

Oracle ACE

Tech Superstars Unite

Get worldwide recognition as an Oracle ACE



Oracle.com
Profile Page



Travel Support



Swag & Certification
Exam Credits



Networking
Events



Exclusive Content from
Product Development



Your Own Oracle
Cloud Account

✕ @oracleace

📄 [linkedin.com/groups/72183](https://www.linkedin.com/groups/72183)

🦋 @oracleace.bsky.social

Learn more at:
ace.oracle.com



1,440+ Global ACE Members in 86 Countries

Members Directory: ace.oracle.com/directory





Mastering Design Patterns for Custom Business Processes in Oracle Fusion Applications: Strategies for CX, HCM, and SCM

Real-World Patterns: ESS + BIP + OIC | Groovy-based ESS | Emerging AI Agent Studio

Alex Rabinovich, Oracle ACE Pro
CEO, CID SOFTWARE SOLUTIONS LTD



Agenda

- Custom Business Processes Examples
 1. Parameterised Data Fetch and Update
 2. Event Driven Processing
- End User Interaction
 1. ESS Framework Utilization (standard UI, scheduling, monitoring)
 2. BI Publisher Catalog Report Parameter Page
 3. Classic UI Page Composer Links
 4. Application Composer Actions/CX Smart Actions
 5. Visual Builder Studio Extensions
 6. Dedicated Visual Builder Studio AppUI Screens
 7. AI Agent Exposure
- Covered Design Patterns for Custom Business Processes in FA
 1. BI Publisher as Data Fetch and UI Templating Tool
 2. ESS Job submitting Synchronous Processing in OIC
 3. ESS Job submitting Asynchronous Processing in OIC
 4. Standard/Custom Integration Event triggering an OIC Integration
 5. Groovy based ESS Job
 6. New: AI Agent Studio
- Design Pattern Selection
- Q & A

Custom Business Processes Examples

Real World Examples from CX, SCM and HCM Implementations

SaaS Data Manipulation Process Examples

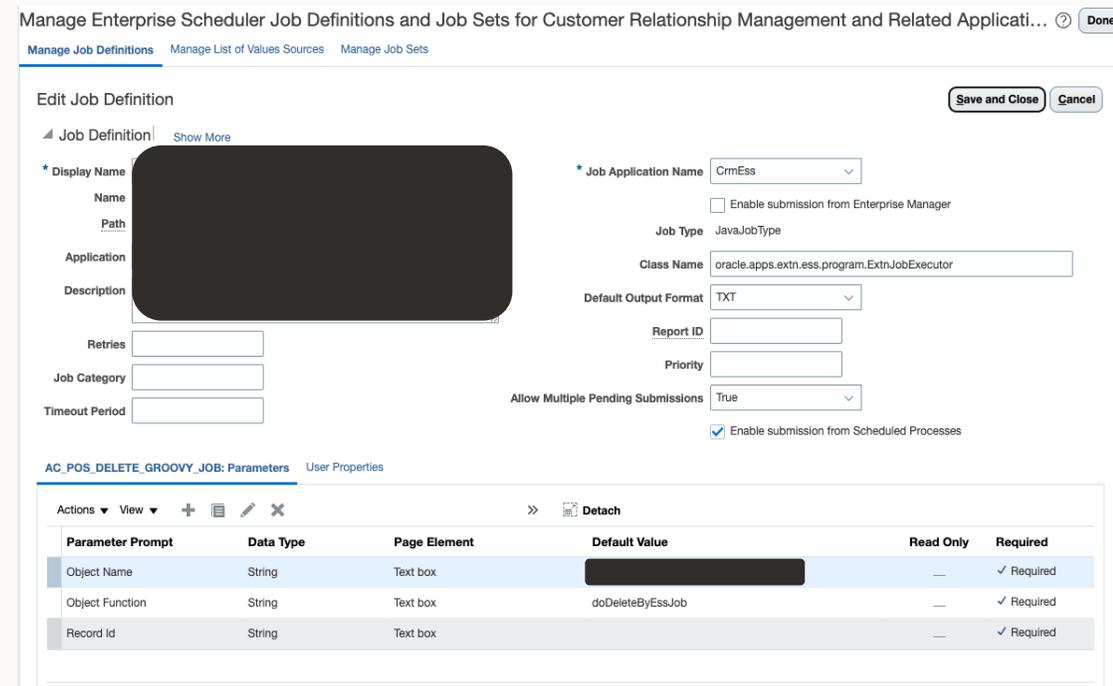
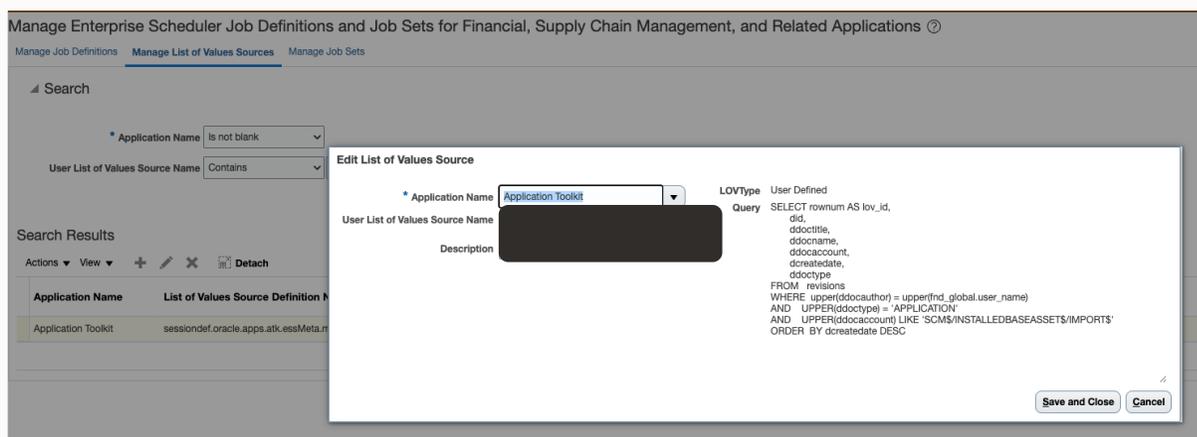
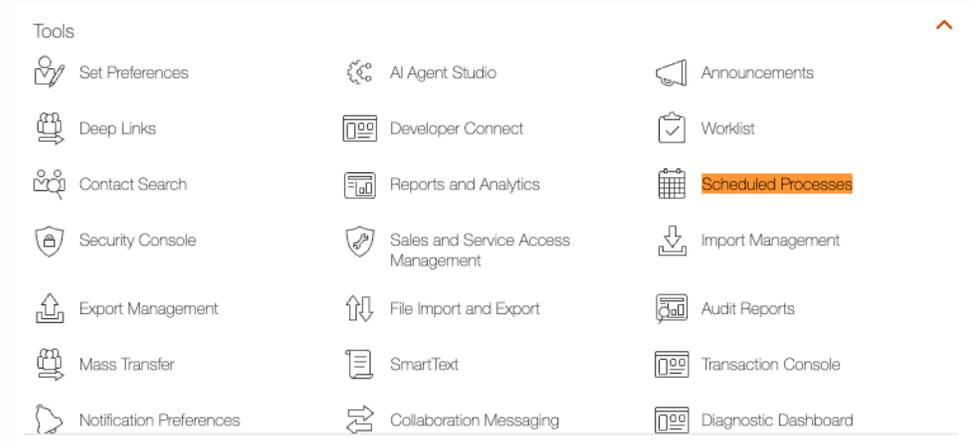
- Data Manipulation Examples
 - Custom post clone HCM anonymization process
 - Enrich subscriptions automatically created from Order Management
 - Update Install Base following Service Logistics transactions
 - Auto create deliveries
 - Release order holds/pauses based on custom criteria
 - Upload sales orders from an excel file
 - Upload file of different formats – import customer product usage
- Data Extract Examples
 - Run a report based on an input file
- Event Driven Examples
 - Populate additional item defaults upon creation
 - Notify relevant business users with sales order status change

End User Interaction

Expose Custom Business Processes To Users

1. End User Interaction – ESS Framework Utilization

- Expose in standard Schedule Processes Menu
- Use Checkbox/Date/Text/Number/LOV Parameters (without dependent SQL)
- Ability to connect to Report or Custom Groovy Function
- Schedule, Monitor Submission for Asynchronous Jobs (*)
- Schedule, Monitor Submission, Execution and Erros for Synchronous Jobs
- View Output for Synchronous/Groovy Jobs (manual screen refreshes)
- Per ESS Job Role in Security Console



2. End User Interaction – BI Catalog Report Parameter Page

- Standard bookmarkable report link
- BI Publisher parameters utilization (including LOV dependency)
- Suitable for submission of async OIC calls
- Supports synchronous OIC calls
- Supports scheduled submissions
- BI Publisher template layout utilization (render Excel, HTML based on OIC results)
- BI Publisher report/folder security

BU LOV: Type: SQL Query

Options Cache Result

SQL Query

Query Builder

```
select name ,organization_id from hr_operating_units
```

LOV Test

Data Model

Properties

- ▲ Data Sets
- ▲ Event Triggers
- ▲ Flexfields
- ▲ List of Values
 - BU LOV
 - ORDER_LOV**
- ▲ Parameters
 - p_bu
 - p_order
- ▲ Bursting

List of Values

+ X

Name	Type	Data Source	Reorder
BU LOV	SQL Query	ApplicationDB_FSCM	▲▼
ORDER_LOV	SQL Query	ApplicationDB_FSCM	▲▼

ORDER_LOV: Type: SQL Query

Options Cache Result

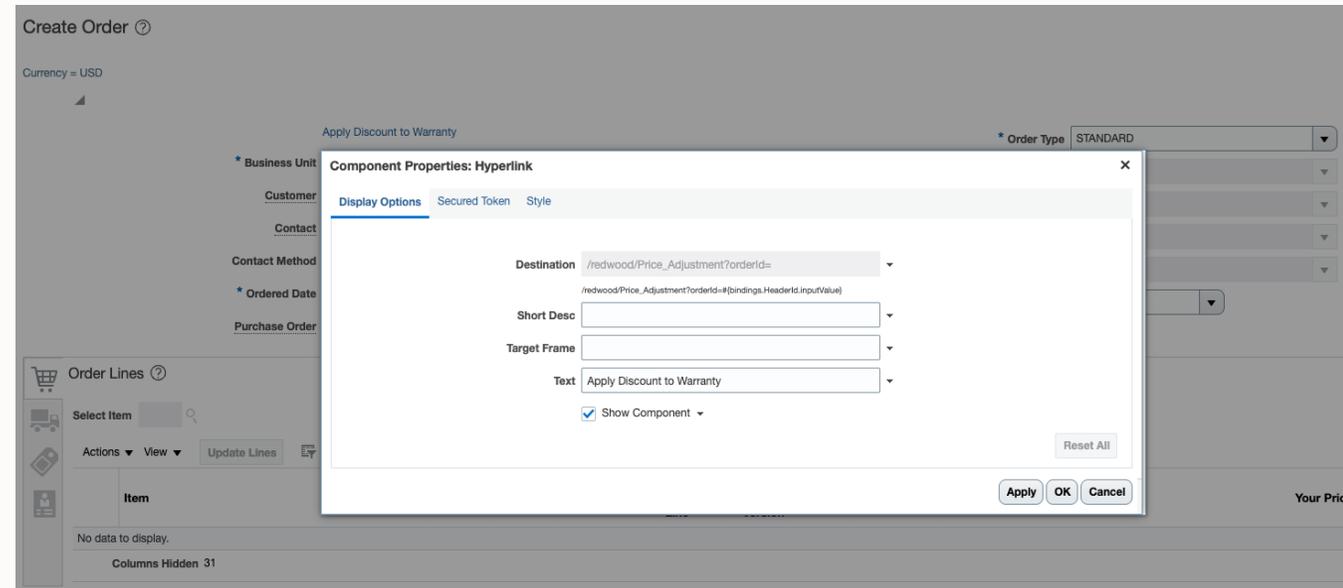
SQL Query

```
select order_number, header_id from doo_headers_all where org_id = :p_bu
```

Query Builder

3. End User Interaction – Classic UI Page Composer Links

- Setup through Page Composer Sandbox
- Allows current record context passing
- Navigate to BI Publisher report (including report submitting an OIC)
- Navigate to Visual Builder AppUI/Redwood URL
- Keep URL definition P2T safe (//analytics, //xmlpserver)



4. End User Interaction – Application Composer Actions/CX Smart Actions

- Setup through Application Composer Sandbox
- Allows current record context passing
- Navigate to BI Publisher report (including report submitting an OIC)
- Navigate to Visual Builder AppUI/Redwood URL
- Keep URL definition P2T safe (//analytics, //xmlpserver)
- Navigate to OIC (Webservice Definition) (can be made P2T safe with OCI API Gateway)

The screenshot shows the 'Application Composer' interface. On the left, a tree view under 'Object Tags' includes 'Service Request' with sub-items like 'Fields', 'Joins', 'Pages', 'Actions and Links', 'Server Scripts', 'Message', 'Milestone', 'Service Request Contact', 'Service Request Reference', and 'Service Request Resource'. The main area is titled 'Service Request: Edit Action or Link'. It has a 'Display Label' field set to 'Update Parts Revision', a 'Name' field with 'Update_Parts_Revision', and a 'Type' radio button set to 'Action'. Below this is a 'Script' section with a 'Method Name' field set to 'ACUpdSRPartRevision' and a code editor containing the following script:

```
def params = []
def returnparams = []
params.SRID = SRID
returnparams = adf.util. [redacted] params, [redacted] /1.0/start')
if (returnparams.msg != 'Success')
{
    throw new oracle.jbo.ValidationException((String)returnparams.msg)
}
```

The screenshot shows the 'Application Composer' interface with the 'Web Services' section active. It features a search bar and a table of web services. The table has columns for 'Name' and 'URL'.

Name	URL
[redacted]ocalFscmREST	https://fa-internal.oracleoutsourcing.com:10663/fscmRestApi/resources/latest/##REL_URL...

The screenshot shows a test results window titled 'Test Alex' with a 'Low' status indicator. It lists several test cases:

- update part revision|
- Filter update part revision
- Search Knowledge
- Update Part Revisions

5. End User Interaction – Visual Builder Studio Extensions

- Setup through Edit Page in Visual Builder Studio
- Regular link addition is possible only in Oracle defined components (new section in Edit Purchase Requisition Delivery Details)
- Allows current record context passing (*)
- Navigate to BI Publisher report (including report submitting an OIC) (*)
- Navigate to Visual Builder AppUI/Redwood URL
- Keep URL definition P2T safe (//analytics, //xmlpserver)
- Advanced solutions – utilization of Lifecycle Events

Getting Started | edit-delivery-billing x | shoppingcart | homepage | fscm

Page Designer | **Action Chains (2)** | Event Listeners (1) | Events | Types | Variables (12) | JavaScript | JSON | Settings

Action Chains > vbAfterNavigateListener > **Diagram** | Tests

Filter | Search

General

- Assign Variable
- Call Action Chain
- Call Function
- Call REST
- Call Variable
- Code
- Fire Data Provider Event
- Fire Event

```

(-) A urlParams = new URLSearchParams(window.location.search)
    A lineId = urlParams.get('lineId')

(-) A lineResponse = Call REST site_prc_extension:fscm/getFscmRestApiResourcesLatestPurchaseRequisitionsReqIdChildLinesLineId
    URI Parameters
    A expand: 'DFF'
    A fields: 'SourceType,DFF'
    A lineId: lineId
    A reqId: $base.flow.variables.activeRequisitionDetails.RequisitionHeaderId
  
```

Settings and Actions Sign Out

Personalization
Set Preferences

Administration
Edit Page in Visual Builder Studio
Setup and Maintenance

Troubleshooting
Run Diagnostics Tests

Applications Help

About This Application

6. End User Interaction – Dedicated Visual Builder Studio AppUI Screens

- Requires VB expertise in addition to BIP

The screenshot displays a web application interface with a dark blue header bar. Below the header, there is a white panel containing two main sections: 'Report Parameters' and 'Job Execution Status'.

Report Parameters:

- A file upload button labeled 'Serials File' with an upward arrow icon. To its right, the text 'File Selected: CSV.csv' is displayed.
- A text input field labeled 'Serial'.
- A date selection field labeled 'From Date' with a calendar icon, showing the date '2/19/2026'.
- A 'Process File' button with a right-pointing arrow icon.

Job Execution Status:

- A 'Refresh' button with a circular arrow icon.
- A table with the following columns: Name, Request Id, Status, Submission Time, and Last Refresh Time.
- A single row of data is visible, with the 'Name' column redacted by a black box. The other values are: Request Id: 5037974, Status: Succeeded (with a green checkmark icon), Submission Time: 19-Feb-2026 5:57 PM, and Last Refresh Time: 19-Feb-2026 5:58 PM.
- A 'View Output' button with a download icon is located to the right of the table row.

7. End User Interaction – AI Agent Exposure

- Role based AI Agents launch screen /fscmUI/redwood/human-resources/ai-studio/agent-explore
- Guided Journeys
- Smart Actions (CX)

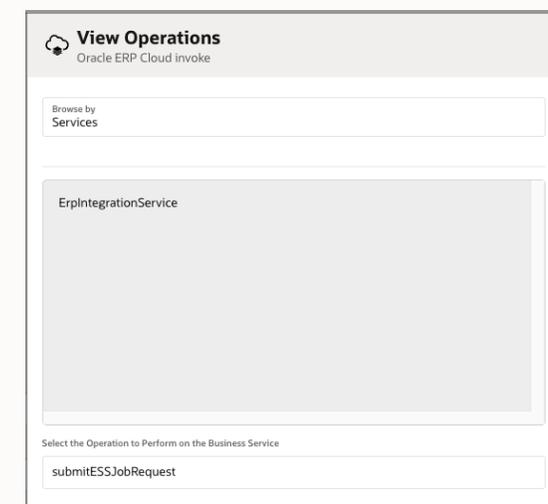
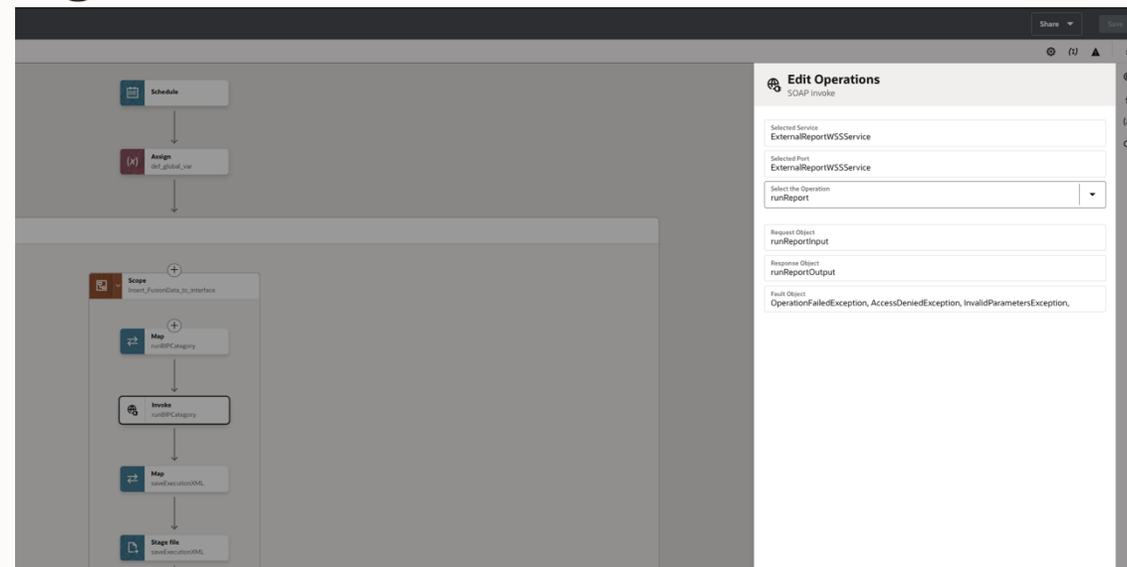
The screenshot displays the 'AI Agent Explore' interface. At the top, there are two tabs: '1 of 1 Needs Attention' and 'Explore 3 Agents'. Below the tabs is a search bar with a red circle icon, a 'Quote' tag with an 'X' to remove it, and the placeholder text 'Search by agent name, code, or description'. Below the search bar is a card for 'Self Service Procurement' featuring the 'Quote To Purchase Requisition Assistant' agent. The card includes a description: 'Automatically create a requisition from a supplier quotation using an AI Agent.' and a stylized leaf icon.

Covered Design Patterns for Custom Business Processes in FA

Cloud native design patterns for custom business logic implementation in FA

1. Covered Design Patterns for Custom Business Processes in FA – BI Publisher as Data Fetch and UI Templating Tool

- Identify fetched data volume and query complexity
- Prefer ESS submission vs. online SOAP BIP call
 - Execution timeout
 - Result output size
 - Impact on online FA users
- ESS + Wait vs. Enable Events = Y
 - Single OIC Process
 - Submitter + Callback



1. Covered Design Patterns for Custom Business Processes in FA – BI Publisher as Data Fetch and UI Templating Tool – Cont'd

- Use Data (XML) and Data (CSV) to skip layout calculation
- Utilize lexical variables to generate dynamic SQL
(example: &P_QUERY_TEXT – useful for Planning modules, custom object access)
- Use Explain Plan (including AI)

Data Model **LOV Test** 🔍 +

Layout

Apply Style Template

Name	Template File	Type	Output Formats	Default Format	Default Layout	Apply Style Template	Active	View Online	Locale	Reorder
Default	Default.rtf	rtf	Data (XML):Data /C <input type="checkbox"/> PDF/X <input type="checkbox"/> PDF/JA <input type="checkbox"/> Zipped PDFs <input type="checkbox"/> FO Formatted XML <input checked="" type="checkbox"/> Data (XML) <input checked="" type="checkbox"/> Data (CSV)	Data (CSV)	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	English (United States)	▲ ▼

Generate RTF layout based on selected Data Model.

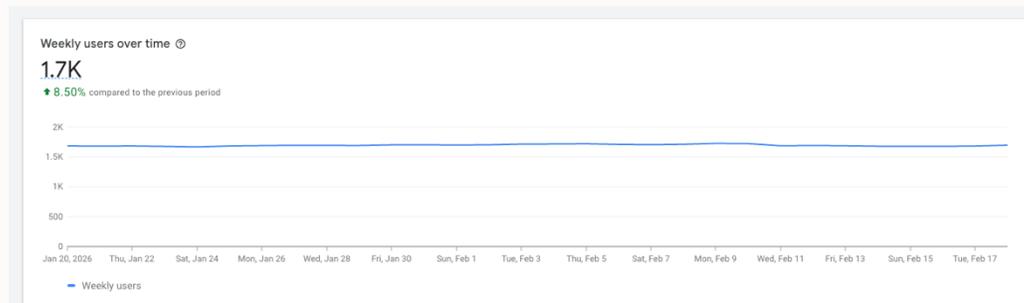
Generate

Auto Generate Layout ✕

* Template Name

1. Covered Design Patterns for Custom Business Processes in FA – BI Publisher as Data Fetch and UI Templating Tool – Cont'd

- Use free/paid tools for easy SQL access
 - Free “CID Fusion Query” chrome extension
 - Recent addition – support for 320k query text



v2.8.0 Dev1 SQL History Run Query Explain Plan Max Records: 10

What's New in Version 2.8.0 Installation About Refresh Metadata Object Browser Environments Logged In (SSO)

```

1 select * from all_objects
2
3 select * from egp_system_items_b where item_number like '%112%'

```

Query Execution Plan

Plan hash value: 991682699

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		89478	67M	82290 (1)	00:00:04
* 1	TABLE ACCESS STORAGE FULL	egp_system_items_b	89478	67M	82290 (1)	00:00:04

Predicate Information (identified by operation id):

```

1 - storage(("ITEM_NUMBER" LIKE '%112%' AND "ITEM_NUMBER" IS NOT NULL)
filter(("ITEM_NUMBER" LIKE '%112%' AND "ITEM_NUMBER" IS NOT NULL))

```

0Y! Pro Tip: Click the extension icon while on any Fusion Apps page to automatically setup and login! You can also

Execution Time: 4.03 seconds

Show 5 entries

OWNER	OBJECT_NAME	SUBOBJECT_NAME	OBJE	SECONDARY	NAMESPACE	EDITION_NAME	SHARING	EDITIONABLE	ORACLE_MAINTAINED
FUSION	ACA_ATTACH_MIGRATE_LOG		4991	N	1	FA_20260214212748	NONE	Y	N
FUSION	ACA_ATTACH_MIGRATE_LOG_N1		25683	N	4		NONE		N
FUSION	ACA_ATTACH_MIGRATE_LOG_U1		25684	N	4		NONE		N
FUSION	ACA_AUTO_SEQ_METADATA_B		4991401						
FUSION	ACA_AUTO_SEQ_METADATA_B=		4991402						

Showing 1 to 5 of 10 entries

Export To Excel Search:

Download Close

Previous 1 2 Next

1. Covered Design Patterns for Custom Business Processes in FA – BI Publisher as Data Fetch and UI Templating Tool – Cont'd

- Access UCM text documents by UCM connection
- Access Query Example:

```
with rws as (select :DOCUMENT str from dual)
select REGEXP_SUBSTR(column_value, '[^,]+', 1, 1) AS item,
       REGEXP_SUBSTR(column_value, '[^,]+', 1, 2) AS qty,
       REGEXP_SUBSTR(column_value, '[^,]+', 1, 3) AS desc
from (select trim(regexp_substr(str, '[^ | chr(10) | | ]'+, 1, level)) column_value
      from rws
      connect by level <= length(str) - length(replace(str, chr(10))) + 1
      ) file_data
```

The screenshot shows the BI Publisher Data Model interface. On the left, a tree view shows the 'Data Model' structure with 'Data Sets' expanded to show 'Parser', 'UCM1', and 'revs'. The main workspace displays a diagram with three data set components: 'G_REVS', 'G_PARSER', and 'G_UCM'. 'G_REVS' and 'G_PARSER' are connected to 'G_UCM'. An 'Edit Data Set - UCM1' dialog is open in the foreground, showing the following configuration:

- Name: UCM1
- Data Source: FA_UCM_PROVISIONED
- Parent Group: G_REVS
- Document ID: G_REVS.DID
- Content Type: Text

The screenshot shows the BI Publisher Data Model interface with a diagram view. The 'Diagram' tab is active, showing a flow from 'G_REVS' and 'G_PARSER' to 'G_UCM'. The 'Global Level Functions' component is also visible. The 'Data Model' tree on the left shows the following structure:

- Data Sets
 - Parser
 - UCM1
 - revs
- Event Triggers
- Flexfields
- List of Values
- Files
- Parameters
- p_file_id
- Bursting

1. Covered Design Patterns for Custom Business Processes in FA – BI Publisher as Data Fetch and UI Templating Tool – Cont'd

- Utilize Oracle's database powerful XML and JSON functions together with CLOB fields in Custom Objects
- Combine fully qualified JSON payload generated by DB with OIC binary input

```

WITH
FileUploadRequest_sql as (
  select 'select ' || listagg(c.column_name || ' ' || cu.attribute_name, ',') ||
    ' from ' || max(t.table_name) ||
    ' where attribute_category = ''FileUploadRequest_c'' and id= ' || :p_upload_rec_id code
  from
  adf_extensible_table_usage tu,
  adf_extension_column_usage cu,
  adf_extension_column c,
  adf_extensible_table t
  where
  tu.context_column_value = 'FileUploadRequest_c'
  and cu.table_usage_id = tu.usage_id
  and cu.column_id = c.column_id
  and c.table_id = t.table_id
),
FileUploadRequest as (
  select c.*
  from FileUploadRequest_sql s,
  xmltable(
    '/ROWSET/ROW' passing dbms_xmlgen.getxmltype(s.code)
    columns
      FileContent_c clob
  ) c
),
fileContent as (
  select FileContent_c from FileUploadRequest
),

```

```

getHeader as (
  select t.SourceOrderNumber OrderNumber, any_value(t.Action) Action,
  JSON_OBJECT(
    'SourceTransactionNumber' value t.SourceOrderNumber,
    'TransactionalCurrencyCode' value any_value(t.CurrencyCode),
    'SourceTransactionId' value any_value(id.header_id+rownum),
    'SourceTransactionSystem' value 'OPS',
    'BusinessUnitName' value any_value(t.BusinessUnit),
    'RequestingBusinessUnitId' value any_value(ou.organization_id),
    'BuyingPartyName' value any_value(t.Customer),
    'BuyingPartyNumber' value any_value(t.CustomeNumber),
    'SubmittedFlag' value decode(any_value(t.SOstatus), 'Draft', 'false', 'true'),
    'DemandClass' value any_value(t.DemandClass),
    'billToCustomer' value
    JSON_ARRAY(
      JSON_OBJECT(
        'PartyName' value any_value(t.BilltoCustomer),
        'AccountNumber' value any_value(t.Billtoaccount)
      ) returning clob
    ) returning clob
  ),
  'shipToCustomer' value
  JSON_ARRAY(
    JSON_OBJECT(
      'PartyName' value any_value(t.ShiptoCustomer),
      'Address1' value any_value(t.Shiptoaddress)
    ) returning clob
  ) returning clob
),
'TransactionType' value any_value(t.OrderType),
'SalesChannel' value any_value(t.SalesChannel),
'Salesperson' value any_value(t.PrimarySalesPerson),
'CustomerPONumber' value any_value(t.PurchaseOrder),
'additionalInformation' value
  JSON_ARRAY(
    JSON_OBJECT(
      'Category' value 'DOO_HEADERS_ADD_INFO',
      'HeaderEffBCustomer_NotesprivateV0' value
        JSON_ARRAY(
          JSON_OBJECT(
            'ContextCode' value 'Customer Notes',
            'deliveryComments1' value any_value(t.DeliveryComments1),
            'deliveryComments2' value any_value(t.DeliveryComments2)
          ) returning clob
        )
    )
  )

```

1. Covered Design Patterns for Custom Business Processes in FA – BI Publisher as Data Fetch and UI Templating Tool – Cont'd

- Utilization of layout templates:
 - Documents generation (PDF, HTML, Excel)
(access to Excel/HTML can be done by Scheduling Service SOAP by BIP Job ID)
Example can be found in the [blog article](#)
 - XSL-TEXT + eText for structured text files (integrations, label printing, etc.)
- Bursting options
 - Email (use carefully after P2T)
 - SFTP with/without encryption
 - UCM – can be combined with OIC processing
(Run a report and attach its output to every order as an attachment)

2. Covered Design Patterns for Custom Business Processes in FA – ESS Job submitting Synchronous Processing in OIC

- Triggered by a dedicated connection in BIP data model
 - Utilize a generic HTTP connection to OIC (one place for P2T change)
 - Possibility of creating a SOAP connection to OIC (additional P2T change)
- Ability to map report parameters to OIC integration (can not be done from query)
- Ability to map **fusionapps_request_id** built in parameter passing current ESS Job Id
- Design a synchronous OIC integration (GET, POST with response, request/response SOAP)
- Pros
 - ESS appears running as long as OIC process is running
 - Resulting XML from HTTP/SOAP connection can be used in BIP template
 - Output/Errors are exposable in ESS's output
- Cons
 - 5 mins limit on sync OIC call
 - Not suitable for large volume processing
 - Last Update user it OIC service user

2. Covered Design Patterns for Custom Business Processes in FA – ESS Job submitting Synchronous Processing in OIC – Cont'd

Administration Administration > HTTP > Update Data Source: OIC

Search All

Update Data Source: OIC

General

Data Source Name: OIC
Server Protocol: https
* Server: [redacted].ocp.oraclecloud.com
* Port: 443
Session Timeout (Minutes): 10
Realm: [redacted]
Username: [redacted]
Password: [redacted]

Security

Allow Guest Access

Available Roles

- (OBSOLETE) Payment Reversal
- (OBSOLETE) Student Enrollment
- (OBSOLETE) Student Payment
- (OBSOLETE) Student Sponsorship Maintenance
- Absence Entitlement Case Transaction Analysis Duty
- Absence Management Transaction Analysis Duty
- AC Accounts Receivable Inquiry
- Academic Coordinator
- AC BI Administrator
- AC BI Author

Allowed Roles

- [redacted] BI Consumer Role

Move, Move All, Remove, Remove All

2. Covered Design Patterns for Custom Business Processes in FA – ESS Job submitting Synchronous Processing in OIC – Cont'd

Administration > Web Services > Update Data Source: oic-lot-update

Update Data Source: oic-lot-update

General

Data Source Name: oic-lot-update

Server Protocol: https

* Server: [redacted].ocp.oraclecloud.com

* Port: 443

* URL Suffix: /ic/ws/integration/v1/flows/soap/[redacted]
(Example: analytics-ws/saw.dll?wsdl)

Session Timeout (Minutes): 100

Complex Type:

WS-Security: 2004

Authentication Type: HTTP

Username: [redacted]

Password: [redacted]

WSDL protected by HTTP basic auth:

Security

Allow Guest Access:

Available Roles:

- (OBSOLETE) Payment Reversal
- (OBSOLETE) Student Enrollment
- (OBSOLETE) Student Payment
- (OBSOLETE) Student Sponsorship Maintenance
- Absence Entitlement Case Transaction Analysis Duty
- Absence Management Transaction Analysis Duty
- Academic Coordinator
- Accept Incentive Compensation Plan Document
- Access Certification Administrator
- Access Certification Transaction Analysis Duty

Allowed Roles:

Move, Move All, Remove, Remove All

2. Covered Design Patterns for Custom Business Processes in FA – ESS Job submitting Synchronous Processing in OIC – Cont'd

The screenshot displays the Oracle Integration Cloud (OIC) configuration interface. On the left, a navigation pane lists various configuration options: Event Triggers, Flexfields, List of Values, Parameters, OrderNumber, and Bursting. The main workspace shows a configuration element labeled 'G_1' with a message: 'No metadata available for elements'. An 'Edit Data Set' dialog box is open, showing the following configuration:

- Name:** [Redacted]
- Data Source:** OIC
- URL Suffix:** ic/api/integration/v1/flows/rest/[Redacted]
- Method:** GET
- Parameters:** Add Parameter

Name	Value (Parameter)
P_ORDER_N	OrderNumber

Buttons for 'OK' and 'Cancel' are visible at the bottom right of the dialog.

2. Covered Design Patterns for Custom Business Processes in FA – ESS Job submitting Synchronous Processing in OIC – Cont'd

- Small volume/single record processing pattern:
 - Submitting ESS Job
 - Caller BIP with HTTP connection, HTML/PDF output
 - Synchronous OIC returning XML in case of success or failure
 - Online REST APIs/BIP queries if needed

3. Covered Design Patterns for Custom Business Processes in FA – ESS Job submitting Asynchronous Processing in OIC

- ESS Job triggering a synchronous OIC integration by GET
- GET integration can return submitted integration id or an error
- Pass **fusionapps_request_id** – allows current user/email identification
- Design a caller synchronous OIC integration (same as before)
- Design an asynchronous OIC integration (POST without response, one-way SOAP, scheduled)
 - Scheduled integration is a preferred pattern (multiple submission will be placed in a queue)
- Pros:
 - Execution time is up to 6 hours
 - Utilization of full OIC capabilities
- Cons:
 - ESS completes immediately, no indication of progress, errors, feedback (possible to create SQL dataset with **DBMS_LOCK.sleep** to wait until OIC completion)
 - Last Update user is OIC service user
 - Outputs of BIPs executed as part of OIC are not visible by submitter user and should be emailed or attached to accessible business record

3. Covered Design Patterns for Custom Business Processes in FA – ESS Job submitting Asynchronous Processing in OIC – Cont'd

- Example of an OIC utility integration fetching user's email for `fusionapps_request_id`
- Utilize the following endpoints:
 - `/ess/rest/scheduler/v1/requests/{requestId}/`
 - `/hcmRestApi/scim/Users`

The screenshot displays the Oracle Integration Cloud (OIC) interface. On the right, the 'View Request Parameters' configuration window is open, showing the following details:

- Operation Name: default
- Resource URI: /calc
- HTTP Method: GET

Below these fields, the 'Specify Query Parameters' section contains a table with the following data:

Name	Data Type
requestId	string

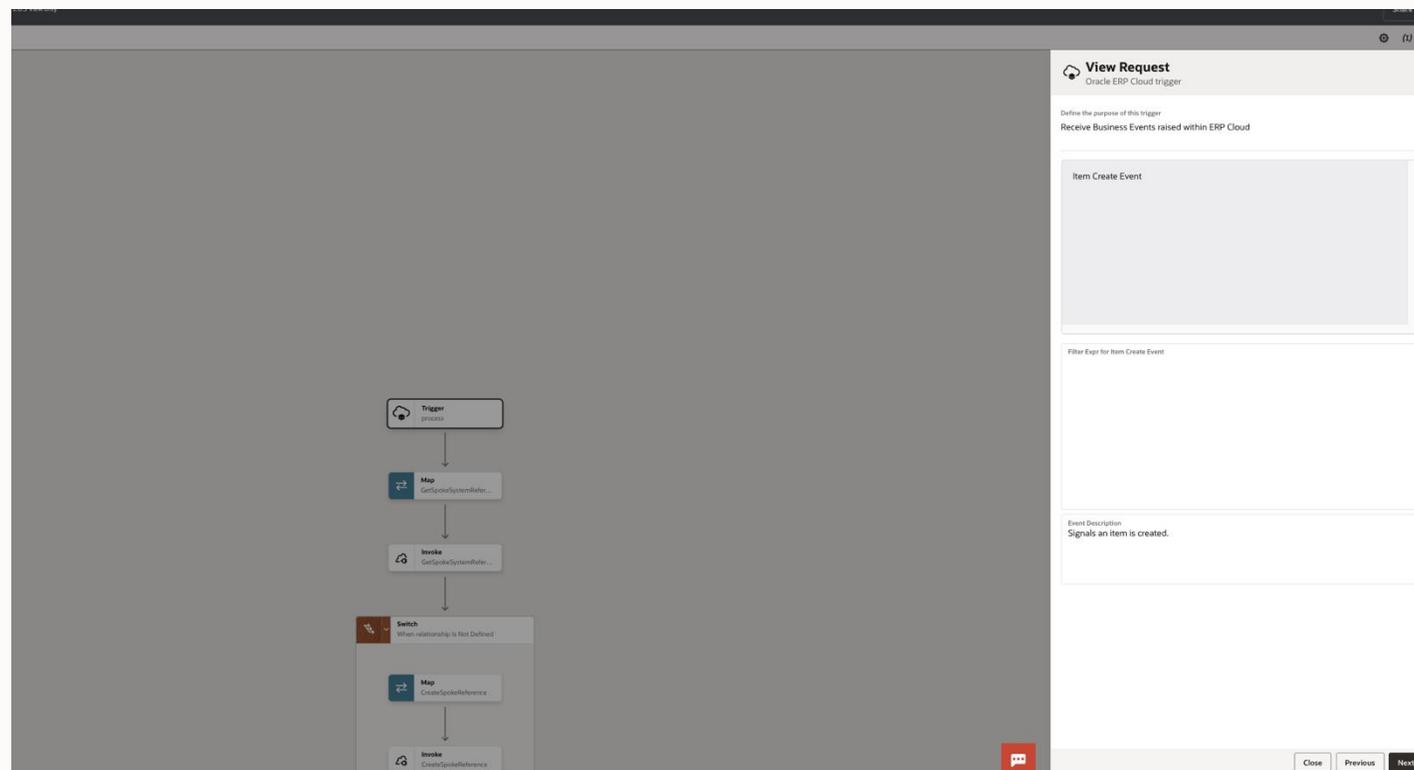
On the left, a process flow diagram is visible, showing a sequence of steps: Trigger (GetUserDetails) → Logger (RequestIDLogger) → Map (GetESSDetails) → Invoke (GetESSDetails) → Logger (ESSJobNameLogger) → Map (GetUserScimDetails) → Invoke (GetUserScimDetails) → Map (GetUserDetails).

3. Covered Design Patterns for Custom Business Processes in FA – ESS Job submitting Asynchronous Processing in OIC – Cont'd

- Large volume data manipulation pattern:
 - Submitting ESS Job
 - Caller BIP with HTTP connection
 - Trigger OIC GET integration
 - Scheduled OIC processing integration
 - Data fetch by BIP processed by additional ESS (can be defined not user triggerable)
 - REST API/FBDI/HDL/Import Management processing
 - Securely expose generated data files to users
 - Error handling and user's progress/error notification
- Additional large volume processing example:
 - User uploads to sharepoint an excel file with sales order for import
 - ESS Job is submitted
 - File fetched from sharepoint, converted to JSON and stored in a custom object CLOB field
 - CLOB field is parsed by XMLTABLE/JSON_TABLE, records are processed
 - User notified/result files published to sharepoint

4. Covered Design Patterns for Custom Business Processes in FA – Standard/Custom Integration Event triggering an OIC Integration

- Use OIC to subscribe to pillar specific FA standard business events
- Events examples:
 - ESS/FBDI load completion
 - Item creation/update
 - Work Order change
 - Sales Order status update
- No impact on P2T process
- No SLA for for triggering time
- Event processing can be lost due to password change
- Backup process to sync unprocessed events



4. Covered Design Patterns for Custom Business Processes in FA – Standard/Custom Integration Event triggering an OIC Integration – Cont'd

- Use OIC to subscribe to FA custom business events
- Prerequisites:
 - Set Profile option **ZCX_CUSTOM_OBJECT_EVENTS** to **Yes** on site level
 - Navigate to **Application Composer > Common Setup** (outside of sandbox)
 - Right-click and select **Generate Integration Event**.
- Create/Use custom object in **Application Composer**
- Create an OIC Integration to Listen to CRUD Events

Application Composer

Application: CRM Cloud

Object: OIC Call

Object Tags: Innovation, Sales, Quality

Objects: Custom Objects, OIC Call, Fields, Pages, Actions and Links, Security, Server Scripts, Standard Objects

Fields

Custom Standard

Display Label	Name	Type	Required	Updatable	Description
Owner	OraZcxOwner	Choice List (Dynamic) <R...	✓	✓	Owner of the record

View Request
Oracle ERP-Cloud trigger

Define the purpose of this trigger
Receive Business Events raised within ERP-Cloud

OICCall_c_Standard_Create

Filter Expr for OICCall_c_Standard_Create

Event Description
This custom object event signals for OICCall_c is created

Close Previous Next

5. Covered Design Patterns for Custom Business Processes in FA – Groovy based ESS Job

- Expose as regular ESS
- Implement logic inside FA without need for OIC
- Useful for HCM implementation without OIC
- Limited to 30 mins execution time (more than triggers, workflows)
- Implementation in Groovy by using:
 - Custom Object Object Function
 - Based on official [documentation](#) the parameters for Scheduled Groovy are not supported
 - Identify ESS parameters by current session available in Groovy
 - BIP Reports for data fetch
 - Data manipulation by using Web Services defined against P2T safe URL <https://fa-internal.oracleoutsourcing.com:10663>
 - Text output for results and error display
- Submission user is used for DML, proper license and access required
- Real life example can be found in the [blog article](#)

5. Covered Design Patterns for Custom Business Processes in FA – Groovy based ESS Job – Cont'd

- Create a custom ESS job definition by cloning the standard **Schedule Custom Groovy Object Functions** job

Manage Enterprise Scheduler Job Definitions and Job Sets for Customer Relationship Management and Related Applications ⓘ

Manage Job Definitions Manage List of Values Sources Manage Job Sets

Done

Save and Close Cancel

Edit Job Definition

Job Definition Show More

Display Name Schedule Custom Groovy Object Functions

Name ExtnJob

Path /oracle/apps/ess/etrv/

Application CRM Application Composer

Description Schedules and executes long-running custom Groovy object functions callable by external systems at the customer-desired time and recurrence.

Retries

Job Category

Timeout Period

Job Application Name CRMEss

Job Type JavaJobType

Class Name oracle.apps.ess.program.ExtnJobExecutor

Default Output Format

Report ID

Priority

Allow Multiple Pending Submissions True

Enable submission from Scheduled Processes

ExtnJob: Parameters User Properties

Actions View ▾ Detach

Parameter Prompt	Data Type	Page Element	Default Value	Read Only	Required
Object Name	String	Text box	---		✓ Required
Object Function	String	Text box	---		✓ Required

5. Covered Design Patterns for Custom Business Processes in FA – Groovy based ESS Job – Cont'd

- Add parameter fields **after** the two required parameters:

Manage Enterprise Scheduler Job Definitions and Job Sets for Customer Relationship Management and Related Applications ?

Manage Job Definitions | Manage List of Values Sources | Manage Job Sets

Done

Save and Close | Cancel

Create Job Definition

Job Definition [Show More](#)

Display Name: Custom ESS Job 1

Name: CustomEssJob1

Path: /custom/

Application: CRM Application Comp

Description: Schedules and executes long-running custom Groovy object functions callable by external systems at the customer-desired time and recurrence.

Retries: 5

Job Category:

Timeout Period:

Job Application Name: CrmEss

Enable submission from Enterprise Manager

Job Type: JavaJobType

Class Name: oracle.apps.extn.ess.program.ExtnJobExecutor

Default Output Format: TXT

Report ID:

Priority:

Allow Multiple Pending Submissions: True

Enable submission from Scheduled Processes

CustomEssJob1: Parameters | User Properties

Actions | View | + | - | Copy from Existing Job Definition | Details

Parameter Prompt	Data Type	Page Element	Default Value	Read Only	Required
Object Name	String	Text box	CustomESS_c	—	✓ Required
Object Function	String	Text box	Job1	—	✓ Required
Param 1	String	Text box	Test Param 1	—	— Required
Param 2	String	Text box	Test Param 2	—	— Required

5. Covered Design Patterns for Custom Business Processes in FA – Groovy based ESS Job – Cont'd

- Use Groovy code to retrieve the current **sessionId**
- Build a BIP report that queries **FND_SESSION_ATTRIBUTES** and **ESS_REQUEST_PROPERTY** to fetch passed parameters
- Execute the report from Groovy and parse the parameters for use in logic

Application Composer

Application: CRM Cloud

Custom

Object Tags: Innovation Sales Quality

Objects

- Custom Objects
 - Custom ESS Job
 - Fields
 - Pages
 - Actions and Links
 - Security
 - Server Scripts
- Standard Objects
 - Customer Contact Profile
 - Launch Custom Profile Specificat
 - Launch Customer Profile Specific
- Common Setup
 - CX Extension Generator
 - Copy Maps
 - Data Quality Rules
 - Global Functions
 - Mashup Content
 - Metadata Manager

Edit Object Function

Definition

- Function Name: Job1
- Type: Row Level
- Returns: String
- Description:

Example:

Privileged

Visibility: Callable by External Sys

Parameters

Function Body

Edit Script

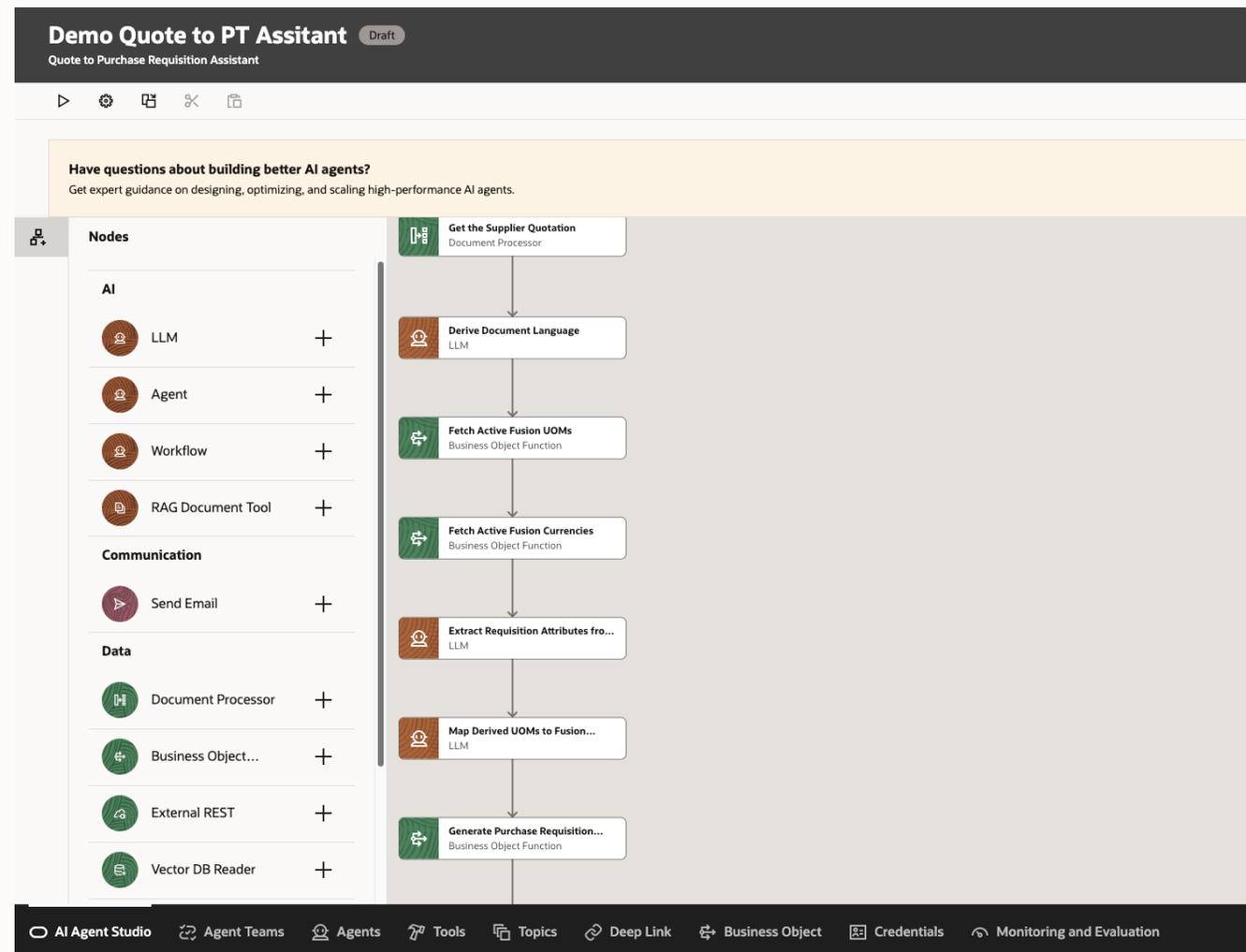
```

1 def output = new StringBuilder()
2 output.append("\n")
3 def sessionId=adf.util.getSession().toString().split('/')[0]
4 output.append("SessionID=" + sessionId + "\n")
5 def json = adf.util.runSqlToJson("""select fsa.attribute_value requestid, erp.name param_name, erp.value param_value
6 from fnd_session_attributes fsa, ess_request_property erp
7 where fsa.session_id='${sessionId}'
8 and fsa.attribute_name='JOB_REQUEST_ID' and to_number(fsa.attribute_value) = erp.requestid
9 and erp.name like 'submit.argument%' order by erp.name""")
10 def res = (List)adf.util.parseJson(json)
11 output.append("Param1=" + res[2]["PARAM_VALUE"] + "\n")
12 output.append("Param2=" + res[3]["PARAM_VALUE"] + "\n")
13 return output.toString()

```

6. Covered Design Patterns for Custom Business Processes in FA – New: AI Agent Studio

- Start using AI Agent Studio
- Utilize and learn from pre-existing templates
- Use the power of LLM for user interaction, unstructured files processing
- Enable in non-prod to start exploration
- Start user adoption with out-of-the-box agents
- Proceed with custom Tools for advanced processes



6. Covered Design Patterns for Custom Business Processes in FA – New: AI Agent Studio – Example

- Business Scenario:
 - User receives multiple formats of product usages from customers
 - Files could be Excel, CSV, PDF
 - Column names could be different
 - The column's order and location is not the same
 - Goal – extract from the files common data and upload it to billing system (exposed by OIC)
- Demo Solution:
 - OIC Integration receiving JSON with single usage payload and returns register status
 - AI Agent Studio – Custom Tool of External URL pointing to OIC integration is defined
 - AI Agent Studio – Custom Agent using pre-built MultiFileProcessor tool, prompt defined with the parsing logic
 - AI Agent Studio – Custom Agent Team of type Workflow using the above Custom Agent, record loops and External URL calls

6. Covered Design Patterns for Custom Business Processes in FA – New: AI Agent Studio – Example – Cont'd

The screenshot displays the Oracle ACE AI Agent Studio interface. At the top, the header shows 'Product: ITEM2 | AcmU5xGIEfGBYPFnHIEB2g' and 'AI Utils'. The instance is 'Process Usage (01.00.0000)' with a tracing level of 'Debug'. The main workspace shows a process flow diagram with two steps: 'Trigger Start' and 'Map Start'. The 'Map Start' step is highlighted with a green background. To the right, the 'Activity stream' panel shows a sequence of events: 'Wire Message received by Trigger Start' (with a JSON payload), 'Message started processing', 'Message received by Data Mapper', 'Data Mapping completed with Message', 'Message sent to reply', and 'Reply to Start completed with Wire Message' (with a JSON payload). The JSON payloads are:

```
{ "Product": "ITEM2", "Quantity": 200, "UsageDate": "2020-01-02" }
```

 and

```
{ "status": "success", "message": "Registered usage of ITEM2 on 2020-01-02 with quantity 200" }
```

6. Covered Design Patterns for Custom Business Processes in FA – New: AI Agent Studio – Example – Cont'd

Custom Usage R

Have questions about b
Get expert guidance on desig

Tool Type
External REST

Tool Code
CUSTOM_USAGE_PROCES

Product
Subscription Management

Description
Custom Single Usage Proc

Require human approval

Enable if you want human approva

Authorization Function

+ Add

Name
Process_Usage

AI Agent Studio

Name
Process_Usage
Only numbers, alphabets, _ and - are allowed

Operation Type
HTTP POST

Resource Path
/ic/api/integration/v2/flows/rest/project/AI_UTILS/PROCESS_USAGE/1.0/process

Description
Process single product usage

Body Template

```
{  
  "Product": "{product}",  
  "Quantity": {quantity},  
  "UsageDate": "{usage_date}"  
}
```

Parameters Sample Queries Headers

+ Add

Name	Data Type	
usage_date	String	
Usage date		
quantity	Number	
Quantity		
product	String	

Cancel Update

6. Covered Design Patterns for Custom Business Processes in FA – New: AI Agent Studio – Example – Cont'd

The screenshot shows the AI Agent Studio interface for a custom agent named "Custom Usage File Processor". The interface includes a header with the agent name and "Custom Agent" label. Below the header, there is a section titled "Have questions about building better AI agents?" with a link to get expert guidance. The main workspace displays a card for the "Custom Usage File Processor" agent, which is currently inactive. The card shows the agent's name, a description "Process different usage file formats an...", and a list of tools: "Tools(1)" containing "MultiFileProcessor" (with a description "This tool type will be used to proce...") and "Topics(0)". The bottom navigation bar includes icons for AI Agent Studio, Agent Teams, Agents, Tools, Topics, Deep Link, Business Object, Credentials, and Monitoring and Evaluation.

The screenshot shows the configuration interface for the "Custom Usage File Processor" agent, labeled as a "Worker Agent". The main area contains a "Prompt" field with the following text:

You are the Document Processor Agent for product usage data.

When files are uploaded:

- Call MultiFileProcessor to process all files (tables from XLSX/CSV, text from PDF/DOCX).

User request: `{{context.$system.$inputMessage}}`

Responsibilities:

1. Process ALL uploaded files.
2. For each data row (skip headers):
 - SrcFileName = source file name
 - SrcFileRec# = row number in source file (1-based, skip header)
 - Product = value from Item/Product column (best match)
 - UsageDate = normalize to YYYY-MM-DD (assume DD/MM/YYYY unless impossible)
 - Quantity = number from Qty/Quantity column
3. Ignore all other columns.
4. Build array of objects with exactly the keys above.
5. Return STRICTLY JSON matching the output schema — nothing else.

Date normalization rules:

- Prefer DD/MM/YYYY (Israel/Europe style)
- Accept MM/DD/YYYY, YYYY-MM-DD, DD-MMM-YY, etc.
- If ambiguous (both day and month ≤12) → use DD/MM/YYYY
- If parse fails → keep original string + note in summary

Example output:

```
{
  "records": [
    {
      "SrcFileName": "usage.xlsx",
      "SrcFileRec#": 1,
      "Product": "ITEM1",
      "UsageDate": "2020-01-01",
      "Quantity": 1
    },
    ...
  ],
  "rowCount": 14,
  "processedFiles": ["usage.xlsx"],
  "summary": "14 rows extracted from 1 file, all dates normalized",
  "error": ""
}
```

If no file or error:

- "records": []
- "rowCount": 0
- "error": "short message"

At the bottom, there is a note: "Type {{ to start seeing context-sensitive expressions." and buttons for "Cancel" and "Update".

6. Covered Design Patterns for Custom Business Processes in FA – New: AI Agent Studio – Example – Cont'd

Custom Usage Processor Published Cancel Save and Close Publish

Have questions about building better AI agents?
Get expert guidance on designing, optimizing, and scaling high-performance AI agents. Ask Oracle Dismiss

Nodes

- AI**
 - LLM +
 - Agent +
 - Workflow +
 - RAG Document Tool +
- Communication**
 - Send Email +
- Data**
 - Document Processor +
 - Business Object... +
 - External REST +
 - Vector DB Reader +
 - Vector DB Writer +
- Logic**

```
graph TD; Start((▶)) --> FileProcessor[File Processor Agent]; FileProcessor --> Loop[Process Records Loop]; subgraph Loop; direction TB; LoopStart((▶)) --> ProcessUsage[Process Usage External REST]; ProcessUsage --> LoopEnd[◻]; end; LoopEnd --> Final[◻];
```

6. Covered Design Patterns for Custom Business Processes in FA – New: AI Agent Studio – Example – Cont'd

The screenshot displays the Oracle AI Agent Studio interface for a 'Custom Usage Processor' in a 'Draft' state. The interface is divided into several sections:

- Nodes Panel (Left):** A sidebar containing various components categorized into AI, Communication, Data, and Logic. The AI category includes LLM, Agent, Workflow, and RAG Document Tool. Communication includes Send Email. Data includes Document Processor, Business Object..., External REST, Vector DB Reader, and Vector DB Writer. Logic is currently empty.
- Workflow Diagram (Center):** A vertical flowchart starting with a play button icon, followed by a 'File Processor Agent' node, then a 'Process Records Loop' node, and ending with a square stop icon.
- File Processor (Right):** A large, empty white area intended for the configuration of the selected 'File Processor Agent' node.
- Bottom Panel:** A navigation bar with icons for AI Agent Studio, Agent Teams, Agents, Tools, Topics, Deep Link, Business Object, Credentials, and Monitoring and Evaluation. Below this, a file explorer shows three files: 'TestData.xlsx' (9.19 KB), 'TestData2.pdf' (11.55 KB), and 'res.csv' (70 bytes).

6. Covered Design Patterns for Custom Business Processes in FA – New: AI Agent Studio – Example – Cont'd

Custom Usage Processor Draft

Nodes

- AI**
 - LLM +
 - Agent +
 - Workflow +
 - RAG Document Tool +
- Communication**
 - Send Email +
- Data**
 - Document Processor +
 - Business Object... +
 - External REST +
 - Vector DB Reader +
 - Vector DB Writer +
- Logic**

Flowchart:

```
graph TD; Start(( )) --> FileProcessor[File Processor Agent]; FileProcessor --> ProcessRecords[Process Records Loop]; ProcessRecords --> End(( ));
```

AI Agent Studio | Agent Teams | Agents | Tools | Topics | Deep Link | Business Object | Credentials | Monitoring and Evaluation

```
{\n  "status": "success",\n  "message": "Registered usage of ITEM5 on 2020-01-05 with quantity 5",\n  "status": "success",\n  "message": "Registered usage of ITEM6 on 2020-01-06 with quantity 6",\n  "status": "success",\n  "message": "Registered usage of ITEM7 on 2020-01-07 with quantity 7",\n  "status": "success",\n  "message": "Registered usage of ITEM8 on 2020-01-08 with quantity 8",\n  "status": "success",\n  "message": "Registered usage of ITEM9 on 2020-01-09 with quantity 9",\n  "status": "success",\n  "message": "Registered usage of ITEM10 on 2020-01-10 with quantity 10",\n  "status": "success",\n  "message": "Registered usage of ITEM11 on 2020-01-11 with quantity 11",\n  "status": "success",\n  "message": "Registered usage of ITEM12 on 2020-01-12 with quantity 12",\n  "status": "success",\n  "message": "Registered usage of ITEM13 on 2020-01-13 with quantity 13",\n  "status": "success",\n  "message": "Registered usage of ITEM14 on 2020-01-14 with quantity 14"}]
```

Context:

Context



```
[{\n  "status": "success",\n  "message": "Registered usage of ITEM1 on 2020-01-01 with quantity 133",\n  "status": "success",\n  "message": "Registered usage of ITEM2 on 2020-01-02 with quantity 200",\n  "status": "success",\n  "message": "Registered usage of ITEM1 on 2020-01-01 with quantity 1",\n  "status": "success",\n  "message": "Registered usage of ITEM2 on 2020-01-02 with quantity 2",\n  "status": "success",\n  "message": "Registered usage of ITEM3 on 2020-01-03 with quantity 3",\n  "status": "success",\n  "message": "Registered usage of ITEM4 on 2020-01-04 with quantity 4",\n  "status": "success",\n  "message": "Registered usage of ITEM5 on 2020-01-05 with quantity 5",\n  "status": "success",\n  "message": "Registered usage of ITEM6 on 2020-01-06 with quantity 6",\n  "status": "success",\n  "message": "Registered usage of ITEM7 on 2020-01-07 with quantity 7",\n  "status": "success",\n  "message": "Registered usage of ITEM8 on 2020-01-08 with quantity 8",\n  "status": "success",\n  "message": "Registered usage of ITEM9 on 2020-01-09 with quantity 9",\n  "status": "success",\n  "message": "Registered usage of ITEM10 on 2020-01-10 with quantity 10",\n  "status": "success",\n  "message": "Registered usage of ITEM11 on 2020-01-11 with quantity 11",\n  "status": "success",\n  "message": "Registered usage of ITEM12 on 2020-01-12 with quantity 12",\n  "status": "success",\n  "message": "Registered usage of ITEM13 on 2020-01-13 with quantity 13",\n  "status": "success",\n  "message": "Registered usage of ITEM14 on 2020-01-14 with quantity 14",\n  "status": "success",\n  "message": "Registered usage of ITEM1 on 2020-01-01 with quantity 1",\n  "status": "success",\n  "message": "Registered usage of ITEM2 on 2020-01-02 with quantity 2",\n  "status": "success",\n  "message": "Registered usage of ITEM3 on 2020-01-03 with quantity 3",\n  "status": "success",\n  "message": "Registered usage of ITEM4 on 2020-01-04 with quantity 4",\n  "status": "success",\n  "message": "Registered usage of ITEM5 on 2020-01-05 with quantity 5",\n  "status": "success",\n  "message": "Registered usage of ITEM6 on 2020-01-06 with quantity 6",\n  "status": "success",\n  "message": "Registered usage of ITEM7 on 2020-01-07 with quantity 7",\n  "status": "success",\n  "message": "Registered usage of ITEM8 on 2020-01-08 with quantity 8",\n  "status": "success",\n  "message": "Registered usage of ITEM9 on 2020-01-09 with quantity 9",\n  "status": "success",\n  "message": "Registered usage of ITEM10 on 2020-01-10 with quantity 10",\n  "status": "success",\n  "message": "Registered usage of ITEM11 on 2020-01-11 with quantity 11",\n  "status": "success",\n  "message": "Registered usage of ITEM12 on 2020-01-12 with quantity 12",\n  "status": "success",\n  "message": "Registered usage of ITEM13 on 2020-01-13 with quantity 13",\n  "status": "success",\n  "message": "Registered usage of ITEM14 on 2020-01-14 with quantity 14"}]
```

Ask Oracle



Design Pattern Selection

Value driven, efficient and supportable solutions

Design Pattern Selection – Checklist

- Define Best match for end user submission UI based on business requirements
- Define security/licensing model
 - Access for submission (screen, ESS, etc.)
 - Permissions for performing changes (does submitter have access to data manipulation API?)
 - Ability to track submission user as last updater/transaction creator
- Define expected response time and volume (to pick sync/async mode)
- Define error handling and user progress notification approach
- Define final/intermediate artifacts delivery approach (OIC email, OWA email, sharepoint, UCM/attachments)
- Define P2T actions if needed

Q & A

Real world use cases, implementation ideas

Thank You

Alex Rabinovich

CID SOFTWARE SOLUTIONS LTD

Alex.Rabinovich@cidsolutions.co.il

<https://www.linkedin.com/in/alex-rabinovich-cid/>

<https://cidsolutions.co.il/blog>

