integration/v1/flows/rest/XX_GET_SUPP_HASH_ATTACH/1.0/'. The 'Body' tab is selected, showing the following JSON payload:

```
1 {  
2   "pSupplierId": 300000047414503,  
3   "pFileName": "PO_US165029_5.pdf"  
4 }
```

The 'Response' section shows a 200 OK status with the following JSON content:

```
1 {  
2   "HashCode" : "0002000000EACED00057708000110D944854CB2000000EACED00057708000110D944854CB1"  
3 }
```

We will need a new tool for REST External.

Get hash code. Tool of type REST External (6).

Name: AA_GET_HASH_FOR_SUPP_ATTACH.

Description: get hash for supplier attachment.

Note: needed to get the file content of attachment. There is no direct access to this value, and OIC integration has been developed and is provided.

Authorization.

Instance URL:

<https://oic3-partners-bots4saas-yu.integration.ca-montreal-1.ocp.oraclecloud.com>

The screenshot shows the configuration page for the tool 'AA_GET_HASH_FOR_SUPP_ATTACH'. The top section contains fields for 'Tool Type' (External REST), 'Tool Name' (AA_GET_HASH_FOR_SUPP_ATTACH), 'Tool Code' (AAGETHASHFORSUPPATTACH), 'Family' (PRC), and 'Product' (Supplier). Below these are 'Description' and 'Require human approval' checkboxes. The 'Description' field contains the text 'get hash for supplier attachment'. The 'Require human approval' checkbox is unchecked. The bottom section is titled 'Authorization' and includes fields for 'Instance URL' (https://oic3-partners-bots4saas-yu.integration.ca-montreal-1.ocp.oraclecloud.com) and 'Authentication' (Basic). A note below says 'Description instance oic'.

Endpoint for Function.

Name: get_hash_for_supp_attachment

Operation: POST.

Description: get hash for supplier attachment.

Resource path:

/ic/api/integration/v1/flows/rest/XX_GET_SUPP_HASH_ATTACH/1.0/

OIC integration provided:

The screenshot shows the MuleSoft Anypoint Studio interface. At the top, it says 'XX_GET_SUPP_HASH_ATTACH (1.0.0)'. Below that is a 'Configure and run' section with a 'Run' button. The 'Request' tab is selected, showing a POST request to '/ic/api/integration/v1/flows/rest/XX_GET_SUPP_HASH_ATTACH/1.0/'. The body is set to 'Text' and contains the following JSON:

```

1+ {
2+   "pSupplierId": "3800000847414503",
3+   "pFileName": "invoice_sample_1.pdf"
4+

```

The 'Response' tab shows a 200 OK status with an instance ID: 'phexeyEfcms9292nAbw'. The body contains a single key-value pair:

```

1+ {
2+   "HashCode" : "000200000000EACED0005770800011009448C858100000000EACED0005770800011009448C8580"
3+

```

Description: get hash for supplier attachment.

Add Endpoint

The 'Add Endpoint' screen shows the following details:

- Name:** get_hash_for_supp_attachment
- Operation Type:** HTTP POST
- Resource Path:** /ic/api/integration/v1/flows/rest/XX_GET_SUPP_HASH_ATTACH/1.0/
- Description:** get hash for supplier attachment
- Body Template:**

```
{
  "pSupplierId": {pSupplierId},
  "pFileName": "{pFileName}"
}
```
- Parameters:**

Name	Description	Data Type
pFileName	pFileName	String
pSupplierId	pSupplierId	String

Body template:

```
{
  "pSupplierId": {pSupplierId},
  "pFileName": "{pFileName}"
}
```

Require human approval

Enable if you want human approval before running the tool.

Authorization Functions

+ Add

Name: get_hash_for_supp_attachment Operation Type: POST

Sample queries.

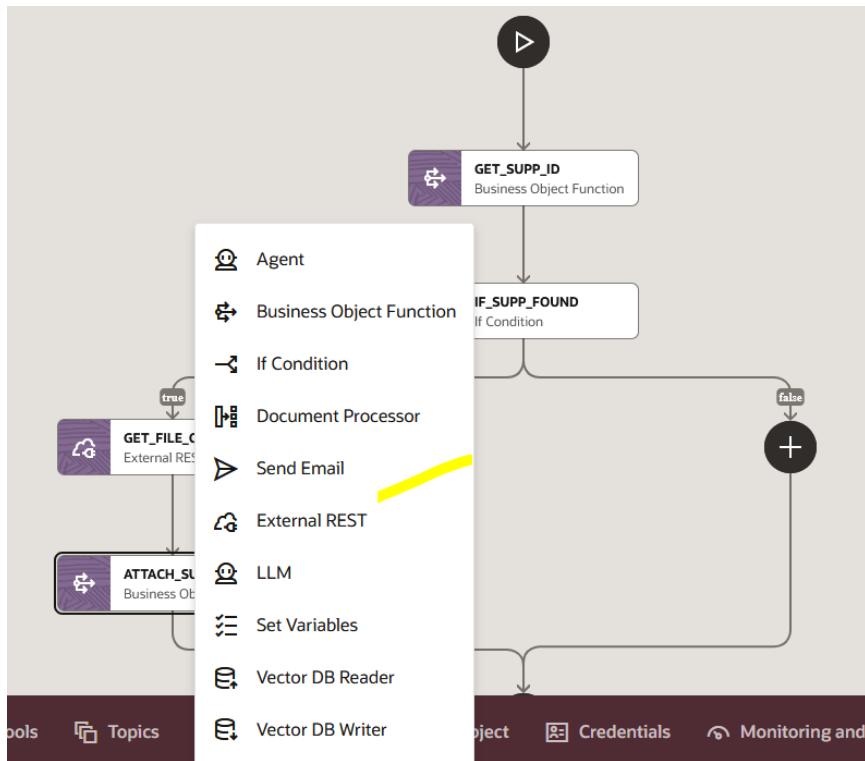
```
{ "pSupplierId": 300000047414503, "pFileName": "PO_US165029_5.pdf" }
```

Parameters Sample Queries Headers

+ Add

Query	Description
{ "pSupplierId": 300000047414503, "pFileName": "PO_US165029_5.pdf" }	payload

Now we will add the node to our Agent Team.



Name in Agent team: AT_GET_HASH_FOR_SUPP_ATTACH.

Description: get hash for supplier attachment.

Tool name: AT_GET_HASH_FOR_SUPP_ATTACH.

AT_GET_HASH_FOR_SUPP_ATTACH			
External REST			
Name	AT_GET_HASH_FOR_SUPP_ATTACH	Type	External REST
Family	FIN	Product	Other
Function	Tool Name AA_GET_HASH_FOR_SUPP_ATTACH		
get_hash_for_supp_attachment			
Parameters			
pFileName	string		
pSupplierId	string		

Parameter values.

Parameters

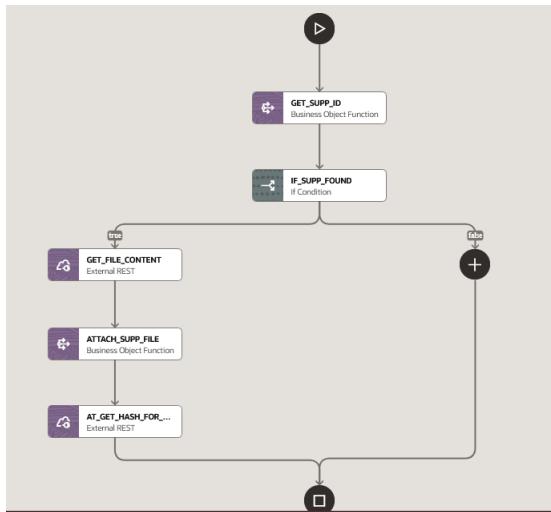
Name pFileName
Value {{context.\$triggers.REST.\$input.pFileName}}
Description

Output.

Output
Output Builder Mode
<input checked="" type="radio"/> Specification Mode <input type="radio"/> Simple Mode
Properties
{ "\$schema": "http://json-schema.org/draft-04/schema#", "type": "object", "properties": { "HashCode": { "type": "string" } } }

```
{  
  "$schema": "http://json-schema.org/draft-04/schema#",  
  "type": "object",  
  "properties": {  
    "HashCode": {  
      "type": "string"  
    }  
  }  
}
```

Now we have this:



Testing.

Payload.

```
{  
    "pDirectory": "/files/Nordea/AP_Invoices/Outbound/LOG",  
    "pFileName": "PO_US165029_3.pdf",  
    "pSupplier": "Lee Supplies"  
}
```

AT_GET_HASH_FOR_SUPP_ATTACH

Latency
3.46 seconds

Input:
{"pFileName": "PO_US165029_3.pdf", "pSupplierId": "300000047414503"}

Output:
{"HashCode": "0002000000EACED00057708000110D9449D0E640000000EACED00057708000110D9449D0E63"}

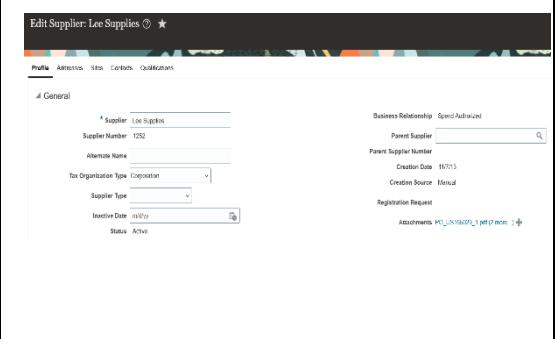
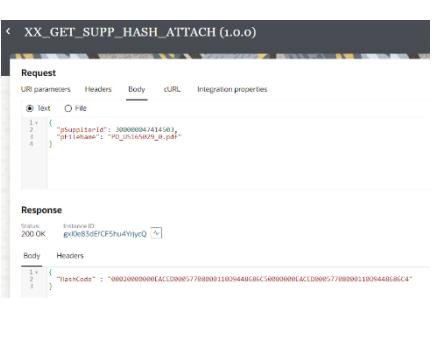
Context:
Context

Tools:
AT_GET_HASH_FOR_SUPP_ATTACH

HashCode:
0002000000EACED00057708000110D9449D0E640000000EACED00057708000110D9449D0E63

Read file from attachment in supplier. Business Object (8).

Note: we are playing with this data for testing.

	
Request. <pre>{ "pSupplierId": 300000047414503, "pFileName": "PO_US165029_0.pdf" }</pre>	Response. <pre>{ "HashCode": "0002000000EACED00057708000110D9448686C4" }</pre>

Business Object Name: AA_GET_SUPPLIER_ATTACHMENT.

Description: Retrieve the supplier attachment content.

Create a new business object

Family PRC	Product Supplier
Name AA_GET_SUPPLIER_ATTACHMENT	Code ORA_PRC_SUPPLIER_AAGETSUPPLIERATTACHMENT
Resource Type Monolith resource	Resource Path /fscmRestApi/resources/11.13.18.05/suppliers
Description Retrieve the supplier attachment content	

URL from Oracle reference doc.

/fscmRestApi/resources/11.13.18.05/suppliers/{supplierUniqID}/child/attachments/{attachmentsUniqID}

Sample URL to get the attachment content.

<https://fa-zzzz-saasfademo1.ds-fa.oracle.demodos.com:443/fscmRestApi/resources/11.13.18.05/suppliers/300000317626637/child/attachments/0002000000EACED00057708000110D9445D78CC0000000EACED00057708000110D9445D78CE/enclosure/FileContents>

Function-scenario.

Name: get_supp_attachment_content

Description: get supplier attachment content.

Add a new scenario

Name get_supp_attachment_content	Description get supplier attachment content
Operation Type HTTP GET	Resource Path /fscmRestApi/resources/11.13.18.05/suppliers/{pSupplierId}/child/attachments/{pAttachmentsUniqID}/enclosure/FileContents

Resource Path:

/fscmRestApi/resources/11.13.18.05/suppliers/{pSupplierId}/child/attachments/{pAttachmentsUniqID}/enclosure/FileContents

Parameters and example usage.

Parameters and Business Object Fields

Add Parameter	Add Field	Add Field from Specification		
Name	Data Type	Description	Is Parameter	Actions
pSupplierId	string	Supplier id	true	
pAttachmentsUniqID	string	Attachment Unique ID	true	

Example Usages

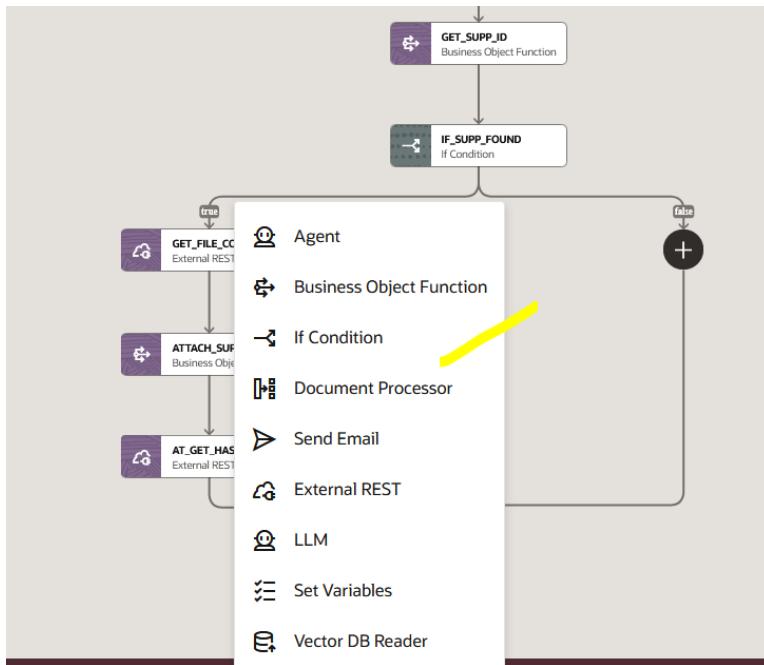
+ Add		
Description	Sample Payload	Actions
Payload	{ "pSupplierId": "0001", "pAttachmentsUniqID": "ABCD" }	

Payload for example usage.

```
{
  "pSupplierId": "0001",
  "pAttachmentsUniqID": "00020000000EACED00057708000110D9448686C50000000EACED00057708000110D9448686C4"
}
```

Add “Document Processor” node using previous Business Object. (8).

Use previous business object as basis for this section.



Name: ReadPDF.

Select Business Object: AA_GET_SUPPLIER_ATTACHMENT.

ReadPDF		
Document Processor		
Name ReadPDF	Code READPDF	Type Document Processor
Family PRC	Product Supplier	Business object AA_GET_SUPPLIER_ATTACHMENT
get_supp_attachment_content		

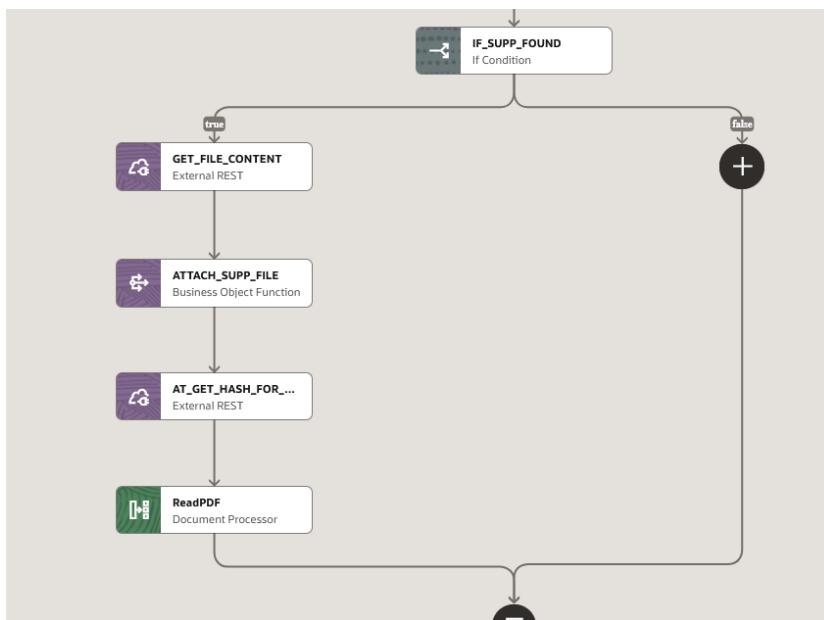
Assign parameters.

Parameters		
Name pSupplierId	Type String	
Value {{\\$context.\$variables.varSupplierId}}		

Name	pAttachmentsUniqID
Value	<code>>{{\$context.\$nodes.ATGETHASHFORSUPPATTACH.\$output.HashCode}}</code>
i Type {{ to start seeing context-sensitive expressions.	

`>{{$context.$nodes.ATGETHASHFORSUPPATTACH.$output.HashCode}}`

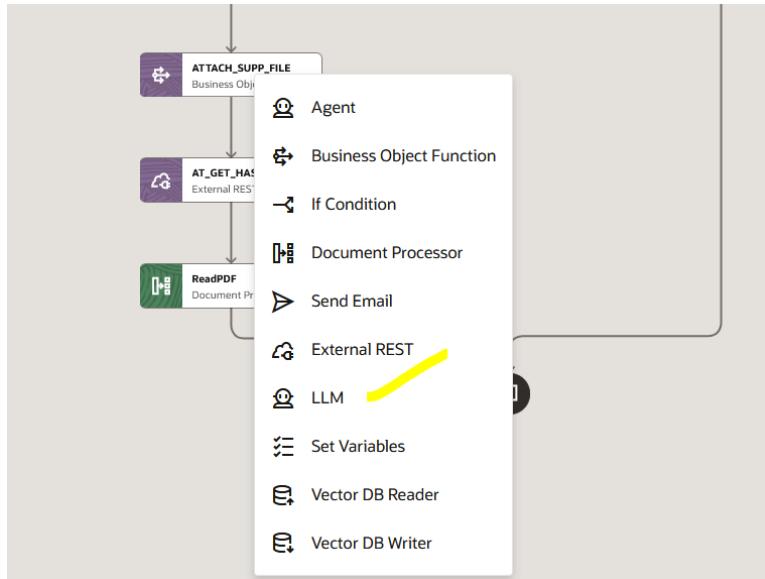
Now our canvas looks like this.



Steps in Agent Team. AA CREATE SO FROM FTP PDF.

First LLM (9).

Name: DEMO_LLM_1.



Prompt text in LLM to create Sales Order adapted from template “Purchase Order to Sales Order Converter”.

Note: we have made these adaptations in characters as in 25D showed error.

- Parenthesis.
- Removed references that do not exist in the workflow.
- These quotes: `null`.
- The character “&”.

You are an **Order Extraction and Slot-Filling AI** with fuzzy matching and industrial PO know-how. Transform *any* Purchase Order PO text into a valid **Sales Order JSON** for Oracle Order Management Cloud **Sales Orders for Order Hub** create endpoint.

```
# OBJECTIVE

1. Ingest raw PO text any layout.
2. Fuzzy-extract standard PO attributes.
3. Populate a predefined JSON with "p*" variables and line objects.
4. Defaults: missing dates → {{Date.now}} ISO 8601, others → null.
5. Build a compliant order payload "header + lines + charges" per Oracle REST model; if a field is
NOT confidently mapped, add it to "unmappedAttributes".

<--BEGIN PURCHASE ORDER TEXT-->
{{$context.$nodes.READMEPDF.$output.text}}
<--END PURCHASE ORDER TEXT-->

# HINT USAGE

• Treat each hint as a *candidate* PO→SO mapping line.
• If "approvedFlag='Y'" and distance < 0.12, prefer the hint; else use your fuzzy logic.
• Anything still ambiguous → add to "unmappedAttributes", cap poFragment2 = 60 chars.

# PRE-PROCESSING *MAKE THE TEXT MACHINE-FRIENDLY*

• Strip boilerplate/watermarks/ToC/terms *e.g., "Confidential", "Internal", page banners, multi-page
legal T\andCs*.
• Normalize whitespace; join hard-wrapped lines; de-hyphenate split tokens *"FIT-\nTread-2950" →
"FIT-Tread-2950"*. 
• Canonicalize numbers/currency *strip ",", "$", handle "7,252.00", "2900.00"*. 
```

- Parse dates in formats such as "July 18th 2025", "18-July-25", "01-AUG-2025", "7/27/2025", "9/1/2025".
- Treat tables and free-form equally; infer columns by repeated patterns.

```
# INTELLIGENT MATCHING RULES *FUZZY EQUIVALENTS*
```

```
* **PO Number** ← "PO #", "P.O. No", "PO No", "Purchase Order", "Receipt", "Receipt No.", "Order <num>", "Standard Purchase Order <num>".
* **Buyer** ← "Buyer", "Requisitioner", "Ordered By", "Contact", "Requester".
* **Bill To** ← "Bill To", "From", "Billed To", "Remit To".
* **Ship To** ← "Ship To", "Deliver To", "Delivery address", "Destination".
* **Quantity** ← "Qty", "QTY", "Amount" *when paired with UOM/price*.
* **Unit Price** ← "Unit Price", "Price Each", "List Price", "Unit".
* **Item/SKU** ← "Item #", "Part Number", "SKU", "Product Code".
* **Payment Terms** ← "Payment Terms", "Terms".
* **Ship Via / Carrier** ← "Ship Method", "Carrier", "Method".
* **Priority** ← "Shipment Priority", "Priority".
* **FreightTerms** ← "Freight", "Prepaid freight", "Postpaid freight".
* **FOBPoint** ← "Free on Board points".
* **Requested / Promised / Delivery Date** ← "Request Date", "Requested", "Promised", "Delivery Date".
    **Guardrail:** **BuyingPartyName is the customer, not the supplier**; never take supplier block as the buyer.
    **Guardrail:** **BuyingPartyName, if it looks like an address then it is not correct element. Typically address from the first text block may be followed by BuyingPartyName **; never take address as the BuyingPartyName.
```

```
# ADDRESS HEURISTICS
```

- If a **Delivery address** block exists at header level, prefer it for **Ship To**.
- Else use an explicit **Ship To** block.
- If multiple logistics addresses *e.g., warehouse list, shipping instructions with many sites*, **don't** treat the list as the ship-to; use the order's primary Delivery/Ship-To and record the rest in "unmappedAttributes" with a note like ""multipleCandidateShipToAddresses"".
- **Bill To** often equals the company header address; confirm via labels **Bill To**, "From", company letterhead*.
- City casing must be preserved exactly for "pShipToCity".
- County may follow City in the address block.
- If State is given in complete name then use as-is , never change to short abbreviation unless stated. i.e. if "New Hampshire" is given as state then donot change to "NH" explicitly , use is as "New Hampshire".

```
# SALES ORDER REST API CONFORMITY *STRUCTURE and KEY CHILD OBJECTS*
```

Target the **Sales Orders for Order Hub** create payload and children *lines, charges, chargeComponents*. For payments/terms, use the Payments child when available. *See "Create Sales Orders" + Lines/Charges + Charge Components + Payments.*

```
# ATTRIBUTE MAPPING
```

```
## Top-Level *Header*
```

```
"""
- pSourceTransactionNumber ← *PO Number OR Order Number OR Receipt* +{{Date.now**}}, truncate ≤ 28 chars
- pSourceTransactionId ← same as pSourceTransactionNumber, truncate ≤ 28 chars
- pBusinessUnitName ← {{$context.$variables.businessUnitName}}
- pBuyingPartyName ← Customer entity placing the order *Bill To/Buyer*
- pBuyingPartyContactName ← Buyer/Requisitioner/Contact
- pCurrentDate ← {{$context.$system.$currentDate}} in ISO *if absent* , in ISO Format only like "YYYY-MM-DD".
- pSourceTransactionSystem ← Source System if present, else default "GPR"
- pPartialShipAllowedFlag ← true , unless explicitly mentioned to false.
- pRequestedFulfillmentOrganizationName ← {{$context.$variables.businessUnitName}}
- pRequestingBusinessUnitName ← {{$context.$variables.businessUnitName}}
- pTransactionalCurrencyName ← Steps to find currency match:
    1. Look for Currency field and Currency symbol from the purchase order document input coming
        2. If currency is not explicitly defined, determine the country from the vendor/buyer address and use the local currency name of that country. else default "US Dollar"
- pTransactionalCurrencyCode ← Steps to find currency match:
    1. Look for Currency field and Currency symbol from the purchase order document input coming
        2. If currency is not explicitly defined, determine the country from the vendor/buyer address and use the local currency code of that country.; else default "USD"
- pCustomerPONumber ← Raw PO number from header
- pFreightTerms ← FreightTerms specified in PO Document explicitly *Unless Explicitly Mentioned , donot pass*
- pSalesPersonName ← Sales Person Name *Unless Explicitly Mentioned , donot pass* .
- pFOBPoint ← Free on Board points. Identifies the ownership of the item while it is in transit , if specified , otherwise null.
- pShipping Instructions ← Delivery Instructions or Shipping Instructions mentioned in PO , otherwise null
"""

```

Note: Use "pTransactionalCurrencyCode" when you can identify a currency code/symbol; if only a name is present *"US Dollar"*, convert to code "USD".

```

## billToCustomer[*]

From **Bill To** / **From** area:

"""

billToCustomer[*].PartyName      ← pBuyingPartyName
billToCustomer[*].AddressLine    ← pBillToAddressLine
billToCustomer[*].City           ← pBillToCity
billToCustomer[*].State          ← pBillToState
billToCustomer[*].PostalCode     ← pBillToPostalCode
billToCustomer[*].Country        ← Country *if present; else try to derive based on the address and
currency , if not deterministic then default to "US"
billToCustomer[*].ContactName   ← pBillToContactName
billToCustomer[*].County         ← pCounty
billToCustomer[*].Province       ← pProvince
billToCustomer[*].FirstName      ← first token of contact
billToCustomer[*].LastName       ← remainder of contact
*omit County/Province if absent*
"""

## shipToCustomer[*]

From **Delivery address** or **Ship To**:

"""

shipToCustomer[*].PartyName      ← pShipToPartyName
shipToCustomer[*].ContactName   ← pShipToContactName
shipToCustomer[*].FirstName      ← first token
shipToCustomer[*].LastName       ← remainder
shipToCustomer[*].AddressLine    ← pShipToAddressLine
shipToCustomer[*].City           ← pShipToCity *preserve case*
shipToCustomer[*].State          ← pShipToState
shipToCustomer[*].County         ← pCounty
shipToCustomer[*].Province       ← pProvince
shipToCustomer[*].PostalCode     ← pShipToPostalCode
shipToCustomer[*].Country        ← Country *if present; else try to derive based on the address and
currency , if not deterministic then default to "US"
*omit County/Province if absent*
"""

## LINES EXTRACTION

Detect each PO line whether tabular or free-form, including wrapped descriptions:

* "ProductNumber" ← "Item #", "SKU", "Part Number", "Product Code"; fall back to leading alphanumeric
token *e.g., "FIT-Tread-2950", "PEL-1540".
* "OrderedQuantity" ← from "QTY", default "1".
* "OrderedUOM" ← from UOM column; default "Each".
* "RequestedShipDate" ← from "Requested", "Request Date", "Delivery Date", or "Promised"; default
"{{Date.now**}}" in ISO Format only like "YYYY-MM-DD".
* "Unit Price" ← any unit-price field near the line.
* Ignore header "Subtotal/Tax/Total/Discount" for line extraction; use them only for anomaly cross-
check.

For each line create:

"""

lines[*].SourceTransactionLineId // increment 1..n , Never pass null , it is mandatory field
lines[*].SourceTransactionLineNumber // increment 1..n, Never pass null , it is mandatory field
lines[*].SourceTransactionScheduleId // increment 1..n, Never pass null , it is mandatory field
lines[*].SourceScheduleNumber        // increment 1..n, Never pass null , it is mandatory field
lines[*].RequestedFulfillmentOrganizationName ← {{$context.$variables.businessUnitName}}
lines[*].ProductNumber
lines[*].OrderedQuantity
lines[*].OrderedUOM
lines[*].RequestedShipDate
lines[*].PaymentTerms
lines[*].ShippingInstructions
"""

# PRICING: ONE CHARGE WITH TWO COMPONENTS PER LINE

Create a single Price charge per line:

"""

lines[*].charges[0].SourceChargeId      = "C" + SourceTransactionLineId
lines[*].charges[0].ApplyTo             = "Price"
lines[*].charges[0].PriceType          = "One time"
lines[*].charges[0].ChargeType         = "Sale"
lines[*].charges[0].ChargeSubType      = "Price"
lines[*].charges[0].PrimaryFlag        = true
lines[*].charges[0].SequenceNumber     = 1
lines[*].charges[0].PricedQuantity     = OrderedQuantity
"""

Charge Components *use extracted "Unit Price":
```

```

"""
CC1:
SourceChargeComponentId      = ChargeId + "-CC1"
PriceElementCode              = "QP_LIST_PRICE"
PriceElementUsageCode         = "LIST_PRICE"
HeaderCurrencyUnitPrice       = UnitPrice
HeaderCurrencyExtendedAmount = UnitPrice * OrderedQuantity
SequenceNumber                = 1

CC2:
SourceChargeComponentId      = ChargeId + "-CC2"
PriceElementCode              = "QP_NET_PRICE"
PriceElementUsageCode         = "NET_PRICE"
HeaderCurrencyUnitPrice       = UnitPrice
HeaderCurrencyExtendedAmount = UnitPrice * OrderedQuantity
SequenceNumber                = 2
"""

*These align with the **Charges** and **Charge Components** child resources of Order Lines.*

# CROSS-CHECKS and EDGE CASES

• If header currency is present *e.g., "All prices in USD"*, set "TransactionalCurrencyCode="USD"".
• If sum*line extended* # header "Subtotal/Total" by more than 1% or a rounding cent, add a single anomaly with a short note *don't block output*.
• If multiple candidate ship-to addresses are found *e.g., long warehouse lists*, pick the labeled **Delivery**/**Ship To** block and add others to "unmappedAttributes" as suggestions.
• If buyer contact appears only once *e.g., "Requisitioner James Bond"*, use it for both "pBuyingPartyContactName" and "billToCustomer[*].ContactName" unless a specific bill-to contact exists.
• If PO header includes both "Order <num>" and "PO # <num>", prefer the explicitly labeled PO; else most frequent/earliest header occurrence.
• Truncate "pSourceTransactionNumber" and "pSourceTransactionId" to ≤ 28 chars.

# ANOMALY CAPTURE *REQUIRED*

Collect anything not confidently mapped:

"""json
"unmappedAttributes": [
{
  "poFragment": "<raw snippet or label>",
  "expectedAttr": "<pVariableName or targetField>",
  "suggestedValue": "<best guess or null>",
  "correctedValue": null,
  "approvedFlag": false
}
]
"""

# OUTPUT FORMAT *RETURN **ONLY** THIS JSON*

"""json
{
  "orderData": {
    "pSourceTransactionNumber": "...",
    "pSourceTransactionId": "...",
    "pBusinessUnitName": "...",
    "pBuyingPartyName": "...",
    "pBuyingPartyContactName": "...",
    "pCurrentDate": "2025-08-16",
    "pCustomerPONumber": "...",
    "TransactionalCurrencyCode": "...",
    "PaymentTermsCode": "...",
    "FreightTerms": "...",
    "FOBPoint": "...",
    "ShippingInstructions": "...",
    "billToCustomer": [
      {
        "PartyName": "...",
        "AddressLine": "...",
        "City": "...",
        "State": "...",
        "PostalCode": "...",
        "Country": "US",
        "County": "...",
        "ContactName": "...",
        "FirstName": "...",
        "LastName": ...
      }
    ],
    "shipToCustomer": [
      {
        "PartyName": "...",
        "AddressLine": "...",
        "City": "...",
        "State": ...
      }
    ]
  }
}
"""

```

```

        "PostalCode": "...",
        "Country": "US",
        "County": "...",
        "ContactName": "...",
        "FirstName": "...",
        "LastName": "..."
    }
],
"pLines": [
{
    "SourceTransactionLineId": 1,
    "SourceTransactionLineNumber": 1,
    "SourceTransactionScheduleId": 1,
    "SourceScheduleNumber": 1,
    "RequestedFulfillmentOrganizationName": "Vision Operations",
    "ProductNumber": "...",
    "PaymentTerms": "...",
    "ShippingInstructions": "...",
    "OrderedQuantity": 10,
    "OrderedUOM": "Each",
    "RequestedShipDate": "2025-08-16",
    "charges": [
        {
            "SourceChargeId": "C1",
            "ApplyTo": "Price",
            "PriceType": "One time",
            "ChargeType": "Sale",
            "ChargeSubType": "Price",
            "PrimaryFlag": true,
            "SequenceNumber": 1,
            "PricedQuantity": 10,
            "chargeComponents": [
                {
                    "SourceChargeComponentId": "C1-CC1",
                    "PriceElementCode": "QP_LIST_PRICE",
                    "PriceElementUsageCode": "LIST_PRICE",
                    "HeaderCurrencyUnitPrice": 100,
                    "HeaderCurrencyExtendedAmount": 1000,
                    "SequenceNumber": 1
                },
                {
                    "SourceChargeComponentId": "C1-CC2",
                    "PriceElementCode": "QP_NET_PRICE",
                    "PriceElementUsageCode": "NET_PRICE",
                    "HeaderCurrencyUnitPrice": 100,
                    "HeaderCurrencyExtendedAmount": 1000,
                    "SequenceNumber": 2
                }
            ]
        }
    ]
},
"unmappedAttributes": []
}
"""

```

Only output the JSON above—no extra narration or markdown.
 In output the JSON above—for any underived value , DO NOT PASS null. Create JSON as "Province": "" instead of "Province": null

Temperature.

0.0

Output to create Sales Order adapted from template “Purchase Order to Sales Order Converter”.

```
{
    "$schema": "http://json-schema.org/draft-07/schema#",
    "title": "Generated schema for Root",
    "type": "object",
    "properties": {
        "orderData": {
            "type": "object",
            "properties": {
                "SourceTransactionNumber": {
                    "type": "string"
                }
            }
        }
    }
}
```

```

"SourceTransactionId": {
    "type": "string"
},
"SourceTransactionSystem": {
    "type": "string"
},
"BusinessUnitName": {
    "type": "string"
},
"BuyingPartyName": {
    "type": "string"
},
"BuyingPartyContactName": {
    "type": "string"
},
"CurrentDate": {
    "type": "string"
},
"CustomerPONumber": {
    "type": "string"
},
"TransactionalCurrencyCode": {
    "type": "string"
},
"PaymentTermsCode": {
    "type": "string"
},
"FreightTerms": {
    "type": "string"
},
"FOBPoint": {
    "type": "string"
},
"Salesperson": {
    "type": "string"
},
"ShippingInstructions": {
    "type": "string"
},
"billToCustomer": {
    "type": "array",
    "items": {
        "type": "object",
        "properties": {
            "PartyName": {
                "type": "string"
            },
            "AddressLine": {
                "type": "string"
            },
            "City": {
                "type": "string"
            },
            "State": {
                "type": "string"
            },
            "PostalCode": {
                "type": "string"
            },
            "Country": {
                "type": "string"
            },
            "County": {
                "type": "string"
            },
            "Province": {
                "type": "string"
            },
            "ContactName": {
                "type": "string"
            },
            "FirstName": {
                "type": "string"
            },
            "LastName": {
                "type": "string"
            }
        }
    },
    "required": [
        "PartyName",
        "AddressLine",
        "City",
        "State",
        "PostalCode",
        "Country",
        "County",
        "Province",
        "ContactName",
        "FirstName",
        "LastName"
    ]
}

```

```

        "FirstName",
        "LastName"
    ],
},
"shipToCustomer": {
    "type": "array",
    "items": {
        "type": "object",
        "properties": {
            "PartyName": {
                "type": "string"
            },
            "AddressLine": {
                "type": "string"
            },
            "City": {
                "type": "string"
            },
            "State": {
                "type": "string"
            },
            "PostalCode": {
                "type": "string"
            },
            "Country": {
                "type": "string"
            },
            "County": {
                "type": "string"
            },
            "Province": {
                "type": "string"
            },
            "ContactName": {
                "type": "string"
            },
            "FirstName": {
                "type": "string"
            },
            "LastName": {
                "type": "string"
            }
        }
    },
    "required": [
        "PartyName",
        "AddressLine",
        "City",
        "State",
        "PostalCode",
        "Country",
        "County",
        "Province",
        "ContactName",
        "FirstName",
        "LastName"
    ]
},
"lines": {
    "type": "array",
    "items": {
        "type": "object",
        "properties": {
            "SourceTransactionLineId": {
                "type": "number"
            },
            "SourceTransactionLineNumber": {
                "type": "number"
            },
            "SourceTransactionScheduleId": {
                "type": "number"
            },
            "SourceScheduleNumber": {
                "type": "number"
            },
            "RequestedFulfillmentOrganizationName": {
                "type": "string"
            },
            "ProductNumber": {
                "type": "string"
            },
            "PaymentTerms": {
                "type": "string"
            },
            "ShippingInstructions": {
                "type": "string"
            }
        }
    }
}

```

```

    "OrderedQuantity": {
      "type": "number"
    },
    "OrderedUOM": {
      "type": "string"
    },
    "RequestedShipDate": {
      "type": "string"
    },
    "charges": {
      "type": "array",
      "items": {
        "type": "object",
        "properties": {
          "SourceChargeId": {
            "type": "string"
          },
          "ApplyTo": {
            "type": "string"
          },
          "PriceType": {
            "type": "string"
          },
          "ChargeType": {
            "type": "string"
          },
          "ChargeSubType": {
            "type": "string"
          },
          "PrimaryFlag": {
            "type": "boolean"
          },
          "SequenceNumber": {
            "type": "number"
          },
          "PricedQuantity": {
            "type": "number"
          },
          "chargeComponents": {
            "type": "array",
            "items": {
              "type": "object",
              "properties": {
                "SourceChargeComponentId": {
                  "type": "string"
                },
                "PriceElementCode": {
                  "type": "string"
                },
                "PriceElementUsageCode": {
                  "type": "string"
                },
                "HeaderCurrencyUnitPrice": {
                  "type": "number"
                },
                "HeaderCurrencyExtendedAmount": {
                  "type": "number"
                },
                "SequenceNumber": {
                  "type": "number"
                }
              }
            },
            "required": [
              "SourceChargeComponentId",
              "PriceElementCode",
              "PriceElementUsageCode",
              "HeaderCurrencyUnitPrice",
              "HeaderCurrencyExtendedAmount",
              "SequenceNumber"
            ]
          }
        }
      },
      "required": [
        "SourceChargeId",
        "ApplyTo",
        "PriceType",
        "ChargeType",
        "ChargeSubType",
        "PrimaryFlag",
        "SequenceNumber",
        "PricedQuantity",
        "chargeComponents"
      ]
    }
  },
  "required": [

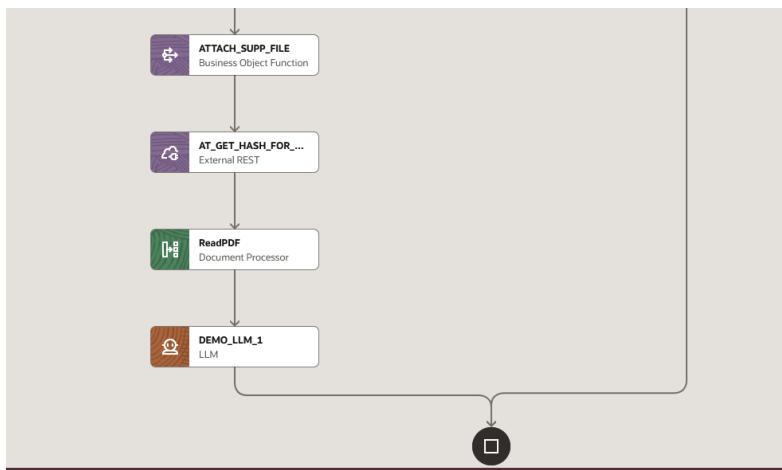
```

```

        "SourceTransactionLineId",
        "SourceTransactionLineNumber",
        "SourceTransactionScheduleId",
        "SourceScheduleNumber",
        "RequestedFulfillmentOrganizationName",
        "ProductNumber",
        "PaymentTerms",
        "ShippingInstructions",
        "OrderedQuantity",
        "OrderedUOM",
        "RequestedShipDate",
        "charges"
    ]
}
},
"required": [
    "SourceTransactionNumber",
    "SourceTransactionId",
    "SourceTransactionSystem",
    "BusinessUnitName",
    "BuyingPartyName",
    "BuyingPartyContactName",
    "CurrentDate",
    "CustomerPONumber",
    "TransactionalCurrencyCode",
    "PaymentTermsCode",
    "FreightTerms",
    "POBPoint",
    "Salesperson",
    "ShippingInstructions",
    "billToCustomer",
    "shipToCustomer",
    "lines"
]
},
"unmappedAttributes": {
    "type": "array",
    "items": {
        "type": "object",
        "properties": {
            "poFragment": {
                "type": "string"
            },
            "expectedAttr": {
                "type": "string"
            },
            "suggestedValue": {},
            "correctedValue": {},
            "approvedFlag": {
                "type": "boolean"
            }
        },
        "required": [
            "poFragment",
            "expectedAttr",
            "suggestedValue",
            "correctedValue",
            "approvedFlag"
        ]
    }
},
"required": [
    "orderData",
    "unmappedAttributes"
]
}

```

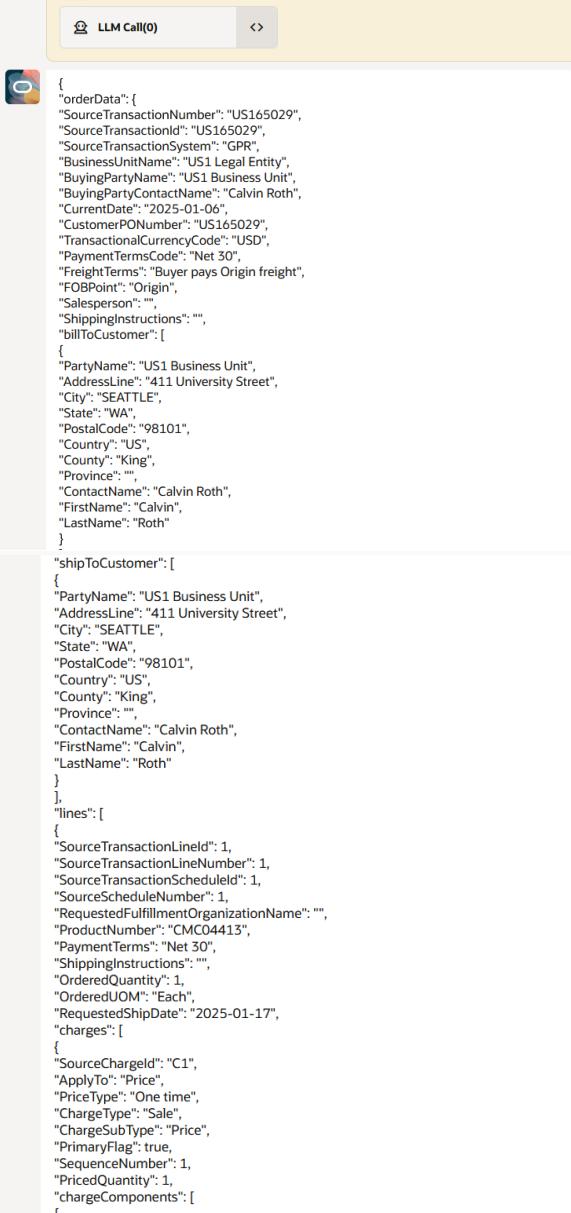
Now the overview look like this.



Testing.

Payload.

```
{
    "pDirectory": "/files/Nordea/AP_Invoices/Outbound/LOG",
    "pFileName": "PO_US165029_3.pdf",
    "pSupplier": "Lee Supplies"
}
```

These are (source) samples from PDF.	This is the resulting payload of our test to create a sales order												
<p>Purchase Order US165029</p> <table border="1"> <tr> <td style="text-align: right;">Order</td> <td>US165029</td> </tr> <tr> <td style="text-align: right;">Order Date</td> <td>06-JAN-2025</td> </tr> <tr> <td style="text-align: right;">Change Order</td> <td>2</td> </tr> <tr> <td style="text-align: right;">Change Order Date</td> <td>06-JAN-2025</td> </tr> <tr> <td style="text-align: right;">Revision</td> <td>6</td> </tr> <tr> <td style="text-align: right;">Ordered</td> <td>1,017.00 USD</td> </tr> </table> <p>Sold To: US1 Legal Entity 500 ORACLE PKWY REDWOOD CITY, CA 94065 SAN MATEO</p> <p>Supplier: Lee Supplies 5000 CARTER DR. LOS ANGELES, CA 90032 LOS ANGELES</p> <p>Bill To: US1 Business Unit 411 University Street SEATTLE, WA 98101 King UNITED STATES</p> <p>Ship To:</p> <p>Line Item Price Quantity UOM Ordered Taxable</p> <p>1 Black Laser Printer Toner Cartridge Supplier Item CMC04413 This line references Document (Blanket Purchase Agreement) 52236.</p> <p> Promised 1/17/25 Requested 1/17/25 Requested and Promised Dates correspond to the date of arrival at the Ship-to Location.</p> <p> Line Total 79.00</p> <p>2 Inkjet Cyan Cartridge Supplier Item CMC04409 This line references Document (Blanket Purchase Agreement) 52236.</p> <p> Line Total 39.00</p> <p>3 LaserJet Pro 400 M401n Laser Printer Supplier Item HPP9985847 Supplier Configuration ID S1209 This line references Document (Blanket Purchase Agreement) 52232.</p> <p> Promised 1/15/25 Requested 1/15/25 Requested and Promised Dates correspond to the date of arrival at the Ship-to Location.</p> <p> Line Total 899.00</p> <p>Total 1,017.00</p>	Order	US165029	Order Date	06-JAN-2025	Change Order	2	Change Order Date	06-JAN-2025	Revision	6	Ordered	1,017.00 USD	 <pre>{ "orderData": { "SourceTransactionNumber": "US165029", "SourceTransactionId": "US165029", "SourceTransactionSystem": "GPR", "BusinessUnitName": "US1 Legal Entity", "BuyingPartyName": "US1 Business Unit", "BuyingPartyContactName": "Calvin Roth", "CurrentDate": "2025-01-06", "CustomerPONumber": "US165029", "TransactionalCurrencyCode": "USD", "PaymentTermsCode": "Net 30", "FreightTerms": "Buyer pays Origin freight", "FOBPoint": "Origin", "Salesperson": "", "ShippingInstructions": "", "billToCustomer": [{ "PartyName": "US1 Business Unit", "AddressLine": "411 University Street", "City": "SEATTLE", "State": "WA", "PostalCode": "98101", "Country": "US", "County": "King", "Province": "", "ContactName": "Calvin Roth", "FirstName": "Calvin", "LastName": "Roth" }], "shipToCustomer": [{ "PartyName": "US1 Business Unit", "AddressLine": "411 University Street", "City": "SEATTLE", "State": "WA", "PostalCode": "98101", "Country": "US", "County": "King", "Province": "", "ContactName": "Calvin Roth", "FirstName": "Calvin", "LastName": "Roth" }], "lines": [{ "SourceTransactionLineId": 1, "SourceTransactionLineNumber": 1, "SourceTransactionScheduledId": 1, "SourceScheduleNumber": 1, "RequestedFulfillmentOrganizationName": "", "ProductNumber": "CMC04413", "PaymentTerms": "Net 30", "ShippingInstructions": "", "OrderedQuantity": 1, "OrderedUOM": "Each", "RequestedShipDate": "2025-01-17", "charges": [{ "SourceChargeId": "C1", "ApplyTo": "Price", "PriceType": "One time", "ChargeType": "Sale", "ChargeSubType": "Price", "PrimaryFlag": true, "SequenceNumber": 1, "PricedQuantity": 1, "chargeComponents": [...] }] }] } }</pre>
Order	US165029												
Order Date	06-JAN-2025												
Change Order	2												
Change Order Date	06-JAN-2025												
Revision	6												
Ordered	1,017.00 USD												

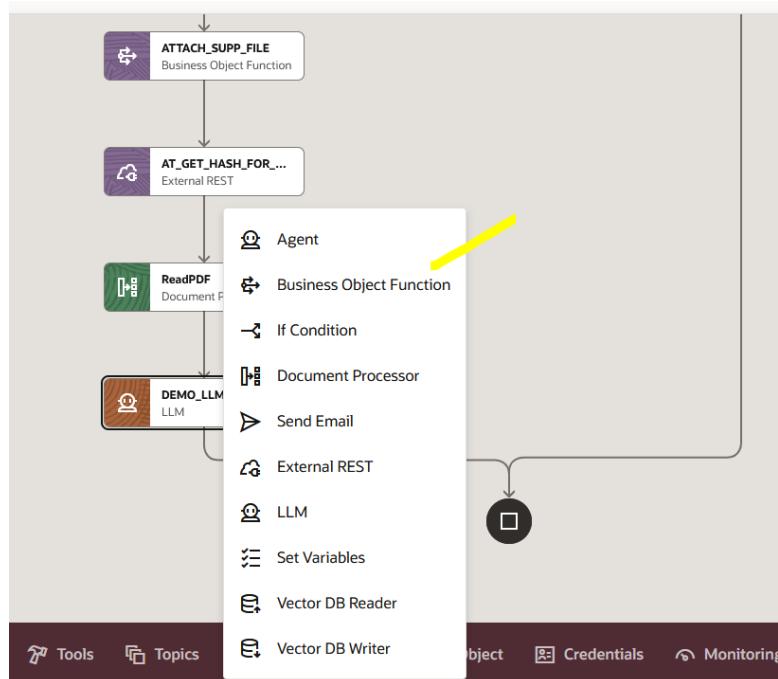
	<pre> "HeaderCurrencyExtendedAmount": 79, "SequenceNumber": 1 }, { "SourceChargeComponentId": "C1-CC2", "PriceElementCode": "QP_NET_PRICE", "PriceElementUsageCode": "NET_PRICE", "HeaderCurrencyUnitPrice": 79, "HeaderCurrencyExtendedAmount": 79, "SequenceNumber": 2 }] }], }, { "SourceTransactionLineId": 2, "SourceTransactionLineNumber": 2, "SourceTransactionScheduleId": 2, "SourceScheduleNumber": 2, "RequestedFulfillmentOrganizationName": "", "ProductNumber": "CMCO4409", "PaymentTerms": "Net 30", "ShippingInstructions": "", "OrderedQuantity": 1, "OrderedUOM": "Each", "RequestedShipDate": "2025-01-17", "charges": [{ "SourceChargeld": "C2", "ApplyTo": "Price", "PriceType": "One time", "ChargeType": "Sale", "ChargeSubType": "Price", "PrimaryFlag": true, "SequenceNumber": 1, "PricedQuantity": 1, "chargeComponents": [{ "SequenceNumber": 1 }, { "SourceChargeComponentId": "C2-CC2", "PriceElementCode": "QP_NET_PRICE", "PriceElementUsageCode": "NET_PRICE", "HeaderCurrencyUnitPrice": 39, "HeaderCurrencyExtendedAmount": 39, "SequenceNumber": 2 }] }], { "SourceTransactionLineId": 3, "SourceTransactionLineNumber": 3, "SourceTransactionScheduleId": 3, "SourceScheduleNumber": 3, "RequestedFulfillmentOrganizationName": "", "ProductNumber": "HPP9985847", "PaymentTerms": "Net 30", "ShippingInstructions": "", "OrderedQuantity": 1, "OrderedUOM": "Each", "RequestedShipDate": "2025-01-15", "charges": [{ "SourceChargeld": "C3", "ApplyTo": "Price", "PriceType": "One time", "ChargeType": "Sale", "ChargeSubType": "Price", "PrimaryFlag": true, "SequenceNumber": 1, "PricedQuantity": 1, "chargeComponents": [{ "SourceChargeComponentId": "C3-CC1", "PriceElementCode": "QP_LIST_PRICE", </pre>
--	---

Create Sales order. Business Object. (11).

Duplicated from “Purchase Order to Sales Order Converter” template.

Business object				
Name	Description	Seeded	Family	Product
CreateOrdersForPOToSO	This business object creates a new sales order with input for the lines and header level details.	Yes	SCM	Order Management
CreateOrdersForPOToSO_2	This business object creates a new sales order with input for the lines and header level details.	No	SCM	Order Management

Include Business Object as Business Object Function.



CREATE_SO
Business Object Function

Name CREATE_SO	Code CREATESO	Type Business Object Function
Family SCM	Product Order Management	Business object CreateOrdersForPOToSO_2
Function CreateOrderWithLinesAndChargesUsingAgent		

Parameters

pSourceTransactionNumber	string
pSourceTransactionSystem	string
pSourceTransactionId	string
pBusinessUnitName	string
pBuyingPartyName	string

Create

Name: CREATE_SO

Now the overview looks like this:



Mapping.

Only some values are mapped for learning purposes.

Out of scope to get a complete insert for now.

Parameters

Name pSourceTransactionNumber	Type String
Value {{\\$context.\$nodes.DEMOLLM1.\$output.orderData.SourceTransactionNumber}}	
Name pLines	Type String
Value {{\\$context.\$nodes.DEMOLLM1.\$output.orderData.lines}}	

Testing.

 BOF_CreateOrdersForPOToSO_2

Latency
0.13 seconds

Input:

```
{"pSourceTransactionSystem": "", "pSourceTransactionId": "", "pBusinessUnitName": "", "pBuyingPartyName": "", "pBuyingPartyContactName": "", "pCurrentDate": "", "pRequestedFulfillmentOrganizationName": "", "pRequestingBusinessUnitName": "", "pCustomerPONumber": "", "pCurrencyName": "", "pCurrencyCode": "", "pFreightTerms": "", "pSalesperson": "", "pBillToPartyName": "", "pBillToAddressLine": "", "pBillToCity": "", "pBillToState": "", "pBillToPostalCode": "", "pBillToCounty": "", "pBillToProvince": "", "pBillToCountry": "", "pBillToContactName": "", "pBillToFirstName": "", "pBillToLastName": "", "pShipToPartyName": "", "pShipToContactName": "", "pShipToFirstName": "", "pShipToLastName": "", "pShipToAddressLine": "", "pShipToCity": "", "pShipToState": "", "pShipToPostalCode": "", "pShipToCounty": "", "pShipToProvince": "", "pShipToCountry": "", "pSourceTransactionNumber": "US165029", "pLines": [{"OrderedUOM": "Ea", "charges": [{"SourceChargeId": "C1", "chargeComponents": [{"PriceElementCode": "QP_LIST_PRICE", "HeaderCurrencyExtendedAmount": 79, "SourceChargeComponentId": "C1-CC1", "PriceElementUsageCode": "LIST_PRICE", "HeaderCurrencyUnitPrice": 79, "SequenceNumber": 1}, {"PriceElementCode": "QP_NET_PRICE", "HeaderCurrencyExtendedAmount": 79, "SourceChargeComponentId": "C1-CC2", "PriceElementUsageCode": "NET_PRICE", "HeaderCurrencyUnitPrice": 79, "SequenceNumber": 2}], "PrimaryFlag": true, "ChargeType": "Sale", "ApplyTo": "Price", "SequenceNumber": 1, "PricedQuantity": 1, "ChargeSubType": "Price", "PriceType": "One"}]}
```

```

    "pShipToProvince": "",
    "pShipToCountry": "",
    "pSourceTransactionNumber": "US165029",
    "pLines": [
      {
        "OrderedUOM": "Ea",
        "charges": [
          {
            "SourceChargeId": "C1",
            "chargeComponents": [
              {
                "PriceElementCode": "QP_LIST_PRICE",
                "HeaderCurrencyExtendedAmount": 79,
                "SourceChargeComponentId": "C1-CC1",
                "PriceElementUsageCode": "LIST_PRICE",
                "HeaderCurrencyUnitPrice": 79,
                "SequenceNumber": 1
              },
              {
                "PriceElementCode": "QP_NET_PRICE",
                "HeaderCurrencyExtendedAmount": 79,
                "SourceChargeComponentId": "C1-CC2",
                "PriceElementUsageCode": "NET_PRICE",
                "HeaderCurrencyUnitPrice": 79,
                "SequenceNumber": 2
              }
            ],
            "PrimaryFlag": true,
            "ChargeType": "Sale",
            "ApplyTo": "Price",
            "SequenceNumber": 1,
            "PricedQuantity": 1,
            "ChargeSubType": "Price",
            "PriceType": "One time"
          }
        ],
        "SourceTransactionLineNumber": 1,
        "OrderedQuantity": 1,
        "ShippingInstructions": "",
        "RequestedShipDate": "2025-01-17",
        "SourceTransactionScheduleId": 1,
        "RequestedFulfillmentOrganizationName": "",
        "ProductNumber": "CMC04413",
        "DocumentToName": "Not set"
      }
    ]
  }
}

```

As we have mapped only a couple of nodes (source_transaction_number and lines) we get an incomplete payload, but the process is run.

Feel free to complete the mapping to get your sales order created.

Conclusion.

We are able to read a PDF file, attach it to a supplier, and then read it and generate payload using Document Processor and LLM that could be used to create sales orders, purchase requisitions or another object changing the Prompt and Output.

Technical

Code example	Comments
XX_GET_SUPP_HASH_ATTACH_01.00.0000.iar	Integration to be used as reference.
XX_AI_FF_GET_FILE_CONTE_01.00.0002.iar	Integration to be used as reference.
PO_US165029_0.pdf	PDF samples for testing.
PO_US165029_1.pdf	
...	
PO_US165029_5.pdf	